

IEEEESTEC

14th 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



Министарство просвете,
науке и технолошког развоја



ISBN: 978-86-6125-242-6

IEEESTEC Student Projects Conference, November 25, 2021, Niš, Serbia
Proceedings 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:

Dušan Vučković
Miloš Marjanović

Editors of Proceedings:

Vera Marković
Danijel Danković
Zlatica Marinković

Miloš Marjanović
Biljana Stošić

Secretary:

Jana Vračar
Aleksandra Stojković
Nikola Mitrović
Milan Stojanović
Sandra Veljković
Neda Stanojević
Marija Živković
Milica Veljović
Anđela Kocić
Nina Dimitrijević
Ana Sentov

Stefan Popović
Nemanja Nedeljković
Anastasija Avramović
Dušan Jovanović
Nikola Stojiljković
Aleksandra Trajković
Strahinja Dimitrijević
Mikica Radenković
Sava Stojanović
Katarina Božinović
Ilija Đorđević Arsić

Technical Editors:

Miloš Marjanović
Jana Vračar
Danijel Danković
Emilija Živanović

CIP - Каталогизacija у публикацији - Народна библиотека Србије, Београд

621.3(082)
004(082)

IEEESTEC Student Project Conference (14 ; 2021 ; Niš)

IEEESTEC 14th Student Project Conference, [Niš, 2021] / organized by
EESTEC LC, Niš ... [et al.] ; [editors Vera Marković ... [et al.]]. - Niš :
Faculty of Electronic Engineering, 2021 (Niš : Unigraf X copy). - 286 str.
: ilustr. ; 30 cm

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

ISBN 978-86-6125-242-6

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

a) Електротехника - Зборници b) Рачунарство - Зборници

COBISS.SR-ID 51185673

Printed by: Unigraf X copy Niš, 2021.

Circulation: 100 copies

Reviewers:

1. Mirjana Perić
2. Aleksandar Milenković
3. Dragana Jovanović
4. Vladimir Sibinović
5. Vladimir Mitić
6. Marko Dimitrijević
7. Dejan Mirković
8. Miloš Bogdanović
9. Vladimir Čirić
10. Jana Vračar
11. Miloš Marjanović
12. Sandra Veljković
13. Nikola Mitrović
14. Milan Stojanović
15. Aleksandra Stojković
16. Filip Filipović
17. Tatjana Nikolić
18. Anđelija Đorđević
19. Nađa Gavrilović
20. Nenad Petrović
21. Milan Simić
22. Ana Vučković
23. Biljana Stošić
24. Ivica Marković
25. Miljana Milić
26. Emilija Živanović
27. Marija Živković
28. Neda Stanojević
29. Vojkan Davidović
30. Vesna Paunović
31. Ivan Anastasijević
32. Goran Nikolić
33. Ivana Kostić
34. Jelena Jovanović
35. Bojan Banković
36. Aneta Prijić
37. Danijel Danković

Table of contents:

Merenje nepoznatih otpornosti i kapacitivnosti primenom Arduino platforme	1
Maglev vozovi-pregled dosadašnjih istraživanja	5
Simulation of One-Week Production for a Photovoltaic System and How Does It Affect the Distribution Grid	9
Sigurnosni IoT sistem sa električnom bravom	13
Sistem automatskog proveravanja i održavanja mera protiv COVID-a 19	17
Realizacija sistema za kontrolu servo motora korišćenjem Nucleo razvojnog okruženja	21
Projektovanje i realizacija sistema za praćenje kvaliteta vode	25
Realizacija kapacitivnog senzora za merenje vlažnosti zemljišta	29
Uređaj za kontrolu budnosti vozača Stay Awake	33
Merenje induktivnosti kalema pomoću Arduino mikrokontrolera	37
Konstrukcija i primena ultrazvučnog senzora za merenje daljine	41
Proširenje SMTP protokola za automatsku end-to-end enkripciju imejl poruka	45
Rešavanje optimizacionih problema primenom genetičkih algoritama – primer putujućeg trgovca	49
Pametni šor	55
Sistem za testiranje RGB matičnih ploča	61
LED Driver Basics and its Most Important Characteristics	65
Merenje provodnosti metala primenom vrtložnih struja	69
Realization of Training for Remote Control of Drones	73
Primena cirkularnog bafera za prijem podataka korišćenjem UART komunikacije i DMA pristupa	77
Mašinsko učenje kao pomoć kod prognoze kliničkog ishoda nakon moždanog udara	81
Projektovanje uređaja za snimanje i reprodukciju okretanja oko jedne ose	85
Praćenje rada i zaštita računarskih mreža pSense sistemom	89
ARM mikrokontrolerski sistem regulacije brzine DC motora zavisne od temperature okoline	93
Praktična realizacija jednobitne ALU	97
Tester UV zaštite naočara	103
Projektovanje razvojnog okruženja za primene operacionih pojačavača	107
Softver za generisanje NBS IPS QR kôdova	111
Induktivnost i jačina magnetnog polja višeslojnih cilindričnih kalemova	115
„UniMer VIR“ - Merna stanica sa osciloskopom, frekvencijskim generatorom i naponskim izvorom	119
Upotreba 12-bitnog digitalno-analognog konvertora MCP4725 u embedded uređajima	123
Sistem zaključavanja korišćenjem daljinskog upravljača realizovan u Arduino okruženju	127
Softverska kontrola SMU Keithley serije 2400 za karakterizaciju PMOS tranzistora IRF9520	131
Mobilna android aplikacija za uspostavljanje kontakta između majstora i poslodavca ...	135
Primena Markovljevih lanaca na prikaz podataka sa meteoroloških stanica o vremenu korišćenjem R jezika	139
Proračun tokova snaga u distributivnim mrežama	143
Model-Driven Approach to VHDL-based Logic Design	147
Termometar sa PIC16F1847 - uređaj primenljiv u svim sferama života	151
Upoređivanje efikasnosti algoritama za rešavanje Sudoku-a	155
Kreiranje verifikacionog okruženja za Split Transaction Open Core Protocol po UVM metodologiji	159
Realizacija kola sa optokaplerima	163
Digitalna platforma “Virtual MIEL”	167
Proračun optimalnih tokova aktivnih snaga primenom linearnog programiranja	171
Realizacija OLED i LCD1602 displeja na Nucleo razvojnom okruženju	175
Projektovanje tajmera korišćenjem PIC mikrokontrolera	181
Simulacija samostalnog fotonaponskog sistema	185
Termoelektrični modul kao izvor napajanja u čvorovima bežičnih senzorskih mreža - pregled	189
Task Prompter	195
Grafen aluminijum-jonske baterije	197
LIMENKO-solarni kolektor od iskorišćenih limenki	201
Sistem za kontrolu brzine DC motora korišćenjem ultrazvučnog senzora i ARM mikrokontrolera	205
Booking Cancellation Prediction Relying on Graph Database in Neo4j	209
Primena proširene realnosti u mobilnim aplikacijama	213
Projektovanje termoregulacionog sistema za pametne kuće	215
Projektovanje i realizacija podizača napona na štampanoj ploči	219
Realizacija laserskog sigurnosnog sistema upotrebom Arduino Uno razvojnog okruženja	225
Odziv VDMOS tranzistora snage na NBT naprezanje, γ -zračenje i odžarivanje	229
Android aplikacija Newsy – društvena mreža za diskutovanje novosti	233
Razvoj aplikacije za optimizaciju položaja pokretnih solarnih panela	237
Ultrazvučni anemometar	241

Projektovanje sistema za kontrolu dnevne svetlosti prostorije u pametnim kućama.....	245
Prototip sistema za otključavanje brave korišćenjem RFID tehnologije.....	249
Projektovanje štampane ploče pametne vage.....	253
Projektovanje uređaja za merenje nivoa glukoze u krvi.....	257
EMC Compliance Testing LED Luminaire in Frequency Bandwidth 9 kHz – 30 MHz.....	261
Elektronski sistem pametne kuće.....	265
Različite realizacije binarnog sabirača.....	269
Realizacija semafora sa dnevnim i noćnim režimom rada korišćenjem Arduina.....	273
Projektovanje pametne raskrsnice.....	277
Pokretanje motora ventilatora na određenoj temperaturi uz pomoć Arduino okruženja i DHT11 senzora.....	279
IEEESTEC na-S-paja.....	283

Merenje nepoznatih otpornosti i kapacitivnosti primenom Arduino platforme

Isidora Jovanović, Natalija Ivković

Sadržaj – U ovom radu opisana je metoda merenja nepoznatih otpornosti i kapacitivnosti korišćenjem Arduino platforme. Merene vrednosti otpornosti i kapacitivnosti za različite otpornike i kondenzatore upoređene su sa nominalnim vrednostima tih elemenata i prikazane tabelarno.

I. UVOD

Laboratorijske vežbe iz Osnova elektrotehnike na Elektronskom fakultetu u Nišu počele su da se izvode pre više od šezdeset godina [1]. U jednom ovako dugom vremenskom periodu dolazilo je do njihovih izmena, osavremenjavanja i uvođenja novih vežbi [2, 3].

U poslednje dve decenije razvijeni su i mnogi programski paketi namenjeni simulaciji rada električnih kola. Samim tim, uvođenjem novih nastavnih planova i programa, uvedena je i primena računara u izvođenje laboratorijskih vežbi iz elektrotehnike. Međutim, nikako se ne sme zanemariti značaj realizacije vežbi na maketama. Studentima je veoma bitno upoznavanje sa osnovnim elementima električnih kola, proveru zakona i merenje fizičkih veličina u realnim uslovima. To je jedino moguće ostvariti kroz direktan kontakt studenata sa maketom i odgovarajućim instrumentima.

Sa druge strane, poslednjih godina prisutna je sve veća upotreba mikrokontrolera, mikroracunara i programabilnih elektronskih komponenti, koja zahteva poznavanje programiranja. Programiranje mikrokontrolera predstavlja složen proces, ali se upotrebom Arduino platforme značajno pojednostavljuje. Arduino je platforma otvorenog koda (*open source*), koja sadrži skup jednostavnih i fleksibilnih komponenti, koje se mogu povezivati u složenije celine. Ova platforma je jednostavna za rad, što je čini pogodnim nastavnim sredstvom u procesu edukacije budućih inženjera. Činjenica je da su laboratorijske vežbe nezamenljive u spajanju teorije i praktične realizacije, pa nije redak slučaj da se Arduino platforma na ovaj način izučava na različitim tehničkim fakultetima širom sveta.

Arduino platforma se može koristiti za merenje mnogih električnih i neelektričnih veličina. U ovom radu biće opisan način merenja električne otpornosti i kapacitivnosti. Realizovana je platforma koja se može iskoristiti za izvođenje laboratorijskih vežbi iz Osnova elektrotehnike i Elektromagnetike.

I. Jovanović i N. Ivković su studenti četvrte godine modula Upravljanje sistemima na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: isidora@elfak.rs, natalija.ivkovic@elfak.rs

II. MERENJE OTPORNOSTI POMOĆU ARDUINO PLATFORME

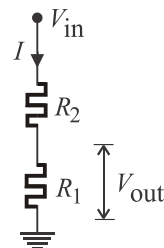
Električna otpornost je jedna od najvažnijih električnih veličina i predstavlja stepen otpora nekog materijala prolasku električne struje. Standardni načini za merenje otpornosti su pomoću instrumenata, kao što su ommetri i multimetri ili pomoću otporničkih mostova, od kojih je najpoznatiji Vitstonov most [4], slika 1.



Slika 1. Instrumenti za merenje otpornosti.

A. Princip rada

Na slici 2 je prikazan princip određivanja nepoznate otpornosti pomoću naponskog razdelnika.



Slika 2. Naponski razdelnik.

$R_2 = R_x$ je nepoznata otpornost, R_1 je poznata otpornost, I je struja koja protiče kroz razdelnik, a napon napajanja je $V_{in} = 5\text{ V}$.

U naponskom razdelniku, prikazanom na slici 2, vrednost otpornosti R_x računamo merenjem napona na otporniku R_1 . Ukoliko napon na otporniku R_1 označimo sa V_{out} , a sa V_x napon na otporniku R_x , izražen u voltima, tada je:

$$V_x = V_{in} - V_{out} = 5 - V_{out} . \quad (1)$$

U tom slučaju, za struju I važi:

$$I = \frac{V_{out}}{R_1} . \quad (2)$$

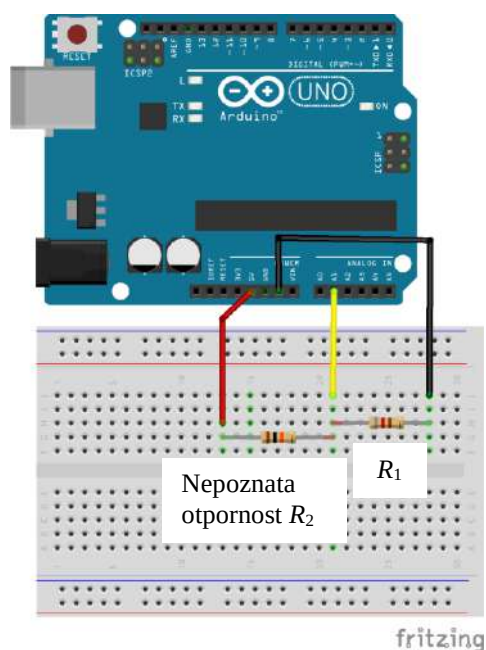
Kako su otpornici R_x i R_1 vezani na red, ista struja protiče kroz njih. Prema tome, nepoznatu otpornost R_x računamo iz relacije:

$$R_x = \frac{V_x}{I} = \frac{V_{in} - V_{out}}{V_{out}} R_1 = \left(\frac{5}{V_{out}} - 1 \right) R_1 . \quad (3)$$

Ovaj izraz predstavlja sastavni deo koda za programiranje Arduina.

B. Šema veze

Fritzing je besplatan program za crtanje šema [5]. Primenom ovog programa, na slici 3 prikazana je realizacija veze za merenje nepoznate otpornosti.

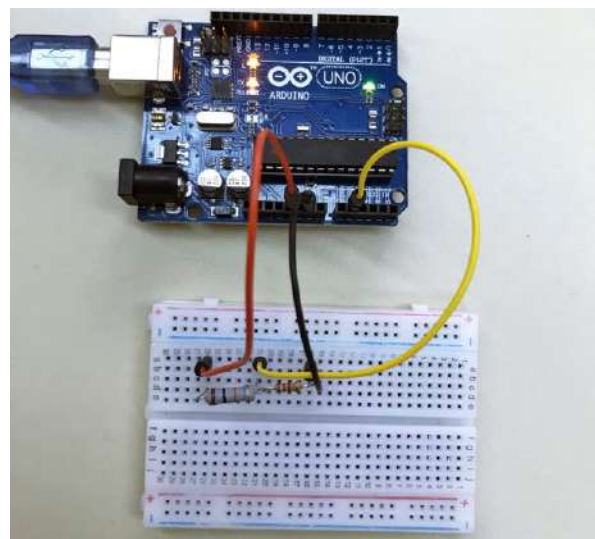


Slika 3. Šema veze nacrtana u programu Fritzing.

C. Praktična realizacija

Na slici 4 prikazana je platforma za merenje nepoznate otpornosti za čiju realizaciju su iskorišćene sledeće komponente [6]: Arduino UNO, protoploča, otpornik poznate otpornosti ($R_1 = 820 \Omega$), otpornik nepoznate otpornosti i konektori.

Rezultati merenja prikazuju se na monitoru računara.



Slika 4. Realizacija platforme za merenje nepoznate otpornosti.

Verifikacija tačnosti realizovane platforme izvršena je u Tabeli I merenjem nekoliko nepoznatih otpornosti, R_{mer} , i poređenjem dobijenih vrednosti sa nominalnim vrednostima tih otpornika, R_{nom} . S obzirom da otpornici imaju i svoju proizvođačku toleranciju, u tabeli je navedena njihova tolerancija δ , kao i vrednosti dobijene merenjem pomoću digitalnog mernog instrumenta, R_{mer1} . Određeno je i odstupanje rezultata primenom izraza

$$\delta_R[\%] = \left| \frac{R_{mer} - R_{nom}}{R_{nom}} \right| \cdot 100 . \quad (4)$$

TABELA I
POREĐENJE MERENJA NEPOZNATE OTPORNOSTI.

Merena vrednost $R_{mer} [\Omega]$	Nominalna vrednost $R_{nom} [\Omega]$	Merena vrednost $R_{mer1} [\Omega]$	$\delta[\%]$	$\delta_R[\%]$
220.50	220	216	± 5	0.23
1854.14	1800	1809	± 5	3.01
3378.40	3300	3270	± 5	2.38
10682.47	10000	10040	± 5	6.82

Može se zaključiti da su izmerene vrednosti otpornika u opsegu njihove tolerancije. Treba imati u vidu da i otpornik poznate vrednosti R_1 ima toleranciju od 5% što takođe utiče na tačnost rezultata merenja dobijenih korišćenjem Arduino platforme, a u obzir nisu uzete ni otpornosti konektora (žice i kontakata).

Osim za merenje pojedinačnih vrednosti otpornika, pomoću realizovane platforme, moguće je približno izmeriti i ekvivalentnu otpornost redne i paralelne veze dva ili više otpornika. I ovde u obzir nije uzeta otpornost kontakata i žica, što će svakako uticati i na tačnost merenja. U Tabelama II i III prikazani su rezultati dobijeni za rednu i paralelnu vezu različitih kombinacija otpornika.

TABELA II
REDNA VEZA DVA OTPORNIKA R_2 I R_3 .

R_2 [Ω]	R_3 [Ω]	Merena vrednost otpornosti R_{mer} [Ω]	Nominalna vrednost otpornosti R_{nom} [Ω]
220	3300	3599.37	3520
10000	10000	21874.06	20000
1800	3300	5220.86	5100
10000	3300	13911.23	13300

TABELA III
PARALELNA VEZA DVA OTPORNIKA R_2 I R_3 .

R_2 [Ω]	R_3 [Ω]	Merena vrednost otpornosti R_{mer} [Ω]	Nominalna vrednost otpornosti R_{nom} [Ω]
220	3300	206.50	206.25
10000	10000	5220.86	5000.00
1800	3300	1193.62	1164.70
10000	3300	2538.72	2481.20

III. MERENJE KAPACITIVNOSTI POMOĆU ARDUINO PLATFORME

Električna kapacitivnost predstavlja odnos naelektrisanja i napona na kondenzatoru [4]. Za merenje ove veličine, pored standardnih mernih uređaja, koriste se i mostovi za naizmeničnu struju [7], slika 5. Mostovi za naizmeničnu struju omogućavaju međusobno poređenje kompleksnih impedansi koje u sebi mogu sadržati kapacitivne i induktivne elemente.



Slika 5. Vinov most za merenje nepoznate kapacitivnosti.

A. Princip rada

Proces tokom koga se vrši opterećivanje, odnosno rasterećivanje kondenzatora u RC kolu, naziva se prelaznim stanjem ili prelaznim režimom [4]. Napon na kondenzatoru se menja po zakonu

$$u_C(t) = E(1 - e^{-t/\tau}), \quad (5)$$

gde je E elektromotorna sila naponskog generatora na koji je RC kolo priključeno, dok je sa

$$\tau = RC \quad (6)$$

obeležena vremenska konstanta izražena u sekundama, R otpornost kola, u omima, a C kapacitivnost kondenzatora u faradima. Vremenska konstanta predstavlja vreme potrebno da napon na kondenzatoru dostigne 63.2% svoje vrednosti, kada je kondenzator napunjen [8].

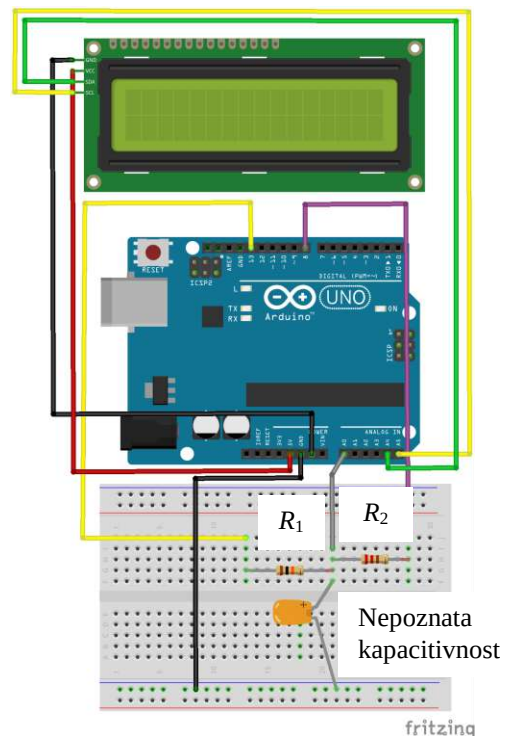
Merenje kapacitivnosti se zasniva na korišćenju RC kola sa poznatom vrednošću otpornosti R i nepoznom vrednošću kapacitivnosti C . Arduino meri napon na kondenzatoru C i beleži vreme potrebno da taj napon dostigne 63.2% vrednosti napona kada je kondenzator potpuno napunjen, što predstavlja vremensku konstantu. Na osnovu izmerene vremenske konstante i poznate vrednosti otpornosti moguće je izračunati vrednost nepoznate kapacitivnosti primenom izraza

$$C = \frac{\tau}{R}, \quad (7)$$

koji predstavlja sastavni deo koda za programiranje Arduina.

B. Šema veze

Za crtanje šeme veze za merenje nepoznatih kapacitivnosti, prikazane na slici 6, i ovde se koristi program Fritzing [5].



Slika 6. Šema veze nacrtana u programu Fritzing.

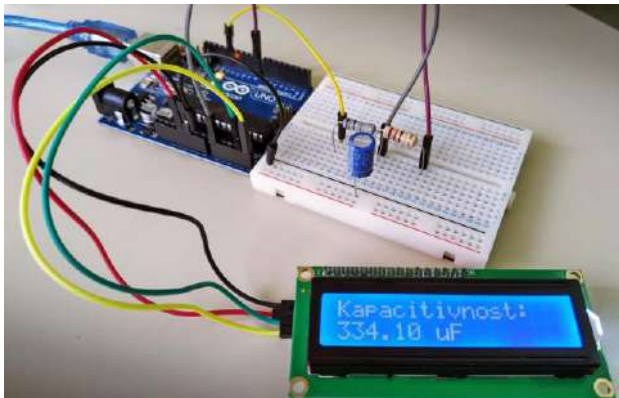
Kondenzator se puni preko otpornika R_1 , dok se prazni preko otpornika R_2 . Rezultati merenja prikazuju se preko LCD-a.

C. Praktična realizacija

Na slici 7 prikazana je platforma za merenje nepoznate kapacitivnosti koja je realizovana pomoću sledećih komponenti [8]: Arduino UNO, protoploča, otpornici $R_1 = 10 \text{ k}\Omega$ i $R_2 = 220 \Omega$, kondenzator nepoznate kapacitivnosti, 16x2 LCD displej i konektori.

Da bismo proverili funkcionalnost realizovane platforme izvršeno je merenje nekoliko nepoznatih kapacitivnosti. Upoređene su vrednosti sa displeja sa nominalnim vrednostima tih kondenzatora. Rezultati su predstavljeni u Tabeli IV. Određeno je i odstupanje rezultata primenom izraza

$$\delta_C [\%] = \left| \frac{C_{\text{mer}} - C_{\text{nom}}}{C_{\text{nom}}} \right| \cdot 100. \quad (8)$$



Slika 7. Realizacija platforme za merenje nepoznate kapacitivnosti.

TABELA IV
MERENJE NEPOZNATE KAPACITIVNOSTI.

Merena vrednost $C_{\text{mer}} [\mu\text{F}]$	Nominalna vrednost $C_{\text{nom}} [\mu\text{F}]$	τ [ms]	δ [%]	δ_C [%]
334.10	330	3341	± 20	1.24
442.00	470	4420	± 20	5.95

Pomoću realizovane platforme, moguće je izmeriti i ekvivalentnu kapacitivnost redne i paralelne veze dva ili više kondenzatora. Rezultati su prikazani u Tabelama V i VI. Ni ovde, kao i kod realizacije platforme za merenje nepoznatih otpornosti nisu uzete u obzir tolerancije priključenih otpornika, niti otpornosti kontakata i samih provodnika. Merenjem je pokazano da su izmerene vrednosti u granicama tolerancije korišćenih kondenzatora, što navodi na zaključak da se primenom ove platforme mogu približno izmeriti vrednosti nepoznatih kapacitivnosti,

kao i ekvivalentna kapacitivnost mešovite veze kondenzatora. Na ovaj način, studenti mogu da potvrde važenje relacija za određivanje redne i paralelne veze dva ili više kondenzatora.

TABELA V
REDNA VEZA DVA KONDENZATORA C_1 I C_2 .

$C_1 [\mu\text{F}]$	$C_2 [\mu\text{F}]$	Merena vrednost kapacitivnosti $C_{\text{mer}} [\mu\text{F}]$	Nominalna vrednost kapacitivnosti $C_{\text{nom}} [\mu\text{F}]$
330	330	159.3	165.00
330	470	183.90	193.87
470	470	218.7	235.00

TABELA VI
PARALELNA VEZA DVA KONDENZATORA C_1 I C_2 .

$C_1 [\mu\text{F}]$	$C_2 [\mu\text{F}]$	Merena vrednost kapacitivnosti $C_{\text{mer}} [\mu\text{F}]$	Nominalna vrednost kapacitivnosti $C_{\text{nom}} [\mu\text{F}]$
330	330	659.90	660
330	470	763.80	800
470	470	890.20	940

IV. ZAKLJUČAK

U radu je prikazana mogućnost korišćenja Arduino platforme za realizaciju laboratorijskih vežbi iz Osnova elektrotehnike. Merenjem nepoznatih otpornosti i kapacitivnosti, kao i merenjem redne i paralelne veze, studenti su u mogućnosti da na jednostavan način provere i potvrde osnovne relacije i zakone elektrotehnike, koristeći savremene tehnologije.

ZAHVALNICA

Autori se zahvaljuju doc. dr Mirjani Perić na izdvojenom vremenu, sugestijama i pomoći pri realizaciji i pripremi ovog rada.

LITERATURA

- [1] A. Vorgučić, *Osnovi elektrotehnike – laboratorijske vežbe*, Elektronski fakultet, Niš, 1972.
- [2] D. N. Mitić, *Osnovi elektrotehnike – laboratorijske vežbe*, Naučna knjiga, Beograd, 1988.
- [3] D. N. Mitić, *Osnovi elektrotehnike – praktikum i radni listovi za laboratorijske vežbe*, Elektronski fakultet, Niš, 2003.
- [4] D. Mitić, *Elektrotehnika 1*, Petograf, Niš, 2007.
- [5] Fritzing – verzija 0.9.3, <http://fritzing.org> (preuzeta: 08.09.2021.)
- [6] <https://www.circuitbasics.com/arduino-ohm-meter/>
- [7] D. Mitić, *Elektrotehnika 2*, Petograf, Niš, 2008.
- [8] <https://www.circuitbasics.com/how-to-make-an-arduino-capacitance-meter/>

Maglev vozovi-pregled dosadašnjih istraživanja

Mladen Krstić, Đorđe Jovanović, Branislav Milenković

Sadržaj - Tema ovog rada se tiče projektovanja, tehnologije, trenutne primene i budućnosti vozova na magnetnu levitaciju (maglev). Saobraćajni sistem koji koristi maglev je stabilniji, brži, ekonomičniji, efikasniji. Maglev sistemi se trenutno primenjuju na vozove velikih brzina. Maglev je oblik pogona koji koristi magnetnu levitaciju da drži voz u vazduhu umesto da korsiti točkove, osovine i ležišta. Sa maglevom, vozilo je u vazduhu na malom rastojanju od vođica koristeći magnete da podigne voz i drži ga u vazduhu (na distanci od ne više od 10cm). U budućnosti ovi maglev vozovi velike brzine mogu biti konkurencija avioindustriji.

I. UVOD

Maglev vozovi se glatkije i tiše kreću od drugih vrsta vozova. Takođe, ne oslanjaju se na vučnu silu ili silu trenja, ubrzanje i usporenje je veće od drugih vrsta vozova, i vremenski uslovi ne utiču na njih. Snaga potrebna za levitaciju nije velika u odnosu na potrošnju energije; većina snage ovih vozova se koristi na savladavanje otpora vazduha, kao što je slučaj sa svim oblicima prevoza koji se kreću velikim brzinama. Ovi vozovi se kontinuirano kreću brzinama većim nego druge vrste vozova. U [1] je pojašnjeno da je magnetna levitacija napredna i efikasna tehnologija. Ona se može iskoristiti u industrijske svrhe, ali i u kancelarijama i domovima. Neke od primena su ventilacija u zgradama, prevozna sredstva, oružja (puške i raketno naoružanje), nuklearni reaktori, liftovi, igračke, olovke. Najveća prednost je zelena energija, dok odsustvo kontakta uklanja efekat sile trenja. Magnetna levitacija povećava efikasnost i životni vek celog sistema i značajno smanjuje troškove održavanja. U ovom radu se prikazuje prednost i potreba za magnetnom levitacijom u inženjeringu u budućnosti i svetu. Može se reći da je magnetna levitacija budućnost letećih vozova i automobila.

II. MAGLEV TEHNOLOGIJA

Maglev tehnologija poznata je već dugi niz godina. Patent na nju su dobila dva Amerikanca u oktobru 1969 godine ali prepreku daljem i razlog sporog razvoja ovog tipa prevoza predstavljale su loše karakteristike postojećih materijala i slabo, za potrebe MAGLEV-a, razvijena precizna elektronika tog vremena.

M. Krstić i Đ. Jovanović su studenti treće godine doktorskih studija, Fakultet za mašinstvo i građevinarstvo, Univerzitet u Kragujevcu, Dositejeva 19, 36000 Kraljevo, Srbija. E-mail: mladenkrstic994@gmail.com

B. Milenković je viši stručni saradnik na Fakultetu primenjenih nauka, Univerzitet UNION-Nikola Tesla, Dušana Popovića 22a, 18000 Niš, Srbija, E-mail: bmilenkovic92@gmail.com

Prva svetska "pruga" na bazi magnetne levitacije koja je korišćena u komercijalne svrhe je izgrađena u Birmingham-u u Velikoj Britaniji 1985 godine. Bila je duga 600 metara i povezivala je aerodrom sa obližnjom železničkom stanicom. Mali šatl je levitirao na visini od 15 milimetara. Ipak zbog brojnih problema posle 11 godina zamenjen je običnim vozom.

Najviše uspeha u razvoju MAGLEV tehnologije imali su Nemačka i Japan. Japan, kao, jedna od vodećih tehnoloških sila sveta sa velikim brojem gusto naseljenog stanovništva ulaže ogromna sredstva u razvoj brzog i efikasnog prevoza. Istraživanja u Japanu u ovoj oblasti počela su još 1970 godine. Japan danas drži svetski rekord koji je postignut na jednoj pruzi i on iznosi 581 km/h. Postignut je 2. decembar 2003 godine.



Slika 1. Maglev voz (Japan) [5].

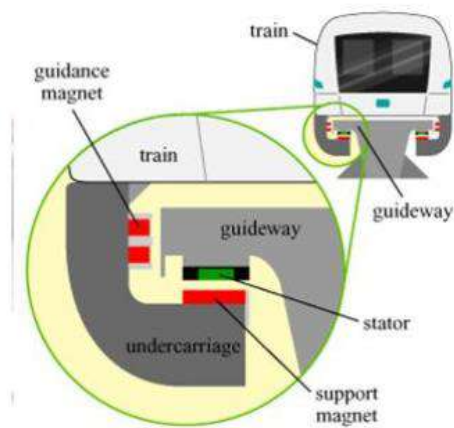
Tipovi maglev tehnologije:

- Elektromagnetna suspenzija
- Elektrodinamička suspenzija
- Inductrack sistem

A. Elektromagnetna suspenzija

U ovom sistemu elektromagneti su pričvršćeni za voz i za vođice šine. Imaju feromagnetne statore na stazi, koji im pomažu da podignu voz u vazduh. Imaju magnete za navođenje na bočnim stranama staze, koji su postavljeni duž cele staze. Računar se koristi za kontrolu visine levitacije voza, koja iznosi od 1 do 15cm. Maksimalna brzina koju mogu dostići ovi vozovi je oko 438km/h, što je

dovoljno da prave konkurenciju avionima. Na vozu se nalazi baterija koja pruža dodatnu energiju za pokretanje kabine, kao na Slici 2.



Slika 2. Elektromagnetna suspenzija [2].

H. Yaghoubi i H. Ziari su u svom radu [2] prikazali da magnetna polja u unutrašnjosti i spoljašnjosti vozila su manja od elektrodinamičke suspenzije. Dokazano je da za komercijalne potrebe ova tehnologija može dostići velike brzine, bez korišćenja točkova i sistema za suspenziju. Zbog nestabilnosti sistema i konstantne korekcije od strane spoljašnjih sistema, mogu se pojaviti problemi usled vibracije.

B. Elektrodinamička suspenzija

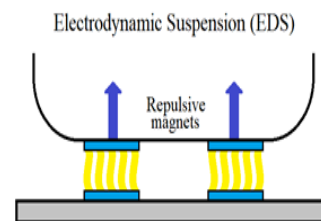
H. Behbahani, H. Yaghoubi, and M. A. Rezvani su u svom radu [3] opisali superprovodni sistem koji se drži na niskim temperaturama, postavljen ispod voza. Ovaj sistem omogućava levitaciju od 10cm. Magnetno polje koje omogućava levitaciju voza je napravljeno korišćenjem superprovodnih magneta. Ukoliko se ovi stalni magneti postave u niz, koristili bi se i kao indutrack sistem. Sila u stazi se kreira indukovanim magnetnim poljem u žicama ili provodnim trakama na stazi. Kod elektrodinamičke suspenzije (EDS skraćeno), i vođice i voz emituju magnetno polje, a voz se održava u vazduhu silama odbijanja i privlačenja između ovih magnetnih polja.

EDS sistemi imaju i veliku manu. Pri malim brzinama, struja koja indukuje se u ovim kalemima i rezultatni magnetni fluks nije dovoljno veliki da izdrži težinu voza. Iz ovog razloga, voz mora da ima neku vrstu točka ili opreme za prizemljenje da podupire voz dok ne dostigne brzinu koja može da podrži levitaciju. Pošto voz može da se zaustavi na bilo kojem mestu, na primer zbog problema funkcionisanja opreme, cela staza mora da podnese težinu prilikom i malih i velikih brzina. Druga mana je da EDS sistem prirodno stvara polje u stazi ispred i iza magneta, što utiče negativno na magnete i stvara otpor vazduha.

H. Behbahani, H. Yaghoubi, i M. A. Rezvani, u svom radu [3] objašnjavaju da ime maglev potiče od

početka reči „magnetna levitacija“. Magnetna levitacija je veoma napredna tehnologija, sa primenom u ekologiji (male i velike turbine na vetar, u domovima, kancelarijama, itd.), stambene zgrade (ventilacija), transportacioni sistemi (voz na magnetnu levitaciju, brzi personalni transport, itd.), vojna industrija (puške, raketno naoružanje), nuklearno inženjerstvo (centrifuga i nuklearni reaktor), građevinsko inženjerstvo (lift), reklamiranje (levitiranje svega unutar ili iznad raznih kadrova može biti odabrano), igračke (vozovi, svemirci koji levitiraju iznad svemirskog broda, itd.) i pribor za pisanje (olovke). Ono što je zajedničko u ovim primenama je manjak kontakta i stoga nema habanja i trenja. Ovo povećava efikasnost, smanjuje troškove održanja, i povećava korisno vreme života sistema. Tehnologija magnetne levitacije se može iskoristiti kao efikasna tehnologija u različitim industrijama. Maglev tehnologija je već privukla pažnju mnogih zemalja širom sveta. Ovaj rad pokušava da prikaže najbitnije upotrebe tehnologije magnetne levitacije. Rezultati jasno prikazuju da se maglev tehnologija može smatrati rešenjem za buduće potrebe u inženjerstvu.

Magneti na vozu imaju veliku marginu između šine i voza, i omogućavaju najveće zabeležene brzine (561 km/h) i kapacitet za velika opterećenja. Takođe su demonstrirane uspešne operacije koristeći superprovodnike na visokim temperaturama sa svojim magnetima na vozu, ohlađeni tečnim azotom. Jaka magnetna polja na vozu bi nepovoljno delovala na putnike koji imaju električne uređaje poput pejsmejкера, hard diskova, kreditnih kartica, što ukazuje na potrebu za magnetnom zaštitom. Ograničenja u pogledu granice induktivnosti ograničavaju maksimalnu brzinu vozila; vozilo mora biti vučeno prilikom vožnje na malim brzinama, što je prikazano na Slici 3.



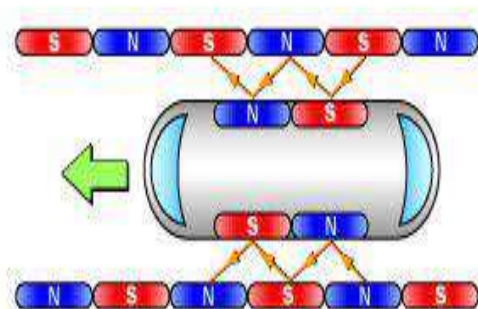
Slika 3. Elektrodinamička suspenzija [2].

C. Indutrack sistem (stalni magnet pasivna suspenzija)

Za ovaj sistem nije potrebna snaga za aktivaciju magneta. Magnetno polje se nalazi ispod vagona, oni mogu da generišu dovoljno sile pri malim brzinama (oko 5km/h) za levitaciju maglev voza. U slučaju nestanka struje vozovi polako usporavaju do zaustavljanja, trajni magneti su poslagani u niz koji pomaže u održavanju voza u vazduhu. Zahtevaju ili točkove ili segmente staze koji se pomeraju u slučaju zaustavljanja voza. Niti Indutrack niti superprovodni EDS nisu u stanju da levitiraju vozila koja miruju, iako Indutrack obezbeđuje levitaciju do nižih brzina, i za ove sisteme su potrebni točkovi. EMS sistemi su bez točkova.

III. RAZVOJ MAGLEV VOZOVA (FAKTORI)

Postoje različiti faktori koji se koriste u razvoju maglev vozova, koji pomažu u pomeranju, obezbeđuju stabilnost i usmeravanje. H. Behbahani i H. Yaghoubi su u radu [4] opisali da neki EMS sistemi, poput HSST/Linimo može da obezbedi i levitaciju i propulziju koristeći linearni motor koji se nalazi u vozu. Ali neki EDS i EMS sistemi imaju mogućnost da održavaju voz u vazduhu koristeći magnete, ali ne mogu da ga potisnu napred. Zbog toga, vozilima je potrebna za propulziju. Linearni motor (propulzioni kalemovi) postavljen na stazi je jedno rešenje, kao na Slici 4.



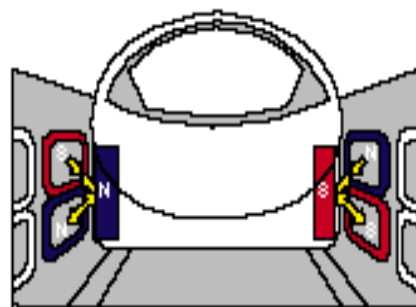
Slika 4. Propulzija [2].

Prema Earnshaw-ovoj teoremi, bilo koja kombinacija statičkih magneta ne može biti u stabilnoj ravnoteži. Stoga, dinamičko magnetsko polje je potrebno da bi se postigla stabilizacija. EMS sistemi se oslanjaju na aktivnoj elektronskoj stabilizaciji, koja stalno meri rastojanje ležišta i šteluje shodno tome elektromagnet. Svi EDS sistemi se zasnivaju na promenljivom magnetnom polju koje stvara električnu struju, i ovo može doprineti pasivnoj stabilnosti. Budući da maglev vozovi u suštini lete, stabilizacija visine frekvencije, kotrljanja i skretanja je neophodna koristeći magnetnu tehnologiju. Uz rotaciju, kretanje napred i nazad, horizontalno i vertikalno mogu biti problem za neke tehnologije.

Neki sistemi koriste rešenja sa nultom strujom (takođe se zovu i sistemi sa nultim fluksom). Koriste kalem tako navijen da ulazi u dva suprotna, alternirajuća polja, tako da je prosečan fluks u petlji nula. Kada je vozilo upravljeno prema napred, ne protiče struja, dok ukoliko skreće sa prave linije, to stvara promenljiv fluks koji generiše polje koje prirodno gura i vuče vozilo nazad u pravu liniju. Sistem vođenja za maglev vozove je prikazan na Slici 5.

Neki sistemi (naročito Swissmetro sistem) predlažu korišćenje tehnologiju maglev vozova koja koristi vakuum tube (vactrain), što se koristi za uklanjanje otpora vazduha. Ovo ima potencijal da poveća brzinu i efikasnost poprilično, budući da se većina energije za obične maglev vozove troši na otpor vazduha. Jedan potencijalan rizik za putnike ovih vozova je da mogu biti podložni riziku od dekompresije kabine osim ako sistemi za monitoring

bezbednosti tunela mogu da ponovo podignu pritisak tube u slučaju kvara voza ili nesreće.



Slika 5. Sistem vođenja [2].

Energija za maglev vozove se troši na ubrzanje voza, i ta energija se može ponovo dobiti kada voz usporava (regenerativno kočenje). Takođe se energija troši na održavanje voza u vazduhu i da se stabilizuje kretanje voza. Glavni deo energije je potreban za guranje voza kroz vazduh, i troši se na otpor vazduha. Takođe, deo energije se koristi za klima uređaje, osvetljenje i druge sekundarne sisteme.

Pri malim brzinama procenat snage (energija po jedinici vremena) koja se troši na levitaciju može biti značajan, čak i do 15% više snage u odnosu na metro ili servisiranje osvetljenih šina. Takođe, na veoma malim odstojanjima energija koja se troši na ubrzanje je veoma velika. No, snaga koja se koristi za savladavanje otpora vazduha se povećava sa trećim stepenom brzine, stoga dominira pri velikim brzinama.

Konvencionalne železnice su vrlo verovatno efikasnije pri malim brzinama. No, budući da ne postoji dodir između šina i vozila, maglev vozovi ne trpe silu trenja pri kotrljanju, što ostavlja samo otpor vazduha i elektromagnetni otpor, potencijalno povećavajući efikasnost trošenja električne energije. Neki sistemi, poput Central Japan Railway Company SCMaglev, koriste gumene točkove pri malim brzinama. Težina elektromagneta u mnogim EMS i EDS dizajnima se čini kao veliki problem. Jako magnetno polje je potrebno da se maglev vozilo podigne i održava u vazduhu. Za Transrapid (nemački maglev), snaga potrebna za levitaciju iznosi 1 do 2 kW/t. Druga putanja za levitaciju jeste da se koriste superprovodni magneti kako bi se smanjila potrošnja električne energije elektromagneta, kao i cena održavanja polja. Većina električne energije ja se koristi za TRS je za propulziju i savladavanje trenje otpora vazduha pri brzinama od 16km/h. Konvencionalni vozovi stoga teže manje o magleva. Budući da veliki izvor buke kod maglev vozova dolazi od razmeštenog vazduha, maglev vozovi proizvode manje buke od konvencionalnih vozova pri istim brzinama. Međutim, studija je zaključila da buka od magleva može biti karakterizovana kao putnički saobraćaj, dok konvencionalni vozovi imaju dodatnih 5-10dB, pri

čemu se smatraju manje nesnosnim pri istoj meri buke. Maglev dizajn eliminiše korišćenje kočenja i dodatne žice za razliku od konvencionalnih vozova, budući da dobijaju napajanje iz zemlje. Njihov dizajn je toliko aerodinamičan da dostižu brzinu od oko 483 km/h mnogo brže od konvencionalnih brzih vozova.

Za maglev sisteme male ili velike brzine nema signaliziranja. To je zato što ovim sistemima diriguju računari. Pored toga, pri ekstremno visokim brzinama ovih sistema, nijedan ljudski kočničar ne bi mogao da reaguje dovoljno brzo da uspori ili zaustavi voz na vreme. Zbog toga ovim sistemima su potrebna posebna prava za put i obično se preporučuje da budu nekoliko metara iznad nivoa zemlje.

Dva radarska tornja za maglev sisteme su u kontaktu sa EMS vozilom u svakom trenutku, i postoji dvosmerna komunikacija između vozila i glavnog računara u centralnom postrojenju. Takođe, ne postoji potreba ni za trubama ili zviždaljkama kao oblicima signalizacije.

Avioni su u teoriji fleksibilni, ali komercijalne vazdušne rute nisu. Maglevi visoke brzine su projektovani da se takmiče sa letovima od 800 kilometara ili manje, u pogledu vremena provedenog na putu. Pored toga, dok maglevi mogu da opslužuju nekoliko gradova na jednoj ruti i ne zavise od vremenskih uslova, avioni ne mogu da se približe tom nivou pouzdanosti ili performanse. Zato što maglev vozila napaja struja, a ne gorivo, troškovi za magleve su manje podložni promenama cena koje se dešavaju na tržištu nafte. Putovanje maglevom takođe nudi značajnu bezbednosnu marginu u odnosu na putovanje avionom, pošto maglevi ne mogu da se sudare sa drugim maglevima ili skrenu sa šina. Avionsko gorivo je značajna opasnost prilikom poletanja i sletanja, pri čemu postoji opasnost od nesreće. U realnim situacijama, brzina magleva je manja od brzine aviona, no maglevi i dalje štede vreme zbog manjeg broja prepreka u odnosu na avionski prevoz. Sa avionskim prevozom, ljudi treba da dođu na vreme na aerodrom zbog čekiranja, bezbednosti, ukrcavanja, itd. Kod avionskog prevoza, vreme se takođe troši (na velikim aerodromima) na plaćanje aerodromskih taksi, čekanje u redu za poletanje i sletanje, što je u slučaju magleva zanemarljivo.

IV. ZAKLJUČAK

Maglev vozovi troše manje energije u poređenju sa konvencionalnim vozovima. Ne zahtevaju veliki motor, budući da rade na linearni motor. Kreću se brže od konvencionalnih vozova, jer ne podležu sili trenja, samo otporu vazduha. Nekompatibilni su sa postojećim železnicama, jer im je potrebna dodatna šina kako bi levitirali, za razliku od tradicionalnih vozova velike brzine. Inicijalna cena je visoka, ali može se smanjiti u budućnosti.

ZAHVALNICA

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja, Republike Srbije, kroz Matematički institut SANU.

LITERATURA

- [1] Shwethasingh, Aradhanasingh. "Magnetic Levitation Methods and Modeling in Maglev Trains", 2014 ISSN: 2277 128 01.
- [2] H. Yaghoubi and H. Ziari, "Development of a maglev vehicle/guideway system interaction model and comparison of the guideway structural analysis with railway bridge structures," Journal of Transportation Engineering, vol. 137, no. 2, pp. 140–154, 2010.
- [3] H. Behbahani, H. Yaghoubi, and M. A. Rezvani, "Development of technical and economical models for widespread application of magnetic levitation system in public transport," International Journal of Civil Engineering, vol. 10, no. 1, pp. 13–24, 2012.
- [4] H. Behbahani and H. Yaghoubi, "Procedures for safety and risk assessment of maglev systems: a case study for long-distance and high-speed maglev project in Mashhad-Tehran route," in Proceedings of the 1st International Conference on Railway Engineering, High-speed Railway, Heavy Haul Railway and Urban Rail Transit, pp. 73–83, China Railway Publishing House, Beijing Jiaotong University, Beijing, China, 2010.
- [5] <https://en.wikipedia.org/wiki/Maglev>
- [6] Global Journal of Researches in Engineering Mechanical & Mechanics Volume 13 Issue 7 Version 1.0 Year 2013 Global Journals Inc. (USA) Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Simulation of One-Week Production for a Photovoltaic System and How Does It Affect the Distribution Grid

Aleksandra Stojkovska

Abstract – Last decade was turn point for Photovoltaic (PV) solar energy system, which has shown its huge potential. In the world, amount of PV installed on/off grid has rapidly increased. The annual potential for electricity production from solar power in Macedonia is estimated at 1,300 - 1,550 kWh/kWp, therefore EVN AD Skopje – distribution company offers a complete ‘key in hand’ solution for photovoltaic systems for all households and businesses in Macedonia in order to encourage citizens to use Renewable energy sources (RES) and solar potential. This paper reviews the installed photovoltaic capacities in a real network, the challenges associated with the integration of the photovoltaic capacities network, and highlights of the future work strategies in the electricity distribution sector.

I. INTRODUCTION

The solar electric generation system usually known as Solar Photovoltaic or solar PV uses sun light to produce electricity, for instance to power electrical appliances. Photovoltaic systems use cells, generally consisting of one or more layers of semi-conducting (generally, silicon) material – P and N type semi-conductors. When each cell is connected appropriately in series/parallel they can form a panel. [1]. When sunlight falls on to the semiconducting cells, it releases electrons and creates an electric field across the layers. Consequently, this causes electrons to flow, creating electricity. The energy cycle of a photovoltaic system in a household is depicted in Figure 1. The system consists of a solar panel, DC/AC inverter and meter. The solar panels are mounted on the roof of the household, and are called roof top solar panels [1].

Photons from the sun are converted into electricity by roof top solar panels

- The electricity produced by the solar panels is Direct Current (DC). This DC is converted into AC current (mains) by the DC/AC inverter so they can be used in the home;
- The usage meters measure the amount of electricity produced by the solar panels;
- Unused electricity (surplus) is exported into the electricity grid and used by the utility company - this set up is called a grid-tied solar panel system.

A. Stojkovska, BSc in Electrical Engineering and Information Technologies, in Power Systems, currently on master studies - European University second cycle of studies at the Faculty of Economics, Study program – Business Management, E-mail: aleksandrasto10@gmail.com

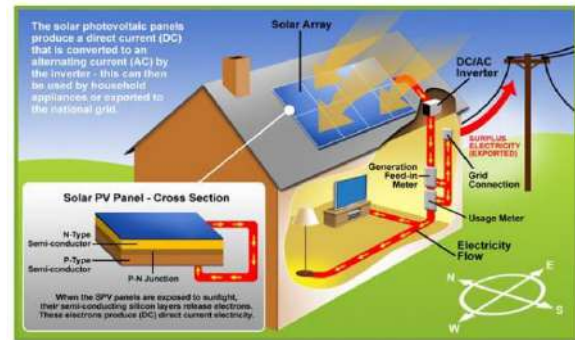


Fig. 1. Energy cycle of Photovoltaic (PV) system in a household [1].

II. WAYS OF REQUESTING FOR SOLAR PANELS FOR HOUSEHOLDS AND COMPANIES

Solar renewable energy in Macedonia is available for use by anyone who is interested, such as households and companies. The whole process of using the services, starts with submitting a request to EVN AD Skopje. The request provides basic information about the facility or location where the PV will be installed.

Based on the request, an offer is prepared, which if it is accepted, a contract will be signed and a detailed personalized solution for the consumer (households or companies) will be made. The EVN service implies full development and implementation of the solution (plan) that should provide the most optimal results and meet the needs of the user. The process continues with the development of a detailed analysis of the potential user's facility/location needs. Based on this analysis, EVN offers a complete ‘key in hand’ solution for photovoltaic system. The service is personalized and therefore for each user the price would be different. The price depends on many factors and most on the size of the PV. [2]

A. Grid connected PV system

Most of the installed photovoltaic systems in the EU (about 90%) are network connected systems, which means that these systems are connected to the distribution voltage network. [6] These systems work fully synchronized and allow primarily use of energy produced by photovoltaic systems and energy intake in case of insufficient production of the photovoltaic system from the distribution

network. This results in a significant reduction in the amount of energy taken from the distribution network.

What it takes to connect to a network:

- Solar photovoltaic cells - which convert solar energy into direct current;
- Inverter - DC electricity produced by solar cells with the inverter is converted into AC electrical energy that is compatible with public grid electricity;
- Electricity meter - is installed next to the permanent electricity meter. It measures the amount of electricity produced in the public grid and shows how much electricity is produced by the photovoltaic system.

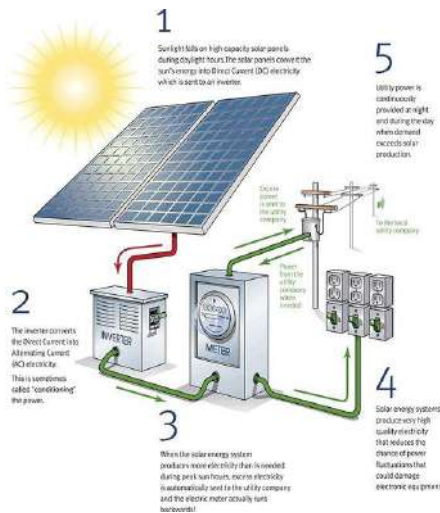


Fig. 2. Energy cycle and Net metering [6].

"Net-Metering" is a measure in the field of alternative energy which provides the owners of PV systems primarily intended to meet their own energy needs with the opportunity to transfer excess unused electricity back to the distribution network. This means that the amount of produced solar electricity (expressed in kWh) that is delivered to the distribution network is subtracted from the amount of electricity taken from the distribution network, which means that the difference in the received electricity is paid ("net" amount).

Households such as houses and buildings will have PV system installed on their rooftops, which will also be connected with the distribution network. Also when the PV system will produce more electrical energy than the household is using - unused electricity (surplus) is exported into the electricity grid and used by the utility company.

III. IMPACTS OF PV SYSTEMS

Generally, grid-connected PV systems are employed to improve the performance of the electric power network. PV systems in this case, are considered as generators and they provide energy at the load side of the distribution

network. This setup reduces the feeder active power loading and thus improves the voltage profile and enhances the life time of shunt capacitors and series voltage regulators. The Load Carrying Capacity (LCC) can be increased by employing PV systems – which is the amount of load a power system can handle without comprising on certain reliability criteria of the existing network [4]. Nevertheless, PV systems can have various negatives impacts on the power systems, particularly if the penetration is high. The impacts of PV systems on electric networks are divided into two categories based on the size of the PV system: impacts of small and medium PV systems and large PV systems. [8]

A. Impacts of large PV system

- Serious power, frequency and voltage fluctuation – this is due to the erratic and unpredictable nature of PV arrays, as they are highly dependent on environmental conditions such as solar irradiation and temperature. For example, shading caused by passing clouds can vary the insolation affecting the production of PV system, thus leading to rapid fluctuations in its output. This will lead to power quality issues. Active power fluctuations cause frequency variations, reactive power and consequently voltage fluctuations;
- Stability of electric power systems - the erratic nature of the PV system output has greater impacts on the power system operation. When a large PV system cannot provide the dispatchable supply in each point of time to the varying demand, it not only fails to deal with the uncontrolled demand, but also the uncontrolled generation. This will result in severe power system instability.

Large PV system means increased ancillary services requirements:

The electric grid must act as a buffer to compensate the power fluctuation of the output of the power PV system. Other generators must be adjusted frequently to deal with this PV power fluctuations to maintain the overall power grid frequency and maintain the nominal operation of the electric power grid.

B. Impacts of small and medium PV systems

Reverse power flow: in the normal operation of a distribution system, the power flow is unidirectional from higher voltage system to lower voltage system, i.e., MV to LV system. At high PV penetration level, especially during noon, there will be some instances when the net production is more than the consumption (the demand) - this results in a reverse power flow from the LV side to the MV side. This reverse power flow results in overloading of the distribution feeders, increased system power losses, hampers the operation of automatic voltage regulators installed along the distribution feeders and

affects the online tap changers of the distribution transformers.

- The voltage control becomes difficult as they affect the capacitor banks and voltage regulators;
- Increases power losses in the system;
- Phase unbalance may take place when PV generations are not distributed evenly;
- Power quality issues: the inverters used to interface PV system with the power grid produce harmonic currents, contributing to the total harmonic distortion of both current and voltage at the point of common coupling.

C. Grid-Connected System - parameters of a project and a generation of it

The project is prepared with the help of the program PVsyst 6.8.4, thus calculates all the necessary data for its construction and consumption that would be achieved during one year.

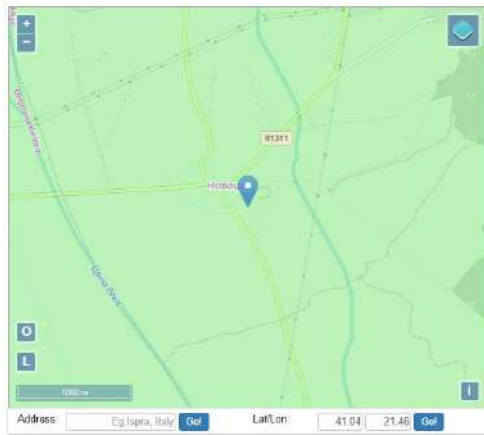


Fig. 3. Place where PV should be built [7].

The photovoltaic power plant should be built on the location Novaci, Republic of Macedonia, with planned power generation of 225 kW. The data download is from the NASA-SSE satellite data database 1983-2005 (modified by user) - Synthetic. The coordinates and altitude are downloaded from Google Maps, while the time zone is taken as Time zone UT + 1. For Novaci the latitude is 41.04° N, while the longitude is 21.46° E and the altitude of 579 m. The input data for the photovoltaic orientation is the optimal plane placement angle (photovoltaic) is 30° to the south, while the azimuth or orientation of the photovoltaic is 0 or to the south.

The input data for the photovoltaic model are the following:

- Type Si-poly, model CS3K - 285P, manufacturer Canadian Solar Inc, planned output 225 kWp
- Inverter model Sunny Tripower 25000TL-30, manufacturer SMA, operating voltage 390-800 V and nominal power 25kW.

Specifically, for this project, have been selected: 9 inverters, 22 photovoltaic modules in series and 36 series of photovoltaic arrays.

Generation of an implemented PVs system in Novaci is show on Figure 4. X-axis is filled with date (one week from summer month). On the other hand, Y-axis show as generated produced power, from the days shown on X-axis.



Fig. 4. Generation of an implemented PVs system in Novaci [5].

Factors that affect the efficiency of PVs, i.e., electricity production are:

- PV generally produce energy at a reduced % of maximum power in cloudy weather. In conditions of dense dark clouds, intense rain and/or snow and in fog, energy production is impossible and reduced to zero;
- Insolation is daily total direct sunlight and is expressed in (W/m²). On average, EU locations receive between 4 and 6 hours of insolation per day, depending on the season and location.
- Number of sunny days;
- One of the main reasons for power loss in a photovoltaic system is when part of the panel or the whole array is shaded. If any cell is shaded or the entire string is shaded, the performance drops and equals the cell with the lowest performance, i.e., the weakest panel according to the over form in that array;
- Setting angle - optimally to the south, at approx. $30\text{--}50^\circ$.

The grid voltage levels will vary and fluctuate throughout the day depending on how much power is being drawn from the grid, and how much solar is being sent back.

Distribution utility is continually adjusting grid voltage to attempt to keep it within required limits. But high grid voltage isn't a problem that can be solved so easily. Distribution utility should invest more on grid connection such as maintaining or preserving a grid, increasing cable size, planning implementation of new PV systems on grid in order to avoid voltage variation.

D. Voltage opportunities with/without PV system connected to the grid

As for the voltage, Macedonia has not implemented huge amount of PVs systems in the grid. Since the 2019

year was the year when the electricity market was liberated, and that is the reason why PVs systems are reduced. Because of the previous stated, the voltage drop is within limits, and not affect the grid much, as will affect if there is more PVs.

Below we can see two examples of circuits with and without PV depicted with references Figure 6. and Figure 7. They are seen gravitational and the lower and upper voltage thresholds in the nodes, which obviously increase with the PV node the voltage drastically increases which means that if the PV is connected to the grid, we will reach a voltage greater than the maximum [4]. These figures are perfect example of showing how PVs system affect the distribution grid in Macedonia, network without/with PV system.

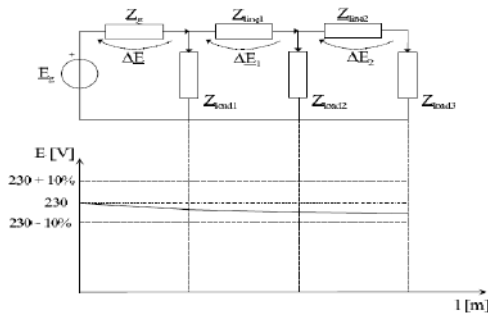


Fig. 5. Network without PV.

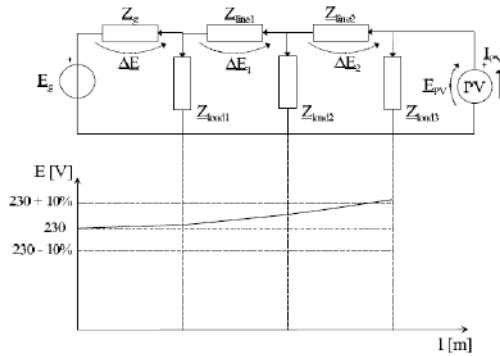


Fig. 6. Network with PV.

IV. CONCLUSION

Photovoltaic systems are one of the most anticipated growing sources of electricity in decades to come. Cell efficiency increases rapidly and costs decrease. The impact of PV on the connected network depends on the size of the PV, installation location and system topology.

Republic of Macedonia is a country that has ideal conditions for the use of PV systems. At the environmental level, we have the opportunity to use the available natural resources - potential for electricity production from solar power in Macedonia is estimated at 1,300-1,550kWh /kWp.

If we keep in mind all aspects of impact of PV system to the grid, it is clear that this process can be relied on future science - based analysis, thus with the hope that soon new regulatory and financial measures in our country will stimulate the purchase of PV.

REFERENCES

- [1] S. Locke, "How does solar power work?", Scientific American: <https://www.scientificamerican.com/article/how-does-solar-power-work/>, 29.08.2021
- [2] „Соларната обновлива енергија достапна за домаќинствата и за компаниите“, ЕВН ИНТЕРН, Издавач, Скопје, 08/2018.
- [3] D. M. Tobnaghi, 'A Review on Impacts of Grid-Connected PV System on Distribution Networks', International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, Vol. 10, No. 1, 2016
- [4] A. F. Povlsen, "Impacts of power penetration from photovoltaic power system in distribution network," report IEA-PVPS Tsk 5, Feb 2002.
- [5] PVsyst software, <https://www.pvsyst.com/>
- [6] How Energy Travels: What Happens with PV Solar Power, <https://medium.com/@solar.dao/how-energy-travels-what-happens-with-pv-solar-power-16a047dbe87e>, 19.09.2021
- [7] PHOTOVOLTAIC GEOGRAPHICAL INFORMATION SYSTEM, https://re.jrc.ec.europa.eu/pvg_tools/en/?fbclid=IwAR3efU3Dpo7-7MeYm2Ca3xleKu8HqIEBBh1xkEA5eEZ_KFkxtwSt6gN3tmk#PVP, 20.09.2021
- [8] МРЕЖНО ИНТЕГРИРАЊЕ НА ОБНОВЛИВИ ИЗВОРИ НА ЕНЕРГИЈА - умножени предавања, Проф. д-р Гога Цветковски, Скопје, 2018.

Sigurnosni IoT sistem sa električnom bravom

Jelena Nedeljković, Marko Tepavac

Sadržaj – Nikome nije prijatno kada čuje da je neki objekat, privatni ili poslovni, obijen. Uprkos tome, ovakve informacije se često nađu u medijima pa uglavnom svi tada osete neku, bar i najmanju dozu straha. U cilju prevencije ovakvih događaja, ljudi se danas sve češće okreću nekim vidovima zaštite svoje imovine. Glavni cilj ovog rada jeste da prikaže moguću realizaciju i primenu sigurnosnih sistema za zaštitu imovine. Za ostvarivanje tog cilja korišćeni su koncepti IoT (*Internet of Things*). Osnovni segmenti ovog sistema su računar *Raspberry Pi* i razvojni sistem *Arduino*.

I. UVOD

Digitalizacija je danas prisutna u svim sferama naših života. Na polju elektronike se razvijaju uređaji koji nisu standardni računarski uređaji ali su digitalizovani, tj. povezani sa internetom u cilju deljenja podataka.

Kada se govori o olakšavanju života primarna stvar jeste obezbeđivanje sigurnosti i zaštite imovine. U tom cilju ovaj rad daje primer jednog sigurnosnog sistema. Konkretno ovaj primer je realizovan pomoću računara *Raspberry-Pi* i razvojnog sistema *Arduino*. U ovom primeru zaštita imovine je realizovana korišćenjem kamere i PIR senzora koji je namenjen detektovanju pokreta. Pomenute periferije su povezane sa računarom *Raspberry Pi* koji je povezan sa internetom kako bi slao podatke, primao uputstva ili oba od strane nekog *servera*. Pomenuti proces se naziva IoT (*Internet of Things*). Kada se detektuje neki pokret vlasnik se obaveštava putem *mail* – a. Takođe, postoji ekran na kome se ispisuje željeni tekst kao i LED koja je uključena tokom snimanja. Takođe, postoji i sistem otključavanja brave pomoću RFID čitača. Sistem opisan u ovom radu može biti okarakterisan kao celina koja se sastoji od dva zasebna sistema.

II. PRINCIP RADA

Ovaj sistem se sastoji od sledećih elemenata:

A) Podsystem za detekciju pokreta, snimanje i fotografisanje, čuvanje fotografije i slanje obaveštenja na *mail* korisnika baziran na *Raspberry Pi* računaru;

B) Podsystem za paljenje LED i ispisivanje željenog teksta na ekran baziran na *Arduino* razvojnem sistemu;

C) Podsystem za automatsko otvaranje vrata pomoću RFID čitača baziran na *Arduino* razvojnem sistemu;

J. Nedeljković je student master studija, studijski program Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: j.nedeljkovic007@gmail.com

M. Tepavac je student IV godine, modul Elektronika, podmodul Multimedijalne tehnologije, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: tepavac.marko98@gmail.com

D) *Hardware*-ska realizacija i maketa projekta.

A. Podsystem za detekciju pokreta, snimanje i fotografisanje, čuvanje fotografije i slanje obaveštenja na *mail* korisnika baziran na *Raspberry Pi* računaru

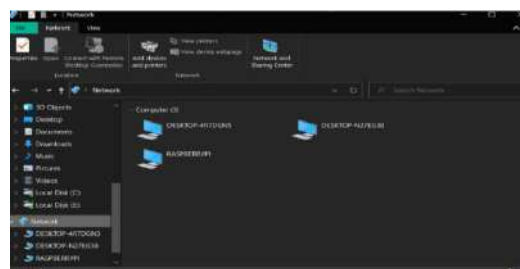
Funkcija ovog podsystema je da kamera, nakon detekcije pokreta pomoću PIR senzora, automatski snimi video fajlove i skladišti ih u označeni direktorijum. Nakon toga, putem *mail* adrese obaveštava se korisnik slanjem informacije o vremenu kada je pokret bio detektovan. Na kraju, računar je podešen da direktorijum u kome se skladište video fajlovi bude dostupan preko lokalne mreže. Na taj način se može pristupiti fajlovima preko drugog računara, tableta ili pametnog telefona. U toku procesa snimanja LED je uključena i ona na taj način smanjuje otpornost fotootpornika koji je povezan na razvojni sistem *Arduino*. Na taj način se započinje izvršavanje procesa na *Arduinu*.

Za realizaciju ovog podsystema korišćeno je sledeće:

- *Raspberry Pi* 4 (Model B)
- *Micro SD* kartica od 8 GB
- Napajanje (5V, 2.5A)
- *HDMI* kabl i monitor
- Tastatura i miš
- *Raspberry Pi* kamera
- Senzor pokreta HC-SR501
- LED

Kako bi folder u kome će biti smešteni video fajlovi bio dostupan na lokalnoj mreži i bio moguć pristup sa nekog drugog računara, tableta ili pametnog telefona, potrebno je instalirati *Samba file server*.

Na ovaj način je direktorijum *Share* postao vidljiv i dostupan u lokalnoj mreži na bilo kom uređaju koji je deo iste mreže (Slika 1).



Slika 1. Lokalna mreža.

Nakon unosa korisničkog imena i lozinke koju smo kreirali može se pristupiti željenom direktorijumu. (Slika 2 i Slika 3).

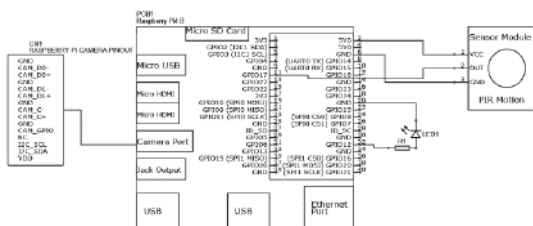


Slika 2. Unos šifre.

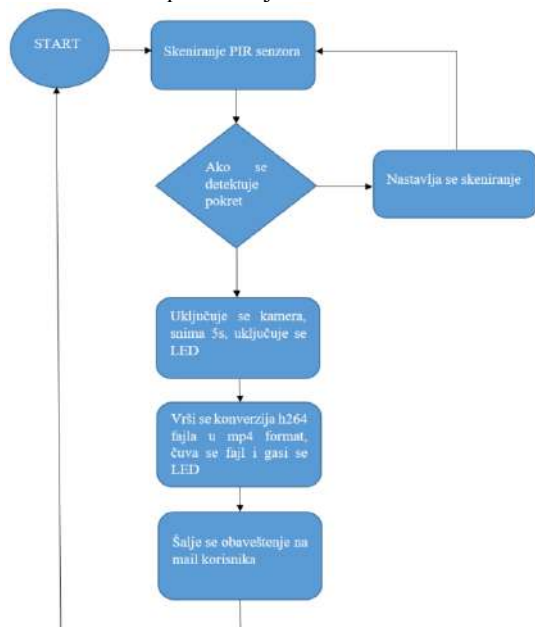


Slika 3. Pristup direktorijumu.

U nastavku je prikazana električna šema povezivanja PIR senzora, kamere i računara *Raspberry Pi* kao i algoritam.



Slika 4. Električna šema povezivanja.

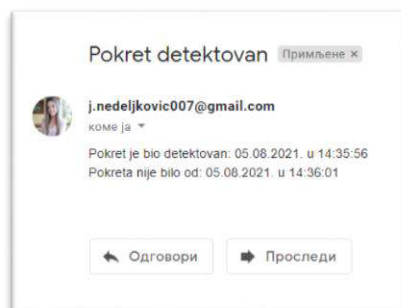


Slika 5. Algoritam.

U programskom okruženju, nakon pokretanja koda, ispisuje se sledeća poruka:

Pokret je detektovan: 05.08.2021. u 14:36:03
 Pocinje snimanje
 Cuvanje slike
 Nije bilo pokreta od: 05.08.2021. u 14:36:11
 Snimanje završeno i email se kreira
 To:j.nedeljkovic007@gmail.com
 From: j.nedeljkovic007@gmail.com
 Subject: Pokret detektovan
 Pokret je bio detektovan: 05.08.2021. u 14:36:03
 Pokreta nije bilo od: 05.08.2021. u 14:36:11

Nakon toga će korisnik biti obavešten putem *mail* poruke.



Slika 6. Obaveštenje korisnika putem *mail* poruke.

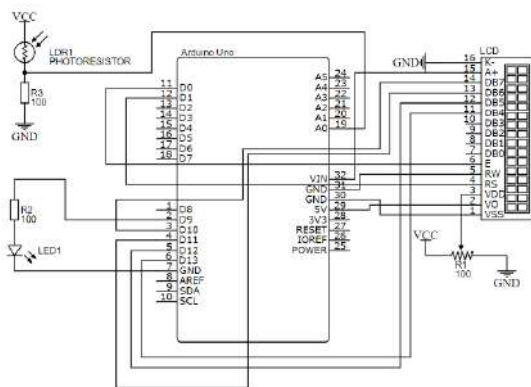
B. Podsystem za uključivanje LED i prikazivanje željenog teksta na ekranu

Funkcija ovog podsystema je ispisivanje teksta na ekran montiran na samom ulazu u kuću i uključivanje LED koja predstavlja spoljnu rasvetu. Unutar makete se nalazi fotootpornik koji je povezan sa razvojnim sistemom *Arduino*, dok je na računaru *Raspbbery Pi* povezana LED. U trenutku detektovanja pokreta *Raspbbery Pi* uključuje LED koja svojim svetlom smanjuje otpornost fotootpornika i na taj način obaveštava razvojni sistem *Arduino* da započne izvršavanje svojih instrukcija. Na ekranu se ispisuje korisnički odabran tekst i uključuje rasvetu. Prilikom povezivanja ekrana neophodno je koristiti potenciometar koji određuje kontrast ekrana.

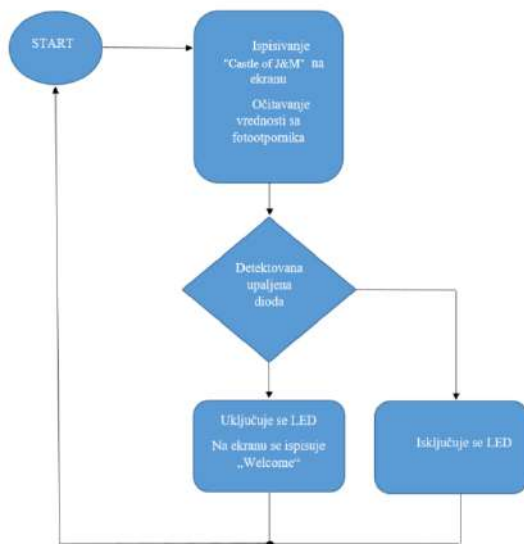
Za realizaciju ovog podsystema koristili smo:

- *Arduino Uno*
- Napajanje preko računara putem USB-a
- Ekran
- LED
- Fotootpornik
- Potenciometar

U nastavku je prikazana električna šema povezivanja ekrana, LED, fotootpornika i razvojnog sistema *Arduino* kao i algoritam.



Slika 7. Električna šema povezivanja.



Slika 8. Algoritam.

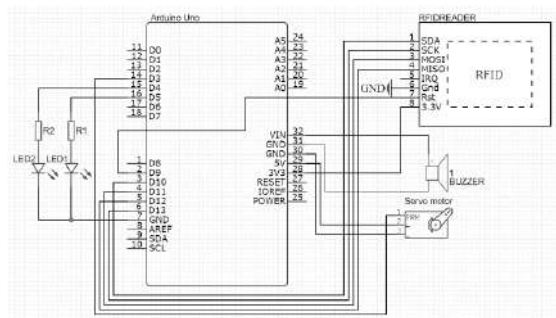
C. Podsystem za automatsko otvaranje vrata pomoću RFID čitača baziran na *Arduino* razvojnog sistemu

Funkcija ovog podsystema je realizacija automatskih vrata pomoću *Arduino Uno* razvojnog sistema. U cilju otvaranja vrata je korišćen servo motor. Postoje i dve LED koje se uključuju u zavisnosti od toga da li je pristup dozvoljen ili odbijen. Od dozvole ili zabrane zavisi i ton koji proizvodi zujalica.

Za realizaciju ovog podsystema koristili smo:

- *Arduino Uno*
- Napajanje preko računara putem USB-a
- Servo motor
- RFID senzor
- Dve LED
- Zujalica

U nastavku je prikazana električna šema povezivanja RFID čitača, LED, zujalice i servo motora sa *Arduinom*.



Slika 9. Električna šema povezivanja.

D. Hardware-ska realizacija i maketa projekta

Konkretno ovaj projekat sigurnosnog IoT sistema sa električnom bravom realizovan je povezivanjem tri podsystema u kompaktnu celinu.

Maketa koja je napravljena za ovaj projekat je realizovana tako da dočara primenu sigurnosnog IoT sistema za obezbeđivanje kuće i garaže.

Na prednjoj strani kuće se nalaze:

- kamera koja je povezana sa računarom *Raspberry Pi*;
- senzor pokreta koji je povezan sa računarom *Raspberry Pi*;
- LED koja je povezana sa razvojnim sistemom *Arduino*;
- ekran koji je povezan sa razvojnim sistemom *Arduino*.

Vrata garaže se otvaraju pomoću kartice i RFID senzora. U nastavku su prikazane fotografije makete koja je izrađena za ovaj projekat.



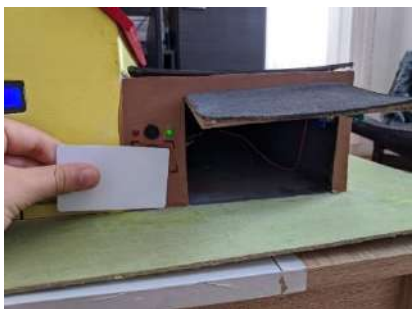
Slika 10. Maketa.



Slika 11. Unutrašnje povezivanja komponenata.



Slika 12. Unutrašnje povezivanje komponenata.



Slika 13. Otvaranje garaže.

III. ZAKLJUČAK

U modernom dobu se obraća posebna vrsta pažnje na zaštitu imovine. Bezbednost je postala prioritet. Kada kažemo bezbednost ne mislimo samo na bezbednost ljudi već i imovine, poslovnih postora, kuće, itd. Danas su provale i krađa česte pojave. Iz ovih razloga sve više ljudi dolazi do zaključka da je neki vid zaštite neophodan. Kada su u pitanju kuće i stanovi, najbolji način zaštite je ugradnja video nadzora i postavljanje alarmnih sistema. U ovom radu prikazan je primer korišćenja bezbednosnog sistema u vidu kamere i senzora koji je namenjen detektovanju pokreta. Takođe, prikazan je i sistem električne brave. Ovo je samo jedan od mnogih načina zaštite. Prethodno predložen i realizovan sistem se može montirati na svim vrstama objekata, bilo da je u pitanju poslovni objekat, porodična kuća ili zgrada.

ZAHVALNICA

Zahvaljujemo se našim prijateljima koji su nam ukazali na određene nepravilnosti prilikom pisanja ovog rada.

LITERATURA

- [1] Scott Fitzgerald and Michael Ahiloh, "The Arduino project book", Septembar 2012.
- [2] Michael Margolis, "Arduino Cookbook", March 2011.
- [3] Wiley Brand, "Arduino for dummies"
- [4] Derek Molloy, "Raspberry Pi", New York, 2016.
- [5] Cornel Armariei, "Arduino Development Cookbook", April 2015

Sistem automatskog proveravanja i održavanja mera protiv COVID-a 19

Lazar Mančić, Jovan Stanišić, Bojan Ćirić

Sadržaj – Ovaj rad predstavlja pregled sistema automatskog proveravanja i održavanja mera protiv COVID-a 19. Sistem ima tri osnovne funkcije, a one su: proveravanje da li osoba nosi masku, merenje temperature osobe i dezinfekciju osobe koja se približi prskalicu sa alkoholom za dezinfekciju. Proveravanje da li osoba nosi masku se vrši softverom za tu funkciju i kamerom koja snima osobu, merenje temperature se vrši uz pomoć senzora za merenje temperature i dezinfekcija se vrši uz pomoć senzora za merenje blizine i pumpice za prskanje alkohola. Prikupljeni podaci i slika sa kamere se prikazuju u interfejsu ovog sistema.

I. UVOD

Od početka pandemije COVID-a 19, do dana današnjeg, nažalost svakodnevno su se pojavljivale i još uvek se pojavljuju nove zaražene osobe, samim tim i novi smrtni ishodi. Čak i pored masovne vakcinacije, virus je još uvek među ljudima. Sve dok raste broj novozaraženih, uvedene mere se moraju poštovati. Kako se poštovanje mera mora proveravati to uglavnom rade zaposleni u nekoj instituciji, čime sebe izlažu riziku da se zaraze.

Ovaj sistem je stvoren upravo s ciljem sprečavanja takve vrste zaraze, odnosno kako bi se zaštitili oni koji proveravaju i održavaju mere. Sistem je osmišljen za primenu u školama, preciznije na ulazima u školama. Sve bi funkcionisalo pomoću kamere i par senzora, tako da bi se na ekranu računara prikazivalo da li neko ima masku ili ne, njegova trenutna temperatura i da li je prošao kroz dezinfekciju i time bi se učenik obavestio da li su sve mere ispoštovane. Naravno za održavanje pristupa u školi bi bila potrebna osoba, ali bi ona bila zaštićenija jer nije u direktnom kontaktu sa učenicima, već bi mogla preko ekrana da vidi da li učenika treba da pusti ili ne, izbegnuto je ručno merenje temperature i prskanje, odnosno direktan prilaz samim osobama koje to obavljaju.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

A. Princip rada sistema

Sistem funkcioniše jednostavno, kamera prenosi snimak uživo do računara koji softverom utvrđuje da li osoba nosi masku ili ne i rezultat provere se prikazuje na ekranu. Temperaturni senzor meri temperaturu učenika i

L. Mančić i J. Stanišić su učenici Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: mancic.lazar.181@tsp.edu.rs, stanicic.jovan.181@tsp.edu.rs

B. Ćirić je profesor Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: bojan.ciric@tsp.edu.rs

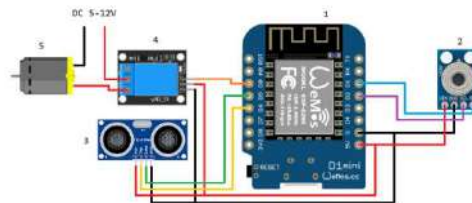
šalje podatke glavnom računaru koji ih prikazuje na ekranu sa ostalim informacijama. Da bi se učenik dezinfikovao, on mora prići do prskalice sa alkoholom i ispružiti ruke, senzor blizine utvrđuje da li je na dovoljnoj blizini. Ukoliko se prskalica aktivira, o tome se obaveštava glavni računar i na ekranu se prikazuje da je učenik dezinfikovao.

B. Hardver

Da bi sistem funkcionisao potreban je određen hardver, koji obuhvata sledeće komponente:

- PC računar / Laptop – Glavni računar koji obrađuje sve podatke i prikazuje ih u interfejsu na ekranu.
- Wemos D1 Mini – Mikrokontroler na kojem su priključeni svi senzori.
- HC-SR04 – Ultrazvučni senzor za merenje blizine.
- GY 906 – Senzor za merenje temperature.
- Pumpica – Pumpica za prskanje dezinfekcionog sredstva.
- PS5 HD Camera – Kamera koja prenosi snimak na osnovu kojeg se proverava da li osoba nosi masku.

C. Povezivanje komponentata na Wemos D1 Mini



Slika 1. Šematski prikaz povezanih komponentata na Wemos D1 Mini.

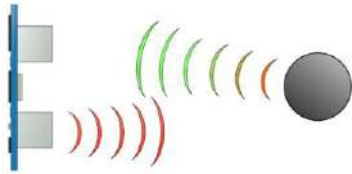
Wemos D1 Mini u ovom sistemu obavlja dve osnovne funkcije, a to je upravljanje dezinfekcijom i merenje temperature putem senzora temperature. Na slici 1 se može videti šematski prikaz povezanih komponenti, sa leve strane Wemos-a su komponente koje obavljaju dezinfekciju, dok sa desne strane je senzor za merenje temperature.

Komponenta broj 1 je sam Wemos D1 Mini, na koji su povezane sve komponente. Crvenim provodnikom je označena veza koja prenosi napon do komponentata, odnosno to je pozitivni kraj napona. Crnim provodnikom je prikazana veza do uzemljenja, ili negativnog kraja napona. Komponenta broj 2 je GY 906 senzor za merenje temperature. Komponenta broj 3 je HC-SR04 ultrazvučni

senzor za merenje blizine. Komponenta broj 4 je relej koji se koristi zajedno sa pumpicom. Komponenta broj 5 je pumpica za prskanje dezinfekcionog sredstva. Za pumpicu i relej je korišćeno napajanje od 5 V (na slici prikazano kao DC 5 – 12 V).

D. Ultrazvučni senzor za merenje daljine

U ovom projektu je korišćen ultrazvučni senzor za merenje daljine HC-SR04. Ovaj senzor koristi ultrazvučne talase kako bi izmerio udaljenost od njega do nekog objekta. Na sledećoj slici je ilustrovano emitovanje ultrazvučnih talasa i njihov povratak do senzora.



Slika 2. Prikaz principa rada ultrazvučnog senzora.

Za izračunavanje daljine, koristi se najjednostavniji oblik zakona puta u odnosu na vreme što se može videti na jednačini (1).

$$s(t) = v \cdot t \quad (1)$$

Jednačina (1) predstavlja opštu jednačinu puta, dok je kod ultrazvučnog senzora ona malo modifikovana. Za v se koristi brzina zvuka, t je vreme između emitovanja i detektovanja talasa, što znači da s koje se dobije je ustvari jednako dveju dužina između senzora i objekta. U jednačini (2) prikazane su vrednost brzine zvuka i izmerena vrednost vremena. U jednačini (3) su primenjene vrednosti iz jednačine (2) i izračunato je rastojanje modifikovanom jednačinom puta.

$$v = 340 \frac{m}{s} = 0,034 \frac{cm}{\mu s} \quad (2)$$

$$t = 294 \mu s$$

$$d = \frac{v \cdot t}{2} = 9,996 \text{ cm} \approx 10 \text{ cm} \quad (3)$$

E. Funkcionisanje Wemos D1 Mini i senzora

Wemos D1 Mini je programiran preko Arduino IDE, prema tome se može reći da je ovaj deo sistema programiran u C / C++ programskom jeziku. Wemos D1 Mini je povezan na Internet putem koga bi mogao da ubaci podatke u bazu podataka koju čita klijentska strana kako bi dobila podatke.

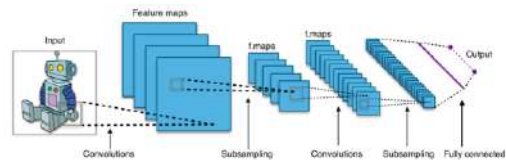
Podaci koji se upisuju u bazu su: temperatura ambijenta i temperatura osobe koja priđe, daljina osobe, koja se meri ultrazvučnim senzorom i da li je aktivirana prskalica ili ne. Kako se ne bi stalno čitala temperatura, ona će biti

detektovana tek kada se utvrdi da je osoba na dovoljnoj distanci i tada se temperatura upisuje u bazu, a u suprotnom se upisuje null vrednost za temperaturu. Distanca se konstantno upisuje u bazu. Vrednost da li je prskalica aktivirana ili ne upisuje se stalno u bazu, odnosno u početku je to vrednost 0, a kada se aktivira pumpica do resetovanja, upisuje se vrednost 1 u bazu podataka. Resetovanje se vrši kada se osoba skloni od senzora blizine, odnosno kada više nije blizu sistema.

F. Software za prepoznavanje da li osoba ima masku

Softver za prepoznavanje da li osoba ima masku je napisan u Python, programskom jeziku, sa korišćenjem određenih biblioteka. Programi ovakve vrste su uglavnom implementirani kroz algoritme mašinskog učenja, uključujući i ovaj softver. Algoritmi mašinskog učenja koji su korišćeni jesu - implementiranje veštačkih neuronskih mreža, odnosno algoritam dubokog učenja. U Python-u postoji gotova biblioteka za korišćenje tih algoritama koja se naziva Tensorflow. Pored Tensorflow-a, biblioteke koje su korišćene su Keras, imutils, numpy, OpenCV i Flask.

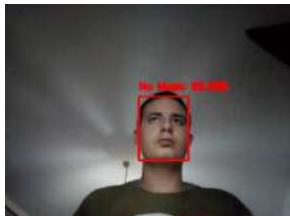
Keras biblioteka je korišćena radi modela koji se usaglašavaju sa Tensorflow-om. Imutils je biblioteka za uzimanje real-time prikaza slike sa kamere, koja se koristi za „posmatranje“ osobe. Numpy je biblioteka koja pruža olakšan rad sa nekim matematičkim algoritmima, poput množenja matrica, operacija nad istim i slično. OpenCV je korišćen za prepoznavanje lica i obradu podataka snimka koji stiže sa kamere. Flask je biblioteka korišćena za pisanje veb servera, on je korišćen da bi podatke sa serverskog dela prosledili interfejsu, odnosno frontend delu, koji je u slučaju ovog projekta zapravo veb stranica. Za prepoznavanje da li osoba nosi masku na licu se koristi konvolucijska neuronska mreža. Takve mreže se uglavnom koriste u projektima dubokog učenja sa obrađivanjem slika ili videa, klasifikovanju slika i sličnih algoritama.



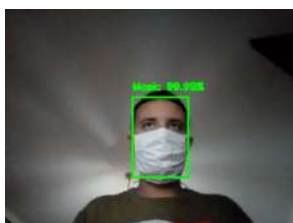
Slika 3. Prikaz rada konvolucijskih neuronskih mreža.

Ovaj sistem ima dva modela neuronskih mreža, prvi model je OpenCV model, koji se koristi da se na nekoj slici pronađu lica osoba, i drugi model Tensorflow model, koji se koristi za prepoznavanje da li na tom pronađenom licu postoji maska. Slika koja se prenosi sa kamere putem biblioteke imutils, u svakom trenutku, se OpenCV bibliotekom obrađuje da bi bila pogodna za prolazak kroz prvi model neuronske mreže, i zatim se istom prosleđuje. Pokreće se neuronska mreža i ona kao rezultat vraća niz sa licima, odnosno lokacije sa dimenzijama na kojoj se nalaze lica. Zatim se for petljom prolazi kroz niz koji je dao prvi

model, i za svako lice se pokreće drugi model, Tensorflow model, i rezultat svake predikcije se čuva u nizu. Oko svakog lica OpenCV modulom se crta zeleni ili crveni pravougačnik sa procentualnom predikcijom u vezi maske. Iako se to iscrta oko svakog lica, sistem će za podatke koje će prikazati u interfejsu gledati osobu koja je najbliže kameri, odnosno osoba čije lice ima najveći okvir. Na slikama 4 i 5 se može videti vizuelni rezultat rada softvera.



Slika 4. Rezultat koji daje software za lice bez maske.



Slika 5. Rezultat koji daje software za lice koje ima masku.

Iako je ovaj program u velikoj većini slučajeva tačan, moguće je da se dogodi greška prilikom prolaska kroz bilo koju od navedene dve neuronske mreže. Neki od faktora koji mogu uticati da prva OpenCV neuronska mreža da pogrešan rezultat, jeste udaljenost osobe od kamere, ugao pod kojim je kamera, da li osoba ima nešto na licu, ili čak ukoliko je maska koju osoba nosi prevelika može doći do toga da neuronska mreža ne može uočiti da je to zapravo lice. Tensorflow neuronska mreža može pogrešiti ukoliko se rukom prelazi preko lica, ili maske, boja maske može uticati na rezultat, kao i navedeni faktori koji utiču na tačnost OpenCV neuronske mreže, jer da bi se Tensorflow slika obradila ona se postavlja na određenu širinu i dužinu, što kod nekih „uhvaćenih“ lica može napraviti problem. Tačnost, tokom treniranja Tensorflow neuronske mreže za prepoznavanje maske na licu osobe se može videti na grafiku koji se nalazi na slici 6.



Slika 6. Preciznost i greška tokom treniranja neuronske mreže za prepoznavanje da li osoba nosi masku.

Na slici 6, crvena linija prikazuje grešku tokom treniranja, dok plava linija prikazuje grešku prilikom testiranja neuronske mreže. Ljubičasta linija prikazuje preciznost mreže prilikom treniranja, a crna (siva) linija prikazuje preciznost mreže prilikom testiranja (validacije). Epoch označava epohu, što je u mašinskom učenju broj ponavljanja treniranja tokom jedne „epohe“. Vrednost epohe za treniranu mrežu sa slike 6 je bila 20.

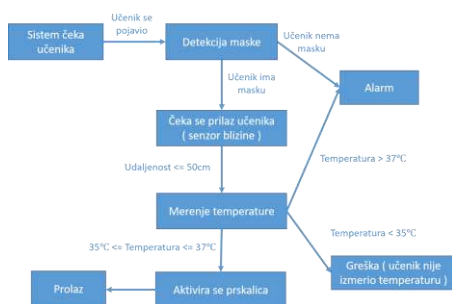
G. Korisnički interfejs i vizualizacija podataka

Korisnički interfejs je deo sistema koji se pojavljuje na ekranu glavnog računara ovog sistema. On je rađen kao klasična veb stranica, korišćeni su HTML, CSS i JavaScript, za samo stilizovanje stranice. Korisnički interfejs je namenjen osobama koje kontrolišu pristup, ali i samim osobama koje koriste sistem, zbog određenih poruka koje sistem daje. Podaci se čitaju iz baze podataka pomoću glavnog računara i pomoću istog se prikazuju u interfejsu, na ekranu tog računara. Ukoliko se on podesi da bude javni ili lokalni server, onda bi svako na javnoj ili lokalnoj mreži mogao pristupiti veb stranici sa interfejsom i videti ono što se dešava na ulazu, odnosno ovo pokazuje da se praćenje podataka i kamere ne mora vršiti samo preko glavnog računara. Ova karakteristika daje portabilnost sistemu, jer osoba ne mora da bude fizički pored sistema da bi videla šta se dešava sa njegovim podacima. Sam korisnički interfejs je veoma jednostavan, pošto se sastoji od samo jedne veb strane. Na njoj se nalazi heder u kome se nalaze naziv škole ili ustanove, naziv samog sistema i trenutni datum i vreme. Podaci koji se prikazuju jesu: da li korisnik ima masku, ambijentna temperatura, temperatura korisnika i da li je dezinfekcija izvršena. Pored toga postoje upozorenja, odnosno poruke, za korisnika sistema, i one se prikazuju pre aktiviranja prskalice i pre merenja temperature. Na slici 7 se može videti osnovni izgled korisničkog interfejsa.



Slika 7. Korisnički interfejs.

Podaci koji se prikazuju u korisničkom interfejsu stižu sa servera, odnosno sa backend strane. Na backend delu, da li korisnik ima masku dolazi iz ranije objašnjenog softvera, dok se podaci o temperaturi, dezinfekciji, blizini i slično, čitaju iz baze podataka koju ažurira Wemos D1 Mini. Kada se podaci učitaju iz baze, odnosno kada se dobije podatak o nošenju maske, po Request-Response, modelu oni se prosleđuju klijentskoj strani, odnosno korisničkom interfejsu.



Slika 8. Algoritamski prikaz funkcije sistema.

III. MOGUĆE PRIMENE SISTEMA

Iako je ovaj sistem prvenstveno namenjen za proveravanje i održavanje mera tokom aktuelne pandemije COVID-a 19, u budućnosti slična situacija može nastupiti, ali za razliku od COVID-a, tada to ne treba da iznenadi ljude, odnosno oni moraju biti spremni na takvu situaciju. Prema tome ovaj sistem pored sadašnjosti svakako bi mogao da se iskoristi i za primenu u budućnosti, ukoliko nesrećnim okolnostima dođe do potrebe za tim. Sistem je realizovan da bude korišćen u školi, ali on ne mora biti nužno korišćen u školama, već se može koristiti i u drugim institucijama u kojima su obavezne provere ovakve vrste, poput banaka, zgrada opštine, i drugih javnih objekata. Sam sistem nije preterano skup, odnosno veće institucije ga sa lakoćom mogu priuštiti, ali i one manje ne bi morale da izdvoje puno novca za samu implementaciju ovog sistema, što otvara veću mogućnost primene istog.

IV. MOGUĆE NADOGRADNJE SISTEMA

Ovaj sistem je sam po sebi veoma fleksibilan odnosno ima mesta za buduću nadogradnju istog. Jedna od najboljih nadogradnji uključuje povezivanje sistema sa sistemom pametnih vrata, i omogućavanje prolaska kroz ista samo u slučaju ispunjenja uslova maske, temperature i dezinfekcije. U tom slučaju, glavni računar, odnosno server, koji je trenutno PC, odnosno laptop računar, mogao bi se zameniti sa Raspberry Pi računarom i time bi se uštedelo dosta prostora, i sistem bi postao skroz automatizovan što znači da ne bi bili potrebni posmatrači, odnosno osobe koje kontrolišu pristup ulaska. Druga od mogućih nadogradnji je dodavanje softvera za prepoznavanje lica. Ovaj softver bi na osnovu lica osobe mogao da utvrdi njen identitet, ukoliko je zabeležen u bazi podataka. Na osnovu tog podatka moglo bi se tačno ustanoviti koja osoba ne nosi masku, nije izmerila temperaturu ili se nije dezinfikovala. Ovako nešto se može primeniti u školama, radnim institucijama, i sličnim objektima gde su poznati identiteti osoba koje će boraviti u njima.

Sistem proveravanja i održavanja mera protiv COVID-a 19 je kao što se moglo videti po stavkama III i IV veoma primenljiv i relativno jeftin, a pored toga i ima mogućnost za razvijanje u budućnosti. Ovaj sistem nije unikatan, već postoje ovakvi sistemi, koji vrše slične funkcije kao i ovaj. Neka od postojećih rešenja već imaju implementiran dodatak pametnih vrata ili neku drugu nadogradnju, ali neka postojeća rešenja nemaju neku funkciju koju ima ovaj. Detekciju maske praktično svi sistemi vrše pomoću konvolucijskih neuronskih mreža, ali zbog same njihove prirode svaki softver za detekciju može dati malo drugačiji rezultat zbog različitog treninga neuronske mreže. Zaključno sa svim prethodno rečenim, iako već postoje sistemi slični ovom, oni su uglavnom skuplji od ovog, što ovom sistemu daje potencijalnu komercijalnu primenu zbog svoje cene, pored njegove dobre praktične primene.

ZAHVALNICA

Zahvaljujemo se našem mentoru Bojanu Ćiriću i našoj školi, Tehničkoj školi u Pirotu, što su nam omogućili da učestvujemo na ovoj konferenciji i time steknemo nova znanja i iskustva u izradi ovog praktičnog projekta.

LITERATURA

- [1] Face Mask Detection - <https://github.com/balajisrinivas/Face-Mask-Detection>
- [2] Flask Documentation - <https://flask-doc.readthedocs.io/en/latest/>
- [3] Convolutional Neural Network - https://en.wikipedia.org/wiki/Convolutional_neural_network
- [4] OpenCV in Python - https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- [5] Wemos D1 Mini - https://www.wemos.cc/en/latest/d1/d1_mini.html
- [6] HC-SR04 Guide - <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- [7] GY 906 Tutorial - <https://www.nn-digital.com/en/blog/2019/11/09/interfacing-gy-906-mlx90614-non-contact-ir-infra-red-temperature-sensor-with-arduino/>
- [8] Corona Virus Disinfectant Tunnel Using Face Mask Detection and Temperature Monitoring - https://ieeexplore.ieee.org/document/9432387?fbclid=IwAR1xPEcWOIFh40Q2C_CGI6M140VroYcmBYhH9rysP0AX3EM4pVHPOKjBkNM
- [9] IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection - <https://www.sciencedirect.com/science/article/pii/S2666285X21000996>
- [10] Convolutional Neural Network 2 - <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>
- [11] Convolutional Neural Network 3 - <https://www.tensorflow.org/tutorials/images/cnn>

Realizacija sistema za kontrolu servo motora korišćenjem Nucleo razvojnog okruženja

Nikoleta Džunić, Rade Tošić

Sadržaj – U ovom radu biće opisan princip kontrole servo motora korišćenjem Nucleo razvojnog okruženja. Kontrola servo motora se vrši pomoću dva tastera kojima se pomera motor i to svaki taster pokreće motor za 20° u jednu od dve strane. Pomoću LED diode se pokazuje trenutna pozicija motora.

I. UVOD

Servo motori su rotacioni aktuatori koji se koriste u aplikacijama gde je potrebno upravljanje ugaonom pozicijom, brzinom i ubrzanjem. Prema tome imaju široko područje primene. Sastoje se od odgovarajućeg motora u kombinaciji sa senzorom koji daje informaciju o poziciji. Postoje i RC (Radio Control) servo motori, koji se najčešće koriste u aplikacijama sa daljinskim upravljanjem (mali roboti, daljinski upravljani automobili, avioni, brodovi itd). [1] U ovom radu biće prakazan jedan od načina za kontrolu servo motora korišćenjem ARM mikrokontrolerskog okruženja.

II. KOMPONENTE KORIŠĆENE PRILIKOM REALIZACIJE PROJEKTA

Za realizaciju projekta korišćene su komponente:

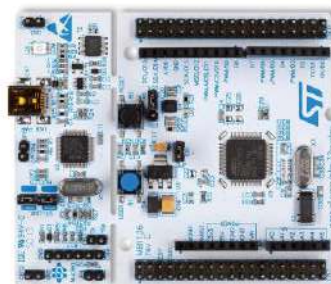
- Nucleo-F302R8 okruženje
- servo motor (SpringRC SM-S2309S)
- LED diode
- otpornici
- tasteri

Nucleo-F302R8 okruženje sadrži STM32F302R8T6 mikrokontroler [2] koji je baziran na Kortex-M4 jezgru. Može da radi na učestanostima do 72 MHz, dok se napon napajanja kreće između 2 V i 3.6 V, međutim najčešće se koristi standardnih 3.3 V. Ovaj mikrokontroler poseduje 51 ulazno-izlazni pin, 9 tajmera, dvanaestobitni analogno-digitalni konvertor ali i dvanaestobitni digitalno-analogni konvertor. Što se tiče komunikacije, ovaj mikrokontroler ima mogućnosti za ostvarivanje serijske (UART), I2C i SPI komunikacije, kao i mogućnost CAN i USB komunikacije.

N. Džunić je diplomirani inženjer elektrotehnike i računarstva, modul Elektronske komponente i mikrosistemi na Elektronskom fakultetu u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikoletadzunic@gmail.com.

R. Tošić je student osnovnih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: radetosic96@gmail.com.

Dakle, STM32F302R8T6 ima tri nezavisna porta za serijsku komunikaciju (U(S)ART portovi), tri nezavisne I2C periferije i dve nezavisne SPI periferije. Naravno, konektori za sve mogućnosti mikrokontrolera razvedeni su po razvojnoj ploči. [3]. Izgled razvojne ploče dat je na slici 1.



Slika 1. Nucleo-F302R8.

RC Servo motor poseduje tri izvoda za povezivanje, jedan upravljački i dva preko kojih se motoru dovodi napajanje. Upravljački signal se koristi za upravljanje ugaonom pozicijom izlazne osovine i najčešće je to PWM (*Puls-Width Modulation*) signal. Trajanje impulsa PWM signala određuje poziciju osovine. Dok god je upravljački signal prisutan, servo motor zadržava poziciju svoje izlazne osovine. Pri pojavi signala, motor rotira osovinu u smeru u kom će se najpre ostvariti zadata pozicija. Kada se pozicija ostvari, osovina se zaustavlja i zadržava poziciju osovine. Upravljački signal servo motora ima periodu od 20 ms, dok pozicija izlazne osovine zavisi od faktora ispune impulsa. [1] U zavisnosti od vrste servo motora različiti faktori ispune odgovaraju osnovnim uglovima servo motora. Za konkretni servo motor koji je korišćen prilikom realizacije ovog projekta uglu od 0° odgovara faktor ispune od 2.5 % dok uglu od 180° odgovara 12.5 %, što se može videti u tehničkoj dokumentaciji [4]. Na osnovu ovih podataka dobijeno je da uglu od 1° odgovara faktor ispune od 1.1 %. Izgled unutrašnjosti servo motora kao i njegov fizički izgled dati su na slici 2.



Slika 2. Izgled servo motora (spolja i iznutra).

LED (*Light Emitting Diode*) dioda je komponenta koja emituje svetlost pri odgovarajućem napajanju. Emisija svetlosti se dešava prilikom direktne polarizacije diode i ova pojava se naziva elektroluminiscencija. Napon vođenja LED diode koje emituju svetlost iz vidljivog dela spektra kreće se od 1.8 V (crvena LED) do 3.5 V (plava LED). Dok intenzitet svetlosti koju emituje LED diode zavisi od otpornika koji služi za ograničenje struje kroz samu diodu. Na slici 3 su prikazane LED diode koje su korišćene prilikom realizacije projekta.



Slika 3. LED diode.

Izgled tastera i otpornika koji su korišćeni prilikom realizacije projekta dat je na slici 4.

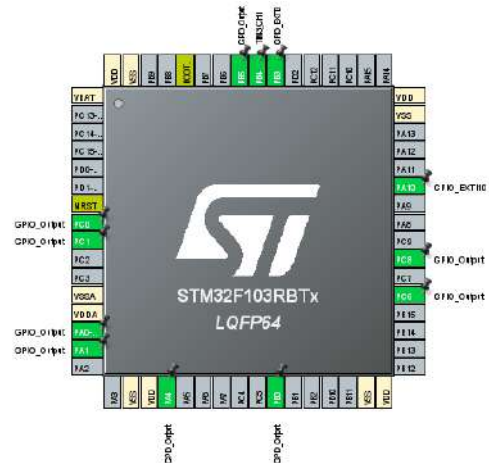


Slika 4. Otpornik i taster.

III. PODEŠAVANJE PERIFERIJA MIKROKONTROLERA

Pre same realizacije projekta i pisanje koda neophodno je u programu *STM32Cube MX*-u podesiti periferije koje će biti iskorišćene. Korišćenjem ovog programa se u velikoj meri olakšava kodiranje, zato što on generiše „kostur“ samog koda u programu *Keil uVision5*. U okviru „kostura“ na odgovarajućim mestima se kasnije pišu naredbe koda kojima se zadaje funkcionalnost projekta.

Kako bi se kontrolisao servo motor neophodno je označiti pin gde se on povezuje sa mikrokontrolerom. Preko ovog pina biće generisan PWM kontrolni signal, korišćenjem tajmerskog modula mikrokontrolera (TIM3 kanal 1). S obzirom da se ugao okretanja servo motora može da bude od 0 ° do 180 ° za indicaciju njegovog pomeraja od 20 ° potrebno je 9 LED dioda, koje se pale i gase. Prema tome neophodno je podesiti 9 pinova da budu izlazni (GPIO OUTPUT). Smer okretanja servo motora zavisi od toga koji taster se pritiska, tako da je potrebno podesiti i pinove za dva tastera. Ovi pinovi se podešavaju na eksterni *interrupt*. Podešeni pinovi u okviru *STM32Cube MX* programu su prikazani na slici 5.



Slika 5. Podešavanje pinova u *STM32Cube MX*-u.

Kada su pinovi podešeni potrebno ih je prilagoditi projektu. Da bi se aktivirali *interrupt*-i neophodno je obeležiti EXTI line 3 i EXTI line15-10. Takođe je potrebno podesiti tajmer tako da se na pinu PB4 generiše PWM signal. Perioda PWM signala kojim se kontroliše servo motor mora da bude 20 ms, jer on radi na frekvenciji od 50 Hz, tako da je potrebno podesiti još neke parametre tajmera, koji se odnose na PWM signal [5].

Na osnovu periode signala, izabranog *Counter Period* od 10 000 i izraza (1) za frekvenciju *prescaler*-a se dobija 500 kHz.

$$f_{\text{Prescaler}} = \frac{\text{Counter Period}}{T} \quad (1)$$

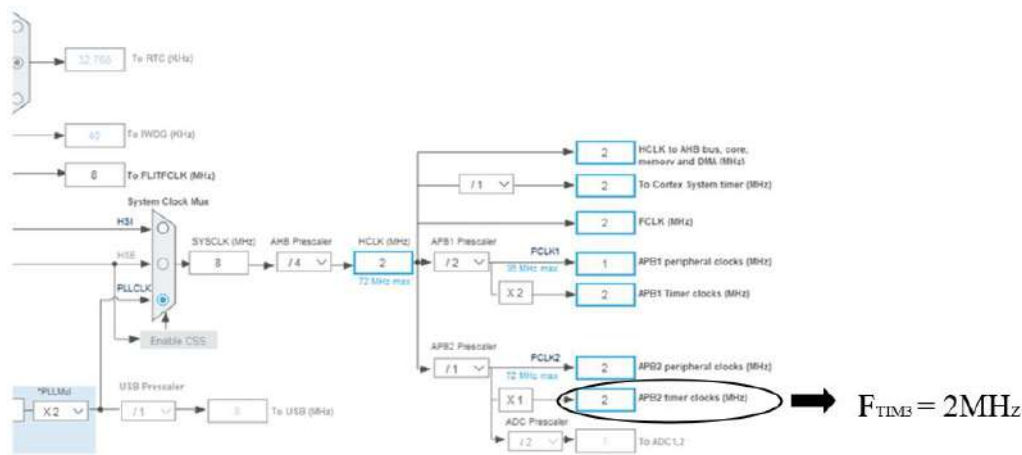
Za *Counter Period* je izabrana vrednost od 10000 kako bi se kasnije u registru upisivali celi brojevi. Da bi početni *Duty Cycle* bio 50% parametru *pulse* se dodeljuje vrednost od 5000, da bi se na osnovu izraza (2) dobila vrednost 50%.

$$\text{Duty Cycle} = \frac{CCR_x}{ARR_x} = \frac{\text{Pulse}}{\text{CounterPeriod}} \cdot 100\% \quad (2)$$

Konačno, korišćenjem vrednosti frekvencije tajmera, koja se može očitati u okviru prozora *clock configuration*, dobijene vrednosti frekvencije *prescaler*-a i izraz (3) dobijena vrednost preskarela je 4.

$$\text{Prescaler} = \frac{f_{\text{TIMs}}}{f_{\text{Prescaler}}} \quad (3)$$

Na slici 6 je prikazan izgled *clock configuration* prozora u okviru *STM32Cube MX*-a. Sa slike se može videti vrednost frekvencije tajmera kao i podešena vrednost *prescaler*-a.



uključujemo onoliko dioda kolika je vrednost promenljive pomeraj. Kod glavne programske petlje je prikazan u sledećem listingu:

```
/* USER CODE BEGIN 3 */
htim3.Instance->CCR1=duty;
switch(pomeraj){
case 0:
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_0,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_1,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_4,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_8,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_6,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_0,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_1,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_0,GPIO_PIN_RESET);
    break;
case 1:
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_0,GPIO_PIN_SET);
    ...
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_0,GPIO_PIN_SET);
    break;
}
/* USER CODE END 3 */
```

V. PRAKTIČNA REALIZACIJA PROJEKTA

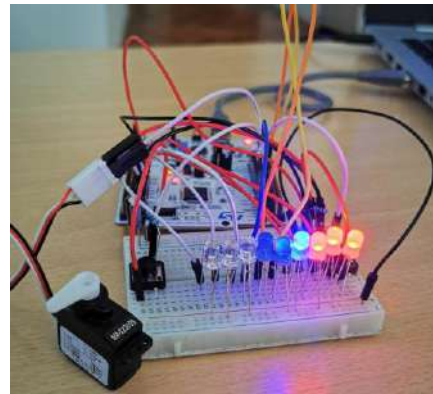
Pre same praktične realizacije potrebno je proveriti signal koji se generiše na pinu PB4 koji služi za kontrolu servo motora. Za ispitivanje signala umesto osciloskopa je iskorišćen logički Analizator. Prilikom provere signal je izgledao kao na slici 7, tako da se na osnovu njega moglo zaključiti da su parametri u *STM32Cube MX* dobro podešeni, jer je frekvencija signala 50 Hz, tako da na pinu imamo dobar kontrolni signal servo motora.



Slika 7. Izgled generisanog PWM signala.

Povezivanje komponenata u kolu prilikom praktične realizacije je moguće korišćenjem protoploče. Pomoću nje je moguće povezivanje komponenata sa *Nucleo* okruženjem bez lemljenja.

Za potrebe ovog projekta na protoploči su povezane devet LED dioda i dva tastera. Tasteri su povezivani preko *pull down* otpornika sa mikrokontrolerom, dok su LED diode sa mikrokontrolerom povezivane preko otpornika od 220 Ω . Servo motor je direktno povezan za mikrokontroler pomoću njegovih izvoda. Međutim kako bi servo motor bio stabilan bilo je neophodno paralelno sa napajanjem koje se dovodi sa mikrokontrolera povezati elektrolitski kondenzator od 10 μF . Prikaz praktične realizacije projekta dat je na slici 8.



Slika 8. Realizovani projekat.

Prvobitni položaj servo motora bio je na 80° jer je tako definisano inicijalizacijom promenljive *duty* na 690 zbog toga su u tom trenutku uključene četiri LED diode. Pritiskom tastera koji se nalazi sa leve strane motor se okreće u levu stranu, odnosno, postavi se u položaj od 100° i uključuje se još jedna LED dioda koja nam pokazuje novi položaj servo motora. Slična situacija se dešava kada se pritisne desni taster, u ovom slučaju se servo motor okreće u desno za 20° i smanjuje se broj uključenih LED dioda.

VI. ZAKLJUČAK

U ovom radu je prikazana jedna od mogućnosti realizacije kontrole servo motora. Data su objašnjenja o korišćenim komponentama i koda pisanog u okviru *Keil uVision5* programu za *Nucleo* okruženje, kao i način kako se podešavaju neke od periferija koje sadrži *Nucleo* okruženje pomoću programa *STM32Cube MX*. Opisani projekat praktično je testiran na protoploči.

ZAHVALNICA

Zahvaljujemo se profesorima sa predmeta Komponente i kola snage na ideji koju su nam dali za izradu ovog projekta.

LITERATURA

- [1] <https://www.automatika.rs/baza-znanja/mehatronika/rc-servo-motori.html>
- [2] <https://pdf1.alldatasheet.com/datasheet-pdf/view/1009255/STMICROELECTRONICS/STM32F302R8T6.html>
- [3] Nikola Mitrović, "Osnove rada sa ARM mikrokontrolerima"
- [4] <http://descargas.cetronic.es/microservo.pdf>
- [5] Miloš Marjanović, "Komponente i kola snage"

Projektovanje i realizacija sistema za praćenje kvaliteta vode

Nikoleta Džunić

Sadržaj – Izrada senzora za proveru kvaliteta vode koristeći hemijsku i optičku metodu. Hemijska metoda podrazumeva realizaciju senzora za merenje električne provodnosti vode, dok optička metoda predstavlja realizaciju senzora za merenje zamućenosti vode. Korišćenjem ovih senzora u kombinaciji sa uslugama koje nude pametni gradovi, ljudima se može podići svest o trenutnom zagađenju vode i samim tim će je manje zagađivati.

I. UVOD

Pametani grad je mesto gde se tradicionalne mreže i usluge čine fleksibilnijim, efikasnijim i održivijim uz upotrebu informacionih, digitalnih i telekomunikacionih tehnologija, kako bi poboljšali poslovanje u korist svojih stanovnika. Karakteristike pametnog grada uključuju uštedu energije i ekološku efikasnost, dok se inicijative pametnih gradova mogu koristiti za borbu protiv klimatskih promena i zagađenja vazduha i vode [1].

Voda čini 70 % površine Zemlje, međutim samo 1-2 % vode se može koristiti za piće. Iako smo u 21. veku, i dalje postoje zemlje čije stanovništvo nije u mogućnosti da sebi obezbedi higijenski ispravno vodosnabdevanje. Pametne vodovodne mreže u pametnim gradovima omogućavaju kontinualno praćenje kvaliteta pitke vode direktno u cevima. Kvalitet pitke vode može se proveriti pomoću hemijske i optičke metode. Hemijska metoda podrazumeva merenje električne provodnosti vode. Električna provodnost [2] je mera ukupne koncentracije rastvorenih soli u vodi. Optička metoda [3] podrazumeva merenje zamućenosti vode. Analiza zamućenosti je proučavanje optičkih svojstava zbog kojih se svetlost kroz vodu rasipa i apsorbuje, a ne prenosi u pravim linijama.

II. EKOLOŠKI STANDARD

Velike količine rastvorenih supstanci mogu sprečiti upotrebu vode za navodnjavanje i piće, pa je provodnost jedan od najvažnijih parametara kvaliteta neorganske vode. Provodnost vode se obično izražava u milisimensima po centimetru (mS/cm). Provodnost prirodne vode se kreće od 0.05 mS/cm - 1.5 mS/cm za slatkovodna jezera i potoke do oko 55 mS/cm za morsku vodu, dok se provodnost vode za piće kreće u rasponu od 0.2 mS/cm do 0.8 mS/cm [4].

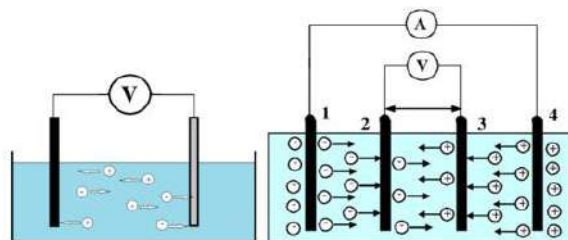
N. Džunić je diplomirani inženjer elektrotehnike i računarstva, modul Elektronske komponente i mikrosistemi na Elektronskom fakultetu u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikoletadzunic@gmail.com.

Drugi, ali ne i manje važan parametar kvaliteta vode jeste njena zamućenost. Zamućenost se javlja kada voda na svom putu pokupi razne vrste čestica. Zamućenost se najčešće meri u nefelometrijskim jedinicama zamućenja (NTU). Maksimalna vrednost zamućenosti pitke vode je 1 NTU, dok u vanrednim prilikama ova vrednost može biti 6 NTU [5].

III. MERENJE ELEKTRIČNE PROVODNOSTI

Postoje dve vrste merenja provodnosti [6]: kontaktna i induktivna. Izbor koja od ovih vrsta će biti korišćena zavisi od količine provodnosti i količine suspendovanih čvrstih materija.

Kontaktni senzori provodnosti poseduju sondu koja se sastoji od dve metalne elektrode. Primenom naizmeničnog napona na elektrodama, električno polje u vodi podstiče kretanje jona napred-nazad stvarajući struju. Merenjem struje i korišćenjem Omovog zakona na jednostavan način se može izračunati električna provodnost rastvora. Neki kontaktni senzori imaju četiri elektrode, gde se kroz spoljne elektrode propušta naizmenična struja, dok se na unutrašnjim elektrodama meri napon. Na slici 1 su prikazani principi kontaktnog merenja električne provodnosti.

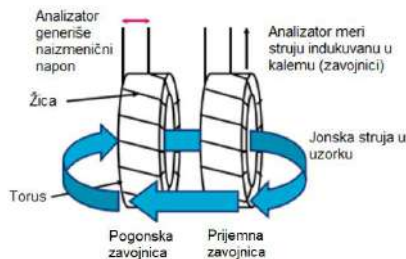


Slika 1. Princip kontaktnog merenja provodnosti.

Kontaktno merenje provodnosti može se realizovati pomoću otporne ili kapacitivne metode. Otporna metoda podrazumeva zamenu jednog od otpornika u šemi oscilatora realizovanog pomoću NE555 integrisanog kola, Vitstonovog mosta ili razdelnika napona sondom koja meri provodnost vode. Dok kapacitivna metoda podrazumeva merenje vremena punjenja i pražnjenja kondenzatora koji je zamenjen sondom.

Induktivno merenje provodnosti se ponekad naziva torusno merenje ili merenje provodnosti bez elektroda. Induktivni senzor sastoji se od dva metalna torusa

namotana u žicu čije je jezgro plastično telo otporno na koroziju. Jedan torus je pogonska zavojnica, dok je drugi prijemna zavojnica. Dovođenjem AC napona na pogonskoj zavojnici u vodi se indukuje napon koji uzrokuje pojavu struje koja se meri na prijemnoj zavojnici. Merenjem te struje možemo odrediti provodnost vode. Na slici 2 je dat princip induktivnog merenja električne provodnosti vode.

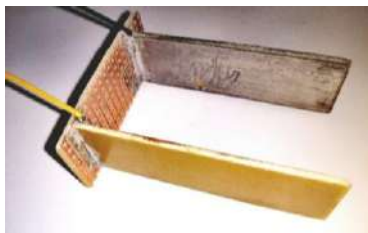


Slika 2. Princip induktivnog merenja provodnosti.

IV. KOMPONENTE KORIŠĆENE ZA REALIZACIJU HEMIJSKE METODE

Pored *Arduino Uno* razvojnog okruženja, LCD displeja, potencijometra i otpornika od $220\ \Omega$ za praktičnu realizaciju hemijske metode korišćena je i sonda za merenje provodnosti vode.

Sonda koja se koristi u praktičnoj realizaciji je samostalno napravljena. Sastoji se od dve elektrode koje predstavljaju bakarne pločice preko kojih je nanet sloj kalaja kako ne bi došlo do neke hemijske reakcije prilikom potapanja u vodu. Da bi pločice stajale na konstantnom rastojanju, od kog provodnost direktno zavisi neophodno je zalemiti pločice na raster ploču. Pored samih pločica na raster ploču se leme i kontakti koji služe za povezivanje sonde u kolo gde se primenjuje. Na slici 3 je dat izgled sonde za merenje električne provodnosti vode.

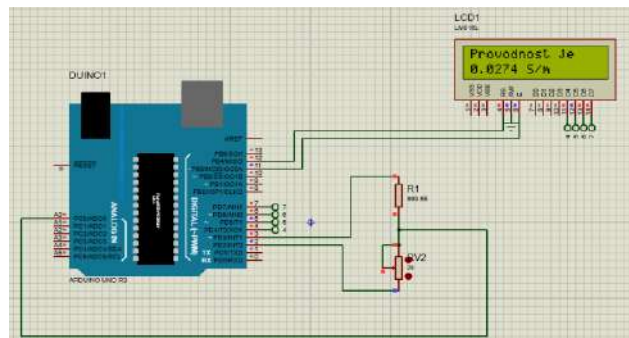


Slika 3. Sonda za merenje provodnosti vode.

V. SIMULACIJA I PRAKTIČNA REALIZACIJA HEMIJSKE METODE

Metoda koja je praktično realizovana jeste merenje provodnosti/otpornosti vode pomoću razdelnika napona. Kod ove metode se impulsno meri napon kako bi se izbegla elektroliza, ovako merenje je moguće menjanjem naponskih nivoa na krajevima kola naponskog razdelnika. U ovom primeru napon na krajevima se menja pomoću

mikrokontrolera svakih 10 ms. Pre same realizacije kola, neophodno je simulirati šemu, kako bi se proverila ispravnost metode. Simulacija je urađena u programu *Proteus 8 Professional* i na osnovu nje moglo je se zaključiti da metoda ispravno radi. Sonda je u simulatoru simulirana pomoću potencijometra od $2\text{ k}\Omega$ kako bi se obuhvatio celi opseg otpornosti vode po ekološkom standardu. Na slici 4 je prikazana šema iz simulatora.

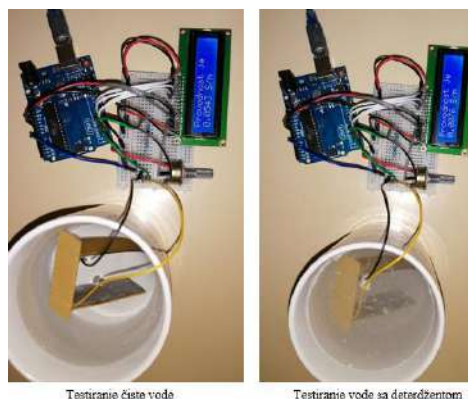


Slika 4. Šema iz simulatora.

Za povezivanje komponenta prilikom praktične realizacije korišćena je protoploča pomoću koje je moguće povezivanje komponenta sa Arduino okruženjem bez lemljenja. Komponente su međusobno povezane na protoploču, prateći šemu iz simulatora. Za razliku od simulacije prilikom praktične realizacije kola korišćene su standardne vrednosti otpornika (tri otpornika od $220\ \Omega$) i kratkospojnici za povezivanje.

VI. TESTIRANJE HEMIJSKE METODE

Testiranje je izvršeno u kućnoj varijanti. Voda koja je testirana nalazila se u keramičkoj šolji na sobnoj temperaturi. Najpre je sonda uronjena u vodu sa česme kada se na LCD-u ispisala vrednost provodnosti koja se nalazi u opsegu vode za piće. Kasnije je u vodi dodata određena količina deterdženta za pranje veša i provodnost se drastično povećala. Na slici 5 je prikazano testiranje čiste vode, kao i vode koja sadrži deterdžent za pranje veša.

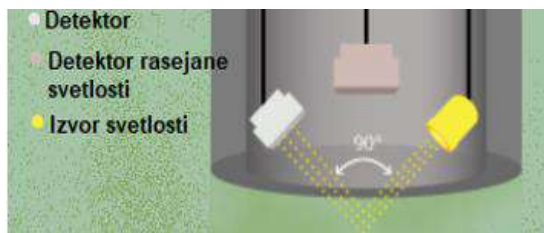


Slika 5. Testiranje hemijske metode.

VII. MERENJE ZAMUĆENOSTI VODE

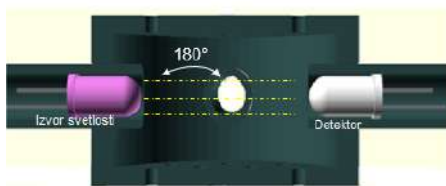
Sva merenja zamućenosti predstavljaju količinu svetlosti koja je propuštena ili rasejana od čestica u uzorku vode. Prema tome postoje dve glavne tehnike optičkog merenja zamućenosti koje se mogu kategorisati kao nefelometrija i tehnika zasnovana na apsorpciji [7].

Nefelometrija podrazumeva da su izvor svetlosti i fotodetektor postavljeni pod uglom od 90° , kao što je prikazano na slici 6. Smatra se da je ugao od 90° najosetljiviji na rasejanu svetlost bez obzira na veličinu čestice od koje se ona rasejava.



Slika 6. Nefelometrija.

Tehnika zasnovana na apsorpciji podrazumeva da su fotodetektor i izvor svetlosti postavljeni pod uglom od 180° , tako da praktično detektor detektuje samo svetlost koja je propuštena od strane čestica u uzorku, kao što je prikazano na slici 7.



Slika 7. Tehnika zasnovana na apsorpciji.

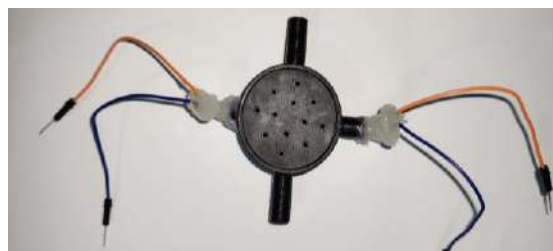
VIII. KOMPONENTE KORIŠĆENE ZA REALIZACIJU OPTIČKE METODE

Pored *Arduino*-a koji služi samo kao izvor napajanja od 5 V, za realizaciju ove metode neophodno je koristiti odgovarajuće otpornike, operacioni pojačavač *LM358N* kao i sondu za merenje zamućenosti vode.

Operacioni pojačavač *LM358N* je dizajniran da radi kao *single supply* operacioni pojačavač, tj. operacioni pojačavač koji se jednostrano napaja širokim opsegom napona. Granice napajanja *LM358N*-a se kreću od 3 V do 12 V, dok mu je maksimalno DC pojačanje 100 dB.

Sonda kod ove metode je takođe samostalno napravljena i sastoji se od IR LED diode koja predstavlja izvor svetlosti i fotodiode koja predstavlja detektor svetlosti. IR LED dioda koja je korišćena za izradu sonde je *LD274*, ona emituje svetlost talasne dužine 950 nm, dok je fotodioda *SFH213* čiji je opseg osetljivosti od 400 nm do 1000 nm. Kućište u kojem se nalaze diode je napravljeno

pomoću 3D štampača. Kućište sadrži niz rupa sa gornje i donje strane kako bi voda nesmetano mogla da uđe u senzor. Na slici 8 je dat izgled sonde.



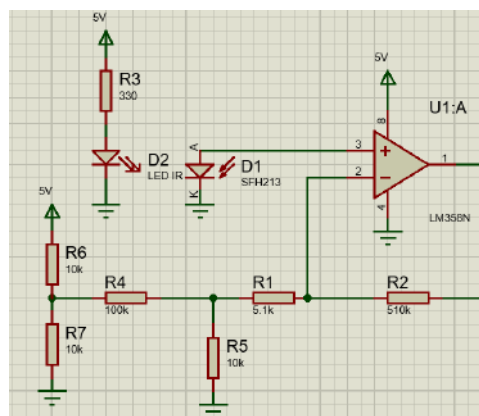
Slika 8. Sonda za merenje zamućenosti vode.

Pre samog 3D štampanja kućišta sonde neophodno je nacrtati njegov 3D model. 3D model je rađen u programu *OpenSCAD*, dok je priprema za štampu obavljena u programu *Ultimaker Cura*.

IX. ŠEMA I PRAKTIČNA REALIZACIJA OPTIČKE METODE

Šema optičke metode se sastoji iz dva dela. Prvi deo podrazumeva kolo za napajanje IR LED diode, dok drugi deo predstavlja pojačanje fotodiode. Za pojačanje fotodiode moguće je iskoristiti najjednostavniju konfiguraciju neinvertujućeg pojačavača, dok kolo LED diode sadrži samo IR LED diodu i otpornik koji ograničava struju koja protiče kroz diodu, tako da od veličine otpornosti otpornika zavisi osvetljenje koje će IR dioda emitovati. Međutim, korišćenjem šeme gde se pojačanje fotodiode realizuje pomoću neinvertujuće konfiguracije dobijeni rezultati prilikom testiranja nisu u potpunosti zadovoljavajući iz razloga što se opseg merenja napona kreće na malom intervalu, a i sam nagib krive koja opisuje pojačanje je vrlo mali.

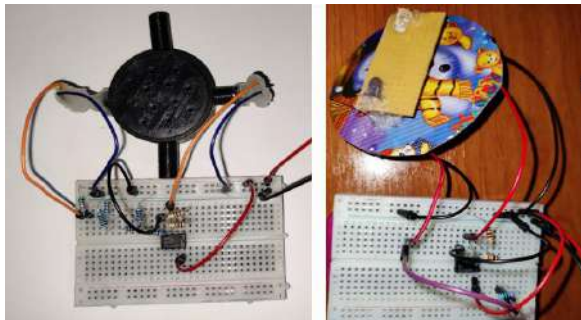
Da bi se to nadomsetilo neophodno je promeniti šemu, odnosno potrebno je dodati *offset* napon na invertujućem ulazu operacionog pojačavača čija vrednosti se oduzima od svake dobijene vrednosti na fotodiodi. Optimizovana šema je prikazana na slici 9.



Slika 9. Optimizovana šema.

Offset napon se kontroliše pomoću naponskog razdelnika i ne sme da bude veći od najmanje izmerene vrednosti napona fotodiode. *Offset*-om karakteristika zavisnosti izlaznog napona od broja kapi mleka se samo preslikava paralelno na nižem opsegu vrednosti, čime se omogućava dodatno pojačanje signala koji se dobija na fotiodi. Povećanjem pojačanja se dobija veći nagib, a samim tim i bolja karakteristika.

Prateći obe šeme praktično je realizovano kolo. Realizovana kola obe šeme data su na slici 10. Sa leve strane slike je dat izgled realizovanog kola sa *offset* naponom, dok se na desnoj strani nalazi kolo realizovano samo pomoću neinvertujuće konfiguracije operacionog pojačavača.



Slika 10. Praktična realizacija optičke metode.

X. TESTIRANJE OPTIČKE METODE

Testiranje optičke metode takođe je izvršeno u kućnoj varijanti pored senzora i vode za testiranje je bio potreban i špric kojim su se ubacivale kapi mleka u posudu sa vodom. Napon na izlazu kola je meren direktno na samom operacionom pojačavaču pomoću multimetra.

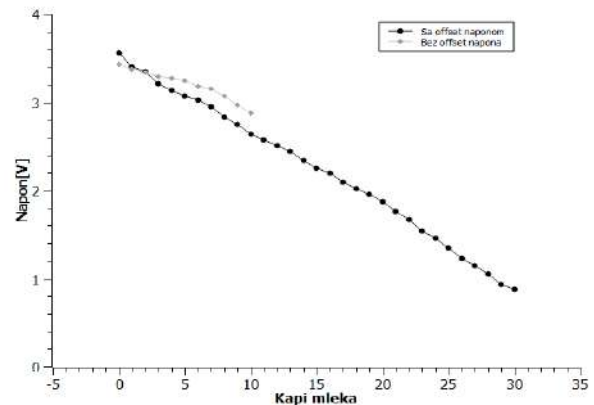
Prilikom izučavanja ove metode najpre je testirano kolo gde je pojačanje fotodiode realizovano pomoću neinvertujuće konfiguracije. Prilikom tog testiranja senzor je izgledao malo drugačije. Kućište nije bilo odštampano tako da ga je bilo potrebno improvizovati kako bi diode bile postavljene na isti način kao što je osmišljeno prilikom projektovanja kućišta. Slika improvizovane sonde je data na slici 11.



Slika 11. Improvizacija sonde.

Prilikom ovog testiranja izvršeno je 11 merenja najpre čiste vode, a zatim svaki put nakon dodavanja kapi mleka. Dok je prilikom testiranja kola sa *offset*-om izvršeno 31 merenje, iz razloga što je prilikom ovog testiranja posuda sa uzorkom vode bila veće zapremine. Zbog povećanja

zapremine posude, zamućenost se nije mogla uočiti golim okom kada se u vodu doda samo 10 kapi mleka zbog čega je nastavljeno sa dodavanjem kapi sve do neke prihvatljive vrednosti zamućenosti koje je vidljivo. Rezultati oba merenja su dati na grafiku koji je prikazan na slici 12.



Slika 12. Rezultati merenja.

Sa grafika se može videti da je šema sa *offset*-om pogodnija za ovaj senzor i da daje mnogo linearniji izlazni napon u poređenju sa šemom bez *offset*-a. Takođe u poređenju sa grafikom dobijenim testiranjem kola bez *offset*-a može se zaključiti da je nagib krive u ovom slučaju veći, a samim tim je veće i pojačanje.

XI. ZAKLJUČAK

Na osnovu rezultata koji su dobijeni prilikom testiranja oba realizovana senzora može se zaključiti da upotreba istih može pomoći u očuvanju pitke vode ukoliko se oni kombinuju sa inovacijama pametnih gradova.

ZAHVALNICA

Zahvaljujem se profesoru Ljubomiru Vračaru na pomoći tokom izrade projekta.

LITERATURA

- [1] <https://www.twi-global.com/technical-knowledge/faqs/what-is-a-smart-city>
- [2] <https://hal.inria.fr/hal-01061736/document>
- [3] Ahmad Fairuz Omar i Mohd Zubir Mat Jafri, "Dizajn i analiza turbidimetra: Pregled senzora optičkih vlakana za merenje zamućenosti vode"
- [4] [https://en.wikipedia.org/wiki/Conductivity_\(electrolytic\)#cite_note-2](https://en.wikipedia.org/wiki/Conductivity_(electrolytic)#cite_note-2)
- [5] <https://www.grf.bg.ac.rs/~vladana/files/Praktikum%202009.pdf>
- [6] <https://thecavepearlproject.org/2017/08/12/measuring-electrical-conductivity-with-an-arduino-part1-overview/>
- [7] <https://www.fondriest.com/environmental-measurements/measurements/measuring-water-quality/turbidity-sensors-meters-and-methods/#TurbidMM1>

Realizacija kapacitivnog senzora za merenje vlažnosti zemljišta

Ljubica Krivokapić

Sadržaj - U radu je dat pregled realizacije senzora za merenje vlažnosti zemljišta uz pomoć dve kapacitivne konfiguracije. Dat je i detaljan prikaz svih komponenti i kola uz primer rada sistema. Ispravan rad sistema praktično je proveren logičkim analizatorom.

I. KAPACITIVNI SENZOR ZA MERENJE VLAŽNOSTI ZEMLJIŠTA

Kapacitet se definiše kao količina naelektrisanja koji materijal može da uskladišti pod datim primenjenim električnim potencijalom. Obično se kondenzatori vizualizuju kao konfiguracije paralelnih ploča slično onom prikazanom na slici 1. U obliku jednačine, kapacitet se može napisati kao odnos između naelektrisanja i potencijala:

$$C = \frac{Q}{V} \quad (1)$$

gde C predstavlja kapacitet, Q je količina naelektrisanja V je količina naelektrisanja. Na slici 1. prikazan je koplanarni kondenzator, koji svoju izloženu površinu tretira kao dielektrični materijal, omogućavajući tako meru dielektrične konstante, koja je za vodu mnogo veća od većine zemljišta. Odnosi se na koplanarni kondenzator kod koga su elektrode kondenzatora jedna pored druge, a dielektrik se nalazi preko njih. Ovo omogućava senzoru da aproksimira sadržaj vode u zemljištu na osnovu promene u kapacitivnosti. Teoretska aproksimacija kapaciteta za postavku na slici 1: površinom, svojstvima dielektričnog materijala i rastojanjem, odnosno kapacitivnost kondenzatora zavisi od površine i rastojanja između elektroda kao i od dielektrične konstante sredine. Shodno tome, može se konstatovati da je dielektrična konstanta materijala direktno proporcionalna kapacitivnosti. Rezonantna učestanost, koja se može izmeriti je [1]:

$$f = \frac{1}{2\pi\sqrt{LC_t}} \quad (2)$$

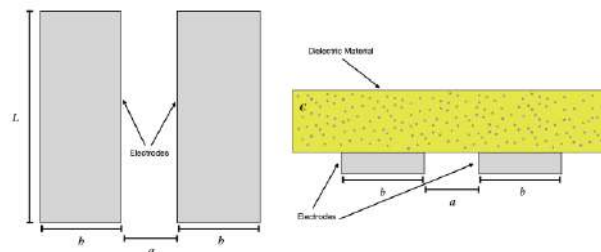
gde f predstavlja rezonantnu frekvenciju sistema, koja je žnsti merljiva, gde je L induktivnost sistema, a C_t ukupna

Lj. Krivokapić je student na modulu EKM, Katedra za mikroelektroniku, Elektronski fakultet Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: krivokapiccc@gmail.com.

kapacitivnost sistema. S obzirom da kapacitivnost sistema zavisi od učestanosti, samim tim se može zaključiti i da je dielektrična konstanta sredine funkcija rezonantne učestanosti, što se može prikazati u obliku:

$$\varepsilon = f(f) \quad (3)$$

I ovaj rezultat je primećen više puta: dielektrično svojstvo medijuma je funkcija rezonantne frekvencije kondenzatora. To je prvi pomen lako merljive veličine koja se odnosi na svojstva izloženog materijala (za razliku od zatvorene karakteristike standardnog kondenzatora).



Slika 1. Crtež koplanarnog kondenzatora koji prikazuje izloženi materijal kao dielektrik (zemlja + voda).

Ovo je princip rada kapacitivnog senzora vlažnosti zemljišta. Ova velika propusnost omogućava protok električne energije slobodno, što povećava kapacitet i rezultira nižom rezonancom. Zbog toga je kapacitet dobra mera vlažnosti zemljišta, zbog relativnog uticaja vode na električnu propusnost u materijalima prisutnim u zemljištu. Direktna kvantifikacija vlažnosti zemljišta nije jednostavan postupak.

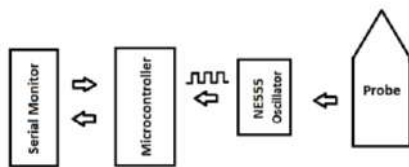
II. REALIZACIJA KAPACITIVNOG SENZORA

Kapacitivni senzor vlažnosti u osnovi se sastoji od para elektroda koje formiraju kondenzator sa zemljom koja se ponaša kao dielektrik između njih. Dve elektrode, zajedno sa zemljom kao dielektričnim materijalom, formiraju kondenzator. Generalno, što je veći sadržaj vode, veći je kapacitet. Dakle, merenjem kapaciteta, možemo odrediti sadržaj vode u zemljištu. Ovde korišćeni kapacitivni senzor vlažnosti koristi integrisano kolo tajmera 555 za pretvaranje njegove rezonantne frekvencije u analogni signal, koji će očitati Arduino ploča. Ovaj analogni signal će biti kalibrisan kako bi se stvorio

empirijski odnos između vlažnosti zemljišta i izlaznog analognog signala senzora. Arduino ploča koja se koristi je Arduino Nano, koja ima 10-bitni ADC koji radi na 5V. Kapacitivni senzor koji se koristi radi na 5V, a sa 10-bitnim ADC -om, radnim opsegom od 3.3V - 5V i opsegom merenja VWC (volumetric water content-volumetrijski sadržaj vode) od 0% - 100% - ovo daje VWC rezoluciju od oko 0,5%, što je donja granica rezolucije za podešavanje merenja. ADC veće rezolucije rezultiralo bi većom rezolucijom u merenju, ali 10 bita je dovoljno za trenutnu analizu [2].

III. PRVA KONFIGURACIJA KAPACITIVNOG SENZORA

Blok šema senzora data je na slici 2. Na slici 2. Možemo videti da nam je za razliku prve konfiguracije potrebni samo NE555 tajmer i elektroda koja predstavlja kondenzator.



Slika 2. Blok šema prve konfiguracije kapacitivnog senzora.

Tajmer NE555 koristi se kao oscilator koji bira signal senzora kondenzatora za celo kolo zasnovano na mikro kontroleru. Podrazumevano, oscilator radi u astabilnom režimu, neprestano menjajući niske i visoke izlazne signale. Učestalost uključivanja astabilnog oscilatora zavisi od vrednosti otpornika i kondenzatora spojenih na kolo stabilnog oscilatora. Dakle, kapacitivni senzor je glavni kondenzator u ovom kolu koji aktivno određuje frekvenciju oscilatora.



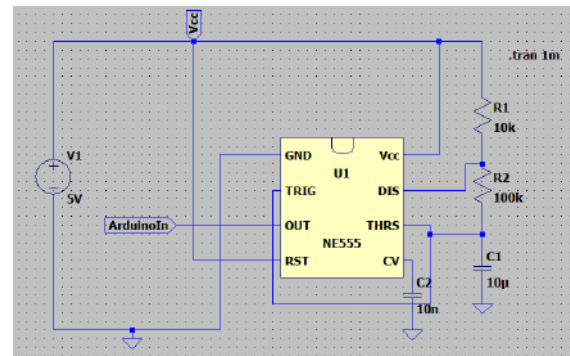
Slika 3. Elektroda kapacitivnog senzora.

IV. NAČIN FUNKCIONISANJA PRVE KONFIGURACIJE KAPACITIVNOG SENZORA

Funkcioniše tako što meri promena kapacitivnosti izazvanih promenama u dielektriku. Ne meri direktno vlažnost (čista voda ne provodi dobro električnu struju), već meri jone koji su rastvoreni u njoj. Na ove jone i njihovu koncentraciju mogu uticati brojni faktori, na primer dodavanjem đubriva će se smanjiti otpor zemljišta. Kapacitivno merenje u osnovi meri dielektrične osobine novonastale sredine koju stvori zemljište i voda je

najvažniji faktor koji utiče na dielektrik. Kapacitivno merenje ima neke prednosti, ne samo da izbegava koroziju sonde, već i daje bolje očitavanje sadržaja vlage u zemljištu, za razliku od upotrebe otpornog senzora vlažnosti zemljišta. Pošto kontakti (plus ploča i minus ploča kondenzatora) nisu izloženi zemljištu, nema korozije samog senzora [3].

Ova konfiguracija meri promenu frekvencije u odnosu na promenu kapacitivnosti kondenzatora, koji je u ovom slučaju realizovan pomoću Pertinaks pločice koja je izrezana u obliku vrha strele radi lakšeg zaranjanja u zemlju. Kako elektroda ne bi korodirala premazana je sa obe strane debelim slojem farbe koja u ovom slučaju služi kao izolator.

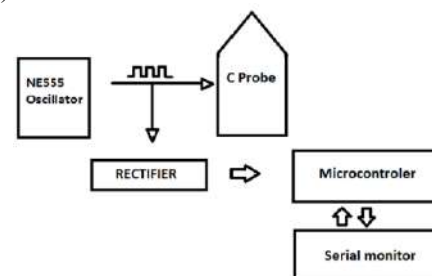


Slika 4. Električna šema prve konfiguracije kapacitivnog senzora u LTspice-u.

Takođe su i konektori prefarbanani do neke dužine kako dubina uranjanja elektrode ne bi uticala na konačne rezultate, već je sasvim svedeno do koje dubine uranjamo elektrodu u zemljište za razliku od komercijalnih kapacitivnih senzora kojih ima na tržištu i kod kojih je tačno obeležena dubina [4].

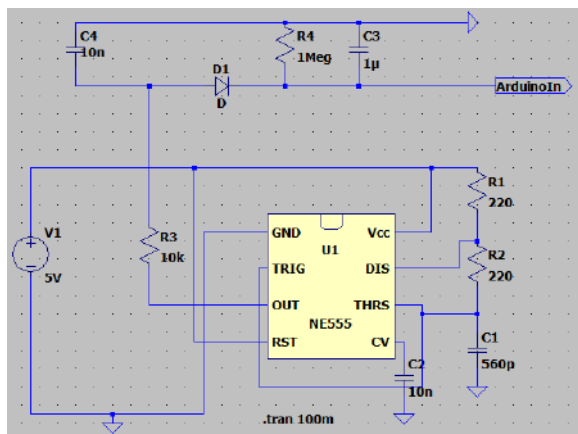
V. DRUGA KONFIGURACIJA KAPACITIVNOG SENZORA

Druga konfiguracija koju obrađujemo u ovom radu je analogni kapacitivni senzor vlažnosti, gde umesto frekvencije merimo pad napona u odnosu na promenu vlažnosti zemljišta. Izvor napona od 5 V napaja NE555 tajmer čiji izlazni signal napaja niskopropusni filter (otpornik od 10 kΩ i ravanski kondenzator koji detektuje vlažnost).



Slika 5. Blok šema druge konfiguracije kapacitivnog senzora.

Napon vrha do vrha (peak to peak) talasnog oblika zavisi od efektivne dielektrične konstante zemlje. Detektor vršnog napona daje analogni izlazni signal koji dobijamo preko ADC-a mikrokontrolera. Tokom električne karakterizacije sa senzorskom sondom u vazduhu primetili smo da kada se uvede sonda osciloskopa između otpornika od 10 kΩ i diode, tada se izlazni napon jako menja, što ukazuje na to da 1μF i 10MΩ sonde značajno menjaju ponašanje kola.



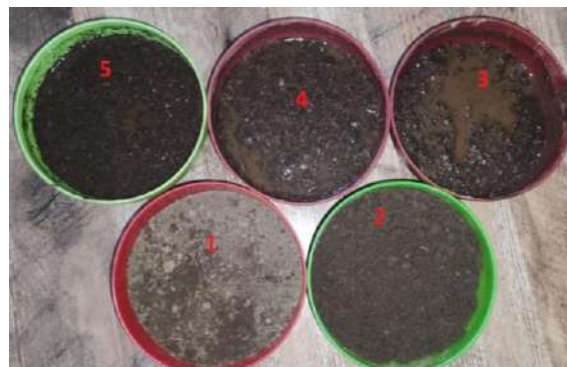
Slika 6. Šema druge konfiguracije kapacitivnog senzora.

Komponenta CPROBE sa slike 4 je takođe korišćena i u ovoj realizaciji. Izlazni signal NE555 je pravougaonog talasnog oblika koji radi na oko 1,9 MHz teorijski, ali u praktičnoj realizaciji izlazna frekvencija je podešena na 300 kHz. Radni ciklus ovog pravougaonog signala iznosi oko 33% Duty cycle (radni ciklus NE555 uvek treba da bude veći od 50%) verovatno su uzrokovani blizinom radne frekvencije do fizičke granice frekvencije za uređaj NE555. Ispravljač je električni uređaj koji pretvara naizmeničnu struju (AC), koja periodično menja smer, u jednosmernu struju (DC), koja teče samo u jednom smeru.

VI. KALIBRACIJA SENZORA

Kalibracija senzora izvršena je gravimetrijskom metodom. Vlažnost zemljišta se može aproksimirati tako što se izdvoji 5 posuda sa istom količinom zemljišta svaka posuda treba da ima po 500 g suvog zemljišta. Odvojena 2,5 kg zemlje se prvo osuši u rerni na preporučenih 105 °C, na plehu gde je na raspoređena jednako ka svim stranama, i peče se sve dok se zemlja potpuno ne osuši. Nakon toga se prvenstveno izmeri težina svake prazne činije ponaosob i zapiše se u tabeli, nakon toga se u svakoj činiji dosipa 500g suve ispečene zemlje. Kada je svaka činija napunjena, prvo se izmere parametri sa oba senzora u 100% suvoj zemlji i upišu u tabelu. Nakon toga se u svakoj činiji dolije po određena količina vode koja predstavlja tačan procenat vlažnosti, počevši sa 20% što na 500 g vode iznosi 100ml vode, pa na 40% što iznosi 200 ml vode, i tako sve do 100% vlažnosti što iznosi 500ml vode[5]. Na slici 7 se

nalaze činije sa zemljom korišćenje za merenje vlažnosti zemljišta.



Slika 7. Činije sa različitim procentima vlage.

$$\text{Soil moisture (\%)} = \frac{\text{weight of water contained in the soil}}{\text{weight of dry soil samples}} \times 100\%$$

$$\text{Soil moisture (\%)} = \frac{(b - a) - (c - a)}{(c - a)} \times 100\%$$

Parametar *a* predstavlja težinu same činije, parametar *b* predstavlja težinu mokrog zemljišta sa činijom i parametar *c* predstavlja težinu suvog zemljišta i činije. (1=20%, 2=40%, 3=60%, 4=80%, 5=100%). Takođe svaka činija sa određenom količinom vlage se takođe meri na vagi, kako bi se sve te vrednosti kasnije iskoristile za volumetrijsku računici, da bi matematički dobili procenat vlažnosti u zemljištu [6].

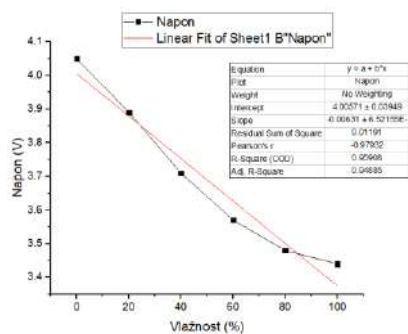
TABELA I
GRAVIMETRIJSKA METODA PROVERE KOLIČINE VLAŽNOSTI U DATIM UZORCIMA

1	Procenat vlage	Težina činije	Težina suvog zemljišta	Težina vlažnog zemljišta	Sm(%)
2	0%	60g	557g	557g	0%
3	20%	60g	557g	655g	18.70%
4	40%	60g	560g	755g	39%
5	60%	60g	565g	857g	57.80%
6	80%	60g	565g	952g	76.60%
7	100%	60g	564g	1040g	94.40%

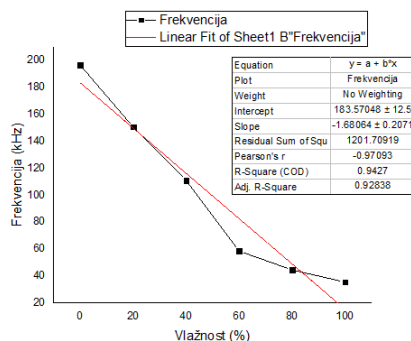
TABELA II
REZULTATI DOBIJENI IMPLEMENTACIJOM OBA SENZORA

1	0%	20%	40%	60%	80%	100%	0
2	%	195.600	150.800	115.500	58.800	68.600	35.930
3	V	4.05	3.89	3.71	3.57	3.48	3.64

Nakon svakog dodavanja vode, sačeka se 30 min pa se promeša sadržaj činije u kome je dosipana voda kako bi se ona lepo slegla i rasporedila, takođe voda se ravnomerno sipa po celoj površini zemlje kako bi se ravnomerno i upila. Napomena o vremenu potrebnom za merenje validnih rezultata: kapacitivnom senzoru vlažnosti može biti potrebno izvesno vreme da se izjednači i daje stabilno očitavanje, pa sačekajte približno 1 minut nakon što se senzor smiri na datu vrednost. Trebalo bi da se slegne za 1-5 minuta, u zavisnosti od nivoa zasićenosti zemljišta i od toga koliko je dobro izmešano vlažno zemljište. Pod pretpostavkom uspešne kalibracije, podaci se mogu uzeti i analizirati. Na slici 8 i 9 su dobijeni grafici za rezultate prve i druge konfiguracije kapacitivnog senzora.



Slika 8. Rezultati prve konfiguracije kapacitivnog senzora.



Slika 9. Rezultati druge konfiguracije kapacitivnog senzora.

VII. ZAKLJUČAK

Iz datih rezultata vidimo da su rezultati digitalnog kapacitivnog senzora, kod koga se u odnosu na promenu vlažnosti zemljišta menja frekvencija precizniji u odnosu

na analognu verziju senzora. Ovo daje korisnicima mogućnost da odrede sadržaj vode u zemljištu merenjem težine prema suvom uzorku zemljišta - dobijajući trenutnu aproksimaciju sadržaja vlažnosti u zemljištu. Napon je zatim linearno koreliran sa aproksimacijama gravimetrijske vlage, kako bi se dobio efikasan odnos između očitavanja sa kapacitivnog senzora i sadržaja vode u tlu. Ova procedura kalibracije pokazala je da su jeftini senzori vlažnosti zemljišta kapacitivnog tipa sposobni da predvide nivo vode u zemljištu sa visokim stepenom tačnosti, sa malo zahteva izvan samog uređaja, što je u direktnoj suprotnosti sa vremenom koje je potrebno da se tradicionalno meri sadržaj vode u zemljištu.

LITERATURA

- [1] Calibration of Capacitive soil moisture sensor (SKU:SEN0193)
- [2] World Agriculture: Towards 2015/2030, An FAO perspective, edited by Jelle Bruinsma, Earthscan Publications Ltd London, 2003.
- [3] Food and Agriculture Organization of the United Nations (FAO). Crops and Drops - Making the Best Use of Water for Agriculture; FAO: Rome, Italy, 2002.
- [4] H Eller and A Denoth. A capacitive soil moisture sensor. Journal of Hydrology, 185(1- 4):137–146, 1996.
- [5] Đ. Bošnjak, Metode istraživanja i određivanja fizičkih svojstava zemljišta, Jugoslovensko društvo za proučavanje zemljišta, Novi Sad, Srbija, 1997.
- [6] R.B. Grisso et al, Calibration and validation of a low-cost capacitive moisture sensor to integrate the automated soil.

Uređaj za kontrolu budnosti vozača Stay Awake

Veljko Aleksić, Aleksandra Aleksić, Predrag Šubarević

Sadržaj - Prema studiji iz 2008. godine koju je sproveo Švedski nacionalni institut za istraživanje bezbednosti na putevima, oko 20 odsto svih saobraćajnih nesreća uzrokovano je umorom vozača. „Umor je posebna tema za one kojima je vožnja deo profesionalnih obaveza“, kaže Džoša Haris, direktor kampanje za bezbednost na putevima u Velikoj Britaniji. „Ovi vozači često su za volanom duži vremenski period, a neretko upravljaju masivnim vozilima koja mogu prouzrokovati veliku štetu u slučaju sudara. Tehnologija igra važnu ulogu u težnjama da se postigne bezbednost na putevima i pozdravljamo razvoj sistema koji poboljšavaju bezbednost“, kaže on. Prema studijama NHTSA - Fatality Analysis Reporting System, u državi Teksas, 2016. godine, 22,1% saobraćajnih nesreća sa fatalnim ishodom potiče od „uspavanog“ vozača. Od toga čak 1/3 čine čeonni sudari sa drugim vozilom.

I. UVOD

Gore su samo neke navedene studije koje govore o opasnosti pospanog vozača na putu. Kao što možemo zaključiti iz studija, procenat umornih vozača na putu koji će izazvati saobraćajnu nesreću sa smrtnim ishodom, učestvuje sa udelom od oko 20% svih nesreća na putevima. Zanimljiv je i taj podatak da je od tog procenta čak 87% vozača poštovao dozvoljeni vremenski opseg u vožnji, dakle nije prekoračio dozvoljeno vreme vožnje pred odmor.

Ovaj procenat je dovoljno bio zabrinjavajući da bi nas podstakao na detaljno proučavanje ovog problema i realizaciju uređaja Stay Awake. Kao što samo ime govori (ostani budan) Stay Awake je uređaj koji će na vreme upozoriti vozača da je umoran, i da mu je hitno potreban odmor.

Najnoviji u nizu izveštaja SZO o Globalnom stanju bezbednosti na putevima ukazuje na to da saobraćajne nezgode godišnje odnesu živote preko 1350000 ljudi.

Na svake 24 sekunde jedna osoba izgubi život u saobraćajnim nezgodama na putevima i neko ostane bez oca, majke, sestre, brata, deteta...

Da bismo realizovali kvalitetan uređaj moramo se pozabaviti suštinom problema.

Mnogi vozači koji su zaspali za volanom u toku vožnje, priznaju da nisu to ni osetili. Psihijatri tvrde (Fraunhofer Institute for Systems and Innovation Research ISI) da san dolazi polako i neprimetno, pogotovu kad mu je saveznik umor vozača. Razgovor sa suvozačem ili

pomoćnim vozačem, može biti od pomoći što se tiče budnosti ali to uzrokuje smanjenu koncentrisanost na vožnju, slično razgovoru telefonom. Najčešći vinovnici (krivci) nesreća na putu, usled sna za volanom, su profesionalni vozači (vozači kamiona i autobusa). Često su njihove relacije duge i zahtevaju višednevnu vožnju. Da bi autoprevoznika preduzeća imala veću efikasnost pribegavaju modelu da na put krenu dva vozača kako bi jedan uvek bio sposoban za vožnju dok drugi odmara do naredne smene.

Psihijatri tvrde da je i to čak jedan od uzroka usnulog vozača. Naime, kako tvrdi Berit Brogaard, (D.M.Sci., Ph.D., profesor filozofije i direktor na Brogaard Lab for Multisensory Research na univerzitetu u Majamiju) prisustvo usnulog kolege nas takođe podstiče na san. Prema Metju Vokeru (engleski naučnik i profesor neuronauke i psihologije na Kalifornijskom univerzitetu u Berkliju), autoru knjige „Zašto spavamo“, 1,2 miliona nesreća je prouzrokovano pospanošću svake godine u Sjedinjenim Državama, a „nesreće izazvane pospanom vožnjom premašuju one izazvane alkoholom i drogom zajedno“.

Kako zapravo to počinje?

Prema ispovesti vozača, koji su imali iskustva sa snom u toku vožnje, prvo što su primetili je bilo to da se ne sećaju detalja nekog dela puta koji su prošli u zadnjih pet do deset minuta. Nakon toga se javlja peckanje u očima ili delimični pad kapaka. Ova situacija dovodi do podsvesnog spuštanja očnih kapaka vozača i kratke usnulosti. Ovaj fenomen nazvan je „mikrospavanje“, a to su kratki, nevoljni periodi nepažnje koji traju samo nekoliko sekundi. Iako ove epizode sna mogu biti kratke i do 5 sekundi, put koji pređe vozač svojim vozilom pri dozvoljenoj brzini na auto-putu može biti preko 150m.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA STAY AWAKE UREĐAJA

Način rada Stay Awake uređaja je taj da prati baš očne kapke vozača i primeti čak i delimičnu „zatvorenost“ oka. Istraživanjem konkurentnih uređaja zaključili smo da većina njih prati prvo satnicu vožnje, zatim učestalost u danima, pa na osnovu toga procenjuju umor vozača. Istraživanja AAA (American Automobile Association) pokazuju da to ne mora da bude presudan faktor jer je velika većina njih zaspala za volanom u periodu kad je bilo dozvoljeno da voze. Naime, većina vozača, vreme za odmor vrlo često ne koriste na pravi način, već odlaze u restoran ili su čak primorani da učestvuju prilikom istovara ili utovara robe. Sve ovo skraćuje vreme za san. Iz ovoga jasno zaključujemo da matematičkim izračunavanjem ne

V. Aleksić i A. Aleksić su učenici III razreda ETŠ „Nikola Tesla“, Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: info@etstesla.ni.ac.rs

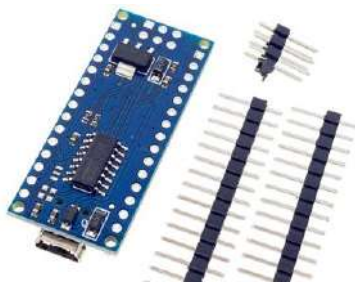
P. Šubarević je mas. inž. mehatronike i upravljanja, ETŠ „Nikola Tesla“, Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: info@etstesla.ni.ac.rs

možemo predvideti umor vozača. Pored ovoga i uslovi vožnje doprinose umoru, tako da ako je vozač dugo izložen direktnom suncu to znatno povećava umor kao i noćna vožnja. Dakle postoji niz faktora koji se ne mogu predvideti, a veoma su bitni.

Druga vrsta konkurentnih uređaja radi na principu "uspravnog" držanja glave. Ukoliko glava vozača klone i ugao vrata se savije od 15° do 20° uređaj reaguje. Ovde se jasno vide mnogi nedostaci. Da bi glava klonula oči moraju pre toga biti zatvorene, a vozač je već usnuo. Onda kad je glava već klonula, verovatno je kasno za buđenje. Vrlo čest slučaj je da profesionalni vozači spavaju zavaljeni u stolici restorana "uspravne" glave. Pored toga to zavisi i od trenutnog položaja tela vozača, glava može biti udobno naslonjena tako da ne mora klonuti i tako aktivirati uređaj.

Treća vrsta konkurentnog uređaja radi na principu snimanja lica vozača i praćenja očnih kapaka pomoću kamere. Ovo rešenje je najbliže našem konceptu, jer prati direktno očne kapke. Nedostatak ovog uređaja je taj što je uređaj instaliran ili na kontrolnoj tabli ili na vetrobranu vozila, tako da sa relativno velike daljine prati lice vozača. Problem u radu i preciznosti ovakvih uređaja, čini često zakretanje glave vozača radi praćenja retrovizora, saobraćajnih znakova ili pak skretanja sa tranzita. Udaljeno sočivo kamere u takvim slučajevima gubi potreban ugao snimanja, što dovodi do grešaka u radu uređaja. Sam algoritam rada ovakvih uređaja, iziskuje obradu velikog broja podataka i time povećava mogućnost greške.

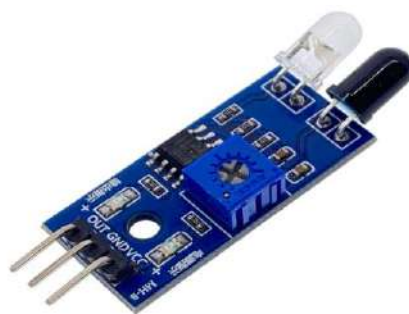
Naš uređaj je instaliran na glavi vozača po principu Hi-Fi slušalica. Težina je 78 gr. Tako da je lagan i ne predstavlja nikakav problem vozaču kod dugotrajnog nošenja. Izrađen je od PET-G plastike i neškodljiv organizmu, a senzor je podešen da stoji na samoj ivici oka i svojom pozicijom ne ograničava vidno polje vozača, čak ni kod perifernog vida. Napaja se punjivom baterijom od 9V. Uređaj koristi kontroler Arduino Nano AT mega 328P (slika 1).



Slika 1. Kontroler Arduino NANO.

Ovo je kontroler veoma malih dimenzija i pogodan je za primenu u ovakvim uslovima gde smo ograničeni gabaritom i težinom uređaja. Na sebi ima 14 digitalnih i 8 analognih pinova. Radi u taktu od 16MHz, a napajanje se vrši u opsegu od 5V-12V. Zadatak mu je da pročita signale sa IR modula (slika 2), i na osnovu parametra kontroliše zvučni signal na uređaju. IR modul je zasnovan na principu

odbijanja svetlosti od očnog kapka i taj signal se prosleđuje kontroleru na obradu.



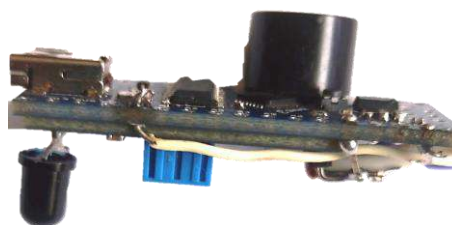
Slika 2. IR modul.

Senzorski modul prilagodljiv je ambijentalnom svetlu. Ima par infracrvenih dioda koje emituju i primaju svetlost. Ova svetlost nije u vidnom rasponu (750 nm do 3 mm, odnosno od 4,5*10¹⁴ do 10¹² Hz) tako da ne smeta vozaču tokom vožnje. Domet detekcije senzora može se menjati podešavanjem potenciometra, i to od 2cm do 30cm, tako da nam to daje veliku slobodu u dizajnu samog uređaja.

Ugao koji je podržan od strane modula je 35°, što nam omogućava vrlo lako podešavanje uređaja. Dimenzije elektronskog dela uređaja su date u tabeli 1, a kompaktnost sklopa na slici3.

TABELA I
DIMENZIJE MODULA SA KONTROLEROM

Širina	22mm
Dužina	43mm
Visina	15mm
Težina	8 gr



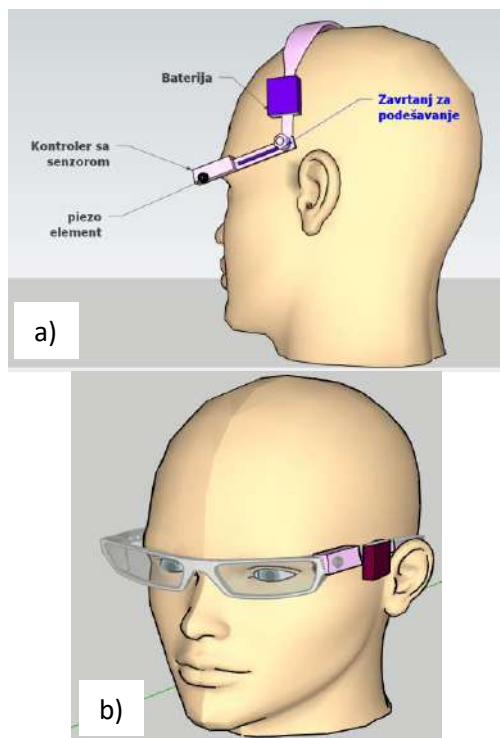
Slika 3. Modul sa kontrolerom.

Ovako male dimenzije su nam omogućile da uređaj bude kompaktan i lak za korišćenje. Pored malih dimenzija i preciznosti u radu ovaj uređaj je daleko konkurentniji i sa cenom od ostalih, jer njegova izrada košta manje od 30 eura.

III. OPIS RADA

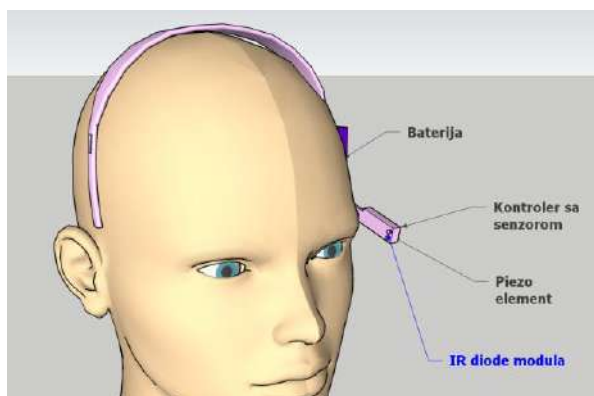
Kao što se da videti, (slika 4), uređaj se može podesiti u odnosu na dimenziju glave korisnika. Zavrtanj za

podešavanje služi za promenu nagiba kao i dubine kontrolera.

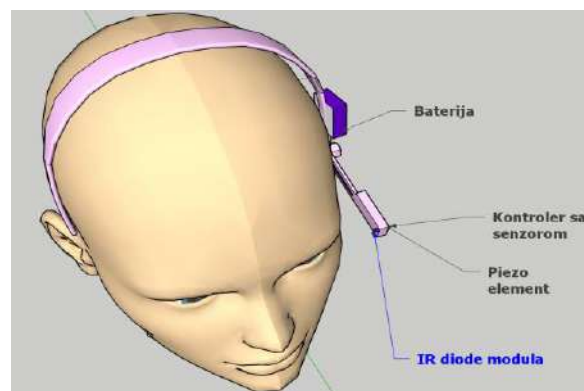


Slika 4. Stay Awake a) – varijanta sa posebnim nosačem, b) – sistem se po potrebi kači na naočare.

Uključivanjem uređaja piezo element se oglašava “pištanjem” u periodu od 1500 milisekundi i time potvrđuje da je uređaj spreman za rad. Pored podešavanja dimenzije, nikakva druga podešavanja nisu dalje potrebna. Bitno je pomenuti da diode modula koje vire sa uređaja (slika 5, 6.) moraju biti usmerene ka kapku oka što je to bolje moguće. Testiranje uređaja se obavlja tako što uređaj se podesi po visini i dubini, a oko je otvoreno. Zaklapanjem kapaka u periodu od samo 1000 milisekundi, piezo element se oglašava “pištanjem” i tako stavlja do znanja da je uređaj ispravno podešen.

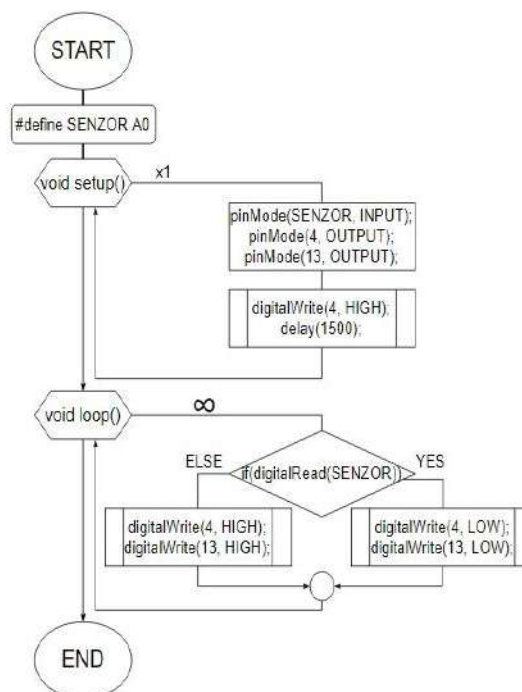


Slika 5. Usmeravanje IR dioda.



Slika 6. Usmeravanje IR dioda.

Ukoliko u toku vožnje, vozač zaklopi očne kapke u periodu od 1000 milisekundi piezo će se oglasiti pištanjem i upozoriti vozača da je umoran i da mu je potreban hitan odmor. Softversko rešenje nam omogućava podešavanja svih parametara ali je uzeto vreme od 1000 milisekundi, za koje će se oglasiti piezo element, na osnovu toga što je svaki čovek individua za sebe, pa neko brže ili sporije trepće. Prema Harvard Database of Useful Biological Numbers prosečno trajanje treptaja je između 100 i 400 milisekundi, a sve preko 1000 milisekundi se smatra mikrospavanjem. Sporiji treptaj od 1000 milisekundi ukazuje na to da su očni kapci vozača već “teški” i da je vreme za odmor i spavanje. Nakon oglašavanja piezo elementa vozač se budi i nakon dizanja kapaka piezo element prestaje sa “pištanjem”. Kodiranje kontrolera izvršeno je u Arduinovom razvojnom okruženju. Detaljan princip rada prikazan je algoritmom (slika 7).



Slika 7. Algoritam.

U avgustu 2020. novom rezolucijom Generalne
pštine UN-a proglašena je Druga decenija akcije za



Ovaj projekat smatramo malim doprinosom ovoj akciji i bili bismo presrećni da bar jedan život bude sačuvan zahvaljujući našem projektu.

LITERATURA

- [10] Matthew Walker, professor of neuroscience and psychology and the director of the Center for Human Sleep Science at the University of California (2017). "Why We Sleep"

Merenje induktivnosti kalema pomoću Arduino mikrokontrolera

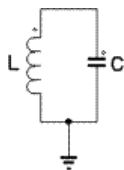
Ana Simić

Sadržaj - Tema rada je merenje induktivnosti kalema merenjem periode oscilovanja napona u paralelnom oscilatornom LC kolu pomoću Arduino mikrokontrolera.

I. UVOD

Kalem je komponenta koja se često koristi u elektrotehnici. Induktivnost je osnovna osobina kalema koja predstavlja odnos između magnetnog fluksa u kalemu i jačine električne struje kroz namotaje kalema.

Paralelno oscilatorno kolo (slika 1) čini paralelna veza kondenzatora i kalema. Kada se oscilatornom kolu saopšti energija bilo kroz struju kalema ili kroz napon na kondenzatoru i kolo prepusti samom sebi tada se u oscilatornom kolu javljaju sinusne oscilacije napona i struje koje su određene parametrima oscilatornog kola - kapacitivnošću kondenzatora C i induktivnošću kalema L.



Slika 1. Paralelno oscilatorno kolo.

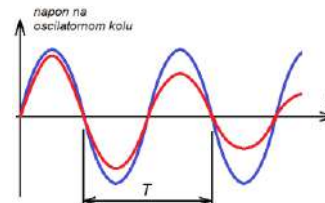
$$\omega = \frac{1}{\sqrt{LC}} \quad (1)$$

Mereći periodu oscilovanja napona na paralelnom oscilatornom kolu može se izmeriti kružna učestanost, a poznavajući jedan od parametara oscilatornog kola, u ovom slučaju kapacitivnost kondenzatora, može se odrediti vrednost preostalog parametra - induktivnost kalema [1][2].

$$L = \frac{T^2}{4\pi^2 C} \quad (2)$$

Kod idealnog oscilatornog kola nema gubitaka energije. Amplituda napona u oscilatornom kolu je konstantna. Paralelno oscilatorno kolo sa realnim elementima ima gubitke energije te električna energija izlazi van oscilatornog kola zbog čega se i amplituda

napona u oscilatornom kolu smanjuje i vremenom nestaje. Na grafiku (slika 2) prikazani su oblici napona na idealnom i realnom oscilatornom kolu.

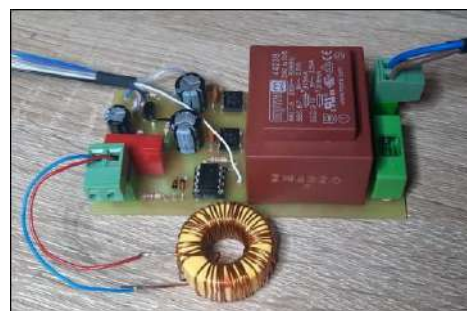


Slika 2. Napon na idealnom (grafik plave boje) i realnom oscilatornom kolu (grafik crvene boje).

Iako se kod realnog oscilatornog kola zbog gubitaka energije oscilacije prigušuju moguće je izmeriti periode prvih nekoliko oscilacija i potom, pomoću mikrokontrolera softverski obraditi rezultate i na LCD displeju prikazati induktivnost kalema.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Uređaj (slika 3) sastoji se iz paralelnog oscilatornog kola koga čine kondenzator poznate kapacitivnosti i kalem čija se induktivnost meri, kola za pobudu oscilatornog kola, kola za detekciju prolaska napona na oscilatornom kolu kroz nulu (zero cross kola), Arduino Uno mikrokontrolera, LCD 16x2 displeja sa I²C interfejsom izvora napajanja.



Slika 3. Deo uređaja za merenje induktivnosti bez Arduino mikrokontrolera i LCD displeja.

A. Hardver

Kako bi se u oscilatornom kolu uspostavile oscilacije napona u kolo se pomoću tranzistora U1 unosi početna energija. Uključenje tranzistora U1 obavlja mikrokontroler preko pina 4 (slika 4).

A. Simić, učenica Tehničke škole u Loznici, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, e-mail: anasimic@tehnickalo.edu.rs


```

    c=2.82E-6;//kapacitivnost kondenzatora
}

void loop()
{
    //pobuda oscilatornog kola
    i=0;
    digitalWrite(pobuda, LOW);
    delay(50);
    digitalWrite(pobuda, HIGH);
    delay(150);
    Serial.print("");

    //Razlika
    m=i--;
    for (j=0; j<m; j++){b[j]=n[j]-n[j-1];}

    //Sortiranje
    for(j=0;j<m; j++){
        for(k=0;k<m-1;k++){
            if
            (b[k]>b[k+1]){e=b[k];b[k]=b[k+1];b[k+1]=e;}}
            T=b[m/2];
            l=(T*T/(4.0*pi*pi*c))*(1.0E-6);

    //Ispis rezultata
    lcd.setCursor(3,0);
    lcd.print("T=");
    if(T>1000){ lcd.print(T/1000);
                lcd.print("ms  ");}
    else {lcd.print(T);
          lcd.print("us   ");}
    lcd.setCursor(3,1);
    lcd.print("L=");
    if(l>1000){ lcd.print(l/1000);
                lcd.print("mH   ");}
    else {lcd.print(l);
          lcd.print("uH   ");}
}

void napon_nula(){
    if(i>39) i=39;
    n[i]=micros();
    i++;}

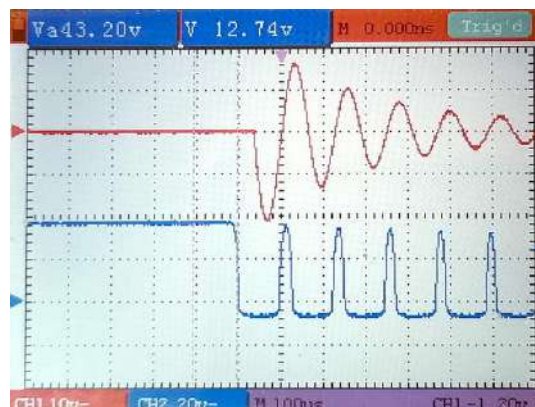
```

III. TESTIRANJE UREĐAJA

Testiranje uređaja za merenje induktivnosti kalema realizovano je upoređivanjem ispisane periode oscilovanja na LCD displeju i periode koji je izmerena pomoću digitalnog osciloskopa. Vrednost ispisane induktivnosti kalema proverena je računskim putem na osnovu poznate i prethodno izmerene kapacitivnosti kondenzatora.

Na slici 7 prikazani su vremenski oblici napona na oscilatornom kolu (grafik crvene boje) od trenutka kada se isključuje pobuda oscilatornog kola i napon na izlazu naponskog komparatora (grafik plave boje). Iako se na osnovu prikazanog merenja vidi da je upotrebljeni naponski komparator spor da isprati promene napona na

oscilatornom kolu njegov kratkotrajni impuls sasvim je dovoljnog trajanja da izazove prekid mikrokontrolera.



Slika 7. Vremenski oblik napona na oscilatornom kolu i izlaz naponskog komparatora.

Provera, odnosno merenje kapacitivnosti kondenzatora koji je upotrebljen u oscilatornom kolu vršeno je sa digitalnim multimetrom MY64. Kapacitivnost upotrebljenog kondenzatora je $2.82\mu\text{F}$. Izmerena induktivnost kalema nije proverena drugim meračem induktivnosti.

Maksimalna vrednost induktivnosti koja je izmerena u toku testiranja rada uređaja iznosi $3,2\text{mH}$.

IV. ZAKLJUČAK

Predstavljeni uređaj za merenje induktivnosti kalema je relativno jeftin i praktičan za upotrebu. Promena kapacitivnosti kondenzatora u oscilatornom kolu i izmena tog parametra u programu omogućava merenje induktivnosti kalema u različitim mernim opsezima.

ZAHVALNICA

Zahvalnost na realizaciji ovog istraživačkog projekta uputila bih Tehničkoj školi u Loznici, mom mentoru, profesoru dipl. inž. elek. Tomislavu Mijiću kao i mojoj porodici na neizmernoj moralnoj i finansijskoj podršci.

LITERATURA

- [1] https://en.wikipedia.org/wiki/LC_circuit
- [2] http://mikro.elfak.ni.ac.rs/wp-content/uploads/Lec_06_KalemoviTransformatori_2018.pdf

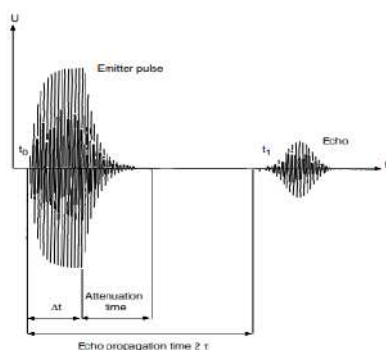
Konstrukcija i primena ultrazvučnog senzora za merenje daljine

Lazar Šipić

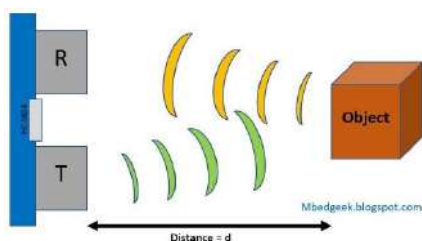
Sadržaj - U okviru rada prezentovan je postupak izrade ultrazvučnog senzora za merenje daljine. Napravljen je kratak uvod u signale i njihovu obradu. Takođe, objašnjeni su načini obrade signala, odnosno filtriranje i pojačanje signala. Glavni deo rada obuhvata simulaciju i eksperimentalno testiranje prijemnog i predajnog pojačavača, koji zajedno čine ultrazvučni senzorski sistem. Na kraju su prikazani grafici i tabele dobijenih rezultata i njihovo poredjenje sa idealnim vrednostima.

I. UVOD

Ultrazvučni daljinomer generiše zvučne talase na frekvenciji od 40 kHz. Zvučni talasi se reflektuju od objekta i vraćaju u prijemnik (echo), senzor daje informacije o vremenu, koje je bilo potrebno da se zvučni talasi prošire od senzora do objekta i nazad. Na slici 1. možemo videti kako su signali iz predajnika veće amplitude nego reflektovani signali, što znači da bi bili detektovani prvo se moraju pojačati, dok na slici 2 vidimo kako se talasi prostiru od predajnika do objekta i od objekta do prijemnika.



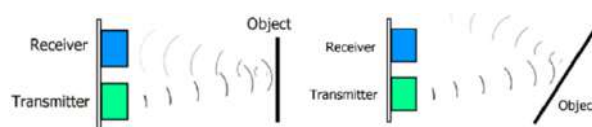
Slika 1. Prikaz emitovanog i reflektovanog signala [1].



Slika 2. Prikaz kretanja talasa (R- prijemnik, T- predajnik) [1].

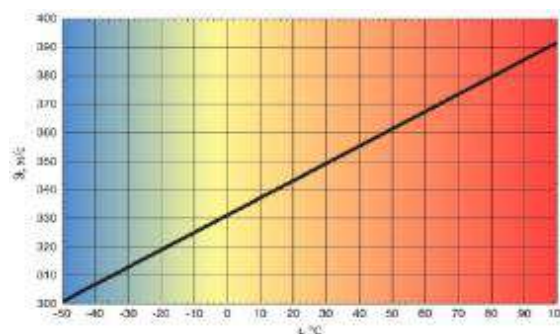
L. Šipić je student na modulu EKM, Katedra za mikroelektroniku, Elektronski fakultet Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sdevlin998@gmail.com.

Na očitavanja ultrazvučnih daljinomera ne utiče sunčeva svetlost niti boja predmeta, kao što je slučaj sa infracrvenim senzorima. Ultrazvučni talas se reflektuje sa gotovo bilo koje površine, čak i providne, ali može biti teško odrediti rastojanje do lepršavih ili malih predmeta. Osim toga, na očitavanja utiče ugao pada talasa. Ako je senzor usmeren pod uglom od 90 sepeni na predmet, merenja će biti najpreciznija. Osim toga, ako je upadni ugao prevelik, onda talas, odbijen od objekta, ne ulazi u prijemnik, što će dovesti do pogrešnog merenja.



Slika 3. Primer pravilnog merenja i nepravilnog merenja [2].

Na slici 4 je prikazana promena brzine zvuka sa porastom temperature. Sa povećanjem temperature vazduha za 1 °C, brzina zvuka u njemu raste za 0,6 m/s. Da bismo se brzo prilagodili promenljivoj temperaturi okoline, potrebno je u merni sistem dodati temperaturni senzor i koristiti rezultate njegovih merenja. Neki ultrazvučni daljinometri imaju ugrađene senzore temperature.



Slika 4. Zavisnost brzine zvuka od temperature [3].

Izračunavanje udaljenosti objekta se vrši na osnovu dobijenog vremena na osnovu sledeće relacije:

$$s = \frac{vT}{2} \quad (1)$$

v je brzina zvuka (≈ 340 m/s); T je vreme kretanja talasa od senzora do objekta i nazad. Brzina zvuka je ovde uzeta kao konstanta (340 m/s), ali to u praksi neće dati najbolje

rezultate, zato što na slici 4. vidimo kako brzina zvuka u vazduhu zavisi od temperature. Da bismo dobili najtačnije rezultate, odnosno da bi određena distanca bila što tačnija, mora se uvek uzeti u obzir temperatura. Formula za izračunavanje brzina zvuka na 0 stepeni sledi:

$$C_0 = \sqrt{\frac{P_0 \cdot k}{\rho_0}} \text{ [m / s]} \quad (2)$$

gde C_0 predstavlja brzinu zvuka u vazduhu na 0 stepeni, P_0 atmosferski pritisak, ρ_0 gustinu vazduha na 0 stepeni i k predstavlja adijabatski indeks [9]. Kako je $P_0 = 101,325 \text{ kPa}$ (standardno), $\rho_0 = 1.293 \text{ kg / m}^3$, $k = 1.402$:

$$C_0 = \sqrt{\frac{101325 \cdot 1,402}{1,2935}} = 331,4 \text{ [m/s]} \quad (3)$$

Brzina zvuka u vazduhu na 0 stepeni iznosi 331,4 m/s. Zatim je potrebno izračunati brzinu zvuka na nekoj temperaturi t . Računica će se odvijati preko sledeće formule:

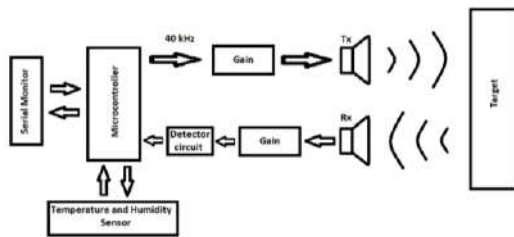
$$C = C_0 + (0.6 \cdot t) \text{ [m / s]} \quad (4)$$

gde ćemo za t uzeti 25 stepeni. Zamenom vrednosti dobijamo:

$$C = 331,4 + (0.6 \cdot 25) \\ C = 346,4 \text{ [m / s]}$$

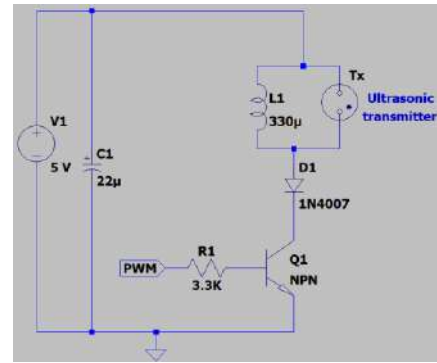
III. OPIS BLOKOVA

Na slici 5 prikazana je blok šema ultrazvučnog daljinomera koji uz pomoć senzora za merenje temperature i relativne vlažnosti vazduha određuje što preciznije brzinu ultrazvučnih talasa.



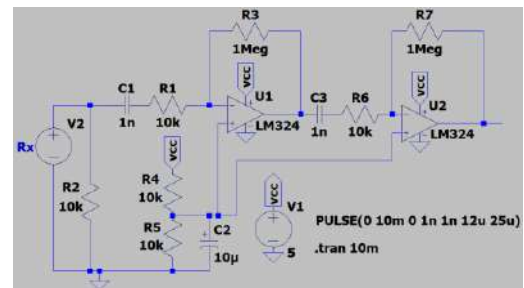
Slika 5. Blok šema ultrazvučnog senzora za merenje daljine.

Dobijeni rezultati se ispisuju na monitoru (*serial monitor*). Električna šema sa slike 6 nije neophodna za funkcionisanje ultrazvučnog daljinomera, ali će domet biti znatno skraćen. Dovoljno je samo prikazati ultrazvučni emiter na izlaz mikrokontrolera i funkcionisaće bez problema. Cilj ovog kola (slika 6) jeste baš to, povećanje dometa, gde uz pomoć par aktivnih i pasivnih komponenti rešavamo problem.



Slika 6. Pojačavački blok predajnika (Tx) nacrtan u LT Spice-u.

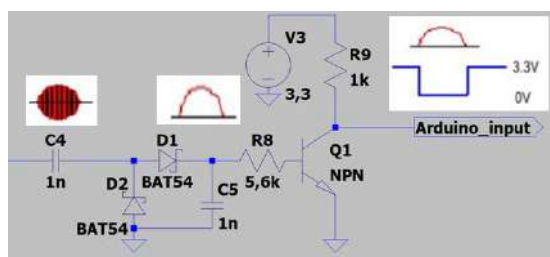
Na PWM ulaz šaljem 40kHz frekvenciju sa mikrokontrolera i time palimo i gasimo npn tranzistor Q1. Kada se na bazi tranzistora Q1 pojavi logička "1" tada će tranzistor da provede i kroz kalem će proteći struja (smatramo da je vreme logičke "1" dovoljno dugo da struja kroz kalem postane konstantna). U narednom koraku dolazi logička "0" i tranzistor prestaje da vodi. U trenutku ukidanja impulsa kalem teži da zadrži uspostavljeni smer struje, pa na svojim krajevima indukuje napon suprotnog polariteta. Kalem akumulira energiju u magnetnom polju koje nastaje pod dejstvom promene struje kroz njega. Akumulacija energije predstavlja punjenje, a njeno oslobađanje pražnjenje kalema. Ako je vreme logičke "0" dovoljno dugo, kalem će se u potpunosti isprazniti, pa će struja kroz njega prestati da teče. C1 predstavlja decoupling kondenzator čija je uloga u kolu da smanju ulazni šum i padove napona, tj poskakivanje napona. Uloga diode D1 je da zaštiti tranzistor od napona suprotnog polariteta koji se javlja na kalemu L1. Na slici 7 prikazan je pojačavački blok primljenog signala (Rx). Ovaj blok je neophodan zato što primljeni signal je reda mV i potrebno ga je pojačati kako bi taj signal bio od koristi, tj. da bi bismo ga detektovali pomoću naseg kola za detekciju signal mora biti veće amplitude.



Slika 7. Pojačavački blok signala sa prijemnika nacrtan u LT Spice-u.

Naponski generator V2 imitira prijemnik Rx. Podesen je tako da daje PWM signale frekvencije 40 kHz i amplitude 10 mV. Amplituda signala zavisi od daljine, sto je daljina veća onda će amplituda primljenog signala biti manja i obrnuto. Uloga otpornika R2 je prilagodjenje impedance, da

bi se smanjio ulazni šum kako bi signal bio što "čistiji". Kondenzator C1 blokira svaku DC komponentu jer može poremetiti signal, tj. dati signalu offset i tada bi signal bio podignut. Za pojačanje signala koristili smo operacioni pojačavač LM324 u single supply modu. Da bi operacioni pojačavač radio u single supply modu neophodno je bilo dovesti na neinvertujući pin VCC/2, tj. 2.5 V kako bi podigli ulazni signal da bi bio "vidljiv". Uloga kondenzatora C3 je ista kao i kod C1. Što se pojačanja tiče, isto je za oba pojačavačka bloka. S obzirom da signal dovodimo na invertujući ulaz, pojačanje bi bilo $- R2 / R1$. Za prvi pojačavač bi glasilo $- R3 / R1 = 100$. U praktičnoj realizaciji koristili smo operacioni pojačavač LM324 koji u sebi sadrži 4 međusobno nezavisna operaciona pojačavača, od kojih su dva iskorišćena. Na slici 8 prikazan je blok za detekciju signala. Signal iz drugog pojačavača vodimo u naredni blok, tj. blok za detekciju čija je uloga da detektuje svaki put kada se na njegovom ulazu pojavi logička "1". Nakon pojačanja, signal će proći kroz kolo detekcije koje se sastoji od polutalasnog kola za ispravljanje. Ovo kolo implementira dve šotkijeve diode. Ispravljeni signal odlazi na tranzistor. Kada nema signala, izlaz je 3,3 V (HIGH). Ako je signal prisutan, to će uzrokovati pad izlaznog napona na 0 V (LOW). Signal dalje ulazi u arduino gde se dalje procesira da li ce i kada biti prekinuto merenje vremena.



Slika 8. Blok za detekciju reflektovanih impulsa nacrtan u LT Spice – u.

Poslednji blok predstavlja blok za merenje temperature i vlažnosti vazduha čija je uloga da što preciznije odredimo brzinu kretanja ultrazvuka. Senzor je prikazan na slici 9.



Slika 9. DHT11 senzor temperature i relativne vlažnosti vazduha.

Vrednosti dobijene pomoću senzora DHT 11 prikazane su na slici 10.

```
16:48:30.454 ->
16:48:30.454 -> Distance (cm): 20.00
16:48:30.454 -> Humidity (%): 45.00
16:48:30.454 -> Temperature (C): 23.00
16:48:33.050 ->
16:48:33.050 ->
16:48:33.050 -> Distance (cm): 20.00
16:48:33.050 -> Humidity (%): 45.00
16:48:33.096 -> Temperature (C): 23.00
16:48:34.492 ->
16:48:34.492 ->
16:48:34.492 -> Distance (cm): 19.00
16:48:34.492 -> Humidity (%): 45.00
16:48:34.538 -> Temperature (C): 23.00
```

Slika 10. Prikaz dobijenih rezultata pomoću DHT 11 senzor.

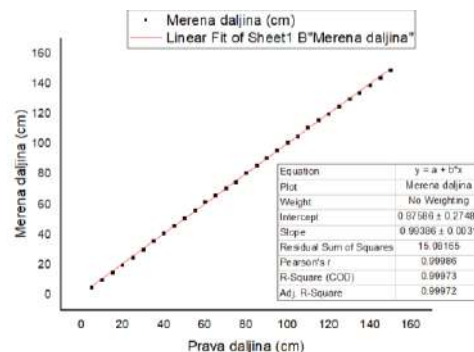
IV. PRAKTIČNA REALIZACIJA

TABELA I

VREDNOSTI DOBIJENE KORIŠĆENJEM SENZORA U ODNOSU NA PRAVE VREDNOSTI

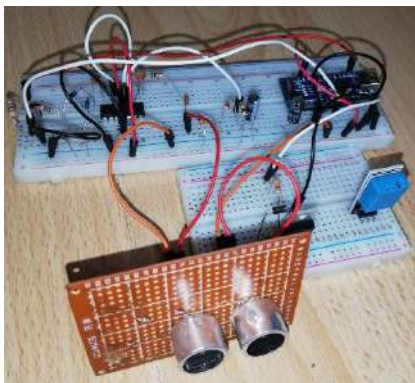
	A(X)	B(Y)			
Long Name	Prava dalji	Merena dal			
Units	cm	cm			
Comments					
F(x)=					
1	5	5	16	80	81
2	10	10	17	85	86
3	15	15	18	90	91
4	20	20	19	95	96
5	25	25	20	100	101
6	30	30	21	105	105
7	35	36	22	110	111
8	40	41	23	115	116
9	45	46	24	120	120
10	50	51	25	125	125
11	55	56	26	130	130
12	60	62	27	135	134
13	65	66	28	140	139
14	70	71	29	145	144
15	75	75	30	150	149

Na osnovu merenja kreiran je grafik u programu Origin.



Slika 11. Grafični prikaz merenja daljine pomoću ultrazvučnog senzora.

Na slici 12 prikazana je praktična realizacija ultrazvučnog senzora za merenje daljine. Na slici 13 je prikazano eksperimentalno merenje, dok su na slici 14 prikazani eksperimentalni rezultati merenja.



Slika 12. Prikaz praktične realizacije ultrazvučnog senzora.



Slika 13. Eksperimentalno merenje.



Slika 14. Dobijeni rezultati.

V. ZAKLJUČAK

Ultrazvučni senzori imaju široku primenu, osim merenja udaljenosti pokretnih objekata, koriste se za merenje nivoa, protoka, debljine i sastava nekih materijala. Na trenutnom projektu može se još puno raditi. Odličan je kao početak razumevanja kako ceo koncept funkcioniše, počevši od prostiranja ultrazvučnih talasa kroz vazduh pa sve do programiranja mikrokontrolera kako bi dobijene podatke pretvorili u korisne informacije. Nakon toga treba nastaviti razvijanje celokupne elektronike i posvetiti se većoj preciznosti samog uređaja, povećanju dometa i rešavanju najvećeg problema ultrazvučnog senzora, tj detekciji malih i sitnih objekata. Ultrazvučni senzor ne može detektovati objekte koji su previše mali, uski, niski.. Pokazivaće kao da se tu ne nalazi ništa. Jos jedan interesantan i još precizniji projekat je merenje brzine vetra pomoću ultrazvučnog senzora gde pored savršene elektronike i programiranje mora da bude na visokom nivou.

ZAHVALNICA

Rad je realizovan u laboratorijama Elektronskog fakulteta, kao deo diplomskog rada na modulu Elektronske komponente i mikrosistemi. Autor se zahvaljuje mentoru prof. dr Ljubomiru Vračaru na pruženom znanju i pomoći.

LITERATURA

- [1] Aplikacije ultrazvučnog senzora . Dostupno na: <https://www.migatron.com/ultrasonic-detections-and-control-applications/>
- [2] Princip rada. Dostupno na: <https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq/>
- [3] Proračuni za brzinu ultrazvuka. Dostupno na: <http://www.sengpielaudio.com/SpeedOfSoundPressure.pdf>, <http://www.sengpielaudio.com/calculator-speedsound.htm>
- [4] Senzori i pretvarači prof. dr Ljubomir Vračar. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Senzori-i-pretvaraci_Predavanja.pdf
- [5] Autonomni mikrosistemi prof. dr Ljubomir Vračar http://mikro.elfak.ni.ac.rs/wp-content/uploads/Autonomni_mikrosistemi_Predavanja.pdf
- [6] Korišćeni program.Dostupno na: <https://www.originlab.com/>
- [7] Datasheet korišćenih komponenti.

Proširenje SMTP protokola za automatsku *end-to-end* enkripciju imejl poruka

Nađa Gavrilović, Mirjana Živković

Sadržaj – Proteklih godina, sa razvojem Interneta, sve je veća potreba da komunikacija korisnika na mreži bude što sigurnija. Jedan od najčešće korišćenih mrežnih servisa za komunikaciju je elektronska pošta. U osnovnoj verziji SMTP protokola, koji vrši prenos imejl poruka kroz mrežu, sadržaj poruke koja se šalje nije ni na koji način zaštićen. Ekstenzija SMTP protokola, koja je danas u širokoj upotrebi (STARTTLS) koristi TLS protokol na transportnom nivou i omogućava zaštitu samo komunikacionih kanala prilikom prenosa sa servera na server upotrebom simetrične enkripcije. Cilj ovog rada je projektovanje i implementacija proširenja SMTP protokola koje omogućava asimetrično šifriranje poruka i osigurava potpunu *end-to-end* imejl enkripciju, od izvora do krajnjeg odredišta. U radu je detaljno prikazana implementacija SMTP protokola sa predloženim proširenjem, koje dodatkom u SMTP klijent/server komunikaciji omogućava kompletno očuvanje poverljivosti imejl poruka.

I. UVOD

Elektronska pošta (*e-mail*) danas predstavlja jedan od najzastupljenijih mrežnih servisa, kao i najčešće korišćenih Internet usluga. Standardni mrežni protokol koji omogućava prenos elektronske pošte i definiše način slanja imejl poruka jeste SMTP, dok su POP3 i IMAP pristupni protokoli i definišu prijem i preuzimanje poruka na klijentskoj strani [1]. Uređaji koji prenose imejl poruke od izvora do odredišta nazivaju se agentima i njihova klijent/server komunikacija definisana je SMTP protokolom [2].

U osnovnoj verziji SMTP protokola, poruka koja se šalje nije ni na koji način zaštićena od neovlašćenog pristupa. Imejl poruka se šalje u *plaintext* formatu, tj. u formi čistog teksta. Na taj način, omogućeno je bilo kom napadaču da presretne saobraćaj i da pročita sadržaj imejl poruke, koji može biti poverljive prirode. Takođe, bilo ko može pristupiti SMTP serveru na standardnom SMTP portu (25) i poslati imejl poruku proizvoljne sadržine, čak i sa proizvoljnim informacijama u zaglavlju poruke (primalac, pošiljalac i sl.).

Jedna od postojećih opcija za zaštitu imejl saobraćaja je STARTTLS ekstenzija, koja koristi TLS protokol i štiti

N. Gavrilović je student doktorskih akademskih studija na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: nadja.gavrilovic@elfak.ni.ac.rs

M. Živković je student master akademskih studija na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: mirjana.zivkovic@elfak.rs

komunikacioni kanal mejl transfer agenata na transportnom nivou, upotrebom simetrične enkripcije. Takav bezbednosni mehanizam štiti prenos poruka između agenata, ali ne obezbeđuje potpunu *end-to-end* enkripciju imejl poruka, od izvora do odredišta [3, 4].

Imejl saobraćaj može se zaštititi i upotrebom S-MIME i PGP standardnih protokola. Iako obezbeđuju *end-to-end* enkripciju poruka, oba protokola zahtevaju generisanje ključa za simetrično šifriranje na strani izvora. Zatim, neophodno je i asimetrično šifriranje i distribucija ključa primaocu za svaku pojedinačnu poruku, što dodatno usložnjava operaciju slanja imejla. Dodatno, ovakav proces zaštite nije transparentan krajnjim imejl korisnicima [3, 4].

Rešenje koje omogućava očuvanje privatnosti sadržaja imejl poruka, uz potpunu transparentnost krajnjim korisnicima i bez usložnjavanja komunikacije je enkripcija imejl poruka upotrebom asimetričnih kriptografskih algoritama.

Ovi algoritmi podrazumevaju korišćenje dva različita ključa – javnog i privatnog, koji omogućavaju enkripciju i dekripciju poruka u cilju očuvanja njihove poverljivosti. Upotrebom para privatni-javni ključ omogućeno je slanje enkriptovane poruke sa izvora, koju može dekriptovati samo odredište kome je poruka namenjena, čime je obezbeđeno *end-to-end* šifriranje podataka. Najšire korišćeni algoritam sa javnim ključem jeste RSA algoritam [3].

Cilj ovog rada je projektovanje i implementacija SMTP protokola sa proširenjem koje omogućava šifriranje poruka, upotrebom algoritma sa javnim ključem. Na taj način, osigurava se potpuna *end-to-end* imejl enkripcija od izvora do krajnjeg odredišta, čime se sprečava neovlašćen pristupi imejl poruci, bilo tokom prenosa, bilo pristupom usputnim mejl agentima. U radu će detaljno biti prikazana implementacija SMTP protokola sa predloženim proširenjem, koje dodatkom u SMTP klijent/server komunikaciji omogućava očuvanje poverljivosti imejl poruka na mreži.

Nakon uvoda, u drugom poglavlju rada biće predstavljen SMTP protokol, kao i proces SMTP komunikacije servera i klijenta tokom slanja imejl poruke. Treće poglavlje biće posvećeno kriptografiji sa javnim ključem, dok će četvrto poglavlje predstaviti RSA algoritam. Peto poglavlje predstaviće implementaciju SMTP protokola sa mogućnošću automatskog šifriranja imejla. Zaključak će biti dat u šestom poglavlju.

II. SIMPLE MAIL TRANSFER PROTOCOL

SMTP je standardni mrežni protokol za prenos elektronske pošte. Koriste ga agenti (MTA – *Message Transfer Agent*) za međusobni transfer poruka, kao i korisnički agenti (MUA – *Mail User Agent*), kako bi inicijalno poslali poruku do agenta za transfer poruka [5].

Prenos imejl poruke započinje kada SMTP klijent otvori konekciju ka SMTP serveru. Svaka imejl transakcija sadrži niz komandi kojima klijent definiše izvor i odredište mejla, kao i samu poruku. Komande su četvoroslovne, dok se eventualni parametri odvajaju razmakom. Na komande klijenta server odgovara trocifrenim kodom, koji daje informaciju o prihvatanju komande i stanju komunikacije. Svaka cifra u odgovoru ima specifično značenje [5].

Početak SMTP komunikacije servera i klijenta dat je u nastavku:

```
S: 220 foo.com Simple Mail Transfer
    Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
```

Klijent šalje zahtev za otvaranje konekcije i predstavlja se. Server prihvata konekciju, i šalje dodatne opcije, dok kodom 250 označava kako je sa komunikacijom sve u redu i zahtevi klijenta su uspešno obrađeni [2].

Sama mejl transakcija sastoji se od tri komande: MAIL, RCPT i DATA. Primer ovog dela komunikacije dat je u nastavku:

```
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
```

MAIL komandom klijent definiše imejl adresu pošiljaoca. Nakon toga, jednom ili uz više RCPT komandi, klijent definiše primaoca imejla. Server prihvata sve primljene parametre (kodom 250 u odgovoru), osim drugog primaoca poruke, u čijem slučaju kod u odgovoru sugerise da adresa primaoca nije dostupna [2].

Treći korak transakcije započinje DATA komandom od strane klijenta, nakon koje se šalje sama imejl poruka. Nakon DATA komande, svaka naredna poruka klijenta tumači se kao deo poruke za slanje, sve do prijema tačke u zasebnoj liniji, koja označava kraj poruke. Komunikacija se završava QUIT komandom, na zahtev klijenta. Kada se prenos poruke završi, klijent može da uputi i druge zahteve ka serveru, kao na primer zahtev za verifikaciju imejl adrese

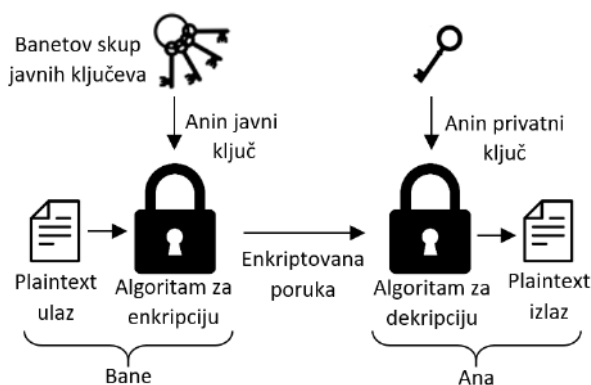
[2]. Primer dostavljanja sadržaja imejl poruke i završetka komunikacije dat je u nastavku:

```
C: DATA
S: 354 Start mail input; end with
    CRLF>.<CRLF>
C: Blah blah blah. . .
C: . . .etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing
    transmission channel
```

III. KRIPTOGRAFIJA SA JAVNIM KLJUČEM

Kriptografija je nauka koja se bavi metodama očuvanja tajnosti informacija. U zavisnosti od broja ključeva koji se koriste u kriptografskom sistemu, on može biti simetričan i asimetričan [3].

Kriptografija sa javnim ključem je asimetrična, što podrazumeva korišćenje dva različita ključa - javnog i privatnog, od kojih se jedan koristi u procesu enkripcije, a drugi tokom dekripcije. Privatni ključ poznat je samo primaocu poruke, i pomoću njega se poruka može dešifrovati. Javni ključ poznat je bilo kome i koristi se za šifriranje poruka. Osim sa ciljem enkripcije podataka, kriptografija sa javnim ključem koristi se i za kreiranje digitalnog potpisa, kao i razmenu tajnih ključeva [3]. Šema za enkripciju javnim ključem data je na slici 1.



Slika 1. Enkripcija javnim ključem.

Svaki učesnik u komunikaciji generiše par ključeva (javni-privatni). Javni ključ se postavlja u javni registar ili se na drugi način distribuira, dok se privatni ključ drži u tajnosti. Svaki korisnik ima kolekciju javnih ključeva korisnika sa kojima komunicira. Kao što je dato na slici 1, kada jedan učesnik komunikacije (Bane), šalje poverljivu poruku drugom (Ani), enkripcija poruke vrši se Aninim javnim ključem. Primljenu poruku Ana dekriptuje svojim privatnim ključem. Na ovaj način, osigurano je da samo Ana može videti inicijalni sadržaj poruke. Neki algoritmi (npr. RSA algoritam) imaju osobinu da se bilo koji od dva

ključa može koristiti za enkripciju, odnosno dekripciju podataka [3].

IV. RSA ALGORITAM

Jedna od prvih uspešnih implementacija asimetričnog kriptografskog algoritma bila je od strane Ron Rivesta, Adi Šamira i Lena Adlemana, prvi put objavljena 1978. godine. Šema RSA (Rivest-Šamir-Adleman) predstavlja najrasprostranjeniji pristup enkripciji sa javnim ključem [3].

RSA algoritam se bazira na primeni stepenovanja na konačnom polju korišćenjem modularne aritmetike i prostih brojeva. Bezbednost algoritma zasniva se na teškoćama vezanim za faktorisanje velikih brojeva [3]. Inicijalni *plaintext* se enkriptuje po blokovima, gde svaki blok ima binarnu vrednost manju od zadate vrednosti n . Za *plaintext* blok M i blok šifrovanog teksta C , enkripcija (1) i dekripcija (2) su u sledećem formatu:

$$C = M^e \bmod n \quad (1)$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n \quad (2)$$

Vrednost n poznata je i primaocu i pošiljaocu. Pošiljalac zna vrednost parametra e , a samo primalac zna vrednost parametra d . Zbog toga, javni ključ karakterišu vrednosti $PU = \{e, n\}$, a privatni ključ $PR = \{d, n\}$. Sve vrednosti (privatni, javni ključ, vrednost n) predstavljaju velike cele brojeve, koje učesnici komunikacije pronalaze na algoritmom definisan način [3].

Na osnovu prethodno navedenog, RSA šifriranje /dešifriranje izgleda ovako [3]:

1. Pošiljalac dobija javni ključ primaoca $PU = \{e, n\}$
2. Pošiljalac izračunava $C = M^e \bmod n$, gde je $0 \leq M < n$
3. Primalac dešifriranje obavlja korišćenjem tajnog ključa $PR = \{d, n\}$
4. Primalac izračunava $M = C^d \bmod n$

Da bi ovaj algoritam zadovoljavao uslove enkripcije sa javnim ključem, mora da zadovolji sledeće uslove [3]:

- Moguće je naći vrednosti e , d i n tako da je $M^{ed} \bmod n = M$ za svako $M < n$.
- Relativno je lako izračunati $M^{ed} \bmod n$ i $C^d \bmod n$ za sve vrednosti $M < n$.
- Neizvodljivo je izračunati d , pomoću vrednosti e i n .

V. IMPLEMENTACIJA SISTEMA

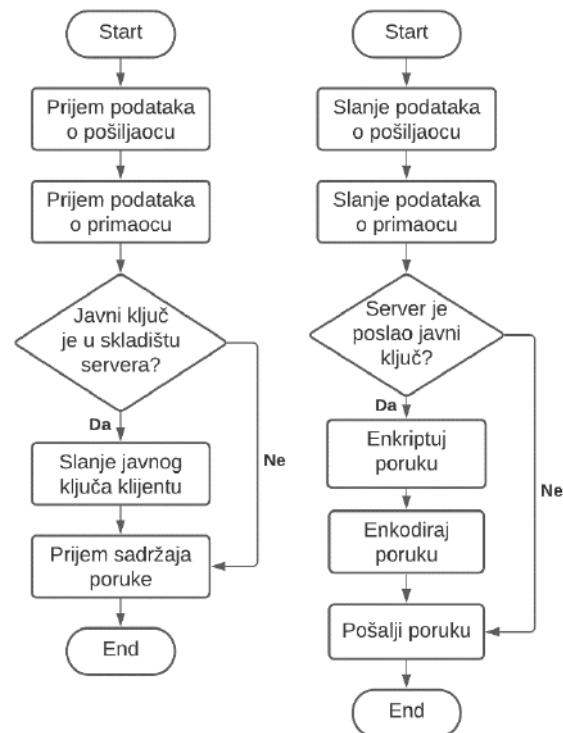
U ovom poglavlju biće dato predloženo proširenje SMTP rotokola koje omogućava automatsko šifriranje imejla. Implementacija sistema je izvršena u programskom jeziku *Python*, korišćenjem *Python Cryptography Toolkit* (*Pycrypto*) paketa. Ovaj paket predstavlja kolekciju heš

funkcija (*Crypto.Hash*), različitih algoritama za enkripciju (*Crypto.Cipher*), algoritama sa javnim ključem (*Crypto.PublicKey*), itd.

Realizovane su dve aplikacije, od kojih jedna predstavlja SMTP server, dok je druga u ulozi SMTP klijenta i zapravo predstavlja MUA uređaj odgovoran za slanje imejl poruke. Obe aplikacije imaju osnovne funkcionalnosti predviđene SMTP protokolom i nadograđene su sistemom za šifriranje poruka.

Osnovna prednost implementacije predloženog sistema jeste mogućnost enkripcije imejl poruka javnim ključem na samom izvoru, u cilju sprečavanja napadača od čitanja poverljivih poruka tokom celokupnog prenosa poruke kroz mrežu. U nastavku poglavlja će biti opisan kompletan tok SMTP komunikacije servera i klijenta, sa naglaskom na izmenama predloženim u ovom radu.

Šifriranje imejl poruka upotrebom asimetričnog kriptografskog algoritma omogućeno je projektovanjem proširenja klijent/server SMTP komunikacije. Na slici 2a prikazan je algoritam rada implementiranog SMTP servera tokom prenosa jedne imejl poruke, dok je na slici 2b dat algoritam rada klijenta tokom istog procesa.



Slika 2. Algoritam rada SMTP servera (a) i klijenta (b).

Na početku komunikacije implementirane aplikacije prate tok osnovne SMTP transakcije, opisane u drugom poglavlju rada (prvi i drugi blok, slike 2a i 2b). Nakon što klijent pošalje podatke o pošiljaocu i primaocu poruke, implementirani SMTP server ima zadatak da omogući enkripciju poruke koja će biti poslata u nastavku komunikacije. Server u tom trenutku klijentu šalje javni

ključ primaoca imejl poruke, ako ga ima u svojoj bazi (treći i četvrti blok, slika 2a). Nakon prijema ključa, klijent izvršava enkripciju svoje poruke javnim ključem (treći i četvrti blok, slika 2b). Kao kriptografski algoritam u implementaciji korišćen je prethodno opisani RSA algoritam. Pre samog slanja poruke, klijent će izvršiti base64 encoding i takvu poruku će proslediti serveru (peti i šesti blok, slika 2b). Sadržaj poruke se završava znakom \".\" kako bi bio u skladu sa SMTP standardom. Na samom kraju komunikacije, server prima poruku (peti blok, slika 2a) i prosleđuje je ka primaocu, kao u protokolom definisanoj osnovnoj SMTP komunikaciji.

Važno je napomenuti da, kao što se može primetiti na slikama 2a i 2b, u slučaju da javni ključ nije prisutan, SMTP protokol je istovetan zvaničnom i bez izmena. Ova osobina predloženo SMTP proširenje čini kompatibilnim sa originalnom SMTP specifikacijom. U nastavku je dat primer kompletne komunikacije implementiranih SMTP servera i klijenta sa predloženim proširenjem osnovnog SMTP protokola:

```
C: EHLO author@example.com
S: 250-smtpclient.com
S: 250-SIZE
S: 250-8BITMIME
S: 250-HELP
C: MAIL FROM author@example.com
S: 250 OK
C: RCPT TO recipient@example.com
S: 250 OK -----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuVa
AwkxX3zmQWxeNWUBUHcmLf
hZf3X2nkm+h18v8Xj4kaib44ZEPULwC7b7UHtkiwQE
xfWwL+b3P25x2my4xJ6KRy
+uCJVeTMw3rslDac861vwMscXyNvZ1aVKmH2jZHDuM
ggC86oJucLmcRiiFm8+Q6J
Gu4Yha3yJ2WcKJtZHQIDAQAB
-----END PUBLIC KEY-----e
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C:
n0ouyS8xC7yYV5qu1b6rZDATvalYYc0Z41U+7zcBrH5NSoa
K81JW/A05XfYA/PYB1Ur043/reK5eRCDwrNqc++fuxK9v5
+vOVxW+erHi2QLad1gs30Wpgr6p2oKoyB1JE1Nvo55HMiVb
owGivlpMfoaIdRMEDaq0EBQ9TpXd8s=
S: 250 OK
C: QUIT
S: 221 Bye
```

Kao što je prethodno rečeno, sam SMTP protokol u osnovnoj verziji ne pruža dovoljno sigurnosti i privatnosti

korisnicima, ako nije prethodno obezbeđen neki mehanizam zaštite. Postojeći protokoli za zaštitu imejl saobraćaja koriste kombinaciju simetrične i asimetrične enkripcije imejl poruka i usložnjavaju proces slanja, značajno uključujući izvor poruke u sam proces enkripcije.

Prednost rešenja projektovanog i implementiranog u ovom radu je što proces šifriranja poruke obezbeđuje potpunu tajnost poruke tokom njenog celokupnog prenosa do odredišta, dok ujedno čitav proces čini transparentnim imejl korisnicima. Takođe, implementirano rešenje omogućava neometan rad SMTP servera i onda kada nije zahtevano šifriranje poruke, što osigurava kompatibilnost predloženog rešenja sa osnovnom verzijom protokola. Korišćeni RSA algoritam je široko rasprostranjen i njegova sigurnost se redovno testira. Takođe, sam proces enkripcije poruke je dovoljno brz da ne utiče na performanse rada sistema i korisničko iskustvo prilikom upotrebe elektronske pošte.

VI. ZAKLJUČAK

U ovom radu projektovano je i implementirano proširenje SMTP protokola, koje omogućava automatsko šifriranje imejl poruke na izvoru, tako da je obezbeđena zaštita sadržaja poruke do njenog odredišta. Detaljno je prikazana implementacija predloženog dodatka SMTP klijent/server komunikaciji, implementiranog u cilju zaštite imejl poruka od neovlašćenog pristupa. Demonstrirana je upotreba kriptografskog algoritma sa javnim ključem, u cilju očuvanja poverljivosti imejl poruka na mreži.

LITERATURA

- [1] A.S.Tanenbaum, D. Wetherall, Computer networks, 5th Edition, 2011.
- [2] J. Klensin, "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008.
- [3] W. Stallings, Network Security and Cryptography 7th edition, Pearson Education, 2017.
- [4] S.A. Sheikh, M. Tariq Banday, "Secure E-mail Communications Through Cryptographic Techniques-A Study", In: Gao XZ., Tiwari S., Trivedi M., Mishra K. (eds) Advances in Computational Intelligence and Communication Technology, vol 1086. Springer, Singapore, 2021.
- [5] N. Gavrilović, "Design and Implementation of SMTP Network Protocol Extension for Proof of Client's Work", Proceedings on IEEEESTEC 12th student projects conference, Faculty of Electronic Engineering, Niš, November, 2019, pp. 321-324, ISBN: 978-86-6125-215-0

Rešavanje optimizacionih problema primenom genetičkih algoritama – primer putujućeg trgovca

Marjan Dorđević

Sadržaj – U ovom radu detaljno je opisan način rešavanja optimizacionih problema primenom genetičkih algoritama. Kao primer za optimizaciju uzet je problem putujućeg trgovca. Programski kod je napisan koristeći programski jezik Python. Kroz rad je objašnjen detaljan postupak kreiranja genetičkog algoritma za rešavanje problema putujućeg trgovca.

I. GENETIČKI ALGORITAM

U oblasti veštačke inteligencije genetički algoritam (GA) je pretraživačka heuristika koja oponaša proces prirodne selekcije. Ova heuristika (takođe ponekad nazivana metaheuristika) se rutinski koristi da generiše korisna rešenja za optimizaciju i probleme pretrage. Genetički algoritmi pripadaju većoj klasi evolucionih algoritama (EA) koji generišu rešenja za optimizaciju problema korišćenjem tehnika insprisanih prirodnom evolucijom, kao što su nasleđivanje, mutacija, selekcija i krosing-over [1].

A. Princip rada genetičkih algoritama

Rad genetičkih algoritama se može opisati u sedam koraka:

1. Definisanje svih potrebnih parametara problema i genetičkog algoritma
2. Formiranje početne populacije
3. Određivanje cene hromozomima (formiranjem funkcije pogodnosti)
4. Selekcija hromozoma koji će opstati za parenje
5. Ukrštanje – obično se iz boljeg dela populacije odaberu roditelji koji će na neki način ukrstiti svoj genetički material i dati jednog ili više potomaka
6. Mutacija, pri kojima se menja genetički sadržaj hromozoma
7. Ispitivanje konvergencije, da bi se utvrdilo da li ima osnova da se tok algoritma prekine. Ukoliko uslov nije ispunjen, vratiti se na korak 3 [1].

II. PROBLEMI OPTIMIZACIJE

U genetičkom algoritmu, populacija rešenja, kandidati (tzv. pojedinaca, bića ili fenotipa) problema optimizacije evoluiraju ka boljim rešenjima. Svako rešenje kandidat ima skup osobina (svojih hromozoma ili genotipa) koji može da

M. Dorđević je student Doktorskih studija na katedri za Upravljanje sistemima, Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14 18000 Niš, Srbija, E-mail: marjan.djordjevic@outlook.com

bude mutiran ili izmenjen; tradicionalno rešenja su predstavljena binarno, kao nizovi nula i jedinica, ali i druga kodiranja su takođe moguća [1].

Evolucija obično počinje od populacije nasumično generisanih pojedinaca, i to je proces koji se ponavlja, sa populacijom koja se u svakoj iteraciji zove generacija. U svakoj generaciji fitness svakog pojedinca u populaciji se ocenjuje; fitness je obično vrednost objektna funkcije u problemu optimizacije koji se rešava. Što više fit (zdravih) pojedinaca je stohastički selektovano iz trenutne populacije, i genom svake individue se menja (kombinuje i eventualno mutira) kako bi se formirala nova generacija. Nova generacija dobijenih rešenja kandidata se zatim koristi u sledećoj iteraciji algoritma. Uglavnom, algoritam se završava ili kada je proizveden maksimalni broj generacija ili je dostignut zadovoljavajući nivo fitnessa za populaciju [1].

Tipičan genetski algoritam zahteva:

1. Genetički prikaz domena rešenja
2. Fitness funkciju za procenu domena rešenja

Standardna reprezentacija svakog rešenja je niz bitova. Nizovi drugih tipova i struktura se u suštini mogu koristiti na isti nači. Glavna osobina koja čini ove genetske reprezentacije pogodne je da su njihovi delovi jako usklađeni zbog fiksne veličine što olakšava jednostavne operacije ukrštanja. Promenljiva dužina reprezentacije se takođe može koristiti, ali je implementacija ukrštanja u ovom slučaju složenija [2].

A. Inicijalizacija

Veličina populacije zavisi od prirode problema, ali obično sadrži nekoliko stotina ili hiljada mogućih rešenja. Često se početna populacija generiše nasumično, čime se omogućava čitav niz mogućih rešenja (prostor pretrage). Povremeno, rešenja mogu biti „posejana“ u oblastima gde je moguće naći optimalna rešenja [2].

B. Selekcija

Selekcija predstavlja izbor jedinki iz trenutne populacije koje će biti korišćenje za dobijanje naredne generacije. Jedinke mogu biti izabrane na osnovu vrednosti njihove funkcije pogodnosti, pri čemu se jedinkama sa boljom funkcijom pogodnosti daje veća verovatnoća da budu izabrane. Ovakva selekcija se često naziva i ruletska

selekcija, gde se pretpostavlja da na krugu veći isečak pripada jedinkama sa većom funkcijom pogodnosti. U literaturi se često koristi i turnirska selekcija. Kod nje se na slučajan način bira nekoliko jedinki, nakon čega se organizuje tzv. turnir, čiji je pobednik jedinka sa najboljom funkcijom pogodnosti, koja će učestvovati u selekciji [2].

C. Ukrštanje

Ukrštanje predstavlja kombinovanje gena dva roditelja, pri čemu se dobijaju dva nova potomka. U literaturi se najčešće javljaju jednopoziciono, dvopoziciono i uniformno ukrštanje. Kod jednopozicionog se uniformno bira gen i , gde prvi potomak nasleđuje gene prvog roditelja do i -tog i gene drugog roditelja od i -tog, a drugi potomak gene drugog roditelja do i -tog i prvog roditelja od i -tog. Kod dvopozicionog ukrštanja se uniformno biraju dva gena, i i j . Prvi potomak nasleđuje gene pre i -tog i nakon j -tog od prvog roditelja, a između gena i i j od drugog roditelja. Nasuprot tome, drugi potomak nasleđuje gene pre i -tog i nakon j -tog od drugog roditelja, a gene između i -tog i j -tog od prvog roditelja. Kod uniformnog ukrštanja, za svaki gen, odabir roditelja je slučajan [3].

D. Mutacija

Mutacija predstavlja slučajnu promenu nekih gena određene jedinke. Ona se na svaku jedinku primenjuje sa nekom malom verovatnoćom. Ukoliko je verovatnoća mutacije premala, ili se uopšte ne primenjuje, može doći do preuranjene lokalne konvergencije. Ako je vrednost velika, genetički algoritam će prostor rešenja pretraživati bez jasnog fokusa [3].

Iako je pretpostavka da će kvalitetni geni sa velikom verovatnoćom preći u narednu generaciju, to ne mora uvek da bude slučaj. Tako se može desiti da neki potomci budu lošiji od roditelja ili čak i da najbolja jedinka ne pređe u narednu generaciju. Jedan od najlošijih ishoda je da se u nekom trenutku dostigne rešenje koje je dobrog kvaliteta, a da se zatim izgubi primenom operatora ukrštanja i mutacije. Zato se često primenjuje princip elitizma, koji se zasniva na ideji da se najbolja ili više najboljih jedinki prenesu direktno u narednu generaciju [3].

E. Odabir parametara

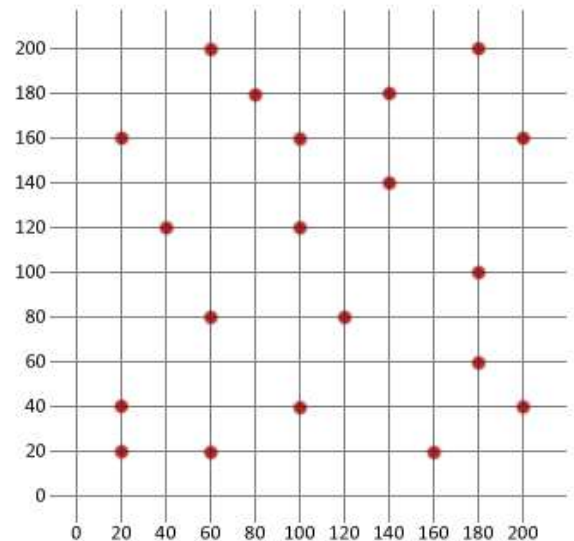
Pri implementaciji genetičkih algoritama, veoma važnu ulogu igra odabir parametara. Parametri mogu biti fiksni ili im se vrednosti mogu na određen način menjati iz generacije u generaciju. Njihove vrednosti variraju u zavisnosti od problema, a neke koje se često u praksi koriste su:

- Verovatnoća mutacije – između 0.005 i 0.01
- Verovatnoća ukrštanja – između 0.8 i 0.95
- Veličina populacije – između 10 i 150 jedinki. [3].

III. PROBLEM PUTUJUĆEG TRGOVCA

A. Formulacija problema

U ovom delu rada, biće opisan način konstruisanja genetičkog algoritma za rešavanje dobro poznatog problema putujućeg trgovca. Rešavanje ovog problema se svodi na sledeće: Ako je data lista gradova i poznato je kolika je udaljenost između svaka dva grada, koja je najkraća moguća putanja kojom bi trgovac mogao da obiđe svaki grad i da se vrati u grad iz kojeg je krenuo?



Slika 1. Primer rasporeda gradova koje trgovac treba da poseti.

Uzimajući sve u obzir, postoje dva važna ograničenja kojih se trgovac mora pridržavati:

1. Svaki grad dozvoljeno je posetiti samo jednom
2. Putujući trgovac se mora vratiti u grad iz kojeg je krenuo, tako da ukupna distanca mora da bude izračunata uzimajući u obzir ovo ograničenje

B. Pristup rešavanju problema

Na samom početku biće navedene neke definicije koje su prilagođene konkretnom problemu putujućeg trgovca:

- Gen – grad predstavljen svojim koordinatama (x i y)
- Hromozom – jedinstvena putanja koja zadovoljava uslove iskazane u prethodnom poglavlju
- Populacija – skup svih mogućih putanja
- Roditelji – dve putanje koje se mogu kombinovati i tako stvoriti novu putanju
- Selekcija – skup roditelja koji su izabrani da kreiraju novu populaciju (kreirajući sledeću generaciju mogućih putanja)

- Fitnes – funkcija koja daje informaciju o tome koliko je dobra putanja (u ovom slučaju, koliko je rastojanje)
- Mutacija – način da se uvede varijacija u populaciji tako što se proizvoljno zamene dva grada u putanji
- Elitizam – način da se najbolji geni prenesu u sledeću generaciju

Genetički algoritam, konstruisan za konkretan problem, izvršavaće se u sledećim koracima:

1. Kreiranje početne populacije
2. Određivanje vrednosti fitnes funkcije
3. Selekcija genoma za ukrštanje
4. Ukrštanje
5. Mutacija
6. Ponavljanje

IV. KREIRANJE GENETIČKOG ALGORITMA

Na početku biće upotrebljeni neki standardni paketi u programskom jeziku Python kako bi se olakšala konstrukcija genetičkog algoritma.

```
import numpy as np, random, operator,
pandas as pd, matplotlib.pyplot as plt
```

Slika 2. Standardni paketi u programskom jeziku Python.

A. Kreiranje klase Grad i Fitnes

Prvo će biti kreirana klasa *Grad* koja će omogućiti kreiranje u upravljanje gradovima. Ovo podrazumeva jednostavno unošenje koordinata (x i y) konkretnog grada. Unutar klase *Grad*, implementirana je funkcija *rastojanje* za računanje udaljenosti između dva grada. Takođe implementirana je i funkcija za jednostavniji prikaz gradova uz pomoć koordinata (slika 3) [4].

```
1 class Grad:
2     def __init__(self, x, y):
3         self.x = x
4         self.y = y
5
6     def Rastojanje(self, grad):
7         xRas = abs(self.x - grad.x)
8         yRas = abs(self.y - grad.y)
9         Rastojanje = np.sqrt((xRas**2)+(yRas**2))
10        return Rastojanje
11
12    def __repr__(self):
13        return "(" + str(self.x) + ", " + str(self.y) + ")"
```

Slika 3. Definisanje klase *Grad*.

Nakon kreiranja klase *Grad*, potrebno je kreirati i klasu *Fitnes*. U ovom slučaju, vrednost fitnes funkcije biće tretirana kao obrnuto srazmerna vrednosti rastojanja. Ovo znači da treba maksimizovati vrednost fitness funkcije, što

je vrednost veća to je rešenje bolje. Prema pravilu broj 2 iz odeljka formulacija problema, trgovac mora krenuti i završiti svoje putovanje na istom mestu, tako da je to rešeno u delu kreiranja klase *Fitnes* (slika 4) [4].

```
1 class Fitnes:
2     def __init__(self, putanja):
3         self.putanja = putanja
4         self.rastojanje = 0
5         self.fitnes = 0.0
6
7     def udaljenost(self):
8         if self.rastojanje == 0:
9             udaljenost_put = 0
10            for i in range(0, len(self.putanja)):
11                izGrada = self.putanja[i]
12                uGrada = None
13                if i+1 < len(self.putanja):
14                    uGrada = self.putanja[i+1]
15                else:
16                    uGrada = self.putanja[0]
17                udaljenost_put = udaljenost_put + izGrada.rastojanje(uGrada)
18            self.rastojanje = udaljenost_put
19            return self.rastojanje
20
21    def fitnesPut(self):
22        if self.fitnes == 0:
23            self.fitnes = 1/float(self.udaljenost)
24        return self.fitnes
```

Slika 4. Definisanje klase *Fitnes*.

B. Kreiranje početne populacije i određivanje vrednosti fitnes funkcije

Sada treba kreirati početnu populaciju, tj. prvu generaciju. Da bi bilo moguće to uraditi, treba napraviti funkciju koja će proizvesti putanje koje zadovoljavaju prethodno zadate uslove. Za kreiranje genoma, nasumice je generisan redosled posjećivanja gradova od strane putujućeg trgovca (slika 5) [4].

```
1 def kreirajPut(ListaGradova):
2     putanja = random.sample(ListaGradova, len(ListaGradova))
3     return putanja
```

Slika 5. Kreiranje individualnih genoma.

Ovaj postupak kreira samo individualni genom, ali za potrebe rešavanja problema potrebno je kreirati celu populaciju. Ovo je izvedeno tako što je prethodna funkcija za kreiranje genoma upotrebljena u petlji koja se izvršava onoliko puta kolika je veličina populacije. Ovakav način kreiranja populacije je upotrebljen samo pri inicijalnom kreiranju. Naredne generacije će biti kreiranje kroz procese ukrštanja i mutacije (slika 6) [4].

```
1 def inicijalPopulacija(velPop, ListaGradova):
2     populacija = []
3
4     for i in range(0, velPop):
5         populacija.append(kreirajPut(ListaGradova))
6     return populacija
```

Slika 6. Kreiranje početne populacije.

Kao sledeći korak u rešavanju problema, potrebno je odrediti vrednost fitnes funkcije. Ovo se može protumačiti kao postupak rangiranja najsposobnijih individua u populaciji. Kao izlaz, ova funkcija će dati identifikacioni

broj putanje kao i vrednost fitnes funkcije za nju (slika 6 i slika 7) [4].

```
1 def Inicijalizacija(velPop, listaGradova):
2     populacija = []
3
4     for i in range(0, velPop):
5         populacija.append(kreirajPut(listaGradova))
6     return populacija
7
8 def rangPut(populacija):
9     fitnesRez = {}
10    for i in range(0, len(populacija)):
11        fitnesRez[i] = fitnes(populacija[i]).fitnesPut()
12    return sorted(fitnesRez.items(), key = operator.itemgetter(1), reverse = True)
```

Slika 7. Fitnes funkcija.

C. Selekcija za proces ukrštanja

Postoji nekoliko načina za selekciju roditelja koji će biti iskorišćeni za stvaranje sledeće generacije. Najčešći načini su rulet selekcija (*eng. roulette wheel selection*) ili turnirska selekcija [4].

- Rulet selekcija – vrednost fitness funkcije svakog pojedinca se meri u odnosu na celokupnu populaciju i tako se određuje verovatnoća da taj pojedinac bude izabran.
- Turnirska selekcija – izdvaja se set nasumično izabranih pojedinaca populacije. Onaj pojedinac sa najvećom vrednosti fitness funkcije je izabran da bude pri roditelj. Postupak se ponavlja za izbor drugog roditelja [4].

Još jedan način za izbor genoma za ukrštanje je tzv. Elitizam. Kod ovog načina ukrštanja, najposposobniji pojedinci u populaciji će preneti svoj genetički materijal u sledeću generaciju, time omogućavajući opstanak najposposobnijih pojedinaca [4].

Za problem koji je aktuelan, geni za ukrštanje biće odabrani u dva koraka. Prvo će biti upotrebljen izlaz funkcije *Rangiranje_Putanja* za određivanje konkretnih putanja. Pošto je u pitanju ruletska selekcija, ovde se nasumično izabran broj poredi sa fitnes funkcijom pojedinačnog hromozoma. Takođe, najbolje putanje treba sačuvati pa je upotrebljen i elitizam. Na kraju, ova funkcija kao rezultat vraća listu putanja koje su iskorišćene za proces ukrštanja (slika 8) [4].

```
44 def Selekcija(popRang, elitVel):
45     RezSel = []
46     fd=pd.DataFrame(np.array(popRang), columns=["Indeks", "Fitnes"])
47     fd['kum_sum'] = fd.Fitnes.kum_sum()
48     fd['kum_per'] = 100*fd.kum_sum/fd.Fitnes.sum()
49
50     for i in range(0, elitVel):
51         RezSel.append(popRang[i][0])
52     for i in range(0, len(popRang)-elitVel):
53         izaberi = 100*random.random()
54         for i in range(0, len(popRang)):
55             if izaberi <= fd.iat[i,3]:
56                 RezSel.append(popRang[i][0])
57                 break
58     return RezSel
```

Slika 8. Prvi korak u selekciji roditelja.

Drugi korak se zasniva na tome da iz liste koja je kreirana u prvom koraku budu izvučene putanje koje će se ukrstiti i tako stvoriti potomke (slika 9) [4].

```
69 def Par_Ukr(populacija, RezSel):
70     Par_Ukr = []
71     for i in range(0, len(RezSel)):
72         index = RezSel[i]
73         Par_Ukr.append(populacija[index])
74     return Par_Ukr
```

Slika 9. Drugi korak u selekciji roditelja.

D. Ukrštanje

Kako su izabrani hromozomi roditelji, sada se može pristupiti kreiranju generacije potomaka. Ovaj proces kod genetičkih algoritama se naziva ukrštanje. Ako se pretpostavi da su hromozomi predstavljeni nizovima 0 i 1 i da dva pravila koja su navedena na početku ne važe, najprostije se može izabrati tačka ukrštanja i postojeći nizovi bi bili podeljeni i tako stvorili potomke (slika 9) [4].

Međutim, kod rešavanja problema putujućeg trgovca sve lokacije moraju biti posećene samo jednom. Da bi ovo pravilo ostalo na snazi, uvode se specijalna funkcija koja se naziva uređeno ukrštanje. Kod ovakvog načina ukrštanja, nasumice se bira pod-skup gena iz skupa prvog roditelja a ostatak se nadopunjuje sa genima drugog roditelja. Pri tome se mora voditi računa da ne postoji dupliranje gena u izabranom pod-skupu prvog roditelja. Na kraju, da bi se obezbedio opstanak najposposobnijih hromozoma, i ovde je upotrebljen princip elitizma (slika 10 i slika 11) [5].

```
76 def Ukrstanje(roditelj_1, roditelj_2):
77     potomak = []
78     potomak_1 = []
79     potomak_2 = []
80
81     genA = int(random.random() * len(roditelj_1))
82     genB = int(random.random() * len(roditelj_2))
83
84     startGen = min(genA, genB)
85     endGen = max(genA, genB)
86
87     for i in range(startGen, endGen):
88         potomak_1.append(roditelj_1[i])
89
90     potomak_2 = [item for item in roditelj_2 if item not in potomak_1]
91
92     potomak = potomak_1 + potomak_2
93     return potomak
```

Slika 10. Uređeno ukrštanje.

```
96 def UkrstajPopu(Par_Ukr, elitVel):
97     potomci = []
98     duzina = len(Par_Ukr) - elitVel
99     uzorci = random.sample(Par_Ukr, len(Par_Ukr))
100
101     for i in range(0, elitVel):
102         potomci.append(Par_Ukr[i])
103     for i in range(0, duzina):
104         potomak = Ukrstanje(uzorci[i], uzorci[len(Par_Ukr)-i-1])
105         potomci.append(potomak)
106     return potomci
```

Slika 11. Princip elitizma kod ukrštanja.

E. Mutacija

Mutacija ima veoma važnu ulogu pri projektovanju genetičkog algoritma. Ona pomaže izbegavanju lokalne

konvergencije uvodeći nove putanje koje pomažu pri istraživanju drugih delova prostora rešenja. Slično kao kod procesa ukrštanja, aktuelni problem putujućeg trgovca mora biti posebno razmotren i sa polja mutacije. Opet, ako bi se hromozom sastojao od niza 0 i 1, mutacija bi značila da postoji mala verovatnoća da gen pređe sa 0 na 1 i obratno [4].

Međutim, kako odavde treba ispoštovati određena pravila, određeni gradovi se ne mogu izostaviti. Umesto ovoga biće korišćena mutacija zamene. Ovo znači da će sa predodređenom verovatnoćom, dva grada zameniti mesta na putanji. Ovo će biti izvedeno funkcijom *Mutacija* (slika 12) [5].

```
109 def Mutacija(poedinac, stopaMut):
110     for zamena in range(len(poedinac)):
111         if(random.random() < stopaMut):
112             zameni_sa = int(random.random() * len(poedinac))
113
114             grad_1 = poedinac[zamena]
115             grad_2 = poedinac[zameni_sa]
116
117             poedinac[zamena] = grad_2
118             poedinac[zameni_sa] = grad_1
119     return poedinac
```

Slika 12. Mutacija primenjena na jednu putanju.

Nakon primene na jednoj od putanja, sada funkcija *Mutacija* može biti primenjena na celu populaciju (slika 13).

```
122 def MutPop(populacija, stopaMut):
123     Mutir_Pop = []
124
125     for i in range(0, len(populacija)):
126         mutirani_poj = Mutacija(populacija[i], stopaMut)
127         Mutir_Pop.append(mutirani_poj)
128     return Mutir_Pop
```

Slika 13. Mutacija cele populacije.

F. Ponavljanje

Na kraju je potrebno samo sklopiti sve delove ove slagalice. Prvo je potrebno rangirati putanje upotrebom funkcije *Rangiranje_Putanja*. Nakon toga upotrebom funkcije *Selekcija* biraju se potencijalni roditelji hromozomi. Na kraju se izabrani hromozomi ukrštaju i tako stvaraju novu populaciju upotrebom funkcija *Razmnožavanje* i *Mutacija* (slika 14) [5].

```
130 def sledGen(trenutGen, elitVel, stopaMut):
131     RangPop = rangPut(trenutGen)
132     RezSel = Selekcija(RangPop, elitVel)
133     Pop_za_Ukr = Par_Ukr(trenutGen, RezSel)
134     potomci = UkrsPopu(Pop_za_Ukr, elitVel)
135     sledGen = MutPop(potomci, stopaMut)
136     return sledGen
```

Slika 14. Proces ponavljanja.

Poslednji korak u kreiranju genetičkog algoritma je njegovo izvršavanje. Sve što treba uraditi je kreiranje

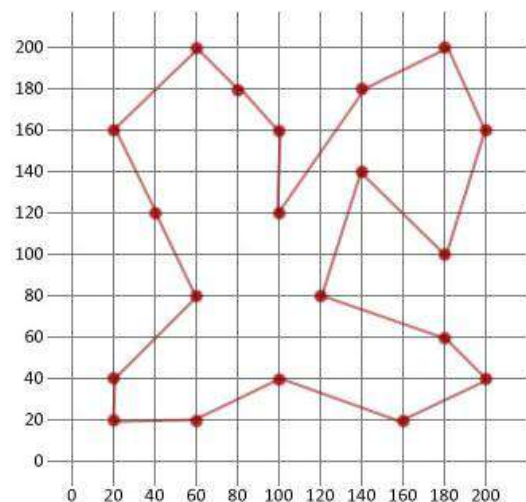
početne populacije i nakon toga se može prolaziti kroz proizvoljan broj generacija (slika 15) [5].

```
150 def genAlgoritam(populacija, velPop, elitVel, stopaMut, generacije):
151     pop = inicijalizacija(velPop, populacija)
152     print("Pocetno rastojanje:" + str(1/rangPut(pop)[0][1]))
153
154     for i in range(0, generacije):
155         pop = sledGen(pop, elitVel, stopaMut)
156
157         print("Konacno rastojanje:" + str(1/rangPut(pop)[0][1]))
158         indeks_najbolje_put = rangPut(pop)[0][0]
159         najbolja_put = pop[indeks_najbolje_put]
160         return najbolja_put
161
162 listaGradova = []
163
164 for i in range(0,35):
165     listaGradova.append((grad(int(random.random() * 200), y=int(random.random() * 200))))
166
167 genAlgoritam(inicijalizacija, velPop = 100, elitVel = 20, stopaMut = 0.01, generacije = 500)
```

Slika 15. Pokretanje genetičkog algoritma.

V. REZULTATI

Prilikom izvršavanja genetičkog algoritma upotrebljeni su sledeći parametri. Broj hromozoma u svakoj generaciji je 100, broj sačuvanih elitnih hromozoma 20, verovatnoća mutacije 0.01, broj izvršavanja koda kroz 500 generacija (slika 16 i slika 17) [4].



Slika 16. Optimalna putanja između gradova.

```
Pocetno rastojanje: 1998
Konacno rastojanje: 945
[60, 200|20, 160|40, 120|60, 80|
 20, 40|20, 20|60, 20|100, 40|160, 20|200, 40|
 180, 60|120, 80|140, 140|180, 100|200, 160|180, 200|
 140, 180|100, 120|100, 160|80, 180|
```

Slika 17. Rezltat nakon izvršenja programa.

VI. ZAKLJUČAK

U ovom radu prikazan je način rešavanja optimizacionih problema pomoću genetičkih algoritama. Kao reprezentativni primer uzet je problem putujućeg trgovca. U radu je za pisanje programa korišćen programski jezik *Python*. Kroz rad su dat detaljan opis način pisanja programa.

Kao eventualno poboljšanje postojećeg programa, može se ispitati zavisnost rastojanja od svake generacije u procesu izvršenja. Na taj način se može uvideti koliki je optimalni broj generacija za koje program postiže zadovoljavajuće rezultate. Samim tim program se može dodatno optimizovati u pogledu trajanja izvršenja. Ovo će možda biti tema u nekom od sledećih radova.

LITERATURA

- [1] <http://elibrary.matf.bg.ac.rs/bitstream/handle/123456789/1894/genetski%20algoritam.pdf?sequence=1>
- [2] Risshal Kurbanss, "Algoritmi veštačke inteligencije", 1. izdanje, pp. 227-279.
- [3] Sebastian Raschka, Vahid Mirjalili, "Python mašinsko učenje", 3. izdanje, pp. 259-285
- [4] <https://www.theprojectspot.com/tutorial-post/applying-a-genetic-algorithm-to-the-travelling-salesman-problem/5>
- [5] <https://repozitorij.pmf.unizg.hr/islandora/object/pmf%3A5227/datastream/PDF/view>

Pametni šor

Nikola Simić, Marko Milijanović, Aleksandar Krajačević, Ognjen Milojević, Nikola Velemir, Filip Goldberger, Albert Anđelkovski, Branislav Hrcan, Matej Milankov, Tomislav Jovanović

Sadržaj – U ovom radu je opisan princip rada projekta “Pametni šor”, kao i komponente i alati korišćeni za njegovu izradu.

projekta: pametnisor.pilab021.com.

I. UVOD

Pametni šor je platforma za upravljanje *IoT* (*Internet of Things*) projektima razvijena sa ciljem jednostavne distribucije podataka između *IoT* čvorova i servera i njihove kontrole, ali i sa ciljem edukacije u sferi *IoT*-a.

Glavne komponente koje su korišćene za izradu ovog projekta su mikroracunar *Raspberry Pi* [1] i centralni web server sa podrškom za *NodeJs* [2] i *Python* [3].

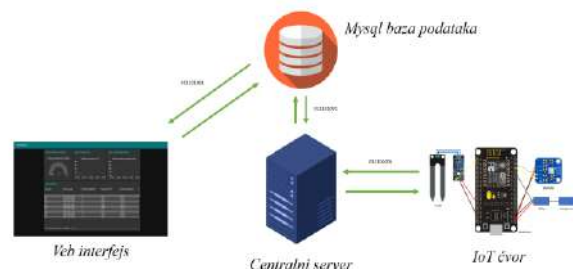
II. PRINCIP RADA

Centralni server pametnog šora konstantno razmenjuje podatke sa *IoT* čvorovima preko različitih protokola, oni se zatim skladište u bazu podataka i prikazuju u interfejsu veb aplikacije čvora (slika 1). Čvorovi su izrađeni na različitim platformama i koriste različite protokole za komunikaciju sa centralnim serverom. Zbog ograničenog broja strana u ovom dokumentu biće opisan princip rada samo jednog čvora – “Beočin 1” (slika 2) koji je izrađen na platformi *NodeMCU* [4] koja se zasniva na *ESP8266* mikročipu [5] i koji koristi *MQTT* protokol [6] za razmenu podataka sa serverom. Pregled ovog, drugih čvorova, njihovih merenja i detaljnijeg opisa samog projekta možete naći na veb sajtu

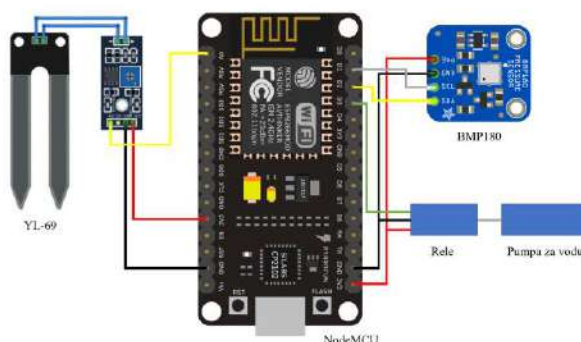
N. Simić i O. Milojević su studenti druge godine na smeru Računarstvo i automatika, M. Milijanović je student druge godine na smeru Softversko inženjerstvo i informacione tehnologije, Fakultet tehničkih nauka Novi Sad, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21102 Novi Sad, Srbija.
E-mail: nikola7simic@gmail.com, ognjenmilojevic88@gmail.com marko.milijanovic@yahoo.com.

A. Krajačević je bivši učenik smera Elektrotehničar informacionih tehnologija, srednja elektrotehnička škola “Mihajlo Pupin” Novi Sad, Futoška 17, 21102 Novi Sad, Srbija.
E-mail: aleksandarkrajacevic@gmail.com.

N. Velemir je učenik IV razreda smer Elektrotehničar informacionih tehnologija, F. Goldberger je učenik IV razreda smer Elektrotehničar računara, A. Anđelkovski, B. Hrcan, M. Milankov i T. Jovanović su učenici III razreda smer Administrator računarskih mreža, srednja elektrotehnička škola “Mihajlo Pupin” Novi Sad, Futoška 17, 21102 Novi Sad, Srbija.
E-mail: vnexa021@gmail.com, filip.gold159@gmail.com, albertandjelkovski@gmail.com, hrcanbaki@gmail.com, matej.milankov@gmail.com, tomlavjovanovic04@gmail.com.



Slika 1. Razmena podataka između *IoT* čvora i centralnog servera.



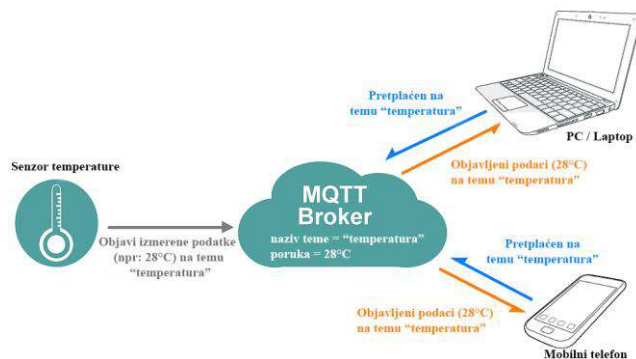
Slika 2. Šema čvora „Beočin 1“.

“Beočin 1” je *IoT* čvor koji meri vlažnost zemljišta, vazdušni pritisak i temperaturu. Koristi senzor vlažnosti zemljišta *YL-69* [7] i barometarski senzor *BMP180* [8]. U okviru čvora priključena je i pumpa za vodu koja se uključuje u slučaju niskog očitavanja vlažnosti zemljišta.

Ovaj čvor koristi *MQTT* protokol [6] za razmenu podataka sa centralnim serverom. *MQTT* je jednostavan protokol koji je baziran na sistemu pretplate i objave (slika 3) koji je namenjen prenosu kratkih poruka. Očitavanja senzora se objavljuju na njima namenjenu temu (*topic*) kao što je na primer “/senzor1/temperatura”, a uređaji koji žele da prime poruku moraju da budu pretplaćeni na tu istu temu. Poruka se prvo objavljuje do *broker*-a, a on je zatim prosleđuje pretplatnicima. Prednost ovog pristupa je to što je omogućena jednostavna distribucija iste poruke do svih zainteresovanih korisnika, a da pri tome nije potrebno da uređaj koji objavljuje poruku

zna detalje o pretplatnicima (njihovu adresu, stanje veze itd). Pametni šor koristi *mosquito broker* koji je instaliran na *Raspberry Pi-u*.

Očitavanja sa *YL-69* senzora se objavljuju na svakih pet sekundi, a očitavanja sa *BMP180* senzora se objavljuju na svakih minut.



Slika 3. Mehanizam pretplate i objave kod *MQTT* protokola.

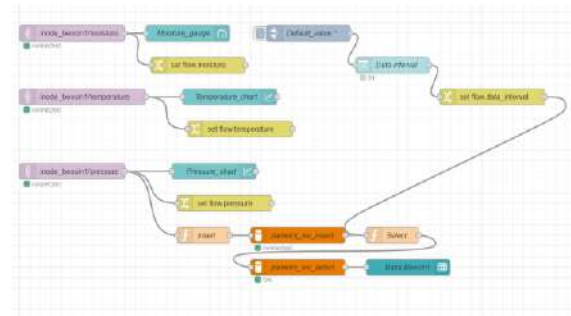
Dalji proces obrade podataka odvija se na serveru. Alat koji je korišćen za izradu serverskog dela i veb prikaza podataka za ovaj čvor je *Node-RED* [9].

Node-RED je programski alat koji funkcioniše na principu *flow-based* [10] programiranja. Prvobitno je razvijen od strane *IBM*-ovog *ETS* tima (*Emerging Tehnology Services team*), a sada je deo *JS* fondacije.

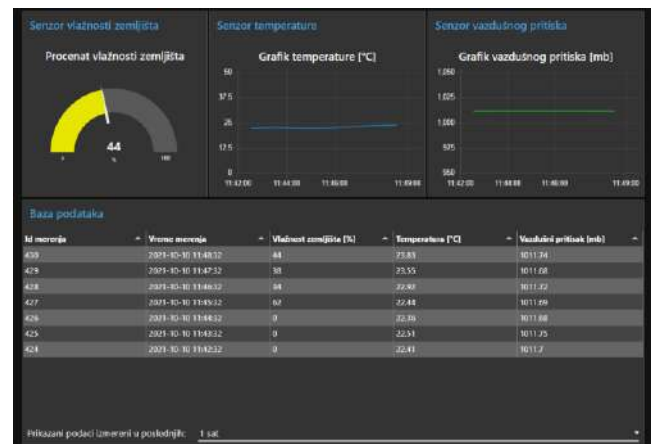
Node-RED pokreće *NodeJs* okruženje, a *flow-editor-u* se pristupa preko ip adrese na kojoj radi *Node-RED* server kroz veb pretraživač. U *editor-u* aplikacija se kreira prevlačenjem čvorova (*nodes*) iz menija na radni prostor i njihovim međusobnim povezivanjem. Klikom na dugme *Deploy* aplikacija je pokrenuta. Paleta čvorova se može proširiti i na lak način se mogu instalirati novi čvorovi kreirani od strane *Node-RED* zajednice, a tokovi (*flows*) koji se kreiraju mogu biti lako podeljeni kao *JSON* datoteke.

Tok „Beočin 1“ (slika 4) zadrži tri *MQTT_in* čvora, svaki pretplaćen na zasebnu *MQTT* temu za svaki podatak koji se dobija sa *IoT* čvora. Pri pristizanju podataka na odgovarajuću temu oni se prosleđuju, zapisuju se u bazu podataka i prikazuju se na komandnoj tabli (veb interfejsu ugrađenom u *Node-RED*) (slika 5). Korisniku je omogućeno da izabere prikaz podataka upisanih u poslednjih 1, 2, 6, 12 ili 24 sata tako što će odgovarajući interval izabrati iz liste na dnu tabele.

Svako može da pristupi detaljnom opisu čvora i pregledu njegovih očitavanja preko veb portala „Pametni Šor“ koji se nalazi na adresi: pametnisor.pilab021.com (slika 6, slika 7).



Slika 4. Tok „Beočin 1“.

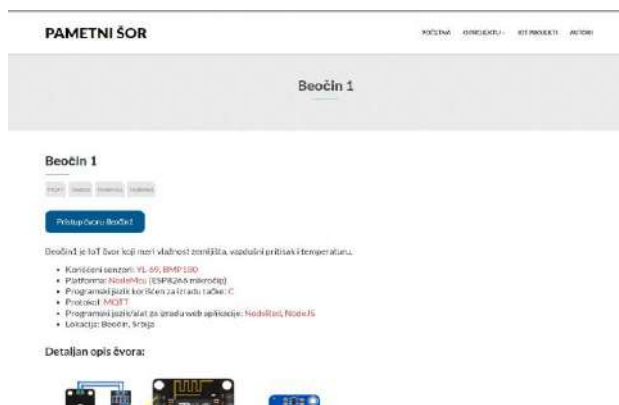


Slika 5. *Node-RED* komandna tabla.

*Napomena: Zbog problema sa hosting provajderom, *NodeRed* server je trenutno instaliran na *Raspberry Pi* koji se nalazi na udaljenoj lokaciji. U budućnosti *NodeRed* server će biti smešten na centralnom serveru kao i sve ostale aplikacije.



Slika 6. Veb portal „Pametni Šor“.



Slika 7. Stranica čvora "Beočin 1" na veb portalu.

III. IMPLEMENTACIJA

A. Hardverska implementacija

Glavna hardverska komponenta čvora „Beočin 1“ je sama platforma na kojoj je izrađen, *NodeMCU* [4] (slika 8).

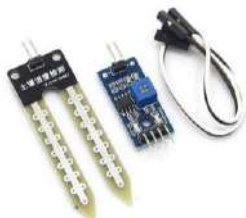


Slika 8. NodeMCU.

NodeMCU je jeftina *open-source IoT* platforma zasnovana na *ESP8266* mikročipu [5]. *NodeMCU* takođe pruža neke od najbitnijih odlika mikrokontrolera kao što su *GPIO* (*general purpose input - output*) pinovi, *PWM* (*pulse-width modulation*), *A/D* konvertor itd.

Na *NodeMCU* su povezani senzor vlažnosti zemljišta *YL-69* i barometarski senzor *BMP180*.

YL-69 [7] (slika 9) senzor se sastoji iz dva dela: ploče sa elektronikom i sonde sa dva pada koja detektuje količinu vode. Senzor ima ugrađeni potenciometar za podešavanje osetljivosti digitalnog izlaza, ugrađenu *LED* diodu koja označava da li je senzor uključen i *LED* diodu koja označava da senzor odašilje signal na digitalni izlaz. Izlazni signal može biti digitalni ili analogni. Što je vlažnost veća napon izlaznog signala je manji i obrnuto.



Slika 9. YL-69 senzor vlažnosti zemljišta.

BMP180 [8] (slika 10) senzor je digitalni senzor pritiska koji meri apsolutni pritisak vazduha oko njega. Raspon merenja koji pruža ovaj senzor je od 300 hPa do 1100 hPa (hPa = mb). Pošto temperatura utiče na pritisak senzor *BMP180* dolazi i sa senzorom za temperaturu.



Slika 10. BMP180 barometarski senzor.

Pored ova dva senzora preko relea je povezana pumpa za vodu koja se uključuje u slučaju očitavanja niske vlažnosti zemljišta.

Što se tiče hardverskih komponenti samog serverskog dela, potreban je server koji je u ovom slučaju zakupljen od hosting provajdera i hardver koji je sposoban da pokreće *MQTT Broker*. Za te potrebe ovaj projekat koristi mikroračunar *Raspberry Pi* [1] (slika 11).



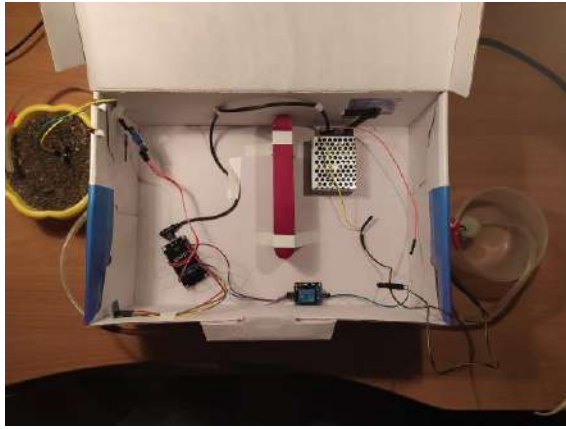
Slika 11. Raspberry Pi model 3 b+.

Raspberry Pi 3 b+ model koristi četvorojezgarni *ARM Cortex-A53* procesor koji radi na frekvenciji od 1,4 GHz. Posедуje 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 se napaja preko eksternog napajanja od 5 V sa kojeg je takođe izvedeno napajanje za pumpu za vodu. Oba senzora se napajaju preko 3.3 V pinova sa samog *NodeMCU*-a. Elektronika čvora se nalazi u privremenom kućištu (slika 12) dok se ne izradi posebno kućište koje će moći da se montira unutar objekata kao i spolja. Kućište za čvor je u fazi projektovanja i biće izrađeno uz pomoć 3D štampača.

B. Softverska implementacija

Softverska implementacija *IoT* čvora zavisi od same platforme na kojoj je čvor izgrađen i želje korisnika u slučaju da ima više izbora. U slučaju čvora „Beočin 1“ korišćeno je razvojno okruženje *Arduino* [11] i program je napisan u programskom jeziku *C* [12]. Najbitniji deo koda jeste konektovanje i obavljanje komunikacije preko *MQTT* protokola (slika 13, slika 14).



Slika 12. Privremeno kućište za čvor „Beočin 1“.

```
51 void mqtt_reconnect() {
52   while(!client.connected()) {
53     Serial.print("Attempting MQTT connection...");
54
55     if(client.connect("ESP8266-Beocin1")) {
56       Serial.println("Connection successful");
57     } else {
58       Serial.print("Connection failed, rc=");
59       Serial.print(client.state());
60       Serial.println(" try again in 5 seconds");
61       delay(5000);
62     }
63   }
64 }
```

Slika 13. Funkcija za ponovno konektovanje sa MQTT broker-om.

```
177 if(current_time - timer_2 > 60000){
178   client.publish("/node_beocin1/temperature", pub_temp_value);
179   client.publish("/node_beocin1/pressure", pub_pressure_value);
180   timer_2 = current_time;
181 }
```

Slika 14. Objavljivanje pročitanih podataka na njima namenjene teme.

Softverska implementacija na serverskoj strani uključuje kreiranje tabele u bazi podataka u koju će podaci da se opisuju, i kreiranje samog 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 (slika 15).



Slika 15. Pisanje funkcije u sastavu *function node*-a.

Centralni server je standardni veb server (Apache, PHP, MySQL + podrška za *NodeJs* i *Python*), zakupljen kod hosting provajdera. Povezani IoT čvorovi u realnom vremenu komuniciraju sa centralnim serverom i razmenjuju podatke.

IV. MERENJA

Tokom testiranja čvor se nalazi unutar stambenog objekta i samim tim manje su varijacije vrednosti temperature i vazdušnog pritiska (slika 16). Izradom kućišta i montiranjem samog na otvoreno merenja će prikazivati realnu sliku spoljne temperature koja će moći i da se primeti na grafiku temperature u grafičkom interfejsu. U budućnosti svaki čvor će pored svoje primarne funkcije na sebi imati i senzor za merenje kvaliteta vazduha i izmereni podaci će biti javno dostupni.

id merenja	Vreme merenja	Vlažnost vazduha [%]	Temperatura [°C]	Vazdušni pritisak [mbar]
1320	2021-10-31 15:11:19	36	21,27	1005,02
1321	2021-10-31 15:41:19	35	21,61	1005,04
1322	2021-10-31 15:11:19	46	21,3	1005,21
1323	2021-10-31 17:41:19	45	21,41	1005,14
1324	2021-10-31 17:11:19	55	21,28	1005,07
1325	2021-10-31 17:11:14	55	21,15	1005,12
1326	2021-10-31 16:52:28	58	21,26	1005,04
1327	2021-10-31 16:52:18	58	21,25	1004,98
1328	2021-10-31 16:53:08	58	21,26	1005,02

Slika 16. Tabela sa zabeleženim merenjima čvora "Beočin 1".

V. ZAKLJUČAK

IoT tehnologije postaju nezaobilazne u svim sferama života i samim tim su potrebni sistemi kao što je Pametni Šor koji će na jednostavan način obavljati komunikaciju sa IoT čvorovima, čuvati dobijene podatke i pružiti lak način da se korisnik edukuje u ovoj oblasti.

Zbog ograničenja koja su definisana ovaj rad ne može da prikaže sve mogućnosti ovog projekta. Pregled svih čvorova, detaljniji opisi i tekstovi o alatima, protokolima i temama vezanim za IoT se nalaze na veb portalu: pametnisor.pilab021.com.

Unapređenja predviđena za sledeći period: Nastaviti sa dodavanjem novih IoT čvorova, istraživanjem novih tehnologija u ovoj oblasti i ažuriranjem veb portala.

ZAHVALNICA

Zahvaljujemo se mentorima Ušan Branislavu, Skala Vladimiru, Vukobrat Milanu i Pavlović Jovi na usmeravanju i sugestijama pri izradi ovog rada kao i ostalim članovima sekcije na pruženoj podršci.

LITERATURA

- [1] Raspberry Pi: <https://www.raspberrypi.org>
- [2] Node.js: <https://nodejs.dev/learn>
- [3] Python: <https://www.python.org/>
- [4] NodeMCU: <https://electropeak.com/learn/nodemcu-esp8266-on-arduino-ide/>

- [5] ESP8266: <https://www.espressif.com/en/products/socs/esp8266>
- [6] MQTT: https://www.rtrk.uns.ac.rs/sites/default/files/materijali/predavanja/Orm2_Bel'eske_Predavanje12.pdf
- [7] YL-69: <https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/>
- [8] BMP180: <https://randomnerdtutorials.com/esp32-with-bmp180-barometric-sensor/>
- [9] Node-RED: <https://nodered.org/about/>
- [10] Flow-based programming: https://en.wikipedia.org/wiki/Flow-based_programming
- [11] Arduino IDE: <https://www.arduino.cc/en/software>
- [12] C: <https://www.tutorialspoint.com/cprogramming/index.htm>

Sistem za testiranje RGB matričnih ploča

Aleksandar Vučković

Sadržaj – Osnovni cilj ovog rada jeste da se analizira i kreira sistem koji bi pružio pomoć korisniku prilikom ispitivanja matričnih štampanih ploča sa RGB svetlećim diodama.

I. UVOD

Obzirom da u poslednje vreme elektronika brzo napreduje, svetleće diode su se drastično promenile od svog nastanka. Tako, danas postoji mnoštvo tipova svetlećih dioda [1] koje se razlikuju po tipu kućišta, tehnologiji izrade, boji koju emituju, oblasti primene... Ovde su od značaja svetleće diode koje mogu da emituju pun spektar boje tzv. *full color diode* [2] (mešanjem boja dobija se pun spektar boja) koje su zalemljene na štampanu ploču na određenim pozicijama čime formiraju matricu – RGB matrična štampana ploča.

Ove štampane ploče imaju primenu u svetlosnim saobraćajnim znacima koji se najčešće mogu videti na autoputevima ili gradskim saobraćajnicama u zoni škola. Svaki od ovih znakova se sastoji od nekoliko desetina pa i do nekoliko stotina pojedinačnih matričnih ploča a sve to sa ciljem da se prikaze tekst, izgled saobraćajnog znaka, specifičan oblik i sl.

Razlog za razvoj ovog sistema je proizašao iz potrebe da se sa veicom sigurnošću utvrdi ispravnost matrične štampane ploče koja će biti ugrađena u određeni saobraćajni znak. Ukoliko u nekom trenutku dođe do problema sa saobraćajnim znakom, sumnja na neispravnost matričnih ploča je jako mala obzirom da su prošle detaljno testiranje.

II. KARAKTERISTIKE RGB LE DIODA I LED

DRAJVERA

Proizvođač LE dioda daje podatke o ovoj komponenti u svom *datasheet*-u na osnovu koga se dobijaju potrebni podaci prilikom projektovanja štampane ploče. Najvažniji podaci koji su potrebni za projektovanje štampane ploče jesu tip kućišta LE diode, nazivni napon i struja pri određenim uslovima, maksimalne vrednosti napona i struje, osvetljenost LE diode (u mcd), preporučeni izgled otiska na ploči (eng. *footprint*), maksimalna disipacija snage kao i da li je u pitanju LE dioda sa zajedničkom anodom ili zajedničkom katodom.

Kada govorimo o prikazu na svetlosnom saobraćajnom znaku, potrebno je uključiti određene LE

dioda ili određenu grupu LE dioda. Pošto ukupan broj LE dioda po saobraćajnom znaku prelazi i broj od 1000 komada, neophodno je koristiti specijalizovane čipove odnosno LED drajvere. Ova integrisana kola, čiji je tipičan predstavnik čip TLC5955 [3], imaju najčešće 48 *constant-current* izlaza (po 3 za svaku LE diodu) i koriste SPI protokol komunikacije za prenos podataka između drajvera i *master* uređaja. Podaci koje LED drajver primi se manifestuju uključivanjem određenih izlaza koji mogu snabdevati LE diode različitim vrednostima struje na osnovu čega se dobija željeni prikaz na svetlosnom saobraćajnom znaku. Pored kontrole prikaza, ovi drajveri imaju dve veoma bitne karakteristike vezane za ispravnost LE diode – detekcija greške prekida LE diode i detekcija greške kratkog spoja diode. Zahvaljujući ovim osobinama, omogućena je povratna informacija o ispravnosti pojedinačnih LE dioda.

III. ZAHTEVI SISTEMA

Kako bi se proverila ispravnost matrične štampane ploče potrebno je da se ispitaju sledeći zahtevi:

- da li svetli tačno određena dioda kao što je to softverom definisano,
- da li svetli tačno određenom bojom, na tačno određenoj poziciji,
- da li sve diode svetle istom jačinom,
- da li linije veze na štampanoj ploči prenose dobro signale,
- da li su naponi od 5 i 3,3 V u spoju,
- da li ploča prikazuje onaj šablon definisan softverom i
- da li funkcionišu grejači na ploči.

Gore navedeni zahtevi su proizašli iz potrebe da se pokrije što više testnih procedura i da se na taj način pokaže da je štampana ploča sa LE diodama ispravna i funkcionalna. Uz to, obzirom da ovaj sistem u suštini predstavlja jedan uređaj koji radi po unapred utvrđenoj proceduri, greške prilikom testiranja su svedene na minimum.

IV. KOMPONENTE SISTEMA

Ceo ovaj sistem će se sastojati iz više komponenata koje su međusobno povezane u jednu celinu. Osnovne komponente sistema za testiranje ploča su:

- Aluminijumsko kućište,
- WEB kamera,
- CPU ploča,
- Ekstenziona ploča,

A. Vučković, dipl. inž. elektrotehnike i računarstva, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.vuckovic@elfak.rs

- Štampana ploča za električno ispitivanje ploče sa LE diodama,
- Kablovi i konektori za povezivanje,
- Napajanja.

A. Aluminijumsko kućište

Kućište je izrađeno od aluminijuma jer je kao materijal lak za obradu i ima malu masu, a solidnu čvrstinu. Kućište je oblika kocke sa stranicom od 350 mm. Otvara se sa prednje strane ka spolja, na gornjoj strani se nalazi otvor za objektiv kamere koja se nalazi spolja, a preko nje se nalazi poklopac koji sprečava da svetlost iz spoljašnjosti padne na objektiv kamere. Sa donje strane se nalaze 4 odstoynika od po 20 mm u 4 temena kvadrata a donja stranica kvadrata je izrezana po sredini i dijagonalno ka temenima kvadrata.

Razlog za ovakav oblik donje stranice je taj što je centralni deo izrezan da bi tu prošli kablovi do štampane ploče sa LE diodama, a 4 tanke dijagonalno izrezane površine imaju ulogu vodiča za klizače. Ti klizači se nalaze u unutrašnjosti testera i njihova osnovna namena je da, bez obzira na dimenziju ploče, ploča uvek bude postavljena na podjednako udaljenosti od sve 4 stranice kvadrata odnosno da bude postavljena na sredinu donje stranice.

Sa zadnje stranice kućišta se nalaze 2 DIN šine (industrijska šina predviđena za montažu prekidača, osigurača, napajanja...) koje su postavljene horizontalno i paralelno jedna u odnosu na drugu. Gornja DIN šina je predviđena za 4 napajanja a donja za 3 štampane ploče. Izgled ovog kućišta se može videti na slici 1.



Slika 1. Izgled aluminijumskog kućišta.

B. WEB kamera

Možda i najteži zadatak za ovaj sistem je bio odabrati kameru koja će imati zadovoljavajuću rezoluciju, da ne zauzima previše mesta, da ima što bolji odnos cene i kvaliteta, da odlično reaguje na promene svetlosti (da ima što brži odziv) i da joj se karakteristike ne smanjuju previše kada snima u mraku.

Od više testiranih kamera, najbolje se pokazala kamera *Logitech C920 HD Pro* koja je zadovoljila većinu ovih uslova. Kamera snima u *Full HD* rezoluciji u 60 frejma po sekundi ima širok ugao snimanja svetlosti koja

pada na senzor i koristi USB interfejs za komunikaciju. Sam proizvođač nudi aplikaciju kojom je moguće podešavati rezoluciju kamere, broj frejmova po sekundi, autofokus ili manualni fokus zatim osvetljenje, kontrast, jačinu boje itd. Ova aplikacija nudi dosta mogućnosti podešavanja video snimka što dosta pomaže.

C. CPU ploča

Centralnu procesorsku jedinicu ovog sistema predstavlja CPU (Central Processing Unit) ploča koja je kontrolisana softverom. Koristi mikrokontroler *Atmel AT89C51ED2* [4] proizvođača *Microchip*. U pitanju je 8-bitni mikrokontroler koji ima 4 porta sa po 8 ulazno-izlaznih pinova (GPIO). Iako je mikrokontroler nastao 2003. godine i dalje poseduje mnoštvo funkcija da zadovolji današnje zahteve za kontrolu LE dioda.

Ova ploča ima mogućnost komuniciranja korišćenjem protokola *Ethernet* i *RS232* kao i *I2C* [5] koja je od značaja za ovaj sistem. Što se tiče komunikacionih protokola koje ploča koristi razlika je uglavnom u brzini prenosa podatka, daljini prenosa, jednostavnosti i pristupu ploči. Uglavnom se putem serijske komunikacije čitaju ili upisuju podaci kada se ploča ne nalazi na mreži odnosno kada ploča nema pristup internetu dok se putem protokola *Ethernet* komunikacije lako pristupa ploči putem njene *MAC* ili *IP* adrese i u većini slučajeva se na ovaj način vrši upis željenog softvera u memoriju mikrokontrolera. Na slici 2. se može videti izgled CPU ploče.



Slika 2. Izgled CPU ploče.

Obzirom da se nekad dešava da se CPU ploča nalazi daleko od štampane ploče sa LE diodama, signali koji se šalju putem flet kablova do LED drajvera mogu da oslabe do nivoa na kom drajveru neće biti jasno da li je poslata logička 1 ili logička 0. Iz tog razloga se izlazni signali stabilisu kako bi se bezbedno preneli kroz flet kabl. Kolo koje se koristi za ovu svrhu je osmo-ulazni neinvertujući bafer sa tri-state izlazima.

D. Ekstenziona ploča

Ekstenziona ploča ima jednostavnu ulogu – proslediti signale sa CPU ploče na sve štampane ploče sa LE

diodama. Ponekad je potrebno drajvovati veliki broj ploča istovremeno a to je jako teško postići jednim flet kablom pa se iz tog razloga koristi ekstenziona ploča.

Ekstenziona ploča ili skraćeno ekstenzija može proslediti signale do 4 vertikalna flet kabla koji mogu da drajvuju do 8 ploča po vertikali odnosno 32 ploče. Na ulazu ekstenzije se nalazi šesnaestopinski flet konektor koji prima signale sa CPU ploče i odmah ih prosleđuje na ulaze istog bafera koji se nalazi pre flet konektora na CPU ploči. U ovom slučaju se na ulazu svakog pina nalazi *pull-up* otpornik na koji je jedino vezan kondenzator ka masi. Razlog za ovaj način povezivanja je da se eventualno nastale smetnje prilikom prenosa signala kablovskim putem otklone i da se stabilan signal baferuje radi sigurnosti čime se povećava efikasnost prenosa signala. Obzirom da ploča ima 1 ulaz i 4 izlaza, potrebno je da se pronađe način kako će se na tačno određenoj vertikali poslati signal. To se rešava uz pomoć multipleksera koji će signale glavne podatke, *shift* registar i dozvolu rada proslediti onda kada stigne potreban podatak sa CPU ploče. Nakon izlaska iz multipleksera, podaci se ponovo šalju na bafer i putem desetopinskih konektora dalje na štampane ploče. Obzirom da LED drajveri imaju mogućnost slanja povratnih informacija *master* uređaju o greškama LE diode (kratak spoj ili prekid LE diode), ovi podaci će proći istu proceduru samo u suprotnom smeru (od matične štampane ploče do CPU ploče).

E. Štampana ploča za električno ispitivanje ploče sa LE diodama

Glavna uloga ove ploče jeste provera ispravnosti ploče sa LE diodama u električnom smislu. Ploča ispituje ispravnost grejača ploče (kod određenih matičnih ploča), proverava tačnost signala nakon njihovog prolaska kroz ploču i proverava da li negde na ploči postoji spoj napona napajanja od 5 i 3,3 V.

Štampana ploča je osmišljena tako da pokriva što više načina testiranja. Obzirom da ploča služi za ispitivanje i na neki način za generisanje podataka o stanju ploče sa LE diodama, potrebno je te podatke poslati CPU ploči na što pouzdaniji i efikasniji način. Za tu svrhu se koristi integrisano kolo *MCP23017* [6] koje podržava protokol za I2C komunikaciju kako bi prenelo podatke od i do CPU ploče.

Ovo kolo poseduje 2 porta sa po 8 ulazno-izlaznih pinova i može da prenosi 16-bitne podatke. Ovi portovi mogu da rade zajedno ili posebno, zavisno od kontrole nad njima, a svaki port može da bude i ulazni i izlazni u zavisnosti od želje korisnika. Takođe, poseduje 3 selektorska pina za podešavanje I2C adrese kola tako da se na jednoj ploči može naći do 8 ovakvih kola. Upravo zbog ovih karakteristika je ovo kolo iskorišćeno. Port A je izabran kao izlazni i on kontroliše kada i koje napajanje će se poslati na klemu za napajanje ploče sa LE diodama dok je port B izabran kao ulazni i na ovom portu su dovedeni signali za SPI interfejs za LED drajvere kao i signal greške.

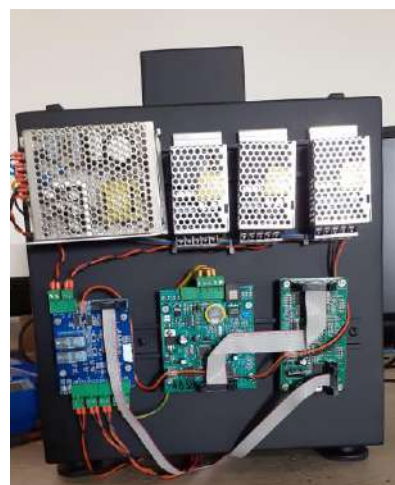
Kolo će ove signale obraditi i putem I2C komunikacije poslati CPU ploči. Takođe, kada bude bilo potrebno uključiti napajanje za ploču sa LE diodama, CPU ploča će generisati podatak koji će *MCP23017* obraditi i podići pin za kontrolu releja na visok naponski nivo čime će se ulazni napon proslediti na izlaz.

Ova štampana ploča treba da ispita sledeće:

- ispitivanje ispravnosti grejača na ploči merenjem struje,
- ispitivanje napona napajanja kao i da li su u spoju,
- ispitivanje signala za podatke i kontrolne signale.

Na osnovu dobijenih podataka, štampana ploča komunicira sa CPU pločom putem protokola I2C komunikacije na osnovu čega CPU ploča vraća informaciju korisniku o ispravnosti štampane ploče sa LE diodama.

F. Kablovi i konektori za povezivanje



Slika 3. Izgled povezanog sistema.

Izgled zadnje stranice kućišta na koju su montirane štampane ploče i napajanja se može videti na slici 3. Da bi se povezale štampane ploče potrebni su određeni tipovi kablova pa se tako koriste šesnaestožilni provodnici za povezivanje CPU i ekstenzione ploče i desetožični provodnici za povezivanje ekstenzione ploče sa štampanom pločom sa LE diodama a kasnije i za povezivanje štampane ploče za električno ispitivanje ploče sa LE diodama. Pored ovih kablova koriste se i i kablovi za napajanje koji su većeg preseka u odnosu na prethodno napomenute kablove.

Što se tiče konektora, koriste se flet konektori prilikom međusobnog povezivanja ploča i koriste se terminal kleme za ostala povezivanje (napajanje i kleva za protokol I2C komunikacije).

G. Napajanja

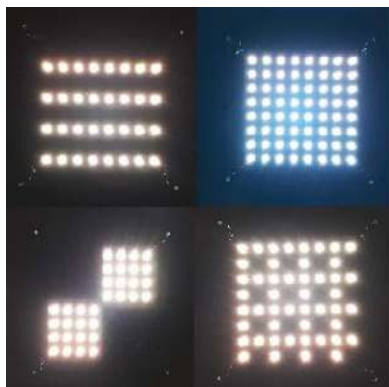
Da bi ceo ovaj sistem funkcionisao, potrebno je odabrati adekvatna napajanja za njegov rad. Obzirom da su potrebni naponi različiti, potrebno je onoliko napajanja

koliko postoje različitih vrednosti napona napajanja. Konkretno, potrebno je 12 V za napajanje sve tri ploče, dok su napajanja od 3,3 i 5 V potrebna za ploču za električno ispitivanje ploče sa LE diodama i naravno za samu ploču koja se testira. Na osnovu matričnih ploča sa LE diodama koje postoje, potrebno je bilo izračunati maksimalnu potrošnju svih matričnih ploča koje se koriste kada rade u režimu gde je njihova potrošnja najveća (kada je potrebna bela boja svetlosti). Na osnovu podataka iz tehničke dokumentacija za RGB diode izabrana su napajanja i to: 12 V – 25 W, 5 V – 25W (2 komada) i 3,3 V – 75 W.

V. POTROŠNJA SISTEMA I TESTNI ŠABLONI

Ovako kreiran sistem ne traži puno snage, osim za napajanje matrične ploče koja se testira. Kada govorimo o potrošnji CPU ploče, ekstenzione ploče i štampane ploče za električno ispitivanje štampanih ploča ona je reda veličine nekoliko vati što zavisi od režima rada. Za slučaj kada se testira matrična štampana ploča, maksimalna potrošnja celog sistema ne prelazi snagu od 40W.

Na slici 4. može se videti par šablona koje kamera treba da prepozna na osnovu video snimka. Testiranje nije usko vezano za prikazane šablone jer postoji veliki broj šablona koji se može iskoristiti. Najvažnije je da prilikom testiranja matrične štampane ono što kamera opaža bude upravo određeni šablon koji se očekuje od CPU ploče koja u memoriji mikrokontrolera čuva određeni broj šablona koji je moguće prikazati. Obzirom da su matrične štampane ploče uglavnom u formatu 8x8 (8 RGB LE dioda horizontalno i 8 vertikalno) moguće je kreirati različite šablone.



Slika 4. Testni šabloni.

VI. ZAKLJUČAK

Ovaj sistem je prvenstveno osmišljen kao pomoć pri testiranju ploče sa RGB LE diodama na dosta složeniji

način od onog koji se danas koristi. Međutim, iako je način testiranja složeniji, posao korisnika se svodi na ubacivanje ploče, podešavanje parametara i uključivanje testera u proces testiranja. Proces izrade samog kućišta je bio dosta složen jer ima dosta specifičnih delova koje je bilo potrebno konstruisati i povezati u jednu celinu. Takođe, što se tiče hardverskog dela, prilikom dizajniranja same ploča bilo je potrebno pokriti što više načina ispitivanja ploča sa RGB LE diodama a uz to i pravilan raspored komponenata na ploči kao i razlivanje mase bez uticaja na funkcionalnost. Ono što nedostaje jeste softver koji će da kontroliše ceo sistem i koji će da omogućava korisniku lako rukovanje prilikom testiranja.

Sistem je sklopljen, izvršeno je kabliranje i raspored delova sistema prema unapred utvrđenom redosledu, ispitivan je prostor za pozicioniranje kamere, unutar testera je postavljeno više različitih ploča sa RGB LE diodama, ispitivan je ulazak svetlosti u unutrašnjost sistema i eventualni propusti su ispravljani tako da je sistem spreman za ispitivanje ploča što se tiče hardverskog dela.

ZAHVALNICA

Autor ovog rada se zahvaljuje Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, posebno prof. dr Danijelu Dankoviću i asistentu Nikoli Mitroviću, na pruženoj pomoći i svakodnevnoj dostupnosti, kao i u davanju smernica prilikom realizacije ovog projekta.

LITERATURA

- [1] A. Prijic „Poluprovodničke komponente“, dostupno na: <http://mikro.elfak.ni.ac.rs/wp-content/uploads/Poluprovodni%C4%8Dke-komponente-diode-2021-1.pdf>
- [2] NSSM124DT, „Full color LED“, Nichia Corporation, datasheet, dostupno na: [https://www.nichia.co.jp/specification/products/led_spec/NSSM124DT-E\(4436A\).pdf](https://www.nichia.co.jp/specification/products/led_spec/NSSM124DT-E(4436A).pdf)
- [3] TLC5955, „PWM LED driver“, Texas Instruments, datasheet, dostupno na: https://www.ti.com/lit/ds/symlink/tlc5955.pdf?ts=1634002701108&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTLC5955
- [4] AT89C51ED2, „8-bit flash microcontroller“, Atmel, datasheet, dostupno na: <https://www.farnell.com/datasheets/299889.pdf>
- [5] M. Marjanović, „Modeliranje i simulacija mikroelektronskih komponenata i kola“, dostupno na: <http://mikro.elfak.ni.ac.rs/wp-content/uploads/Arduino-zbirka-resenih-projekata.pdf>
- [6] MCP23017, „16-Bit I/O Expander with Serial Interface“, Microchip, datasheet, dostupno na: <https://ww1.microchip.com/downloads/en/devicedoc/2000195>

LED Driver Basics and its Most Important Characteristics

Mihajlo Đorđević

Abstract – In this manuscript for the IEEEESTEC Student's papers Conference, the design rules and application of LED drivers are considered. Besides, it also covers additional rules on which designers need to pay attention when designing a LED driver in their custom applications. Considering that LEDs are current-driven devices, current control in LED drivers is also presented and how this affects efficiency. In addition to the current, it is explained why the size of the application itself has its share of inefficiency and additional functions such as dimming and overvoltage protection. At the very end, a practical example of a LED driver in the form of an integrated circuit is shown, and a typical application example of an LED driver application.

I. INTRODUCTION

Due to the increase in energy regulations, most people are now familiar with features such as long life and energy savings associated with LEDs or light-emitting diodes in general. However, many are unaware that these innovative light sources require specialized devices necessary for their proper operation called LED drivers. LED drivers can be a confusing part of LED technology, mainly because there are many different types and variations of the driver. Also known as LED power supplies, LED drivers share a remarkable resemblance to ballasts for fluorescent bulbs or transformers for low voltage bulbs; more precisely, they supply LEDs with the electricity they need for proper operation and best performance. For this reason, this paper can help you understand the very concept and type of LED drivers, how they differ, and what to look for when choosing a driver for your LED lighting application.

Let's start primarily with why we need a driver for LEDs. LEDs are designed to operate on low voltage direct current (DC) with voltages of 12V or even 24V. However, most places supply higher voltage (120-277V), alternating current. The LED driver "corrects" higher voltage, more precisely alternating current to low voltage, i.e., to direct current, and thus also protects the LEDs from voltage fluctuations which can cause a change in the current supplied to the LEDs.

The output of the diode, more precisely the LED light, is proportional to the current supply of the diode - a specific range of current measured in amperes. For this reason, too much or too little current can cause the light output to change faster or degrade due to higher

temperatures inside the LED itself. In short, LED drivers not only convert higher voltage (alternating current) to low voltage (direct current), they also maintain the voltage and current flowing through the LED circuit at a nominal level.

Using the driver itself is crucial in preventing damage to LEDs because the forward voltage, which represents the amount of volts needed to emit electricity and illuminate the emitting (light) diode, changes with temperature in high power diodes. As the temperature rises, the forward voltage of the LED decreases, causing the LEDs to draw more current, which is why the LED will continue to heat up and attract more and more current until the diode itself burns out. For this reason, the LED driver is a stand-alone power supply with outputs aligned with the LEDs' electrical characteristics.

For the successful selection of LED drivers, it is necessary to pay attention to some of the essential characteristics such as the correct choice of LED driver type, current control, efficiency and size, dimming, which will be explained in more detail later. [1]

II. LED DRIVER TYPES

We recognize two types of LED drivers as shown in Fig.1., such as those that use DC low voltage in the range of usually 5V up to 36V DC, and those that use high AC voltage from 90V up to 277V AC called Off-Line or AC LED drivers.

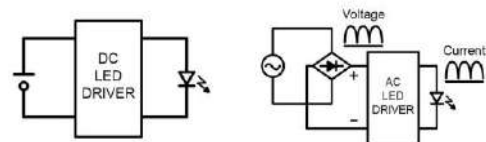


Fig.1. DC and AC type of LED driver.

A low-voltage DC input for LED drivers is recommended in most applications because they are highly efficient and reliable. For smaller applications, more dimming and output options are available compared to high-voltage AC drivers. However, for larger projects such as general lighting projects for residential or commercial lighting, in that case, AC drivers would be a better solution.

In this paper, the topic bases on low-voltage DC LED drivers in the form of integrated circuits, but we have also mentioned other types to understand the breadth of this very topic.

M. Đorđević is with the Department of Microelectronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: mihajlo628@hotmail.com

III. LED CURRENT CONTROL

LEDs are current-driven devices whose brightness is proportional to their forward current, controlled in two ways. One way is to use an LED current-voltage I-U curve to determine the voltage applied to the LED to generate the desired forward current. This method is usually achieved using a voltage source and a ballast resistor (a ballast resistor defined as a resistor in a circuit used to reduce current), as shown in Fig.2.

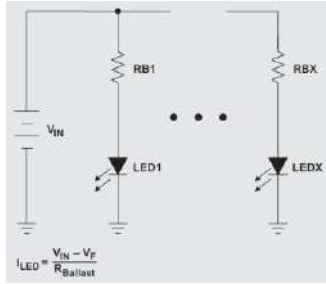


Fig.2. LED current control using a voltage source and a ballast resistor [2].

However, this method is not perfect and has several drawbacks. Each change in the LED forward voltage changes the current flowing through the LED itself. With a nominal forward voltage of 3.6V, the current flowing through the LED in Fig.2 has a value of 20mA. If this voltage increases and changes to 4V within the specified voltage tolerance, the current value will drop to 14mA. Also, depending on the available input voltage, voltage drop and power dissipation across the ballast resistor consumes more power and reduces battery life. Another method for regulating the current of the LED, which is primarily recommended, is to start the LED with a constant current source, shown in Fig.3. eliminating the current change due to changes in forward voltage, which automatically represents the constant brightness of the LED. In this case, the reference supply voltage and the value of the current-sense resistor are the ones that determine the LED current. Multiple LEDs should be connected in series configuration so that the same current flows in each LED, while parallel connection of LEDs requires the ballast resistor as mentioned earlier in each LED series, which leads to reduced efficiency and uneven current matching.

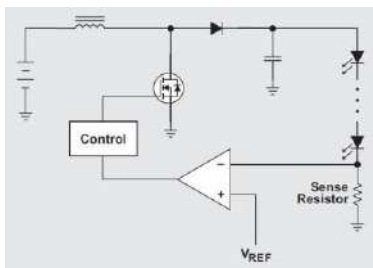


Fig.3. Constant current source LED current control [2].

IV. EFFICIENCY AND SIZE

One of the more critical problems, especially in portable applications, is the battery's durability and the power supply itself. Therefore, for an LED driver to be valid and helpful, it must also be efficient. Measuring the efficiency of a LED driver differs from that of a regular power supply. The efficiency of a standard power supply defines as the output power divided by the input power, which differs in the case of LED drivers. With LED drivers, power output is not a parameter of interest. What is essential is the amount of input power necessary to generate the desired brightness of the LED, which is easily determined by dividing the power of the LED by the input power.

Defining efficiency in this way means that the power dissipated on the current-sense resistor with the current sensitivity contributes to the power loss in the power supply. Good advice, in this case, is to choose a power supply with a reference voltage of 0.25V versus a power supply with a reference voltage of 1V, which leads to improved efficiency. A power supply with a lower voltage more sensitive to current is more efficient regardless of the input voltage or LED current. In addition to all other conditions, a lower reference voltage significantly improves efficiency and extends battery life.

Another important feature of LED drivers is their small size. One of the factors that contribute to the size of the component itself is the switching frequency. Higher switching frequencies allow the use of smaller passive components. Modern LED drivers intended for portable applications should be seamlessly "switched" at frequencies up to 1MHz. All frequencies higher than 1MHz are not recommended because it doesn't reduce the dimensions of the circuit itself but reduces efficiency and shortens battery life due to higher switching losses.

V. LED DIMMING

Today's LED applications, in most cases, require the option of LED dimming. In applications such as LCD backlighting, dimming provides adjustment for both contrast and brightness. There are two types of dimming such as analog and PWM dimming. With analog dimming, 50% of the illumination is achieved by applying 50% of the maximum current to the LEDs. Disadvantages of the analog dimming method include changing the color of the LED, as well as the need for an analog control signal that is generally not readily available.

On the other hand, PWM dimming is achieved by applying the total value of the current to the LEDs at a reduced duty cycle. To achieve a brightness of 50%, full current is applied at a duty cycle of 50%. The frequency of the PWM signal must have a value above 100Hz to ensure that the pulsation of the PWM is not visible to the human eye. To ensure maximum flexibility and ease of integration,

the LED driver should be able to accept PWM signal frequencies up to 50kHz.

VI. OVERVOLTAGE PROTECTION

A power supply with a constant current source generates a constant output current regardless of the load. If there is an increase in load resistance, the supply voltage must also increase to obtain the constant current necessary for the supply. If the power supply encounters excessive load resistance, or if the load switches off, the output voltage may increase above the voltage of the integrated circuit itself or other discrete components. For this reason, the operation of the power supply in the constant current mode requires protection against overvoltage.

For LED drivers operating in constant power supply mode, there are several ways to protect against overvoltage, where one of them is to place the Zener diode in parallel with the LED itself. This mode limits the output voltage to the breakdown voltage of the Zener diode by adding the value of the reference supply voltage. During the overvoltage state, the output voltage increases to the breakdown point of the Zener diode, where it begins to conduct. The output current flows through the Zener diode, then through the current-sense resistor to the ground. The power supply continues to generate a constant output current while the Zener diode limits the maximum output voltage.

Another interesting way to protect against overvoltage is to monitor the output voltage and turn it off when it reaches its limit (overvoltage trip point). Turning off the power source when an overvoltage occurs reduces energy loss, thereby extending battery life in the event of a failure.

VII. PRACTICAL APPLICATION

When we now summarize all the parameters mentioned above and characteristics that need attention when choosing the right LED driver, we can also show a practical example of such a driver in the form of an integrated circuit. It is TPS61042 [2], an LED driver integrated circuit from Texas Instruments, whose block diagram with highly integrated control logic is shown in Fig.4. In addition to the block diagram, the actual look of the integrated circuit itself is shown in its QFN package measuring 3mm x 3mm. Transistor Q1 is an integrated low-resistance MOSFET that significantly contributes to the high efficiency of the component itself, while transistor Q2 is the part of the circuit that protects the circuit from load-disconnect. The obtained reference voltage of 0.25V contributes to the reduction of current-sense resistor losses. The PWM dimming mode is easily implemented on this integrated circuit by applying an external PWM signal to its control CTRL pin at high frequencies up to 50kHz. Since it is an integrated circuit like TPS61042, it can be said that this circuit is perfectly synchronized with the PWM dimming frequency.

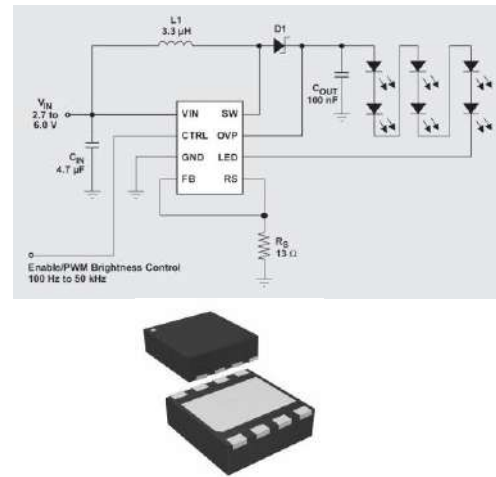


Fig.4. Block diagram and actual look of the TPS61042.

Fig.5. shows a typical application scheme for an LED driver that can drive four LEDs with a current of 20mA operating in the supply voltage range from 1.8V to 6V. The realized LED driver consists of an integrated circuit TSP61042, two small ceramic capacitors for filtering inputs and outputs, a coil, a current-sense resistor, and a diode.

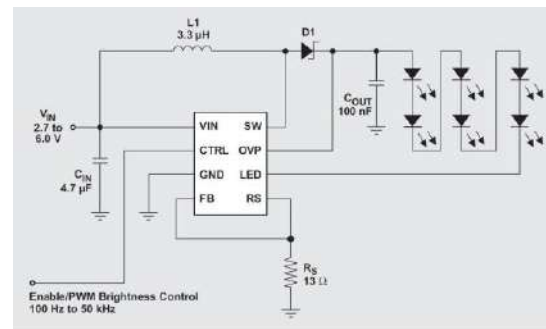


Fig.5. Typical application of the TPS61042.

The overall efficiency of the application mainly depends on the selection of the inductor. A lower inductor value increases the switching frequency and switching losses yielding a lower efficiency. A lower inductor dc resistance has lower copper losses, giving a higher efficiency. Therefore, the efficiency can typically vary $\pm 5\%$ depending on the selected inductor. The converter regulates the LED current by regulating the voltage across the current sense resistor (R_S). The voltage across the sense resistor is regulated to the internal reference voltage of 252mV. The LED current can be calculated with the help of the equation:

$I_{LED} = 0.252V / R_S$; where a LED current of 19mA is obtained for the value of the sense resistor of 13Ω.

The current programming method is used when the brightness of the LEDs is fixed or controlled by a PWM signal applied to the CTRL (control) pin. When using a

PWM signal on the CTRL pin, the LED brightness is only dependent on the PWM duty cycle, independent of the PWM frequency, or amplitude, which simplifies the system. Such a small circuit shows the high level of integration that is achieved with today's LED drivers. In addition, it has capabilities such as load disconnection, overvoltage protection, as well as PWM dimming, which performs the adjustment of the brightness of LEDs with frequencies in the range from 100Hz up to 50kHz. In addition to the essential characteristics of this component and its theoretical presentation of the application, its way of connecting in a circuit and all its components is of great importance. For this reason, we will also consider the layout guidelines of the TPS61042, shown in Fig.6. In all switching power supplies the layout is an important step in the design, especially at high peak currents and switching frequencies. If the layout is not carefully done, the regulator might show noise problems and duty cycle jitter. The input capacitor CIN should be placed as close as possible to the input pin for good input voltage filtering. The inductor L1 and diode must be placed as close as possible to the switch pin to minimize noise coupling into other circuits. Since the feedback pin and network is a high impedance circuit, the feedback network should be routed away from the inductor.

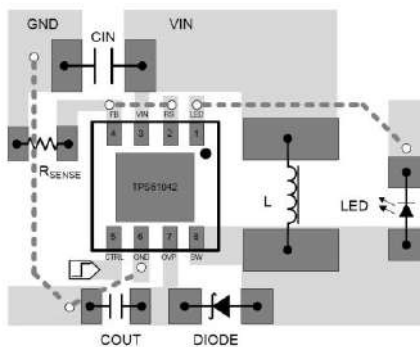


Fig.6. Layout guidelines of the TPS61042.

VIII. CONCLUSION

The LED driver is a stand-alone power supply that regulates the power required for one or a series of LEDs. Light-emitting diodes are low-energy diodes, i.e., lighting devices with long service life and low energy consumption, which is why they demand specialized power sources such as LED drivers. The LED driver is something like cruise control in a car (automatic vehicle speed control system); the required power level changes with the temperature of the LED. Without the proper LED driver, the LEDs would become too hot and unstable, leading to malfunction and poor performance. Based on this work, we can now solve these problems and ensure the proper functioning of LEDs by successfully selecting the appropriate LED driver.

ACKNOWLEDGEMENT

I want to take this opportunity to thank professor Danijel Danković for the invitation and the opportunity to participate and be a part of the IEEEESTEC Student's Projects Conference this year as well.

REFERENCES

- [1] "Understanding LED Driver Technology" available at: https://www.vossloh-schwabe.com/fileadmin/user_upload/Service_und_Downloads/Literatur/Broschueren_und_Kataloge/Leitfaden_Treiber_EN.pdf
- [2] "LED Driver Considerations" available at: <http://www.lunaleds.com/technologies.html>
- [3] "TPS61042" available at: <https://www.ti.com/product/TPS61042?keyMatch=TPS61042&tisearch=search-everything&usecase=GPN>
- [4] "LED Driver consideration" available at: <https://www.ti.com/lit/an/slyt084/slyt084.pdf>

Merenje provodnosti metala primenom vrtložnih struja

Milica Stojanović

Sadržaj – Vrtložne struje su struje indukovane u provodniku kada se magnetno polje oko provodnika menja. Metoda beskontaktnog merenja provodnosti se zasniva upravo na pojavi vrtložnih struja. Merenjem parametara impedanse kalema, za različite vrednosti udaljenosti kalema od provodnika i za različite vrednosti električne provodnosti provodnika, moguće je odrediti vrednost električne provodnosti za provodnik čija se provodnost traži.

I. UVOD

Električna provodnost predstavlja meru sposobnosti materijala da provodi električnu struju ili da prenosi električni napon. Jedinica u SI sistemu za električnu provodnost je simens po metru (S/m) i odnosi se na temperaturu od 25°C, ukoliko nije drugačije naglašeno.

Statičku električnu provodnost je moguće izračunati kao recipročnu vrednost otpornosti merene standardnom U/I metodom. Međutim, određivanje električne provodnosti metala može biti od interesa i pri metalurškim ispitivanjima i detekciji defekata. U tom slučaju nije moguće primeniti tradicionalne metode za merenje otpornosti i provodnosti.

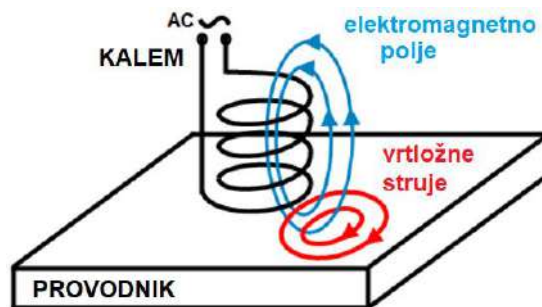
Zbog toga je realizovana metoda beskontaktnog merenja električne provodnosti koja se bazira na pojavi vrtložnih struja u provodniku. Vrtložne struje su struje indukovane u provodniku kada se magnetno polje oko provodnika menja. Kada se magnetno polje kalema nalazi u blizini provodnika, u provodniku će se indukovati vrtložne struje. Pod dejstvom vrtložnih struja generiše se novo magnetno polje koje će izazvati slabljenje originalnog polja. Promena originalnog magnetnog polja će uticati na vrednost impedanse kalema, slika 1. Kako je eksperimentalno utvrđeno da su logaritmi faze impedanse kalema i logaritmi električne provodnosti linearno zavisni, merenjem parametara impedanse kalema, može se odrediti nepoznata vrednost električne provodnosti [1][2].

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Kako bi bilo moguće primeniti metodu za beskontaktno merenje električne provodnosti, potrebno je unapred odrediti vrednost pobudne frekvencije tako da debljina uzorka čija se električna provodnost meri bude bar tri puta veća od dubine prodiranja vrtložnih struja. Zatim su

M. Stojanović, je master student Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: s.micika@elfak.rs

potrebna dva uzorka poznatih električnih provodnosti kako bi se generisala kalibraciona kriva. Pored toga, električna



Slika 1. Princip nastanka vrtložnih struja [2].

provodnost standardnih uzoraka treba biti dovoljno tačna, jer se u suprotnom može javiti još veća greška. Električne provodnosti ovih uzoraka su obeleženi sa σ_1 i σ_2 , respektivno. Za oba provodnika se određuju po dve vrednosti impedanse, za različite vrednosti udaljenosti kalema od provodnika, kako bi se eliminisao uticaj liftoff efekta. Fazni signali standardnih uzoraka se dobijaju kao:

$$\begin{cases} Phs_1 = \frac{X_{11} - X_{12}}{R_{11} - R_{12}} \\ Phs_2 = \frac{X_{21} - X_{22}}{R_{21} - R_{22}} \end{cases} \quad (1)$$

gde su R_{11} i R_{12} otpornosti, a X_{11} i X_{12} rezistanse impedanse prvog standardnog uzorka za dve vrednosti udaljenosti kalema od provodnika i R_{21} i R_{22} otpornosti, a X_{21} i X_{22} rezistanse impedanse drugog standardnog uzorka za dve vrednosti udaljenosti kalema od provodnika. [1]

Kalibraciona kriva je oblika

$$\ln \left| \frac{\Delta X}{\Delta R} \right| = k \ln(\sigma) + c, \quad (2)$$

gde su

$$\begin{aligned} k &= \frac{\ln|Phs_2| - \ln|Phs_1|}{\ln(\sigma_2) - \ln(\sigma_1)} \\ c &= \ln|Phs_1| - k \ln(\sigma_1) \end{aligned} \quad (3)$$

Ovaj metod je predstavljen algoritmom na slici 2.

Za praktičnu realizaciju ovog algoritma korišćeni su instrumenti Analog Discovery 2 i Hantek 1833C za

merenje parametara impedanse kalema, prikazani na slikama 3 i 4; provodnik od mesing poznate provodnosti $\sigma_1=16,24\text{MS/m}$, provodnik od bakra poznate provodnosti $\sigma_2=51,5\text{MS/m}$ i provodnik od aluminijuma čija se provodnost određuje. Udaljenosti kalema od provodnika, na kojima su mereni parametri impedanse, su izabrane na tri načina, i to:

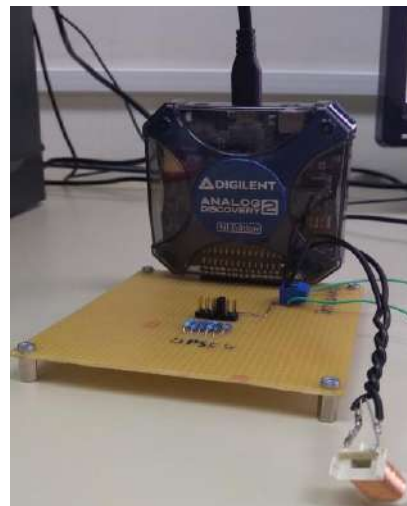
- $d_1=0,5\text{mm}$ i $d_2=1\text{mm}$,
- $d_1=0\text{mm}+\Delta d=0,5\text{mm}$ i $d_2=0,5\text{mm}+\Delta d=1\text{mm}$, pri čemu je $\Delta d=0,5\text{mm}$ debljina kalemskog tela i
- $d_1=\Delta d=0,5\text{mm}$ i $d_2\rightarrow\infty$

Sva merenja su izvršena prvo na frekvenciji 1kHz, a zatim i na frekvenciji 10kHz.



Slika 2. Algoritam beskontaktnog merenja provodnosti [1].

Nakon što je postavljena vrednost pobudne frekvencije na 1kHz, odnosno 10kHz u drugom slučaju, merene su vrednosti impedanse kalema Z_{11} , Z_{12} , Z_{21} i Z_{22} , za dva provodnika poznatih električnih provodnosti, kao što je navedeno u algoritmu na slici 2. Impedanse su merene za sva tri navedena slučaja izbora udaljenosti kalema od provodnika. Za svaki od slučaja se, pomoću parametara impedanse R i X , računaju Phs_1 i Phs_2 , primenom formule 1. Pomoću ovih vrednosti, kao i vrednosti poznatih električnih provodnosti, računaju se parametri k i c primenom formule 3, i formira se kalibraciona kriva primenom formule 2. Kalibraciona kriva se može aproksimirati linearnom funkcijom.



Slika 3. Instrument za merenje parametara impedanse kalema Analog Discovery 2.



Slika 4. Instrument za merenje parametara impedanse kalema Hantek 1833C.

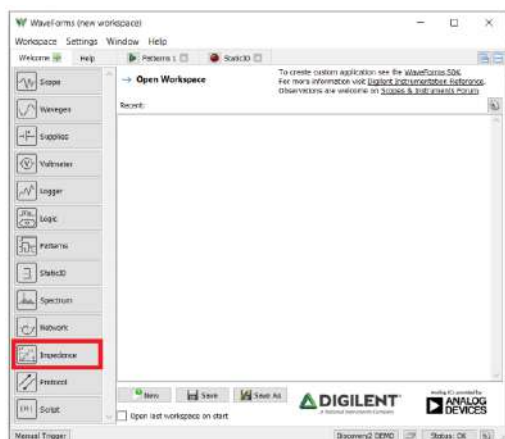
Zatim se za provodnik nepoznate provodnosti, mere impedanse kalema, za iste vrednosti udaljenosti kalema od provodnika, kao malo pre i na osnovu dobijenih vrednosti se izračunava faza Phs . Zamenom dobijene vrednosti faze u koalibracionu krivu, dobija se vrednost tražene električne provodnosti provodnika.

III. INSTRUMENTI ZA MERENJE PARAMETARA IMPEDANSE

A. Analog Discovery 2

Analog Discovery 2, slika 3, je multifunkcionalni akvizicioni modul, pomoću kojeg je moguće vršiti merenja, generisanje signala, snimanje signala i analize elektroonskih kola različitih vrsta. Iako je malih dimenzija, može zameniti veliki broj instrumenata u laboratoriji, a njegove performance se mogu proširiti korišćenjem dodatnih adaptera, kao što su BNC ili impedansni adapter [3].

Za ovaj instrument postoji programski paket WaveForms koji pruža softversku podršku. Klikom na odgovarajuću karticu u glavnom prozoru programskog paketa, bira se način rada Analog Discovery-a 2, pri čemu on može da se ponaša kao osciloskop, signal generator, izvor napajanja, voltmetar, analizator logičkih kola, analizator spectra, analizator mreže ili uređaj za merenje impedanse. Na slici 5 je prikazan izgled glavnog prozora WaveForms softvera [3].



Slika 5. Izgled glavnog prozora WaveForms-a.

B. Hantek 1833C

Hantek 1833C, slika 4, je ručni, prenosivi LCR metar, instrument za merenje parametara kao što su induktivnost, kapacitivnost, otpornost [4].

IV. REZULTATI MERENJA

U nastavku će, u tabeli 1, biti prikazane vrednosti provodnosti aluminijumskog provodnika dobijene nakon izvršenih merenja, primene algoritma i formula 1, 2 i 3. Iz date tabele se može videti da se dobijene vrednosti veoma razlikuju. Na rezultate merenja mogu uticati razni parametri, među njima udaljenost kalema od provodnika, vrednost pobudne frekvencije, vrednost permeabilnosti provodnika.

TABELA I
REZULTATI IZRAČUNAVANJA NEPOZNATE PROVODNOSTI

	Analog Discovery 2	
	1kHz	10kHz
$d_1=0,5\text{mm};$ $d_2=1\text{mm}$	22,3215 MS/m	50,728M S/m
$d_1=\Delta d=0,5\text{mm};$ $d_2=0,5\text{mm}+\Delta d=1\text{mm}$	23,897M S/m	13,967M S/m
$d_1=\Delta d=0,5\text{mm};$ $d_2\rightarrow\infty$	25,3303 MS/m	27,7364 MS/m
	Hantek 1833C	
	1kHz	10kHz
$d_1=0,5\text{mm};$ $d_2=1\text{mm}$	33,569M S/m	/
$d_1=\Delta d=0,5\text{mm};$ $d_2=0,5\text{mm}+\Delta d=1\text{mm}$	55,094M S/m	30,355M S/m
$d_1=\Delta d=0,5\text{mm};$ $d_2\rightarrow\infty$	35,999M S/m	26,7459 MS/m

Promena udaljenosti kalema od provodnika utiče na impedansu kalema, iako je faza kalema neosetljiva na tu promenu. Samim tim će uticati i na krajnje rezultate. Kako bi se smanjio uticaj promene udaljenosti kalema od provodnika, za referentnu vrednost impedanse se, umesto vrednosti impedanse kalema merene kada se kalem nalazi na beskonačnoj udaljenosti od provodnika, uzima vrednost impedanse kalema merene na rastojanju od 1mm kalema od provodnika.

Rezultati merenja će zavisiti i od vrednosti pobudne frekvencije. Od vrednosti pobudne frekvencije zavisi dubina prodiranja vrtložnih struja u provodniku, pa je veoma važno da se za vrednost pobudne frekvencije izabere vrednost kojom će se omogućiti da debljina uzorka bude bar tri puta veća od dubine prodiranja vrtložnih struja. Za tanje uzorke koriste više frekvencije kako bi se smanjila dubina prodiranja vrtložnih struja.

Takođe, na merene rezultate može uticati vrednost permeabilnosti provodnika. S obzirom da je sa porastom vrednosti permeabilnosti, uticaj na dobijene vrednosti veći, za primenu ovog algoritma poželjno je birati neferomagnetne provodnike kao što su provodnici od bakra, aluminijuma, mesinga.

V. ZAKLJUČAK

Usled nedostatka opreme kojom bi se izmerila stvarna vrednost provodnosti aluminijumskog provodnika, ne može se reći koji izbor parametara za izračunavanje je tačan. Ono što se može primetiti iz rezultata jeste da je ovaj metod merenja provodnosti veoma osetljiv, tj. vrednost tražene električne provodnosti se veoma menja, za male promene parametara R i X. Izborom vrednosti pobudne frekvencije, rastojanja između kalema i provodnika na kojem će se meriti parametri impedanse i tipa provodnika, može se uticati na krajnje rezultate i smanjiti greška. Takođe, krajnji

rezultat će zavisi i od tačnosti i stabilnosti instrumenata koji se koriste, ali i od uslova u kojima se merenja vrše.

ZAHVALNICA

Zahvalnost za realizaciju ovog rada, na pruženoj pomoći i saradnji, uputila bih prof. dr Draganu Živanoviću, kao i Katedri za metrologiju i mernu tehniku na Elektronskom fakultetu, Univerziteta u Nišu.

LITERATURA

- [1] Cuiping Wang, Mengbao Fan, Binghua Cao, Bo Yi, Wei Li, „Novel Noncontact Eddy Current Measurement of Electrical Conductivity“, *IEEE Sensors Journal*, 2018, DOI: 10.1109/JSEN.2018.2870676
- [2] Eddy Current Testing using the Bode 100 – Lukas Heinzle
- [3] Digilent, Inc. (Oktobar 2015.). *Analog Discovery 2TM Reference Manual* [Online]. Dostupno na: <https://reference.digilentinc.com/reference/>, [pristupljeno oktobra 2021].
- [4] Web site: <http://www.hantek.com/products/detail/16180> [pristupljeno oktobra 2021.]

Realization of Training for Remote Control of Drones

Uroš Ercegovac, Slavko Popović, Teodora Radić and Miljana Vuković

Abstract - In this paper development and benefits of educational project “Our Friend Mr. Robot” is explained -training for drone control and programming. The goal of the project is that through three trainings that will be held in Serbia, Portugal and Slovenia, we will come to a solution on how to get a picture of the ripeness of strawberries with the help of a drone, which should be programmed, to send a mechanical hand to the exact location and harvest fruit.

I. INTRODUCTION

We live in a time where technology has advanced so much that life without electronic devices is almost unimaginable. At every step and in every area, we can see devices that improve and facilitate human work and life. The use of drones in agriculture can help a person to do the job easier and much faster, as well as to significantly reduce the cost of the final product. There are a couple of basic rules for how a drone moves in relation to the speed of the engines. When a drone flies in a constant position above a certain point, then the drone is in hover mode. In this flight mode, the driving force of the buoyancy is equal to the weight of the drone. The driving force of the buoyancy arises as a consequence of the operation of all four engines. If the buoyancy force is greater than the weight of the drone, it will move vertically upwards. If the buoyancy force is less than the weight of the drone, it will move vertically downwards. The amount of buoyancy depends on the speed of rotation of the motor rotor. If the speed is lower and the buoyancy force is lower and vice versa. To ensure lateral movement, in the plane of the center of gravity of the drone, without changing the flight altitude, it is necessary for the rotors of the three engines to rotate at the same speed, and the rotor of the fourth engine to rotate at a different speed. Movement in one of the four directions can also be provided by increasing the rotor speed of two adjacent motors. During this movement, the drone is tilted to the side, so it is necessary to pay attention to the maximum size of the tilt angle. For rotational movement in the plane of the center of gravity of the drone, it is necessary to reduce the speed of rotation of the rotor opposite the engines.

The project was realized using and programming the Tello EDU Drone [1]. This drone was chosen because it allows the user to program it through three programming languages (applications). List of components of the Tello EDU drone (Fig 1.):

U. Ercegovac, S. Popović, T. Radić and M. Vuković are students of the Vocational School “Milenko Brzak Uča” Ruma, Vuka Karadžića 70 , 22400 Ruma, Serbia, E-mail: igor.tomic@yahoo.com

- Propellers
- Engines
- Drone Status Indicator
- Camera
- Power Button
- Antennas
- Vision Positioning System
- Battery
- Micro USB port
- Propeller protectors

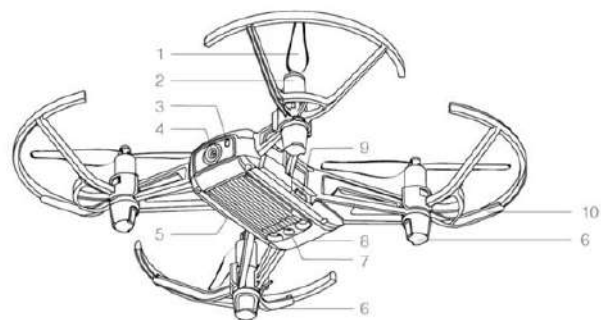


Fig. 1. Components of the Tello EDU drone.

The basic construction requirements are as follows:

1. The plane of the engine rotor must be above the plane in which the center of gravity of the quadcopter is located. If the plane in which the rotors are located is also the plane in which the center of gravity is located, there is a danger of the quadcopter overturning during the maneuver around the horizontal plane, due to the moments of gravity around the center of gravity. Rollover means loss of buoyancy and fall. This implies that the center of gravity of the drone should be as low as possible and that any additional cargo or equipment is placed below the base of the drone. This is a condition for greater mobility of the drone, because when the drone is tilted, the reference plane of the propeller additionally goes down.
2. The vertical axis of all engines must lie in the same circle with the center located on the central axis of the center of gravity of the structure.
3. Symmetry of the structure for easier regulation and better stability
4. Secured space for batteries and other equipment
5. As little mass as possible but to meet the strength requirements.
6. Space above and below the rotor with as few construction parts as possible and, if possible, the existence

of elements that would take over most of the impact energy in the event of a fall, while being easily replaceable [4].

II. DRONE CONTROL

All participants were introduced to manual drone control. For manual drone control Tello EDU drones we use Tello app.

The programming languages (applications) we used were: DroneBlocks, Scretch and Python.

A. Tello app

Tello app is available at Google Play and Apple Store and it is very easy to install and use. On Fig. 2. Is an image of the application. The drone is controlled with the help of virtual joysticks. There are few more button such as: TakeOff and Laning, Flights Modes, Camera/Video swithc and indicators for battery, Wi-Fi and Bluetooth.

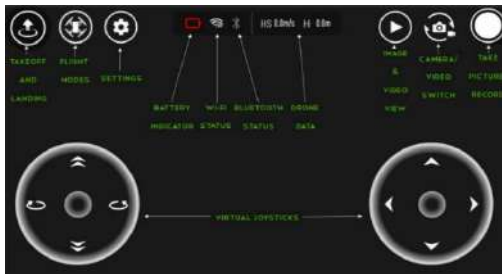


Fig. 2. Tello app.

B. DroneBlocks

DroneBlocks is a drag and drop block programming environment that supports many of the leading drones, such as: Phantom 3, Phantom 4, Mavic Pro, Mavic Air, Spark, and Tello [2]. This is a free application that is available on different platforms: iOS App Store, Google Play Store and Chrome App Store. One of the biggest advantages is that a mobile phone can be used to make a program and operate a drone. After a couple of introductory tasks training participants had to simulate movement from an Fig. 3.

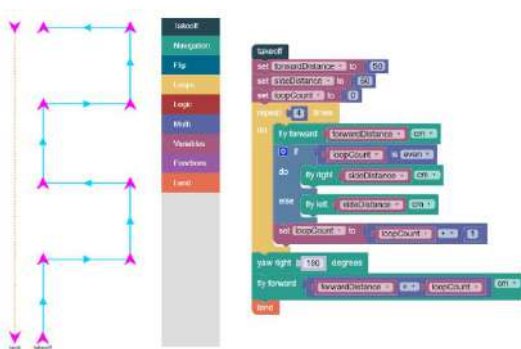


Fig. 3. DroneBlocs code for control drone.

A great feature of DroneBlocks is that you can view JavaScript in your block mission. That way, users can learn the basics of JavaScript. On Fig. 4. is presented solution in JavaScript for drone movement from Fig. 3.

```
var forwardDistance;
var sideDistance;
var loopCount;

takeOff();
forwardDistance = 50;
sideDistance = 50;
loopCount = 0;

for (var count = 0; count < 4; count++) {
  flyForward(forwardDistance, "cm");
  if (loopCount % 2 == 0) {
    flyRight(sideDistance, "cm");
  } else {
    flyLeft(sideDistance, "cm");
  }
  loopCount = loopCount + 1;
}

yawRight(180);
flyForward(forwardDistance * loopCount, "cm");
land();
```

Fig. 4. Tello app.

C. Scratch programming language

Scratch is a visual programming language. Its drag-and-drop interface with colorful blocks makes it one of the most intuitive programming languages to learn. Unlike traditional programming languages like Python where the coder must write out the code, Scratch empowers young people and adults alike to easily build interactive games and programs by connecting code blocks together. In the Scratch programming language main task was to create program that will control the movement of the drone by pressing a key on the keyboard (Fig. 5).

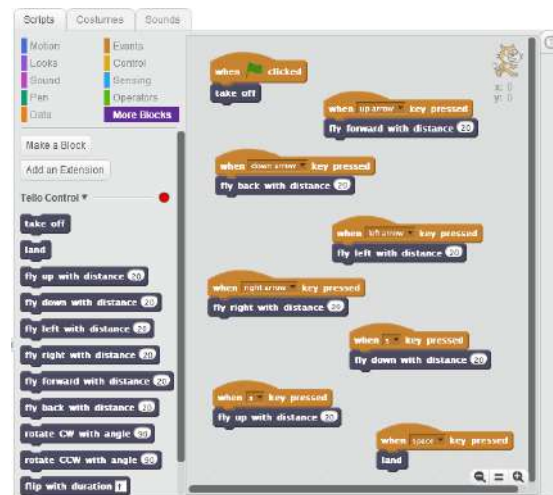


Fig. 5. Control drone movement in Scretch.

D. Python programming language

Python is programming language that lets you work quickly and integrate systems more effectively. To program Tello drone we need libraries. We need djitellopy library and opencv-python library [3]. In Python we demonstrate how to create program (Fig. 45.) to control drone by keyboard and we add a button that will save the current camera image to a file which will allow us to get a picture of the strawberries and check if any of the strawberries is ready to be harvested.

```
import KeyPress as kp
from djitellopy import tello
import time
import cv2
kp.init()
me = tello.Tello()
me.connect()
global img
me.streamon()
me.takeoff()
def getKeyboardInput():
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 50
    if kp.getKey("LEFT"):
        lr = speed
    elif kp.getKey("RIGHT"):
        lr = -speed
    if kp.getKey("UP"):
        fb = speed
    elif kp.getKey("DOWN"):
        fb = -speed
    if kp.getKey("w"):
        ud = speed
    elif kp.getKey("s"):
        ud = -speed
    if kp.getKey("a"):
        yv = speed
    elif kp.getKey("d"):
        yv = -speed
    if kp.getKey("q"): me.land(); time.sleep(3)
    if kp.getKey("i"): me.takeoff()
    if kp.getKey("z"):
        cv2.imwrite(f'Resources/Images/{time.time()}.jpg', img)
        time.sleep(0.3)
    return [lr, fb, ud, yv]
while True:
    vals = getKeyboardInput()
    me.send_rc_control(vals[0], vals[1], vals[2], vals[3])
    img = me.get_frame_read().frame
    img = cv2.resize(img, (360, 240))
    cv2.imshow("Image", img)
    cv2.waitKey(1)
```

Fig. 5. Python code for control drone by keyboard.

E. Competition

After completing the training, all students participated in a competition (Fig. 6.) that consisted of three parts: manual drone control, drone control with the help of the DroneBlocks and the Python. There were 4 teams. In each team was one participant from each country. Mission 1. was to use Tello App and cross over the given obstacles. In Mission 2. they have to write program in Python in order to mimic motion. Goal of Mission 3. was to write program in DroneBlocks which will allow the drone to move according to a certain scheme.

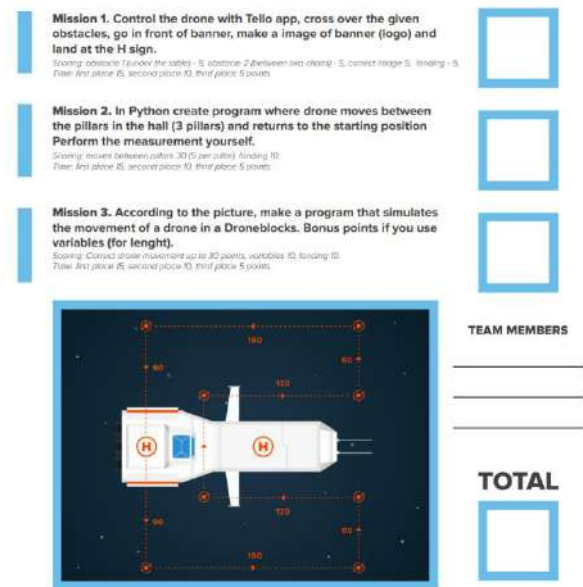


Fig. 6. Competition.

III. CONCLUSION

The competition was conducted with the aim of seeing which part of drone control the students best understood and mastered the best. The results showed that the students understood manual drone control best – via Tello App. The next task they did best was with DroneBlocks. The weakest results were achieved when students use Python which is expected taking into account that the students had a basic knowledge of programming.



Fig. 7. Awarding medals after the competition.

Team members and all participants are extremely satisfied with the project activities and believe that the similar activities should continue in the future so this project can be developed and be applied in various spheres of agriculture.

ACKNOWLEDGEMENT

We would like to thank our mentors, professor of programing, Zoran Negovanović and professors of informatics, Igor Tomić and Vesna Ilić, for the time and commitment to the project.

REFERENCES

- [1] <https://www.ryzerobotics.com/tello/downloads>
- [2] <https://learn.droneblocks.io/p/introduction-to-drone-programming-with-tello>
- [3] <https://djitellopy.readthedocs.io/en/latest/tello/>
- [4] Miloš S. Marina, Petar D. Mirosavljević, Pregledni rad: Tehnike leta dronova u vazdušnom saobraćaju i transportu, Univerzitet u Beogradu, Saobraćajni fakultet, Beograd

Primena cirkularnog bafera za prijem podataka korišćenjem UART komunikacije i DMA pristupa

Nikola Mitrović, Milan Đorđević, Sandra Veljković

Sadržaj - U ovom radu prikazan je princip primene cirkularnog bafera za prijem podataka korišćenjem UART komunikacije i DMA pristupa. Rad detaljnije opisuje strukturu cirkularnog bafera, a onda i mogućnosti njegove primene. Princip rada prikazan je na primeru prijema podatka nepoznate dužine koji korišćenjem AT komandi šalje modul ESP8266-01. Validnost prijema je prikazana korišćenjem terminala.

I. TEORIJSKI UVOD

Razvoj savremenih komunikacionih tehnologija podstaknut je velikim razvojem komponentata. Zahvaljujući razvoju mikroelektronike, na jednom čipu moguće je integrisati veliki broj kola i podsistema koji značajno proširuju mogućnosti sistema. Moderni mikrokontroleri, npr. oni sa ARM M3-Cortex jezgrom objedinjuju preko 40 perifernih jedinica, mogu paralelno obavljati veliki broj instrukcija, čak mogu pokretati i operativne sisteme [1]. Jedan od zadataka i problema sa kojim se susreće moderna industrija čak nije više dominantno usmeren na razvijanje novih i moćnijih čipova (iako je to naravno i dalje dominantna delatnost), već i na pripremu odgovarajuće tehničke literature koja inženjerima omogućava da do maksimuma iskoriste resurse postojećih mikrokontrolera. Ipak, kako se nove komponente i softverski alati konstantno razvijaju, ovaj zadatak postaje sve komplikovaniji.

Jedan od specifičnih problema koji će biti detaljnije analiziran u ovom radu je primena cirkularnog bafera za prijem podataka korišćenjem UART (*Universal Asynchronous Receive Transmit*) komunikacije i DMA (*Direct Memory Access*) pristupa, konkretno kada je potrebno primiti podatak nepoznate dužine. U drugom delu rada će biti prikazana primena na jednom primeru serijske komunikacije.

Od korišćenih resursa, biće detaljnije opisan princip rada cirkularnog bafera, UART komunikacije i DMA pristupa, a nakon toga će biti prikazana njihova sprega na primeru komunikacije sa ESP8266 modulom. Taj modul je izabran jer prilikom komunikacije šalje poruke promenljive dužine čija se sadržina ne može uvek predvideti.

N. Mitrović i S. Veljković su studenti 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

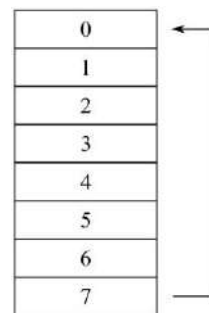
M. Đorđević je student na Akademiji Tehničko Vaspitačkih Strukovnih Studija – Odsek Niš, Aleksandra Medvedeva 20, 18000 Niš, Srbija, E-mail: milan94srb@gmail.com

II. KORIŠĆENI RESURSI

Glavni resurs čije je razumevanje potrebno za opis primene je cirkularni bafer. On može imati raznolike namene, a jedna od njih može biti i u okviru UART komunikacije. Nakon toga, sama komunikacija može biti sa ili bez DMA pristupa.

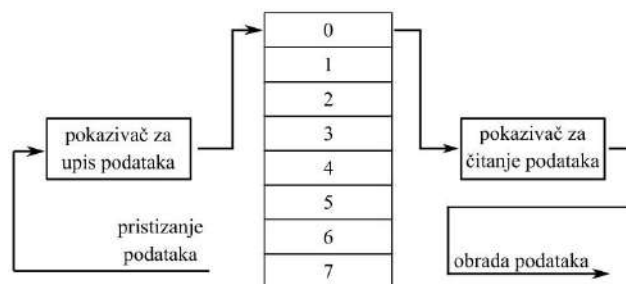
A. Cirkularni bafer

Cirkularni bafer je poseban tip strukture podataka koju je moguće definisati na novijim tipovima kontrolera [2]. Bafer ima naziv cirkularni jer nakon pristupa poslednjoj lokaciji u baferu, naredna lokacija za pristup je početna, pa se podaci upisuju „u krug“, odnosno cirkularno. Na slici 1 prikazan je dijagram cirkularnog bafera.



Slika 1. Dijagram rada cirkularnog bafera.

Princip rada cirkularnog bafera najjednostavnije je objasniti korišćenjem dva pokazivača (*pointers*). Jedan pokazivač odnosi se na upis podataka (*Write buffer pointer*), a drugi pokazivač se odnosi na čitanje podataka (*Read buffer pointer*). Korišćenjem ova dva pokazivača moguće je kontrolisati protok podataka kroz bafer, kao što je prikazano na slici 2.



Slika 2. Dijagram rada pokazivača cirkularnog bafera.

Pokazivač za upis pokazuje na lokaciju u koju treba upisati naredni primljeni podatak. Lokacija upisa se inkrementira nakon što se podaci upišu u nju. Pokazivač za čitanje pokazuje na lokaciju sa koje treba očitati naredni podatak. Lokacija se inkrementira nakon što se njen sadržaj smesti u odgovarajuću promenljivu, odnosno nakon što se izvrši *fetching* podataka.

Na ovaj način, cirkularni bafer vrši emulaciju FIFO (*First In First Out*) pristupa, gde prvi podatak koji se čita jeste prvi podatak koji je upisan. Glavna prednost korišćenja ovog tipa bafera je ta što nije potrebno tačno odrediti dužinu bafera. Svakako je potrebno definisati kolika je dužina bafera (10, 15, 120 memorijskih lokacija), ali se u bafer mogu upisivati i podaci koji su manje i veće dužine od veličine bafera. Kod standardnog bafera (linijskog) od recimo 20 lokacija, upis bi se izvršio na tih 20 lokacija, i nakon toga ne bi bilo moguće dalje upisivati podatke u taj bafer. Kod cirkularnog bafera, ne postoji takvo ograničenje, pa je dosta pogodniji za primene kada je potrebno upisivati podatke napoznatih dužina.

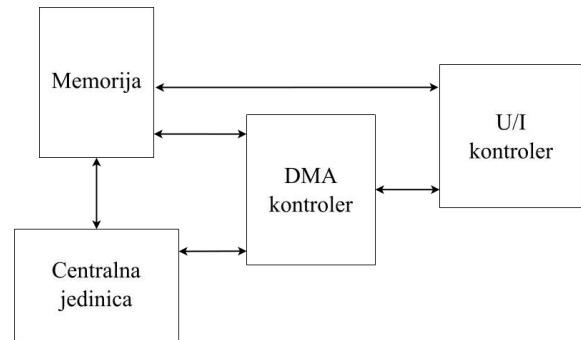
B. UART komunikacija

Jedan od najjednostavnijih i dosta primenjenih tipova serijske komunikacije je UART komunikacija. UART komunikacija je tip komunikacije za serijsko slanje podataka [3]. UART uređaj koji šalje (predajnik, eng. *transmitter*) podatke, konvertuje paralelnu formu podataka u serijsku formu podataka, i kao takvu je šalje prema UART prijemniku. Prijemnik prima podatke u serijskoj formi, i konvertuje ih nazad u paralelnu formu podataka, nakon čega se ti podaci dalje obrađuju. Komunikacije se obavlja preko dve veze (konekcije), od TX pina predajnika do RX pina prijemnika, u oba smera.

C. DMA pristup

Prenos podataka kod MCU, odnosno upis i ispis (ulaz-izlaz) podataka može biti realizovan na tri načina. Prvi je najjednostavniji. Informacije se upisuju u registre neke periferalne jedinice, i kada taktni signal redom odvede *Program Counter* do adrese registra te periferalne jedinice, dolazi do prosleđivanja tih podataka memoriji centralne jedinice MCU. Ovaj način je najsporiji. Drugi način podrazumeva identičan proces, samo se do adrese registra ne dolazi redom, već korišćenjem prekida (interrupta). Ovo je brži način. Treći, najbrži način podrazumeva tehniku direktnog pristupa memoriji. Direktni pristup memoriji (skr. DMA od eng. *Direct Memory Access*) je način upisa ili ispisa podataka kada brzina MCU da izvrši ove instrukcije nije dovoljno velika, odnosno kada je potrebno povećati ovu brzinu [4]. Podrazumeva postojanje dodatnog kontrolera (DMA kontrolera, koji je najčešće integrisan u MCU) koji direktno kontroliše magistralu memorije MCU. DMA omogućava da podatak sa neke periferalne jedinice (kao npr. UART ili nekog tajmera) dospe direktno do memorije MCU, bez da ga sam MCU postavi na tu

memorijsku lokaciju. Dakle, DMA omogućava efikasniju upotrebu prekida (interrupta), odnosno omogućava veću protočnost podataka. Osnovna blok šema DMA prenosa prikazana je na slici 3.

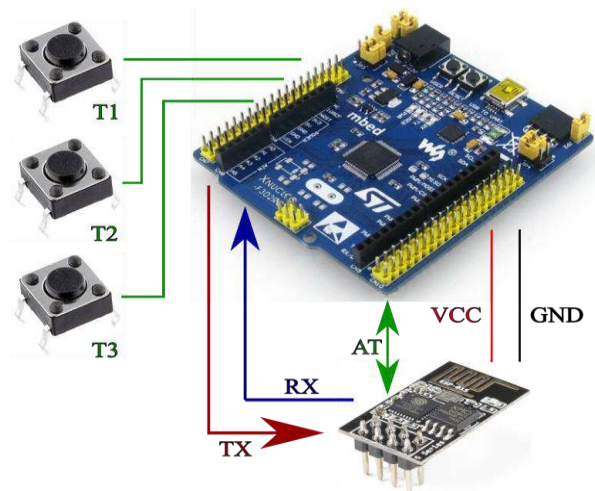


Slika 3. Blok šema DMA prenosa.

Najčešća primena DMA tehnike je kada je potrebno izvršiti prijem podataka a brzina komunikacija je jako velika. Potrebno je izbegnuti slučaj u kome se deo podataka koje treba primiti gubi jer MCU ne reaguje dovoljno brzo da izvrši prijem svakog nadolazećeg podatka. Korišćenjem DMA, nema gubitka podataka.

III. PRIMER PRIMENE

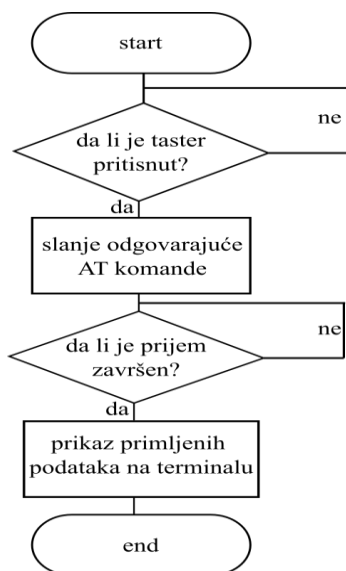
Primer primene cirkularnog bafera pokazan je komunikaciji između STM32 mikrokontrolera (MCU) i ESP8266-01 modula. Moguće je koristiti bilo koji MCU koji ima mogućnost kreiranja strukture podataka kakav je cirkularni bafer. Blok šema kola za primenu prikazana je na slici 4.



Slika 4. Blok šema sistema za primer primene cirkularnog bafera.

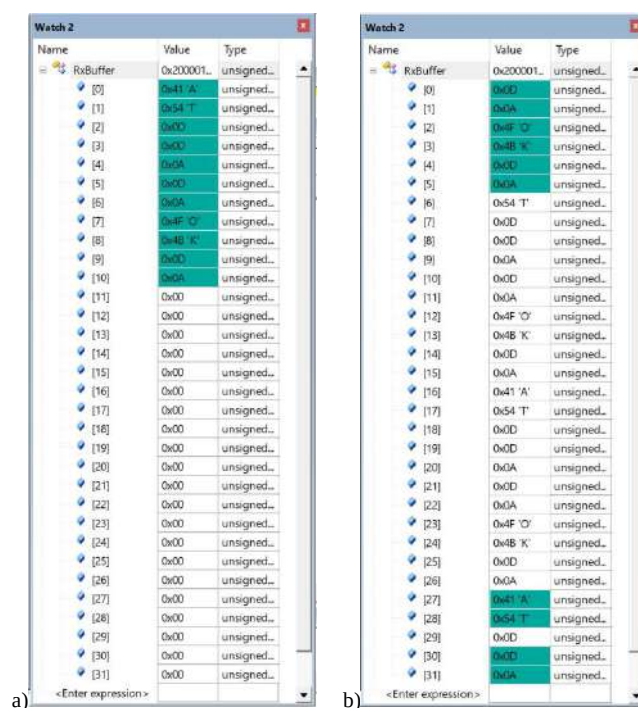
STM32F302 je MCU kompanije STM sa Cortex-M3 jezgrom [5]. Spada u grupu MCU opšte namene. Ima 64 pina i jako širok opseg perifernih jedinica. Radni napon je 3.3 V. Od interesa za ovaj rad je periferna jedinica za UART komunikaciju. STM32F302 ima tri ovakve

Proces komunikacije prikazan je algoritmom datim na slici 5.



Slika 5. Algoritam procesa prijema podatka.

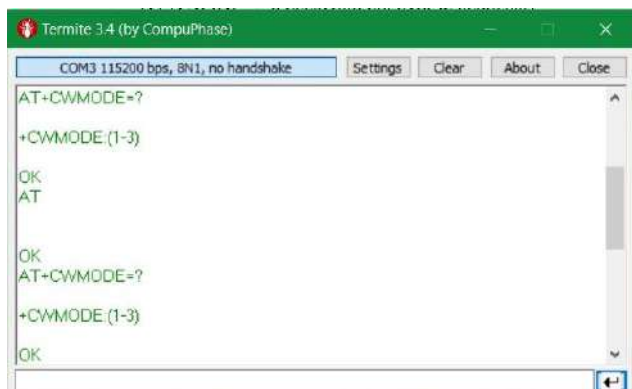
Testiranje primene kao i sadržaj registra tokom rada biće prikazani softverskim alatom *Keil Microvision* [8]. Ovaj alat omogućava debugging mode koji može pratiti sadržine promenljivih u realnom vremenu. Na slici 6 prikazan je primer prijema AT komande za inicijalizaciju i AT komande za resetovanje (pritisak tastera *T1* i *T2*). Na slici 6a) je prikazan prijem podatka kada je cirkularni bafer pre prijema prazan. Ponavljanjem pritiska tastera, dolazi do slanja AT naredbi, pa se prijemni bafer puni odgovorima naredbi. Na slici 6b) prikazan je primer okreta bafera (*rollover*). Kada pristigne odgovor za čiji prijem nema dovoljno memorijskih lokacija na dnu bafera, problem gubitka podataka se zaobilazi tako što se „višak“ podataka prepisuje preko podataka koji se nalazi na početnim memorijskim lokacijama bafera (vrhu bafera). Na ovaj način, bez obzira na dužinu primljene poruke, nema gubitka podataka.



Na slici 7 prikazan je prijem odgovora AT komande koja se aktivira pritiskom na taster T3. U zavisnosti od toga koji režim rada Wi-Fi modula je aktivan, postoji više različitih odgovora koji imaju različite dužine koji se mogu dobiti ovom AT komandom. Na slici 7a) prikazan je

Watch 1				Watch 2			
Name	Value	Type		Name	Value	Type	
0x00000000	0x00000000	char		0x00000000	0x00000000	char	
[0]	0x00 'A'	char		[0]	0x00 'A'	char	
[1]	0x04 'T'	char		[1]	0x04 'T'	char	
[2]	0x28 '+'	char		[2]	0x28 '+'	char	
[3]	0x43 'C'	char		[3]	0x43 'C'	char	
[4]	0x57 'W'	char		[4]	0x57 'W'	char	
[5]	0x4D 'M'	char		[5]	0x4D 'M'	char	
[6]	0x4F 'O'	char		[6]	0x4F 'O'	char	
[7]	0x44 'D'	char		[7]	0x44 'D'	char	
[8]	0x45 'E'	char		[8]	0x45 'E'	char	
[9]	0x3D '='	char		[9]	0x3D '='	char	
[10]	0x3F '?'	char		[10]	0x3F '?'	char	
[11]	0x00	char		[11]	0x00	char	
[12]	0x00	char		[12]	0x00	char	
[13]	0x0A	char		[13]	0x0A	char	
[14]	0x28 '+'	char		[14]	0x28 '+'	char	
[15]	0x43 'C'	char		[15]	0x43 'C'	char	
[16]	0x57 'W'	char		[16]	0x57 'W'	char	
[17]	0x4D 'M'	char		[17]	0x4D 'M'	char	
[18]	0x4F 'O'	char		[18]	0x4F 'O'	char	
[19]	0x44 'D'	char		[19]	0x44 'D'	char	
[20]	0x45 'E'	char		[20]	0x45 'E'	char	
[21]	0x3A ':'	char		[21]	0x3A ':'	char	
[22]	0x28 '('	char		[22]	0x28 '('	char	
[23]	0x31 '!'	char		[23]	0x31 '!'	char	
[24]	0x2D '-'	char		[24]	0x2D '-'	char	
[25]	0x33 '3'	char		[25]	0x33 '3'	char	
[26]	0x29 '9'	char		[26]	0x29 '9'	char	
[27]	0x00	char		[27]	0x57 'W'	char	
[28]	0x0A	char		[28]	0x4D 'M'	char	

Sadržina bafera *TxBuffer* prikazuje se i na terminalu na odgovarajuć način. Odgovarajućim ispisivanjem potvrđuje se ispravnost primene, kao što je prikazano na interfeisu terminala *Termite* na slici 8.



80

U ovom radu je prikazan primer primene cirkularnog bafera za prijem podatka nepoznate dužine korišćenjem UART komunikacije i DMA pristupa. Primena je pokazana na primeru komunikacije između STM32 mikrokontrolera i ESP8266-01 modula. Dalje unapređenje ovog pristupa ogleda se u formiranju biblioteke koja je u mogućnosti da obradi ceo set AT komandi i bude skladna sa različitim verzijama modula. Kako AT komande nisu tip komunikacije karakterističan samo za Wi-Fi module, primena se može proširiti i na druge module koji koriste ove principe.

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026. Autori se zahvaljuju prof. dr Danijelu Dankoviću na motivaciji i pomoći prilikom izrade rada.

- [1] Tehnička specifikacija za ARM Cortex-M3. Dostupno na: https://www.keil.com/dd/docs/datashts/arm/cortex_m3/r2p0/di0337g_cortex_m3_r2p0_trm.pdf pristupljeno 10.10.2021.
- [2] D. Ponikvar, "The use of a circular buffer", Dostupno na: <https://www.fmf.uni-lj.si/~ponikvar/STM32F407%20project/Ch13%20The%20use%20of%20circular%20buffer.pdf>, pristupljeno 10.10.2021.
- [3] Universal Asynchronous Receiver/Transmitter (UART) - User Guide, Texas Instruments, 2010. Dostupno na: <https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf>, pristupljeno 10.10.2021.
- [4] Communication peripheral FIFO emulation with DMA and DMA timeout in STM microcontrollers, STM, 2009. Dostupno na: https://www.st.com/resource/application_note/cd0025668.pdf, pristupljeno 10.10.2021.
- [5] Tehnička dokumentacija za STM32F302R8. Dostupno na: <https://www.st.com/resource/en/datasheet/stm32f302c8.pdf> pristupljeno 10.10.2021.
- [6] Tehnička dokumentacija za okruženje xNucleo F302R8. Dostupno na: <https://static.chipdir.ru/lib/DOC001552162.pdf>, pristupljeno 10.10.2021.
- [7] Tehnička dokumentacija za modul ESP8266-01. Dostupno na: <https://www.microchip.ua/wireless/esp01.pdf>, pristupljeno 10.10.2021.
- [8] Softverski alat Keil Microvision. Dostupno na: <https://www.keil.com/download/>, pristupljeno 10.10.2021.

Mašinsko učenje kao pomoć kod prognoze kliničkog ishoda nakon moždanog udara

Emilija Čojbašić

Sadržaj – Do sada su razvijeni brojni modeli za predikciju smrtnosti kod spontane intracerebralne hemoragije (ICH), što je jedan od tipova moždanog udara sa visokom smrtnošću [1][2]. Predikcija kliničkog ishoda kod ICH značajna je pomoć neurologu kod donošenja odluka o optimalnom tretmanu pacijenta i personalizovanoj terapiji. U ovom radu razvijen je neuro-fazi model za predikciju smrtnosti nakon spontanog ICH na bazi inicijalnih kliničkih parametara i upoređen sa publikovanim modelima na bazi veštačkih neuronskih mreža i logističke regresije. U radu je korišćen skup podataka o pacijentima sa spontanom ICH publikovan u studiji [3], gde su pacijenti tretirani u petogodišnjem periodu u univerzitetskom kliničkom centru koji pripada tercijarnoj zdravstvenoj zaštiti. Razmatrani su pacijenti stariji od 18 godina koji su imali su dokaze o spontanom ICH na kompjuterizovanoj tomografiji mozga. Analizirani su podaci o 411 pacijenata (199 muškaraca i 212 žena), prosečne starosti od 67,35 godina, od čega je 256 (62,29 %) pacijenata preminulo tokom bolničkog tretmana a 155 (37,71 %) pacijenata je preživelo posledice moždanog udara. Razvijeni neuro-fazi modeli pokazali su superiornost u odnosu na standardne modele logističke regresije, dok je tačnost klasifikacije bila gora u odnosu na model zasnovan na veštačkim neuronskim mrežama publikovan u [3]. Sa druge strane, razvijeni neuro-fazi modeli imaju druge prednosti koje su diskutovane u radu.

I. UVOD

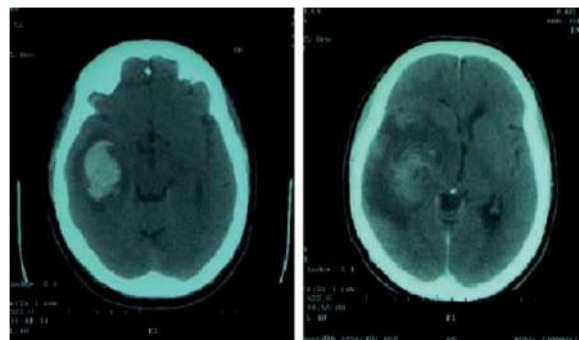
Akutni moždani udar (MU, šlog) je treći uzrok smrtnosti i prvi uzrok invaliditeta u razvijenim zemljama sveta: procenjuje se da svakih 45 sekundi jedna osoba doživi, a svaka tri minuta jedna osoba umire od posledica MU. Incidencija MU je 150–200 slučajeva na 100000 osoba godišnje. Radi se stoga o veoma čestom oboljenju sa visokim stepenom smrtnosti ($\approx 30\%$; najveći u prvim mesec dana). Ne manje ozbiljna je i činjenica da je ovo neurološka bolest sa najvećim stepenom invaliditeta [1][2].

Moždani udar se klasifikuje prema patologiji, pa razlikujemo (a) *ishemički MU*, koji nastaje usled nedovoljnog priliva ili potpunog prekida dotoka krvi u određeni region mozga (b) *hemoragički MU*, koji nastaje usled rupture krvnog suda ili vaskularne malformacije, sa izlivom krvi u mozak [1].

Vrsta hemoragičkog moždanog udara, spontana ili primarna intracerebralna hemoragija (ICH), je druga najzastupljenija forma šloga, koja obuhvata 10–30% prvih moždanih udara [1]. Smrtnost u prvim 30 dana kod ICH je

E. Čojbašić je student treće godine studija Računarstva i informatike na Elektronskom Fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija i studentski član IEEE, E-pošta: ecojbasic@elfak.rs

30–55% sa polovinom smrtnih ishoda u akutnoj fazi, posebno tokom prvih 48 časova [1][2][4]. Kompjuterizovana tomografija (CT) mozga bolesnika sa intracerebralnim krvavljenjem prikazana je na slici 1.



Slika 1. CT bolesnika sa intracerebralnim krvavljenjem [1].

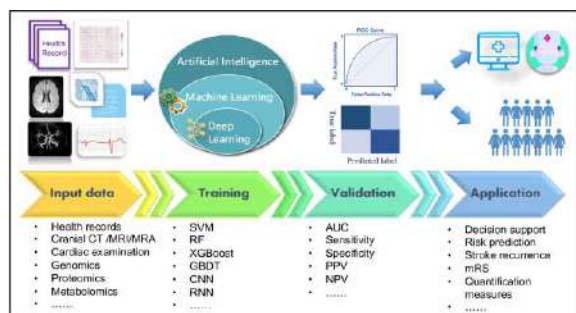
Mnoge studije su identifikovale široki opseg faktora koji imaju prediktivnu vrednost za procenu ishoda nakon akutnog ICH. Iz tih studija, razvijeni su brojni modeli za predikciju ishoda za prognozu smrtnosti i funkcionalnu prognozu [3][4].

U kliničkim studijama, mnoge od promenljivih mogu biti međusobno povezane, i proces selekcije često dovodi do isključivanja nekih varijabli iz statističkih modela [3][4][5][6]. Time se čini kompromis u pogledu prognostičke snage modela i dovodi do manje tačne klasifikacije slučajeva, čime statistički modeli često gube specifičnost i preciznost koja je neophodna za primenu u kliničkoj praksi kod donošenja odluka.

Veštačka inteligencija i mašinsko učenje su danas prisutni u tretmanu moždanog udara, a neki od načina za njihovu primenu klasifikovani su uopšteno na slici 2 [7]. Veštačka inteligencija (engl. Artificial intelligence, skraćeno AI) [8] je podoblast računarstva i predstavlja široko područje računarske nauke koje ima za globalni cilj da mašine funkcionišu i zaključuju podražavajući ljudski mozak. Danas posebno aktivna oblast, mašinsko učenje (engl. Machine Learning, skraćeno ML) [9], je podoblast veštačke inteligencije čiji je cilj konstruisanje algoritama i računarskih sistema koji su sposobni da se adaptiraju na analogne nove situacije i uče na bazi iskustva. Razvijene su različite tehnike mašinskog učenja za različite zadatke.

U ovom radu korišćeni su sistemi mašinskog učenja poznati pod pseudonimom ANFIS, što je oznaka za „adaptive network-based fuzzy inference system” [10][11], a što označava neuro-fazi sistem koji je vrsta veštačke

neuronske mreže bazirane na Takagi–Sugeno fazi sistemu. Pošto integriše neuronske mreže i fazi sisteme ima potencijale da obuhvati benefite obe tehnologije u jedinstvenom okviru. Fazi logika (FL) predstavlja generalizaciju klasične Bulove logike, a na njoj i teoriji fazi skupova zasnovani fazi sistemi mogu se posmatrati kao generalizacija konvencionalnih ekspertskih sistema baziranih na pravilima. Sa druge strane, veštačke neuronske mreže (ANN) predstavljaju paralelne, distribuirane, adaptivne sisteme za procesiranje informacija koji razvijaju svoju funkcionalnost kao odgovor na izloženost podacima i informacijama.



Slika 2. Razvoj, validacija i primena tehnika veštačke inteligencije u tretmanu moždanog udara [7].

Ovo istraživanje je sprovedeno sa ciljem da se razviju ANFIS neuro-fazi modeli mašinskog učenja za prognozu smrtnosti nakon spontanog ICH, na bazi inicijalnih kliničkih parametara, kao prilog modelima zasnovanim na veštačkim neuro-mrežama i logističkoj regresiji predloženim u [3], u cilju upoređenja rezultata.

II. METODE

A. Pacijenti i klinički podaci

Podaci o pacijentima korišćeni u ovom radu preuzeti su iz rada [3] i prikupljeni su u studiji sprovedenoj u univerzitetskom kliničkom centru koji je ustanova tercijarne zdravstvene zaštite u toku petogodišnjeg perioda.

Ukupno 892 pacijenata je u posmatranom periodu primljeno u bolnicu sa dijagnozom ICH. Iz različitih razloga (trauma, tumor, pacijenti prezentovani više od 6h nakon šloga, pacijenti prebačeni iz drugih bolnica, pacijenti na antikoagulantnoj terapiji, pacijenti koji su podvrgnuti hirurškom tretmanu, itd.) isključen je iz analize ukupno 481 pacijent dok je konačno analizirano 411 pacijenata sa ICH dijagnozom na prijemu u bolnicu: 199 muškaraca i 212 žena, prosečne starosti 67,35 godina (Tabela 1). Među njima je 256 (62,29%) pacijenata umrlo tokom bolničkog lečenja, a 155 (37,71%) pacijenata je preživelo i otpušteno je iz bolnice nakon lečenja, što direktno potvrđuje literaturne podatke o visokoj smrtnosti kod ICH.

TABELA I
NEKE KLINIČKE KARAKTERISTIKE PACIJENATA [3]

Varijabla	Svi (n=411)	Preživeli (n=155)	Premrli (n=256)
Starost (godine)			
Srednja±SD	67,35±10,71	67,12±10,30	67,49±10,97
Opseg	26-93	41-93	26-91
Pol (n)			
Muški	199	69	130
Ženski	212	86	126
GCS skor (3-15)			
Srednji±SD	9,19±4,54	12,41±3,45	7,25±3,99
Opseg	3-15	3-15	3-15
Srednji KP (mmHg)			
Srednji±SD	127,99±24,27	126,05±21,68	129,16±25,68
Opseg	53,33-220	53,33-200	66,67-220

SD=standardna devijacija; GCS=Glasgow Coma Scale; KP=krvni pritisak.

Klinički podaci uključeni u analizu sastojali su se od starosti, pola, istorije istovremenih bolesti i porodične istorije. Podaci prikupljeni tokom prezentacije uključivali su interval od pojave simptoma do prijema, Glazgov koma skor (GCS) [1][2][3], nivo svesti i krvni pritisak (KP). GCS je određen prema originalnim uputstvima (otvaranje očiju na skali 1–4, verbalni odgovor na skali 1–5 i najbolji motorni odgovor na skali 1–6). Takođe je određivan nivo svesti (1:budnost, 2:pospanost, 3:stupor, 4:koma). Vrednosti arterijskog krvnog pritiska (KP) merene su u ležećem položaju a srednji KP se izračunavao kao 1/3 sistolnog + 2/3 dijastolnog krvnog pritiska.

Dijagnoza spontanog ICH zasnovana je na neurološkim nalazima, a potvrđivana je CT nalazom mozga u roku od 24 sata od početka simptoma. Praćeni ishod bio je mortalitet tokom hospitalizacije. Svi pacijenti su primljeni na odeljenje intenzivne nege neurologije i njima su istovremeno upravljali tim intenzivne nege i neurolog. Nijedan posmatrani pacijent nije imao ograničenja u nezi.

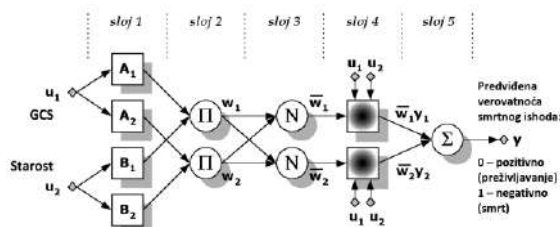
B. Logistička regresija i ANN model

U studiji [3] logistička regresija (LR) [12] je korišćena kao standardni pristup za analizu ishoda preživljavanja merenog kao dihotomna promenljiva, koja se može tumačiti kao verovatnoća smrti (0: preživljavanje, 1: smrt). Ukupna tačnost konačnog modela određena je poređenjem predviđanih vrednosti sa stvarnim ishodima.

ANN korišćena u [3] bila je standardna neuronska mreža tipa višeslojni perceptron sa četiri sloja: ulaznim slojem, dva skrivena sloja i izlaznim slojem. Pojedinačni mrežni izlaz bio je isti, ishod preživljavanja pacijenata, izražen kao verovatnoća smrtnog ishoda.

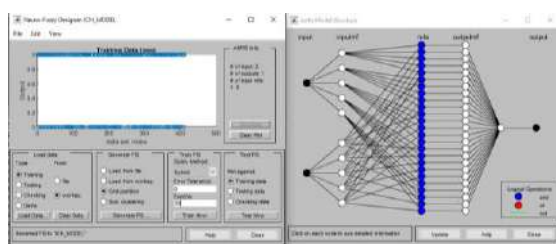
C. ANFIS model

ANFIS model za predikciju kliničkog ishoda kod ICH koji je predložen u ovom radu prikazan je na slici 3. Prikazani model je tipičan predstavnik familije ANFIS modela koji su razmatrani sa različitim brojem i selekcijom kliničkih parametara na ulazu. Model na slici 3 koji se ističe svojim performansama i jednostavnošću ima dva ulaza (GCS i starost) i jedan izlaz (prognozirana verovatnoća smrtnog ishoda).



Slika 3. ANFIS model sa dva ulaza (GCS i starost) i jednim izlazom (verovatnoća smrtnog ishoda) sa pet slojeva i po dva fazi skupa po ulazu.

Originalni hibridni algoritam obučavanja ANFIS sistema, koje je korišćen kod obučavanja razvijenih ANFIS modela, sastoji se od dva sukcesivna dela (algoritma) [10][11]: 1) metod povratne propagacije, odnosno najstrijmijeg spusta (BP/GD), koji proračunava signale greške (definisana kao izvod kvadratne greške u odnosu na izlaz svakog elementa) rekurzivno od izlaznog sloja unazad ka ulaznim elementima, i 2) rekurzivni metod najmanjih kvadrata (RLSE), kojim se određuje optimalni set posledičnih parametara. Sa fiksniranim vrednostima uzročnih parametara, izlaz se može izraziti kao linearna kombinacija posledičnih parametara. ANFIS modeli su projektovani i obučavani Matlab softverskim paketom (The MathWorks Inc., USA), i njegovim dodatkom Fuzzy Logic Toolbox, verzijama 2021a. Na slici 4 pokazan je interfejs za obučavanje ANFIS sistema i struktura jednog ANFIS modela sa 2 ulaza.



Slika 4. Grafički interfejs za obučavanje ANFIS modela u MATLAB paketu i struktura jednog ANFIS modela.

Skup podataka je nasumično podeljen u skupove za treniranje, validaciju i testiranje. Skup za testiranje (70%, 287 pacijenata) je prezentiran neuro-fazi mreži tokom treniranja, i mreža je hibridnim algoritmom podešena prema grešci procene. Validacioni skup (15%, 62 pacijenata) je korišćen da se u toku obučavanja procenjuje

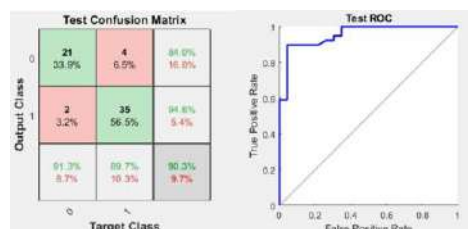
generalizacija mreže i da se prekine obučavanje kada generalizacija prestane da se poboljšava. Konačno, skup za testiranje (15%, 62 pacijenata) nije imao uticaja na obučavanje i korišćen je da se obezbedi nezavisna mera performansi ANFIS modela nakon obučavanja.

III. REZULTATI

Izbor ulaznih kliničkih parametara za razvijene ANFIS modele mašinskog učenja baziran je na analizi uticajnosti parametara iz literature, posebno uz pokušaj da se zadrži jednostavnost modela i skroman broj ulaza kako bi se napravio iskorak ka upotrebljivosti i razumljivosti modela i jednostavnosti primene u kliničkoj praksi. S obzirom na to i osetljivost ANFIS mreža na broj ulaza, razmatrani su modeli sa dva i tri ulaza. U studiji [3] pacijenti koji su umrli imali su verovatnije niži GCS skor i nivo svesti pri prijemu. Umrli pacijenti su imali takođe verovatnije niži tzv. MG indeks (srednji KP/GCS skor), što ukazuje i na značaj krvnog pritiska na predikciju ishoda. Nivo svesti, verbalni odgovor, starost, muški pol i krvni pritisak uključeni su u konačnu regresionu jednačinu, dok su kod ANN modela ulazne varijable mreže bile su starost, pol, puls i srednji arterijski pritisak, tri parametra GCS-a i nivo svesti. U ovom radu kod ANFIS modela sa dva ulaza kao ulazne promenljive odabrani su klinički parametri GCS skor i starost, dok je kod ANFIS modela sa tri ulaza dodat i nivo svesti. Alternativno je moguće razmotriti modele sa dva ulaza MG indeks i starost čime bi se uključio uticaj krvnog pritiska sa samo dve ulazne promenljive. Regresioni model [3] je tačno klasifikovao 73,55% preživelih pacijenata i 82,81% umrlih pacijenata, sa ukupno 79,32% tačnih klasifikacija. U nezavisnom uzorku za testiranje, ANN [3] je tačno predvidela ishode u 90,47% preživelih pacijenata i u 95,12% umrlih pacijenata, odnosno ukupno sa 93,55% tačnih predviđanja (Tabela 2).

TABELA II
TAČNOST KLASIFIKACIJE ZA RAZNE MODELE
(CUT-OFF VREDNOST $P=0.5$)

Model	Procenat tačno klasifikovanih ishoda
Logistička regresija	79,32
ANN	93,55
ANFIS 2 ulaza	83,90
ANFIS 3 ulaza	90,30



Slika 5. Matrica konfuzije i ROC kriva za ANFIS model sa tri ulaza na test skupu pacijenata (62 pacijenata).

Performanse razvijenih ANFIS modela su očekivano, s obzirom na primenjenu tehnologiju i broj ulaza, po rezultatima između regresionih i ANN modela. Ukupno je ANFIS najbolji model sa dva ulaza tačno klasifikovao 83,9% ishoda dok je najbolji ANFIS model sa tri ulaza pokazao tačnost u 90,30% slučajeva. Matrica konfuzije i ROC kriva za ANFIS model sa tri ulaza su prikazani na slici 5. Sva tri tipa modela su generalno pokazala dobre performanse, sa jasnom superiornošću modela ANN.

IV. DISKUSIJA I ZAKLJUČCI

Do danas je mali broj prognostičkih modela postiglo stepen tačnosti neophodan za praktičnu primenu u kliničkim uslovima. Studija [3] uz druge publikovane rezultate i rezultate prezentovane u ovom radu ukazuje na to da se statistički modeli kao i modeli zasnovani na mašinskom učenju, ako su dostupni, mogu primeniti kao prediktori mortaliteta kod pacijenata sa spontanom ICH. Pri tome su modeli zasnovani na mašinskom učenju po performansama superiorni u odnosu na standardne modele logističke regresije [13][4][7].

Model ANN za predikciju mortaliteta kod pacijenata sa ICH zbog većeg broja ulaza efikasnije koristi informacije koje su sadržane u kliničkim podacima, a ti zaključci se delimično odnose i na neuro-fazi ANFIS modele. Rezultatima se sugeriše potencijalna primenljivost modela zasnovanih na tehnikama mašinskog učenja kod predviđanja ishoda za pacijente sa spontanom ICH kod kliničkog odlučivanja [3][4]. Poboľjšani prognostički sistemi mogu dovesti do efikasnije identifikacije pacijenata kojima je potrebno intenzivno lečenje, u smislu dostupnosti novih i efikasnijih hirurških tehnika [14] ili novih terapijskih mogućnosti [6]. Iako dobijeni rezultati sugerišu superiornost ANN modela u odnosu na ANFIS modele, ANFIS modeli ipak imaju izvesne prednosti, a to su na primer mogućnost razmene ekspertskog znanja neurologa sa modelom u oba smera budući da je reč o vrsti fazi sistema, superiorni algoritam obučavanja, potencijalni pozitivan uticaj velikog broja parametara modela na mogućnost podešavanja i performanse i drugo.

Važno ograničenje modela mašinskog učenja predloženih u [3] i ovom radu je korišćenje samo početnih kliničkih parametara, dok podaci dobijeni daljom dijagnostikom nisu uzeti u obzir. Iako isključenje neuroradioloških faktora značajno slabi modele mašinskog učenja, ideju o razvoju prognostičkih modela zasnovanih samo na početnim kliničkim podacima pri prijemu i dalje vredi razmatrati zbog dobijanja brzi i rane predikcije smrtnosti nakon ICH, što je u nekim situacijama na odjeljenjima hitne pomoći od praktične vrednosti.

Nakon što se ANFIS i ANN modeli mašinskog učenja razviju, obuče i validiraju, spremni su za implementaciju na različitim računarskim platformama. U fazi aplikacije, rezultati predikcije su dostupni gotovo odmah. Ovu tehniku

mogu lako koristiti svi jer se modeli zasnovani na mašinskom učenju lako implementiraju.

Može se zaključiti da su dobijeni rezultati potvrdili primenljivost modela mašinskog učenja, ANN i ANFIS, zasnovanih na kliničkim parametrima dostupnim na prijemu, za predviđanja mortaliteta nakon spontanog ICH. Dalja proučavanja prednosti i ograničenja ovih pristupa potrebna su kod većih prospektivnih uzoraka osoba sa ICH [3][4][7], pre nego što se mašinsko učenje može pouzdano preporučiti za primenu u kliničkom okruženju.

ZAHVALNOST

Autorka rada iskazuje svoju iskrenu zahvalnost doc dr med.sc. Stevi Lukiću sa Medicinskog fakulteta Univerziteta u Nišu za nadzor nad njenim radom, dragocene smernice i podršku.

LITERATURA

- [1] V. Kostić, *Neurologija za studente medicine*, Medicinski fakultet Univerziteta u Beogradu, 2007.
- [2] A. L. Berkowitz, *Clinical Neurology and Neuroanatomy*, McGraw-Hill Education, 2016.
- [3] S. Lukić et al, "Artificial neural networks based early clinical prediction of mortality after spontaneous intracerebral hemorrhage", *Acta Neurol Belg*, 112:375–382, 2012.
- [4] T. Gregorio et al, "Prognostic models for intracerebral hemorrhage: systematic review and meta-analysis", *BMC Medical Research Methodology*, 18:145, 2018.
- [5] J. Pinhoa, A. S. Costa, J. M. Araújo, J. M. Amorim, C. Ferreira, "Intracerebral hemorrhage outcome: A comprehensive update", *Journal of the Neurological Sciences*, 398, 54–66, 2019.
- [6] C. Cordonnier, A. Demchuk, W. Ziai, C. Anderson, "Intracerebral hemorrhage: current approaches to acute management", *The Lancet Journal*, volume 392, issue 10154, p1257–1268, 2018.
- [7] L. Ding, C. Liu, Z. Li, Y. Wang, "Incorporating Artificial Intelligence into Stroke Care and Research", *Stroke*, 51: e351–e354, 2020.
- [8] P. Janićić, M. Nikolić, *Veštačka inteligencija*, Matematički fakultet Univerziteta u Beogradu, 2021.
- [9] M. Nikolić, A. Zečević, *Mašinsko učenje*, Matematički fakultet Univerziteta u Beogradu, 2019.
- [10] R.-S. Jang, "ANFIS: Adaptive Network based Fuzzy Inference Systems", *IEEE Trans. Syst., Man, Cybern.*, Vol. 23, No. 3, pp. 665–685, 1993.
- [11] R.-S. Jang, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Pearson, 1997.
- [12] S. Domínguez-Almendros, N. Benítez-Parejo, A. Gonzalez-Ramirez, "Logistic regression models", *Allergol Immunopathol*, 39(5):295–305, 2011.
- [13] D. Edwards, H. Hollingsworth, A. Zazulia, M. Diringer "Artificial neural networks improve the prediction of mortality in intracerebral hemorrhage". *Neurology* 53:351–357, 1999.
- [14] A.L. de Oliveira Manoel, "Surgery for spontaneous intracerebral hemorrhage". *Crit Care* 24, 45, 2020.

Projektovanje uređaja za snimanje i reprodukciju okretanja oko jedne ose

Mikica Radenković, Sava Stojanović

Sadržaj – U ovom radu je prikazan process projektovanja i realizacije uređaja za snimanje i reprodukciju okretanja oko jedne ose pomoću STM32 nucleo H755ZI-Q okruženja.

I. UVOD

U ovom radu je predstavljeno projektovanje u programu za simuliranje elektronskih kola *Proteus*, korišćenje razvojnog okruženja *CubeIDE* i praktično korišćenje razvojnog okruženja STM32 nucleo. Mikrokontroleri nude ogromne mogućnosti merenja različitih fizičkih veličina, obradu izmerenih podataka i kontrolu rada aktuatora. Merenje podataka zavisi od fizičke veličine i načina njenog merenja, obrada podataka od moći mikroprocesora i željene izlazne veličine, a rad aktuatora od promene u sistemu koji želimo da ostvarimo. Kretanje oko jedne ose predstavlja najprostiji vid kretanja kada je reč o rotacionom kretanju. Svakodnevno koristimo potencioetre za podešavanje zvuka, osvetljenja, brzine itd. pretvarajući naš fizički pokret u promenu intenziteta željene veličine. Ovde ćemo pokazati primer snimanja okretanja potenciometa od strane korisnika i reprodukciju tog okretanja na servo motoru.

II. PRINCIP FUNKCIONISANJA

Uređaj se sastoji od rotacionog potenciometa kog korisnik okreće 10 sekundi, a po isteku tog vremena servo motor reprodukuje korisnikovo okretanje.

Za realizaciju projekta korišćene su komponente:

- razvojno okruženje STM32 nucleo H755ZI-Q (slika 1)
- servo motor SpringRC SM-S2309S (slika 2)
- potencijometar
- le dioda
- taster
- otpornici

Uređaj vrši snimanje okretanja potenciometa aktivacijom snimanja pritiskom tastera, uključuje se signalna le dioda, koja ukazuje da je započeto snimanje, i korisnik može da okreće potencijometar čime se vrši promena analogne vrednosti napona. Potencijometar je

M. Radenković i S. Stojanović su studenti osnovnih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: radenkovicmikica99@gmail.com, ssava329@protonmail.com

povezan na pin mikrokontrolera i registruje promene napona koje pretvara u određenu celobrojnu vrednost i zatim pamti 1 vrednost u sekundi. Snimanje traje 10 s nakon čega se signalna dioda isključuje, a dalje okretanje potenciometa neće biti zabeleženo. Kada se završi proces snimanja mikrokontroler šalje odgovarajući signal servo motoru o položaju zupčanika u kom je potrebno da budu, servo motor okreće svoje zupčanike čime se reprodukuje snimljeno kretanje dokle god korisnik ne započne novo snimanje.



Slika 1. Razvojno okruženje STM32 nucleo H755ZI-Q.



Slika 2. Servo motor SpringRC SM-S2309S.

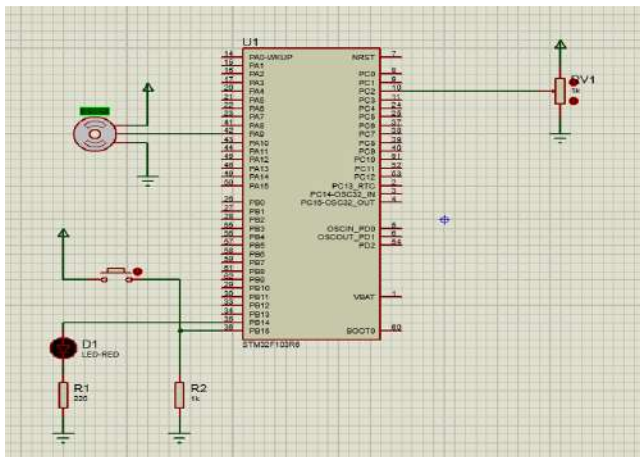
Razvojno okruženje STM32 nucleo H755ZI-Q [1] je bazirano na mikroprocesoru STM32H755ZIT6 [2] sa 144 pinova podeljenih u 8 serija (a-g) za povezivanje sa uređajima i jednom serijom za napajanje, dvojezgarim Arm Cortex M7 i Arm Cortex M4 procesoru koji rade na maksimalnim frekvencijama 480 MHz i 240 MHz respektivno. Posедуje 2 MB fleš memorije i 1 MB SRAM-a, internet konektor brzine 10/100 Mbps, integrisani naponski izvor (Internal SMPS), USB OTG FS konektor i integrisani ST-LINK/V3 debugger/programmer.

Servo motor se koristi kada je potrebno upravljati ugaonom pozicijom. Sastoji se od odgovarajućeg motora u kombinaciji sa senzorom koji daje informaciju o poziciji. Sa

mikrokontrolerom se povezuje sa 3 priključka, jednim upravljačkim i dva naponska. Upravljački signal se koristi za upravljanje ugaonom pozicijom izlazne osovine i najčešće je to PWM (Pulse width modification) signal. SpringRC SM-S2309S [3] može da rotira od 0-180°. Kontrolise se PWM signalom periode 900 μs (4,5%) - 2100 μs (10,5%) u zavisnosti od ugla. Potreban mu je napon napajanja od 5V i troši oko 300mA struje prilikom rotacije.

III. PROJEKTOVANJE UREĐAJA U PROGRAMU PROTEUS

U programu za simulaciju elektronskih kola "Proteus" prvo je napravljena električna šema uređaja (slika 3). Zbog nepostojanja modela našeg mikoro-procesora korišćen je model mikroprocesora STM32F103R6 sa 64 pina pa je servo motor SpringRC SM-S2309S povezan na pin PA9, a u praktičnoj realizaciji korišćen je PE9. Na pin PB15 je povezan taster za pokretanje snimanja u pull-down konfiguraciji. Na pin PB14 je povezana signalna le dioda sa otpornikom. Srednji pin potencijometra, na kome se menja vrednost napona, je povezan na pin PC2.



Slika 3. Električna šema uređaja u programu „Proteus“.

IV. PISANJE KODA U PROGRAMU "CUBEIDE"

Program CubeIDE [4] je razvila kompanija STM kao razvojno okruženje za sopstvene mikroprocesore. Po izboru mikroprocesora ili mikrokontrolera korisnik vrši podešavanje frekvencija, deklaraciju ulazno-izlaznih pinova, definisanje potrebnih konvertora, sabirača, kontrolera... Zatim okruženje generiše inicijalni kod. Taj kod korisnik dalje oblikuje.

Sve frekvencije su inicijalno podešene na 64 MHz, i nije bilo potrebe menjati ih. Potrebni ulazno-izlazni pinovi deklarirani su kao aktivni, podešen je izbor jezgra koji ih koristi i na koji način i imenovani su. Za snimanje vrednosti sa potencijometra korišćena je analogno digitalna konverzija na kanalu IN0 konvertora ADC3 u osmobitnoj rezoluciji preko funkcije prekida (interrupt). Analogno digitalna

konverzija predstavlja pretvaranje analognog signala u digitalni. U ovom slučaju analognu vrednost napona sa potencijometra intenziteta 0-3,3V pretvaramo u digitalnu vrednost intenziteta 0-255. Kada u kodu pozovemo funkciju interapta očitace se trenutna vrednost napona. Upravljanje servo motorom vrši se tajmerom (brojačem). Kod mikrokontrolera tajmer služi za generisanje PWM signala. Tajmer 1 preko kanala 1 je podešen kao generator PWM signala. Duty cycle servo motora mora biti između 4.5% i 10.5%, pa je potrebno izvršiti preskalaranje vrednosti PWM signala koji se šalje servo motoru u odnosu na vrednost koja se dobija AD konverzijom. Željena vrednost se dobija po formuli:

$$duty cycle = \frac{(IN - IN_{min}) * (OUT_{max} - OUT_{min})}{(IN_{max} - IN_{min}) * OUT_{min}}$$

gde je IN vrednost AD konverzije, $IN_{min}=0$, $IN_{max}=255,0$, $OUT_{min}=45,0$ i $OUT_{max}=105,0$.

U kodu su deklarirane promenljive (slika 4), deklarirane vrednosti duty cycle-a (slika 5). Snimanje se pokreće pozivanjem funkcije prekida koji inicijalnu vrednost "flag" postavlja na vrednost 1 (slika 6). U glavnom kodu se vrši snimanje pamteći vrednosti u pomoćnu promenljivu "bafer". Kada se završi funkcija snimanja, računa se duty cycle na osnovu dobijenih vrednosti i šalje se odgovarajući signal na izlaz na kom je povezan servo motor (slika 7).

```
72 /* USER CODE BEGIN 0 */
73 uint8_t flag;
74 uint32_t i=0, k=0;
75 uint32_t vrednost[10];
76 char bafer[20];
77 /* USER CODE END 0 */
```

Slika 4. Deklarisane promenljivih.

```
/* USER CODE BEGIN 1 */
uint16_t dutyCycle=45;
/* USER CODE END 1 */
```

Slika 5. Deklarisane početne vrednosti duty cycle-a.

```
552 /* USER CODE BEGIN 4 */
553 // ubacujemo interrupt na taster koji uključuje snimanje
554 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
555 {
556     /* Prevent unused argument(s) compilation warning */
557     UNUSED(GPIO_Pin);
558
559     /* NOTE: This function should not be modified, when the callback is needed,
560        the HAL_GPIO_EXTI_Callback could be implemented in the user file
561     */
562     flag=1;
563 }
564 /* USER CODE END 4 */
```

Slika 6. Pokretanje prekida i početak snimanje.

V. PRAKTIČNA REALIZACIJA

Praktična realizacija projekta je povezana na protoploči (slika 8). Povezivanje je izvršeno na osnovu inicijalno zamišljene električne šeme uređaja. Okruženje se napaja

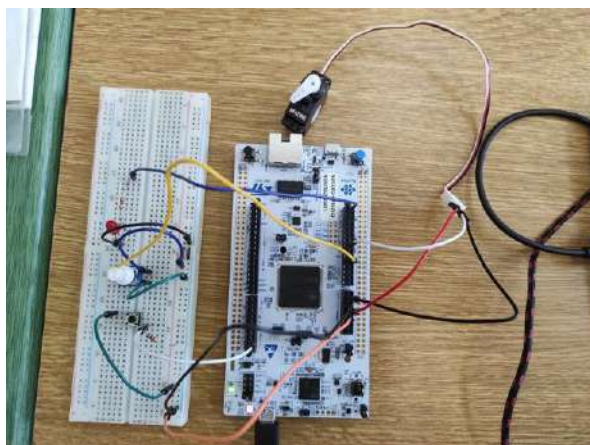
USB kablom iz lap-topa preko kog se i vrši komunikacija sa lap-topom. Otpornici su snage 0,25W.

```

156 /* USER CODE BEGIN 3 */
157
158 if (flag==1)
159 {
160     HAL_GPIO_WritePin(lcd_GPIO_Port, led_Pin, GPIO_PIN_SET); // uključuje diodu
161     for(i=0; i<10; i++)
162     {
163         vrednost[i]=HAL_ADC_GetValue(&hadc3); // namti ih vrednosti nize
164         sprintf(buf, "%d \r\n", vrednost[i]);
165         HAL_UART_Transmit(&huart2, (uint8_t*)buf, strlen(buf), 10);
166         HAL_UART_Transmit(&huart2, (uint8_t*)buf, strlen(buf), 10);
167         HAL_Delay(1000);
168     }
169     HAL_GPIO_WritePin(lcd_GPIO_Port, led_Pin, GPIO_PIN_RESET); // isključuje diodu
170     i=0;
171     flag=0;
172 }
173
174 for(k=0; k<10; k++)
175 {
176     dutyCycle=((vrednost[k]*(105.0-45.0))/(255.0))+45.0;
177     htim.Instance->CCR1=dutyCycle;
178     HAL_Delay(1000);
179 }
180 k=0;
181
182 }
183 /* USER CODE END 3 */

```

Slika 7. Glavni kod.



Slika 8. Praktična realizacija uređaja.

VI. ZAKLJUČAK

Ovaj rad predstavlja primer upoznavanja sa razvojnim okruženjem STM32 Nucleo i njegovim mogućnostima. Projekat je realizovan u okviru predmeta Komponente i kola snage na osnovnim studijama na modulu Elektronske komponente i mikrosistemi na Elektronskom fakultetu.

ZAHVALNICA

Zahvaljujemo se profesorima sa Katedre za mikroelektroniku na pomoći koju su nam pružili tokom izrade projekta, posebno asistentu Milošu Marjanoviću.

LITERATURA

- [1] <https://www.st.com/en/evaluation-tools/nucleo-h755zi-q.html>
- [2] <https://www.st.com/en/microcontrollers-microprocessors/stm32h755zi.html>
- [3] https://www.rhydolabz.com/robotics-servo-motors-c-155_157/sms2309s-micro-servo-11-kgcm-p-1315.html
- [4] <https://www.st.com/en/development-tools/stm32cubeide.html>

Praćenje rada i zaštita računarskih mreža pfSense sistemom

Svetlana Mančić, Nađa Gavrilović

Sadržaj – Maliciozne aktivnosti na mreži su problem koji se danas nameće svim korisnicima Interneta. Napretkom alata i tehnika za sajber napade, oni postaju sve sofisticiraniji, što njihovu detekciju čini sve težim zadatkom. U današnje vreme, jedan od primarnih zahteva korisnika mreže jeste sigurna mrežna infrastruktura koja će omogućiti siguran prenos i skladištenje podataka. Cilj ovog rada je projektovanje i implementacija sistema za zaštitu dela računarske mreže Elektronskog fakulteta u Nišu, baziranom na pfSense sistemu instaliranim na APU4 uređaju. PfSense sistem konfigurisan je tako da vrši ulogu *firewall* uređaja i loguje sve informacije o blokiranim paketima. Na postavljenom pfSense sistemu instaliran je i Snort IDS paket za detekciju napada na mreži. Dodatno, podaci koje generišu pfSense i Snort IDS prikupljeni su, vizuelizovani i analizirani upotrebom ELK (*ElasticSearch-Logstash-Kibana*) sistema. U radu je detaljno opisana implementacija predloženog sistema za zaštitu mreže, kao i proces prikupljanja, skladištenja i analize podataka. Takođe, dat je vizuelni prikaz analize podataka dobijenih u toku rada.

I. UVOD

Tehnološki napredak modernog doba omogućio je da se Internet koristi u veoma važnim oblastima, kao što su bankarstvo, zdravstvo, školstvo itd. Mogućnosti koje Internet pruža, neprestan porast broja korisnika, dostupnost i lakoća pristupa informacijama dovode u pitanje bezbednost korisnika. Maliciozne aktivnosti postaju sve sofisticiranije, dok tehnike njihovog prikrivanja postaju sve razvijenije. Bezbedna mrežna infrastruktura postala je primarni zahtev korisnika mreže, kako bi prenos i skladištenje njihovih podataka bili zaštićeni [1].

Veliki broj korisnika mreže nije dovoljno informisan o bezbednosti na Internetu, samim tim se i ne štiti adekvatno od različitih pretnji. Posledica su sve češći napadi na korisnike i njihove poverljive podatke različitim tehnikama socijalnog inženjerstva (eng. *Social Engineering*) [2].

U cilju zaštite korisnika mreže i same mrežne infrastrukture, jedne od najvažnijih komponenti sigurnosnih sistema su *firewall* uređaji i sistemi za detekciju napada (IDS). *Firewall* predstavlja sistem za zaštitu mreže, koji prati aktivnosti na računarskoj mreži i kontroliše dolazni i odlazni saobraćaj upotrebom definisanih bezbednosnih

pravila [3]. Kako bi zaštita bila potpuna, IDS sistemi koriste se u cilju detekcije neautorizovanih pristupa mreži i generisanja upozorenja. Jedan od najčešće korišćenih sistema za detekciju napada je Snort IDS [1].

Cilj ovog rada je projektovanje i implementacija sistema za zaštitu dela računarske mreže Elektronskog fakulteta u Nišu pfSense sistemom, instaliranim na APU4 uređaju. PfSense postavljen je tako da vrši *firewall* ulogu i loguje sve informacije o blokiranim paketima. Dodatno, u cilju proširenja funkcionalnosti pfSense sistema, na njemu je instaliran i Snort paket za detekciju mrežnih napada. Svi podaci koje pfSense i Snort generišu, šalju se do ELK (*ElasticSearch-Logstash-Kibana*) sistema, gde se skladište, analiziraju i vizuelno prikazuju. U radu će biti opisana implementacija predloženog sistema za zaštitu mreže, kao i proces prikupljanja podataka koje sistem generiše, njihovog skladištenja i krajnje vizuelizacije. Takođe, biće dat vizuelni prikaz analize prikupljenih mrežnih podataka, koji olakšava detekciju malicioznih aktivnosti.

Nakon uvoda, u drugom poglavlju biće data teoretska osnova neophodna za implementaciju predloženog sistema. Biće diskutovan princip rada pfSense sistema i Snort IDS paketa, kao i Logstash alata koji je korišćen za prikupljanje podataka sa pfSense sistema u cilju dalje analize. U trećem poglavlju biće opisan implementirani sistem. U četvrtom poglavlju biće dat prikaz rada predloženog sistema. Zaključak će biti dat u petom poglavlju.

II. TEORETSKA OSNOVA

A. PfSense

Projekat pfSense je besplatna *firewall* distribucija koja se bazira na FreeBSD operativnom sistemu. Pored toga što je moćna i prilagodljiva *firewall* platforma, pfSense ima mogućnost rutiranja. Dodatno, uključuje veliki broj softverskih paketa koji proširuju osnovne funkcionalnosti pfSense sistema. Osim podešavanja iz komandne linije, sistemom se može upravljati i u okviru dostupnog veb interfejsa [4].

Za razliku od većine komercijalnih *firewall* sistema, PfSense je softver koji je moguće instalirati na proizvoljnom hardveru, koji ispunjava minimalne zahteve i pogodan je za potrebe različitih okruženja [4]. Postoji nekoliko načina implementacije pfSense sistema – u ulozi *firewall* uređaja, rutera, sviča i *wireless* rutera [5].

U SOHO (*Small Office Home Office*) okruženju, *firewall* funkciju i rutiranje često obavlja isti uređaj. Kod

S. Mančić je student osnovnih akademskih studija na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: svetlanamancic@elfak.rs

N. Gavrilović je student doktorskih akademskih studija na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: nadja.gavrilovic@elfak.rs

mreža sa više mrežnih segmenata, pfSense se može koristiti za njihovo povezivanje i rutiranje između njih [5].

Kada bi nad nekoliko interfejsa bio kreiran *bridge* kako bi se interfejsi povezali (bez podele *broadcast* domena) pfSense bi mogao obavljati funkciju sviča. Međutim, ovaj način implementacije se retko koristi, jer su podešavanja standardnog sviča jednostavnije i daju bolje performanse jer se ne vrši filtriranje paketa [5].

Moderne mreže uključuju i bežično povezivanje. Korišćenjem pfSense sistema moguće je povezati bežične uređaje uz dodavanje *wireless* mrežne kartice, koju podržava FreeBSD. Međutim, bolja opcija je upotreba standardnog *wireless* rutera, podešavanje pristupne tačke, a zatim povezivanje rutera na LAN port pfSense sistema [5].

Najčešći način implementacije pfSense sistema je *firewall*. U tom slučaju, pfSense se postavlja tako da predstavlja vezu lokalne mreže i Interneta. Port koji je povezan na Internet je WAN interfejs, a lokalna mreža je na LAN interfejsu. Ukoliko se koristi ovaj način implementacije, osim *firewall* funkcije, pfSense može vršiti i rutiranje. Broj interfejsa nije ograničen na dva, već može biti i više interfejsa na kojima su lokalne mreže [5].

Osnovni zadatak pfSense *firewall* uređaja je uspostavljanje barijere između pouzdane lokalne mreže i nepouzdanе spoljne mreže, odnosno Interneta. Svaki *firewall* ima sposobnost filtriranja paketa, odnosno ispitivanja ulaznog i izlaznog saobraćaja i donošenja odluke o prosleđivanju ili odbacivanju ispitivanog paketa [5].

Ingress filtering se odnosi na filtriranje paketa koji dolaze na lokalnu mrežu sa spoljne mreže. *Egress filtering* se odnosi na filtriranje paketa koji se sa lokalne mreže šalju na Internet ili drugi interfejs [4].

Firewall pravila imaju tri opcije po kojima mogu postupiti [5]:

- *Pass* - propušta pakete kroz *firewall*,
- *Block* - odbacuje pakete bez obaveštavanja pošiljaoca,
- *Reject* - odbacuje pakete i šalje *port unreachable* poruku pošiljaocu na osnovu čega on može da zaključiti da je *firewall* odbacio paket.

U okviru liste pravila definisanih na jednom interfejsu, pravilo koje će se primeniti nalazi se *first match* politikom, tj. pravila se čitaju s vrha liste ka dnu i prvo pravilo sa kojim se nađe poklapanje se primenjuje [4].

B. Snort IDS

Snort je jedan od najčešće korišćenih sistema za detekciju napada na mrežu. Zasniva se na potpisima, koji definišu šta se može smatrati malicioznom aktivnošću na mreži i tako omogućavaju detekciju napada [6]. Setovi potpisa predstavljaju Snort pravila, koja se formalno definišu na sledeći način [7]:

```
<akcija><protokol><izvorna_IP_adresa><izvorni_port>  
<smernost><odredišna_adresa><odredišni_port><opcije>
```

Pravilo može sadržati jednu ili više opcija, od kojih se svaka sastoji od ključne reči koja definiše opciju, i argumenata koji navode detalje opcije [7]. Ako se polja iz paketa podudaraju sa nekim pravilom, Snort će generisati upozorenje, koje se prikazuje u realnom vremenu [8]. Snort je dostupan kao paket koji je moguće instalirati i podešavati kao dodatak funkcionalnosti pfSense sistema.

C. Logstash

ELK sistem predstavlja kolekciju tri javno dostupna alata koji se koriste u cilju prikupljanja, skladištenja i vizuelizacije podataka različitog porekla i strukture. Logstash komponenta služi za sakupljanje i transformaciju podataka, koji se šalju do Elasticsearch komponente. Ona sadrži indekse za skladištenje podataka definisane strukture, dok se Kibana alat koristi za krajnju vizuelizaciju prikupljenih podataka.

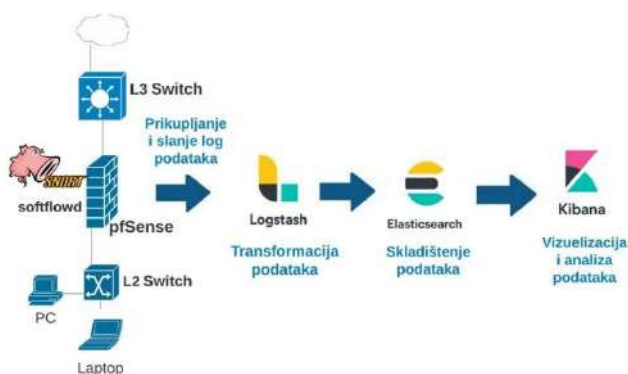
Logstash je ključna komponenta ELK sistema koja služi za dinamičko prihvatanje podataka sa više izvora, njihovu transformaciju i slanje bez obzira na format i složenost. Podaci su pre prikupljanja često skladišteni na različitim sistemima, u različitim formatima. Svaki tip podatka se može transformisati korišćenjem širokog spektra input, filter i output logstash *plugin*-ova [9]. Logstash ima dva neophodna elementa, input i output, i jedan opciono element, filter. Input element prihvata podatke sa različitih izvora, filter ih može modifikovati i proširiti na način koji se definiše u konfiguraciji, a output element šalje podatke odredištu. Dodatno, input i output podržavaju *codec* elemente, koji samostalno omogućavaju kodiranje i dekodiranje podataka bez potrebe za definisanjem posebnog filtera [9]. Najčešće korišćeni tipovi inputa su: *file*, *syslog*, *redis*, *beats* i *udp*. U ovom radu korišćeni su *udp* i *syslog*, koji funkcionišu tako što čitaju podatke dobijene preko mreže. U konfiguraciji, potrebno je definisati port na kom Logstash prima podatke, pri čemu je moguće definisati više različitih portova, ako postoji više izvora [9]. Filter komponenta vrši procesiranje i transformaciju podataka. U slučaju da se navede više njih, primenjuju se u redosledu navođenja. Dostupan je veliki broj filtera, od kojih su korišćeni *geoip*, *grok* i *mutate*. *Geoip* filter dodaje informacije vezane za geografsko lociranje IP adresa. *Grok* filter transformiše nestruktuirane podatke u struktuirane i čini ih pogodnim za primenu Elasticsearch upita. *Mutate* filter omogućava da se vrše izmene atributa podataka kao što su preimenovanje, brisanje i zamena [9]. Output tipovi koji se najčešće koriste su *elasticsearch*, *file*, *graphite* i *statsd*. U radu je korišćen *elasticsearch* output, koji upisuje obrađene podatke u Elasticsearch indekse, u kojima se podaci skladište i dostupni su za dalju analizu i vizuelizaciju [9].

III. PREGLED IMPLEMENTIRANOG SISTEMA

Sistem predložen u ovom radu sastoji se od pfSense *firewall* sistema instaliranog na APU4 hardveru, sa dodatno

instaliranim Snort IDS sistemom. APU4 ima 30GB SSD i 4GB RAM memorije, poseduje jedan serijski, četiri gigabitna i dva USB porta. Kako bi NetFlow funkcionalnosti bile dostupne na pfSense sistemu, neophodna je instalacija i *softflowd* paketa, koji loguje sav mrežni saobraćaj koji prođe kroz instalirani sistem.

Prikaz arhitekture predloženog sistema dat je na slici 1. U cilju analize mrežnog saobraćaja, pfSense podatke iz svojih *firewall* i Snort logova šalje na ELK sistem. Logstash transformiše podatke i šalje ih ElasticSearch komponenti, koja se sastoji od 5 mašina ukupnog kapaciteta 320GB, gde se podaci skladište. Kibana, poslednja komponenta ELK kolekcije, vizuelizuje podatke iz ElasticSearch indeksa. Na isti način kao i pfSense, paket *softflowd* šalje svoje dodatne logove na ELK sistem, u cilju dalje analize. U okviru rada, konfigurisana je Logstash komponenta, dok su ElasticSearch klaster i Kibana prethodno bili postavljeni i konfigurisani.



Slika 1. Prikaz arhitekture sistema.

Povezivanje je izvršeno tako da se jedan od gigabitnih portova APU4 uređaja koristi za menadžment pristup pfSensu radi podešavanja i interfejs kome je on dodeljen i nazvan je LAN_ZA_MENADZMENT. Dva gigabitna porta koriste se za veze sa L2 i L3 svičevima i njihovi interfejsi nazvani su LAN_KA_RACUNARIMA i LAN_KA_SVICU, respektivno.

Na interfejsu LAN_ZA_MENADZMENT mrežna podešavanja (IP adresa, maska i *gateway*) su statički podešena. Na portovima koji se koriste za veze sa svičevima kreirani su svi potrebni VLAN-ovi koji postoje na delu fakultetske mreže koja se štiti. Kako bi pfSense mogao biti postavljen između L2 i L3 sviča (kao na slici 1), bez podele broadcast domena, neophodno je povezati interfejse LAN_KA_RACUNARIMA i LAN_KA_SVICU kreiranjem *bridge*-a. Potrebno je kreirati novi interfejs i njemu dodeliti *bridge*, kako bi filtriranje paketa moglo da se vrši između L2 i L3 sviča.

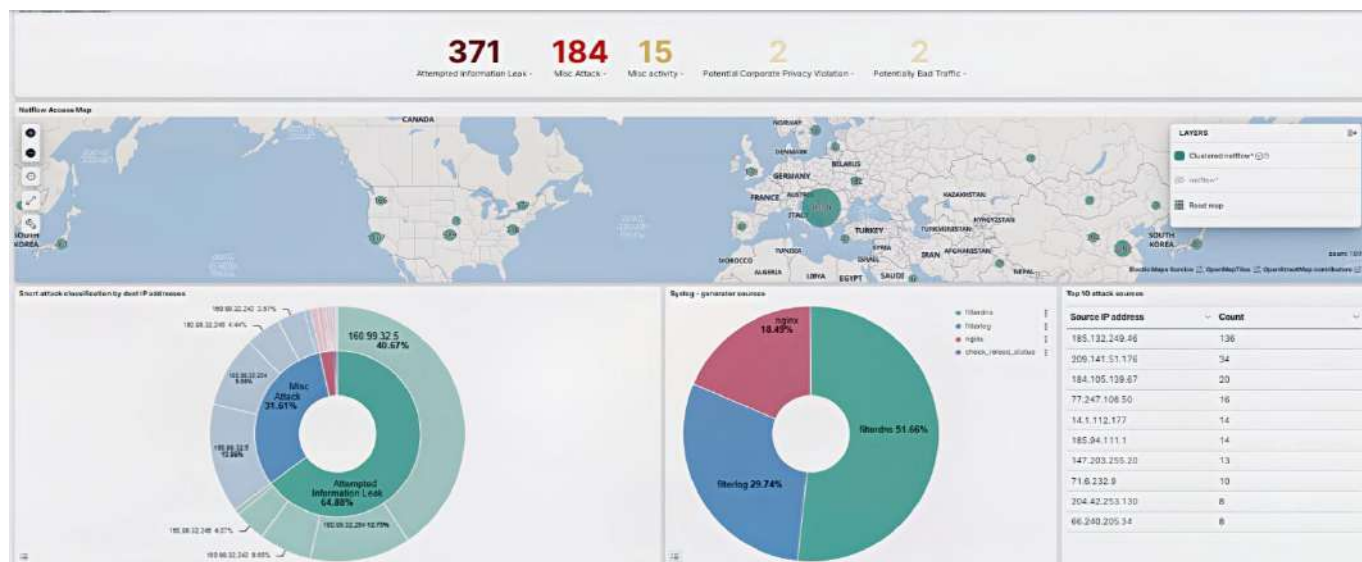
Tokom instalacije sistema, kreirana su *firewall* pravila za filtriranje saobraćaja i instalirani su paketi Snort i *softflowd*. U okviru podešavanja vezanih za Snort, izvršeno je preuzimanje javno dostupne baze Snort pravila. Bitno je podesiti koja se mreža štiti Snort IDS sistemom konfiguracijom HOME_NET opcije na interfejsima. HOME_NET se podešava tako što se kreira alijas koji obuhvata deo fakultetske mreže koji se štiti predloženim sistemom. EXTERNAL_NET opcija se konfiguriše tako da obuhvata sve mreže koje nisu uključene u HOME_NET.

Zatim, Snort IDS je podešen da detekciju napada vrši na *bridge* interfejsu. Na interfejsu se primenjuje celokupna baza preuzetih pravila. Snort je konfigurisan u modu detekcije, kako bi slao upozorenja o potencijalnom napadu, ali bez blokiranja izvorišta ili odredišta. Snort upozorenja se upisuju u pfSense log tako da se zajedno sa postojećim pfSense *firewall* logovima šalju Logstash-u.

Logstash prikuplja podatke sa dva izvora preko kanala na različitim UDP portovima. Jedan izvor čine podaci sa pfSense-a (*firewall* i Snort log), a drugi *softflowd* log podaci o celokupnom saobraćaju koji prolazi kroz pfSense. Pojedinačni izvori podataka definisani su posebnim konfiguracionim fajlovima.

Kod input dela pfSense konfiguracionog fajla koristi se postojeći *syslog plugin*. Filter deo konfiguracije ima više uslova. U zavisnosti od tipa log podatka (pošto su formati poruka različiti) primenjuje se odgovarajući *grok* šablon na osnovu kog se od nestruktuiranih podataka dobijaju značajni atributi i njihove vrednosti. U output delu se vrši slanje prikupljenih podataka ka ElasticSearch komponenti. Podaci dobijeni od Snort-a upisuju se u jedan, a od pfSense-a u drugi ElasticSearch indeks. Značajni podaci koje daje Snort su: izvorna i odredišna IP adresa, izvorni i odredišni port, klasa napada, prioritet, akcija, protokol itd. PfSense *firewall* sistem generiše podatke o blokiranim paketima, i to u formi interfejsa, preduzete akcije, izvorne i odredišne IP adrese i porta, itd.

U *softflowd* konfiguracionom fajlu input deo sadrži *udp plugin* za koji je definisan *codec netflow*, koji vrši dekodiranje dobijenih podataka u NetFlow formatu. U filter delu se za *source_ip* i *destination_ip* koristi *geoip* filter za dobijanje koordinata u cilju analize mrežnog saobraćaja iz ugla geografskog položaja. Output upisuje podatke u treći, zasebni netflow ElasticSearch indeks. Značajni podaci koji se dobijaju od *softflowd* paketa su izvorišna i odredišna IP adresa, izvorišni i odredišni port, podaci o geografskom položaju (koordinate, grad, država i sl.), količina prenetih podataka.



Slika 2. Vizuelni prikaz prikupljenih podataka.

IV. PRIKAZ DOBIJENIH PODATAKA

Za vizuelizaciju i analizu podataka dobijenih od pfSense-a, Snort IDS sistema i *softflowd* paketa korišćen je Kibana alat. Kreirani su dijagrami koji omogućavaju praćenje značajnih parametara sistema. Na slici 2 data je komandna tabla (eng. *dashboard*) koja daje vizuelni prikaz prikupljenih podataka u toku poslednjih 7 dana.

Dijagram na vrhu slike brojčano pokazuje različite vrste napada u toku 7 dana, koje je detektovao Snort IDS sistem. Na mapi sveta su obeležene lokacije izvornih IP adresa svih paketa koji prolaze kroz pfSense, detektovanih od strane *softflowd* paketa. Prvi s leva dijagram ispod mape pokazuje različite vrste napada na mrežu u procentima, koje je zabeležio Snort IDS, i za svaku vrstu napada određene adrese ka kojima je upućen maliciozni saobraćaj. Dijagram u sredini procentualno prikazuje udeo svakog tipa log podatka u sa pfSense sistema. Tabela u donjem desnom uglu sadrži deset IP adresa sa kojih je najviše paketa detektovao Snort IDS. Na taj način su zabeleženi najčešći izvori malicioznog saobraćaja ka zaštićenoj mreži.

Implementirani sistem omogućava prikupljanje podataka sa više različitih izvora, njihovo struktuiranje i proširenje podacima o geografskom položaju. Takođe, sistem omogućava rad sa velikom količinom log podataka, jer koristi Elasticsearch skladište velikog kapaciteta. Konačno, vizuelizacija podataka omogućava njihovu jednostavnu analizu, kao i vizuelnu korelaciju podataka iste mreže, ali sa različitih izvora. Na taj način, implementirani sistem pruža mogućnost jednostavne detekcije napada u realnom vremenu.

V. ZAKLJUČAK

U ovom radu implementiran je sistem za zaštitu dela računarske mreže Elektronskog fakulteta u Nišu pfSense sistemom instaliranim na APU4 hardveru. PfSense ima *firewall* ulogu i loguje sve informacije o blokiranim paketima, dok dodatno instalirani Snort IDS služi za detekciju mrežnih napada. Podaci koje generiše predloženi sistem prikupljeni su, analizirani i vizuelizovani upotrebom ELK sistema. U radu su date arhitektura i implementacija predloženog sistema, kao i proces prikupljanja i analize mrežnih podataka. Takođe, dat je i vizuelni prikaz analize prikupljenih podataka.

ZAHVALNICA

Autori se zahvaljuju prof. dr Vladimiru Čiriću na uloženom trudu, savetima, motivaciji i podršci.

LITERATURA

- [1] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges", *Cybersecurity*, Vol. 2, 2019.
- [2] J. M. Kizza, *Guide to Computer Network Security*, 4th. ed. Springer, Publishing Company, Incorporated, 2017.
- [3] X. He, "Research on Computer Network Security Based on Firewall Technology", IOP Publishing, 2020.
- [4] C. Buechler, J. Pingle, "pfSense: The Definitive Guide", 2009.
- [5] D. Zientara, "Mastering pfSense", Packt Publishing, 2016.
- [6] S. Dwiyoatno, W. A. Andriani, A. P. Sari, Sulistiyono, "Implementation of Snort IPS Using pfSense as Network Forensic in Smk XYZ", Atlantis Press SARL, 2020.
- [7] W. Stallings, L. Brown, "Computer security – Principles and Practice", Pearson, 2008.
- [8] B. Caswell, J. Twycross, T. Hesketh-Roberts, "Snort – IDS and IPS Toolkit", Syngress Publishing, 2007.
- [9] S. Chhajer, "Learning ELK Stack", Packt Publishing, 2015.

ARM mikrokontrolerski sistem regulacije brzine DC motora zavisne od temperature okoline

Nemanja Tomić, Lazar Stanojević

Sadržaj – U ovom radu predstavljen je prototip sistema za regulaciju brzine DC motora u zavisnosti od temperature okoline. Sistemom upravlja ARM mikrokontroler na Nucleo razvojnom okruženju. Ako je temperatura ispod 25°C, sistem je neaktivan. Na temperaturama između 25°C i 35°C motor se okreće sa 30% brzine. Iznad 35°C motor se okreće srazmerno temperaturi. Maksimalnu brzinu ostvaruje ukoliko je temperatura 65°C. Iznad ove temperature izbacuje se upozorenje korisniku. Motor radi sa 100% brzine dok se ne dostigne 75°C i tada se isključuje.

I. NUCLEO-F103RB

Za realizaciju prototipa korišćeno je *STM 32 Nucleo 64* razvojno okruženje kompanije *ST microelectronics* sa mikrokontrolerom *STM32F103RB* [1]. Mikrokontroler poseduje 32-bitni *ARM cortex M3* procesor. Može da radi na frekvenciji do maksimalno 72 MHz. Ima 128 KB *flash* memorije i 20 KB *SRAM*-a. Ima 51 ulazno-izlazni pin. Može istovremeno da radi više analogno-digitalnih konverzije sa 12-bitnom rezolucijom. Posедуje *RTC* (*real time clock*), 4 tajmera, po 2 perifernjske jedinice za *I²C* i *SPI* komunikaciju i 3 perifernjske jedinice za *UART* (*universal asynchronous receiver-transmitter*). Prikaz razvojnog okruženja dat je na slici 1.



Slika 1. Nucleo-F302R8.

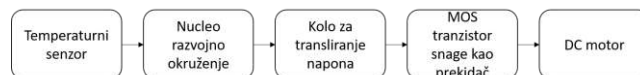
II. HARDVERSKJE KOMPONENTE SISTEMA

Pored Nucleo razvojnog okruženja, za realizaciju prototipa sistema za regulaciju brzine motora korišćeni su

N. Tomić je diplomirani inženjer elektrotehnike i računarstva, modul Elektronske komponente i mikrosistemi na Elektronskom fakultetu u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: n.tomic@elfak.rs

L. Stanojević je student osnovnih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: lazar.st@elfak.rs

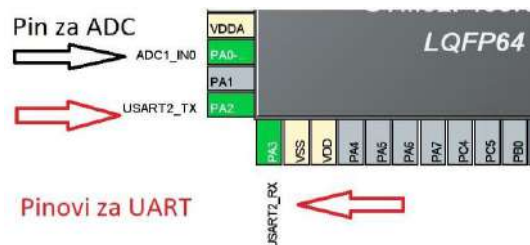
temperaturni senzor TMP36 [2], TXS0108E kolo za transliranje napona sa 3.3V na 5V [3], MOS tranzistor snage IRF510 [4] koji radi kao prekidač i DC motor. Blok šema realizovanog prototipa data je na slici 2. Kolo za transliranje napona bilo je neophodno jer mikrokontroler daje na izlaznom pinu napon logičke jedinice od 3.3V, što nije dovoljno za uključivanje MOS tranzistora snage. Treba napomenuti da je Nucleo razvojno okruženje povezano sa računarom koji preko serijskog terminala prima određene podatke, tj. šalju se podaci upozorenja kada motor radi na temperaturi od 65 do 75 °C, a na temperaturi iznad 75 °C i ispod 25 °C se šalje podatak da je motor isključen. Prilikom testiranja prototipa korišćen je laboratorijski izvor DC napona i osciloskop.



Slika 2. Blok šema realizovanog prototipa.

III. PODEŠAVANJE PERIFERIJA MIKROKONTROLERA KORIŠĆENJEM PROGRAMA CUBE MX

Cube MX [5] služi za podešavanja periferija mikrokontrolera i automatsko generisanje koda za inicijalizaciju iskorišćenih periferija. Potrebno je bilo da se postave parametri za *UART* serijsku komunikaciju, podesi tajmer i *AD* konvertor (*ADC*) mikrokontrolera. Pinovi za *UART* komunikaciju su izvedeni na razvojnom okruženju za direktno povezivanje sa računarom preko *USB*-a. Ukoliko se odabere, *UART1* ili *UART3* potrebno je naknadno povezivanje preko dodatnog *USB* konektora, zato je odabran *UART2* za direktno povezivanje okruženja sa računarom. Na slici 3 prikazani su Rx (*PA3*) i Tx (*PA2*) pinovi za *UART* komunikaciju, kao i *ADC* pin (*PA0*).



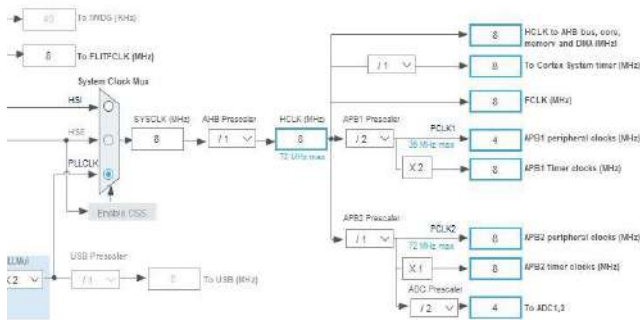
Slika 3. Postavljanje pinova za ADC i UART.

Podešavanje UART modula podrazumeva postavljanje asinhronog *conversion mode*-a tako da se podaci šalju serijski bez posredovanja taktnog signala. Postavljen je *baud rate* na 115200 bits/s (slika 4).



Slika 4. Cube MX podešavanja za UART.

Temperaturni senzor povezan je na pin PA0, tako da se vrednost temperature dobija kao rezultat AD konverzije. AD konverzija rađena je u *continuous mode*-u koji omogućava najbržu konverziju. Ukoliko se HCLK (radni takt mikrokontrolera) postavi na 8 MHz, *prescaler* (1/2) postavlja takt ADC konverzije na 4 MHz (slika 5).



Slika 5. Podešavanje takta AD konverzije.

Vreme potrebno da se završi jedna ADC konverzija u ciklusima iznosi:

$$T_{conv}(cycles) = Sampling\ time + 12.5 \quad (1)$$

Ako se za *sampling time* (vreme uzorkovanja) izabere 239.5 ciklusa (slika 6), na osnovu jednačine (1) se dobija vreme potrebno za jednu AD konverziju, izraženo u taktnim ciklusima. Vreme dobijeno u ovom slučaju, izraženo u taktnim ciklusima, je dato u jednačini (2):

$$T_{conv}(cycles) = 239.5 + 12.5 = 252 \quad (2)$$

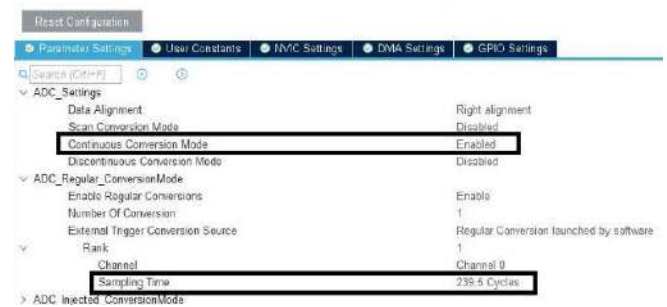
Pošto je frekvencija AD konverzije 4 MHz vreme jedne AD konverzije iznosi 63 μ s.

Da bi se impulsni signal doveo na DC motor potrebno je podešavanje tajmera kojim će se podešavati *duty cycle* impulsnog signala. DC motor će biti kontrolisan PWM (*pulse-width modulation*) signalom preko pina PA15 (slika 7). *Duty cycle* će zavisiti od izmerene temperature. Frekvencija tajmera se u kartici *clock configuration* podešava na $f_{TIM2} = 8$ MHz. Da bi se dobila frekvencija koja

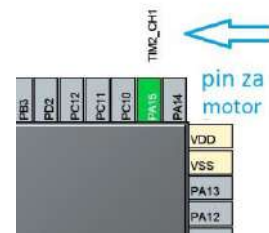
je odgovarajuća za skaliranje postavljamo *prescaler* na 7999 da bi dobili frekvenciju tajmera od $f_{TIM2(prescaler)} = 1$ KHz. CCRx (*Pulse*) predstavlja vrednost koju ima registar preko kojeg se podešava *duty cycle*, pa je pogodno da se *counter period* (*AutoReload* registar) podesi na 100 (slika 8), da bi vrednost CCRx bila vrednost *duty cycle*-a izražena u procentima. Računanje vrednosti za *duty cycle* je dato u jednačini (3):

$$duty cycle = CCRx / ARR_x = (Pulse) / (Counter Period) \times 100\% \quad (3)$$

Perioda signala za kontrolu DC motora je: *Counter Period* / $f_{TIM2(prescaler)} = 100$ ms. Nakon ovih podešavanja, Cube MX generiše osnovni kod sa inicijalizovanim periferijama: UART, AD konvertor, tajmer.



Slika 6. Podešavanje parametara AD konvertora.



Slika 7. Postavljanje pina za PWM kontrolu DC motora, korišćenjem tajmera mikrokontrolera.



Slika 8. Podešavanje parametara tajmera.

IV. SOFTVER MIKROKONTROLERA

Softver mikrokontrolera (*firmware*) pisan je na programskom jeziku C, korišćenjem Keil uVision5 okruženja [6]. Prvo je bilo potrebno da se uvede niz

promenljivih za različite namene, kao što je datu u narednom listingu:

```
/* USER CODE BEGIN 0 */
int senzorNapon, dutycycle;
float senzorVrednost, temperatura;
uint8_t poruka[20] = "Upozorenje\r\n";
uint8_t poruka1[25] = "Sistem je
isključen\r\n";
char txPodatak1[40];
/* USER CODE END 0 */
```

Koriste se promenljive `senzorNapon`, `dutycycle` da se u njima upisuju celobrojni tipovi podataka, a u `senzorVrednost`, `temperatura` se upisuju realni brojevi. Potrebna je inicijalizacija dve poruke, stringa koje se šalju računaru pri određenim uslovima.

Promenljiva `senzorVrednost` prvo dobija vrednost koja se čita sa temperaturnog senzora putem AD konverzije. Promenljiva `temperatura` čuva vrednost izmerene temperature. Funkcija `sprintf(txPodatak1, "T=%.2f\r\n", temperatura)` u string `txPodatak1` čuva vrednost dobijenu sa senzora. Ovaj podatak se *UART* serijskom komunikacijom šalje računaru koji preko serijskog terminal prikazuje ceo string. Listing koda početka glavnog programa dat je u nastavku:

```
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    senzorVrednost=HAL_ADC_GetValue(&hadc1);
    temperatura=senzorVrednost/4095.0*100.0;
    sprintf(txPodatak1, "T = %.2f \r\n", temperatura);
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
}
```

U nastavku se proveravaju uslovi, kao što je opisano u sadržaju rada. Prvi je da ukoliko je temperatura između 25°C i 35°C, DC motor se okreće sa 20% maksimalnog intenziteta i ispisuje se vrednost izmerene temperature. Ukoliko je zadovoljen uslov iz *if* narebe, komandom `htim2.Instance->CCR1= 20` se vrednost `CCR1` registra menja i iznosi 20. Menja se *duty cycle* signala za kontrolu DC motora. Listing koda dat je u nastavku:

```
if (temperatura>25.0 && temperatura<35.0)
{
    htim2.Instance->CCR1=20;
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
    sprintf(txPodatak1, "Temperatura = %.2f stepeni\r\n", temperatura);
}
```

Drugi uslov je da, ukoliko je temperatura između 35°C i 65°C, DC motor se okreće srazmerno vrednosti temperature. Ispisuje se izmerena vrednost preko *UART*-a i `CCR1` dobija vrednost promenljive *duty cycle*. Listing koda dat je u nastavku:

```
else if (temperatura>35.0 && temperatura<65.0)
{
    dutycycle=((temperatura-35.0)*2.66)+30;
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
    sprintf(txPodatak1, "Temperatura = %.2f stepeni\r\n", temperatura);
    htim2.Instance->CCR1=dutycycle;
}
```

U slučaju da je vrednost temperature između 65°C i 75°C, motor nastavlja da se okreće maksimalnom brzinom, šalje se poruka o izmerenoj temperature, ali i poruka upozorenja. Listing koda dat je u nastavku:

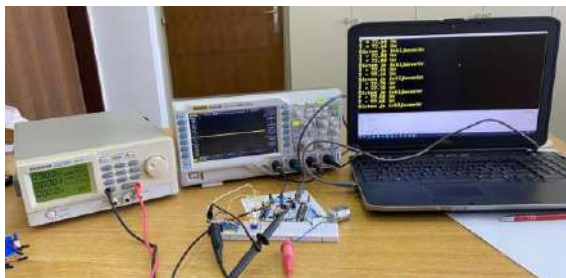
```
else if (temperatura>65.0 && temperatura<75.0)
{
    htim2.Instance->CCR1=100;
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
    sprintf(txPodatak1, "Temperatura = %.2f stepeni\r\n", temperatura);
    HAL_UART_Transmit(&huart2, poruka, strlen(poruka), 10);
    HAL_Delay(500);
}
```

U slučaju da je temperatura veća od 75°C motor se isključuje tako što se *duty cycle* stavlja na nulu. Preko *UART*-a se šalje poruka da je motor isključen. Zadnji slučaj posle koga se *while* zatvara (glavna programska petlja) je kada je vrednost temperature manja od 25°C. Takođe, kao i kada je vrednost veća od 75°C, motor se isključuje i šalje se odgovarajuća poruka na serijski terminal. Listing koda dat je u nastavku:

```
else if (temperatura>75.0)
{
    htim2.Instance->CCR1=0;
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
    sprintf(txPodatak1, "Temperatura = %.2f stepeni\r\n", temperatura);
    HAL_UART_Transmit(&huart2, poruka1, strlen(poruka1), 10);
    HAL_Delay(500);
}
else if (temperatura<25.0)
{
    htim2.Instance->CCR1=0;
    HAL_UART_Transmit(&huart2, (uint8_t *)txPodatak1, strlen(txPodatak1), 10);
    sprintf(txPodatak1, "Temperatura = %.2f stepeni\r\n", temperatura);
    HAL_UART_Transmit(&huart2, poruka1, strlen(poruka1), 10);
    HAL_Delay(500);
}
/* USER CODE END 3 */
```

V. PRAKTIČNA REALIZACIJA PROTOTIPA

Opisani prototip ARM mikrokontrolerskog sistema za kontrolu brzine DC motora u zavisnosti od izmerene temperature okoline realizovan je u laboratoriji (slika 9) i potvrđena je funkcionalnost sistema. U praktičnoj realizaciji i testiranju sistema korišćene su elektronske komponente nabrojane u poglavlju II ovog rada.



Slika 9. Testiranje prototipa opisanog sistema u laboratoriji.

VI. ZAKLJUČAK

U ovom radu predstavljen je prototip sistema za regulaciju brzine DC motora u zavisnosti od temperature okoline zasnovan na ARM mikrokontroleru. Za realizaciju prototipa korišćeno je Nucleo razvojno okruženje, temperaturni senzor, DC motor i kolo za drajvovanje motora zasnovano na MOS tranzistoru snage. Funkcionalnost sistema proverena je u laboratorijskim uslovima. Ovakav sistem se sreće svakodnevno od ventilatora do složenih industrijskih primena.

ZAHVALNICA

Ovaj rad predstavlja projektni zadatak iz predmeta Komponente i kola snage, na modulu Elektronske komponente i mikrosistemi u VIII semestru.

LITERATURA

- [1] STM32F103. Datasheet. Dostupno na: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>
- [2] TMP36. Datasheet. Dostupno na: https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf
- [3] TXS0108E. Datasheet. Dostupno na: https://www.ti.com/lit/ds/symlink/txs0108e.pdf?ts=1634205212896&ref_url=https%253A%252F%252Fwww.google.com%252F
- [4] IRF510. Datasheet. Dostupno na: <http://www.irf.com/product-info/datasheets/data/irf510.pdf>
- [5] Cube MX. Dostupno na: <https://www.st.com/en/development-tools/stm32cubemx.html>
- [6] Keil uVision5. Dostupno na: <https://www2.keil.com/mdk5>

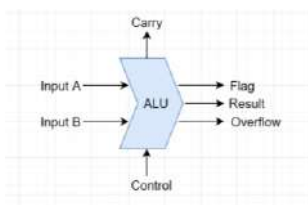
Praktična realizacija jednobitne ALU

Katarina Božinović

Sadržaj – U ovom radu je dat teorijski opis aritmetičko logičke jedinice, izvršene su simulacije, VHDL opis, projektovanje i praktična realizacija kola jednobitne aritmetičko logičke jedinice.

I. UVOD

Aritmetičko-logička jedinica, ALU (*Arithmetic Logic Unit*), je kompleksno digitalno kolo koje predstavlja sintezu aritmetičke i logičke jedinice. Više ALU-a nalazi se u računaru uključujući više njih u Centralnoj procesorskoj jedinici, CPU (*Central Processor Unit*), Grafičkoj procesnoj jedinici, GPU (*Graphics Processing Unit*) i Jedinici pokretnog zarez, FPU (*Floating point unit*) [1]. Aritmetičko logička jedinica je deo računara u kome se izvršavaju operacije nad podacima, zbog toga se često naziva i izvršnom jedinicom [1]. Operacije koje se izvršavaju mogu se podeliti u dve grupe: aritmetičke i logičke. U aritmetičke operacije uglavnom se ubrajaju: sabiranje i oduzimanje, a kod složenijih ALU jedinica mogu se sresti i množenje i deljenje. Logičke operacije uključuju sve operacije koje manipulišu logičkim vrednostima. Logičke vrednosti su: tačno ili netačno, što se podjednako predstavlja i kao 0 i 1 [2]. Logičke (Bulove) vrednosti dobile su ime po engleskom matematičaru Džordžu Bulu, koji je osmislio Bulovu algebru i smatra se utemeljivačem teorije računarstva [2]. Osnovni logički operatori su AND (I), OR (ILI) i NOT (NE). Sve Bulove funkcije mogu se izvesti iz ova tri osnovna operatora [2]. Kod tipične aritmetičko logičke jedinice sreću se neke od sledećih logičkih operacija: AND, OR, NOT, NAND, NOR, XOR, XNOR, komparacija, pomeranje bitova unutar podatka levo ili desno, itd. Na Slici 1 je prikazan blok dijagram tipične aritmetičko logičke jedinice. Ulazi se mogu podeliti u više grupa: postoje kontrolni ulazi, preko kojih se dovode signali od kojih zavisi način rada aritmetičko logičke jedinice i postoje ulazi za podatke, A i B.



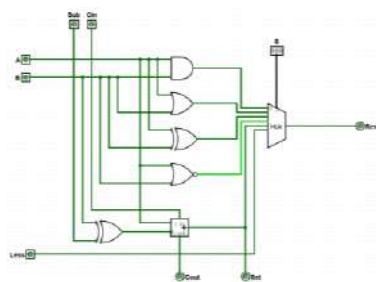
Slika 1. Blok dijagram aritmetičko logičke jedinice.

K. Božinović je student osnovnih akademskih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: katarina.n.bozinovic@gmail.com.

Takođe, postoji više vrsta izlaza: za rezultat (*Result*), za prenos (*Carry*), *Flag* ili *Overflow*. *Flag* predstavlja neko obaveštenje sistemu, dok *Overflow* izlaz upućuje na neki vid prekoračenja. Sve operacije obavljaju se nad podacima koji se dovode na ulaze A i B.

II. JEDNOBITNA ARITMETIČKO LOGIČKA JEDINICA

Jednobitna ALU sa Slike 2 može vršiti sledeće operacije: AND, OR, XOR, NOR, sabiranje sa prenosom i oduzimanje. Postoje četiri jednobitna ulaza: A, B, Sub, Cin, Less i jedan trobitni selekcionni ulaz S. A i B su ulazi za bitove nad kojima se vrše operacije. Sub određuje da li se vrši sabiranje (Sub = 0), ili oduzimanje (Sub = 1), dok je Cin ulazni bit prenosa. Cin obavezno mora biti 1 ukoliko se vrši oduzimanje.



Slika 2. Jednobitna aritmetičko logička jedinica.

Za svaku logičku operaciju (AND, OR, XOR, NOR) vrši se dovođenje bita sa ulaza A na prvi ulaz i bita sa ulaza B na drugi ulaz, dvoulaznog logičkog kola. Izlaz svakog logičkog kola vodi se na multiplekser 8/1. Od 8 ulaza MUX-a koriste se samo 6. Selektionim trobitnim ulazom S određuje se koji ulaz se propušta na izlaz, samim tim se kontroliše koju operaciju ALU obavlja, pa se ovaj ulaz može nazvati i kontrolni. Zapravo, ALU u svakom trenutku sa datim ulazima obavlja svaku moguću operaciju, ali se selektionim ulazom bira koji od rezultata obavljenih operacija će se proslediti na izlaz, tako da korisnik stiče utisak da izborom selektionog ulaza bira i način rada ALU.

Sabiranje i oduzimanje vrši se uz pomoć osnovnog sabirača koji je dostupan kao gotovo kolo u Logisim simulatoru. Sabirač koji je dostupan kao gotovo kolo ima ulaze: a, b i Cin, dok su izlazi: Cout, bit izlaznog prenosa i Set koji predstavlja rezultat, dok A i B predstavljaju ulazne podatke. Da bi se izvršilo sabiranje, Sub treba biti 0. Oduzimanje se vrši tako što se Sub i Cin postavljaju na vrednost logičke 1, na taj način se uz pomoć XOR kola vrši

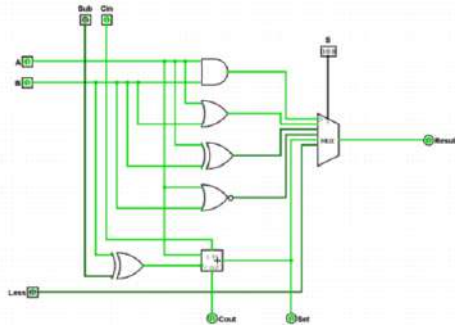
komplement drugog ulaznog bita. Tada operacije koje slede odgovaraju izrazima (1) i (2):

$$b = B \oplus 1 = \bar{B} \quad (1)$$

$$\text{Set} = A + \bar{B} + \text{Cin} = A + \bar{B} + 1 = A - B \quad (2)$$

Na Slici 3 prikazan je primer u kome je izlaz jednobitne ALU rezultat operacije AND. Ulazi A i B imaju vrednost logičke 1, tako da je rezultat operacije takođe 1.

Selekcionni ulaz S ima vrednost "000", što znači da se propušta upravo rezultat iz formule (3). Takođe, izlazi Set i Cout su jednaki logičkoj 1, zato što se sa datim ulazim obavljaju sve operacije, pa tako i sabiranje ($\text{Sub} = 0$).



Slika 3. Primer - Operacija AND.

U zavisnosti od vrednosti signala S propuštaju se sledeći rezultati:

- S = 000 - propušta se rezultat operacije AND,
- S = 001 - propušta se rezultat operacije OR,
- S = 010 - propušta se rezultat operacije XOR,
- S = 011 - propušta se rezultat operacije NOR,
- S = 100 - propušta se rezultat operacije sabiranja/oduzimanja,
- S = 101 - predviđeno je da se propusti operacija komparacije, koja se koristi tek prilikom kaskadnog vezivanja više jednobitnih ALU,
- S = 110 ili 111, izlaz je nedefinisan, jer su ulazi, čiji sadržaj bi se prenosio, neiskorišćeni.

III. VHDL OPIS ALU KOLA

Pre opisa glavnog kola, potrebno je prvobitno opisati kolo jednobitnog potpunog sabirača, a zatim hijerarhijski organozovati projekat. Na Slici 4 prikazan je opis kola sabirača. Kod se sastoji od entiteta i arhitekture. Entitet sadrži sve ulaze i izlaze komponente, dok arhitektura opisuje način rada komponente. Za potrebe opisa korišćeni su i pomoćni signali: m, n i l. Pomoćni signali čuvaju rezultate određenih operacija i olakšavaju konverziju iz tipa *std_logic_vector* u tip *unsigned*. Ova konverzija je neophodna zato što je nad tipom *std_logic_vector* moguće vršiti samo logičke operacije, dok je nad podacima tipa *unsigned* dozvoljeno vršiti samo aritmetičke operacije.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.NUMERIC_STD.ALL;
4
5 entity adder is
6     port(a,b:IN STD_LOGIC;
7         cin:IN STD_LOGIC_VECTOR(0 DOWNTO 0);
8         cout:OUT STD_LOGIC_VECTOR(0 DOWNTO 0);
9         set:OUT STD_LOGIC);
10 end adder;
11
12 architecture arch of adder is
13     signal m:std_logic_vector(0 downto 0);
14     signal n:std_logic_vector(0 downto 0);
15     signal l:unsigned(0 downto 0);
16 begin
17     set<= a XOR b XOR cin(0);
18     m(0) <= a AND b;
19     n(0) <= a XOR b;
20     l <= unsigned(m) + unsigned(n);
21     cout<= STD_LOGIC_VECTOR(l)AND cin(0);
22 end arch;
```

Slika 4. VHDL opis potpunog sabirača.

Formule (3) i (4) po kojima su definisani izlazi sabirača izvedene su iz tablice istinitosti (TABELA 1):

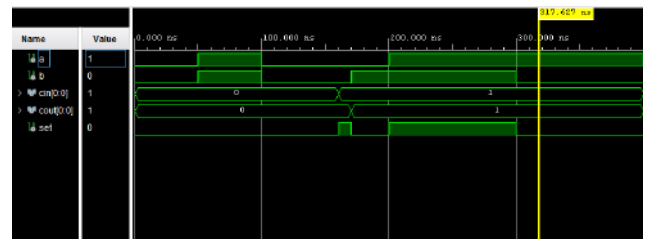
TABELA I
TABLICA ISTINITOSTI JEDNOBITNOG POTPUNOG SABIRAČA

A	B	Cin	Cout	Set
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A \oplus B \oplus \text{Cin} \quad (3)$$

$$\text{Cout} = (A \cdot B) + (\text{Cin} \cdot (A \oplus B)) \quad (4)$$

Vremenski dijagram izlaznih i ulaznih signala sabirača prikazan je na Slici 5. U trenutku označenom kursorom može se videti da ulazi a i b sabirača imaju vrednosti 1 i 0, respektivno, dok je Cin jednako logičkoj 1. Tada izlazi Cout i Set imaju vrednosti 1 i 0, respektivno.



Slika 5. Vremenski dijagram izlaznih i ulaznih signala sabirača.

Pored osnovnih delova koda, entiteta i arhitekture, u opisu jednobitne ALU korišćena je prethodno formirana komponenta potpunog sabirača (Slika 6). Pre instanciranja sabirača potrebno je u arhitekturi ključnom rečju *COMPONENT* deklarirati ovu komponentu, pod uslovom da je opis deklarisan komponente u istom projektu. Zatim se, preko labele u1, vrši instanciranje sabirača, tako što se uz pomoć naredbe *PORT MAP* ulazi i izlazi sabirača povezuju sa ulazima, izlazima ili pomoćnim signalima

glavnog kola. Pomoćni signali *pom1* i *pom2* neophodni su u ovom opisu. Njihova uloga je da sačuvaju rezultate određenih operacija, koji su potrebni za dalju analizu. Rezultati operacija u realizovanom hardveru predstavljaju izlaze iz logičkih kola koja izvršavaju te operacije. U VHDL-u jedan signal istovremeno ne može biti i ulaz i izlaz, zbog toga pomoćni signal čuva izlaznu vrednost i prosleđuje je odgovarajućem ulazu, to je suština njihove upotrebe.

Multiplexer je opisan uz pomoć naredbe *SELECT*. Ova naredba spada u konkurentne naredbe dodele sa izborom vrednosti [11]. Izračunata vrednost selekcionog izraza se poredi sa vrednostima iz svih grana, a signal dobija vrednost izraza iz grane gde se javilo slaganje [11]. Ukoliko se prosleđuje vrednost sa poslednja dva ulaza, tada *S* ima vrednost "110" ili "111", tada se na izlaz prosleđuje nepoznata vrednost 'X' (tip podataka u VHDL-u), što se poklapa sa simulacijom, s obzirom da ovi ulazi nisu povezani. Treba obratiti pažnju na ključnu reč *OTHERS* koja je neophodna ukoliko se radi sa tipom podataka *std_logic* ili *std_logic_vector*. Pored vrednosti logičke 0 (LOW) i logičke 1 (HIGH), u VHDL-u postoji još šest mogućih vrednosti koje jedan bit može da ima i potrebno je da se svaka od njih obuhvati kao mogućnost, kao i svaka moguća kombinacija tih vrednosti kod višebitnih signala, kao što je ovde selekcionog signala *S*. Dakle, sve moguće vrednosti selekcionog signala moraju biti pokrivena, što se upravo postiže ključnom reči *OTHERS* [3].

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity ALU is
5     port(a,b,sub,less:in std_logic;
6           s:in std_logic_vector(2 downto 0);
7           cin:in std_logic_vector(0 downto 0);
8           cout:out std_logic_vector(0 downto 0);
9           R, set:out std_logic);
10 end ALU;
11
12 architecture arch of ALU is
13     -- deklaracija potpunog sabirača koji ne koristi
14     component adder
15         port(a,b:in std_logic;
16               cin:in std_logic_vector(0 downto 0);
17               cout:out std_logic_vector(0 downto 0);
18               set:out std_logic);
19     end component;
20     signal pom1, pom2: std_logic;
21     begin
22         pom1 <= sub xor b;
23         u1: adder
24         port map (a=>a,b=>pom1,cin=>cin,cout=>cout,set=>pom2);
25         with s select
26             R <= A and B when "000",
27                 A or B when "001",
28                 A xor B when "010",
29                 A nor B when "011",
30                 pom2 when "100",
31                 less when "101",
32                 'X' when OTHERS;
33         set <= pom2;
34     end arch;

```

Slika 6. VHDL opis jednobitne ALU.

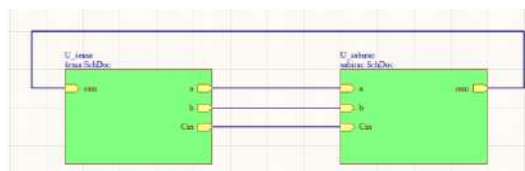
Na Slici 7 prikazan je vremenski dijagram izlaznih i ulaznih signala ovog primera ALU jedinice. U trenutku označenom kursorom, ulazi imaju vrednost $A = 1$, $B = 0$, $sub = 0$, $less = 0$, $S = 010$, $Cin = 1$, dok su vrednosti izlaza $Cout = 1$, $R = 1$ i $Set = 0$. Dakle, na izlaz je prosleđena operacija sa drugog ulaza multipleksera, a to je operacija XOR (isključivo ILI).



Slika 7. Vremenski dijagram ulaznih i izlaznih signala jednobitne ALU.

IV. REALIZACIJA PROTO PLOČE U ALTIIUM DESIGNER-U

U programu Altium Designer je isprojektovano kolo jednobitne aritmetičko logičke jedinice. Na početku realizacije potrebno je nacrtati šemu kola, kao i prilikom opisa, tako i prilikom projektovanja projekat je organizovan hijerarhijski. Postoje dva šematika, glavnog kola i sabirača. Ove dve šeme su ravnopravne po važnosti, ali je neophodno da se međusobno povežu na posebnom šematičku koji je većeg prioriteta, tako da se stvara hijerarhijski poredak svih šema u projektu. Na Slici 8 može se videti kako izgleda povezivanje ove dve šeme.



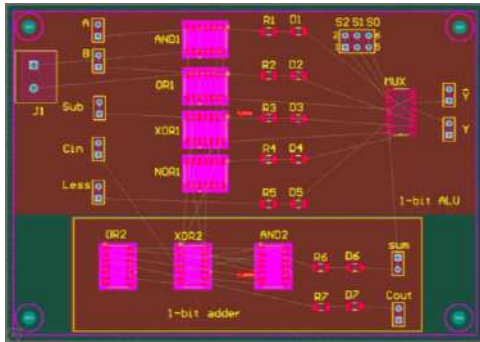
Slika 8. Prikaz „Glavne šeme“ projekta.

Pre prelaska u PCB projekat potrebno je definisati oblik i dimenzije pločice. U ovom slučaju pločica je dimenzija 7 cm x 10 cm. Potrebno je da pločica ima određene rupe za montažu u kućište ili u ovom slučaju, za montažu šrafova koji bi služili kao postolje, za pritupačniji rad na ploči. Nakon uspešno povezanih šema, svim komponentama se pridružuju odgovarajući otisci koji će se pojavljivati u 2D prikazu pločice. Nakon toga, iz šematika se može preći u PCB projekat, naredbom *Update PCB document*.

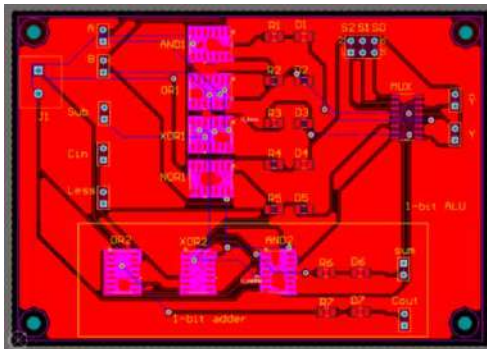
S obzirom da u projektu postoje dve šeme, sve komponente, koje se pojavljuju u vidu svojih *footprint*-ova, biće raspoređene u dva posebna *room*-a. Svaki od *room*-ova odgovara jednom od šematika.

U PCB projektu su dostupna dva načina prikaza, 2D i 3D. U 2D režimu mogu se videti otisci komponenti, dok je za 3D prikaz potrebno komponentama pridružiti odgovarajuće 3D modele. Na Slici 9 može se videti raspored komponenti na pločici pre rutiranja, dok je na Slici 10 prikazana pločica nakon izvršenog rutiranja i razlivanja mase. Obe slike su zabeležene u 2D načinu prikaza. Rutiranje je izvršeno naredbom *Auto Route*, gde se prilikom povezivanja poštuje pravilo najmanje razdaljine između dve komponente. Pločicu je moguće prikazati i u

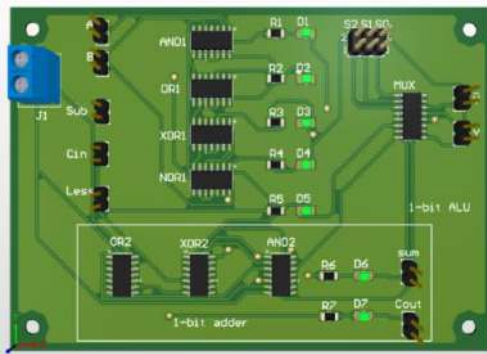
3D režimu, gde se komponente pojavljuju u vidu svojih 3D modela. Ovaj prikaz verodostojno pokazuje kako bi pločica izgledala ukoliko bi se realizovala (Slika 11).



Slika 9. 2D prikaz ploče pre rutiranja.



Slika 10. Prikaz ploče nakon rutiranja i razlivanja mase.

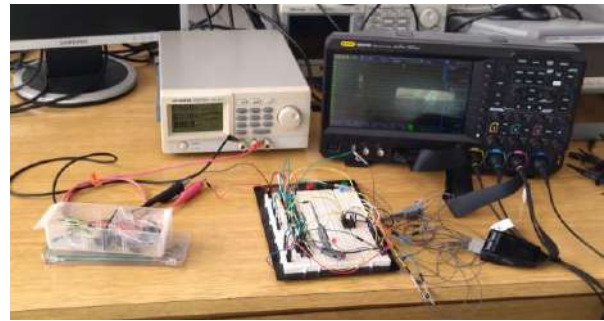


Slika 11. 3D prikaz ploče.

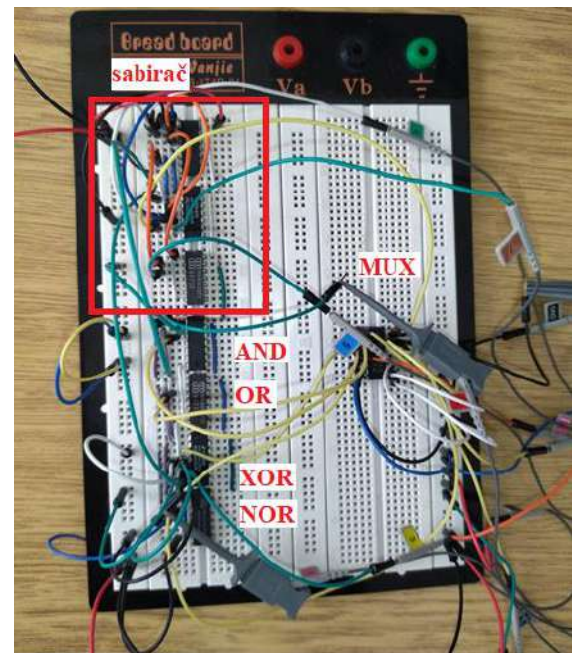
V. PRAKTIČNA REALIZACIJA

Na samom kraju izvršena je praktična realizacija jednobitne aritmetičko logičke jedinice i njeno testiranje. Za realizaciju korišćena su integrisana kola iz serije 7400: AND - MC74HC08AN [4], XOR - MM74HC86N [5], OR - M54HC32F1 [6], NOR - M74HC02B1 [7], MUX - M74HC151B1 [8]. Za praktičnu realizaciju je pored navedenih kola bila potrebna proto ploča, žice i izvor napajanja kao što se može videti na Slici 12. Za testiranje je

korišćen osciloskop Rigol MS05104 [9] koji ima ulogu i logičkog analizatora.



Slika 12. Prikaz radnog okruženja.



Slika 13. Prikaz povezane proto ploče.

Na Slici 13 prikazan je izgled povezanog kola na proto ploči, koje kao celina predstavlja jednobitnu ALU. Svi važni delovi kola su obeleženi na slici, treba naglasiti da se i sam sabirač sastoji od AND, OR i XOR kola. Zbog preglednosti korišćena su različita kola za izvršenje logičkih operacija i za realizaciju sabirača, iako bi po jedno integrisano kolo bilo dovoljno za uspešnu realizaciju.

VI. TESTIRANJE REALIZOVANOG KOLA

Na narednoj slici prikazane su zabeležene vrednosti ulaza i izlaza kola aritmetičko logičke jedinice (Slika 14). Ovo je samo jedan izdvojen primer svih zabeleženih operacija. Konkretno, radi se o operaciji AND, izabrana je baš ova operacija da bi mogla da se stvori analogija sa simulacijom iste ove operacije koja je prikazana u ranijim delovima teksta. S obzirom da je ova operacija na ulazu najmanje težine multipleksera, selekcionirani ulazi S2S1S0 su

jednaki "000". Ulaz *A* ima vrednost logičke 1, dok *B* ima vrednost logičke 0, rezultat operacije AND je 0, što odgovara izlazu *Y*.



Slika 14. Primer propuštanja operacije AND na izlaz.

ZAHVALNICA

Zahvaljujem se profesorima sa Katedre za mikroelektroniku na pomoći koju su mi pružili tokom izrade projekta, posebno Profesoru dr Danijelu Dankoviću i asistentu Nikoli Mitroviću.

LITERATURA

- [1] ALU, dostupno na: <https://www.computerhope.com/jargon/a/alu.htm>
- [2] 13.2 Loic Operations, dostupno na: <http://wwwcs.uwm.edu/classes/cs315/Bacon/Lecture/HTML/ch13s02.html>
- [3] Goran j. Dorđević, *Arhitekture mikrosistema*, Niš, Srbija, 2009., dostupno na: <https://sites.ooble.com/site/ads202021amseembedded/predavanja>
- [4] AND -C74HC08AN datasheet, dostupno na: <https://www.alldatasheet.com/datasheetpdf/pdf/12156/ONSEMI/MC74HC08AN.html>
- [5] XOR -MM74HC86N datasheet, dostupno na: https://eu.ouser.com/datasheet/2/308/1/MM74HC86_D-2315791.pdf
- [6] OR - 54HC32F1 datasheet, dostupno na: <https://www.alldatasheet.com/view.jsp?Searchword=M54HC32F1-SGS>
- [7] NOR -74HC02B1 datasheet, dostupno na: https://www.lldatasheet.com/view_datasheet.jsp?Searchword=M74HC02B1
- [8] MUX - M74HC151B1 datasheet, dostupno na: <https://www.alldatasheet.com/datasheetpdf/pdf/15539/PHILIPS/74HC151.html>
- [9] Rigol MS05104, dostupno na: https://beyondmeasure.rigoltech.com/acton/attachment/1579/f-0907/1/-/-/-/-/MSO5000_datasheet.pdf

Tester UV zaštite naočara

Anđela Panić

Sadržaj - Tester služi za testiranje UV zaštite naočara. Princip rada baziran je na merenju relativnog odnosa intenziteta UV zraka pre i nakon prolaska kroz sočivo naočara.

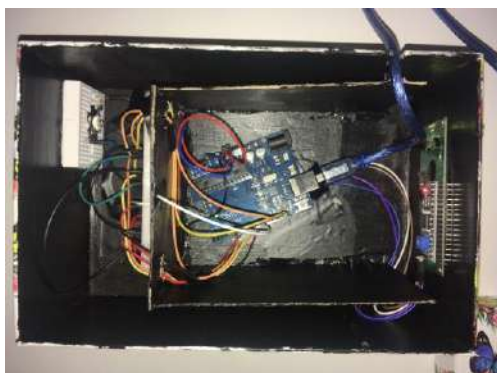
I. UVOD

Izlaganje UV zračenju bez adekvatne zaštite može biti rizično po zdravlje ljudi [1]. UV zračenja mogu naneti štetu koži, ali i drugim delovima ljudskog tela kao što su oči. Većina naočara za sunce i naočara za vid koje se koriste ne poseduju potrebnu UV zaštitu. Uređaj zvani fotometar obavlja određeno merenje na osnovu koga se može ustanoviti da li određene naočare poseduju UV zaštitu. Međutim, visoka cena fotometra omogućava njegovu dostupnost samo malom broju korisnika.

Budući da je za testiranje UV zaštite naočara dovoljno odrediti relativni odnos intenziteta UV zraka pre i nakon prolaska kroz sočivo naočara, moguće je, bez preciznog merenja apsolutnog intenziteta UV zračenja, napraviti jednostavan i jeftin tester UV zaštite naočara.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Princip rada testera (slika 1) baziran je na dva merenja intenziteta UV zračenja. Prvo merenje se izvodi bez prisustva naočara. Izmeren intenzitet UV zračenja zapravo predstavlja intenzitet zračenja koje dolazi do sočiva naočara. Potom se vrši drugo merenje, sa prisustvom naočara, koje omogućava merenje inteziteta UV zračenja koje je prošlo kroz sočivo naočara. Relativni odnos rezultata ova dva merenja ukazuje na apsorpciju UV zračenja u sočivu naočara, odnosno, govori da li postavljene naočare poseduju UV zaštitu.



Slika 1. Unutrašnji izgled testera.

A. Panić, učenica Tehničke škole u Loznici, Trg Jovana Cvijića 3, 15 300 Loznica, Srbija. e-mail: andjelapanic27@gmail.com

A. Delovi testera

Tester se sastoji iz:

- Mikrokontrolera Arduino UNO R3,
- Analognog UV senzora GYML8511,
- UV LED diode (365 nm),
- LCD displeja 16x2 (sa I²C interfejsom),
- Tastera,
- Kutije.

B. Hardver

Centralni deo testera čine UV LED dioda i senzor UV zračenja koji se fizički nalaze jedno naspram drugog (slika 2). Sočivo naočara čije se UV protekcija utvrđuje stavlja se između UV LED diode i UV senzora.



Slika 2. UV LED dioda i senzor UV zračenja.

UV LED dioda pobuđuje se direktno preko otpornika sa Arduino mikrokontrolera. Napon sa UV senzora dovodi se na analogni ulaz mikrokontrolera.

U testeru je korišćena UV LED dioda čija je talasna dužina emitovane svetlosti 365 nm. Analogni UV senzor GYML8511 osetljiv je na zračenja u UVB i UVA delovima UV spektra, a najveću osetljivost ima na 365 nm [2]. Izlazni napon senzora ima linearnu zavisnost u funkciji intenziteta UV zračenja koja je stabilna u širem temperaturnom opsegu [3].

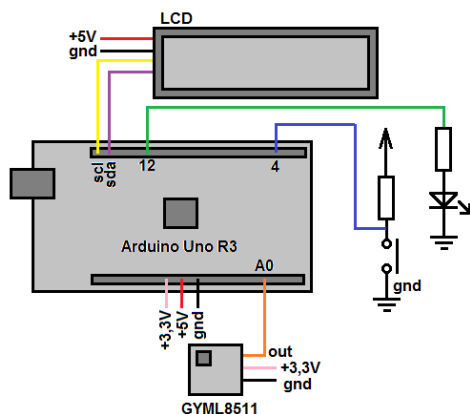
Mikrokontroler Arduino UNO R3 ima zadatak da pobuđuje UV LED diodu i da meri vrednost jednosmernog napona koji dobija od UV senzora. Vrednost merenja kao i

potrebna uputstva za korisnika testera UV zaštite naočara mikrokontroler ispisuje na LCD displeju.

Kako bi se sprečio uticaj svetlosti okoline u kojoj se obavlja testiranje, kompletan uređaj nalazi se u kutiji čija je unutrašnjost obojena u crnu boju. Merenje se obavlja kada je kutija zatvorena. Ovim je istovremeno od UV svetlosti LED diode zaštićena i osoba koja testira naočare.

Tester poseduje taster koji omogućava korisniku da pobudi UV LED diodu i izvrši potrebno merenje, pri čemu prvo dolazi do kratke pauze koja korisniku daje dovoljno vremena da zatvori kutiju pre nego što se UV LED dioda pobudi.

Sve komponente testera napajaju se sa +5 V sem UV senzora čije napajanje iznosi +3.3 V (slika 3).



Slika 3. Električna šema testera.

C. Softver

Za pisanje programa korišćen je Arduino IDE Software.

Program ima zadatak da izvrši dva merenja UV zračenja, sa i bez naočara i da njihov relativni odnos prikaže na LCD displeju (slika 4). Kroz niz uputstva koje program ispisuje na LCD displeju rukovanje ovim testerom omogućeno je i korisnicima koje ne moraju da znaju kako tester funkcioniše.



Slika 4. Ispis rezultata merenja.

U mikrokontroler Arduino Uno unet je sledeći program:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,2);
int taster = 4;
int uvled = 12;
int a1,a2;
float b;

void setup() {
  pinMode(taster, INPUT);
  pinMode(uvled, OUTPUT);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Andjela PanicE41");
  delay(1000);
  lcd.setCursor(1,1);
  lcd.print("Tehnicka skola");
  delay(1000);
  lcd.setCursor(0,1);
  lcd.print("    Loznica    ");
  delay(1000);
  lcd.setCursor(0,1);
  lcd.print("    2021/2022    ");
  delay(1000);
  lcd.setCursor(4,1);
  lcd.print("UV tester");
  delay(2000);
  lcd.clear(); }

void loop() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Uklonite naocare");
  delay (1000);
  lcd.setCursor(0,1);
  lcd.print("ZatvoriteUredjaj");
  delay (3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Potom pritisnite");
  lcd.setCursor(0,1);
  lcd.print("taster");

  while (digitalRead(taster)==1){};
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Prvo merenje...");
  delay(2000);
  digitalWrite(uvled, HIGH);
  delay(1000);
  a1=analogRead(A0);
  digitalWrite(uvled, LOW);
  lcd.setCursor(0,1);
  lcd.print("Zavrsono...");
  delay(2000);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Stavite naocare");
  delay (1000);
  lcd.setCursor(0,1);
  lcd.print("ZatvoriteUredjaj");
  delay (3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Potom pritisnite");
  lcd.setCursor(0,1);
  lcd.print("taster");
```

```

while (digitalRead(taster)==1){};
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Drugo merenje...");
delay(2000);
digitalWrite(uvled, HIGH);
delay(1000);
a2=analogRead(A0);
digitalWrite(uvled, LOW);
lcd.setCursor(0,1);
lcd.print("Završeno...");
delay(2000);

b=((a1-a2)*100.0)/a1;

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Rel. slabljenje");
lcd.setCursor(0,1);
lcd.print("je: ");
lcd.print(b);
lcd.print(" %");
delay(2000);

while (digitalRead(taster)==1){};
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Novo");
lcd.setCursor(0,1);
lcd.print("testiranje...");
delay(2000);}

```

III. ZAKLJUČAK

Sa konstantnim smanjenjem ozonskog omotača, velika je verovatnoća da će se rizik i posledice od UV zračenja povećati i u budućnosti. Da bi sačuvali svoje oči i vid, ljudi treba da koriste efikasna sredstva za zaštitu. Ovaj uređaj omogućava jednostavnu, efikasnu i brzu proveru UV zaštite kod naočara. Zbog jednostavnog načina i relativno jeftinih komponenti, uređaj može biti dostupan velikom broju korisnika, čime bi se predupredile štetne posledice UV zračenja na zdravlje ljudi.

ZAHVALNICA

Svoju zahvalnost za realizaciju ovog istraživačkog rada uputila bih pre svega svojoj porodici na podršci i ocu koji mi je pozajmio deo svoje opreme. Takođe bih se posebno zahvalila Tehničkoj školi Loznica i mentoru, profesoru Tomislavu Mijić koji me je savetovao, usmeravao i pružio svoju pomoć prilikom izrade i realizacije projekta.

LITERATURA

- [1] <https://www.framesdirect.com/knowledge-center/uv-coating>
- [2] <http://solar-center.stanford.edu/about/uvlight.html>
- [3] https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/M-L8511_3-8-13.pdf

Projektovanje razvojnog okruženja za primene operacionih pojačavača

Nemanja Tomić

Sadržaj - U ovom radu je predstavljeno integrisano kolo pomoću kojeg se vrši obrada signala u analognoj elektronici i bez kojeg je teško zamisliti rad nekog sadašnjeg analognog kola. Kolo pojačava signal na svom ulazu i vrši različite operacije nad ulaznim signalom da bi se dobio željeni signal na izlazu, pa mu je zbog toga i dat naziv operacioni pojačavač.

I. UVOD

Kolo operacionog pojačavača je nastalo iz potrebe za stabilnim pojačanjem signala u širokom frekvencijskom opsegu. Pre njega, aktivne komponente koje su se koristile nisu imale veliko pojačanje signala ili pojačanje nije bilo stabilno [1]. Ta nestabilnost se ogleda u izuzetno velikoj osetljivosti pojedinih parametara na spoljne uticaje (temperatura, vlažnost, pritisak,...).

Sama ploča je slikovito podeljena na 4 kvadranta i za svaki od kvadranta je napomenuto koje konfiguracije se u njemu mogu dobiti. Dodatne mase i trnove za napajanje se nalaze u gornjem delu ploče, radi lakšeg pristupa. Planirano dualno napajanje je ± 12 V, tako da treba obratiti pažnju koje napajanje se dovodi. Operacioni pojačavač LM2902 se nalazi u centru ploče, a u kvadrantima je naslikan simbol operacionog pojačavača u Top Overlay-u radi estetike i lakšeg uočavanja konfiguracija na ploči, dok su linije veza i razlivena masa odrađene u Top i Bottom Layer-u.

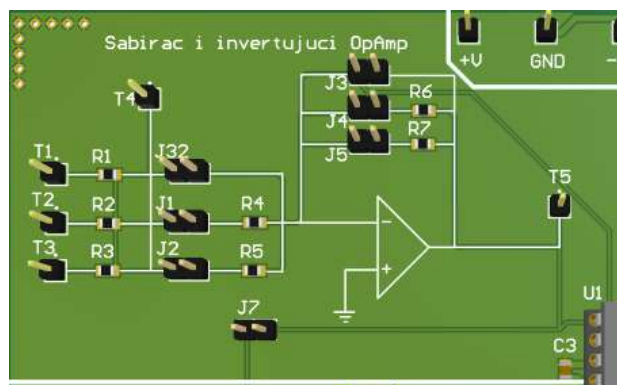
U ovom radu prikazane su konfiguracije invertujućeg i neinvertujućeg pojačavača, sabirača, diferencijalnog pojačavača, integratora i diferencijatora, logaritamskog i eksponencijalnog pojačavača kao i bafera impedanse tj. sleditelja napona.

II. PRIMENE OPERACIONOG POJAČAVAČA

Sabirač – na sabirač se preko trnova T_1 , T_2 i T_3 dovode tri signala koji idu na otpornike R_1 , R_2 i R_3 . Svaki od otpornika se vezuje za džemper J_{32} i dovodi signal na invertujući ulaz operacionog pojačavača. U povratnoj sprezi se u zavisnosti od željenog pojačanja bira određena vrednost otpornika i samim tim džampera J_4 ili J_5 . Izlazni signal se očitava sa trna T_5 , kao što se može videti na slici 1.

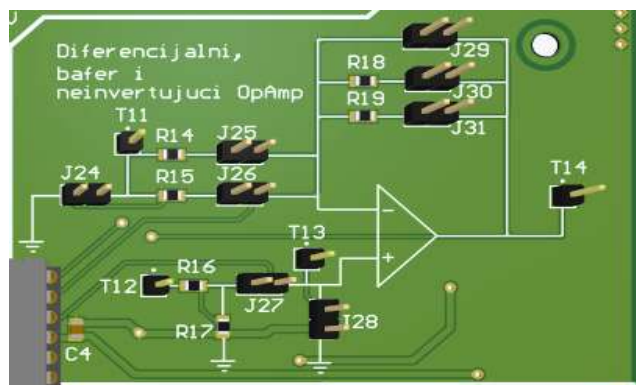
N. Tomić je student na modulu EKM, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, E-mail: n.tomic@elfak.rs

Invertujući pojačavač – preko trna T_4 se dovodi signal na jedan od džampera J_1 ili J_2 (opciono u zavisnosti koju vrednost otpornika tj. pojačanja želimo) i odatle na R_4 od $1\text{ k}\Omega$ i R_5 od $10\text{ k}\Omega$ koji su povezani na invertujući ulaz pojačavača. Neinvertujući ulaz je na masi. U povratnoj sprezi takođe postoji opcija da li će veza biti ostvarena preko džampera J_4 izabrati otpornik R_6 od $1\text{ k}\Omega$ ili na J_5 na otpornik R_7 od $10\text{ k}\Omega$. Izlazni signal se očitava sa trna T_5 , kao što se može videti na slici 1.



Slika 1. Izgled sabirača i invertujuće konfiguracije na štampanoj ploči u Altium-u.

Bafer impedanse – invertujući ulaz se preko džampera J_{29} povezuje na izlaz, dok su džamperi J_{30} , J_{31} , J_{25} i J_{26} otkaćeni. Na neinvertujući ulaz se signal dovodi preko trna T_{13} , dok su džamperi J_{28} i J_{27} otkaćeni. Izlazni signal se očitava sa trna T_{14} , kao što se može videti na slici 2.



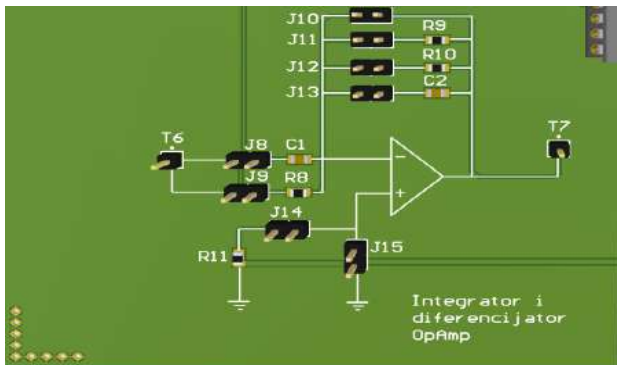
Slika 2. Izgled bafera impedanse, diferencijalnog pojačavača i neinvertujuće konfiguracije na štampanoj ploči u Altium-u.

Diferencijalni pojačavač – signali se dovode na oba ulaza pojačavača; na neinvertujući ulaz se preko trna T_{12}

dovodi signal na otpornike R_{16} i R_{17} redno vezane i džemper J_{27} je povezan, a J_{28} otkačen. Zavisno od toga koje pojačanje želimo, signal se na invertujući ulaz dovodi preko otpornika R_{14} ili R_{15} . Ako izaberemo R_{14} , onda je džemper J_{25} kratkospojen, a J_{26} otkačen. Obrnuto je ako izaberemo otpornik R_{15} . U povratnoj sprezi se nalaze paralelno otpornici R_{18} i R_{19} otpornosti 10 k Ω i 100 k Ω respektivno. Bira se željeni otpornik i njegov redno vezani džemper se povezuje, a drugi se otkači. Džemper J_{29} je otkačen. Signal se očitava sa trna T_{14} , kao što se može videti na slici 2.

Neinvertujuća konfiguracija – signal se na neinvertujući ulaz dovodi preko trna T_{13} dok su džamperi J_{27} i J_{28} otkačeni. Opciono se bira vrednost otpornika koji se vezuju na invertujući ulaz pojačavača (R_{14} ili R_{15}) kao i u povratnoj sprezi između otpornika R_{18} ili R_{19} . Invertujući ulaz je vezan na masu preko džampera J_{24} . Signal se očitava preko trna T_{14} , kao što se može videti na slici 2.

Integrator – preko trna T_6 dovodimo signal na džemper J_9 i otpornik R_8 , dok je džemper J_8 otkačen. U povratnoj sprezi su svi džamperi sem J_{13} otkačeni, a J_{13} ide na kondenzator C_2 . Neinvertujući ulaz se može povezati na masu preko džampera J_{15} , dok je džemper J_{14} otkačen. Da bi se poboljšao rad integratora u povratnoj sprezi se stavlja još jedan otpornik R_9 od 100 k Ω zbog ulaznog offset napona, i on je paralelno vezan kondenzatoru C_2 , i na neinvertujući ulaz se dovede otpornik R_{11} preko džampera J_{14} . Signal se očitava sa trna T_7 , kao što se može videti na slici 3.



Slika 3. Izgled integratora i diferencijatora na štampanoj ploči u Altium-u.

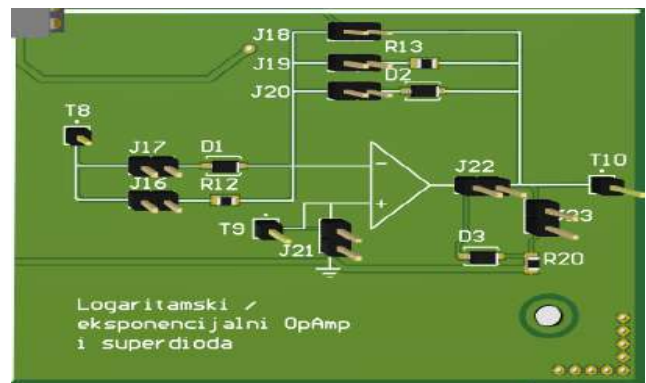
Diferencijator – preko trna T_6 se signal dovodi na džemper J_8 i kondenzator C_1 dok je džemper J_9 otkačen. U povratnoj sprezi se poveže džemper J_{11} i otpornik R_9 od 100 k Ω , dok su ostali džamperi otkačeni. Neinvertujući ulaz je na masi. Signal se očitava sa trna T_7 , kao što se može videti na slici 3.

Ekspnencijalni pojačavač – sa trna T_8 se dovodi signal na džemper J_{17} koji ide na direktno polarisanu diodu

D_1 i u invertujući ulaz, dok je džemper J_{16} otkačen. U povratnoj sprezi je džemper J_{19} povezan i na njega je redno vezan otpornik R_{13} male otpornosti. Džamperi J_{18} , J_{20} i J_{23} su otkačeni, dok je džemper J_{22} kratkospojen. Neinvertujući ulaz je na masi preko džampera J_{21} . Signal se očitava sa trna T_{10} , kao što se može videti na slici 4.

Logaritamski – sa trna T_8 se signal dovodi na džemper J_{16} , dok je džemper J_{17} otkačen. Džemper J_{16} je redno vezan za otpornik R_{12} . U povratnoj sprezi je džemper J_{20} povezan na direktno polarisanu diodu D_2 , dok su džamperi J_{18} , J_{19} i J_{23} otkačeni. Neinvertujući ulaz je na masi preko džampera J_{21} . Džemper J_{22} je kratkospojen. Izlazni signal se očitava sa trna T_{10} , kao što se može videti na slici 4.

Superdioda - sa trna T_9 se signal dovodi na neinvertujući ulaz, dok je džemper J_{21} otkačen. Invertujući ulaz je preko džampera J_{18} povezan na izlaz. Džamperi J_{22} , J_{16} , J_{17} , J_{19} , J_{20} su otkačeni, a džemper J_{23} povezuje konfiguraciju diode i otpornika sa izlazom koji se očitava na trnu T_{10} , kao što se može videti na slici 4.

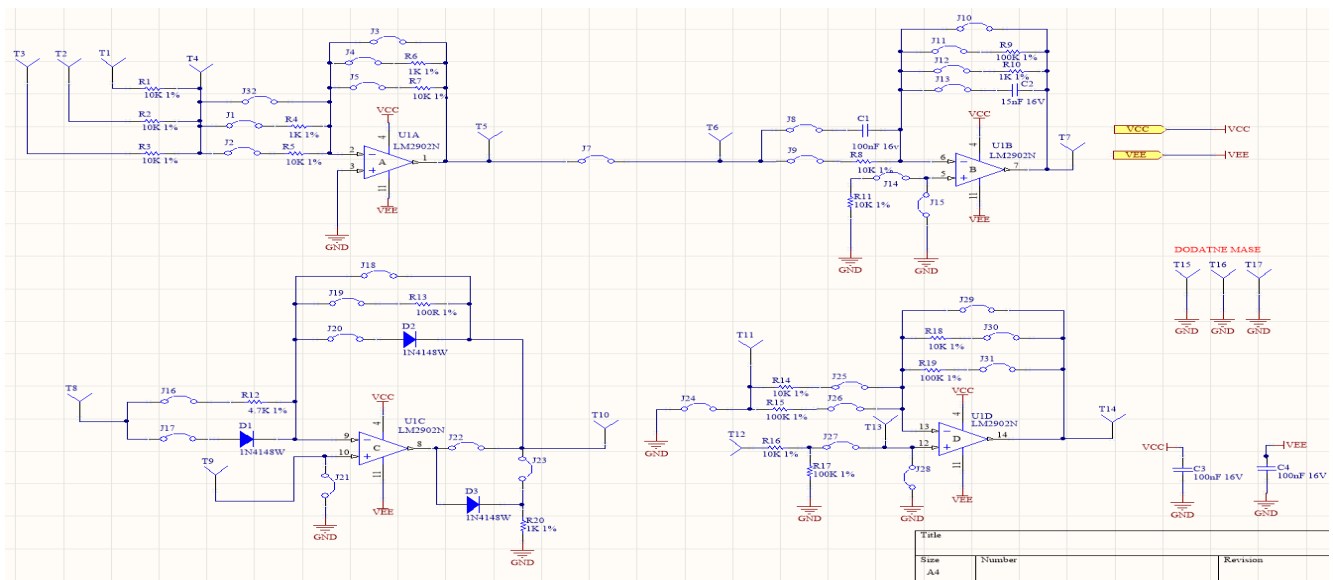


Slika 4. Izgled ekspnencijalnog, logaritamskog pojačavača i superdiode na štampanoj ploči u Altium-u.

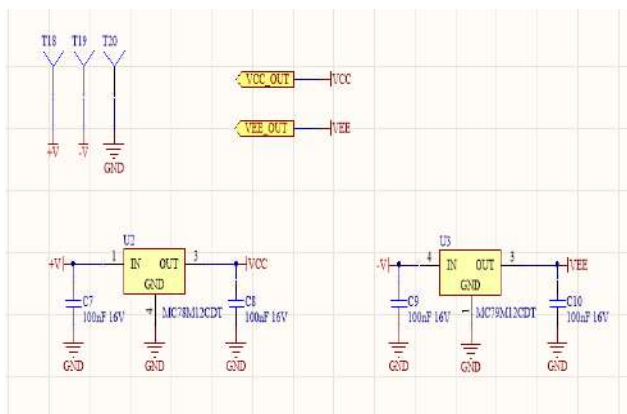
III. PROCES PROJEKTOVANJA PLOČE

Da bi se napravila štampana ploča, potrebno je napraviti odgovarajući šematič na kome treba staviti komponente sa njihovim footprint-ovima. U našem slučaju treba napraviti dva šematika: Jedan za napajanje, koje se nalazi na slici 6, i drugi na kojem treba da budu operacioni pojačavači sa odgovarajućim konfiguracijama, koje treba spojiti.

Svako od komponenti je dodeljen odgovarajući footprint. Na slici 5 se nalaze 4 operaciona pojačavača koja su smeštena u kućištu DIP 14 operacionog pojačavača LM2902. Svaki od operacionih pojačavača ima određenu konfiguraciju za koju može da radi.



Slika 5. Električna šema sa konfiguracijama.



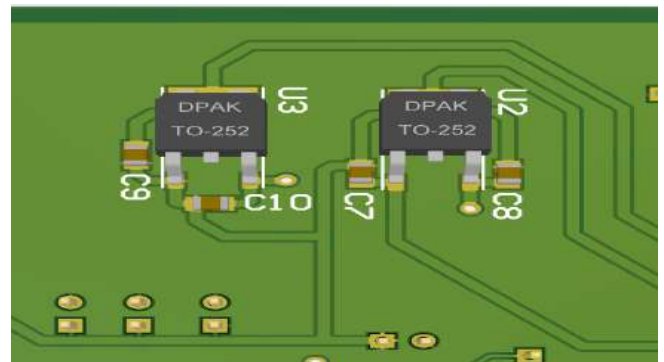
Slika 6. Izgled šematika za napajanje.

Kada su napravljeni odgovarajući šematiki, sve komponente su kompajlirane i prebačene na ploču. Nakon toga su poređane, vodeći računa o pravilima za postavljanje komponenti, odnosno ne preklapati linije veza. Treba definisati odgovarajuće klase i debljinu linija. Podešena je širina linija u delu kola koji se tiče napajanja, jer tu protiče veća struja. (Width_Power). Kada je to urađeno, treba dodeliti klasu linija koja će imati veću širinu. Kada je to definisano, potrebno je da ovom pravilu dodelimo klasu.

Nakon toga, kada je sve definisano, prelazi se na rutiranje komponenti uz poštovanje pravila o štampanim pločama. Korišćen je Auto Route, pa su ručno ispravljani pojedini detalji [3]. Posle rutiranja, potrebno je preko cele površine razliti masu, izuzev delova gde prolaze veze. To se radi iz više razloga, ali osnovni motiv je da se smanji električni šum i da se minimizuju neželjene petlje koje mogu biti ostvarene preko zajedničke mase. Izgled ploče nakon razlivanja mase prikazan je na slici 7.

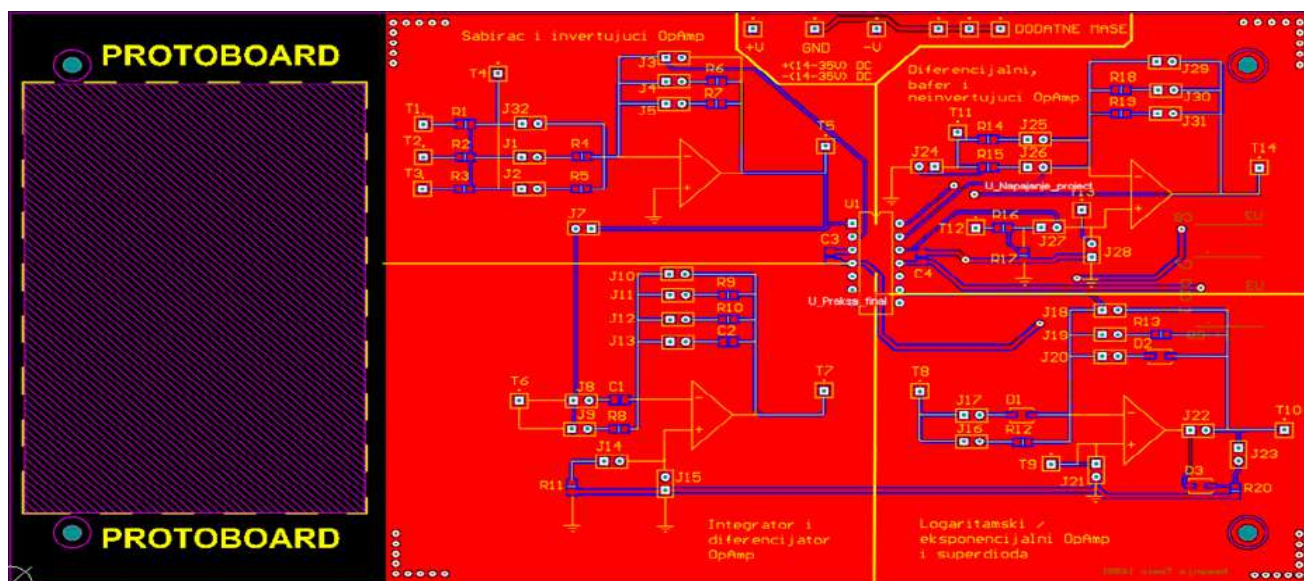
Razlivanjem mase se smanjuju efekti preslušavanja signala između pojedinih veza na štampanoj ploči, odnosno da se ne mešaju signali. Ovo ima veliku važnost pri velikim učestanostima, odnosno kad kroz linije veza prolaze signali koji se mnogo menjaju u vremenu.

Kada su mase razlivenne, potrebno je spojiti ih Via-ma, dakle spojiti gornju i donju površinu ploče, jer su na istom potencijalu. Najbolje mesto za to jesu uglovi pločice. Da bi estetski izgledalo lepše, vrši se poravnanje via horizontalno i vertikalno.



Slika 8. Regulatori napona na štampanoj ploči u Altium-u.

Zadatak izvora napajanja je da naizmenični napon iz mreže pretvori u jednosmerni, svodeći pri tome njegovu amplitudu na željeni nivo, kao i da takav jednosmerni napon učini dovoljno stabilnim u smislu imunosti na varijacije mrežnog napona i šumove. Podela izvora napajanja je na linearne i prekidačke. Kod linearnih izvora napajanja se, u opštem slučaju, ulazni napon vrednosti VIN transformiše u izlazni napon vrednosti VOUT na principu promenljive provodnosti elementa koji se nalazi između ulaza i izlaza, pri čemu je $VOUT < VIN$.



Slika 7. 2D izgled štampane ploče posle razlivanja mase.

U ovom slučaju korišćeni su linearni regulatori napona MC78M12ACDT i MC79M12ACDT za ± 12 V, koji su prikazani na slici 8. Nalaze se sa donje strane ploče. Korišćena su dva regulatora za dualno napajanje koji su smešteni na donji deo ploče.

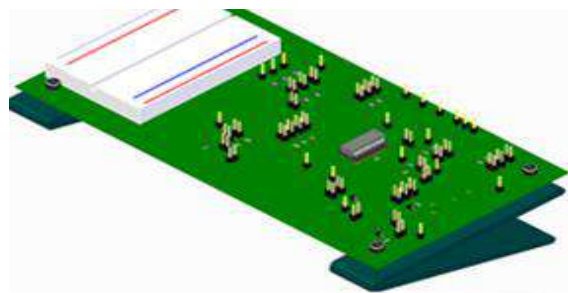
IV. PROCES PROJEKTOVANJA PLOČE I KONAČAN IZGLED

Postolje je napravljeno korišćenjem softverskog alata Solid Edge, sa namerom da drži pločicu. Napravljeno je sa nagibom prema korisniku radi lakšeg rada sa pločicom. Pravljen je od jednog materijala isečeno naredbom Cut na željeni oblik. Na postolju treba izbušiti rupe na koje će pločica biti pričvršćena odgovarajućim šrafovim. Između pločice i postolja, treba da bude odstopnik, jer se sa donje strane ploče nalaze linearni regulatori, odnosno da se oni ne bi grejali. Dimenzije odstopnika ne treba da budu velike, tj. da ne dolaze do prevelikog izražaja.



Slika 9. Izgled postolja sa odgovarajućim odstopnicima.

Kada je napravljeno odgovarajuće postolje i određeno sve što je potrebno da se pločica napravi, treba ih spojiti.



Slika 10. Konačan izgled ploče.

V. ZAKLJUČAK

U radu je objašnjeno i prikazano razvojno okruženje za primene operacionih pojačavača, odnosno konfiguracije operacionog pojačavača koje su stavljene na štampanu ploču, izrađene u softverskom alatu Altium Designer i smeštene na postolje. Objašnjene su sve komponente koje se nalaze na ploči i kojoj konfiguraciji pripadaju.

ZAHVALNICA

Autor se zahvaljuje prof. dr Danijelu Dankoviću i dipl. ing. Nikoli Mitroviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji projekta i pisanju rada.

LITERATURA

- [1] Z. Prijić, Analogni mikroelektronika (Operacioni pojačavači), Predavanja, Niš
- [2] Johan Huijsing, "Operational Amplifier" 2016
- [3] Majid Pakdel, "Fast PCB Design with Altium Designer"

Softver za generisanje NBS IPS QR kôdova

Nikola Milošević, Luka Nikolić

Sadržaj – QR (Quick Response) kôdovi su postali deo naše svakodnevice. Prvobitno kreirani za praćenje automobilskih delova u fabrikama, QR kôdovi su, preko ostalih grana industrije, došli i do primene u skoro svemu što okružuje prosečnog čoveka danas. Jedna od tih primena je u okviru finansijskih transakcija, u cilju pojednostavljenja različitih operacija i smanjenja mogućnosti greške u elektronskom i mobilnom bankarstvu. Narodna banka Srbije je pratila svetske trendove i pre nekoliko godina uvrstila QR kôdove u platni sistem. Mnoga preduzeća su već dodala QR kôdove svojim računima, tako da je njihovo plaćanje izuzetno lako i brzo. Njima se pridružila i Stambena zajednica Bulevar Nikole Tesle 21 Niš, za čije potrebe je razvijen softver za generisanje QR kôdova za plaćanje zajedničkih troškova. U ovom radu je dat pregled korišćene tehnologije, a prikazan je i sâm softver, koji je već u upotrebi.

I. UVOD

Od pronalaska QR (Quick Response) kôdova 1994. godine od strane japanske Denso Wave korporacije [1]–[3], ova tehnologija je sve više prisutna u različitim sferama ljudskog života. U početku su QR kôdovi bili korišćeni za označavanje automobilskih delova u Tojotinoj fabrici automobila, gde su zamenili dotadašnje jednodimenzionalne bar kôdove. Međutim, petnaestak godina kasnije, sa razvojem pametnih telefona, kao čitača, QR kôdovi su počeli masovno da se koriste na reklamnim materijalima, za finansijske transakcije, u obrazovnim institucijama itd.

Dobre osobine i jednostavnost upotrebe QR kôdova je prepoznala Narodna banka Srbije (NBS), pa je od oktobra 2018. godine [4] omogućila korišćenje QR kôdova za jednostavnije plaćanje računa i nazvala ih NBS IPS QR kôdovima. Naime, QR kôd sadrži instrukcije za popunjavanje uplatnice, a mobilna aplikacija neke banke koristi te instrukcije i automatski popunjava uplatnicu.

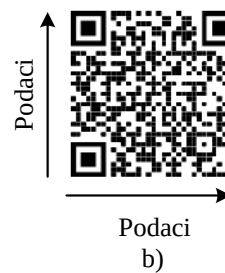
U ovom radu je prikazan softver koji služi za generisanje NBS IPS QR kôdova za plaćanje mesečnih računa za zajedničke troškove u okviru Stambene zajednice Bulevar Nikole Tesle 21, Niš. Na osnovu unetih zaduženja za svaki stan, softver generiše Microsoft Word dokument koji sadrži klasične tekstualne podatke za svaki stan, ali i QR kôdove, tako da je plaćanje računa izuzetno jednostavno i brzo.

II. QR KÔDOVI

Kao što je već rečeno, QR kôdovi su zamenili

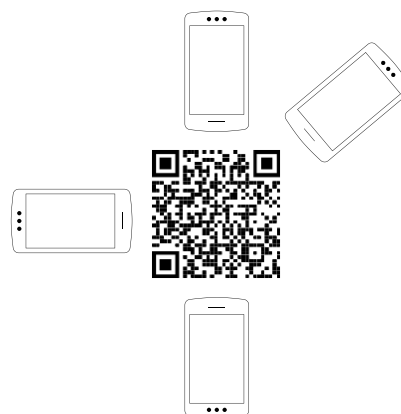
N. Milošević i L. Nikolić su učenici Gimnazije “Bora Stanković”, Vožđa Karađorđa 27, 18000 Niš, Srbija, E-mail: nikola.milosevic05@gmail.com, luka.nikolic973@gmail.com

jednodimenzionalne bar kôdove, prvenstveno zbog mogućnosti da se na istoj površini smesti veći broj znakova. Naime, bar kôd je mogao da sadrži do 20 karaktera. Kod QR kôda je informacija sadržana u dve dimenzije, pa je kapacitet porastao više stotina puta (Slika 1).



Slika 1. Poređenje bar kôda (a) i QR kôda (b).

Pored povećanja kapaciteta, zadatak QR kôdova je bio i da se omogući očitavanje kôda pri bilo kojoj orijentaciji čitača, odnosno čitač je mogao da bude zarotiran u odnosu na QR kôd za bilo koji ugao od 0 do 360 stepeni (Slika 2).



Slika 2. Bilo koja orijentacija čitača QR kôda.

Još jedan važan zahtev je bio da očitavanje bude veoma brzo. Sve to je postignuto dodavanjem kvadratnih

markera u uglovima kôda. Čitač detektuje postojanje markera, a samim tim i postojanje QR kôda i njegovu orijentaciju. Markeri su tako izabrani da je jako mala verovatnoća da u bilo kom tekstu ili slici postoje isti takvi markeri sa istim rasporedom, a da ne pripadaju validnom QR kôdu. Naime, inženjeri Denso Wavea su izvršili analizu ogromnog broja štampanih materijala različitog porekla [3] i došli do zaključka da najmanje korišćena kombinacija crnih i belih površina ima odnos 1:1:3:1:1, pa je taj odnos upotrebljen za kreiranje markera. Marker počinje crnom oblašću neke jedinične širine, iza kog sledi bela oblast iste širine, pa crna oblast trostruke širine, a na kraju su opet bela, pa crna oblast jedinične širine. S obzirom da može da dođe do oštećenja nekog dela QR kôda, koje bi prouzrokovalo pogrešno očitavanje, u QR kôd je ugrađena mogućnost da se takve greške poprave. U slučaju da je oštećenje preveliko, čitač će da signalizira da je QR kôd neispravan ili ga uopšte neće očitati. Povećanjem otpornosti na oštećenja, smanjuje se korisni kapacitet kôda, jer se neki od znakova koriste za popravku onih oštećenih. Zbog fleksibilnosti, predviđeno je da prilikom kreiranja QR kôda može da se bira jedna od 4 moguće otpornosti na oštećenje, označene slovima L, M, Q i H, kao što je prikazano u Tabeli I [3].

TABELA I
MOGUĆNOST POPRAVKE OŠTEĆENOG QR KÔDA

Nivo otpornosti	Maksimalni dozvoljeni procenat oštećenja
L	7 %
M	15 %
Q	25 %
H	30 %

Svaki QR kôd se sastoji iz takozvanih modula, crnih ili belih kvadratića, odnosno QR kôd predstavlja kvadrat čije su stranice jednake određenom broju modula. Prva verzija QR kôda je imala stranicu širine 21 modul. Sa unapređenjem čitača i potrebama da se sve više teksta smesti na QR kôd, povećavan je i kapacitet QR kôda povećavanjem broja modula. Druga verzija je imala stranicu širine 25 modula, treća 29, sve do trenutno najnovije verzije 40, koja ima stranicu širine 177 modula (Slika 3).



Slika 3. Različite verzije QR kôda za fiksnu dimenziju modula.

Na slici 3 su prikazani QR kôdovi različitih dimenzija (verzija), pri čemu sva tri kôda sadrže isti tekst (adresa web sajta konferencije). Ovo je primer pogrešne upotrebe različitih verzija kôda. Verzija QR kôda najmanjeg kapaciteta na slici (verzija 2) je sasvim dovoljna za smeštanje pomenutog teksta, tako da nema smisla koristiti kôdove većeg kapaciteta, koji se teže čitaju i kod kojih je veća mogućnost greške. Zbog toga generator QR kôda treba da koristi najmanju verziju kôda koja je dovoljna za onaj broj znakova koje treba kodirati. Kapacitet QR kôdova za dve najstarije i dve najnovije verzije je prikazan u tabeli II [3].

TABELA II
KAPACITET RAZLIČITIH VERZIJA QR KÔDOVA

Verzija	Broj modula	Nivo otpornosti	Broj znakova	
			Numeričkih	Alfanumeričkih
1	21x21	L	41	25
		M	34	20
		Q	27	16
		H	17	10
2	25x25	L	77	47
		M	63	38
		Q	48	29
		H	34	20
39	173x173	L	6743	4087
		M	5313	3220
		Q	3791	2298
		H	2927	1774
40	177x177	L	7089	4296
		M	5596	3391
		Q	3993	2420
		H	3057	1852

Ukoliko postoji potreba, dalje povećanje kapaciteta QR kôdova bi moglo da se postigne ukoliko bismo umesto crno-belih modula koristili module u boji [5].

Pored prikazanog, postoji još nekoliko vrsta QR kôdova. Na primer, oni koji u sebi mogu da sadrže grafički logo neke kompanije. Međutim, kako nisu od interesa za ovaj rad, neće biti detaljno prikazani.

III. NBS IPS QR KÔD

QR kôd Narodne banke Srbije praktično predstavlja posebno strukturiran tekst, koji aplikacija neke banke prepoznaje i na osnovu njega popunjava podatke za uplatu. Tekst, koji je potreban za razvijenu aplikaciju, se sastoji iz sledećih polja [4]:

- K; Ovo polje ima fiksni sadržaj (PR) i predstavlja identifikator NBS IPS QR kôda,
- V; Verzija koda, koja je trenutno 01,
- C; Znakovni skup, trenutno 1,

- R; Broj računa primaoca plaćanja. Ovaj broj se, u principu, sastoji iz tri dela: trocifreni identifikator banke, 13-cifarni broj računa u banci i dvocifreni kontrolni broj. Ukoliko broj računa nema 13 cifara, treba ga proširiti do 13 cifara dodavanjem odgovarajućeg broj nula ispred broja. Zbog toga ovo polje uvek treba da ima 18 cifara,
- N; Naziv primaoca plaćanja. Ovo polje može sadržati najviše 70 karaktera raspoređenih u najviše tri reda,
- I; Valuta i iznos novčanih sredstava, bez razmaka između njih. Valuta je RSD. Obavezno se upisuje decimalni zarez iza koga se ne moraju pisati nepotrebne decimalne nule (1025, ili 1025,1). Maksimalni iznos može biti 999999999999,99,
- SF; Šifra plaćanja. Za transakcije po nalogu građana, šifra plaćanja je 289,
- S; Svrha plaćanja. Ovo polje može da ima najviše 35 karaktera,
- RO; Poziv na broj odobrenja. Ovo polje se sastoji iz dvocifrene oznake modela i samog poziva na broj. Ukoliko se poziv na broj ne izračunava ni prema jednom modelu, tada se model označava sa 00.

Pomenuta polja se pišu jedno iza drugog i razdvojena su vertikalnim crtama. Primer jednog teksta, spremnog za kreiranje NBS IPS QR kôda je: K:PR|V:01|C:1|R:840000000172166689|N:Elektronski fakultet, Niš|I:RSD0,00|SF:289|S:Primer za NBS IPS QR kôd|RO:0012345. Ogovarajući QR kôd je prikazan na slici 4.



NBS IPS QR

Slika 4. Primer generisanog NBS IPS QR kôda.

IV. PRIKAZ REALIZOVANOG SOFTVERA

U okviru svake stambene zajednice svakog meseca stanari plaćaju račune za zajedničke troškove. To može da bude dosta zamorno, jer je potrebno uneti naziv, adresu i broj računa stambene zajednice, zatim svrhu plaćanja, poziv na broj odobrenja i iznos koji se plaća. Gledajući koliko vremena ovaj posao oduzima ljudima iz okruženja, autori su došli na ideju da naprave program pomoću koga će blagajnik stambene zajednice generisati NBS IPS QR

kôdove za svaki stan ponaosob i zatim ih poslati stanarima. To bi trebalo da značajno ubrza plaćanje i smanji mogućnost dešavanja greške u procesu plaćanja.

Polja, potrebna za generisanje teksta koji se umeće u QR kôd, opisana u prethodnom odeljku, u ovom slučaju imaju sledeći sadržaj:

- K:PR,
- V:01,
- C:1,
- R:160000000050561035,
- N:Stambena zajednica BNT 21, 18000 Niš,
- I:RSD1245,56,
- SF:289,
- S:Zajednički troškovi, 11/2021,
- RO:00xx. Poziv na broj (xx) se menja od 01 do 20, u zavisnosti od broja stana.

Softver je napisan u programskom jeziku C# u Visual Studio 2019 razvojnom okruženju. Za generisanje QR kôda na osnovu teksta je korišćena QRCoder biblioteka [6]. Rezultat se upisuje u Microsoft Word dokument, korišćenjem Microsoft.Office.Interop.Word biblioteke. Microsoft ovu tehniku naziva Automation i, da bi funkcionisala, neophodno je da Word bude instaliran na računaru. Prvo se iz aplikacije startuje Word, pa se kreira novi dokument, popuni podacima i na kraju snimi. Wordu se instrukcije praktično daju VBA (Visual Basic for Applications) komandama, čiji je pregled dat u daljem tekstu. Word se pokreće komandom:

```
Word.Application app=new Word.Application();
```

dok se kreiranje dokumenta pomoću:

```
Word.Document doc=app.Documents.Add();
```

Formiranje novog pasusa, njegovo centriranje i dodavanje teksta se vrši pomoću sledeće sekvence:

```
app.Selection.TypeParagraph();
```

```
app.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;
```

```
app.Selection.TypeText("tekst");
```

QR kôd se kreira u aplikaciji, privremeno sačuva kao PNG slika, pa se umetne u Word pomoću:

```
app.Selection.InlineShapes.AddPicture(path);
```

gde je path putanja do sačuvane slike.

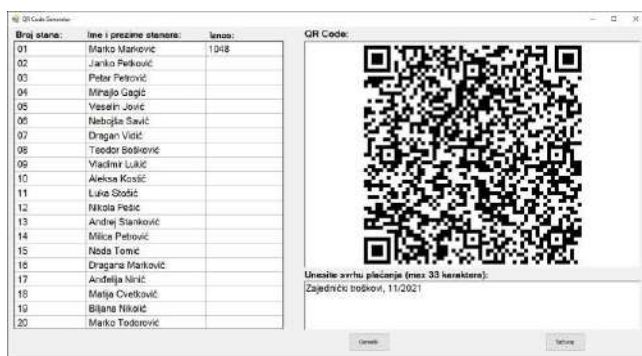
Na kraju, kada su upisani svi podaci, potrebno je sačuvati Word dokument i zatvoriti Word aplikaciju:

```
doc.Close();
```

```
app.Quit();
```

Izgled aplikacije je prikazan na slici 5, dok je izgled generisanog Word dokumenta, za jedan stan, prikazan na slici 6. Kao što se vidi, za ovu dužinu teksta bio je potreban QR kôd verzije 6 sa 41×41 modula. Zbog zaštite

privatnosti, u slikama iz aplikacije su prikazana izmišljena imena stanara.



Slika 5. Izgled forme napisanog softvera.



Slika 6. Izgled generisanog Word dokumenta za jedan stan.

V. ZAKLJUČAK

QR kôdovi su sve značajniji i imaju sve širu primenu, pa ih je pre tri godine Narodna banka Srbije odabrala kao jednu od tehnologija koje će se koristiti u platnom prometu. U ovom radu je prikazan softver, pisan u programskom

jeziku C#, za generisanje NBS IPS QR kôdova za lakše plaćanje zajedničkih troškova u okviru Stambene zajednice Bulevar Nikole Tesle 21 Niš. Na osnovu mesečnih zaduženja, blagajnik Stambene zajednice je u mogućnosti da sada generiše QR kôdove, uz čiju upotrebu stanari mogu da na jednostavan način plate svoje račune.

U ovoj fazi, u okviru softvera nije predviđena mogućnost promena podataka o stambenoj zajednici, kao što su broj i imena stanara, broj računa itd. Međutim, ukoliko se ukaže potreba, aplikacija se može proširiti, tako da svi pomenuti podaci budu promenljivi. Takođe, može se ubaciti podrška za veći broj stambenih zajednica, što bi moglo biti od interesa profesionalnim upravicima stambenih zajednica.

ZAHVALNICA

Zahvaljujemo se svojim profesorima Draganu Iliću i Vladanu Mihajloviću za dragocene lekcije iz programiranja, kao i svojim roditeljima na pomoći prilikom formulacije problema i pisanja ovog rada.

Ovaj rad je delimično realizovan u okviru Erasmus+ projekta ICT_EDUPAND (2020-1-PL01-KA226-096196).

LITERATURA

- [1] C. Aktaş, *The Evolution and Emergence of QR Codes*, 1st ed. Newcastle upon Tyne, GBR: Cambridge Scholar Publishing, 2017.
- [2] S. Tiwari, "An Introduction to QR Code Technology," in *2016 International Conference on Information Technology (ICIT)*, 2016, pp. 39–44, doi: 10.1109/ICIT.2016.021.
- [3] Denso Wave Incorporated, "QR Code," *QR Code.com*, 2021. <https://www.qrcode.com>.
- [4] Narodna banka Srbije, "NBS IPS QR kôd," *NBS IPS QR kôd*, 2018. https://nbs.rs/QRcode/index_lat.html.
- [5] N. Taveerad and S. Vongpradhip, "Development of Color QR Code for Increasing Capacity," in *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 2015, pp. 645–648, doi: 10.1109/SITIS.2015.42.
- [6] Microsoft, "QRCode," *NuGet Gallery*, 2021. <https://www.nuget.org/packages/QRCode/>.

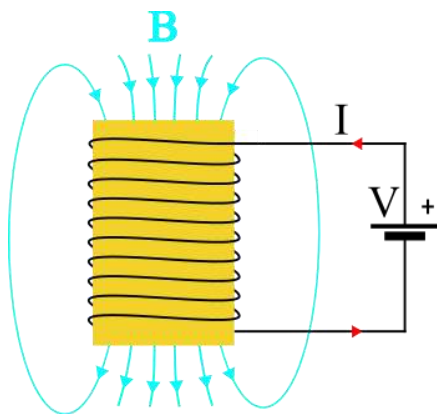
Induktivnost i jačina magnetnog polja višeslojnih cilindričnih kalemova

Milan Stojanović

Sadržaj – U radu su prikazane osnovne osobine i navedene neke od oblasti primena kalemova. Date su jednačine na osnovu kojih se može izračunati induktivnost i jačina magnetnog polja po radijalnoj osi višeslojnih cilindričnih kalemova. Pri teorijskom izračunavanju uzeto je da su broj namotaja i jačina struje koja protiče kroz kalem konstantni. Analizirana je promena navedenih veličina za kalemove različitih dužina i unutrašnjeg poluprečnika. Pokazano je da se pri povećanju unutrašnjeg poluprečnika i konstantnoj dužini kalema induktivnost povećava dok se jačina magnetnog polja smanjuje. U slučaju da je unutrašnji poluprečnik konstantan, a dužina kalema se povećava, i induktivnost i jačina magnetnog polja se smanjuju.

I. UVOD

Kalem je pasivna elektronska komponenta koja u idealnom slučaju poseduje samo reaktivnu otpornost. Vrednost reaktivne otpornosti je direktno proporcionalna frekvenciji signala koji se dovodi na kalem. Induktivnost kalema predstavlja koeficijent proporcionalnosti između otpornosti i frekvencije. Ukoliko se kalem priključi na jednosmerni napon, jednosmerna struja kroz njega se ne uspostavlja trenutno, zbog magnetnog polja koje se u njemu stvara neposredno nakon dovođenja napona. Ovo magnetno polje indukuje elektromotornu silu suprotnog znaka od dovedenog napona, pa samim tim ne dozvoljava trenutno uspostavljanje struje. Nakon nekog vremena, magnetno polje postaje konstantno (Slika 1) i ono dalje ne utiče na proticanje jednosmerne struje.



Slika 1. Magnetno polje koje generiše kalem pri proticanju struje.

M. Stojanović, student doktorskih akademskih studija na modulu Nanotehnologije i mikrosistemi, Elektronski fakultet Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milan.stojanovic@elfak.ni.ac.rs

Odziv kalema na naizmenični signal je nešto drugačiji. Kada je reč o sinusnom signalu, struja kroz kalem, a samim tim i magnetno polje zavise od frekvencije signala [1]. Magnetno polje koje nastaje pri proticanju struje kroz kalem ima brojne primene. U industriji se koristi u elektromagnetnim ventilima, tj. uređajima za aktiviranje hidrauličnih ili pneumatskih ventila. Kalem je deo elektromagneta koji se ugrađuje u mehanizmima za automatsko zaključavanje. Ovakvi sistemi za zaključavanje su pogodni zato što imaju malu potrošnju energije. U medicini se primenjuju u aparatima za dijalizu, za kontrolu protoka krvi. Takođe, imaju značajnu primenu u automobilske industriji, i dr [2]. Još jedan primer primene kalemova je generisanje magnetnog polja, signalom tačno određene frekvencije, za ispitivanje bioloških efekata [3].

Induktivnost kalema zavisi od njegovih geometrijskih parametara, tj. od oblika i dimenzija tela oko koga se formira kalem kao i od broja namotaja i debljine provodnika koji formira kalem. Za dobijanje većih vrednosti induktivnosti proizvode se višeslojni kalemovi koji mogu biti namotani na različite načine. Višeslojni kalemovi motani po pravilnom redu imaju mali faktor dobrote (Q faktor) kao i veliku sopstvenu kapacitivnost, pa se uglavnom primenjuju u radiodifuziji i kao prigušnice. Kako bi se dobila veća vrednost Q faktora i manja vrednost sopstvene kapacitivnosti koriste se unakrsno i nasumično motani višeslojni kalemovi. Uz poboljšanje dva navedena parametra, ovakvim načinom motanja se postiže veća mehanička čvrstoća, čak i u odsustvu tela kalema [4].

II. IZRAČUNAVANJE INDUKTIVNOSTI I JAČINE MAGNETNOG POLJA VIŠESLOJNIH CILINDRIČNIH KALEMOVA

Telo kalema najčešće je cilindričnog oblika. Za izračunavanje induktivnosti cilindričnih kalemova, motanih unakrsno ili nasumično koristi se jednačina (1), koja se u literaturi može naći pod nazivom Vilerov (Wheeler) obrazac [4].

$$L = 78,7 \frac{d^2 N^2}{3d^2 + 9l + 10(r_2 - r_1)} \cdot 10^{-3} \quad (1)$$

U jednačini (1), d predstavlja prečnik kalema i može se izračunati kao $d = r_1 + r_2$, pri čemu je r_2 spoljašnji, a

r_1 unutrašnji poluprečnik kalema. N predstavlja broj namotaja, a l je dužina kalema (dužina provodnog sloja) kao što se može videti na Slici 2. Vrednosti svih veličina su izražene u cm, a vrednost dobijene induktivnosti izražava se u uH. Za izračunavanje jačine magnetnog polja na

radijalnoj osi kalema postoji nekoliko jednačina. Jednačina (2) je jedna od njih [5].

U jednačini (2) μ_0 predstavlja megnetnu permeabilnost vakuuma, a z udaljenost tačke na radijalnoj osi u odnosu na centar kalema.

$$B(z) = \frac{\mu_0 NI}{2l(r_2 - r_1)} \left[\left(\frac{l}{2} - z \right) \ln \left(\frac{r_2 + \sqrt{r_2^2 + \left(\frac{l}{2} - z \right)^2}}{r_1 + \sqrt{r_1^2 + \left(\frac{l}{2} - z \right)^2}} \right) + \left(\frac{l}{2} + z \right) \ln \left(\frac{r_2 + \sqrt{r_2^2 + \left(\frac{l}{2} + z \right)^2}}{r_1 + \sqrt{r_1^2 + \left(\frac{l}{2} + z \right)^2}} \right) \right] \quad (2)$$

Prikazane jednačine se mogu iskoristiti za kalemove bez jezgra, tzv. vazdušne kalemove. Drugim rečima ovi kalemovi nemaju telo oko koga se formiraju namotaji. Međutim, ove jednačine su validne ukoliko su namotaji formirani oko tela kalema čija je relativna magnetna permeabilnost $\mu_r \approx 1$.

od ostalih geometrijskih parametara i može se izračunati koristeći jednačinu (3).

$$r_2 = r_1 + \frac{d_0^2 N}{l} \quad (3)$$

Nakon utvrđivanja veze između svih geometrijskih parametara, može se preći na izračunavanje vrednosti induktivnosti i jačine magnetnog polja.

A. Induktivost višeslojnih cilindričnih kaleмова

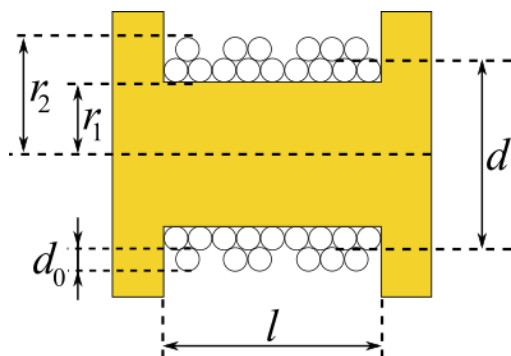
U Tabeli I prikazane su vrednosti geometrijskih parametara kalema koje se koriste za izračunavanje induktivnosti.

TABELA I
GEOMETRIJSKI PARAMETRI KOJI SE KORISTE ZA IZRAČUNAVANJE
INDUKTIVNOSTI VIŠESLOJNOG CILINDRIČNOG KALEMA

l [mm]	$d=r_1+r_2$ [mm]				
	$r_1 = 5$	$r_1 = 10$	$r_1 = 15$	$r_1 = 20$	$r_1 = 25$
120	20	30	40	50	60
150	18	28	38	48	58
200	16	26	36	46	56
240	15	25	35	45	55
300	14	24	34	44	55

Na osnovu vrednosti iz Tabele I, može se zaključiti da, ukoliko je dužina kalema konstantna, prečnik kalema će imati veću vrednost, ako je unutrašnji poluprečnik veći. U drugom slučaju, ukoliko se dužina kalema povećava, a vrednost unutrašnjeg poluprečnika je konstantna, prečnik kalema se smanjuje.

Zavisnost induktivnosti kalema od vrednosti unutrašnjeg poluprečnika, izračunata na osnovu jednačine (1), predstavljena je na Slici 3, a zavisnost induktivnosti kalema od dužine kalema na Slici 4.



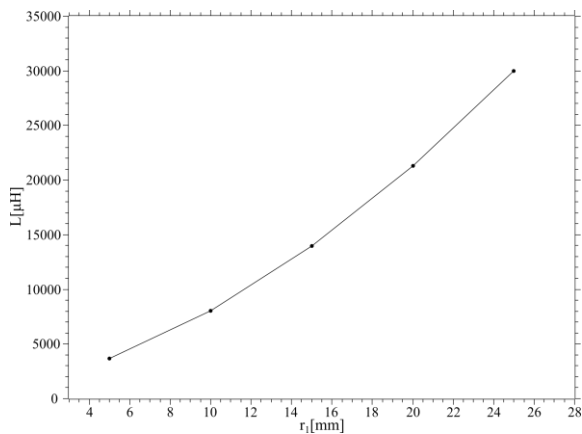
Slika 2. Poprečni presek tela cilindričnog kalema.

III. REZULTATI

Analiza induktivnosti i jačine magnetnog polja višeslojnog kalema urađena je za kalem kog koga je broj namotaja konstantan ($N=1200$), a vrednosti geometrijski parametara kalema se menjaju. Telo kalema je cilindričnog oblika i njegova relativna magnetna permeabilnost je $\mu_r \approx 1$, tako da ne utiče na jačinu magnetnog polja. Drugim rečima, ispituju se osobine vazdušnog kalema.

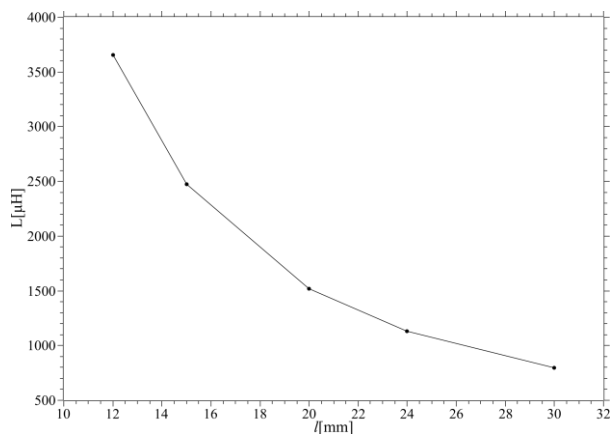
U prvom delu analize, menja se vrednost unutrašnjeg poluprečnika, a dužina kalema je konstantna, a u drugom delu je unutrašnji poluprečnik konstantan, a dužina kalema se menja. Debljina provodnika od koga se formira kalem je $d_0 = 1 \text{ mm}$.

Jedan od geometrijskih parametara kalema je, kako je već rečeno, spoljašnji poluprečnik. Njegova vrednost zavisi



Slika 3. Zavisnost induktivnosti višeslojnog cilindričnog kalema, konstantne dužine sa konstantnim brojem namotaja, od vrednosti unutrašnjeg poluprečnika.

Kao što se može videti, sa porastom unutrašnjeg poluprečnika, induktivnost višeslojnog kalema se povećava ukoliko su broj namotaja i dužina kalema konstantni.



Slika 4. Zavisnost induktivnosti višeslojnog cilindričnog kalema, konstantnog unutrašnjeg poluprečnika sa konstantnim brojem namotaja, od dužine kalema.

U drugom slučaju, kada su unutrašnji poluprečnik i broj namotaja višeslojnog kalema konstantni, induktivnost se smanjuje ukoliko se dužina kalema povećava.

B. Jačina magnetnog polja na radijalnoj osi višeslojnih cilindričnih kaleмова

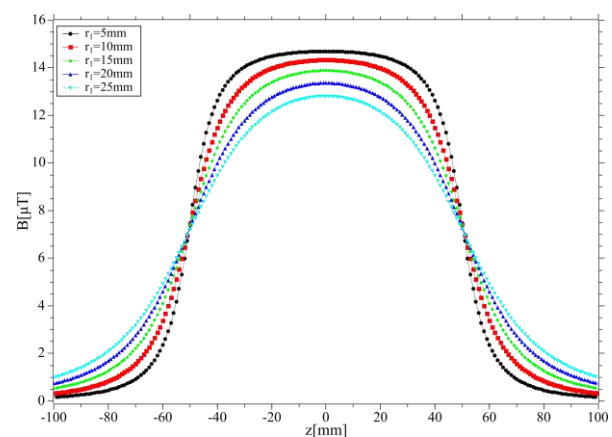
Za izračunavanje jačine magnetnog polja na radijalnoj osi višeslojnog kalema iskorišćene su vrednosti geometrijskih parametara prikazane u Tabeli II, a sama jačina magnetnog polja izračunata je koristeći jednačinu (2). Rezultati koji će biti prikazani dobijeni su pod pretpostavkom da kroz kalem protiče jednosmerna struja jačine $I = 1 \text{ A}$.

TABELA II

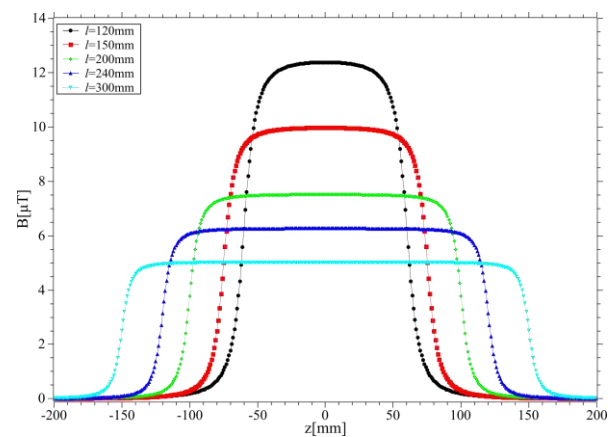
GEOMETRIJSKI PARAMETRI KOJI SE KORISTE ZA IZRAČUNAVANJE JAČINE MAGNETNOG POLJA VIŠESLOJNOG CILINDRIČNOG KALEMA

l [mm]	r_2 [mm]				
	$r_1 = 5$	$r_1 = 10$	$r_1 = 15$	$r_1 = 20$	$r_1 = 25$
120	15	20	25	30	35
150	13	18	23	28	33
200	11	16	21	26	31
240	10	15	20	25	30
300	9	14	19	24	29

Na Slici 5 i Slici 6 su predstavljene promene jačine magnetnog polja koje generiše višeslojni cilindrični kalem po njegovoj radijalnoj osi, pri proticanju struje kroz njega, za različite vrednosti unutrašnjeg poluprečnika i dužine kalema, respektivno.



Slika 5. Jačina magnetnog polja na radijalnoj osi višeslojnog cilindričnog kalema konstantne dužine, pri konstantnom broju namotaja i različitim vrednostima unutrašnjeg poluprečnika.



Slika 6. Jačina magnetnog polja na radijalnoj osi višeslojnog cilindričnog kalema konstantnog unutrašnjeg poluprečnika, pri konstantnom broju namotaja i različitim vrednostima dužine kalema.

Na slikama se može videti da je magnetno polje najjače i konstantno u unutrašnjosti kalema, dok njegov intenzitet van kalema naglo opada.

Kada je dužina kalema konstantna, magnetno polje unutar kalema opada ukoliko se unutrašnji poluprečnik povećava. To se može objasniti time što se namotaji provodnika koji generišu magnetno polje udaljavaju od radijalne ose kalema. Međutim, magnetno polje se van kalema ponaša suprotno i može se videti da je na određenom rastojanju magnetno polje kalema veće, pri većim vrednostima unutrašnjeg poluprečnika.

U drugom slučaju, magnetno polje se smanjuje sa porastom dužine kalema. Pri porastu dužine i fiksnom broju namotaja, debljina namotaja odnosno broj slojeva se smanjuje jer broj namotaja u jednom sloju raste. Upravo smanjenje broja slojeva dovodi do smanjenja magnetnog polja, jer se na taj način zapravo smanjuje zapremina provodne oblasti koja generiše magnetno polje.

IV. ZAKLJUČAK

Konstrukcija kalema zavisi od oblasti primene i zahteva koje treba da zadovolji. Kada je o reč o višeslojnim cilindričnim kalemovima sa konstantnim brojem namotaja, vrednosti induktivnosti i jačine magnetnog polja koje kalemovi generišu po radijalnoj osi se menja sa promenom geometrijskih parametara tela oko koga se formira kalem. Ukoliko se dužina kalema ne menja, sa porastom unutrašnjeg poluprečnika kalema, induktivnost kalema

raste, ali će jačina generisanog magnetnog polja biti manja. U suprotnom, kada se unutrašnji poluprečnik kalema ne menja, i induktivnost kalema i jačina generisanog magnetnog polja po radijalnoj osi će se smanjivati.

ZAHVALNICA

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije, br. 451-03-9/2021-14/200102.

LITERATURA

- [1] C. Wei, X. Youan, Y. Zhiyong, Z. Zhonghao, „Magnetic field analysis of solenoid driven by alternating current”, *Journal of Physics: Conference Series*, 2019, Volume 1237, DOI:10.1088/1742-6596/1237/3/032073.
- [2] Web cite: <http://www.professionaldude.com/?solenoid-applications-in-the-modern-world,9> (pristupljeno 10.10.2021.)
- [3] K. Matsubara, K. Wada, Y. Suzuki, „Design of a coil geometry for generating magnetic field to evaluate biological effects at 85 kHz”, *Electr. Eng. Jpn.* 2018; 205: 55–63. <https://doi.org/10.1002/eej.23117>
- [4] S. Ristić, „Elektronske komponente“, Niš, Elektronski fakultet, 2011.
- [5] M. Basharat, M. Ding, H. Cai, Y. Li, J. Fang, „Design and Analysis of Multilayer Solenoid Coil for Faraday Modulator“, *MATEC Web of Conferences*, 2017 114. 04004. 10.1051/mateconf/201711404004.

„UniMer VIR“- Merna stanica sa osciloskopom, frekvencijskim generatorom i naponskim izvorom

Viktor Savić, Ivan Janka, Robert Lukač Žorž

Sadržaj – U ovom radu je prikazan uređaj koji služi kao univerzalna stanica za merenje mnogih fizičkih veličina u elektrotehnici i drugim oblastima, korišćenjem „Arduino“ mikrokontrolerske hardversko - softverske razvojne platforme za kontrolu rada. Uređaj obuhvata univerzalni instrument, osciloskop, izvore napona, generator funkcije, senzore za merenje ambijentalnih parametara i modul za merenje radioaktivnosti.

I. UVOD

Kompletan uređaj se zasniva na klasičnoj elektronici i mikrokontrolerskom hardversko-softverskom „Arduino“ okruženju. Planirano je da poseduje jedinicu koja prikuplja i skladišti sve podatke (SD modul sa karticom), a postoji mogućnost da ih preko LAN i/ili GSM modula šalje na daljinu drugim uređajima (računar, telefon, tablet i dr.).

Za prikaz svih relevantnih podataka merenja i upravljanja uređajem, koriste se 2.4" Touch LCD, LCD 4h20 (I2C), kao i VGA monitor, koji se koristi za prenos slike. Kao prateći modul uz monitor je i USB tastatura, što zajedno sa 2.4" Touch LCDom čini konzolu za upravljanje uređajem pored moguće aplikacije na pametnim uređajima.

Uređaju se napajanje obezbeđuje pomoću VRLA olovne, akumulatorske baterije napona 12V, kapaciteta 7Ah ili 9Ah. Kao alternativa punjenju akumulatorske baterije preko mreže, ponuđeno je opciono rešenje punjenja preko solarnog panela, kada je to ambijentalno moguće.

II. PRINCIP RADA „UNIMER VIR“ – MERNE STANICE

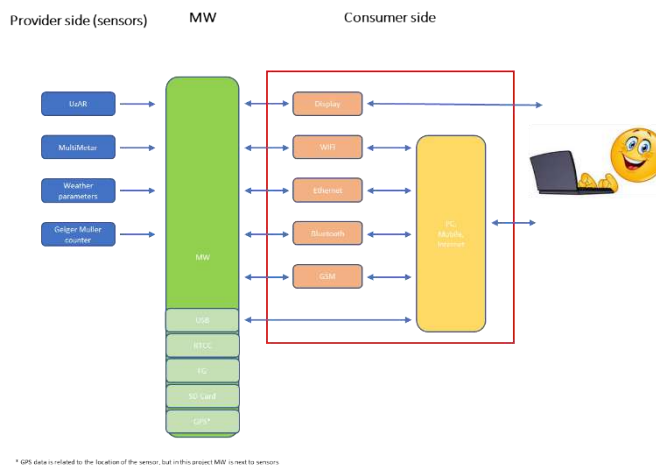
Univerzalna merna stanica „UniMer VIR“ efikasno doprinosi smanjenju broja uređaja za merenje osnovnih veličina u elektrotehnici (otpornosti, kapacitivnosti i induktivnosti; naizmeničnog i jednosmernog napona i struje, faznog pomeraja između napona i struje). Takođe, uključuje i dodatne uređaje: osciloskop, generator funkcija, promenljivi izvor napona, izvor standardnih napona (± 12 V,

± 5 V, ± 15 V, ± 3.3 V), merač temperature/vlažnosti vazduha i drugih ambijentalnih parametara, merač radioaktivnosti i matematičku jedinicu koja vrši razne proračune, na osnovu kojih pravi statistiku.

„UniMer VIR“ obuhvata sve prethodno navedene uređaje u jednu univerzalnu stanicu za merenje fizičkih veličina u oblasti elektrotehnike pod različitim uslovima u objektima i van njih, uključujući ambijentalne parametre i radioaktivnosti okoline.

Arhitektura stanice (Slika 1.) je troslojna i prepoznati su sledeći slojevi:

- sloj provider-a koji obuhvata module i senzore zadužene za pribavljanje podataka,
- srednji sloj (middleware) je zadužen za povezivanje komponenti i pružanje nekih usluga: tačno vreme, generator f-ja, upis na SD-u i GPS podatke o položaju.
- klijentski sloj je zadužen za komunikaciju korisnika sa uređajem. Obuhvata nekoliko modula kojima se mogu preuzeti i prikazati podaci i upravljati senzorima.



Slika 1. Blok šema arhitekture merne stanice.

Da bi se omogućila saradnja modula različitih arhitektura i dizajna, definisane su poruke za razmenu podataka (upit i odgovor), koje moraju biti razumljive svim stranama. Izbor ovakve arhitekture je uslovljen sledećim razlozima:

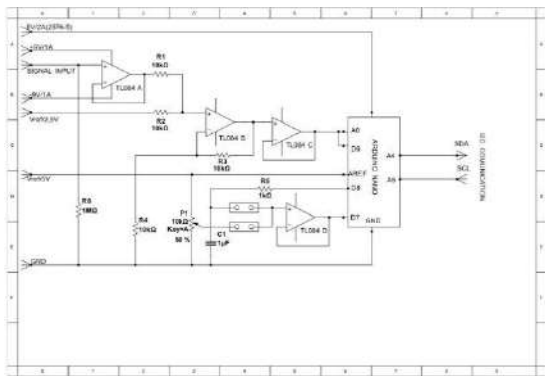
- postoji veliki broj modula, senzora i različitih komunikacionih protokola koje ne može podržati jedno „Arduino“ razvojno okruženje,

V. Savić i R. Lukač Žorž su učenici IV razreda srednje elektrotehničke škole „Mihajlo Pupin“ – Futoška 17, Novi Sad, Srbija, E-mail: viktorsavicprviprojekat.16@gmail.com, robertlukaczoraz@gmail.com

I. Janka je učenik III razreda srednje elektrotehničke škole „Mihajlo Pupin“ – Futoška 17, Novi Sad, Srbija, E-mail: ivanikajanika@gmail.com.

- Funcionalni blokovi „UniMer VIR“-a sa pripadajućim električnim, elektronskim i “Arduino” komponentama su:

Uređaj za Akviziciju Podataka (UzAP) (Slika 2.) je osmišljen kao uređaj koji je zasnovan na „Arduino“ NANO [3] razvojnom okruženju, koji prikuplja očitane vrednosti napona i bez ikakve dalje obrade ih šalje na serijski port u vidu binarnog signala. Ovo znači da je neophodan i klijentski program koji će moći prikupljene podatke da prikaže u nekom smislenom obliku i po potrebi sačuva za dalju obradu i analizu.



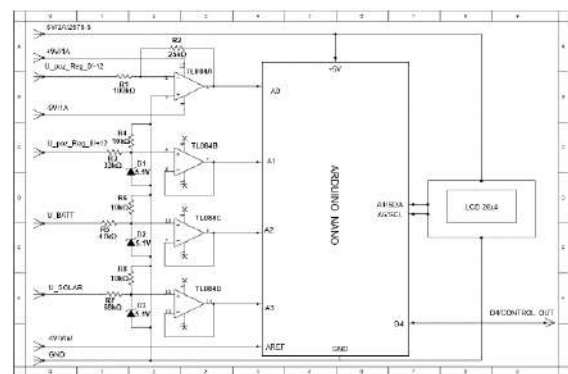
B. Multimeter (Multiple value Meter)

Instrumentaciono pojačalo se sastoji iz dva operaciona pojačavača čiji se ulazi vezuju na neki potencijal i još jednog oduzimajućeg pojačavača koje pravi razliku između

[illegible]

C. Modul za merenje napona

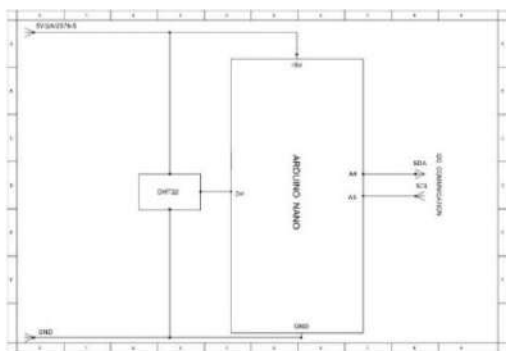
Modul za merenje napona (Slika 4.) služi za merenje naponskih veličina na univerzalnoj stanici čije je vrednosti potrebno u realnom vremenu prikazati na ekranu ili na osnovu kojih odlučiti koju akciju treba obaviti. Za ovaj uređaj se koristi „Arduino“ UNO/NANO [2-3]. okruženje, jer nije potrebno mnogo analognih i digitalnih pinova, a nisu kompleksne operacije. Za prikaz izmerenih vrednosti napona se koristi LCD 20x4 (I2C) [5]. Koristi se integrisano kolo sa četiri operaciona pojačavača, sa velikim ulaznim otporom, TL084. Kao zaštita ulaza kontrolera koriste se zener diode.



D. Meteorološka stanica

Postoji mogućnost da se meteorološka stanica nadogradi dodatnim modulima i senzorima za merenje

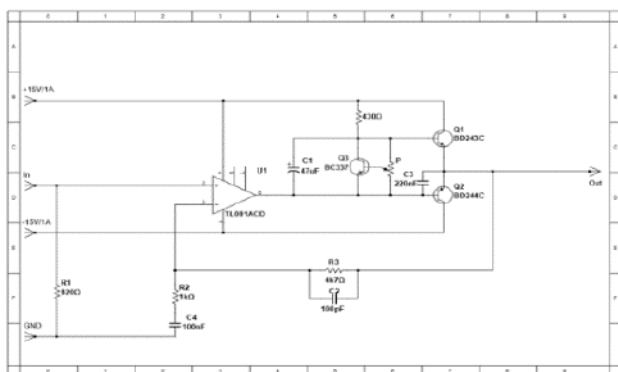
ostalnih ambijentalnih veličina (pritisak, količina padavina, brzina vetra i dr.).



Slika 5. Meteorološka stanica.

E. Pojačavač za generator funkcije – AD9833

Pojačavač za generator funkcije – AD9833, (Slika 6.) je modul koji generiše signale nezavisno od rada kontrolera. Ovom modulu se samo zadaju određene vrednosti u registar koje određuju oblik signala i frekvenciju i one su zapamćene sve dok je uređaj pod naponom. Sam izlazni signala on osciluje oko 2,5 V DC.

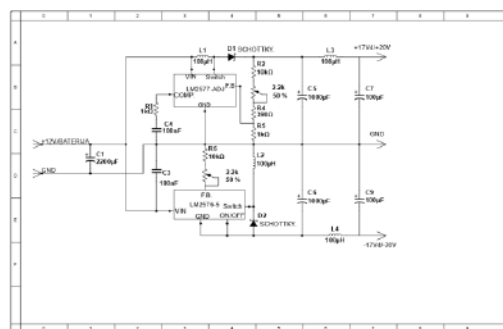


Slika 6. Pojačavač za generator funkcije – AD9833.

Njegova maksimalna i minimalna vrednost iznose $\pm 2,5V$ tj. ukupno 5 Vpp. Izlazna struja generatora je mala pa se na izlaz mora staviti pojačavač napona, struje i snage. Sam generator se nalazi na srednjem, sloju.

F. Negativan i pozitivan BOOST

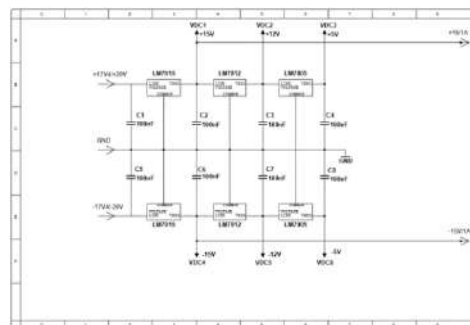
Negativan i pozitivan BOOST pretvarač sa IC 2576 [13] i 2577 [12] kolima prikazan je na slici 7. Univerzalna stanica se napaja iz VRLA AGM olovne baterije, čiji je nominalni napon 12 V. Princip rada elektronskih sklopova ove vrste je da zahtevaju pozitivno i negativno napajanje tzv. simetrično napajanje. Potrebni su stabilni naponi od +18 V i -18 V. Oba napona se dobijaju pomoću prekidačkih pretvarača, tj. pošto je reč o višim naponima po apsolutnoj vrednosti u odnosu na napon baterije, pomoću tzv. BOOST pretvarača.



Slika 7. Negativan i pozitivan BOOST pretvarač.

G. 78xx i 79xx serije stabilizatora napona

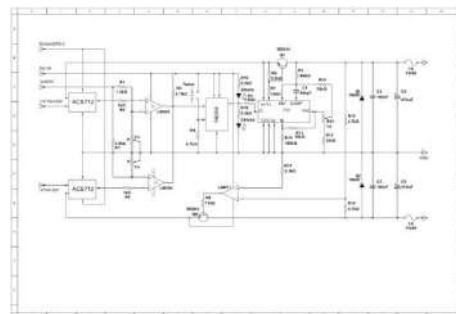
78xx i 79xx serije stabilizatora napona prikazane su na Slici 8, gde pored promenljivog izvora pozitivnog i negativnog napona (0 do $\pm 12V$), uređaj poseduje i neke standardno zadate izvore napona. 78XX serija stabilizatora se koristi za stabilizaciju pozitivnog napona, dok se 79XX serija stabilizatora koristi za stabilizaciju negativnog napona.



Slika 8. 78xx i 79xx serije stabilizator napona.

H. Linearni stabilizator pozitivnog napona

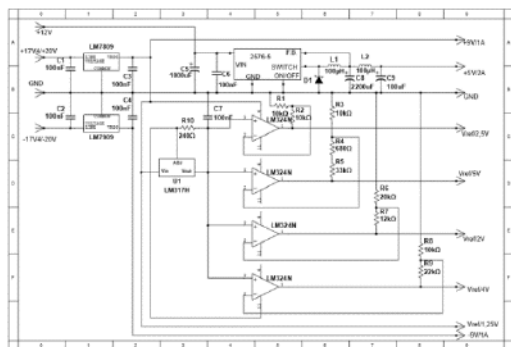
U promenljivom izvoru napona linearni stabilizator pozitivnog napona (Slika 9.) je realizovan sa IC LM723 [14]. Ovo IC kolo sadrži temperaturno kompenzovan izvor referentnog napona tipično 7.15 V, operacioni pojačavač za povratnu spregu i izlazni stepen koji pogoni bipolarni tranzistor snage.



Slika 9. Promenljivi izvor napona.

I. Jedinica za napajanje kontrolera

I. Jedinica za napajanje kontrolera (Slika 10.), modula i izvor referentnih napona je posebna jedinica u kojoj se nalaze samo izvori napajanja potrebni za navedene sklopove, kao i stabilni izvori referentnih napona (2 V, 2,5 V, 4 V, 5 V) za merenja. 7809 i 7909 stabilizatori napona se koriste za napajanje operacionih pojačavača.

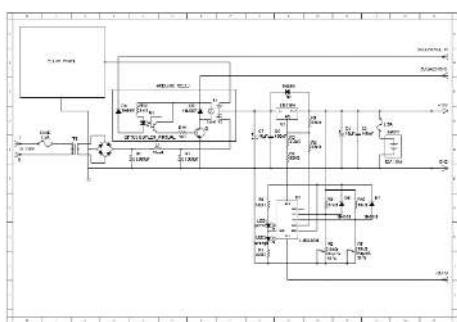


Slika 10. Jedinica za napajanje kontrolera.

BUCK pretvarač sa 12 V na 5 V se koristi za napajanje svih kontrolera i njihovih pratećih modula. Za BUCK pretvarač se koristi već gore navedeno IC LM2576 [13]. Za razliku od gornje konfiguracije u kojoj se koristi kao BOOST pretvarač za negativan napon, na ovoj šemi je zasnovan na njegovoj osnovnoj šemi BUCK pretvarača.

J. Punjač baterije

Punjač baterije koristeći mrežu ima torusni transformator 50 Hz, sa Grecovim ispravljačem i elektrolitskim kondenzatorom velike kapacitivnosti prikazan je na Slici 11. Na izlaz pored kondenzatora može da se doda još jedno LC kolo za umanjeno brujanje. Opciono rešenje je solarni panel, a izbor između dva izvora za punjenje baterije vrši kontroler koji meri napon na njemu u realnom vremenu.



Slika 11. Punjač baterije.

III. ZAKLJUČAK

Potencijalni korisnici „UniMer VIR“-a su: u raznim laboratorijama preduzeća/ institucija, kabinetima u školama

i fakultetima, istraživačkim /naučnim institutima, servisima za dijagnostiku, popravku i reparaciji električnih - elektronskih uređaja i aparata. Upotrebljiv je od strane svih onih kojima je potreban jedan uređaj za merenje raznorodnih veličina, šireg spektra u raznim ambijentalnim uslovima i na različitim lokacijama. Opšte karakteristike uređaja su: preciznost u merenju, jednostavna kontrola vrste i opsega merenja, uključen prateći pribor, lagan i malih dimenzija, pristupačne cene za datu funkcionalnost. Od planiranog trenutno su razvijeni sledeći moduli: VGA klijent sa USB/PS2 tastaturom, LCD Touch displej, Ethernet klijent koji će pristupiti serverskoj aplikaciji ili HTML strani na kojoj će prikazivati date podatke, kao i servrski moduli, meteorološka stanica i Uređaj za akviziciju podataka koji čini deo jedinice osciloskopa. Ostali moduli su trenutno u razvoju. U nekoj kasnijoj fazi razvoja projekta, predviđeno je da se za VGA klijenta sa USB/PS2 tastaturom umesto „Arduino“ MEGA kontrolera koriste „Raspberry“ Pi mikroracunar.

ZAHVALNICA

Autori se ovim putem zahvaljuju učenicima Nikoli Stefanoviću (R31) i Stefanu Karaću (IT31) na pomoći pri uređivanju rada i Novici Gutoviću (mentoru projekta), rukovodiocu sekcije „Arduino“ u ETŠ „Mihajlo Pupin“-Novi Sad, na tehničkoj podršci i veoma uspešnoj saradnji.

LITERATURA

- [1] <https://store.arduino.cc/arduino-mega-2560-rev3>
- [2] <https://store.arduino.cc/arduino-uno-rev3>
- [3] <https://store.arduino.cc/arduino-nano-every>
- [4] <https://fritzing.org/projects/dht22-temperaturehumidity>
- [5] <https://create.arduino.cc/projecthub/electropeak/arduino-2-4-touch-screen-lcd-shield-tutorial-fe6f05>
- [5] <https://www.instructables.com/How-to-Use-I2C-Serial-LCD-20X4-Yellow-Backlight/>
- [6] <https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390>
- [7] <https://lastminuteengineers.com/a6-gsm-gprs-module-arduino-tutorial/>
- [8] https://create.arduino.cc/projecthub/Niv_the_anonymous/esp8266-beginner-tutorial-project-6414c8
- [9] <https://www.instructables.com/Arduino-Webserver-Controller-Using-Ethernet-ENC28J/>
- [10] <https://datasheetspdf.com/pdf/570845/AllegroMicroSystems/ACS712/1>
- [11] https://en.wikipedia.org/wiki/Buck_converter
- [12] <https://www.ti.com/lit/ds/symlink/lm2577.pdf>
- [13] <https://www.ti.com/lit/ds/symlink/lm2576.pdf>
- [14] <https://www.ti.com/lit/ds/symlink/lm723.pdf>
- [15] https://en.wikipedia.org/wiki/Instrumentation_amplifier
- [16] <https://www.instructables.com/Girino-Fast-Arduino-Oscilloscope>
- [17] <http://lig-membres.imag.fr/krakowia/Files/MW-Book/Chapters/Intro/intro.html>
- [18] <https://github.com/smaffer/vgax>
- [19] <https://www.instructables.com/Connect-Arduino-UNO-to-USB-Keyboard>

Upotreba 12-bitnog digitalno-analognog konvertora MCP4725 u embeded uređajima

Dušan Jovanović, Nikola Stojiljković

Sadržaj - U okviru ovog projekta prikazana je upotreba 12-bitnog DAC konvertora MCP4725 kontrolisanog korišćenjem I^2C komunikacije za generisanje izlaznog analognog signala u opsegu od 0-5V sinusnog i trougaonog oblika, kao i za posmatranje protoka paketa duž magistrale.

I. UVOD

Kola čija je osnovna funkcija konverzija digitalnog u analogni signal nazivaju se digitalno-analogni konvertori (u daljem tekstu skraćeno DAC). Svaki DAC okarakterisan je pomoću više parametara gde su najznačajniji rezolucija i brzina uzorkovanja [1]. Rezolucija DAC zapravo predstavlja broj mogućih različitih analognih nivoa koje D/A konvertor može proizvesti u zadatom opsegu vrednosti. Ako se pretpostavi da svakom binarnom stanju u digitalnom domenu odgovara jedan vrednost izlazne analogne veličine, onda se rezolucija može izraziti i kao broj bita m DAC konvertora jer je tada broj mogućih stanja 2^m . Zbog složenosti i potrebe za precizno usklađenim komponentama, većina DAC-a su implementirani kao integrisana kola (IC). MCP4725 predstavlja jedno takvo integrisano kolo sa 12-bitnom rezolucijom [2].

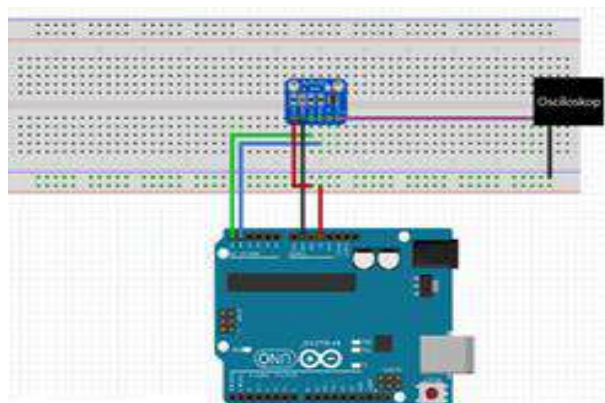
Prenos podataka preko magistrala (žica), tj. komunikacija može biti paralelna ili serijska. U paralelnoj komunikaciji prenosi se veliki broj podataka istovremeno. Najčešće preko 8, 16 ili 32 linija. Iako je ovaj tip komunikacije brz, zahteva veliki broj konekcija, pa se zato uglavnom sreće unutar integrisanih kola. Za povezivanje uređaja uglavnom se koristi serijska komunikacija gde se podaci prenose bit po bit. Brzina prenosa podataka je manja nego kod paralelnog prenosa, ali se koristi manji broj linija veze (1-4). Da bi se postigla sinhronizacija u prenosu podataka jedna od linija na magistrali je takt (*clock*). To su sinhroni serijske magistrale. Postoje i asinhroni serijske magistrale, gde nema prenosa taktnog signala, već sami uređaji koji komuniciraju imaju integrisana kola za sinhronizaciju. I^2C (*Integrated-Integrated Circuit*) je sinhroni serijski *half-duplex* princip komunikacije realizovan sa dve veze. Koristi se za povezivanje periferija na mikrokontrolere i funkcioniše po principu komunikacije između *master* uređaja i *slave* uređaja. I^2C magistrala

D. Jovanović i N. Stojiljković su studenti četvrte godine na modulu za Elektronske komponente i mikrosisteme Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija
E-mail: dusann.jovanovic@elfak.rs, stojiljkovicn@elfak.rs

dozvoljava postojanje više *master* uređaja i više *slave* uređaja na jednoj magistrali. Master uređaj je glavni jer diktira učestanost signala takta. Za prenos podataka koriste se dve veze: SCL (*Serial Clock*) koja prenosi taktni signal i SDA (*Serial Data*) koja prenosi podatke [3].

II. ELEKTRIČNA ŠEMA I ELEMENTI KOLA

Prvi korak pri realizaciji projekta bio je kreiranje šeme u okviru programskog paketa *Fritzing*. Na slici 1 je ta ista prikazana.



Slika 1. Električna šema kola sa elementima u *fritzing*-u.

A. Arduino UNO

Arduino Uno je štampana ploča sa mikrokontrolerom zasnovana na ATmega328 mikrokontroleru (prikazana na slici 2). Ima 14 digitalnih ulaza/izlaza (od kojih se 6 mogu koristiti kao PWM izlazi), 6 analognih ulaza, kristalni oscilator od 16 MHz, USB vezu, priključak za napajanje, ICSP konektore za programiranje i taster za resetovanje. Sadrži sve što je potrebno za podršku mikrokontroleru; jednostavno se povezuje na računar pomoću USB kabla, preko koga se i napaja. Napajanje može biti i DC adapterom ili baterijom[4].



Slika 2. Arduino Uno Rev3.

B. MCP4725

MCP4725 je kolo male snage, visoke preciznosti, u vidu 12-bitnog DAC-a sa trajnom memorijom u sebi (EEPROM). Samo kolo poseduje eksterni pin A0 za selekciju adrese korišćene za I²C komunikaciju. U zavisnosti od toga kako je ovaj pin povezan, da li je vezan za VDD ili VSS promeniće se i sama adresa. Poseduje I²C interfejs sa dve veze sa različitim brzinama prenosa podataka: standardni način prenosa (100 kHz), brzi način prenosa (400 kHz) ili super brzi način prenosa podataka (3.4 Mbs). Modul je prikazan na slici 3 [5].



Slika 3. Modul sa integrisanim kolom MCP4725.

III. PRINCIP RADA

Sam projekat je koncipiran po sledećem principu: Izvršeno je povezivanje samog modula MCP4725 sa Arduino Uno R3 mikrokontrolerom. Modul je priključen na napajanje od 5V koje potiče od Arduino okruženja, gde je unapred izvršeno povezivanje eksternog A0 pina za selekciju adrese, tako da je adresa samog modula fiksno 0x60. Takođe, na šemi nisu upotrebljeni *pull-up* otpornici jer su oni već implementirani na samom modulu. SCL i SDA pinovi sa modula su povezani direktno sa SCL i SDA pinovima na Arduino okruženju. Izlazni priključak modula se povezuje za pozitivan kraj sonde osciloskopa, dok je negativan kraj sonde osciloskopa vezan za masu.

Priključivanjem Arduino okruženja preko USB porta, i *upload*-ovanjem samog koda, na ekranu osciloskopa se dobijaju određeni oblici signala zavisno od toga koji je kod *upload*-ovan na Arduino okruženju.

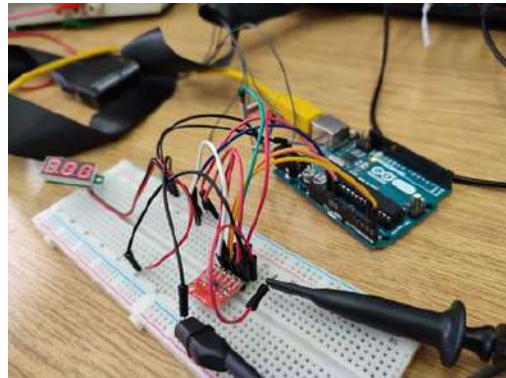
Za trougaoni signal princip funkcionisanja samog programa bazira se na postojanju brojača čija se vrednost inkrementira od 0 do 4095 (gde je 0 – 0V a 4095 – maksimalna vrednost na izlazu samog DAC-a koja u ovom slučaju iznosi 5V) pri čemu će se inkrementiranjem vrednosti stvoriti uzlazni *ramp* signal, a zatim nakon izlaska iz date petlje, ponoviće se isti postupak samo u suprotnom smeru. Brojač počinje od 4095, tačnije od maksimalne vrednosti, i spušta se ka 0, čime se dobija silazni *ramp* signal. Na taj način dolazi do generisanja trougaonog signala na ekranu osciloskopa.

Za generisanje signala sinusnog oblika korišćene su *look-up* tabele, za različite rezolucije koje mogu da se odaberu, počevši od najniže kojom bi se dobio signal stepeničastog oblika, pa sve do najviše rezolucije kojom je moguće dobiti signal pravilnog sinusnog oblika. Pored prikazivanja signala sinusnog i trougaonog oblika,

nad ovim modulom je i izvršena upotreba logičkog analizatora RIGOL MSO5104 [6].

Sam logički analizator, kao uređaj namenjen za posmatranje digitalnih signala sa jako velikim brojem kanala (za razliku od osciloskopa koji ima 2-4 logički analizator može imati i 80), omogućio je vizuelizaciju prenosa podataka preko linije za podatke u okviru I²C komunikacionog protokola. Sama logička analiza izvršena je nad dva signala, jednog amplitude 2V i drugog amplitude 1V koji su se smenjivali u periodu od 2 sekunde.

Na slici 4 prikazana je praktična realizacija u okviru laboratorije:



Slika 4. Praktična realizacija projekta.

IV. PROGRAMSKI KOD

Kod za dobijanje signala sinusnog oblika je preobimam da bi bio prikazan u okviru ovog rada. Za detaljnije posmatranjesamog koda, može se pristupiti pomoću linka[7]. Sledi programski kod za signal trougaonog oblika, koji je prikazan na slici 5. Još jedna stvar koja je zanimljiva da se uoči jeste kako se šalju i primaju podaci korišćenjem I²C komunikacije, programski kod za taj deo prikazan je na slici 6.

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

void setup(void) {
  Serial.begin(9600);
  dac.begin(0x60);
}

void loop(void) {
  uint32_t brojac;

  for (brojac = 0; brojac < 4095; brojac++)
  {
    dac.setVoltage(brojac, false);
  }
  for (brojac = 4095; brojac > 0; brojac--)
  {
    dac.setVoltage(brojac, false);
  }
}
```

Slika 5. Arduino kod za signal trougaonog oblika.

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

void setup(void) {
  Serial.begin(9600);
  dac.begin(0x60);
}

void loop(void) {
  uint32_t brojac;

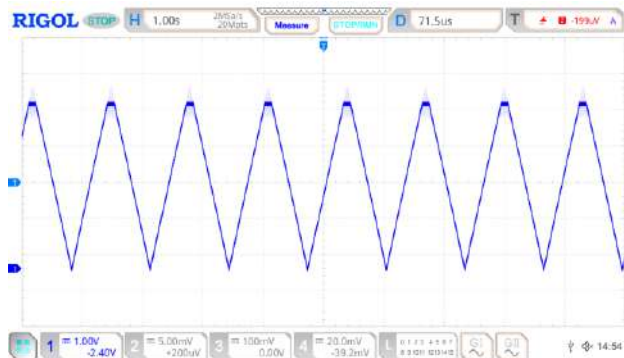
  dac.setVoltage(2000, false);
  delay(2000);
  dac.setVoltage(1000, false);
  delay(2000);
}
```

Slika 6. Arduino kod za logičku analizu.

V. REZULTATI MERENJA

Nakon povezivanja samog modula sa Arduino okruženjem i priključivanja osciloskopa, dobijeni su sledeći rezultati:

Na slici 7, prikazanoj ispod, prikazan je signal trougaonog oblika dobijen upotrebom modula MCP4725 kada *counter* ide do svoje maksimalne vrednosti od 4095. Kao što je već naznačeno, MCP4725 predstavlja 12-bitni digitalno-analogni konvertor koji ima ukupnu rezoluciju od 4096 naponskih nivoa. Zbog nesavršenosti samog modula, uočeno je da pri vrednostima *counter-a* iznad 4067, dolazi do odsecanja trougaonog signala. To praktično znači da dolazi do gubitka 28 naponskih nivoa. Kako bi bio prikazan najverodostojniji trougaoni signal, vrednost brojača je smanjena.

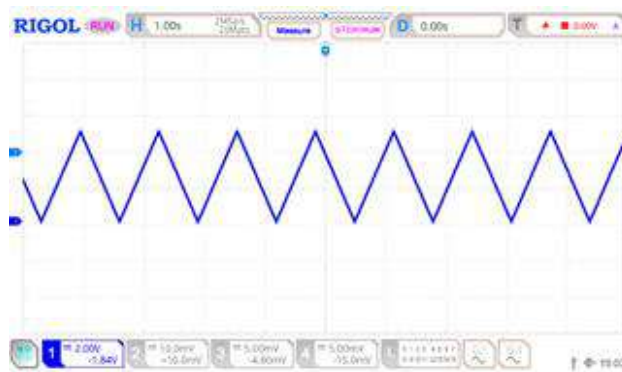


Slika 7. Trougaoni signal na izlazu modula sa odsečenim vrhom.

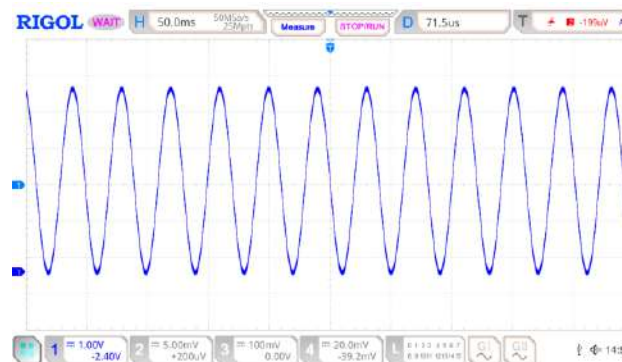
Na slici 8, prikazanoj ispod, prikazan je signal trougaonog oblika dobijen upotrebom modula MCP4725 nakon smanjenja vrednosti *counter-a*.

Drugi signal koji smo pratili na osciloskopu, jeste signal sinusnog oblika. Dobijeni rezultati su prikazani na slici 8.

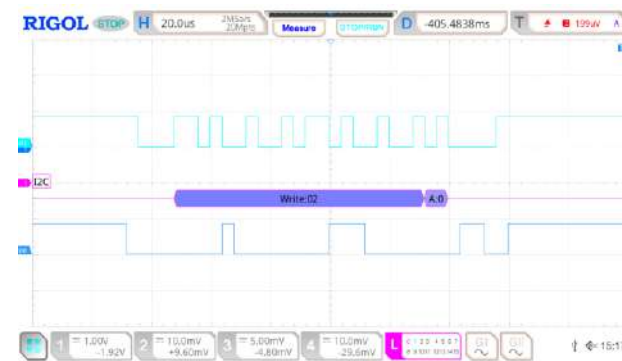
Poslednji deo predstavlja posmatranje toka podataka uz pomoć logičkog analizatora. Dobijeni rezultati su prikazani na sledećoj slici, slici 9.



Slika 8. Trougaoni signal na izlazu modula.



Slika 8. Sinusni signal na izlazu modula.



Slika 9. Dobijeni rezultati uz pomoć logičkog analizatora.

Logički analizator je odličan alat koji se koristi za verifikaciju i debugiranje digitalnih kola. Sam logički analizator može prikazati više različitih signala odjednom i analizirati njihov odnos u vremenu. Na slici 6 prikazan je kod koji je korišćen. Samim kodom definisano je generisanje dva različita signala na izlazu konvertora. Kao što je već rečeno, MCP4725 sa Arduino okruženjem komunicira pomoću I^2C komunikacionog protokola.

Logičkim analizatorom je omogućeno posmatranje I^2C write operacije ili I^2C read operacije između Arduino okruženja i MCP4725 digitalno-analognog konvertora u bilo kom trenutku.

VI. ZAKLJUČAK

Na osnovu rada sa modulom MCP4725, došli smo do zaključka da jedno ovako jednostavno kolo sa malim *footprint*-om može vrlo lepo poslužiti kao DAC konvertor u cilju generisanja signala željenog oblika. Vrlo je jednostavno povezati kolo u kome će se ovakav jedan modul upotrebiti, jer ne zahteva nikakve dodatne komponente osim provodnika putem kojih će na odgovarajući način biti povezan sa mikrokontrolerom.

Jedan od problema sa upotrebom ovog kola jeste činjenica što je referentni napon koji ovo kolo koristi sam napon napajanja koji ima svoju neku toleranciju određenu od strane samog proizvođača, koja u manjoj ili većoj meri može uticati na dobijeni izlazni signal. [8]

ZAHVALNICA

Autori se zahvaljuju Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, a posebno

asistentu Nikoli Mitroviću na pruženoj pomoći i saradnji, tokom realizacije projekta.

LITERATURA

- [1] <http://ww1.microchip.com/downloads/en/devicedoc/22039d.pdf>
- [2] Robert Oshana, in DSP Software Development Techniques for Embedded and Real-Time Systems, 2006
- [3] Miloš Marjanović, Modeliranje i simulacija mikroelektronskih komponenata i kola, zbirka rešenih arduino projekata, 2020
- [4] ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [5] ww1.microchip.com/downloads/en/devicedoc/22039d.pdf
- [6] www.butterfly.com/PDF/RIGOL/mso5000/MSO5000_UserGuide_EN.pdf
- [7] <https://create.arduino.cc/editor/dusanjovanovic1/ae49c29f-b455-4950-a9df-2ffad2146228/preview>
- [8] www.bestmicrocontrollerprojects.com/mcp4725.html

Sistem zaključavanja korišćenjem daljinskog upravljača realizovan u Arduino okruženju

Nemanja Nedeljković

Sadržaj – U ovom radu je realizovan sistem zaključavanja koji se sastoji od daljinskog upravljača (IR receiver), dok je za ispis šifre korišćen LCD displej. Rad upravljača uz LCD displej su omogućeni preko Arduino Uno okruženja. Detaljan opis korišćenih komponenata i demonstracija primene će biti detaljnije objašnjeni u nastavku rada.

I. UVOD

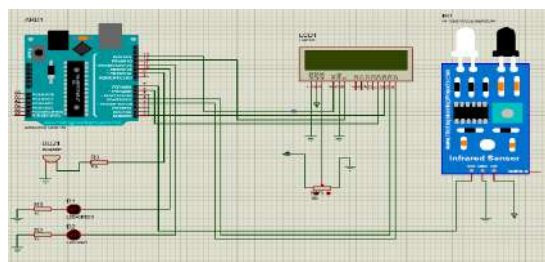
Naši preci su razvili potrebu da svoje stvari zadrže za sebe koristeći mehaničke uređaje poznate kao brave [1]. U početku su te brave bile samo jednostavni čvorovi napravljeni od užadi ili drugih materijala (koristili su se samo za otkrivanje pokušaja da ih neko otvori), ali kako je vreme prolazilo i razvijale se nove tehnologije, pojavile su se prave brave od drveta i metala. Savremeni istoričari nisu sigurni koja je drevna civilizacija prva stvorila mehaničke brave, a mnogi veruju da su ih Egipćani, dok su Grci i Rimljani razvili nezavisno jedan od drugog. Istorija mehaničkih brava započela je pre više od 6 hiljada godina u starom Egiptu, gde je bravar prvi put uspeo da stvori jednostavnu, ali efikasnu bravu sa iglama koja je u potpunosti napravljena od drveta. Tokom prvog milenijuma pre nove ere, brave su konačno počele da se poboljšavaju sa tehnologijama i dizajnom, uvođenjem metala kao primarnog materijala. Koristeći gvozdene brave, Rimljani su konačno uspeali ne samo da imaju vrlo jaku zaštitu od napada grubom silom, već su i ključevi po prvi put bili mali da bi se mogli nositi u džepovima. Napredak je došao u 18. veku, kada je tehnološki napredak omogućio inženjerima da stvore male i čvrste mehanizme. Ovaj novi talas inovacije zaključavanja predvođen je pronalascima Roberta Barronina, Joseph Bramaha i drugih.

Savremeni sistemi kontrole zaključavanja zasnivaju se na skeniranju otisaka prstiju, glasovne identifikacije, prepoznavanju lica i radio-frekvencijske identifikacije (RFID). U ovom radu biće predstavljen jedan od savremenih sistema zaključavanja zasnovan na daljinskom upravljaču. Ideja projekta je da se unosom odgovarajuće šifre omogućava otključavanje brave, što se signalizira zelenom LED diodom i zujalicom. Šifra se unosi putem daljinskog upravljača i pritiskom na odgovarajuće dugme se aktivira poređenje sa inicijalno unetom šifrom.

N. Nedeljković je student IV godine studija na studijskom programu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: neca.nedeljkovic@gmail.com

II. ELEKTRIČNA ŠEMA I KOMPONENTE SISTEMA

Pre praktične realizacije urađena je simulacija projekta u programskom paketu Proteus. Električna šema data je na slici 1. U ovom projektu Arduino je glavni deo sistema. *Arduino Uno* je ploča mikrokontrolera zasnovana na *ATmega328* mikrokontroleru [2]. Infracrveni (IC) prijemnik je komponenta prikazana na slici 2. Infracrveni prijemnik ima tri pina: prvi pin je Vout, drugi pin je GND, treći pin je Vcc. Kada se pritisne daljinski upravljač, šalje se infracrveni modulirani signal. Ovaj signal sadrži informacije o izdatoj komandi koju IC prijemnik prikuplja. Korišćen je LCD displej 16x2 za prikaz odgovarajućih poruka. Za podešavanje kontrasta LCD-a korišćen je potenciometar. LED su u ovom projektu iskorišćene za indikaciju rada sistema zaključavanja, sa odgovarajućim otpornicima za ograničenje struje. Zujalica je iskorišćena za zvučnu signalizaciju stanja sistema zaključavanja.



Slika 1. Električna šema urađena u programskom paketu Proteus.

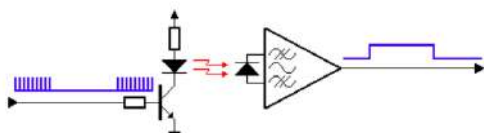


Slika 2. Izgled IC daljinskog upravljača i prijemnika HX1838.

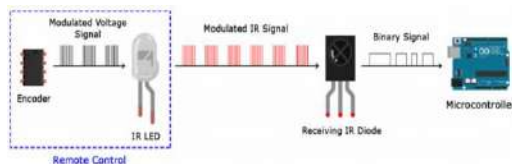
III. PRINCIP RADA SISTEMA

IC daljinski upravljač i prijemnik međusobno komuniciraju prenošenjem i dekodiranjem signala u obliku impulsnog IC zračenja (slika 3). Infracrveno zračenje (IC) je vrsta elektromagnetnog zračenja sa talasnim dužinama od 700 nm do 1 mm. Pošto ljudi mogu videti svetlost samo sa talasnim dužinama od 400 nm (ljubičastih) do 700 nm (crvenih), IC zračenje je nevidljivo za ljudsko oko. Za bežični prenos signala između daljinskog upravljača i

prijemnika potrebna je jasna linija vidljivosti, signal se ne može prenositi kroz zidove. IC LED dioda na daljinskom upravljaču nije jedini izvor IC zračenja. Svaki objekat koji ima temperaturu zrači i u infracrvenom spektru. Ambijentalno IC zračenje može ometati komunikaciju između daljinskog upravljača i prijemnika. Da bi se omogućio nesmetani prijem signala sa daljinskog upravljača koristi se modulacija signala. Kod IC modulacije signala, koder na IC daljinskom upravljaču pretvara binarni signal u modulirani električni signal. Ovaj električni signal se šalje na predajnu LED diodu. Prenosna LED pretvara modulirani električni signal u modulirani IC svetlosni signal. IC prijemnik zatim demodulira IC svetlosni signal i pretvara ga nazad u binarni pre nego što informacije prosledi mikrokontroleru (slika 4). Modulirani IC signal je povorka svetlosnih impulsa određene, noseće, frekvencija. Noseća frekvencija koju koristi većina predajnika je 38 kHz, jer je u prirodi retka i stoga se može razlikovati od ambijentalne buke. Na ovaj način će IC prijemnik znati da je signal od 38 kHz poslat sa predajnika i da nije prikupljen iz okruženja. Prijemna dioda detektuje sve frekvencije IC svetlosti, ali IC modul ima filter opsega frekvencija i propušta samo IC na 38 kHz. Zatim se pojačava modulirani signal pomoću predpojačavača i pretvara se u binarni signal pre nego što bude poslat u mikrokontroler.

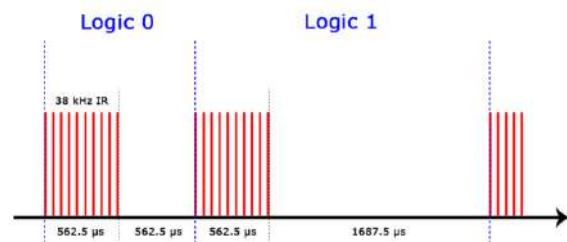


Slika 3. Poslati i detektovani signal IC predajnikom (levo) i prijemnikom (desno).



Slika 4. Ilustracija prenosa modulisanog IC signala.

Obrazac po kome se modulirani IC signal pretvara u binarni definisan je protokolom prenosa. Postoji mnogo protokola za IC prenos: Sony, Matsushita, RC5... U ovom radu korišćen je NEC protokol (slika 5). Logička "1" počinje sa 562.5 μ s dugim HIGH impulsom od 38 kHz, nakon čega sledi 1687.5 μ s dugačak LOW impuls. Logička "0" se prenosi sa 562.5 μ s dugim HIGH impulsom, nakon čega sledi 562.5 μ s dugački LOW impuls. Ostali protokoli se razlikuju samo po trajanju pojedinačnih HIGH i LOW impulsa. Svaki put kada se pritisne dugme na daljinskom upravljaču, generiše se jedinstveni heksadecimalni kod. Ovo su informacije koje se modulišu i šalju preko IC predajnika. Da bi dešifrovao koji taster je pritisnut, prijemni mikrokontroler mora znati koji kod odgovara svakom tasteru na daljinskom upravljaču.



Slika 5. Ilustracija NEC protokola.

IV. PROGRAMSKI KOD

Dati su delovi koda za čitanje poruka koje stižu na IC prijemnik sa daljinskog upravljača i ispis određene šifre na LCD displeju. Za rad sa funkcijama za upravljenje i kontrolu LCD displeja koristi se biblioteka *LiquidCrystal*, koja se uključuje preko header fajla na početku koda. Za rad sa IC prijemnikom i daljinskim upravljačem koristi se biblioteka *IRremote*. Izlaz IC prijemnika povezan je na pin 7 Arduino okruženja, za šta se u kodu koristi labela RECV_PIN. Nakon toga kreira se objekat klase IRrecv i povezuje se sa izlaznim pinom prijemnika (pin 7). U ovom slučaju, *irrecv* objekat upravlja o IC protokolu i obradi signala prijemnika. Zatim se kreira objekat nazvan *results* klase *decode_results*. Ovaj objekat skladišti dekodirane vrednosti iz objekta prijemnika. Promenljiva *key_value* sa početnom vrednošću 0 se koristi za skladištenje vrednosti ključa. Takođe se kreira LCD objekat, imenovana instanca za rad sa LCD bibliotekom, u ovom projektu lcd i deklarišu se pinovi na koje je povezan [5].

```
#include <IRremote.h>
#include <LiquidCrystal.h>
const int RECV_PIN = 7;
LiquidCrystal lcd(13, 12, 6, 5, 4, 3);
IRrecv irrecv(RECV_PIN);
decode_results results;
unsigned long key_value = 0;
```

Definisani su pinovi na koje su povezani zupjalica, crvena i zelena LED. U promenljivu *sifra* je upisana šifra koja otključava bravu. Promenljive *primljen_podatak** prihvataju pristigle podatke s upravljača. String *primljena_sifra* je niz prethodno uneta 4 karaktera. Promenljiva *fleg* je pomoćna logička promenljiva.

```
#define buzzer 9
#define redLed 11
#define greenLed 10
String sifra = " 1 2 3 4 ";
String primljen_podatak;
String primljen_podatak1;
...
String primljena_sifra;
int fleg = 0;
```

U okviru funkcije *void setup()* koja se izvršava samo jednom, postavljena je brzina prenosa podataka za serijsku komunikaciju na 9600. Zatim se inicijalizuje prijemnik

funkcijom `enableIRIn()`. Nakon toga definisani su pinovi LED i zujalice kao izlazni korišćenjem funkcije `pinMode()`. Takođe, inicijalizovan je interfejs LCD-a i specificirane su dimenzije displeja (16x2), korišćenjem funkcije `lcd.begin(16,2)`. Osim toga prethodno deklarirani podaci `primljen_podatak*` su ovde postavljeni da budu prazni.

```
void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();
  lcd.begin(16, 2);
  pinMode(tasterPin, INPUT);
  pinMode(redLed, OUTPUT);
  ...
  primljen_podatak1 = "prazan";
  primljen_podatak2 = "prazan";
  ...
  lcd.print("Unesite sifru:");
}
```

Funkcija `void loop()` predstavlja beskonačnu petlju i ovde je napisan korisnički deo programa. Pri svakom prolasku kroz beskonačnu petlju prvo se proverava da li su svi primljeni podaci "prazni". Nakon toga kada se primi kod sa upravljača, `irrecv.decode(&results)` vraća vrednost `true` i ostatak koda se izvršava. Kada se neprestano pritiska taster, sa upravljača stiže `0xFFFFFFFF`, pa se ta vrednost zamenjuje sa prethodnom ključnom vrednošću koja je sačuvana kao `key_value`.

```
void loop(){
  if(primljen_podatak1=="prazan"||primljen_podatak2=="prazan"||...){
    if (irrecv.decode(&results)){
      if (results.value == 0xFFFFFFFF){
        results.value = key_value;
        ...
      }
    }
  }
}
```

Nakon toga se koristi struktura `switch case` za proveru primljenog koda u odnosu na različite kodove koji su poznati za svaki od tastera na daljinskom upravljaču. Odgovarajuće vrednosti ključa se zatim pamte u promenljivu `primljen_podatak`. Konačno, poziva se funkcija `resume()` koja resetuje prijemnik i priprema ga za prijem sledećeg unosa.

```
switch(results.value){
  ...
  case 0xFF02FD:
    primljen_podatak = "OK ";
    break ;
  ...
}
key_value = results.value;
irrecv.resume();
```

Sledeći korak je da se ispita da li je `fleg` jednak 0 ili je došlo do resetovanja pritiskom na odgovarajući taster na upravljaču za reset, ako nije, ponovo se unose karakteri, ako jesete, ispituje se nekoliko uslova koji nam služe da pritisnute karaktere na daljinskom upravljaču spakuju u odgovarajuće promenljive `primljen_podatak*` i ujedno se štampaju na LCD-u. Pored ova 4 podatka postoji i `primljen_podatak5` koji služi da signalizira da je njegovim

pritiskom došlo do kraja unosa karaktera i da dolazi do poređenja prethodno uneta 4 karaktera sa inicijalno unetom šifrom. Ispituje se poklapanje unete i definisane šifre, ako su iste zelena LED se uključuje i aktivira se zujalica preko funkcije `beep()`. Međutim, ako se unese pogrešna šifra zasvetleće crvena LED i zujalica će zazujati 2 puta i nakon toga treba krenuti sa unosom druge šifre, obaveštenje o tome pojavljuje se na LCD displeju. Ako je pritisnut pogrešan taster prilikom upoređivanja dve šifre, tada će takođe zasvetleti crvena LED i zujalica će zazujati 3 puta. Konačno, treba resetovati primljene podatke, tako da će svi biti "prazan", za naredni prolaz kroz glavnu programsku petlju.

```
if(fleg==0){
  lcd.print(primljen_podatak);
  if(primljen_podatak1 == "prazan")
  {
    primljen_podatak1 = primljen_podatak;
    lcd.setCursor(0,2);
    lcd.print("");
    delay(1000);
  }
  ...
  else if (primljen_podatak5 == "prazan")
  {
    primljen_podatak5 = primljen_podatak;
    if(primljen_podatak5 == "OK ")
    {
      lcd.setCursor(0,0);
      lcd.print(primljena_sifra);
      delay(1000);
      lcd.clear();
      if(primljena_sifra == sifra)
      {
        digitalWrite(greenLed,HIGH);
        lcd.setCursor(0,1);
        lcd.print("Dobra sifra");
        delay(2000);
        beep(1000);
        lcd.clear();
        primljena_sifra=" ";
        digitalWrite(greenLed,LOW);
      }
    }
    else
    {
      digitalWrite(redLed,HIGH);
      lcd.setCursor(0,0);
      lcd.print("Pogresna sifra");
      ...
      primljena_sifra=" ";
      digitalWrite(redLed,LOW);
    }
  }
  else
  {
    digitalWrite(redLed,HIGH);
    lcd.setCursor(0,0);
    lcd.print("Pogresan taster!");
    ...
    lcd.print("Unesite ponovo!");
    primljena_sifra=" ";
    digitalWrite(redLed,LOW);
  }
  primljen_podatak1 = "prazan";
  ...
}
```

```

delay(1000);
fleg = 0;
}
}

```

Funkcija za zujalicu data je u nastavku:

```

void beep(unsigned char delaysms)
{
    analogWrite(9, 20);
    delay(delaysms);
    analogWrite(9, 0);
    delay(delaysms);
}

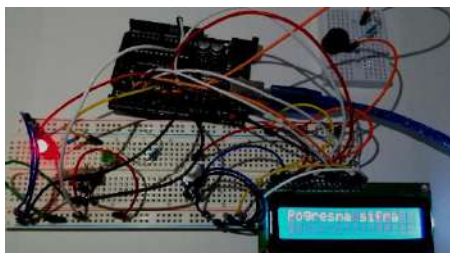
```

V. REZULTATI TESTIRANJA SISTEMA

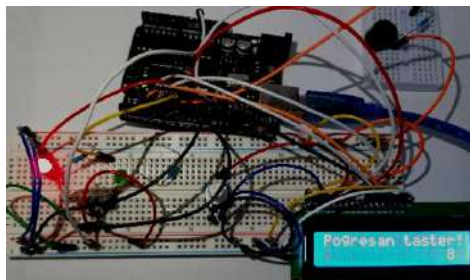
U zavisnosti od tačnosti unete šifre dobija se odgovarajuća poruka na LCD displeju, aktiviraju se LED diode i zujalica. Unosi se šifra od 4 karaktera preko daljinskog upravljača. Ako je korisnik uneo pogrešnu šifru, pritisnuo odgovarajuće dugme, na ekranu LCD displeja će se ispisati “Pogrešna šifra” i aktiviraće se crvena LED (slika 6), kao i zujalica, koja će dva puta uzastopno zazujati. Ako je korisnik uneo dobru šifru ili pogrešnu šifru, nije pritisnuo odgovarajuće dugme, na ekranu LCD displeja će se ispisati “Pogrešan taster” i aktiviraće se crvena LED (slika 7), kao i zujalica, koja će tri puta uzastopno zazujati. Ako je korisnik uneo dobru šifru, pritisnuo odgovarajuće dugme, na ekranu LCD displeja će se ispisati “Dobra šifra”, i aktiviraće se zelena LED (slika 8), kao i zujalica, koja će jednom zazujati. Korisnik može poništiti unete karaktere, pritiskom na odgovarajući taster i tako otpočeti unos šifre od početka.

VI. ZAKLJUČAK

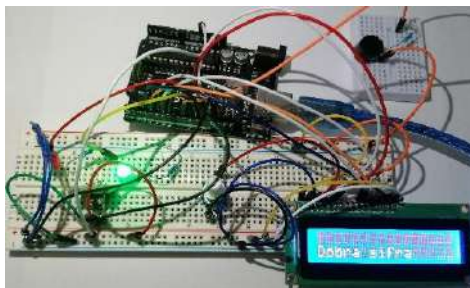
Iako su se ljudi od davnina koristili različitim metodama zaključavanja, razvoj savremene mikroelektronike omogućio je druge znatno naprednije načine za zaključavanje. U ovom radu predstavljena je praktična realizacija sistema za zaključavanje putem unosa šifre preko daljinskog upravljača upravljnog Arduino mikrokontrolerskom platformom. Za unos šifre umesto daljinskog upravljača može se koristiti tastatura ili sistem daljinskog slanja i prijema podataka pomoću RFID kartica/odašiljača. Ovo će biti predmet budućih istraživanja.



Slika 6. Prikaz prve situacije kada je uneta pogrešna šifra i pritisnut odgovarajući taster.



Slika 7. Prikaz druge situacije kada je uneta pogrešna šifra ili dobra šifra, ali je pritisnut pogrešan taster.



Slika 8. Prikaz treće situacije kada je uneta dobra šifra i kada je pritisnut odgovarajući taster.

ZAHVALNICA

Autor se zahvaljuje Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, a posebno asistentima Nikoli Mitroviću i Milošu Marjanoviću na pruženoj pomoći i saradnji, kao i usmeravanju tokom realizacije ovog projekta.

LITERATURA

- [1] Istorija ključeva i brava. Dostupno na: <http://www.historyofkeys.com/locks-history/history-of-locks/>
- [2] ATmega 328 datasheet. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [3] HX1838 tehnička dokumentacija. Dostupno na: <http://eeshop.unl.edu/pdf/VS1838-Infrared-Receiver-datasheet.pdf>
- [4] Arduino Projects Book, ed. S. Fitzgerald, M. Shiloh, T. Igoe, Arduino LLC, 2012.
- [5] Arduino funkcije. Dostupno na: <https://www.arduino.cc/reference/en/>

Softverska kontrola SMU Keithley serije 2400 za karakterizaciju PMOS tranzistora IRF9520

Strahinja Dimitrijević

Sadržaj – U ovom radu prezentovan je postupak merenja strujno-naponskih karakteristika PMOS tranzistora snage IRF9520 korišćenjem izvorno-merne jedinice (*Source – Measure Unit*-a) Keithley serije 2400 koji je softverski kontrolisan uz pomoć programa napisanog u programskom jeziku C# u okruženju *Visual Studio 2019*. U radu će biti navedena funkcija u programu koja poziva uređaj i izvršava merenje, kao i sama šema za povezivanje uređaja sa merenom komponentom. Na kraju će biti prikazani rezultati merenja.

I. UVOD

Instrumenti za merenje i izvori napajanja jesu nezaobilazni uređaji svake laboratorije i neophodni su za realizaciju bilo kog eksperimenta. Oni pružaju mogućnost da se njihovim korišćenjem izmere karakteristike bilo koje komponente sa dva ili više izvoda. Naročitu prednost pružaju uređaji koji u sebi imaju integrisani merni instrument i izvor napajanja – oni nose naziv izvorno-merne jedinice (eng. *Source Measure Unit*, skraćeno *SMU*). SMU uređaji predstavljaju elegantno rešenje u slučaju kada je potrebno minimizovati količinu i zaokupiranost potrebnom opremom kao i kabliranje koje može u određenim situacijama da bespotrebno opterećuje merenje [1]. Jedan primer takvog instrumenta o kome će kasnije biti više reči je SMU Keithley 2400.

Kao što je dobro poznato, uređaji za merenje moraju biti u neposrednom kontaktu sa merenom komponentom. Međutim, sam korisnik mernog uređaja ne mora nužno da bude u direktnom kontaktu sa instrumentom. Određeni instrumenti omogućavaju korisniku udaljeni (eng. *remote*) pristup, što znači da korisnik može softverski, preko određenog interfejsa, zadavati naredbe instrumentu i dobijati natrag izmerene podatke. Tu se već zalazi u domen virtuelne instrumentacije, koja je postala sve prisutnija sa razvojem i primenom računara u svim sferama inženjerstva i tehnologije. Virtuelna instrumentacija omogućava korisnicima da samostalno definišu funkcije mernog instrumenta u skladu sa svojim potrebama i zahtevima, odnosno u skladu sa specifičnom primenom [2].

Udaljena tj. softverska kontrola instrumenta ne rešava samo problem udaljenosti i problem preokupiranosti opremom. Korišćenjem okruženja za razvoj aplikacija u određenom programskom jeziku, korisnik ima

neograničene mogućnosti što se tiče dizajna aplikacije koja će pokrenuti merenje, preuzeti rezultate merenja, prikazati ih grafički, izvršiti određene analize i na kraju završiti sa merenjem gašenjem uređaja.

Razvojno okruženje u kome će biti realizovana aplikacija za kontrolu SMU-a je *Visual Studio 2019*. *Visual Studio* je IDE (*Integrated Development Environment*) koje podržava mnoge aspekte razvoja softvera i omogućava korisnicima da u njemu pišu programski kod koji se kasnije može realizovati u aplikaciju [3]. Korišćeni programski jezik je C#. Završni program sastoji se od veoma velikog broja linija koda, pa će u ovom radu biti navedena funkcija koja upravlja uređajem, dok će celokupni kod biti dostupan na linku koji vodi do sajta *Git Hub*.

II. KORIŠĆENA OPREMA I POVEZIVANJE KOMPONENTE SA INSTRUMENTOM

A. Keithley 2400

Uređaj koji će biti korišćen u cilju merenja karakteristika tranzistora jeste SMU Keithley 2400 [4]. Izgled njegovog prednjeg i zadnjeg dela nalazi se na Slikama 1a) i 1b) respektivno.



a)



b)

Slika 1. a) Prednji deo SMU-a Keithley 2400, b) Zadnji deo SMU-a Keithley 2400.

S. Dimitrijević je student četvrte godine osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: strax.dimitrijevic@elfak.rs

Po definiciji, SMU predstavlja uređaj koji u sebi sadrži integrisanih pet uređaja. On kombinuje korisne osobine digitalnog multimetra, izvora napajanja, strujnog izvora, elektronskog opterećenja i generatora impulsa [5]. Kao takav, predstavlja sofisticirani instrument koji se može koristiti za veliki broj primena. Osnovne karakteristike instrumenta Keithley 2400 što se tiče maksimalnih vrednosti nalaze se u izvodu iz tehničke dokumentacije samog instrumenta, u Tabeli 1.

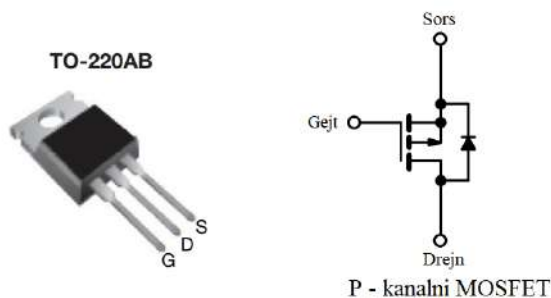
TABELA I
MAKSIMALNE VREDNOSTI SNAGE, NAPONA I STRUJE ZA KEITHLEY 2400

Model	2400
Maksimalna struja (izvorni/merni opseg)	1A
Maksimalni napon (izvorni/merni opseg)	200V
Rezolucija merenja (struja/napon)	1pA / 100nV
Maksimalna izlazna snaga	20W

SMU nudi neke prednosti u odnosu na klasične digitalne multimetre i izvore napajanja. Na primer, SMU ima mogućnost automatske linearne promene napona i struje do ili od negativnih i pozitivnih vrednosti. Velika prednost je da prilikom merenja nije potrebno ponovno povezivanje za različite režime merenja; konektori ostaju sve vreme povezani na merenu komponentu, tj DUT (eng. *Device Under Test*).

B. Tranzistor IRF9520

Tranzistor prikazan na Slici 2 je PMOS tranzistor IRF9520 [6].



Slika 2. Izvod iz tehničke dokumentacije tranzistora IRF9520.

Ovaj komercijalni tranzistor je često u upotrebi i predstavlja primer tipičnog PMOS tranzistora snage. Vrednost napona praga tranzistora je $V_{TH} = -3 \text{ V}$, sa maksimalnim naponom na drejnu $V_{DS} = -100 \text{ V}$ i

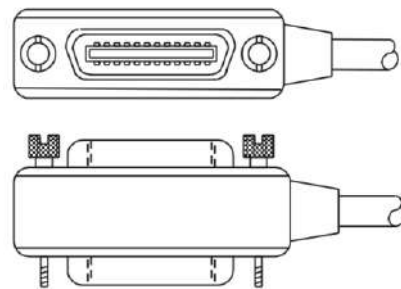
maksimalnom konstantnom strujom drejna $I_D = -6.8 \text{ A}$ na sobnoj temperaturi [6].

C. Povezivanje instrumenta sa računarom

Glavna osobina instrumenta Keithley 2400 koju eksploatišemo jeste njegova mogućnost povezivanja sa računarom. To činimo uz pomoć dva interfejsa koji su na raspolaganju: GPIB i RS-232 interfejs.

GPIB bus jeste IEEE-488 interfejs. Za povezivanje preko ovog interfejsa neophodno je uneti adresu uređaja, koja se automatski ispisuje na displeju za vreme uključivanja uređaja. RS-232 je serijski interfejs koga mi nećemo koristiti.

Za povezivanje uređaja na GPIB magistralu potreban je odgovarajući GPIB konektor, koji je prikazan na Slici 3.

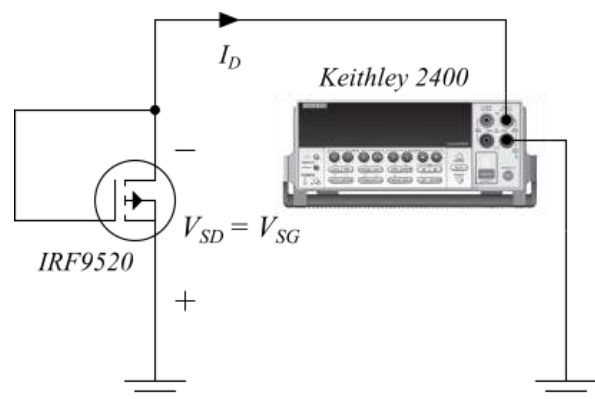


Slika 3. GPIB IEEE-488 konektor.

Međutim, za praktične primene, pošto noviji računari nemaju IEEE-488 konektor, koristi se GPIB na USB drajver, koji znatno olakšava povezivanje i kompatibilan je sa verzijama USB1.1/USB2.0 [7].

D. Šema merenja

Električna šema po kojoj će uređaj i komponenta biti povezani nalazi se na Slici 4.



Slika 4. Električna šema PMOS tranzistora i SMU-a.

Kao što se vidi sa slike, PMOS tranzistor je povezan u tzv. diodnoj konfiguraciji, u kome su gejt i drejn

kratkospojeni. To za posledicu ima da je tranzistor uvek u zasićenju, pa prilikom merenja nema potrebe paziti na to koliki napon dovodimo na drejn. Struja drejna u zasićenju data je izrazom (1).

$$I_D = \frac{\beta}{2} (V_{GS} - V_T)^2 \quad (1)$$

Postupak merenja strujno – naponskih karakteristika je sledeći: pošto se radi o PMOS tranzistoru, SMU na njegov drejn (gejt) dovodi negativan napon. U našem slučaju, polarizacija se ne obavlja čisto naponski, već strujno, što znači da SMU „forsira“ struju I_D u drejn tranzistora, i istovremeno meri napon na drejnu, koji je jednak naponu na gejtju. Time SMU dobija dve vrednosti: jednosmernu struju drejna i određeni napon na gejtju pri odgovarajućoj struji drejna, i te vrednosti je moguće predstaviti grafički, čime se na kraju i dobija karakteristika tranzistora.

III. METODI I REZULTATI MERENJA

A. Algoritam i programski kod

Algoritam za program koji pokreće merenje, prikuplja podatke i završava merenje prikazan je na Slici 5.



Slika 5. Algoritam osnovnog izvorno – mernog režima.

Ovaj algoritam se može realizovati unutar funkcije u programskom jeziku C#:

```

public void PMOS_snagas() //
{
    device_2400.Write("*RST");
    device_2400.Write(":SOUR:FUNC CURR");
    device_2400.Write(":SOUR:CURR:MODE FIX");
    device_2400.Write(":SOUR:CURR:RANG:AUTO ON");
    device_2400.Write(":SOUR:CURR:LEV 0");
    device_2400.Write(":SENS:VOLT:PROT 50");
    device_2400.Write(":SENS:FUNC 'VOLT'");
    device_2400.Write(":SENS:VOLT:RANG:AUTO ON");
    device_2400.Write(":FORM:ELEM VOLT");
    device_2400.Write(":OUTP ON");
}
  
```

Ovaj deo koda vrši inicijalnu konfiguraciju: govori uređaju da treba da se ponaša kao strujni izvor i zadaje početnu struju od 0 A, koja će se kasnije inkrementirati za određenu vrednost; zatim, govori uređaju da je merena veličina napon i definiše se maksimalna vrednost napona (eng. *compliance*). Na kraju nastupa uključivanje izlaza nakon čega se očekuje čitanje i obrada merenih podataka.

Unutar petlje koja inkrementira početnu vrednost struje i meri odgovarajući napon nalaze se sledeće naredbe:

```

device_2400.Write(":SOUR:CURR:LEV " + (pocetna_struja *
Math.Pow(konstanta, brojac)).ToString()); //zadaje struju
device_2400.Write(":SENS:FUNC 'VOLT'"); //meri napon
  
```

Nakon toga slede naredbe koje uzimaju izmerene vrednosti sa uređaja i iscrstavaju grafik.

Na kraju, potrebno je resetovati uređaj na standardna podešavanja i isključiti izlaz:

```

device_2400.Write("*RST");
device_2400.Write(":OUTP OFF");
  
```

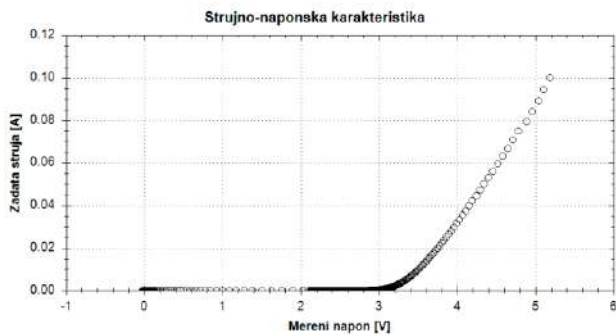
Treba napomenuti da su algoritam, kao i odgovarajući kod samo deo jedne mnogo veće aplikacije koju su projektovale kolege pri Laboratoriji za primenjenu fiziku na Elektronskom fakultetu u Nišu, te da je u ovom radu prezentovana samo konceptualna struktura softverske kontrole uređaja i osnove izvorno-mernog procesa. Aplikacija je realizovana tako da nudi i grafički korisnički interfejs koji omogućava korisniku da zada početnu i krajnju vrednost struje i napona, kao i sam grafički prikaz izmerene karakteristike. Ceo kod je dostupan na linku:

<https://github.com/Scar3CroWing/MGCPMS>

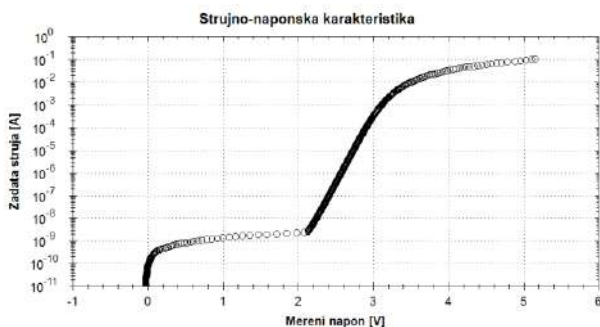
B. Rezultati merenja

Nakon povezivanja tranzistora sa instrumentom i pokretanja aplikacije, dobijaju se karakteristike tranzistora IRF9520 u linearnoj i logaritamskoj razmeri koje su prikazane na Slikama 6 i 7, respektivno. Na x-osi nalazi se napon na drejnu (gejtju) koji uređaj meri, dok je na y-osi struja drejna koju forsira uređaj. Pri pokretanju aplikacije moguće je unutar grafičkog korisničkog interfejsa podesiti broj merenih tačaka. U ovom primeru, broj tačaka iznosi 300. Manji broj tačaka imaće za posledicu brže, ali znatno nepreciznije merenje, dok će merenje sa većim brojem

tačka trajati proporcionalno duže, ali će kvalitet karakteristike biti приметно bolji.



Slika 6. Prenosna strujno-naponska karakteristika tranzistora IRF9520 u linearnoj razmeri.



Slika 7. Prenosna strujno-naponska karakteristika tranzistora IRF9520 u logaritamskoj razmeri.

Kao što se vidi sa grafika, izmerene strujno – naponske karakteristike tranzistora IRF9520 odgovaraju idealnim karakteristikama PMOS tranzistora. Takođe, jasno se vidi da je napon praga tranzistora oko – 3 V, što se poklapa sa vrednošću iz tehničke dokumentacije.

Za povezivanje komponente sa instrumentom iskorišćen je testni adapter (eng. *Test-Fixture*) koji pruža mogućnost vrlo lakog i brzog povezivanja komponenta, pri čemu je u proces merenja moguće uključiti i integrisana kola sa većim brojem izvoda.

Bitna napomena jeste da je aplikacija projektovana tako da se sve strujno – naponske karakteristike prikazuju u prvom kvadrantu bez obzira da li se radi o NMOS ili PMOS tranzistoru.

IV. ZAKLJUČAK

Softverska kontrola izvorno-mernog uređaja pruža jako velike mogućnosti što se tiče merenja i obrade merenih rezultata. Sa samo nekoliko linija koda moguće je

konfigurisati osnovni izvorno-merni režim rada, a na korisniku je da izmerene vrednosti prikaže grafički ili na bilo koji drugi način, što zavisi od konkretne primene. Takođe, jednom projektovanu aplikaciju nije potrebno menjati prilikom karakterizacije bilo kog drugog tranzistora. Naravno, ukoliko konkretni zadatak zahteva komplikovanija merenja sa više tranzistora u različitim konfiguracijama merenja, aplikacija se dodatno usložnjava, a broj potrebnih uređaja postaje veći. Ovo se delimično može rešiti korišćenjem višekanalnih SMU – a.

Najveća mana ovakvog načina kontrole SMU – a jeste to što različite serije Keithley uređaja imaju različitu sintaksu za osnovne naredbe. Da je umesto serije 2400 bio korišćen, na primer, Keithley serije 2600, bilo bi neophodno rekonstruisati celokupni programski kod i prilagoditi ga datom uređaju iz serije. Ovo može biti otežavajuća okolnost ukoliko se korisnik prvi put susreće sa novijom serijom Keithley uređaja, gde je neophodno voditi računa i o drajverima za sam uređaj, kao i o sintaksi. Međutim, kada se jednom konfiguriše uređaj i za njega napiše odgovarajući kod, to će važiti i za sve ostale uređaje u toj seriji.

ZAHVALNICA

Veliku zahvalnost dugujem profesoru Goranu Ristiću i asistentu Nikoli Mitroviću na jako korisnim sugestijama oko ideje i pisanja rada. Praktični deo ovog rada je realizovan u Laboratoriji za primenjenu fiziku Elektronskog fakulteta u Nišu.

Takođe, veliku zahvalnost dugujem kolegama Stefanu Iliću i Aleksandru Jevtiću koji su mi pružili nesebičnu pomoć prilikom sticanja znanja koja su bila ključna za realizaciju ovog rada.

LITERATURA

- [1] Vojkan Davidović, *Predavanja iz predmeta Karakterizacija Komponenta*
- [2] Metrologija električnih veličina – Laboratorijski praktikum, Vežba br. 4, *Virtuelni Instrumenti*, Elektronski Fakultet, Univerzitet u Nišu.
- [3] Microsoft, *Welcome to the Visual Studio IDE*, dostupno na: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- [4] Keithley Instruments Inc, *2400 Series SourceMeter User's Manual*, Cleveland, Ohio, USA, May 2002.
- [5] Tektronix, *Source Measure Units – Selector Guide*
- [6] Vishay Electronics, *IRF9520 Datasheet*
- [7] Contec, *Low Price High – Speed GPIB Communication Micro Converter for USB2.0; datasheet.*

Mobilna android aplikacija za uspostavljanje kontakta između majstora i poslodavca i pronalazak prakse za učenike srednjih stručnih škola

Aleksandar Cvetković, Pavle Janevski

Sadržaj - U ovom radu je predstavljena društveno korisna mobilna android aplikacija za uspostavljanje kontakta između majstora i poslodavca - "Majstor". Aplikacija takođe omogućava pronalazak adekvatne prakse učenicima koji pohađaju srednje stručne škole. U prvom delu rada opisana je praktična primena i arhitektura aplikacije. Drugi deo rada objašnjava korisnički interfejs i funkcionalnosti aplikacije.

I. UVOD

Ideja za aplikaciju nastala je iz dva razloga koja smo videli kao probleme u našoj sredini:

- Pronalaženje odgovarajućeg majstora oduzima previše vremena. Vrlo često, kada želimo da nađemo majstora za neki kvar u našem domu, potrošimo mnogo vremena i energije. Neretko se na kraju ispostavi da majstor zapravo nije bio dovoljno kompetentan.
- Nedostatak prakse učenicima srednjih stručnih škola. Motivacija su bili učenici srednjih stručnih škola koji kažu da je veoma teško pronaći praksu koja će im pomoći u učenju i razvoju dalje karijere.

Iz želje da pomognemo mnogobrojnim ljudima koji imaju ove probleme, razvijena je aplikacija „Majstor“, koja daje rešenje za ova dva problema.

II. TEHNIČKI OPIS I STRUKTURA APLIKACIJE

Aplikacija je razvijena u okviru integrisanog razvojnog okruženja za android aplikacije "Android Studio" [1]. Objektno orijentisano programiranje jedan je od osnovnih koncepata programiranja koji je korišćen [2]. Aplikacija se sastoji iz više od 25 xml layout-a i java klasa koje će upravo pokazati zašto je objektno orijentisano programiranje bitno za strukturu aplikacije. Korišćena je baza podataka Google servisa Firebase [3], koja omogućuje čuvanje svih potrebnih podataka kao i interakciju dve vrste korisnika. U aplikaciji je korišćeno nekoliko komponenata iz biblioteke „Android Arsenal“ [4], kao i slike koje su obrađivane u programu „Adobe xd“ [5].

A. Cvetković je student na smeru Računarstvo i informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: alek.cvetkovic01@gmail.com

P. Janevski je student na smeru Softversko inženjerstvo, Elektrotehnički fakultet, Univerzitet u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija, E-mail: pavle.jane@gmail.com

A. Struktura i celine aplikacije

Aplikaciji pristupaju tri vrste korisnika:

- Korisnik koji ima neki kvar kod kuće i želi da unajmi majstora (u daljem tekstu „poslodavac“).
- Korisnik koji je majstor po profesiji i želi da pronađe posao (u daljem tekstu „majstor“).
- Korisnik koji se prijavljuje u ulozi srednje stručne škole u cilju pronalaženja prakse svojim učenicima (u daljem tekstu „škola“).

Sistem se sastoji iz sledećih celina i ekrana aplikacije:

- Deo za autentifikaciju (Registracioni deo, Deo za prijavljivanje)
- Glavni ekrani aplikacije (dostupni iz menija):
 - Deo za majstore i škole: Svi poslovi, Pregovori sa poslodavcima, Ugovori sa poslodavcima
 - Deo za poslodavce: Moji poslovi, Ponude majstora, Pregovori sa majstorima, Ugovori sa majstorima
 - Ekran za informacije i podešavanje naloga
- Sporedni ekrani aplikacije:
 - Deo za majstore i škole: Prijavljivanje za posao, Razgovor (poruke) sa poslodavcem, Prihvatanje ugovora
 - Deo za poslodavce: Prihvatanje ili odbijanje ponuda majstora, Razgovor (poruke) sa majstorom, Kreiranje ugovora

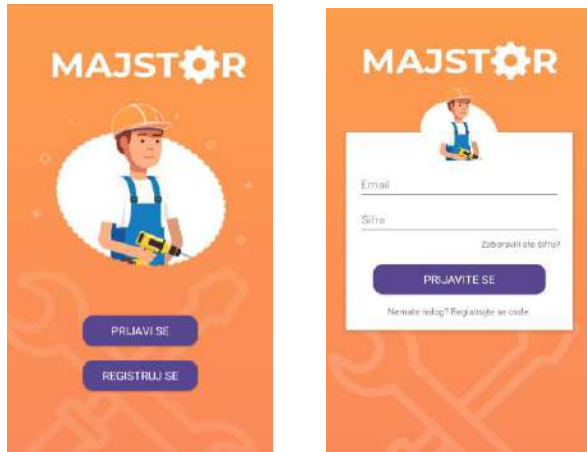
Aplikacija podržava dve vrste naloga kojima pristupaju odgovarajući korisnici – „standardni nalog“ (namenjen poslodavcima i majstorima) i „školski nalog“ (namenjen isključivo školama). Uz slike koje prate opis, u narednim poglavljima, prvo će biti opisane funkcionalnosti standardnog naloga, a zatim školskog naloga.

III. KORISNIČKI INTERFEJS I FUNKCIONALNOSTI

Kada korisnik prvi put uđe u aplikaciju (slika 1. a)) prvo mora da prođe kroz deo za autentifikaciju. U ovoj celini razlikujemo deo za prijavljivanje (slika 1. b)) i deo za registrovanje (slika 2. a)). Prilikom registrovanja bira se vrsta naloga (standardni ili školski). Vrsta naloga je jedini parametar koji se kasnije ne može promeniti i od koga zavise ostali parametri za registraciju.

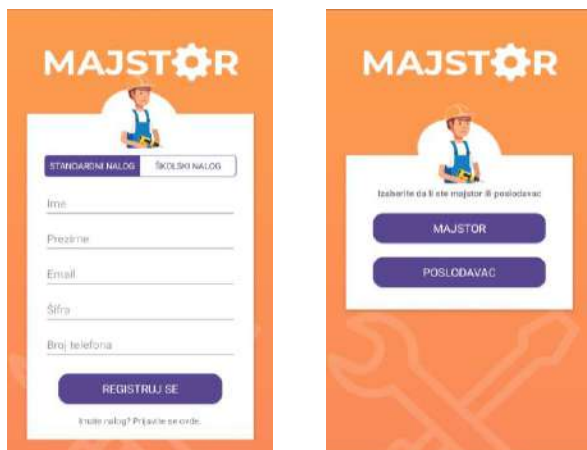
A. Funkcionalnosti standardnog naloga

Nakon kreiranja standardnog naloga (slika 2. a)), koji je opisan u ovom poglavlju, korisnik bira da li želi da koristi aplikaciju u svojstvu majstora ili poslodavca (slika 2. b)). Ovaj izbor kasnije se lako može promeniti u podešavanju naloga, što je korisno kada su majstoru potrebne popravke za koje nije specijalizovan.



Slika 1. a) Početna strana.

b) Prijavljivanje.



Slika 2. a) Registrovanje.

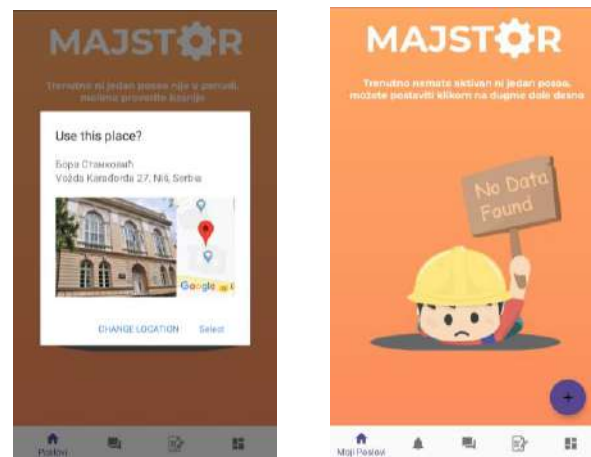
b) Izbor korisnika.

Nakon kreiranja naloga, od korisnika se zahteva da unese svoju adresu stanovanja (slika 3. a)). Lokacija korisnika je jako bitna majstoru da bi znao koliko je daleko posao koji je u ponudi. Vođeno je računa o bezbednosti privatnih informacija tako da majstor može da vidi samo približnu udaljenost posla sve dok ne bude sklopljen ugovor o poslu, koji poslodavac može sklopiti samo sa jednim majstorom.

Kroz naredni deo teksta naizmenično će biti opisane funkcionalnosti kojima pristupaju majstor i poslodavac da bi se stekla jasna slika o tome kako interaguju dve vrste korisnika.

Sličnost koja se može primetiti kod poslodavca (slika 3. b)) i majstora (slika 7) jeste meni za navigaciju u dnu

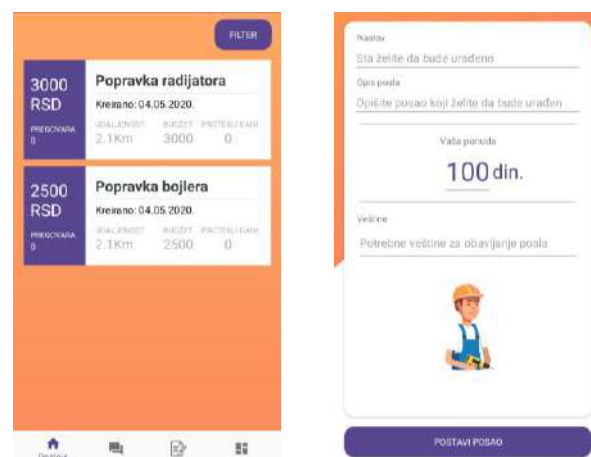
ekrana pomoću koga se korisnik kreće između ekrana dostupnih u aplikaciji.



Slika 3 a). Izbor lokacije.

b) Ekran Moji poslovi.

Prvi ekran kome pristupaju i majstor i poslodavac je ekran o poslovima. Kod poslodavca (ekran Moji Poslovi slika 3. b)), nalaze se kartice sa informacijama o poslovima koje je on do tada postavio. U donjem desnom uglu postoji dugme koje omogućava otvaranje ekrana za postavljanje posla (slika 4. b)). Kod majstora (ekran Poslovi slika 4. a)), moguće je videti kartice sa svim poslovima koji su u ponudi. Vrlo korisnu funkcionalnost za majstore na tom ekranu predstavlja opcija filtriranja (slika 5. a)) gde na osnovu kategorija za koje je specijalizovan, brzo i lako može izdvojiti poslove za koje je zainteresovan. Da bi majstor video detalje o poslu koji je u ponudi, potrebno je da klikne na karticu i ona će se rasklopiti (slika 5 b) – rasklopljena, slika 4. a)) – pre rasklapanja). Klikom na dugme „prijava se“ otvara se ekran za prijavljivanje (slika 6. a)), gde majstor može da pošalje prijavu za izabrani posao.

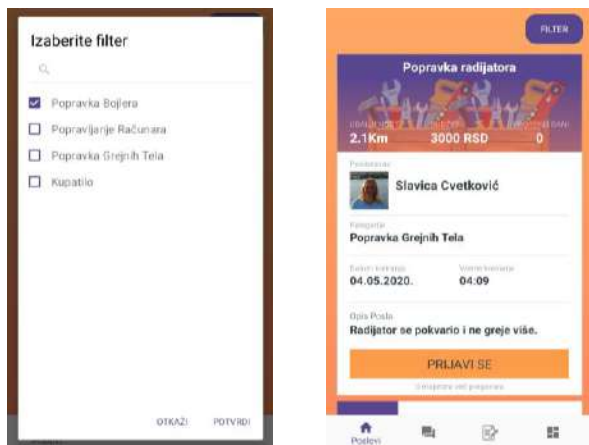


Slika 4. a) Ekran Poslovi.

b) Postavljanje posla.

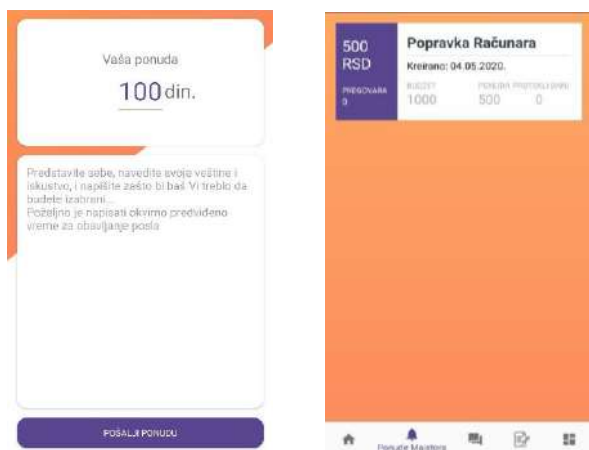
Drugi ekran kod poslodavca (ekran Ponude Majstora slika 6. b)), sadrži kartice sa ponudama koje su majstori

poslali za postavljene poslove. Klikom na neku od kartica, kartica se rasklapa i poslodavac može da vidi sve detalje o toj ponudi (slika 7. a) – rasklopljena, slika 6. b) – pre rasklapanja). Sada poslodavac klikom na strelicu pored majstorove slike može da otvori majstorov profil gde može da vidi ocene i komentare drugih poslodavaca koji će mu pomoći da odluči da li želi da započne pregovore sa tim majstorom (slika 7. b)). Klikom na dugme „prihvati“ (slika 7. a)), poslodavac prihvata da započne pregovore sa izabranim majstorom. Moguće je istovremeno pregovarati sa više majstora o jednom poslu.



Slika 5. a) Filtriranje poslova.

b). Kartica posao.

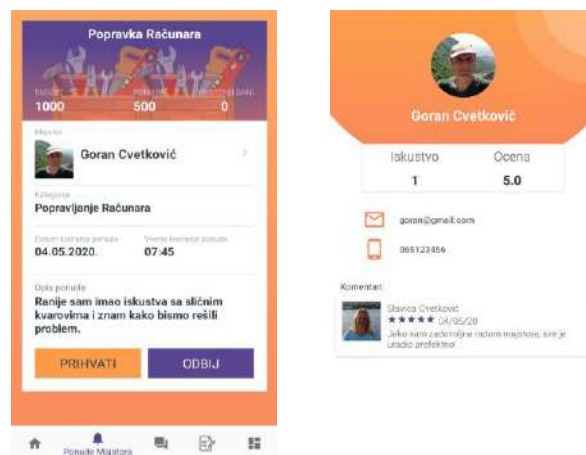


Slika 6. a) Prijavljivanje.

b) Ponude majstora.

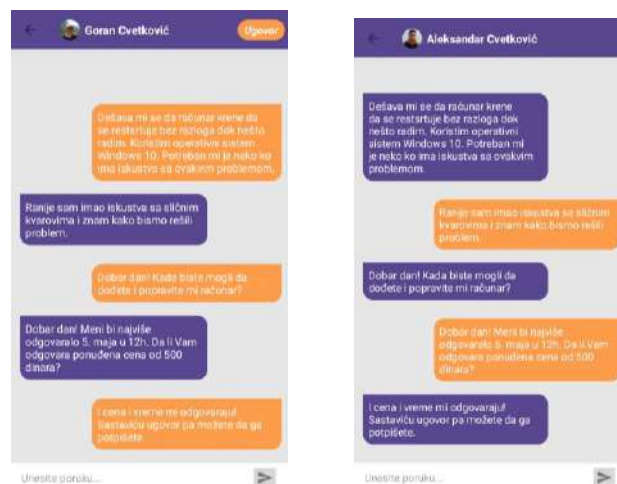
Naredni ekran (ekran Pregovori), koji postoji i kod majstora i kod poslodavca, sadrži kartice sa podacima o pregovorima koji su aktivni. Klikom na karticu, otvara se ekran za Razgovor (međusobno dopisivanje i dogovaranje, slika 8. a) – poslodavac, slika 8. b) – majstor). Kod poslodavca u gornjem desnom uglu postoji dugme pomoću kojeg može da otvori ekran za kreiranje ugovora (slika 9. a)). Ugovor nije pravni dokument, već predstavlja postignut dogovor, koji se za jedan posao može sklopiti samo sa jednim majstorom i daje mogućnost poslodavcu da kasnije oceni majstora. Nakon što poslodavac kreira

ugovor, kod majstora se, na ekranu za dopisivanje, pojavljuje dugme u gornjem desnom uglu pomoću kojeg može da vidi i prihvati ugovor. Sa ekrana za dopisivanje (slika 8. a)) poslodavac može, klikom na majstorovu sliku u gornjem delu ekrana, da otvori majstorov profil (slika 7. b)).



Slika 7. a). Kartica ponuda.

b). Majstorov profil.



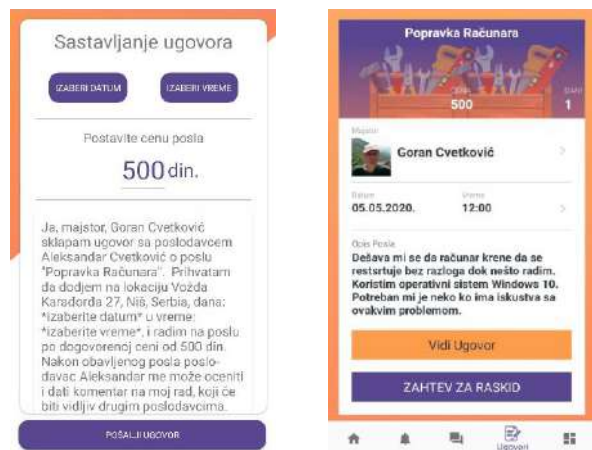
Slika 8. a). Razgovor poslodavca.

b) Razgovor majstora.

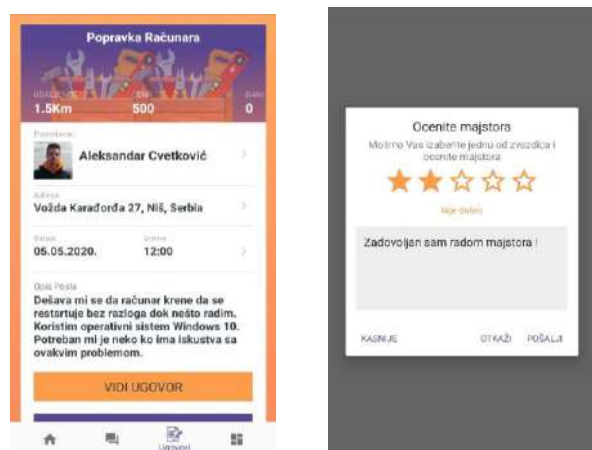
Ekran Ugovori (slika 9. b) – poslodavac, slika 10. a) – majstor). Sadrži kartice sa podacima o ugovorima koji su do sada sklopljeni. Pored mogućnosti da vide detalje o ugovoru, majstor i poslodavac, mogu da raskinu ugovore tako što šalju zahtev za raskid. U slučaju da obe strane pošalju zahtev, ugovor se raskida (majstor ne može da dobije lošu ocenu). Majstor (slika 10. a)), može po prvi put da vidi adresu poslodavca kao i da klikom na strelicu pored adrese otvori mapu sa optimalnom rutom do njega. Majstor i poslodavac mogu da vide datum za koji je zakazan posao i da klikom na strelicu pored datuma dodaju podsetnik u kalendar.

Poslodavac ima mogućnost da nakon što je prošao termin određen ugovorom, oceni majstora ocenom od 1 do 5 i napiše komentar koji će kasnije služiti drugim

poslodavcima kao uverenje o kvalitetu majstora (slika 10. b)).



Slika 9. a) Kreiranje ugovora. b) Ugovor poslodavac.



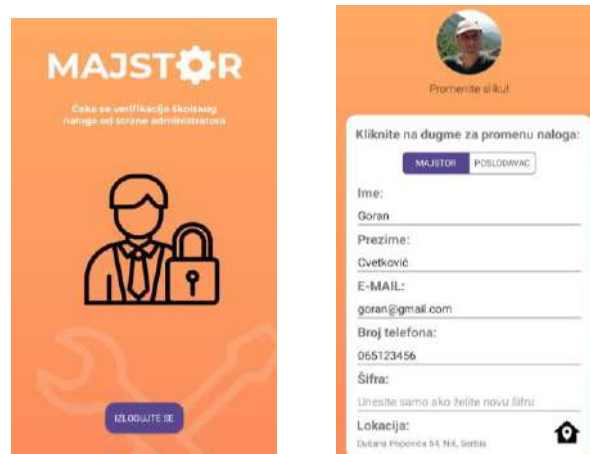
Slika 10. a). Ugovor – majstor. b) Ocenjivanje.

B. Funkcionalnosti školskog naloga

Prilikom registracije (slika 2. a)), moguće je odabrati kreiranje školskog naloga. Školski nalog ima većinu potpuno istih funkcionalnosti kao i majstor kod standardnog naloga, osim par privilegija zbog čega je školski nalog dostupan za korišćenje tek kada administratori potvrde da nalog stvarno pripada određenoj srednjoj školi (slika 11. a)). Ponude za posao sa školskog naloga su besplatne. Kartice sa ponudama pojavljuju se kod poslodavca ispred ponuda kreiranih standardnim nalogom (imaju prioritet). Kartice se izgledom razlikuju od ostalih, da bi privukle pažnju i pružile šansu učenicima da dobiju praksu.

Podešavanje naloga je ekran kome pristupa svaki korisnik, gde ima opciju da promeni parametre svog

naloga. U slučaju standardnog naloga korisnik može da izabere da li će pristupiti funkcionalnostima majstora ili poslodavca (slika 11. b)).



Slika 11. a) Škola – verifikacija. b) Podešavanje naloga.

IV. ZAKLJUČAK

Aplikacija pomaže poslodavcu da brzo i lako pronađe odgovarajućeg majstora (koji može biti i učenik srednje stručne škole) za rešavanje svog problema po ceni koja njemu odgovara, a sa druge strane pomaže majstoru ili školi da pronađu poslove za koje su kvalifikovani. Ako se više majstora prijavi za jedan posao, poslodavac može da izabere onog koji njemu najviše odgovara, po ceni i po opisu profila. Majstori mogu biti prepoznati ako dobro rade svoj posao i poslodavac ih dobro oceni. Njihova komunikacija je optimizovana delom za pregovore gde mogu brzo i lako da uspostave dogovor. Korišćenjem školskog naloga se stvara odnos u kome je zadovoljan poslodavac, jer je rešio svoj problem bez plaćanja i bez mnogo utrošenog vremena i u kome je zadovoljna škola, jer je dobila realnu i primenljivu praksu za svoje učenike.

LITERATURA

- [1] Razvojno okruženje Android Studio. Dostupno na: <https://developer.android.com/studio>
- [2] Koncept objektno orijentisanog programiranja. Dostupno na: https://sr.wikipedia.org/sr-el/Objektno-orijentisano_programiranje
- [3] Baza podataka Google servisa Firebase. Dostupno na: <https://firebase.google.com/>
- [4] Android Arsenal. Dostupno na: <https://android-arsenal.com/>
- [5] Softver za obradu slika Adobe XD. Dostupno na: <https://www.adobe.com/products/xd.html>

Primena Markovljevih lanaca na prikaz podataka sa meteoroloških stanica o vremenu korišćenjem R jezika

Miljan Jeremić

Sadržaj - Ovaj rad predstavlja primenu tzv. Markovljevih lanaca (*Hidden Markov Model*), kao i upotrebu paketa za primenu ovog modela i predstavljanje numeričkih podataka u R programskom jeziku. Koristeći R jezik sa primenom u oblasti nauke o podacima (*Data Science*), *markovchain* i *HMM* paketi su adekvatni za numeričko predstavljanje rezultata korišćenjem podataka o padavinama i temperaturama po mesecima u toku godine. Rezultati pokazuju važan doprinos i značajna poboljšanja u odnosu na prethodne napore drugih programa i programskih jezika koji mogu statistički obrađivati podatke. Rad predstavljen ovde može pomoći u rešavanju problema obrade statističkih podataka za proveru parametara koji karakterišu slivove reka, nivoa padavina i promene temperatura u Srbiji.

I. UVOD

Analiza istraživačkih podataka se bavi procedurama i tehnikama za analizu podataka kao i interpretacijom dobijenih rezultata, prikazujući ih najčešće numerički i vizuelno.

Obrada i analiza podataka se poboljšavaju vizuelno kodiranjem informacija kroz aktivnost vizualizacije [1].

R sa svojim mnoštvom paketa može se koristiti ne samo za statističku analizu, već i za GIS analizu i rad sa prostornim podacima.

Ovaj rad predstavlja prikaz primene Markovljevih lanaca kako bi se prikazala projekcija vremenskih prilika na osnovu podataka i njihovih vrednosti o padavinama i temperaturama (minimalne, maksimalne i srednje vrednosti) prikupljenih sa meteoroloških stanica u Srbiji.

II. MARKOVLJEVI LANCI

U relativno jednostavnu, ali i veoma interesantnu klasu slučajnih procesa pripadaju i Markovljevi lanci. Mogu da opisuju slučajne pojave ili sisteme koji se menjaju tokom vremena, a njihova jednostavna struktura omogućava nam da saznamo mnogo toga o njihovom ponašanju u nekoj bliskoj budućnosti. Ako uzmemo u obzir poznavanje nekih informacija o prošlim stanjima, uslovna raspodela budućih stanja sistema zavisi samo od onih najskorijih. Dakle možemo da kažemo da predviđanje budućnosti sistema zavisi samo od sadašnjeg stanja, a ne i od puta kojim je sistem došao do tog stanja.

Lanci Markova predstavljaju jedan od najprostijih matematičkih modela za opisivanje pojava u realnom životu.

M. G. Jeremić, Knjaževačka gimnazija, Karađorđeva 16, 19350 Knjaževac, Srbija, E-mail: miljan.jeremic@gmail.com

Pravilo da su najjednostavniji modeli često i najkorisniji za analizu praktičnih problema ne zaobilazi ni Markovljeve lance. Njima se mogu modelirati mnoge pojave, kao što su pojave u biologiji, psihologiji, sportu i dr [2].

Prilikom posmatranja određenih sistema, ponekad nam je važno da odredimo kako se stanje nekog sistema menja s vremenom. Ako stanje sistema predstavlja vrednost slučajne promenljive, sledi da nas zanima promena slučajne promenljive u zavisnosti od vremena. Ponašanje posmatranog sistema opisuje se stohastičkim procesom. Poseban oblik stohastičkog procesa je Markovljev proces odnosno Markovljev lanac.

Markovljevo svojstvo koje opisuje činjenicu da raspodela verovatnoće slučajne promenljive X_t u trenutku $t = t_n$ zavisi samo od vrednosti x_{n-1} sistema u trenutku t_{n-1} , a ne zavisi od vrednosti sistema u ranijim trenucima. Radi jednostavnijeg zapisa, stanje sistema u trenutku t_n ćemo označavati sa X_n i govoriti o njemu kao o n -tom koraku sistema. U zavisnosti od toga da li je T diskretan ili neprekidan skup, razlikujemo procese Markova sa diskretnim i neprekidnim vremenom. Sve vrednosti koje može da uzme slučajna veličina X_t čine skup stanja ili prostor stanja koji označavamo sa S . Ako je skup S diskretan tj. konačan ili prebrojiv tada se proces Markova zove lanac Markova.

Da bismo Markovljev lanac u potpunosti opisali moramo, osim matrice verovatnoće prelaska, poznavati i vektor početnih vrednosti, koji predstavlja stanje Markovljevog lanca u početnom trenutku posmatranja, tj. vrednost slučajne promenljive X_0 .

Objasnilo bismo kratko klasifikaciju stanja Markovljevog lanca kako bismo lakše formulisali naredne definicije [3]:

- Za dva stanja i i j , putanja od i do j je niz prelaza koji počinju u i i završavaju u j , tako da je pri svakom prelazu u nizu verovatnoća pozitivna.
- Stanje j je dostupno iz stanja i ako postoji putanja koja vodi od i do j .
- Dva stanja su povezana ako je stanje j dostupno iz i , i ako je stanje i dostupno iz j .
- Skup stanja S u Markovljevom lancu je zatvoren skup ako ne postoji stanje izvan skupa S koje je dostupno iz nekog stanja iz S .
- Stanje i je apsorbujuće ako je $p_{ii} = 1$.
- Stanje i je prelazno stanje ako postoji stanje j koje je dostupno iz i , ali stanje i nije dostupno iz stanja j . Prelazno stanje omogućuje da

napustimo stanje i ali se više nikada u njega ne vraćamo.

- Stanje koje nije prelazno, naziva se rekurentno stanje (povratno stanje).

Stanje i je periodično sa periodom $k > 1$ ako je k najmanji broj takav da sve putanje vode iz stanja i ponovo u stanje i , i koje imaju dužinu jednaku deliocu broja k . Ako rekurentno stanje nije periodično, ono je aperiodično.

III. R JEZIK I NJEGOVI PAKETI

R jezik uglavnom koristi podatke iz *Excel* dokumenata u CSV ili XLSX formatima, kao i sa web stranica koji se uvoze.

Primenom Markovljevih lanaca, na samom početku, podaci o padavinama se postavljaju u stanje početnih vrednosti, označenih sa 0 ili 1. Na osnovu tog početnog vektora stanja o padavinama, kao i podacima o srednjoj maksimalnoj temperature, podaci se procesiraju kroz process Markovljevih lanaca gde dobijamo vektor koji prikazuje da li je dan kišovit, vedar, ili topao (*rain*, *clear*, *warm*), i na kraju se vrši analiza obrađenih podataka.

Podaci obrađeni korišćenjem R jezika najčešće se izvoze u *Excel* datoteke ili tekstualne tabele. Takvi podaci se kasnije lako mogu preneti na web aplikaciju koju korisnici mogu videti kroz pregledače nakon obrade.

Da bi prikazali podatke obrađene u jeziku R i direktno poslali web aplikaciji, neophodno je ponekad da se obezbedi web servis za ubrzani prikaz obrade podataka na jeziku R.

IV. NUMERIČKI PRIKAZ PODATAKA UZ PRIMENU MARKOVLJEVIH LANACA KORIŠĆENJEM R JEZIKA

R programski jezik je dizajniran za obavljanje statističkih analiza. Istraživanja programskih jezika pokazuju da se upotreba R brzo povećava, očigledno u vezi sa sve većim prikupljanjem podataka. R se može koristiti na dva različita načina. Najčešće, R se koristi kao interaktivni alat. Na primer, u ljsuci (*shell*) R konzole, korisnik bi mogao upisati komande. R ima na stotine ugrađenih funkcija koje mogu obavljati hiljade statističkih zadataka.

Paket *markovchain* je novi alat u jeziku R koji omogućava i sadrži klase i metode za kreiranje i upravljanje (na primer, iscrtavanje, štampanje, izvoz) Markovljevih lanaca sa diskretnim vremenom (*DTMC*). Osim toga, pruža funkcije za obavljanje statističkih podataka (prilagođavanje) i crtanje slučajnih varijanti) i analizu verovatnoća (analiza svojstava *DTMC*).

Paket *HMM* je jednostavan za postavljanje, primenu i zaključivanje sa diskretnim vremenom i diskretnim prostorom skrivenih Markovljevih modela.

Sledeći deo programskog koda pokazuje kako započeti korišćenjem paketa *markovchain* i *HMM*.

```
library("readxl")
library(markovchain)
library(HMM)
```

Skup poodataka učitavamo iz Excel-ovog fajla (*HMM1.xlsx*) gde se nalaze vrednosti padavina i temeperatura za period od 36 meseci. Ovaj skup podataka može se iskoristiti i za generisanje klima dijagrama, samo što su nazivi kolana sada vertikalno postavljeni u prvoj koloni (okrenuto ulevo), što je prikazano na slici 1.

MESEC	Tsrmx	Tsrmn	Tmin	Precipitation
1	12.6	2.0	-2.5	33
2	11.5	1.2	-5.5	33.7
3	15.6	3.3	-2	19.2
4	22.1	4.4	1	5.4
5	25.3	12.3	3.5	104.8
6	31.0	15.2	10.5	10
7	34.9	15.9	10.5	11.4
8	31.7	16.5	12	73.5
9	23.2	9.9	3	64.9
10	16.2	6.7	0.5	115.5
11	8.7	0.8	-7.5	115.6
12	4.9	-2.2	-6.5	34.8

Slika 1. Prikaz dela skupa podataka za obradu.

Na osnovu učitanih podataka iz Excel dokumenta u R jezik za obradu, počinje proces obrade učitanih podataka kroz memorisanje u jenu promenljivu tipa vektor, odnosno matrica [4].

Na osnovu učitanih podataka, u prvom delu obrade traže se kišni dani uz uslove da je količina padavina veća od 30 mm.

```
> spring$Precipitation[1:36]
[1] 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1
1 1 0 0 0 1
```

Na osnovu toga, kad za takav uslov koji je ispunjen imamo vrednost 1, a u ostalim slučajevima to se predstavlja vrednošću 0, vrši se sada klasifikacija svakog posmatranja kao specifičnog stanja koje se predstavlja kao: kiša, vedro ili toplo. Sve se to zapisuje u jedan vektor, u našem slučaju od 36 elemenata opservacije.

```
> spring$markov_states[1:36]
[1] "rain" "rain" "clear" "warm" "warm" "warm" "warm"
"warm" "warm"
[10] "clear" "clear" "rain" "rain" "clear" "rain" "warm"
"warm" "warm"
[19] "warm" "warm" "warm" "clear" "rain" "clear" "rain"
"rain" "rain"
[28] "clear" "warm" "warm" "warm" "warm" "warm" "warm"
"clear" "clear" "rain"
```

Na osnovu podataka o padavinama, kao i podacima o prikazu primene metode Fit u vektoru *markov_states* dobijam jednu tabelu sa podacima koliko ima kišnih, sušnih ili toplih dana sa manjim ili većim padavinama.

```
> e_table
```

```
clear 9 0
rain 0 10
warm 6 11
```

>

Na osnovu ove tabele računaju se verovatnoće ovih događaja i to se zapisuje u matricu B za dalju obradu.

> B

```
      0      1
clear 1.0000000 0.0000000
rain  0.0000000 1.0000000
warm  0.3529412 0.6470588
```

>

Nakon ovih dosadašnjih obrada, dobija se jedna tabela sa verovatnoćama koja prikazuje vrednosti za 3 parametra prikaza stanja vremena.

```
>table(spring$markov_states)/length(spring$markov_states)
)
```

```
clear rain warm
0.2500000 0.2777778 0.4722222
```

>

Na osnovu ovih rezultata verovatnoća stanja vremena, dolazi se do inicijalizacije modela Markovljevih lanaca koji je ovde i prikazan [5].

> # Initialise HMM

```
> hmm = initHMM(S, V_labels, startProbs = pi, transProbs = A, emissionProbs = cbind(B[,0:1]))
```

>

> print(hmm)

```
$States
[1] "clear" "rain" "warm"
```

```
$Symbols
[1] "H" "L" "M"
```

```
$startProbs
clear rain warm
0.25 0.28 0.47
```

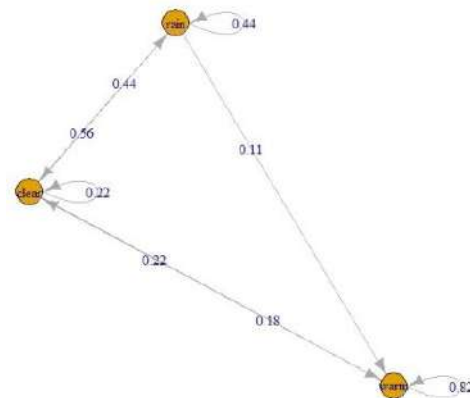
```
$transProbs
to
from clear rain warm
clear 0.2222222 0.5555556 0.2222222
rain 0.4444444 0.4444444 0.1111111
warm 0.1764706 0.0000000 0.8235294
```

```
$emissionProbs
symbols
states H L M
clear 1.0000000 1.0000000 1.0000000
rain 0.0000000 0.0000000 0.0000000
warm 0.3529412 0.3529412 0.3529412
```

V. GRAFIČKI PRIKAZ PODATAKA UZ PRIMENU MARKOVLJEVIH LANACA KORIŠĆENJEM R JEZIKA

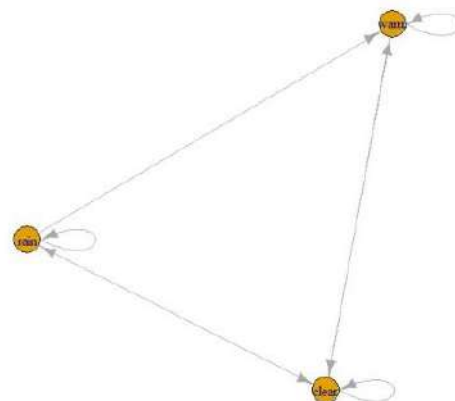
Za grafički prikaz Markovljevih lanaca koristi se biblioteka u R programskom jeziku pod imenom *igraph*.

Na osnovu prikaza vrednosti verovatnoća u procesu kreiranja modela Markovljevog lanca, *\$transProbs*, prikazan je usmeren graf sa vrednostima potega između čvorova (*rain*, *warm*, *clear*).



Slika 2. Prikaz grafa sa vrednostima u čvorovima.

Sledeći graf prikazuje povezanost čvorova na osnovu podataka iz modela Markovljevih lanaca, korišćenjem *simple_A\$estimate* parametra i tranzicione matrice između čvorova (*rain*, *warm*, *clear*), generisanog u prethodnim koracima.



Slika 3. Prikaz povezanosti čvorova u grafu na osnovu podataka u matrici.

VI. ZAKLJUČAK

U radu je pokazano kako se probabilistički koncepti koji su izraženi kroz jednačine mogu implementirati kao objekti i metode.

Na osnovu podataka u vektoru, odnosno matrici, sa verovatnoćama i prognozom kakav će dan biti, možemo na osnovu vrednosti zaključiti da će dan biti topao, a po

verovatnoći i sledeći dan bi bio takav, jer vrednost čvora *warm* ima najveću verovanoću u matrici sa vrednošću 0.82. Takođe se može videti da je i verovatnoća posmatranog stanja najveća za vrednost *warm*, u vrednosti od 0.47.

Ovde smo demonstrirali upotrebu modela sa pronalaženjem skora, otkrivanjem latentnog varijabilnog lanca i primenili proceduru obuke. Podaci dobijeni primenom programskog jezika R i vizualizovani korišćenjem tri pomenuta paketa za numeričke vrednosti pružaju dobru polaznu osnovu za korišćenje Markovljevih lanaca koji bi se primenjivao i u ekologiji za prognoze visine vodostaja reka, ili temperature vazduha na osnovu podataka sa meteoroloških stanica.

Na kraju se primenom paketa *igraph* vrši vizuelizacija modela Markova koja prikazuje povezanost čvorova u grafu, sa ili bez težina čvorova.

ZAHVALNICA

Na kraju ovoga rada 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] Marković J., Lanci Markova sa diskretnim vremenom i promenom, Beograd, 2014
- [2] S. Lipschutz, Theory and problems of Probability, Schaum's Outline series, McGraw-Hill Book Company New York, 1968.
- [3] W. L. Winston, Operations research, Applications and algorithms, Thomson learning Brooks/Cole, 2004.
- [4] <https://rpubs.com/mingo1226/hmm>
- [5] <https://towardsdatascience.com/hidden-markov-model-implemented-from-scratch-72865bda430e>

Proračun tokova snaga u distributivnim mrežama

Jovana Stanković

Sadržaj - Cilj ovog rada je prikazati opšti značaj tokova snaga, kao i probleme na koje se nailazi pri proračunu tokova snaga u distributivnim mrežama. U radu je opisan iterativni metod za proračun tokova snaga u radijalnim distributivnim mrežama zasnovan na primeni Kirhofovih zakona. Na temelju ovog metoda rešen je problem proračuna tokova snaga i naponskih prilika na primeru distributivne mreže IEEE 33.

I. UVOD

Razvojem društva, zahtevi koji se postavljaju pred elektroenergetsku mrežu u vidu pouzdanosti i sigurnosti snabdevanja električnom energijom postali su sve veći. Potrošači očekuju od mreže da bude pouzdana i da električna energija zadovoljavajućeg kvaliteta bude dostupna u svakom trenutku. Da bi se to ostvarilo izuzetno je važno održati stabilne naponske prilike u mreži.

Distributivne mreže, kao završna karika u prenosu električne energije do potrošača, čine veoma bitan deo elektroenergetskog sistema. Da bi se ispunili zahtevi po pitanju električne energije usled rasta industrijske proizvodnje i povećanja gustine opterećenja u urbanim delovima gradova neophodan je ubrzani razvoj ovih mreža. Poslednjih decenija, distributivnu mrežu je bilo potrebno konstantno nadograđivati i prilagođavati novim zahtevima. Ova činjenica povećala je potrebu za analizom stanja distributivnih mreža pri različitim konfiguracijama i opterećenjima, sve u cilju određivanja optimalnih uslova za njen rad.

Osnovni proračuni koji se sprovode, kako za efikasno upravljanje postojećim sistemom, tako i za potrebe eksploatacije, planiranja i razvoja sistema, jesu proračuni tokova snaga. Tokovi snaga kroz mrežu uzrokuju padove napona na elementima mreže i na taj način značajno utiču na kvalitet snabdevanja potrošača električnom energijom, pa je analiza tokova snaga neizostavan deo za ostvarenje optimalnog rada sistema.

II. TOKOVI SNAGA

Problem određivanja tokova snaga može se opisati kao zadatak nalaženja napona i snaga injektiranja u svim čvorovima i tokova snaga po svim vodovima mreže [1]. Poznavanje stanja sistema je preduslov za verifikaciju rešenja koja se dobijaju prilikom istraživanja raznih varijanti za njegovo unapređenje. Proračunima tokova snaga prati se uticaj promene konfiguracije mreže do koje

J. Stanković je student na modulu Elektroenergetika, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: stankovicjovana@elfak.rs

dolazi zbog dodavanja novih novih ili zbog isključivanja postojećih elemenata sistema. Tako dobijamo uvid u opterećenje vodova i napone čvorova koji potencijalno mogu uzrokovati probleme u radu sistema ukoliko su van propisanih granica.

Prvobitni metodi koji su korišćeni za potrebe proračuna tokova snaga bili su primenjivani u prenosnim mrežama. Neki od tih metoda su: Gaussov Gauss-Seidelov, Newton-Raphsonov metod i njegove modifikacije. Ovi metodi se sve do danas uspešno i široko koriste za analizu rada prenosnog sistema. Međutim, pokušaj primene ovih metoda u distributivnim mrežama pokazao je loše konvergencijske sposobnosti. Razlog za to leži u različitoj prirodi prenosnih i distributivnih mreža, što presudno utiče na date proračune

Distributivne mreže, za razliku od prenosnih mreža, karakteriše [2]:

- radijalni pogon ili relativno slaba potencijalna upetljanost;
- velika dimenzionalnost (veliki broj vodova i transformatorskih stanica);
- znatno veći odnos aktivne otpornosti i reaktanse vodova ($R/X \gg 0$);
- često nepouzdate i nepotpune informacije o opterećenjima.

Ove fizičke razlike između prenosnih i distributivnih mreža dovele su do toga da poznati metodi za proračun tokova snaga nisu postigli zadovoljavajuću tačnost i brzinu konvergencije. Iz tog razloga, za potrebe dobijanja informacija o stanju sistema i efikasnije analize distributivnih mreža bilo je neophodno razviti drugačije algoritme u kojima bi se iskoristila specifična radijalna priroda ovih mreža [3].

III. METODI ZA PRORAČUN TOKOVA SNAGA DISTRIBUTIVNIH MREŽA

Poslednjih decenija, na osnovu poznavanja karakteristika distributivnih mreža, razvijen je niz vrlo efikasnih iterativnih numeričkih postupaka koji umesto matričnog koriste pristup orjentisan ka granama mreže koja se analizira. Na taj način, potisnuta je potreba za rešavanjem matričnih sistema jednačina, a zahtevi za memorijom i vreme proračuna su znatno smanjeni [3].

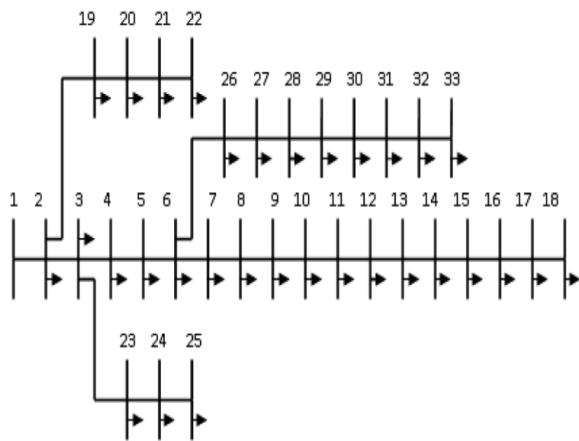
Osnovni metod koji se koristi u analizi distributivnih mreža je Shirmohamadijev iterativni metod [4]. Algoritam je veoma jednostavan, budući da se bazira na dva osnovna zakona elektrotehnike, prvom i drugom Kirhofovom zakonu. Ovaj metod u osnovi je razvijen za proračun tokova snaga radijalnih uravnoteženih mreža. Međutim, zahvaljujući

dobroj matematičkoj zasnovanosti, Shirmohamadijem metod, uz određena poboljšanja i modifikacije, vrlo efikasno je primenjen i za proračun tokova snaga slabo upetljenih [4] i neuravnoteženih mreža [5].

IV. PRORAČUN TOKOVA SNAGA IEEE33 DISTRIBUTIVNE MREŽE

U okviru ovog poglavlja biće izvršena implementacija iterativnog postupka koji se zasniva na algoritmu „korak napred/korak nazad” i primeni Kirhofovih zakona. Efikasnost ovog pristupa leži u njegovoj jednostavnosti pri implementaciji i brzini proračuna. Verifikacija algoritma je izvršena je na primeru IEEE 33 test mreže u programskom paketu Matlab. Za potrebe proračuna razvijen je programski kod u okviru .m fajla.

Ova mreža je radijalna distributivna mreža koja se sastoji od 33 čvorova i 32 grane. Jednopolna šema mreže i numeracija čvorova prikazani su na slici 1.



Slika 1. Jednopolna šema IEEE 33 mreže.

Proračun tokova snaga se sastoji u proračunu promenljivih stanja distributivne mreže na bazi poznatog napona izvora napajanja mreže, električnih parametara vodova i poznatih potrošnji u svim čvorovima mreže. Čvor sa indeksom 1 je izvorni čvor mreže, dok su u ostalim čvorovima priključeni potrošači sa specificiranim snagama potrošnje.

Podaci o opterećenjima dobijaju se prikupljanjem injektirane aktivne i reaktivne snage u svakom čvoru mreže. Podaci o vodovima odnose se na vrednosti otpornosti i reaktansi svake grane mreže.

Realizacija proračuna

Proračun tokova snaga posmatrane mreže izvršen je u programskom paketu Matlab. Ulazni podaci programa su: napon napojnog čvora, snaga koju potrošači preuzimaju iz pojedinih čvorova, impedanse vodova, informacije o povezanosti čvorova i željena tačnost. Snage koje potrošači

preuzimaju iz pojedinih čvorova sastoje se od aktivne i reaktivne komponente. U programu se pretpostavlja da je ta snaga konstantna, što je uobičajna pretpostavka za ovakve tipove proračuna. Impedanse grana su kompleksne veličine i prikazane su preko vrednosti otpora i reaktansi. Podaci o opterećenjima čvorova, električnim parametrima vodova i povezanosti čvorova date mreže preuzeti su iz reference [6]. Napon izvornog čvora iznosi 12,66 kV dok se za baznu snagu uzima vrednost od 100MVA.

Nakon učitavanja ulaznih podataka mreže sve veličine svode se na relativne vrednosti. Naponi svih potrošačkih čvorova postavljaju se na vrednost napona izvornog čvora i pokreće se iterativni proces. Iterativni postupak čine sledeći koraci:

Korak 1: Na osnovu podataka o snagama injektiranja računaju se injektirane struje u svaki čvor mreže pomoću izraza:

$$I_i^{(v)} = \left(\frac{P_i + Q_i}{U_i^{(v-1)}} \right)^*, i = 1, 2, \dots, n, \quad (1)$$

gde su:

$I_i^{(v)}$ - struja injektiranja u čvoru i , u v -toj iteraciji,

$U_i^{(v-1)}$ - napon čvora i u $v-1$ iteraciji,

P_i - aktivna injektirana snaga u čvoru i ,

Q_i - reaktivna injektirana snaga u čvoru i ,

n - broj čvorova u mreži.

Korak 2: U ovom koraku vrši se izračunavanje struja koje teku kroz grane mreže, i to počevši od grane sa najvećim indeksom („korak unazad”). Algoritam najpre proverava koji čvorovi se nalaze na ograncima mreže i ne napajaju nijednu granu. Ukoliko krajnji čvor grane i ne napaja nijednu granu, struja kroja teče kroz tu granu jednaka je struji potrošnje datog čvora. U suprotom program traži indekse svih grana koje ističu iz datog čvora i privremeno ih skladišti. Na osnovu tih podataka vrši se izračunavanje struja kroz datu granu mreže. Izračunavanje struja kroz granu i , koja povezuje čvorove i i $i+1$, vrši se prema sledećoj relaciji:

$$J_i^{(v)} = -I_{i+1}^{(v)} + \sum_{k \in \alpha_{i+1}} J_k^{(v)}, i = n-1, \dots, 2, 1, \quad (2)$$

gde je:

$J_i^{(v)}$ - struja grane i na v -toj iteraciji,

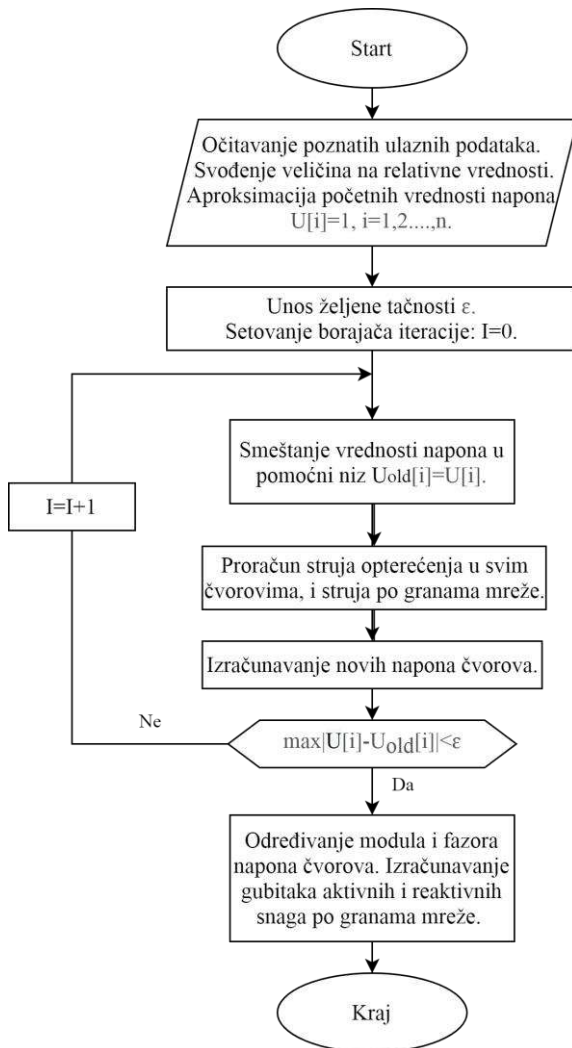
α_{i+1} - skup grana koji ističu iz čvora $i+1$.

Korak 3: Na osnovu izračunatih struja koje teku kroz grane mreže računaju se nove vrednosti napona čvorova počevši od prvog čvora („korak unapred”). Napon i -tog čvora računa se prema relaciji:

$$U_i^{(v)} = U_{i-1}^{(v)} - Z_{i-1} J_{i-1}^{(v)}, i = 2, 3, \dots, n. \quad (3)$$

Ovi koraci ponavljaju se sve dok se ne ostvari željena tačnost, odnosno sve dok razlika napona svih čvorova dve uzastopne iteracije ne bude manja od unapred zadate vrednosti ϵ .

Izloženi postupak proračuna prikazan je u obliku blok dijagrama na slici 2.



Slika 2. Blok dijagram predloženog algoritma proračuna.

U nastavku će biti prikazan programski kod koji je primenjen za potrebe proračuna. U programu su implementirani prethodno opisani koraci proračuna, a sama realizacija proračuna u Matlabu dodatno je pojašnjena kroz propratne komentare u okviru koda. Podaci neohodni za proračun nalaze se u dva .m fajla i u programu se učitavaju korišćenjem funkcije load. U fajlu linedata33bus.m nalaze se podaci o parametrima vodova i informacije o povezanosti čvorova, odnosno konfiguraciji mreže, dok fajl loaddata33bus.m sadrži podatke o snagama opterećenja čvorova koji su prikazani preko aktivne i reaktivne komponente.

Prikaz matlab koda:

```

LD=load('linedata33bus.m'); %parametri
vodova
BD=load('loaddata33bus.m'); %podaci o
opterećenjima cvorova

Sbase=100; %bazna snaga (MVA)
Vbase=12.66; %bazni napon (KV)
Zbase=(Vbase^2)/Sbase;

%svodjenje velicina na relativne
vrednosti
LD(:,4:5)=LD(:,4:5)/Zbase;
BD(:,2:3)=BD(:,2:3)/(Sbase*1000);

N=max(max(LD(:,2:3)));
Sload=complex(BD(:,2),BD(:,3));
V=ones(size(BD,1),1); %inicijalizacija
napona cvorova

Z=complex(LD(:,4),LD(:,5));
Iline=zeros(size(LD,1),1);
E=0.0001; %zadata tačnost
I=0;
K=0;
while K==0
    Iload=conj(Sload./V); %struje
    opterećenja
    V2=V;

    %korak nazad
    %izracunavanje struja kroz grane
    mreze
    for j=size(LD,1):-1:1
        c=[];
        e=[];
        [c e]=find(LD(:,2:3)==LD(j,3));
        if size(c,1)==1 %ukoliko vektor c
            sadrzi samo jedan element grana j ne
            napaja nijednu drugu granu, struja kroz
            datu granu jednaka je struji
            injektiranja u završnom cvoru te grane
            Iline(LD(j,1))=Iload(LD(j,3));
        else
            Iline(LD(j,1))=Iload(LD(j,3))+
            sum(Iline(LD(c,1))-Iline(LD(j,1)));
        end
    end

    %korak napred
    %izracunavanje napona cvorova
    for j=1:size(LD,1)
        V(LD(j,3))=V(LD(j,2))-
        Iline(LD(j,1))*Z(j);
    end

    I=I+1; %broj iteracija
    if max(abs(V2-V))<E %ispitivanje uslova
        konvergencije
  
```

```

K=1;
end
end

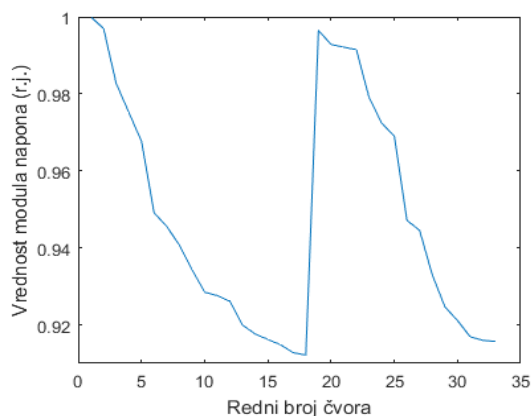
Ilinea=abs(Iline);
Voltage=abs(V)
P=real(Z.*(Ilinea.^2))*100000
Q=imag(Z.*(Ilinea.^2))*100000
S=(Z.*(Ilinea.^2))*100000
Puk=sum(P) %ukupni gubici aktivne
snage
Quk=sum(Q) %ukupni gubici reaktivne
snage

```

Analiza rezultata proračuna

Kao rezultat program daje sledeće podatke: vrednosti napona čvorova, struje koje teku kroz grane mreže, gubitke snaga po granama, kao i ukupne gubitke aktivne i reaktivne snage mreže. Sa tako dobijenim rezultatima moguće je izvršiti analizu mreže.

Na slici 3 dat je grafički prikaz modula napona po čvorovima mreže. Kako dozvoljeni pad napona u distributivnim mrežama iznosi $\pm 10\%$ nominalnog napona, na osnovu prikazanih vrednosti zaključuje se da se naponi svih čvorova nalaze se u dozvoljenim granicama. Makimalni napon je napon čvora referentnog čvora 1, a minimalni napon javlja se u čvoru 18. Čvor 18 je najudaljeniji od referentnog čvora pa je samim tim napon tog čvora najniži.



Slika 3. Grafički prikaz vrednosti modula napona po čvorovima mreže.

Na osnovu određenih struja grana i napona čvorova izračunati su gubici aktivne i reaktivne snage u svim granama mreže. Ukupni gubici aktivne snage u svim vodovima mreže iznose 205.2004 kW, a gubici reaktivne snage 136.7110 kVAR.

Konvergenција se dostiže posle svega 4 iteracije. Ovim je potvrđeno da je primenjeni metod vrlo efikasan sa aspekta brzine proračuna. Ova brzina konvergencije dostiže se zbog činjenice da se u radialnoj mreži sa ovakvom početnom aproksimacijom napona i poznatim snagama potrošnje svih potrošačkih čvorova, već u prvoj iteraciji dobijaju se vrednosti struja po svim granama sa velikom tačnošću. Zatim je sa tako određenim strujama moguće značajno poboljšati početne pretpostavljene vrednosti napona u svim čvorovima.

V. ZAKLJUČAK

Poznavanje stanja distributivne mreže je preduslov za verifikaciju rešenja koja se dobijaju prilikom istraživanja raznih varijanti za njeno unapređenje. Iz tog razloga veoma je važna primena efikasnog algoritma za proračun tokova snaga mreže.

Pojava digitalnih računara i raznih softverskih programskih alata znatno je olakšala i povećala tačnost proračuna tokova snaga. U okviru rada izvršena je analiza IEEE33 test mreže u programskom paketu Matlab. Posmatranjem dobijenih rezultata dolazi se do zaključka da se rezultati nalaze unutar dozvoljenih granica i da je data mreža dobro projektovana za dati naponski nivo. Ovim proračunom potvrđene su dobre konvergenćijske sposobnosti i tačnost primenjenog metoda.

ZAHVALNICA

Autor se zahvaljuje profesoru Aleksandru Janjiću na podršci i pruženoj pomoći u toku pisanja ovog rada.

LITERATURA

- [1] D. Tasić, *Analiza elektroenergetskih mreža*. Elektronski fakultet u Nišu, 2010.
- [2] N. Rajaković and D. Tasić, *Distributivne i industrijske mreže*. Akademska misao, 2008.
- [3] D. Popović, D. Bekut and V. Treskanica, *Specijalizovani DMS algoritmi*. DMS Group, 2004.
- [4] D. Shirmohammadi, H. Hong, A. Semlyen and G. Luo, "A compensation-based power flow method for weakly meshed distribution and transmission networks", *IEEE Transactions on Power Systems*, 1988, vol. 3, no. 2, pp. 753-762.
- [5] C. Cheng and D. Shirmohammadi, "A three-phase power flow method for real-time distribution system analysis", *IEEE Transactions on Power Systems*, 1995, vol. 10, no. 2, pp. 671-679.
- [6] M. Baran and F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing", *IEEE Transactions on Power Delivery*, 1989, vol. 4, no. 2, pp. 1401-1407.

Model-Driven Approach to VHDL-based Logic Design

Nenad Petrović

Abstract – In this paper, it is explored how the adoption of model-driven approach could be beneficial to automated logic design. As outcome, a framework for automated generation of VHDL systems starting from user-defined model relying on Eclipse Modeling Framework is introduced. Two methods of system modelling are covered: functional and state machine-based. Evaluation is performed in case of simple COVID-19 protection system design. According to the achieved results, the proposed approach significantly speeds up the time required for digital system prototyping.

I. INTRODUCTION

In 1983, US Department of Defense (DoD) granted the support to research project aiming the development of standardized methods for documenting the behavior of digital systems [1, 2]. It involved three major companies from this field: Texas Instruments, IBM and Intermetrics [1]. As outcome, a programming language-alike product named Very High-Speed Integrated Circuits Hardware Description Language (VHDL) was released in 1985 and transferred to IEEE for standardization that would enable its wider adoption. Moreover, it was also extended with aspects enabling simulation of digital systems based on their description. The first standardized version has emerged in 1987 and called IEEE 1076-1987, while subsequent revisions were done in 1993 and 2008 as well and is widely adopted even nowadays. While almost any possible digital system can be modelled using VHDL, only a subset of their functionalities can be synthesized and physically implemented [1, 2]. However, despite its intuitiveness, the manual creation and verification of complex systems relying on VHDL can be quite challenging task [1, 2]. Therefore, in this paper the focus is on adoption of model-driven approach to enable automated logic design for purpose of rapid prototyping, starting from high-level system description. For this purpose, we rely on metamodel which defines the structure of user-created model instances. Moreover, Ecore within Eclipse Modelling Framework (EMF) [3] is used to generate adapter classes that enable model instance parsing and traversal which are crucial for code generation.

II. BACKGROUND

A. VHDL

In case of VHDL-based logic design, crucial asset is .vhd file which describes a digital system and consists of 3 main parts [1, 2] as summarized Table I.

N. Petrović is PhD student and teaching assistant at the Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nenad.petrovic@elfak.ni.ac.rs

TABLE I
.VHD FILE STRUCTURE

Part	Description	Template
Libraries and packages import	Provide the basic capabilities, such as operators, types and functions	library lib_name; use lib_name.package_1.all
entity	Defines system interface in form of ports. Direction mode (in, out, inout or buffer) and type have to be provided for each of them.	entity <e_name> is port (port1: <mode> <type>; port2: <mode> <type>; end entity ;
architecture	Describes the behavior of a system.	Architecture <a_name> of <e_name> is --user-defined enums --signals --constants --components begin --behavior description end

Furthermore, architecture gives the ability to specify system's behavior using various level of abstraction [1, 2], as shown in Table II.

TABLE II
ARCHITECTURE SPECIFICATION ABSTRACTION LEVELS

Abstraction level	Template
Concurrent assignments and logic operators	<sig ₁ > <= <in ₁ > op <in ₂ > <sig ₂ > <= <sig ₁ > op <sig ₂ > <out> <= <sig ₁ > op ... <sig _n > op = {and, or, xor, not}
Conditional assignments	<signal_name> <= expr1 when cond1 else expr2 when cond2 else ... exprN;
Selection-based assignment	with <input_name> select <signal_name> <= expr1 when cond1, ... exprN when others;
Component-based	<instance_name> : <component_name> portmap (port1=>signal1,... portN=>signalN);

The main properties of VHDL language are [1, 2]: 1) not case-sensitive 2) comments start with "--" 3) assignments and declarations end with ";" 4) all user-defined names start with letters and different than keywords. When it comes to data, the following main types are supported [1, 2]: bit {0,1}; Boolean {true, false}; character {256 ASCII characters}, integer, real, bit_vector {end_index downto begin_index}.

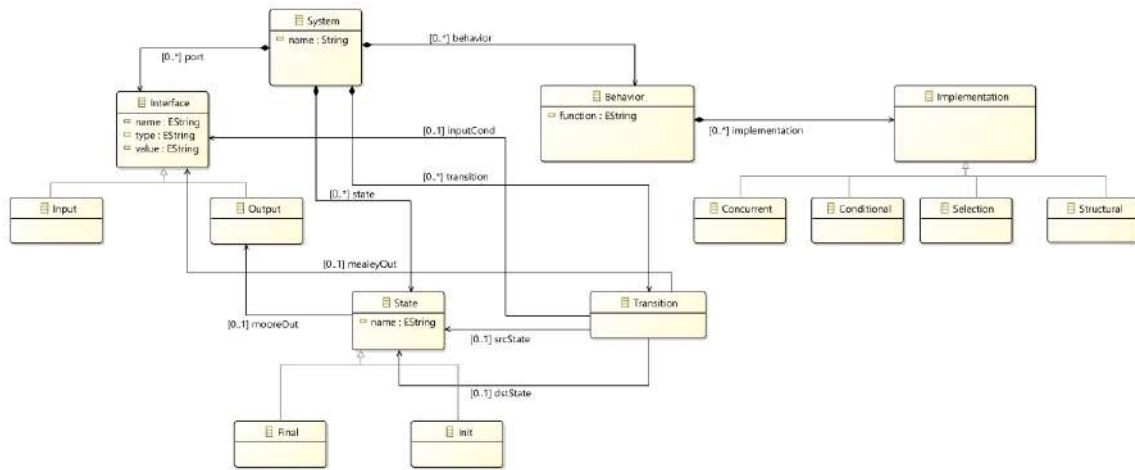


Fig. 1. Logic Design Metamodel (LDM).

B. Ecore

Ecore provides the capabilities for construction of meta-models that can be further used as templates for user-created model instances whose structure complies with meta-model definition. Furthermore, it provides automatically generated adapter classes for model instance manipulation that enable model parsing and traversal, which is crucial for parameters retrieval within the VHDL code generation algorithm.

In related works, Ecore was adopted for various purposes. In [4], the approach based on meta-models was leveraged for automated Gmock test generation in area of automotive software. On the other side, in [5], Ecore was used for automatic animated 3D web presentation generation starting from user-provided model. Finally, in [6] we used it for component allocation problem representation for purpose of multi-objective optimization. In all these cases, such approach resulted with quite fast code generation time, even better than other methods, such as ontologies [5].

III. IMPLEMENTATION

A. Logic Design Metamodel (LDM)

The goal of this metamodel is to define structure of digital system model instances, covering the cases of design based on either logic functions or state machine. The highest-level concept is *System* which consists of *Interface* and *Behavior*. Furthermore, *Interface* could be either *Input* (entering the system as it is) or *Output* (produced as result of transformation inside the system). On the other side, *Behavior* refers to process of transformation performed against the *Inputs* in order to produce the *Outputs*. When it comes to function-based design, the transformation itself is described as logic function of *Inputs* within the string-type *function* attribute. On the other side, in case of state machine representation,

System consists of *States* and *Transitions*. Each *State* has distinct name, while reference to the *Output* with value is optional, and is used only in case of Moore's finite state automaton. Furthermore, there are two types of special states – *Initial* and *Final*. On the other side, *Transition* has references to source and destination *States* (*srcState* → *dstState*) and input condition (*Input* type) which triggers the transition. Additionally, for Mealy's automaton, the output value (*Output* type) should be specified as well.

Considering that VHDL supports different ways of specifying architecture, *Behavior* has *Implementation* which could be either *Concurrent*, *Conditional*, *Selection* or *Structural*. Each type of *Implementation* corresponds to one of the previously mentioned levels of abstraction used for architecture definition. In this paper, for functional design we make use of concurrent assignments, while conditional assignments are used for state-based representation. In Fig. 1, the LDM metamodel is depicted in form of UML class diagram.

B. Code generation algorithms

Transformation algorithm takes the Ecore model instance in either functional or state machine form as input and produces VHDL system code as output.

In this paper, for case of functional design, the focus is on concurrent assignments behavior description, while it is assumed that logic function representing system is given as sum of products. In Table III, pseudo-code of VHDL generation algorithm in case of concurrent assignment is shown. First, the model is parsed and its elements assigned to Ecore adapter classes generated automatically starting from metamodel. Moreover, the interface elements (*Inputs* and *Outputs*) are retrieved together with their parameters (name and type), so the corresponding entity element can be generated. After that, the *Behavior* element is retrieved together with function string for purpose of architecture code generation. At the beginning of architecture definition, for each of input ports, corresponding negation element is defined. Furthermore, the function string is

processed and broken down into products. Each product is separated into literals (denoting ports) and corresponding concurrent assignment lines generated. Finally, the last line of behavior specification defines OR operation against the all of the previously generated products.

TABLE III
CONCURRENT ASSIGNMENT VHDL GENERATION ALGORITHM

```

Input: model_path
Output: VHDL code
Steps:
1. system=parse(model_path)
2. inputs:=system.inputs
3. name:=system.name
4. "entity <system.name> is port("
5. For each input i in inputs
6.   "<i.name> : <i.direction> <i.type>"
7. For each output o in outputs
8.   "<o.name><o.direction><o.type>);"
9. "end entity"
10. F_string:=system.behavior.function;
11. Terms:=f_string.split("+");
12. Literals:=distinct_chars(terms);
13. "architecture <system.name>arch of <system.name>"
14. "begin"
15. For each literal in literals
16.   "<literal>n<=not <literal>"
17. End for;
18. i:=1
19. For each term t in terms
20.   l:=term.split("*");
21.   "<i>i<=<l[o]>and<l[1]>and<l2>);"
22.   i++
23. End for;
24. "<o.name><=)"
25. For each term t in terms
26.   If(t is terms.last())
27.     "<t>);"
28.   Else
29.     "<t> or"
30. "end architecture"
31. End

```

However, when it comes to state machine-based design, we focus on Mealy's automatons, the following VHDL process template for conditional assignments based on [ref] is parametrized for each of the states *model.state[i]*, as shown in Fig. 2.

```

WHEN <model.state[i]> =>
  IF (Input = <transition[i][1].inputCond>) THEN
    nextState <= <transition[i][1].dstState>;
    result <= <transition[i][1].mealyOut>;
  ELSE
    nextState <= <transition[i][0].dstState>;
    result <= <transition[i][0].mealyOut>;
  END IF;

```

Fig. 2. VHDL process template for Mealy automaton.

In this case, for i^{th} state, the corresponding transition is performed by updating the *nextState* VHDL variable with *dstState* value only in case that *inputCond* of transition for input value "1" holds (denoted as *transition[i][1]*).

Moreover, the corresponding output for that transition is also generated and placed within *result* variable. Otherwise, the *nextState* and *result* would be taken for transition in case of "0", denoted as *transition[i][0]*.

IV. EXAMPLE CASE STUDY

The case studies used for evaluation of the proposed model-driven framework are based on author's previous works related to COVID-19 indoor safety systems [8-10]. In what follows, an example for state-based Mealy automaton depicted in Fig. 3 will be described.

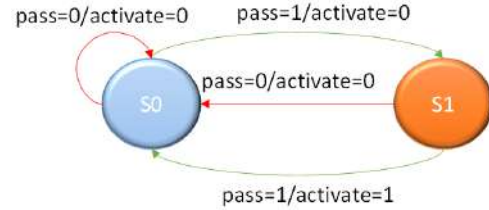


Fig. 3. Indoor air quality control system for COVID-19 prevention.

This system represents indoor air quality control device aiming the reduction of COVID-19 spread. In literature, it was stated that certain air quality conditions (such as temperature 26-27°C and less than 40% humidity) suitable for coronavirus [8]. In this case, we have two states representing different air quality checks: S0 – humidity and S1 – temperature. There are two possible input values for PASS: "1" – air quality parameter is inside the given range; "0" – outside. Finally, there are two possible outcomes for output variable ACTIVATE: "1" – filtering and conditioning device will be started; "0" – otherwise. In Fig. 4, an excerpt of VHDL code describing the system behavior based on conditional assignments is given.

```

BEGIN
  nextState <= S0;
  CASE currentState IS
    WHEN S0 =>
      IF (pass = '1') THEN
        nextState <= S1;
        count <= "0";
      ELSE
        nextState <= S0;
        count <= "0";
      END IF;
    WHEN S1 =>
      IF (pass = '1') THEN
        nextState <= S0;
        count <= "1";
      ELSE
        nextState <= S0;
        count <= "0";
      END IF;
  END CASE;
END PROCESS;

```

Fig. 4. VHDL process for Mealy automaton of COVID-19 indoor air quality control system.

V. EVALUATION

In Table IV the results for code generation times for two approaches (functional and state machines) are given in case of different COVID-19 protection systems (air quality; temperature and mask check) and sizes (number of products within sum and states). Moreover, the obtained results (including the time needed for manual Ecore model creation) are compared to estimations of manual .vhd file creation without relying on the proposed method. It can be seen that state machine-based code generation requires more time, while the speed-up achieved using the proposed approach increases with model size.

Evaluation was performed on a laptop equipped with Intel i7 7700HQ 2.8GHz quad-core processor, 16GB DDR4 RAM and M1 SSD disk.

TABLE IV
RESULTS SUMMARY

System	Approach	Size	Gen. time [s]	Model. time [s]	Speed-up [times]
Air quality [8]	Terms	1	0.015	68	12
	States	2	0.051	154	17
	Terms	2	0.027	72	15
	States	3	0.079	171	19
Temperature and mask check [9, 10]	Terms	3	0.042	77	18
	States	4	0.097	194	26

VI. CONCLUSION AND FUTURE WORK

In this paper, a model-driven approach to function and state machine-based automated VHDL logic design relying on code generation was introduced. According to the achieved results, its adoption is highly beneficial, as it reduces the time necessary for creation of VHDL systems, especially for larger and complex solutions.

In near future, it is planned to adopt a tool based on the presented work for student exercises check, in order to support the first cycle of Logic Design course (accreditation 2019) at second year of Computer Science

within Faculty of Electronic Engineering, University of Niš, Serbia.

ACKNOWLEDGEMENT

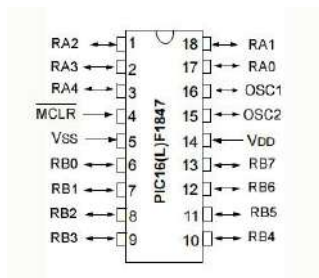
This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

- [1] B. J. LaMeres, *Introduction to Logic Circuits & Logic Design with VHDL*, Springer, 2017.
- [2] VHDL tutorial [online]. Available on: https://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html, last accessed: 26/10/2021.
- [3] Eclipse Modeling Framework [online]. Available on: <https://www.eclipse.org/modeling/emf/>, last accessed: 26/10/2021.
- [4] N. Petrović, M. Radenković, S. Cvetković, D. Rančić: "Model-driven automated gMock test generation for automotive software industry", XV International SAUM, pp. 1-4, 2021.
- [5] N. Petrović, M. Radenković, V. Roblek, M. Khokhobaia, I. Gagnidze, "Model-driven Approach to Interactive 3D Web Presentation Generation", 15th International Online Conference on Applied Electromagnetics - IIEC 2021, pp. 137-140.
- [6] N. Petrović, I. Al-Azzoni, J. Blank, "Model-driven multi-objective optimization approach to 6G network planning", TELSIS 2021, pp. 223-226.
- [7] Melay machine finite state machine design in vhdl [online], available on: <https://www.engineersgarage.com/melay-machine-fsm-vhdl-code/>, last accessed: 26/10/2021.
- [8] N. Petrović, Đ. Kocić, "IoT for COVID-19 Indoor Spread Prevention: Cough Detection, Air Quality Control and Contact Tracing", 2021 IEEE 32nd International Conference on Microelectronics (MIEL), pp. 297-300, September 2021. <https://doi.org/10.1109/MIEL52794.2021.9569099>
- [9] N. Petrović, "COVID-19 Safety Systems Design Exercises in Computer Science University Courses", YuInfo 2021, pp. 1-6.
- [10] N. Petrović, Đ. Kocić, "IoT-based System for COVID-19 Indoor Safety Monitoring", IcETRAN 2020, pp. 603-608

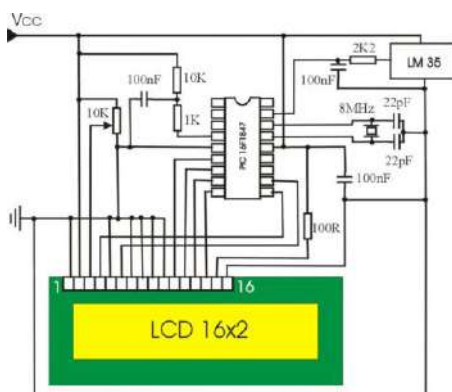


Slika 3. LCD displej.



Slika 4. Raspored pinova kod mikrokontrolera.

Pre same praktične realizacije, termometar je šematski realizovan, što je prikazano na Slici 5.



Slika 5. Šema realizovanog termometra.

Za razvoj ovog termometra korišćena je razvojna ploča proizvođača Mikroelektronika EasyPIC V7 [6].

U mikrokontroler PIC 16F1847 unet je *.HEX fajl sledećeg koda:

```
sbit LCD_RS at LATB4_bit;
sbit LCD_EN at LATB5_bit;
sbit LCD_D4 at LATB0_bit;
sbit LCD_D5 at LATB1_bit;
sbit LCD_D6 at LATB2_bit;
sbit LCD_D7 at LATB3_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;

const float VREF = 5.0;
```

```
float temp,temp_f,temp_res;
char txt[15];
const char character1[] =
{10,14,17,12,2,17,14,0};

void Slovo_sh(char pos_row, char pos_char)
{
    char i;
    Lcd_Cmd(64);
    for (i = 0; i<=7; i++)
        Lcd_Chrc(character1[i]);
    Lcd_Cmd(_LCD_RETURN_HOME);
    Lcd_Chrc(pos_row, pos_char, 0);
}

const char character2[] =
{2,4,14,17,16,17,14,0};

void Slovo_cj(char pos_row, char pos_char)
{
    char i;
    Lcd_Cmd(72);
    for (i = 0; i<=7; i++)
        Lcd_Chrc(character2[i]);
    Lcd_Cmd(_LCD_RETURN_HOME);
    Lcd_Chrc(pos_row, pos_char, 1);
}

void main()
{
    // Kodovi za konfigurisanje portova i
    // inicijalizaciju AD konverzije i displeja:

    ANSELA = 0x07;
    // Konfigurovanje pina A0, A1, A2 kao analogni

    ANSELB = 0x00;
    // Konfigurisanje PORTB kao digitalni port

    TRISA = 1; // Konfigurovanje PORTA kao
    ulazni
    TRISB = 0; // Definisavanje PORTB kao
    izlazni

    ADC_Init(); // inicijalizacija A/D
    konvertora
    Lcd_Init(); // Inicijalizacija LCD
    displeja
    Lcd_Cmd(_LCD_CLEAR); // Brisanje LCD
    displeja
    Lcd_Cmd(_LCD_CURSOR_OFF);
    // Isključeno prikazivanje kursora

    Lcd_Out(1, 6, "Autori:");
    Lcd_Out(2, 1, " Vuka in Kuli ");
    Slovo_sh (2,7 );
    Slovo_cj (2,15);
    Delay_ms(2500);

    Lcd_Out(2, 1, " Milica Ili ");
    Slovo_cj (2,14);
    Delay_ms(2500);

    Lcd_Out(2, 1, "Marko Arsenovi ");
    Slovo_cj (2,15);
    Delay_ms(2500);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 1, "Temp.C=");
    Lcd_Chrc(1,14,223);
    Lcd_Chrc(1,15,'C');
    Lcd_Out(2, 1, "Temp.F=");
    Lcd_Chrc(2,14,223);
    Lcd_Chrc(2,15,'F');
```

```
temp_res = 0.0;

// Beskonacna petlja
//Kodovi za izracunavanje temperature i
prikaz dobijenih vrednosti:

do {
    temp_res = ADC_Get_Sample(0); // AD
konverzija
    temp = (temp_res*VREF)/10.240;
    // Odredjivanje temperature u celzijusima

    temp_f=temp*1.8+32.0;
    //Odredjivanje temperature u farenhajtima

    FloatToStr(temp, txt);
    //konverzija realnog broja

    txt[4] = 0;
    Lcd_Out(1,8,txt);
    FloatToStr(temp_f, txt);
    txt[4] = 0;
    Lcd_Out(2,8,txt);

    Delay_ms(300);
    // Vremensko kašnjenje 300ms

} while (1);
}
```



Slika 6. Provera koda na razvojnoj ploči.

III. RESULTAT

Rezultat ovog rada je realizovan sistem za merenje temperature sa mikrokontrolerom PIC 16F1847 i integrisanim senzorom temperature LM35. Kada ovaj sistem za merenje temperature bude priključen na napajanje pojaviće se ime prvog autora i biće prisutno 2.5 sekunde, potom će biti prikazano ime drugog autora u trajanju od 2,5 sekunde, pa ime trećeg autora u trajanju od 2,5 sekunde. Posle toga, biće prikazana merena temperatura kako u celzijusima tako i u farenhajtima. Ovaj termometar može biti korišćen u različitim složenim sistemima gde je

potrebno meriti temperaturo. To mogu biti složeni sistemi u industriji ali i za kućne potrebe.



Slika 7. Izgled realizovanog termometra.

IV. ZAKLJUČAK

Ovaj termometar može poslužiti za merenje temperature u svim sferama života. Tačnost je zadovoljavajuća za mnoge primene a pored tog uređaj ima nisku cenu. Može biti uz odgovarajuću doradu napravljen i kao termostat ili sa više diskretnih izlaza koji mogu biti iskorišćeni za povezivanje na PLC.

Nedostaci su veći broj provodnika, kao i način povezivanja jer je bitno gde je napajanje senzora, a gde izlaz.

ZAHVALNICA

Autori rada se zahvaljuju nastavnicima Tehničke škole Vlasotince na izdvojenom vremenu, sugestijama i pomoći pri realizaciji projekta i pisanju rada.

LITERATURA

- [1] <http://ww1.microchip.com/downloads/en/devicedoc/41453b.pdf>
- [2] <https://html.alldatasheet.com/html-pdf/1188696/TGS/LM35/455/7/LM35.html>
- [3] <https://www.mikroe.com/mikroc-pic>
- [4] <https://www.winstar.com.tw/products/character-lcd-display-module/16x2-datasheet.html>
- [5] https://www.yxcxtal.com/u_file/photo/20200311/HC-49S-Datasheet.pdf
- [6] <https://www.mikroe.com/easypic-v7>

Upoređivanje efikasnosti algoritama za rešavanje Sudoku-a

Milica Aleksić, Đorđe Antić

Sadržaj – U ovom radu izvršeno je poređenje nekoliko algoritamskih pristupa programiranja ograničenja (engl. *constraint programming*) za rešavanje problema klasične igre Sudoku. Poređeni algoritmi implementirani su u programskom jeziku Python. Algoritmi su testirani na 100 Sudoku zagonetki dimenzije 9×9 i dobijeni rezultati su tabelarno prikazani.

istraživanja u ovoj oblasti služe za otkrivanje novih načina rešavanja problema programiranja ograničenja, utvrđivanja uskog grla nekih rešenja, registrovanja novih primena ovih tehnika, itd.

I. UVOD

Sudoku je igra koja se bazira na logičkom raspoređivanju brojeva u matrici dimenzija 9×9 . Matrica je podeljena na 9 podmatrica veličine 3×3 . Na početku igre je veći deo matrice prazan, dok se samo u malom broju polja nalaze brojevi od 1 do 9. Cilj igre je da se popuni svako polje u matrici nekom od cifara $\{1, 2, \dots, 9\}$ tako da budu ispunjeni sledeći uslovi:

1. U bilo kom redu ne smeju da se pojave dva ista broja,
2. Bilo koja kolona ne sme imati dva ista broja,
3. Istaknute podmatrice veličine 3×3 ne smeju da sadrže dva ista broja.

								8
1	8				2	3		
	6			5	7			1
	7		9	6				
	9		7		4		1	
				8	1		4	
6			2	4			8	
		4	5				9	3
5								

Slika 1. Primer početne matrice u Sudoku-u.

U radu McGuire et al. pokazali su da standardni sudoku problem dimenzije 9×9 mora da sadrži najmanje 17 inicijalno popunjenih ćelija kako bi imao jedinstveno određeno rešenje, što zovemo logički rešiv problem. Razmatrani algoritmi u ovom radu primenjivani su upravo na logički rešive sudoku zadatke [1].

Automatsko rešavanje sudoku zagonetki predstavlja interesantan problem u oblasti istraživanja inteligentnih sistema. O tome svedoči veliki broj radova u naučnoj literaturi, videti npr. [2]-[5]. Naročito su aktuelna istraživanja za velike dimenzije sudoku matrica na kojima se dobija jasnija slika za poređenje efikasnosti algoritamskih rešenja. Osim poređenja postojećih tehnika rešenja,

M. Aleksić i Đ. Antić su studenti na modulu Računarstvo i informatika na Elektronskom Fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija., E-mail: aleksicmilica@elfak.rs, djordje.antic@elfak.rs

II. PRINCIP RADA

Na primeru od 100 Sudoku zagonetki primenjene su sledeće algoritamske tehnike:

- Pretraživanje sa vraćanjem (engl. *Backtracking*),
- Programiranje zadovoljenja ograničenja (engl. *Constraint satisfaction programming*),
- Celobrojno linearno programiranje (engl. *Integer Linear programming*).

Sve tri navedene paradigme koriste definiciju problema koji se rešava preko tri osnovne komponente:

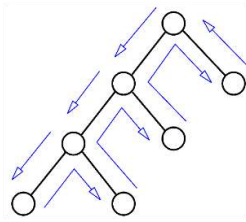
1. Skup od n promenljivih $X = \{x_1, x_2, \dots, x_n\}$, čije vrednosti je potrebno odrediti (polja sudoku-a).
2. Skup od n domena $D = \{D_1, D_2, \dots, D_n\}$, pri čemu je svaki skup $D_i \subseteq \{1, 2, \dots, 9\}$ skup dozvoljenih vrednosti za promenljivu x_i .
3. Skup od p ograničenja $C = \{C_1, C_2, \dots, C_p\}$, gde svako ograničenje definiše dozvoljene kombinacije vrednosti promenljivih [6].

Ograničenja se odnose na uslove rešenja igre sudoku, tj. na broj dozvoljenih pojavljivanja svake od cifara. Ograničenja se definišu pomoću relacija (jednačine, nejednačine ili apstraktne relacije) koje podržavaju lako testiranje. Drugim rečima, za uvedene relacije potrebno je da postoji efikasno algoritamsko rešenje koje služi kao filter za pronalaženje mogućih vrednosti neke promenljive. Glavna razlika između primenjenih metoda u ovom radu upravo se ogleda u načinu zadavanja relacija i samim algoritamskim rešenjima filtriranja.

A. Backtracking

Backtracking je tip iscrpljujuće (*brute-force*) pretrage koja se zasniva na rekurzivnom ispitivanju po dubini. Ova tehnika isprobava nekim utvrđenim redom sve moguće kombinacije do trenutka pronalaženja tačnog rešenja. Postepeno se grade kandidati rešenja, tj. vrednosti za promenljive $x_1, x_2, \dots, x_n$, a odbacuju se one za koje se utvrdi da ne dovode do ispravnog rezultata. Najpre se prolazi kroz jednu celu granu kombinacija vrednosti, i tek kada se sa neuspehom isprobaju sve njene opcije prelazi se na narednu granu koja počinje od prethodne definisane

vrednosti promenljive x_i . Tehnika je bazirana na rekurzivnom pozivu funkcije prilikom isprobavanja svih mogućih kombinacija [7].



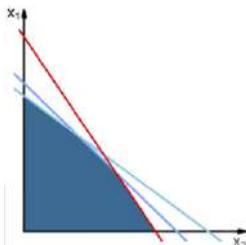
Slika 2. Backtracking tehnika.

B. Constraint satisfaction programming (CSP)

Constraint satisfaction programming je algoritamski pristup poznat u veštačkoj inteligenciji i operacionim istraživanjima. Predstavlja važnu programsku paradigmu za modeliranje i rešavanje složenih kombinatornih problema. Cilj je pronalaženje adekvatne kombinacije vrednosti za promenljive $x_1, x_2, \dots, x_n$ na osnovu dostupnih domena $D_1, D_2, \dots, D_n$ tako da postavljena ograničenja budu zadovoljena. Koristi se neki tip zaključivanja kako bi se postupno redukovali domeni promenljivih bez provere svih mogućih kombinacija vrednosti. U konkretnom slučaju sudoku problema, cilj je simulacija postupaka koje čovek koristi prilikom rešavanja sudoku zagonetki olovkom na papiru. Upotrebljena je heuristika najmanjeg domena, tj. najograničenije promenljive.

C. Linear programming

Linearno programiranje (LP) je tehnika koja se koristi za rešavanje kontinualnih optimizacionih problema. Zadatak je odrediti maksimum ili minimum linearne funkcije koja se naziva funkcija cilja. Specijalno, za probleme diskretne optimizacije koristi se celobrojno linearno programiranje (ILP) koje podrazumeva da se optimum funkcije traži na skupu celobrojnih vrednosti za optimizacione promenljive. Prilikom optimizacije funkcije promenljive moraju biti nenegativne i moraju zadovoljavati linearna ograničenja definisana u obliku matematičkih jednačina ili nejednačina. Problemi koji se rešavaju pomoću linearnog programiranja predstavljaju jedan od najznačajnijih problema donošenja odluka. LP se koristi za rešavanje praktičnih zadataka kao što su planiranje ekonomskog razvoja [8].



Slika 3. Grafički pristup rešavanju problema.

Problem matematičkog programiranja se zadaje preko:

1. Jedne ili više optimizacionih promenljivih
2. Funkcije cilja
3. Skupa ograničenja

Postoje različiti alati u rešavanju linearnog programiranja među koje spadaju i algoritmi. Postoji veći broj programskih paketa za rešavanje ovakvog problema. Tu spadaju MOSEK pisan u C++, LINDO interaktivni softver prilagođen Windows okruženju i grafičkim problemima, i PuLP biblioteka u Python-u koja omogućava opis matematičkih problema [9].

III. IMPLEMENTACIJA ALGORITAMA

A. Backtracking

Rešavanje sudoku-a pomoću backtracking tehnike postignuto je na sledeći način. Pronalaze se prazna polja u sudoku matrici. Za svako pronađeno prazno polje se redom proverava koji broj počevši od jedinice zadovoljava rešenje. Ukoliko je konstatovano da nijedan broj od 1 do 9 ne ispunjava ograničenja sudoku igre, jer narušava neko od pravila jedinstvenosti, zaključuje se da je izbor za prethodno popunjeno polje bio loš. Pronalazi se prethodno popunjeno polje i inkrementira se njegova vrednost sve dok se ne pronađe rešenje koje zadovoljava sve uslove sudoku igre.

```
def find_empty_grid(x): # trazi prazna mesta ili '0'.
    for i in range(len(x)):
        for j in range(len(x[0])):
            if x[i][j] == 0:
                return (i, j) # koordinate praznih mesta, vrsta i kolona
    return None

def solve_array(x):
    find = find_empty_grid(x) # trazi prazna mesta
    if not find:
        return True
    else:
        row, col = find

        for i in range(1,10):
            if valid(x, i, (row, col)): # proverava ogranicenja
                x[row][col] = i

                if solve_array(x):
                    return True

                x[row][col] = 0

        return False

def valid(x, num, pos): # proverava ogranicenja
    # proverava reda
    for i in range(len(x[0])):
        if x[pos[0]][i] == num and pos[1] != i:
            return False

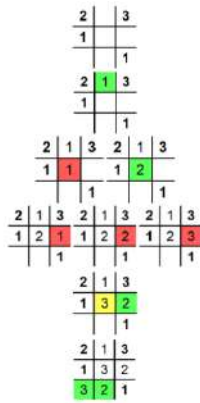
    # proverava kolone
    for i in range(len(x)):
        if x[i][pos[1]] == num and pos[0] != i:
            return False

    # proverava podmatrice
    box_x = pos[1] // 3
    box_y = pos[0] // 3

    for i in range(box_y*3, box_y*3 + 3):
        for j in range(box_x*3, box_x*3 + 3):
            if x[i][j] == num and (i,j) != pos:
                return False

    return True
```

Slika 4. Implementacija Backtracking algoritma.



Slika 5. Backtracking po koracima.

B. Constraint Satisfaction programming (CSP)

Za rešavanje Sudoku-a kreiran je skup promenljivih $X = \{A1, A2, \dots, I8, I9\}$. Slovicima od A do I označeni su redovi, a brojevima od 1 do 9 kolone. Sadrži ukupno 81 element. Domeni promenljivih definisani su skupom brojeva od 1 do 9.

Skup ograničenja se sastoji od tri grupe ograničenja. Prvu grupu čine ograničenja kolona:

$AllDiff(A1, B1, C1, D1, E1, F1, G1, H1, I1), \dots$

$AllDiff(A9, B9, C9, D9, E9, F9, G9, H9, I9)$,

drugu ograničenja redova:

$AllDiff(A1, A2, A3, A4, A5, A6, A7, A8, A9), \dots$

$AllDiff(I1, I2, I3, I4, I5, I6, I7, I8, I9)$

I treću ograničenja podmreža dimenzije 3×3 :

$AllDiff(A1, A2, A3, B1, B2, B3, C1, C2, C3) \dots$

$AllDiff(G7, G8, G9, H7, H8, H9, I7, I8, I9)$.

```
def getRemainingValues(board): #vraca moguće vrednosti za celu sudoku
    RV=[]
    for row in range(9):
        for col in range(9):
            if board[row][col] != 0:
                RV.append(['x'])
            else:
                RV.append(getDomain(board, row, col))
    return RV

def getDomainLength(lst): #vraca velicu domena za to polje
    if 'x' in lst or lst == []:
        return 10
    else:
        return len(lst)

def getNextMRVRowCol(board): #trazi el sa najmanjim domenom
    rv=getRemainingValues(board)
    rvMap = list(map(getDomainLength,rv))
    minimum = min(rvMap)
    if minimum == 10:
        return (-1,-1)
    index = rvMap.index(minimum)
    return(index // 9, index % 9)

def isEmptyDomainProduced(board, row, col, choice):
    rv=getRemainingValues(board)
    element =rv.pop(row*9 + col)
    if [] in rv:
        rv.insert(row*9+col,element)
        return True
    else:
        rv.insert(row*9+col,element)
        return False
```

```
def solveCSPFH(board):
    expandedNodes=0
    location = getNextMRVRowCol(board)
    if location[0] == -1:
        return True
    else:
        expandedNodes+=1
        rv=getRemainingValues(board)

        row = location[0]
        col = location[1]
        for choice in rv[row*9+col]:
            board[row][col] = choice
            cpy = rv
            rv = getRemainingValues(board)

            if not isEmptyDomainProduced(board, row, col, choice):
                if solveCSPFH(board):
                    return True
            board[row][col] = 0
            rv = cpy
        return False

def getDomain(board, row, col): #fja za trazenje domena za polje
    RVCell = [1,2,3,4,5,6,7,8,9]
    for i in range(9):
        if board[row][i] != 0:
            if board[row][i] in RVCell:
                RVCell.remove(board[row][i])
    for i in range(9):
        if board[i][col] != 0:
            if board[i][col] in RVCell:
                RVCell.remove(board[i][col])
    boxRow = row - row%3
    boxCol = col - col%3
    for i in range(3):
        for j in range(3):
            if board[boxRow+i][boxCol+j] != 0:
                if board[boxRow+i][boxCol+j] in RVCell:
                    RVCell.remove(board[boxRow+i][boxCol+j])
    return RVCell
```

Slika 6. Implementacija CSP-a.

Za svako polje u tablici određen je domen. Za već popunjena polja iz unapred definisanog domena izbačene su sve vrednosti osim zadate. Domen ostalih polja modifikovan je na sledeći način. Za svaku vrednost iz domena proveravano je da li narušava neka od ograničenja. Ukoliko narušava izbacuje se iz domena. Traži se polje sa najmanjim brojem elemenata u domenu. To polje predstavlja naredno polje u koje će biti upisana vrednost. Proverava se da li je moguće da upisivanje elementa u to polje dovodi do ispražnjena domena nekog drugog elementa radi zadovoljavanja ograničenja. Ta tehnika se naziva proveravanje unapred (engl. *Forward Checking*).

C. Integer Linear programming (ILP)

Za logički rešiv sudoku problem rešenje je jedinstveno. Drugim rečima ne postoje bolja i gora rešenja sudoku-a. Zbog toga ovo nije pravi optimizacioni problem, pa ciljnu funkciju možemo proizvoljno da izaberemo. Sudoku tabla se sastoji od 9×9 polja koje sadrže vrednosti od 1 do 9 pritom su vrednosti u svakoj vrsti različite, kao i u koloni i u podmreži dimenzije 3×3 . Promenljive čije se vrednosti određuju imaju $729 = 9 \times 9 \times 9$ parametara i indeksiramo ih sa v (označava vrednost u polju), r (označava red) i c (označava kolonu). Ograničenja koje nameće igra sudoku je moguće matematički zapisati indikatorskim promenljivim:

$$\sum_{v=1}^9 x_{vrc} = 1, \text{ za } r, c \in [1,9]$$

$$\sum_{r=1}^9 x_{vrc} = 1, \text{ za } v, c \in [1,9]$$

$$\sum_{c=1}^9 x_{vrc} = 1, \text{ za } r, v \in [1,9]$$

$$\sum_{r=3p-2}^{3p} \sum_{c=3q-2}^{3q} x_{vrc} = 1, \\ \text{ za } v \in [1,9] \text{ i } p, q \in [1,3].$$

Prva jednačina predstavlja ograničenje kojim se definiše da se samo jedna vrednost može naći u jednom polju. Druga i treća da se vrednost može samo jedanput naći u koloni i vrsti, a četvrta da se vrednost samo jedanput može naći u podmreži dimenzije 3×3 . Zatim se kreira instanca problema. Prvi parametar označava naziv problema. Drugi parametar koji je opcioni predstavlja šta je neophodno uraditi sa funkcijom. Kreiraju se promenljive čije se vrednosti izračunavaju. Problemu se dodaju ciljna funkcija i ograničenja. Moguća je i provera statusa problema. Rešavanje LP problema postiže se pozivanjem ugrađene solve funkcije [10].

```

VALS=ROWS=COLS=range(1,10)
Boxes = [
    [(3 * i + k + 1, 3 * j + 1 + 1) for k in range(3) for l in range(3)]
    for i in range(3)
    for j in range(3)
]
prob = LpProblem("Sudoku_Problem")
choices = LpVariable.dicts("Choice", (VALS, ROWS, COLS), cat="Binary")
for r in ROWS:
    for c in COLS:
        prob += lpSum([choices[v][r][c] for v in VALS]) == 1
for v in VALS:
    for r in ROWS:
        prob += lpSum([choices[v][r][c] for c in COLS]) == 1
    for c in COLS:
        prob += lpSum([choices[v][r][c] for r in ROWS]) == 1
    for b in Boxes:
        prob += lpSum([choices[v][r][c] for (r, c) in b]) == 1
for (v, r, c) in input_data:
    prob += choices[v][r][c] == 1
prob.solve()

```

Slika 7. Implementacija ILP-a.

IV. REZULTATI

Upotrebljene su funkcije koje nam pruža paket *time* u programskom jeziku Python radi merenja vremena potrebnog za izvršenje prethodno navedenih algoritama. Na bazi od 100 primera Sudoku zagonetki dobijena su sledeća vremena:

Algoritam	Vreme (s)
Backtracking	11.5458
Constraint Satisfaction Programming	0.0669
Linear Programming	8.4409

V. ZAKLJUČAK

Kao što je i očekivano *Backtracking* tehnika korišćena u *Brute-force* pristupu daje najgore rešenje. Algoritmi bazirani na veštačkoj inteligenciji daju značajno optimalnija rešenja, naročito CSP. Vreme potrebno za izvršenje linearnog programiranja znatno je veće od vremena potrebno za izvršenje CSP što se može pripisati implementaciji gotovih funkcija u sklopu uključenog paketa, koje su korišćene za izvršenje ovog algoritma. Ipak, *Backtracking* tehnika ostaje nezaobilazna za rešavanje sudoku zagonetki koje nisu logički rešive, tj. koje nemaju jedinstveno rešenje. Da bi se redukovala kombinatorna eksplozija koja nastaje prilikom *brute-force* pretrage istraživanja u ovoj oblasti usmerena su ka hibridnim algoritmima. Primenjuje se jedno ili više algoritamskih rešenja programiranja ograničenja kao pretprocesiranje. Iscrpljujuća pretraga nastupa isključivo u slučaju da pretprocesiranje nije dovelo do rešenja.

ZAHVALNICA

Autori rada se zahvaljuju prof. dr Jovani Džunić, sa Elektronskog fakulteta u Nišu na pruženoj pomoći tokom izrade ovog rada.

LITERATURA

- [1] G. McGuire, B. Tugemann, G. Civario, There is no 16-clue Sudoku: solving the Sudoku minimum number of clues problem. Comput Res Repos. arXiv:1201.0749, 2012.
- [2] N. Musliu and F. Winter, "A Hybrid Approach for the Sudoku Problem: Using Constraint Programming in Iterated Local Search," in *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 52-62, Mar.-Apr. 2017, doi: 10.1109/MIS.2017.29.
- [3] H. Simonis, "Sudoku as a Constraint Problem", 2005.
- [4] B. Crawford, C. Castro, E. Monfroy, „Solving Sudoku with Constraint Programming”, In book: *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, 2009.
- [5] I. Lynce, J. Oukanin, Sudoku as a SAT Problem, *Proceedings of AIMATH 06*, 2006.
- [6] Solving Sudoku with AI <https://towardsdatascience.com/solving-sudoku-with-ai-d6008993c7de> (posećen 25.10.2021)
- [7] How to solve sudoku using artificial intelligence. <https://medium.com/swlh/how-to-solve-sudoku-using-artificial-intelligence-8d5d3841b872> (posećen 23.10.2021.)
- [8] P. Stanimirović, N. Stojković, M. Petković, "Matematičko programiranje", Prirodno-matematički fakultet, Univerzitet u Nišu, Niš, 2007.
- [9] S. Mitchell, M. O'Sullivan, I. Dunning, "PuLP: A Linear Programming Toolkit for Python", Department of Engineering Science, The University of Auckland, Auckland, 2011
- [10] Using Integer Linear Programming to Solve Sudoku Puzzles <https://towardsdatascience.com/using-integer-linear-programming-to-solve-sudoku-puzzles-15e9d2a70baa> (posećen 25.10.2021.)

Kreiranje verifikacionog okruženja za Split Transaction Open Core Protocol po UVM metodologiji

Milan Spasić

Sadržaj – Svrha rada jeste da što bolje objasni šta je verifikacija i čemu je ona namenjena. Verifikacija *Open Core Protocol*-a u konfiguraciji *Split Transaction* daje potpunu sliku mogućnosti *UVM* metodologije i *SystemVerilog*-a. Za realizaciju i verifikaciju okruženja korišćeni su profesionalni alati.

I. VERIFIKACIJA

Verifikacija je proces provere ispravnosti bloka ili celog dizajna u skladu sa funkcionalnom specifikacijom. Verifikacija se najčešće izvršava u toku realizacije dizajna. Svako odstupanje rada dizajna od funkcionalne specifikacije označava se kao potencijalna greška. U pogledu na verifikacioni pristup imamo: *IP* verifikaciju i *SoC* verifikaciju. *IP* verifikacija se zasniva na *Coverage Driven Verification (CVD)*, odnosno *UVM* metodologiji, dok se *SoC* verifikacija bazira na komunikaciji procesorskog jezgra sa periferijama. Jezik za opis hardvera i verifikaciju koji se slaže sa pomenutom metodologijom jeste *SystemVerilog*. Njegov koncept zasniva se na objektno - orijentisanom programskom jeziku. Bitno je napomenuti da se *UVM* verifikacija oslanja na randomizaciju. Tabela I prikazuje razliku između randomizovanog i direktnog testiranja.

TABELA I
RANDOMIZOVANO I DIREKTNO TESTIRANJE

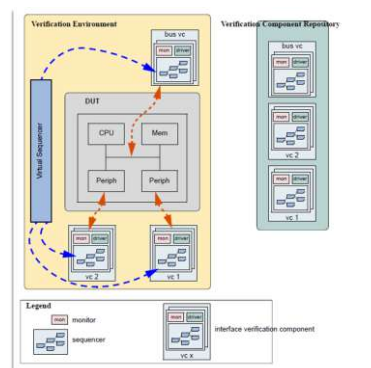
Directed testing methodology	Constrained-random testing methodology
Tests / Runs / Simulations	Tests / Runs / Simulations
T1 * 1 = 1	T1 * 150 = 150
T2 * 1 = 1	T2 * 150 = 150
...	...
T3000 * 1 = 3000	T20 * 150 = 3000

II. VERIFIKACIONO OKRUŽENJE

Verifikaciono okruženje mora biti napravljeno tako da se može ponovo upotrebiti, odnosno mora biti *reusable*. Najčešći naziv za verifikaciono okruženje jeste *VIP* (*Verification Intellectual Property*). Na slici 1 nalazi se

Milan Spasić je verifikacioni inženjer i student master studija na studijskom programu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 1800 Niš, Srbija. E-mail: milan.spasic@elfak.rs

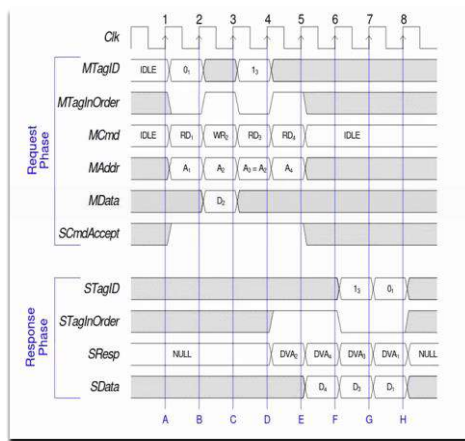
verifikaciono okruženje koje sadrži tri verifikaciona interfejsa. Možemo uočiti da se ova tri interfejsa nalaze na repozitorijumu, odnosno oni su tipa *reusable* i koriste se više puta. Svaki interfejs oponaša signale koji su od ključnog značaja za verifikaciju DUT-a. U okviru jednog UVC (*Universal Verification Component*) nalaze se komponente *driver*, *monitor*, *sequencer*, itd. *Sequencer* je komponenta koja šalje randomizovane sekvence do *driver*-a a zatim ih *driver* prosleđuje na *interface* do DUT-a. *Monitor* kupi svaku promenu na interfejsu od strane okruženja ali i od strane DUT-a. Virtualni *sequencer* je komponenta koja kontroliše rad svih *sequencer*-a.



Slika 1. Primer verifikacionog okruženja.

III. OPEN CORE PROTOCOL SPLIT TRANSACTION

U ovom poglavlju biće ukratko objašnjen *Open Core Protocol (OCP)*. *OCP* je visokoefikasni i magistralno nezavisni interfejs između dva IP jezgara, odnosno on se definiše kao interfejs od tačke do tačke između dva entiteta. Bidirekcionni interfejs koji je sinhronizovan signalom clock i načinom semplovanja na pozitivivnu ivicu clock-a. Jedan entitet se ponaša kao *master* a drugi kao *slave*. Sadrži dva tipa osnovnih komandi *Read* i *Write* i pet dodatnih komandi. *OCP* omogućava *pipeline* transfer. Visoka efikasnost se postiže *burst* transferom. Za realizaciju *Split Transaction* konfiguracije pomenutog protokola korišćene su oznake (*Tags*). One omogućavaju raspodelu transfera u redosledu onakvim kakav je *master* zahtevao. Sekvenca *Split Transaction* Open Core Protocol-a prikazana je na slici 2. Sastoji se iz dve faze, faze zahteva i faze odgovora.



Slika 2. Faze Split Transaction Open Core Protocol-a.

Master šalje komandu na interfejs (*MCmd*). Kada se podigne signal *MTagInOrder* master tada ukazuje *slave*-u da taj transfer ima prioritet za odgovor. *Slave* u zavisnosti od toga koju je komandu dobio odgovara na interfejs. Ukoliko *MTagInOrder* nije na logičkoj jedinici, onda oznaka *MTagID* se uvećava. Nakon izvršavanja prioritetnih transfera, *slave* ima slobodu odgovora na označene vrednosti. Signal *SCmdAccept* je prihvatni signal, odnosno signal koji ukazuje da je *slave* prihvatio zahtev.

A. Signali OCP-a

Tabela II prikazuje osnovne signale OCP protokola.

TABELA II
OSNOVNI SIGNALI OCP

Clk	1 bit	Varies	Clock input
MAddr	Configure	Master	Transfer address
MCmd	[3:0]	Master	Transfer command
MData	Configure	Master	Write data
SCmdAccept	1 bit	Slave	Slave accept transfer
SData	Configure	Slave	Read data
SResp	[1:0]	Slave	Transfer response

B. Odgovor slave-a

Tabela III prikazuje varijacije odgovora *slave*-a.

TABELA III
SLAVE ODGOVOR

SRESSP[1:0]	RESPONSE	
00	NO RESPONSE	NULL
01	DATA VALID/ ACCEPT	DVA
10	REQUEST FAILED	FAIL
11	RESPONSE ERROR	ERR

IV. OCP SPLIT TRANSACITON UVM OKRUŽENJE

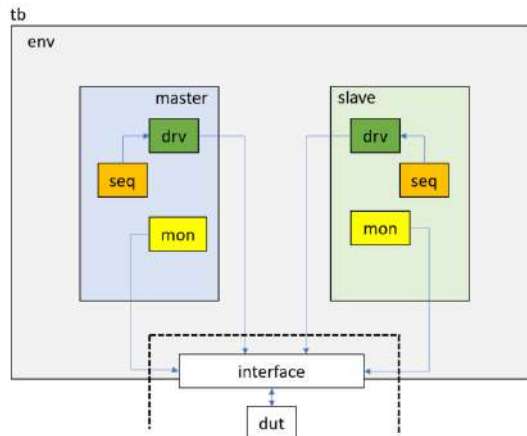
Kreiranje verifikacionog okruženja za *Split Transaction OCP* počinje generisanjem *template*-a u alatu. Nakon toga bitno je izbaciti proste signale na interfejsu sa ciljem provere da li generisan *template* u funkcionalnosti. Pre same verifikacije moramo definisati verifikacioni plan. On se sastoji iz niza podataka funkcionalne specifikacije koje treba proveriti i pokriti sve moguće varijacije. Obizrom da se radi o Split Transaction konfiguraciji bitno je proveriti ispravnost slanja komande, njihovo unakrsno presecanje, kao i vreme *slave* odgovora koje protekne. Bitno je definisati tvrdnje koje će proveraviti ispravnost nekih ekspresija. Npr. ukoliko se na interfejsu pojavi komanda IDLE to znači da se u tom trenutku ne dešava ništa, a ako se ipak pojavi neki signal za vreme ove komande *assertion* će se izvršiti i ukazaće na grešku. U tabeli IV prikazane su stavke iz funkcionalne verifikacije koje treba pokriti. Pokrivenost ili *coverage* je mera provere ispravnosti dizajna. Prilikom puštanja regresije (*skup testova*), polja regresije ukoliko su ispravna se uvećavaju i pokazuju nam stepen funkcionalne ispravnosti. Osim toga važno je da i pokrivenost koda bude na maksimalnih 100%.

TABELA IV
PROVERA FUNKCIONALNOSTI

Cg MCmd i cross Write, Read IDLE	3 bins IW, IR, RW, RI, WI, WR, WW, RR,
Cg master delay	bins
Cg slave delay	bins
Cg slave response delay	bins
Asserts	STagInOrder, MTagInOrder, MCmd, ValidData-SResp
Code coverage 100 %	Definisan na 100 procenata.

A. UVM okruženje

Na slici 3 prikazano je UVM okruženje za *Split Transaction Open Core Protocol*.

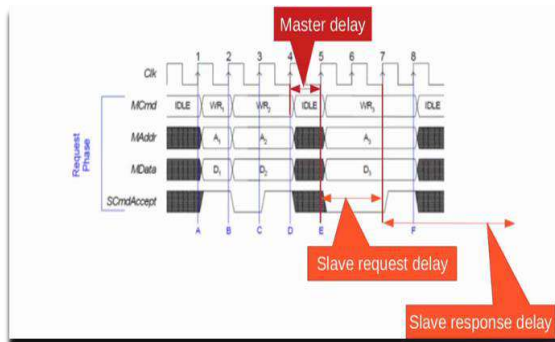


Slika 3. Okruženje za Split Transaction OCP.

UVM okruženje za konfiguraciju Split Transaction sastoji se od *testbench*-a, *environment*-a i dva agenta. Obzirom da se radi o aktivnim agentima zato što je OCP bidirekcion protokol, sadržaj agenta je isti, međutim njihova logika se razlikuje. *Master* šalje zahteve a *slave* odgovara. *Driver* je aktivna komponenta koja pokreće i oponaša signale OCP-a. *Monitor* je pasivna komponenta koja kupi sve promene na interfejsu. Kada se jedan transfer završi odnosno obe faze izvrše, driver ispisuje *done*. Nakon toga sledi novi transfer. U *environment*-u se može naći konfiguracioni objekat koji određuje aktivnost agenta. Ukoliko se radi o pasivnom agentu, kreira se samo monitor.

B. Kašnjenje u fazama zahteva i odgovora

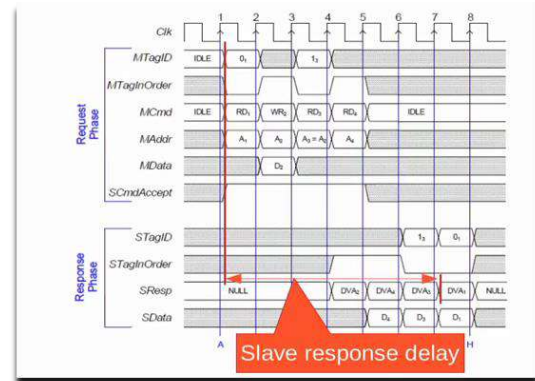
Na slici 4 prikazana je faza zahteva.



Slika 4. Faza zahteva OCP Split Transaction.

Tokom faze zahteva uočavamo moguća kašnjenja. *Master delay* se može okarakterisati kao vreme koje protekne tokom izvršavanja komande IDLE. Ako pogledamo *slave request delay* na slici, možemo zaključiti

da je to vreme koje protekne sve dok *slave* ne podigne *SCmdAccept* signal na logičku jedinicu.



Slika 5. Faza odgovora OCP Split Transaction.

Na slici 5. *waveform* pokazuje *slave response delay*. *Slave response delay* je vreme koje protekne od trenutka kada je *master* na interfejsu poslao komandu do trenutka kada *slave* odgovara na tu komandu. Obzirom da se radi o randomizovanim signalima, kašnjenja su ograničena u klasi transakcije.

V. DRIVER I MONITOR

U radu će biti predstavljene najbitnije komponente i biće ilustrovan njihov rad. Najviše će biti reči o drajverima i monitoru. Na slici 6 prikazan je kod koji omogućava funkcionalnost *master driver*-a OCP Split Transaction. *Driver* dobija randomizovanu komandu *MCmd* i randomizovan *MTagInOrder*. Ukoliko je *MCmd* jedinica onda je komanda za upis, ukoliko je broj 2 onda je to komanda za čitanje. Ako se radi o komandi za upis, master šalje randomizovanu adresu i podatak, a ukoliko se radi o čitanju šalje samo adresu. Nakon izvršenja komande sačekamo da *slave* podigne *SCmdAccept*. *Master* i *slave driver* rade u *forever* petlji, odnosno sve dok *sequencer* ne prestane sa slanjem transakcije ove dve komponente će raditi.

Slika 7 pokazuje rad *slave driver*-a. U okviru *slave driver*a kreiramo lokalne promenljive koje će skupljati podatke sa interfejsa. U ovom slučaju dve promenljive tipa *queue*, čuvaju sve komande koje su poslate sa *MTagInOrder*-om, i komande sa oznakom. Nakon toga proverom onoga što se nalazi u tim lokalnim promenljivama, *slave* odgovara. Ukoliko je *STagInOrder* na jedinici onda će *slave* odgovoriti sa podatkom iz *order_queue*-a u kome se nalaze komande sa podignutim *MTagInOrder*-om. Ukoliko se radi o drugom slučaju onda će biti aktivan *non_order_que*. U slučaju da je komanda za čitanje *slave* odgovora na *SResp* signal sa DVA i podatkom *SData*. U slučaju komande upisa od strane *master*-a onda *slave* odgovara samo sa DVA.

```

task my_uvc_master_driver:drive_transaction(my_uvc_transaction transaction);
  vif.MCmd <= transaction.MCmd;
  vif.MTagInOrder <= transaction.MTagInOrder;

  if (transaction.MTagInOrder == 0) begin
    MCount++;
    vif.MTagID <= MCount;
  end
  if (transaction.MCmd == 1) begin //write
    vif.MData <= transaction.MData;
    vif.MAddr <= transaction.MAddr;
    wait (vif.SCmdAccept == 1);
  end
  else if (transaction.MCmd == 2) begin //read
    vif.MAddr <= transaction.MAddr; //read mcmd
    vif.MData <= 'bz';
    wait (vif.SCmdAccept == 1);
  end
  else // idle
    vif.MAddr <= 'bz';
    vif.MData <= 'bz';
  end
  num_sent++;
  $um_info(get_type_name(), $formatf("Item %0d Sent: ...", num_sent), UVM_H3GH);
endtask : drive_transaction

```

Slika 6. Master driver.

```

task my_uvc_slave_driver:send_response(my_uvc_transaction response);
  vif.SRespDelay <= response.SRespDelay;
  repeat(response.SRespDelay)
    @posedge vif.sig_clock;
  grancica = non_order_queue.size();
  id = $random_range(0,grancica-1);

  if (order_queue.size() == 0 && non_order_queue.size() == 0) begin
    vif.SResp <= HALL;
    vif.StagInOrder <= 'bz';
    $um_info(get_type_name(), $formatf(".....SIZE.....ORDER & NON ORDER: %0d %0d", order_queue.size(), non_order_queue.size()), UVM_H3GH);
  end
  if (response.StagInOrder == 1 && order_queue.size() == 0 && non_order_queue.size() != 0) begin
    vif.StagInOrder <= 1;
  end
  if (response.StagInOrder == 0 && order_queue.size() != 0 && non_order_queue.size() == 0) begin
    vif.StagInOrder <= 1;
  end
  if (vif.StagInOrder == 1 && order_queue.size() != 0) begin
    if (order_queue.pop_front() == 1) begin
      vif.SData <= response.SData;
      vif.SResp <= OVA;
    end
    else begin
      vif.SData <= 'bz';
    end
  end
  else if (vif.StagInOrder == 0 && non_order_queue.size() != 0) begin
    if (non_order_queue[id] == 1) begin
      vif.StagID <= tag_id[id];
      vif.SData <= response.SData;
      vif.SResp <= OVA;
    end
    else begin
      vif.StagID <= tag_id[id];
      vif.SData <= 'bz';
      vif.SResp <= HALL;
    end
    non_order_queue.delete[id];
    tag_id.delete[id];
  end
  num_sent++;
endtask : send_response

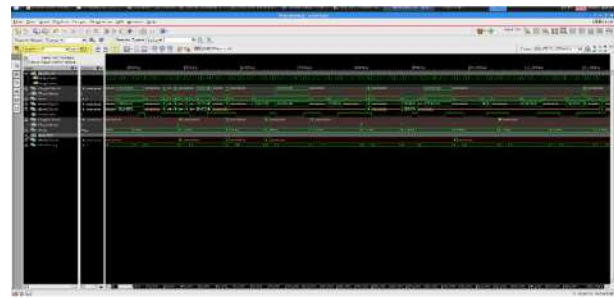
```

Slika 7. Slave driver.

Monitor u ovom radu skuplja sve promene komande *MCmd* i kašnjenja *MDelay*, *slave reponse delay*, *slave request delay*. Osim toga *monitor* će skupljati i sve adrese i podatke poslate od strane i *master*-a i *slave*-a. Za verifikaciju *Split Transaction* konfiguracije koristimo *covergroup*-e sa za *MCmd* i za kašnjenja.

VI. SIMVISION I VMANAGER

Rezultati simulacije prikazni su u sledećim slikama. Na prvoj slici koja sledi za nama nalazi se simulacija jedne randomizovane sekvence sa fazama zahteva i odgovora. Na slici 9 nalazi se stepen pokrivenosti *Open Core Protocol*-a po *Split Transaction* konfiguraciji. Cilj od 100 % pokrivenosti koda i funkcionalne pokrivenosti pokazatelj je da je uspešno završen proces verifikacije. Rezultati su prikazani u *Vmanager*.



Slika 8. Signali u Simvision alatu.



Slika 9. Stepen pokrivenosti u Vmanager alatu.

VII. ZAKLJUČAK

Na osnovu obrazloženog rada, možemo zaključiti da je proces verifikacije dizajna, koji samo jedan od procesa proizvodnje čipa, veoma složen i kompleksan. U radu prikazan je process verifikacije *Open Core Protocol Split Transaction* protokola. Sam process verifikacije nije jednostavno odraditi upravo iz razloga što profesionalni alati nisu javno dostupni. Obizrom na današnju složenost dizajna, možemo pretpostaviti da će proces verifikacije u budućnosti zahtevati više vremena.

ZAHVALNICA

Zahvaljujem se kompaniji HDL Design House koja je omogućila korišćenje alata.

LITERATURA

- [1] Ray Selami „The UVM Primer: Introduction to the UVM methodology“, 2013
- [2] Accellera „Universal Verification Methodology, 2011
- [3] IEEE Standard for SystemVerilog-Unified HardwareDesign, Specification, and Verification Language IEEE Std 1800™, 2012
- [4] Open Core Protocol Specification www.accellera.org
- [5] Eduard Cerny „SVA: The Power of Assertions in SystemVerilog“, 2015
- [6] Stuart Sutherland, Tom Fitzpatrick „UVM Rapid Adoption: A Practical Subset of UVM“, 201
- [7] HDL Design House, Tools and Training Presentations, 2020.

Realizacija kola sa optokaplerima

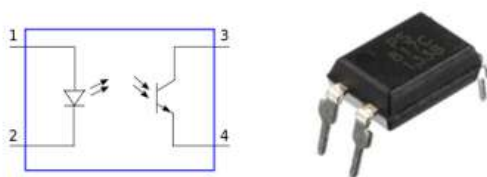
Mikica Radenković

Sadržaj – U ovom radu je prikazana primena i funkcionalnost optoelektronske komponente optokaplera u svrhu realizacije laboratorijskih vežbi.

načina: računskim putem, primenom simulatora i praktično realizovano u laboratoriji.

I. UVOD

Optokapler je komponenta koja se sastoji od LED diode i fototranzistora integrisanih u jednom kućištu. [1, 2]. Optokapler predstavlja komponentu sa svetlosnom spregom između ulaza i izlaza. Ulaz i izlaz su međusobno galvanski izolovani (slika 1.).



Slika 1. Električni simbol i izgled komponente.

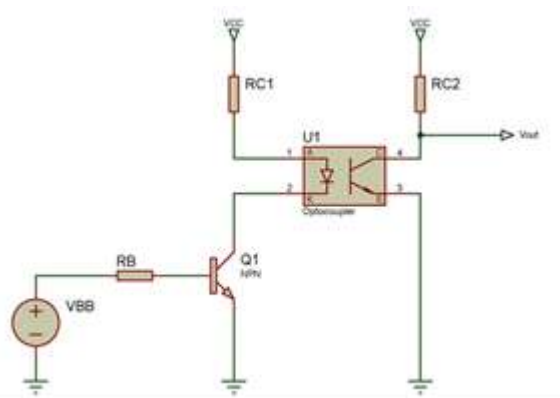
II. PRINCIP FUNKCIONISANJA OPTOKAPLERA

Signal na ulazu izaziva emisiju svetlosti LE diode. Svetlost predstavlja pobudu fototranzistora, tako da se na izlazu pojavljuje odgovarajući signal. Galvanska izolacija čini optokapler pogodnim za primenu u svim uređajima kod kojih korisnik može doći u dodir sa potencijalno opasnim nivoima signala. Optokapler se takođe može koristiti za prilagođenje naponskih nivoa između dva kola, kao i za smanjenje uticaja elektronskog šuma. Optokapler se može polarisati da radi u dva režima, u aktivnom režimu rada ili u zasićenju. Kada je izlazni tranzistor u zasićenju, optokapler predstavlja prekidač sa svetlosnom spregom. Na ulaz optokaplera se dovodi digitalni signal, koji može predstavljati izlaz nekog senzora. Na izlazu optokaplera se kao opterećenje najčešće pojavljuje neko logičko kolo. Treba primetiti da postoje dve nezavisne mase, ulaznog i izlaznog kola.

III. PRIMER LABORATORIJSKE VEŽBE

Na slici 2. je prikazano kolo koje služi za optičku spregu dva kola pomoću optokaplera [3]. Kolo čija električna šema prikazana na slici 3. rešavano je na više

M. Radenković je studentkinja master akademskih studija modula Elektronika i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mikica.radenkovic@elfak.rs.



Slika 2. Kolo za optičku spregu pomoću optokaplera.

Računskim putem se određuje radna tačka tranzistora Q1 ($V_{BE} = 0.7 \text{ V}$, $\beta = 100$) ako su poznati elementi u kolu: $V_{BB} = 5 \text{ V}$, $R_B = 43 \text{ k}\Omega$, $V_{LED} = 2 \text{ V}$, $R_{C1} = 680 \Omega$, $V_{CC} = 12 \text{ V}$. Potrebno je odrediti vrednost otpornosti otpornika R_{C2} tako da napon na izlazu bude $V_{OUT} = 5 \text{ V}$ ako je struja kolektora fototranzistora 20 puta manja od struje koja teče kroz LE diodu u optokapleru.

Za određivanje radne tačke tranzistora Q1, potrebno je odrediti kolektorsku struju I_{C1} , kao i napon između kolektora i emitora. Naponska jednačina (1) po drugom Kirhofovom zakonu za bazno kolo tranzistora Q1 ima oblik

$$V_{BB} = R_B I_B + V_{BE} \quad (1)$$

iz koje se za baznu struju dobija jednačina (2)

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} = 0.1 \text{ mA.} \quad (2)$$

Kako tranzistor radi u normalnoj aktivnoj oblasti, tada važi za kolektorsku struju jednačina (3)

$$I_{C1} = \beta \cdot I_B = 10 \text{ mA.} \quad (3)$$

Napon između kolektora i emitora ovog tranzistora izračunava se iz naponske jednačine (4) po drugom Kirhofovom zakonu

$$V_{CC} = R_{C1} I_{C1} + V_{LED} + V_{CE}, \quad (4)$$

odakle se dobija jednačina (5)

$$V_{CE} = V_{CC} - V_{LED} - R_{C1}I_{C1} = 3.2 \text{ V}, \quad (5)$$

tako da je radna tačka tranzistora Q1 (V_{CE} , I_{C1}) = (3.2 V, 0.01 A).

Kolektorska struja izlaznog kola optokaplera izračunava se iz jednačine (6)

$$I_{C2} = \frac{I_{LED}}{20} = \frac{I_{C1}}{20} = 0.5 \text{ mA}. \quad (6)$$

Iz naponske jednačine (7) po drugom Kirhofovom zakonu kolektorskog kola optokaplera

$$V_{CC} = R_{C2}I_{C2} + V_{CE}, \quad (7)$$

i kako je $V_{OUT} = V_{CE} = 5 \text{ V}$ dobija se da je otpornost otpornika na osnovu jednačine (8)

$$R_{C2} = \frac{V_{CC} - V_{OUT}}{I_{C2}} = 14 \text{ k}\Omega. \quad (8)$$

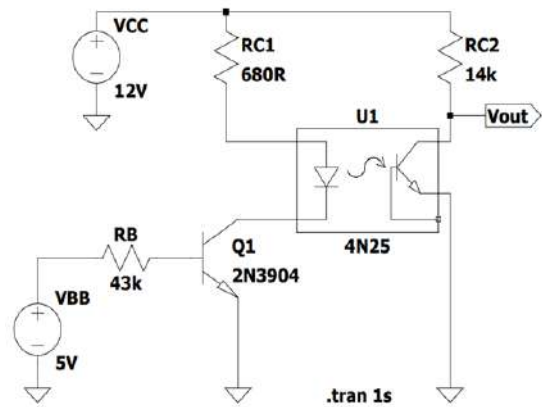
Nakon dobijanja vrednosti parametara kola koji su potrebni za proveravanje istih u simulatoru i laboratoriji, vrši se izbor najbliže vrednosti otpornosti otpornika R_{C1} , R_{C2} i R_B iz standardnog niza nazivnih vrednosti otpornosti. Potom je potrebno izračunati vrednost napona na izlazu kola V_{OUT} i odrediti radnu tačku tranzistora Q1. Dobijene vrednosti struja I_{C1} , I_{C2} i napona V_{OUT} proveriti u simulatoru. Uzeti u obzir da je pad napona na diodi $V_{LED} = 1.1 \text{ V}$ i da su struje $I_{C1} = 15.8 \text{ mA}$ i $I_{C2} = 0.79 \text{ mA}$ očitane u simulatoru.

IV. PRAKTIČNA REALIZACIJA

Laboratorijska oprema korišćena prilikom rada u laboratoriji:

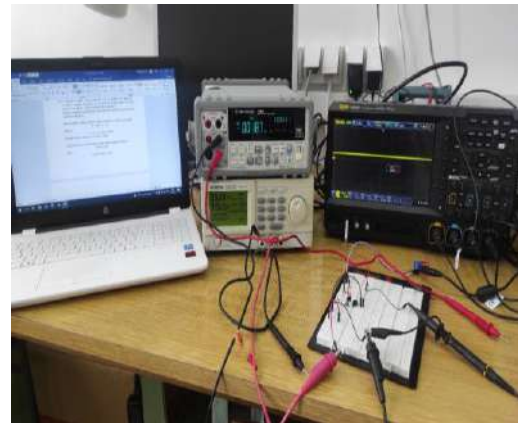
- PSP-2010 DC izvor napajanja [4]
- U3606A Digitalni multimetar [5]
- MSO5104 Digitalni osciloskop [6]
- Optokapler 2N25 [1-2]
- Otpornici [1]
- Tranzistor 2N3904 [7]
- Konektori za povezivanje.

LTSpice [8] je simulator koji se koristi za simulaciju kola i proveru vrednosti u samom kolu. U sledećim primerima se koristi da bi se uporedili rezultati dobijeni u simulatoru i rezultati dobijeni merenjem u laboratoriji. Na slici 3. je prikazana električna šema u simulatoru LTSpice u kome je vršena simulacija kola i provera vrednosti parametara.



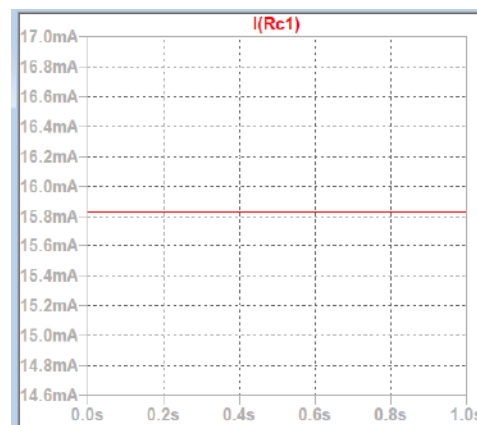
Slika 3. Kolo u simulatoru.

Na slici 4. je prikazano radno okruženje u laboratorijskim uslovima i praktična realizacija kola za optičku spregu pomoću optokaplera.



Slika 4. Radno okruženje i praktična realizacija kola za optičku spregu pomoću optokaplera.

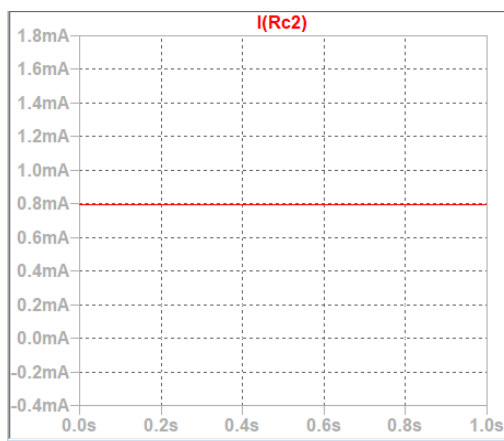
Struja kolektora I_{C1} kroz tranzistor Q1 određena u simulatoru prikazana je na slici 5, dok je rezultat dobijen merenjem u laboratoriji prikazan na slici 6.



Slika 5. Struja kolektora I_{C1} određena u simulatoru.

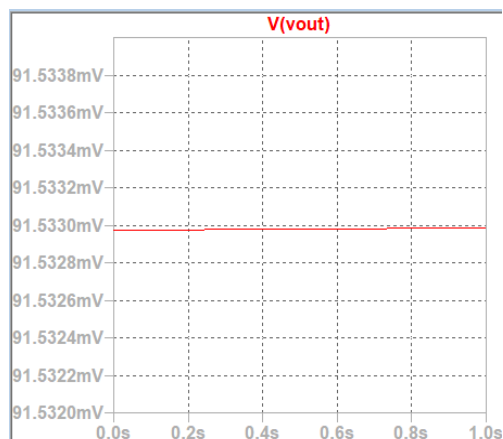
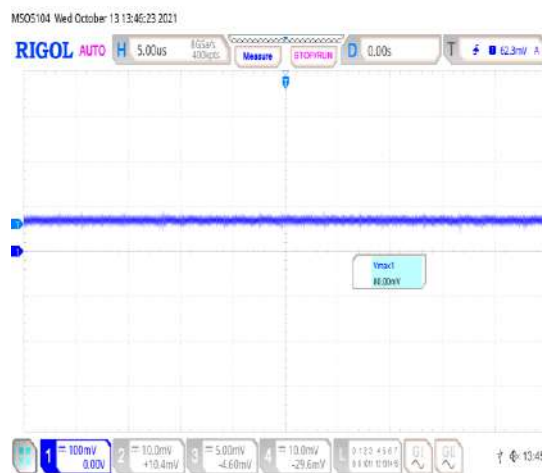
Slika 6. Struja kolektora I_{C1} dobijena merenjem.

Kroz tranzistor optokaplera protiče struja kolektora I_{C2} , na slici 7. prikazan je rezultat dobijen u simulaciji, dok je na slici 8. prikazan rezultat dobijen merenjem u laboratoriji.

Slika 7. Struja kolektora I_{C2} određena u simulatoru.Slika 8. Struja kolektora I_{C2} dobijena merenjem.

Izlazni napon V_{OUT} kola se očitava na izlazu iz optokaplera kao što je i prikazano na električnoj šemi. Na

slici 9 prikazana je vrednost izlaznog napona V_{OUT} merena u simulatoru, dok je vrednost izmerena u laboratoriji prikazana na slici 10.

Slika 9. Vrednost izlaznog napona V_{OUT} dobijena u simulatoru.Slika 10. Vrednost izlaznog napona V_{OUT} merena u laboratoriji.

Rezultati simulacije dobijeni uz pomoć simulatora LTSpice i rezultati dobijeni praktičnim merenjem u laboratoriji se razlikuju. Razlika potiče iz činjenice da simulator radi sa komponentama koje su idealnih karakteristika, odnosno nema nikakvih odstupanja od vrednosti koja je zadata, dok u praksi to nije slučaj. Prilikom povezivanja kola koriste se komponente koje nisu idealne, imaju svoja odstupanja i pri tome dolazi do stvaranja grešaka merenja. Klasični predstavnik je otpornik koji ima određenu toleranciju i automatski stvara grešku prilikom merenja, ukoliko se oni vežu redno greška se povećava, jer i njihova tolerancija raste.

V. ZAKLJUČAK

Optoelektronske komponente čine širok opseg komponenata, počevši od običnih fotodioda do najnovijih dispeja koji se mogu naći u upotrebi. Svaka od

optoelektronskih komponenata odlikuje se sebi svojstvenim karakteristikama, a te karakteristike su od značaja za edukaciju zbog raznovrsnosti primene.

Ispitivanjem optoelektronskih komponenata u laboratorijskim uslovima stiče se znanje u vidu sklapanja električnih šema i korišćenje opreme za ispitivanje samih karakteristika komponenata, merenja određenih veličina u električnom kolu. Ključnu stvar predstavlja činjenica da su optoelektronske komponente kompatibilne sa velikim brojem razvojnih okruženja, koja olakšavaju njihovu primenu i pružaju mogućnost sticanja novih veština u pogledu softverskih okruženja.

ZAHVALNICA

Autor se zahvaljuje nastavnicima i saradnicima Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu na pruženoj pomoći i saradnji tokom realizacije ovog projekta.

LITERATURA

- [1] „Uvod u poluprovodničke komponente i njihovu primenu”, Aneta Prijić, Danijel Danković i Zoran Prijić, drugo,

izmenjeno i dopunjeno elektronsko izdanje, Univerzitet u Nišu, Elektronski fakultet, Niš, 2020.

- [2] „Laboratorijski praktikum sa zadacima za samostalan rad iz predmeta Komponente za telekomunikacije”, Danijel Danković, Edicija: Pomoćni udžbenici, Univerzitet u Nišu, Elektronski fakultet, Niš, 2018.
- [3] „Praktikum laboratorijskih vežbi sa primerima zadataka iz predmeta osnovi optike”, Emilija Živanović, Edicija: Pomoćni udžbenici, Univerzitet u Nišu, Elektronski fakultet, Niš, 2018.
- [4] Tehnička dokumentacija PSP-2010 DC izvor napajanja: <https://www.gwinstek.com/en-GB/products/detail/PSP-Series>
- [5] Tehnička dokumentacija U3606A Digitalni milimetar: <https://www.keysight.com/zz/en/assets/7018-02447/brochures/5990-5315.pdf>
- [6] Tehnička dokumentacija MSO5104 Digitalni osciloskop: https://beyondmeasure.rigoltech.com/acton/attachment/1579/f-0907/1/-/-/-/-/MSO5000_datasheet.pdf
- [7] Tehnička dokumentacija tranzistor 2N3904: <https://www.onsemi.com/pdf/datasheet/2n3903-d.pdf>
- [8] Uputstvo za korišćenje LTSpice programa: <http://mikro.elfak.ni.ac.rs/predmeti/elektronske-komponente/>

Digitalna platforma “Virtual MIEL”

Milan Danković

Sadržaj – U ovom radu detaljno je opisan razvoj digitalne platforme “Virtual MIEL”. Platforma je razvijena za potrebe održavanja 32. međunarodne konferencije o mikroelektronici, MIEL 2021. Razvijenu platformu, u istom ili modifikovanom obliku, moguće je primeniti i za druge naučne skupove, uključujući i IEEEESTEC konferenciju.

I. UVOD

U prethodne dve godine uočeno je da pandemija nove bolesti Kovid-19, izazvane korona virusom, ne predstavlja samo globalnu zdravstvenu krizu, već dalekosežno utiče i na privredni i društveni razvoj, uzrokujući višestruke posledice po živote ljudi. Iz tih razloga došlo je do znatnih izmena u decenijski ustaljenim režimima rada i ponašanja ljudi širom zemaljske kugle. Ove nagle, spontane, unapred nepripremljene promene u režimima rada pogodile su sve starosne grupe, pa tako nisu ni mladi pošteđeni. Iako pandemija još traje, brojna istraživanja već sada ukazuju da je život mladih u Srbiji značajno promenjen u prethodne dve godine [1], a da će efekti tih promena biti vidljivi i po završetku pandemije.

IEEEESTEC konferencija [2], koja je tokom prethodne decenije izrasla u jedan od najznačajnijih skupova namenjen učenicima srednjih škola i studentima tehničkih fakulteta, takođe je zbog pandemije bolesti Kovid-19 morala da menja svoj ustaljeni format. Tradicionalno okupljanje (u sklopu proslave Dana Elektronskog fakulteta u Nišu, ilustrovano na slici 1) više stotina učenika i studenata tokom održavanja IEEEESTEC konferencije zamenjeno je virtuelnim događajem.



Slika 1. Atmosfera sa svečanog otvaranja jubilarne 10. IEEEESTEC konferencije [2].

Organizatori IEEEESTEC konferencije upotreбили su nekoliko ustaljenih rečenica, stalno korišćenih tokom

M. Danković je student Softverskog inženjerstva na Elektrotehničkom fakultetu, Univerziteta u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija, E-mail: dankovicmilan@gmail.com

pandemije bolesti Kovid-19, i konferenciju premestili u virtuelni svet:

“Ove godine IEEEESTEC konferencija susrela se, kao i ceo svet, sa pandemijom neviđenih razmera. Ovaj izazov stavio je na test celo društvo, ali nam i pokazao da je nauka potrebna više nego ikad. Zato se selimo u virtuelni svet, nastavljamo tradiciju dugu 13 godina i ovogodišnju konferenciju organizujemo online” [2].

Slična situacija je i sa drugim domaćim i međunarodnim naučnim skupovima, pa je iz tih razloga krenula potražnja za digitalnim platformama koje omogućavaju održavanje naučnih skupova i konferencija u virtuelnom svetu. Treba napomenuti da su i pre pojave pandemije bolesti Kovid-19 mnoge kompanije posvetile pažnju i pokazale interesovanje za ovakvim platformama. Jedna od takvih aplikacija je *MatchAbout* [3], koja omogućava da se čitav događaj prenesu u digitalni svet u realnom vremenu, ali i da učesnici isprate sve vezano za sam događaj (i pre početka događaja, a i po završetku).

U ovom radu opisan je razvoj digitalne platforme “Virtual MIEL” namenjene organizaciji 32. Međunarodne naučne konferencije o mikroelektronici, MIEL 2021 [4].

II. OPIS ORGANIZACIJE NAUČNE KONFERENCIJE

Iskustvo organizatora konferencija ukazuje da se tok organizacije konferencija uglavnom odvija kroz tri faze:

I faza: komunikacija sa potencijalnim autorima (slanje poziva, davanje neophodnih informacija vezanih za konferenciju, podsećanja...). Aktivnosti u ovoj fazi uglavnom se odvijaju putem elektronske pošte, objavama na promo stranicama stručnih magazina ili časopisa iz te naučne oblasti, objavama preko sajta konferencije ili sajtova organizatora konferencije...

II faza: odvija se kroz aktivnosti od prijema početne verzije rada, pa sve do prijema finalne verzije rada koja se objavljuje u zborniku radova sa konferencije. Nakon prijema početne verzije rada, organizatori konferencije određuju relevantne recenzente iz oblasti iz koje je rad, koji procenjuju naučni doprinos rada. Na osnovu prispelih recenzija organizatori donose odluku o prihvatanju rada (ili odbijanju) i zahtevaju od autora da dostave finalnu verziju urađenu u skladu sa zahtevima recenzenata. Organizacioni odbor konferencije analizira finalnu verziju rada i donosi konačnu odluku o njegovom objavljivanju u zborniku radova sa konferencije. Aktivnosti u ovoj fazi odvijaju se putem elektronske pošte i/ili besplatnih platformi, kao što su *Open Journal System* [5], *EasyChair* [6] i slične.

III faza: prezentovanje radova tokom održavanja konferencije. Uglavnom se odvija preko različitih

platformi, kao što su *MS Teams* [7], *Zoom* [8], *Google Meet* [9] i slične.

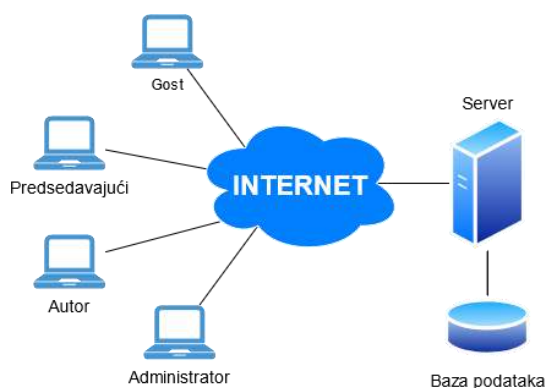
Bez obzira što mnoge od navedenih platformi omogućavaju uspešnu realizaciju više faza u praksi i organizatori najvećih naučnih konferencija u svetu koriste napred pomenuti pristup. Sa ciljem grupisanja aktivnosti na jednom mestu, osnovna ideja je razvijanje sistema kome se pristupa preko sajta konferencije, a kojim može da se pokrije veći broj faza tokom organizacije konferencije. Ono što je najbitnije, sve se odvija na jednom mestu, preko sajta konferencije, gde potencijalni autori inače i traže informacije o konferenciji.

IEEEESTEC konferencija ima razvijenu kvalitetnu platformu koja pokriva I i II fazu organizacije jedne naučne konferencije (kartica *Project application* na početnoj strani sajta konferencije). Ova platforma je u funkciji već nekoliko godina, a pozitivno iskustvo pri korišćenju imaju kako organizatori konferencije tako i autori i recenzenti. Osoblje Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu organizator je i konferencije IEEEESTEC i konferencije MIEL. Da bi upotpunili sistem organizovanja naučnih konferencija došli su na ideju da se razvije digitalna platforma koja bi omogućavala realizaciju III faze. Preduslov da se razvije i koristi takva platforma je pravljenje potpuno nove verzije sajta MIEL konferencije (<http://miel.elfak.ni.ac.rs/>), imajući u vidu da je stara verzija sajta (<http://miel.elfak.ni.ac.rs/old/>) kreirana pre 30 godina.

III. STRUKTURA I OPIS PLATFORME

A. Struktura platforme

Platforma je realizovana na bazi dinamičkog sajta postavljenog na veb serveru koji podržava *PHP* [10] i *AJAX* [11]. Arhitektura veb aplikacije prikazana je na slici 2. Postoji server na kome je *MySQL* [12] baza podataka u kojoj se čuvaju kredencijali korisnika i ostale potrebne informacije. Informacije o sadržaju veb aplikacije dobijaju se preko odgovarajućeg *API* veb servisa [13]. Server uz pomoć *PHP* upita i podataka iz baze podataka kreira statički *HTML* kod [14] koji se prosleđuje korisniku sajta.



Slika 2. Arhitektura aplikacije.

B. Kategorije korisnika

Platformom su obuhvaćene sledeće kategorije korisnika: *gost*, *autor*, *predsedavajući konferencije* i *administrator sistema*.

Gost: Gost je tip korisnika koji ima pristup opštim informacijama konferencije i mogućnost pregledanja video prezentacija objavljenih radova nakon što, na lični zahtev, dobije token od organizatora konferencije (predsedavajući konferencije i administrator).

Autor: Autor je tip korisnika koji se loguje u sistem, ima pristup istim funkcionalnostima kao i gost, ali može dodavati i svoje radove kao i prateće materijale (video snimak, poster i prezentaciju). Ove funkcionalnosti dostupne su autorima do određenih vremenskih trenutaka kada se prelazi u sledeću fazu pripreme konferencije. Autor ima mogućnost pregleda svojih podataka (profila) i njihovu izmenu. Autor nakon registrovanja na sistem čeka potvrdu svog naloga od strane organizatora konferencije.

Predsedavajući: Predsedavajući je tip korisnika koji se loguje u sistem i ima mogućnost kreiranja programa konferencije i odobravanja pristupa određenim delovima veb aplikacije ostalim korisnicima. Takođe, ima mogućnost dodavanja i brisanja radova i pratećih materijala bilo kog autora.

Administrator: Administrator je tip korisnika koji se loguje u sistem i ima skup privilegija koji predstavlja uniju skupova privilegija svih ostalih korisnika kao i mogućnost menjanja statusa korisnika (proglašavanje korisnika predsedavajućim).

C. Funkcionalni zahtevi platforme

Osnovne funkcionalnosti koje bi platforma trebalo da obezbedi različitim kategorijama korisnika MIEL konferencije:

Registracija korisnika. Prilikom registracije korisnika unose se *e-mail* adresa, lozinka i druge potrebne informacije. U slučaju da u sistemu već postoji nalog sa unetom *e-mail* adresom ili nisu ispravno uneti svi potrebni podaci, korisniku se prikazuje odgovarajuća poruka greške. Nakon uspešnog kreiranja korisničkog naloga, korisnik čeka odobrenje naloga od strane predsedavajućeg ili administratora.

Autorizacija korisnika. Prilikom prijavljivanja na sistem, korisnik unosi *e-mail* adresu i lozinku. U slučaju da kombinacija *e-mail* adrese i lozinke ne odgovara nijednom nalogu, korisniku se prikazuje odgovarajuća poruka greške.

Osnovne funkcionalnosti korisnika. Svi korisnici imaju mogućnost pregledanja opštih informacija o konferenciji, to jest pregled detalja o mestu i vremenu održavanja, pregled plana i programa konferencije. Korisnicima je omogućen i pregled galerije prethodnih konferencija kao i kratkog izveštaja o njima. Svaka grupa korisnika ima dostupne sve funkcionalnosti korisnika niže u hijerarhiji kao i navedene dodatne funkcionalnosti.

Funkcionalnosti autora.

- Pregled i izmena profila. Svaki autor može videti i modifikovati svoj profil, na kome su prikazane osnovne informacije. Informacije o autoru su ime i prezime, pol, e-mail, institucija i slika.

- Postavljanje sadržaja. Autor ima mogućnost objavljivanja naučnog rada i pratećih materijala.

- Pregled i ocenjivanje i komentarisane radova ostalih autora. Svaki autor nakon odobravanja i objavljivanja radova može čitati i oceniti radove ostalih autora. Autor može postaviti pitanje autoru rada koji čita u vidu komentara.

Funkcionalnosti predsedavajućeg.

- Kreiranje programa konferencije. Predsedavajući može da kreira program konferencije koji će biti objavljen među opštim informacijama o konferenciji.

- Odobravanje/odbijanje naloga i akcija gosta i autora. Predsedavajući može da odobri ili odbije zahtev za kreiranje naloga ili pristup zaštićenom delu veb aplikacije ostalim korisnicima.

Funkcionalnosti administratora.

Administrator ima privilegije za pristup i izmenu svih delova veb aplikacije. Administrator može korisniku dodeliti status predsedavajućeg.

D. Vizuelni izgled digitalne platforme

Nakon prijavljivanja na sistem administratoru i predsedavajućem se otvaraju kartice kao što je ilustrovano na slici 3. Preko kartice UPLOAD INFO administrator ili predsedavajući pravi kompletnu strukturu za pregled radova. Autoru se nakon prijavljivanja otvaraju samo kartica UPLOAD FILES FOR YOUR PAPER (preko koje može da postavi video, prezentaciju, pdf ili word dokument) i kartica BROWSE PAPERS FROM MIEL (preko koje vrši pregled materijala svih radova). Gostu se otvara samo kartica BROWSE PAPERS FROM MIEL.

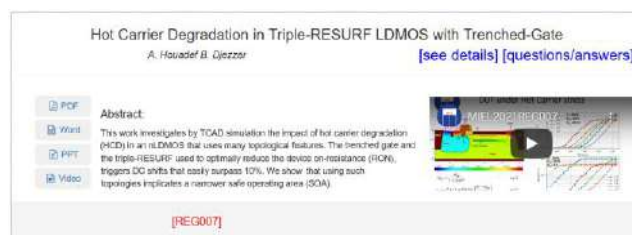


Slika 3. Izgled kartica nakon prijavljivanja administratora ili predsedavajućeg konferencije.

Nakon što je administrator/predsedavajući kreirao strukturu radova (sadržaj) i autori uneli video materijal (i opcionalno pdf, word i ppt.) za svaki rad se pojavljuje prikaz kao na slici 4. Korisnik može da pregleda film ili materijal koji je postavljen. Video materijal koji korisnik može da pregleda može da se podigne sa servera gde je postavljena platforma ili sa youtube kanala (kao što je slučaj na slici 4). Administrator ima mogućnost selekcije izvora video materijala, iz baze podataka ili sa youtube kanala.

Osnovni razlog razvoja ovakvog tipa digitalne platforme podstaknut je iskustvom organizatora MIEL

konferencije u periodu pandemije bolesti Kovid-19. Naime, na brojnim konferencijama javljala su se dva ključna problema, različite vremenske zone autora i tehnički problemi prezentera ili slušaoca (internet, konekcija, protok). Oba ključna problema imala su uticaj gotovo na sve održane konferencije u prethodnom periodu. Kako bismo prevazišli ove problem, ova platforma omogućava autorima postavljanje materijala pre konferencije, a posetiocima da taj materijal mogu da pregledaju tokom trajanja konferencije (više dana). Međutim, osnovni uslov svih konferencija koje se odvijaju pod pokroviteljstvom IEEE-a je da se učesnicima konferencije omogućiti postavljanje pitanja i dobijanje odgovora. Pri čemu IEEE zbog različitih vremenskih zona čak i preporučuje da ova komunikacija može da bude u pisanom obliku i može da se odvija više dana.



Slika 4. Prikaz rada preko kartice BROWSE PAPERS FROM MIEL 2021.

Pristup odeljku za komunikaciju, odnosno postavljanje pitanja i davanje odgovora omogućen je preko linka [question/answers], kao što je ilustrovano na slici 4. Na slici 5. prikazan je izgled odeljka za komunikaciju.

Hot Carrier Degradation in Triple-RESURF LDMOS with Trenched-Gate



Slika 5. Prikaz odeljka za postavljanje pitanja i odgovora.

IV. EVALUACIJA DIGITALNE PLATFORME “VIRTUAL MIEL”

Po završetku konferencije MIEL 2021 autori radova popunili su anonimnu anketu koja je sadržala 12 pitanja, od kojih su 3 pitanja otvorenog tipa. Pitanja su bila tematski podeljena i ticala su se: kvaliteta radova, angažovanja tima

koji je organizovao MIEL konferenciju i značaja digitalne platforme “Virtual MIEL”. Dva pitanja iz te ankete direktno su se ticala digitalne platforme “Virtual MIEL”, a rezultati su prikazani na slikama 6 i 7.



Slika 6. Rezultati evaluacije od strane autora. Odgovor na pitanje MIEL konferencija realizovana je korišćenjem digitalne platforme “Virtual MIEL”. Kolika je verovatnoća da ovu platformu preporučite vašim kolegama (1-nikakva, ..., 10-izuzetno velika).



Slika 7. Rezultati evaluacije od strane autora. Odgovor na pitanje Sveobuhvatno, kako bi ste ocenili digitalnu platform “Virtual MIEL” (izvanredna, veoma dobra, dobra, slab utisak, nezadovoljavajuća).

Rezultati anketa prikazani na slikama 6. i 7. jasno ukazuju da je više od 90% autora zadovoljno korišćenom digitalnom platformom i da bi je preporučili i za druge naučne konferencije i skupove. Ovakve komentare autori su pisali i kod pitanja otvorenog tipa. Neki su preporučivali da u slučaju kada se konferencije održavaju uživo poželjno je koristiti ovakvu platformu za autore koji iz nekog razloga nisu u mogućnosti da prisustvuju konferenciji.

Među odgovorima koje su napisani kod pitanja otvorenog tipa posebno se ističu sledeći komentari:

“From my experience over the last 18 months, your setup was one of the best I have seen, congratulations. The quality and reliability of the internet setup, the guidelines for the contributors, the style of the communication were excellent indeed”

“Digital platform was organized not forcing audience to follow a given schedule”

V. ZAKLJUČAK

32. međunarodna konferencija o mikroelektronici održana je korišćenjem digitalne platforme “Virtual MIEL”. Organizatori konferencije i autori radova veoma su zadovoljni načinom na koji je izvršeno prezentovanje radova i međusobna komunikacija. Digitalna platforma je namenski razvijena za potrebe ove konferencije ali može biti modifikovana i prilagođena drugim naučnim skupovima.

U anketi koja je sprovedena za potrebe ovog rada mnogi autori su istakli da način kada se konferencije održavaju uživo nije moguće zameniti niti jednom platformom. Međutim, ističu da je potreba za ovakvom platformom velika, čak i u slučaju kada se konferencije održavaju uživo, a bila bi namenjena autorima koji u tom trenutku imaju neke druge obaveze i nisu u mogućnosti da prisustvuju. Ovo jasno ukazuje da sa razvojem ove platforme treba nastaviti u budućem periodu.

ZAHVALNICA

Inspiracija za ovaj rad potiče sa predmeta “Principi softverskog inženjerstva” koji se sluša na smeru SI na Elektrotehničkom fakultetu Univerziteta u Beogradu. Ključna znanja vezana za ovaj rad stečena su iz ovog predmeta, a višegodišnje učestvovanje na konferencijama i stečeno iskustvo doprinelo je kompletiranju zadatka.

LITERATURA

- [1] M. Stošić i S. Đorđević, “Život mladih u Srbiji: uticaj kovid-19 pandemije: rad, obrazovanje, bezbednost, mentalno zdravlje, poverenje u institucije, volontiranje, migracije”, Dosije studio, Beograd, 2020. Dostupno na: <https://koms.rs/wp-content/uploads/2020/12/Zivot-mladih-u-Srbiji-uticaj-kovid-19-pandemije.pdf>
- [2] IEEEESTEC konferencija studentskih projekata, Dostupno na: http://ieee.elfak.ni.ac.rs/sr/about_ieeestec_sr/
- [3] Aplikacija MatchAbout. Dostupno na: <https://www.matchabout.com/#/welcome>
- [4] IEEE International Conference on Microelectronics, MIEL, Dostupno na: <http://miel.elfak.ni.ac.rs/>
- [5] Open Journal System. Dostupno na: <https://openjournalsystems.com/>
- [6] EasyChair. Dostupno na: <https://easychair.org/>
- [7] MS Teams. Dostupno na: <https://teams.microsoft.com/>
- [8] Zoom. Dostupno na: <https://zoom.us/>
- [9] Google Meet. Dostupno na: <https://meet.google.com/>
- [10] PHP. Dostupno na: <https://www.php.net/manual/en/intro-whatis.php>
- [11] AJAX. Dostupno na: https://www.w3schools.com/whatis/whatis_ajax.asp
- [12] MySQL. Dostupno na: <https://www.mysql.com/>
- [13] Web API. Dostupno na: https://en.wikipedia.org/wiki/Web_API
- [14] HTML. Dostupno na: <https://html.com>

Proračun optimalnih tokova aktivnih snaga primenom linearnog programiranja

Teodora Denić

Sadržaj - U ovom radu analizirana je primena linearnog programiranja za proračun optimalnih tokova aktivnih snaga. Formiran je matematički model za proračun optimalnih tokova aktivnih snaga primenom linearnog programiranja i na konkretnom primeru elektroenergetskog sistema prezentovana je primena formiranog matematičkog modela. Rezultati postavljenog problema linearnog programiranja dobijeni su kao izlaz kreiranog Matlab koda za primenu simpleks algoritma na proračun optimalnih tokova aktivnih snaga.

I. UVOD

Optimizacija je proces nalaženja najboljeg mogućeg rešenja pri datim uslovima. Postupak nalaženja optimuma naziva se i matematičko programiranje i ono se generalno izučava kao deo operacionih istraživanja [1].

Sa stanovišta elektroenergetike postoji više modela optimizacije koji mogu da se primene za rešavanje odgovarajućih problema. Jedan od značajnih problema u elektroenergetici jeste određivanje optimalnih tokova snaga elektroenergetskog sistema. Po pravilu, ovo je nelinearni problem i rešava se tehnikama nelinearnog programiranja. U pojedinim slučajevima može se izvršiti linearizacija opšteg nelinearnog modela, što omogućava primenu neke od metoda linearnog programiranja. Na ovaj način mogu se značajno pojednostaviti složeni proračuni, koji zahtevaju i veliki memorijski prostor prilikom izrade računarskih programa. U ovom radu analizirana je primena linearnog programiranja za proračun optimalnih tokova aktivnih snaga.

II. LINEARNO PROGRAMIRANJE I SIMPLEKS ALGORITAM

Linearno programiranje (LP) predstavlja osnovnu disciplinu operacionih istraživanja. To je klasičan optimizacioni metod koji se može primeniti za rešavanje problema sa linearnom kriterijumskom funkcijom i linearnim ograničenjima tipa jednakosti i nejednakosti [1]. Linearno programiranje bilo je predmet brojnih naučnih istraživanja, a čak četiri Nobelove nagrade dodeljene su za doprinose ovoj oblasti. Jednostavno se može kompjuterski implementirati, čime se značajno pojednostavljuje rešavanje složenih optimizacionih problema, a do rešenja se dolazi za konačan broj iteracija.

T. Denić je student na modulu Elektroenergetika, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: teodra.denic@elfak.rs

Jedan od najefikasnijih i najčešće korišćenih algoritama razvijen za rešavanje problema linearnog programiranja jeste simpleks algoritam. Simpleks metod se zasniva na sukcesivnom poboljšanju početnog dopustivog baznog rešenja, sve dok se ne dobije optimalno rešenje.. Efikasnost simpleks algoritma zavisi od broja dopustivih baznih rešenja koje treba ispitati. Simpleks algoritam u svakoj iteraciji pretražuje samo susedna dopustiva bazna rešenja. Na ovaj način se redukuje računarsko vreme u svakoj iteraciji. Međutim, broj iteracija može biti veoma veliki jer kod problema LP sa m ograničenja i n promenljivih u standardnoj formi dopustiva oblast može imati $n!/(n-m)!m!$ ekstremnih tačaka. Broj iteracija je važan faktor kod simpleks algoritma. Zbog svega navedenog od izuzetne važnosti je adekvatno postaviti i modelovati problem LP.

III. MATEMATIČKI MODEL ZA PRORAČUN OPTIMALNIH TOKOVA AKTIVNIH SNAGA PRIMENOM LINEARNOG PROGRAMIRANJA

Kao efikasan postupak za numeričko rešavanje sistema nelinearnih algebarskih jednačina, Njutn-Rafsonov metod (*Newton-Raphson method*) nalazi široku primenu za potrebe proračuna tokova snaga u složenim mrežama. U osnovi Njutn-Rafsonovog metoda leži iterativni postupak sa kvadratnom konvergencijom, pa se do rešenja sa željenom tačnošću može doći u nekoliko iteracija. Međutim, opšti model za proračun tokova snaga, dobijen Njutn-Rafsonovim postupkom, zahteva u svakoj iteraciji izračunavanje inverzne Jakobijeve matrice, koja je u opštem slučaju velike dimenzije. Zbog toga se kod primene klasičnog Njutn-Rafsonovog metoda za proračun tokova snaga u složenim mrežama postavlja povećan zahtev za memorijom, kao i složenija izrada računarskog programa. U cilju prevazilaženja ovih nedostataka razvijeni su raspregnuti metodi za proračun tokova snaga [2].

Za proračun tokova aktivnih snaga, a u cilju formiranja matematičkog modela koji se može rešiti bez velikih numeričkih poteškoća, razvijen je linearni metod, koji se često naziva i DC metod. Linearni model formira se modifikacijom opšteg modela za proračun tokova snaga i koristi se za približan proračun tokova aktivnih snaga, posebno u proračunima na tržištu električne energije [2]. Korišćenjem linearnog modela, uz uvažavanje troškova angažovanja generatorskih jedinica i dodatnih ograničenja, biće formiran matematički model za proračun optimalnih tokova aktivnih snaga primenom LP.

Najpre će se formirati kriterijumska funkcija. Neka je dat opšti slučaj funkcije troškova agregata [3] u generatorskom čvoru i , $i = 1, \dots, N_g$:

$$F_i = a_i + b_i P_{gi} + c_i P_{gi}^2, \quad (1)$$

gde su a_i , b_i i c_i unapred definisani koeficijenti (parametri troškova) i P_{gi} je aktivna snaga generisanja agregata u generatorskom čvoru i . Ukupan broj generatorskih čvorova je N_g . Kriva koja odgovara funkciji troškova može se podeliti na pravougaone segmente oblika:

$$F_i = a_i + b_i P_{gi}, \quad (2)$$

tako da se za rešavanje problema može primeniti LP.

Funkcija cilja se određuje kao suma troškova angažovanja generatorskih jedinica u svim čvorovima:

$$F' = \sum_{i=1}^{N_g} F_i, \quad (3)$$

odnosno u razvijenom obliku:

$$F' = \sum_{i=1}^{N_g} (a_i + b_i P_{gi}) = \sum_{i=1}^{N_g} a_i + \sum_{i=1}^{N_g} b_i P_{gi}. \quad (4)$$

Član $\sum_{i=1}^{N_g} a_i$ predstavlja konstantnu brojnu vrednost, pa ne utiče na rezultat optimizacije i može se eliminisati. Minimizacija troškova angažovanja generatorskih jedinica predstavlja kriterijum po kome se vrši optimizacija:

$$\min F' = \sum_{i=1}^{N_g} b_i P_{gi}. \quad (5)$$

Pri optimizaciji je potrebno zadovoljiti i odgovarajuća ograničenja, odnosno adekvatno modelovati ograničenja koja se realno postavljaju. Kod linearnih proračuna tokova aktivnih snaga zanemareni su gubici u mreži, pa je suma svih injektiranih snaga u mreži jednaka nuli, odnosno:

$$\sum_{i=1}^N P_i = 0, \quad (6)$$

gde je P_i snaga injektiranja u čvor i , dok N predstavlja ukupan broj čvorova. Postavljaju se i sigurnosna ograničenja u odnosu na dozvoljene kapacitete grana oblika:

$$|P_{ij}| \leq P_{ijmax}, \quad i, j = 1, \dots, N, \quad (7)$$

gde je P_{ij} opterećenje grane mreže između čvorova i i j , a P_{ijmax} dozvoljeni kapacitet posmatrane grane. Na kraju, postavlja se donja i gornja granica uglova napona θ_i za sve

čvorove mreže $i = 1, \dots, N$. Usvaja se granica stabilnosti pri malim poremećajima [2]:

$$-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}, \quad i = 1, \dots, N, \quad (8)$$

mada su realno ti uglovi dosta manji. Primena formiranog matematičkog modela za proračun optimalnih tokova aktivnih snaga biće prezentovana na konkretnom primeru EES-a u narednom odeljku.

IV. PRIMER PRIMENE MATEMATIČKOG MODELA ZA PRORAČUN OPTIMALNIH TOKOVA AKTIVNIH SNAGA

Na slici 1 dat je sistem sa 3 generatorska i jednim potrošačkim čvorom. Parametri grana mreže dati su na slici. Dozvoljeni kapaciteti pojedinih grana su:

$$\begin{aligned} P_{12}^{max} &= 150\text{MW}, P_{13}^{max} = 200\text{MW}, \\ P_{14}^{max} &= 100\text{MW}, \\ P_{23}^{max} &= 100\text{MW}, P_{34}^{max} = 150\text{MW}. \end{aligned} \quad (9)$$

Troškovi agregata u generatorskim čvorovima 1, 2 i 4 modelovani su sledećim linearnim funkcijama:

$$\begin{aligned} F_1 &= 100 + 4.5 P_{g1} [\text{NJ/MWh}], \\ F_2 &= 150 + 3.5 P_{g2} [\text{NJ/MWh}], \\ F_4 &= 120 + 4 P_{g4} [\text{NJ/MWh}]. \end{aligned} \quad (10)$$

Ograničenja po snagama proizvodnje za pojedine agregate su:

$$\begin{aligned} 50\text{MW} &\leq P_{g1} \leq 150\text{MW}, \\ 50\text{MW} &\leq P_{g2} \leq 250\text{MW}, \\ 50\text{MW} &\leq P_{g4} \leq 150\text{MW}. \end{aligned} \quad (11)$$

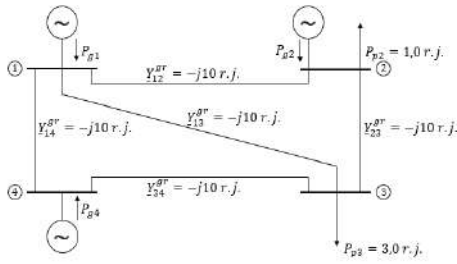
Dozvoljeni kapaciteti pojedinih grana, troškovi agregata u generatorskim čvorovima i ograničenja po snagama proizvodnje za pojedine agregate usvojeni su proizvoljno.

Treba odrediti optimalnu raspodelu tokova aktivnih snaga u datoj mreži tako da ukupni troškovi angažovanja agregata budu minimalni. Proračun je urađen primenom linearnog „DC” modela za proračun tokova aktivnih snaga. Za baznu snagu je usvojena vrednost 100MW. Za balansno referentni čvor usvojen je čvor 1 (važi da je $\theta_1 = 0$). U proračunu nisu razmatrani tokovi reaktivnih snaga i odgovarajuća ograničenja. Takođe, nisu razmatrana ni ograničenja po naponima čvorova.

Ukupni troškovi angažovanja agregata su:

$$F_{uk} = 450 P_{g1} + 350 P_{g2} + 400 P_{g4}. \quad (12)$$

U prethodnom izrazu snage P_{g1} , P_{g2} i P_{g4} su u relativnim jedinicama, a konstantan član je eliminisan, jer ne utiče na rezultat optimizacije.



Slika 1. Elektroenergetski sistem sa četiri čvora [4].

Pri optimizaciji treba zadovoljiti i odgovarajuća ograničenja. Pre svega, to su jednačine injektiranja po čvorovima. Prema "DC" modelu, potrebno je zadovoljiti sledeću jednačinu:

$$\mathbf{B}\boldsymbol{\theta} = -\mathbf{P}, \quad (13)$$

gde je $\mathbf{B}$ matrica admintansi date mreže, $\boldsymbol{\theta}$ vektor uglova fazora napona čvorova i $\mathbf{P}$ je vektor injektiranja aktivnih snaga po čvorovima. Na osnovu datih podataka, za mrežu koja se analizira, prethodna jednačina postaje:

$$\begin{bmatrix} -30 & 10 & 10 & 10 \\ 10 & -20 & 10 & 0 \\ 10 & 10 & -30 & 10 \\ 10 & 0 & 10 & -20 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = - \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = - \begin{bmatrix} P_{g1} - P_{p1} \\ P_{g2} - P_{p2} \\ P_{g3} - P_{p3} \\ P_{g4} - P_{p4} \end{bmatrix} = - \begin{bmatrix} P_{g1} \\ P_{g2} - 1 \\ -3 \\ P_{g4} \end{bmatrix}. \quad (14)$$

Pored ograničenja tipa jednakosti, postavljaju se i ograničenja tipa nejednakosti. Ograničenja po proizvodnji aktivne snage pojedinih agregata mogu se predstaviti sledećim izrazima:

$$\begin{aligned} 0.5 \text{ r.j.} &\leq P_{g1} \leq 1.5 \text{ r.j.}, \\ 0.5 \text{ r.j.} &\leq P_{g2} \leq 2.5 \text{ r.j.}, \\ 0.5 \text{ r.j.} &\leq P_{g4} \leq 1.5 \text{ r.j.} \end{aligned} \quad (15)$$

Za uglove napona usvaja se granica stabilnosti pri malim poremećajima:

$$-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}, i = 1, \dots, 4. \quad (16)$$

Na kraju, preostaju sigurnosna ograničenja po pitanju dozvoljenih prenosnih kapaciteta za pojedine vodove. Ova ograničenja se mogu predstaviti sledećim izrazom:

$$-\mathbf{P}_{gr}^{max} \leq \mathbf{P}_{gr} \leq \mathbf{P}_{gr}^{max}. \quad (17)$$

U relaciji (17) sa $\mathbf{P}_{gr}$ označen je vektor aktivnih snaga grana, dok je sa $\mathbf{P}_{gr}^{max}$ označen vektor dozvoljenih aktivnih snaga

po granama. Imajući u vidu "DC" model, vektor $\mathbf{P}_{gr}$ može se izračunati na sledeći način:

$$\mathbf{P}_{gr} = -\mathbf{B}_{gr}\mathbf{A}\boldsymbol{\theta}. \quad (18)$$

gde je $\mathbf{B}_{gr}$ dijagonalna matrica sa dijagonalnim elementima koji predstavljaju susceptanse pojedinih grana, $\mathbf{A}$ je matrica incidencije grana-čvor, dok je $\boldsymbol{\theta}$ vektor uglova fazora napona čvorova. Za analiziranu mrežu matrice $\mathbf{B}_{gr}$ i $\mathbf{A}$ imaju sledeći oblik:

$$\mathbf{B}_{gr} = \begin{bmatrix} -10 & 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix}, \quad (19)$$

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \quad (20)$$

Vrste u matricama $\mathbf{B}_{gr}$ i $\mathbf{A}$ odgovaraju granama 1-2, 1-3, 1-4, 2-3 i 3-4, respektivno. Zamenom dobijenih vrednosti, ograničenje po dozvoljenim prenosnim kapacitetima postaje:

$$-\begin{bmatrix} 1.5 \\ 2 \\ 1 \\ 1 \\ 1.5 \end{bmatrix} \text{ r.j.} \leq \begin{bmatrix} 10(\theta_1 - \theta_2) \\ 10(\theta_1 - \theta_3) \\ 10(\theta_1 - \theta_4) \\ 10(\theta_2 - \theta_3) \\ 10(\theta_3 - \theta_4) \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 2 \\ 1 \\ 1 \\ 1.5 \end{bmatrix} \text{ r.j.} \quad (21)$$

Na ovaj način formirana su sva potrebna ograničenja.

Uvode se smene: $x_1 = P_{g1}$, $x_2 = P_{g2}$, $x_3 = P_{g4}$, $x_4 = \theta_2$, $x_5 = \theta_3$, $x_6 = \theta_4$. Uglovi fazora napona mogu biti i negativni. Kako bi se zadovoljio uslov nenegativnosti svih promenljivih, zahtevan standardnom formom, uvode se dodatne promenljive, tako da se uglovi fazora napona mogu izraziti kao razlika nenegativnih promenljivih:

$$\begin{aligned} x_4 &= x_7 - x_8 \\ x_5 &= x_9 - x_{10} \\ x_6 &= x_{11} - x_{12}, \\ x_7, x_8, x_9, x_{10}, x_{11}, x_{12} &\geq 0 \end{aligned} \quad (22)$$

Pored toga, potrebno je uvesti i balansne i veštačke promenljive u cilju kreiranja odgovarajuće forme, kako bi se mogao primeniti simpleks algoritam. Nakon što se uvedu odgovarajuće balansne i veštačke promenljive i uvažavajući činjenicu da je $\theta_1 = 0$, dobija se sistem ograničenja (23) u kanoničnoj formi:

$$x_1 + 10x_7 - 10x_8 + 10x_9 - 10x_{10} + 10x_{11} - \dots \quad (23)$$

$$\begin{aligned}
& -10x_{12} + x_{35} = 0 \\
& x_2 - 20x_7 + 20x_8 + 10x_9 - 10x_{10} + x_{36} = 1 \\
& 10x_7 - 10x_8 - 30x_9 + 30x_{10} + 10x_{11} - \\
& \quad -10x_{12} + x_{37} = 3 \\
& x_3 + 10x_9 - 10x_{10} - 20x_{11} + 20x_{12} + x_{38} = 0 \\
& x_1 - x_{13} + x_{39} = 0.5 \\
& x_1 + x_{14} = 1.5 \\
& x_2 - x_{15} + x_{40} = 0.5 \\
& x_2 + x_{16} = 2.5 \\
& x_3 - x_{17} + x_{41} = 0.5 \\
& x_3 + x_{18} = 1.5 \\
& x_7 - x_8 + x_{19} = \pi/2 \\
& -x_7 + x_8 + x_{20} = \pi/2 \\
& x_9 - x_{10} + x_{21} = \pi/2 \\
& -x_9 + x_{10} + x_{22} = \pi/2 \\
& x_{11} - x_{12} + x_{23} = \pi/2 \\
& -x_{11} + x_{12} + x_{24} = \pi/2 \\
& -10x_7 + 10x_8 + x_{25} = 1.5 \\
& 10x_7 - 10x_8 + x_{26} = 1.5 \\
& -10x_9 + 10x_{10} + x_{27} = 2 \\
& 10x_9 - 10x_{10} + x_{28} = 2 \\
& -10x_{11} + 10x_{12} + x_{29} = 1 \\
& 10x_{11} - 10x_{12} + x_{30} = 1 \\
& 10x_7 - 10x_8 - 10x_9 + 10x_{10} + x_{31} = 1 \\
& -10x_7 + 10x_8 + 10x_9 - 10x_{10} + x_{32} = 1 \\
& 10x_9 - 10x_{10} - 10x_{11} + 10x_{12} + x_{33} = 1.5 \\
& -10x_9 + 10x_{10} + 10x_{11} - 10x_{12} + x_{35} = 1.5 \\
& x_i \geq 0, \quad i = 1, \dots, 34; i \neq 4; i \neq 5; i \neq 6.
\end{aligned}$$

U prethodnom sistemu ograničenja (23) sa $x_{13} \dots x_{34}$ označene su balansne promenljive, dok su sa $x_{35} \dots x_{41}$ označene veštačke promenljive.

Za ovako postavljeni problem može se primeniti dvofazni simpleks algoritam za rešavanje problema LP. Cilj prve faze je nalaženje inicijalnog dopustivog baznog rešenja originalnog problema dovođenjem vrednosti veštačkih promenljivih na nulu. U tu svrhu formira se veštačka kriterijumska funkcija W kao suma svih veštačkih promenljivih. Veštačka kriterijumska funkcija minimizira se primenom simpleks algoritma.

$$\min W = \sum_{i=35}^{41} x_i \quad (24)$$

U drugoj fazi dopustivo bazno rešenje dobijeno na kraju prve faze se optimizuje prema originalnoj kriterijumskoj funkciji.

$$\min F_{uk} = 450x_1 + 350x_2 + 400x_3. \quad (25)$$

Na slici 2 može se videti izlaz Matlab programa kreiranog u cilju rešavanja postavljenog problema LP primenom simpleks algoritma.

OptimalBFS = 1x35 table

	x1	x2	x3	x4	x5	x6	x7
1	0.5000	2.1250	1.3750	0	0	0	0.0125

Kao rezultat optimizacije dobijeni su sledeći rezultati: Pg1=0.5000 r.j, Pg2=2.1250 r.j, Pg4=1.3750 r.j, teta2=0.0125 rad, teta3=-0.0875 rad, teta4=0.0250 rad

Pri tome je vrednost ukupnih satnih troškova: Fuk=1518.7500 NJ/h

Slika 2. Izlaz Matlab programa.

Dobijeni su sledeći rezultati: $x_1 = P_{g1} = 0.5000$ r.j, $x_2 = P_{g2} = 2.1250$ r.j, $x_3 = P_{g4} = 1.3750$ r.j, $x_4 = \theta_2 = 0.0125$ rad, $x_5 = \theta_3 = -0.0875$ rad, $x_6 = \theta_4 = 0.0250$ rad. Pri tome je vrednost ukupnih satnih troškova: $F_{uk}^{min} = 1518.7500$ NJ/h. Na osnovu dobijenih rezultata, primenom linearnog „DC” modela, mogu se izračunati tokovi aktivnih snaga po granama: $P_{12} = -12.5$ MW, $P_{13} = 87.5$ MW, $P_{14} = -25$ MW, $P_{23} = 100$ MW, $P_{34} = -112.5$ MW.

V. ZAKLJUČAK

Jedan od značajnih problema elektroenergetike jeste određivanje optimalnih tokova snaga. U ovom radu analizirana je primena linearnog programiranja za proračun optimalnih tokova aktivnih snaga. Korišćen je simpleks metod, koji se smatra najefikasnijom i najpopularnijom metodom za rešavanje problema LP. Kako realni elektroenergetski sistemi imaju stotine, pa i hiljade čvorova, ovaj pristup pogodan je za realizaciju računarskih programa, jer omogućava rešavanje problema velike dimenzionalnosti na brz i efikasan način.

Nakon što je formiran matematički model za proračun optimalnih tokova aktivnih snaga primenom LP, prezentovana je primena formiranog matematičkog modela. Konkretni primer EES-a rešen je u programskom okruženju Matlab, a dobijeni rezultati pokazuju da se linearno programiranje može efikasno koristiti za proračun optimalnih tokova aktivnih snaga.

ZAHVALNICA

Autor se zahvaljuje profesoru dr Draganu Tasiću.

LITERATURA

- [1] A. Savić, D. Šošić, G. Dobrić, M. Žarković, *Metode optimizacije - Primena u elektroenergetici*, Beograd, Akademska misao, 2018.
- [2] D. Tasić, *Analiza elektroenergetskih meža i sistema*, Niš, Elektronski fakultet u Nišu, 2010.
- [3] J. A. Momoh, *Electric power system applications of optimization*, CRC PRESS, 2009.
- [4] N. Rajaković, M. Čalović, P. Stefanov, A. Savić, *100 rešenih zadataka iz analize elektroenergetskih sistema*, Beograd, Elektrotehnički fakultet Beograd, 2002

Realizacija OLED i LCD1602 displeja na Nucleo razvojnom okruženju

Mikica Radenković, Ana Sentov, Katarina Božinović

Sadržaj – U ovom radu je prikazana primena OLED i LCD1602 displeja na Nucleo razvojnom okruženju.

I. UVOD

U ovom projektu biće prikazan sistem za prikazivanje vremena. Za kontinualno očitavanje vremena korišćen je interni RTC (*Real Time Clock*) modul Nucleo okruženja. Za prikazivanje vremena (časova, minuta i sekundi) korišćen je OLED displej. Displej LCD1602 je korišćen za prikazivanje odgovarajuće poruke u skladu sa proteklom vremenom. Na početku svakog minuta na LCD1602 displeju se prikazuje poruka: „Projekat iz optoelektronike“, tako što je u prvom redu ispisano „Projekat iz“, a drugom „optoelektronike“. Prilikom ispisivanja poruke, karakteri se pomeraju svake sekunde, tako da se tekst u oba reda pomera u različitom smeru, sve dok se cele reči ne ispišu na displeju i ne nestanu u potpunosti sa displeja. Nakon toga, potrebno je da oba reda stoje na displeju 3 sekunde. Smer klizanja teksta se menja u zavisnosti od toga da li je počeo parni ili neparni minut.

II. HARDVERSKI DEO

Potrebne komponente za realizaciju projekta su: Nucleo okruženje, LCD1602 displej, OLED displej, protoploča i kratkospojnici. U slučaju da je izabrano neko drugo okruženje, koje ne poseduje interni RTC modul (npr. Arduino), tada bi se na listi potrebnih komponenti našao i eksterni RTC modul.

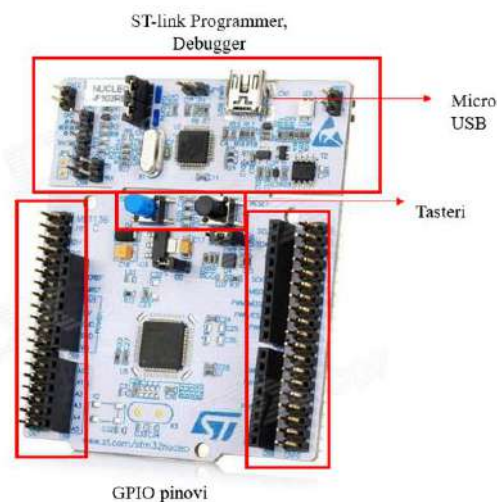
Razvojno okruženje STM32 Nucleo-64 sa STM32F103RB MCU (*Microcontroller Unit*) poseduje 64 GPIO (*General-Purpose Input/Output*) pina, dva korisnička LED-a, dva tastera na ploči - reset i korisnički, kao i micro USB, kao što se može uočiti na slici 1. Jedna od karakteristika je fleksibilno napajanje (3.3V, 5V, 7-12V). Neke od karakteristika STM32F103RB mikrokontrolera su:

- STM32F103RBT6 u LQFP64 paketu
- ARM®32-bit Cortex®-M3 CPU
- 72 MHz maksimalna frekvencija
- VDD from 2.0 V to 3.6 V

M. Radenković i K. Božinović su studentkinje master akademskih studija modula Elektronika i mikrosistemi,

A. Sentov je studentkinja osnovnih akademskih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: radenkovicmikica99@gmail.com, ana.sentov@elfak.rs, katarina.n.bozinovic@gmail.com.

- 128 KB Flash
- 20 KB SRAM
- GPIO (51) sa eksternim interaptom
- 12-bit ADC (2) sa 16 kanala
- RTC
- Tajmeri (4)
- I2C (2)
- USART (3)
- SPI (2)
- USB 2.0
- CAN



Slika 1. Razvojno okruženje STM32 Nucleo-64.

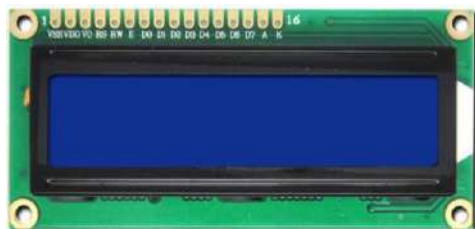
Micro USB je konektor pomoću kog se ovo okruženje povezuje sa računarom i na taj način se vrši njegovo programiranje odgovarajućim softverskim alatima. U ovom slučaju korišćeni programi su: STM32CubeMX i Keil uVision5. STM32 Nucleo-64 je okruženje koje pruža mnogo mogućnosti. Poželjno je imati rasporedi pinova ovog okruženja ispred sebe na početku svakog projekta, jer se oznake na okruženju u većini slučajeva ne odnose na redni broj porta i pina. Na slici 2 može se videti raspored pinova ovog okruženja.

LCD – *Liquid Crystal Display* [2] je vrsta displeja čiji se rad zasniva na svojstvu tečnih kristala. Tečni kristali su materijali koji pokazuju osobine i tečnih i čvrstih materijala. Princip rada ove vrste displeja se zasniva na prozračnosti tečnih kristala, pa tako tečni kristal ne može direktno da emituje svetlost, zato se koristi pozadinski izvor svetlosti i propuštaju se promenljive količine belog svetla kroz aktivni filter. U zavisnosti od toga da li su u

pitanju *Passiv Matrix* ili *Active Matrix* displej, adresiranje piksela sa vrši na drugačiji način, ali kod oba tipa dovođenjem određenog napona na piksel vrši se ispisivanje nekog sadržaja na displeju. Izgled LCD displeja koji se koristi u ovom projektu prikazan je na slici 3.



Slika 2. Prikaz oznaka portova i pinova.



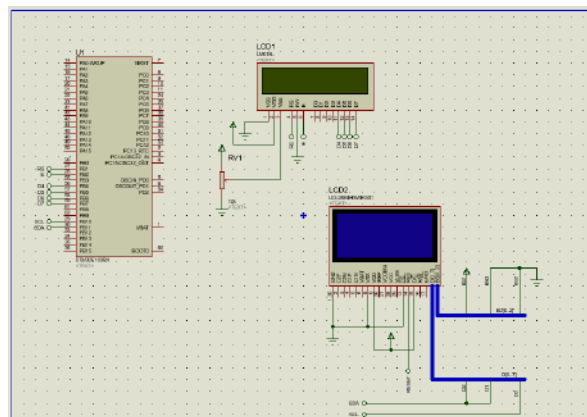
Slika 3. LCD displej.

OLED SSD1306 [3] predstavlja moderniji tip displeja, pre svega zbog svojih malih dimenzija. OLED (*Organic Light Emmiting Diode*) predstavlja vrhunac tehnologije u području izrade displeja i izvora svetlosti. Ovakav tip displeja je sačinjen od super tankih slojeva organskog materijala koji počinju da svetle čim se kroz njega propusti mala struja [4]. Displeji bazirani na OLED tehnologiji mogu da prikazuju pun spektar boja. Modul koji je korišćen u ovom projektu pored OLED displeja koristi SSD1306 drajver koji omogućava veliki broj podešavanja i olakšava rad sa displejem [4]. Povezuje se preko samo 4 pina, što dosta olakšava rad i rasterećuje mikrokontroler, sa kojim komunicira preko I2C komunikacije. Na slici 4 je prikazan OLED sa drajverom SSD1306 koji je korišćen u ovom projektu [4].



Slika 4. OLED SSD1306 modul.

III. SIMULACIJA U PROGRAMU PROTEUS



Slika 5. Šema u programu Proteus.

Pre same fizičke realizacije projekta potrebno je i poželjno da se prvo osmisli i uz pomoć simulatora realizuje šema projekta. Na slici 5. prikazana je šema projekta. Pinovi PB4, PB5, PB6 i PB7 uz pomoć labela su povezani sa pinovima na LCD displeju, D4, D5, D6 i D7. RS i EN pin LCD displeja su povezani na pinove PB1 i PB2 mikrokontrolera. PB10 i PB11 su povezani sa pinovima SCL i SDA OLED displeja, respektivno. Ostali pinovi LCD-a povezani su na odgovarajući način. OLED displej koji se koristi za fizičku realizaciju ovog projekta ima samo 4 pina, dok je ovaj u simulaciji drugačiji, ali je povezan na odgovarajući način tako da ima istu ulogu. Mnogi pinovi se ne koriste pa su ostali nepovezani. Sama simulacija je verodostajna predstava fizičke realizacije projekta, jer su korišćeni identični pinovi za povezivanje.

IV. SOFTVERSKI ALATI

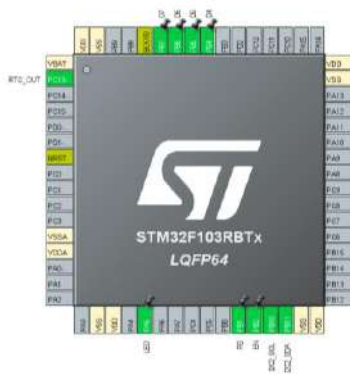
Softverski alat koji je korišćen za realizaciju ovog projekta sastoji se od dva nezavisna ali međusobno povezana programa: STM32CubeMX i Keil uVision5. Sam proces pisanja koda odvija se u dve faze:

Pre samog pisanja koda potrebno je izvršiti odgovarajuća podešavanja u programu STM32CubeMX. Na samom početku potrebno je izabrati odgovarajuće okruženje koje se koristi, da bi sam program generisao odgovarajući korisnički interfejs. U ovom slučaju potrebno je izabrati NUCLEO-F103RB (slika 6). Kao što je prikazano na slici 7. prvo se obeležavaju pinovi koji se koriste i poželjno je označiti pinove tako da je očigledna njihova namena. Svi pinovi su podešeni kao izlazni, osim pinova za I2C komunikaciju koji se automatski zauzimaju po njenom uključivanju. Pored I2C komunikacije koju je potrebno uključiti, za potrebe ovog projekta uključuje se i RTC – *Real Time Clock*, čija su podešavanja prikazana na slici 8. Kao format za pronos podataka izabran je binarni tip, što je kasnije propraćeno i u kodu. Nakon svih podešavanja, u delu koji se naziva *Clock Configuration*, potrebno je podesiti frekvenciju rada mikrokontrolera, kao i

to da li će se koristiti interni ili eksterni *clock*. U ovom projektu je jako važno izabrati interni *clock* zbog ispravnog rada RTC modula. Na kraju potrebno je generisati kôd i preći u program Keil uVision5



Slika 6. Izbor razvojnog okruženja u programu STM32CubeMX.



Slika 7. Podešavanja u STM32CubeMX-u.



Slika 8. Podešavanje RTC-a.

U drugoj fazi projekta softverski alat je izgenerisao određeni „kostur“ kôda koji je potrebno dopuniti i prilagoditi zadatku. Predefinisani delovi kôda su tu radi lakšeg manipulisanja bibliotekama, podacima i promenljivama, kao i lakšeg pristupanju i pozivanju

funkcija. Potrebno je poštovati označene delove kôda koji su predviđeni za korisnika, jer na taj način se kôd čuva i ukoliko dođe do neke promene u STM32CubeMX-u i ponovnog generisanja kôda. U nastavku će biti u delovima prikazan kôd korišćen u ovom projektu.

Na samom početku potrebno je uključiti biblioteke koje su potrebne u radu sa displejima i pored toga uključena je i biblioteka za rad sa stringovima:

```
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include "STM_MY_LCD16X2.h" //Biblioteka za LCD displej
#include "fonts.h" //Biblioteka za fontove za OLED displej
#include "ssd1306.h" //Biblioteka za OLED displej
#include "string.h" //Biblioteka za prosleđivanje stringova
/* USER CODE END Includes */
```

- U ovom delu kôda deklarišu se sve potrebne promenljive:

```
/* USER CODE BEGIN PV */
RTC_TimeTypeDef sTime; //RTC struktura preko koje se prosleđuje vreme
uint8_t sati, minuti, sekunde; //Promenljive u kojoj se čuvaju sati, minuti i sekunde
char time1[10], time2[10], time3[10]; //Promenljive koje se prosleđuju na displej
uint8_t i = 1; //Brojač za pomeranje teksta za određeni broj karaktera
uint8_t k = 0; //Brojač za pojavljivanje teksta 3s na displeju
char poruka1[16] = "Projekat iz ";
//Tekst koji se ispisuje u prvom redu
char poruka2[16] = "optoelektronike "; //Tekst se ispisuje u drugom redu
uint8_t fleg = 0, fleg1 = 0; //Flegovi koji služe za odabir smera kretanja poruka
/* USER CODE END PV */
```

U korisničkom kodu 0 predviđeno je pisanje funkcija. Funkcije nisu neophodne, ali u ovom projektu su korišćene da bi preglednost i optimalnost kôda bila bolja. U naredne tri IF petlje proverava se da li je vrednost svake od promenljiva *sati*, *minuti* ili *sekunde* jednocifreni ili dvocifreni broj. Ukoliko je jednocifreni, na mestu desetice postavljamo nulu, međutim ukoliko je dvocifreni uklanjamo tu nulu, tako što je prepisujemo stvarnom vrednošću desetice, postavljanjem kursora za par piksela ulevo.

```
/* Private user code -----*/
/* USER CODE BEGIN 0 */
void OLED_Vremena(void){
//Prvi IF služi sa proveravanje sati
if(sati < 9) {
    SSD1306_GotoXY(0, 0);
    //Postavljanje kursora u prvi red OLED-a
    sprintf((char*)time1, "H: 0%d", sati);
    //Ispisivanje nule na mestu desetice
```

```

        SSD1306_Puts(time1,&Font_11x18, SSD1306_
COLOR_BLACK); //Prosleđivanje sata, fonta i boje
teksta na OLED displeju
    }
    else {
        SSD1306_GotoXY(0, 0);
        sprintf((char*)time1, "H: %2d", sati);
        //Brisanje nule na mestu desetica
        SSD1306_Puts(time1,&Font_11x18,SSD1306_COL
OR_BLACK);
    }
//Drugi IF služi sa proveravanjem minuta
if(minuti < 9) {
    SSD1306_GotoXY(0, 15); //Postavljanje
kursora u drugi red OLED-a
    sprintf((char*)time2, "M: 0%d", minuti);
    //Ispisivanje nule na mestu desetica
    SSD1306_Puts(time2,&Font_11x18,
SSD1306_COLOR_BLACK);
}
else {
    SSD1306_GotoXY(0, 15);
    sprintf((char*)time2, "M: %2d", minuti);
    //Brisanje nule na mestu desetica
    SSD1306_Puts(time2,&Font_11x18,
SSD1306_COLOR_BLACK);
}
//Treći IF služi sa proveravanjem sekundi
if(sekunde < 9) {
    SSD1306_GotoXY(0, 30);
    //Postavljanje kursora u treći red OLED-a
    sprintf((char*)time3, "S: 0%d", sekunde);
    //Ispisivanje nule na mestu desetica
    SSD1306_Puts(time3,&Font_11x18,
SSD1306_COLOR_BLACK);
}
else {
    SSD1306_GotoXY(0, 30);
    sprintf((char*)time3, "S: %2d", sekunde);
    //Brisanje nule na mestu desetica
    SSD1306_Puts(time3,&Font_11x18,
SSD1306_COLOR_BLACK);
}
    SSD1306_UpdateScreen(); //Ažuriranje ekrana
zbog "mrtvih" piksela
}
/* USER CODE END 0 */

U korisničkom kodu 2 se vrši inicijalizacija
displejeva i postavljanje početnog vremena.

/* USER CODE BEGIN 2 */

    SSD1306_Init(); //Inicijalizacija OLED
displeja
    SSD1306_Fill(SSD1306_COLOR_WHITE); //OLED
displej se boji u belo
    LCD1602_Begin4BIT(GPIOB, RS_Pin, EN_Pin,
D4_GPIO_Port, D4_Pin, D5_Pin, D6_Pin, D7_Pin);
//Inicijalizacija LCD displeja
    sTime.Hours = 10; //Postavljanje sata na
željeno vreme
    sTime.Minutes = 10; //Postavljanje minuta na
željeno vreme
    sTime.Seconds = 50; //Postavljanje sekunde na
željeno vreme
    HAL_RTC_SetTime(&hrtc,&sTime,RTC_FORMAT_BIN);
//Postavljanje RTC trenutnog vremena

/* USER CODE END 2 */

Glavni korisnički kod 3:

```

```

/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */

//=====OLED=====//
    HAL_RTC_GetTime(&hrtc, &sTime, RTC_FORMAT_BIN);
    //Očitavanje vremena sa RTC
    sati = sTime.Hours; //Promenljiva sati uzima
vrednosti sata sa RTC-a
    minuti = sTime.Minutes; //Promenljiva
minuti uzima vrednosti minuta sa RTC-a
    sekunde = sTime.Seconds; //Promenljiva sekunde
uzima vrednosti sekundi sa RTC-a
    OLED_Vremena(); // Pozivanje funkcije za prikaz
vremena
//=====LCD=====//
    if(sekunde == 59) { //Proveravamo da li sekunde
imaju vrednost 59 tj. da li će se promeniti minut
        fleg = 1;
        //Postavljamo fleg na 1
    }
    if (minuti % 2 == 0) { //Proveravamo da li
je minut paran
        fleg1 = 0; //Postavljamo fleg1 na
    }
    else if (minuti % 2 == 1) { //Proveravamo
da li je minut neparan
        fleg1 = 1; //Postavljamo fleg1 na
1
    }
    if(fleg == 1) { //Provera da li su
sekunde došle do 59
        if (fleg1 == 0) { // Ukoliko je minut
paran brojač broji od 1 do 16
            for (i = 1; i<16; i++) {

                LCD1602_setCursor(1,i); // y=1, x
pozicija zavisi od FOR petlje i kreće se unapred
                LCD1602_print("Projekat iz");
                //Štampanje odredjenog teksta u prvom redu
                LCD1602_setCursor(2,16-i); //Postavlja
kursor na y=2, a x pozicija zavisi od FOR petlje
i kreće se unazad
                LCD1602_print("optoelektronike");
                //Štampanje odredjenog teksta u drugom redu
                HAL_Delay(250);
                LCD1602_clear(); //Sadržaj LCD displeja se
briše

                //Ažuriranje vremena
                HAL_RTC_GetTime(&hrtc, &sTime, RTC_FORMAT_BIN);

                sati = sTime.Hours;
                sekunde = sTime.Seconds;
                minuti = sTime.Minutes;

                OLED_Vremena(); //Pozivanje funkcije za proveru
ispisa vremena na OLED displeju
            }
            LCD1602_clear(); //Brisanje LCD displeja

            //Brojač koji broji 3s za istovremeno ispisivanje
teksta
            for (k = 0; k < 3; k++) {
                LCD1602_setCursor(1,1); //Postavljanje LCD
kursora u prvom redu, na prvoj poziciji

                LCD1602_print("Projekat iz"); //Štampanje
odredjenog teksta u prvom redu
            }
        }
    }
}
}

```

```

        LCD1602_setCursor(2,1);
//Postavljanje LCD kursora u drugom redu, na
prvoj poziciji
        LCD1602_print("optoelektronike");
//Štampanje odredjenog teksta u drugom redu

//Ažuriranje vremena
HAL_RTC_GetTime(&hrtc, &sTime, RTC_FORMAT_BIN);
sati = sTime.Hours;
sekunde = sTime.Seconds;

sekunde = sTime.Seconds;
OLED_Vremena();

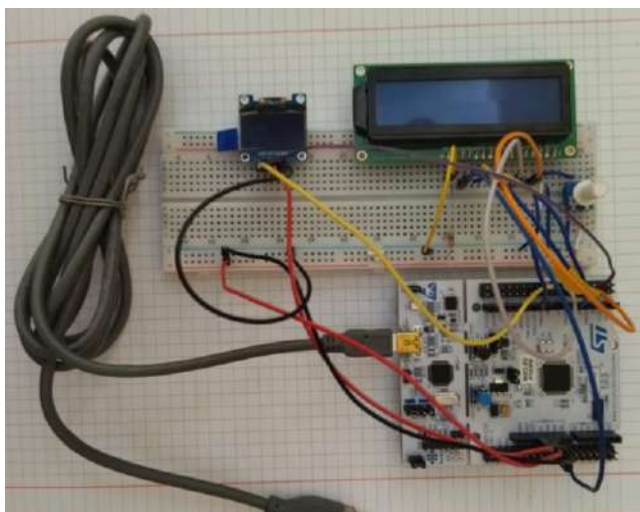
//Pozivanje funkcije za proveru ispisa vremena
na OLED displeju
HAL_Delay(1000);
HAL_RTC_GetTime(&hrtc, &sTime, RTC_FORMAT_BIN);
sati = sTime.Hours;
sekunde = sTime.Seconds;
sekunde = sTime.Seconds;

LCD1602_clear();
}

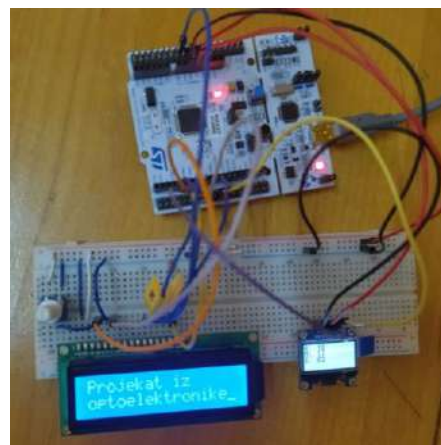
```

V. PRAKTIČNA REALIZACIJA

Za povezivanje komponenata u kolu korišćena je protoploča. Komponente su povezivane na osnovu šeme realizovane u simulatoru. Na slici 9. je prikazan izgled realizovanog projekta, dok se na slici 10. može videti ispisivanje vremena na OLED displeju i odgovorajuće poruke na LCD-u sinhronizovano sa vremenom.



Slika 9. Fizička realizacija projekta.



Slika 10. Prikaz poruka na displeju.

VI. ZAKLJUČAK

U ovom projektu prikazan je princip rada LCD i OLED displeja, jedan deo njihovih mogućnosti, kao i primena I2C komunikacije. Prikazivanje vremena se na sličan način vrši i na digitalnim satovima. Jedna vrsta unapređenja projekta bila bi prikazivanje temperature na OLED displeju, očitavanjem nekog temperaturnog senzora koji bi sa mikrokontrolerom komunicirao preko I2C ili neke druge komunikacije. Prikazivanje sadržaja na LCD displeju može biti raznovrstan, pa je tako osim ovih poruka moguće prikazati bilo koje druge, kao i neke prostije slike ili animacije. Na LCD-u je takodje moguće prikazivati i vreme i temperaturu, ali OLED ima bolji kontrast, veću osvetljenost i veću brzinu osvežavanja, pa je zbog toga pogodniji.

ZAHVALNICA

Autori se zahvaljuju nastavnicima i saradnicima Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu na pruženoj pomoći i saradnji tokom realizacije ovog projekta.

LITERATURA

- [1] Komponente i kola snage, Miloš Marjanović, Katedra za mikroelektroniku, Elektronski fakultet Niš, 2020.
- [2] Tehnička specifikacija za LCD1602.
- [3] Tehnička specifikacija za OLED3306.
- [4] Optoelektronika, Osnove serijskih komunikacija, Nikola Mitrović, Katedra za mikroelektroniku, Elektronski fakultet Niš, 2020.

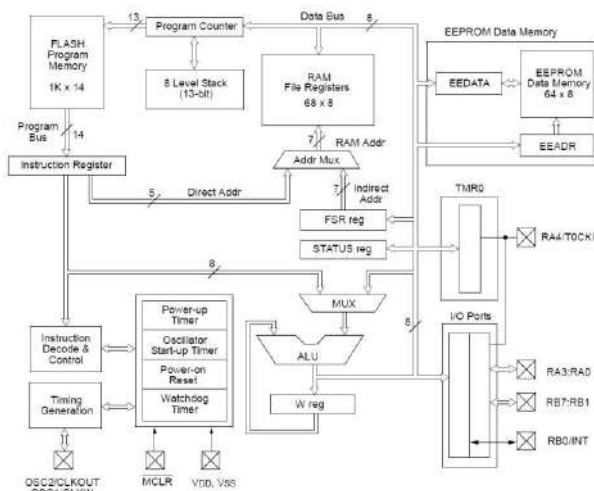
Projektovanje tajmera korišćenjem PIC mikrokontrolera

Bogdan Krstić

Sadržaj-U ovom radu je opisana realizacija tajmera pomoću PIC mikrokontrolera 16F84A-04. Tajmer ima mogućnost brojanja do 10 minuta. U realizaciji su korišćeni sedmosegmentni displeji za prikaz stanja tajmera.

I. TEORIJSKI UVOD O MIKROKONTROLERU

PIC 16F84A-04 spada u 8-bitne mikrokontrolere RISC arhitekture. PIC(Peripheral Interface Controller) mikrokontroleri su mikrokontroleri američke kompanije *Microchip Technology* [3]. Blok šema korišćenog mikrokontrolera data je na slici 1.



Slika 1. Blok šema mikrokontrolera 16F84A-04.

Centralna procesorska jedinica izvršava korisnički program čime upravlja procesom obrade i uzajamnom komunikacijom ostalih delova mikrokontrolera. U CPU nalaze se: ALU, dekodir instrukcija (*Instruction Decoder*), registar instrukcija (*Instruction Register*), akumulator (*W reg*), programski brojač (*Program Counter*), registar uslova (*Status reg*), *FSR reg*, *MUX*, *Addr Mux*... *FSR* registar koristi se kod indirektnog adresiranja. Multiplekser (*MUX*) omogućava da se u ALU prosleđuju upravljački signali ili podaci koji se obrađuju. Adresni multiplekser (*Addr Mux*) omogućava da se vrši direktno ili indirektno adresiranje memorijskih lokacija u RAM memoriji.

Programska memorija (*FLASH*) je kapaciteta

1024×14, odnosno ima 1024 14-bitne memorijske lokacije i namenjena je za smeštanje korisničkog programa. EEPROM memorija je kapaciteta 64×8 (64B), odnosno ima 64 8-bitne memorijske lokacije i namenjena je za smeštanje podataka koje treba sačuvati i kada nema napajanja. Jedan od takvih podataka je na primer zadata temperatura kod temperaturnih regulatora. EEPROM memoriji pristupa se preko *EEADR* i *EEDATA* registara po posebnoj proceduri kako bi se izbegao slučajan upis. U registar *EEDATA* upisuje se podatak koji treba upisati u ovu memoriju ili koji je pročitao iz ove memorije. U registar *EEADR* upisuje se adresa memorijske lokacije EEPROM memorije kojoj se pristupa. RAM memorija za podatke je kapaciteta 68×8 (68B), odnosno ima 68 8-bitnih memorijskih lokacija i namenjena je za smeštanje podataka, kao što su promenljive ili međurezultati, koje program koristi u svom izvršavanju i čiji gubitak u slučaju prekida napajanja ne predstavlja problem za rad uređaja nakon ponovnog uključenja napajanja. Stek (*Stack*) memorija ili magacinska memorija, koja je kapaciteta 8×13, odnosno ima 8 13-bitnih lokacija, namenjena je za čuvanje povratnih adresa (to su adrese od kojih se nastavlja izvršenje programa nakon završetka potprograma ili prekidnih potprograma).

Programski brojač (*PC - Program Counter*) je 13-bitni brojač u kome se nalazi adresa instrukcije koja se izvršava. Ako ta instrukcija nije **CALL** ili **GOTO**, kojima se ostvaruje programski skok, tada se nakon izvršenja instrukcije sadržaj programskog brojača automatski povećava za jedan. Ako se tekućom instrukcijom zahteva skok, tada se nakon izvršenja instrukcije sadržaj programskog brojača menja za više od jedan. Inače ovaj programski brojač može da adresira 8k memorijskih lokacija. Brojač (*TMR0*), koji predstavlja tajmersku jedinicu, je osmobični binarni brojač koji radi nezavisno od programa i koji na svaki četvrti takt impuls iz taktnog generatora povećava svoju vrednost za jedan sve dok ne dostigne maksimalnu vrednost (255) a onda brojanje ponovo kreće od nule. Ako se brojač pobuđuje impulsima iz taktnog generatora tada tajmerska jedinica radi kao tajmer, a ako se brojač pobuđuje impulsima iz spoljnog izvora (senzora) tada tajmerska jedinica radi kao brojač koji broji proizvode na traci, broj obrtaja osovine, prolaznike i slično.

PORTA i *PORTB* su ulazno/izlazni portovi (*I/O Ports*) koji čine U/I jedinicu. *PORTA* je petobitni a *PORTB* osmobični. Takti generator ili generator takta (*Timing Generator*) generiše takt impulse (takt) koji se koriste pri

B. Krstić je student druge godine osnovnih akademskih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: bkrstic@elfak.rs

formiranju upravljačkih signala i na taj način sinhronizuju rad ostalih komponenata MCU. Sigurnosni tajmer (Watchdog Timer - WDT) služi za resetovanje MCU ako dođe do „zaglavljivanja“ programa. Power-on Reset (POR) resetuje MCU prilikom uključanja napajanja. Power-up Timer (PWRT) ima ulogu da, nakon uključanja napajanja, MCU drži u resetovanom stanju bar 72 ms dok se ne stabilise napon napajanja. Oscillator Start-up Timer (OST) ima ulogu da, nakon stabilizacije napona napajanja, MCU drži u resetovanom stanju sve dok se ne stabilise oscilacije taktog generatora.

PIC16F84A-04 ima osamnaest nožica, odnosno pinova i izrađuje se u varijanti kućišta DIL18 (slika 2) i SMD kućištu. Uloge pinova mikrokontrolera PIC16F84A-04 date su na slici 3.



Slika 2. Izgled mikrokontrolera 16F84A-04.

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 brojač TMR0 pobuđuje impulsima iz spoljnog izvora, tada se impulsi dovode na ovaj pin.
Pin br.4	MCLR	Reset ulaz i ulaz napona programiranja mikrokontrolera (12 do 13,5V).
Pin br.5	Vss	Ulaz za masu napajanja.
Pin br.6	RB0/INT	Multi 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 PCLC u programskom modu.
Pin br.13	RB7	Sedmi pin porta B. Dodatna funkcija je „Data“ linija PCLD 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	Multi pin porta A.
Pin br.18	RA1	Prvi pin porta A.

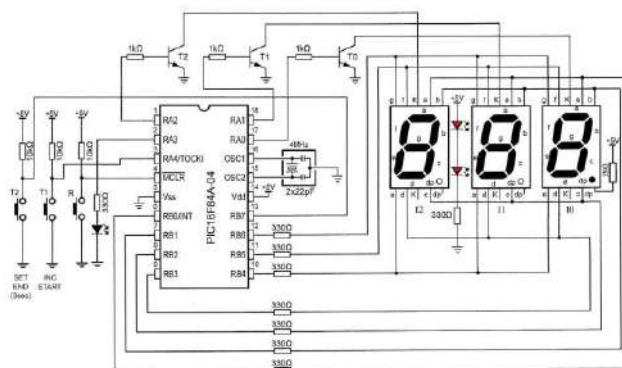
Slika 3. Uloge pinova mikrokontrolera 16F84A-04.

Fizički, port je registar unutar MCU koji je preko dodatnih digitalnih kola i tranzistora povezan sa pinovima mikrokontrolera. MCU PIC16F84A04 ima dva porta, petobitni PORTA i osmobitni PORTB. Ova dva porta čine ulazno-izlaznu jedinicu koja omogućava vezu MCU sa spoljnim svetom. Pinovi porta mogu biti ulazni i izlazni. Ako je pin izlazni, tada se na njemu postavlja ono stanje (0 ili 1) koje je i u memorijskoj ćeliji porta sa kojom je povezan. Ako je pin ulazni, tada se stanje (0 ili 1), koje je na njemu uspostavljeno delovanjem spoljnog sveta, upisuje u memorijsku ćeliju porta sa kojom je povezan. Dakle, preko ulaznih pinova u portove se upisuju kombinacije

nula i jedinica a preko izlaznih pinova MCU saopštava okolini kombinacije nula i jedinica koje se nalaze u portovima. Svi pinovi porta A i B mogu biti ulazni ili izlazni. Za definisanje ulaznih i izlaznih pinova porta A koristi se registar TRISA a za definisanje ulaznih i izlaznih pinova porta B koristi se registar TRISB. Ako je odgovarajući bit TRIS registra setovan, odnosno ima vrednost logičke jedinice, tada je pin, kome je pridružen taj bit, ulazni. Ako je odgovarajući bit TRIS registra resetovan, odnosno ima vrednost logičke nule, tada je pin, kome je pridružen taj bit, izlazni. U toku rada MCU može se menjati namena pinova čime se menja smer prenosa podataka. Registri PORTA i PORTB nalaze se u Bank 0 a registri TRISA i TRISB u Bank 1.

II. REALIZACIJA TAJMERA

Za realizaciju tajmera upotrebljene su sledeće elektronske komponente: MCU, 3 LED displeja za prikazivanje vrednosti vremena, 3 tranzistora NPN tipa koji služe kao prekidači, 3 tastera i oscilatorno kolo za generisanje signala u određenim frekvencijama. MCU je povezan sa ostalim elektronskim komponentama kao na slici 4. Pinovi RB7 i RA4 su ulazni a svi ostali pinovi porta A i B su izlazni. Nakon uključanja napajanja, LED dioda, indikatori I2, I1 su isključeni, a na indikatoru I0 je cifra 0.



Slika 4. Električna šema tajmera.

Tajmer služi za merenje vremenskog intervala koji se može podesiti na bilo koju vrednost od 1 s do 9 min 59 s. Za formiranje željenog vremenskog intervala od 1s do 9 min 59 s, koriste se promenljive xm i xs čije su početne vrednosti 0. Promenljiva xm služi za podešavanje minuta a promenljiva xs za podešavanje sekundi. Svakim pritiskom na taster T1 (INC), promenljiva xm se inkrementira. Ako je taster T1 stalno pritisnut, tada se xm povećava nakon svake 0,4s. Ako je trenutna vrednost promenljive $xm=9$, tada se, pritiskom na taster T1, promenljivoj xm dodeljuje vrednost 0. Kada se podesi željeni broj minuta xm , to se potvrđuje pritiskom na taster T2 (SET), nakon čega se podešava broj desetice sekundi, a da bi to bilo moguće neophodno je da se prethodno otpusti taster T2. Svakim pritiskom na taster T1 (INC), promenljiva xs povećava se za 10. Ako je taster

T1 stalno pritisnut, tada se xs povećava nakon svake 0,4s. Ako je trenutni broj desetica promenljive xs 5, tada se, pritiskom na taster T1, promenljiva xs smanjuje za 50. Kada se podesi željeni broj desetica promenljive xs , to se potvrđuje pritiskom na taster T2 (SET), nakon čega se podešava broj jedinica sekundi, a da bi to bilo moguće neophodno je da se prethodno otpusti taster T2. Svakim pritiskom na taster T1 (INC), promenljiva xs povećava se za 1. Ako je taster T1 stalno pritisnut, tada se xs povećava nakon svake 0,4s. Ako je trenutni broj jedinica promenljive xs 9, tada se, pritiskom na taster T1, promenljiva xs smanjuje za 9. Kada se podesi željeni broj jedinica promenljive xs , to se potvrđuje pritiskom na taster T2 (SET) [2].

Pritiskom na taster T1, na RA4 uspostavlja se logička nula, četvrti bit porta A se resetuje čime započinje odbrojanje, dekrementiranjem promenljivih xm i/ili xs (zavisno od njihovih trenutnih vrednosti) nakon isteka svake sekunde. Nakon svakog dekrementiranja, promenljive xm i xs bekapuju se u EEPROM memoriji, kako bi bile sačuvane u slučaju prekida napajanja. Nakon ponovnog uspostavljanja napajanja, arhivirane vrednosti se dodeljuju promenljivim xm i xs . Ako je bilo koja arhivirana vrednost veća od nule, tada se automatski nastavlja odbrojanje vremena. Stanja promenljivih xm i xs prikazuju se na 7-segmentnim LED indikatorima. Kada promenljive xm i xs dostignu vrednost 0, prekida se njihova promena i uključuje se LED dioda. Ako se pre isteka definisanog vremena, taster T2 (END) drži pritisnut 3 sekunde, tada se promenljivim xm i xs dodeljuje vrednost 0, čime se završava odbrojanje vremena. Za merenje vremenskog intervala od 3 sekunde, koristi se promenljiva $x3$.

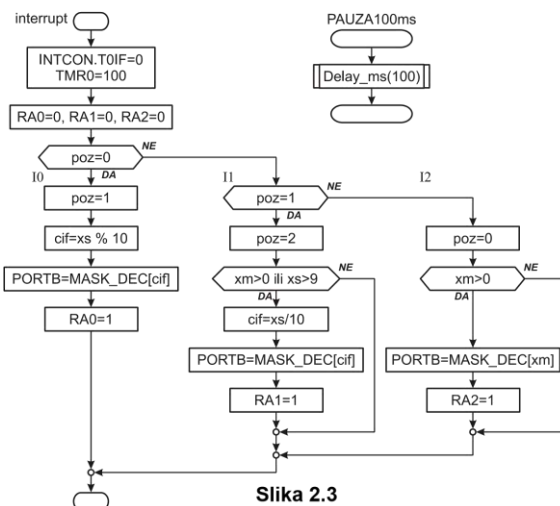
Pinovi RA0, RA1 i RA2 su izlazni. Kada je promenljiva $poz=0$, tada je RA0=1, RA1=0, RA2=0, tranzistor T0 provodi, tranzistori T1 i T2 su zakočeni, indikator I0 je uključen a indikatori I1 i I2 su isključeni. Kada je promenljiva $poz=1$, tada je RA0=0, RA1=1, RA2=0, tranzistori T0 i T2 su zakočeni, tranzistor T1 provodi, indikatori I0 i I2 su isključeni a indikator I1 je uključen. Kada je promenljiva $poz=2$, tada je RA0=0, RA1=0, RA2=1, tranzistori T0 i T1 su zakočeni, tranzistor T2 provodi, indikatori I0 i I1 su isključeni a indikator I2 je uključen. Stanja na pinovima RA0, RA1 i RA2 menjaju se nakon svakih 5ms, tako da indikatori rade sa vremenskim multipleksom. Na indikatoru I0 prikazuje se cifra jedinica promenljive xs , na indikatoru I1 prikazuje se cifra desetica promenljive xs a na indikatoru I2 prikazuje se cifra jedinica promenljive xm . Indikator I1 uključuje se samo ako je $xm>0$ ili $xs>9$. Indikator I2 uključuje se samo ako je $xm>0$. Vremenski interval od 5ms dobija se uz pomoć s brojača TMR0. Algoritam toka glavnog programa dat je na slici 5.

Brojač se pobuđuje internim takt signalom, čija je perioda 1 μ s, pri čemu je faktor deljenja preskalera 32 tako da brojač menja stanja nakon svake 32 μ s. Početno stanje brojača je 100. Kada brojač promeni stanje sa 255 na 0,

odnosno nakon 156 promena stanja, nastupa prekid (interrupt) zbog prekoračenja maksimalne vrednosti brojača. Dakle, prekid se javlja nakon svake 32 μ s $\times$ 156=4,992ms. Kada nastupi prekid, brojač se pripremi za novi ciklus brojanja, a uključuje se indikator koji je u toku prethodne 5ms bio isključen. Za pamćenje pozicije indikatora koji je bio uključen u prethodnom prekidu, koristi se promenljiva "poz". Da bi preskaler bio dodeljen brojaču i da bi faktor deljenja preskalera bio 32, potrebno je da sadržaj registra OPTION_REG bude 0B00000100. Jednodimenzionalni niz (vektor) "MASK_DEC[]" sadrži maske decimalnih cifara za sedmosegmentni LED indikator sa zajedničkom katodom (slika 6). Svaki element niza (maska) je maska one decimalne cifre koja je jednaka indeksu elementa niza. Elementi niza deklarirani su kao konstante da bi kao takvi bili smesteni u programsku memoriju. Pritiskom na taster R, na pinu MCLR uspostavlja se logička nula čime se MCU resetuje. S obzirom da je broj upisa u EEPROM 10 $\cdot$ 10, to znači da je broj punih radnih ciklusa (od 9:59 do 0) tajmera, 10 $\cdot$ 10 /599=16694. Ako se iz programa izbaci deo za EEPROM, tada je broj punih radnih ciklusa tajmera neograničen. Algoritam toka interrupt funkcije dat je na slici 7.

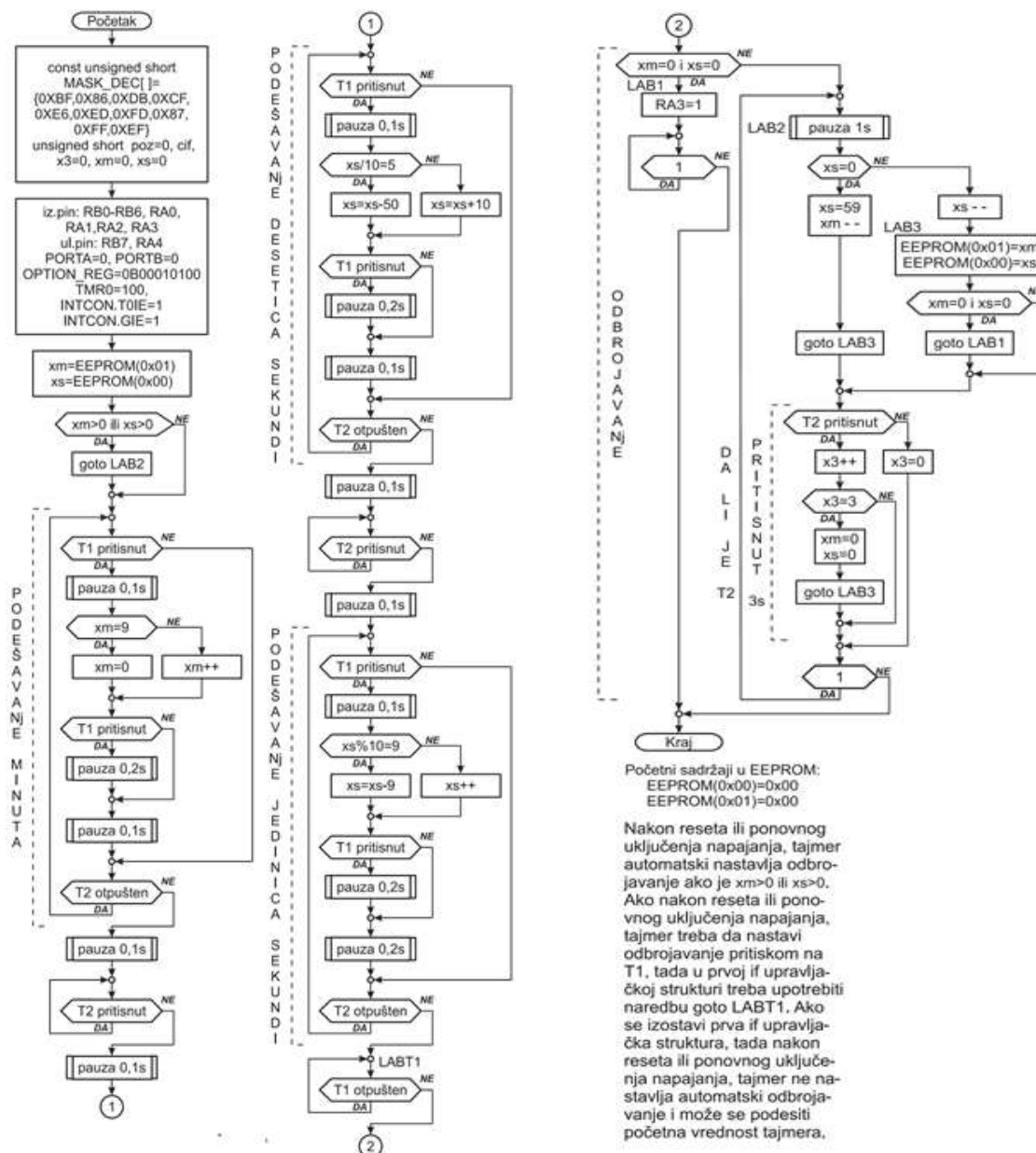
Izlazni pinovi porta B	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	
Priključci 7S LED indikatora	dp	g	f	e	d	c	b	a	HEX
Decimalna cifra									
0	1	0	1	1	1	1	1	1	0XBF
1	1	0	0	0	0	1	1	0	0X86
2	1	1	0	1	1	0	1	1	0XDB
3	1	1	0	0	1	1	1	1	0XCF
4	1	1	1	0	0	1	1	0	0XE6
5	1	1	1	0	1	1	0	1	0XED
6	1	1	1	1	1	1	0	1	0XFD
7	1	0	0	0	0	1	1	1	0X87
8	1	1	1	1	1	1	1	1	0XFF
9	1	1	1	0	1	1	1	1	0XEF

Slika 6. Maske decimalnih cifara za sedmosegmentni LED indikator sa zajedničkom katodom.



Slika 2.3

Slika 7. Algoritam ISR-a mikrokontrolera.



Slika 5. Algoritam glavnog programa mikrokontrolera.

VI. ZAKLJUČAK

Korišćenjem PIC mikrokontrolera i poznavanjem njegove arhitekture kao i upotrebom razvojnog okruženja MikroC PRO for PIC realizovan je tajmer čiji je opseg od 1s do 9 min i 59 sec. Prikazani sistem ima širok spektar upotrebe u svakodnevnom životu.

LITERATURA

- [1] PIC16F84A datasheet, Microchip, 2001.
- [2] Nebojša Matić, PIC mikrokontroleri, MikroElektronika, Beograd, 2003.
- [3] Milan Verle, PIC mikrokontroleri, MikroElektronika, Beograd, 2008.
- [4] Milan Verle, PIC microcontrollers Programming in C, MikroElektronika, Beograd, 2009.

Simulacija samostalnog fotonaponskog sistema

Neda Stanojević, Aleksandar Pantić

Sadržaj – U ovom radu objašnjena je procedura projektovanja jednog samostalnog (*engl. stand-alone*) fotonaponskog (PV) sistema pomoću softvera *PVsyst*. Rad ima za cilj da omogući osnovne informacije o tipu sistema koji se projektuje, kao i o rezultatima simulacije.

I. UVOD

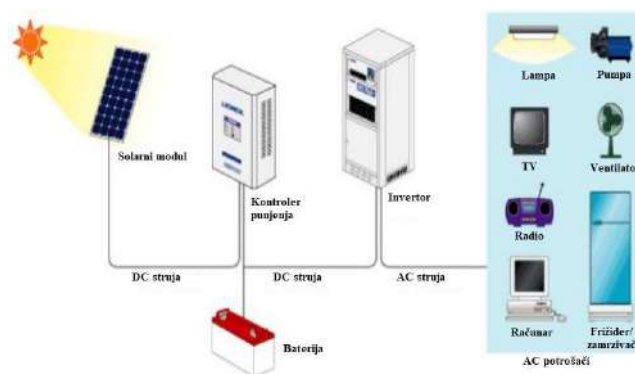
Od davnina su ljudi na različite načine koristili energiju dobijenu od Sunca, a danas se smatra da je solarna energija osnova za dobijanje svih drugih vidova energije na Zemlji. Svake godine, kako u celom svetu, tako i u Srbiji, raste broj instaliranih fotonaponskih sistema čija snaga se u proseku kreće od nekoliko kW do nekoliko desetina GW. Na tržištu je dostupan veliki izbor fotonaponskih modula, invertora, akumulatora i ostalih komponenata koje su potrebne za izgradnju jednog kompletnog fotonaponskog sistema, što pruža mogućnost da se projektuje sistem prema potrebama, željama i budžetu korisnika.

Danas je dostupan veliki broj licenciranih softverskih paketa koji olakšavaju projektovanje fotonaponskih sistema i pružaju rezultate simulacije sa svim karakteristikama sistema koji se projektuje. U ovom radu korišćen je softverski paket *PVsyst*, jedan od najkorišćenijih programa za dizajn, optimizaciju i analizu kompletnog PV sistema i koji sadrži, kako veliku bazu lokacija, tako i veliku bazu svih komponenata dostupnih na tržištu [1].

II. SAMOSTALNI PV SISTEM

Fotonaponska ćelija je ćelija čiji se rad bazira na fotonaponskom efektu, a njena uloga je da detektuje svetlost, a zatim da je pretvori u električnu energiju. Najčešće se prave od monokristalnog silicijuma, polikristalnog silicijuma i tankog filma, a u zavisnosti od ovog materijala zavisi i njihova efikasnost. Izlazna snaga PV ćelije je mala, te se one grupišu i tako formiraju fotonaponske module. U zavisnosti od tipa solarne ćelije koju sadrže, vrši se i podela PV modula. Iako najveću efikasnost imaju PV moduli na bazi monokristalnog silicijuma, najzastupljeniji su PV moduli na bazi polikristalnog silicijuma zbog niže cene, jednostavne proizvodnje i dobrih karakteristika u ekstremnim uslovima. Svi fotonaponski sistemi mogu se podeliti na: autonomne,

samostalne (*engl. stand-alone*) sisteme, hibridne sisteme i mrežne sisteme (*engl. grid-connected*). Samostalni fotonaponski sistemi su sistemi koji za svoj rad ne zahtevaju priključenje na električnu mrežu (*grid-off*), a njihova osnovna uloga je da napajaju stambeni objekat potrebnom strujom. Ovi sistemi se najčešće koriste na udaljenim ili nepristupačnim objektima, pokretnim objektima, za uličnu rasvetu, signalizaciju, alarmne sisteme, video nadzore, bazene za navodnjavanje. Pri dimenzionisanju samostalnog fotonaponskog sistema mora se voditi računa o poziciji objekta, potrošačima i vremenu autonomije. Svaki sistem ovog tipa sastoji se od izabranih solarnih modula, kontrolera koji kontroliše punjenje/praznjenje, invertora koji snabdeva uređaje kojima je potreban naizmenični napon, akumulatora koji skladišti energiju i isporučuje je u odsustvu Sunca ili u toku noći, kablova, montažnih delova i opterećenja, odnosno uređaja. Na slici 1 prikazana je šema samostalnog fotonaponskog sistema [2].



Slika 1. Samostalni fotonaponski sistem.

III. PROJEKTOVANJE SAMOSTALNOG PV SISTEMA

Kada se projektuje fotonaponski sistem, osim odabira optimalnih komponenti, neophodno je obratiti pažnju i na meteorološke uslove lokacije na kojoj je planirana montaža sistema. Softverski alati koje nudi *PVsyst* korisniku dosta olakšavaju ovaj posao. Dakle, softver sadrži veliku bazu podataka lokacija, tako da je moguće projektovati sistem za bilo koju lokaciju na svetu, kao i veliku bazu podataka svih komponenti različitih proizvođača potrebnih za realizaciju bilo kog tipa fotonaponskih sistema. *PVsyst* predlaže korisniku optimalnu komponentu prema definisanim parametrima, vrši proračune i daje ekonomsku procenu troškova. Kao i za svaki drugi softverski paket koji služi za projektovanje fotonaponskih sistema, tako je slučaj i sa *PVsyst*-om, da su dve osnovne grupe parametara koje treba

N. Stanojević i A. Pantić su studenti doktorskih studija na modulu Nanotehnologije i mikrosistemi na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: neda.s@elfak.rs, aleksandar.pantic@elfak.ni.ac.rs

definisati: lokacija na kojoj će se projektovati PV sistem i unos podataka sa osnovnim meteorološkim podacima za tu lokaciju. Softverski paket sastoji se iz dva glavna dela. Prvi je *Preliminary design* koji omogućava korisniku ispitivanje polazne ideje, odnosno, takozvanog idejnog (preliminarnog projekta). Drugi deo je *Project design* koji pruža detaljne rezultate simulacije i detaljne ekonomske proračune troškova. Preporučljivo je da korisnik uradi prvo idejni projekat kako bi ispitao svoju ideju.

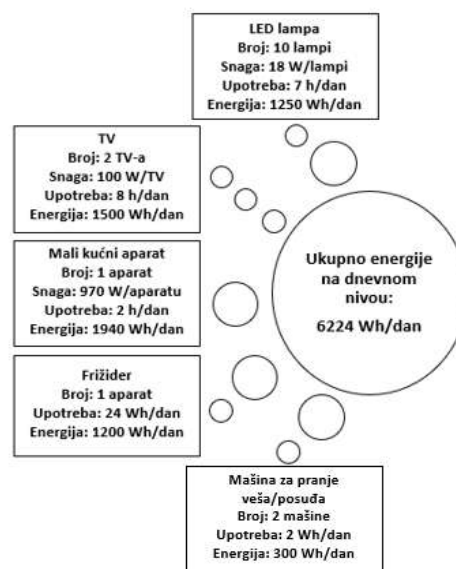
Osnovni parametri koji utiču na kvalitet i efikasnost projektovanog sistema su: geografski položaj sistema, uglovi orijentacije i uticaj senki, karakteristike PV modula i ostalih komponenata, kao i usaglašenost PV modula i invertora [1].

A. Preliminarni dizajn projekta (*Preliminary design*)

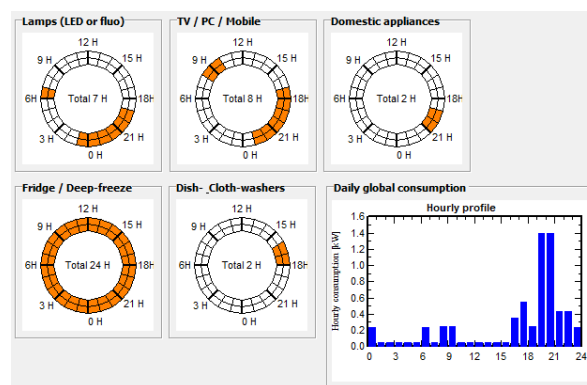
Sistem koji je projektovan je za domaćinstvo u blizini grada Niša sledećih geografskih koordinata: 43,32° geografske širine i 21.90° geografske dužine. Geografska visina je 200 m, a albedo efekat defisan na 0.2 %. Albedo efekat je koeficijent refleksije i računa se kao odnos između reflektovane energije zračenja i intenziteta globalnog zračenja, a izražava se u procentima [2]. Softver nudi tabelu sa prosečnim vrednostima albeda za različite površine. Uglavnom se, kao prosečna vrednost, uzima vrednost albeda od 0.2 %. Ali, na primer, ako je lokacija za koju se projektuje fotonaponski sistem na planini, onda se može uzeti veća vrednost albeda (0.8 %) zbog većeg broja dana kada je planina prekrivena snegom.

Datoteke sa meteorološkim podacima, kao što su prosečne dnevne, mesečne i godišnje temperature, maksimalne i minimalne temperature za to područje, vlažnost, brzina vetra, oblačnost, insolacija, itd., mogu se uzeti iz mnogih izvora koje nudi softver, a u projektu je korišćen *Meteonorm 7.2*.

Glavni deo preliminarnog projekta je definisanje potrošača. Kada se definišu potrebe korisnika podrazumeva se definisanje spiska svih električnih potrošača koji će biti priključeni na projektovani sistem. Na fotonaponski sistem povezani su sledeći potrošači: 10 LED lampi snage 18 W po lampi sa dnevnim korišćenjem od 7 h, 2 televizora snage 100 W po uređaju koji se koristi 8 h dnevno, frižider snage 1.2 kW koji radi 24 h, 2 mašine za pranje veša/posuđa snage 150 W po jednom pranju koje se koriste 2 h dnevno i jedan mali kućni aparat snage 970 W po uređaju koji se koristi 2 h u toku dana. Simulator daje podatke da ukupna energija koja je potrebna na dnevnom nivou iznosi 6224 Wh po danu. Na slici 2 prikazano je definisanje potrošača koji su povezani na projektovani fotonaponski sistem, a na slici 3 definisanje korišćenja uređaja u *PVsyst*-u u toku dana po satima, a u cilju izračunavanja ukupne energije.



Slika 2. Definisanje potrošača povezanih na fotonaponski sistem.



Slika 3. Definisanje korišćenja potrošača na dnevnom nivou.

B. Dizajn glavnog projekta (*Project design*)

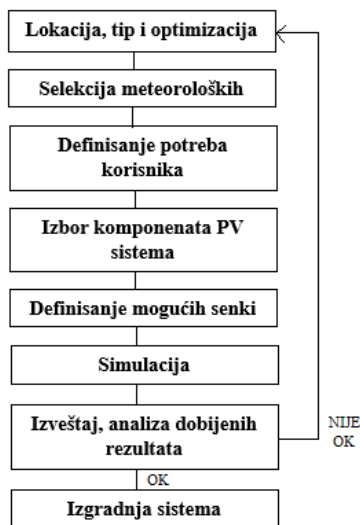
Nakon urađenog idejnog projekta koji je pokazao da zamišljeni fotonaponski sistem ima perspektivu i da će biti zadovoljene potrebe korisnika, može se preći na rad potpune simulacije u okviru dela *Project design*. Pri projektovanju sistema potrebno je slediti algoritam prikazan na slici 4 [4].

Lokacija i potrošači koji će biti priključeni na sistem već su definisani u okviru preliminarnog projekta, ali je potrebno definisati orijentaciju. Objekat ima orijentaciju sever-jug ($azimuth=0^\circ$), a nagibni ugao modula je 36° ($tilt=36^\circ$) i ovo je prikazano na slici 5. Azimut ugao definiše se kao ugao između juga i mesta na kom će fotonaponski panel biti postavljen. Optimalan azimut ugao u solarno podne je 0° .

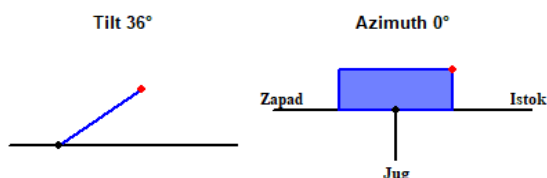
Moduli se montiraju na krov objekta, a izabran je sistem sa fiksiranim fotonaponskim modulima. Takođe, pri projektovanju sistema je posebno važno izabrati optimalne komponente. Najvažnije karakteristike kojima korisnik

treba da se vodi pri izboru komponenata za sistem koji projektuje su efikasnost, pouzdanost, kapacitet, cena, izgled, integrisana elektronika. S obzirom da na tržištu postoji veliki izbor komponenata različitih proizvođača, različitih cena i karakteristika, svakako da izbor nije ni malo lak. Vrlo je važno da projektant održava balans između cene i kvaliteta, a poznato je i očekivano da će kvalitetnije komponente veće efikasnost imati i višu cenu [1].

Pri izboru optimalne baterije mora se voditi računa o njenim cenama na tržištu, jer je moguće da u toku životnog veka projektovanog sistema korisnik mora menjati akumulator jednom ili nekoliko puta. U ovom projektu korišćene su dve paralelno vezane litijum-jonske baterije, modela *Powerwall2*, proizvođača *Tesla*, kapaciteta 536 Ah. Moduli su od monosilicijuma, modela *Black ESM 340 W* sa 72 solarne ćelije, snage 340 Wp, nominalnog napona od 34 V i efikasnosti od 17.5 %. Izabran je univerzalni kontroler sa DC-DC konverterom maksimalne efikasnosti od 97 %. Za projekat je korišćeno 11 solarnih modula koji su vezani paralelno na ukupnoj površini krova od 21.3 m². Maksimalna snaga instaliranih modula je 3740 Wp. U okviru projektovanja sistema uzimaju se u obzir i gubici. Najznačajniji faktor koji utiče na gubitak energije je zagrevanje sistema (20 W/m²K), ali osim ovoga definišu se i gubici zbog prljanja modula (3 %), gubici zbog kvaliteta i nepodudaranja modula i omski gubici žica za povezivanje [3].

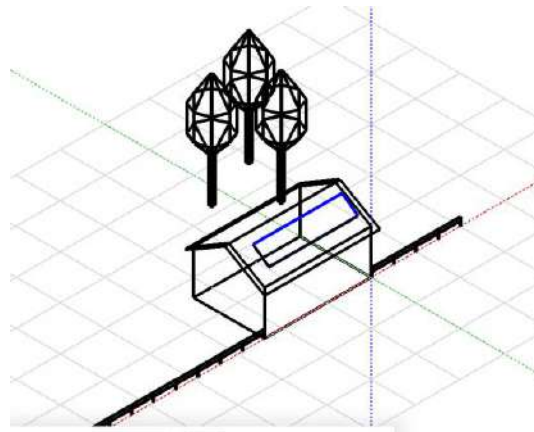


Slika 4. Algoritam za projektovanje fotonaponskih sistema.



Slika 5. Optimizacija uglova fotonaponskih modula.

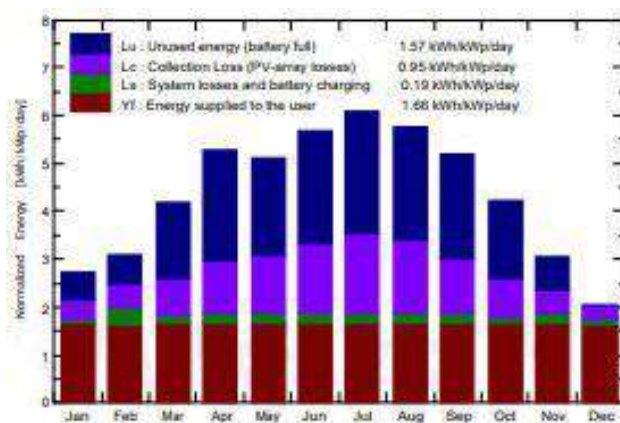
Pomoću opcije *Near Shading* projektant može da skicira objekat za koji projektuje fotonaponski sistem, ali i okolne objekte bliže ili dalje koji mogu stvarati senke i time uticati na efikasnost sistema. Senke mogu stvarati i sami paneli ukoliko su oni montirani suviše blizu [3]. Na slici 6 prikazan je objekat za čije potrebe je projektovan fotonaponski sistem.



Slika 6. Analiza uticaja senke na projektovani PV sistem.

IV. REZULTATI SIMULACIJE

Nakon definisanja svih parametara, pokreće se simulacija. Rezultati simulacije u vidu grafikona, tabela i proračuna, kao i glavni podaci o samom sistemu, dati su u izveštaju. Dostupna energija sistema je 4551 kWh godišnje, a PR faktor koji predstavlja odnos korisne energije i energije zračenja oko 70 %. Normalizovana proizvodnja po instaliranom kWp po mesecima prikazana je na slici 7 [6].



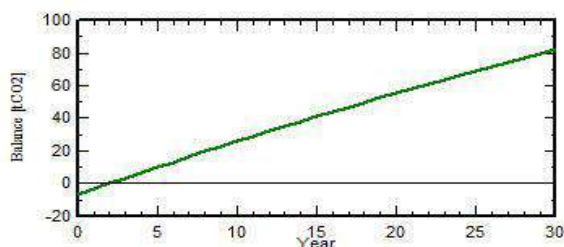
Slika 7. Proizvedena energija po kW instalirane snage PV sistema.

Sa slike 7 može se videti da sistem zadovoljava potrebe korisnika, ali i da je količina dostupne energije mnogo veća u toku letnjih meseci. Jako je bitno naglasiti da se tada taj višak proizvedene energije može koristiti za još neki potrošač koji nije definisan od strane korisnika na

početku projektovanja sistema. Naravno, mora se uzeti u obzir i činjenica da će se uređaji ređe ili češće koristiti na dnevnom nivou, odnosno u toku godine.

U okviru softverskog paketa PVsyst moguće je predvideti i troškove, kao i isplativost projektovanog sistema. Proračun koji je softver napravio definiše da je cena ovog sistema oko 7.000 EUR, ali treba uzeti u obzir da najveći deo troškova čine same komponente.

U izveštaju nakon odrađene simulacije postoje i podaci o redukciji emisije ugljen-dioksida. Ako se uzme u obzir da će se sistem, uz redovno održavanje, koristiti 30 godina, u proračunu je navedeno da će se sačuvati 82.2 tone ugljen-dioksida (slika 8). Proračun balansa emisije ugljen-dioksida PVsyst izvodi u skladu sa preporukama *International Energy Agency* (IEA). U izveštaju je navedeno da će se uštedeti 752 gr ugljen-dioksida po 1 kWh.



Slika 8. Odnos uštede ugljen-dioksida tokom životnog veka projektovanog sistema.

V. ZAKLJUČAK

U radu je opisana procedura projektovanja jednog samostalnog fotonaponskog sistema korišćenjem softverskog simulatora PVsyst. Ovaj sistem idealan je za korišćenje na mestima gde nije moguće povezivanje na distributivnu mrežu. U izveštaju je dostupna potpuna tehnološka, ali i ekonomska analiza, što je veoma korisno za inženjere koji se bave projektovanjem PV sistema, ali i za istraživače koji svoja istraživanja baziraju na solarnoj energiji.

ZAHVALNICA

Ovaj rad je realizovan u okviru projekta finansiranog od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije (TR-33035).

LITERATURA

- [1] R. Mitić, S. D. Pantić, S. Aleksić, D. Pantić, "Projektovanje i simulacija karakteristika samostalnog PV sistema snage do 5 kW", *Zbornik radova 57. konferencije ETRAN, Zlatibor*
- [2] D. Pantić, "Solarne komponente i sistemi- predavanja", 2020.
- [3] M. Đorđević, M. Savić, A. Vulović, M. Rajković, "Simulacija mrežnog fotonaponskog sistema", *IEEEESTEC 7th Student Project Conference*, Niš, ISBN 978-86-6125-114-6
- [4] N. Stanojević, A. Pantić, F. Filipović, B. Banković, S. Aleksić, M. Petronijević, D. Pantić, "CAD analysis of grid-on photovoltaic power plant design and cost-effectiveness", *TELSIKS*, 2021.
- [5] National Electronics and Computer Technology Center (NECTEC), "Stand alone PV system design using PVsyst"
- [6] PVsyst Website, <http://www.pvsyst.com/en/>

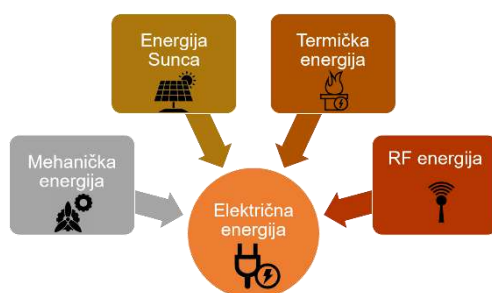
Termoelektrični modul kao izvor napajanja u čvorovima bežičnih senzorskih mreža - pregled

Miloš Marjanović

Sadržaj - Prikupljanjem različitih vidova energije iz okoline i njenim pretvaranjem u električnu energiju omogućen je rad samonapajajućih senzorskih čvorova i elektronskih komponenti. U ovom radu dat je literaturni pregled različitih samonapajajućih sistema zasnovanih na termoelektričnim modulima.

I. UVOD

Proces prikupljanja i pretvaranja energije iz različitih izvora u okolini u električnu energiju naziva se *energy harvesting*. Pored baterija ovo je često korišćen način napajanja čvorova bežičnih senzorskih mreža (*Wireless Sensor Network node* – WSN čvor) i drugih *embedded* sistema. Energija Sunca, radio talasa, termička i mehanička energija mogu biti izvori u *energy harvesting* sistemima za napajanje WSN čvorova (slika 1). Najzastupljenija su fotonaponska rešenja, gde se WSN čvorovi napajaju preko solarnih panela, kako na otvorenom, tako i u zatvorenom prostoru. Mehanička energija može se konvertovati u električnu pomoću tri mehanizma: elektrostatičkog, piezoelektričnog, elektromagnetnog. Na primer, vibracije ili pritisak, kao oblici mehaničkog naprezanja, izazivaju deformacije piezoelektričnog materijala što dovodi do generisanja napona. Pretvaranje termičke u električnu energiju zasnovano je na Zibekovom efektu i u tu svrhu koriste se termoelektrični moduli (TEM) [1]. Usled razlike u temperaturi dolazi do generisanja napona. U ovom radu detaljno će biti opisan princip rada termoelektričnih modula i biće analizirana primena tih modula kao izvora napajanja u čvorovima WSN-a.



Slika 1. Najčešći izvori energije u *energy harvesting* sistemima.

M. Marjanović je student doktorskih studija na modulu Nanotehnologije i mikrosistemi, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milos.marjanovic@elfak.ni.ac.rs

U pogledu efikasnosti, posle baterija (do 90 %) najbolji odnos uložene i dobijene snage pokazuju izvori na bazi vibracija i radio talasa - do 50 % [2]. Slede solarne ćelije sa efikasnošću do 24 %, dok najmanju efikasnost pokazuju termoelektrični moduli (do 3 %). S druge strane, višak neiskorišćene toplote je dosta prisutan u industriji, gde ovi moduli, iako slabo efikasni, mogu biti primenjeni kao izvori napajanja za čvorove WSN-a, pa će iz tih razloga u ovom radu biti opisani.

II. TERMOELEKTRIČNI MODULI

Termoelektrični efekti (Zibekov, Peltijev, Džulov i Tomsonov) definišu vezu između termičke i električne energije u metalima i poluprovodnicima. Ovde će akcenat biti na Zibekovom efektu u poluprovodnicima s obzirom da TEM-ovi zasnivaju rad na ovom efektu. Analiziraćemo strukturu sa slike 2 koju čine dva poluprovodnička elementa p- i n- tipa (termoelementi) koji su povezani preko metalnih kontakata tako da čine termoelektrični par. Termoelementi se tipično prave od dopiranih legura Bi-Te i Pb-Te. Kada se ovakva struktura izloži temperaturnom gradijentu (s jedne strane strukture je temperatura T_{hot} – vruća strana, a s druge strane je temperatura T_{cold} – hladna strana) generisaće se Zibekov napon:

$$V = \alpha \Delta T = \alpha (T_{hot} - T_{cold}), \quad (1)$$

gde je $\alpha = \alpha_p - \alpha_n$ – ekvivalentni Zibekov koeficijent za poluprovodnik p- i n-tipa, a ΔT je temperaturna razlika. Naime, usled temperaturne razlike, pokretni nosioci naelektrisanja sa vruće strane difunduju ka hladnoj strani. Nagomilavanje nosilaca naelektrisanja rezultira neto naelektrisanjem (negativnim za elektrone i pozitivnim za šupljine) na hladnoj strani i pojavom napona. Konvencionalni TEM-ovi sadrže veliki broj termoelektričnih parova (slika 2) koji su električno redno povezani preko metalnih kontakata, a termički paralelno preko keramičkih ploča. Metalni kontakti prvog i poslednjeg termoelementa u nizu predstavljaju izvođe TEM-a kao generatora (TEG). Generisani napon otvorenog kola TEG-a pri temperaturnim razlikama od nekoliko desetina °C je reda stotina mV, tako da je prilikom njegovog korišćenja za napajanje uređaja uvek potrebno kolo za upravljanje električnom energijom.

Termičke otpornosti keramičkih ploča utiču na promenu raspodele temperature unutar TEG-a (slika 3), a time i na vrednost generisanog napona:

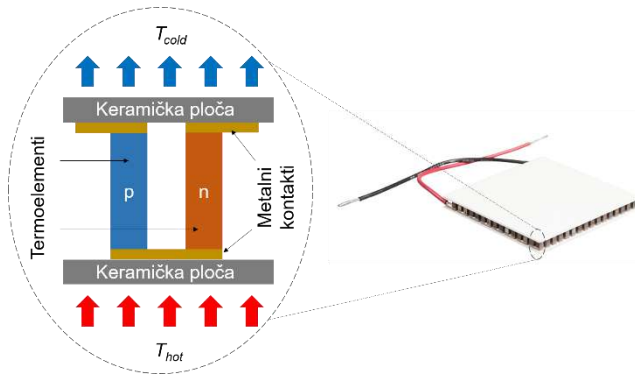
$$V_{TEG} = N\alpha\Delta T' = N\alpha \frac{R_{thTEG}}{R_{thTEG} + 2R_{thP}} \Delta T, \quad (2)$$

gde je N broj termoelektričnih parova. Termičke otpornosti svih termoelektričnih parova i keramičkih ploča su, respektivno:

$$R_{thTEG} = \frac{l}{2N\lambda A}, \quad (3)$$

$$R_{thP} = \frac{l_p}{\lambda_p A_{TEG}}, \quad (4)$$

gde su l dužina termoelementa, A površina njegovog poprečnog preseka, l_p debljina keramičke ploče, A_{TEG} spoljašnja površina TEG-a (istovremeno i keramičkih ploča), λ termička provodnost termoelektričnog materijala i λ_p termička provodnost keramičkog materijala.



Slika 2. Termoelektrični modul: unutrašnja struktura i fizički izgled termoelektričnog para.

Električna otpornost svih termoelektričnih parova čini unutrašnju otpornost TEG-a:

$$R_{TEG} = 2N\rho \frac{l}{A}, \quad (5)$$

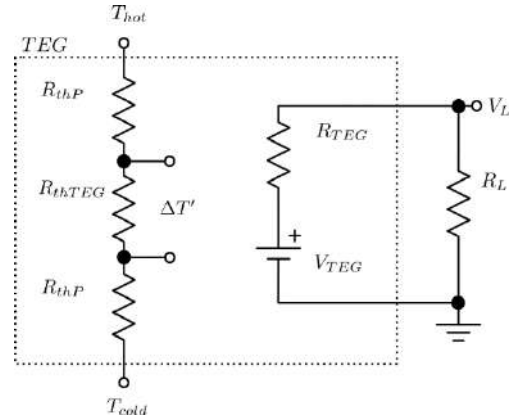
gde je ρ specifična električna otpornost termoelektričnog materijala i ova veličina je temperaturno zavisna. Kada je na TEG priključeno opterećenje R_L , kroz njega će proticati struja:

$$I_L = \frac{V_{TEG}}{R_{TEG} + R_L}. \quad (6)$$

Napon na potrošaču će biti: $V_L = V_{TEG} - R_{TEG}I_L$, tako da je snaga koju TEG predaje potrošaču jednaka $P_L = V_L I_L$.

U uslovima proticanja struje kroz TEG, dolazi do izražaja i Peltijev efekat koji efektivno smanjuje vrednost generisanog napona jer dovodi do smanjenja temperature

T_{hot} i porasta temperature T_{cold} proporcionalno jačini struje. Usled toga je ovaj efekat više izražen pri malim vrednostima opterećenja R_L .



Slika 3. Ekvivalentno termo-električno kolo TEG-a sa priključenim opterećenjem.

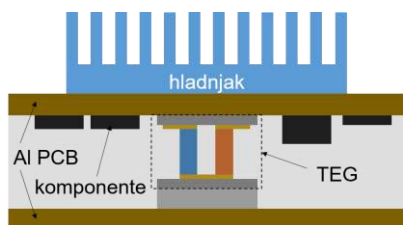
Velika primena termoelektričnih modula u različitim industrijama dovodi do proširenja tržišta ovih komponenta. Prema studiji objavljenoj u [3], godišnja stopa rasta potražnje za TEM-ovima biće 7.3% od 2021. do 2027. godine, te se očekuje da će do tada tržište TEM-ova biti vredno 866 miliona dolara. Standardni TEM-ovi (slika 2) su ostvarili najveći udeo na tržištu u 2020. godini, u poređenju sa mikro TEM-ovima realizovanim u tehnologiji tankih filmova. Zahvaljujući razvoju 5G telekomunikacionih sistema i automobilske industrije prognozira se da će u budućnosti na tržištu biti dominantni mikro TEM-ovi [3]. Fleksibilni mikro TEM-ovi omogućiće komercijalizaciju nosivih mikrosistema koji toplotnu energiju čoveka konvertuju u električnu [4].

III. PRIMENA TERMoeLEKTRIČNIH MODULA KAO IZVORA NAPAJANJA U ČVOROVIMA WSN-A

Savremeni čvorovi bežičnih senzorskih mreža imaju potrošnju od oko 100 μ W do 10 mW. S obzirom na potrebnu električnu energiju i dug životni vek, TEM-ovi su pogodni izvor napajanja u ovakvoj vrsti mikrosistema.

U radu [5] predstavljen je čvor WSN-a koji sadrži PIC mikrokontroler koji vrši akviziciju vrednosti temperature sa senzora i šalje podatke preko RF modula. Čvor je realizovan u kompaktnoj formi korišćenjem štampanih ploča sa aluminijumskim jezgrom (PCB) i TEM-om PE-031-10-20 kao generatorom (slika 4). Ovaj TEG sadrži 31 termoelektrični par. Donja štampana ploča koristi se kao kolektor toplote za vruću stranu TEG-a, a gornja za odvođenje toplote. PCB sa aluminijumskim jezgrom obezbeđuje ravnomerniju raspodelu temperature u poređenju sa PCB-om od FR-4 materijala [6]. Na gornjem PCB-u montiran je niskoprofilni aluminijumski hladnjak da obezbedi bolji prenos toplote ka ambijentu. Elektronske komponente sistema smeštene su na gornjem PCB-u, dok je

prostor između štampanih ploča ispunjen termoizolatorskom penom da bi se smanjila neželjena razmena toplote. Pored TEG-a, elektronski deo čvora čini blok za upravljanje i skladištenje električne energije baziran na *boost* kolu LTC3108 sa superkondenzatorom. Ukupne dimenzije čvora WSN-a su $(52 \times 35 \times 17.6)$ mm. Čvor je projektovan da obezbedi pouzdan hladni start. Eksperimentalno je pokazano da je minimalna temperaturna razlika od $\Delta T = 13.5^\circ\text{C}$ potrebna da bi TEG generisao napon od 40 mV, pri kome *boost* kolo počinje da daje napon dovoljan za rad sistema, i pri čemu je vreme hladnog starta 90 s. Pri temperaturnoj razlici od 25°C ovo vreme skraćuje se na 20 s i tada je prikupljena snaga 2.04 mW. Za opisani čvor WSN-a razvijen je model i urađene su FEM elektro-termičke stacionarne [7] i SPICE vremenski zavisne [8] simulacije. Ovaj SPICE model može se koristiti u procesu projektovanja i optimizacije čvora WSN-a da bi se izabrao odgovarajući TEG sa više različitih aspekata (efikasnosti, vremena hladnog starta, dimenzija čvora, maksimalne temperature primene).

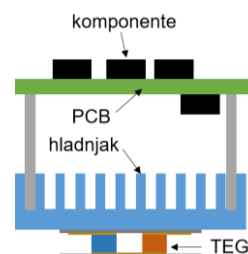


Slika 4. Čvor bežične senzorske mreže napajan termoelektričnim generatorom, adaptacija na osnovu [5].

U radu [9] je opisan bežični senzorski čvor za primenu u industriji, konkretno valjaonici. Štampana ploča sa TEG-om CP20151, dimenzija $(15 \times 15 \times 5)$ mm, kolom za upravljanje energijom LTC3108, temperaturnim senzorom, mikrokontrolerom i *Xbee* transmitterskim kolom nalazi se u sendvič strukturi između dve metalne ploče. Jedna metalna ploča se šrafi na kraj vruće cevi, a druga metalna ploča je okrenuta ka ambijentu, tako da je temperaturna razlika na krajevima TEG-a 7.5°C . Sistem ne poseduje dodatni hladnjak. Pokazano je da od trenutka postavljanja sistema na izvor toplote, vreme za prikupljanje energije potrebne za ispravan rad mikrokontrolera je 3 minuta. Maksimalna dobijena snaga je 0.14 mW i koristi se za analogno-digitalnu konverziju podataka sa senzora, buđenje transmitterskog kola i slanje paketa od 10 bajtova podataka.

Čvorovi WSN-a mogu biti projektovani i tako da TEG i hladnjak budu van štampane ploče. U radu [10] predstavljen je sistem za prikupljanje energije sa kolom za upravljanje energijom namenjen za napajanje čvora bežične senzorske mreže u zatvorenom prostoru. Sistem se može montirati na vruće cevi instalacija za grejanje i toplu vodu, i ima dimenzije $(5 \times 6 \times 7)$ cm. Iskorišćen je TEG prilagođenog dizajna, dimenzija $(50 \times 50 \times 3.4)$ mm, sa 127 termoelektričnih parova. Preko TEG-a postavljen je hladnjak, a u hladnjak su montirani metalni nosači štampane ploče (slika 5). Na PCB-u se nalazi kolo DC/DC konvertora

i to: *charge-pump* kolo ultraniskog startnog napona S882Z-18, *boost* prekidačko konvertersko kolo TPS61020 i superkondenzatori za skladištenje energije. Eksperimentalno je pokazano da će TEG generisati 250 mV, koliko je potrebno za početak rada *charge-pump* kola, pri temperaturi vruće strane od 60°C . Kada ovo kolo generiše napon od 0.9 V, uključuje se *boost* kolo koje na izlazu daje napon potreban za napajanje čvora WSN-a od 3.3 V. Maksimalna izlazna snaga kola za upravljanje energijom je 1.1 mW.



Slika 5. Sistem za napajanje čvora senzorske mreže pomoću TEG-a, adaptacija na osnovu [10].

Autori u [11] nisu opisali izgled čvora WSN-a, ali su vršili eksperiment kada se koristi TEG TMG-127-1.0-2.5 sa 127 termoelektričnih parova i hladnjak u kombinaciji sa kolom za upravljanje energijom TPS60303. Pokazali su da je potrebna temperaturna razlika od 34°C da bi se startovalo kolo za podizanje napona. Napon TEG-a pri kome počinje da radi *boost* kolo je 900 mV, a pri tom se na izlazu tog kola dobija 3 V. Kolo čvora WSN-a za očitavanje podataka sa pasivnog senzora, obradu tih podataka i slanje putem RF modula troši 31.8 mW. Prototip čvora WSN-a prikazan je u [12]; koristi se TEG od 127 termoelektričnih parova sa dvostepenim kolom DC/DC konvertora, gde je omogućeno startovanje sistema pri ulaznim naponima već od 20 mV. Sistem može da se startuje sa $84 \mu\text{W}$ ulazne snage pri temperaturnoj razlici od 2.6°C , za vreme od 196 s.

Prototip senzorskog sistema zasnovanog na TEG-u TEC1-12706, za primene u spoljašnjoj sredini dat je u radu [13]. Sistem koristi Sunčevu energiju koju preko sočiva koncentriše u metalnu pločicu koja je spojena sa toplom stranom TEG-a. S druge strane TEG-a je hladnjak. Za upravljanje električnom energijom koristi se kolo LTC3105, koje pri ulaznom naponu od 250 mV generiše izlazni napon dovoljan za napajanje senzorskog sistema zasnovanog na Arduinu. Sistem poseduje i kolo za punjenje baterije i bateriju. Prototip je testiran tokom celog dana, kada su različiti intenziteti sunčevog zračenja. Eksperimentalno je utvrđeno da je srednja dobijena energija 133.6 mW.

Čvor WSN-a za merenje temperature u industriji, na nivou prototipa, opisan je u [14]. Sistem sadrži dva TEG-a TGM287-1.0-1.3, koji su preko bakarnih pločica svojom vrućom stranom postavljeni na topli zid. Bakarne pločice postavljene su i između hladne strane TEG-a i bakarnog hladnjaka. Oko TEG-ova korišćen je *aerogel* da spreči prenos toplote na hladnu stranu TEG-ova. TEG-ovi su provodnicima spojeni sa ostatkom sistema: kolom za

upravljanje električnom energijom, superkondenzatorom, mikrokontrolerskim sistemom. Autori su pokazali da se pri temperaturnoj razlici od 14°C generiše dovoljan napon TEG-ova za pokretanje kola LTC3108. Dobijena ukupna snaga pri toj temperaturnoj razlici je 13.1 mW.

Još jedan industrijski čvor WSN-a prezentovan je u [15]. Sistem zasnovan na akcelometru se koristi za detekciju istrošenosti oštrice testere. TEG QC-32-0.6-1.2 se montira na mašinu gde postoji disipacija toplote, dok se senzorski deo nalazi na nosaču oštrice. WSN čvor može da isporuči 1.8 mW snage pri temperaturnoj razlici od 19°C, pri čemu je potrebno vreme hladnog starta 58 s od trenutka uključenja sistema testere.

U radu [16] predstavljen je čvor WSN-a baziran na TEG-u kod koga se energija dobija iz vrlo malih temperaturnih razlika na međupovršini zemlja-vazduh. TEG se nalazi na nosaču koji se pobija u zemlju, a ostatak sistema je odvojen na posebnoj štampanoj ploči. Autori su pokazali da je maksimalna dobijena energija tokom dana 27.2 mW, a 6.3 mW tokom noći.

Senzorski sistem sa 3 TEG-a GM200-71-14-16 predstavljen je u radu [17]. TEG-ovi su postavljeni jedan preko drugog, pri čemu se na poslednjem nalazi hladnjak. Sistem se postavlja na ruku korisnika i koristi temperaturnu

razliku između kože korisnika i ambijenta za dobijanje električne energije. Napon sa TEG-ova vodi se u dvostepeni sistem za upravljanje energijom. Posle 65 s od trenutka postavljanja sistema TEG-ova može se dobiti 1.62 mW snage pri temperaturnoj razlici od 5°C.

Autori rada [18] prezentovali su prototip senzorskog čvora koji se napaja pomoću TEG-a sa 127 termoelektričnih parova od silicijuma. Pored posebnog dizajna TEG-a autori su predstavili i sopstveni dizajn integrisanog kola za upravljanje energijom. U radu je pokazano da je napon hladnog starta kola 200 mV, te da se dobija izlazna snaga od 34 mW pri temperaturnoj razlici od 70°C.

Razvijeni su i hibridni čvorovi WSN-a koji sadrže baterijsko i *energy harvesting* napajanje, kao što je prezentovano u [19]. Ovaj sistem se sastoji iz 4 redno povezana TEG-a koji se nalaze između aluminijumske ploče preko koje se dovodi toplota i cevi kroz koju kruži rashladna tečnost. Cirkulacija rashladne tečnosti se vrši korišćenjem DC pumpe. Kolo za upravljanje električnom energijom i senzorski čvor su udaljeni od *energy harvesting* sistema. Opisani sistem pri temperaturnoj razlici od 23°C može da prikupi 450 mW, što je manje od potrošnje pumpe, te ovakav sistem služi samo za dopunu baterije.

TABELA I

UPOREDNI PREGLED KARAKTERISTIKA ČVOROVA BEŽIČNIH SENZORSKIH MREŽA KOJI KORISTE TERMoeLEKTRIČNI MODUL KAO IZVOR NAPAJANJA

Ref.	Opis kola	Dimenzije TEG-a (L×W×H) mm	Broj termoelektričnih parova TEG-a	Kolo za upravljanje električnom energijom (PWR MNG)	Minimalni napon za startovanje PWR MNG kola (mV)	Maksimalna snaga pri određenoj temperaturnoj razlici (mW)
[5]	Kompaktan čvor senzorske mreže	15×15×4.3	31	LTC3108	40	2.04 @ 25°C
[9]		15×15×5	31	LTC3108	50	0.14 @ 7.5°C
[15]	Distribuiran čvor senzorske mreže	8×8×2.6	32	LTC3107	20	1.80 @ 19°C
[10]	Sistem TEG-a i kola za upravljanje energijom	50×50×3.4	127	S8827-18 TPS61020	250	1.10 @ 40°C
[11]		30×30×4.3	127	TPS60303	900	31.8 @ 45°C
[13]		40×40×4	127	LTC3105	250	133.6
[14]	Prototip čvora senzorske mreže	40×40×3.6	2×287	LTC3108	50	13.1 @ 14°C
[17]		30×30×3.4	3×71	LTC3108 BQ25504	20	1.62 @ 5°C

IV. ZAKLJUČAK

U tabeli I dat je uporedni pregled najbitnijih karakteristika različitih čvorova bežičnih senzorskih mreža sa TEG-om kao izvorom napajanja. Sistemi koji su realizovani kao kompaktni uređaji koriste TEG-ove manjih površina sa manjim brojem termoelektričnih parova, za razliku od sistema koji su urađeni na nivou prototipa. Najzastupljenije kolo za podizanje napona i upravljanje električnom energijom je LTC3108 kompanije Analog Devices. Ovom kolu potreban je ulazni napon od 20-50 mV

da bi startovalo i na izlazu dalo maksimalno 5 V. U zavisnosti od primene čvora WSN-a i dostupne temperaturne razlike, maksimalne dobijene snage su pretežno reda nekoliko mW.

ZAHVALNICA

Ovaj rad je realizovan u okviru projekta finansiranog od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije (ev. br. 451-03-9/2021-14/200102).

LITERATURA

- [1] M. Nikitin, Thermoelectrics for Power Generation - A Look at Trends in the Technology, *IntechOpen*, 2016.
- [2] Jose Pedro de Matos, Wireless Sensor Networks Running on Harvested Energy, Ph.D Thesis, University of Coimbra, 2015.
- [3] Tržište termoelektričnih modula. Dostupno na: <https://www.kbvresearch.com/thermoelectric-modules-market/>
- [4] V. Karthikeyan et al. "Wearable and flexible thin film thermoelectric module for multi-scale energy harvesting", *Journal of Power Sources*, Vol. 455, 2020.
- [5] A. Prijić et al, "Thermal Energy Harvesting Wireless Sensor Node in Aluminum Core PCB Technology", *IEEE Sensors Journal*, vol. 15, no. 1, 2015, pp. 337-345.
- [6] Z. Prijić et al, "Design Techniques for Wireless Sensor Network Nodes Powered by Ambient Energy Harvesting", in *Proc. 2019 IEEE 31st International Conference on Microelectronics (MIEL)*, 2019, pp. 37-44.
- [7] D. Milić et al, "Characterization of commercial thermoelectric modules for application in energy harvesting wireless sensor nodes", *Applied Thermal Engineering*, vol. 121, 2017, pp. 74-82.
- [8] M. Marjanović et al, "A Transient Modeling of the Thermoelectric Generators for Application in Wireless Sensor Network Nodes", *Electronics*, vol. 9, no. 6, p. 1015, 2020.
- [9] J. M. Lopera et al, "Practical Issues in the Design of Wireless Sensors Supplied by Energy Harvesting Thermoelectric Generators", *IEEE Transactions on Industry Applications*, vol. 55, no. 1, 2019, pp. 996-1005.
- [10] W. Wang et al, "Thermoelectric Energy Harvesting for Building Energy Management Wireless Sensor Networks", *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.
- [11] S. Dalola et al, "Characterization of Thermoelectric Modules for Powering Autonomous Sensors", *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 1, 2009, pp. 99-107.
- [12] M. Guan et al, "Design and experimental investigation of a low-voltage thermoelectric energy harvesting system for wireless sensor nodes", *Energy Conversion and Management*, vol. 138, 2017, pp. 30-37.
- [13] D. Charris et al, "A Portable Thermoelectric Energy Harvesting Unit to Power Up Outdoor Sensors and Devices", in *Proc. IEEE Sensors Applications Symposium (SAS)*, 2019, pp. 1-6.
- [14] L. Hou et al, "Thermal Energy Harvesting WSNs Node for Temperature Monitoring in IIoT", *IEEE Access*, vol. 6, 2018, pp. 35243-35249.
- [15] M. Magno et al, "SmarTEG: An Autonomous Wireless Sensor Node for High Accuracy Accelerometer-Based Monitoring", *Sensors*, vol. 19, no. 12, 2019, p. 2747.
- [16] L. Sigrist et al, "Thermoelectric Energy Harvesting From Gradients in the Earth Surface", *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, 2020, pp. 9460-9470.
- [17] C. Xia et al, "Human Body Heat Based Thermoelectric Harvester with Ultra-Low Input Power Management System for Wireless Sensors Powering", *Energies*, vol. 12, no. 20, 2019, p. 3942.
- [18] J. Im et al, "Self-Powered Autonomous Wireless Sensor Node by Using Silicon-Based 3D Thermoelectric Energy Generator for Environmental Monitoring Application", *Energies*, vol. 13, no. 3: 674, 2020.
- [19] G. Verma, V. Sharma, "A Novel Thermoelectric Energy Harvester for Wireless Sensor Network Application", *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, 2019, pp. 3530-3538.

Task Prompter

Lazar Becić

Abstract – Project named Task Prompter is designed to help people remember various tasks and to alert them when a deadline for that task is close so as to urge them to complete it. It is a site made with JavaScript, HTML and CSS.

I. INTRODUCTION

In this day and age, people have numerous distractions that make it incredibly easy for us to forget our responsibilities and that can lead to some serious consequences. The author's intention is to reduce the chances of this problem happening by making a project called Task prompter.

II. THE PRINCIPLE OF WORK AND IMPLEMENTATION

Upon opening the Task prompter the user is to input a task and a time frame relating to it in their respective spaces (Fig. 1). Clicking the button “Add +” places that task in a newly constructed table. Apart from the task's name and the inputted date, there are also options to remove and rearrange the selected task. On the right-hand side of those options is the time remaining until the task's deadline. When a task is close to its deadline it will be highlighted and a sound will be played.

Fig. 1. Input appearance.

The code is separated into 3 parts: declaration of variables and input; the creation of a table with all of its contents and

L. Becić is the first-year student of the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia. E-mail: lazarbecic1531@gmail.com

the specification of all of its functions.

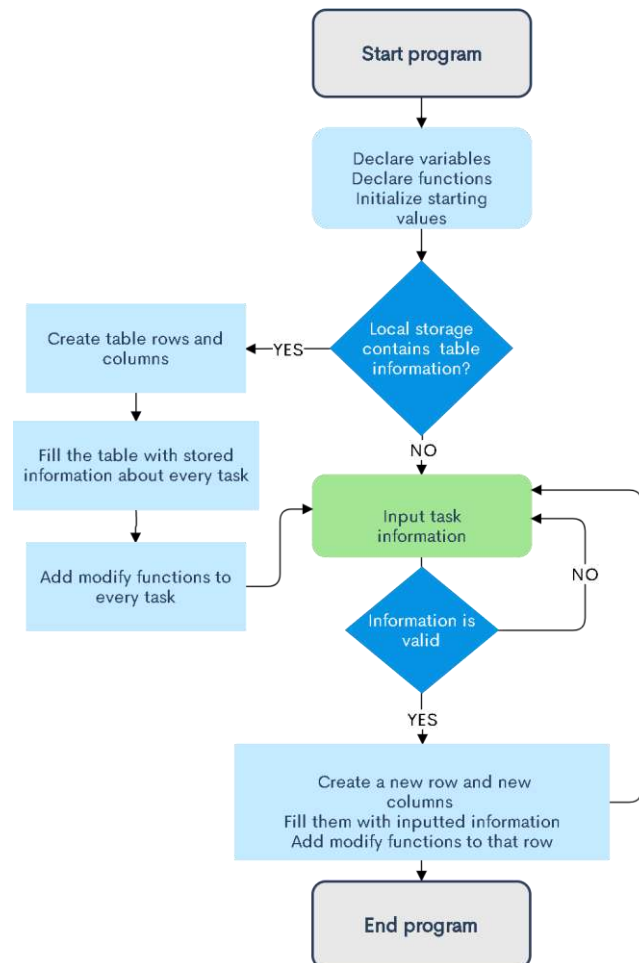


Fig. 2. Main program algorithm.

As can be seen on Fig. 2, the program consists of multiple functions that either create, change or delete data located in the table. Some of the most important functions are:

Function Time() – calculates time remaining until a tasks deadline [1].

Function OnDelRow() – deletes selected row.

Function create_row() (Fig. 3.) – creates a new row along with columns related to it and gives them appropriate information [2].

```
function_create_row(task, date){
  let mytable = document.getElementById('mytable').getElementsByName('tbody')[0];
  let newRow = mytable.insertRow(mytable.rows.length);

  let newCell = newRow.insertCell(0);
  let text = document.createTextNode(task);
  newCell.appendChild(text);

  newCell = newRow.insertCell(1);
  text = document.createTextNode(date);
  newCell.appendChild(text);
  newCell.className = "classDates";

  newCell = newRow.insertCell(2);
  newCell.innerHTML = '<button type="button" class="button-on-30 deleteBtn">delete</button>';

  newCell = newRow.insertCell(3);
  newCell.innerHTML = '<input type="number" name = "switch_number" id="switch_number" style="font-size: 18px; display:inline-block; width: 80px;" value="" placeholder = "position">';

  newCell = newRow.insertCell(4);
  newCell.innerHTML = '<button type="button" class="button-on-30 SwitchBtn">place</button>';

  newCell = newRow.insertCell(5);
  newCell.innerHTML = Time(date);
}
```

Fig. 3. Code of the function create_row().

Function Sort() – sorts all tasks according to the deadline in ascending order.

Function remember() – sends valid input to local storage.

Function fill() – creates a table with the information stored in the local storage along with function create_row.

III. THE RESULT

Task prompter is simple to use and will alert the user of the upcoming deadlines as long as the program is being used. The existence of this program will not only give a reminder of the task's deadline but also remind the user about the task's existence, and of their upcoming deadlines as long as the program is being used. Then seeing all their incomplete tasks will prompt them to dedicate some of their time to complete them before the ever closer deadline (Fig. 4.).

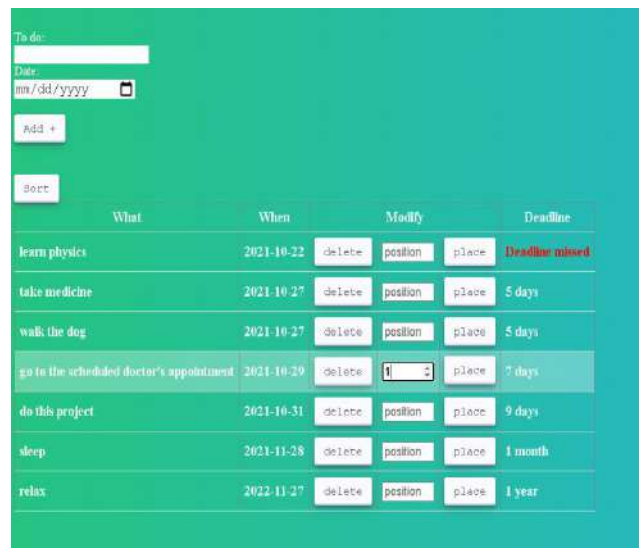


Fig. 4. A picture of the finished site.

IV. CONCLUSION

This product can be improved further by it having the ability to be downloaded onto smartwatches or phones so that the user can keep track of their respected tasks even while being away from their computer which will, in turn, make their chances of forgetting some of their tasks slim to none. Another improvement that can be done is to make the program usable while being offline. That would allow the user to keep track of their tasks while not having internet access. Creating a new note on a phone for every task would keep the user from forgetting their tasks, but for that to happen he needs to check notes regularly and there would be no warning a certain amount of time before the task's deadline.

ACKNOWLEDGEMENT

The author would like to thank Milan Stojanović who helped with the realization of this project as well as assistant professor Emilija Živanović who initiated this project.

REFERENCES

- [1] https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Timeouts_and_intervals
- [2] Marijn Haverbeke "Eloquent JavaScript", third edition, 2018.

Grafen aluminijum-jonske baterije

Dejan Anđelković

Sadržaj- U ovom radu je opisan princip rada grafen aluminijum-jonskih baterija kao i njihove prednosti i mane u odnosu na litijum-jonske baterije

I. UVOD

Litijum-jonske baterije su postale najčešće korišćena tehnologija za skladištenje energije u modernom društvu. Njihova primena i njihov procenat tržišta baterija raste iz godine u godinu i ne pokazuje znake usporavanja. Ali, litijum-jonske baterije su daleko od savršenih. Njihov kapacitet je ograničen i već duže vreme predstavljaju glavni faktor ograničenja portabilnih uređaja. Razvoj novog tipa baterije većeg kapaciteta i bržeg punjenja je postao glavni cilj svih istraživača u svetu skladištenja energije.

Jedan od obećavajućih koncepta nove tehnologije baterija je aluminijum-jonska baterija.

II. PRINCIP RADA

A. Istorija litijum-jonskih baterija

Brzo punjenje jedan je od ključnih faktora baterije portabilnih elektronskih uređaja i električnih vozila. Za platforme energetskog skladištenja koje se oslanjaju na reverzibilne procese oksido-redukcije, proces punjenja je već prešao na red veličine minuta. Tipičan primer ovakvog nivoa unapređenja je prisutan i kod aluminijum-jonskih baterija.

U zadnjih par godina, aluminijum-jonske baterije su stekle reputaciju zbog svojih neverovatnih brzina kako punjenja ali isto tako i pražnjenja. Formiranje elektrode od čistog aluminijuma pruža ozbiljne prednosti kao što su niska cena, nezapaljivost ali i izuzetno visok kapacitet skladištenja energije. Takođe, stabilni aluminijumski elektroda-elektrolit interfejs eliminiše kompleksnost interfaznog sloja koji je često prisutan u litijum-jonskim sistemima. Kao rezultat, slobodni joni aluminijuma se ređe vezuju sa katodom pa performanse aluminijum-jonske baterije u teoriji ne opadaju čak ni nakon 10 000 ciklusa punjenja i pražnjenja [1].

B. Princip rada aluminijum-jonskih baterija

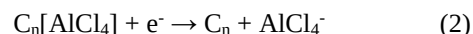
Princip rada aluminijum-jonskih baterija je u svojoj osnovi vrlo sličan litijum-jonskim baterijama. To je tip punjivih baterija u kojima joni aluminijuma pružaju energiju

Dejan Anđelković je student treće godine doktorskih akademskih studija na Elektronskom Fakultetu u Nišu. E-mail adresa: dejan.andjelkovic@elfak.rs

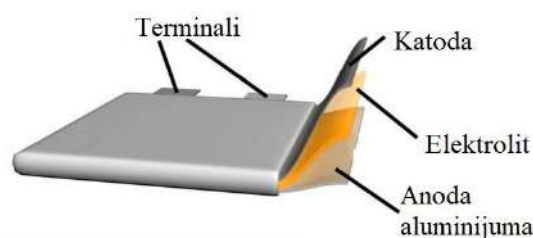
kretanjem od pozitivne elektrode baterije, anode, ka negativnoj elektrodi, katodi. Pri punjenju, joni aluminijuma vraćaju se negativnoj elektrodi, i mogu razmeniti do tri elektrona po jonu. To znači da kaliranje jednog Al_3^+ jona je ekvivalentno tri Li^+ jona u konvencionalnim interkalarnim katodama. S obzirom da je jonski radijus Al_3^+ (0.54 Å) sličan Li^+ (0.76 Å), znatno viši modeli elektrona i Al_3^+ jona mogu biti prihvaćeni katodom bez značajne pulverizacije. Trovalentni nosilac naelektrisanja, Al_3^+ , je istovremeno i prednost i mana ovakvog tipa baterije. Dok je prenos tri jedinice naelektrisanja jednim jonom prednost i dozvoljava znatno veći energetski kapacitet, elektrostatička interkalacija korišćenih materijala sa trovalentnim katjonom je suviše snažna za jasno definisani elektrohemijski rezultat. Polureakcija anode [2] je izražena jednačinom broj 1:



Polureakcija katode [2] je izražena jednačinom broj 2:



Inertnost aluminijuma, njegovo jednostavno skladištenje u ambijentalnom okruženju smatraju se bitnim prednostima za proizvodnju ovakvog tipa baterije. Dodatno, aluminijum ima veću gustinu ($2,7 \text{ g cm}^{-3}$ pri temperature od 25°C) od litijuma, kalijuma, magnezijuma, kalcijuma i cinka kao i sposobnost da razmeni tri elektrona. To dalje dovodi do veće količine energije skladištene po jedinici zapremine u odnosu na baterije bazirane na prethodno navedenim metalima. Zbog toga, očekuje se da komercijalne aluminijum-jonske baterije budu još manjih dimenzija.



Slika 1. Grafički prikaz aluminijum-jonske baterije [1].

C. Problemi aluminijum-jonskih baterija

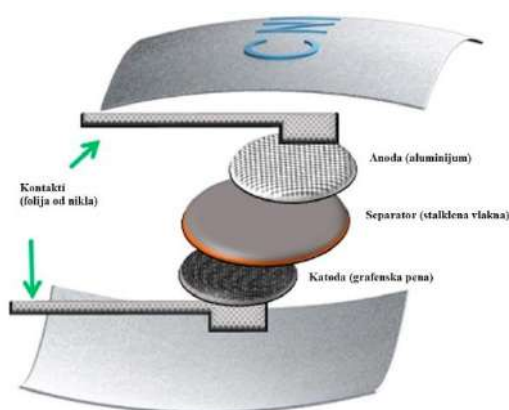
U teoriji, aluminijum-jonske baterije svakako deluju primamljivo, ali rani prototipi ovih baterija su ukazali na ozbiljne, fundamentalne probleme. Izlazni napon je bio vrlo nizak, u određenim slučajevima 0,55 V, ali je bio evidentan

i problem katode. Više različitih materijala su probani za izradu katode i svi su se pokazali nepouzdanim. Neki su se pri pražnjenju baterije raspali a drugi su pokazali veliki nivo stresa, što je dovodilo do naglog pada kapaciteta već u prvih 100 ciklusa punjenja i pražnjenja.

III. ULOGA GRAFENA

A. Katode od grafena

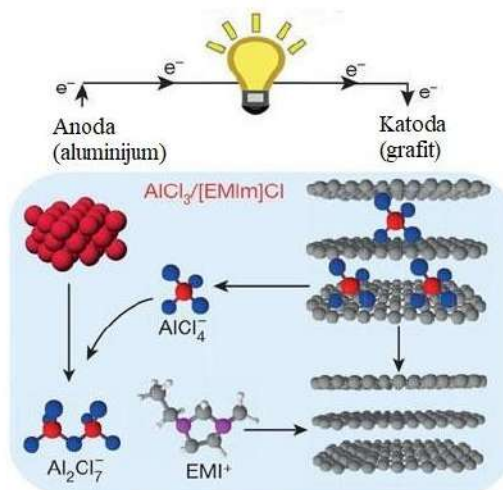
Nedavna istraživanja su otkrila grafit kao jedinu validnu opciju za materijal katode aluminijum-jonskih baterija. Dok se baterija ne prazni, joni aluminijum hlorida ($AlCl_4^-$) se skladište u katodi. Kada baterija uđe u stanje pražnjenja, joni migriraju ka anodi gde reaguju sa aluminijumom i formiraju $Al_2Cl_7^-$. Analizom katode utvrđeno je da se anjoni $AlCl_4^-$ reverezibilno interkaliraju između grafitnih slojeva.



Slika 2. Grafički prikaz grafitne aluminijum-jonske baterije [4].

Katode prirodnog grafita imaju dva problema. Prvo, maksimalni napon baterije i brzina pražnjenja su umanjeni difuzijom $AlCl_4^-$ jona između atomskih slojeva, što dalje dovodi do drugog problema a to je intenzivno širenje grafita. Oba problema su rešena deponovanjem grafena na nikl u kiselom okruženju, čime se postiže ugljenična struktura sa velikim atomskim razmacima koja dozvoljava ulaz i izlaz jona $AlCl_4^-$ rapidno bez ikakve načinjene štete, kao što se može videti na slici broj 2.

Katoda od grafena je sačinjena od trodimenzijonalne mreže grafitnih slojeva kako bi se dalje unapredio kapacitet baterije. Elektrolit korišćen u ovom slučaju je organskog tipa i jednostavan je za pripremu: mešanjem imidazolijum hlorida (EMI^+Cl^-) čvrstog stanja sa bezvodnim prahom $AlCl_3$ daje kao proizvod jonsku tečnost, kao što se može videti na slici broj 3. Zbog takvog tipa jonske tečnosti, 3 glavna jona su primećena u radu ovakve baterije: aluminijumski mono-kompleks ($AlCl_4^-$), aluminijumski duo-kompleks ($Al_2Cl_7^-$) i organski katjon (EMI^+). Zbog prisustva ovakvog elektrolita, aluminijumska elektroda postaje anoda dok elektroda grafita/grafena postaje katoda.



Slika 3. Grafički prikaz elektrohemijskih procesa aluminijum-jonske baterije sa grafitnom katodom [3].

Rezultujuće elektrohemijsko ponašanje je sledeće: elektroni sa anode će se vezati sa aluminijumskim duo-kompleksom i samim tim će ga redukovati na dva aluminijum mono-kompleksa, a nuspojava reakcije će biti svež atom aluminijuma deponovan na anodi. Na strani katode neće biti novih proizvoda, već će aluminijum mono-kompleks biti apsorbovan površinama ugljenika [3].

B. Performanse grafen aluminijum-jonske baterije

Prve praktične izrade ovakve baterije pokazale su vrlo obećavajuće performanse. Napon pražnjenja baterije bio je iznad 2 V a energetski kapacitet je bio višestruko veći od litijum-jonskih baterija. Dalja testiranja su pokazala da je kapacitet baterije opao manje od 5% nakon 7000 ciklusa punjenja i pražnjenja [2], što je znatno bolja retencija kapaciteta u odnosu na litijum-jonske baterije. Najbitniji napredak je vreme potrebno za kompletno punjenje baterije. Prosečno vreme je bilo 100 puta brže od litijum-jonskih baterija, manje od 60 sekundi. Kako je aluminijum inertan, baterija može da se savije ili da se „preklopi“ bez ikakvih posledica. Baterija je čak nastavila sa radom kada je bila probušena bušilicom u toku procesa pražnjenja.

IV. ZAKLJUČAK

Grafen aluminijum-jonske baterije svakako deluju kao obećavajuće unapređenje litijum-jonskih baterija. Efektivno trostruki kapacitet uz 100 puta brži proces punjenja pruža svetlu budućnost svetlu baterija, a samim tim i velikom delu industrije kako portabilnih elektronskih uređaja, tako i elektronskih vozila.

Početak njihove komercijalne primene se već očekuje, ali je trenutno aktuelna pandemija usporila proces razvoja i proizvodnje čitave industrije, pa samim tim i uvod grafen aluminijum-jonskih baterija.

ZAHVALNICA

Autor se zahvaljuje prof. dr Danijelu Dankoviću na mentorstvu i podršci, kao i ostalim članovima Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu.

LITERATURA

- [1] Aluminium-Ion Battery, <https://graphenemg.com/energy-storage-solutions/aluminum-ion-battery/>
- [2] Preeti Bhauriyal, Arup Mahata, Biswarup Pathak, "The staging mechanism of AlCl₄ intercalation in a graphite electrode for an aluminium-ion battery", Discipline of Chemistry, Indian Institute of Technology (IIT) Indore, Indore, DOI: 10.1039/C7CP00453B
- [3] Shen, X., Sun, T., Yang, L., "Ultra-fast charging in aluminum-ion batteries: electric double layers on active anode", Nat Commun 12, 820 (2021), DOI: 10.1038/s41467-021-21108-4
- [4] Anthony S. Childress, Prakash Parajuli, Jingyi Zhu, Ramakrishna Podila, Apparao M. Rao, "A Raman spectroscopic study of graphene cathodes in high-performance aluminum ion batteries", Clemson Nanomaterials Institute, Dept. of Physics and Astronomy, DOI: 10.1016/j.nanoen.2017.06.038

LIMENKO-solarni kolektor od iskorišćenih limenki

Rastko Petrović, Tatjana Živadinović, Kaja Gavrilović, Jana Simić

Sadržaj - Solarni kolektori se poslednjih nekoliko godina sve češće vide na krovovima raznih stambenih objekata. Glavni razlog ove pojave je smanjenje troškova grejanja (korišćenje obnovljivih izvora energije). Naš kolektor se razlikuje od običnih solarnih kolektora, upravo zato što je sam prijemnik napravljen od recikliranih limenki [1, 2].

I. UVOD

Solarni kolektor od limenki (može se videti na slici 1) kao dogrevač vazduha može se koristiti bilo gde: u kući, vikendici, garaži, a primenu može naći i u poljoprivredi. Njime se mogu dogrevati plastenici, prostorije u kojima borave životinje... Uštedom energije koju on omogućava investicija će se brzo isplatiti.

Ovim proizvodom smanjuju se troškovi grejanja, povećava stepen korišćenja obnovljivih izvora energije i smanjuje zagađenje životne sredine. Jednim kolektorom se može dogrevati prostor od 15-20 m², a efikasnost zavisi od izolacije prostora kao i količine sunčevog zračenja. S obzirom da Srbija ima prosečno 270 sunčanih dana godišnje, kolektor je idealan dodatak postojećem sistemu grejanja i štedi 20-35 % troškova grejanja [3].

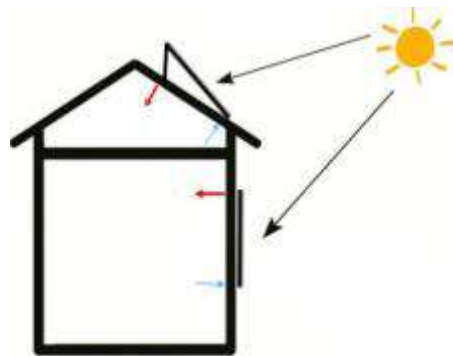


Slike 1. Solarni kolektori od limenki (Limenka).

Još jedna prednost „Limenka“ je jednostavna i brza montaža. Kako bi najbolje iskoristili sunčeve zrake potrebno je postaviti kolektor na južnu stranu objekta vertikalno na zid ili na krov pod uglom do 45° (može se

R. Petrović, T. Živadinović, K. Gavrilović i J. Simić su učenici druge godine Gimnazije „Bora Stanković“, Vožđa Karađorđa 27, 18000 Niš, Srbija, E-mail: ekolim.nis@gmail.com

videti na slici 2). Za montažu je potrebno izbušiti dva otvora u zidu/krovu kroz koje bi strujao vazduh. Za kolektor sa ventilatorom potreban je i dovod električne energije.



Slika 2. Montaža solarnog kolektora.

II. KONSTRUKCIJA I PRINCIP RADA SOLARNOG KOLEKTORA OD LIMENKI

Kolektor funkcioniše na principu kruženja vazduha. Hladan vazduh iz prostorije se kroz otvor na zidu sprovodi u donji deo kolektora, struji kroz limenke u kojima se pod uticajem sunčeve energije zagreva i penje ka otvoru u gornjem delu kolektora. Tako zagrejan, vazduh se vraća nazad u prostoriju (može se videti na slici 3).

Proizvedenu toplotu treba odmah upotrebiti za grejanje vazduha unutar stambenog prostora. Ukoliko nema sunca potrebno je prekinuti dovod vazduha u solarni kolektor, jer bi u suprotnom prostorija počela da se hladi.



Slika 3. Princip rada solarnog kolektora.

Za izradu solarnog kolektora dimenzija kolektora 0.7 x 1.4 m potrebno je 60 limenki (ili 10 x 6 limenki). Solarni apsorber izradjen je od limenki koje su ofarbane

mat-crnom bojom otpornom na visoke temperature. Limenke se buše i spajaju lepljenjem jedne na drugu kako bi kroz njih cirkulisao vazduh.

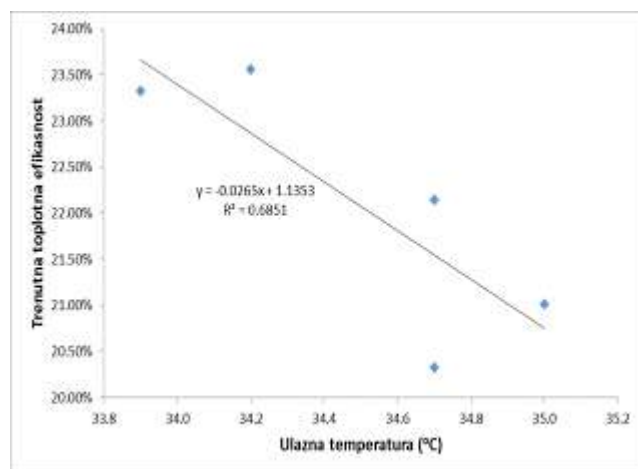
Kućiste za solarni kolektor napravljeno je od voodotporne šper-ploče debljine 20 mm, dok je njegova prednja strana od polikarbonatne ploče debljine 6 mm. Na zadnjoj strani kućista postavljen je stirodur kao izolacija i izbušena dva otvora za povezivanje sa prostorijom koju grejemo.

Postoje dve verzije ovog kolektora. Prva, pasivna verzija radi na principu prirodne cirkulacije toplog vazduha i ne koristi dodatnu energiju za rad. Druga verzija ima ugrađen ventilator koji povećava efikasnost grejanja, ali je neophodan električni priključak za rad.

III. ISPITIVANJE PRIJEMNIKA SUNČEVE SVETLOSTI

Efikasnost projektovanog solarnog kolektora ispitivan je u laboratoriji za termotehniku L10 pri Mašinskom fakultetu u Nišu. Došlo se do zaključka da pri uticaju ulazne temperature fluida na trenutnu toplotnu efikasnost vazdušnog prijemnika sunčeve energije za ukupnu količinu dozračene sunčeve energije dobijeno je $I_b = 850 \text{ W/m}^2$ i protok 0.0056 kg/s . Sa povećanjem ulazne temperature fluida, toplotna efikasnost se smanjuje.

Dobijeni rezultati (može se videti na slici 4) se objašnjavaju činjenicom, da ako se ulazna temperatura fluida povećava, a solarni fluks i temperature okoline ostaju konstantni, razlika izlazne i ulazne temperature fluida se smanjuje, i zbog toga biće manja korisna energija.



Slika 4. Rezultati ispitivanja.

Iz ovog se izvlači aksiom da toplotna efikasnost može biti i veća od 24 % ukoliko se obezbede uslovi većeg protoka fluida, kvalitetnijim transparentnim staklom (safirsko staklo) kao i boljom toplotnom izolacijom samog prijemnika [4, 5].

Na osnovu izvršenog pregleda i obavljenih merenja došlo se do sledećih zaključaka:

- konstrukcija PSE-prijemnik sunčeve energije je takva da u normalnim uslovima upotrebe nigde nisu nastale trajne deformacije ili bilo kakva oštećenja pojedinih elemenata;

- svi elementi PSE su tako spojeni da u toku rada nije došlo do deformacija, ugiba ili pregrevanja površina absorbera.

Na osnovu svega izrečenog može se zaključiti da ispitani PSE u svemu odgovara svojoj osnovnoj nameni, tako da se može koristiti kao uređaj za dogrevanje prostorija u prelaznom periodu, za sušenje voća i povrća, začinskog bilja i tome slično.

IV. ZAKLJUČAK

Prema osnovi ispitivanja LIMENKA na Mašinskom fakultetu u Nišu dokazana je njegova energetska efikasnost. LIMENKO je pouzdana investicija. Ovaj proizvod ima srazmernu cenu shodnu svojim mogućnostima. Proizvod iznosi 10.823,28 dinara. Ova cena obuhvata troškove potrebnog materijala za nabavku po važećim maloprodajnim cenama i izradu kolektora.

Uzimajući u obzir da je proizvod namenjen isključivo za dogrevanje prostorija, najveću primenu ima u poljoprivrednim objektima (sušare, voćnjaci, farme...). Razvoj proizvoda za korišćenje u poljoprivredi: dogrevanje plastenika u periodu decembar-mart. U ovom periodu sade se krompir, salata, kupus, paradajz, krastavac i paprika, kao i rasad tako da bi povećanje od 5°C bilo od velike pomoći kako ove kulture ne bi izmrzle. Standardni plastenici su dimenzija $8 \times 5 \text{ m}$ tako da bi za dogrevanje bilo potrebno 6 kolektora koji bi se rasporedili sa strane plastenika da prave što manju senku.

Jedan od kolektora trenutno zagreva prostorije Niškog ZooPlaneta, odnosno prostorije u kojima se nalaze živina i stoka [6].

Usavršavanjem našeg solarnog kolektora od limenki dobijamo veću primenu u grani energetike. Planiramo i razvoj drugih proizvoda koji za osnovu imaju solarni kolektor od limenki. Jedan od tih proizvoda je aparat za sušenje voća i povrća u kome bi izvor toplog vazduha za sušenje bio solarni kolektor [6, 7].

ZAHVALNOST

Zahvaljujemo se na podršci našem mentoru, Ani Veljković, prof. geografije u Gimnaziji "Bora Stanković".

LITERATURA

- [1] Solarni panel od recikliranih limenki, <https://nationalgeographic.rs/zuti-okvir-2021/a26936/niski-djaci-solarni-paneli-od-recikliranih-limenki.html>
- [2] Solarni panel od recikliranih limenki, <https://www1.wdr.de/radio/cosmo/programm/sendungen/radio-forum/region/panel-108.html>

- [3] Jozef Baruhović, Solarna fotonaponska postrojenja u Srbiji, http://www.planeta.rs/60/10_energija.htm#.YYmCIFXMKM8
- [4] Marko Mančić, Metodologija tehno-ekonomske optimizacije poligeneracionih sistema zasnovanih na korišćenju obnovljivih izvora energije, doktorska disertacija, Mašinski fakultet, Univerzitet u Nišu, 2018.
- [5] Saša Pavlović, Istraživanje optimalnih parametara solarnih paraboličnih koncentrišućih prijemnika toplote sa aspekta primene u sistemima poligeneracije, doktorska disertacija, Mašinski fakultet, Univerzitet u Nišu, 2017.
- [6] Omladinski klimatski forum, <https://www.wwfadria.org/?4619366/Mladi-u-Srbiji-predstavili-svoja-klimatska-reenja-u-okviru-omladinskog-projekta-WWF-a>
- [7] Srbija i životna sredina, <https://www.bbc.com/serbian/lat/srbija-57012094>

Sistem za kontrolu brzine DC motora korišćenjem ultrazvučnog senzora i ARM mikrokontrolera

Stefan Nikolić, Jovan Mijalković

Sadržaj – U ovom radu prikazan je sistem kontrole DC motora zasnovan na ARM mikrokontroleru. Brzina DC motora se menja proporcionalno u zavisnosti od izmerenog rastojanja. Za merenje rastojanja koristi se ultrazvučni senzor.

I. UVOD

Sistem sa ultrazvučnim senzorom i DC motorom je realizovan na Nucleo razvojnom okruženju [1]. Ukoliko je rastojanje koje meri senzor manje od 100 cm, smanjuje se brzina motora. Brzina motora proporcionalno se smanjuje u zavisnosti od rastojanja. Ukoliko je rastojanje manje od 50 cm, motor treba da se zaustavi. Ukoliko se poveća rastojanje motor se ponovo pokreće. Za merenje rastojanja upotrebljen je ultrazvučni senzor HC-SR04 [2], dok je za kontrolu DC motora iskorišćen *half-H driver* L293D [3].

II. KARAKTERISTIKE KORIŠĆENIH KOMPONENATA

A. Ultrazvučni senzor HC-SR04

HC-SR04 je veoma precizan senzor za merenje razdaljine, greška merenja je oko 3 mm, dok mu je opseg merenja od 2 cm do 400 cm. Na slici 1 prikazan je izgled ultrazvučnog senzora.



Slika 1. HC-SR04 ultrazvučni senzor.

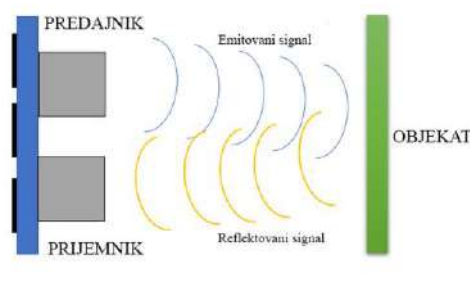
Od pinova za konekciju senzor sadrži Vcc pin za napajanje (5 V), *Trigger Pulse Input* pin, *Echo Pulse Output* pin, *Ground* pin. Da bi senzor izmerio rastojanje, potrebno je na *Trigger Pulse* pin dovesti kratkotrajni impuls u trajanju od maksimalno 10 μs. Kada se detektuje *trigger* impuls, senzor emituje 8 kratkih ciklusa ultrazvuka na frekvenciji od 40 kHz i postavlja *echo* pin na logičku jedinicu čime počinje merenje rastojanja. Kada se ultrazvuk odbije od objekta senzor ga detektuje (slika 2), *echo* pin se

S. Nikolić i J. Mijalković su studenti na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefan.s.nikolic@gmail.com, mijalkovic.jovan@gmail.com

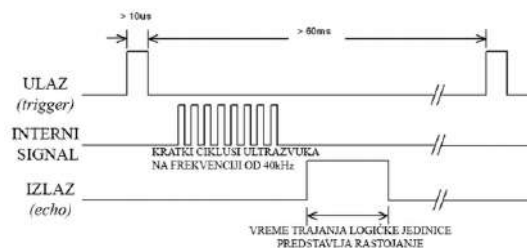
setuje na logičku nulu i time je merenje završeno. Slika 3 prikazuje dijagram prilikom merenja rastojanja. Za preciznost merenja vreme između dva *trigger* impulsa bi trebalo da bude veće od 60 ms. Da bi se izračunalo rastojanje koristi se formula 1:

$$d = t \cdot \frac{c}{2} \quad (1)$$

gde je d – rastojanje u cm, t – dužina trajanja logičke jedinice *echo* pina u μs, c – brzina zvuka koja iznosi 340 m/s=0.034 cm/μs. Uzima se $t/2$ jer je vreme t proteklo od trenutka kada je zvuk od predajnika putovao do objekta i nazad do senzora.



Slika 2. Ilustracija principa rada ultrazvučnog senzora.

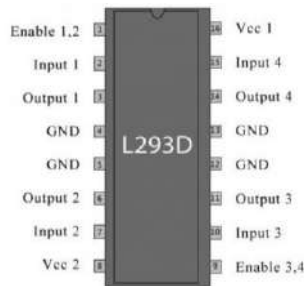


Slika 3. Dijagram merenja rastojanja.

B. Half-H driver L293D

Integrisano kolo L293D koristi se kao drajver induktivnih opterećenja kao što su releji, solenoidi, DC i bipolarni *stepper* motori. Svaki izlazni pin, unutar čipa, ima Darlingtonov tranzistor (struja do 600 mA). Raspored pinova kola L293D je prikazan na slici 4. Sadrži 4 ulaza i 4 izlaza, a rad drajvera se omogućava u paru (1 i 2, 3 i 4). Kada se dovede *enable* signal na pin broj 1 aktivira se par *input* 1 i *input* 2 (pinovi 2 i 7) koji predstavljaju ulaze, a

odgovarajući izlazi su pinovi 3 i 6, isto je za dozvolu rada drajvera 3 i 4 preko pina broj 9. U ovom slučaju ulazi su na pinovima 10 i 15, a izlazi na 11 i 14.



Slika 4. L293D sa pinovima.

C. DC motor

Elektromotori su uređaji koji pretvaraju električnu energiju u mehaničku. Elektromotori 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. Izgled DC motora je prikazan na slici 5.



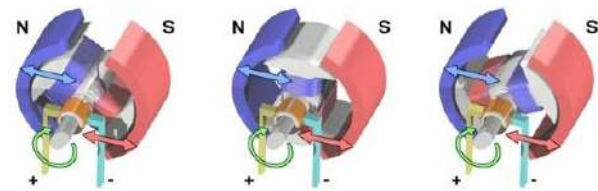
Slika 5. DC motor.

Prema konstrukciji motore možemo podeliti na motore sa četkicama (*brushed*), motore bez četkica (*brushless*) i step motore. Za realizaciju projekta iskorišćen je motor sa četkicama.

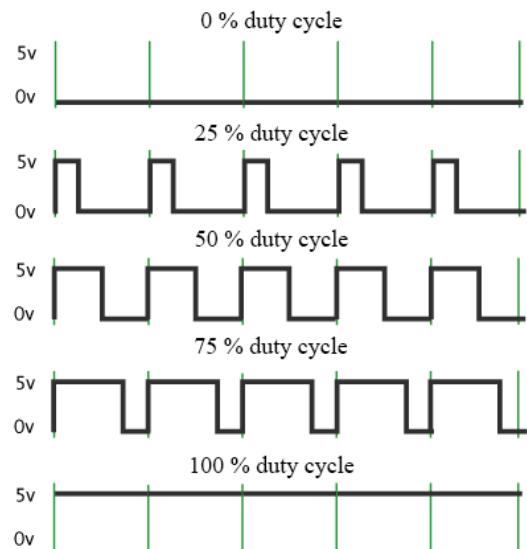
Motor sa četkicama sastoji se od statora (stalni magneti), rotora, četkica i komutatora. Komutator je mehanički prekidač koji periodično obrće smer struje u motoru. Čini ga par međusobno izolovanih bakarnih kontakata polucilindričnog oblika pričvršćenih oko rotora. Četkice služe za zatvaranje strujnog kola između izvora preko komutatora. Kada se kalem napaja, magnetno polje se generiše oko armature. Leva strana armature je gurnuta od strane levog magneta i privučena prema desnom magnetu i to izaziva rotaciju. Kada armatura postaje horizontalno poravnata, komutator obrće smer struje kroz kalem obrtanjem magnetnog polja. Tako se proces ponavlja (slika 6).

Najjednostavniji način kontrole brzine DC motora je upotrebom širinsko-impulsne modulacije (eng. PWM – *pulse width modulation*). To je tehnika dobijanja analognih vrednosti pomoću digitalnih impulsa konstantne amplitude. PWM modul generiše periodičan signal čiji se faktor

ispune (eng. *duty cycle*) može kontrolisati od strane mikrokontrolera (slika 7).



Slika 6. Princip rada DC motora sa četkicama.



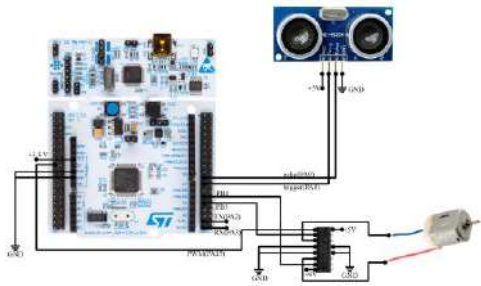
Slika 7. PWM za različite vrednosti faktora ispunje.

III. REALIZACIJA SISTEMA

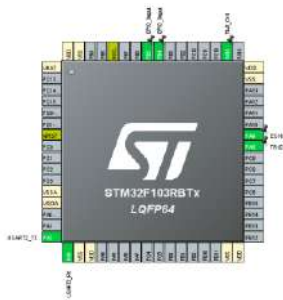
Blok šema realizovanog sistema prikazana je na slici 8. Za pisanje softvera mikrokontrolera korišćeni su STM32CubeMX [4], koji omogućava definisanje funkcionalnosti periferija mikrokontrolera, dok je za pisanje koda na programskom jeziku C korišćen Keil uVision5 [5].

Na slici 9 prikazani su definisani pinovi. Za realizaciju sistema potrebno je definisati izlazni *trigger* pin (PA8), *echo* pin (PA9) definisan kao ulazni i pin za PWM signal koji će kontrolisati brzinu DC motora. PWM signal definisan je korišćenjem tajmera mikrokontrolera. U ovom slučaju izabran je *timer 2 channel 1* (PA15). Smer rotacije motora biće kontrolisan signalima preko pinova PB4 i PB5. Pinovi PA2 i PA3 služe sa serijsku komunikaciju.

Kao što je na početku rečeno, ultrazvučni senzor počinje da meri rastojanje kada se na *trigger* pinu pojavi impuls od 10 μ s. Kako ovo nije moguće uraditi uz pomoć postojećih funkcija *delays*, jer je minimalno vreme 1 ms, potrebno je kreirati novu funkciju uz pomoć internog tajmera. Za ovo je upotrebljen *Timer 4*, a na slikama 10 i 11 je prikazano kreiranje posebne funkcije nazvane *usDelay*.



Slika 8. Blok šema sistema.



Slika 9. Definisani pinovi za realizaciju projekta.

```
/* USER CODE BEGIN PFP */
void usDelay(uint32_t uSec);
/* USER CODE END PFP */
```

Slika 10. Deklaracija funkcije *usDelay*.

```
/* USER CODE BEGIN 4 */
void usDelay(uint32_t uSec){
    if(uSec < 2) uSec = 2;
    TIM4->ARR = uSec - 1;
    TIM4->EGR = 1;
    TIM4->SR &= ~1;
    TIM4->CR1 |= 1;
    while((TIM4->SR & 0x0001) != 1);
    TIM4->SR &= ~(0x0001);
}
/* USER CODE END 4 */
```

Slika 11. Funkcija *usDelay*.

Promeljive koje se koriste su date na slici 12. Promeljiva *speedofSound* predstavlja brzinu zvuka podeljenu sa 2, odnosno $c/2$ iz jednačine 1, izražena u $\text{cm}/\mu\text{s}$. *Distance* služi za čuvanje informacije o izmerenom rastojanju, dok *duty* služi za kontrolisanje brzine motora. Promeljiva *numTicks* služi za merenje vremena za koje je *echo* pin postavljen u stanje logičke jedinice.

```
/* USER CODE BEGIN 0 */
//Brzina zvuka u cm/usec
const float brzinazvuka = 0.0343/2;
float rastojanje;
int duty;

char uartBuf[100];
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */
    uint32_t numTicks = 0;
    /* USER CODE END 1 */
```

Slika 12. Promeljive u kodu.

Pre početka glavnog dela koda potrebno je omogućiti generisanje PWM signala potrebnog za kontrolu motora i to je urađeno linijom koda prikazanoj na slici 13.

```
/* USER CODE BEGIN 2 */
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
/* USER CODE END 2 */
```

Slika 13. Linija koda za omogućavanje PWM signala.

Deo koda koji služi za slanje *trigger* impulsa i merenje trajanja *echo* signala prikazan je na slici 14.

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    //Trig pin se setuje na logicku nulu
    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);
    usDelay(3);

    //1. Slanje signala u trajanju od 10us na trig pin
    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_SET);
    usDelay(10);
    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);

    //2. Ceka se prednja ivica echo pina
    while(HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_RESET);

    //3. Meri se vreme trajanja logicke jedinice na echo pinu
    numTicks = 0;
    while(HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_SET)
    {
        numTicks++;
        usDelay(2); //2.8usec
    };
};
```

Slika 14. Deo koda za merenje *echo* signala.

Na početku se *trigger* setuje na logičku nulu, a ubrzo posle toga se šalje logička jedinica u trajanju od 10 μs korišćenjem prethodno kreirane funkcije *usDelay*. U trenutku kada se na *echo* pinu pojavi logička jedinica počinje merenje gde se promenljiva *numTicks* povećava za jedan na svakih 2 μs . Merenje se završava u trenutku kada signal padne na logičku nulu. Usled internog kašnjenja, 2 μs u realnom vremenu predstavljaju kašnjenje od oko 2.8 μs , pa je *numTicks* u formuli za računanje rastojanja pomnožen sa 2.8. Time određujemo trajanje logičke jedinice na *echo* pinu. Na slici 15 je prikazan drugi deo glavnog koda, odnosno deo u kome se računa rastojanje i menja brzina motora.

Rastojanje se računa uz pomoć jednačine 1, a *duty cycle* po jednačini 2:

$$D = (d - 50) \cdot 2 \quad (2)$$

gde je D – *duty cycle*, a d – rastojanje. Sistem je projektovan na sledeći način, ako je rastojanje veće od 100 cm *duty cycle* je 100%, odnosno motor se okreće maksimalnom brzinom, ako je rastojanje manje od 50 cm *duty cycle* je 0%, odnosno motor se ne okreće. Za rastojanja između 50 cm i 100 cm, brzina motora se menja proporcionalno sa rastojanjem, a *duty cycle* se računa po jednačini 2. Sistem je realizovan i testiran povezivanjem komponenta na protoploči i programiranjem mikrokontrolera na Nucleo razvojom okruženju.

```

//4. Racunanje rastojanja u cm
rastojanje = (numTicks + 0.0f)*2.8*brzinazvuka;

//5. Serijska komunikacija
sprintf(uartBuf, "Rastojanje (cm) = %.1f\r\n", rastojanje);
HAL_UART_Transmit(&huart2, (uint8_t *)uartBuf, strlen(uartBuf), 100);

HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
duty = (rastojanje - 50)*2;
sprintf(uartBuf, "Duty = %d\r\n", duty);
HAL_UART_Transmit(&huart2, (uint8_t *)uartBuf, strlen(uartBuf), 100);

if(rastojanje < 50)
{
    htim2.Instance->CCR1=0;
}
else if(rastojanje > 100)
{
    htim2.Instance->CCR1=100;
}
else
{
    htim2.Instance->CCR1=duty;
}
HAL_Delay(1000);
}
/* USER CODE END 3 */
}

```

Slika 15. Deo koda za računanje rastojanja i promenu brzine DC motora

ZAHVALNICA

Ovaj rad predstavlja projekat u okviru predmeta Komponente i kola snage, na IV godini OAS. Autori se zahvaljuju predmetnim profesorima i asistentima na pomoći i sugestijama u procesu izrade projekta.

LITERATURA

- [1] Razvojno okruženje STM32 Nucleo-64 sa STM32F103RB MCU, Dostupno na: https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-f103rb.html
- [2] Ultrazvučni senzor HC-SR04, Dostupno na: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [3] *Half-H driver* L293D, Dostupno na: <https://www.ti.com/lit/ds/symlink/l293.pdf>
- [4] STM32CubeMX, Dostupno na: <https://www.st.com/en/development-tools/stm32cubemx.html>
- [5] Keil -uVision5, Dostupno na: <https://www2.keil.com/mdk5/uvision/>

Booking Cancellation Prediction Relying on Graph Database in Neo4j

Nenad Petrović

Abstract – COVID-19 has caused huge losses in tourism industry due to travel restrictions and other measures. However, it is identified that machine learning has huge potential for negative impact reduction in this sector, providing the ability to bring decisions proactively, relying on predictions. In this paper, it is examined how the travel booking data can be leveraged for reservation cancellation prediction. As outcome, classification model implemented relying on Neo4j graph database is introduced and evaluated on realistic dataset.

I. INTRODUCTION

The ongoing COVID-19 pandemic has affected tourism sector significantly worldwide. Travel restrictions and other government measures applied as a response to novel circumstances, has caused negative consequences when it comes to tourism – from booking and flight cancellations to job loss [1].

However, state-of-art information technology has huge potential when it comes to tourism revival during the pandemic. Machine learning and artificial intelligence are identified among key-enablers of both safe tourism and pandemic's negative impact reduction on this sector [2].

In this paper, it is explored how the subclass of machine learning algorithms, known as classification methods, can be leveraged to predict booking cancellation based on traveler's data. Identifying the cancellations proactively can lead to better business and management decisions, mitigating the possible losses in travel industry [3]. For implementation, we rely on Neo4j graph database [4] and its Graph Data Science Library (GDS) [5] for classification. Finally, the proposed classification model created within Neo4j is trained and evaluated on realistic, publicly available dataset [6].

II. BACKGROUND

A. Neo4j

Neo4j is based on property graph database model. The *graph* itself consists of *nodes* which can be connected using *relationships* [7]. These nodes describe entities from a particular domain. On the other side, relationships define connection between the *source* and *target* nodes. Moreover, it is possible to assign properties (in form of

key-value pairs) to both the nodes and relationships, which are used to describe them in more details. Cypher [8], a declarative query language is used for efficient data retrieval in Neo4j.

Furthermore, since late 2020, Neo4j has become extendable with Graph Data Science Library [4, 9], which enables the usage of machine learning algorithms against the data stored within graph. It currently supports two machine learning tasks – classification and relationship discovery. Despite the limited set of machine learning-related capabilities, it was shown that Neo4j is superior to many other both graph- and non-graph solutions when it comes to these two tasks [9].

B. Classification in Neo4j

In this paper, the focus is on adoption of classification in Neo4j. In general, classification consists of predicting the right class label (output variable) to the observed sample from dataset (Neo4j node in this context), based on its features (input variables). However, in order to make new predictions, it is necessary to train model observing correctly labelled samples (train dataset). Once the training is completed, the model's performance can be evaluated using different metrics common for classification algorithms, such as accuracy and F1-score, which are supported in Neo4j:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$F - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (2)$$

In what follows, the meaning of the variables used for number of observations within the eq. 1 and 2 is given: TP – true positive; TN – true negative; FP – false positive; FN – false negative.

For training of classification model, Neo4j provides a method within Graph Data Science plugin – `gds.alpha.ml.nodeClassification.train` [10]. In Table I, the set of most relevant classification parameters for training, adjustable in Neo4j is given. Each call of this function can include multiple model configurations with different parameters (within `params` attribute), while the one with the best performance according to the selected metric is returned to user.

N. Petrović is PhD student and teaching assistant at the Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nenad.petrovic@elfak.ni.ac.rs.

TABLE I
NEO4J CLASSIFICATION ALGORITHM PARAMETERS

Algorithm parameter	Type	Description
modelName	String	Unique name of the classification model which is about to be created.
feature Properties	List of String	Names of node properties used as input variables for classification.
target Property	Integer	Class label property (output variable in classification) of a node.
holdoutFraction	Float [0-1]	Percentage of trainset reserved for esting purpose.
validationFolds	Integer	Number of train graph divisions used for model selection.
Metrics	F1_WEIGHTED F1_MACRO ACCURACY	Model evaluation method used for performance estimation.
Params	List of Map	List of various model configurations consisting of parameters from Table II.
randomSeed	Integer	Seed value for random number generator within training phase.

Moreover, for each of the model configurations, the set of parameters given in Table II can be customized [9].

TABLE II
CLASSIFICATION MODEL PARAMETERS

Model parameter	Type	Description
Penalty	String	Learning rate value of the model – how much it modifies the learned weights with new observations.
batchSize	Integer	Number of observations (nodes) per batch used for training within one iteration.
minEpochs	Integer	Minimum number of epochs (passes through the dataset) during training.
maxEpochs	Integer	Maximum number of training epochs.
patience	Integer	Maximum number of performed iterations without improvement of loss value without stopping.
tolerance	Float	Minimum value of acceptable loss value before stopping.

III. IMPLEMENTATION

In this section, the implementation of classification model for booking cancellation prediction relying on Neo4j will be described. Fig. 1 depicts the machine learning workflow for classification algorithm in Neo4j with corresponding procedures used within each step. Green rectangles denote the decisions and steps performed manually by user, while the blue ones stand for automatic steps executed within Neo4j.

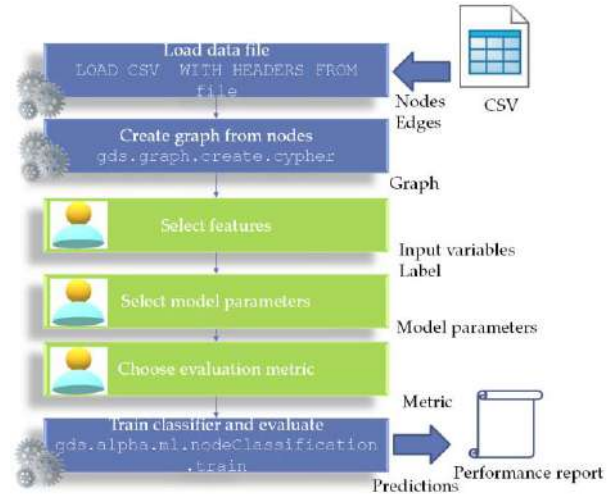


Fig. 1. Neo4j classification workflow.

First, user has to load data from CSV file into Neo4j using the LOAD CSV WITH HEADERS FROM file command as shown in Fig. 2. The dataset is imported as set of nodes with label “Booking” (each of them corresponding to distinct row in CSV file) with properties (column values for corresponding row) into Neo4j environment. In this command, file parameters denotes the CSV file path, while MERGE part is used to define node label and select the desired columns.

```
LOAD CSV WITH HEADERS FROM 'file:///hotel_bookings.csv' AS row
WITH row WHERE row.hotel IS NOT NULL
MERGE (b:Booking {hotel: row.hotel, is_canceled: row.is_canceled,
lead_time:row.lead_time,
stays_in_week_nights:row.stays_in_week_nights
...
previous_cancellations:row.previous_cancellations});
```

Fig. 2. Loading CSV file into Neo4j as set of nodes.

After that, it is needed to construct Neo4j graph based on this set of nodes, as classification algorithm can be executed only on this data structure in Neo4j. For this purpose, gds.graph.create.cypher method is used, as given in Fig 3. It has three obligatory parameters: 1) graph name 2) cypher query returning the desired nodes 3) cypher query that filters relationships between nodes. In this case, relationships between nodes are not leveraged, as classification relies on data analysis performed against node features only.

```
CALL gds.graph.create.cypher(
  'BookingGraph',
  'MATCH (n:Booking) WHERE n.lead_time IS NOT NULL RETURN id(n) as id,
  toInteger(n.is_canceled) as is_canceled, toInteger(n.lead_time) as
  lead_time, toInteger(n.previous_cancellations) as previous_cancellations,
  ...
  toInteger(n.stays_in_week_nights) as stays_in_week_nights',
  'MATCH (start:Booking)-[link]->(end:Booking)
  RETURN ID(link) as link, ID(start) as source, ID(end) as target'
)
```

Fig. 3. Creating graph from set of nodes in Neo4j.

Finally, the classifier is trained by invoking `gds.alpha.ml.nodeClassification.train` procedure from Graph Data Science Library, as illustrated in Fig. 4. Three main parts of this command can be identified:

- Algorithm input and output definition: specifying input graph name, model name, set of node features used as input for classification, target label and metric
- Model parameter definition: creating one or many model candidates with different parameter values (from Table II)
- Best model selection: According to the metric value specified as criterion, the candidate model will be compared and the one with the best possible performance metric value will be stored in Neo4j catalog with given modelName

```
CALL gds.alpha.ml.nodeClassification.train('BookingGraph', {
  modelName: 'cancel_prediction',
  featureProperties:
    ['lead_time', 'previous_cancellations',
     ...
     'stays_in_week_nights'],
  targetProperty: 'is_canceled',
  randomSeed: 2,
  holdoutFraction: 0.2,
  validationFolds: 5,
  metrics: [ 'ACCURACY'],
  params: [
    {
      penalty: 0.07,
      maxEpochs: 20,
      batchSize: 200
    },
    ...
    { penalty: 0.8 }
  ]
}) YIELD modelInfo
RETURN
{penalty: modelInfo.bestParameters.penalty} AS winningModel,
modelInfo.metrics.ACCURACY.outerTrain AS trainGraphScore,
modelInfo.metrics.ACCURACY.test AS testGraphScore
```

Fig. 4. Graph-based classification in Neo4j.

IV. EXPERIMENTS AND EVALUATION

For purpose of evaluation, a laptop equipped with Intel i7 7700HQ 2.8GHz quad-core processor with 16GB DDR4 RAM and M1 SSD disk was used. When it comes to data used within the experiments, the publicly available open dataset about hotel reservation cancellations worldwide from 2014 to 2017 [6] was used. From this dataset 10 502 records were imported into Neo4j graph – 80% for training and 20% for test. The main features from (among 32 total) used for reservation classification are summarized in Table III (the output variable row is gray).

TABLE III
HOTEL CANCELATION PREDICTION DATASET FEATURES

Feature	Type	Description
is_canceled	Boolean	The target (output) variable denoting whether the reservation was cancelled (1) or not (0).

hotel	String	Identifier of a hotel.
lead_time	Integer	Number of days between booking creation and arrival date.
arrival_date_year, month, week, day	Integer, String, Integer, Integer	Starting date of the reservation.
previous_cancellations	Integer	Number of previous cancellations by the same person.
previous_bookings_not-canceled	Integer	Number of past travels without cancellation .
adults	Integer	Number of adults within the reservation.
children	Integer	Number of children among the guests for given booking.
country	String	Travel destination country,
total_nights	Integer	The stay duration.
reserved_room_type	Char (A-H)	Category of the reserved room.
Is_repeated_guest	Boolean	Has the user booked the travel reservation to the same destination in past.

Furthermore, in Table IV, the results of evaluation for the implemented classification model are given. Various aspects were considered related to different components, such as execution time and the achieved model performance. The last column shows the environment parameters and/or conditions under whom these values were obtained. Considering the values, it can be noticed that Neo4j is quite fast for classification, performing all the steps except the CSV import in less than one second within our experiments. Compared to other solutions used for classification of booking reservation, such as RapidMiner case study in [3], Neo4j shows superior processing time. However, the processing time increases with number of features, nodes and candidate models. The best model shows slightly higher performance in case of training set, as expected.

TABLE IV
EVALUATION SUMMARY

Aspect	Step	Value	Conditions
Execution time [s]	Model import	49	0.015
	Graph creation	0.19	15 features
	Model training	0.27	2 model candidates
	Model training	0.91	8 model candidates
Model performance	Accuracy	0.83	Train set penalty=0.3 maxEpochs=30
	F1 score	0.88	
	Accuracy	0.79	Test set
	F1 score	0.85	

V. CONCLUSION AND FUTURE WORK

In this paper, a classification approach based on Neo4j graph was adopted for hotel booking cancellation prediction. According to the obtained results, this approach shows promising classification performance and fast execution times even in case of larger datasets.

In near future, it is planned to adopt the same model in context of post-COVID tourism by taking into account the data about new daily cases and the applied government measures, such as travel prohibitions, obligatory vaccination record certificates.

ACKNOWLEDGEMENT

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

- [1] P. J. Vindhya, R. R. Varma, "The Impact of COVID19 in Tourism Sector", Indian Journal of Applied Research vol. 11(10), pp. 61-63, 2021. <https://doi.org/10.36106/ijar/4313909>
- [2] N. Petrović, V. Roblek, M. Radenković, V. Nejković, N. Papachashvili, "Smart Technologies for the Post-COVID-19 Tourism Industry", 15th International Online Conference on Applied Electromagnetics - IIEC 2021, pp. 133-136.
- [3] N. Petrović, V. Roblek, N. Papachashvili, "Decision Support Based on Data Mining for Post COVID-19 Tourism Industry" SAUM 2021, pp. 1-4.
- [4] Neo4j [online]. Available on: <https://neo4j.com/>, last accessed: 01/11/2021.
- [5] The Neo4j Graph Data Science Library Manual v1.7 [online]. Available on: <https://neo4j.com/docs/graph-data-science/current/>, last accessed: 01/11/2021.
- [6] Hotel Booking Cancellation Prediction [online]. Available on: <https://www.kaggle.com/sanjana08/hotel-booking-cancellation-prediction>, last accessed: 01/11/2021.
- [7] Graph database concepts [online]. Available on: <https://neo4j.com/docs/getting-started/current/graphdb-concepts/>, last accessed: 01/11/2021.
- [8] Cypher Query Language [online]. Available on: <https://neo4j.com/developer/cypher/>, last accessed: 01/11/2021.
- [9] Announcing: Graph-Native Machine Learning in Neo4j! [online]. Available on: <https://neo4j.com/blog/announcing-graph-native-machine-learning-in-neo4j/>, last accessed: 01/11/2021.
- [10] Node Classification [online]. Available on: <https://neo4j.com/docs/graph-data-science/current/algorithms/ml-models/node-classification/>, last accessed: 01/11/2021.

Primena proširene realnosti u mobilnim aplikacijama

Vukašin Dobromirović

Sadržaj - U ovom radu je obrađena tehnologija Proširena realnost (Augmented Reality). Koristi se macOS operativni sistem, program Xcode kao i swift programski jezik. Korišćenje aplikacija zahteva posjedovanje iPhone mobilnog telefona.

I. UVOD

Kroz evoluciju tehnologije, ekspanzija virtualne i proširene stvarnosti eksponencijalno se povećala u poslednjoj deceniji. Iako je virtualna stvarnost napravila velike korake poslednjih godina, sistemi proširene stvarnosti u nadolazećoj fazi će poremetiti sektore širom sveta. Od zdravstvene zaštite preko inženjerstva do obrazovanja i svega između, proširena stvarnost ima sposobnost da oblikuje našu maštu, nadogradi naše kutije alata i izvrši revoluciju načina na koji pristupamo obrazovanju, preduzetništvu, a posebno kreativnosti. A kako proširena stvarnost postaje sve češća na svim uređajima – posebno pametnim telefonima – samo je pitanje vremena kada će se svačija stvarnost proširiti tehnologijom [1].

II. POKEMON3D

Pokemon3D je aplikacija u kojoj je uz AR (Augmented Reality) tehnologiju detaljnije obrazložen i oblik uživo praćenje. Ukratko, kada igrač želi da pozove svog pokemona potrebno je da otkrije lice svoje kartice i pojaviće se 3D model željenog pokemona (slika 1). Razvoj aplikacije je objašnjen u 3 faze.



Slika 1. Primena aplikacije.

Vukašin Dobromirović, student prve godine osnovnih akademskih studija na Elektronskom fakultetu, Univerziteta u Nišu, E-mail: vukasinveliki500@gmail.com

A. Ubacivanje slike kartice i prepoznavanje iste

Potrebno je slikati svoju karticu u visokoj rezoluciji radi boljeg prepoznavanja. Zatim sliku kartice ubaciti u folder Assets.xcassets i podesiti joj dimenzije u razmeri 1:1. Potrebno je uneti identične dimenzije kartice kao u realnosti. Po istom principu treba raditi za svaku narednu karticu. Nakon toga sledi kodiranje. Program koristiti deklaraciju ARImageTrackingConfiguration [2] (slika 2).

```
// Create a session configuration
let configuration = ARImageTrackingConfiguration()

if let imageToTrack =
    ARReferenceImage.referenceImages(inGroupNamed: "Pokemon
    Cards",
                                     bundle: Bundle.main) {
    configuration.trackingImages = imageToTrack
    configuration.maximumNumberOfTrackedImages = 3
}
```

Slika 2. Deo koda koji prikazuje deklaraciju ARImageTracking Configuration.

B. Konvertovaje 3D modela u USDZ format fajla

Svaki 3D model se danas može naći u mnogo formata. Među najpoznatijima su .obj, .fbx, .dae itd. Da bi ova aplikacija prepoznala i prikazala 3D model potrebno je da taj fajl bude prebačen u .usdz format. To se odrađuje online preko raznih besplatnih sajtova [3] ili preko novog beta programa Reality Converter [4] koji Apple kompanija teži da usavrši jos više.

Zatim je potrebno kreirati .scn fajl. Najlakši način je nakon selektovanja fajla izabrati Editor pa Convert to sceneKit file format (slika 3).



Slika 3. Primer gotovog modela.

C. Povezivanje 3D modela i slike kartice

Potrebno je napraviti površinu na kojoj će stajati model, koja se naziva planeNode. Koristiti deklaraciju

SCNScene. Zatim povezati model i konačna verzija aplikacije je spremna za korišćenje (slike 4 i 5).

```
let node = SCNNode()
if let imageAnchor = anchor as? ARImageAnchor {
    let plane = SCNPlane(width:
        imageAnchor.referenceImage.physicalSize.width,
        height:
            imageAnchor.referenceImage.physicalSize.height)
    plane.firstMaterial?.diffuse.contents = UIColor(white: 1.0, alpha: 0.5)
    let planeNode = SCNNode(geometry: plane)
    planeNode.eulerAngles.x = -.pi / 2
    node.addChildNode(planeNode)
}
```

Slika 4. Deo koda gde je upotrebljena planeNode scena.

```
if let pokeScene = SCNScene(named: "art.scnassets/eevee.scn") {
    if let pokeNode = pokeScene.rootNode.childNodes.first {
        pokeNode.eulerAngles.x = .pi / 2
        planeNode.addChildNode(pokeNode)
    }
}
```

Slika 5. Deo koda gde je upotrebljena pokeNode scena.

III. MAGICPAPER

Primena aplikacije MagicPaper je povezana sa oblikom Live video. Aplikacija je jednostavnog karaktera gde kamera telefona povezuje sliku koju treba prethodno ubaciti u sam projekat (slika 6). Potrebno je skinuti video koji mora biti u .mp4 formatu [5]. Nakon toga je potrebno iseći i napraviti sliku baš one koje želimo zameniti video klipom. Ubaciti sliku u projekat i kodom povezati sa AR tehnologijom. Koristiti deklaraciju SKVideoNode. Takođe, jako bitna stvar koja ne sme biti izostavljena je podešavanje scene tj. podešavanje rezolucije i veličine videa. To odraditi deklaracijom SKScene (slika 7).

IV. ZAKLJUČAK

Svrha ovih aplikacija je da ljudima približi savremene tehnologije, konkretno AR tehnologiju. Statistika kaže da broj preuzimanja aplikacija koje sadrže AR tehnologiju prelazi 13 miliona, što govori o tome da je zagarantovana svetla budućnost ovoj tehnologiji.

ZAHVALNICA

Zahvaljujem se profesorki Emiliji Živanović i istraživaču Milanu Stojanoviću za pomoć u pisanju rada.



Slika 6. Primer novine.



Slika 7. Primena aplikacije.

LITERATURA

- [1] Proširena realnost. Dostupno na: <https://www.apple.com/augmented-reality/>
- [2] Apple dokumentacija. Dostupno na: <https://developer.apple.com/documentation/arkit/arimagetrackingconfiguration>
- [3] Primer besplatnog sajta. Dostupno na: <https://products.aspose.app/3d/conversion/fbx-to-usdz>
- [4] Program Reality Converter. Dostupno na: <https://developer.apple.com/news?id=01132020a>
- [5] Sajt koji je korišćen za skidanje klipa. Dostupno na: <https://cloudconvert.com/mp4-converter>

Projektovanje termoregulacionog sistema za pametne kuće

Anastasija Avramović, Aleksandra Trajković

Sadržaj – Cilj ovog projekta je realizacija pametne peći koršćenjem kola za drajvovanje releja, lcd displeja, temperaturnog senzora i tastera. Pritiskom odgovarajućeg tastera temperatura se povećava ili smanjuje, čime se kontroliše uključivanje/isključivanje peći ili klime.

I. UVOD

Pametne kuće su sistemi koji osposobljavaju interakciju između uređaja u kući i stanara. Taj sistem čini mreža podataka koja međusobno prenosi informacije i na taj način se izvršava zahtev. Sistemi se međutim ne odnose samo na aparate u kući već i na stvari poput grejanja i svetla. Mogu raditi na struju, preko bluetooth-a, kao i internet [1]. U svetu kakav je danas, u kome postoji sve veća potreba za energijom i gde raste svest o potrebi za očuvanjem prirode, koncept pametne kuće se sve više razvija i poboljšava. Merenje potrošnje i inteligentnije upravljanje potrošačima omogućavaju smanjenje troškova i povećanje komfora i sigurnosti, što podstiče i ekonomski opravdava razvoj uređaja koji se bave tim problemom.

II. KORIŠĆENE KOMPONENTE ZA REALIZACIJU PROJEKTA

A. Arduino UNO

Arduino Uno je mikrokontrolerska ploča bazirana na ATmega328 (prikazana na slici 1). Ima 14 digitalnih ulaza/izlaza (od kojih se 6 mogu koristiti kao PWM (*Pulse-width modulation*) izlazi), 6 analognih ulaza, kristalni oscilator od 16 MHz, USB vezu, priključak za napajanje, ICSP (*In Circuit Serial Programming*) zaglavlje i dugme za resetovanje. Sadrži sve što je potrebno za podršku mikrokontroleru; jednostavno se povezuje na računar pomoću USB kabla ili se napaja DC adapterom ili baterijom. Analogni pinovi mogu biti samo ulazni, dok digitalni pinovi mogu biti ulazni i izlazni [2].

A. Avramović i A. Trajković su studenti četvrte godine na modulu za Elektronske komponente i mikrosisteme Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anastasija.avram@elfak.rs, trajkovic.aleksandra@elfak.rs



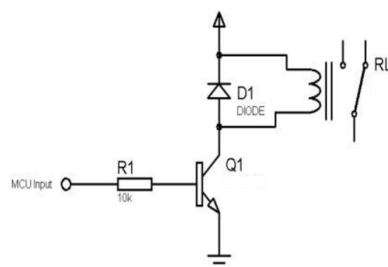
Slika 1. Arduino UNO.

B. Releji

Releji (prikazan na slici 2) se koristi za prekidanje ili uspostavljanje strujnog kola putem elektromagneta koji otvara ili zatvara strujne kontakte [4]. Materijali za kontakte su plemeniti metali: platina, paladijum, iridijum, rodijum, srebro, zlato, volfram i molibden. Zbog visoke cene platine, zlata i rodijuma koriste se platinirani kontakti: osnova je bakar ili srebro, a iznad je pločica od platine ili zlata [7]. Releji se kao i sva druga induktivna opterećenja, nikada ne povezuju direktno na pin bilo kog mikrokontrolera. Moramo imati kolo za drajvovanje koje čine tranzistor i zaštitna diode (prikazano na slici 3).



Slika 2. Releji.



Slika 3. Kolo za drajvovanje releja.

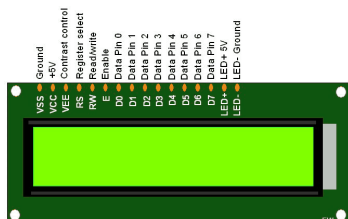
LM35 je integrirani analogni temperaturni senzor (prikazan na slici 4) čiji je električni izlaz proporcionalan stepenu Celzijusa. LM35 senzor ne zahteva nikakvu eksternu kalibraciju ili trimovanje da bi obezbedio tipične tačnosti. Niska izlazna impedansa LM35, linearna zavisnost izlaznog napona od temperature i precizna inherentna kalibracija čine povezivanje sa kolom za očitavanje ili upravljanje posebno lakim. Koristiti se za merenje temperature između -55°C do 150°C. Može se lako povezati sa bilo kojim mikrokontrolerom koji ima ADC funkciju ili bilo kojom razvojnom platformom kao što je *Arduino* [6].



Slika 4. Temperaturni senzor LM35.

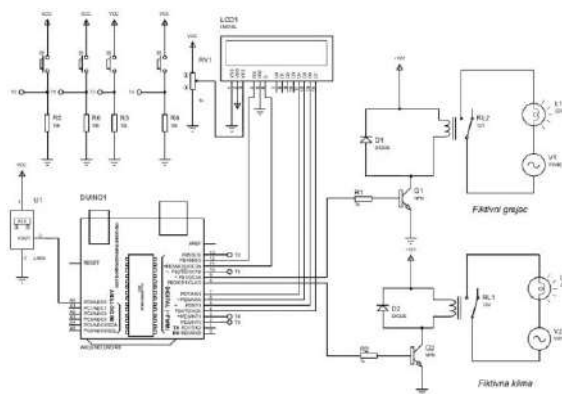
D. LCD displej

LCD (Liquid Crystal display) (prikazan na slici 5) 16x2, što znači da može prikazivati 2 reda od po 16 alfa-numeričkih karaktera. Izvodi LCD displeja su: VSS – masa; VDD – napajanje, najčešće 5V; VEE=V0 pin za podešavanje kontrasta displeja (kontrast se podešava korišćenjem potencijometra); RS (register select) pin kontroliše prenos podataka- kada se na logičkoj jedinici prenose podaci, a kada je na logičkoj nuli prenose se komande; R/W (read/write) pin se postavlja na masu, zato što se vrši samo pisanje u displej; E (enable) pin služi za dozvolu rada; D0-D7 se koriste za prenos podataka. Najčešće se koristi 4-bitni prenos podataka. Iskorišćeni su digitalni pinovi Arduina za povezivanje sa LCD-om. LCD displeji sa pozadinskim osvetljenjem imaju i dva dodatna pina LED+ i LED- koji kontrolišu pozadinsko osvetljenje: LED+ se preko otpornika za ograničenje struje povezuje na napajanje, a LED- na masu. Svaki od segmenata LCD-a je sastavljen iz matrice, najčešće 7x5 ili 8x5, polja ispunjena tečnim kristalom. Brisanje sadržaja sa LCD-a postiže se korišćenjem funkcije `mojLCD.clear()` [7].



Slika 5. LCD displej.

III. PRINCIP RADA SISTEMA



Slika 6. Kreirana šema u softverskom alatu *Proteus*.

Šema je kreirana u softverskom alatu *Proteus* (prikazana na slici 6). Temperaturni senzor LM35 je povezan na analogni pin A0 na Arduinou i meri temperaturu. Na Arduino su povezana dva releja, sa pratećim kolima za drajvovanje. Kada temperatura padne ispod vrednosti TG, uključi se relej broj 1. Relej će uključiti potrošač, u ovom slučaju peć. Kada temperatura poraste iznad TG, relej1 se isključuje. Kada temperatura dođe iznad vrednosti TK uključuje se relej broj 2. Ovaj relej će uključiti klimu. Kada temperature padne ispod TK isključuje se relej2. Zatim, sistem sadrži displej na kome se sve vreme ispisuju vrednosti temperature TK odnosno TG i temperature očitana sa senzora.

Kod [8] je pisan u softverskom alatu *Arduino*. Postupak određivanja vrednosti temperature prikazan je funkcijom void tajmerISR (prikazan na slici 7). U ISR-u se na svaku sekundu, u celobrojnu promenljivu senzor, korišćenjem funkcije analogRead(), upisuje očitana vrednost. Očitana vrednost biće iz opsega 0-1023. Zatim je potrebno odrediti realnu vrednost napona na izlazu senzora, s obzirom da proizvođači u tehničkoj dokumentaciji [6] daju zavisnost temperature u funkciji napona na senzoru. Postavljanjem proporcije, napon se određuje kao senzor*5.0/1024.0. Ova vrednost upisuje se u promenljivu temperatura. Na senzoru će biti prikazana celobrojna vrednost temperature, izvršiće se zaokruživanje temperature korišćenjem funkcije round. Zaokružena vrednost temperature biće smestena u promenljivu roundtemp. Promenljiva flegStampa biće korišćena u delu programa za informisanje korisnika o proteklom vremenu.

```
void tajmerISR(){
  senzor=analogRead(senzorPin);
  napon=senzor*5.0/1024.0;
  temperatura=(napon-0.5)*100.0;
  roundtemp=round(temperatura);
  flegStampa = 1;
}
```

Slika 7. Funkcija void tajmerISR.

```

if (flagStampa) {
  mojLCD.clear();
  mojLCD.print("T=");
  mojLCD.setCursor(3,0);
  mojLCD.print(roundTemp);
  mojLCD.setCursor(5,0);
  mojLCD.write(byte(0));

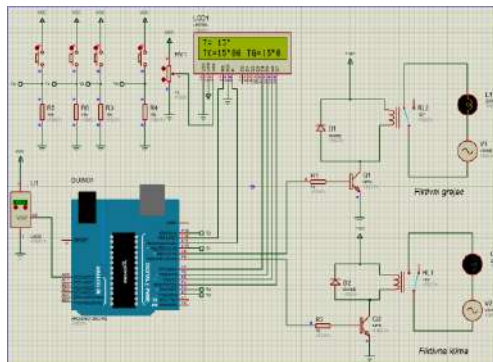
  mojLCD.setCursor(0,1);
  mojLCD.print("TK="); //temperatura kojom se kontroliše klima
  mojLCD.setCursor(3,1);
  mojLCD.print(TK);
  mojLCD.setCursor(5,1);
  mojLCD.write(byte(0));

  mojLCD.setCursor(0,1);
  mojLCD.print("TG="); //temperatura kojom se kontroliše grejač
  mojLCD.setCursor(3,1);
  mojLCD.print(TG);
  mojLCD.setCursor(5,1);
  mojLCD.write(byte(0));
  flagStampa = 0;
}

```

Slika 8. Deo koda kojim se ispisuju temperature.

Kad istekne jedna sekunda u glavnoj petlji void loop (slika 8), osvežimo displej za novi unos temperature. Na lcd-u u prvom redu se prikazuje temperatura očitana sa senzora, dok se u donjem redu prikazuju temperature koje korisnik zadaje kojom se uključuju/isključuju klima ili peć. Sa slike 9 se vidi da je temperatura ambijenta 13°C, korisnik želi da uključi peć kada je temperatura manja od 15°C, sa simulacije se vidi da je relej1, koji predstavlja fiktivni grejač uključen. Temperatura kojom se kontroliše klima iznosi 15 °C, relej2 odnosno fiktivna klima je isključena jer je temperatura ambijenta manja od 15 °C. Iščitanje tastera se vrši pulling metodom.



Slika 9. Simulacija u radu.

III. ZAKLJUČAK

Realizacija termoregulacionog sistema za pametne kuće postignuta je u softverskim alatima *Proteus* i *Arduino*. U projektu su iskorišćena četiri tastera kojim korisnik manuelno može da podešava željene temperature kojima se podešava odgovarajuća vrednost temperature koje iniciraju uključivanje/isključivanje peći ili klime kako bi se osećao komfornije u svom pametnom domu.

ZAHVALNICA

Autori se zahvaljuju Nikoli Mitroviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji ovog projekta.

LITERATURA

- [1] <https://www.procar.rs/blog/pametne-kuce-i-njihove-prednosti.html>
- [2] M. Marjanović „Modeliranje i simulacija mikroelektronskih komponenti i kola”. Dostupno na: <http://mikro.elfak.ni.ac.rs/wp-content/uploads/Arduino-zbirka-resenih-projekata.pdf>
- [3] Lj. Vračar „Senzori i pretvarači”. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Lab_vezba_LM35.pdf
- [4] V. Paunović „Materijali za elektroniku”. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Lab_slides.pptx
- [5] Tehnička dokumentacija za Arduino UNO. Dostupno: <https://www.farnell.com/datasheets/1682209.pdf>
- [6] Tehnička dokumentacija za LM35. Dostupno na: <https://www.electroschematics.com/wp-content/uploads/2010/02/LM35-DATASHEET.pdf>
- [7] Tehnička dokumentacija za LCD displej. Dostupno na: <https://components101.com/displays/16x2-lcd-pinout-datasheet>
- [8] <http://basicarduinotutorial.blogspot.com/2017/08/project-iii-7-thermostat-with-lcd-and-3.html>
- [9] Kod je dostupan na: <https://github.com/trajkovicewa16/file1/blob/main/klima>

Projektovanje i realizacija podizača napona na štampanoj ploči

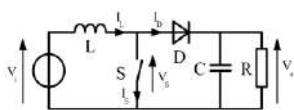
Rade Tošić

Sadržaj – Podizač napona je električno kolo koje na izlazu daje veći napon u odnosu na napon koji se dovodi na ulazu. Realizacija ovog kola odrađena je u softveru *Altium Designer*. Pre toga bilo je neophodno napraviti pločicu u *Solid Edge*-u. Nakon kreiranja pločice i njeno eksportovanje u .dxf fajl prelazi se na projektovanje štampane ploče u *Altium Designer*-u. U *Altium Designer*-u se crta šema uređaja, postavljaju se komponente na željena mesta, postavljaju se odgovarajuća pravila koja se odnose na štampane ploče i izrađuje se lista materijala. Kada pločica bude realizovana u *Altium Designer*-u, prelazi se na smeštanje iste u kućište koje je projektovano za uređaj. Smeštanje pločice u kućište, kao i njegov finalni izgled, se projektuje u *Solid Edge*-u. Nakon projektovanja sledi praktična realizacija podizača napona.

I. UVOD

Podizač napona (*Boost converter*, bust konvertor, podizač napona) je DC-DC konvertor snage kod koga je vrednost izlaznog napona veća nego vrednost ulaznog napona. Podizač napona je *SMPS* klase (*switched-mode power supply*) koji se sastoji od najmanje dve poluprovodničke komponente (diode i tranzistora), barem jednog izvora napajanja, kondenzatora, kalema (ili njihova kombinacija). Filtri napravljeni od kondenzatora (ponekad u kombinaciji sa kalemom) se obično dodaju izlazu konvertora sa ciljem da se smanje oscilacije napona. Izvor napajanja za podizač napona može biti bilo koji DC izvor napajanja, poput baterije, solarnog panela, DC generatora.

Ključni princip kod bust konvertora je sposobnost kalema da se odupre promenama u strujnom kolu. Kod bust konvertora, izlazni napon je uvek veći od ulaznog [1]. Osnovna šema *boost converter*-a prikazana je na slici 1.



Slika 1. Osnovna šema *Boost converter*-a.

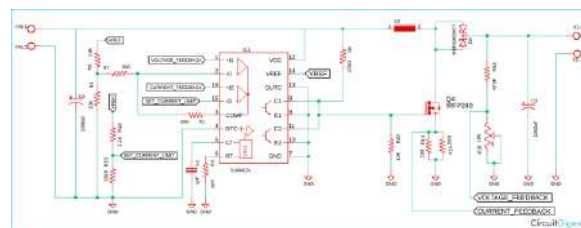
II. OPIS ELEKTRIČNE ŠEME REALIZOVANOG KOLA

IC TL494 [2] ima zadatak da obezbedi odgovarajuće impulse za okidanje tranzistora. Elektrolitski kondenzator

R. Tošić je student osnovnih studija modula Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: radetosic96@gmail.com.

C4 je ulazni kondenzator i on je tu da zadovolji velike trenutne potrebe koje se javljaju kada se MOSFET prekidač zatvori i kalem počne da se puni. Otpornici R7 i R1 čine upravljačku petlju, uz pomoć ove povratne sprege, izlazni PWM signal se linearno menja. Bez otpornika u povratnoj sprezi komparator će delovati kao kolo koje se uključuje / isključuje samo na zadatom naponu. Otpornici R10 i R15 postavljaju ograničenje struje u kolu.

Na neinvertujući ulaz prvog operacionog pojačavača (pin 1) koji služi za pojačavanje greške dovodi se uzorak izlaznog napona koji se uzima sa razdelnika napona R19 i R18 a na invertujući ulaz se dovodi referentni napon sa razdelnika napona R2 i R8. Otpornici R2 i R8 podešavaju upravljački napon za povratnu petlju. Razlika ta dva napona predstavlja grešku. Šema celokupnog projekta prikazana je na slici 2.



Slika 2. Šema projekta.

Kontrola izvora napajanja vrši se sistemom povratne sprege tako što se izlazni napon poredi sa referentnim naponom, traži se greška i vrši kompenzacija promenom *duty cycle*-a, tako da kroz potrošač protiče konstantna struja, tj. napon bude stabilisan. PWM kolo konvertuje izlaz iz komparatora (*compensated error amplifier*) u *duty cycle*, tako što izlaz iz komparatora upoređuje sa testerastim signalom koji generiše oscillator. Sa pojačavača grešaka napon se dalje vodi na PWM komparator koji u zavisnosti od tog napona određuje širinu impulsa za okidanje izlaznih tranzistora.

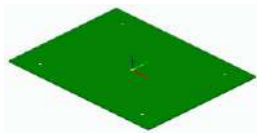
Postavljanjem odgovarajućih vrednosti na pinove 5 i 6, može se postaviti frekvencija komutacije ovog IC. Za ovaj projekat je korišćena vrednost kondenzatora od 1 nF i vrednost otpornika od 10 kΩ što nam daje približno frekvenciju od 100 kHz.

Dioda MBR10020CT [3] sprečava pražnjenje kondenzatora preko prekidača, kada je zatvoren. MBR20100 je dvostruka Šotki dioda, dakle dve diode su u jednom kućištu sa zajedničkom katodom, pin 1 i pin 3 su anode svake od dioda.

Tranzistor IRFP250n [4] radi u prekidačkom režimu – položaj radne tačke, odnosno režima rada, određuju otporici $R4$ i $R20$. Kada impuls otvori tranzistor on provede tako da kroz kalem protekne velika struja što izazove namagnetisanje njegovog jezgra. Kada ulazni impuls padne na nulu tranzistor prestaje da vodi, dođe u stanje zakočenja i struja kroz kalem je mala i jednaka struji potrošača. To sada izaziva naglo razmagnetisavanje kalema, što dalje prouzrokuje indukovanje dodatnog napona u namotajima kalema pa imamo veći napon od ulaznog. Taj povišeni napon se preko dioda dovodi na kondenzator $C3$ koji se puni na vrednost izlaznog napona. On omogućava konstantan napon na izlazu [5].

III. IZRADA PLOČICE I KUĆIŠTA U SOLID EDGE-U

Pre realizacije projekta u *Altium*-u potrebno je napraviti pločicu u *Solid Edge*-u [6]. Na osnovu dimenzije kućišta, broja komponenata i njihove veličine određuje se veličina pločice. Koristeći dobijene dimenzije iscertava se pravougaonik koji se naredbom *extrude* izvuče do zadovoljavajuće debljine, koju pločica treba da ima. Nakon toga potrebno je izbušiti rupe za montažu pomoću naredbe *Hole*, prečnika 3.5 mm. Dve rupe se pozicioniraju pomoću ove naredbe, a druge dve se dobijaju *mirror*-ovanjem u odnosu na x osu. Na taj način dobija se pločica koja se eksportuje u .dxf fajl kako bi mogla da se uveze u *Altium*. Izgled tako kreirane pločice prikazan je na slici 3.



Slika 3. Pločica izrađena u *Solid Edge*-u.

Po završetku projektovanja u *Altium*-u vrši se eksportovanje 3D modela pločice sa svim komponentama u *Solid Edge*. Sledeći korak je smeštanje pločice u kućište. To se radi poravnavanjem dna pločice sa stubićima (ležištima), na kućistu. Zatim je potrebno aksijalno postaviti rupe na pločici sa rupama na stubićima. Za ovo se biraju dve rupe koje su dijagonalno. Na ovaj način pločica se pozicionira u kućište. Tada se javio problem da su nožice komponenata virile iz kućista. Bilo ih je potrebno odseći. Sledeći korak je prilagođavanje samog kućišta. Iseku se rupe za hađenje, kao i rupe za pristup klemama za ulazni / izlazni napon. Na kraju su označeni ulazi i izlazi, sa ciljem olakšavanja upotrebe uređaja krajnjem korisniku. Finalni izgled projektovanog uređaja u *Solid Edge*-u sa i bez poklopca dat je na slici 4.



Slika 4. Finalni izgled projektovanog uređaja.

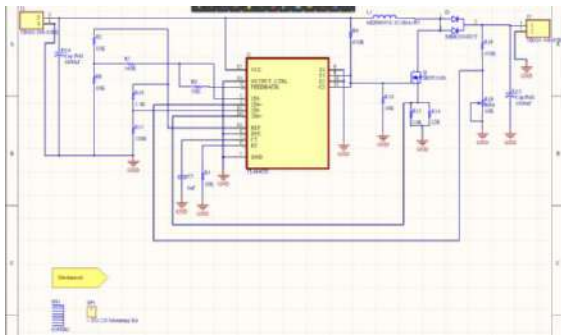
IV. IZRADA PROJEKTA U ALTIVM DESIGNER-U

Pre same realizacije projekta bilo je potrebno odrediti dimenzije kako same pločice tako i najviše komponente koja se na nju montira. U ovom slučaju to je hladnjak. Najvažnija karakteristika hladnjaka jeste njegova termička otpornost koja zavisi od snage disipacije komponente koja se povezuje za njega, njegove temperature kao i temperature ambijenta. Na osnovu proračuna dobijena termička otpornost hladnjaka je iznosila $14.05\text{ }^{\circ}\text{C/W}$, međutim, pošto je bolje da termička otpornost bude manja od izračunate izabrani hladnjak je sa termičkom otpornošću od $3.30\text{ }^{\circ}\text{C/W}$ kako bi u potpunosti bili sigurni da je hladnjak odgovarajući.

Kada je pitanje hladnjaka rešeno može se nastaviti sa pronalaženjem kućišta za sam uređaj. Visina hladnjaka je 27.3 cm što znači da kutija treba da bude nešto malo veća od te vrednosti. Istraživanjem gotovih 3D modela na internetu dolazi se do odgovarajućeg kućišta. Visina samog kućišta je 35.25 cm, tako da uređaj bez problema može da se smesti u njega.

Sledeći korak je realizacija ploče u *Altium*-u [6]. Pre samog početka izrade projekta treba uvesti pločicu koja je prethodno napravljena u *Solid Edge*-u. Nakon toga može se kreirati šematik i crtati šema. Šema izrađena u *Altium Designer*-u je prikazana na slici 5. Prilikom crtanja šeme svaka komponenta se bira iz određene biblioteke i podešava se njen *footprint*. Međutim većina ovih komponenata nisu često korišćene tako da ih nema u listi podrazumevanih komponenti u *Altium*-u, tako da ih je potrebno pronaći i za svaku od njih instalirati biblioteku. Prilikom ubacivanja svake od komponenata u šematik u okviru kartice *properties* sa desne strane treba podesiti *designator*, kako bi označili šta predstavlja komponenta, *value*, kako bi definisali vrednost kao i tolerancije za određene komponente poput otpornika i kondenzatora i naravno još jednom proverili *footprint*-ove svake komponente.

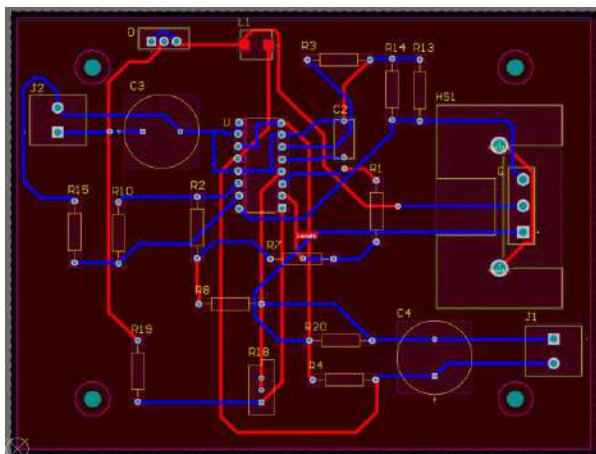
Osim samih električnih komponenata u šemi postoje i mehaničke komponente koje su podjednako važne za realizaciju projekta. Tako da ih je neophodno navesti u programu kako bi se našle na listi materijala. Ovde su hladnjak i držač za hladnjak jedine mehaničke komponente.



Slika 5. Šema izrađena u Altium Designer-u.

Nakon iscrtavanja šeme, projekat nije gotov. Najvažniji deo jeste postavljanje komponenti sa šeme na ploču koja je uvezena. Pomoću funkcije *Update PCB Document Ploca.PcbDoc* u okviru kartice *Design* sve komponente, tačnije njihovi *footprint*-ovi, koji se nalaze u šematičku smeštaju se na štampanu ploču. Naravno treba pomenuti da prvobitni raspored komponenata nije bio kao na slici 6 već su naknadno poređane. Po završetku ređanja usledila je provera koji od 3D modela nedostaje i njihovo uvoženje na pločicu pomoću naredbe *3D body* u okviru kartice *Place*. Kao što je već napomenuto, većina otisaka, simbola kao i 3D modela komponenata mogu se naći na internetu. Međutim sa nekim od tih 3D modela mogu se javiti problemi. Dešavalo se to da *footprint* i 3D model nisu kompatibilni iako su oba skinuta sa istog sajta. Ti problemi su rešeni crtanjem modela komponenata direktno u Altium-u u okviru kartice *Place* bira se *3D body*, a zatim se u *properties* izabere određeni oblik komponenete i na kraju se upisuje visina komponenata.

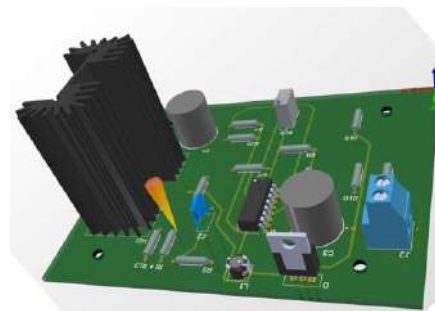
Kada svaka komponenta ima 3D model i da su oni zaključani zajedno sa otiskom, prelazi se na rutiranje. U okviru kartice *Route* postoji opcija *Auto Rout* koja je od velike pomoći, gde nakon toga mora da se odradi samo još par sitnijih prepravki. Na kraju, dobija se konačni izgled pločice u Altium Designer-u prikazan na slici 6.



Slika 6. Konačni izgled pločice u Altium Designer-u.

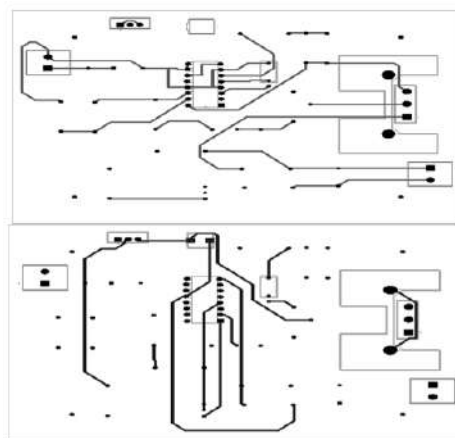
Veoma je važno pomenuti pravila koja su definisana u Altium-u neka od njih su: električna pravila, pravila vezana za linije veze, pravila vezana za SMD komponente, *manufacturing* i *placement* pravila. Program je javio ukupno 33 greške, što znači da su 33 puta prekršena neka od navedenih pravila. Kako bi rešili greške neophodno je bilo predefinisati pravila. Pravila se menjaju u okviru prozora *Design Rule Check* koja se nalazi u kartici *tools*.

Pritiskom tastera 3 na tastaturi prelazi se na 3D prikaz pločice. Na slici 7 se može videti izgled realizovane pločice. Takođe prikazom 3D modela može se ustanoviti da ne postoje kolizije na pločici.



Slika 7. 3D prikaz pločice.

Jedan od važnijih koraka je dodavanje gerber fajlova. Gerber fajlovi su jako bitni jer mašine koje vrše proizvodnju štampanih ploča mogu da ih čitaju. Oni predstavljaju maske koje se koriste za proizvodnju štampanih ploča. Na slici 8 prikazani su gerber fajlovi.



Slika 8. Gerber fajlovi.

Poslednji korak u realizaciji štampane ploče je pravljenje liste materijala neophodnih za njenu realizaciju. U okviru Altium-a moguće je automatski izraditi listu materijala, BOM, pomoću opcije *Bill of Materials* iz padajućeg menija *Project Reports* u kartici *Reports*. U tabeli koja je izvezena u Excel-u mogu se videti podaci o komponentama koje su korišćene poput *designator*-a, *footprint*-a, broja istih komponenata itd. Takođe moguće je

dodati i kod dobavljača kako bi olakšali posao ljudima koji vrše porudžbinu.

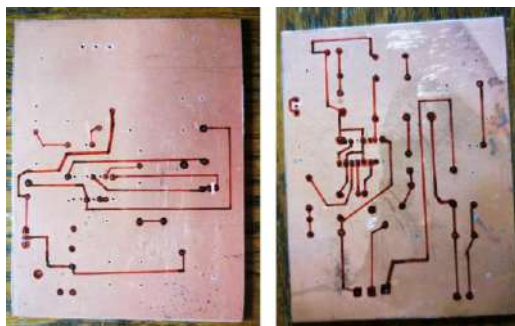
V. IZRADA PLOČICE

Kada je projekat u *Altium*-u bio gotov, bilo je moguće da se na osnovu njega izradi i uređaj. Za izradu projeketa korišćena je pločica od vitroplasta. Ono što je specifično kod nje je da je dvostruko kaširana. Korišćenje ovakve pločice je neophodno jer se u projektu linije veze nalaze sa obe strane.

Nakon izbora vrste pločice sledi iscrtavanje dimenzija, a nakon toga pločica se iseče. Za to je korišćen tanak list bonseka.

Sledeći korak je bušenje rupa za komponente. Da bi znali gde se rupe tačno nalaze, tj. gde da ih bušimo moramo prvo da ih obeležimo. Odštampa se na papiru šema u razmeri 1:1 i postavi se na pločicu. Uz pomoć kirner-a (mali metalni šiljak, nalik šrafčigeru) obeleži se gde treba da se buše rupe. Sada je sve spremno za bušenje. Sam proces bušenja obavlja se jednom malom, veoma tankom burgijom.

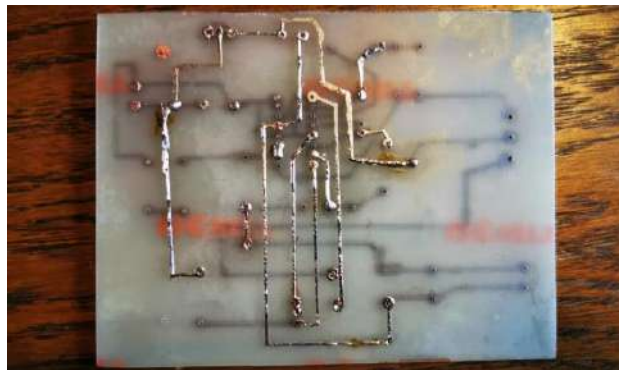
Važno je napomenuti da je prvo potrebno izbušiti rupe kako bi u sledećem koraku bilo lakše iscrtavanje linija veza (lakše je za orijentaciju pre svega, mada mogu se prvo i iscrtati linije veza pa da se onda odradi bušenje). Dakle, sledeći korak je iscrtavanje linija veza. Linije veza iscrtavaju se rapidografom debljine 0.8 mm. Te linije služe kao neka vrsta rezista, odnosno, posle nagrizanja samo na tim mestima će ostati bakar. Na slici 9 je prikazan izgled pločice sa gornje i donje strane nakon bušenja rupa i iscrtavanja inija veza rapidografom.



Slika 9. Izgled pločice sa gornje i donje strane nakon bušenja rupa i iscrtavanja inija veza rapidografom.

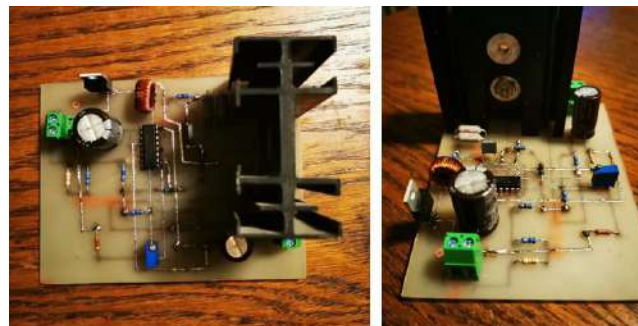
Dakle, kao što je već napomenuto, nakon iscrtavanja linija veza sledi nagrizanje. Cilj je da na pločici ostanu samo bakarne veze. Nagrizanje se vrši pomoću hidrogena i sone kiseline u razmeri 1:4. Sam proces nagrizanja vrši se tako što se pločica stavi u jednu posudu i zatim se doda mešavina hidrogena i sone kiseline. Ta mešavina nagriza bakar koji je višak. Nakon toga pločica se izvadi i opere vodom. Naravno, posle pranja, potrebno je i ošušiti pločicu. Kada se dobro osuši spremna je za dalji rad.

Naredni korak je kalaisanje veza (nanošenje kalaja na linije veza). To se radi u cilju pojačavanja/poboljšavanja veza i da bi kasnije bilo lakše lemljenje komponenata. Na slici 10 je prikazan izgled pločice nakon kalaisanja linija veza.



Slika 10. Pločica nakon kalaisanja linija veza.

Ostalo je još samo da se postave komponente i da se zaleme. Tako da je to sledeći korak. Komponente se postavljaju po određenom pravilu. Odnosno, prvo se postavljaju komponente koje su manje termički osetljive (pasivne komponente kao što su otpornici), a zatim aktivne komponente (integrisano kolo, dioda itd). Nakon što se postave komponente zaleme se, a viškovi u vidu predugačkih nožica se iseku. Konačni izgled pločice prikazan je na slici 11.



Slika 11. Konačni izgled pločice.

VI. ZAKLJUČAK

Na osnovu testiranja realizovanog uređaja može se zaključiti da ovo kolo radi kako treba. Na izlazu se dobija veći napon od ulaznog. Testiranje je izvršeno u kućnoj varijanti pomoću ispravljača laptopa i multimetra.

ZAHVALNICA

Zahvaljujem se profesorima sa predmeta Projektovanje štampanih ploča, posebno prof. dr Zoranu Prijiću na smernicama prilikom realizacije projekta.

LITERATURA

- [1] Dokić, Branko L. (2007). Energetska elektronika - pretvarači i regulatori. Beograd/Banja Luka: Akademska misao/Elektrotehnički fakultet.
- [2] https://www.ti.com/lit/ds/symlink/tl494.pdf?ts=1635739934441&ref_url=https%253A%252F%252Fwww.google.com%252F
- [3] <https://www.onsemi.com/pdf/datasheet/mbr20100ct-d.pdf>
- [4] <https://www.vishay.com/docs/91212/91212.pdf>
- [5] <https://circuitdigest.com/electronic-circuits/designing-high-power-high-efficiency-boost-converter-using-tl494>
- [6] Materijal sa vežbi iz predmeta Projektovanje štampanih ploča, Elektronski fakultet Niš

Realizacija laserskog sigurnosnog sistema upotrebom Arduino Uno razvojnog okruženja

Andela Jovanović

Sadržaj – U ovom radu će biti predstavljeno jednostavno kolo za realizaciju laserskog sigurnosnog sistema upotrebom *Arduino Uno* razvojnog okruženja. Cilj projekta bio je dizajniranje sigurnosnog sistema zasnovanog na laseru i fotootporniku. Rad detektora sistema je baziran na principu razdelnika napona. Imamo rednu vezu fotootpornika i otpornika sa fiksnom vrednošću. Vrednost fotootpornika je promenljiva u zavisnosti od količine svetlosti koja pada na njegovu površinu, a samim tim dolazi i do promene izlaznog napona na ovom razdelniku. Ovaj napon je ujedno i napon koji se dovodi do analognog ulaza *Arduina* i na osnovu te vrednosti dolazi do aktivacije alarma. Da bi se postigla kontrola ovog sistema, zbog svoje jednostavnosti, *Arduino Uno* je izabran kao glavni kontroler koji kontroliše rad celokupnog sistema. Ovaj projekat je vrlo jednostavan i pomaže da saznamo više o komponentama koje se često koriste.

I. UVOD

Verovatno ste kao mlađi gledali stari *Western* film u kojem se dobri momci kriju i provlače žicu u visini članaka oko svog skrovišta, vezujući je za konzervu napunjenu kamenjem. Kada loši momci pokušaju da se prišunjuju usred noći, oni zakače žicu i povlače limenku koja proizvodi zvuk i alarmira dobre momke dižući iz ih sna. Laserski sigurnosni sistem radi po veoma sličnom principu, samo što umesto žice imamo laserski snop svetlosti koji okružuje područje koje želimo da zaštitimo, a umesto limenke pune kamenja za alarmiranje koristimo piezo zujalicu koja proizvodi zvučni signal.

Ovaj jednostavan laserski sigurnosni sistem ima pet bitnih komponenti pored *Arduino Uno* razvojnog okruženja: taster, laser, fotootpornik kao detektor, LED diode kao vizuelni i piezo zujalicu kao zvučni indikator. Laser je koncentrisani izvor svetlosti koji emituje jednobojni ravan snop svetlosti. Fotootpornik kao detektor je osetljiv na svetlost i sa većim intenzitetom svetlosti njegova otpornost opada. LED diode emituju svetlost kada su direktno polarisane i kada kroz njih protiče struja. Piezo zujalica se širi i skuplja zavisno od vrednosti napona koji se primenjuje na nju, promenom napona tokom vremena piezo menja oblik dovoljno brzo da stvori talas pritiska koji naše uho tumači

A. Jovanović, Katedra za mikroelektroniku, Univerzitet u Nišu, Elektronski fakultet Niš, Aleksandra Medvedeva 14, Niš, Srbija, E-mail: andjela.andjix@elfak.rs

kao zvuk. Kada je laserski snop svetlosti prekinut i ne može doći do detektora, njegov naponski izlaz se menja, a ostatak kola oseti promenu i šalje vizuelne i zvučne signale upozorenja.

II. DETALJI, REALIZACIJA I RAD REALIZOVANOG LASERSKOG SIGURNOSNOG SISTEMA

Za realizaciju ovog kola potrebno je pregledno i tačno rasporediti i povezati komponente na protoboard ploči, koja će se kasnije povezati zajedno sa *Arduino Uno* razvojnim okruženjem, kako bi se omogućilo jednostavno i korektno očitavanje parametara. Zatim je potrebno programirati *Arduino Uno* mikrokontroler.

A. Komponente sistema

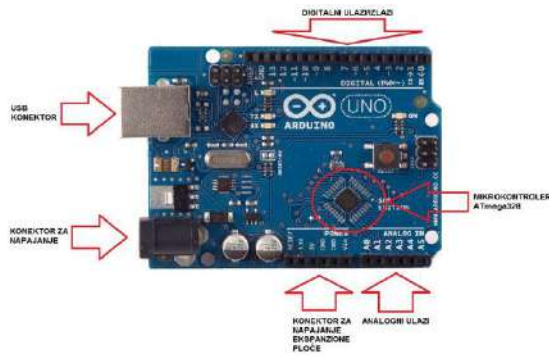
Kao što mu samo ime govori, cilj projekta je realizacija laserskog sigurnosnog sistema upotrebom pristupačnih i jednostavnih komponenti. Pored lasera, komponente koje su upotrebljene u projektu su:

- Glavni kontroler: *Arduino Uno* razvojno okruženje
- Displej za prikaz informacija: Serijski monitor ugrađen u *Arduino IDE* softveru
- Izvor svetlosti: Laser
- Detektor pokreta: Fotootpornik
- Alarm: LED diode kao vizuelni i piezo zujalica kao zvučni
- Aktivacija/deaktivacija sistema: Taster

a. *Arduino Uno* razvojno okruženje

Arduino Uno razvojno okruženje je u suštini elektronska platforma otvorenog koda. To je štampana ploča sa mikrokontrolerom čija je svrha programiranje interakcija sa hardverskim uređajima kao što su motori, svetla, senzori i mnogi drugi. *Arduino Uno* razvojno okruženje je pogodno za ovaj projekat zahvaljujući svojim mnogobrojnim mogućnostima i sposobnosti komunikacije sa različitim komponentama koje su upotrebljene u ovom sistemu. *Arduino Uno* razvojno okruženje kao glavnu komponentu koristi mikrokontroler ATmega328. Pored njega poseduje 14 ulazno/izlaznih pinova (od kojih se 6 mogu koristiti kao PWM), 6 analognih ulaza, keramički oscilator od 16MHz, USB konektor, konektor za napajanje, konektore za ekspanziju ploče i taster za reset. Okruženje poseduje sve ono što je neophodno mikrokontroleru da radi, samo je potrebno prikačiti ga na napajanje kako bi se pokrenuo [1].

Na slici 1 je prikazan izgled jednog *Arduino Uno* razvojnog okruženja.



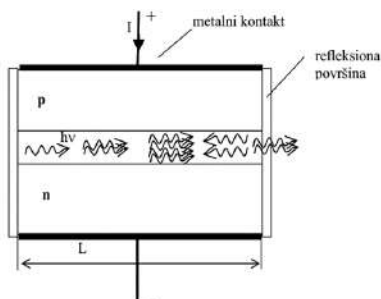
Slika 1. Izgled *Arduino Uno* razvojnog okruženja.

b. Laser kao izvor svetlosti

Rad laserske diode (eng. LASER – *Light Amplification by Stimulated Emission of Radiation*) se zasniva na procesu stimulisanе emisije zračenja. Stimulisana emisija svetlosti predstavlja elektromagnetno zračenje nastalo prelaskom atoma na osnovni ili na niža energetska stanja, izazvano spoljašnjim zračenjem iste energije. Ovakva emisija svetlosti daje koherentnu (elektromagnetni talasi su u istoj fazi), monohromatsku (elektromagnetni talasi imaju istu talasnu dužinu) i polarizovanu (elektromagnetni talasi osciluju u istoj ravni) svetlost.

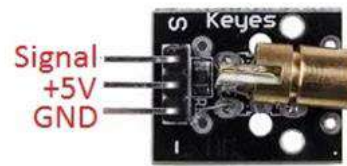
Radi ostvarenja pojačanja zračenja putem stimulisanе emisije, potrebno je prethodno „pumpanjem“ lasera ostvariti inverziju populacije, a oblast u kojoj se ostvaruje inverzija populacije se naziva aktivna oblast. Inverzija populacije znači da je veća popunjenost elektronima pri dnu provodne nego pri vrhu valentne zone. Za generisanje laserskog zračenja neophodno je da aktivna oblast bude smeštena unutar rezonatora. Ovaj rezonator obezbeđuje višestruki prolazak svetlosti kroz aktivnu oblast, a pri svakom novom prolasku intenzitet svetlosti se putem stimulisanе emisije pojačava.

Na slici 2 je prikazana osnovna struktura laserske diode. Imamo p i n spoj između kojih se nalazi aktivna oblast u kojoj se obavlja višestruki prolazak svetlosti između dva visoko reflektujuća ogledala, čime se intenzitet svetlosti pojačava pre nego što svetlost napusti aktivnu sredinu [3].



Slika 2. Strukturni prikaz laserske diode.

Na slici 3 je prikazan laserski modul upotrebljen u projektu. Modul poseduje tri pina: pin za napajanje od 5V, pin za masu i pin za signal koji može ostati nepovezan ali se može koristiti za očitavanje napona na modulu.

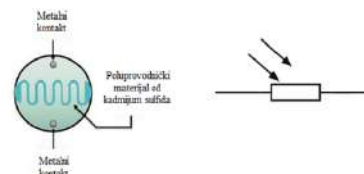


Slika 3. Laserski modul KY-008.

c. Fotootpornik kao detektor

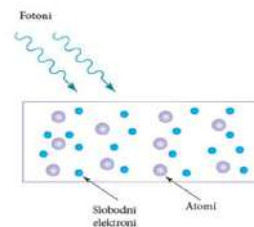
Fotootpornik je tip otpornika čija otpornost opada pri povećanju inteziteta svetlosti. Drugim rečima, intezitet struje kroz fotootpornik povećava se sa povećanjem inteziteta svetlosti koja na njega pada.

Na slici 4 je prikazana struktura jednog fotootpornika sa njegovim električnim simbolom. Fotootpornik je načinjen od keramike koja je odličan izolator, sam poluprovodnički materijal koji je osetljiv na svetlost je kadmijum sulfid i on je postavljen na samoj površini fotootpornika u obliku cik-cak linije. Razlog zašto se koristi baš cik-cak oblik je kako bi se dobila maksimalna dužina ovog materijala da bi dobili očekivanu snagu i otpornost, a da komponenta i dalje ostane malih fizičkih dimenzija. Ova linija odvaja par metalnih kontakata, a otpor ovih kontakata treba da bude što manji kako ne bi imao veliki uticaj na sam rad fotootpornika.



Slika 4. Osnovna struktura fotootpornika i njegov električni simbol u električnim šemama.

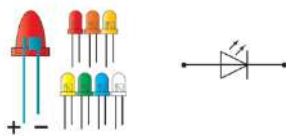
Na slici 5 je prikazan način generisanja slobodnih elektrona u fotootporniku. Kada na fotootpornik padne svetlost, neki valentni elektroni apsorbuju energiju svetlosti i kidaju svoje veze sa osnovnim atomima, te postaju slobodni elektroni, a upravo ovi slobodni elektroni koji se kreću čine električnu struju [2].



Slika 5. Generisanje slobodnih elektrona u fotootporniku.

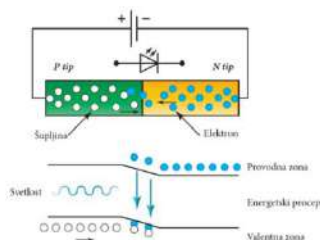
d. LED dioda kao vizuelni alarmni indikator

LED dioda (eng. *Light Emitting Diode*) predstavlja svetleću diodu koja pretvara električnu u energiju svetlosti. Struja kroz LED diodu teče kada se ona direktno polariše jednosmernim naponom i kao rezultat toga ona će svetleti, a jačina inteziteta svetlosti zavisi od jačine struje koja protiče kroz nju u datom trenutku. Na slici 6 prikazan je grafički izgled LED diode sa njenim električnim simbolom. Radi lakšeg raspoznavanja, anodni priključak je uvek duži od katodnog.



Slika 6. Grafički izgled LED diode i njen električni simbol u električnim šemama.

Na slici 7 je prikazan princip rada jedne LED diode. Da bi poluprovodnik emitovao svetlost, neophodno je da postoji veliki broj pobuđenih elektrona, a pobuđivanje elektrona s nižeg na viši energetske nivo najbolje se postiže direktnom polarizacijom p-n spoja. Usled direktne polarizacije, elektroni prelaze iz n-oblasti u p-oblast, gde se rekombinuju sa šupljinama. Pri rekombinaciji elektrona i šupljina, emituju se fotoni svetlosti tj. LED dioda svetli [3].



Slika 7. Princip rada LED diode.

e. Piezo zujalica kao zvučni alarmni indikator

Piezo zujalica radi po principu inverzije piezoelektričnog efekta koji je otkriven 1880. godine od strane braće Jacques i Pierre Curie-a. Kada se određeni piezoelektrični materijali izlože naizmeničnom polju električne energije, element piezo zujalice se rasteže i sabija u skladu sa frekvencijom ulaznog napona. Kao rezultat toga, proizvodi se talas koji naše uho registruje kao zvuk. Piezo zujalice imaju radni opseg između 3 i 250 V. Na slici 8 je prikazan izgled piezo zujalice.



Slika 8. Fizički izgled piezo zujalice.

f. Taster za aktivaciju/deaktivaciju sistema

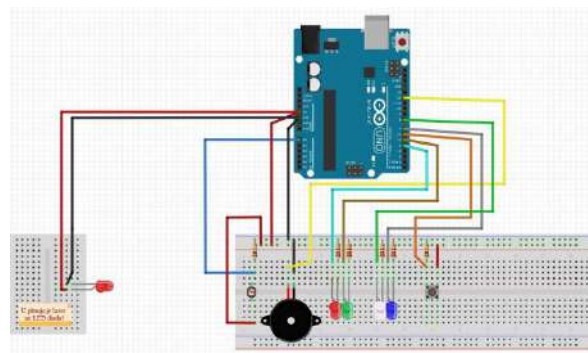
Taster predstavlja vrstu prekidača u električnim kolima koji prekidaju ili spajaju dva ili više provodnika samo kada je pritisnut od strane korisnika. Inače se automatski, putem opruge vraća u stalni – početni položaj. Struja koju taster prekida nije velika, obično je reda do 500 mA. Za prekidanje većih struja koriste se prekidači, a za struje preko 10 A se koriste sklopke. Izgled tastera upotrebljenog u ovom projektu je prikazan na slici 9.



Slika 9. Fizički izgled tastera.

B. Realizacija i princip rada Sistema

Na slici 10 je prikazana električna šema projekta koja je nacrtana u programu *Fritzing* [4]. U šemi, ključne komponente jesu laserska dioda i fotootpornik.



Slika 10. Šema povezivanja laserskog sigurnosnog sistema.

Sistem radi na sledeći način: Pri pokretanju rada sistema prvo se vrši kalibracija fotootpornika. Laserski snop prvobitno nije usmeren ka fotootporniku kako bi se prvo uzeli uzorci svetlosti iz osvetljenja prirodnom svetlošću i na osnovu njih postavio prag za tu svetlost. Nakon postavljanja tog praga za prirodnu svetlost, laserski snop se usmerava na površinu fotootpornika i sistem uzima uzorke svetlosti iz osvetljenja laserskom svetlošću pa takođe na osnovu njih postavlja drugi prag za lasersku svetlost. Uslov za uzimanje uzoraka laserske svetlosti je da oni imaju veću vrednost od praga postavljenog prirodnom svetlošću. Nakon određivanja i ovog praga, fotootpornik je kalibrisan i sistem može da počine sa regularnim radom.

Zatim imamo crvenu i zelenu LED diodu od kojih je zelena indikator da je sistem neaktivan, dok je crvena indikator da je sistem aktivan. Takođe pored ove dve LED diode imamo još dve LED diode, belu i plavu, koje se koriste za vizuelno alarmiranje. Imamo i piezo zujalicu za zvučno alarmiranje. I na kraju taster kojim se vrši aktivacija ili deaktivacija sistema.

IV. ZAKLJUČAK

Laserski sigurnosni sistemi imaju mnoge prednosti, jednostavni su za ugradnju i mogu se efikasno koristiti unutar ili van kuće. Sistemi se mogu koristiti kao veoma efikasan alarm za velike granične površine nalik dvorišta oko kuće ili čak i za bazene, gde se sistem može postaviti tako da vrši detekciju i aktivira alarm kada mala deca dođu do par centimetara od ivice bazena. U zatvorenom prostoru detektor i laser se obično postavljaju na mestima koja nisu toliko upadljiva, npr. unutar utičnica, dok se napolju mogu sakriti ispod biljaka ili žbunja i neće naštetiti travnjacima ili drugoj vegetaciji. Nedostatak je u tome što ovi sistemi mogu biti i izuzetno skupi. Iako neki planovi sigurnosnog sistema ciljaju na zaštitu jedne prostorije, planovi koji štite veća područja ili celu kuću koštaju mnogo više i dosta ih je teže priuštiti. Ovakvi sistemi pored lasera i fotootpornika za detekciju pokreta koriste wireless komunikaciju sa sistemom upotrebom aplikacije preko interneta, kamere unutar i van kuće, senzore za prozore koji detektuju da li je staklo polomljeno, detektore dima, detektore ugljen monoksida, temperaturne senzore itd. Sistem koji je realizovan u okviru ovog projekta je najjeftinija moguća varijanta laserskog sigurnosnog sistema i cenovnik tih komponenti je prikazan na slici 11.

Komponenta	Broj	Cena (RSD)	Ukupna cena (RSD)
Arduino Uno razvojno okruženje	1	1090,00 ^[5]	1090,00
Protoboard ploča	2	480,00 ^[6]	960,00
Kablovi za protoboard ploče	1 komplet	600,00 ^[7]	600,00
Laserski modul KY-008	1	330,00 ^[8]	330,00
Foto otpornik	1	95,00 ^[9]	95,00
Piezo zujalica	1	220,00 ^[10]	220,00
LED diode (crvena, zelena, bela i žuta)	4	20,00 ^[11]	80,00
Taster	1	28,00 ^[12]	28,00
Otpornici (220Ω i 10kΩ)	6	1,00 ^[13]	6,00

Slika 11. Cena komponenti upotrebljenih u sistemu.

LITERATURA

- [1] W. Durfee (2011) „Arduino Microcontroller Guide”
- [2] Ranko V. , Radovan B. (2020) „Osnove elektronike za I razred elektrotehničke škole”
- [3] Predavanja iz predmeta Laserska elektronika (2021) „LED i laserske diode”
- [4] Programski paket Fritzing. Dostupno na: <https://fritzing.org/download/>
- [5] Proelectronic radnja – Arduino Uno: <http://www.proelectronic.rs/Arduino%20razvojni%20sistemi/17027-ARDUINO%20UNO%20R3>
- [6] M&G electronic radnja – Protoboard ploča: <https://www.mgelectronic.rs/protoboard-plo%C4%8Da-300100-kontakta>
- [7] M&G electronic radnja – Kablovi za protoboard ploče: <https://www.mgelectronic.rs/komplet-kablova-za-protoboard-plo%C4%8De>
- [8] Proelectronic radnja – Laserski modul KY-008: <http://www.proelectronic.rs/Arduino%20razvojni%20sistemi/17149-ARDUINO%20LASER%20MODUL%20KY-008>
- [9] Proelectronic radnja – Fotootpornik: <http://www.proelectronic.rs/Optoelementi/Fotootpornik/13421-FOTO%20OTPORNİK%20LDR03>
- [10] Proelectronic radnja – Piezo zujalica: <http://www.proelectronic.rs/Zvucnici/Buzeri-piezo%20zujalice/5720-BUZER%20BZ-05%20%28KPE260%29>
- [11] Proelectronic radnja – LED diode (crvena, zelena, bela i žuta): <http://www.proelectronic.rs/Optoelementi/Le%20diode/Led%2010%20mm>
- [12] Proelectronic radnja – Taster: <http://www.proelectronic.rs/Tasteri/Impulsni/15356-MINI%20TASTER%206X6mm%20X5mm%20MEKI%20TASTER>
- [13] Proelectronic radnja – Otpornici (220Ω i 10kΩ): <http://www.proelectronic.rs/Otpornici/0%2C25w%20otpornik/Tolerancija%205%20%25/74-OTPORNİK%200%2C25W%20220R>

Odziv VDMOS tranzistora snage na NBT naprezanje, γ -zračenje i odžarivanje

Sandra Veljković, Nikola Mitrović

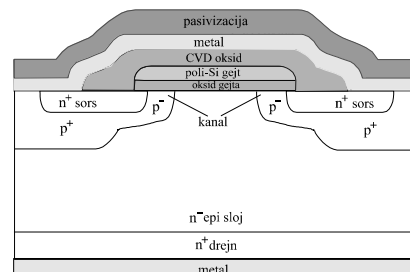
Sadržaj - U okviru ovog rada razmatrane su promene napona praga do kojih dolazi usled izlaganja VDMOS tranzistora snage NBT naprezanju, zatim jonizujućem zračenju i na kraju, odžarivanju. Detaljno je opisana primena ovih snažnih komponenata, kao i neka od istraživanja koja se sprovede kako bi se ispitala njihova pouzdanost. Prikazan je ceo proces eksperimenta, koja je aparatura korišćena, kao i koji su sve uslovi važili tokom njegovog sprovođenja. Nisu sve komponente izlagane potpuno istim uslovima naprezanja. Prva grupa komponenata je ozračivana bez prethodnog naprezanja, druga nakon 1 h, a treća nakon 168 h prethodnog NBT naprezanja. Na kraju je data analiza promena do kojih dolazi.

I. UVOD

Iako su VDMOS (*Vertical Double Diffused Metal-Oxide Semiconductor*) [1] komponente poznate više od 50 godina, interesovanje za njihovim istraživanjem i primenom ne opada. Pre svega, cena proizvodnje je prihvatljiva jer je tehnologija koja se koristi relativno jednostavna. Masovna proizvodnja je opravdana jer se često ugrađuju i u uređaje za posebne namene ali i u komercijalno dostupne elektronske uređaje. Zbog specifičnog seta karakteristika koje poseduju, polje njihove primene je veoma veliko. Neke od tih primena su: kao senzori i dozimetri jonizujućeg zračenja, u audio opremi, u auto-elektronici [1-3]. Kako je korišćenje VDMOS tranzistora veoma rasprostranjeno, veoma je bitno da te komponente rade tačno i precizno, odnosno pouzdano. Zato se proučava kako ove komponente reaguju na različita naprezanja kao što su naponsko-temperaturna naprezanja, injekcija vrućih nosilaca, prisustvo jakih električnih polja, izlaganje jonizujućem zračenju i slično [4-8]. Tokom godina, uočeno je da dolazi do promene u radu, koja je posledica promena njihovih električnih parametara od kojih je najvažniji napon praga (V_T) ali i pokretljivost nosilaca u kanalu i struje curenja. Te nestabilnosti parametara tranzistora su posledica promena u gustini naelektrisanja zahvaćenog u oksidu gejta (ΔN_{ot}) i površinskih stanja (ΔN_{it}) i mogu da dovedu do otkaza kola u kojima komponente rade. Iako se ovi procesi godinama proučavaju, mehanizmi koji uzrokuju promene električnih parametara do danas nisu u potpunosti razjašnjeni. Iz tog razloga, sprovede se ispitivanja koja imaju za cilj da se razjasni koji procesi kako deluju na rad VDMOS tranzistora snage. Poprečni

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

presek ove komponente je prikazan na Slici 1.



Slika 1. Poprečni presek dve polućelije p-kanalnog VDMOS tranzistora snage.

Tokom godina, razvijani su različiti testovi i metode za ispitivanje promena odgovarajućih parametara i kako se to odražava na komponente. Već više decenija se intenzivno radi na tome, jer su u toku svoje eksploatacije komponente izložene različitim vidovima naprezanja, odnosno, rade pod težim uslovima. Kako nije praktično da se testovi sprovedu u dugom periodu (više desetina godina), razvijeni su testovi koji se sprovedu vremenski kraće, ali uslovi kojima se izlažu su dosta rigorozniji od realnih situacija. Time se omogućava da se predvidi vreme pouzdanog rada komponenata u dužem vremenskom periodu [9, 10].

Pri korišćenju ovih snažnih tranzistora u uređajima koji se ugrađuju u nuklearne elektrane, u medicinske uređaje ili satelite koji se koriste u komunikacijama, dolazi do konstantnog izlaganja ili s vremena na vreme uticaju jonizujućeg zračenja. Mnoga istraživanja su pokazala da postoji analogija između efekata koji se javljaju pri izlaganju jonizujućem zračenju i jakim električnim poljima, odnosno, da oba naprezanja dovode manje ili više do generacije naelektrisanja u oksidu i površinskih stanja. Rezultati koji se dobijaju ispitivanjem efekata naprezanja oksida gejta, mogu u mnogome da daju objašnjenja koja se tiču pouzdanog rada VDMOS tranzistora snage [9, 13-17]. Takođe, od velike važnosti su promene koje se javljaju usled primenjenih visokih napona i temperatura. Veoma važne su promene do kojih pri tome dolazi, što je u inostranoj literaturi poznato pod nazivom NBTI (*Negative Bias Temperature Instability*). Ovi testovi se najčešće sprovedu kod p-MOS tranzistora, pri čemu polja pri kojima se promene javljaju u oksidu gejta su od 2 MV/cm do 6 MV/cm, a temperature od 100 °C do 250 °C [3, 5, 15, 16, 18]. Pored toga što se ove temperature javljaju pri standardnom radu komponenata, karakteristične su i za

sprovođenje testova žarenja ili u inostranoj literaturi „burn-in“ testovi [19]. U okviru ove grupe testova sprovode se temperaturno-naponsko naprezanje gejta (HTGB - *High Temperature Gate Bias*), naprezanje inverznom polarizacijom (HTRB - *High Temperature Reverse Bias*) i skladištenje pri visokoj temperaturi [2, 19, 20]. Tokom rada u elektronskim uređajima koji se koriste u industriji, često postoji zahtev da komponente naizmenično rade a zatim ne rade. Takođe, u nekim primenama, u uređajima postoje rezervne komponente koje u početku ne rade i za to vreme su izložene jonizujućem zračenju, a zatim se uključuju kada se za tim javi potreba. Pri daljem radu, komponente mogu biti izložene uticajima koji odgovaraju NBT naprezanju.

U okviru ovog rada izvršeno je ispitivanje p-kanalnih VDMOS tranzistora snage. Vršeno je NBT naprezanje za dva vremenska perioda, γ -zračenje i na kraju termičko odžarivanje.

II. SPROVEDENI EKSPERIMENT

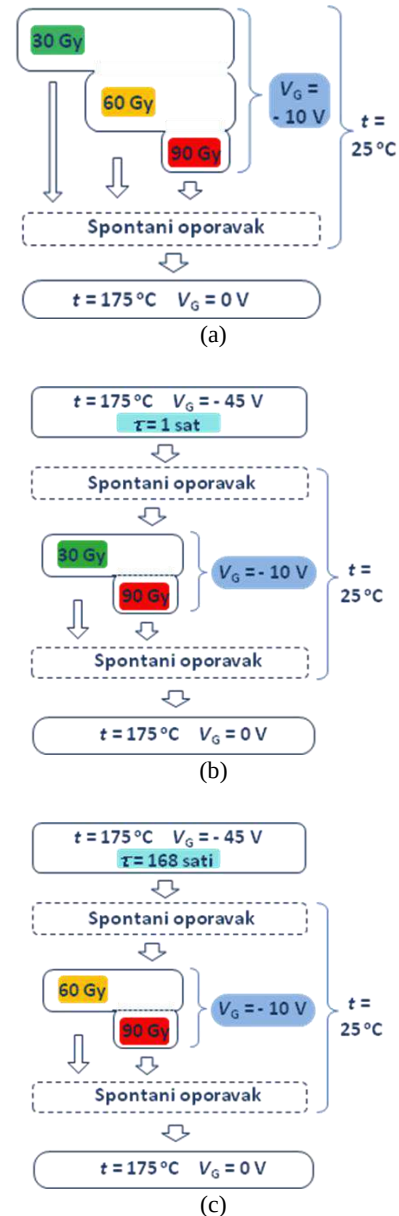
Kao komponente koje su testirane u ovom eksperimentu, korišćeni su p-kanalni VDMOS tranzistori snage. Ove komponente su komercijalno dostupne pod oznakom IRF9520. Neke od osnovnih karakteristika ovih komponenata su te da su proizvedene u standardnoj poli-Si tehnologiji i debljina gejta je 100 nm. Kod ovih tranzistora maksimalna struja koja može protiče je 6.8 A, dok je napon praga koji je izmeren pre eksperimenata bio $V_{TO} = -3.6$ V. Komponente su pakovane u plastično kućište TO-220, imaju 1650 ćelija i njihova struktura je heksagonalna.

U eksperimentu, koji je šematski predstavljen na Slici 2, testirane komponente su podeljene u tri grupe. Prva grupa komponentata nije bila NBT naprezana, dok su druga i treća bile izložene NBT naprezanju u trajanju od 1 h i 168 h, respektivno. Naponsko temperaturno naprezanje negativnom polarizacijom je sprovedeno na Elektronskom fakultetu u Nišu, u Laboratoriji za mikroelektroniku i elektronske komponente. Uređaj koji je korišćen za ovaj vid naprezanja je *Hereus HEP2* sistem komora, koji se koristi kao sistem za proveru pouzdanosti. 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 1 h za drugu grupu komponenata, a treća grupa je izlagana istoj temperaturi 168 h, odnosno jednu nedelju.

Nakon što su uzorci prošli prvu fazu eksperimenta, bili su izlagani jonizujućem zračenju. Pri zračenju je primenjena polarizacija gejta od -10 V, dok su ostali izvodi bili uzemljeni. Čitav deo eksperimenta koji se tiče zračenja, sproveden je u Institutu za nuklearne nauke “Vinča” u Srbiji, u Laboratoriji za zaštitu od zračenja i zaštitu životne sredine. Testirane komponente se postavljaju ispred izvora jonizujućeg zračenja - Co-60, i brzinom doze od 0.5 Gy(SiO₂)/min se ozračuju do željene

doze. Kako bi rezultati mogli da se upoređuju, komponente su ozračene do iste ukupne doze od 90 Gy.

Nakon γ -zračenja, poslednja faza eksperimenta je bila termičko odžarivanje. Ovaj deo eksperimenta je sproveden na Elektronskom fakultetu u Nišu i za njegovu realizaciju je korišćen sterilizator. U njemu su odžarivani tranzistori tokom 168 h, bez polarizacije i na temperaturi od 175 °C.

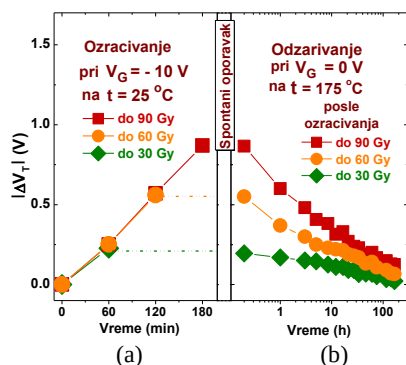


Slika 2. Eksperimentalna procedura sprovedena za (a) prvu, (b) drugu i (c) treću grupu testiranih komponenata.

Kako bi se izvršila električna karakterizacija, pre svega pratila promena V_T , tokom svih faza eksperimenta, naprezanja su prekidana. U tačno određenim vremenskim intervalima i na sobnoj temperaturi, snimane su prenosne karakteristike, i kao SMU (*Source Measure Unit*) korišćen je Keysight Technologies B2901A. Ovaj uređaj se kontrolisao preko kompjutera na koji je povezan pomoću GPIB-a.

III. REZULTATI I NJIHOVA ANALIZA

Kao što je pomenuto, električna karakterizacija vršena je tokom naprežanja, i rezultati svih testiranih grupa su prikazani na Slici 3, Slici 4. i Slici 5. Na Slici 3. prikazane su promene napona praga po apsolutnoj vrednosti (ΔV_T) za tranzistore koji prethodno nisu bili NBT naprežani, dok su na Slici 4 i Slici 5 prikazane promene napona praga po apsolutnoj vrednosti za tranzistore koji su prethodno izlagani NBT stresu. NBT naprežanje dovelo je do pomeranja karakteristika u desnu stranu, što praktično znači da dolazi do povećanja vrednosti napona praga, što je prikazano na Slici 4 (a) i Slici 5 (a). Može da se uoči da NBT naprežanje u trajanju od 168 h dovodi do značajnijeg povećanja napona praga nego izlaganje komponenata kratkotrajnom NBT naprežanju u trajanju od 1 h. Za prvu, a nakon spontanog oporavka za drugu i treću grupu komponenata, usledilo je ozračivanje do ukupne doze od 90 Gy, što se vidi na Slici 3 (a), Slici 4 (b) i Slici 5 (b). Prva grupa komponenata je podeljena u tri podgrupe. Prva je ozračena do 30 Gy (60 min), druga do 60 Gy (120 min), dok je treća podgrupa primila ukupnu dozu od 90 Gy (180 min), što je prikazano na Slici 3 (a). U okviru druge grupe komponenata, izvršena je podela na dve podgrupe, pri čemu je jedna ozračivana do 30 Gy a druga do 90 Gy (Slika 4 (b)). Podela treće grupe je slična drugoj grupi, sa tim što su podgrupe ozračivane do 60 Gy i 90 Gy, što se vidi na Slici 5 (b).

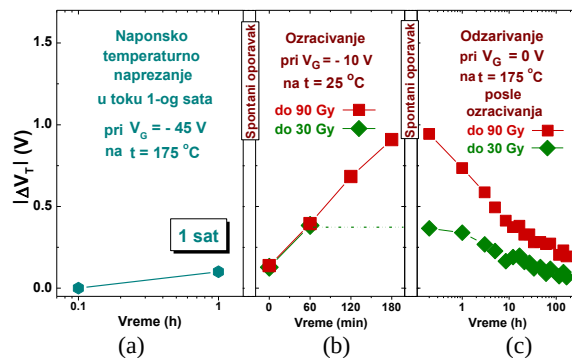


Slika 3. Porast ΔV_T IRF9520 tranzistora snage u toku (a) ozračivanja i (b) odžarivanja.

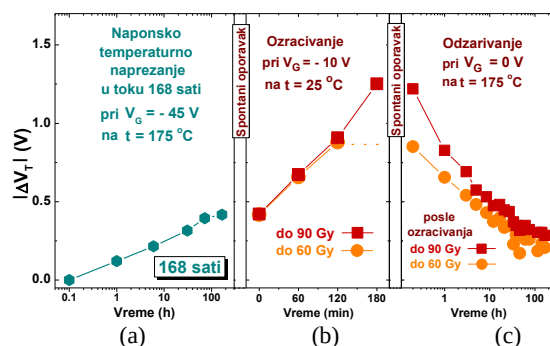
S obzirom da p-kanalni tranzistori u praksi rade pri negativnoj polarizaciji, polarizacija koja je primenjena pri ozračivanju je -10 V. To ozračivanje dovelo je do daljeg značajnog porasta napona praga po apsolutnoj vrednosti. Kako se radi o p-kanalnom VDMOS tranzistoru, to praktično znači da su vrednosti napona praga nakon ozračivanja još negativnije. Promene napona praga se javljaju jer dolazi do formiranja naelektrisanja u oksidu - N_{ot} i površinskih stanja - N_{it} . Do njihovog formiranja dolazi usled elektrohemijjskih procesa u kojima učestvuju šupljine, čestice vodonika, kao i defekti u oksidu i na međupovršini.

Tokom primene γ -zračenja, dolazi do generisanja parova elektron-šupljina u oksidu gejta. Iako najveći broj

elektrona napusti oksid kroz poluprovodnik, kod nekih parova dolazi i do rekombinacije. Međutim, veliki broj šupljina, koje su slabije pokretne od elektrona, bude zahvaćen na defektima. Usled toga dolazi do povećanja gustine naelektrisanja u oksidu gejta i na međupovršini.



Slika 4. Porast ΔV_T IRF9520 tranzistora snage u toku (a) NBT naprežanja od 1-og sata, (b) ozračivanja i (c) odžarivanja.



Slika 5. Porast ΔV_T IRF9520 tranzistora snage u toku (a) NBT naprežanja od 168 sati, (b) ozračivanja i (c) odžarivanja.

Treba napomenuti da krajnje vrednosti ΔV_T zavise od eventualnog prethodnog stresiranja komponenata. Konačne vrednosti ΔV_T su svakako izraženije kod komponenata koje su izlagane NBT naprežanju od 168 h, a zatim 1 h, dok su najmanje kod komponenata ozračenih bez prethodnog izlaganja stresu. Uočeno je da tokom samog zračenja, dolazi do nešto bržeg porasta napona praga kod grupe komponenata koje prethodno nisu izlagane NBT stresiranju. To je posledica većeg broja dostupnih prekursora naelektrisanja u oksidu gejta i površinskih stanja u toj grupi komponenata [21-24]. Takođe, mogu da se uporede grupe koje su pre zračenja izlagane NBT naprežanju u kraćem i dužem vremenskom intervalu. Promene V_T su nešto izraženije kod treće grupe komponenata, odnosno grupe izlagane naponsko temperaturnom naprežanju u trajanju od 168 h. Nakon ponovnog spontanog oporavka, usledilo je odžarivanje svih komponenata na temperaturi od 175 °C, bez primene polarizacije na gejtu i u trajanju od 168 h. Proces odžarivanja je prikazan na Slici 3 (b), Slici 4 (c) i Slici 5 (c). Kod svih komponenata ovaj proces je doveo do značajnog opadanja vrednosti ΔV_T , koji je najizraženiji na samom početku. Kao što može da se primeti, krajnje

vrednosti ΔV_T na Slici 4(c) nisu niže od vrednosti ΔV_T neposredno nakon NBT naprezanja. S druge strane, na Slici 5(c) primetno je da su konačne vrednosti ΔV_T ispod vrednosti neposredno nakon NBT naprezanja. Kako komponente nisu bile polarisane tokom odžarivanja, svi efekti koji se javljaju su termički aktivirani. To praktično znači da su elektrohemijske reakcije i procesi do kojih dolazi, posledica difuzije neutralnih molekula H_2 i pozitivnih jona H^+ .

IV. ZAKLJUČAK

U radu su analizirani uticaji NBT naprezanja, jonizujućeg zračenja pri negativnoj polatizaciji gejta i odžarivanja na promene napona praga kod IRF9520 komercijalnih tranzistora snage. Prikazan je porast ΔV_T po apsolutnoj vrednosti kod ozračenih komponenata koje prethodno nisu izlagane, kao i onih koje su izlagane kraćem (1 h) i dužem (168 h) NBT naprezanju. Uočeno je da tokom zračenja, dolazi do nešto bržeg porasta napona praga kod grupe komponenata koje prethodno nisu izlagane NBT stresiranju, što je posledica većeg broja dostupnih prekursora naelektrisanja u oksidu gejta i površinskih stanja u toj grupi komponenata. Takođe, ΔV_T u toku zračenja su veće kod komponenata koje su prethodno napezane duže u odnosu na promene nakon kraćeg NBT stresa. Tokom odžarivanja, vrednosti ΔV_T značajno opadaju. Međutim, kod grupe komponenata koja je prethodno kraće NBT napezana, ΔV_T ne pada niže od vrednosti nakon samog naprezanja, dok kod komponenata prethodno duže izlaganih NBT naprezanju ΔV_T opada i do nižih vrednosti od onih nakon NBT naprezanja.

ZAHVALNICA

Prikazani rezultati dobijeni su u okviru programa za finansiranje naučnoistraživačkog rada (pod brojem 451-03-9/2021-14/200102) koji finansira Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije i u okviru projekta "Osobine tankih i ultratankih oksidnih slojeva" (F-148), koji finansira Srpska akademija nauka i umetnosti-SANU. Rezultatima predstavljenim u ovom radu je u mnogome doprineo nedavno preminuli akademik Ninoslav Stojadinović i svojim naučnim i istraživačkim aktivnostima i značajnom saradnjom sa autorima.

LITERATURA

- [1] V. Benda et al., "Power Semiconductor Devices", New York-Oxford, Oxford University Press, 2000.
- [2] N. Stojadinović et al., "Effects of high electric field and elevated-temperature bias stressing on radiation response in power VDMOSFETs", *Microelectron. Reliab.*, vol. 42, pp. 669-677, 2002.
- [3] A. Demesmaeker et al., "Bias temperature reliability of p-channel high-voltage devices", *Microelectron. Reliab.*, vol. 37, pp. 1767-1770, 1997.
- [4] T. P. Ma and P. V. Dressendorfer, "Ionizing Radiation Effects in MOS Devices and Circuits", 1989, Wiley, New York.
- [5] S. Gamerith and M. Polzl, "Negative bias temperature stress in low voltage p-channel DMOS transistors and role of nitrogen", *Microelectron. Reliab.*, vol. 42, pp. 1439-1443, 2002.
- [6] N. Stojadinović et al., "Effects of electrical stressing in power VDMOSFETs", *Microelectron. Reliab.*, vol. 45, pp. 115-122, 2005.
- [7] G. S. Ristić, et al. "Defect behaviors in n-channel power VDMOSFETs during HEFS and thermal post-HEFS annealing", *Appl. Surf. Sci.*, vol. 252 (8), 2006, 3023-3032.
- [8] X. R. Ye et al., "VDMOSFET HEF degradation modelling considering turn-around phenomenon", *Microelectron. Reliab.*, vol. 50, pp. 37-41, 2018.
- [9] D. Danković et al., "NBT Stress-Induced Degradation and Lifetime Estimation in p-channel Power VDMOSFETs", *Microelectronics Reliability*, vol. 46, pp. 1828 - 1833, 2006.
- [10] S. Veljković, "Mogućnosti predviđanja pouzdanosti tehničkih sistema i elektronskih komponenata", Studentska konferencija IEEEESTEC 2020, Niš, pp. 211-214, 2020.
- [11] D. Danković et al., "Određivanje perioda pouzdanog rada p-kanalnih VDMOS tranzistora snage podvrgnutih kontinualnim i impulsnim NBT naprezanjima", Zbornik radova LVI konferencije za ETRAN, (str. MO1.1-1- MO1.1-4), Zlatibor, 11-14 Jun 2012.
- [12] D. Danković et al., "Implications of Negative Bias Temperature Instability in Power MOS Transistors" in *Micro Electronic and Mechanical Systems*, edited by Kenichi Takahata, IN-TECH Press, Boca Raton, vol. 342, pp.19.319-19. 2009.
- [13] V. Huard et al., "NBTI degradation: From physical mechanisms to modelling", *Microelectron. Reliab.*, vol. 46, pp. 1-23, 2006.
- [14] J. H. Stathis, S. Zafar, "The negative bias temperature instability in MOS devices: A Review", *Microelectron. Reliab.*, vol. 46, pp. 270-286, 2006.
- [15] D. K. Schroder, J. A. Babcock, "Negative bias temperature instability: Road the cross in deep submicron silicon semiconductor manufacturing", *J. Appl. Phys.*, vol. 94, pp. 1-18, 2003.
- [16] S. Ogawa et al., "Interface trap generation at ultrathin SiO₂ (4-6 nm) - Si interfaces during negative-bias temperature aging", *J. Appl. Phys.*, vol. 77, pp. 1137-1148, 1995.
- [17] N. Stojadinović et al., "Negative bias temperature instability mechanisms in p-channel power VDMOSFETs", *Microelectron. Reliab.*, vol. 45, pp. 1343-1348, 2005.
- [18] K.O. Jepson and C.M. Svensson, "Negative bias stress of MOS devices at high electric fields and degradation of MNOS devices", *J. Appl. Phys.*, vol. 48, pp. 2004-2014, 1977.
- [19] N. Stojadinović et al., "Effects of burn-in stressing on radiation response of power VDMOSFETs", *Microelectron. J.*, vol. 33, pp. 899- 905, 2002.
- [20] S. Djorić-Veljković et al, "Effects of burn-in stressing on post-irradiation annealing response of power VDMOSFETs", *Microelectron. Reliab.* vol. 43, pp. 1455-1460, 2003.
- [21] V. Davidović et al., "NBTI and irradiation effects in p-channel power VDMOS Transistors", *IEEE Trans. Nucl. Sci.*, vol. 63, pp. 1268-1275, 2016.
- [22] D. Danković et al., "Radiation and Annealing Related Effects in NBT Stressed P-Channel Power VDMOSFETs", *Microelectronics Reliability*, vol. 61, In Press, 2021.
- [23] S. Veljković et al., "Efekti zračenja i odžarivanja kod naponsko temperaturno napezanih p-kanalnih VDMOS tranzistora snage", Zbornik radova LXV konferencije za ETRAN, (str. MO1.1-1- MO1.1-4), Republika Srpska, 2021.
- [24] S. Veljković et al., "Effects of Bias Temperature Stress and Irradiation in Commercial p-Channel Power VDMOS Transistors", *Proc. 32th International Conference on Microelectronics (MIEL 21)*, Niš, October 2021, pp. 345-350, 2021.

Android aplikacija Newsy – društvena mreža za diskutovanje novosti

Dorđe Mitrović

Sadržaj – U ovom radu prikazan je razvoj aplikacije Newsy. Aplikacija je rađena u tehnologiji *React Native*, koristi *Firebase* za autentikaciju i čuvanje, i pristup tekstualnim i multimedijalnim podacima, i *Native Module*, pisanim u Javi, za internu bazu podataka i API zahtevima. U aplikaciji se najnovije novosti, a i one najstarije, mogu diskutovati pisanjem novih postova, i reagovanjem na njih. Postove mogu videti svi korisnici aplikacije, kao i interaktovati sa postom.

I. UVOD

Danas su telefoni neizostavna stvar, svako danas poseduje barem jedan mobilni uređaj. Telefoni mogu poslužiti za ozbiljnije stvari, ali mogu služiti i za zabavu tokom umornog dana nakon posla. Kako je tehnologija napredovala, *Android* je postao vodeći sistem koji koristi danas najveći procenat mobilnih telefona, ali se može naći i na pametnim satovima, televizorima, i drugim mobilnim uređajima. Kako je rasla prisutnost mobilnom telefonu, kao i njegova praktičnost, cilj razvoja bilo koje aplikacije jeste zapravo da bude laka za korišćenje, da korisnik može da se bez ikakvih problema snađe u korišćenju funkcija i pogodnosti aplikacije. Međutim osim toga bitno je i kako izgledaju sve te pogodnosti, da bude privlačno ljudskom oku, jer će zapravo lep dizajn privući, a funkcionalnost zadržati.

Danas su društvene mreže neizostavni deo svakog telefona. Komunikacija je nikad razvijenija, slanje multimedijalnog sadržaja je nikad lakše, i postajace sve lakše. Društvene mreže danas nude veoma veliki broj mogućnosti, i tek će se razvijati. U svetu društvenih mreža gde ovladavaju giganti poput Instagrama, Twittera i ostalih, uglavnom sve novosti mogu da se saznaju preko istih. Međutim da bi se saznalo više o tim novostima, potrebno je „izguglati“ te novosti. Danas uglavnom društvene mreže ne nude opširne informacije o novostima. Tu nastupa aplikacija Newsy.

Aplikacija Newsy učitava vesti, sa globalnog i sa lokalnog nivoa, sa svih izvora, i nudi čitanje celokupnih članaka prikupljenih sa raznih novinarskih izvora. Listanje članaka je veoma lako, i može se pronaći članak o istom događaju, prikupljen sa različitih izvora. Novosti mogu da se lajkuju, međutim ne mogu da se komentarišu, već može da se otvori diskusija na osnovu te vesti, gde će

drugi moći da odgovore komentarom. A moguće je i privatno diskutovanje o nekoj novosti, odnosno komuniciranje privatnim porukama između osoba. Novosti mogu da se prosleđuju u porukama.

II. REALIZACIJA APLIKACIJE

Source kod projekta je 90% pisan u tehnologiji *React Native* [1], dok je ostatak pisan u Javi. Za korišćenje svih mogućnosti *React Native*, neophodno je nakon instaliranja *Node Package Manager-a* (NPM) [2] instalirati dodatne biblioteke za lakši rad, ali i neophodan rad. Pisani kod u Javi je rađen korišćenjem *Android Studio* [3]. Za preuzimanje novosti koristi se besplatan API koji nudi *NewsCatcher*. Za autentikaciju, globalnu bazu podataka, i smeštaj multimedijalnih podataka, koriste se besplatne usluge platforme *Firebase* [4]. Listu instaliranih biblioteka, kao i svi kodovi koji su pisani za izradu ove aplikacije može se naći na *Githubu* [5].

A. *React Native*

React Native je programski jezik zasnovan na *JavaScript-u* i široko se koristi za izradu mobilnih aplikacija, za *Android* i za *iOS*. Razlog za korišćenje *React Native-a* je zbog jednostavnosti kreiranja aplikacije, kao i brzina izrade. Kod *React Native-a* može da se piše u *Visual Studio Code-u* [6], koji je za izradu ovakvih aplikacija, najbolji.

Tokom izrade aplikacije koristio se *Metro* server, koji se pokreće preko terminala. Kada je *Metro* server pokrenut, svaka izmena u kodu aplikacije odmah osvežava aplikaciju na emulatoru, ili mobilnom uređaju koji se koristi za testiranje aplikacije.

Za testiranje aplikacije, dizajna kao i svih funkcionalnosti koristio se emulator koji nudi razvojno okruženje *Android Studio*. Pokretanje aplikacije kao i lokalnog servera za aplikaciju, vrši se pomoću NPM komandi.

Ceo dizajn aplikacije kao i mnoge funkcionalnosti aplikacije rađene su u *React Native-u*. *React Native* koristi takozvane *JSX* elemente za kreiranje svih vidljivih komponenti u aplikaciji. Za izradu aplikacije su se koristili već gotovi *JSX* elementi uzeti iz biblioteke pod nazivom ‘*react-native*’, i zbog toga se na početku svakog fajla, moraju uvesti te komponente, kako bi bile spremne za korišćenje (slika 1).

Đ. Mitrović je student treće godine na modulu Računarstvo i informatika, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: djokatru@elfak.rs

```
import React, {useCallback, useEffect, useL
import {View,
  Text,
  TouchableOpacity,
  SafeAreaView,
  TextInput,
  TouchableHighlight,
  TouchableWithoutFeedback,
  Image,
  NativeModules,
  ActivityIndicator,
  ToastAndroid} from 'react-native';
```

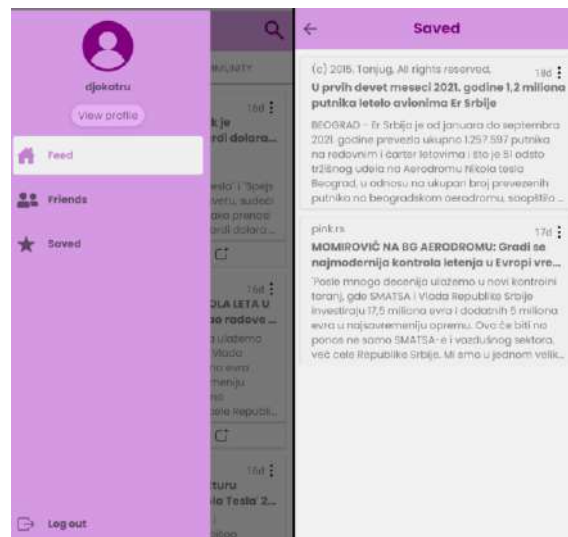
Slika 1. Importovanje JSX elemenata.

Osim JSX elemenata se importuju i različite ikonice iz biblioteke *vector-icons*, koje mogu da se koriste u dizajnu aplikacije. Ikonice daju zabavan karakter aplikaciji, ali i olakšavaju njeno korišćenje.

Kako bi se ulepšao dizajn aplikacije, osim pravilnim rasporedom JSX elemenata, moraju se ti JSX elementi i urediti različitim bojama, oblicima i rasporedom prostora. To se postiže preko stilova. Stilovi za svaku stranicu su grupisani, i nalaze se u posebnom izdvojenom folderu.

U *React Native-u* se svaka, takozvana stranica, koju vidimo (npr. *Feed*, *Options...*) pravi kreiranjem funkcija, čija povratna vrednost zapravo predstavlja izgled stranice, ili kreiranjem komponenata, čijom se funkcijom *Render*, pokazuje izgled ekrana. Što se tiče performansi, iste su i prilikom funkcija, i prilikom komponenata, međutim *source* kod funkcije, izgleda mnogo čistije, i zahteva manje kodiranja od komponenata. Za manipulaciju podacima unutar funkcije ili komponente koriste se *React Hookovi*. *UseState* pamti podatak unutar komponente ili funkcije. Nakon svake promene podataka koji se pamti pomoću *UseState*, ekran se ponovo renderuje, kako bi se promenili podaci. *UseEffect* je *hook*, čije se telo izvršava nakon svakog renderovanja ekrana, ili nakon promene unapred zadatih elemenata, dok *UseLayoutEffect* je *hook* koji može da promeni izgled elemenata prilikom renderovanja. Koristi se da se promeni naslov header-a kada se učitava nečiji profil. Koriste se još i *UseCallback*, za *callback* funkcije, i *UseNavigation*, za manipulisanje navigacijom ekrana.

Za usmeravanja i pomeranje između funkcija ili komponenata, koristi se *React Navigation*, koji nudi mnogo različitih načina navigacije. Za kreiranje ove aplikacije koristile su se *NativeStack* i *Drawer* navigacije (slika 2). *NativeStack* navigacija nudi preusmeravanje ekrana u vidu steka, odnosno svaki put kada korisnik ode na neku stranicu, može se vratiti klikom strelice nazad. *Drawer* navigacija nudi lak izbor stranica na koje korisnik želi da se usmeri otvaranjem menija sa leve strane.



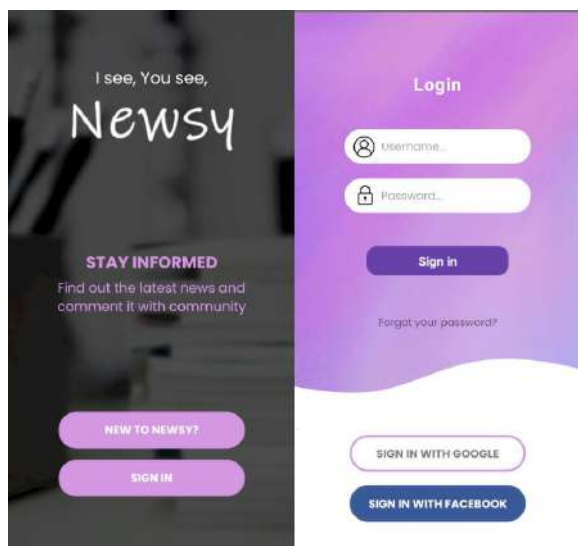
Slika 2. Navigacija u aplikaciji.

B. Firebase

Korišćene usluge platforme *Firebase* su: Autentikacija, Firestore, Realtime Baza podataka i Storage. Pomoću navedenih usluga *Firebase*, mogu se ostvariti većina zahteva aplikacije.

Autentikacija omogućava da svaki korisnik može da poseduje zasebni nalog, koji će koristiti za pristup i korišćenje aplikacije. Kada se novi korisnik pojavi on se upućuje na *Welcome* stranicu, gde bira da li će da se registruje ili prijavi na nalog. Da li je korisnik prijavljen ili nije, koristi se *UseState* hook, preko kojeg se zatim određuje da li će se korisnik uputiti na *Welcome* stranici ili na *Feed* stranici, i stavlja se odgovarajuća stranica na početak *NativeStack* navigacije. Umesto *useState* hook moguće je korišćenje i *React Context-a*, odnosno *Redux-a*. To bi mu davalo na funkcionalnosti, ali kako bi aplikacija imala što bolje performanse, odnosno kako bi se odmah moglo ustanoviti da li je korisnik prijavljen, koristi se *useState* hook, a tek nakon toga se dobijaju potrebni podaci, koji kasnije mogu da se čuvaju u *Context-u*, ali je u ovoj aplikaciji odabrana *SQLite* baza podataka.

Firebase omogućava da korisnik može da se registruje i pristupa aplikaciji i putem *Google* naloga, i *Facebook-a* (slika 3). Radi lakše realizacije ovakvih načina prijavljivanja, a i lakšeg kreiranja modernog dizajna, koriste se dve biblioteke, i to su *Google-Signin* i *Fbsdk-next*. Kada se korisnik prvi put prijavi u aplikaciji, usmeruje se na posebnu *First Time* stranicu, gde unosi svoje ime, prezime i svoj unikatni *username*. U slučaju zaborava šifre, moguće je resetovati šifru, pomoću linka koji se šalje na mail korišćen tokom registrovanja. Korišćenjem *dinamičkih linkova*, korisnik se upućuje u aplikaciji, gde bira svoju novu šifru.



Slika 3. Autentikacija u aplikaciji.

Realtime baza podataka se koristi u slučaju kada treba osvežiti neki podatak istog trenutka kada se podatak promeni u toj bazi podataka. Ovaj rezultat se postiže odgovarajućom primenom *event listenera* koju ova baza podataka nudi.

Firestore je baza podataka koja je jednostavnija za korišćenje i predstavlja NoSQL bazu podataka. U *Firestore* se smeštaju ogromne količine postova, podataka korisnika, novih novosti i još mnogo stvari. Za razliku od *Realtime* baze podataka, *Firestore* funkcioniše na osnovu kolekcija, što mu daje na organizaciji podataka. U svakoj kolekciji može biti neka nova kolekcija, ili neki dokument, u kojem se nalaze različita polja koja čuvaju podatke.

U slučaju kada je potrebno sačuvati sliku sa neke novosti, ili čuvanje profilnih fotografija, odnosno avatara, koristi se *Storage*. *Storage* se nalazi na serveru *Firebase*, i pristupa se, u ovoj aplikaciji, za čitanje fotografija, ali mogu se smeštati i bilo koji fajlovi.

Po ovim sposobnostima *Firebase*, dostiže se pravi osećaj društvene mreže. Svaka novost koja stigne, kao i svaka reakcija na novost (slika 4) se čuva u *Firestore*, gde se takođe čuvaju i svi podaci vezani za korisnike. Svaka novost ima svoj jedinstveni ID broj, i svaki *Post* zajednice mora da ima u bazi podataka ID broj novosti, kako bi mogao da se poveže za tu novost, a i zbog bolje organizovanosti podataka. Svakom korisniku koji je kreiran u aplikaciji preko *Firebase Autentikacije*, odmah se njegovi podaci smeštaju u *Firestore* bazu podataka, kako bi se njegovim podacima kasnije lakše pristupalo. Svaki korisnik svoje avatare čuva u *Storage*. Kada se kreira neki korisnik, automatski ima ikonicu korisnika umesto avatara. Ta ikonica se ne postavlja na *Storage*, kako bi se uštedela memorija. Izuzetak je kada se korisnik prijavi putem Google naloga ili Facebook naloga. Tada mu se preuzima profilna fotografija, i postavlja na *Storage*, kao njegov avatar. Svaka poruka koja se pošalje u aplikaciji, obavlja se preko *Realtime* baze podataka, koja odmah automatski

obaveštava korisnika kome je poruka poslata. Isto preko *Realtime* baze podataka, ostvaruje se indikator *Seen*, odnosno da li je poruka pregledana od druge strane.



Slika 4. Interfejs aplikacije.

Svako korišćenje *Firebase* u aplikaciji je asinhrono, odnosno ne obavlja se istog trenutka kada se ukaže na komandu. Potrebno je dobro sinhronizovati komande, kako bi došlo do željenog rezultata. Do smanjenja ovog problema, a i lepše preglednosti aplikacije, uvodi se u program indikator koji ukazuje da se neki podaci učitavaju, i stranica ne pokazuje svoje komponente dok se ne učitaju podaci.

C. Native Moduli

Nekada je neke funkcionalnosti znatno teže ili nemoguće ostvariti u *React Native-u*, a neophodni su za ostvarivanje cilja aplikacije, ili znatno bolje efikasnosti. Tada nastupaju *Native Moduli*, koji preko *React Bridge-a* se povezuju sa *React Native* aplikacijom i obavljaju različite funkcionalnosti. Kodovi *Native Modula* mogu biti pisani u Javi za *Android*, ili u *Swiftu* ili Objektivnom C za *iOS*.

U ovoj aplikaciji se koriste *Native Moduli* pisani u programskom jeziku *Java*, čiji je kod pisan u *Android Studio*. Izrađeni su *Native Moduli* za specifične REST API zahteve za preuzimanje novosti sa besplatnog *NewsCatcher* API servisa, kao i za lokalnu *SQLite* bazu podataka.

REST API zahtevi vraćaju novost u obliku JSON objekta, koji se odgovarajućim parse-ovanjem dovodi u oblik pogodan za čitanje i upravljanje tokom izrade aplikacije. Novosti koje su dobijene ovim zahtevima se smeštaju u *Firestore* bazu podataka, sa posebnim ID brojem.

SQLite je lokalna baza podataka koja se kreira i čuva u svakom uređaju na kome je aplikacija instalirana, i briše se automatski, brisanjem aplikacije. Ona je pogodna za čuvanje ID-a korisnika, kome onda može da se pristupi u

svakom trenutku u kome je potrebno. ID korisnika sačuvanog u SQLite bazi podataka odgovara ID-u u kome je korisnik sačuvan u *Firestore* bazi podataka. Prijavom, ili registrovanjem korisnika u aplikaciji, u ovoj bazi podataka koja ima samo jednu tabelu, pridružuje se ID korisnika. Ta tabela je kreirana instaliranjem aplikacije. Kasnije odjavom se brišu svi podaci koji su bili u tabeli.

Native Moduli se obavljaju asinhrono, i samim tim svaki poziv neke funkcije iz *Native* Modula, mora imati odgovarajuću *callback* funkciju, koja će vratiti podatke koji su potrebni, ili obaviti neku stvar koja je potrebna.

D. Ostale biblioteke

Prilikom izrade ove aplikacije korišćeno je dosta dodatnih biblioteka kako bi se olakšao razvoj, a i pojednostavio kod za čitanje i ubuduće prepravljavanje i doradivanje.

Gifted-chat se koristi kako bi se olakšao dizajn za razmenu poruka, odnosno chat. Nudi dosta funkcija koje se lako uče i mogu dosta da doprinesu razvoju chata, kao što su da li je poruka pregledana od druge strane, ili moderan dizajn.

Image-picker koji se koristi kako bi se prilikom promene profilne fotografije odabrala odgovarajuća fotografija sa mobilnog uređaja. Ta fotografija se potom postavlja na *Storage Firebase* servera.

Safe-area-context služi za pravilno raspoređivanje margina i padding-a na JSX elemente za bolje iskustvo korisnika. Ukoliko kamera zauzima mali kružić kod gornje ivice ekrana, ova biblioteka sprečava da neki element bude tu.

Google-Signin i *Fbsdk-next* pomažu prilikom prijavljivanja korisnika preko *Google* ili *Facebook* naloga.

Omogućavaju njihovo implementiranje i pomažu kod kreiranja dizajna za prijavljivanje.

Options-menu znatno pomaže kod kreiranja posebnog menija za svaki post. Kombinacijom ikonice, JSX elemenata i ovog menija, dobija se moderan dizajn menija.

III. ZAKLJUČAK

Aplikacija je zabavnog i edukativnog karaktera. Mišljenja mnogih ljudi se nalazi u aplikaciji, i nudi mogućnost časkanja o svim glavnim globalnim i lokalnim dešavanjima. Veoma je jednostavna za korišćenje i omogućava prijavljivanje sa *Google* ili *Facebook* naloga, pa nije čak potrebno ni kreiranje novog naloga.

U bliskoj budućnosti je moguće usavršavanje aplikacije za proširivanje funkcionalnosti i daljeg olakšavanja korišćenja.

ZAHVALNICA

Autor se zahvaljuje asistentu Nikoli Mitroviću na usmeravanju i sugestijama, kao i pruženoj pomoći pri radu.

LITERATURA

- [1] React Native, dostupno na: <https://reactnative.dev/>
- [2] Node package manager, dostupno na: <https://www.npmjs.com/>
- [3] Android Studio, dostupno na:
<https://developer.android.com/studio>
- [4] Firebase platforma, dostupno na: <https://firebase.google.com/>
- [5] Source kod projekta
<https://github.com/djordjemitrovic00/Newsy>
- [6] Visual Studio Code, dostupno na:
<https://code.visualstudio.com/>

Razvoj aplikacije za optimizaciju položaja pokretnih solarnih panela

Danilo Strahinović

Sadržaj - Ovaj rad predstavlja realizaciju rotirajućeg fotonaponskog sistema kao i internet aplikaciju za upravljanje tim sistemom. Dat je i detaljan opis komponenata koje su korišćene u ovom sistemu. Ovakav sistem ima prednost u odnosu na osnovni sistem sa analognim senzorima jer rad sistema ne zavisi od klimatskih uslova u okolini. Takođe su dati eksperimentalni rezultati koji ukazuju na mogućnost unapređenja projekta.

I. TEORIJSKI UVOD

Moderni solarni sistemi se sastoje od solarnih panela koji su pričvršćeni na pokretnu strukturu koja se pomera u zavisnosti od položaja sunca. Ovi sistemi su znatno napredniji od standardnih solarnih sistema gde su paneli fiksirani i nepokretni [1]. Energetska efikasnost solarnih sistema sa praćenjem položaja sunca je znatno veća, čak i do 40 %, što predstavlja značajan korak u unapređenju eksploatacije solarne energije [2]. Zbog dostupnosti interneta i pojave pametnih uređaja kreirana je internet aplikacija radi lakšeg upravljanja sistemom za solarno praćenje.

II. ODABIR KOMPONENATA

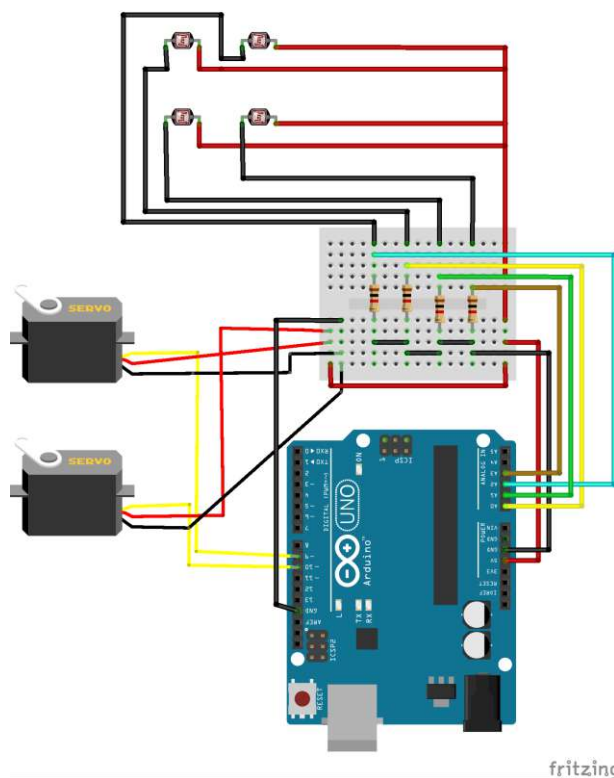
Cilj ovog rada je projektovanje i realizacija rotirajućeg fotonaponskog sistema koji će na osnovu vrednosti analognih senzora slati podatke logičkoj jedinici, koja će na osnovu obrađenih podataka pozicionirati solarni panel. Pre svega, glavni elementi sistema za solarno praćenje su algoritam za praćenje sunca, kontrolna jedinica, sistem za pozicioniranje, pogonski mehanizam, senzorski uređaji i granični prekidači.

Analogne senzore u ovoj realizaciji predstavljaju četiri fotootpornika koji su pozicionirani na strukturi na kojoj je i solarni panel. Ovo omogućava da ugao upadnih zraka bude uvek 90 stepeni što dalje dovodi do znatnog povećanja efikasnosti. Vrlo su i pogodni za korišćenje i očitavanje analognih vrednosti na osnovu kojih se vrši pozicioniranje panela. Logičku jedinicu ovog sistema predstavlja okruženje *Arduino Uno* [3]. Ovako okruženje je izabrano zbog širokog spektra mogućnosti koje nudi svojim korisnicima. Sistem za pozicioniranje ovog sistema čine dva servo motora SG90 [4]. Nivo od 5 V je dovoljan za napajanje oba motora, a budući da solarni panel na svojim izlazima daje napon od 6

volti, ovaj sistem možemo smatrati efikasnim. Servo motori iskorišćeni u ovoj realizaciji imaju mogućnost okretanja od 0 do 180 stepeni. ESP8266 je jeftin Wi-Fi mikročip, sa punim TCP/IP stekom i mogućnostima mikrokontrolera, koji proizvodi *Espressif Systems* u Šangaju [4]. Ovaj mali modul omogućava mikrokontrolerima da se povežu na Wi-Fi mrežu i uspostave jednostavne TCP/IP veze.

III. REALIZACIJA HARDVERA

Za praktičnu realizaciju nam je potrebno četiri fotootpornika, četiri otpornika od 1 k Ω , dva servo motora, logička jedinica, uređaj za povezivanje logičke jedinice na internet i naravno proto ploča i nekoliko žica. Cilj rada je projektovati i realizovati rotirajući fotonaponski sistem sa analognim senzorima koji će pozicionirati solarni panel tako da ugao upadnih zraka bude 90 stepeni. Povezivanje ovog sistema izgleda kao na slici 1.

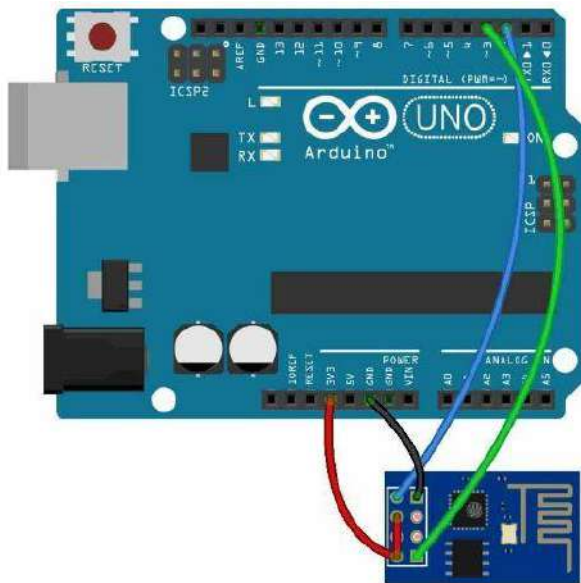


Slika 1. Električna šema sistema.

D. Strahinović je student na modulu Elektronika i mikrosistemi, Elektronski fakultet, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: danilo.strahinovic@protonmail.com

Četiri fotootpornika su povezana preko otpornika na *Arduino* okruženje. Oni čine senzorski deo na osnovu čijih analognih vrednosti će *Arduino* pozicionirati servo motore. Jedan servo motor se pozicionira kao horizontalni, a drugi kao vertikalni tako da ovaj sistem zapravo predstavlja rotirajući fotonaponski sistem sa dve ose. Servo motori su fiksirani na strukturi koja drži solarni panel. Fotootpornici su povezani na pinove A0, A1, A2 i A3, dok su servo motori povezani na pinove 10 i 9.

Za povezivanje sistema na internet korišćen je *Wi-Fi ESP8266* modul. Povezivanje sa *Arduino* okruženjem je prikazano na slici 2.



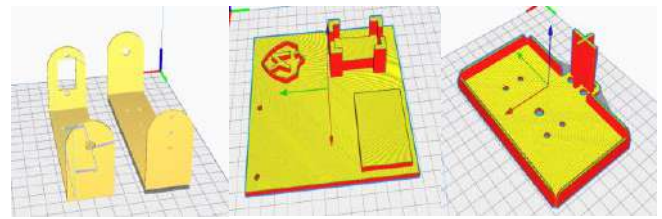
Slika 2. Povezivanje Wi-Fi modula sa *Arduino* okruženjem.

TX pin modula se vezuje na D2 pin *Arduino* okruženja, VCC pin modula se kratkospoji sa CH-EN (*Chip enable*) pinom, zato što zajedno idu na 3.3 V pin *Arduino* okruženja. GND naravno na GND pin. RX pin se vezuje između redne veze otpornika od 1 i 2 k Ω . Drugi krajevi otpornika idu na GND (2K) i na D3 (1K) pinove.

Nakon povezivanja potrebno je konektovati sistem na internet. Zatim je potrebno da se konfiguriše internet server koji daje kompanija komponente. Mora se napomenuti da kompanija daje test server koji radi u lokalu (virtuelni server na personalnom računaru). Što se prenosa podataka tiče, pre svega *Arduino* okruženje pošalje odgovarajuće vrednosti kroz internet modul na njegov server. Zatim sa servera aplikacije se pošalje zahtev na server modula koji ako ima, prosleđuje te podatke. Nakon prihvatanja podataka na server strani aplikacije, isti se obrađuju i odgovarajuće vrednosti se šalju nazad na *Arduino* istim putem samo suprotnim smerom.

Realizacija projekta, s obzirom da se radi o fotonaponskom sistemu koji ima pokretne mehaničke elemente (servo motore), zahteva i realizaciju mehaničke konstrukcije na kojoj se postavlja hardver sistema. Svi

elementi mehaničke konstrukcije (slika 3) su projektovani u Ultimaker Cura [5], alatu za 3D modelovanje i napravljeni na 3D štampaču.



Slika 3. Projektovani i realizovani delovi mehaničke konstrukcije.

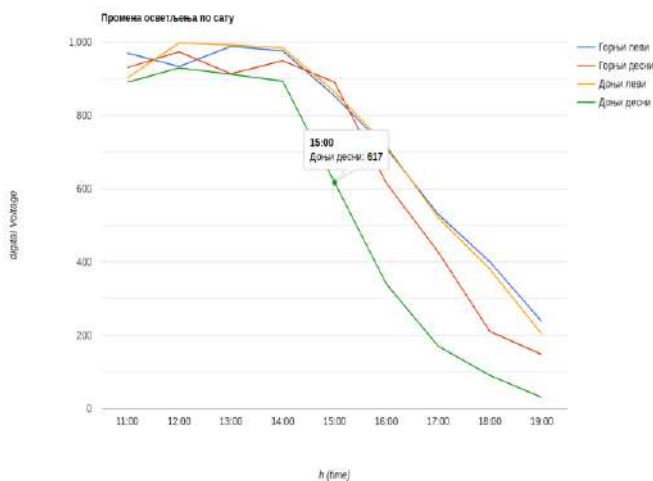
Nakon povezivanja projektovanog hardverskog dela, kao i kreiranja mehaničke konstrukcije koja je projektovana za isti, vrši se celokupno sklapanje projekta i finalna priprema za praktičan rad. Sklopljeni hardverski deo projekta izgleda kao na slici 4.



Slika 4. Izgled sklopljenog hardvera projekta.

IV. EKSPERIMENTALNE VREDNOSTI

Za proveru ispravnosti aplikacije, odnosno da aplikacija može da ima funkcionalnost kod dela za manuelnu optimizaciju, potrebna su merenja koja su pomenuta pri opisu aplikacije. Naravno zbog toga je izvršeno eksperimentalno merenje gde su prikupljeni podaci sa četiri fotootpornika o osvetljenju. Merenje je vršeno u jednom danu (30.09.2021.) i ti podaci izgledaju kao na slici 5.



Slika 5. Izmerene vrednosti fotootpornika u digitalnom obliku napona po satu.

Na slici 5. se vidi kako je napon u digitalnom obliku najveći u podne, kad i jeste najveća osvetljenost i da, kako sunce zalazi, se smanjuje i osvetljenost fotootpornika. Iz grafikona se može zaključiti da senzori rade i da aplikacija može koristiti podatke sa njih. Za kretanje sistema oko vertikalne ose, logička jedinica upoređuje vrednosti sa levih i desnih otpornika. Ukoliko je osvetljenost levih otpornika veća, sistem se kreće u levo, a ako su desni otpornici više osvetljeni, sistem se kreće u desno. Kada je reč o kretanju sistema oko horizontalne ose, logička jedinica upoređuje vrednosti gornjih i donjih otpornika. Ukoliko su gornji otpornici više osvetljeni, sistem se kreće na gore, a u obrnutom slučaju sistem se kreće na dole.

V. INTERNET APLIKACIJA

Aplikacija je pisana u dva programska jezika. U programskom jeziku PHP je obrađena pozadina aplikacije dok je u *Javascript* programskom jeziku pisan izgled interfejsa aplikacije [6]. PHP je izabran zbog jednostavnosti pisanja i pogodnosti za internet aplikacije a *Javascript* za kreiranje vidljivog interfejsa zbog njegovog dizajna i brzine rada. Kod za *Arduino* okruženje je pisan u programskom jeziku C. Za realizaciju aplikacije takođe je korišćen još jedan server koji daje docker paket [6]. Ovo je paket koji kreira virtuelni server na personalnom računaru i funkcioniše kao da je na internetu. *Source* fajlovi za

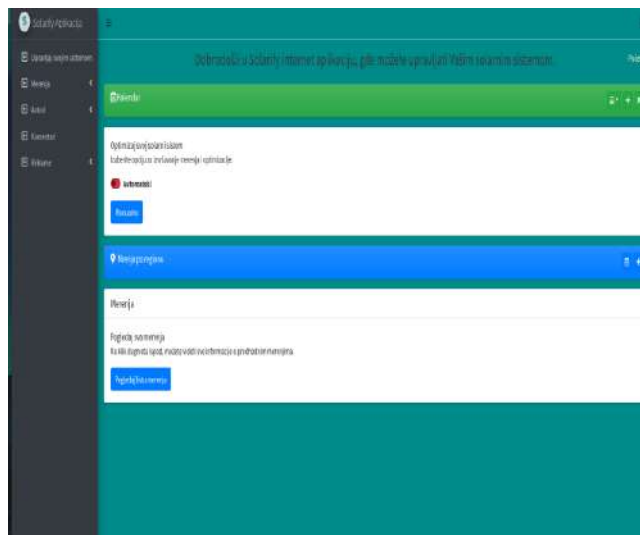
realizaciju aplikacije su dostupni preko platforme GitHub na profil autora, na linku: <https://github.com/ds2208/solar-system>.

Pre svega, aplikacija je zamišljena da ima široku primenu u industriji. Zbog velikog broja korisnika se vodilo računa o zaštiti podataka korisnika kao i njihovih sistema. S tim u obzir, u aplikaciji je implementiran deo za prijavljivanje korisnika (slika 6) po svim trenutnim internet standardima.



Slika 6. Autentifikacija korisnika.

Samo ako je korisnik registrovan, odnosno ako postoje njegovi podaci u bazi će moći da se uloguje. Nakon logovanja interfejs aplikacije izgleda kao na slici 7.

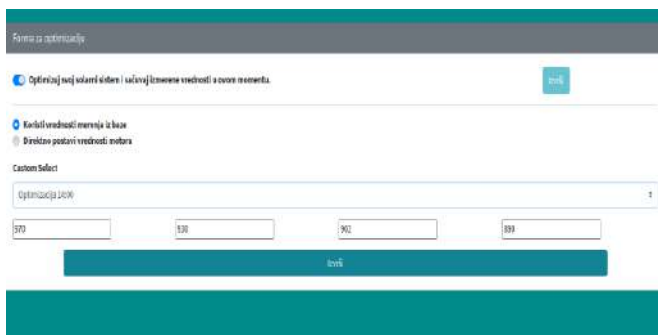


Slika 7. Interfejs aplikacije.

Na početnoj strani aplikacije postoje 3 kartice. Prva je kalendar gde je moguće zakazati optimizaciju kao događaj u budućnosti, to jest da se unapred podesi automatizacija sistema. Na trećoj se mogu izvući informacije o merenjima po regionima. Na četvrtoj klikom na dugme “pogledaj listu merenja” mogu se videti izlistana lista prethodnih merenja i optimizovanja solarnog sistema.

Druga kartica služi za optimizaciju položaja korisničkog solarnog sistema. Na ovoj kartici se nalaze dve komponente, od kojih je prvi prekidač koji služi za automatsko pokretanje optimizacije u aplikaciji. Klikom na prekidač, se aktivira ili deaktivira optimizovanje solarnog sistema koje se izvršava u realnom vremenu na određeni vremenski interval (u ovom slučaju na 1 sat). Ispod prekidača za automatsko uključivanje odnosno isključivanje automatizacije solarnog sistema, postoji dugme koje vodi na novu stranicu koja je zasnovana na manuelnom upravljanju to jest manuelnom optimizovanju sistema. Razlika je najviše u tome što je automatsko uključeno od trenutka aktivacije i izvršava se u određenom intervalu sve dok ga korisnik ne isključi, dok se manuelno izvršava u momentu aktivacije i ne nastavlja dalje sa radom.

Zbog raznolike primene klikom na dugme Manuelno otvara nam se novi interfejs na kome možemo birati vrste izvršenja. Na početku je prikazan jedan prekidač i dugme za okidanje. Ako je prekidač isključen dugme za okidanje će biti dozvoljeno i klikom na isto se okida komanda za optimizaciju solarnog sistema koje, kao što je ranije napomenuto, se izvršava sada u momentu. Klikom na prekidač u levom uglu, zabranjuje se okidanje desnog dugmeta i otvara se mogućnost biranja između dve forme. Prva forma (slika 8) služi za odabir starih vrednosti koje je korisnik sačuvao pri nekoj od prethodnih optimizacija.

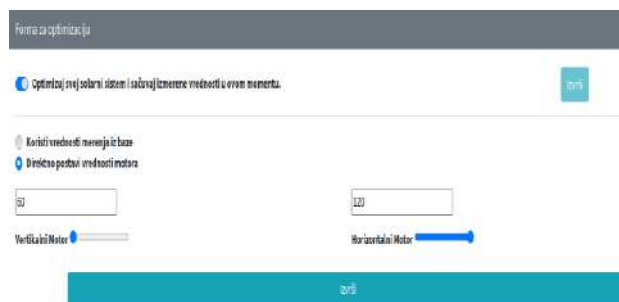


Slika 8. Manuelna optimizacija sa prethodno sačuvanim vrednostima.

S obzirom da projekat koristi četiri fotootpornika, kao senzorske elemente za osvetljenje postoje četiri polja za unošenje podataka koji predstavljaju vrednosti za svaki senzor ponaosob. Ovde se mogu uneti ti podaci ručno, ali takođe, s obzirom da korisnik verovatno ne zna napamet informacije o osvetljenju ima mogućnost da izabere parametre od prethodnih vrednosti i tako skрати posao. Ta mogućnost je data kroz polje za selektovanje jedne vrednosti gde su izlistane sve vrednosti. Klikom na veliko dugme

Izvrši, aplikacija će izračunati poziciju motora po unetim parametrima što znači da će manje opterećivati naš sistem koji će imati i manju potrošnju energije.

Promenom stanja radio dugmeta za promenu forme, na vrednost direktnog postavljanja vrednosti za servo motore, prethodna forma se zatvara a prikazuje se nova forma (slika 9) u kojoj korisnik samostalno može postaviti vrednosti za servo motore i u zavisnosti od tih vrednosti se motori pozicioniraju.



Slika 9. Direktno pozicioniranje motora.

VI. ZAKLJUČAK

U ovom radu je dat teorijski uvod o ideji realizacije, kao i teorijska osnova komponenti koje su bile potrebne. Praktično je realizovan sistem za solarno praćenje kao i internet aplikacija za upravljanje istim. Takođe je dat i prikaz merenih eksperimentalnih vrednosti analognih senzora.

ZAHVALNICA

Autor se zahvaljuje asistentu Nikoli Mitroviću na pomoći u radu i konstantnoj podršci, kao i profesoru dr Vesni Paunović na pomoći pri realizaciji.

LITERATURA

- [1] Prof. Dr. Dragan Pantić, Solarne komponente i sistemi, 2016.
- [2] Sistemi za dvoosno praćenje sunca, primena u solarnim i solarno termičkim sistemima, 2017, Dr Željko V. Despotović, dipl.el.inž Institut Mihalno Pupin, Univerzitet u Beogradu, Volgina 15, Beograd.
- [3] Tehnička dokumentacija za Arduino Uno. Dostupno na: <https://www.farnell.com/datasheets/1682209.pdf>
- [4] Tehnička dokumentacija za motor SG90. Dostupno na: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
- [5] Softverski alat Ultimaker Cura. Dostupno na: <https://ultimaker.com/software/ultimaker-cura>
- [6] Tehnologija Laravel. Dostupno na: <https://laravel.com/docs/8.x/configuration>

Ultrazvučni anemometar

Veljko Vučković

Sadržaj - U ovom radu predstavljeni su princip rada ultrazvučnog anemometra, uređaja za merenje brzine protoka vazduha primenom ultrazvučnih primopredajnika, i električno kolo realizovano za testiranje sistema.

I. ANEMOMETAR I ULTRAZVUČNI ANEMOMETAR

Anemometri [1] su uređaji koji služe za merenje brzine protoka vazduha u atmosferi, aerotunelima i drugim sistemima sa protokom gasova. Za merenje brzine vetra je u najširoj primeni električni anemometar sa rotirajućim čašicama [2].

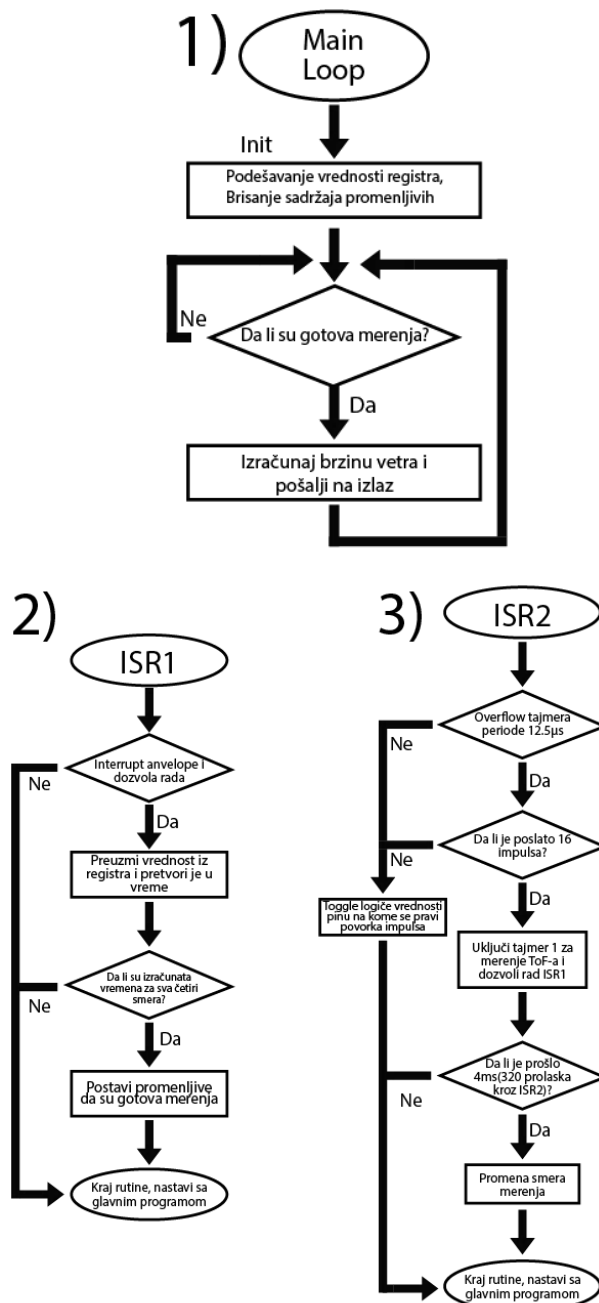
Ultrazvučni anemometri su razvijeni 1950.-ih godina, a koriste ultrazvučne talase za merenje brzine vetra [3]. Oni mere brzinu vetra na osnovu „vremena leta“ ultrazvučnih impulsa između parova ultrazvučnih primopredajnika. Kombinacijom parova ovih senzora se dobijaju merenja po jednoj, dve ili tri ose. Zbog odsustva pokretnih delova su odličan izbor za dugoročnu primenu u autonomnim mernim stanicama i na lokacijama gde atmosferski uslovi, prašina ili salinitet mogu da utiču na pouzdanost standardnih anemometra. Glavni nedostatak im je uticaj konstrukcije anemometra na rezultate merenja, koja zahteva korekcije dobijene merenjima u aerotunelu, zbog malih promena u položaju primopredajnika koje imaju veliki uticaj na kvalitet merenja zbog promene udaljenosti između primopredajnika i promene ugla koji zaklapaju upareni primopredajnici.

II. PRINCIP RADA

Ultrazvučni anemometar naizmenično šalje i prima ultrazvučne signale preko ultrazvučnih primopredajnika koji se nalaze u dve međusobno normalne ose kako bi se dobili dvodimenzionalni podaci o brzini i smeru vetra. Na jedan primopredajnik u paru se šalje niz od 16 impulsa frekvencije 40 kHz (rezonantna frekvencija primopredajnika), koji izazivaju reakciju, tj. generisanje električnog napona, na ostalim primopredajnicima. U svakom trenutku je aktivan samo jedan par primopredajnika, pa preslušavanje signala nema uticaj na merenja. Grubi signal se zatim sa prijemnog senzora šalje na pojačavač, detektor anvelope i detektor prolaska kroz nulu kako bi se moglo precizno odrediti vreme pojave značajnog signala koje se prostom računicom pretvara u brzinu vetra. Proces se ponavlja za oba pravca i s va četiri smera, u odnosu na četiri strane sveta, rezultati se

V. Vučković je diplomirani inženjer elektrotehnike i računarstva na modulu Elektronske komponente i mikrosistemi sa Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: vuckovicveljko98@gmail.com

obrađuju i dobijaju se podaci o apsolutnoj brzini vetra, koji se dalje prosleđuju na računar/displej/internet. Slika 1 prikazuje algoritam rada ultrazvučnog anemometra.



Slika 1. Algoritam rada ultrazvučnog anemometra - 1) Algoritam glavnog programa, 2) Algoritam prve interrupt rutine, 3) Algoritam druge interrupt rutine.

A. Digitalni blok

Rad digitalnog bloka kontrolišu dva signala iz mikrokontrolera koji se dovode na selekzione ulaze multiplexera i dekodera, a dekodera postavlja potrebne logičke nivoe na trostatičke bafere koji se nalaze između linija za PWM signale i primopredajnika. Pomoću ova dva signala se preko multiplexera istovremeno bira primopredajnik na koji je potrebno poslati povorku impulsa, kao i primopredajnik čiji je odziv potrebno obraditi, a dekodera omogućava da impulsi dolaze samo do jednog primopredajnika, čime se smanjuje preslušavanje.

B. Analogni blok

Analogni blok se sastoji iz pojačavača, detektora anvelope i detektora prolaska kroz nulu. Pojačavač se sastoji iz dva stepena izrađena od Darlingtonovih parova pri čemu svaki stepen ima LC rezonantno kolo u kolektorskoj grani čija je rezonantna frekvencija 40 kHz, kako bi se istovremeno dobilo veće pojačanje na ovoj frekvenciji, a i obezbedio se uski propusni opseg pojačavača. Kako bi se umanjio uticaj temperature na karakteristike pojačavača, koriste se dva dodatna Darlingtonova para kod kojih su baza i kolektor vezani. Izlazni signal iz pojačavača je superponiran na određeni jednosmerni nivo napona kako ne bi došlo do deformacije naizmeničnog signala sa primopredajnika. Ovaj signal se dalje prosleđuje na detektor anvelope koji stvara okidni impuls kada anvelopa dostigne značajnu vrednost, kao i na detektor prolaska kroz nulu, pomoću koga se meri vreme za koje se nakon okidanja detektora anvelope pojave 16 impulsa na izlazu primopredajnika, koji su potrebni za precizno računanje „vremena leta” signala između primopredajnika.

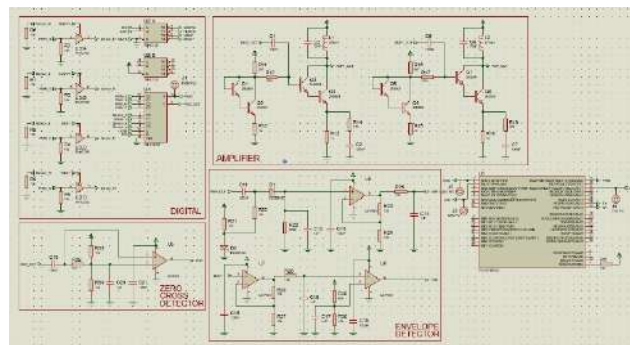


Slika 2. Fizička realizacija sistema, na slici se vide konstrukcija anemometra sa primopredajnicima van krakova anemometra i raster ploča sa zalemljenim komponentama (sredina) i ReadyForPic okruženje (levo).

III. ELEKTRIČNA ŠEMA

Električna šema(koju prikazuje Slika 3) se može podeliti u dva bloka – digitalin (Slika 4) i analogni (Slika 5, Slika 6,

Slika 7), gde se analogni blok dalje može podeliti u tri podbloka - blok pojačavača, blok detektora prolaska kroz nulu i blok detektora anvelope. Rad komponenti kontroliše mikrokontroler PIC18f45k22 kao deo razvojnog okruženja ReadyForPic.



Slika 3. Kompletna električna šema u programu Proteus(delovi šeme su uprošćeni zbog funkcionalnosti softverskog alata).

A. Digitalni blok

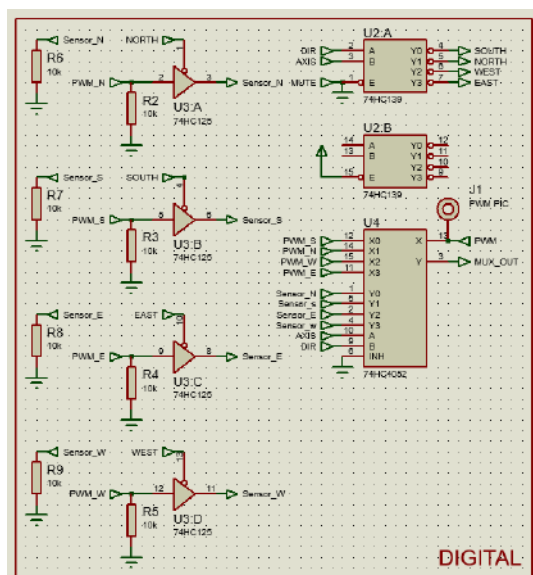
Ovaj deo kola (

Slika 4) određuje koji primopredajnici će biti aktivni, tj. preko multiplexera, dekodera i trostatičkih bafera se PWM signal prosleđuje do jednog primopredajnika u paru, a u isto vreme se drugi primopredajnik povezuje sa analognim blokom kako bi se izvršila obrada signala. Iz mikrokontrolera se signali za odabir pravca i smera prosleđuju na multiplexer i dekodera, a izlazom dekodera se aktiviraju odgovarajući baferi.

Multiplexer koji je primenjen u kolu je CD74HC4052E [4] firme Texas Instruments. Ovo kolo se sastoji iz para multiplexera od kojih se jedan koristi za slanje povorke impulsa, a drugi za prijem odziva sa odgovarajućeg primopredajnika.

Dekoder primenjen u kolu je 74HC139N [5] firme Philips Semiconductors. Ovo kolo ima „active high” ulaze, ali ima „active low” izlaze, na šta se mora obratiti pažnja tokom projektovanja ostatka kola.

Poslednja komponenta iskorišćena u digitalnom bloku je kolo 74HC125 [6], četverostruki trostatički bafer. Ovo kolo se u električnom kolu koristi za izolaciju primopredajnika od povorke impulsa kada se na njih ne šalje povorka, kao dodatna sigurnost protiv prisluškivanja. Ovo kolo ima „active low” ulaze, što u kombinaciji sa dekoderom koji ga kontroliše znači da će samo jedan bafer biti otvoren u svakom trenutku, što znači da je deo multiplexera koji kontroliše slanje povorke uvek povezan sa samo jednim primopredajnikom.



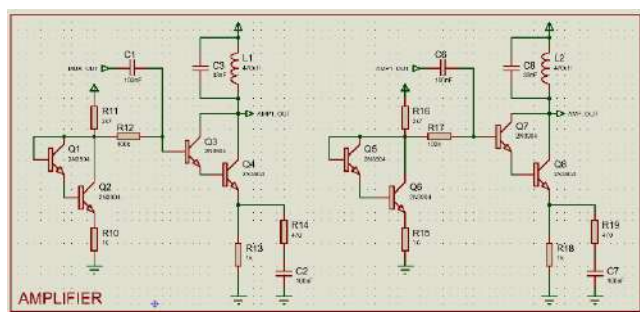
Slika 4. Digitalni blok električne šeme sistema.

B. Analogni blok

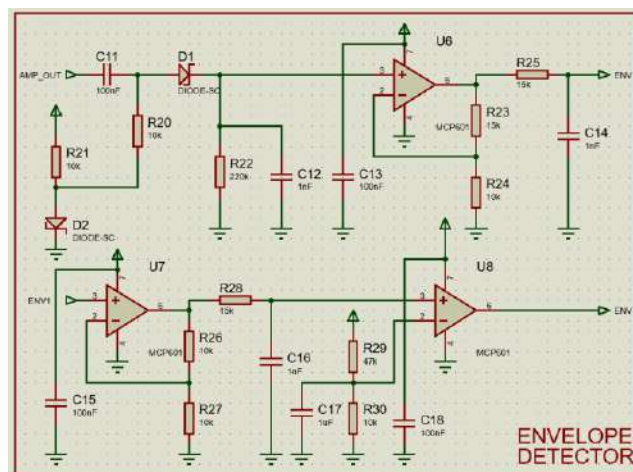
Detektor anvelope koristi aktivne niskopropusne filtre zajedno sa komparatorom kako bi se porast anvelope mogao konvertovati u okidni digitalni signal. Detektor prolaska kroz nulu predstavlja komparator izrađen od operacionog pojačavača.

Na frekvencijama iznad i ispod rezonantne frekvencije LC kola pojačavača pojačanje drastično opada. Zbog tolerancija komponenti i uticaja odstupanja kapacitivnosti i induktivnosti komponenata od idealnih vrednosti, potrebno je odabrati komponente sa veoma precizno određenim vrednostima kako bi se dobili optimalni rezultati.

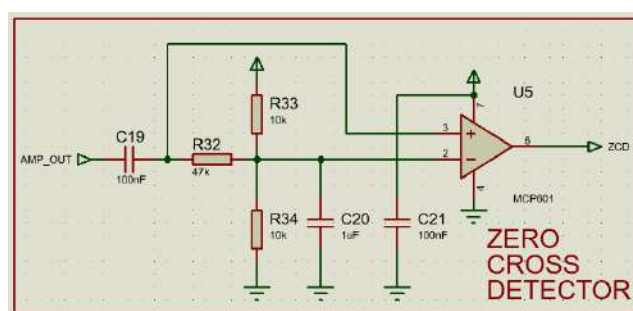
Detektor anvelope čine niskopropusni filteri i operacioni pojačavači kojima se kontroliše pojačavanje ovog dela kola kako bi se dobio željeni nivo anvelope. Na izlazu iz ovog dela kola se nalazi operacioni pojačavač u konfiguraciji komparatora koji stvara impuls od 5V kada nivo anvelope pređe predodređenu vrednost koja se smatra za početak prijema značajnog signala.



Slika 5. Blok pojačavača.



Slika 6. Detektor anvelope.



Slika 7. Detektor prolaska kroz nulu.

IV. PROGRAMSKI KOD

Kod za ovaj projekat je pisan u programskom jeziku BASIC, korišćenjem kompajlera Proton BASIC [7] koji konvertuje BASIC kod u asemblerski kod koji se preko programa mikroBootloader programira u mikrokontroler. Proton BASIC kompajler je napisao Les Džonson sa namerom da stvori prost i fleksibilan kompajler. Proton IDE (Integrated Development Environment) korišćen u realizaciji projekta predstavlja veoma kvalitetno okruženje gde je moguće pisati kod, ispravljati greške i kompajlirati kod u jednom zajedničkom Windows okruženju, a uz pomoć kompatibilnog softvera i programatora moguće je verifikovati programski kod i programirati željeni PICmicro uređaj. U nastavku rada će biti opisani samo važniji delovi koda. Kod se sastoji iz dela za inicijalizaciju, glavne petlje i dve rutine prekida.

A. Inicijalizacija

U ovom delu koda se vrše podešavanja registra tajmera [8], kontrolnih registara za prekidne rutine i deklarisanje promenljivih koje se koriste u programu. Definisani su takođe i parametri vezani za serijsku komunikaciju.

U projektu se koriste dva tajmera, jedan sa periodom od 12.5μs i jedan sa periodom od 2.048ms. Za prvi tajmer se

koristi „Timer4” modul koji postavlja na visoki naponski nivo određeni bit u određenom registru kada registar tajmera dostigne vrednost upisanu u „Period” registar PR4. Ovaj tajmer se koristi za generisanje povorke impulsa koja se šalje na ultrazvučne primopredajnike, kao i za interno brojanje vremena za kontrolu promene smera merenja brzine protoka vazduha. Za drugi tajmer se koristi „Timer1” modul koji kao takt koristi takt sistema. Ovaj tajmer se koristi za merenje vremena leta signala između primopredajnika. Očekivano vreme leta je reda par stotina mikrosekundi, dok je perioda ovog tajmera reda veličine milisekunde, što znači da je prekoračenje maksimalne vrednosti registra tajmera za potrebe normalnog rada sistema nemoguće.

U ovom delu koda se vrše i podešavanja vezana za prekidne rutine. Podešavaju se potrebne vrednosti u registrima vezane za prekid usled promene stanja na pinovima RB0 i RB1, kao i potrebne vrednosti u registrima za postavljanje prioriteta prekidnih rutina.

Pored ovoga se u ovom delu koda nalaze i definicije ulaznih i izlaznih portova.

B. Glavna petlja

U glavnoj petlji koda se vrši korekcija brzine zvuka u zavisnosti od temperature okoline, kao i računica brzine vetra u oba pravca(sever - jug i zapad - istok).

Na početku petlje se vrši korekcija brzine zvuka, zatim se poziva procedura za računicu brzine vetra, ukoliko su završena merenja za svaki smer. Ukoliko je dozvoljena računica, prvo se deaktiviraju prekidne funkcije mikrokontrolera(kako bi se izbeglo neočekivano ponašanje sistema, na primer pokušaj ulaska u jednu prekidnu rutinu dok je druga aktivna), vrši se oduzimanje vremena leta signala u oba pravca(zbog funkcije kojom se vrši računica), poziva se procedura za računicu brzine vetra(proceduri se prosleđuju razlike u vremenu leta za oba smera), vrši se ispis izračunatih brzina preko USB veze ostvarene između računara i *ReadyForPic* okruženja, omogućuju se prekidne funkcije mikrokontrolera i vrednost promenljivih kojima se dozvoljava računica se rača na nulu.

C. Prekidne rutine

U ovom projektu postoje dve prekidne rutine, ISR1 i ISR2. ISR1 je rutina u kojoj se obrađuju prekidna rutina detektora anvelope i prekidna rutina detektora prolaska kroz nulu, a odabir potrebnog dela koda unutar rutine se vrši ispitivanjem signala koji je izazvao prekid, tj. ako je došlo do ulaska u rutinu ISR1 usled okidanja detektora anvelope, izvršiće se jedan blok koda unutar rutine, a ukoliko je došlo do okidanja detektora prolaska kroz nulu, izvršiće se drugi blok koda. Ukoliko je signal iz detektora anvelope izazvao prekid, vrši se čitanje sadržaja registra tajmera 1 kako bi se moglo odrediti kada je došlo do okidanja detektora anvelope(pojava značajnog signala) i smeštanje očitane vrednosti u odgovarajuću promenljivu, nakon čega se pin B.1 postavlja kao ulazni, kako bi se omogućio prijem signala

sa detektora prolaska kroz nulu, uz određenu promenu sadržaja odgovarajućeg registra za kontrolu prekidnih funkcija.

U slučaju pojave signala sa detektora prolaska kroz nulu, druga prekidna rutina broji određen broj impulse, očitava vrednost iz registra tajmera 1, zatim tu vrednost dodaje prethodno dobijenom sadržaju registra, od novodobijene vrednosti oduzima vrednost ekvivalentnu idealnom vremenu slanja 32 impulsa i vrši konverziju konačne vrednosti u sekunde. Ova poslednja izračunata vrednost se prosleđuje proceduri za računicu brzine vetra.

D. Procedura za izračunavanje brzine vetra

Proceduri za računicu se prosleđuju razlike u vremenu leta za pravce sever - jug i zapad - istok. Ukoliko je ova razlika jednaka nuli, izbegava se računica jer bi u tom slučaju došlo do deljenja sa nulom. Ukoliko je ova razlika različita od nule, rešava se kvadratna jednačina $av^2+bv+c=0$, gde „a” predstavlja razliku u vremenu leta, „b” predstavlja dvostruku vrednost rastojanja između ultrazvučnih primopredajnika, a „c” predstavlja proizvod kvadrata brzine zvuka i razlike u vremenu leta.

V. ZAKLJUČAK

U ovom radu su opisani princip rada i električna šema prostog i preciznog ultrazvučnog anemometra. Uparivanje ultrazvučnih primopredajnika i precizno podešavanje rezonantne frekvencije LC kola pojačavača predstavljaju izazove u fizičkoj realizaciji ovog sistema. Zbog postojanja različitih analognih podblokova moguće je izmeniti parametre anemometra kako bi se dobilo veće pojačanje ili ranije okidanje detektora anvelope, a moguće je i izdvojiti određene signale za druge primene. Samo električno kolo zauzima malo mesta na štampanoj ploči, pa je moguća laka integracija u već postojeće sisteme.

LITERATURA

- [1] Anemometar. Dostupno na: <https://www.britannica.com/technology/anemometer>
- [2] Anemometar. Dostupno na: <https://www.britannica.com/technology/revolving-cup-electric-anemometer>
- [3] Anemometar. Dostupno na: <https://en.wikipedia.org/wiki/Anemometer>
- [4] CDx4HC405x, CDx4HCT405x High-Speed CMOS Logic Analog Multiplexers and Demultiplexers, Texas Instruments
- [5] 74HC/HCT139 Dual 2-to-4 line decoder/demultiplexer, Philips Semiconductors
- [6] M74HC125 QUAD BUS BUFFER (3-STATE), STMicroelectronics
- [7] Proton PIC Basic Compiler, A Powerful BASIC Compiler for 8-bit 12F, 16F and 18F PIC Devices, Les Johnson
- [8] PIC18(L)F2X/4XK22, 28/40/44-Pin, Low-Power, High-Performance Microcontrollers with XLP Technology, Microchip Technology Incorporated

Projektovanje sistema za kontrolu dnevne svetlosti prostorije u pametnim kućama

Milan Đorđević, Nikola Mitrović

Sadržaj - U ovom radu prikazano je projektovanje sistema za daljinsku kontrolu osvetljenosti prostorije pomeranjem pokretnih roletni na prozorima. Prikazano je projektovanje hardverskog dela sistema koje uključuje opis i spregu korišćenih komponenta, kao i projektovanje softverskog dela sistema, što podrazumeva projektovanje odgovarajuće aplikacije. Kao programski jezik za pisanje iskorišćen je *JavaScript*, kako za pisanje koda za programiranje mikrokontrolera sa Wi-Fi modulom koji je centralni deo sistema, tako i za pisanje veb aplikacije. Sistem je praktično realizovan.

I. UVOD

Moderne tendencije koje idu ka smanjenju potrošnje energije uređaja zahtevaju projektovanje sistema koji mogu iskoristiti što više prirodnih resursa [1]. Najuočljiviji primer ovakvog pristupa su solarni paneli i moduli montirani na kuće koji obezbeđuju električnu energiju iz okoline. Pored navedene fotoelektrične konverzije, postoje i drugi načini kako je moguće iskoristiti dnevnu svetlost i na taj način smanjiti potrošnju energije. Potrebno je razlikovati termine *energy efficient* (energetski efikasno) i *energy saving* (štednja energije). Razliku između ovih termina najjednostavnije je moguće prikazati na primeru sijalice. Pojam energetski efikasan odnosi se na to da sijalica troši što je moguće manje energije dok svetli. Sa druge strane, štednja energije bi se odnosila na to da li sijalica što je moguće manje uopšte svetli. Baš iz definicije ovog drugog pojma izvire jedan drugi primer upotrebe dnevne svetlosti, a to je za regulisanje osvetljenosti prostorija promenom ugla roletne. Neretko se dešava da roletne nisu pokretne, pa je u nekim situacijama potrebno do kraja zatvoriti roletne i uključiti svetlo, što dovodi do potrošnje energije. Ovo se pogotovo odnosi na noviju arhitekturu gde je tendencija da što veći deo zidova bude od stakla. U ovom radu biće detaljno prikazana realizacija sistema koji reguliše ugao pokretnih roletni korišćenjem projektovane veb aplikacije i na taj način reguliše količinu dnevne svetlosti koja dolazi do prostorije. U poglavlju II će biti opisane korišćene komponente i alati, a u poglavlju III će biti dat prikaz principa rada celokupnog sistema.

M. Đorđević je student na Akademiji Tehničko Vaspitačkih Strukovnih Studija – Odsek Niš, Aleksandra Medvedeva 20, 18000 Niš, Srbija, E-mail: milan94srb@gmail.com

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

II. KORIŠĆENE KOMPONENTE I ALATI

Za realizaciju sistema, potrebno je da sistem može pristupiti Wi-Fi mreži, kako bi mogao da primi podatke poslate sa veb aplikacije. Takođe, sistem bi trebalo da sadrži odgovarajući motor koji vrši pomeranje roletni ili venecijanera.

A. ESP32-WROOM2

Sistem pristupa mreži preko komponente koja je ujedno i centralni deo sistema, a to je mikrokontroler sa Wi-Fi modulom, konkretno ESP32-WROOM2 [2]. Ovaj mikrokontroler (MCU) pored mogućnosti pristupa Wi-Fi mrežama poseduje i veliki broj pinova i drugih perifernih jedinica koje omogućavaju jako široku primenu ovog MCU. Čip pristupa standardnim Wi-Fi mrežama na 2.4 GHz preko 802.11 b/g/n protokola, ali može komunicirati i korišćenjem drugih RF principa (*Bluetooth*, *Ethernet* itd.). ESP32 predstavlja proširenje osnovnog Wi-Fi modula ESP12, a koji je nastao kao unapređenje prvog komercijalno široko rasprostranjenog Wi-Fi modula ESP8266. Konkretno korišćen modul prikazan je na slici 1.

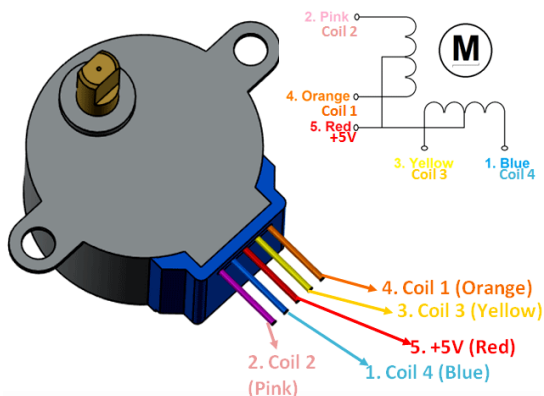


Slika 1. Korišćen modul sa ESP-WROOM2 čipom.

Kod ovog modula, pored samog ESP32 čipa, na ploči je smešteno i CP2102 kolo (konverter UART komunikacije na USB komunikaciju) koje omogućava komunikaciju sa računarom, gde se kao posrednik koristi aplikacija za terminal. Takođe ovo kolo omogućava spuštanje koda na čip (*MCU flashing*), kao i mnoge druge prateće komponente (otpornici, kondenzatori, LED i slično). Modul poseduje 38 pinova, od kojih korisnik može koristiti 28 (10 je sistemskih). Radni napon ovog modula je 3.3 V, ali može da se napaja i sa 5 V preko USB mikro kabla, jer je na ploči povezan AMS3V3 regulator.

B. Motor 28-BYJ48

Korišćeni motor 28-BYJ48 je step motor (koračni motor) koji predstavlja jedan od najzastupljenijih step motora [3]. Motor je prikazan na slici. 2.



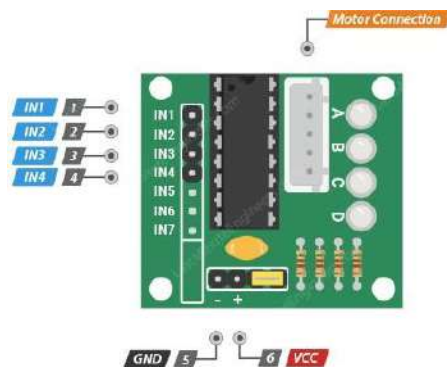
Slika 2. Korišćeni step motor 28-BYJ48 sa rasporedom pinova.

Kao i svi step motori, najveća prednost mu je mogućnost podešavanja pozicije rotora. Motor se sastoji od 4 spregnuta kalem, za koje je napon napajanja i kontrole 5 V, pa se mogu bez naponskih konvertera koristiti sa velikim brojem mikrokontrolera. Kao što je prikazano na slici 2, motor ima pet pinova.

Srednji pin (označen brojem 5) je VCC pin koji je potrebno povezati na napon napajanja. Ostala 4 pina odnose se na kalemov, koji su sastavni delovi motora. Kalemovi su spregnuti, kao na Sl. 2.

Svaki od preostala četiri pina potrebno je povezati na pinove MCU. Postavljanjem nekog od ta četiri pina u stanje *low*, stvara se potencijalna razlika između pina VCC i odgovarajućeg pina, pa dolazi do protoka struje. Time se aktivira odgovarajuće induktivno opterećenje, samim tim i na odgovarajuć način se pokreće motor. Motor se zove koračni jer se kretanje vrši u koracima.

Za pokretanje ovog motora potrebno je da struja prođe kroz kalemov, koji pokreću rotor. Struja koja se može generisati na većini MCU nije dovoljna za uključivanje motora. Zbog toga, u kolima sa ovakvim motorima se gotovo uvek koristi odgovarajuće kolo za kontrolu (drajver). Konkretno drajver iskorišćen za realizaciju ovog projekta je ULN2003, povezan na modul za kontrolu motora [4]. Ovo kolo predstavlja standardno drajversko kolo za motore. Sačinjava ga 7 Darlingtonovih parova koji omogućavaju da izlazna struja na izlaznim pinovima bude do 500 mA, što je sasvim dovoljno za pokretanje ovog motora. Kolo ULN2003 se najčešće montira u modul za kontrolu motora, na koji su povezane prateće komponente koje olakšavaju povezivanje (odgovarajući konektori, LED diode koje su uključene kada je neki od pinova aktivan, džamperi i slično). Ovaj modul prikazan je na slici 3.



Slika 3. Korišćeni modul sa ULN2003 drajverskim kolum.

Modul ima 4 pina koji služe za povezivanje sa 4 pina korišćenog MCU i 4 pina za povezivanje sa pinovima 28-BYJ48 step motora. Snabdeven je sa odgovarajućim otpornicima koji ograničavaju struju koja se isporučuje pinovima motora i odgovarajućim LED koje pokazuju koji od delova motora je trenutno aktivan.

C. Veb aplikacija

Projektovanje veb aplikacije obuhvata projektovanje servera (aplikacije na server koja treba da obezbedi klijent-server komunikaciju) i obezbeđivanje komunikacije sa bazom podataka. Za projektovanje servera iskorišćen je *node.js* [5]. *Node.js* je okruženje otvorenog tipa (open-source) za projektovanje servera. Ekstenzija *.js* upućuje na to da se koristi sintaksa programskog jezika *JavaScript*. Unutar *Node.js* okruženja iskorišćen je paket *Express*, za podizanje servera i paket *Mongoose* za komunikaciju servera i baze. Za kreiranje čitavog projekta kao tekstualni editor korišćen je *Visual Studio Code* [6].

Projektovana aplikacija ima mogućnost komunikacije sa ESP32 modulom korišćenjem *WebSocket* tehnike [7]. Ova tehnika omogućava povezivanje i komunikaciju tačno jednog klijenta i servera u mrežu. Kanal za komunikaciju je stalno otvoren i postoji protok podataka koji ne zavisi od http zahteva. Ovakva tehnika omogućava komunikaciju koja nije vremenski zavisna, već može biti zavisna od događaja. Interfejs projektovane aplikacije prikazan je na slici 4.



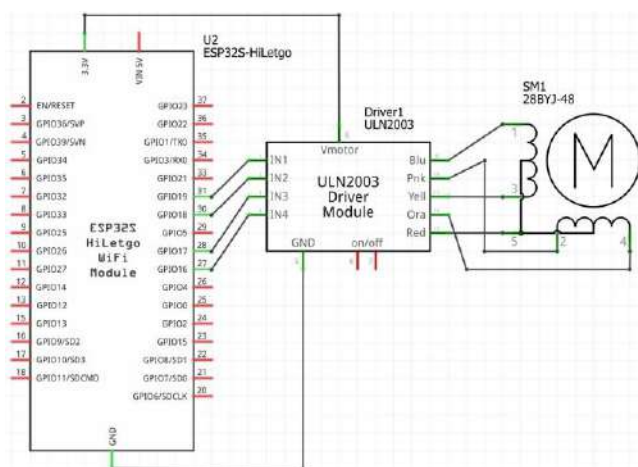
Slika 4. Interfejs projektovane aplikacije.

Aplikacija sadrži neke estetske i funkcionalne elemente. Akcenat ovog rada je na funkcionalnim elementima, dok se estetski elementi mogu detaljnije prilagođavati konkretnijoj primeni. Glavni funkcionalni deo aplikacije je slajder sa trakom koji se nalazi na sredini interfejsa i koji određuje poziciju step motora a samim tim i poziciju rotirajuće roletne na prozorima. Pomeranjem slajdera ulevo po traci ide se ka stanju TZ (totalno zatvorene roletne), a pomeranjem slajdera po traci udesno ide se ka stanju TO (totalno otvorene roletne). Ispod slajdera se nalazi polje koje prikazuje trenutno poziciju slajdera.

Nakon projektovanja servera, potrebno je izvršiti postavljanje i aktiviranje servera, tzv. *deployment*. Ovo je neophodan korak kako bi kreirani softver mogao da bude pristupačan na internetu. Za deployment servera iskorišćena je platforma *Heroku* [8]. Naravno, u realnoj primeni, za ovaj zadatak bio bi odabran odgovarajući hosting, koji bi omogućio prisutnost servera na internetu. U demonstrativne svrhe, za ovu platformu trenutno je iskorišćena *Heroku* platforma. Za prikaz platforme iskorišćen je jezik *html*, a za pojedine elemente prikaza iskorišćen je ponovo *JavaScript*. Veb aplikacija je dostupna na linku: <https://ieeestec-smart-blind.herokuapp.com/>

III. PRINCIP RADA

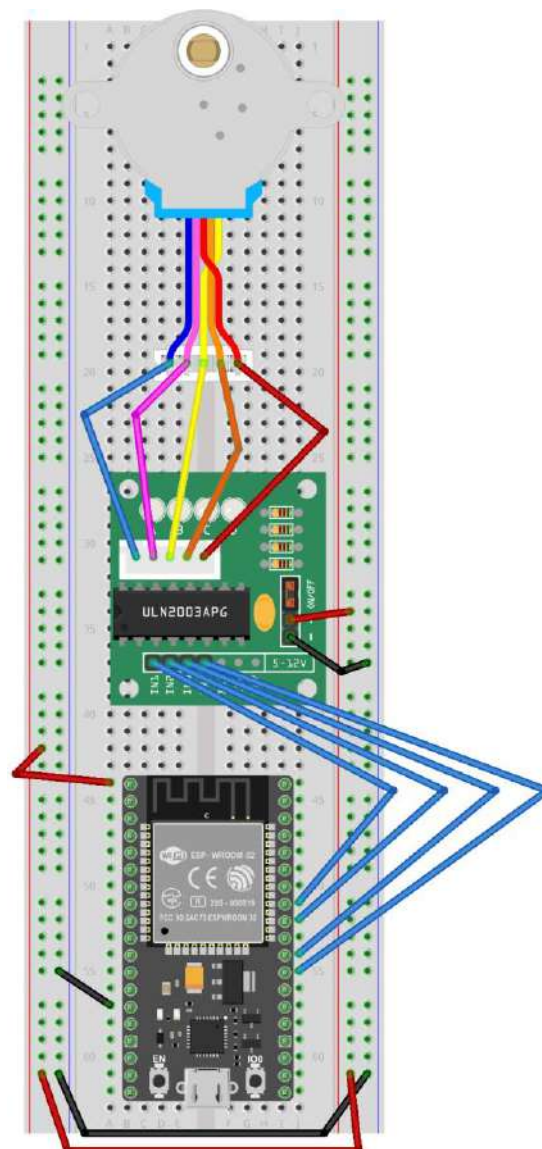
U ovom poglavlju biće detaljnije opisan princip rada MCU i njegova komunikacija sa veb aplikacijom. Šema sistema je prikazana na slici 5.



Slika 5. Šema projektovanog sistema.

ESP32 modul je povezan sa modulom za kontrolu motora preko šest linija. Jedan par je za napajanje kola (3.3 V), a drugi je za uzemljenje (GND). Pinovi ESP32 modula koji kontrolišu poziciju motora su GPIO (*General Purpose Input Output*) pinovi, konkretno pinovi od 16 do 19. Veza između modula za kontrolu i samog motora je uspostavljena povezivanjem odgovarajućeg ženskog konektora na step motoru i muškog konektora na modulu

za kontrolu. Prototip sistema testiran na protoploči prikazan je na slici 6.



Slika 6. Primer praktičnog povezivanja korišćenjem softverskog alata *Fritzing*.

Za pisanje koda za kontrolu MCU koji je glavni deo ESP32 modula iskorišćena je *Espruino* platforma [9]. Ova platforma omogućava pisanje koda i kompajliranje korišćenjem programskog jezika *JavaScript*. Ovaj programski jezik nije konvencionalni programski jezik za programiranje bilo kojih MCU, ali ova platforma je odabrana jer je i veb aplikacija pisana korišćenjem ovog programskog jezika. Identični jezici, strukture podataka i paradigme omogućavaju identično pakovanje podataka prilikom slanja i prijema i na taj način smanjuju mogućnost za grešku ili za gubitak podataka. Zbog ovih razloga, *Espruino IDE (Integrated Development Environment)* se nameće kao dobar izbor platforme za razvoj *Internet of Things (IoT)* sistema.

Glavni deo koda za princip rada sistema u *Espruino IDE* pokazan je na slici 7.

```
1 var wifi = require("wifi");
2 var http = require("http");
3 // potrebni objekti za Wi-Fi povezivanje
4 var WebSocket = require("ws");
5 // objekat za korišćenje tehnike WebSockets
6 var stepperPins = [D19,D18,D17,D16];
7 // pinovi za povezivanje sa modulom za kontrolu
8 var steps = [0b0001,0b0010,0b0100,0b1000];
9 // niz sa mogućim pozicijama motora
10
11 // funkcija za pokretanje motora
12* function doStep(stepIndex) {
13     digitalWrite(stepperPins, steps[stepIndex]);
14     // nizu pinova se prosleđuje odgovarajući element niza za korak
15 }
16 // funkcija za povezivanje na Wi-Fi mrežu korišćenjem WebSockets
17* wifi.connect("Tech_D0006363", {password:"HVPVGYJM"}, function(err){
18     console.log("connected? err=", err, "info=", wifi.getIP());
19     var ws = new WebSocket('ieeeestec-smart-blind.herokuapp.com', {
20         path: '/',
21         port: 80, // default is 80
22         protocolVersion: 13,
23         origin: 'Espruino',
24         keepAlive: 60,
25     });
26 // prijem poruke i aktiviranje motora
27* ws.on('message', function(msg) {
28     var parsedMessage = JSON.parse(msg);
29     if(parsedMessage.step) {
30         doStep(Number(parsedMessage.step));
31     }
32 });
```

Slika 7. Glavni deo koda vezan za kontrolu motora.

Korišćenjem *WebSockets* vrši se prijem podatka *parsedMessage.step*. Ovaj podatak se šalje sa projektovane veb aplikacije, a dobija odgovarajućim pozicioniranjem slajdera. Nakon prijema, ispituje se vrednost podatka i u zavisnosti od vrednosti tog podatka, niz pinova (označeni u kodu kao *stepper Pins*) uključuje se na odgovarajući način. U ovom sistemu, moguće su samo četiri pozicije motora, iako je moguće realizovati finija podešavanja pozicije motora. Na taj način bi slajder bio u potpunosti iskorišćen.

Jedan od glavnih problema koji se javio prilikom razvoja ovog sistema je montiranje. Kako bi sistem sa ovim komponentama zaista mogao da pomera i okreće različite vrste rotirajućih roletni i venecijanera, montaža i kućište za sistem zahtevaju veliki broj specifičnih delova koji će omogućiti montiranja na prozore i vrata. Za ove delove potrebno je koristiti 3D štampače koji omogućavaju projektovanje i izradu delova specifičnih dimenzija i oblika.

Drugi problem koji se javio je napajanje sistema. Zamišljeno je da sistem bude autonoman, odnosno da se napaja preko odgovarajuće 18650 baterije. Ipak, to bi primoralo korisnika da povremeno mora da vrši punjenje uređaja. Za kompletno pouzdan sistem, plan za montiranje i primenu bi trebalo da postoji još prilikom projektovanja

same prostorije odnosno zida. Planom bi trebalo predvideti kablove koji bi bili prekriveni delovima zida, a koji bi omogućili konstantno napajanje za sisteme koji se nalaze pored roletni ili delova koji određuju poziciju roletni.

IV. ZAKLJUČAK

U radu je opisano projektovanje sistema za kontrolu dnevne osvetljenosti u prostoriji kontrolom nad rotirajućim roletnama. Hardverski resursi (korišćene komponente) i softverski resursi (korišćeni alati i moduli) su detaljno opisani. U radu su navedeni i problemi na koje se naišlo prilikom realizacije sistema. Glavna tačka napretka za budući razvoj ogleda se u projektovanju odgovarajuće kutije i u finijem podešavanju motora.

ZAHVALNICA

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026. Autori se zahvaljuju prof. dr Danijelu Dankoviću na motivaciji i pomoći prilikom izrade rada.

LITERATURA

- [1] B. Radenković, M. Despotović-Zrakić, Z. Bogdanović, D. Barać, A. Labus i Ž. Bojović, *Internet inteligentnih uređaja*, Fakultet organizacionih nauka, 2017.
- [2] Tehnička specifikacija za ESP32-WROOM2. Dostupno na: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf, pristupljeno 10.10.2021.
- [3] Tehnička specifikacija za 28BYJ-48. Dostupno na: <https://robocraft.ru/files/datasheet/28BYJ-48.pdf>, pristupljeno 10.10.2021.
- [4] Tehnička specifikacija za ULN2003. Dostupno na: <https://www.ti.com/lit/ds/symlink/uln2003a.pdf>, pristupljeno 10.10.2021.
- [5] Okruženje node.js. Dostupno na linku: <https://nodejs.org/en>, pristupljeno 10.10.2021.
- [6] Softverski alat Visual Studio Code. Dostupno na: <https://code.visualstudio.com/download>, pristupljeno 10.10.2021.
- [7] I. Fette i A. Melnikov, *The WebSocket Protocol*, 2011.
- [8] Platforma Heroku. Dostupno na: <https://www.heroku.com/>, pristupljeno 10.10.2021.
- [9] Softverski alat Espruino. Dostupno na: <https://www.espruino.com/Download>, pristupljeno 10.10.2021.

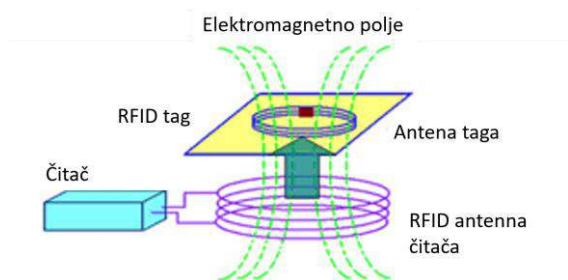
Prototip sistema za otključavanje brave korišćenjem RFID tehnologije

Aleksa Mladenović, Sara Stanojić, Mihajlo Usković, Nemanja Ćirić, Andrija Kostić,
Sara Mišić, Mateja Grujić

Sadržaj – U ovom radu biće predstavljen prototip sistema za otključavanje brave korišćenjem RFID tehnologije. Sistem je zasnovan na Arduinu, koristi RFID modul RDM6300 sa 125 kHz RFID tagovima i servo motor za kontrolu brave.

I. RFID TEHNOLOGIJA

Radio Frequency IDentification (RFID) je sistem daljinskog slanja i prijema podataka koji se dosta koristi u kontroli pristupa, praćenju pokretnih (automobila, kamiona) i nepokretnih (knjige iz biblioteke) objekata. RFID sistem sastoji se iz dva dela: taga koji je lociran u objektu koji treba da bude identifikovan i čitača (transivera) [1]. RFID se pojavio osamdesetih godina prošlog veka a u širokoj je upotrebi i danas. Princip rada RFID sistema ilustrovan je na slici 1. Čitač se sastoji iz RF modula, kontrolne jedinice i namotaja antene koja stvara visokofrekventno elektromagnetno polje. S druge strane, tag je pasivna komponenta koja se sastoji iz namotaja antene i čipa. Kada se tag nađe u elektromagnetnom polju čitača, zbog indukcije se generiše napon u namotajima antene taga i tako se napaja čip. Čip taga može da prima i šalje podatke do čitača tehnikom koja se zove *load manipulation*. Naime, uključivanje/isključivanje antene taga utiče na potrošnju energije antene čitača što se može izmeriti kao promena napona. Ove promene napona predstavljaju logička stanja 0 i 1, koje čitač prihvata kao podatke iz taga. Postoji i drugi način za prenos podataka koji se naziva *backscattered coupling*. U ovom slučaju, tag koristi deo primljene snage da generiše drugo elektromagnetno polje koje će detektovati antena čitača.



Slika 1. Princip rada RFID sistema.

A. Mladenović, S. Stanojić, M. Usković, N. Ćirić, A. Kostić, S. Mišić, M. Grujić su učenici ETŠ "Mija Stanimirović" na smeru Elektrotehničar informacionih tehnologija, Bulevar Svetog cara Konstantina 82-84, Niš, Srbija. E-mail: aleksa.mladenoviccc03@gmail.com

U većini okruženja, RFID postiže 99,5% do 100 % očitavanja u prvom skeniranju [2]. Svaki RFID tag ima u internom čipu upisan jedinstveni kod koji se, nakon prijema u čitaču, može dalje obrađivati. RFID sistemi se klasifikuju u tri frekventna područja. Svako ima svoje karakteristike i tipično područje primene:

1. *Low Frequency* – 100-500 kHz, a najčešće 125 kHz, najkraćeg dometa, najmanje brzine očitavanja i prenosa podataka;
2. *High Frequency* – 10-15 MHz, a najčešće 13.56 MHz, kratkog do srednjeg dometa signala, srednje brzine očitavanja i prenosa;
3. *Ultra High Frequency* (UHF) – rade u rasponu od 433 – 915 MHz i na 2.45 GHz, najvećeg dometa signala i najveće brzine prenosa. Kod ovih tagova ne sme biti prepreke između čitača i taga – UHF radio-talas ne prodire tako dobro kroz materijale i zahteva više energije za prenos u datom opsegu nego talas niže frekvencije.

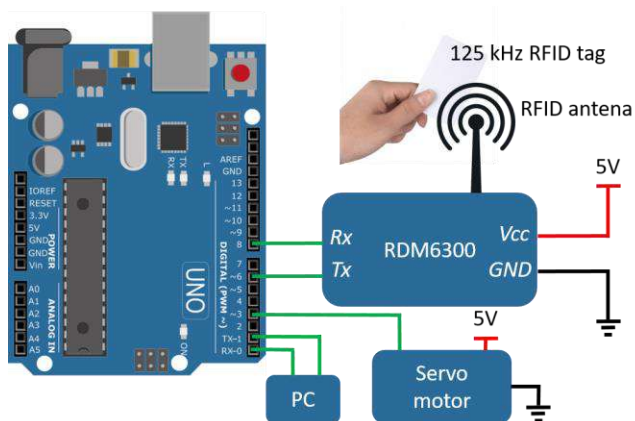
U mnogim kompanijama postoje prostorije sa serverima kojima pristup ima samo jedna ovlašćena osoba, koja korišćenjem RFID taga može da otključa bravu. U ovom radu biće predstavljen prototip sistema za otključavanje brave prostorije sa ograničenim pristupom, korišćenjem RFID tehnologije.

II. REALIZACIJA RFID SISTEMA ZA OTKLJUČAVANJE BRAVE

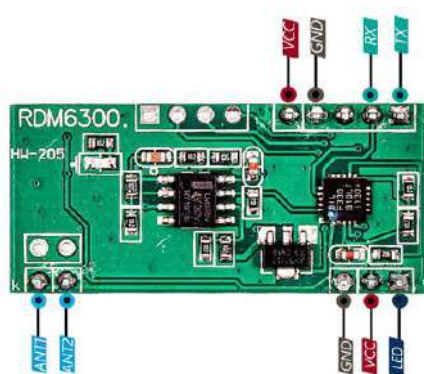
A. Hardverska realizacija

Blok šema RFID sistema za kontrolu brave prostorije sa ograničenim pristupom prikazana je na slici 2. Sistem je razvijen na Arduino platformi. Koristi se RFID pasivni tag (kartica) noseće frekvencije 125 kHz i RFID čitač RDM6300 sa eksternom antenom. Za kontrolu brave iskorišćen je servo motor. U procesu testiranja prototipa sistema Arduino je povezan sa računarom kako bi se pratio tok izvršenja programa mikrokontrolera.

RDM6300 RFID modul za čitanje kartica (slika 3) je dizajniran za isčitavanje tagova koji rade na 125 kHz. Spada u *Low Frequency* čitače i udaljenost sa koje može da očitava karticu je od 20-50 mm. Radi na naponu od 5 V. Ima dva pina za napajanje i dva pina za masu. Preko UART (*Universal asynchronous receiver-transmitter*), komunikacionog protokola komunicira sa mikrokontrolerom (RX i TX pin).



Slika 2. Blok šema RFID sistema za kontrolu brave.

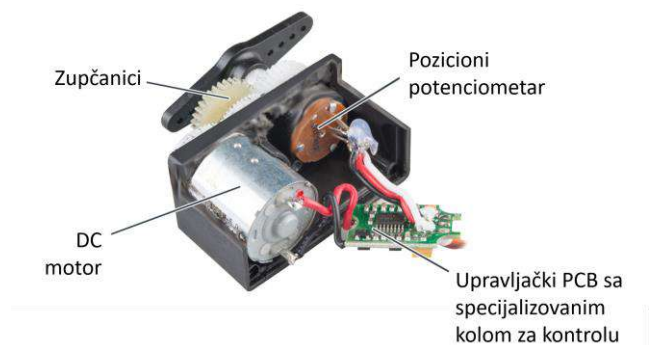


Slika 3. RDM6300 RFID čitač sa označenim pinovima [3].

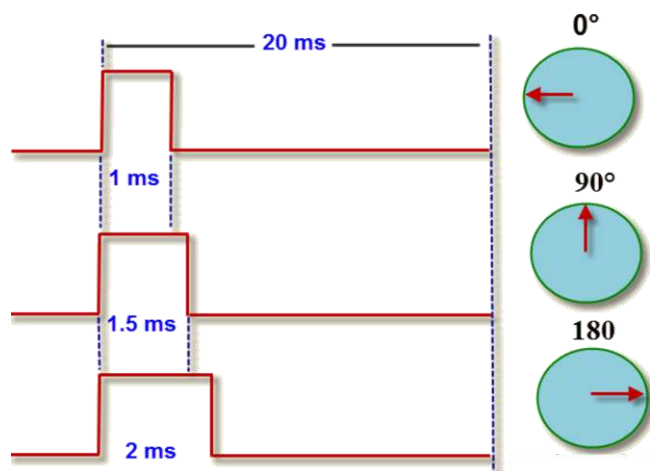
RFID čitač povezan je na Arduino razvojno okruženje. Na razvojnom okruženju nalazi se mikrokontroler ATmega328P, koji upravlja radom sistema. Pinovi 0 i 1 iskorišćeni su za UART komunikaciju između mikrokontrolera i računara, pinovi 6 i 8 za UART komunikaciju između RFID čitača i mikrokontrolera. Treba napomenuti da se komunikacija s računarom odvija korišćenjem UART modula mikrokontrolera, dok je UART komunikacija sa RFID modulom ostvarena korišćenjem softverske biblioteke. PWM pin 3 Arduina iskorišćen je za povezivanje servo motora.

RC Servo motori se koriste u aplikacijama gde je potrebno upravljanje ugaonom pozicijom, brzinom i ubrzanjem [4]. RC servo motori su sastavljeni iz tri dela: motora sa reduktorom, uređaja povratne sprege i upravljačke ploče. Kod RC servoa, uređaj povratne sprege čini najčešće potencijometar. Motor, preko niza zupčanika (koji čine reduktor), okreće izlaznu osovinu i potencijometar istovremeno. Upravljački modul "očitava" otpornost potencijometra i na taj način utvrđuje trenutnu poziciju izlazne osovine. Konstrukcija RC servo motora prikazana je na slici 4. Pored pinova za napajanje (5V), servo motori imaju pin za upravljanje. Upravljački signal je PWM (*Pulse Width Modulation*) signal. Trajanje impulsa PWM signala određuje poziciju osovine (slika 5). Dok god je upravljački signal prisutan, servo motor zadržava

poziciju svoje izlazne osovine. Pri pojavi signala motor rotira osovinu u smeru u kom će se najpre ostvariti zadata pozicija. Kada se pozicija ostvari osovinu se zaustavlja i zadržava poziciju osovine. Upravljački signali servo motora imaju frekvenciju od 50 Hz. Ako servo motor treba da zauzme ugao od 0° potrebno je da impuls upravljačkog signala traje 1 ms (faktorom ispunje od 5%). Da bi se servo motor pozicionirao na ugao od 180° potrebna širina impulsa je 2 ms (faktor ispunje 10%).



Slika 4. Konstrukcija RC servo motora.



Slika 5. PWM upravljački signal RC servo motora za različite položaje osovine.

B. Softverska realizacija

U realizaciji softvera korišćene su funkcije iz biblioteka *SoftwareSerial.h* za implementaciju UART komunikacije između RFID čitača i mikrokontrolera kao i *Servo.h* za upravljanje servo motorom. U glavnoj programskoj petlji čeka se paket podataka preko softverskog UART-a, kada podaci pristignu poziva se funkcija *extract_tag()* koja vrši razdvajanje delova pristiglog paketa i vraća vrednost očitano taga. Sledi upoređivanje vrednosti očitano taga sa tag šifrom. Ako je iskorišćen pravi tag vrši se otključavanje brave određen vremenski period, nakon čega se automatski ponovo zaključava. Ukoliko se tag ne podudara sa šifrom brava ostaje zaključana. Deo koda dat je u nastavku.

```

void loop() {
    if (ssrfid.available() > 0) {
        bool call_extract_tag = false;
        int ssvalue = ssrfid.read(); // čitanje
        if (ssvalue == -1) { // nisu očitani podaci
            return;
        }
        if (ssvalue == 2) { // tag detektovan
            buffer_index = 0;
        } else if (ssvalue == 3) { // tag je upisan
            prijemni_registar
            call_extract_tag = true;
        }
        if (buffer_index >= BUFFER_SIZE) {
            Serial.println("Greska: Prepunjen bafer");
            return;
        }
        buffer[buffer_index++] = ssvalue; // ako je
        sve u redu, kopira se očitana vrednost u bafer
        if (call_extract_tag == true) {
            if (buffer_index == BUFFER_SIZE) {
                unsigned tag = extract_tag();
                if (tag == TAG_SIFRA)
                {
                    mojServo.write(180);
                    Serial.println("Vrata su otključana!");
                    delay(vreme);
                    mojServo.write(0);
                    Serial.println("Vrata su se
zaključala!");
                }
                else
                {
                    mojServo.write(0);
                    Serial.println("Vrata su zaključana!");
                }
            } else {
                buffer_index = 0;
                return;
            }
        }
    }
}

```

Paket podataka (*frame*), koga šalje RFID čitač u mikrokontroler, sastoji se iz 14 bajtova i to: 1 bajt zaglavlja (*head*), 10 bajtova podataka, 2 bajta *checksum*-e i 1 stop bajt (slika 6). Vrednost bajta zaglavlja, kada je očitana tag, je 2. Paket korisnih podataka ima 2 bajta za verziju i 8 bajtova za tag. Stop bajt ima vrednost 3 kada je prenos paketa završen.

02	verzija	tag	checksum	03
----	---------	-----	----------	----

Slika 6. Paket podataka (*frame*) koji šalje RFID čitač u mikrokontroler.

Funkcija za razdvajanje pristiglog paketa na delove, pod nazivom *extract_tag()*, zasniva se na nekoliko for petlji pomoću kojih se u odgovarajuće promenljive izdvajaju pojedini elementi niza. Pri tom vrši se pretvaranje stringa u celobrojnu vrednost, za šta je napisana posebna funkcija *hexstr_to_value()*. Izdvojene vrednosti start bajta, verzije, taga, checksum-e i stop bajta štampaju se preko serijskog terminala korišćenjem funkcije *Serial.print()*. Listing koda funkcije *extract_tag()* dat je u nastavku.

```

unsigned extract_tag() {
    uint8_t msg_head = buffer[0];
    uint8_t *msg_data = buffer + 1;
    uint8_t *msg_data_version = msg_data;
    uint8_t *msg_data_tag = msg_data + 2;
    uint8_t *msg_checksum = buffer + 11;
    uint8_t msg_tail = buffer[13];
    Serial.println("-----");
    Serial.print("Zaglavlje poruke: ");
    Serial.println(msg_head);
    Serial.println("Message-Data (HEX): ");
    for (int i = 0; i < DATA_VERSION_SIZE; ++i) {
        Serial.print(char(msg_data_version[i]));
    }
    Serial.println(" (version)");
    for (int i = 0; i < DATA_TAG_SIZE; ++i) {
        Serial.print(char(msg_data_tag[i]));
    }
    Serial.println(" (tag)");

    Serial.print("Message-Checksum (HEX): ");
    for (int i = 0; i < CHECKSUM_SIZE; ++i) {
        Serial.print(char(msg_checksum[i]));
    }
    Serial.println("");

    Serial.print("Message-Tail: ");
    Serial.println(msg_tail);

    Serial.println("--");
    long tag = hexstr_to_value(msg_data_tag,
DATA_TAG_SIZE);
    Serial.print("Extracted Tag: ");
    Serial.println(tag);
    long checksum = 0;
    for (int i = 0; i < DATA_SIZE; i+=
CHECKSUM_SIZE) {
        long val = hexstr_to_value(msg_data + i,
CHECKSUM_SIZE);
        checksum ^= val;
    }
    Serial.print("Extracted Checksum (HEX): ");
    Serial.print(checksum, HEX);
    if (checksum == hexstr_to_value(msg_checksum,
CHECKSUM_SIZE)) {
        Serial.print(" (OK)");
    } else {
        Serial.print(" (NOT OK)");
    }
    Serial.println("");
    Serial.println("-----");
    return tag;
}

```

Listing koda funkcije za konvertovanje *hex* stringa u *long* celobrojnu vrednost dat je u nastavku.

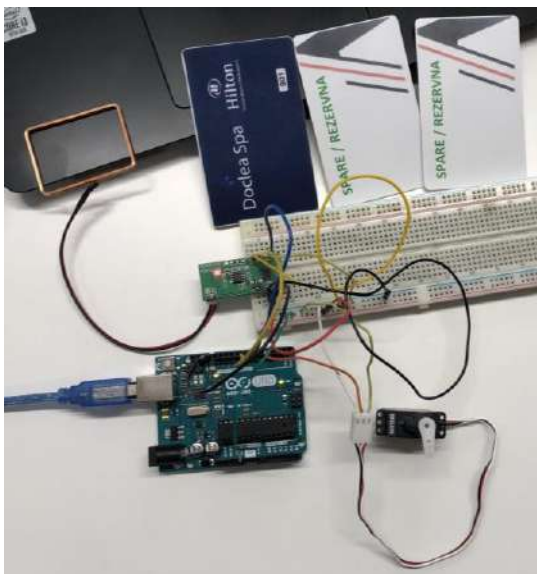
```

long hexstr_to_value(char *str, unsigned int
length) {
    char* copy = malloc((sizeof(char) * length) +
1);
    memcpy(copy, str, sizeof(char) * length);
    copy[length] = '\0';
    long value = strtol(copy, NULL, 16);
    free(copy);
    return value;
}

```

III. REZULTATI TESTIRANJA PROTOTIPA

Opisani sistem realizovan je na protoploči (slika 7) i testiran sa 3 tag kartice. Kod jedne od kartica definisan je kao šifra. Dve kartice, čiji kod se ne podudara sa šifrom, su uspešno očitane i preko serijskog terminala je prikazan očitani tag (slika 8). Servo motor bio je u položaju “zaključano”. Prilikom očitavanja taga, čiji se kod podudara sa šifrom, preko serijskog terminala su prikazani parametri datog taga (slika 9). Servo motor je zauzeo položaj “otključano”, a zatim se nakon definisanog vremena vratio u položaj “zaključano”.



Slika 7. Realizacija sistema za otključavanje brave na protoploči.

```
Zaglavlje poruke: 2
Message-Data (HEX):
51 (version)
0035CDE0 (tag)
Message-Checksum (HEX): 49
Message-Tail: 3
--
Extracted Tag: 3526112
Extracted Checksum (HEX): 49 (OK)
-----
Vrata su zaključana!
```

Slika 8. Poruka primljena preko serijskog terminala kada su očitani tagovi sa pogrešnom šifrom.

```
Zaglavlje poruke: 2
Message-Data (HEX):
01 (version)
0035ED62 (tag)
Message-Checksum (HEX): BB
Message-Tail: 3
--
Extracted Tag: 3534178
Extracted Checksum (HEX): BB (OK)
-----
Vrata su otključana!
Vrata su se zaključala!
```

Slika 9. Poruka primljena preko serijskog terminala kada su očitani tagovi sa ispravnom šifrom.

IV. ZAKLJUČAK

U ovom radu opisan je proces projektovanja jednog sistema za kontrolu brave prostorije s ograničenim pristupom korišćenjem RFID tehnologije. Sistem je realizovan i testiran na protoploči. Eksperimentalno je pokazano da se brava otključava samo kada se očitava tag sa ispravnom šifrom. Za opisani sistem je moguće projektovati štampanu ploču i odgovarajuće kućište, čime bi predloženi prototip postao finalni uređaj. Mogućnosti proširenja funkcionalnosti uređaja su ogromne: povezivanje na internet, izrada odgovarajuće mobilne ili desktop aplikacije.

ZAHVALNICA

Ovaj rad realizovan je u okviru projekata "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu. Autori se zahvaljuju realizatorima projekata na pruženom znanju i pomoći tokom izrade rada.

LITERATURA

- [1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, Wiley, 2010.
- [2] RFID identifikacija. Dostupno na: <https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2007-01-179.pdf>
- [3] Interfacing RDM6300 RFID Reader Module with Arduino. Dostupno na: <https://electropeak.com/learn/interfacing-rdm6300-125khz-rfid-reader-module-with-arduino/>.
- [4] Servo motori. Dostupno na: <https://www.automatika.rs/baza-znanja/mehatronika/rc-servo-motori.html>

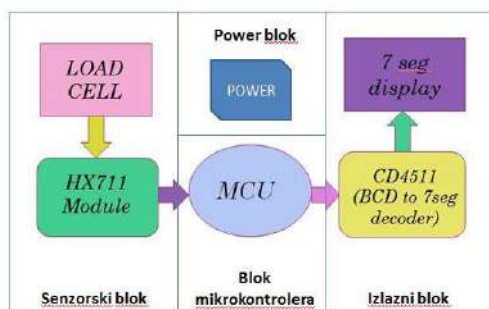
Projektovanje štampane ploče pametne vage

Nina Dimitrijević, Ana Sentov

Sadržaj – Projekat pod nazivom „Smart Scale” je nastao na stručnoj praksi u kompaniji „HTEC Group” u Nišu u okviru „Summer Embedded Intership 2020”. U ovom radu predstavljen je postupak projektovanja štampane ploče pametne vage zasnovane na ARM mikrokontroleru.

I. OPIS PROJEKTA

Zadatak projekta „Smart Scale” je bio realizovati vagu koja će sa senzora očitavati podatke i to prikazivati na displeju. Na slici 1. prikazan je blok dijagram projekta, gde se mogu uočiti njegovi funkcionalni blokovi. Ceo projekat je podeljen na 4 najvažnija bloka, a to su: blok napajanja (power blok), senzorski blok, blok mikrokontrolera i izlazni blok.



Slika 1. Blok dijagram projekta.

Senzor pritiska TAL220 (slika 2) analogni signal šalje do modula HX711 [1] koja konvertuje analogni u digitalni signal. Dalje digitalni signal putuje do mikrokontrolera STM32F401RC [2] koji uz određenu komandu prosleđuje taj digitalni signal na dekodirani CD4511BE [3] kako bi on te signale pretvorio u signale koji su potrebni za ispisivanje cifara na 7-segmentnom displeju 5611AH.

Senzor TAL220 pretvara mehaničku silu u električni analogni signal. Kako bi se javila sila naprezanja potrebno je da jedan deo senzora bude pričvršćen, a drugi kraj da bude slobodan i da se na njemu postavlja težina. Uz pomoć 3D štampača odštampani su potrebni delovi. Dakle, kada se postavi neka težina na senzor, javlja se sila naprezanja, i tada analogni signal putuje do HX711 modula.

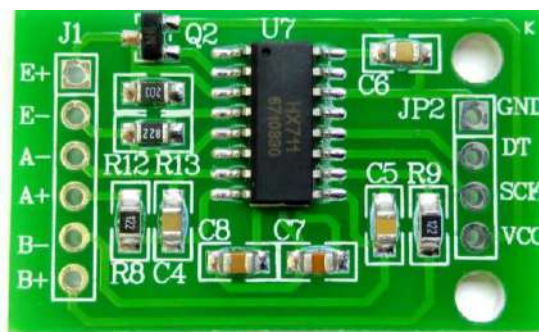
N. Dimitrijević, student master studija na modulu Elektronika i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nina.dimitrijevic@elfak.rs

A. Sentov, student četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ana.sentov@elfak.rs



Slika 2. TAL220 senzor pritiska.

Na slici 3. prikazan je modul HX711 koji ima integrisan HX711 čip, ADC (Analog-to-Digital Converter) i PGA (Programmable Gain Amplifier). Posедуje 2 pina za komunikaciju, jedan pin za slanje i primanje podataka i drugi pin za primanje clock-a sa mikrokontrolera. U zavisnosti koliko clock-a primi uslediće pojačanje signala određene vrednosti. Tako da ukoliko se sa mikrokontrolera proslede 25 clock-a, signal sa senzora će se pojačati 128 puta. Takođe, princip čitanja podataka nije standardizovan pa je potrebno detaljnije proučiti tehničku specifikaciju modula.



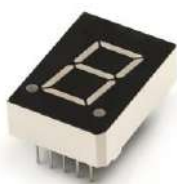
Slika 3. Modul HX711.

Mikrokontroler sa kojeg se manipuliše svim komponentama i podacima jeste STM32F401RC (slika 4), koji je programiran u STM32CubeIDE okruženju na Ubuntu distribuciji Linux operativnog sistema.

Komponenta koja prima digitalni signal sa mikrokontrolera jeste dekodirani CD4511BE čija je funkcija da konvertuje BCD kod u 7-segmentni kod. Dakle, vrednost koja se dobije sa senzora jeste neki broj koji se deli na cifre i svaka cifra se posebno šalje na određeni 7-segmentni displej. S obzirom da je BCD kod 4-bitni i da sama komponenta ima odvojene pinove za svaki bit (pin D, pin C, pin B i pin A), za određenu cifru se šalju određeni bitovi na prethodno pomenute pinove. 7-segmentni displej, čija je oznaka 5611AH do koga stiže već dekodirani 7-segmentni kod prikazana je na slici 5.



Slika 4. Mikrokontroler STM32F401RC.



Slika 5. 7 Segment Display – 5611AH.

Sva stanja u kojima može da se nađe uređaj tj. vaga prikazana su na slici 6. Prvo kad se uključi vaga na napajanje prolazi kroz *init* stanje tačnije inicijalizaciju, što znači da uključuje sve potrebne biblioteke, sve .c i .h fajlove koji su potrebni. Takođe, dok je u inicijalnom stanju, kalibriše se i postavlja nule na displeju. Kada prođe ovo stanje, vaga se konstantno nalazi u *scale* stanju i tada konstantno prima podatke sa senzora i ispisuje to na displeju. S obzirom da postoje dodatna 3 tastera, *ON/OFF* taster za uključivanje i isključivanje, *Calibrate* taster za dodatnu kalibraciju i *Tare* taster za tariranje. Kada se pritisne taster za kalibraciju, vaga uđe u *calibrate state*, uključi se LE dioda i tada je potrebno da korisnik isprati određene korake: stavi teg od 100 g na vagu i sačeka da se LE dioda isključi i da tada skloni teg sa vage i na displeju će se prikazati nule. *Tare* taster služi da ukoliko korisnik ne želi da izmeri npr. težinu posude, pritisne taster za tariranje i na displeju se prikažu nule.

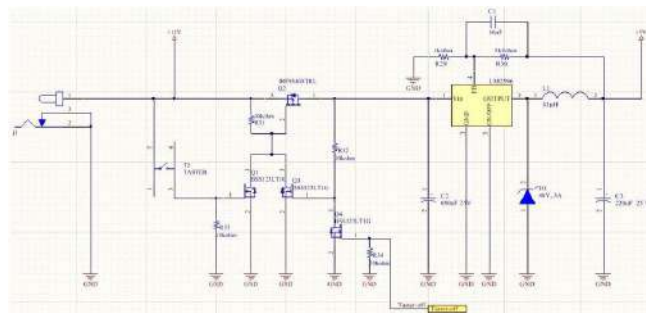
II. DIZAJNIRANJE I IZRADA PCB-A

A. Projektovanje PCB-a

Projektovanje štampane ploče i crtanje električnih šema rađeno je u programu *Altium Designer* [4]. Nakon odabira komponenata, čitanja njihove dokumentacije, podele projekta na blokove, odnosno funkcionalne celine, sledi izrada električnih crteža. Prvi blok je power blok čija je električna šema prikazana na slici 7.

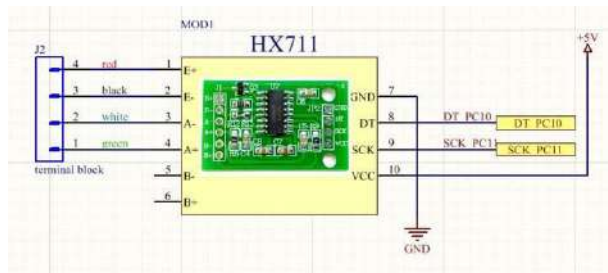
```
switch(current_state)
{
    case INIT:
    {
        StateInit();
    }
    break;
    case SCALE:
    {
        StateScale();
    }
    break;
    case CALIBRATE:
    {
        StateCalibrate();
    }
    break;
    case TARE:
    {
        StateTare();
    }
    break;
    default:
    break;
}
```

Slika 6. Deo koda gde su prikazana stanja.



Slika 7. Električna šema power bloka.

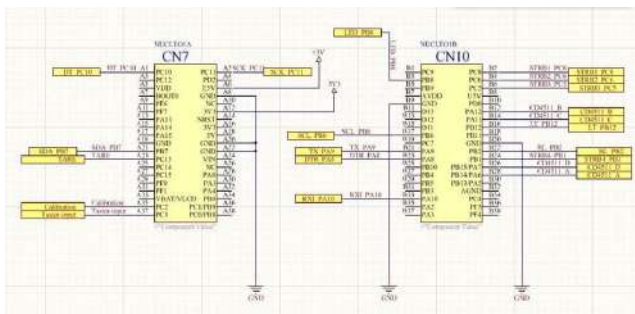
Senzorski blok, prikazan na slici 8, prikazuje povezivanje komponente TAL220 (senzora pritiska) sa modulom HX711.



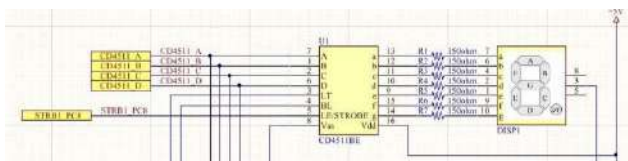
Slika 8. Električna šema senzorskog bloka.

Blok mikrokontrolera prikazan je na slici 9. Izlazni blok sadrži komponente za prikaz podataka koji su pristigli iz senzorskog bloka i koji su obrađeni u mikrokontroleru. Najbitnije komponente u ovom bloku su dekodir CD4511BE i 7-segmentni displej koje su opisane u prethodnom poglavlju. Na slici 10. je prikazan deo električne šeme ovog izlaznog bloka tj. način povezivanja dekodera i 7-segmentnog displeja. Kao što se sa slike može uočiti povezani su preko otporne mreže. Merenje je

prikazano na četiri 7-segmentna displeja, pa bi maksimalna moguća izmerena vrednost bila 9.999.



Slika 9. Električna šema bloka mikrokontrolera.



Slika 10. Električna šema izlaznog bloka.

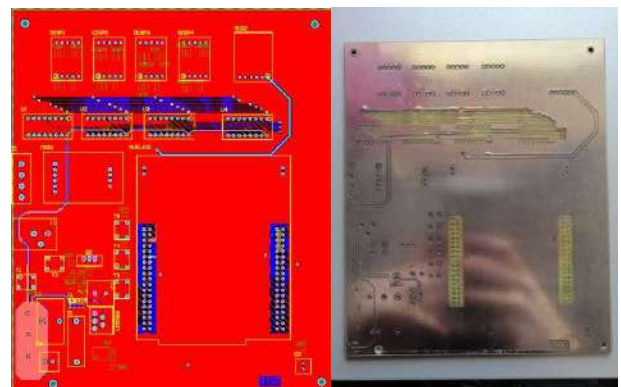
Nakon crtanja električnih šema funkcionalnih blokova, kreće se sa 3D modeliranjem. Komponente se po načinu na koji ostvaruju električni kontakt sa linijama veza na ploči dele na: *through-hole* i SMD (*Surface Mounting Devices*). Komponente se mogu montirati sa jedne ili sa obe strane štampane ploče. Oblik električnog i fizičkog kontakta komponente i ploče naziva se otisak (*footprint*) komponente. Kako bi se napravio *footprint* komponente potrebno je vrlo precizno znati njegove dimenzije. Kada su formirane sve komponente prelazi se na sledeći korak, a to je raspored komponenata tj. *layout*. Raspored komponenata treba da bude optimalan kako bi se lepo definisale linije veze - *routing*. Na pločici za Smart Scale nalaze se i SMD i *through-hole* komponente.

Izrada proizvodne dokumentacije podrazumeva generisanje i proveru *GERBER* fajlova. Ovi fajlovi sadrže podatke, o na pr. koordinatama za bušenje rupa sa definisanim prečnicima. *GERBER* je standardni format za uređaje u proizvodnji štampanih ploča. Pored ovih fajlova koji se šalju u fabriku za izradu PCB pločice, šalje se i lista materijala potrebnih za izradu (*bill of materials- BOM*).

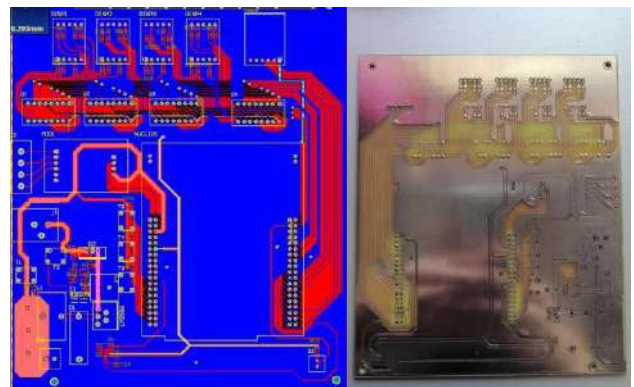
Na slici 11 može se videti kako izgleda gornja strana pločice nakon rutiranja i razlivanja mase. Leva slika je iz programa *Altium Designer*, dok se na desnoj slici može videti kako izgleda gornja strana ploče nakon izrade. Na slici 12 može se videti kako izgleda donja strana pločice nakon rutiranja i razlivanja mase. Leva slika je iz programa *Altium Designer*, dok se na desnoj slici može se videti kako izgleda donja strana ploče nakon izrade.

Kada se završi dizajniranje pločice u programu, može se videti kako bi ona u stvarnosti izgledala. Na slici 13 (levo) vidimo 3D model pločice sa njenim dimezijama kako bi ona izgledala nakon lemljenja svih komponenata. Zahvaljujući 3D modelima komponenata mogu se unapred

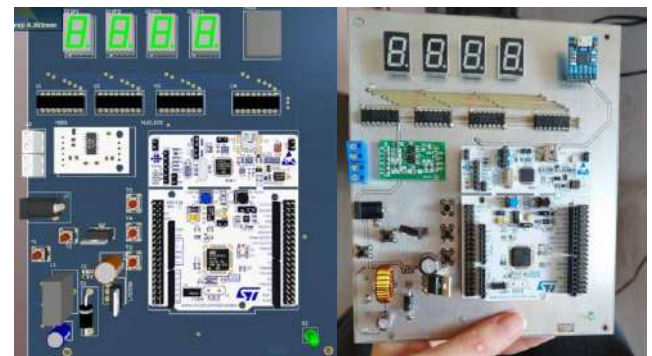
uočiti neki problemi, kao što su nedostatak prostora ili preklapanje sa nekim drugim komponentama zbog prevelikih dimenzija.



Slika 11. Top layer (levo iz programa, desno top layer izrađene pločice).



Slika 12. Bottom layer (levo iz programa, desno top layer izrađene pločice).

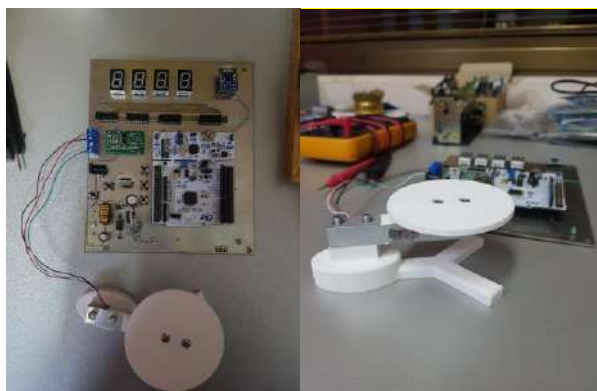


Slika 13. 3D model pametne vage (levo), finalni izgled pločice sa zalemljenim komponentama (desno).

Na slici 13 (desno) može se videti izgled pločice nakon lemljenja. Sve komponente su lemljene ručno lemilicom i srebrnom pastom. Komponente koje su *through-hole* lemljene su sa donje strane pločice.

III. REZULTAT PROJEKTA I PROBLEMI

Dizajnirano je i postolje za senzor koje je odštampano na 3D štampaču. Krajnji izgled pločice i senzora sa postoljem prikazano je na slici 14.



Slika 14. Finalni izgled projekta.

Tokom izrade ovog projekta bilo je različitih problema. Prilikom projektovanja pločice predviđene su određene komponente koje kasnije, kada je došlo vreme lemljenja, nisu bile dostupne u prodavnici. Veoma je bitno da se nabavka komponenata uskladi sa projektovanjem. Jedan od problema koji se javio jeste da kod modula HX711, izlazni pin mase modula i pin mase komponente

HX711 nisu bili međusobno povezani, pa je to lemljenjem kratkospojeno.

IV. ZAKLJUČAK

U ovom radu opisan je uređaj pametne vage zasnovane na ARM mikrokontroleru. Za pametnu vagu projektovana je i izrađena štampana ploča. PCB projekat realizovan je u programu Altium Designer. Pojedini delovi uređaja urađeni su na 3D štampaču.

ZAHVALNICA

Veliku zahvalnost dugujemo menotrima Marku Brelaku i Dušanu Dimitrijeviću, koji su nam sve vreme pomagali u izradi ovog projekta.

LITERATURA

- [1] 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales. Dostupno na: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- [2] STM32F401RC. Dostupno na: <https://www.st.com/en/microcontrollers-microprocessors/stm32f401rc.html>
- [3] CMOS BCD-to-7-Segment Latch Decoder Drivers datasheet. Dostupno na: <https://www.ti.com/lit/ds/symlink/cd4511b.pdf>
- [4] Altium Designer. Dostupno na: <https://www.altium.com/altium-designer>

Projektovanje uređaja za merenje nivoa glukoze u krvi

Vanja Stošić

Sadržaj – U ovom radu dat je teorijski pregled i analiza uređaja za merenje nivoa glukoze u krvi (u daljem tekstu glukometar) i njegovo projektovanje u programu *Altium Designer*.

I. UVOD O GLUKOMETRIMA

Glukometar određuje koncentraciju glukoze u rastvoru/krv. Normalni nivo glukoze u krvi je u ljudskom telu strogo regulisan, u rasponu od 70-150 mg/dL. Uređaji koje mogu koristiti neobučeni, nemedicinski stručnjaci - zaduženi su za redovno merenje, evidentiranje i skladištenje vrednosti hemoglukoze od kuće.

Većina glukometara je zasnovana na elektrohemijskoj tehnologiji koja koristi elektrohemijske test trake za izvođenje merenja. Mala kap rastvora koji se testira stavlja se na test traku za jednokratnu upotrebu koju glukometar koristi za merenje glukoze. U svakoj traci koja se koristi za određivanje glukoze u krvi nalazi se enzim koji se naziva glukoza-oksidaza. Ovaj enzim zatim reaguje sa glukozom u uzorku krvi i stvara kiselinu koja se naziva glukonska kiselina. Tako nastala glukonska kiselina tada reaguje sa drugom hemikalijom u testnoj traci koja se naziva ferocijanid. Ferocijanid i glukonska kiselina se zatim međusobno spajaju i formiraju ferocijanid. Čim se formira ferocijanid, uređaj (tj. glukometar) pokreće električnu struju kroz uzorak krvi na traci. Ova struja se stalno prati dok je traka na mestu, omogućavajući uređaju da nadgleda kada se stavi krv. Ova generisana struja može očitati ferocijanid i identifikovati količinu glukoze prisutne u uzorku krvi na test traci. Nakon što se hemijska reakcija stabilizuje, ADC očitava napon i upoređuje ga pomoću tabele za pregled da bi se dobila proporcionalna vrednost glukoze u mg/dL. Ova vrednost se šalje računaru domaćinu radi informisanja o vrednosti glukoze. Taj broj se zatim prikazuje na ekranu glukometra [1].

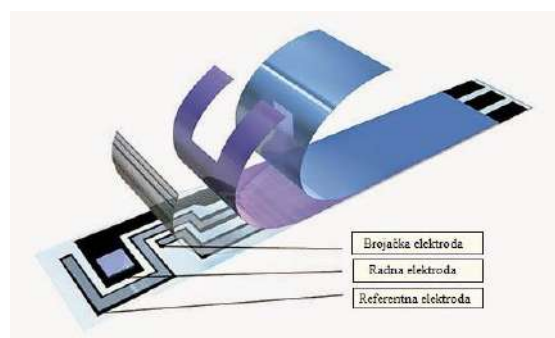
II. PRINCIP RADA GLUKOMETRA

Biosenzor koji se koristi u glukometru ima elektroenzimski pristup, što znači da koristi prednosti oksidacije glukoze enzimom glukoza-oksidaza. Prisustvo glukoza-oksidaze katalizuje hemijsku reakciju glukoze sa kiseonikom, što uzrokuje povećanje pH, smanjenje parcijalnog pritiska kiseonika i povećanje vodonik peroksida zbog oksidacije glukoze u glukonsku kiselinu.

V. Stošić je student master studija sa Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: vanja.5498@gmail.com

Test traka meri promene u jednoj ili više ovih komponenti kako bi se odredila koncentracija glukoze. Trake koje se koriste u ovom dizajnu imaju tri terminala ili elektrode kao na slici 1:

1. Referentna elektroda (Reference electrode) – drži se na konstantnom naponu u odnosu na radnu elektrodu kako bi se pogurale željene hemijske reakcije
2. Radna elektroda (Working electrode) – na njoj se proizvodi naelektrisanje tokom hemijske reakcije i povezana je sa transimpedansnim pojačavačem
3. Okidačka (brojačka) elektroda (Trigger/Counter electrode) – dovodi struju do radne elektrode



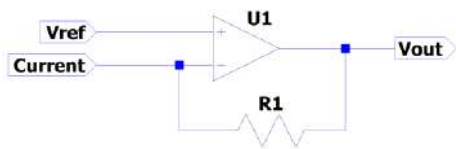
Slika 1. Izgled testne trakice.

Većina dizajna merača glukoze koristi samo dve elektrode, referentnu elektrodu i radnu elektrodu. Na referentnu elektrodu primenjuje se precizan referentni napon, a na operacioni pojačavač napon napajanja. Na ovaj način se održava precizna razlika potencijala preko radne elektrode i referentne elektrode. Ovaj napon je stimulus koji pokreće izlaznu struju test trake.

Uzorak rastvora se stavlja na test traku i dolazi do reakcije glukoze sa enzimom. Elektroni nastaju tokom hemijske reakcije. Protok elektrona će odgovarati protoku struje kroz radnu i referentnu elektrodu. Ova struja će se menjati u zavisnosti od koncentracije glukoze. Izlaz transimpedansnog pojačavača će se posmatrati kao varijacija napona sa promenljivom koncentracijom glukoze u rastvoru [2].

A. Pretvarač struje u napon

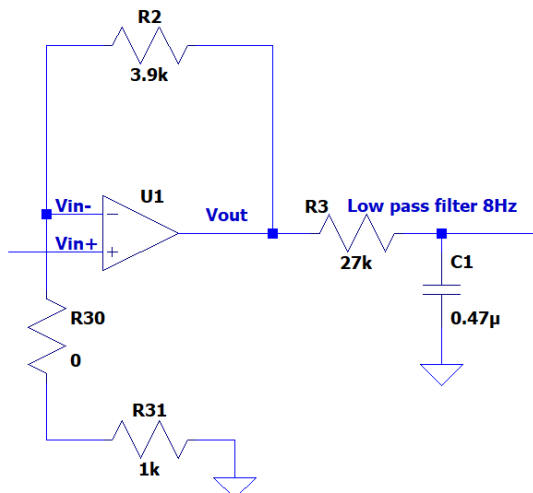
Izlaz koji generiše test traka je struja koja predstavlja koncentraciju glukoze. Ova struja se mora pretvoriti u napon kako bi se mogla pravilno filtrirati i tretirati. Ova konverzija se vrši pomoću pretvarača struje u napon koji ima jedno napajanje, nisko-ulazni offset napon i otpornost iz povratne sprege (slika 2).



Slika 2. Strujno-naponski konvertor.

B. Pojačanje i filtriranje signala

Blok sa slike 3. je podeljen na pojačavački deo i zatim niskopropusni filter sa graničnom frekvencijom od 8 Hz i dizajniran je da eliminiše šumove visoke frekvencije.



Slika 3. Blok za pojačanje i filtriranje signala.

III. IZGLED GLUKOMETRA, METODOLOGIJA I SIMULACIJA

Na slici 4. je prikazan izgled glukometra i blokova koji ga čine [3]. Sam glukometar poseduje tri eksterna porta, senzorski port (1) koji služi za umetanje test trake; kalibracioni port (2) za kalibrisanje novog paketa testnih traka pre prvog korišćenja; port za podatke (3) preko kojeg se glukometar puni ili služi za prenos/manipulaciju podataka (USB, RS-232, bežični..).

Kompletno uzorkovanje i analizu vrši integrisano kolo analognog *front-enda* (4) i ono se sastoji od *DA/AD* konvertora, operacionih pojačavača, transimpedansnog pojačavača, linearnih regulatora napona, temperaturnih senzora, termistora i detaljnije je opisano u nastavku rada, deo po deo.

Poželjno je da se temperatura krvi izmeri na test traci, ali obično se meri temperatura okoline u blizini test trake. Ovo merenje se može izvršiti pomoću samostalnih IC senzora temperature ili pomoću termistora ili PN spoja zajedno sa ADC-om. Korišćenje termistora u konfiguraciji *half-bridge* pobuđenog istim referentnim naponom kao i

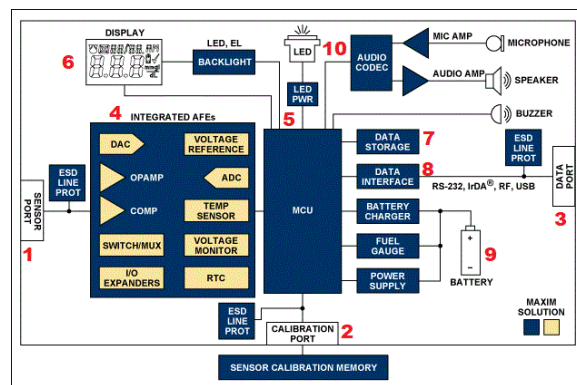
ADC daje tačnije rezultate jer ovaj dizajn eliminiše greške oko referentnog napona.

Mikrokontroler (5) je kontrolna jedinica ovog uređaja i parametar u vidu napona koji se iz prethodnog bloka prosleđuje mikrokontroleru, se provlači kroz unapred definisane tabele sa nivoima glukoze u krvi. Kada se odredi nivo glukoze, ta vrednost se prosleđuje LC displeju (*Liquid Crystal Display*) (6) preko LCD drajvera koji je integrisan unutar mikrokontrolera. Ta vrednost se takođe upisuje u memoriju (7).

Ako se radi o prenosnim glukometrima, sadržaće i unutrašnju punjivu bateriju putem koje će moći da radi (9).

Interfejs podataka (8) ima mogućnost otpremanja rezultata testova na računar putem *Bluetooth-a*.

Zvučni indikatori (10) se kreću od jednostavnih zujalica do naprednijih merača zvuka za osobe oštećenog vida [4].



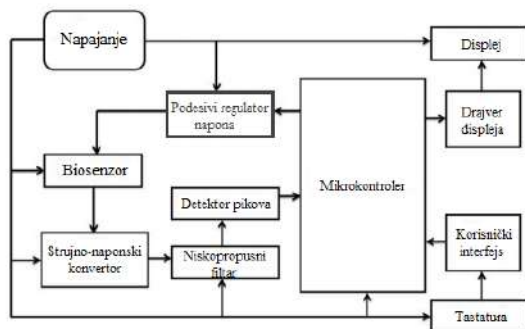
Slika 4. Funkcionalni blokovi glukometra [3].

IV. BLOK DIJAGRAM GLUKOMETRA

Sam uređaj se sastoji od više zasebnih blokova kao na slici 5. koji obavljaju određene funkcije i simultano rade čineći proizvod kompletnim. Blok biosenzora prikuplja informacije sa testne trake i generisanu struju šalje na strujno-naponski pretvarač za konverziju u napon. Regulator napona održava referentni napon konstantnim. Blok niskopropusnog filtra služi za odsecanje šumova, a blok za detekciju pikova konstantno proverava jačinu struje odnosno koncentraciju glukoze u krvi i te informacije dalje šalje mikrokontroleru koji ih dalje obrađuje i prosleđuje preko drajvera do displeja. Tastatura i korisnički interfejs se koriste za manevrisanje gotovih podataka (brisanje, ponovni unos, memorisanje..).

Da bi uređaj mogao da primi različite test trake, napon napajanja treba precizno promeniti/podesiti. Nakon akvizicije, potrebno je obraditi napon koji odgovara amperometrijskoj struji i dobiti odgovarajuću vrednost koncentracije glukoze. U tu svrhu se koristi mikrokontroler. Postoji nekoliko uobičajeno dostupnih test traka i definisana je tablica za pregled koja prikazuje odnos između koncentracije glukoze i struje dobijene test trakama

različitih proizvođača. Izbor test traka je izvršen pomoću tastature koja je bila povezana sa mikrokontrolerom.



Slika 5. Blok šematski prikaz predloženog prenosivog višetrakastog glukometra.

V. KOMPONENTE I INTEGRISANA KOLA

Komponente i integrisana kola korišćena za ovaj rad su SMD (*Surface Mount Device*) tipa kućišta. Za projektovanje ovog uređaja su bila potrebna četiri operaciona pojačavača LMC6484 [5] koji se nalaze u jednom integrisanom kućištu. Kod četverostranog prekidača CD4066BM [6] iskorišćen je jedan, dok su ostali pinovi prekidača spušteni na masu. Kao alternativa mogao je da se koristi i MAX4575 analogni prekidač, ali je korišćenje čipa poželjnije i jeftinije.

Konkretno za konfiguraciju komponente LMC6484 svaki od operacionih pojačavača pojedinačno bio je iskorišćen u konfiguraciji bafera impedanse, integratora i konvertora struje u napon.

Regulatori korišćeni za podešavanje referentnog napona, i to:

- TPS76925DBVT [7] - linearni regulator za fiksni referentni napon sa 5 V na 2,5 V (obzirom da su single-supply (jednostrano napajanje)) i
- TPS76901DBVT - linearni promenljivi regulator kod koga se eksternim otpornicima u konfiguraciji delitelja napona podešava referentni napon na 2,1 V.

VI. PROJEKTOVANJE ŠTAMPANE PLOČE GLUKOMETRA KORIŠĆENJEM ALTIIUM DESIGNERA

Na slici 6. prikazana je električna šema projektovanog glukometra sa linijama veza, simbolima komponenata i unutrašnjim konfiguracijama integrisanih kola, realizovana u *Altium Designer*-u [8].

U donjem levom uglu električne šeme mogu se uočiti konfiguracije sa dva regulatora napona. Krajnja leva konfiguracija je sastavljena pomoću regulatora napona TPS76901DBVT, dva elektrolitska kondenzatora i dva otpornika. Ovaj podesivi regulator napona je deliteljem napona preko otpornika vrednosti $R_3=130\text{ k}\Omega$ i $R_4=180\text{ k}\Omega$ respektivno, podešen na fiksni izlazni napon od 2,1 V.

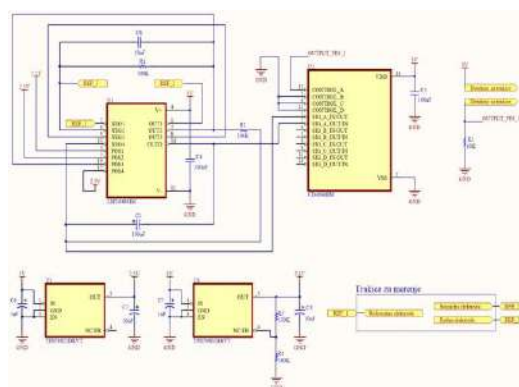
Druga konfiguracija je sa regulatorom napona TPS76925DBVT i dva elektrolitska kondenzatora i ona na izlazu daje fiksni napon od 2,5 V po *datasheet*-u.

U gornjem levom uglu se nalazi integrisano kolo LMC6484IM sa četiri operaciona pojačavača. Konfiguracija prvog pojačavača je ispitivanje da li će se neki napon generisati sa referentne elektrode koja je povezana na invertujući ulaz pojačavača (pin 2), dok se na neinvertujućem ulazu (pin 3) nalazi referentni napon od 2,1 V.

Radna elektroda je povezana na invertujući kraj drugog operacionog pojačavača (pin 6), a fiksni napon od 2,5 V sa regulatora napona TPS76925DBVT je prosleđen na neinvertujući ulaz (pin 5). Drugi pojačavač služi kao konvertor struje u napon (transimpedansni pojačavač), jer struja teče kroz radnu elektrodu od invertujućeg ulaza i otpornika R_1 iz povratne sprege, te njihov proizvod daje napon na tom ulazu. Izlaz iz ovog pojačavača (pin 7) se prosleđuje na sledeća dva: na neinvertujući ulaz trećeg pojačavača (pin 10) i on čini bafer impedanse, tako da se prosleđeni napon na izlazu (pin 8) šalje se na AD konvertor mikrokontrolera. Na četvrti pojačavač se napon prosleđuje na invertujući ulaz (pin 13) u kolo integratora i čiji izlaz (pin 14) se prosleđuje na prekidač CD4066.

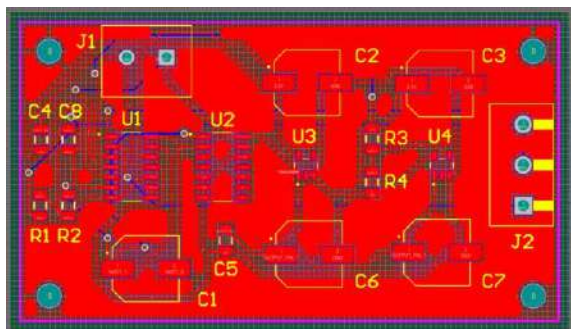
Četverostrani bilateralni prekidač CD4066 se nalazi u gornjem desnom delu električne šeme. Kolo za resetovanje mora biti uključeno da bi se ustanovilo vreme početka pre svakog vremenskog perioda integracije. Vreme završetka integracije nastupa kada se očita merenje. CD4066 kratkospaja kondenzator C_1 ako digitalni izlaz (OUTPUT_PIN_2) pređe u stanje logičke jedinice. Neiskorišćeni kontrolni ulazi povezani su sa masom. Digitalni izlaz sa prekidača se unosi na AD konvertor mikrokontrolera kada je data vrednost napona na izlazu očitana.

Na slici u donjem desnom uglu je prikazan blok dijagram elektroda unutar testne trakice sa izlaznim pinovima. Krajnji gornji desni ugao slike prikazuje detektor testne trakice i on predstavlja sam ulaz u glukometar. Preko pull-down otpornika je ostvarena konfiguracija detektora i sa pina OUTPUT_PIN_1 se prenosi stanje logičke jedinice ako je na visokom naponskom nivou.



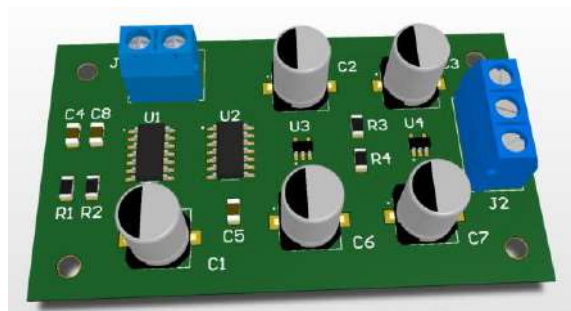
Slika 6. Električna šema glukometra u Altium Designer-u.

Na slici 7. je prikazan izgled pločice nakon rutiranja koje je odrađeno opcijom *Auto Route* kojom program pronalazi optimalnije putanje za linije veza između komponenata i razlivanja mase opcijom *Polygon Pour*. Obzirom da su za projektovanje ove ploče izabrane SMD tipovi kućišta komponenata, broj via je veći nego što bi bio da su korišćeni tipovi kućišta through-hole. Crvenom bojom prikazane su izrutirane linije veza sa gornje strane ploče (*Top Layer*) dok su plavom bojom prikazane veze sa donje strane (*Bottom Layer*).



Slika 7. Izgled pločice nakon rutiranja i razlivanja mase.

Na taj način su svi koraci za projektovanje pločice izvršeni i na slikama 8. i 9. su prikazani finalni izgled pločice u 3D-u, kao i spisak komponenata respektivno.



Slika 8. Konačan izgled pločice prikazan u 3D-u.

Comment	Description	Designator	Footprint	Libref	Quantity
Cap Pol1	Polarized Capacitor (Radial)	C1, C2, C3, C4, C5	CAP_881P2A100P	Cap Pol1	5
Cap	Capacitor	C4, C5, C6	C1006	Cap	3
OST7A003161	Connector Terminal Block 2 Position Sems Solder Straight Thru-hole 14	J1	*OST_CST7A003161	OST7A003161	1
1765520008	Conn Terminal Block 3 POS 5.08mm Solder ST Thru-hole 15A	J2	*1765520008	1765520008	1
Res1	Resistor	R1, R2, R3, R4	C1006	Res1	4
Res1	Resistor	R5	ANGU_R3	Res1	1
LMC6484M	Ultra-low Bias Current, Precision CMOS Rail-to-Rail Input and Output Quad Operational Amplifier 16-SOIC-40 RoHS	U1	SOIC127P0000175-149	LMC6484M	1
CD4068M	70-µA, 20-V, 1:1 (PSR), 4-channel analog buffer 14-SOIC-16 RoHS	U2	SOIC127P0000175-149	CD4068M	1
TPS7A02SD8V7	2.5V-4.5V input, 1A, high efficiency step-down buck converter in SOT23 and SOT143 package 5-SOT-23-40 RoHS	U3	SOT195P0000145-0N	TLV425680BVT	1
TPS7A02SD8V7	2.5V-4.5V input, 1A, high efficiency step-down buck converter in SOT23 and SOT143 package 5-SOT-23-40 RoHS	U4	SOT195P0000145-0N	TLV425680BVT	1

Slika 9. Spisak korišćenih materijala (BOM - Bill Of Materials).

VII. ZAKLJUČAK

Veliki nedostatak današnjeg glukometra je to što su instrumenti i test trake jednog proizvođača nekompatibilni sa onima drugog proizvođača. Stoga bi bilo korisno ako bi se mogao razviti uređaj koji bi mogao da primi većinu test traka dostupnih na tržištu i koji bi takođe bio skalabilan.

U ovom radu prikazan je teorijski pregled i dokumentacija o samom uređaju i bolesti za koju je namenjen, kao i princip rada uređaja. I konačno, izvršena je realizacija jednog kompletiranog glukometra u programu *Altium Designer*.

ZAHVALNICA

Autor se zahvaljuje mentoru prof. dr Danijelu Dankoviću na pruženom znanju i pomoći. Izražava se zahvalnost i mast. inž. Nikoli Mitroviću na korisnim sugestijama, smernicama i stručnoj pomoći pri izradi rada.

LITERATURA

- [1] Anoop A. E., „Madhu Mohan N., Simulation of a Multi-Strip Blood Glucometer“, https://www.researchgate.net/publication/269631389_Simulation_of_a_Multi-Strip_Blood_Glucometer
- [2] Yanez, M. „Glucose Meter Fundamentals and Design“, January 2013, https://www.nxp.com/docs/en/application-note/AN4364.pdf?fbclid=IwAR1nXeVXhQBmJ_g-lU_fIQPhUJpSDBLeRXOTWz7rk0rJPNtWzRyPaMDfAc
- [3] Maxim Integrated. Now Part of Analog Devices. <https://www.maximintegrated.com/en/design/technical-documents/tutorials/4/4659.html>
- [4] DiCristina, J., „BLOOD GLUCOSE METERS“, <https://www.shorturl.at/hiAQ1>
- [5] Texas Instruments, „Operational Amplifier“, LMC6484 Datasheet, June 2020 https://www.ti.com/lit/ds/symlink/lmc6484.pdf?ts=1631799235189&ref_url=https%253A%252F%252Fwww.google.com%252F
- [6] Texas Instruments, „Bilateral Switch“, CD4066 Datasheet, February 2020, <https://www.ti.com/lit/ds/symlink/cd4066b-mil.pdf>
- [7] Texas Instruments, „LINEAR REGULATORS“, TPS769xx Datasheet, May 2001, https://www.ti.com/lit/ds/symlink/tps769.pdf?ts=1631813303622&ref_url=https%253A%252F%252Fwww.google.com%252F
- [8] Altium Designer – PCB Design Software, dostupno na <https://www.altium.com/altium-designer/>

EMC Compliance Testing LED Luminaire in Frequency Bandwidth 9 kHz – 30 MHz

Natalija Antić, Vesna Javor

Abstract – The main goal of this paper is testing levels of conductive electromagnetic interference emitted by LED luminaire (LED, *Light Emitting Diode*) according to the standard CISPR 15 (EN 55015) in the frequency range from 9 kHz to 30 MHz. Line Impedance Stabilization Network (LISN) ROHDE & SCHWARZ ENV216 is placed between the power supply and the Equipment Under Test (EUT) in order to ensure precise impedance at the EUT power supply connector and to prevent the influence of high frequency noise of the power supply on the measurement results. ROHDE & SCHWARZ ESRP3 EMI Test Receiver that is controlled remotely using ELEKTRA software is used to measure EMC interference in accordance with relevant standards. The conclusion is that conductive electromagnetic interference emitted by the tested luminaire is above the limits of the prescribed standards. Afterwards, the testing is repeated with modified wiring method so that electromagnetic compatibility is verified.

I. INTRODUCTION

All electric devices or installations influence each other when interconnected or close to each other. The purpose of electromagnetic compatibility (EMC) is to keep all those side effects under reasonable control. The Electromagnetic Compatibility Directive 2014/30/EU ensures that electrical and electronic equipment does not generate and is not affected by electromagnetic disturbance [1].

II. TEST SETUP

Conducted disturbances at the electric power supply interface are measured by applying instrumentation, test sites, procedures and methods as indicated in the references of Table I.

TABLE I
OVERVIEW OF STANDARDIZED CONDUCTED DISTURBANCE MEASUREMENT MODES

Interface	Frequency range	Reference
Electric power supply interface	9kHz to 30MHz	CISPR 16-1-1 (receiver) CISPR 16-1-2 (ancillary equipment: AMN) CISPR 16-2-1 (Measurement method)

N. Antić is with the Department of Power Engineering, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: natalija.antic@elfak.rs.

V. Javor is with the Department of Power Engineering, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: vesna.javor@elfak.ni.ac.rs.

Limits for the assessment of conducted disturbance voltages at the AC or DC electric power supply interface terminals for the frequency range from 9 kHz to 30 MHz are given in Table II.

TABLE II
DISTURBANCE VOLTAGE LIMITS AT THE ELECTRIC POWER SUPPLY INTERFACE

Frequency range	Limits ^a dB(μV)	
	Quasi-peak	Average
9 kHz to 50 kHz	110	-
50 kHz to 150 kHz	90 to 80 ^b	-
150 kHz to 0.5 MHz	66 to 56 ^b	56 to 46 ^b
0.5 MHz to 5.0 MHz	56 ^c	46 ^c
5 MHz to 30 MHz	60	50

^a At the transition frequency, the lower limit applies.
^b The limit decreases linearly with the logarithm of the frequency in the ranges 50 kHz and 150 kHz to 0.5 MHz.
^c For lighting equipment incorporating exclusively electrodeless lamps, the limit in the frequency range of 2.2 MHz to 3.0 MHz is 73 dB(μV) quasi-peak and 63 dB(μV) average.

A. Test equipment

The output terminals of the artificial mains network and the electric power supply terminals of the EUT are positioned at 0.8 ± 0.05 m apart and connected by the two power conductors of a flexible three-core or two-core cable of 0.8 ± 0.05 m length (Fig. 1). If the electric power supply cable of the EUT is longer than necessary to be connected to the AMN, the length of this cable in excess of 0.8 m should be folded back and forth parallel to the lead, so as to form a bundle with a length between 0.3 m and 0.4 m [2].

Ground reference plane (GRP) is an essential part of the conducted emission test (Fig. 2). A proper measurement is impossible without a GRP. Even a Class II EUT without safety earth connection must be tested over a GRP, since it provides a return path for stray capacitance from the EUT.

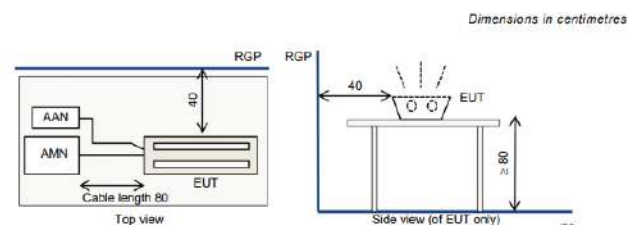
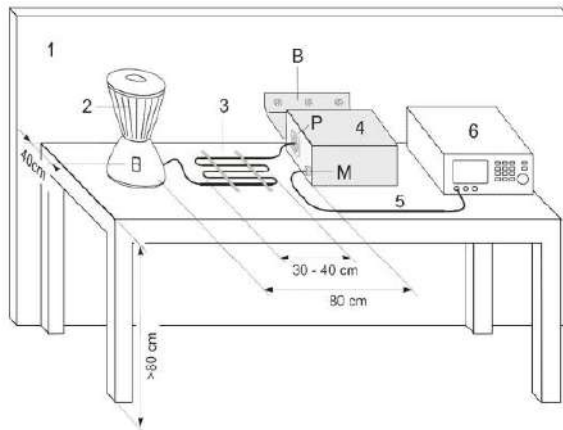


Fig. 1. Vertical RGP setup.

The GRP should be:

- at least 2 m x 2 m, and at least 0.5 m in both directions larger than the boundary of the EUT;
- made of copper, aluminium or steel, though the thickness is not too important;
- bonded to the local supply safety earth (this is for safety only and it is not necessary for the measurement);
- bonded by a very short, low-inductive strap to the reference terminal of the AMN/LISN.



- 1 – Ground reference plane
 2 – Equipment Under Test
 3 – Folded cable
 4 – LISN
 5 – Coaxial cable for connecting LISN and EMI test receiver
 6 – EMI test receiver
 B – Connection on GRP
 M – Connector on EMI test receiver
 P – EUT connector

Fig. 2. Standard compliant setup for conducted emission measurements.

B. Equipment Under Test (EUT)

Tested luminaire contains three LED modules, LED driver with DALI (Digital Addressable Lighting Interface), EM (Emergency) converter and the battery pack that supplies one LED module in case of power supply failure. The wiring diagram of EUT is shown in Fig. 3.

If the EUT has the possibility to reduce the output power (dimming), the electromagnetic disturbance of the EUT shall be measured at the maximum and minimum light output. If the EUT is equipped with battery and EM converter, in the event of a power failure, the level of electromagnetic interference is also measured in the case when the power supply is realized with battery (Emergency mode).

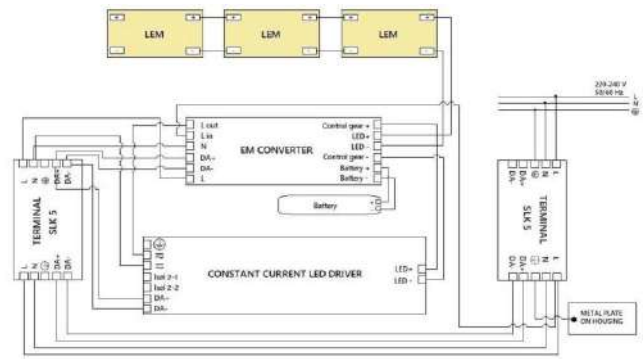


Fig. 3. EUT wiring diagram.

III. MEASUREMENT RESULTS

Measurements are carried out in normal laboratory conditions. The luminaire is placed on an insulating table at 80 cm height and a distance of 40 cm from the vertical reference ground plane, according to standard requirements.

A. Operating mode: Maximum dimlevel

The first measurement is performed at the maximum output power of the luminaire. After scanning process, ELEKTRA software automatically generates a graphical representation of the results (Fig. 4) as well as the table with listed frequencies on which the critical points are present (Fig. 5). On the frequency 0.166 MHz and 0.332 MHz for average signal detector (CAV – CISPR Average) measured signals have values that are above the prescribed standard limit. The average signal value is 64.17 dB μ V at the frequency 0.166 MHz. Maximum CAV value for this frequency is 55.17 dB μ V, hence the measured signal value is 9 dB above the limit prescribed by the standard. At a frequency of 0.332 MHz, the CAV signal value was exceeded by 6.35 dB.

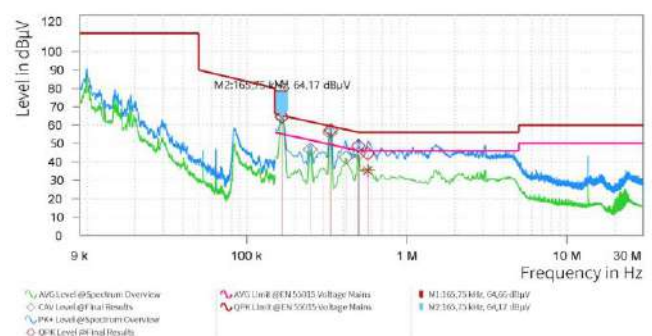


Fig. 4. Results' chart for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, maximum dimlevel.

Frequency [MHz]	QPK Level [dBμV]	QPK Limit [dBμV]	QPK Margin [dB]	CAV Level [dBμV]	CAV AVG Limit [dBμV]	CAV Margin [dB]	Correction [dB]	Line	Meas. BW [kHz]
0.166	64.66	65.17	0.51	64.17	55.17	-9.00	9.50	L1	9,000
0.249				46.64	51.79	5.15	9.50	L1	9,000
0.332				55.75	49.39	-6.35	9.51	L1	9,000
0.335	57.25	59.34	2.09				9.51	L1	9,000
0.416				42.39	47.54	5.15	9.51	N	9,000
0.497	48.59	56.06	7.47				9.51	N	9,000
0.999				44.22	46.02	1.80	9.51	N	9,000
0.569	44.45	56.00	11.55				9.52	N	9,000

Fig. 5. Listed frequencies at which the critical points are present for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, maximum dimlevel.

The following practical precaution measures need to be taken into account in a lighting system to minimize EMC:

- Minimize the differential mode loop area of the lamp wires going from the driver to the light source by keeping the wires close together (bundling). This will minimize the magnetic field and reduce the radiated EMI.
- Keep the length of the incoming mains wire inside the luminaire as short as possible.
- Do not route any wiring over and/or along the driver enclosure to avoid any coupling/crosstalk with internal components of the driver [3].

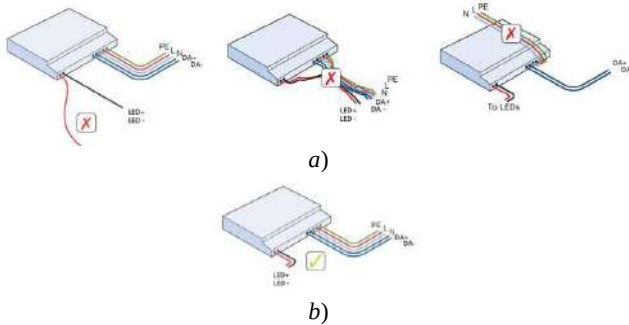


Fig. 6. a) Examples of wrong positioning of the conductors and b) the right position of the conductors.

Two wiring irregularities were observed which are a potential cause of poor results CAV signal values. A conductor that connects the L_{in} connector on the EM converter and the L on the SLK5 terminal is longer than required and is positioned so that it lies below the EM converter. By connecting the conductor to the opposite terminal, the connection of these connections is achieved with a shorter conductor so that passing of the conductor under the EM converter is avoided (Fig. 7).

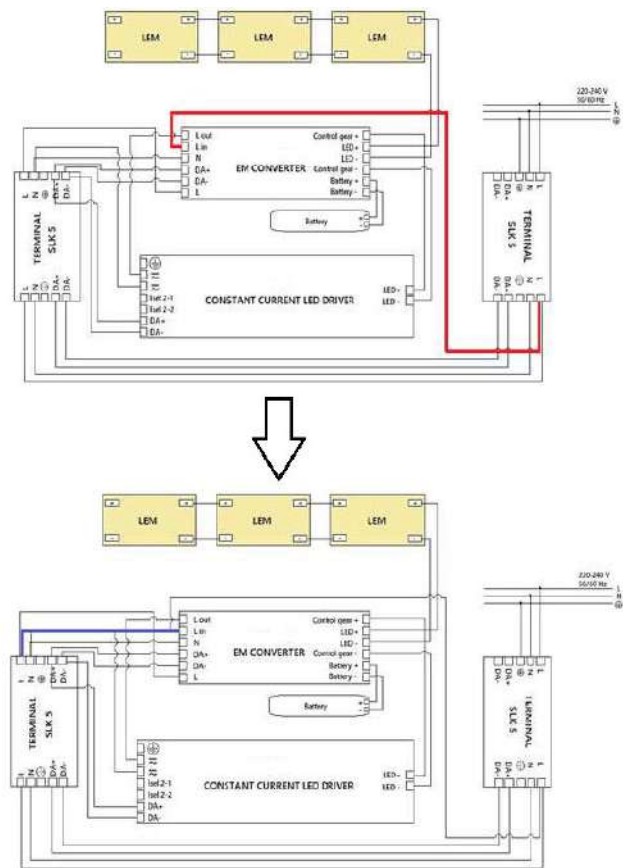


Fig. 7. Modification of EUT wiring configuration in order to improve EMC performance.

Results' chart of repeated measurements on luminaire with modified wiring configuration is shown in Fig. 8 and the table with measured signal values is shown in Fig. 9.

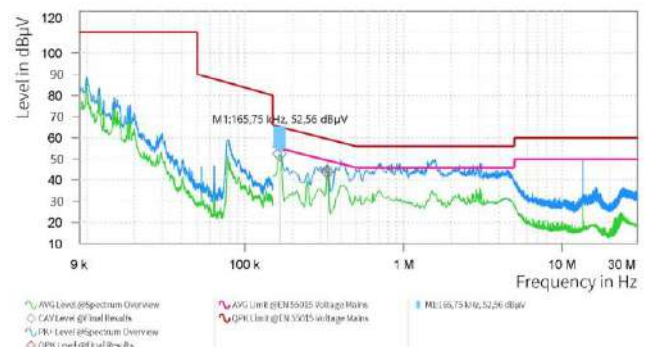


Fig. 8. Results' chart for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, maximum dimlevel – modified EUT wiring configuration.

Frequency [MHz]	QPK Level [dBμV]	QPK Limit [dBμV]	QPK Margin [dB]	CAV Level [dBμV]	CAV: AVG Limit [dBμV]	CAV Margin [dB]	Correction [dB]	Line	Meas. BW [kHz]
0.166				52.56	55.17	2.61	9.50	L1	9,000
0.332				43.93	49.39	5.46	9.51	L1	9,000

Fig. 9. Listed frequencies on which the critical points are present for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, maximum dimlevel – modified EUT wiring configuration.

The critical signal values were again measured at the frequencies 0.166 MHz and 0.332 MHz, but there is a satisfactory margin of 2.61 dB and 5.46 dB, respectively, and the luminaire successfully passed the testing.

B. Operating mode: Minimum dimlevel

Figure 10 shows the test results at EUT's minimum light output. The blue line on the graph shows the peak value interference signal. A high value of this magnitude is detected at the frequency 1.577 MHz which requires re-scanning to calculate the quasi-peak value at this critical frequency. After re-scanning, the quasi-peak value of 43.97 dBμV is measured. The maximum allowed quasi-peak value for this frequency is 56 dBμV (red line at the graphic), therefore luminaire passed test in this operating mode. Table with listed results is shown in Fig. 11.

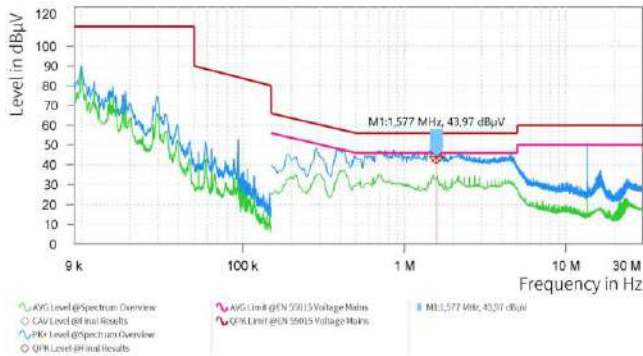


Fig. 10. Results' chart for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, minimum dimlevel.

Frequency [MHz]	QPK Level [dBμV]	QPK Limit [dBμV]	QPK Margin [dB]	CAV Level [dBμV]	CAV: AVG Limit [dBμV]	CAV Margin [dB]	Correction [dB]	Line	Meas. BW [kHz]
1.577	43.97	56.00	12.03				9.52	L1	9,000

Fig. 11. Listed frequency on which the critical point is present for the conducted electromagnetic interference at the electric power supply interface, frequency range 9kHz – 30MHz, minimum dimlevel.

C. Operating mode: Emergency mode

This operating mode is obtained by disconnecting Line Impedance Stabilization Network from the main power supply. Then the battery contained in the luminaire

supplies one led module of luminaire for a limited time interval. Although the EUT is disconnected from mains supply, conducted disturbance signals on coaxial connector of the Line Impedance Stabilization Network can still be detected and measured on EMI test receiver. In Fig. 12 the test result chart in this mode is shown.

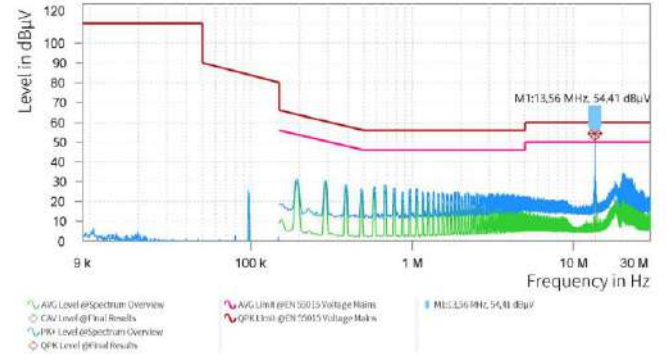


Fig. 12. Results' chart for the conducted electromagnetic interference at the electric power supply interface, frequency range 9 kHz – 30 MHz, emergency mode.

On the frequency of 13.56 MHz the quasi-peak value of 54.41 dBμV is measured. Given that the quasi-peak limit for this frequency is 60 dBμV, the EUT has successfully passed the test in this mode with a margin of 5.59 dBμV.

IV. CONCLUSION

The tested luminaire meets the requirements of the standards after the adoption of recommendations for a wiring configuration that improves EMC performance. Measured levels of emitted conductive interference at the main supply lines are below the prescribed limits for the frequency range from 9 kHz to 30 MHz for all operating modes. Test method and measuring equipment must be used according to strictly established rules defined by the standards. If luminaire is capable for dimming emitted light, except for measuring on the main supply lines, it is necessary to perform measurements of emitted interference on the control lines for all operating modes considered in this paper.

REFERENCES

- [1] Study on the Evaluation of the Electromagnetic Compatibility Directive 2014/30/EU (EMCD) Final Report; doi: 10.2873/139099, ISBN: 978-92-76-22757-1; Publications Office of the European Union, Luxembourg 2021.
- [2] CISPR 15 ED9 IEC CIS/F: Limits and methods of measurement of radio disturbance characteristics of electrical lighting and similar equipment, International Electrotechnical Commission, IEC, 2018.
- [3] Trust Sight LED Emergency Driver Design-in Guide, Philips Lighting Holding, 05/2017.

Elektronski sistem pametne kuće

Marina Stojilković, Aleksa Tripković, Pavle Mitić, Nikola Milošević, Lazar Rašović

Sadržaj – U ovom radu prikazana je detaljna i sofisticirana realizacija pametne kuće sa određenim senzorima i bezbednosnim sistemima pomoću mikrokontrolera Arduino. Samo kolo je prvobitno realizovano u okruženju Tinkercad, a kasnije je izrađeno u fizičkom obliku. Cilj ovog rada jeste da se na konkretan način podigne ljudska svest o pametnim kućama i njihovim sposobnostima.

I. UVOD

Sam razvoj pametnih kuća krenuo je i sa razvojem tehnologije. Prvo su krenuli kompjuteri, telefoni, pa je pređeno i na razvoj saobraćaja i pametnih kuća. Kuće koje uz pomoć senzora regulišu određene uslove koje mi želimo, u zavisnosti od našeg uslova, jesu pametne kuće. Ovakve kuće znatno doprinose očuvanju okoline i životne sredine, zato što mogu smanjiti potrebu za korišćenje neobnovljivih izvora energije (rude, nafta, zemni gas...) [1]. Pamente kuće su direktno umrežene sa njihovim korisnikom pomoću određenih aplikacija, interneta, bluetooth-a. Prvobitni zadatak pametnih kuća je bio da pomogne starijim osobama ili osobama sa invaliditetom da imaju normalan život samostalno. Kasnije su krenule da se grade i u običnim uslovima života zbog svojih osobina. Ljudi su primetili da je život u takvim kućama postao mnogo jednostavniji. Dakle, pametne kuće su električni sistemi koji uz pomoć veštačke inteligencije pomažu ljudima [2].

II. OSNOVE ISTRAŽIVANJA

A. Arduino

Arduino je open-source (besplatna i otvorena) mikrokontrolerska platforma koja je nastala u Italiji 2011. godine. Njegov hardver se sastoji od ploče sa pinovima (ulazima i izlazima) i samog mikrokontrolera.

Od pinova postoje:

- ulazni pinovi (A0 – A5)
- izlazni pinovi (1 – 13)
- pin za uzemljenje GND (eng. ground)
- pin za vezu sa kompjuterom.

Arduino okruženje programira se u jeziku veoma sličnom C i C++ programskom jeziku.

M. Stojilković, A. Tripković, P. Mitić, N. Milošević, L. Rašović su učenici prve godine računarskog smera, Gimnazija „Bora Stanković“, Vožda Karađorđa 27, 18000 Niš, Srbija, E-mail: marinalara2812@gmail.com, atripko07@gmail.com, pavle.mitic2006@gmail.com, nikola.milosevic060@gmail.com, lazarrasovic11@gmail.com.

Radno okruženje Arduino IDE (slika 1) je se sastoji iz određenih tastera koji imaju svoju funkciju (u gornjem levom uglu, od leva ka desno):

1. Pokrenuti kompajler (eng. compiler)
2. Ažurirati (eng. upload) kod na Arduino
3. Novi dokument
4. Otvoriti postojeći dokument
5. Sačuvati postojeći dokument.



Slika 1. Arduino IDE.

B. Protoploča

Proto ploča (eng. protoboard) je osnova za izradu elektronskih prototipova. Omogućava jednostavno spajanje elektronskih komponenata, bez potrebe za lemljenjem i pravljenjem trajnih spojeva. Za spajanje se koriste izolovani provodnici.

Vertikalne linije označavaju da će komponente povezane u istoj vertikali biti na istom potencijalu. Da bi se dovela struja, potreban je izvor energije koji će se sprovesti na plus (crvena horizontala) i na minus (plava horizontala) šinama.

C. Otpornici

Otpornik je jedna od osnovnih komponenti u elektronici i služi za ograničavanje protoka struje u kolu, kao i za uspostavljanje smanjenja napona. [3]. Sastavljen je od dva provodnika povezana otpornim materijalom. Otpor se izražava u omima (Ω), a otpornost pojedinačnih otpornika beleži se pomoću određenih kombinacija boja na samom otporniku. Podela otpornika: stalni otpornici (pružaju otpor koji je unapred uspostavljen) i promenljivi otpornici (tj. potencijometri koji omogućavaju podešavanje otpora u opsegu od 0 do neke maksimalne vrednosti) [4].

D. Fotootpornici

Fotootpornik (slika 2) je otpornik čiji se električni otpor menja s promenom količine svetlosti. Ako svetlost padne na otpornik, on absorbira tu svetlost i izbacuje valentne elektrone u slobodnu zonu i samim tim dovodi do postojanja energije [5].



Slika 2. Fotootpornici.

E. Temperaturni senzor

Temperaturni senzori su komponente koje pri različitim temperaturama daju različite napone. Merenje temperature u ovom projektu vrši se temperaturnim senzorom TMP36 (slika 3) koji ima opseg od -40°C do 125°C i naponski izlaz 0 do 1,75V [6].



Slika 3. Temperaturni senzor TMP36.

F. Metal-oksid poluprovodnički gasni senzor

MQ3 (slika 4) je jedan od najkorišćenijih senzora u MQ seriji koji je i upotrebljen u ovom projektu. On je metal-oksidni poluprovodnički (eng. Metal Oxide Semiconductor - MOS) tip senzora. Kada je SnO_2 poluprovodnički sloj zagrejan na visoke temperature, kiseonik je apsorbovan na površini. U čistom vazduhu, elektroni u kalaj-oksidu privučeni su na kiseonikove molekule. Kao rezultat toga SnO_2 sloj postaje otporniji [7].



Slika 4. Gasni senzor MQ3.

G. Ultrazvučni senzor

U ovom sistemu se kao senzor detekciju pokreta koristi ultrazvučni senzor HC-SR04. Ovaj senzor radi po principu merenja vremena stizanja zvuka od predmeta do prijemnog modula. Senzor stvara ultrazvučne talase

frekvence 40 kHz koji se odbijaju od predmeta i vraćaju senzoru [8].



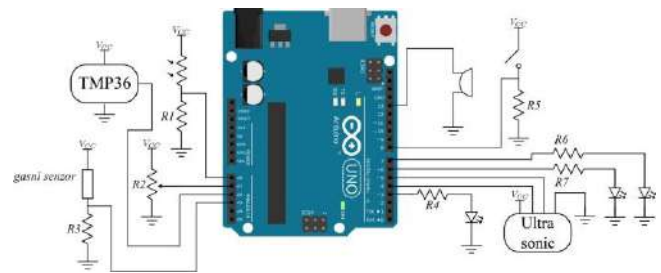
Slika 5. Ultrazvučni senzor.

H. Tinkercad

Tinkercad je besplatno digitalno okruženje namenjeno za izradu i uređivanje kola. Tinkercad se može koristiti online. Na sajtu Tinkercad, ugrađeno je okruženje Arduino IDE što omogućava testiranje koda na samom sajtu.

III. PROJEKTOVANJE DELOVA ELEKTRONSKOG SISTEMA PAMETNE KUĆE

Prvo je potrebno izraditi online simulaciju, a kasnije napraviti program na Arduino IDE platformi. Pametna kuća se sastoji iz raznih bezbednosnih sistema i senzora. Kompletan električni šema data je na slici 6.



Slika 6. Električna šema elektronskog sistema pametne kuće.

A. Rasveta

Rasveta funkcioniše na sledeći način: kada nema svetlosti uključuju se svetla u ovom slučaju simulirana žutom LED diodom, a isključuju se kada ima svetlosti. Za ovaj rad, od hardvera potrebni su: otpornici (jedan od $1\text{ k}\Omega$ a drugi 330Ω), dioda (žute boje) i jedan foto-otpornik koji se spajaju na proto ploču, a potom i na Arduino. Za softver je pre svega u void setup() potrebno definisati pinove za ulaz (INPUT) i izlaz (OUTPUT) podataka pomoću pinMode() naredbe. U void loop() uvodi se promenljiva koja sa ulaznog pina povezanog sa foto-otpornikom čita i čuva količinu svetlosti pročitane sa tog foto-otpornika. Zatim je neophodno postaviti konsataciju koja glasi: ako nema svetla, potrebno je uključiti sijalicu, a u suprotnom je isključiti. Kod je dat na slici 7.

B. Protivpožarni sistem

Protivpožarni sistem pri detekciji dima aktivira piezo zujalicu kao obaveštenje za mogući požar. Za protiv požarni sistem od hardvera korišćeni su: MQ3 senzor, provodnici, piezo zujalica i otpornik od 10 kΩ koji se spajaju po pravilima na protoploču. Na početku softverskog koda u void setup() treba deklarirati ulazni pin za MQ3 i izlazni za piezo. Kod je prikazan na slici 8.

```

#include <Arduino.h>
void setup()
{
  pinMode(A0, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  int svetlo = analogRead(A0);
  delay(5);
  if(svetlo > 200) {
    digitalWrite(13, HIGH);
  }
  else{
    digitalWrite(13, LOW);
  }
}

```

Slika 7. Softver rasvete.

```

int piezo = A1;
int mq = A0;

void setup()
{
  pinMode(mq, INPUT);
  pinMode(piezo, OUTPUT);
}

void loop()
{
  float mqINPUT;
  mqINPUT = analogRead(mq);
  if(mqINPUT >= 390){
    digitalWrite(piezo,HIGH);
  }
  else{
    digitalWrite(piezo,LOW);
  }
}

```

Slika 8. Softver protivpožarnog sistema.

C. Termostat

Sistem funkcioniše tako što se na analogne ulaze dovode signali sa temperaturnog senzora i signal sa ulaznog potencijometra koji se koristi za zadavanje željene temperature. Međusobnim poređenjem ova dva ulaza radi se regulacija temperature uključivanjem grejanja ili hlađenja koji su simulirani crvenom i plavom diodom. Pregled trenutne i zadate temperature vrši se preko LCD displeja sa 16x2 karaktera. Od hardvera su nam potrebni: LCD 16x2 displej, otpornici, temperaturni otpornik TMP36, crvena i plava dioda i pet otpornika (četiri stalna i jedan promenljivi). U prvoj liniji softverskog koda, potrebno je deklarirati biblioteku koja omogućava interakciju sa LCD 16x2 displejom #include <LiquidCrystal.h>. U void setup() treba navesti određene pinove LCD-a, a u void loop() treba deklarirati promenljive koje čuvaju sadašnju i željenu temperaturu pročitane sa ulaza. Uz proporciju analogni ulaz pročitani sa

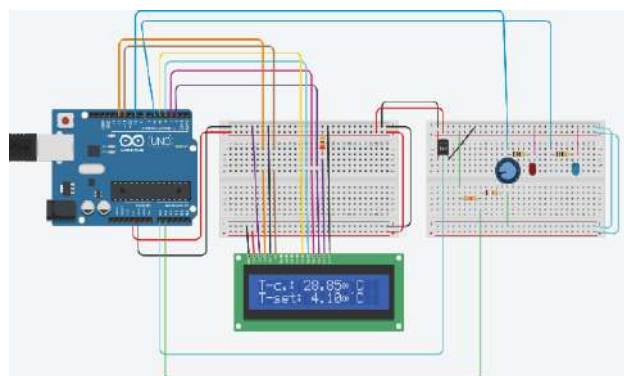
10-bitnog AD konvertora mikrokontrolera računa se u stepenima Celzijusovim. Sledeća je if.else() naredba u kojoj se navode uslovi. U if() grani proveravamo da li je ulaz sa temperaturnog senzora jednak 0V, ako jeste na displeju se prikazuje greska i prestaje se sa izvršavanjem daljeg koda. Ovo je urađeno kao sigurnosna mera u slučaju kvara na senzoru ili nedostatku istog. U else() grani postavljamo uslov: ako je sadašnja temperatura jednaka željenoj temperaturi, onda LED nisu uključene; ako je sadašnja temperatura veća od željene temperature, onda je samo plava LED uključena (simulacija hlađenja), u suprotnom samo crvena (simulacija grejanja). Napisani kod prikazan je na slici 9. Simulacija ovog dela sistema u Tinkercadu prikazana je na slici 10.

```

1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3
4 void setup() {
5   Serial.begin(9600);
6   lcd.begin(16, 2);
7 }
8
9 void loop() {
10  float sadašnjatemp = (float)analogRead(A0) * 0.45 - 40.00;
11  float zeljenatemp = (float)analogRead(A1) * 0.45 - 40.00;
12
13  lcd.clear();
14
15  if(analogRead(A0) == 0.0){
16    lcd.print("GRESKA");
17    delay(5000);
18    exit;
19  }
20  else{
21    lcd.print("T-C: ");
22
23    lcd.print(sadašnjatemp);
24    lcd.print("C");
25
26    lcd.setCursor(0, 2);
27    lcd.print("T-set: ");
28    lcd.print(zeljenatemp);
29    lcd.print("C");
30
31    delay(1000);
32  }
33  if(sadašnjatemp==zeljenatemp){
34    digitalWrite(6, LOW);
35    digitalWrite(9, LOW);
36  }
37  else if(sadašnjatemp>zeljenatemp){
38    digitalWrite(6, HIGH);
39    digitalWrite(9, LOW);
40  }
41  else if(sadašnjatemp<zeljenatemp){
42    digitalWrite(9, HIGH);
43    digitalWrite(6, LOW);
44  }
45 }

```

Slika 9. Softver termostata.



Slika 10. Simulacija kola termostata u Tinnkercad-u.

D. Ultrazvučni senzor za detekciju pokreta

Senzor pokreta funkcioniše tako što meri prvobitnu razdaljinu tela, a zatim računa sledeću poziciju tela. Upoređivanjem dolazimo do zaključka da li se telo pomerilo ili ne, ako jeste pokrenuto se piezo zujalica. Kako ovaj sistem ne bi konstantno radio, potreban je taster za uključanje i isključanje ovog sistema. Od hardvera su potrebni: provodnici, ultrazvučni senzor prekidač i otpornik. Na samom početku koda potrebno je uneti promenljive, kao i ulazne i izlazne podatke za prekidač i senzor. Potrebno je odraditi određenu matematiku da bi se dobile tačne vrednosti razdaljine tela zbog otpornosti određenih otpornika. Nakon toga potrebno je napisati if() naredbu za promenu položaja tela.

IV. PRAKTIČNA PROVERA ELEKTRONSKOG SISTEMA PAMETNE KUĆE

Kada su i hardverski i softverski delovi svih sistema završeni i testirani, usklađuju se na jednu protoploču i jedan Arduino. Na slici 11 prikazana je praktična realizacija pojedinih delova elektronskog sistema pametne kuće.

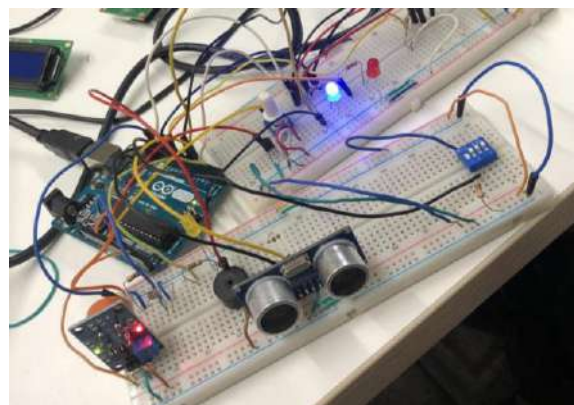
V. ZAKLJUČAK

Ova ideja može se integrisati u druge sisteme i takođe može se unaprediti tehnologijom daljinskog pristupa putem WiFi mreže koje bi omogućilo podešavanja i pregled podataka sa udaljenih lokacija putem pametnih uređaja.

ZAHVALNICA

Ovaj rad realizovan je u okviru projekta "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu.

Zahvalnost za realizaciju ovog projekta upućena je timu sa Elektronskog fakulteta u Nišu koji su podržali ideju, a posebno profesoru Milošu Marjanoviću koji nas je na jednostavan i zanimljiv način uputio u mikroelektroniku.



Slika 10. Realizacija elektronskog sistema pametne kuće na protoploči.

LITERATURA

- [1] Srdjan D. Stojanović, nedeljnik "Vreme", dostupno na: <http://vremeje.rs/sta-je-pametna-kuca/>
- [2] Organizacija ProCar, članak "Pametna Kuća", dostupno na: <https://www.procar.rs/blog/pametne-kuce-i-njihoveprednosti.html>
- [3] Organizacija STEM, otpornici, dostupno na: <https://www.stem.ba/arduino-elektronika/tutorijali/item/303-otpornici>
- [4] Elektroničar, blog „Praktična elektronika“, dostupno na: <https://prakticnaelektronika101.wordpress.com/komponente/otpornik/>
- [5] O fotootpornicima, dostupno na <https://fizis.rs/гимназија/iv-разред/физика-чврстог-стања/fotootpornik/>
- [6] Analog Devices, datasheet "TMP 36", dostupno na: <https://html.alldatasheet.com/html-pdf/49108/AD/TMP36/19/1/TMP36.html>

Različite realizacije binarnog sabirača

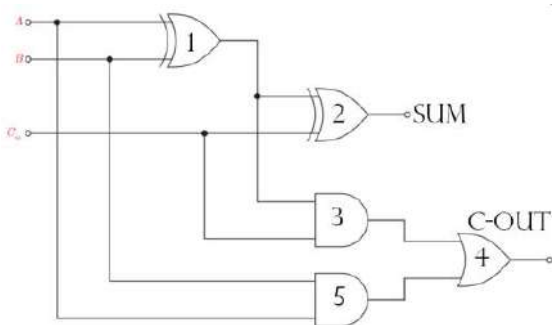
Vuk Radonjić, Teodora Obradović, Đorđe Držić, Aleksandar Crnatović, Aljoša Paunović

Sadržaj – U ovom radu predstavljene su dve različite realizacije binarnog sabirača, korišćenjem logičkih kola i Arduina. Predstavljeni prototip je praktično realizovan, korišćenjem web aplikacije Tinkercad i korišćenjem realnih komponenta i razvojnog okruženja Arduino.

I. PRINCIP RADA POLU-SABIRAČA I POTPUNOG SABIRAČA

Četvorobitni kalkulator je sastavljen iz 4 potpuna sabirača. Potpuni sabirači se sastoje od tri XOR kola, dva AND kola i jednog OR kola (slika 1). [1]

U potpunom sabiraču nalaze se tri ulaza – bit prvog broja (A), bit drugog broja (B) i prenos koji potiče od sabiranja prethodnih cifara prvog i drugog broja (C_{IN}). Prvi sabirač za C_{IN} koristi vrednost 0. Zbog toga se C_{IN} kod prvog sabirača može ignorisati. A i B prvo prolaze kroz XOR kolo označenu brojem 1 i kroz AND kolo označeno brojem 5 (slika 1). Pošto je C_{IN} za prvi potpuni sabirač 0, XOR kolo broj 2 će imati vrednost 1 ako XOR kolo broj 1 ima vrednost 1, a vrednost 0 ako XOR kolo broj 1 ima vrednost 0. Zbog toga će kod prvog sabirača SUM imati istu vrednost kao XOR broj 1. Vrednost promenljive C_{OUT} će biti jednaka 1 samo ako je vrednost AND kola broj 5 jednaka 1. Za sva ostala sabiranja se uzima da je vrednost promenljive C_{IN} jednaka vrednosti promenljive C_{OUT} prethodnog sabirača.



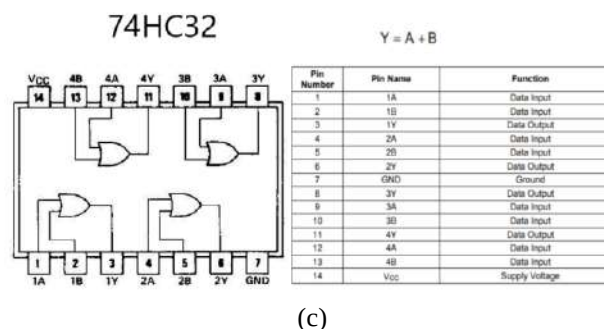
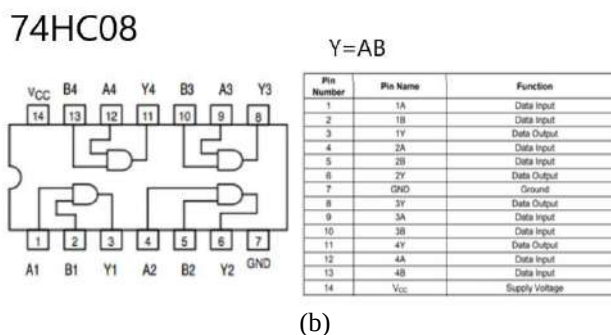
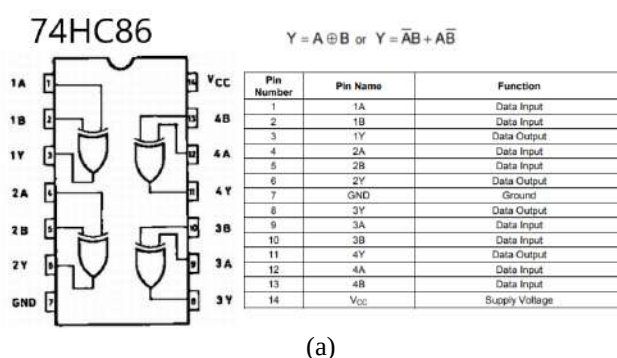
Slika 1. Šema potpunog sabirača [2].

V. Radonjić, T. Obradović, Đ. Držić, A. Crnatović, A. Paunović su učenici specijalizovanog odeljenja za informatiku i računarstvo, Gimnazija „Bora Stanković“, Vožda Karadžića 27, 18000 Niš, Srbija, E-mail: vuk.radonjic@borastankovic.edu.rs, tmobradovic@gmail.com, djordje.drizic@borastankovic.edu.rs, aleksandar.crnatovic@borastankovic.edu.rs, aljosa.paunovic@borastankovic.edu.rs

II. PROJEKTOVANJE ČETVOROBITNOG SABIRAČA KORIŠĆENJEM LOGIČKIH KOLA

A. Komponente

Jedan način da se projektuje četvorobitni sabirač je korišćenjem logičkih kola. Logička kola koja se koriste su 74HC86 (slika 2 (a)), 74HC08 (slika 2 (b)) i 74HC32 kolo (slika 2 (c)). [2]



Slika 2. Logička kola za projektovanje četvorobitnog sabirača: (a) 74HC86, (b) 74HC08 i (c) 74HC32.

Osim ovih logičkih kola koristi se 8 prekidača. Izlazni deo kola čini 5 LED dioda koje prikazuju rezultat binarnog sabiranja.

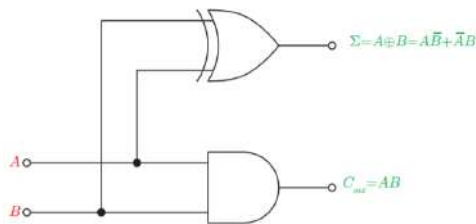
B. Polu-sabirač

Binarno sabiranje korišćenjem polu-sabirača se vrši na način prikazan u tabeli 1:

TABELA I
TABLICA ISTINITOSTI POLU-SABIRAČA

Ulaz		Izlaz	
A	B	C _{OUT}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Zapaža se obrazac koji se može formirati na osnovu XOR i AND kola, odnosno 74HC86 i 74HC08 logičkih kola. Šema polu-sabirača je prikazana na slici 3.



Slika 3. Šema polu-sabirača [3].

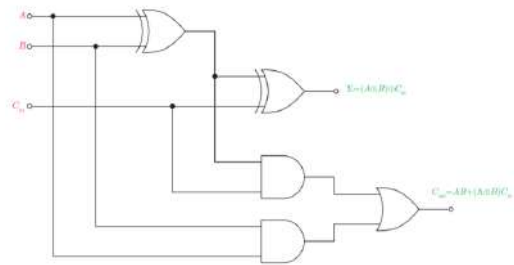
D. Potpuni sabirač

Binarno sabiranje korišćenjem potpunog sabirača se vrši na način prikazan u tabeli 2.

TABELA II
TABLICA ISTINITOSTI POTPUNOG SABIRAČA

Ulaz			Izlaz	
A	B	C _{IN}	C _{OUT}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

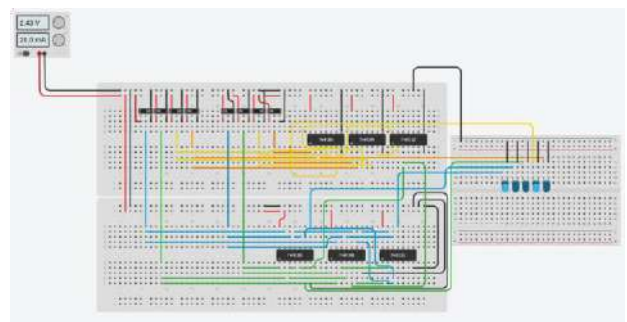
Zapaža se obrazac koji se može formirati pomoću XOR, OR i AND kola, odnosno 74HC86, 74HC32 i 74HC08 logičkih kola. Šema potpunog sabirača je prikazana na slici 4.



Slika 4. Šema potpunog sabirača [3].

A. Realizacija četvorobitnog sabirača

Pomoću polu-sabirača i sabirača može se formirati četvorobitni sabirač – električno kolo koje može da sabere dva binarna broja sa najviše 4 cifre. Četvorobitni sabirač se može sastaviti u web aplikaciji Tinkercad [4] (slika 5). Na ulazu su dva broja (A i B). U ovom primeru broj A je 101, a B je 1101. Kada se brojevi A i B saberu na izlazu se dobija broj 10010. To je prikazano pomoću LED dioda.



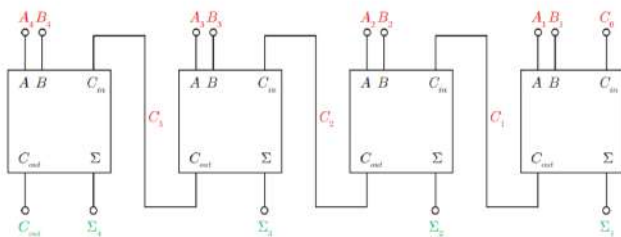
Slika 5. Realizacija četvorobitnog sabirača u Tinkercad-u.

Za svaku cifru na mestu n u brojevima A i B (čitano zdesna na levo) važi tablica istinitosti prikazana u tabeli 3.

TABELA III
TABLICA ISTINITOSTI ČETVOROBITNOG SABIRAČA

Ulaz			Izlaz	
C _{n-1}	A _n	B _n	Σ _n	C _n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Ovo važi za svaku cifru u brojevima A i B. Iz tog razloga se četvorobitni sabirač sastoji iz 4 potpunog sabirača koji rade paralelno (slika 6). Kod prvog potpunog sabirača C_0 ima vrednost 0.



Slika 6. Blok dijagram četvorobitnog sabirača [3].

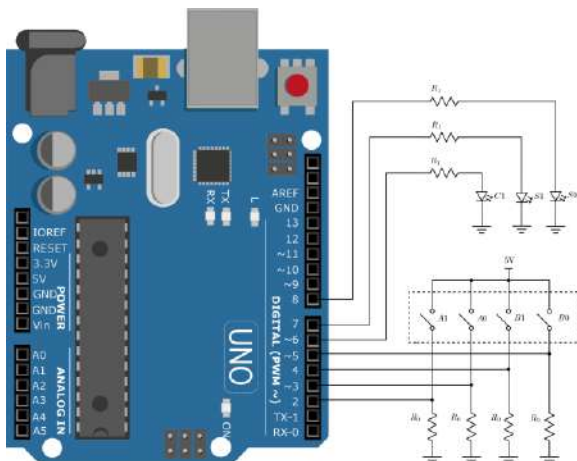
III. PROJEKTOVANJE POTPUNOG DVOBITNOG SABIRAČA KORIŠĆENJEM ARDUINA

A. Arduino razvojno okruženje

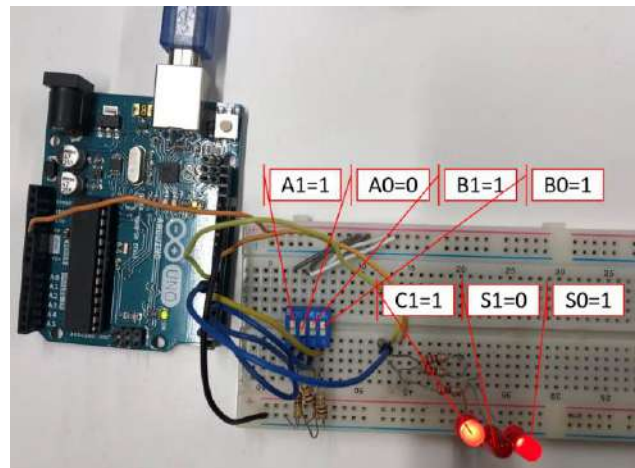
Arduino je zajednički naziv za familiju mikrokontrolerskih platformi. Namenjen je razvoju aplikacija gde je potrebno na jednostavan način ostvariti vezu između hardverskih i softverskih komponenti [5].

B. Hardver

Postoji 4 prekidača (A_1, A_0, B_1, B_0) koji su konfigurisani sa pull-down otpornicima (slika 7). Vrednost pull-down otpornika je 10 k Ω . U praktičnoj realizaciji je korišćen jedan DIP prekidač sa 4 prekidača (slika 8). Prekidač A_1 predstavlja prvu cifru broja A, a prekidač A_0 predstavlja drugu cifru broja A, čitajući sleva na desno. Analogno su postavljeni i prekidači B_1 i B_0 . Izlazni deo kola čine 3 LED diode koje prikazuju rezultat binarnog sabiranja. Za ograničenje struje kroz LED diode korišćeni su otpornici čija je otpornost 220 Ω .



Slika 7. Realizacija potpunog dvobitnog sabirača korišćenjem Arduinoa.



Slika 8. Praktična realizacija četvorobitnog sabirača.

C. Kod

Komandama #define pinovima su dodeljeni nazivi, kako bi se kod lakše unapređivao, menjao i bolje razumeo. Definisane su promenljive a0, a1, b0, b1 tipa bool. U voidsetup() je definisano koji su pinovi za ulaz, a koji za izlaz. U voidloop() je očitana vrednost malopre definisanih promenljivih pomoću unosa korisnika. Zatim su izračunati bitovi manje vrednosti pomoću odgovarajuće logike. To je smešteno u promenljive s0 i c0. Logikom za sabiranje cifara veće vrednosti dodeljena je vrednost promenljivoj s1, a logikom izračunavanja prenosa za bitove više vrednosti dodeljena je vrednost promenljivoj c1. Zavisno od vrednosti promenljivih c1, s1 i s0 uključuju se LED diode u kolu. Kod je prikazan u nastavku:

```
#define a0pin 3
#define a1pin 2
#define b0pin 5
#define b1pin 4
#define sum0pin 8
#define sum1pin 7
#define carrypin 6
bool a0,a1,b0,b1;
voidsetup(){
  pinMode(a0pin,INPUT);
  pinMode(a1pin,INPUT);
  pinMode(b0pin,INPUT);
  pinMode(b1pin,INPUT);
  pinMode(sum0pin,OUTPUT);
  pinMode(sum1pin,OUTPUT);
  pinMode(carrypin,OUTPUT);
}
voidloop(){
  a0=digitalRead(a0pin);
  a1=digitalRead(a1pin);
  b0=digitalRead(b0pin);
  b1=digitalRead(b1pin);
  bool s0=a0^b0;
  bool c0=a0&b0;
  bool s1=( a1^b1) ^c0;
  bool c1=(a1&b1)|(c0&( a1^b1));
  digitalWrite(sum0pin,s0);
  digitalWrite(sum1pin,s1);
  digitalWrite(carrypin,c1);
}
```

IV. PRIMENA ČETVOROBITNOG SABIRAČA

A. Prednosti četvorobitnog sabirača napravljenog pomoću logičkih kola

Predmet informatika i računarstvo se sve više uvodi u osnovne škole. Četvorobitni sabirač napravljen pomoću logičkih kola omogućava učenicima da provere svoja sabiranja u binarnom sistemu. Učenici mogu da vide na kom principu rade današnji kalkulatori. Mogu da prouče logiku kojom mašine sabiraju. Primena ovog prototipa je da pomogne učenicima da bolje shvate binarni brojni sistem i sabiranje u njemu.

B. Prednosti dvobitnog sabirača napravljenog pomoću Arduino razvojnog okruženja

Prednost sabirača napravljenog pomoću Arduino razvojnog okruženja je mogućnost ponovnog korišćenja Arduina za naredni prototip. Lakše se menja, unapređuje i prenosi.

C. Moguća unapređenja sabirača

Sabirač se može unaprediti dodavanjem još prekidača, odnosno povećanjem broja cifara sabiraka. Može se unaprediti i dodavanjem konvertora, tako da može da

prikazuje ulaz i izlaz i u dekadnom sistemu. Na principu Bulove algebre može se napraviti kolo koje može i da oduzima.

ZAHVALNICA

Ovaj rad realizovan je u okviru projekta "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu. Autori se zahvaljuju realizatorima projekta na pruženom znanju i pomoći tokom izrade rada.

LITERATURA

- [1] Albert Paul Malvino, „Digital computer electronics“.
- [2] Tehničke specifikacije logičkih kola 74HC86, 74HC08 i 74HC32.
- [3] Z. Prijić, Slajdovi sa predavanja iz predmeta „Digitalna Mikroelektronika“ na Elektronskom fakultetu, Katedra za mikroelektroniku, Srbija, 2007. Dostupno online: http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Digitalna_combinational_logic.pdf
- [4] Web aplikacija Tinkercad, Dostupno online: <https://www.tinkercad.com/learn/circuits>
- [5] „Arduino programsko okruženje“, <https://www.elektronika.ftn.uns.ac.rs/elektronika-e2/wp-content/uploads/sites/87/2018/03/Arduino-uputstvo.pdf>

Realizacija semafora sa dnevnim i noćnim režimom rada korišćenjem Arduina

Veljko Spasić, Danilo Jovanović

Sadržaj – U ovom radu prikazan je plan smanjenja zastoja u noćnom saobraćaju pomoću promene načina rada semafora u noćnim uslovima. Predstavljen je prototip semafora koji prati osvetljenost i prelazi u režim treptućeg žutog svetla kada padne mrak. Za realizaciju je korišćen Arduino.

I. UVOD

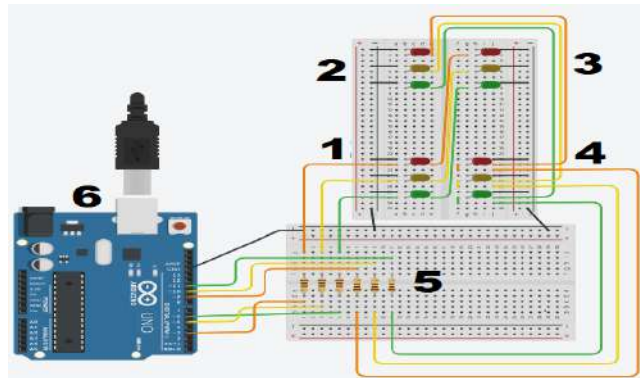
Još od 1868. godine semafori su bili jedni od glavnih elemenata u saobraćaju, služe za regulisanje saobraćaja i prikazivanje mogućnosti prolaza na raskrsnici. Električni semafor kakvog danas znamo bio je postavljen 1912. godine a prvi put je u 1920. godini jedna cela raskrsnica bila kontrolisana pomoću semafora. Semafori se postavljaju na prilazu raskrsnici, na vertikalnom stubu. Svetla su postavljena vertikalno, jedno ispod drugog. Najviše je postavljeno crveno, a ispod njega slede žuto i zeleno svetlo. Prvi semafor u Srbiji bio je postavljen 1939. godine i regulisao je saobraćaj između ulice Kneza Miloša i Takovske u Beogradu, još tad je postojala ideja našeg današnjeg projekta a to je uključivanje treptuće žutog svetla u večernjim satima.

Žuto treptuće svetlo upozorava vozače da semafor ne radi i da smanje brzinu i povećaju opreznost. U ovom slučaju potrebno je primeniti druga pravila za određivanje prvenstva prolaza: saobraćajna pravila, saobraćajne znakove ili saobraćajac koji reguliše saobraćaj. U ovom radu biće opisana realizacija semafora koji u noćnim satima menja program rada u treptuće žuto.

II. IZRADA SEMAFORA

Na slici 1 data je električna šema prototipa semafora koji se kontroliše pomoću Arduina. Brojevima 1-4 obeležene su LED diode, za 4 semafora na raskrsnici. Na proto ploči smešteno je 12 dioda, po 3 za svaki semafor. Povezani su paralelno, 1 i 3 sa 2 i 4 zato što 1 i 3 odnosno 2 i 4 predstavljaju naspramne semafore na raskrsnici. Na protoploči označenoj brojem 5 povezani su otpornici koji služe da ograniče struju kroz LED diode. Korišćeni su otpornici otpornosti 100 Ω . Arduino razvojno okruženje obeleženo je brojem 6. Arduino okruženje sadrži mikrokontroler ATmega 328P koji upravlja radom semafora.

V. Spasić i D. Jovanović su učenici Elektrotehničke škole Rade Metalac u Leskovcu, smer Elektrotehničar informacionih tehnologija. E-mail: vex2005vex@gmail.com, dakijo.bojnik@gmail.com.



Slika 1. Šema semafora koji se upravlja preko Arduina.

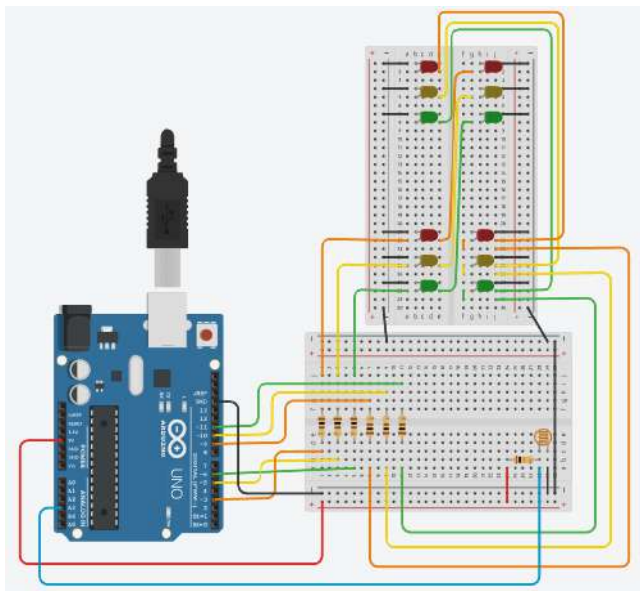
Modifikacija hardvera sa ciljem da se realizuje semafor koji ima noćni režim rada podrazumeva dodavanje senzora osvetljenosti koji će pratiti da li je noć ili dan. U ovom projektu iskorišćen je fotootpornik. Fotootpornik formira razdelnik napona sa otpornikom stalne otpornosti 10 k Ω . Napon razdelnika napona vodi se na analogni ulaz Arduina. Preko ovog ulaza softverski će biti očitano stanje osvetljenosti. Za projektovanje šeme korišćena je web aplikacija Tinkercad.

Fotootpornici pripadaju grupi optoelektronskih komponenata. Otpornost im se menja zbog interakcije svetlosti sa materijalom od koga su napravljeni. Nazivaju se još i svetlosno zavisni otpornici (*Light Dependent Resistors – LDR*). Proizvode se od poluprovodničkih jedinjenja (InSb, ZnS, PbS, CdS, CdSe, itd.). Kada fotoni upadne svetlosti prodru u materijal, svoju energiju predaju elektronima tog materijala. Elektroni raskidaju kovalentne veze sa matičnim atomima i postaju slobodni. Višak slobodnih elektrona rezultuje smanjenjem specifične električne otpornosti materijala. Efekat se naziva fotoprovodnost. Fotoosetljivi materijal se pravi u obliku izuvijane trake kako bi otpornik bio što duži (slika 2).



Slika 2. Fizički izgled fotootpornika.

Šema povezivanja prototipa semafora sa dnevnim i noćnim režimom rada prikazana je na slici 3. Napon za razdelnika napona vezuje se na analogni pin A3.



Slika 3. Šema povezivanja semafora sa dnevnim i noćnim režimom rada.

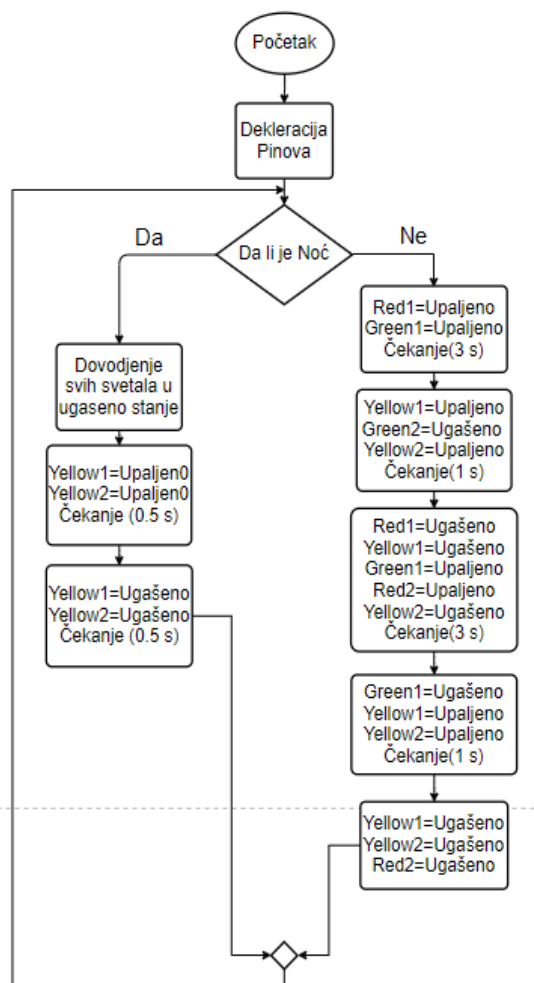
U programu Arduina pre svega su definisani pinovi koji se koriste za kontrolu odgovarajućih LED dioda i ti pinovi deklarirani su kao izlazni. S druge strane, pin A3 na koji je doveden napon s razdelnika napona u kome je fotootpornik deklarisan je kao ulazni. U okviru glavne programske petlje pre svega se vrši očitavanje vrednosti osvetljenosti. U tu svrhu koristi se analogni – digitalni konvertor mikrokontrolera. Za očitavanje vrednosti konverzije koristi se funkcija `analogRead()`. Očitana vrednost smešta se u promenljivu `PhotoResistor`. Sledi ispitivanje uslova da li je noć ili dan u zavisnosti od toga uključuju se odgovarajuće LED diode. Dijagram toka programa prikazan je na slici 4. Za podešavanje stanja LED-a koristi se funkcija `digitalWrite()` kojoj se prosleđuje pin na koji treba da postavi odgovarajuće logičko stanje. Moguća logička stanja su HIGH – za uključeni LED i LOW – za isključeni LED. Za defisanje trajanja pojedinih stanja semafora koristi se funkcija `delay()`, čiji je parametar vreme izraženo u milisekundama. Kompletan kod programa za Arduino koji upravlja dnevnim i noćnim načinom rada semafora dat je u nastavku.

```
//Dekleracija pinova
#define Red1 3
#define Yellow1 5
#define Green1 6
#define Red2 9
#define Yellow2 10
#define Green2 11
#define PRPin A3
int PhotoResistor;
void setup()
{
//Dekleracija tipa pina
pinMode(Red1,OUTPUT);
pinMode(Yellow1,OUTPUT);
```

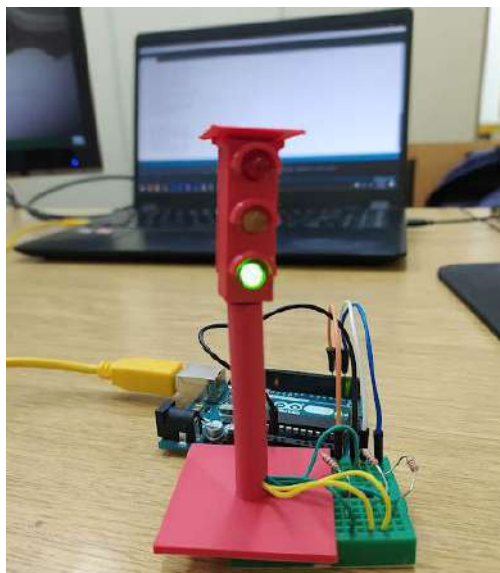
```
pinMode(Green1,OUTPUT);
pinMode(Red2,OUTPUT);
pinMode(Yellow2,OUTPUT);
pinMode(Green2,OUTPUT);
pinMode(PRPin,INPUT);
}

void loop()
{
//Očitavanje vrednosti foto-otpornika
PhotoResistor=analogRead(PRPin);
//Ako je dan onda...
if(PhotoResistor<=100)
{
//Otvaramo 2. ulicu, a 1. zatvaramo
digitalWrite(Red1,HIGH);
digitalWrite(Green2,HIGH);
delay(3000);
//Upozoravamo da se 2. ulica zatvara, a 1.
će se uskoro otvoriti
digitalWrite(Yellow1,HIGH);
digitalWrite(Green2,LOW);
digitalWrite(Yellow2,HIGH);
delay(1000);
//Zatvaramo 2. ulicu, a otvaramo 1.
digitalWrite(Red1,LOW);
digitalWrite(Yellow1,LOW);
digitalWrite(Green1,HIGH);
digitalWrite(Red2,HIGH);
digitalWrite(Yellow2,LOW);
delay(3000);
//Upozoravamo da se 1. ulica zatvara, a 2.
će se uskoro otvoriti
digitalWrite(Green1,LOW);
digitalWrite(Yellow1,HIGH);
digitalWrite(Yellow2,HIGH);
delay(1000);
digitalWrite(Yellow1,LOW);
digitalWrite(Yellow2,LOW);
digitalWrite(Red2,LOW);
//Zatim se sve vraća na početno stanje
}
//Ako je noc onda...
if(PhotoResistor>=100)
{
//Uključuje se trepćuće zuto i signalizira
dopušten prolaz za obe ulice uz povećanu
opreznost
digitalWrite(Yellow1,HIGH);
digitalWrite(Yellow2,HIGH);
delay(500);
digitalWrite(Yellow1,LOW);
digitalWrite(Yellow2,LOW);
delay(500);
}
}
```

Za predstavljeni projekat napravljena je maketa. Delovi za maketu izrađeni su pomoću 3D štampača. LED diode postavljene su u odštampani model semafora, dok su ostale komponente povezane na protoploči. Na slici 5 prikazana je praktična realizacija jednog od semafora na raskrsnici, kada je uključeno zeleno svetlo.



Slika 4. Dijagram toka programa Arduina za semafor sa dnevnim i noćnim režimom rada.



Slika 5. Praktična realizacija 3D modela semafora sa ugrađenim LED diodama.

III. ZAKLJUČAK

Mogućnosti za bolju regulaciju saobraćaja uvek postoje. U ovom radu predstavljen je jedan od načina za realizaciju semafora sa dnevnim i noćnim režimom rada. Ovo može da doprinese, kao prvo smanjenju gužvi u noćnom saobraćaju, kao i povećanju sigurnosti u noćnom saobraćaju. U urbanim sredinama, gde se frekvencija saobraćaja tokom noći ne smanjuje, ovakvo rešenje ne bi bilo najbolje moguće, ali u tome se ogledaju mogućnosti za unapređenje ovog projekta. Jedna od mogućnosti je merenje frekvencije vozila raskrsnici i regulisanje vremena trajanja pojedinih stanja semafora.

ZAHVALNICA

Zahvaljujemo se srednjoj školi TŠ "Rade Metalac" iz Leskovca na povezivanju sa Elektronskim fakultetom u Nišu koji je organizator IEEE STEM projekta. Posebno se zahvaljujemo razrednom strešini Jasminki Milosavljević koja nas je podstakla na ovaj poduhvat, direktoru škole Danku Dankoviću na angažovanju oko našeg projekta i asistentu Milošu Marjanoviću na izdvojenom vremenu za održavanje *online* predavanja o praktičnoj elektronici. Ovaj rad realizovan je u okviru projekta "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu.

LITERATURA

- [1] Semafor. Dostupno na: <https://naslovi.net/2018-12-11/dnevnik/london-prvi-semafor-napunio-150-godina/22654461>
- [2] Semafor. Dostupno na: <https://ltablece.com/svetlosni-signali-semafori/semafori-za-regulisanje-kretanja-vozila>
- [3] Fotoopornici. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/EK_Predavanja_01.pdf
- [4] Arduino – rešeni primeri. Dostupno na: <http://mikro.elfak.ni.ac.rs/wp-content/uploads/Arduino-zbirka-resenih-projekata.pdf>

Projektovanje pametne raskrsnice

Dušan Savić, Stefan Lukić, Andrija Kostić, Dušan Marjanović

Sadržaj – U ovom radu je dat opis projektovanja pametne raskrsnice. Kao centralna jedinica iskorišćeno je *Arduino Uno* okruženje, a svetla na semaforima su predstavljena LE diodama. Kolo za raskrsnicu je realizovano na protoploči.

I. TEORIJSKI UVOD

Pametne raskrsnice su pravac evolucije saobraćaja u 21. veku, i one omogućavaju smanjenje gužvi u saobraćaju. One funkcionišu po principu da je na semaforima za automobile i generalno drumski saobraćaj uvek zeleno svetlo, sve dok pešak ne pritisne taster na semaforu. Pritisak tastera uključuje crveno svetlo i tako zaustavlja kretanje automobila, dok se sa druge strane uključuje zeleno svetlo za pešake.

Povećanje bezbednosti saobraćaja, pogotovo pešaka u saobraćaju je jedan od osnovnih problema na kome se radi u pametnim gradovima. Ovakav tip raskrsnice je moguće implementirati u okolini školskih i predškolskih ustanova kako bi se povećala bezbednost učenika.

II. KORIŠĆENE KOMPONENTE

U ovom radu iskorišćene su sledeće komponente: *Arduino Uno* okruženje kao centar sistema, LE diode (zeleno, žuto i crveno), zatim četvoropinski taster kao i odgovarajući potrebni otpornici.

A. Okruženje Arduino Uno

Iskorišćeno je *Arduino Uno* razvojno okruženje sa ATmega328P mikrokontrolerom [1]. *Arduino* ima 14 digitalnih pinova, označenih sa 0-13, kao i 6 analognih pinova. Korišćeno okruženje prikazano je na slici 1. Digitalni pinovi se mogu koristiti kao ulazni i izlazni.

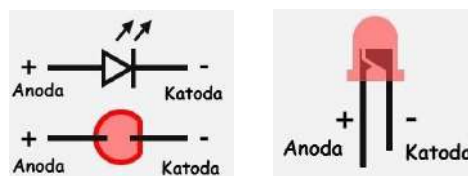


Slika 1. Korišćeno *Arduino Uno* okruženje.

D. Savić, S. Lukić, A. Kostić i D. Marjanović su učenici prvog razreda srednje škole, smer informacione tehnologije, ETŠ „Mija Stanimirović“, Bulevar Svetog cara Konstantina br 82-84, 18000 Niš, Serbia, E-mail: sdusan67@gmail.com, stefi2006luk@gmail.com

B. LE diode

Diode koje emituju svetlost (*Light Emitting Diodes-LED*) pripadaju grupu optoelektronskih komponenta [2]. Na slici 2 prikazan je električni simbol LE diode.

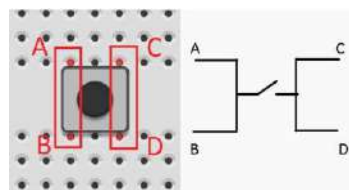


Slika 2. Simbol i kućište korišćenih LED.

Emitovanje svetlosti se dešava prikazom direktne polarizacije diode i ova pojava se naziva elektroluminacija. U ovom radu iskorišćena je žuta, crvena i zelena LE dioda. Ove diode simuliraju svetla na semaforu.

C. Taster

Za aktiviranje promene svetla na semaforu iskorišćen je četvoropinski taster. Na slici 3 je prikazana principijelna šema tastera.

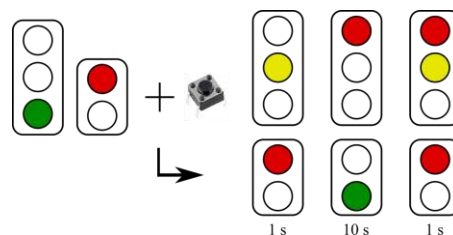


Slika 3. Princip rada tastera.

Kada je taster u pritisnutom stanju, krajevi A i B spajaju se sa onim što je povezano na krajeve C i D.

III. REALIZACIJA PROJEKTA

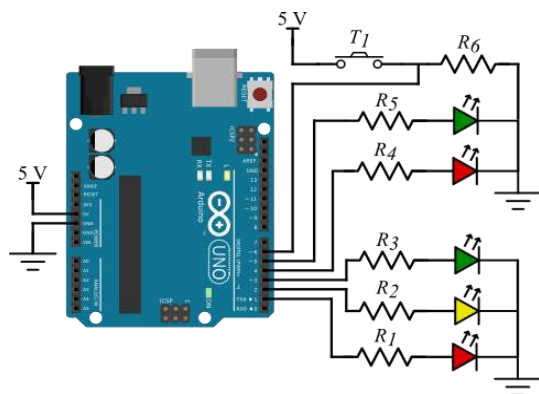
Prvo će biti objašnjen princip rada raskrsnice, a onda će biti prikazan način realizacije. Princip rada raskrsnice prikazan je na slici 4.



Slika 4. Dijagram rada semafora.

Kada taster nije pritisnut, svetlo na semaforu ulice (semafor sa tri svetla) drži zeleno svetlo uključeno. Dok je zeleno svetlo na ulici uključeno, uključeno je i crveno svetlo za pešake. Pritiskom tastera, treba pokrenuti promenu sa zelenog preko žutog na crveno svetlo na semaforu za ulicu. Samim tim, menja se i svetlo na semaforu za pešake, pa pešaci mogu da pređu ulicu. Kada istekne vreme za prolazak pešaka, sistem se vraća u početno stanje.

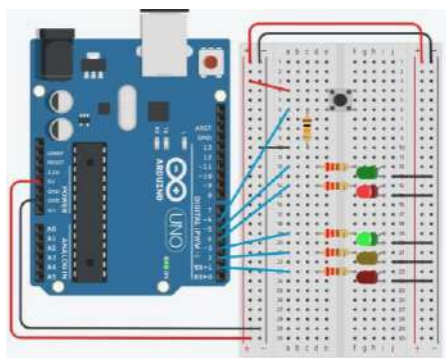
Električna šema za realizaciju ovakvog sistema prikazana je na slici 5.



Slika 5. Električna šema za realizaciju semafora.

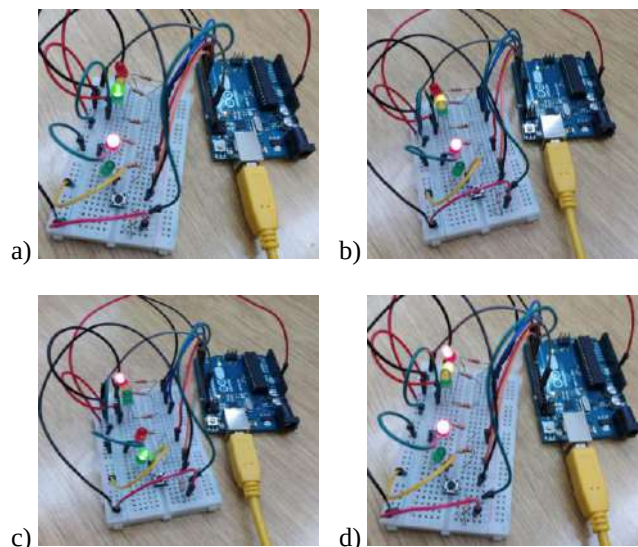
LED koje predstavljaju svetla uličnog semafora povezane su na pinove 1-3. LED koje predstavljaju svetla semafora za pešake su povezane na pinove 4 i 5. Sve iskorišćene LED su povezane na pinove preko otpornika od 220 Ω . Ovi otpornici služe da bi ograničili struju koja protiče kroz LED. Taster u pull-down konfiguraciji sa otpornikom od 10 k Ω je povezan na pin 6. Potrebno je naravno povezati katode svake od LED na masu (GND), gde je povezana i masa Arduino okruženja.

Pre praktične realizacije, potrebno je napisati kod za Arduino okruženje, a onda i odraditi simulaciju projekta. Za simulaciju iskorišćen je program *TinkerCad* [3]. Uz pomoć ovog programa moguće je simulirati rad Arduino okruženja. Šema projekta realizovanog korišćenjem ovog programa prikazana je na slici 6.



Slika 6. Šema korišćenjem TinkerCad, aktivna simulacija.

Kada je ispravan rad proveren simulacijom, kolo je praktično povezano na protoploči. Izgled praktično realizovanog sistema je prikazan na slici 7.



Slika 7. Praktična realizacija sistema a) taster nije pritisnut, uključeno je zeleno svetlo za ulicu i crveno za pešake; b) taster je pritisnut, uključuje se žuto svetlo na ulici; c) uključeno je crveno svetlo za ulicu, a zeleno svetlo za pešake; d) ponovo je uključeno crveno svetlo za pešake i uskoro se uključuje zeleno za ulicu.

Praktičnom realizacijom potvrđena je ispravnost kola i napisanog koda. Kao što je moguće videti sa slika, pritiskom tastera aktivira se potreban niz LED.

IV. ZAKLJUČAK

U ovom radu opisano je projektovanje pametne raskrsnice koja bi povećala bezbednost pešaka u saobraćaju. Sistem je moguće unaprediti dodavanjem nekih vremenskih ograničenja između dva pritiska tastera.

ZAHVALNICA

Ovaj rad realizovan je u okviru projekta "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu. Autori se zahvaljuju profesorki Radmili Lučić kao i osoblju Katedre za mikroelektroniku sa Elektronskog fakulteta u Nišu za niz održanih radionica u sklopu projekta.

LITERATURA

- [1] Tehnička dokumentacija za Arduino Uno. Dostupno na: <https://www.farnell.com/datasheets/1682209.pdf>
- [2] Tehnička dokumentacija za LED. Dostupno na: <https://www.vishay.com/docs/83171/tlur640.pdf>
- [3] TinkerCad. Dostupno na: <https://www.tinkercad.com>

Pokretanje motora ventilatora na određenoj temperaturi uz pomoć Arduino okruženja i DHT11 senzora

Luka Nikolić, Petar Zdravković

Sadržaj - U ovom radu je prikazan primer projektovanja uređaja sa ventilatorom koji se uključuje kada temperatura prostorije poraste iznad zadate vrednosti. U radu je predstavljena šema za ovaj uređaj. Projekat je prvo proveren u simulaciji, a onda je sistem i praktično realizovan.

električne struje na pinovima je oko 20 mA. Na okruženju je povezan i oscilator od 20 MHz. Digitalni pinovi se mogu koristiti i kao ulazni i kao izlazni pinovi. Korišćeno okruženje prikazano je na slici 1.

I. TEORIJSKI UVOD

U doba naglog razvoja tehnologije elektronski uređaji koji nas okružuju postaju sve razvijeniji i napredniji. Velika većina uređaja ima mogućnost da u zavisnosti od nekog spoljnog parametra promeni neko stanje, uključi neko svetlo ili slično. Ovakvi uređaji nazivaju se pametni uređaji.

Prvi pametni uređaji su bili jako jednostavni i podrazumevali su uređaje koji samostalno uključuju svetlo kada padne mrak. Vremenom, inženjeri su projektovali uređaje koji mogu da dobijaju informacije sa sve većeg i većeg broja senzora (senzor osvetljenja, temperature, magnetnog polja itd.). Ovakvi uređaji su sada prisutni i u industrijskim mernim uređajima, ali i u svakodnevnom životu. Jedan primer takvog uređaja, ventilatora koji se uključuje kada temperatur padne ispod određene zadate vrednosti je prikazan u ovom radu.

II. KORIŠĆENE KOMPONENTE

Za realizaciju uređaja prikazanog u ovom radu, kao glavna jedinica sistema iskorišćeno je okruženje *Arduino Uno*. Ventilator je predstavljen DC motorom. Kao temperaturni senzor iskorišćen je senzor DHT11. Uređaj ima i interfejs koji se prikazuje korišćenjem LCD1602 displeja.

A. Arduino Uno

Arduino Uno [1] je okruženje sa ATmega328P mikrokontrolerom koje sadrži 14 digitalnih pinova (od kojih 6 mogu da se koriste kao PWM (*Pulse Width Modulation*) izlazi), 6 analognih ulaza, USB (*Universal Serial Bus*) ulaza, utičnice za napajanje i restart dugmeta. Ima 32 KB *flash* memorije, 2 KB SRAM (*Static Random Access Memory*) memorije i 1 KB EEPROM memorije. Radni napon ovog mikrokontrolera je 5 V. Jačina

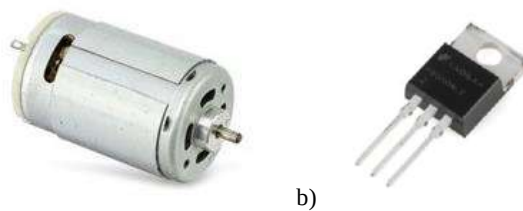
L. Nikolić i P. Zdravković su učenici drugog razreda prirodno-matematičkog smera gimnazije „Bora Stanković“, Vožđa Karađorđa 27, 18000 Niš. E-mail: petarzdravkovic73@gmail.com



Slika 1. Korišćeno okruženje *Arduino Uno*.

B. DC Motor i kolo za kontrolu motora

DC Motor (*Direct current motor*) [2] je komponenta koja koristi energiju jednosmerne struje i pretvara je u mehaničku. Sastoji se iz rotora u samom motoru i statora koji se nalazi na spoljašnjoj strani motora. Kada jednosmerna struja teče kroz motor magnetno polje se stvara oko statora što privlači i odbija magnetne na rotoru stvarajući moment. Da rotor ne bi prestao da se okreće motor u sebi sadrži komutator.



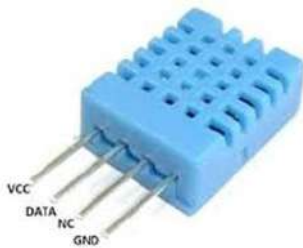
Slika 2. a) Korišćeni DC motor; b) Korišćeni IRF510 MOS transistor za kontrolu motora.

Budući da *Arduino Uno* okruženje ne može da isporuči dovoljno jaku jednosmernu struju za pokretanje motora mora se koristiti kolo koje se sastoji iz MOS tranzistora i diode. MOS transistor [3] ima ulogu da pojača intenzitet električne struje kako bi se rotor motora okretao dovoljno brzo, a dioda koja se vezuje paralelno na motor

ima ulogu da spreči povratak struje u neki drugi deo kola što bi moglo da ošteti mikrokontroler.

C. DHT11 temperaturni senzor

DHT11 [4] je temperaturni i senzor vlažnosti sa digitalnim signalnim izlazom. Senzor obuhvata komponentu za merenje vlažnosti otpornog tipa i komponentu za merenje temperature NTC (*negative temperature coefficient*, tj. otpornik sa negativnim temperaturnim koeficijentom) tipa. Ovaj senzor je prikazan na slici 3.



Slika 3. Korišćeni senzor DHT11.

Jednostavan interfejs čini ga pogodnim za povezivanje na različite mikrokontrolere, pa i na Arduino razvojno okruženje. Merni opseg za merenje vlažnosti vazduha je od 20 – 90 %, sa tačnošću od ± 5 %, a merni opseg za merenje temperature je 0 – 50 °C, sa tačnošću ± 2 °C. Raspored pinova ovog senzora dat je u tabeli I.

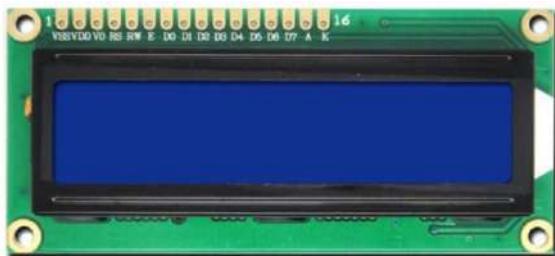
TABELA I: RASPORED PINOVA DHT-A

Pin 1	Napon napajanja
Pin 2	Komunikacioni serijski interfejs
Pin 3	NC (no connection)
Pin 4	Masa

D. LCD displej

LCD displej [5] predstavlja sredstvo za prikazivanje rezultata i tekstualnih karaktera projektovanog uređaja.

Za uređaj je iskorišćen displej koji može da ispisuje karaktere u dve vrste sa 16 karaktera po vrsti poznat kao LCD1602. Ovaj displej prikazan je na slici 4. Svaki karakter koji se prikazuje na displeju je kreiran u polju od 5x8=40 piksela i ima 2x16x40=1280 piksela ukupno.



Slika 4. Iskorišćeni 1602LCD.

Radni napon napajanja je između 4.7 V i 5.3 V, sa potrošnjom struje od 1 mA bez pozadinskog osvetljenja, do 120 mA sa potpunim pozadinskim osvetljenjem. LCD modul može da radi u 8-bitnom i 4-bitnom modu rada. U predstavljenom projektu displej radi u 4-bitnom modu rada što znači da koristi samo 4 pina za podatke u poređenju sa 8 pinova u 8-bitnom modu rada čime se smanjuje broj potrebnih pinova sa Arduino okruženja.

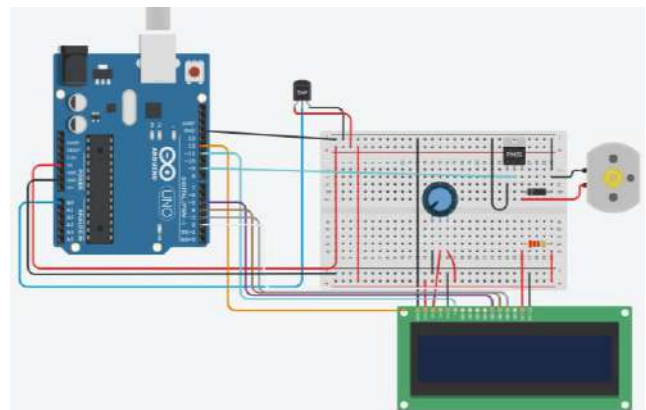
III. REALIZACIJA SISTEMA

Za realizaciju sistema, prvo je potrebno odraditi simulaciju, kako bi se otklonile eventualne greške. Za simulaciju rada uređaja iskorišćen je alat *Tinkercad* [6].

A. Tinkercad

Tinkercad je logički simulator kreiran od strane kompanije Autodesk (za potrebe realizacije projekata baziranih na Arduino platformi). Predstavlja besplatnu kolekciju online softverskih alata dostupnu svima. Jedan od alata je *Tinkercad Circuits* pomoću koga je moguće virtuelno kreirati i programirati projekte na Arduino okruženju bez potrebe za fizički dostupnim hardverom.

Šema projekta nacrtana korišćenjem alata *Tinkercad* prikazana je na slici 5.



Slika 5. Šema nacrtana korišćenjem alata *Tinkercad*.

B. Princip rada

Na električnoj šemi se kao prvo nalazi senzor DHT11 čija je prva nožica povezana na napon od 5 V, druga na analogni pin Arduino okruženja A2, treća nije povezana (ne koristi se), a četvrta na masu (GND). On ima ulogu da očitava temperaturu u prostoriji i njegova očitavanja se prikazuju na LCD displeju. Prvi pin displeja (GND) je povezan na masu, drugi pin (VCC) na napon od 5 V, treći (*contrast*) je povezan na srednju nožicu potencijometra (ostale nožice potencijometra su povezane na napon i masu), RS (*register select*) pin je povezan na digitalni pin 12 Arduino okruženja, RW (*Read/Write*) na masu, *Enable* na digitalni pin 11 Arduino okruženja, pinovi DB4, DB5,

DB6 i DB7 povezani su na digitalne pinove Arduino okruženja 5, 4, 3, 2 redom, LED anoda pozadinskog osvetljenja je preko otpornika od 220 Ω povezana na 5 V, a LED katoda je povezana na masu. Poslednji deo šeme jeste kolo motora. Sastoji se iz MOS tranzistora čija je gejt (*gate*) nožica povezana na masu, drejn (*drain*) nožica na digitalni pin 9 Arduino okruženja i sors (*source*) nožica na jednu od nožica DC motora, druga nožica motora je povezana na masu a paralelno na motor vezana je dioda. Ovo kolo omogućava provod dovoljno jake jednosmerne struje za pokretanje motora kada je temperatura prostorije veća od određene granice koju korisnik postavlja (u ovom primeru je to 25°C).

Nakon kreiranja šeme projekta i povezivanja komponenti moguće je preći na programiranje Arduino okruženja koje se u *Tinkercadu* svodi na kreiranje i postavljanje blokova u skladu sa algoritmom.

C. Pisanje koda

Iako je moguće koristiti *Tinkercad* za pisanje koda za Arduino okruženje, za realizaciju ovog uređaja iskorišćen je softver *Arduino IDE*. Ovaj softver je iskorišćen pre svega jer omogućava dodavanje biblioteke za senzor DHT11. Delovi koda sa objašnjenjem linija prikazani su na slici 6.

```
// Uključivanje biblioteka LiquidCrystal.h i DHT.h za rad sa LCD-om i DHT senzorom
#include <LiquidCrystal.h>
#include "DHT.h"
#define DHTPIN A2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
// deklarisanje konstanti (brojevi Arduino pinova na koje su povezani pinovi LCD-a)
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define motorPin 9

// funkcija za podešavanje, kod između zagrada {} se izvršava samo jednom
// kada se uključi napajanje ili pritisne reset
void setup() {
  Serial.begin(9600); // inicijalizacija serijske UART komunikacije
  // i podešavanje brzine na 9600 bita u sekundi
  Serial.println(F("DHTxx test!"));

  dht.begin();
  lcd.begin(16, 2); // postavljanje broja kolona i redova LCD-a
  pinMode(motorPin, OUTPUT); // konfiguirisanje pina motorPin kao izlaznog
}

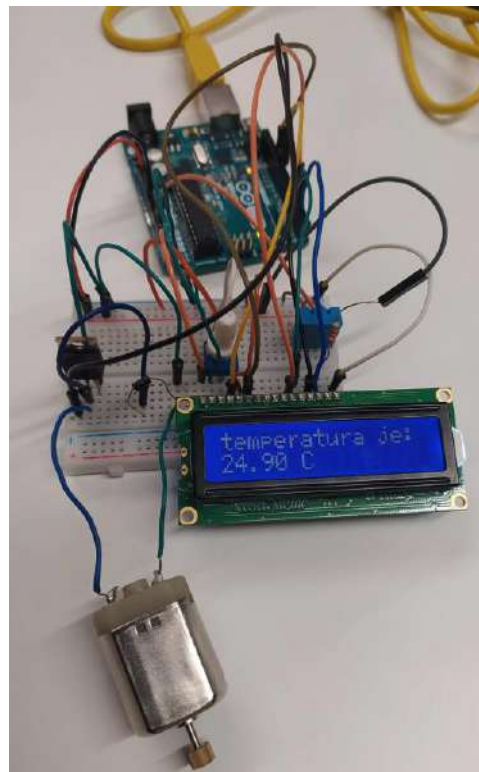
void loop() {
  float temp = dht.readTemperature(); // izračunavanje vrednosti
  // temperature očitava na analognom pina
  int motorStanje = digitalRead(motorPin); // čitanje pina motora

  // ako je temperatura veća od 25 stepeni, uključiti motor
  if(temp > 25){
    digitalWrite(motorPin, HIGH); // postavljanje pina motorPin na visok nivo
    delay(5000); // ovaj deo programa se koristi radi provere
    digitalWrite(motorPin, LOW); // da li sistem funkcioniše
    delay(1000);
  }
  // ako je motor uključen i ako je temperatura manja od 20 stepeni, isključiti motor
  if(motorStanje == HIGH){
    if(temp < 20){
      digitalWrite(motorPin, LOW);
    }
  }
  Serial.println(temp); // štampanje vrednosti temperature
  lcd.setCursor(0, 0); // na virtuelnom terminalu
  lcd.print("temperatura je: "); // štampanje vrednosti
  lcd.setCursor(0, 1); // temperature na LCD displeju
  lcd.print(temp); // uz prethodni tekst
  lcd.setCursor(6, 1);
  lcd.print("°C");
  delay(500);
}
```

Slika 6. Kod Arduino okruženja za pokretanje sistema.

D. Praktična realizacija sistema

Nakon provere rada u simulaciji, sistem je i praktično realizovan sa realnim komponentama. Slika realizovanog sistema prikazana je na slici 7.



Slika 7. Izgled povezanog uređaja u radu.

Prilikom praktične realizacije naišlo se na nekoliko problema. Nakon prvobitnog povezivanja, nije bilo ispisivanja karaktera na displeju. Kako bi bila izvršena provera da li sistem uopšte radi iskorišćeno je slanje podataka preko serijskog porta, da bi se ustanovilo koji deo koda eventualno ne funkcioniše. Kada je potvrđen ispravan rad sistema preko serijskog terminala, naredni korak je bio proveriti povezivanje komponenta na protoploči. Prilikom provere, ustanovljeno je da dve žice koje povezuje Arduino okruženje i LCD1602 nisu bile povezane na odgovarajući način (bila su im zamenjena mesta). Nakon ispravnog povezivanja, dobijeno je ispisivanje karaktera na displeju.

Ispisivanje karaktera na displeju ipak nije bilo u skladu sa očekivanjima. Ne zbog samih karaktera koji se ispisuju, već zbog načina na koji se ispisuju. Svi karakteri su bili ispisani blede i bili su teško čitljivi. Kada se smanji osvetljenost prostorije, sve je teže i teže pročitati vrednosti koje se ispisuju. Prikaz ovog slučaja prikazan je na slici 8. Ustanovljeno je da je potrebno podesiti i kontrast LCD1602 kako bi se ostvarilo ispisivanje karaktera kakvo je očekivano. Kontrast se podešava okretanjem potencijometra čiji je srednji pin povezan na pin 3 displeja.



Slika 8. Problem prilikom praktične realizacije, loš kontrast.

IV. ZAKLJUČAK

U ovom radu je prikazan proces projektovanja ventilatora koji se uključuje kada temperatura poraste iznad predviđene. Svaka od korišćenih komponenti je detaljnije objašnjena. Prikazan je ceo izvorni kod za *Arduino* za ovu primenu. Sistem je i praktično realizovan. Dalje unapređenje sistema ogleda se u tome da se korisniku omogući da sam podešava temperaturu nakon koje će se uključiti ventilator.

ZAHVALNICA

Ovaj rad realizovan je u okviru projekta "STEM visits IEEEESTEC conference" (2021 IEEE Pre-University STEM Grant) i "Vijuge pokreni za IEEEESTEC se spremi" (Centar za promociju nauke) na Elektronskom fakultetu u Nišu. Autori se zahvaljuju profesorki Emiliji Živanović kao i osoblju Katedre za mikroelektroniku sa Elektronskog fakulteta u Nišu za niz održanih radionica u sklopu projekta.

LITERATURA

- [1] Tehnička dokumentacija za Arduino Uno. Dostupno na: <https://www.farnell.com/datasheets/1682209.pdf>
- [2] Tehnička dokumentacija za DC motor. Dostupno na: https://www.arduino.cc/documents/datasheets/DCmotor6_9V.pdf
- [3] Tehnička dokumentacija za MOS tranzistor IRF510. Dostupno na: <https://www.vishay.com/docs/91015/irf510.pdf>
- [4] Tehnička dokumentacija za senzor DHT11. Dostupno na: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [5] Tehnička dokumentacija za LCD1602. Dostupno na: https://www.waveshare.com/datasheet/LCD_en_PDF/LCD1602.pdf
- [6] Softverski alat Tinkercad. <https://www.tinkercad.com>
- [7] Softverski alat Arduino IDE. <https://www.arduino.cc/en/software>

IEEEESTEC na-S-paja

Danijel Danković, Miloš Marjanović, Jana Vračar, Nikola Mitrović, Sandra Veljković, Emilija Živanović

Sadržaj - U ovom radu prikazan je razvoj IEEEESTEC konferencije studentskih projekata od nastanka do danas. Opisana je saradnja Elektronskog fakulteta Univerziteta u Nišu i ostalih eminentnih naučnoistraživačkih i obrazovnih ustanova i institucija tokom organizacije IEEEESTEC konferencije – IEEEESTEC nas spaja.

I. UVOD

IEEEESTEC konferenciju studentskih projekata organizuju IEEE studentski ogranak Niš [1], EESTEC lokalni komitet Niš [2] i Elektronski fakultet, Univerziteta u Nišu [3]. Od samog osnivanja Konferencija se održava pod pokroviteljstvom društva IEEE Electron Devices (EDS)/ Solid-State Circuits (SSC), društva IEEE Microwave Theory and Techniques (MTT) i sekcije IEEE Srbija i Crna Gora [4]. Konferencija je nastala kao rezultat saradnje IEEE studentskog ogranka Niš i EESTEC lokalnog komiteta Niš, što je jasno iz imena Konferencije (IEEE + EESTEC = IEEEESTEC).

Tokom godina IEEEESTEC konferenciju su počele da podržavaju i druge naučnoistraživačke i obrazovne ustanove i institucije iz Srbije i inostranstva. Potencijale IEEEESTEC konferencije među prvima je prepoznao Akademik Ninoslav Stojadinović (1950-2020), osnivač i prvi predsednik Ogranka SANU u Nišu [5]. U znak sećanja na preminulog Akademika Ninoslava Stojadinovića najznačajnija nagrada IEEEESTEC konferencije poneće njegovo ime.

II. CPN I IEEEESTEC

Elektronski fakultet u Nišu i Centar za promociju nauke (CPN, slika 1) [6] su kroz niz zajedničkih aktivnosti ostvarili veoma čvrstu saradnju. Zajednički učesnici svih ovih aktivnosti su učenici više desetina srednjih škola i studenti nekoliko različitih fakulteta iz Srbije i inostranstva, a najvažnije zajedničko mesto okupljanja je IEEEESTEC konferencija. Centar za promociju nauke je finansirao i trenutno finansira nekoliko projekata koji su od velike važnosti za IEEEESTEC konferenciju:

- "Pretočimo znanje u praktičan rad" [7],
- "Praktičnim radom od znanja",
- "Idemo i mi na IEEEESTEC konferenciju",

D. Danković, M. Marjanović, J. Vračar, N. Mitrović, S. Veljković i E. Živanović su sa Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: danijel.dankovic@elfak.ni.ac.rs

- "Za IEEEESTEC se spremi" [8] i
- "Vijuge pokreni za IEEEESTEC se spremi".

Na regionalnom nivou predstavnici Elektronskog fakulteta u Nišu su kroz niz projekata uspostavili zajedničku saradnju sa predstavnicima Regionalnog centra za profesionalni razvoj zaposlenih u obrazovanju iz Niša i predstavnicima Centra za stručno usavršavanje u obrazovanju iz Leskovca, kao i predstavnicima njihovih Naučnih klubova [9-11]. Ovi Centri i Naučni klubovi primer su prave decentralizacije, zato što se u njima organizuje veliki broj izložbi, predavanja, panel-diskusija, radionica, zimskih i letnjih škola i drugih događaja na različite teme, a sve je to lako pristupačno i učesnicima van Beograda. Naučni klubovi prepoznali su značaj saradnje sa Elektronskim fakultetom u Nišu, pa su na glavnoj svečanosti organizovanoj povodom pet godina postojanja i rada Naučnog kluba Leskovac, održanoj u ponedeljak 5. jula 2021. godine, dodelili značajna priznanja pojedincima sa Elektronskog fakulteta (Danijelu Dankoviću i Milošu Marjanoviću), kao i samom Elektronskom fakultetu Univerziteta u Nišu (slika 2).

ОВАЈ ПРОГРАМ ФИНАНСИРА



Slika 1. Centar za promociju nauke finansira ovaj projekat.



Slika 2. Zahvalnica Elektronskom fakultetu u Nišu.

Na Elektronskom fakultetu u Nišu godinama unazad održavaju se radionice u sklopu projekata finansiranih od strane CPN-a, IEEE-a, i drugih relevantnih naučnoistraživačkih i obrazovnih ustanova i institucija. Pored opreme koja je obezbeđena ovim projektima za realizaciju radionica Elektronski fakultet stavio je na raspolaganje i svoju opremu koja se koristi u nastavi i nauci, kao i namenski prostor. Naime, radionice se održavaju u okviru Studentskog kreativnog centra koji funkcioniše u okviru Startup centra Niš (slika 3) [12].



a)



b)

Slika 3. Fotografije sa radionica: a) održane 21. novembra sa učenicima Gimnazije Bora Stanković Niš [13]; b) održane 4. oktobra 2021. godine sa učenicima ETŠ Mija Stanimirović Niš [14].

III. IEEEESTEC ŠIRI KRUG PRIJATELJA

Značaj IEEEESTEC konferencije prepoznala su i druga društva u okviru IEEE-a, WIE Affinity Group i Computational Intelligence (CIS) [4]. Zavod za unapređivanje obrazovanja i vaspitanja (ZUOV) [15] je takođe prepoznao značaj IEEEESTEC konferencije, i 2019. godine je odobrio stručni skup “12. međunarodna

IEEEESTEC konferencija”, kod S389/2019, kao oblik stručnog usavršavanja – konferencija. 43 nastavnika i profesora prisustvovalo je Konferenciji i oni su dobili uverenja o učešću. Tokom Konferencije oni su prisustvovali prezentacijama radova, radionicama i predavanjima o značaju praktične nastave, a na kraju su učestvovali u finalnoj diskusiji i izvođenju zaključaka.



Slika 4. Stručno usavršavanje nastavnika tokom IEEEESTEC konferencije.

Prava potvrda značaja IEEEESTEC konferencija došla je od krovne ustanove, Ministarstva prosvete nauke i tehnološkog razvoja Republike Srbije, pod čijim se pokroviteljstvom Konferencija sada i održava (slika 5) [16].



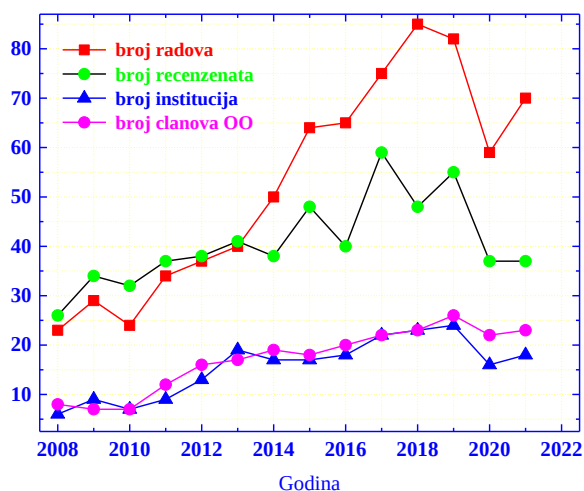
Slika 5. Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije [16].

Imajući u vidu činjenicu da je IEEEESTEC konferencija podržana od strane svih relevantnih činioca u oblasti nauke i obrazovanja, vrlo lako je stekla i naklonost velikog broja kompanija iz Srbije i inostranstva. Kompanije iz godine u godinu obezbeđuju značajne nagrade za autore nagrađenih radova.

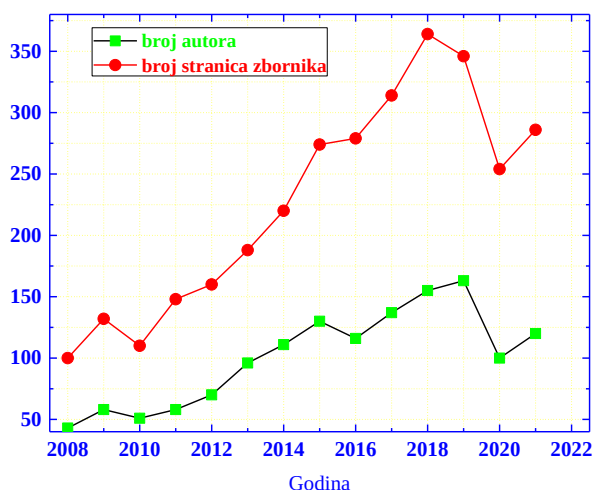


Slika 6. Dodela nagrada tokom IEEEESTEC konferencije.

Napred navedeni akteri u velikoj meri dali su doprinos značajnom napretku IEEESTEC konferencije (slike 7 i 8).



Slika 7. Broj radova, recenzenata, institucija i članova organizacionog odbora IEEESTEC konferencije tokom godina.



Slika 8. Broj autora i stranica zbornika IEEESTEC konferencije tokom godina.

IV. IEEE NA IEEESTEC-U

IEEE (Institute of Electrical and Electronics Engineers) je najveće svetsko udruženje posvećeno unapređivanju tehnoloških inovacija vezanih za elektronske uređaje i tehnološkom razvoju [17]. IEEE ima više od 400.000 članova u 160 zemalja sveta, od toga oko 110.000 studenata. Veliki broj članova učestvuje na preko 1600 IEEE konferencija u 80 zemalja širom sveta.

IEEE je neprofitabilna organizacija koja je poslednjih nekoliko godina finansirala veliki broj projekata koji promovišu rad sa srednjoškolcima. Autori su već imali iskustva i učestvovali u realizaciji EDS-ETC (Engineers Demonstrating Science an Engineer Teacher Connection)

programa pod nazivom “Let’s Put the Knowledge into Practical Work” [18]. U 2021. godini autori u saradnji sa kolegicama i kolegama sa Katedre za mikroelektroniku učestvuju u realizaciji još jednog značajnog projekta pod pokroviteljstvom IEEE-a, “STEM visits IEEESTEC conference”. Projekat je dobijen u okviru TryEngineering IEEE Pre-University STEM Portal Grant for 2021. Ocenjen je kao kreativan i inovativan i plasiran je u samom vrhu [19].

Tokom realizacije projekta u različitim fazama obuhvaćeno je oko 100 srednjoškolaca iz 5 različitih škola (Gimnazija Bora Stanković Niš, ETŠ Mija Stanimirović Niš, ETŠ Nikola Tesla Niš [20], TŠ Rade Metalac Leskovac [21] i Tehnička škola Vlasotince [21]). Tokom realizacije projekta sa učenicima je održano 10 online predavanja, dva takmičarska kviza i 10 radionica na Elektronskom fakultetu u Nišu. Osnovni cilj STEM obrazovanja je da se učenici istovremeno obrazuju kroz četiri naučne discipline, a to su nauka (Science), tehnologija (Technology), inženjerstvo (Engineering) i matematika (Mathematics). Na radionicama je upravo omogućeno učenicima da počev od osnovnih fizičkih zakona opisanih matematičkim formulama, preko crtanja električnih šema dođu do rešenja koje su fizičkom realizacijom svojih projekata i proveravali. Kao rezultat ovakvog rada 30 učenika realizovalo je projekte i napisalo 10 radova za IEEESTEC konferenciju. Ovaj projekat privukao je veliku pažnju studentskih ogranaka IEEE-a, pa su na jednoj od radionica prisustvovali i predstavnici IEEE Student Branch-a iz Slovenije, Eva Boneš, IEEE Student Branch-a iz Belgije, Olivier Leblanc, kao i IEEE Regional Student Representative, Theodoros Chatzinikolau iz Grčke (slike 9 i 10).



Slika 9. Realizatori projekta “STEM visits IEEESTEC conference” sa predstavnicima IEEE Student Branch-a iz Slovenije (Eva Boneš), IEEE Student Branch-a iz Belgije (Olivier Leblanc), kao i IEEE Regional Student Representative (Theodoros Chatzinikolau iz Grčke).

Ovo je bila prilika da promoviramo TryEngineering IEEE Pre-University STEM Portal Grant program (slika 11) [19] kod studentskih ogranaka IEEE-a u tri zemlje iz Evrope.



Slika 10. Neformalno druženje članova IEEE Student Branch Niš i gostiju iz inostranstva.



Slika 11. TryEngineering IEEE Pre-University STEM Portal Grant program [19]

V. ZAKLJUČAK

Organizacioni odbor će raditi i uvoditi novine kako bi konferencija bila interesantnija i učenicima srednjih škola i studentima. Iako je Konferencija popularna, što jasno pokazuju statistički podaci predstavljeni u radu, radiće se i dalje na promociji i popularizaciji naučno-istraživačkog rada i pisanja radova. Ovaj rad je deo aktivnosti na projektima koji na različite načine promovisu i pomažu održavanje IEEEESTEC konferencije. Organizatorima Konferencije mnogo znači pomoć svih relevantnih činioaca u oblasti nauke i obrazovanja.

ZAHVALNICA

Zahvaljujemo svim ustanovama i institucijama koje podržavaju IEEEESTEC konferenciju.

Zahvaljujemo svim članovima Katedre za mikroelektroniku Elektronskog fakulteta Univerziteta u Nišu, koji nam daju ogromnu podršku prilikom organizacije IEEEESTEC 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] 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/>
- [6] Centar za promociju nauke. [OnLine] Dostupno na: <http://www.cpn.rs/o-centru/>
- [7] M. Marjanović, D. Danković, and D. Danković, "Projekat Centra za promociju nauke "Pretočimo znanje u praktičan rad" realizovan u Naučnim klubovima Niš i Leskovac", in *Proc. 12th IEEEESTEC 2019*, Niš, Serbia, 2019, pp. 105-108.
- [8] D. Danković, and M. Marjanović, "Projekat Centra za promociju nauke "Za IEEEESTEC se spremi" na Elektronskom fakultetu u Nišu", in *Proc. 13th IEEEESTEC 2020*, Niš, Serbia, 2020, pp. 199-200.
- [9] Regionalni centar za profesionalni razvoj zaposlenih u obrazovanju u Nišu. Dostupno na: <http://www.rcnis.edu.rs/>
- [10] Centar za stručno usavršavanje u obrazovanju Leskovac [OnLine] Dostupno na: <http://www.csuleskovac.edu.rs/>
- [11] Naučni klubovi. [OnLine] Dostupno na: <http://www.cpn.rs/prograi/mreza-naucnih-klubova/>
- [12] Startup centar Niš, Centar za inovativno preduzetništvo mladih. Dostupno na: <https://www.startupcentarnis.org/#home>
- [13] Gimnazija Bora Stanković Niš. Dostupno na: <https://www.borastankovic.edu.rs/>
- [14] ETŠ Mija Stanimirović Niš. Dostupno na: <https://www.etsmijastanimirovic.edu.rs/>
- [15] Zavod za unapređivanje obrazovanja i vaspitanja. [OnLine] Dostupno na: <https://zuov.gov.rs/>
- [16] Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije. Dostupno na: <https://www.mpn.gov.rs/>
- [17] Institute of Electrical and Electronics Engineers, IEEE. [OnLine] Dostupno na: <https://www.ieee.org/>
- [18] EDS-ETC (Engineers Demonstrating Science an Engineer Teacher Connection) Program. [OnLine] Dostupno na: <https://eds.ieee.org/chapters/the-eds-etc-program>
- [19] IEEE TRYEngineering. [OnLine] Dostupno na: <https://tryengineering.org/news/announcing-the-2021-ieee-pre-university-stem-portal-grant-recipients/>
- [20] ETŠ Nikola Tesla Niš. Dostupno na: <http://nikola.etstesla.ni.ac.rs/>
- [21] TŠ Rade Metalac Leskovac. Dostupno na: <https://www.rademetalac.edu.rs/>
- [22] Tehnička škola Vlasotince. Dostupno na: <https://tsvlasotince.edu.rs/>