IEEESTEC

16th 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 Science, Technological Development and Innovation

ISBN: 978-86-6125-273-0

IEEESTEC Student Projects Conference, November 23, 2023, 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:
Miloš Marjanović

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

Secretary:
Jana Vračar
Aleksandra Stojković
Nikola Mitrović
Milan Stojanović
Sandra Veljković
Neda Stanojević
Anđela Kocić
Stefan Nikolić
Nemanja Nedeljković
Anastasija Avramović
Nikola Stojiljković
Strahinja Dimitrijević
Mikica Radenković
Teodora Popović
Sandra Miljković

Technical Editors:
Miloš Marjanović
Jana Vračar
Danijel Danković
Emilija Živanović

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

621.3(082)
004(082)

IEEESTEC Student Project Conference (16 ; 2023 ; Niš)
IEEESTEC 16th Student Project Conference, [November 23, 2023, Niš] / organized by EESTEC LC, Niš ... [et al.] ; [editors Vera Marković ... [et al.]]. - Niš : Faculty of Electronic Engineering, 2023 (Niš : Grafika Galeb). - 332 str. : ilustr. ; 30 cm

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

ISBN 978-86-6125-273-0

а) Електротехника -- Зборници б) Рачунарство -- Зборници

COBISS.SR-ID 130177801

Printed by: Grafika Galeb, Niš, 2023.
Circulation: 100 copies

Reviewers:

1. Nikola Mitrović
2. Miloš Marjanović
3. Milan Stojanović
4. Aleksandra Stojković
5. Sandra Veljković
6. Neda Stanojević
7. Jana Vračar
8. Goran Nikolić
9. Jelena Jovanović
10. Vladimir Sibinović
11. Ksenija Pešić
12. Đorđe Čikić
13. Marko Dimitrijević
14. Emilija Živanović
15. Milena Frtunić Gligorijević
16. Filip Filipović
17. Vladimir Mitić
18. Milan Simić
19. Nikola Danković
20. Goran Miljković
21. Milica Jovanović
22. Miljana Milić
23. Ana Vučković
24. Nenad Petrović
25. Marija Milijić
26. Anđelija Đorđević
27. Aleksandar Milenković
28. Nikola Stojanović
29. Nikola Vučić
30. Sandra Đošić
31. Novak Radivojević
32. Jelena Nedeljković
33. Milica Stojanović
34. Ivana Kostić
35. Vojkan Davidović
36. Anđela Jovanović
37. Zlatica Marinković
38. Natalija Ivković
39. Isidora Jovanović
40. Anđela Đorđević
41. Mina Nikolić
42. Nevena Tufegdžić
43. Stefan Stankov
44. Danijel Danković
45. Nevena Veselinović
46. Milan Danković
47. Dimitrije Anđelković
48. Sara Savić
49. Lana Tasić
50. Milica Petrušijević
51. Aleksandar Pavlovski
52. Vladimir Rudić
53. Vesna Paunović
54. Nikola Vukotić

Table of contents:

Testiranje sistema za praćenje meteoroloških parametara 1
Prototip sistema mikrotalasne pećnice zasnovan na ARM mikrokontroleru 7
Digitalni uglomer baziran na ADXL345 akcelerometru 11
Projektovanje i emulacija kulera za računar 15
Detektor zvuka upotrebom modula MAX4466 i mikrokontrolera sa ARM arhitekturom 19
Kontrola rada ventilatora korišćenjem ARM mikrokontrolera 23
Projektovanje korisničkog interfejsa aparata za kafu zasnovanog na ARM mikrokontroleru 27
Emulacija mašine za pranje veša bazirana na Blue Pill razvojnom okruženju 31
Prepoznavanje nekih elemenata znakovnog jezika pomoću mašinskog učenja 35
Implementacija VPN tehnologije za unapređenje bezbednosti i privatnosti u Internet of Things (IoT)... 39
Razvoj sigurnosnog sistema zaključavanja uz pomoć Arduino okruženja i 4x4 *Keypad*-a 43
Sistem automatskog zalivanja biljaka 45
Projektovanje *Cut the Rope* mašine korišćenjem razvojnog okruženja sa mikrokontrolerom ARM arhitekture 47
Automatizacija procesa pakovanja čaja primenom SCADA sistema 53
Sušara za sušenje šljiva 57
Rukavica za prevođenje znakovnog jezika 61
Multifunkcionalno pametno vozilo 65
Održavanje ravnoteže četvoronožnog robota 69
Sistem za pomoć pri paralelnom parkiranju 71
Aplikacija za učenje srpskog jezika „*Serbiapp*“ 75
Uređaj za zaštitu električnih potrošača u domaćinstvu 79
Vizuelno-zvučno unapređen sistem bezbednosti 81
Udaljena OET laboratorija – Pločasti kondenzator 85
Igrica za testiranje refleksa realizovana korišćenjem digitalnih kola 89
Makro programibilna tastatura (Macro KeyPad) 95
Pregled konvergencije proširene stvarnosti sa sveprisutnim učenjem radi unapređenja obrazovanja 97
Arduino Visor – robotska glava 107
Agrotemp 111
Algorithms for Drone Positioning 113
Pametni sistem za detekciju pokreta zasnovan na Arduino platformi 117
Pristup dubokog učenja za predviđanje hidroloških vremenskih serija sa modelom ekstremnog mašinskog... 121
Unapređenje učenja studenata kroz implementaciju sistemskog rešenja za zadatke uz korišćenje OCR... 125
Verifikacija RTL modela rutera sa jednim ulaznim i tri izlazna kanala UVM metodologijom 131
Praktična realizacija naizmeničnog pretvarača sa trijakom primenom integrisanog kola TCA785 137
Praktična realizacija estimatora brzine u pogonu motora jednosmerne struje sa jednokvadrantnim čoperom 141
Merenje parametara okoline pomoću *Arduino Mega* u *LabView* okruženju i slanje podataka... 145
High Frequency PWM Generator and MOSFET Bootstrapping Circuit 149
Istraživanje haosa u Čua oscilatoru 153
Pametna invalidska kolica 157
Merenje isparljivih organskih jedinjenja 161
Automatizacija kreiranja modela registarskih blokova pri verifikaciji ASIC čipova UVM metodologijom 165
Programiranje igre Moj broj 169
Generisanje građevina u računarskoj igri „Minecraft” korišćenjem Shap-e modela 173
Uporedna analiza algoritama za generisanje realističnih pahuljica 177
Distribuirani Girvan-Newman algoritam za detekciju zajednica 181
Segmentacija slika primenom dekompozicije tenzora i nenegativne faktorizacije matrica 185
ROBO-Clean: robot za prečišćavanje vazduha 189
Projektovanje pasivne LED matrice korišćenjem *Constant Current* LED drajvera 193
Realizacija uređaja za testiranje brzine refleksnih reakcija 199
Razvoj meteorološkog sistema za prevenciju od bujica 203
Primena kola TCRT5000 za merenje udaljenosti 207
TCAD projektovanje i simulacija bipolarnog tranzistora 211
Projektovanje i realizacija prenosivog ECG uređaja baziranog na STM32 platformi 217
Realizacija 3D skenera korišćenjem *Raspberry Pi* 4 platforme 221
Verifikacija AMBA APB protokola primenom UVM metodologije 227
Korišćenje Arduino razvojnog okruženja u Python-u 231
Primena mašinskog učenja za detekciju i klasifikaciju anemije 235
Bluetooth interfejs za FM radio na bazi Arduina 241
Manuelni ulazni interfejs FM radija na bazi Arduina 245
Sedmosegmentni displej za FM radio prijemnik 249

Interfejs za FM prijemnik sa Arduinom 253
Analiza pojačavača malih signala realizovanog korišćenjem bipolarnog tranzistora... 255
Primena kamere u autoindustriji 259
Izlazni pojačavač za FM radio 263
Realizacija jednostavnog optičkog skenera 267
IoT sistem za nadzor košnica 271
Projektovanje i realizacija rezonantnog LLC pretvarača 275
Kontrola prava pristupa pomoću razvojnog sistema Arduino 281
Kontrola roletni korišćenjem servo motora u zavisnosti od izabranog moda rada 285
Analiza probojnog napona gasnih odvodnika Littelfuse CG2300 pri različitim eksperimentalnim uslovima 289
Kontrola servo motora korišćenjem ARM mikrokontrolera 293
Projektovanje i implementacija aplikacije za vizualizaciju topologije lokalnih računarskih mreža 297
Praktična realizacija signal generatora pomoću analognih komponenata 303
Praktična realizacija signal generatora korišćenjem ICL8038 kola 307
Ionizing Radiation Dosimeter Calibrated as Geiger Counter Based on Arduino Microcontroller... 309
Praktična realizacija sistema za merenje visokih temperatura korišćenjem MAX6675 kola 313
Implementacija Scrum metodologije u okviru razvoja softvera 317
VTŠFit Aplikacija za prevenciju povreda tokom vežbanja korišćenjem veštačke inteligencije 321
Linear Circuit Solver Based on Polymorphism and Automatic Differentiation 325
Strukture integrisanih kola na primeru SE555P preciznog tajmera 329

Testiranje sistema za praćenje meteoroloških parametara

Hristina Todorović

Sadržaj – Rad je baziran na praćenju promena parametara poput temperature, vlažnosti vazduha, vazdušnog pritiska pod različitim meteorološkim uslovima, istovremenim poređenjem rada više senzora slične namene ima za cilj da pre svega prikaže pouzdanost današnjih senzora za meteorologiju, uporedi ih međusobno, kao i da prikaže prednost u brzini, efikasnosti i veličini u odnosu na one korišćene od davnina.

I. Uvod

Ključni pojmovi za dati rad i uglavnom oni koji bivaju permutovani jesu meteorološki elementi pod koje spadaju: temperatura, pritisak, vlaga, vetar koji su u radu i praćeni, dok se pojam meteoroloških pojava odnosi na: kišu, sneg, grad i maglu na primer. Samo vreme se može definisati kao stvarno stanje meteoroloških elemenata i pojava u datom trenutku na određenom mestu [1]. Termometri su instrumenti koji koriste osobine promene dužine i zapremine čvrstih tela, zapremine tečnosti ili pritiska gasa u sudu konstantne zapremine, električne otpornosti, boje (talasne dužine) zagrejanog tela za merenje temperature [2]. Hemijske materije koje upijaju vlagu, na primer kalcijum hlorid i fosfor pentoksid koriste se kao higroskopne materije kojima se meri apsolutna vlažnost vazduha. Sama vlažnost vazduha predstavlja stepen zasićenosti vazduha vodenom parom, a izražava se u procentima (%) [3]. Barometar aneroid se koristi za merenje vazdušnog pritiska pored metalnog i živinog barometra [4]. Vazdušni pritisak opada sa visinom i to približno jedan milibar za svakih 10 metara povećanja visine. Vreme je nešto sa čime moramo da živimo, ali, ako ga bolje upoznamo možemo svesno uticati na smanjenje njegovih negativnih uticaja, a mnogo efikasnije koristiti njegova pozitivna svojstva. Zato je i cilj ovog rada da ispita pouzdane i precizne metode za merenje navedenih meteoroloških elemenata.

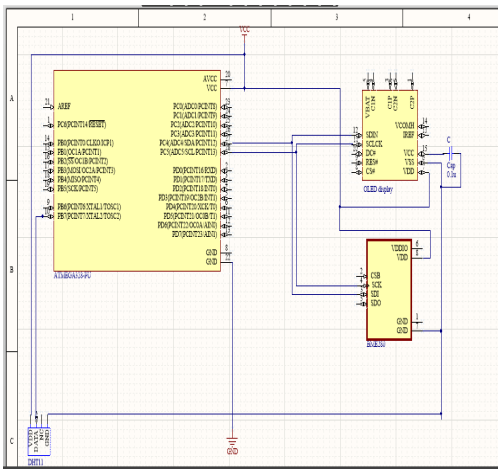
II. Realizacija i testiranje uređaja

A. Šema uređaja u Altium Designer-u

Pre svega, na Slici 1 je dat simbol mikrokontrolerske ploče sa svim datim pinovima od kojih su za realizaciju uređaja za praćenje meteoroloških parametara iskorišćeni:

H. Todorović je diplomirani inženjer Elektrotehnike i računarstva na modulu Elektronske komponente i mikrosistemi Elektronskog fakulteta u Nišu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: hristina.t@elfak.rs

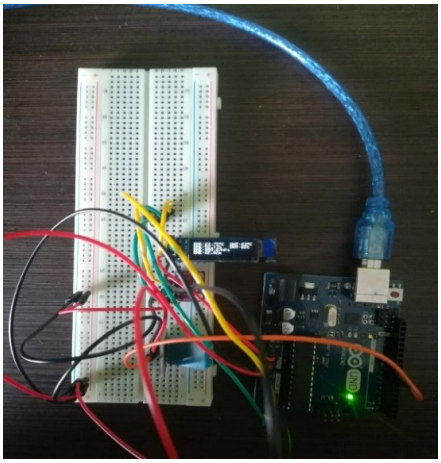
- VCC - napajanje komponenata DHT11 (VDD), BME280 (VDD) i OLED (VDD);
- GND - masa uređaja;
- A4 - linija za podatke kod I²C protokola (SDA);
- A5 - linija za takt (SCL);
- PIN 7 - linija za podatke kod DHT11.



Slika 1. Šematik u Altium Designer-u.

Mikrokontrolerska ploča je na napajanje povezana preko USB kabla, a dalje to svojim pinovima za 5 i 3,3 V prenosi na komponente. DHT11 koristi samo jedan pin za slanje podataka DATA i to na pin 7 mikrokontrolera. BME280 ima pinove za I²C komunikacioni protokol i time komunicira sa mikrokontrolerom, kao što je slučaj i sa OLED displejem. Ova dva uređaja su paralelno povezana na A4 i A5 pin, jer su kod Arduino Uno ploče ova dva pina rezervisana za I²C tip komunikacije. U toku izrade bilo je neophodno proveriti I²C adresu BME280 pomoću koda iz sekcije Examples, radi prepoznavanja same komponente obzirom da ona i može imati dve adrese. Na Slici 2 dat je izgled uređaja koji je predviđen za testiranje, dok je u svakom trenutku promena stanja parametara vidljiva na displeju.Mikrokontrolerska ploča je na napajanje povezana preko USB kabla, a dalje to svojim pinovima za 5 i 3,3 V prenosi na komponente. DHT11 koristi samo jedan pin za slanje podataka DATA i to na pin 7 mikrokontrolera. BME280 ima pinove za I²C komunikacioni protokol i time komunicira sa mikrokontrolerom, kao što je slučaj i sa OLED displejem. Ova dva uređaja su paralelno povezana na A4 i A5 pin, jer su kod Arduino Uno ploče ova dva pina rezervisana za I²C tip komunikacije. U toku izrade bilo je neophodno proveriti I²C adresu BME280 pomoću koda iz

sekcije Examples, radi prepoznavanja same komponente obzirom da ona i može imati dve adrese. Na Slici 2 dat je izgled uređaja koji je predviđen za testiranje, dok je u svakom trenutku promena stanja parametara vidljiva na displeju.



Slika 2. Realizovan uređaj.

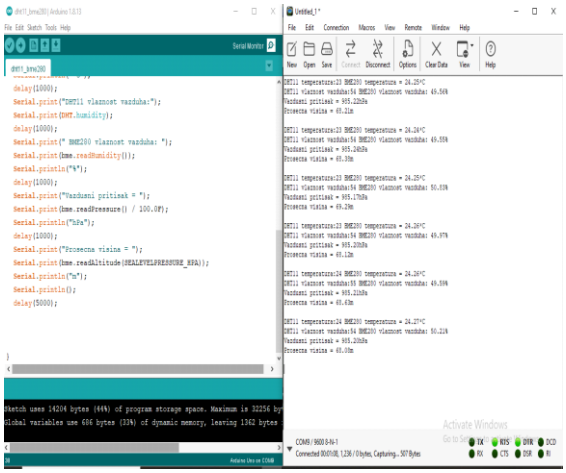
B. Testiranje uređaja

Za potrebe testiranja stabilnosti, preciznosti i osetljivosti komponenata pri merenju temperature i vlažnosti vazduha u kućnim uslovima korišćena je šoljica sa prokljućalom vodom iz koje se kondenzovala vodena para i jedna činija kojom su čitav uređaj kao i šoljica izolovani (Slika 3). Nakon nekoliko minuta, činija je pomerena i uređaj je ostao uključen do stabilizovanja komponenti na sobnu temperaturu i vlažnost vazduha.
Tokom testiranja pored praćenja parametara na displeju, parametri su prikazivani i na *Serial Monitor*-u softverskog alata Arduino IDE.



Slika 3. Testiranje uređaja u laboratorijskim uslovima.

Što se odnosa vazdušnog pritiska i visine tiče, na svakih 10 metara visine, vazdušni pritisak se smanjuje za oko 1 milibar i on je u ovom merenju konstantan, jer se ni relativna, ni apsolutna visina nisu menjale tokom merenja, a normalni vazdušni pritisak za područje Niša je 993,25mb. Upoređivana su merenja temperature i vlažnosti vazduha. Za prikaz prikupljenih podataka koristi se terminal CoolTerm (Slika 4).



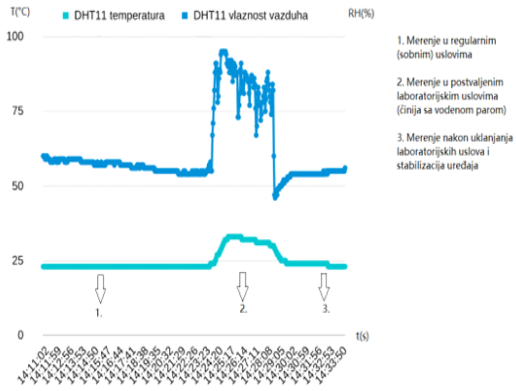
Slika 4. Terminal *CoolTerm*.

Prvo su prikazani parametri koje komponente mere zasebno i tu se jasno vidi da sa porastom temperature u obe varijante raste i vlažnost vazduha (Slika 5 i Slika 6), jer on sam biva sve više zasićen vodenom parom, zatim su tokom vremena upoređeni temperatura i vlažnost vazduha kod ovih dveju komponenti (Slika 7 i Slika 8).

Na graficima se vidi da je BME280 osetljiviji na promene, što nam pokazuje da bi ova komponenta bila pogodnija za neke zatvorene uređaje kod kojih promene nisu toliko česte, niti se odigravaju naglo, dok je DHT11 precizniji, ali sporije reaguje na promene, samim tim je u redu postaviti ga na spoljnjem obodu neke manje meteorološke stanice. Ono što je zanimljivo je da se BME280 sporije stabilizuje od DHT11. U nastavku sledi detaljnija analiza testiranja.

Kao što je već navedeno, grafici su proizišli iz merenja pod tri različita uslova i to, kao što je dato na Slici 5:

1. Gde je uređaj postavljen da neko vreme meri sobnu temperaturu, vlažnost vazduha, kao i pritisak, dok je u međuvremenu posmatrano ponašanje i brzina promena parametara prikazanih na displeju;
2. Zatim je u šoljicu dodata prokljućala voda, čija je uloga bila da podigne temperaturu, kao i vlažnost vazduha pomoću već formirane vodene pare, međutim da bi se to efikasnije i brže izvelo, uređaj je zajedno sa šoljicom postavljen u okviru jedne posude, kao što je prikazano na Slici 3;
3. Parametri su praćeni sve dok nakon nekoliko minuta posuda nije sklonjena, a zatim je sve do stabilizacije parametara merenih ovim komponentama, uređaj ostao uključen. Svi navedeni uslovi važe i za Slike 6, 7 i 8.

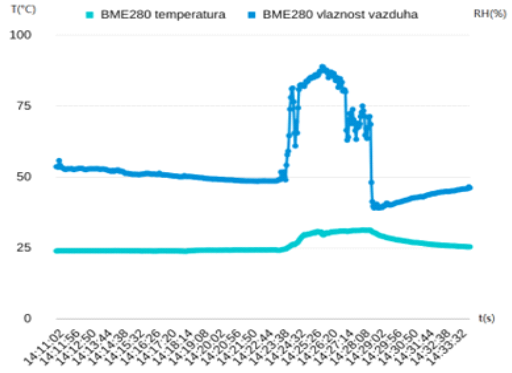


Slika 5. DHT11 promena temperature i vlažnosti vazduha u toku vremena.

Postupak merenja ponovljen je i kako bi se podaci sačuvali u fajl i kasnije mogli da prevedu u grafike.

Ovako sačuvanim podacima, formirani su grafici koji porede merenja na DHT11 i BME280 (Slika 5 i Slika 6), na kojima se lakše može videti da se pri temperaturi od 25°C vlažnost vazduha kod DHT11 kreće negde oko 60%, dok je to kod BME280 blago niža vrednost oko 50%.

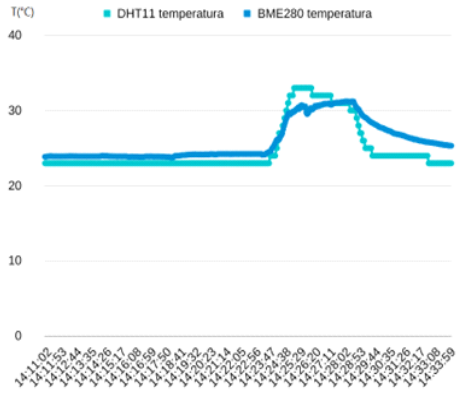
Uvođenjem laboratorijskih uslova, primećujemo drastične promene, ne toliko temperature, koja raste kod DHT11 do 33°C, dok je to kod BME280 za tri stepena niža temperatura, koliko velike promene zasićenosti vodene pare u vazduhu i to porasta od oko 30% kod oba senzora od početne vrednosti vodene pare.



Slika 6. BME280 promena temperature i vlažnosti vazduha u toku vremena.

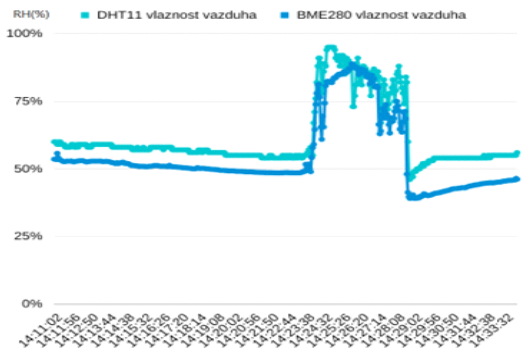
Na Slici 7. formiran je grafik samo za poređenje izmerenih temperatura, radi lakšeg poređenja dva senzora. Ono što se uočava pod uslovima br.1 (datim na Slici 5) jeste da BME280 meri blago povišenu temperaturu u odnosu na DHT11, mada je to i navedeno u tehničkoj dokumentaciji za ovu komponentu (Specifikacije za merenje temperature, "Od značaja je znati da je merenje temperature ovde unutrašnje i koristi se pre svega zbog kalibracije senzora pritiska i vlažnosti. Senzori se zagrevaju, pa je ova temperatura za nijansu viša od stvarne.").

Međutim tokom merenja u laboratorijskim uslovima, DHT11 dostiže višu temperaturu, jer je manje osetljiv na promene i brže se stabilizuje od BME280. U poslednjoj fazi merenja, primećujemo opet sporiju stabilizaciju temperature kod BME280 i ovde se uočava veća razlika nego na početku od oko 3°C.



Slika 7. Promena temperature u toku vremena na DHT11 i BME280.

Na poslednjem grafiku upoređena je vlažnost vazduha u tri faze merenja i tu se moglo uočiti da tokom sve tri faze DHT11 beleži vlažnost vazduha za prosečno 10% veću u prvoj fazi, a oko 15% veću u fazi stabilizacije uređaja, dok su se u laboratorijskim uslovima razlike smenjivale, međutim svakako je potvrđeno da DHT11 jeste manje osetljiv i pri ovoj vrsti merenja.



Slika 8. Promena vlažnosti vazduha u toku vremena na DHT11 i BME280.

III. KORIŠĆENE KOMPONENTE

A. Arduino Uno

Arduino je mikrokontrolerska ploča namenjena razvoju aplikacija kod kojih je potrebno na jednostavan

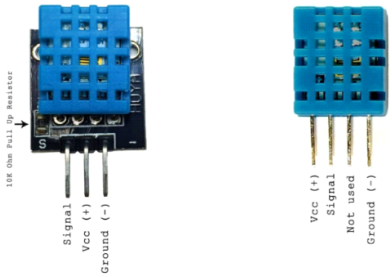
način ostvariti spregu između hardverskih i softverskih komponenti [5]. Hardver se sastoji od štampane ploče zasnovane na 8-bitnom Atmel AVR, ili 32-bitnom Atmel ARM mikrokontroleru. Razvoj softvera je omogućen korišćenjem Open-source integrisanog razvojnog okruženja. Veza sa hardverom ostvarena je preko serijskog USB interfejsa koji se nalazi na samoj ploči i preko kojeg se program koji je napisan i kompajliran u razvojnom okruženju prebacuje u programsku FLASH memoriju mikrokontrolera. Model Arduino UNO sadrži 8-bitni ATMega328 mikrokontroler. Verzija korišćena u radu je opremljena USB interfejsom, zajedno sa 6 analognih ulaza i 14 digitalnih ulaza/izlaza opšte namene [6]. Arduino IDE (engl. *Integrated Development Enviornmen*t) je softversko okruženje koje se koristi za razvoj softvera za različite tipove Arduino platformi. Moguće ga je besplatno preuzeti preko web stranice http://www.arduino.cc/ [7].

B. Digitalni senzor temperature i vlažnosti vazduha DHT11

Digitalni senzor temperature i vlažnosti vazduha DHT11 ima opšte karakteristike:

Opsega relativne vlažnosti vazduha: 20-90% RH;
Preciznosti relativne vlažnosti vazduha: +-5% RH;
Opsega temperature: 0-50°C;
Preciznosti temperature: +-2%°C;
Radnog napona: 3 V do 5.5 V.

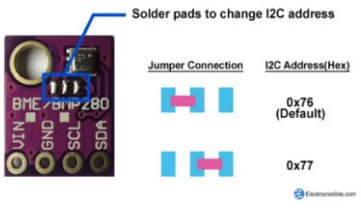
DHT11 meri temperaturu površinski montiranim NTC temperaturnim senzorom (termistorom) ugrađenim u jedinicu. Termistori su promenljivi otpornici koji svoju otpornost menjaju sa temperaturom. Klasifikovani su prema načinu na koji menjaju otpornost u zavisnosti od temperature. Kod *Negative Temperature Coefficient* (NTC) termistora sa povećanjem temperature, otpornost se smanjuje, dok se kod *Positive Temperature Coefficient* (PTC) termistora otpornost povećava sa povećanjem temperature. [8] NTC termistori su najčešći tip, koji se koristi kod DHT11 senzora. DHT11 detektuje količinu vodene pare mereći električnu otpornost između dve elektrode. Senzor vlažnosti vazduha je podloga koja zadržava vlagu sa dve elektrode na svojoj površini. Kada vodena para biva apsorbovana podlogom, joni su oslobođeni iz podloge, što je uzrokovano povećanom provodnošću među elektrodama. Promena u otpornosti je onda srazmerna relativnoj vlažnosti. Veća relativna vlažnost spusta otpornost među elektrodama, dok niža povećava.DHT11 koristi samo jednu žičanu vezu da prosleđuje podatke do Arduina. Pull-up otpornik od 10 kΩ je neophodan između signal žice i napona od 5 V da bismo bili sigurni da je napon uvek na visokom nivou. Postoje dve varijante kućišta i to jedna sa četiri pina, od kojih je jedan NC, i druga PCB varijanta sa ugrađenim pull-up otpornikom (Slika 9). U ovom radu je korišćena PCB verzija kućišta.



Slika 9. Kućište DHT11 senzora sa tri i četiri pina.

C. Senzor vazdušnog pritiska, visine, temperature i vlažnosti vazduha BME280

Za početak je važno, radi daljih merenja, napraviti razliku između apsolutne i relativne visine. Termin apsolutna visina se odnosi na visinu iznad nivoa mora *(Mean Sea Level)*, dok relativna visina označava visinu iznad površine tla *(Above Ground Level)*. BME280 ne meri direktno visinu, već je pretpostavlja na osnovu očitavanja vazdušnog pritiska. Obzirom da je BME280 dobar u merenju pritiska, može izuzetno precizno izračunati i relativnu visinu. Ukoliko je potrebno prikazati apsolutnu visinu, BME280 mora imati podatak o trenutnom pritisku nivou mora. SEA_LEVEL_PRESSURE konstanta u softverskom kodu na primer, se osvežava u zavisnosti od toga gde se senzor nalazi. Ovaj senzor raspolaže elementima koji mere vazdušni pritisak, visinu, temperaturu i vlažnost vazduha. Radi na naponu od 3,3 V do 5 V. BME280 može se povezivati preko I²C ili SPI protokola (Slika 10). Obzirom na korišćenje manjeg broja pinova i jednostavnije realizacije u ovom projektu je korišćen I²C protokol, koji je objašnjen u narednom poglavlju [9]. BME280 se može lako povezati na bilo koji mikrokontroler koristeći I²C. Prednost korišćenja I²C je što možemo koristiti mnogo senzora na jednu magistralu. Ovaj modul poseduje opciju da promeni I²C adresu odlemljivanjem jedne i zalemljivanjem druge veze, tako da dva senzora BME280 mogu biti povezana na istu magistralu (Slika 10).

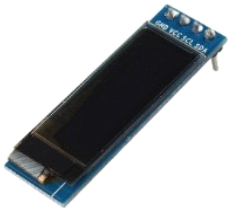


Slika 10. Promena adrese kod BME280.

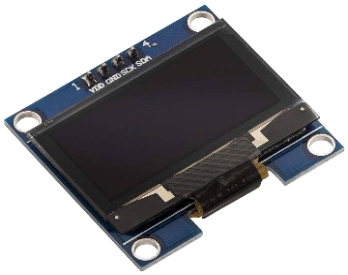
D. OLED display

OLED (Organic Light Emitting Diode) displeji su dipleji sačinjeni od super tankih organskih slojeva, nalik na plastične folije, koji počinju da svetle čim se kroz njih pusti

slaba struja [10]. Organski proces korišćen kod OLED-a, naziva se elektrofosforescencija. Kod OLED displeja, za razliku od klasičnih LED displeja i u odnosu na LCD (Liquid Crystal Display), nije potrebno pozadinsko osvetljenje kako bi se ostvarila slika. Zbog toga, nema potrebe za polarizatorima i drugim pomoćnim sredstvima koja se koriste kod LCD. Za prikaz rezultata, iskorišćen je OLED displej, zbog primera korišćenja više slave uređaja na jedan master kod I²C-a. Drugu grupu displej modula čine OLED displeji za koje je drajver SSD1306 kolo [11]. SSD1306 drajver komunicira sa MCU korišćenjem I²C komunikacije. Pored SDA i SCL pinova potrebnih za I²C komunikaciju, drajver ima i pin za napajanje (mnogo praktičniji vrednosti, od 3,3 V i pin za masu (GND)).Treba razlikovati dva modula, jedan sa 128x32 piksela, drugi sa 128x64 piksela (Slika 11 i Slika 12). Oba imaju varijacije koje se mogu povezivati korišćenjem I²C i SPI komunikacije. SSD1306 kontroler drajver ima 30 pinova, ali korišćenjem komunikacije se kontrola celog drajvera može vršiti sa dva pina *(Serial Data (*SDA*) i Serial Clock (*SCL)).



Slika 11. OLED display 128x32.



Slika 12. OLED display 128x64.

IV. I²C KOMUNIKACIONI PROTOKOL

I²C *(„Inter-Integrated Circuit")* komunikacioni protokol koristi se za mala rastojanja, najčešće za povezivanje integrisanih kola na štampanoj ploči. Za ovaj protokol su potrebne dve fizičke linije tipa open drain. Ove linije se nazivaju SCL (*Serial Clock*) i SDA (*Serial Data*). Na I²C magistrali se mogu vezati 112 kola, a njima se pristupa preko 7-bitne adrese [12].

U ovom tipu komunikacije uređaj može da bude postavljen kao master (*master*) ili kao slejv (*slave*). Master uređaj je najčešće mikrokontroler, a slave uređaji su uglavnom neke periferijske jedinice (senzori, memorije ili slično). U I²C komunikaciji može da se nalazi samo jedan master uredaj, a do 2^7 slejv uređaja od kojih svaki mora imati svoju adresu po kojoj je kasnije master uređaj i raspoznaje [13]. Vrlo bitan deo komunikacije su dva *pull-up* otpornika koji su povezani na SCL i SDA liniju. Oni omogućavaju da su SDA i SCL sve vreme podignuti na visoki naponski nivo (logičku jedinicu, odnosno napon napajanja). Komunikacija se vrši tako da uređaji spuštaju napon na linijama na logičku nulu ili tako što ga ne spuštaju, pa napon ostaje na jedinici. Dakle, ni u jednom trenutku nije potrebno uvesti dodatni element koji treba da podiže liniju komunikacije na nivo logičke jedinice (jer to rade otpornici), već samo treba u određenim intervalima spušati napon.

V. SOFTVERSKI KOD U ARDUINO IDE

U početnom delu koda se uključuju biblioteke i defnišu promenljive. Void setup je deo koda u kome započinjemo serijsku komunkaciju, a zatim i proveravamo ispravnost komponenti koje koristimo kako bismo krenuli sa merenjima. Void loop u ovom slučaju poziva samo jednu funkciju print.Values(), koja je opisana iza glavnog koda i služi za očitavanje i prikaz rezultata merenja i to funkcijama:

```
display.print("BME:");
display.print(bme.readTemperature());
display.print((char)247);
display.print("C");
```

gde display.print jeste funkcija biblioteke za displej, koja umesto na Serial Monitor rezultate merenja prikazuje direktno na displej.

Zatim, bme.readTemperature() prosleđuje podatke za temperaturu BME280, a analogno to rade:

```
bme.readHumidity();
bme.readPressure();
bme.readAltitude(SEALEVELPRESSURE_HA;
DHT.read(DHT11_PIN);
DHT.temperature;
DHT.humidity;
```

svaka za svoj parametar respektivno, dok je (char)247 oznaka za stepen Celzijus.

Deo koda

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 0);
```

definiše veličinu, boju i poziciju teksta takođe funkcijama koje se nalaze u biblioteci za displej.

VI. ZAKLJUČAK

Pored standardnih funkcija meteorološke stanice, ono što je dobijeno kao zaključak testiranja jedne ovakve stanice je da komponente koje su prevashodno namenjene samo merenju određenih parametara bolje izvršavaju svoju funkciju i služe nameni, dok one koje su kombinacija više

parametara pored primarnih, te sporedne mogu koristiti i u druge svrhe, eventualno biti zasebne, ne tako, pouzdane stanice. Sam uređaj se naravno može proširiti dodavanjem naprednijih varijanti senzora, kao što su DHT22 i BME680, zatim dodavanjem anemometara ili snabdevanjem alternativnim izvorom energije, kao što je to na primer solarni panel, da bi sama meteorološka stanica funkcionisala napolju i nezavisno od računara.

ZAHVALNICA

Zahvaljujem se profesoru Danijelu Dankoviću, koji je pomogao u ideji i realizaciji čitavog projekta, kao i asistentkinjama Sandri Veljković i Jani Vračar i asistentu Nikoli Mitroviću na uloženom vremenu oko ispravki u radu i pružanju neophodnih informacija tokom same izrade rada. Posebna zahvalnost profesorkama Emiliji Živanović i Vesni Paunović na podršci, kao i čitavoj Katedri za mikroelektroniku na Elektronskom fakultetu u Nišu.

LITERATURA

[1] "Pogode i nepogode - 150 godina srpske meteorologije", Galerija nauke i tehnike SANU 22. april - 22. Maj 1998.
[2] "Osnovi teoretske fizike", Univerzitet u nišu, Elektronski fakultet, prof. dr Goran Ristić
[3] "Merenje vlažnosti vazduha", prof.Ljiljana Folić
[4] "Vazdušni pritisak", Meteologos, Svet meteorologije (pristupljeno 21.04.2023.)
[5] "Arduino programsko okruženje", Fakultet Tehničkih nauka, Novi Sad (dostupno na: https://www.elektronika.ftn.uns.ac.rs/elektronika-e2/wp-content/uploads/sites/87/2018/03/Arduino-uputstvo.pdf A (pristupljeno 22.04.2023.)
[6] "Prvi koraci sa Arduinom: koje ploče i kompleti mogu biti zanimljiviji za početak",(dostupno na: https://www.hwlibre.com/sr/kits-arduino/, pristupljeno 23.04.2023.)
[7] Arduino UNO datasheet (dostupno na; https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf, pristupljeno 23.04.2023.)
[8] MAKE AN ARDUINO TEMPERATURE SENSOR (THERMISTOR TUTORIAL) (PRISTUPLjENO 28.04.2023., DOSTUPNO NA: https://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial)
[9] "BME280 Pinout, Specs, Applications-Temperature, Pressure & Humidity Sensor Module." (dostupno na: https://electrocredible.com/bme280-pinout-specs-applications/, pristupljeno 23.04.2023.)
[10] "DISPLEJI OLED -3D", Vesna Paunović, Katedra za mikroelektroniku, pristupljeno 28.04.2023.)
[11] "Optoelektronika - Rad sa OLED displejem", Optoelektronika - Laboratorijske vežbe, Vesna Paunović, Nikola Mitrović, Katedra za mikroelektroniku, Elektronski fakultet Niš, 6. maj 2022.
[12] "I²C komunikacija", Miloš Marjanović, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu (dostupno na: https://mikro.elfak.ni.ac.rs/wp-content/uploads/I HYPERLINK "https://mikro.elfak.ni.ac.rs/wp-content/uploads/I%B2C-komunikacija.pdf"²C-komunikacija.pdf, pristupljeno: 23.04.2023.)
[13] "Osnove serijskih komunikacija UART, I²C i SPI", Nikola Mitrović, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, 21.04.2022.

Prototip sistema mikrotalasne pećnice zasnovan na ARM mikrokontroleru

Kristina Milenković, Teodora Đorđević, Miljana Đorđević

Sadržaj - U radu je realizovan prototip sistema mikrotalasne pećnice korišćenjem ARM mikrokontrolera. Sistem se sastoji iz tri tastera. Jedan taster podešava broj desetina sekundi, dok drugi podešava broj sekundi. Kada korisnik klikne na treći taster mikrotalasna pećnica se uključuje što se emulira uključivanjem crvene LE diode i DC motora koji se vrti (imitira okretanje ubačene hrane) podešeno vreme. Podešeno vreme prikazuje se na 2 sedmosegmentna displeja. Dok se motor vrti, na displeju se prikazuje odbrojavanje vremena unazad do 0. Kad istekne predviđeno vreme uključuje se piezo zujalica, tri kratkotrajna tona.

I. Uvod

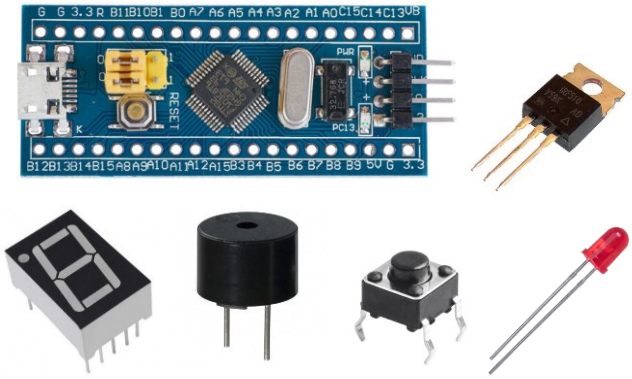
Mikrotalasne pećnice su jedan od najčešće korišćenih kućnih aparata u današnjim domaćinstvima. One su postale ključni deo kuhinje, omogućavajući brzo i jednostavno pripremanje hrane. Mikrotalasne pećnice rade na principu korišćenja mikrotalasnog zračenja za zagrevanje hrane. Ove pećnice emituju visokoenergetske mikrotalase koji prodru u hranu i generišu toplotu. Mikrotalasi prodiru duboko u hranu, zagrevajući je iznutra prema spolja. Jedna od ključnih prednosti mikrotalasnih pećnica je brzina zagrevanja hrane. U poređenju sa konvencionalnim rernama ili šporetima, mikrotalasne pećnice mogu brzo zagrejati hranu za samo nekoliko minuta. Većina modernih modela ima predefinisane programe za različite vrste hrane, kao što su povrće, meso ili supa. Postavljanjem vremena i odabirom odgovarajućeg programa, pećnica će sama prilagoditi snagu i vreme zagrevanja za tu vrstu hrane. Važno je napomenuti da postoji nekoliko sigurnosnih mera koje treba poštovati prilikom korišćenja mikrotalasnih pećnica. Na primer, neki materijali, poput metala ili aluminijumske folije, ne smeju se koristiti u mikrotalasnoj pećnici jer mogu izazvati varnice. Takođe je važno koristiti posude koje su sigurne za korišćenje u mikrotalasnoj pećnici.

U ovom radu biće projektovan i emuliran sistem mikrotalasne pećnice zasnovane na ARM mikrokontroleru. Sistem omogućava podešavanje željenog vremena, dok se DC motor upravljan korišćenjem NMOS tranzistora snage koristi za emulaciju okretanja hrane u pećnici. Za indikaciju se koriste sedmosegmentni displeji da prikažu vreme, LED da emulira uključeni grejač i piezo zujalica da signalizira završetak ciklusa pripreme hrane.

K. Milenković, T. Đorđević, M. Đorđević su studenti četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: kristinaroze@elfak.rs

II. Komponente korišćene za izradu projekta

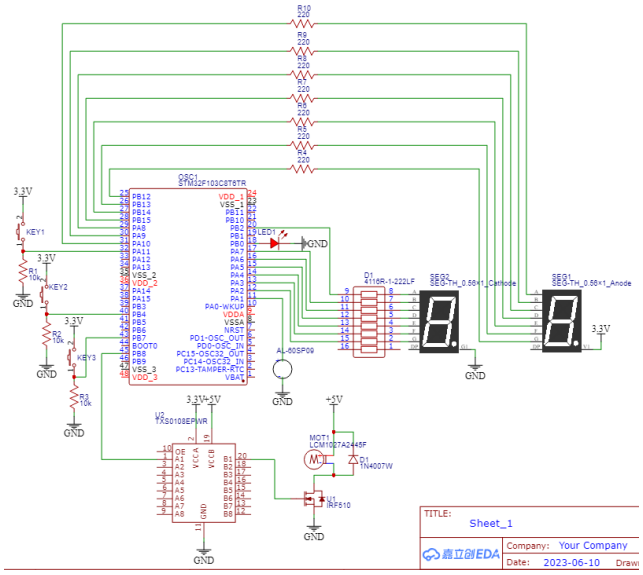
BluePill je razvojno okruženje koje omogućava korisnicima izradu prototipova uz pomoć STM32 mikrokontrolera. Pored ARM mikrokontrolera u ovom projektu korišćen je NMOS tranzistor snage IRF510 u ulozi prekidača za pokretanje DC motora. Za prikaz podešenog vremena koriste se dva sedmosegmentna displeja, i to jedan sa zajedničkom katodom i jedan sa zajedničkom anodom. Za ograničenje struje kroz LED u displeju korišćene su otporne mreže od 220 Ω. Za podešavanje vremena koriste se tasteri u *pull-down* konfiguraciji, sa otpornicima od 10 kΩ. Uporebljeno je ukupno 3 tastera. Za indikaciju uključenog grejača koristi se crveni LED, sa kojim je redno vezan otpornik od 220 Ω da ograniči struju kroz diodu. Za zvučnu indikaciju koristi se piezo zujalica. Izgled korišćenih komponenata prikazan je na slici 1.



Slika 1. Izgled pojedinih elektronskih komponenti korišćenih u projektu.

III. Opis šeme sistema

Električna šema prikazana je na slici 2. Centralni deo sistema je razvojno okruženje *BluePill* sa STM32 mikrokontrolerom. Dva sedmosegmentna displeja služe za prikaz vremena, oni su povezani preko otporne mreže i otpornika 220 Ω na *BluePill*. Na razvojno okruženje je povezan i DC motor preko tranzistora – pin B8. Za aktiviranje tranzistora korišćen je translator napona sa 3.3 V na 5 V. Koristi se LE dioda, povezana na pin B0, koja se uključuje u isto vreme kad i motor. Tri tastera su povezana u *pull-down* konfiguraciji preko otpornika od 10 kΩ na mikrokontroler preko pinova B4, B7 i A11. Piezo zujalica je povezana na pin A1.



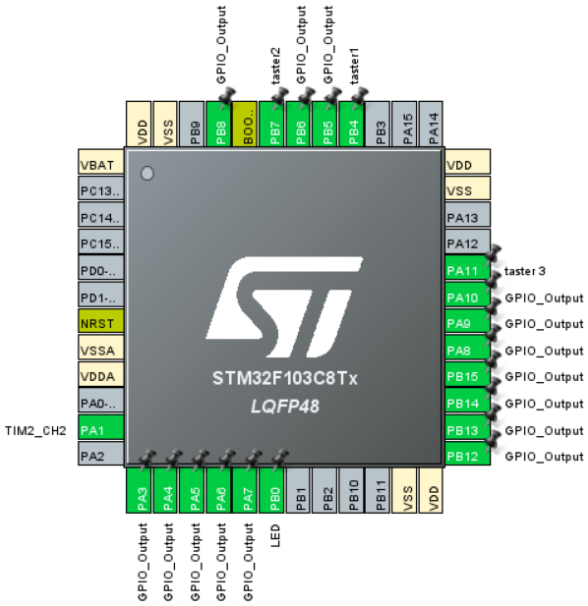
Slika 2. Električna šema realizovanog prototipa.

IV. PODEŠAVANJA U CUBEMX

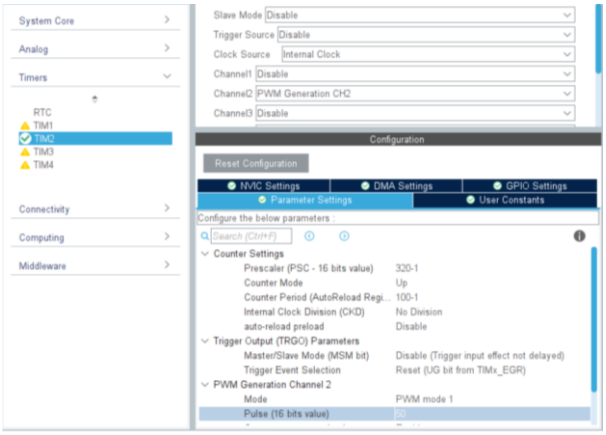
CubeMx je grafički alat koji omogućava vrlo lako konfigurisanje STM32 mikrokontrolera i generisanje kompletnog C projekta za: IAR, KEIL, AC6, ATOLLIC, itd. Na slici 3 prikazani su iskorišćeni pinovi mikrokontrolera. U ovom projektu korišćeno je 14 GPIO pinova kao izlazi za 2 sedmosegmentna displeja i još 2 za DC motor i LED. Tri GPIO ulaza je iskorišćeno za tastere. U *CubeMx*- u postoje 4 tajmera koji poseduju po 4 kanala. U ovom projektu korišćen je tajmer 2 (TIM2) za kontrolu piezo zujalice. TIM2 zauzima pin A1, gde će biti generisan PWM signal za generisanje tona. U *clock source* je izabrana opcija *Internal Clock* i u kanalu 2 podešen je tip na: PWM Generation CH2. Frekvencija tona koji će biti generisan je 250 Hz, to znači da je perioda signala 4 ms. Ako vrednost parametra *Counter Period* postavimo na 100, preskalirana frekvencija će biti 25 kHz. Ako je osnovna frekvencija tajmera 8 MHz, to znači da je vrednost parametra *Prescaler* potrebno postaviti na 320. Faktor ispune PWM signala je inicijalno postavljena kroz parametar *Pulse* na vrednost 50. Ova vrednost može se menjati u toku izvršenja programa. Na slici 4 prikazana su podešavanja tajmera mikrokontrolera koji će upravljati generisanjem zvuka piezo zujalice.

V. PROGRAMSKI KOD

Keil MDK je kompletno okruženje za razvoj softvera za niz ARM Cortex-M baziranih mikrokontrolera. Na početku su deklarisane 2 funkcije displejK() i displejA(). Jedna funkcija je vezana za sedmosegmentni displej koji inicira brojeve od 0 do 5 (desetice), a druga funkcija je za drugi sedmosegmentni displej i inicira brojeve od 0 do 9 (jedinice). Na slici 5 je ilustrovano kako je postavljanjem SET i RESET stanja na pojedine pinove omogućeno prikazivanje cifara na sedmosegmentnim displejima.



Slika 3. Prikaz iskorišćenih pinova mikrokontrolera.



Slika 4. Podešavanje tajmera mikrokontrolera TIM2.

```
void displejK(uint8_t br){

  if(br==0){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
  }
  if(br==1){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
  }
```

Slika 5. Funkcija za sedmosegmentni displej sa zajedničkom katodom.

U nastavku se definišu sve promenljive. Prvih 10 promenljivih su vezane za gore pomenute funkcije, kako bi se pozivale određene cifre koje je potrebno ispisati na displeju. Flegovi su definisani zbog tastera. Desetice, jedinice i promenljiva "a" su definisane zbog nove funkcije koja objedinjuje oba displeja. *DutyCycle* i promenljiva "i" su vezani za deo koda koji kontroliše piezo zujalicu.

```
uint8_t nula=0;
 uint8_t jedan=1;
 uint8_t dva=2;
 uint8_t tri=3;
 uint8_t cetiri=4;
 uint8_t pet=5;
 uint8_t sest=6;
 uint8_t sedam=7;
 uint8_t osam=8;
 uint8_t devet=9;
 int fleg=0;
 int fleg2=0;
 int fleg3=0;
 int jedinica;
 int desetica;
 int a=0;
 uint16_t dutyCycle=50;
 int i;
```

Slika 6. Deklaracija promenljivih.

Napisana je funkcija koja objedinjuje oba displeja. Ovde se inicijalizuje promenljiva "a" koja služi da se dobiju jedinice i desetice. Desetice se dobijaju sa naredbom a/10, a jedinice a%10. Zatim se *if* naredbama ispituje da li je jedinica jednaka 0, ukoliko jeste prikazuje se 0 na displeju. Isti je postupak za sve cifre jedinica i desetica, s tim da se cifre desetice ispisuju na drugom displeju. Deo koda ove funkcije ilustrovan je na slici 7.

```
void display(int a){

  int desetica=a/10;
  int jedinica=a%10;

  if(jedinica==0){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
  }
  else if(jedinica==1){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8 , GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13 , GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
  }
```

Slika 7. Deo funkcije za ispisivanje cifara desetica i jedinica na sedmosegmentim displejima.

U glavnoj programskoj petlji ispituje se pritisak tastera. Ukoliko je pritisnut taster za podešavanje cifre desetice, kroz *if* strukturu se donosi odluka koja cifra će biti prikazana na displeju, deo koda dat je na slici 8. Promenljiva „a" se uvećava za 10, s obzirom da se ovim tasterom podešava cifra desetice. Slična logika upotrebljena je za proveru tastera kojim se podešavaju cifre jedinice i inkrementiraju te cifre. U ovom slučaju vrednost promenljive „a" uvećava se za 1.

```
    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4) && fleg==0){
    HAL_Delay(500);
    displejK(jedan);
    fleg=1;
    HAL_Delay(500);
      a=a+10;

  }
  if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4) && fleg==1){
    HAL_Delay(500);
    displejK(dva);
    fleg=2;
    HAL_Delay(500);
    a=a+10;
```

Slika 8. Deo koda za proveru tastera i inkrementiranje cifre desetice na displeju.

Klikom na treći taster, vrednost koja je zadata prvim i drugim tasterom treba odbrojavati unazad. Treći taster je povezan na pin A11. Ukoliko je ovaj taster pritisnut uključuje se LE dioda setovanjem pina B0, kao i pina B8 koji kontroliše DC motor. Pomoću *while* petlje koja se pokreće sve dok je promenljiva „a" veća ili jednaka sa 0 pomoću funkcije *display(a)*, promenljiva „a" se smanjuje sa „a--" za 1 na svaku sekundu, što je postignuto korišćenjem funkcije *delay*. Kada se odbroji do nule, LED pin se prebacuje u stanje RESET, kao i pin za kontrolu DC motora. Za aktivaciju piezo zujalice, koja treba da emituje tri kratkotrajna tona iskorišćena je *while* petlja u okviru koje se postavlja vrednost faktora ispune PWM signala koji se dovodi na zujalicu. Treba napomenuti da je u kodu potrebno inicijalizovati tajmer i postaviti početnu vrednost faktora ispune na nulu. Kod je prikazan na slici 9.

```
if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_11) && fleg3==0){
  HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET); //ukljucivanje led diode

  HAL_GPIO_WritePin(GPIOB,GPIO_PIN_8, GPIO_PIN_SET);  //ukljucivanje motora
   fleg3=1;


      while (a>= 0)


{

   display(a);
   a--;

HAL_Delay(1000);

 }

HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET); //iskljucivanje led diode
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_8, GPIO_PIN_RESET);  //iskljucivanje motora

while (i<3){
  htim2.Instance-> CCR2=dutyCycle;
  HAL_Delay(500);
  htim2.Instance-> CCR2=0;
  HAL_Delay(500);
  i++;

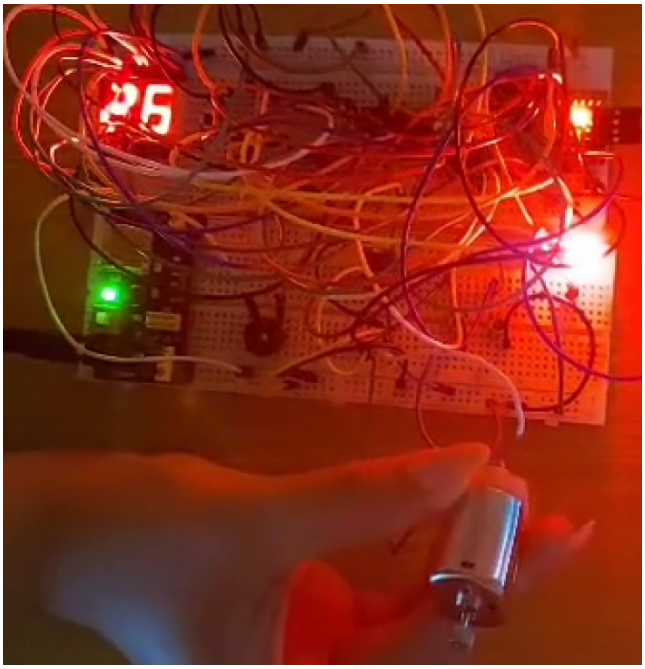
}

  htim2.Instance-> CCR2=0;
```

Slika 9. Deo koda za proveru tastera za početak odbrojavanja podešenog vremena unazad sa uključivanjem-isključivanjem LED- a, DC motora i pezo zujalice.

VI. Praktična realizacija

Na slici 10 prikazana je realizacija na protoploči. Zbog velikog broja provodnika slika nije mogla da bude preglednija. Na slici 10 se vidi da je trenutno vreme 26, uključena je crvena LED dioda. Iako fotografija ne može da dočara, DC motor se vrti.



Slika 10. Praktična realizacija sistema.

VII. Zaključak

Minimalnim izmenama moguće je usavršiti ovaj sistem da bude kompatibilan i za druge primene. Vrlo lako se štoperica može podesiti da odbrojava od 99 do 0, ali dodavanjem još jednog sedmosegmentnog displeja moguće je "pokriti" i sve trocifrene brojeve. Moguće je dodati i displej na kome će se ispisivati poruke poput: "Hrana je spremna.", "Podesi vreme" i sl. Na osnovu ovakvog funkcionalnog prototipa moguće je izraditi štampanu ploču uređaja.

Zahvalnica

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

Literatura

[1] STM32F103C8 Blue Pill razvojno okruženje. Dostupno na: https://components101.com/microcontrollers/stm32f103c8t8-blue-pill-development-board.
[2] Tehnička specifikacija za STM32F103. Dostupno na: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf.
[3] Tehnička dokumentacija za MOS transistor snage IRF510. Dostupno na: https://www.vishay.com/docs/91015/irf510.pdf.
[4] Nikola Mitrović, Miloš Marjanović, Danijel Danković, *„Primena ARM mikrokontrolera“*, Akademska Misao, 2023. ISBN: 978-86-7466-951-8

Digitalni uglomer baziran na ADXL345 akcelerometru

Anastasija Avramović

Sadržaj – U radu je realizovan digitalni uglomer baziran na ADXL335 akcelerometru. Ovaj uređaj je vrlo koristan alat za merenje uglova u različitim primenama. Na osnovu *Nucleo* okruženja i akcelerometra omogućeno je precizno merenje uglova i dobijanje tačnih rezultata. Digitalni uglomer ma široku primenu u različitim industrijama, poput građevinarstva, automobilske industrije, avioindustrije, sportske opreme i mnogih drugih.

I. Uvod

U modernom dobu, precizno merenje uglova i nagiba igra ključnu ulogu u različitim industrijskim sektorima. Ovaj rad istražuje realizaciju digitalnog uglomera baziranog na ADXL335 akcelerometru i Nucleo okruženju, koji predstavljaju napredne tehnološke alate za merenje i praćenje promena u uglovima i nagibima [1].

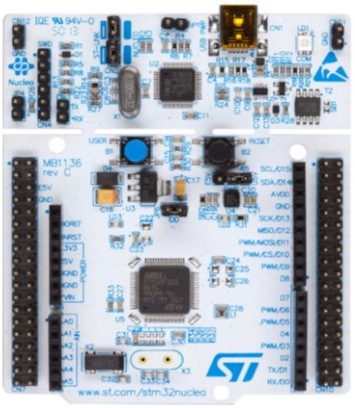
Ideja za ovaj projekat proizlazi iz sve veće potrebe za tačnim i pouzdanim merenjima uglova u različitim primenama. Digitalni uglomeri postaju ključni instrumenti u industrijama kao što su građevinarstvo, automobilska industrija, avioindustrija, sportska oprema i mnoge druge. Njihova sposobnost da pruže precizne podatke o položaju i orijentaciji ima široku primenu u projektima koji zahtevaju kontrolu i praćenje geometrije.

II. Korišćene komponente za realizaciju projekta

A. Nucleo STM32F103RB

Nucleo STM32F103RB je razvojno okruženje za mikrokontrolere bazirane na ARM Cortex-M3 jezgru. STM32F103RB mikrokontroler je osnovna komponenta *Nucle*o okruženja. Ovo okruženje je dizajnirano da omogući laku izradu prototipova i razvoj softvera za širok spektar aplikacija u okviru popularnih razvojnih okruženja kao što su *Keil MDK-ARM, IAR EWARM i GCC-based toolchains*. Velika prednost ovog okruženja ogleda se u tome što podržava različite komunikacijske interfejse koji se mogu koristiti za različite vrste komunikacije između mikrokontrolera i drugih uređaja. Među podržanim interfejsima su *USB, CAN, I2C, SPI i UART* [2]. *Nucleo* okruženje prikazano je na slici 1.

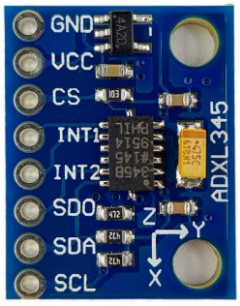
A. Avramović je student master akademskih studija na modulu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anastasijaavram@elfak.rs



Slika 1. Nucleo okruženje.

B. Akcelerometar ADXL345

ADXL345 je digitalni 3-osni akcelerometar visoke rezolucije, koji omogućava merenje statičkog gravitacionog ubrzanja. Podržava komunikaciju preko *SPI ili I2C* protokola. Izlazni podaci su formatirani kao 16-bitni broj u dvostrukom komplementu. Visoka rezolucija merenja od 13 bita do 16 g, omogućava merenje uglova manjih od 1°. Pored toga, poseduje specijalne funkcije poput detekcije slobodnog pada i *FIFO* bafera koji rasterećuju *host* procesor. Uređaj radi na napajanju od 2 do 3,6 V i poseduje *low-power mod*. ADXL345 se koristi u različitim primenama, uključujući medicinsku instrumentaciju, uređaje za video igre i industrijsku instrumentaciju, kao i zaštita *HDD-a* [3]. Prikazan je na slici 2.

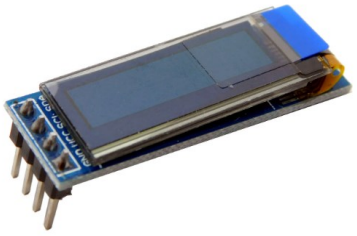


Slika 2. Akcelerometar ADXL345.

C. OLED displej

OLED SSD1306 (slika 3) modul predstavlja moderniji tip displeja za primenu u embedded sistemima, pre svega zbog svojih malih gabarita [4]. Povezuje se preko

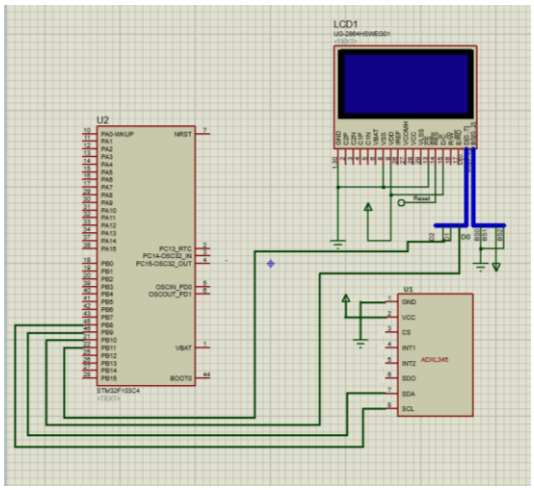
samo 4 pina, korišćenjem *I2C* komunikacije, što rasterećuje *MCU*. Koristi SSD1306 drajver koji omogućava veliki broj podešavanja.



Slika 3. OLED displej.

III. Opis šeme sistema

Za projektovanje električne šeme korišćen je softverski paket *Proteus*. Električna šema prikazana je na slici 4. Centralni deo sistema je razvojno okruženje *Nucleo* sa STM32F103 mikrokontrolerom [2]. *OLED* [4] se fizički povezuje na razvojno okruženje, *I2C* protokol služi za komunikaciju. Displej služi za prikazivanje vrednosti izmerenog ugla. Na razvojno okruženje je povezan i akcelerometar, takođe preko *I2C* protokola [3].

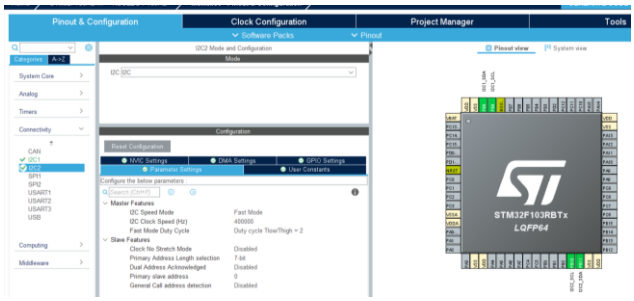


Slika 4. Električna šema sistema.

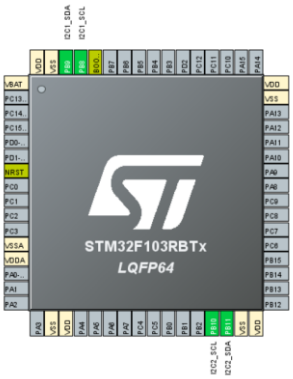
IV. Podešavanja u CubeMx

CubeMx je grafički alat koji omogućava vrlo lako konfigurisanje STM32 mikrokontrolera i generisanje kompletnog C projekta za*: IAR, KEIL, AC6, ATOLLIC*, itd. Na slici 5 se može videti podešavanje I2C protokola koji je neophodan za komunikaciju sa OLED [4] displejem. Pinovi PB11 i PB10 služe kao *SDA* i *SCL*. Preko *SDA* linije se šalju podaci, dok *SCL* služi za generisanje signala takta za *OLED*. Koristi se *I2C fast mode* za brzo slanje podataka.

Takođe je potrebno podesiti *I2C* protokol za komunikaciju sa akcelerometrom. Pinovi PB9 i PB8 služe kao *SDA* i *SCL*. Na slici 6 su prikazani svi potrebni pinovi Ovim su završena sva podešavanja koja se tiču *CubeMX-a*.



Slika 5. Podešavanje I2C protokola.



Slika 6. Svi potrebni pinovi.

V. Kod u *Keil-u*

Keil MDK je kompletno okruženje za razvoj softvera za niz ARM *Cortek-M* baziranih mikrokontrolerskih uređaja. Najpre je bilo potrebno ubaciti dodatne biblioteke *fonts* i *ssd1306* za rad sa *OLED* displejem. Biblioteke se uključuju naredbom *#include* "ime_biblioteke.h". Zatim, jedan deo koda odnosi se na deklaraciju promenljivih. Drugi deo koda namenjen je akcelerometru. U okviru toga definisane su funkcije "adxl_upisi" i "adxl_procitaj" a unutar *USER CODE 3* opisano je merenje ugla i prikazivanje izmerene vrednosti na *OLED* displeju.

Funkcija "adxl_upisi" služi za upisivanje vrednosti u registar ADXL345 senzora. Funkcija ne vraća nikakvu vrednost, već zahteva dva argumenta: oznaku registra i vrednost koja se prosleđuje u taj registar. Funkcija koristi *HAL* biblioteku za *I2C* komunikaciju i zahteva pointer na handler I2C komunikacije. Podaci se šalju u nizu od dve vrednosti (2 bajta) na adresu koja odgovara ADXL345 senzoru. Funkcija "adxl_procitaj" služi za čitanje registara ADXL345 senzora. Funkcija takođe ne vraća nikakvu vrednost, već zahteva dva argumenta: oznaku registra i broj bajtova koje želimo da pročitamo.

Za čitanje registara koristi se funkcija "*HAL_I2C_Mem_Read*", koja ima pointer adresu sa koje vrši učitavanje registara zapisanu kao 1 bajt, "data rec" u koju se smeštaju podaci koji se čitaju sa adrese registra. Za

svaku od tri ose se čitaju po 2 registra, što čini ukupno 6 bajtova podataka. Funkcije su prikazane na slici 7. Funkcija "adxl_upiši" se koristi za inicijalizaciju uređaja i konfiguraciju komunikacije sa njim. Nakon inicijalizacije, funkcija "adxl_procitaj" se koristi za čitanje podataka sa senzora, a zatim se ti podaci obrađuju kako bi se izračunao ugao rotacije.

```
void adxl_upisi (uint8_t reg, uint8_t vrednost)
{
uint8_t podaci[2];
podaci[0]=reg;
podaci[1]=vrednost;

HAL_I2C_Master_Transmit(&hi2c1,adxl_adresa,podaci,2,10);
}

void adxl_procitaj(uint8_t reg, uint8_t brojBajtova)
{
HAL_I2C_Mem_Read(&hi2c1,adxl_adresa,reg,1,data_rec,brojBajtova,100);
}
```

Slika 7. Funkcije akcelerometra.

Unutar *USER CODE 3* (slika 8) nalazi se deo koda koji se odnosi na merenje ugla. U ovom delu koda se koristi funkcija adxl_procitaj za čitanje 6 bajtova. Nakon što su podaci pročitani, njihove vrednosti se smeštaju u promenljive x, y i z.

Kada se dobiju pravilne vrednosti x, y i z, mogu se pomnožiti sa faktorom osetljivosti 0.0078 kako bi se dobilo ubrzanje u g. Ugao rotacije se računa kao arcsin(xg) i konvertuje se u stepene, što se čuva u promenljivoj angle koja je tipa float. Kako bi se izbegle decimale, angle se konvertuje u celobrojnu vrednost, koja se čuva u promenljivoj angle_int.

```
adxl_procitaj(0x32,6);
x=(data_rec[1]<<8|data_rec[0]);
y=(data_rec[3]<<8|data_rec[2]);
z=(data_rec[5]<<8|data_rec[4]);

xg=x*0.0078;
yg=y*0.0078;
zg=z*0.0078;

//izracunavanje ugla rotacije


angle=asin(xg)*180.0/PI;
int angle_int=(int)angle;
```

Slika 8. Deo koda koji se odnosi na merenje ugla.

U nastavku sledi deo koda koji se odnosi na prikazivanje vrednosti izmerenog ugla (slika 9). Funkcija *SSD1306_Fill()* služi za uključivanje ili isključivanje svih dioda displeja u zavisnosti od argumenta koji joj je prosleđen (‘0’ isključuje, ‘1’ uključuje sve piksele). Funkcijom *SSD1306_GotoXY()* vrši se pozicioniranje na displeju. *SSD1306_Puts()* je funkcija kojom se ispisuje tekst na displeju.

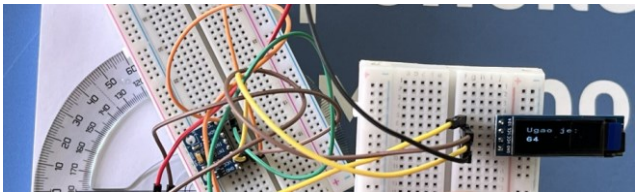
Kao argumente uzima sam tekst napisan pod navodnicima, font kojim će on biti ispisan, a ‘1’ na kraju služi za uključivanje piksela koji će prikazivati tekst. Pošto se na *OLED* displeju samo mogu prikazivati promenljive tipa string, neophodno je vrednost angle_int koje su tipa integer smestiti u novu promenljivu koja je tipa string. U tu svrhu se koristi funkcija *sprintf()* koja kao argumente uzima promenljivu tipa string u koju se smeštaju vrednosti, zatim “%d” što označava da je brojna vrednost koja se smešta u novu promenljivu dekadno prikazana. Funkcija *SSD1306_UpdateScreen()* služi za osvežavanje displeja odnosno za primenu svih prethodno navedenih funckija za pozicioniranje i ispis teksta na displeju.

```
SSD1306_Fill(0);
SSD1306_GotoXY(10, 10);
SSD1306_Puts("Ugao je:", &Font_11x18, 1);
SSD1306_GotoXY(10, 40);
sprintf(NekiString, "%d", angle_int);
SSD1306_Puts(NekiString, &Font_11x18, 1);
SSD1306_UpdateScreen();
HAL_Delay(2000);
SSD1306_Fill(0);
SSD1306_UpdateScreen();
```

Slika 9. Deo koda za prikazivanje izmerene vrednosti ugla.

VI. Praktična realizacija

Praktična realizacija ovog sistema prikazana je na slici 10. Ovaj sistem za merenje ugla pokazao se kao veoma precizan i funkcionalan. Uz pomoć uglomera, izmeren je ugao od 64 stepena, što ukazuje na to da je sistem u potpunosti ispravan i radi kako treba.



Slika 10. Praktična realizacija sistema.

VII. Zaključak

Digitalni uglomeri su korisni elektronski uređaji koji pružaju brže i preciznije merenje ugla u različitim aplikacijama i mogu biti dizajnirani sa dodatnim funkcijama koje olakšavaju upotrebu. Ovaj projekat može se unaprediti na mnogo načina, što zavisi od volje i kreativnosti pojedinca. Umesto *OLED-a* može se iskoristiti bilo koji displej. Takođe može se kombinovati sa žiroskopom, kako bi se poboljšalo očitavanje ugla i povećala preciznost.

ZAHVALNICA

Autor se zahvaljuje Milošu Marjanoviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji ovog projekta.

LITERATURA

[1] Digital protractor Dostupno na: https://m-testco.com/digital-precision-protractor/
[2] Tehnička specifikacija za NucleoF103. Dostupna na: https://os.mbed.com/platforms/ST-Nucleo-F103RB/
[3] Tehnička dokumentacija za akcelerometar ADXL345 Dostupno na: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf
[4] Tehnička dokumentacija za SSD1306 drajver. Dostupna na: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf

Projektovanje i emulacija kulera za računar

Sava Dinić, Andrija Gogić, Damjan Dinić, Uroš Vasiljković, Vladimir Rudić

Sadržaj – U ovom radu je realizovan sistem koji uz pomoć *Blue pill* mikrokontrolerskog okruženja kontroliše brzinu okretanja DC motora koji emulira kuler (hladnjak). U zavisnosti od temperature na laptopu, brzina okretanja kulera (DC motora) biće različita. Brzina motora se smanjuje i povećava proporcionalno temperaturi. Kolo se uključuje i isključuje tasterom. Trenutna temperatura koju meri senzor se prikazuje na sedmosegmentnim displejima. Takođe, 3 LE diode se koriste za signalizaciju temperature. Ako je temperatura preko 15°C uključuje se zelena dioda i kuler radi na 20% maksimalne brzine. Preko 23°C, DC motor radi na 40% maksimalne brzine. Kada temperatura pređe 30°C, uključuje se žuta LE dioda (signal za upozorenje) i kuler radi na 60% maksimalne brzine. Na 42°C kuler radi na 80%. Na temperaturama preko 52°C, uključuje se crvena LE dioda (signalizacija da je temperatura prevelika), gde kuler radi na maksimalnoj brzini, odnosno na 100%. Pored LE dioda, za signalizaciju se koriste OLED displej, kao i serijski monitor, koji prikazuju na koliko procenata radi kuler.

I. Uvod

Hlađenje računara je ključni aspekt njegovog pravilnog funkcionisanja i održavanja optimalnih radnih temperatura komponenti. Kako računari postaju sve snažniji i zahtevniji, generišu više toplote, što može dovesti do pregrevanja i smanjenja performansi. Procesor je jedna od najvažnijih komponenti u računaru koja proizvodi veliku količinu toplote tokom rada. Da bi se sprečilo pregrevanje, procesori su opremljeni hladnjacima koji koriste ventilatore i/ili hladnjake kako bi odvodili toplotu sa površine procesora. Takođe, termalna pasta se koristi za poboljšanje prenosa toplote između procesora i hladnjaka. Kućište računara je opremljeno ventilatorima koji pomažu u cirkulaciji vazduha unutar kućišta. Ovi ventilatori pomažu u izvlačenju toplote koja se generiše od komponenti kao što su grafička kartica, napajanje i memorijske komponente. Naprednija opcija hlađenja računara je tečno hlađenje. Ova metoda koristi rashladnu tečnost koja se pumpa kroz cevi postavljene na komponente kao što su procesor i grafička kartica. To omogućava brži i efikasniji prenos toplote od komponenti do hladnjaka koji se nalazi izvan kućišta računara. Tečno hlađenje često se koristi u računarima koji zahtevaju ekstremno hlađenje, kao što su računari za igranje ili serverske konfiguracije. Da bi se postigla optimalna temperatura unutar računara, važno je pravilno upravljati protokom vazduha. To uključuje postavljanje ventilatora na odgovarajućim mestima kako bi se osiguralo pravilno usmeravanje svežeg vazduha i izvlačenje toplote. Uz napredak tehnologije, proizvođači računara stalno rade na unapređenju sistema hlađenja kako bi se obezbedilo optimalno funkcionisanje računara. Pravilno hlađenje računara, osim što održava stabilnost sistema i poboljšava performanse, produžava i životni vek komponenti.

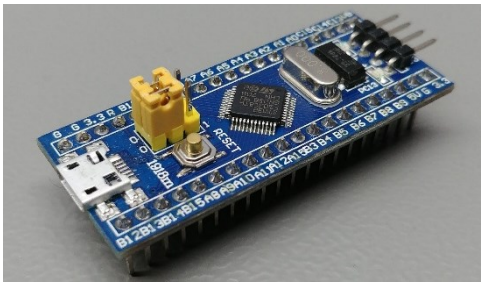
Ovaj rad se bavi projektovanjem i implementacijom sistema za kontrolu brzine rada kulera za računar. Brzina DC motora se kontroliše PWM signalom čiji faktor ispune je srazmeran sa temperaturom koju očitava senzor. Izmerena temperatura i brzina okretanja motora prikazuju se na sedmosegmentim displejima, OLED-u i šalju se putem serijske komunikacije do računara.

S. Dinić, A. Gogić, D. Dinić, U. Vasiljković, V. Rudić su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: savadinic@elfak.rs

II. Korišćene komponente za realizaciju projekta

A. Blue pill

Blue pill (slika 1) predstavlja razvojno okruženje koje omogućava korisnicima izradu prototipova uz pomoć STM32F103C8T6 mikrokontrolera [1].



Slika 1. Blue pill razvojno okruženje.

B. Temperaturni sensor kty 81-210

Temperaturni sensor *kty 81-210* (slika 2) je otporni senzor sa pozitivnim temperaturnim koeficijentom (PTC). Kalibrisan je da meri temperature u opsegu od -55°C do 150°C, sa tolerancijom ±1%. Otpornost na sobnoj temperaturi mu je 2 kΩ. Ovaj senzor dolazi u kućištu SOD70 [2].



Slika 2. Temperaturni sensor *kty 81-210*.

C. OLEDSSD1306

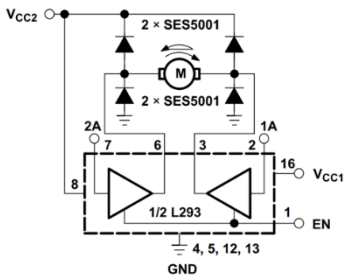
Kod OLED displeja nije potrebno pozadinsko osvetljenje da bi se ostvarila slika. OLED SSD1306 (slika 3) modul predstavlja moderniji tip displeja za primenu u *embedded* sistemima, pre svega zbog svojih malih gabirata[3]. Povezuje se preko samo 4 pina, korišćenjem I2C komunikacije i to linije za takt (SCL - *Serial Clock*), linije za podatke (SDA - *Serial Data*), napajanje i masu.



Slika 3. OLED SSD1306 modul.

D. L293D

Integrisano kolo L293D predstavlja drajver za induktivna opterećenja. Na njemu se mogu povezati dva DC motora. Za potrebe ovog projekta bio je dovoljan jedan. U zavisnosti od temperature šalje se PWM signal od 0 do 100% faktora ispune, koji određuje kojom brzinom će raditi DC motor. Šema za povezivanje L293D [4] sa DC motorom data je na slici 4.

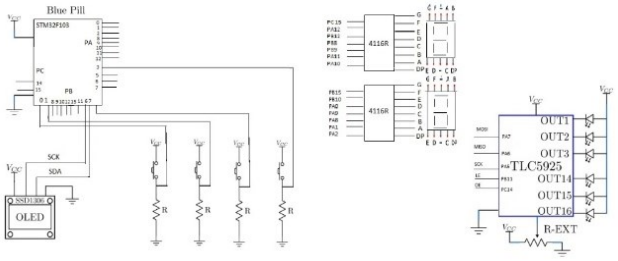


Slika 4. Šema za povezivanje IK L293D sa DC motorom.

Pored ovih komponenata korišćene su: DC motor, taster u *pull-down* konfiguraciji, dva sedmosegmentna displeja i dve otporne mreže, LE diode i otpornici za regulisanje struje, UART (FTDI232) konvertorski čip, HW131 izvor napajanja od 5V za DC motor.

III. Opis šeme sistema

Električna šema prikzana je na slici 5. Glavni deo sistema je razvojno okruženje *Blue pill* sa STM32F103 mikrokontrolerom. Taster je u *pull-down* konfiguraciji i služi za uključivanje i isključivanje sistema. OLED se direktno povezuje na pinove mikrokontrolera, a koristi I2C protokol za komunikaciju. Na šemi sa slike 5 su prikazane oznake linija veze za podatke SDA i takt SCL. Displej služi da prikazuje brzinu okretanja DC motora kulera. Tri LE diode različitih boja koriste se za signalizaciju upozorenja i vezane su redno sa otpronicima od 220Ω za ograničenje struje kroz njih. Zelena dioda signalizira da je sve u redu sa temperaturom, žuta dioda upozorava da je temperatura velika i crvena signalizira da je temperature visoka i da bi računar trebalo isključiti. Dva sedmosegmentna displeja sa otpornim mrežama su povezana na izlaze mikrokontrolera i prikazuju izmerenu temperaturu u svakom trenutku. Temperaturni senzor vezuje se u razdelnik napona sa otpornikom od 2kΩ. Za pokretanje motora koristi se kolo L293D koje kontroliše brzinu okretanja DC motora. Za napajanje DC motora sa stabilnih 5V koristi se eksterni izvor HW131. Za slanje informacija od računara koristi se UART (FTD1232) konvertorska pločica, na kojoj su dostupni pinovi Rx i Tx.

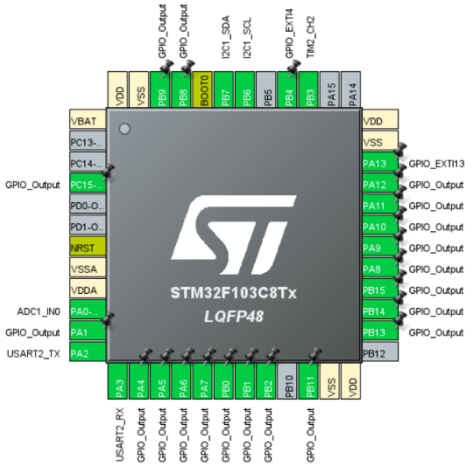


Slika 5. Električna šema sistema.

IV. Podešavanja u CubeMX-u

CubeMX je grafički alat koji omogućava vrlo lako konfigurisanje STM32 mikrokontrolera i generisanje kompletnog C projekta za: IAR, KEIL, AC6, ATOLLIC, itd. Pinovi PB6 i PB7 su pinovi koji se koriste za I2C komunikaciju sa OLED-om, odnosno SCL - Serial Clock i SDA - Serial Data. I2C komunikacija je u fast modu. Pinovi PA3-PA9 i PB0-1, PB4-5, PB8-9 i PB12 se redom postavljaju na *GPIO output,* jer će se oni koristiti kao izlazi za kontrolu sedmosegmentnih displeja. Takođe, pinovi PA10-12 se koriste kao izlazi za aktivaciju LE dioda kao indikatora. PA13 se koristi za detekciju hardverskog prekida i na njega se povezuje taster koji uključuje i isključuje ceo sistem. Za UART se postavljaju pinovi PA2 za Tx, a PA3 za Rx, gde se mod podešava kao asinhroni. TIM2 će biti upotrebljen za generisanje PWM signala za kontrolu brzine motora, gde će se koristiti kanal 2. Pošto se upotrebljava interni oscilator frekvencije 64 MHz, potrebno je preskalirati frekvenciju PWM signala na 10 kHz, što je optimalna frekvencija za DC motor. Da bi se postigla ta vrednost, treba podesiti vrednost prescalera na 64-1, dok u vrednost AR registra treba podesiti na 100. Početna vrednost ispune signala (*pulse*) je podešena na 25, ali ova vrednost nije ključna jer će se tokom izvršavanja programa menjati faktor ispune u realnom vremenu. Za očitavanje

temperaturnog senzora korišćen je AD konvertor na pinu PA0 podešen u kontinualnom (*continuous*) modu.



Slika 6. Iskorišćeni pinovi mikrokontrolera.

V. KOD MIKROKONTROLERA

Keil MDK je kompletno okruženje za razvoj softvera za niz ARM Cortex-M baziranih mikrokontrolerskih uređaja. U ovom projektu jedan deo koda odnosi se na deklaraciju promenljivih i inicijalizaciju tajmera. Drugi deo koda namenjen je temperaturnom senzoru i ostalim funkcijama na osnovu očitane temperature: pokretanju DC motora, uključivanje LE diode, ispisivanju vrednosti na OLED-u, UART-u , kao i na sedmosegmentnim displejima. Kod je prikazan na slikama 7-10. Na slici 7 dat je kod za pozivanje funkcija koje vrše inicijalizaciju periferijskih jedinica mikrokontrolera. Na slici 8 data je prekidna funkcija za očitavanje pritiska tastera za uključivanje sistema. Ako je sistem isključen isključuju se svi displeji i motor, a ako je uključen izvršava se kod sa slike 9, kojim se u zavisnosti od očitane temperature postavljaju stanja LED-ova.

```
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_ADC1_Init();
  MX_I2C1_Init();
  MX_USART2_UART_Init();
  MX_TIM2_Init();
  MX_TIM1_Init();
  /* USER CODE BEGIN 2 */
HAL_ADC_Start(&hadc1); // Inicijalizacija ADC-a
HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1); //Inicijalizacija PWM-a
SSD1306_Init();
  SSD1306_Fill(SSD1306_COLOR_BLACK);
  SSD1306_UpdateScreen();
  HAL_Delay(500);
  SSD1306_Fill(SSD1306_COLOR_WHITE);
  SSD1306_UpdateScreen();
  HAL_Delay(500);
```

Slika 7. Inicijalizacija ADC-a, PWM-a, UART-a i OLED-a.

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
  fleg=!fleg;
  /* Prevent unused argument(s) compilation warning */
  UNUSED(GPIO_Pin);
  /* NOTE: This function Should not be modified, when the callback is needed,
           the HAL_GPIO_EXTI_Callback could be implemented in the user file
   */
}
/* USER CODE END 4 */
```

Slika 8. Prekidna funkcija za taster.

```
  else
  {
  temp=adcPravaVrednost;
    if(temp>=0 && temp<17)
    {HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
        }

    if(temp>=17 && temp<=32)
    {HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
    }

    if(temp>32 && temp<=50)
```

Slika 9. Deo koda za postavljanje stanja LED-ova u zavisnosti od očitane temperature.

Na slici 10 prikazan je deo koda kojim se za određeni temperaturni opseg šalju informacije na OLED, na serijski terminal preko UART-a, i postavljaju odgovarajuće vrednosti PWM signala za kontrolu brzine okretanja motora. Na slici 11 prikazan je deo koda za prikaz vrednosti na sedmosegmentnom displeju.

```
  if(temp>=0 && temp<17)
  {
      if(temp == 0){SSD1306_Clear();}
  HAL_UART_Transmit(&huart2,"Kuler ne radi\r\n",14,10);
SSD1306_GotoXY(x,y);
       SSD1306_Puts("  Kuler ne radi   ",&Font_7x10, SSD1306_COLOR_WHITE);
       SSD1306_UpdateScreen();
     HAL_GPIO_WritePin(GPIOB,GPIO_PIN_14,GPIO_PIN_RESET);
     HAL_GPIO_WritePin(GPIOB,GPIO_PIN_11,GPIO_PIN_RESET);
htim2.Instance->CCR1=0; }
  if(temp>=17 && temp<=22)
  { HAL_UART_Transmit(&huart2,"Kuler radi na 20%\r\n",18,10);
SSD1306_GotoXY(x,y);
       SSD1306_Puts("Kuler radi na 20% ",&Font_7x10, SSD1306_COLOR_WHITE);
       SSD1306_UpdateScreen();
     HAL_GPIO_WritePin(GPIOB,GPIO_PIN_14,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_11,GPIO_PIN_RESET);
htim2.Instance->CCR2=51;  }
  if(temp>22 && temp<=32)
```

Slika 10. Deo koda koji se odnosi na slanje poruka preko UART-a, štampanje na OLED-u, kao i slanje odgovarajućeg PWM signala na kolo za kontrolu DC motora.

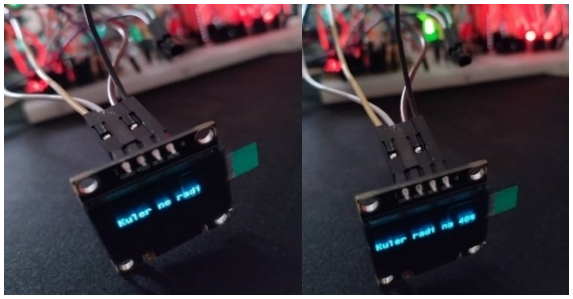
```
//56
if(temp==56) {
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_15, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);

HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
}
```

Slika 11. Kod za štampanje odgovarajuće vrednosti temperature preko dva sedmosegmentna displeja.

VI. PRAKTIČNA REALIZACIJA

Praktična realizacija ovog sistema prikazana je na slikama 12-14. U zavisnosti od temperature kuler će raditi sa određenim faktorom ispune (što je temperatura veća, kuler će raditi sa većim faktorom ispune). Temperature koje temperaturni senzor očitava su u rangu od 10°C do 80°C i prikazuju se na dva sedmosegmentna displeja. Pritiskom tastera korisnik određuje uključivanje i isključivanje kola. Kada je kolo uključeno, na terminalu se štampa poruka na koliko procenata kuler radi u zavisnosti od temperature računara (na primer: Kuler je isključen, Kuler radi na 20%, itd.).



Slika 12. Isključen sistem i rad kulera na 40% maksimalne brzine, sa uključenim zelenim LED-om.



Slika 13. Rad kulera na 60% maksimalne brzine, sa uključenim žutim LED-om, prikaz na OLED-u i serijskom monitoru.



Slika 14. Rad kulera na 100% brzine, sa uključenim crvenim LED-om, prikaz informacije na OLED-u i serijskom monitoru, gde je izmerena temperature prikazana na sedmosegmentnim displejima.

VII. ZAKLJUČAK

U ovom radu prikazano je projektovanje i emuliran je rad kulera za prenosni računar, korišćenjem DC motora i ARM mikrokontrolera. Ovaj projekat se može unaprediti na mnogo načina u zavisnosti od primene kulera. Iako je signalizacija urađena korišćenjem LED-ova, OLED-a, sedmosegmentih displejeva, moguće je dodati i zvučnu signalizaciju. Takođe, postoji mogućnost razvijanja mobilne aplikacije za podešavanje i praćenje rada kulera.

ZAHVALNICA

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

LITERATURA

[1] Tehnička specifikacija za Blue Pill. Dostupna na:https://pdf1.alldatasheet.com/datasheet-pdf/view/201590/STMICROELECTRONICS/STM32F103C8.html

[2] Tehnička dokumentacija za temperatirni senzor ktz 81-210Dostupnona: https://datasheetspdf.com/pdf-file/68970/NXP/KTY81-210/1

[3] Tehnička dokumentacija za SSD1306 kontroler. Dostupna na: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf

[4] Tehnička dokumentacija za L293D. Dostupno na: https://www.ti.com/lit/ds/symlink/l293.pdf

Detektor zvuka upotrebom modula MAX4466 i mikrokontrolera sa ARM arhitekturom

Nikola Stojiljković, Strahinja Dimitrijević

Sadržaj – U ovom radu opisana je realizacija sistema kojim se pomoću ARM mikrokontrolera i modula MAX4466 detektuje zvuk i na serijskom terminalu ispisuju odgovarajuće digitalne vrednosti. Iskorišćeno je Nucleo razvojno okruženje koje poseduje ARM mikrokontroler. Kao indikator granične vrednosti zvuka upotrebljena je crvena LE dioda, kada zvuk premaši graničnu vrednost prikazuje se odgovarajuća poruka na terminalu i uključuje LE dioda.

I. UVOD

U današnje vreme detekcija zvuka predstavlja jedan od najčešće merenih i detektovanih veličina. Prenos informacija se od davnina odvijao pomoću zvuka, a u inženjerskom smislu, detekcija prisustva zvučnih talasa predstavlja osnov za uspešnu komunikaciju.

Za detekciju zvuka koriste se dobro poznati mikrofoni koji se javljaju u nekoliko različitih vrsta. Kapacitivni, elektretni, MEM mikrofoni [1] i tome slično. U ovom projektu iskorišćen je mikrofonski modul MAX4466 koji u srebi sadrži mikrofon sa predpojačavačkim kolom.

Pored same detekcije, neophodno je često znati i kada je intenzitet zvuka veći od neke granične vrednosti (na primer kod mikseta), pa se često koristi svetlosna indikacija u vidu LE diode. Pored toga, postoji i tekstualna indikacija preko serijskog terminala kojom se potvrđuje da se signal sa mikrofona koji je analogni uspešno konvertovao u digitalni signal.

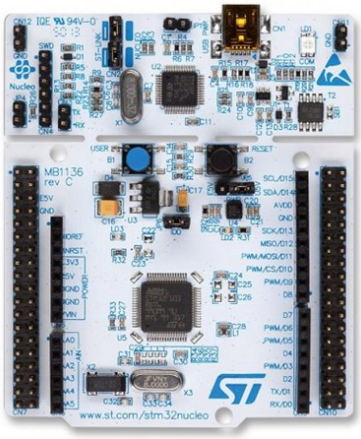
II. KOMPONENTE KORIŠĆENE U REALIZACIJI PROJEKTA

U nastavku su navedene sve komponente neophodne za realizaciju ovog projekta sa najvažnijim teorijskim objašnjenjima, kako bi preciznije objasnili sam princip rada. Korišćene komponente su: STM32 Nucleo-64 [2] razvojno okruženje, MAX4466 [3] mikrofonski modul, LE dioda.

A. STM32 Nucleo-64 razvojno okruženje

Razvojno okruženje STM32 Nucleo-64 (slika 1.) sa STM32F103RB mikrokontrolerom pripada familiji ARM mikrokontrolera. *Arm Cortex-M* je koncept korišćenja 32-bitnih RISC ARM *(Advanced RISC Machine)* arhitektura za kreiranje mikroprocesora i mikrokontrolera.

STM32 Nucleo-64 razvojno okruženje poseduje mogućnost povezivanja sa Arduino razvojnim okruženjem i STM *Morpho* razvojnim okruženjem za primenu u kompleksnim aplikacijama. Ima fleksibilno napajanje od 3,3 V, 5 V, 7 V-12 V. Poseduje dve korisničke LE diode i dva tastera kao i mini USB konektor. Konfiguriše se korišćenjem softverskog alata *STM32CubeMX* [4] dok se odgovarajući *firmware* piše u odgovarajućem programu, pri čemu je ovde upotrebljen *Keil uVision* [5].



Slika 1. STM32 Nucleo-64 razvojno okruženje.

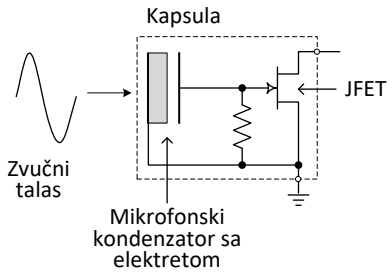
B. MAX4466 mikrofonski modul

Mikrofon je elektronski uređaj koji zvučne talase koji do njega dopiru pretvara u analogni električni signal. Postoji veliki broj tipova mikrofona koji se razlikuju po principu rada. Senzor zvuka koji koristimo u ovom radu (slika 2) sadrži takozvani kondenzatorski mikrofon sa elektretom. Elektret je dielektrični materijal sa permanentno ugrađenim statičkim naelektrisanjem. Postoji analogija sa permanentnim magnetom – kao što oko magneta postoji permanentno magnetno polje, tako i oko elektreta postoji permanentno električno polje. [6] Mikrofon se formira smeštanjem elektretske folije u prostor između dve elektrode, čime se formira struktura kondenzatora sa konstantnim naelektrisanjem. Varijacije vazdušnog pritiska (zvučni talasi) pomeraju jednu od elektroda napred-nazad, menjajući tako rastojanje između elektroda, a time i kapacitivnost ove strukture. Pošto je naelektrisanje konstantno, a prema poznatoj Jednačini 1:

N. Stojiljković, S. Dimitrijević, studenti master studija na studijskom programu Elektroninka i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stojiljkovicn@elfak.rs, strax.dimitrijevic@elfak.rs.

$$Q = CV\,, \qquad (1)$$

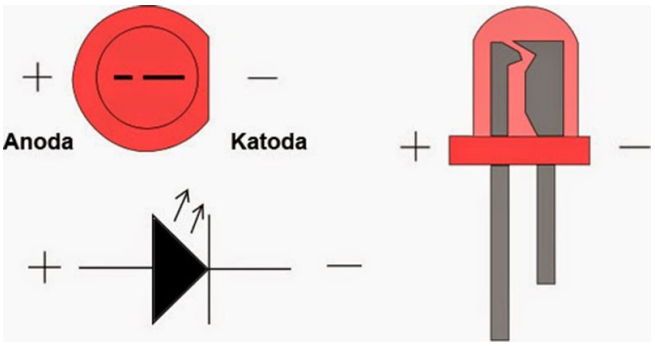
promena kapacitivnosti dovodi do promene napona na kondenzatoru. Na taj način, mikrofon deluje kao naponski izvor modulisan zvukom. Pošto su varijacije napona veoma male, one se moraju pojačati. Prvi stepen pojačanja je *FET* tranzistor (koji je zajedno sa mikrofonom ugrađen u istu kapsulu). Drugi stepen pojačanja je izveden na pločici senzora zvuka uz pomoć operacionog pojačavača. Modul senzora zvuka se povezuje sa mikrokontrolerom preko pina za analogni ulaz.



Slika 2. Mikrofonska kapsula.

C. LE dioda

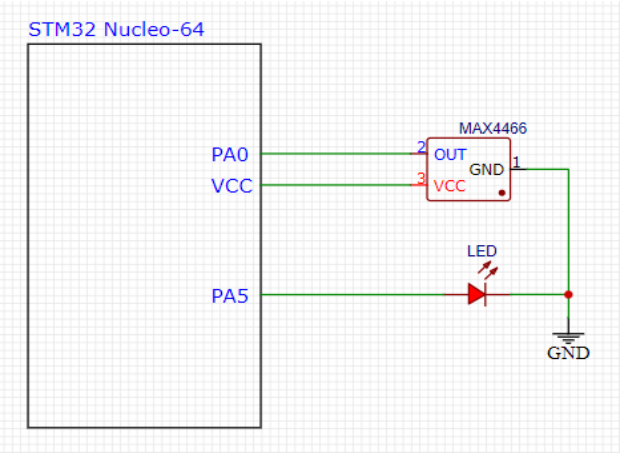
Direktno polarisane LE diode (slika 3.) će provoditi struju jer se veliki broj nosilaca naelektrisanja kreće ka kontaktima anode i katode. Elektroni iz provodne zone (sa višom energijom) se rekombinuju sa šupljinama iz valentne zone (sa nižom energijom). Procesom rekombinacije se oslobađa energija u vidu toplote (fononi) i/ili svetlosnog zračenja (fotoni). Ukoliko dominiraju fotoni javlja se efekat fotoluminiscencije – zračenje u vidu vidljive, IC ili UV svetlosti. Za izradu LE dioda se koriste poluprovodna jedinjenja sa direktnim energetskim procepom [7].



Slika 3. Izvodi i električni simbol LE diode.

III. Električna šema

U nastavku, na slici 4, je predstavljena električna šema po kojoj je realizovan ovaj projekat.



Slika 4. Električna šema u softverskom alatu EasyEDA [8].

IV. Princip rada

Modul MAX4466 povezan je preko pina *OUT* na interno ugrađen A/D konvertor razvojnog okruženja STM32 Nucleo-64. Detektor zvuka detektuje prisustvo zvuka u svom okruženju. Ulaz je senzor zvuka, a izlaz LE dioda koja svetli ako je izmereni nivo zvuka veći od granične vrednosti. Nivo zvuka se sempluje konstantno što je zadato prilikom podešavanja A/D konvertora biranjem countinous moda. Prisustvo zvuka je detektovano, odnosno LE dioda se pali, ukoliko je očitana digitalna vrednost odmerka veća od granične vrednosti.

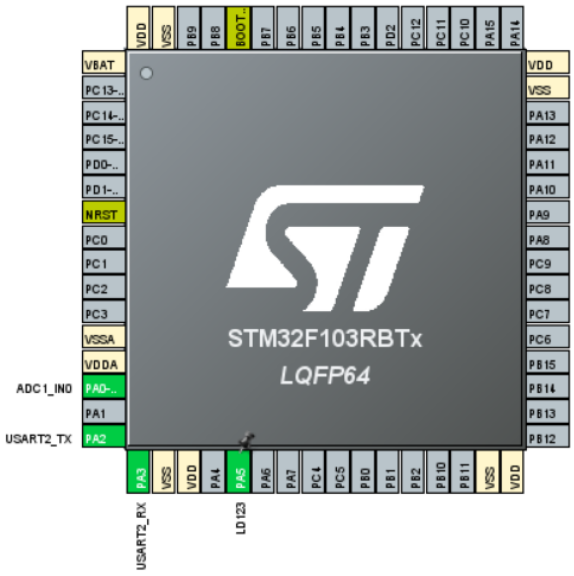
Pored ovih pinova imamo još jedan pin koji služi da se poveže odgovarajuća LE dioda zadužena za indikaciju granične vrednosti detektovane vrednosti zvuka.

Nivo zvuka očitavamo na terminalu koji je povezan pomoću UART-a.

Kako bi se izvršila konfiguracija pinova razvojnog okruženja STM32 *Nucleo-64* i podesilo odgovarajuće vršenje A/D konverzije iskorišćen je softverski alat *STM32CubeMX*. U nastavku, na slici 5. data su podešavanja tajmera za potrebe vršenja A/D konverzije, dok je na slici 6. prikazano podešavanje pinova.



Slika 5. Podešavanje A/D konvertora.



Slika 6. Konfiguracija pinova u STM32CubeMX

V. PROGRAMSKI KOD

Programski kod za potrebe ovog projekta napisan je u softverskom alatu *Keil uVision* i prosleđen mikrokontroleru. Za potrebe očitavanja vrednosti zvuka koristi se virtuelni terminal. Najpre su definisane dve promenljive u delu koda predviđenom za korisnika.

```
/* USER CODE BEGIN 0 */
uint32_t adcVrednost;
char bafer[40];
/* USER CODE END 0 */
```

Nakon toga potrebno je pokrenuti odgovarajući tajmer za vršenje A/D konverzije.

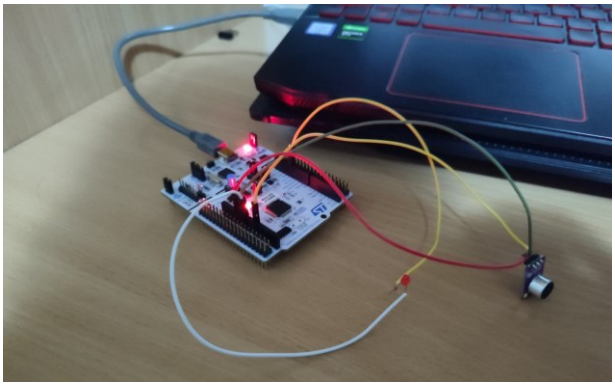
```
/* USER CODE BEGIN 2 */
HAL_ADC_Start(&hadc1);
/* USER CODE END 2 */
```

U okviru glavne programske petlje nalazi se programski kod u nastavku.

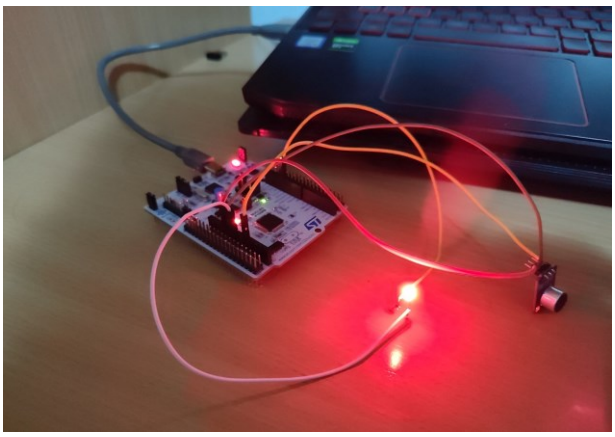
```
/* USER CODE BEGIN 3 */
adcVrednost=HAL_ADC_GetValue(&hadc1);
        if(adcVrednost < 4000)
                {sprintf(bafer,"MIC = %d \r\n",adcVrednost);
HAL_UART_Transmit(&huart2,(uint8_t*)bafer,strlen(bafe
r),10);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
                }
        else
        {
HAL_UART_Transmit(&huart2, "prekoracenje\r\n", strlen
("prekoracenje\r\n"), 10);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
        }
```

VI. PRAKTIČNA REALIZACIJA

Nakon napisanog progamskog koda za mikrokontroler izvršeno je praktično povezivanje (slika 7 i slika 8) svih komponenata kako bi se proverila funkcionalnost samog projekta. Konekcija između okruženja STM32 Nucleo-64 i ručunara obezbeđuje se putem USB- a, kako bi se obezbedilo odgovarajuće napajanje i prosledio programski kod.



Slika 7. Praktična realizacija projekta.



Slika 8. Praktična realizacija projekta.

Kako bi se proverila funkcionalnost povezanog sistema pored LE diode koja se uključuje kada modul detektuje određeni zvuk kosrišćen je i virtuelni terminal. Kada se proizvede zvuk određene jačine, a u opsegu od 20 Hz do 20 kHz uključuje se crvena LE diode i na virtuelnom terminalu prikazuje proračunata ADC vredsnost. Ukoliko je zvuk van ovog opsega na virtuelnom terminalu se prikazuje poruka „prekoračenje“.

Detektor zvuka prikazan u ovom projektu pokriva opseg od 20 Hz do 20 kHz, dakle ograničen je u opsegu koji čuje ljudsko uvo, što može predstavljati nedostatak ovog sistema. Ako je potrebno detektovati zvuk koji se

nalazi u osegu ultrazvuka potrebna je upotreba nekog drugog modula.

Prilikom rada sa ovim modulom nisu uočeni nedostaci koji su predstavljali prepreku za realizaciju celokupnog sistema.

VII. Zaključak

U ovom radu demonstriran je jedan od načina detekcije zvuka kao i detekcija prekoračenja intenziteta zvuka. Konfiguracija mikrokontrolera, uključujući podešavanje pinova i izrada programskog koda vrlo su jednostavni i izmene se takođe jednostavno unose za potrebe drugačijih zahteva.

Princip rada prikazan u ovom projektu samo je prvi korak na kome se mogu nadograditi mnoge druge funkcije kao što je određivanje tačne frekvencije tona koji detektuje mikrofon. Nakon detekcije tona moguće je, u zavisnosti od kreativnosti i potrebe dizajna, pokretati određene aktuatore ili kontrolisati rad nekog drugog uređaja ili kola.

Međutim, sve ove modifikacije izlaze iz okvira trenutnih mogućnosti i zahteva ovog projekta.

Zahvalnica

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

Posebna zahvalnost asistentu Milošu Marjanoviću koji nam je obezbedio sav potreban materijal za izradu ovog projekta kao i sugestije koje nam je uputio tokom izrade istog.

Literatura

[1] V. Paunović, Mikroelektromehanički sistemi, Elektronski fakultet Niš, Dostupno na: https://mikro.elfak.ni.ac.rs/wp-content/uploads/MEMS-3.pdf
[2] Tehnička specifikacija za STM32 Nucleo-64. Dostupno na: https://www.st.com/en/evaluation-tools/nucleo-f401re.html
[3] Tehnička specifikacija za modul MAX4466. Dostupno na: https://pdf1.alldatasheet.com/datasheetpdf/view/73367/MAXIM/MAX4466.html
[4] Softverski alat STM32CubeMX. Dostupno na: https://www.st.com/en/development-tools/stm32cubemx.html
[5] Softverski alat Keil uVision. Dostupno na: https://www2.keil.com/mdk5/uvision/
[6] M. Marjanović, DSP mikrofon, Elektronski fakultet Niš.
[7] A. Prijić, Poluprovodničke komponente, Elektronski fakultet Niš, 2021. Dostupno na: http://mikro.elfak.ni.ac.rs/wpcontent/uploads/Poluprovodničke-komponente-diode-2021-1.pdf
[8] Softverski alat EasyEDA. Dostupno na: https://easyeda.com

Kontrola rada ventilatora korišćenjem ARM mikrokontrolera

Nikola Branković, Stefan Janković, Nikola Janković

Sadržaj – U radu je realizovan sistem koji uz pomoć BluePill razvojnog okruženja kontroliše rad ventilatora u zavisnosti od temperature u prostoriji i temperature praga koju korisnik podešava. Kada temperatura pređe određenu vrednost, uključiće se ventilator i radiće sve dok temperatura ne opadne. Na TFT displeju će se ispisivati trenutna i podešena temperatura kao i stanje ventilatora, da li hladi ili ne, dok će dve LED diode da signaliraju da li je došlo do prekoračenja temperature, crvena, ili je temperatura prostorije ispod podešenog praga, plava.

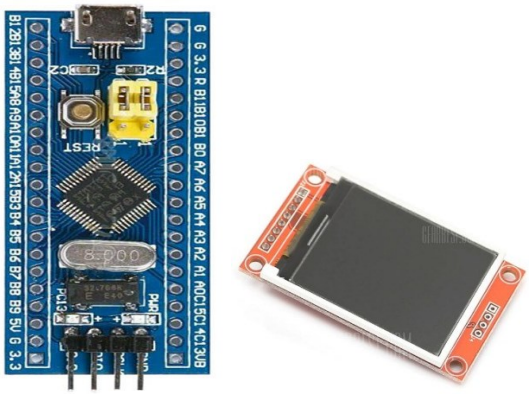
I. Uvod

Ventilatori su električni uređaji koji se koriste za cirkulaciju vazduha u prostorijama. Oni pomažu u rashlađivanju, provetravanju i poboljšanju kvaliteta vazduha u zatvorenim prostorijama. Ventilatori se sastoje od rotirajućih lopatica koje guraju vazduh ispred sebe. Kada lopatice rotiraju, one stvaraju razliku pritisaka, što rezultira strujanjem vezduha. Jedna od glavnih prednosti korišćenja ventilatora je što su energetski efikasniji u poređenju sa klima uređajima. Ventilatori troše manje energije jer ne hlade vezduh, već samo stvaraju protok vazduha koji pruža osećaj hlađenja. Uz standardne ventilatore, postoji i širok izbor modernih ventilatora. To uključuje daljinske upravljače, tajmerske funkcije za automatsko isključivanje ventilatora nakon određenog vremena i različite načine rada kao što je, na primer, noćni tiši režim rada.

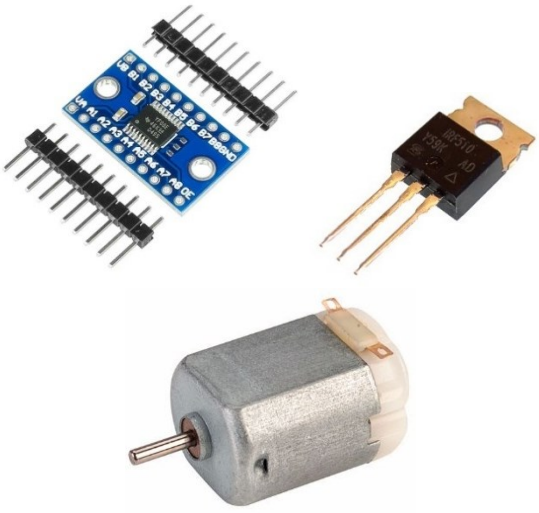
U ovom radu je opisan princip uključivanja/isključivanja ventilatora preko tranzistora snage IRF510, koji se kontroliše mikrokontrolerom.

II. Korišćene komponente za realizaciju projekta

BluePill STM32F103C8T6 je razvojno okruženje koje koristi mikrokontroler serije STM32, baziran na ARM konceptu [1]. BluePill okruženje je prikazano na slici 1. TFT displej ST7735 [2] prikazan na slici 1, zasnovan je na tehnologiji tankog filma. Ovaj model ima rezoluciju 128x160 i dijagonalu displeja 1.8 inča. Ima mogućnost prikaza 18 različitih boja. Komunikacija se vrši preko SPI protokola. TXS1080E je bidirekcioni naponski translator [3], kod kojeg pinovi A služe za naponske nivoe od 1.4V do 3.6V, a pinovi B za naponske nivoe od 1.65V do 5.5V, prikazan na slici 2. Takođe, treba napomenuti da iako je TXS1080E bidirekcioni translator, pinovi A i B ne mogu da se koriste kao ulazi ili izlazi istovremeno. IRF510 je MOSFET tranzistor snage [4], zapakovan u TO220 kućište, prikazan na slici 2. DC motor je vrsta elektro-motora koji pretvaraju DC signal u mehaničku energiju. Jedan primer DC motora je prikazan na slici 2.



Slika 1. BluePill razvojno okruženje i ST7735 TFT displej.



Slika 2. TXS1080E translator napona, IRF510 tranzistor u T0220 kućištu i DC motor.

N. Branković, S. Janković, N. Janković su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikolabrankovic@elfak.rs

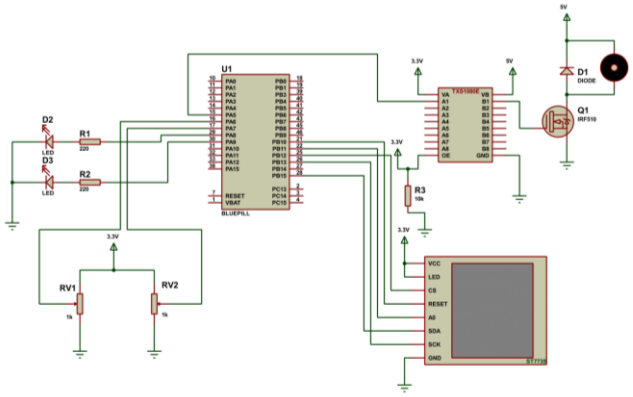
III. Opis šeme sistema

Električna šema je prikazana na slici 3. U centru sistema je BluePill, koji vrši kontrolu celog sistema. DC motor se kotroliše preko tranzitora, a sam tranzistor se kontroliše preko BluePill-a uz pomoć translatora napona TXS. Pin zadužen za uključivanje i isključivanje motora je A5 i on je povezan na A0 pin na translatoru. Pin B0 na translatoru je povezan na gejt tranzistora. Drejn tranzistora je povezan na motor, a sors na masu. Drugi kraj motora je povezan na 5V, koji se dovode eksterno.

TFT displej je povezan na BluePill preko SPI komunikacinog protokola. Na displeju će se ispisivati trenutna temperatura u prostoriji, kao i podešeni temperaturni prag. U softveru CubeMX je uključen SPI2 na BluePill-u. Pinovi obezbeđeni za SPI2 su B13 (*Clock* pin), B14 (MISO – *Master Input Slave Output*), B15 (MOSI – *Master Output Slave Input*). Ostali pinovi sa displeja su *ChipSelect* povezan na B12, *Reset* na B10 i A0 na B11.

LED diode služe za indikaciju prelaza temperature iznad dozvoljenog praga. Plava LED dioda označava da temperatura prostorije nije prešla dozvoljeni prag i ona je preko otpornika od 220 Ω povezana na A8, dok crvena LED dioda označava da je došlo do prelaza temperature preko dozvoljene i ona je povezana preko otpornika od 220 Ω na A9.

Na pinove A6 i A7 su povezani potenciometri koji služe za emulaciju temperaturnog senzora, jedan za podešavanje praga, a drugi za temperaturu sobe. Na translatoru TXS na VA je dovedeno 3.3V sa BluePill-a, a na VB je dovedeno 5V sa eksternog napajanja.

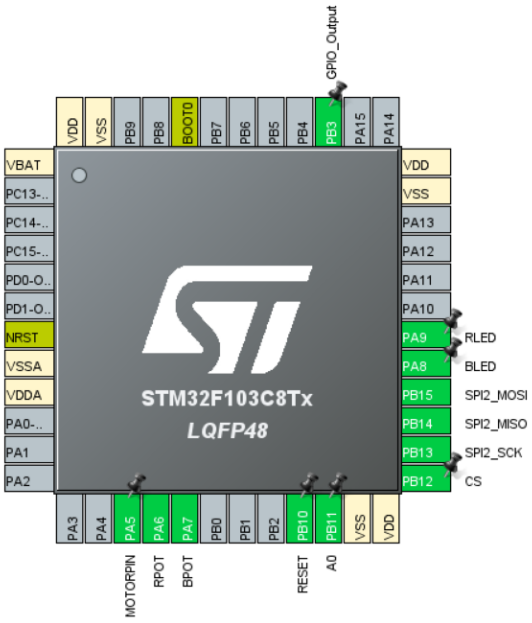


Slika 3. Električna šema.

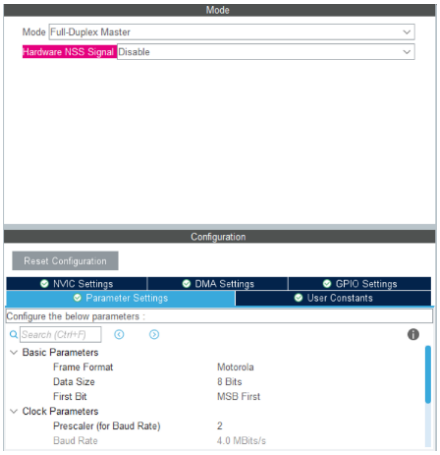
IV. Podešavanja u CubeMx

CubeMX je softver koji nam omogućava lako konfigurisanje STM32 mikrokontrolera, podešavanje njegovih pinova, uključivanje različitih periferija i generisanje kompletnog početnog koda. Iskorišćeni pinovi mikrokontrolera prikazani su na slici 4.

Za komunikaciju sa TFT displejom nam je neophodna SPI komunikacija. STM32F103C8T6 ima mogućnost za povezivanje dva uređaja preko SPI komunikacije. U kategoriji Connectivity je aktiviran SPI2. Izabran je SPI2 zbog rasporeda pinova koji je odgovarao prilikom praktične realizacije. Kroz CubeMX vrši se podešavanje parametara SPI komunikacije, tj. da li će se vršiti Full-Duplex ili Half-Duplex komunikacija, kao i podešavanje brzine komunikacije, što se može videti na slici 5. Pinovi koje koristi SPI2 su B13 za *Clock*, B14 za *MISO* i B15 za *MOSI*. Za *ChipSelect*, *Reset* i *A0* na TFT displeju izabrani su pinovi B10, B11 i B12, respektivno, i podešeni su kao izlazni *GPIO Output*.



Slika 4. Kompletno podešavanje pinova u CubeMX.



Slika 5. Podešavanje SPI u CubeMX-u.

Sledeći korak je uključivanje A/D konvertora. BluePill ima dva A/D konvertora, što je jedna od glavnih razlika i prednosti u odnosu na Arduino razvojna okruženja. U kategoriji *Analog* su uključena oba A/D konvertora. A/D konvertori su potrebni za očitavanje vrednosti sa potenciometara. Prilikom uključivanja A/D konvertora, biraju se pinovi koji će biti zaduženi za

konverziju, a za to su izabrani pinovi A6 za ADC1 i A7 za ADC2.

Na kraju su podešeni pinovi za LED diode, i za to su izabrani pinovi A8 i A9 i podešeni su kao *GPIO Output*, i pin zadužen za uključivanje i isključivanje DC motora, za to je izabran pin A5 i podešen je kao *GPIO Output*.

V. PROGRAMSKI KOD

Keil microVision 5 je okruženje namenjeno za razvoj softvera za mikrokontrolere bazirane na ARM tehnologiji. Prvi deo koda se odnosi na prikupljanje vrednosti sa A/D konvertora i njihovo skaliranje, kao i ispisivanje tih vrednosti na TFT displej, dok je drugi deo zadužen za kontrolu DC motora pomoću poređenja dobijenih vrednosti. Kod za prikupljanje podataka sa potenciometara se može videti na slici 6, dok se deo koda za ispisivanje na TFT i kontrolu DC motora može videti na slici 7.

```
//glavni kod - pocetak
HAL_ADC_Start(&hadc1);
HAL_ADC_Start(&hadc2);
if(HAL_ADC_PollForConversion(&hadc1,5) == HAL_OK){

pod=HAL_ADC_GetValue(&hadc1);
}

if(HAL_ADC_PollForConversion(&hadc2,5) == HAL_OK)
{
sim=HAL_ADC_GetValue(&hadc2);
}

HAL_Delay(50);

podsc=scale(pod);
simsc=scale(sim);

HAL_Delay(50);

sprintf(ZaIspis_sim, " %d ", simsc);
sprintf(ZaIspis_pod, " %d ", podsc);
```

Slika 6. Inicijalizacija A/D konvertora i prikupljanje vrednosti sa njih i skaliranje tih vrednosti u temperaturu pozivanjem na funkciju scale.

```
ST7735_WriteString(5,45,"Trenutna", Font_7x10, ST7735_WHITE, ST7735_BLACK); //simulirano sim
ST7735_WriteString(5,55,"temperatura:",Font_7x10, ST7735_WHITE, ST7735_BLACK);
ST7735_WriteString(5,75,"Podesena", Font_7x10, ST7735_WHITE, ST7735_BLACK); //podesena pod
ST7735_WriteString(5,85,"temperatura:",Font_7x10, ST7735_WHITE, ST7735_BLACK);
ST7735_WriteString(35,65,ZaIspis_sim, Font_7x10, ST7735_WHITE, ST7735_BLACK);
ST7735_WriteString(35,95,ZaIspis_pod, Font_7x10, ST7735_WHITE, ST7735_BLACK);
ST7735_WriteString(60,65,"oC", Font_7x10, ST7735_WHITE, ST7735_BLACK);
ST7735_WriteString(60,95,"oC", Font_7x10, ST7735_WHITE, ST7735_BLACK);

if(simsc>podsc){
  flag=false;
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
  motor(flag);
  ST7735_WriteString(5,15, "Ventilator:", Font_7x10, ST7735_WHITE, ST7735_BLACK);
  ST7735_WriteString(5,25,"Ne hladi", Font_7x10, ST7735_BLACK, ST7735_BLACK);
  ST7735_WriteString(5,25,"Hladi", Font_7x10, ST7735_WHITE, ST7735_BLACK);
  HAL_Delay(50);
}
else if(simsc<podsc){
  ST7735_WriteString(5,15, "Ventilator:", Font_7x10, ST7735_WHITE, ST7735_BLACK);
  ST7735_WriteString(5,25,"Ne hladi", Font_7x10, ST7735_WHITE, ST7735_BLACK);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
      HAL_Delay(50);
  motor(flag);
  flag=true;
}
HAL_Delay(100);
//glavni kod - kraj
```

Slika 7. Ispisivanje na TFT displej i kontrola rada DC motora.

Takođe su deklarisane dve funkcije, motor() i scale(), prikazane na slici 8. Funkcija motor() je zadužena za podešavanje pina A5 na logičku nulu ili jedinicu u zavisnosti od stanja flega. Funkcija scale() služi da vrednost očitanu sa pinova za A/D prevede u temperaturu.

```
void motor(bool flag){

  if (flag==false){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
  }
  else {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
  }

}

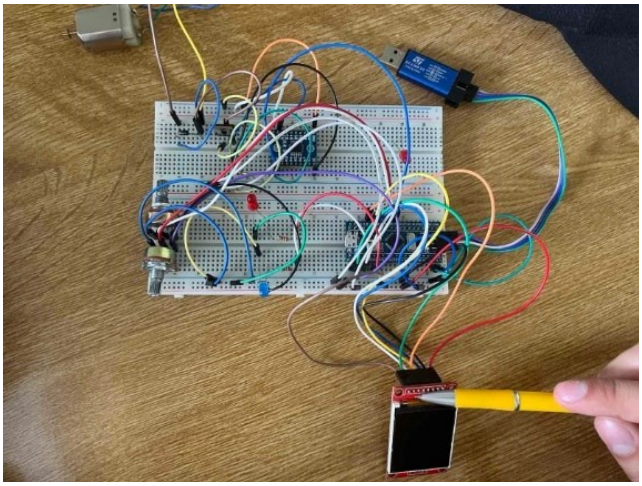
int scale(uint32_t ulaz){
  int x;
  return x=(12*ulaz)/1023;

}
```

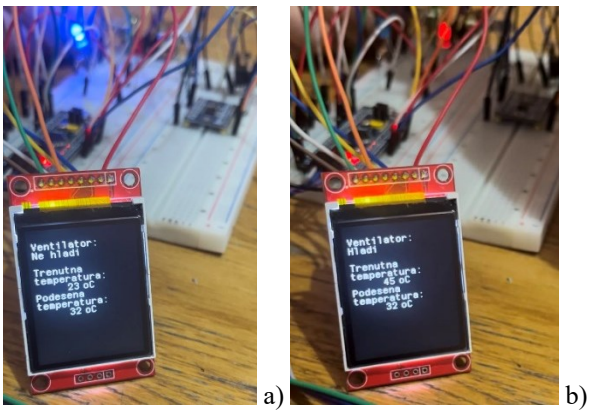
Slika 8. Deklaracija funkcija za kontrolu DC motora i skaliranje vrednosti sa potenciometara u temperature.

VI. PRAKTIČNA REALIZACIJA

Praktična realizacija ovog sistema na protoploči je prikazana na slici 9. Dok je trenutna temperatura niža od podešene temperature, motor će biti isključen, na displeju se prikazuje poruka „Ne hladi“ i svetli plavi LED (slika 10a). Ukoliko je temperatura prostorije veća od temperature praga, uključiće se crvena LED dioda i uključiće se motor (slika 10b). Motor će raditi dok temperatura ne opadne ispod zadate. Na displeju se prikazuje poruka „Hladi“.



Slika 9. Praktična realizacija - prikaz celog sistema.



Slika 10. Prikaz različitih scenarija rada ventilatora: kada je izmerena temperature a) niža, b) viša od podešene vrednosti.

VII. Zaključak

Ovaj projekat može se unaprediti na mnogo načina, a dalja unapređenja ili izmene će zavisiti od same primene sistema. Umesto temperaturnog senzora se može ubaciti gasni senzor uz pomoć kojeg bi se uključivala ili isključivala ventilacija prostorije. Takođe mogao bi da se koristi i drugi tip displeja, a na njemu bi mogla da se prikazuje osim trenutnih temperatura i stanja motora, kao i njegova brzina. Sama brzina motora bi se vršila potenciometrom, a sve to bi se ispisivalo na ekranu.

Zahvalnica

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

Literatura

[1] BluePill-STM32F103C8T6: https://components101.com/microcontrollers/stm32f103c8t8-blue-pill-development-board
[2] ST7735 TFT Displej: https://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf
[3] TXC1080E: https://www.ti.com/lit/ds/symlink/txs0108e.pdf?ts=1686387504824
[4] IRF510: https://www.vishay.com/docs/91015/irf510.pdf

Projektovanje korisničkog interfejsa aparata za kafu zasnovanog na ARM mikrokontroleru

Milica Đorđević, Anđela Ružić, Sara Savić

Sadržaj - U ovom radu prikazano je projektovanje sistema koji simulira rad aparata za kafu, korišćenjem DC motora, TFT displeja i ARM mikrokontrolera. Odabir napitka i količine šećera vrši se pomoću tri tastera, a na displeju se prikazuje odgovarajući tekst. Pored predloženog korisničkog interfejsa aparata za kafu, korišćenjem MOS tranzistora snage i DC motora simulira se rad ovakvog uređaja.

I. Uvod

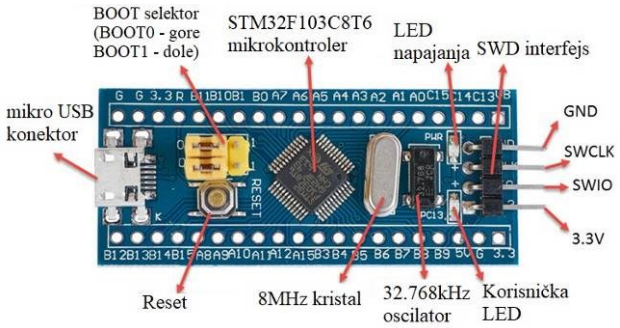
Aparat za kafu je uređaj koji se nalazi u svakom domu i služi za pripremu odabrane kafe. Od prvih jednostavnih metoda pripreme do sofisticiranih modernih aparata prošlo je mnogo vremena [1]. Priprema napitka od kafe počela je u 15. veku, a prvi filteri za kafu od tkanine i papira pojavili su se u 18. veku. Preteča modernog aparata za kafu bila je *espresso* mašina koju je izumeo Italijan, Angelo Moriondo, 1884. godine. Taj aparat je koristio paru i pritisak za brzu pripremu kafe, međutim mašina je bila velika i teška, pa nije bila praktična za kućnu upotrebu. Godine 1901. ovu mašinu unapredio je *Luigi Bezzera* dodavanjem funkcionalnosti kao što su kontrola temperature i pritiska. Danas postoji mnogo brendova i modela aparata za kafu, od jednostavnih kućnih aparata do komercijalnih mašina koje koriste vrhunsku tehnologiju za pripremu kafe.

U ovom radu biće opisan sistem koji simulira rad aparata za kafu zasnovan na ARM mikrokontroleru. Aparat za kafu osmišljen je tako da se nakon poruke na displeju, pomoću jednog taster bira željeni napitak, pri čemu su u ponudi 1. Domaća kafa, 2. Espresso i 3. Topla čokolada. Nakon odabira napitka, pritiskom na drugi taster, koji u ovom slučaju predstavlja potvrdu da je željeni napitak odabran, sledi biranje količine šećera. Količina šećera predstavljena je brojem kašičica, gde je moguće birati između 0, 1 i 2. Nakon potvrde počinje priprema napitka. Na TFT displeju se uz poruku "*Loading*…" puni progres bar, takođe, uključuje se crvena LE dioda koja označava da je napitak u pripremi, i pokreće se DC motor. Nakon određenog vremena, isključuje se dioda i zaustavlja se rad DC motora, a na displeju se ispisuje poruka "Preuzmite napitak", to je znak da je proces završen.

II. Komponente korišćene za izradu projekta

A. STM32F103 Blue Pill

Blue Pill [2] je razvojno okruženje koje je korišćeno za realizaciju ovog sistema, prikazano je na slici 1. U centru se nalazi STM mikrokontroler STM32F103C8T6 [3] zasnovan na ARM Cortex-M RISC arhitekturi. Radi na frekvenciji od 72 MHz, ima ugrađenu memoriju velike brzine, kao i veliki broj ulazno/izlaznih pinova i periferijskih jedinica. Poseduje dva analogno-digitalna konvertora, tri tajmera opšte namene plus jedan tajmer za generisanje PWM signala, takođe sadrži standardne i napredne komunikacione interfejse i to: dva I^2C i SPI interfejsa, tri USART interfejsa, USB i CAN. Dužina reči je 32 bita, a ima i dovoljno memorije za izvršenje operacija u realnom vremenu.



Slika 1. *Blue Pill* razvojno okruženje.

B. ST7735 TFT displej

TFT (*Thin Film Transistor*) displeji predstavljaju podskup LCD (*Liquid Crystal Display*). TFT ima rezoluciju 128 x 160 piksela, može prikazivati do 18 boja i koristi SPI protokol za komunikaciju [4]. Na slici 2. prikazan je izgled prednje i zadnje strane sa slotom za SD karticu.



Slika 2. Izgled prednje i zadnje strane TFT displeja.

C. MOS tranzistor snage IRF510

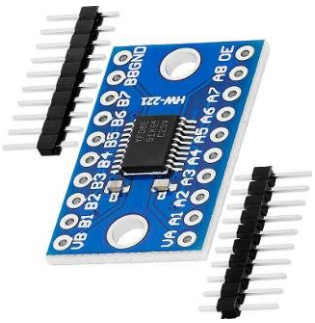
MOS tranzistori snage spadaju u snažne komponente koje imaju visoku ulaznu impedansu, kontrolišu se

M. Đorđević, A. Ružić, S. Savić, studenti četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milicadjordjevic@elfak.rs, andjela.r@elfak.rs, sarassavic@elfak.rs.

naponom pa je upravljanje jednostavno. MOS tranzistor snage je unipolarna komponenta, što znači da ima veliku brzinu prekidanja, pogodan je za prekidačke izvore napajanja i koristi se u kolima za upravljanje radom elektromotora. U upotrebi su uglavnom NMOS zbog veće pokretljivosti elektrona u odnosu na šupljine. U ovom radu korišćen je MOS tranzistora IRF510 [5].

D. TXS0108E translator napona

TXS0108E [6] je 8-bit bidirekcioni translator napona koji na pinove A prima/šalje napone do 3.3 V, a na pinove B šalje/prima napone do 5 V, prikazan je na slici 3.



Slika 3. Translator napona TXS0108E.

E. DC motor

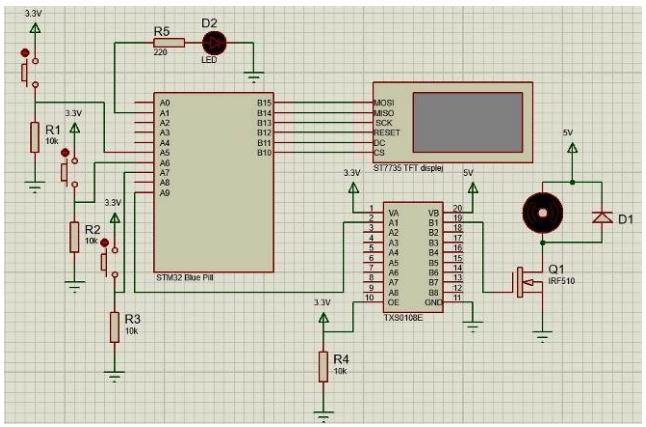
DC motor je vrsta električnih motora koji pretvara električnu energiju jednosmerne struje (DC) u mehaničku energiju. DC motori imaju dva izvoda - po jedan za pozitivni i negativni napon napajanja. Motor se pokreće primenom jednosmernog napona, a zaustavlja isključivanjem napona.

III. POVEZIVANJE ŠEME SISTEMA

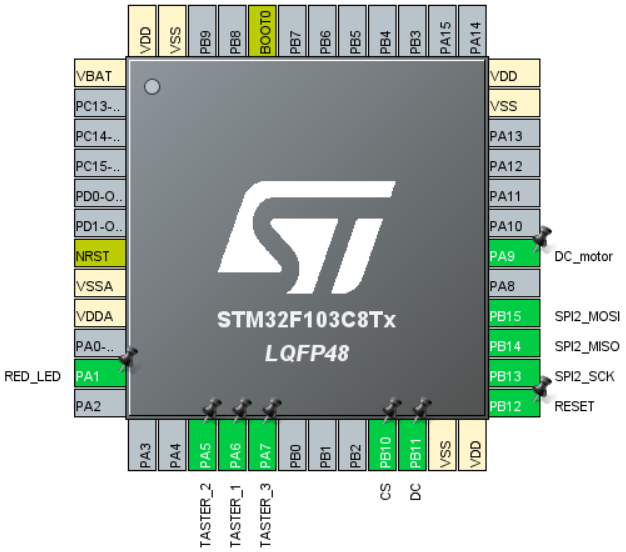
U centralnom delu ovog sistema nalazi se razvojno okruženje *Blue Pill* čiji su pinovi mikrokontrolera iskorišćeni za kontrolu sistema. Zatim, jedan deo sistema predstavlja kolo za kontrolu rada DC motora i čini ga sam motor, drajver, MOS tranzistor snage i napajanje. Pored toga, TFT displej povezan je preko 8 svojih pinova na pinove mikrokontrolera, napajanje i masu. Crvena LE dioda vezana je preko otpornika (R=220 Ω) za izlazni pin *Blue Pill*-a na strani anode, dok je katoda direktno vezana na masu. Tasteri su vezani u *pull-down* konfiguraciji, tako da kada taster nije pritisnut ulazni pin očitava napon logičke nule, a kada je taster pritisnut očitava se napon logičke jedinice. Kompletna električna šema prikazana je na slici 4.

IV. PODEŠAVANJA U *CUBEMX*

Podešavanje pinova u softverskom alatu *CubeMX* prikazano je na slici 5. pri čemu su korišćeni pinovi označeni zelenom bojom i odgovarajućim labelama. TFT displej kontroliše se SPI komunikacijom za koju su definisani sledeći pinovi: B15 kao MOSI (*Master Output Slave Input)* pin, B13 taktni signal, B12 reset pin, B10 *Chip Select* pin i B11 je DC pin. Tri tastera povezana su jednim krajem preko otpronika na masu i na ulazne pinove *Blue Pill*-a A5, A6 i A7, a drugim krajem na napon od 3.3 V. Za LE diodu rezervisan je izlazni pin A1. Povezivanje DC motora izvedeno je korišćenjem kola TXS0108E i MOS tranzistora snage, tako što je na pinu za dozvolu rada kola TXS0108E povezan otpornik u *pull-down* konfiguraciji, a na pinove za napone od 3.3 V i 5 V dovedeni su naponi sa odgovarajućih napajanja, dok su pinovi A1 i B1 kola TXS0108E zaduženi za prenos signala za uključivanje/isključivanje DC motora, koji se na drajver dovodi sa izlaznog pina A9. Alat *CubeMX* iskorišćen je za generisanje kostura koda u skladu sa podešavanjima za GPIO i SPI, po proceduri opisanoj u [7].



Slika 4. Električna šema realizovanog prototipa.



Slika 5. Podešavanje pinova u softverskom alatu CubeMX.

V. KOD U *KEIL*-U

Keil MDK je okruženje za razvoj softvera za niz mikrokontrolerskih uređaja baziranih na *ARM Cortex-M*. Prvi korak u pisanju koda jeste povezivanje programa sa dodatnim bibliotekama potrebnim za rad sa TFT displejem. Nakon njihovog dodavanja potrebno ih je, na početku, uključiti u kod naredbom *#include*. Sledeći deo posvećen je deklarisanju privatnih promenljivih koje će se koristiti u kodu, u ovom kodu to su promenljive *flag1*, *flag2* i *flag3*, 8-bitni integer-i. Zatim se prelazi na deo koda koji se izvršava samo jednom na početku programa, čiji je deo prikazan za slici 6. U ovom delu, pinovi A1 i A9 se inicijalno postavljaju na vrednost logičke nule, inicijalizuje se rad displeja i linijama koda u nastavku prikazana je funkcija iz biblioteke drajvera za ispisivanje poruke na displej *ST7735_WriteString*. Parametri ove funkcije su: koordinate tačke od koje će se tekst ispisivati (x,y), tekst, font, boja teksta i boja pozadine.

```
/* USER CODE BEGIN 2 */
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
ST7735_Init();
ST7735_FillScreen(ST7735_MAGENTA);
HAL_Delay(500);
ST7735_WriteString(30, 30, "APARAT", Font_11x18, ST7735_WHITE, ST7735_MAGENTA);
HAL_Delay(200);
ST7735_WriteString(52, 60, "ZA", Font_11x18, ST7735_WHITE, ST7735_MAGENTA);
HAL_Delay(200);
ST7735_WriteString(41, 90, "KAFU", Font_11x18, ST7735_WHITE, ST7735_MAGENTA);
HAL_Delay(3000);
ST7735_FillScreen(ST7735_MAGENTA);
```

Slika 6. Deo koda koji se izvršava samo jednom na početku.

U glavnom delu programa, beskonačnoj petlji, ispituju se stanja odgovarajućih tastera i *flag*-ova i u zavisnosti od njih na displej se prosleđuju određene poruke i vrši se odabir napitka i količine šećera. Na slici 7. je deo koda beskonačne petlje koji prikazuje upotrebu tastera za potvrdu (trećeg tastera), nakon što je odabran željeni napitak i količina šećera, počinje "priprema" napitka, a indikatori za to su, u ovom sistemu, crvena LE dioda i DC motor koji se aktiviraju kada se na displeju pojavi poruka "*Loading…*" postavljanjem pinova A1 i A9 na napon logičke jedinice.

```
if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7)==1 && flag2==2){
  ST7735_FillScreen(ST7735_MAGENTA);
  ST7735_WriteString(30, 30, "OK", Font_11x18, ST7735_WHITE, ST7735_MAGENTA);
  HAL_Delay(1500);
  ST7735_FillScreen(ST7735_MAGENTA);
  ST7735_WriteString(20, 20, "Loading...", Font_11x18, ST7735_BLACK, ST7735_MAGENTA);
  HAL_Delay(300);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 1);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
```

Slika 7. Deo koda beskonačne petlje za početak "pripreme" napitka.

Na slici 8. prikazana je funkcija za crtanje pravougaonika, koja je takođe deo biblioteke drajvera. Funkcija *ST7735_FillRectangle* za argumente uzima: koordinate x i y početne pozicije, širinu i visinu pravougaonika i boju. Ovom fukcijom vrši se iscrtavanje progres bara pravougaonik po pravougaonik čime se simulira vreme potrebno za pripremu napitka. Sve dok se progres bar puni dioda i motor ostaju uključeni, a kada se proces završi, oni se isključuju, čime se signalizira da je napitak spreman.

```
ST7735_FillRectangle(10, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(20, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(30, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(40, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(50, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(60, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(70, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(80, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(90, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(100, 40, 5, 10, ST7735_BLACK);
HAL_Delay(1000);
ST7735_FillRectangle(110, 40, 5, 10, ST7735_BLACK);
```

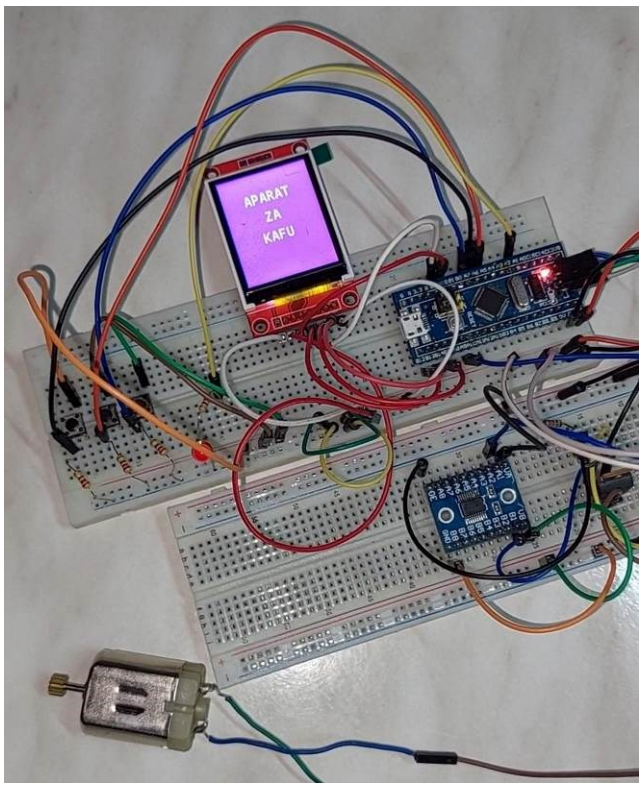
Slika 8. Kreiranje progres bara.

VI. PRAKTIČNA REALIZACIJA

Praktična realizacija sistema prikazana je na slikama 9 i 10. Proces započinje učitavanjem koda u Blue Pill. Na početku programa, na displeju se ispisuje pozdravna poruka. Zatim se od korisnika traži odabir napitka pritiskom na prvi taster. Bira se jedan od tri napitka, tako što se svakim pritiskom tastera selektuje sledeći i sve tako u krug. Nakon selektovanja željenog napitka pritiskom trećeg tastera, tj. tastera za potvrdu, ispisuje se poruka da je napitak izabran, nakon čega se sada od korisnika zahteva da izabere količinu šećera. Količina šećera bira se kao broj kašičica šećera, pri čemu to mogu biti 0, 1 i 2. Istom metodom, ali pritiskom na drugi taster, naizmenično se selektuje jedna od ove tri opcije. Ponovnim pritiskom na taster za potvrdu započinje "priprema" napitka. Uključuje se DC motor i LE dioda. Na displeju se, uz poruku učitavanja, popunjava progres bar. Nakon isteka vremena za pripremu, indikatori se isključuju, a na displeju se ispisuje poruka "Preuzmite napitak", čime se program kompletira.

VII. ZAKLJUČAK

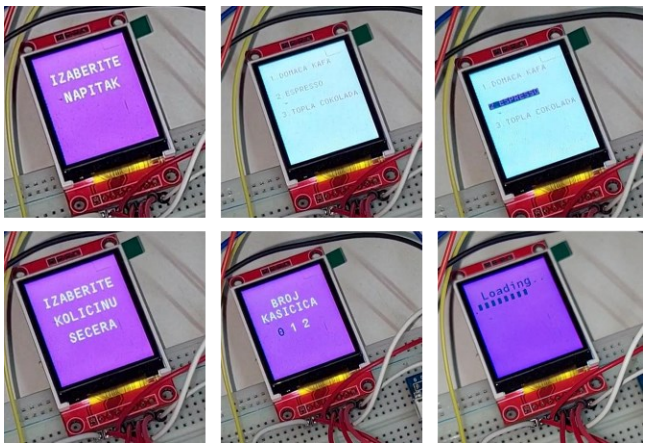
Praktičnom realizacijom prikazano je projektovanje korisničkog interfejsa aparata za kafu. Ovakav sistem može se unaprediti na razne načine. Na primer, moguća je upotreba drugog tipa displeja, dodavanje tastera i većeg broja opcija koji sistem može da pruži i slično. Zavisno od primene moguće je jednostavno prilagoditi jedan ovakav sistem. Korišćenjem MOS tranzistora snage i DC motora simuliran je rad uređaja, ali je dodavanjem sistema pumpi, senzora temperature i pritiska moguće realizovati funkcionalni aparat za kafu.



Slika 9. Praktična realizacija sistema.

ZAHVALNICA

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



Slika 10. Korisnički interfejs projektovanog aparata za kafu.

LITERATURA

[1] Istorijat kafa aparata. Dostupno na: https://www.smithsonianmag.com/arts-culture/the-long-history-of-the-espresso-machine-126012814/

[2] STM32F103C8 *Blue Pill* razvojno okruženje. Dostupno na: https://components101.com/microcontrollers/stm32f103c8t8-blue-pill-development-board

[3] Tehnička specifikacija za STM32F103. Dostupno na: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf

[4] TFT displej. Dostupno na: https://www.electronics-lab.com/project/using-st7735-1-8-color-tft-display-arduino/

[5] Tehnička dokumentacija za MOS transistor snage IRF510. Dostupno na: https://www.vishay.com/docs/91015/irf510.pdf

[6] Tehnička dokumentacija za TXS0108E drajver. Dostupno na: https://www.ti.com/lit/ds/symlink/txs0108e.pdf?ts=1686387504824

[7] Nikola Mitrović, Miloš Marjanović, Danijel Danković, Primena ARM mikrokontrolera, Akademska Misao, 2023. ISBN: 978-86-7466-951-8

Emulacija mašine za pranje veša bazirana na Blue Pill razvojnom okruženju

Boris Jevtić, Milica Petrušijević

Sadržaj – U ovom radu je realizovan sistem koji korišćenjem Blue Pill razvojnog okruženja emulira rad mašine za pranje veša po ciklusima. Ideja je da emulacija bude osnova na koju će se, usložnjavanjem kola i dodadavanjem odgovarajućih senzora, doći do realnog i primenjivog programatora mašine za pranje i sušenje veša.

I. Uvod

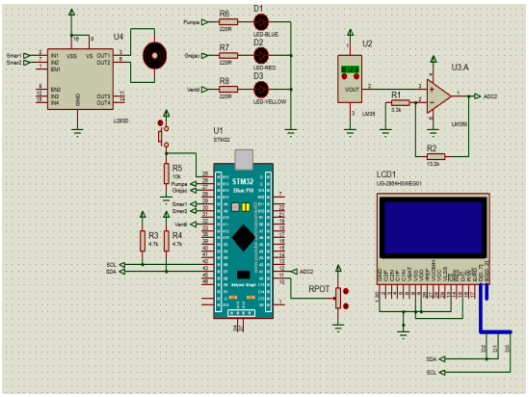
Najraniji pokušaji automatizacije procesa pranja veša datiraju još iz 17. veka. Prve mašine su se oslanjale na ručni rad, da bi u industrijskoj revoluciji koristile paru kao pogonsku snagu za rotiranje bubnja u kom su se prale tkanine. Prve električne veš mašine su se pojavile krajem 19. veka, ali su bile skupe i retko dostupne široj populaciji. U drugoj polovini 20. veka, tehnološki napredak doveo je do sve veće automatizacije i inovacija u industriji veš mašina. Uvođenje mikrokontrolera i elektronike omogućilo je razvoj pametnih veš mašina s programabilnim funkcijama i senzorima koji prate parametre pranja. Ove inovacije su poboljšale performanse, efikasnost i praktičnost veš mašina. U poslednjim decenijama, sve veća pažnja se posvećuje ekološkim aspektima i energetskoj efikasnosti veš mašina. Novi modeli veš mašina su osmišljeni kako bi smanjili potrošnju vode i energije, koristeći napredne tehnologije kao što su senzori za merenje zapremine veša i automatsko prilagođavanje potrošnje. Danas, veš mašine predstavljaju nezaobilazan deo modernog domaćinstva i neprestano se usavršavaju. Uz razvoj pametnih tehnologija, veš mašine postaju sve povezanije s pametnim kućama i mogućnošću daljinskog upravljanja. Brzi ciklusi pranja, tiši rad, veći kapaciteti i poboljšana energetska efikasnost samo su neke od karakteristika današnjih veš mašina koje olakšavaju svakodnevicu.

U ovom radu biće predstavljen inicijalni model programatora veš mašine. Po uključenju uređaja, korisnik bira program. Ponuđeni su programi „30“, „40“ i „60“. Okretanjem klizača potenciometra i pritiskom na taster vrši se odabir željenog programa. Nakon odabira uključuje se LE dioda plave boje koja predstavlja pumpu za uvlačenje vode. Nakon toga, uključuje se LE dioda crvene boje koja predstavlja grejač. Istovremeno počinje i očitavanje vrednosti temperature sa senzora. Kada temperatura dostigne odabranu vrednost (senzor se zagreva eksterno), grejač se isključuje i prelazi se na sledeći deo programa. Motor počinje da se okreće manjom brzinom, naizmenično menjajući smer. Ovaj ciklus se ponavlja nekoliko puta. Nakon toga motor počinje da se okreće većom brzinom (simulirajući centrifugu) i istovremeno se uključuje žuta LE dioda koja predstavlja pumpu za izbacivanje vode iz mašine. Završivši ceo program pranja, korisniku se na OLED displeju ispisuje poruka da je pranje završeno.

B. Jevtić i M. Petrušijević su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milica.petrusijevic@elfak.rs. borisjevtic@elfak.rs.

II. Hardverska realizacija

Za projektovanje električne šeme korišćen je softverski paket *Proteus*. Električna šema prikazana je na slici 1.

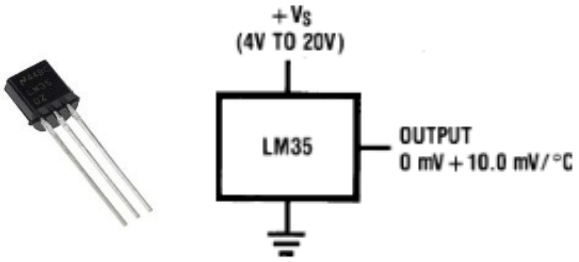


Slika 1. Električna šema sistema.

BluePill je razvojno okruženje bazirano na STM32 mikrokontroleru STM32F103C8Tx [1].

Serija LM35 (slika 2) su precizni temperaturni senzori čiji je izlazni napon linearno proporcionalan temperaturi u °C. LM35 ima prednost u odnosu na linearne temperaturne senzore kalibrisane u Kelvinima, pošto korisnik ne mora da oduzima veliki konstantan napon od izlaza da bi dobio skaliranje °C. Niska izlazna impedansa, linearni izlaz i precizna inherentna kalibracija čine interfejs za očitavanje ili kontrolu kola jednostavnim [2].

Prilikom merenja temperature, izlazni napon sa senzora temperature ima malu vrednost da bi mogao precizno da se očita pomoću analogno-digitalnog konvertora mikrokontrolera. Upotrebljen je operacioni pojačavač LM358N u neinvertujućoj konfiguraciji sa ciljem da se maksimalno iskoristi rezolucija analogno-digitalnog konvertora . Korišćenjem otpornika R_1=3.3 kΩ i R_2=13.2 kΩ dobija se pojačanje od A=5.



Slika 2. Temperaturni senzor LM35 – izgled kućišta i pinovi.

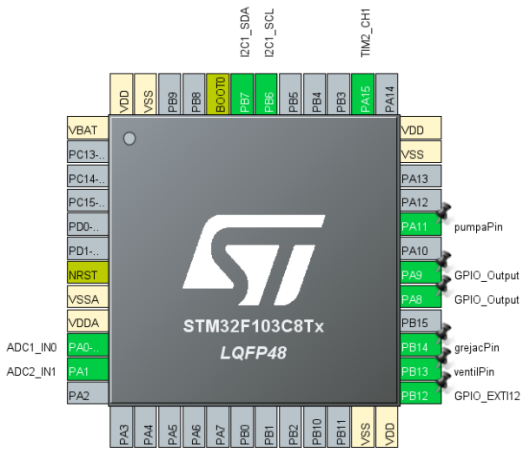
Za upravljanje DC motorom korišćeno je kolo L293D. Brzina okretanja motora određuje se PWM signalom sa željenim faktorom ispune, a smer okretanja motora zavisi od podešavanja naponskih nivoa pinova A8 i A9. Prateći tehničku dokumentaciju [3] na pinove integrisanog kola povezani su i odgovarajući kondenzatori.

Za grafički prikaz podataka iskorišćen je OLED displej. OLED displej komunicira sa mikrokontrolerom preko I2C komunikacionog protokola. Dodati su i pull-up otpornici na pinovima za SDA i SCL [4].

III. SOFTVERSKA REALIZACIJA

U programu *CubeMx* izvršena je inicijalizacija pinova neophodnih za funkcionisanje uređaja. Najpre je odabrana I2C komunikacija na kanalu 1 u *fast* modu i automatski su odabrani pinovi PB7 za SDA i PB6 za SCL. Preko ovih pinova vrši se komunikacija sa OLED-om. Prvi analogno-digitalni konvertor, na pinu PA0, korišćen je za očitavanje položaja klizača potenciometra, dok je drugi AD konvertor, na pinu PA1 iskorišćen za očitavanje temperature sa senzora LM35. Tajmer definisan na kanalu CH1 služi za generisanje PWM signala, promenljivog faktora ispune, čime se reguliše brzina okretanja DC motora. Pinovi PB13, PB14, PA11, PA9, PA8 su definisani kao izlazi pinovi, pri čemu PB13 služi za uključivanje i isključivanje diode koja svetli pri radu ventila. Pin PB14 služi za uključivanje i isključivanje diode koja svetli pri radu grejača, dok pin PA11 služi za uključivanje i isključivanje diode koja svetli pri radu pumpe. Pinovi PA9 i PA8 služe sa određivanje smera okretanja elektromotora. Hardverski interrupt detektuje se na pinu PB12, što se koristi za detekciju pritiska tastera. Prikaz iskorišćenih pinova mikrokontrolera dat je na slici 3.

Na osnovu zadatih podešavanja u alatu *CubeMX* generisan je kostur koda za projekat programatora veš mašine.



Slika 3. Iskorišćeni pinovi mikrokontrolera.

U nastavku je dat kod koji omogućava opisanu funkcionalnost sistema. Na početku programa izvršena je inicijalizacija sledećih biblioteka: "stdio.h", "string.h", "fonts.h", "ssd1306.h". Deklarisane su sledeće promenljive: *uint16_t vrednost, odabranaVrednost, pranje=0, adc2; char baffer[20]; char trideset[]="30"; char cetrdeset[]="40"; char sezdeset[]="60"; volatile uint8_t k=0, duty, i, j; float temp=0.0, senzorNapon*. Na slici 4 su date funkcije. Prva funkcija *centrifuga(int a)* služi za „soft start“ pokretanje motora postepenim povećavanjem brzine okretanja, gde se odabirom vrednosti parametra *a* definiše trajanje ciklusa u sekundama. Druga funkcija *sporoOkretanje(int a)* služi za okretanje motora naizmenično u jednom i drugom smeru gde se odabirom vrednosti parametra *a*, po istom principu i za isti vremenski interval, definiše trajanje ciklusa sporog okretanja.

Kod za inicijalizaciju OLED dislpeja, tajmera i analogno digitalnih konvertora, kao i pozicioniranje teksta na displeju prikazan je na slici 5. Kod glavne programske petlje prikazan je na slici 6. Najpre za odabir moda, u glavnu while pelju ugnježdena je još jedna while petlja koja će dok je K=0 definisati 3 programa rada veš mašine - željenu temperaturu pranja, tj. mod. Trenutno izabrani mod biće prikazan na displeju većim fontom u odnosu na druga dva moguća moda.

Po odabiru programa na ekranu OLED-a ispisuje se naziv programa koji je korisnik odabrao (slika 7). Nakon toga počinje primanje vode i svetli plava LE dioda (slika 8). U toku primanja vode, ispisuje se poruka na displeju OLED-a „Primanje vode“ i naizmenično se ispisuju 3 tačkice u nastavku poruke koje simbolizuju čekanje dok se završi taj proces.

```
75 void centrifuga(int a){
76     for(duty=0;duty<100;duty++){
77     htim2.Instance->CCR1=duty;
78     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,1);
79     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
80     HAL_Delay(25);
81     }
82     htim2.Instance->CCR1=duty;
83     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,1);
84     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
85     HAL_Delay(1000*a);
86     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,0);
87     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
88 }
89
90 void sporoOkretanje(int a){
91     duty=20;
92     for(i=0;i<a;i++){
93     htim2.Instance->CCR1=duty;
94     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,1);
95     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
96     HAL_Delay(1000);
97     }
98     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,0);
99     HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
100    HAL_Delay(3000);
101    for(i=0;i<a;i++){
102    htim2.Instance->CCR1=duty;
103    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,0);
104    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,1);
105    HAL_Delay(1000);
106    }
107    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_8,0);
108    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,0);
109    HAL_Delay(3000);
110  }
```

Slika 4. Kod funkcija *centrifuga()* i *sporoOkretanje()*.

```
146 /* USER CODE BEGIN 2 */
147 SSD1306_Init();
148 HAL_ADC_Start(&hadc1);
149 HAL_ADC_Start(&hadc2);
150 HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1);
151   SSD1306_GotoXY(0,0);
152   SSD1306_Puts("Izaberite program:",&Font_7x10, SSD1306_COLOR_WHITE);
153   SSD1306_GotoXY(0,32);
154   SSD1306_Puts(trideset,&Font_7x10, SSD1306_COLOR_WHITE);
155   SSD1306_GotoXY(55,32);
156   SSD1306_Puts(cetrdeset,&Font_7x10, SSD1306_COLOR_WHITE);
157   SSD1306_GotoXY(113,32);
158   SSD1306_Puts(sezdeset,&Font_7x10, SSD1306_COLOR_WHITE);
159   SSD1306_UpdateScreen();
160 /* USER CODE END 2 */
```

Slika 5. Kod za inicijalizaciju displeja, AD konvertora i tajmera.

```
169   /* USER CODE BEGIN 3 */
170
171 while(pranje==0){
172   while(k==0){
173   vrednost=HAL_ADC_GetValue(&hadc1);
174   if(vrednost<1250){
175     SSD1306_DrawFilledRectangle(0,32,128,20,SSD1306_COLOR_BLACK);
176     SSD1306_GotoXY(0,32);
177     SSD1306_Puts(trideset,&Font_11x18, SSD1306_COLOR_WHITE);
178     SSD1306_GotoXY(55,32);
179     SSD1306_Puts(cetrdeset,&Font_7x10, SSD1306_COLOR_WHITE);
180     SSD1306_GotoXY(113,32);
181     SSD1306_Puts(sezdeset,&Font_7x10, SSD1306_COLOR_WHITE);
182     SSD1306_UpdateScreen();
183     odabranaVrednost=30;
184   }
185   else if(vrednost>1400 && vrednost<2700){
186     SSD1306_DrawFilledRectangle(0,32,128,20,SSD1306_COLOR_BLACK);
187     SSD1306_GotoXY(0,32);
188     SSD1306_Puts(trideset,&Font_7x10, SSD1306_COLOR_WHITE);
189     SSD1306_GotoXY(55,32);
190     SSD1306_Puts(cetrdeset,&Font_11x18, SSD1306_COLOR_WHITE);
191     SSD1306_GotoXY(113,32);
192     SSD1306_Puts(sezdeset,&Font_7x10, SSD1306_COLOR_WHITE);
193     SSD1306_UpdateScreen();
194     odabranaVrednost=40;
```

Slika 6. Kod glavne programske petlje.

```
209   //ODABRAN PROGRAM
210
211   SSD1306_Clear();
212   SSD1306_GotoXY(0,0);
213   SSD1306_Puts("Odabrani program:",&Font_7x10,SSD1306_COLOR_WHITE);
214   SSD1306_GotoXY(55,32);
215   sprintf(baffer, "%d", odabranaVrednost);
216   SSD1306_Puts(baffer,&Font_11x18, SSD1306_COLOR_WHITE);
217   SSD1306_UpdateScreen();
218   HAL_Delay(3000);
219   SSD1306_Clear();
```

Slika 7. Kod za prikaz odabranog moda rada.

```
221   //PRIMANJE VODE
222
223   for(i=0;i<6;i++){
224       HAL_GPIO_WritePin(ventilPin_GPIO_Port,ventilPin_Pin,1);
225       SSD1306_GotoXY(0,29);
226       SSD1306_Puts("Primanje vode",&Font_7x10, SSD1306_COLOR_WHITE);
227       SSD1306_UpdateScreen();
228     HAL_Delay(500);
229     for(j=0;j<3;j++){
230       SSD1306_Puts(".",&Font_7x10, SSD1306_COLOR_WHITE);
231       SSD1306_UpdateScreen();
232       HAL_Delay(500);
233     }
234     SSD1306_DrawFilledRectangle(92,26,21,30,SSD1306_COLOR_BLACK);
235   }
236   HAL_GPIO_WritePin(ventilPin_GPIO_Port,ventilPin_Pin,0);
237   SSD1306_Clear();
238
```

Slika 8. Kod za emulaciju primanja vode.

Kod za emulaciju zagrevanja vode dat je na slici 9. Ovaj deo koda opisuje da sve dok temperatura vode koja se očitava senzorom ne dostigne odabranu vrednost, grejač radi, na displeju se ispisuje proces koji se trenutno odigrava „Grejanje vode“ i trenutna vrednost temperature koju senzor očitava. Tokom ovog procesa svetli crvena dioda. U nastavku se tri puta poziva funkcija „sporoOkretanje“ i time se emulira početak pranja tj. okretanje bubnja mašine. U delu koda sa slike 10 se podešava da se na ekranu OLED-a ispisuje tekst „Izbacivanje vode“, i tokom ovog procesa svetli žuta dioda. Emulacija centrifugiranja omogućena je kodom sa slike 11. Postavljanjem flega *pranje* na 1, proces se završava (slika 12).

```
239   //UKLJUCIVANJE GREJACA
240
241   HAL_Delay(2000);
242   while(temp<odabranaVrednost){
243
244     adc2=HAL_ADC_GetValue(&hadc2);
245     senzorNapon=(adc2*3.3)/4095.0; temp=senzorNapon/0.05;
246   HAL_GPIO_WritePin(grejacPin_GPIO_Port,grejacPin_Pin,1);
247       SSD1306_GotoXY(0,29);
248       SSD1306_Puts("Grejanje vode",&Font_7x10, SSD1306_COLOR_WHITE);
249       SSD1306_UpdateScreen();
250     HAL_Delay(500);
251     for(j=0;j<3;j++){
252       SSD1306_Puts(".",&Font_7x10, SSD1306_COLOR_WHITE);
253       SSD1306_UpdateScreen();
254       HAL_Delay(500);
255     }
256     SSD1306_DrawFilledRectangle(92,26,21,30,SSD1306_COLOR_BLACK);
257     SSD1306_GotoXY(0,40);
258     sprintf(baffer, "%.2f", temp);
259     SSD1306_Puts(baffer, &Font_7x10, SSD1306_COLOR_WHITE);
260     SSD1306_UpdateScreen();
261
262   }
263   HAL_GPIO_WritePin(grejacPin_GPIO_Port,grejacPin_Pin,0);
264     SSD1306_Clear();
265
```

Slika 9. Kod za emulaciju procesa zagrevanja vode.

```
272     //IZBACIVANJE VODE
273
274     for(i=0;i<6;i++){
275     HAL_GPIO_WritePin(pumpaPin_GPIO_Port,pumpaPin_Pin,1);
276     SSD1306_GotoXY(0,29);
277     SSD1306_Puts("Izbacivanje vode",&Font_7x10, SSD1306_COLOR_WHITE);
278     SSD1306_UpdateScreen();
279     HAL_Delay(500);
280     for(j=0;j<2;j++){
281     SSD1306_Puts(".", &Font_7x10, SSD1306_COLOR_WHITE);
282     SSD1306_UpdateScreen();
283     HAL_Delay(500);
284       }
285     SSD1306_DrawFilledRectangle(113,26,20,20,SSD1306_COLOR_BLACK);
286     SSD1306_UpdateScreen();
287     }
288     HAL_Delay(3000);
289     SSD1306_Clear();
290
```

Slika 10. Kod za emulaciju izbacivanja vode.

```
292     //CENTRIFUGA
293
294     centrifuga(10);
295     HAL_GPIO_WritePin(pumpaPin_GPIO_Port,pumpaPin_Pin,0);
296     SSD1306_Clear();
297     pranje=1;
298   }
```

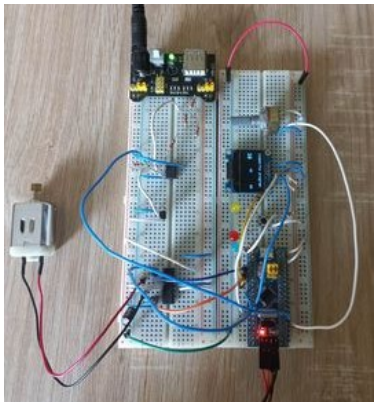
Slika 11. Kod za emulaciju centrifugiranja.

```
299     //PRANJE GOTOVO
300
301
302     SSD1306_GotoXY(18,29);
303     SSD1306_Puts("PRANJE GOTOVO", &Font_7x10,SSD1306_COLOR_WHITE);
304     SSD1306_UpdateScreen();
305     //SSD1306_DrawFilledRectangle(0,0,127,63,SSD1306_COLOR_BLACK);
306
```

Slika 12. Kod za emulaciju kraja procesa pranja.

IV. Praktična realizacija

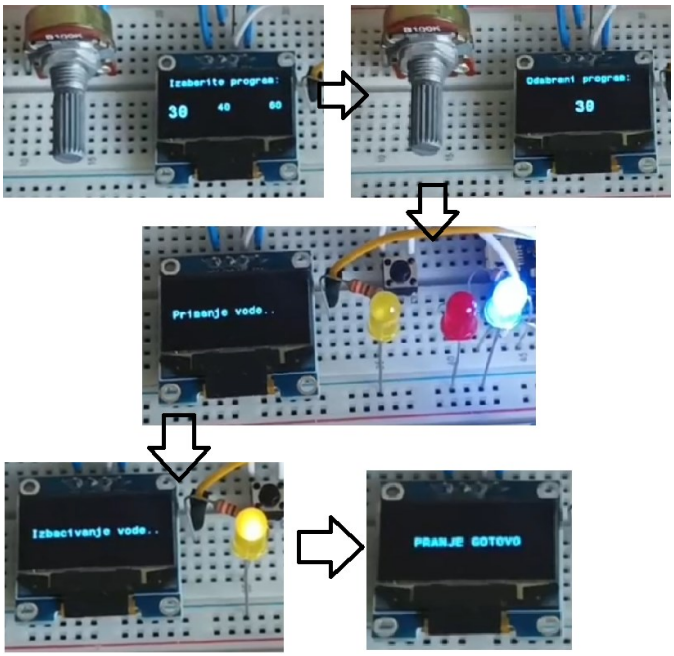
Relizacija opisanog sistema na protoploči prikazana je na slici 13. Na slici 14 ilustrovane su poruke koje se prikazuju na OLED-u tokom emulacije procesa pranja.



Slika 13. Praktična realizacija sistema na protoploči.

V. Zaključak

Projekat je moguće dalje razviti dodavanjem delova prave mašine za pranje veša kao što su motor, grejač, ventil, pumpa, TFT displej… Svaki od ovih delova zahtevao bi i dodatne blokove za njihovo funkcionisanje i kao i korišćenje komponenata snage. Imajući u vidu da se u našoj zemlji mašine za pranje veša ne proizvode već pune dve decenije, smatramo da bi ovaj model bio dobra osnova za razvoj mašine za pranje veša koja bi bila zasnovana na domaćoj pameti i, u perspektivi, domaćim komponentama.



Slika 14. Prikaz poruka na OLED-u tokom emulacije procesa pranja.

Literatura

[1] Tehnička specifikacija za STM32. Dostupno na: https://www.alldatasheet.com/datasheet-pdf/pdf/201590/STMICROELECTRONICS/STM32F103C8.html

[2] Tehnička dokumentacija za LM35 senzor. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/517588/TI1/LM35.html

[3] Tehnička dokumentacija za L293D. Dostupno na: https://www.ti.com/lit/ds/symlink/l293.pdf

[4] Tehnička dokumentacija za OLED displej. Dostupno na: https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pd

Prepoznavanje nekih elemenata znakovnog jezika pomoću mašinskog učenja

Natalija Ivković

Sadržaj − U ovom radu je prikazano kako se metode mašinskog učenja, konkretno slučajna šuma, mogu efikasno primeniti u realnom vremenu za prepoznavanje znakovnog jezika. Kroz par primera predstavljena je dobijena tačnost modela, a cilj je da ovaj rad pruži doprinos razvoju tehnologije za prepoznavanje znakovnog jezika, postavljajući temelj za buduće aplikacije koje unapređuju komunikaciju sa osobama koje koriste znakovni jezik.

I. UVOD

Izraz "znakovni jezik" označava vizuelni način sporazumevanja gluvonemih osoba, posebno ugovorenim, odnosno standardizovanim znacima ruku i prstiju, sa ili bez mimike, sa sopstvenom gramatikom i sintaksom, pomoću koga se prenose namere, predstave, misli i pojmovi [1].

Bitno je istaći to da pokazivanje znakova na znakovnom jeziku mogu da obavljaju i osobe koje nisu gluve, već su fizički u nemogućnosti da govore. Znakovni jezici nisu isti u celom svetu i ne postoji međunarodni znakovni jezik. Svaka zemlja obično ima sopstveni, maternji znakovni jezik, a neke imaju i više od jednog [2].

Srpski znakovni jezik (SZJ) je jezik ravnopravan sa srpskim jezikom i "glas" zajednice gluvih u Srbiji. Koristi ga preko 30.000 ljudi u našoj zemlji. U srpskom znakovnom jeziku znakovi se dele na nekoliko vrsta, dok postoje jednoručna i dvoručna azbuka, kao i jednoručna i dvoručna abeceda [3].

U poslednjih par decenija, mašinsko učenje je postalo jedno od najznačajnijih područja razvoja u oblasti računarstva i veštačke inteligencije. Ovaj interdisciplinarni domen pruža računarima mogućnost da automatski uče iz podataka i stiču znanja bez eksplicitnog programiranja. Mašinsko učenje unapređuje način na koji računari obrađuju informacije i donose odluke.

Mašinsko učenje ima široku primenu u različitim oblastima, uključujući kompjutersku video-analizu, obradu prirodnih jezika, medicinu, finansiju, robotiku i mnoge druge. Ono može biti koristan alat za otkrivanje zločina, prognoziranje tržišnih trendova, prepoznavanje oblika i lica, otkrivanje raznih bolesti, prevođenje jezika i brojne druge zadatke koji zahtevaju automatsku analizu i interpretaciju podataka [4].

N. Ivković je saradnik u nastavi pri Katedri za teorijsku elektrotehniku, na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: natalija.ivkovic@elfak.ni.ac.rs

Ovaj rad predstavlja deo master rada autora [5], gde je za prepoznavanje nekih elemenata znakovnog jezika primenjena metoda Slučajna šuma.

II. SLUČAJNA ŠUMA

Slučajna šuma (Random Forest) je algoritam mašinskog učenja koji se koristi za klasifikaciju i regresiju. On predstavlja ansambl metod, što znači da kombinuje više modela, konkretno stabla odlučivanja, kako bi doneo krajnje predviđanje. Osnovna ideja slučajne šume je da se kreira veliki broj stabala odlučivanja na osnovu slučajno izabranih podskupova podataka i varijabli. Svako stablo se trenira na svom podskupu podataka i glasovima svakog stabla odlučivanja se donosi krajnje predviđanje.

Da bi se što bolje razumeo koncept slučajne šume, neophodno je prvo u kratkim crtama objasniti stabla odlučivanja.

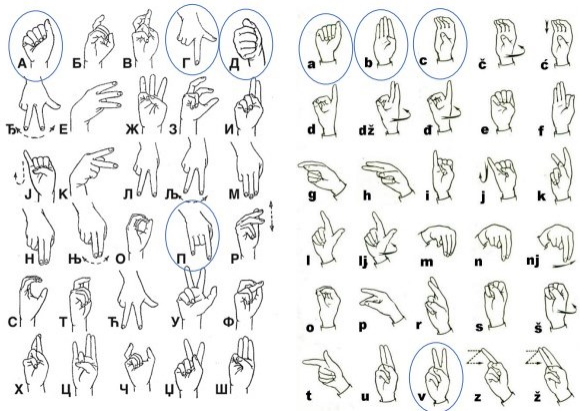
Stablo odlučivanja je jedna od metoda nadgledanog mašinskog učenja, koja se koristi u pronalaženju skrivenog znanja i kod koje je cilj da se kreira model koji predviđa vrednost, odnosno klasu, ciljne promenljive na osnovu više ulaznih promenljivih. Stablo odlučivanja spada u nadgledane metode mašinskog učenja jer se ovde zadaju trening parovi kod kojih je svakom ulaznom podatku X pridružena željena izlazna vrednost y. Svaki ulazni podatak ovde je, zapravo, skup više atributa. Cilj treniranja ovakvih modela je da se na osnovu datih trening parova (X, y) pronađe optimalna funkcija koja mapira ulaz u izlaz, a sa ciljem da se ta funkcija kasnije iskoristi za uspešno predviđanje vrednosti izlaza za nove, do tada neviđene, ulaze. Ukoliko je ciljna vrednost y kontinualna, radi se o problemu regresije, pa se stablo odlučivanja koje rešava takav problem naziva regresiono stablo odlučivanja. Kada su ciljne vrednosti diskretne, kategoričke, radi se o problemu klasifikacije. Tada se stablo odlučivanja naziva klasifikaciono. Stabla odlučivanja se najčešće koriste u problemima klasifikacije.

Kada se ne uzgaja samo jedno stablo odlučivanja, već više stabala istovremeno, dobijamo slučajnu šumu. Na kraju, kada se pojavi novi ulazni podatak, on se propusti kroz sva stabala odlučivanja i konačno predviđanje kod klasifikacije dobijamo kao glas većine, tj. kao onu klasu koju kao svoj rezultat dobije većina stabala, a kod regresije kao prosečnu, odnosno srednju, vrednost dobijenih rezultata [6].

III. REZULTATI

Kao i svaki projekat koji se bazira na mašinskom učenju, i projekat prepoznavanja znakovnog jezika ima tri osnovna dela, a to su: sakupljanje i obrada trening podataka, treniranje modela i testiranje modela.

U master radu [5] istreniran je model mašinskog učenja koji će da prepoznaje osam različitih slova – četiri slova iz jednoručne azbuke i četiri slova iz jednoručne abecede srpskog znakovnog jezika, koja su obeležena na slici 1.



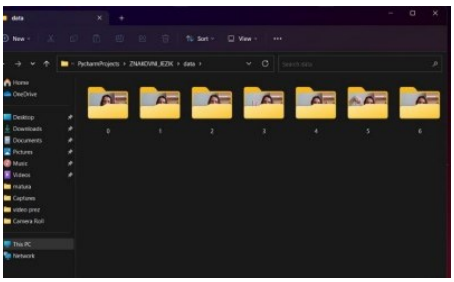
Slika 1. Slova jednoručne abecede i jednoručne azbuke srpskog znakovnog jezika koja će model mašinskog učenja moći da prepozna [7].

A. Sakupljanje i obrada trening podataka

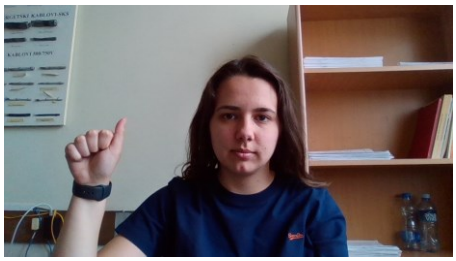
Sakupljanje trening podataka, a kasnije i obrada tih podataka, da bi se dobile korisne informacije, koje će se koristiti pri treniranju samog modela mašinskog učenja, predstavljaju ključnu stvar. Ukoliko nemamo „korisne" trening podatke, naš model neće imati visoku tačnost, odnosno neće vršiti ispravna predviđanja.

Nakon izvršenja koda za sakupljanje podataka, kao rezultat imamo kreiran direktorijum sa nazivom „data" u trenutnom radnom direktorijumu. Taj direktorijum sadrži sedam poddirektorijuma, sa nazivima: „0", „1", „2", „3", „4", „5" i „6", a svaki poddirektorijum sadrži po 400 slika, jednog slova koje će model moći da prepozna. Na slici 2 je prikazan izgled direktorijuma "data", dok su na slikama 3 i 4 prikazani primeri trening podataka iz određenih poddirektorijuma direktorijuma "data".

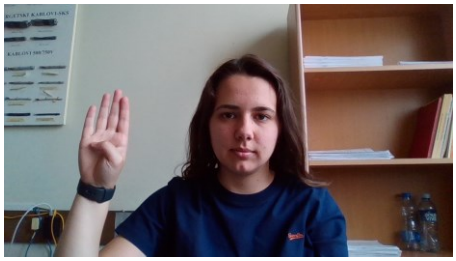
U delu koda za obradu trening podataka se koriste različite Python biblioteke za obradu slika i podataka. Prvo, koristimo biblioteke poput *os* i *cv2* za procesiranje slika i korišćenje komandi operativnog sistema. *Pickle* se koristi za serijalizaciju i de-serijalizaciju objekata, čime se omogućava čuvanje i prenos podataka između programa. Biblioteka *mediapipe* se koristi za analizu karakterističnih tačaka na rukama na slikama, *matplotlib.pyplot* je deo *matplotlib* biblioteke koji se koristi za crtanje grafova i vizualizaciju podataka.



Slika 2. Izgled direktorijuma "data".



Slika 3. Primer podatka za treniranje modela iz klase (poddirektorijuma) "0" – slovo "A".



Slika 4. Primer podatka za treniranje modela iz klase (poddirektorijuma) "1" – slovo "B" (latinica).

Nakon učitavanja potrebnih biblioteka, kreira se model za detekciju ruku i određuju se parametri za detekciju. Zatim se čitaju slike, konvertuju u odgovarajući format za analizu, i karakteristične tačke na rukama se detektuju i čuvaju. Ovi podaci se prikupljaju za sve slike i čuvaju u datoteci u binarnom formatu. Kod se koristi za pripremu podataka za obuku modela mašinskog učenja za prepoznavanje znakovnog jezika na slikama ruku.

B. Treniranje modela

U ovom delu koda se koriste Python biblioteke *pickle*, *numpy*, i *sklearn* za obradu podataka i treniranje modela mašinskog učenja. Najpre se podaci učitavaju iz datoteke i pripremaju za analizu. Zatim se podaci dele na trening i test skupove. Od ukupno 2800 slika, trening skup sadrži 2240 slika (80%), a test skup 560 slika (20%). Nakon toga, koristi se algoritam *Random Forest* za treniranje modela na trening podacima. Na kraju, model se procenjuje na test podacima i čuva za buduću upotrebu.

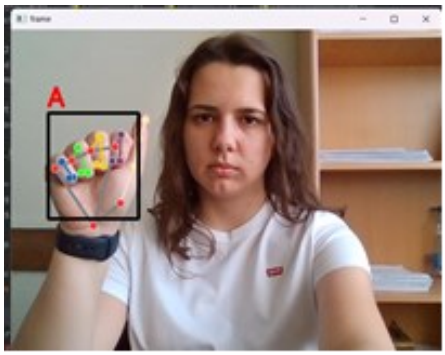
C. Testiranje modela

Nakon sakupljanja podataka i treniranja modela, sledi testiranje modela. Jednostavnije rečeno, testiranje modela predstavlja pokretanje programa da bi se videlo da li program radi ispravno.

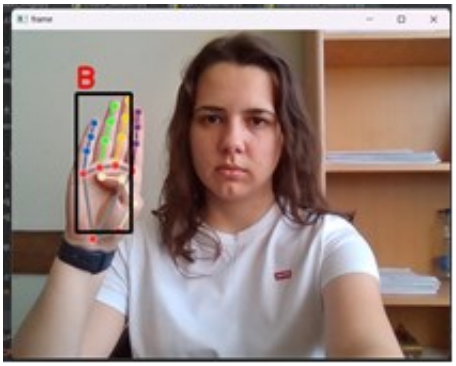
U ovom delu koda vrši se obrada videa kako bi se detektovala pozicija ruke i predviđala pokazana gestikulacija koristeći prethodno obučen model mašinskog učenja.

Kamere očitavaju frejmove videa, a zatim se analiziraju karakteristične tačke na ruci, normalizuju koordinate tih tačaka, i na osnovu tih podataka se koristi model za predviđanje gestikulacije ruke. Rezultat se iscrtava na frejmu i prikazuje korisniku u stvarnom vremenu.

Na slici 5 i slici 6 prikazano je kako izgleda kada model mašinskog učenja prepozna slovo “A” i slovo “B” (latinica), respektivno.



Slika 5. Testiranje modela – slovo “A”.



Slika 6. Testiranje modela – slovo “B” (latinica).

IV. TESTIRANJE PROGRAMA

Program su testirale četiri osobe, tako što je svaka osoba pokazala po četiri puta slova: „A”, „B” (latinica) „C” (latinica), „V” i „П” ispravno i pogrešno (dva puta levom i dva puta desnom rukom). Podskup slova za testiranje programa je izabran nasumično.

A. Pokazivanje ispravnih slova

Rezultati pokazivanja ispravnih slova su dati u tabeli I.

TABELA I
TAČNOST PROGRAMA PRI POKAZIVANJU ISPRAVNIH SLOVA

Osoba	“A”	“B” (latinica)	“C” (latinica)	“V”	„П”
Osoba 1	4/4	4/4	4/4	4/4	4/4
Osoba 2	4/4	4/4	4/4	4/4	4/4
Osoba 3	4/4	4/4	4/4	4/4	4/4
Osoba 4	4/4	4/4	4/4	4/4	4/4

Pokazivanje ispravnih slova i rezultati koji su prikazani u prethodnoj tabeli nam potvrđuju da program radi potpuno ispravno, sa idealnom tačnošću, kada različite osobe pokazuju slova onako kako ta slova zaista izgledaju. To praktično znači da na program ne utiče ko pokazuje slova, već šta se pokazuje.

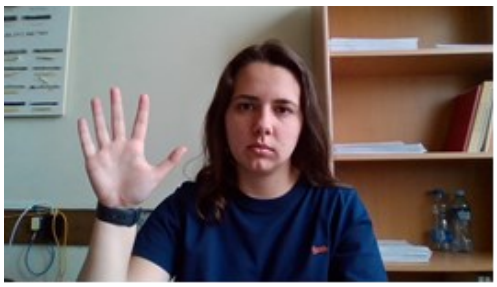
B. Pokazivanje pogrešnih slova

Program je, takođe, testiran i tako što su različite osobe pokazivale pogrešna slova, tj. slova (oblike) koja slično izgledaju kao i slova „A”, „B” (latinica), „C” (latinica), „V” i „П”. Pri pokazivanju pogrešnih slova, program je nekada pravio greške, tako što je, na primer, pri pokazivanju pogrešnog slova „C” (latinica) on prepoznavao slovo „A” (i obrnuto) ili pri pokazivanju pogrešnog slova „B” (latinica) on prepoznavao slovo „V”. Primećeno je da, pri pokazivanju pogrešnih slova, ključnu ulogu ima položaj naše šake, bilo da se radi o levoj ili desnoj. Zavisno od položaja šake kojom pokazujemo pogrešna slova, ključne tačke koje program prepoznaje su različite. Pri prepoznavanju pogrešnih slova, program radi tako što on korisniku prikazuje slovo za koje postoji najveća verovatnoća da ga je pokazao, odnosno čije se karakteristične tačke poklapaju sa najvećom tačnošću sa određenim ispravnim slovom.

Na slikama 7 i 8 su prikazana neka od pogrešnih slova korišćena za ovu analizu.



Slika 7. Pogrešno slovo “A”.



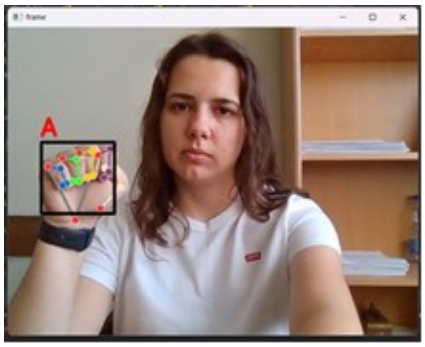
Slika 8. Pogrešno slovo “B” (latinica).

U tabeli II prikazani su rezultati pokazivanja pogrešnih slova.

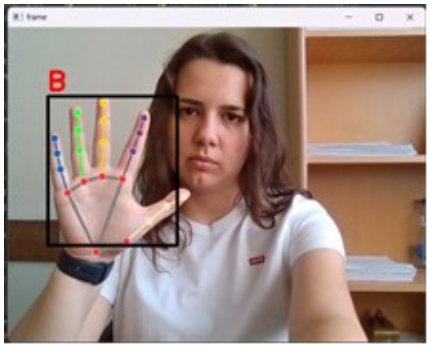
TABELA II
TAČNOST PROGRAMA PRI POKAZIVANJU POGREŠNIH SLOVA

Osoba	“A”	“B” (latinica)	“C” (latinica)	“V”	„П”
Osoba 1	4/4	4/4	3/4	4/4	4/4
Osoba 2	3/4	4/4	4/4	4/4	4/4
Osoba 3	4/4	3/4	3/4	4/4	4/4
Osoba 4	4/4	4/4	4/4	4/4	4/4

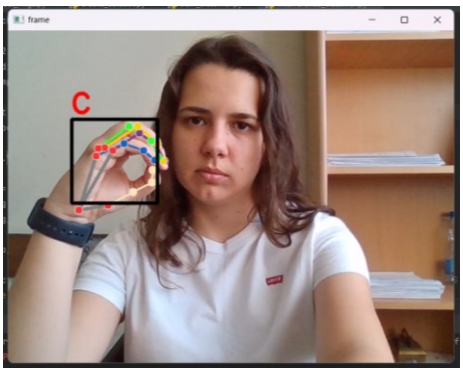
Na slikama 9, 10 i 11 je prikazano kako program prepoznaje pogrešna slova.



Slika 9. Testiranje programa – pokazivanje pogrešnog “A”.



Slika 10. Testiranje programa – pokazivanje pogrešnog “B” (latinica).



Slika 11. Testiranje programa – pokazivanje pogrešnog “C” (latinica).

Prepoznavanje pogrešnih slova sa jedne strane može da bude dobra osobina – da naš model uspešno prepozna određeno slovo iako nismo baš sigurni kako ono treba da izgleda, pa pokažemo nešto slično. Međutim loša strana prepoznavanja pogrešnih slova je ta što, na primer, pogrešno „C” (latinica), slika 11, predstavlja ispravno „O”, tako da program (koji u ovom konkretnom slučaju nije istreniran da prepozna „O”) pravi veliku grešku.

V. ZAKLJUČAK

Ovaj rad ukazuje na to da se mašinsko učenje može uspešno primeniti i u prepoznavanju znakovnog jezika, što bi moglo imati značajnu ulogu u obrazovanju i komunikaciji sa osobama koje imaju poteškoće sa sluhom i govorom. Namera je, da se u budućnosti, proširi skup znakova koje model može da prepozna, čime će se omogućiti preciznija komunikacija.

ZAHVALNICA

Autor se zahvaljuje prof. dr Saši S. Nikoliću na izdvojenom vremenu, sugestijama i pomoći pri pripremi i realizaciji ovog projekta.

LITERATURA

[1] http://www.sgnscg.com/index.php/znakovni-jezik
[2] https://www.sudski-tumaci.rs/vesti/clanak/znakovni-jezici
[3] https://www.opismenise.com/kolumne/znakovni-jezik-osnovni-pokreti/
[4] https://hashdork.com/sr/machine-learning-algorithms-for-begineers/
[5] N. Ivković, “Prepoznavanje znakovnog jezika pomoću mašinskog učenja”, Elektronski fakultet u Nišu, 2023.
[6] Prof. dr M. Milojković, doc. dr M. Milovanović, „Inteligentni sistemi”, Elektronski fakultet u Nišu, predavanja za školsku 2022/2023.
[7] http://www.sgnscg.com/index.php/prstna-azbuka

Implementacija VPN tehnologije za unapređenje bezbednosti i privatnosti u Internet of Things (IoT) sistemima

Jelena Nedeljković

Sadržaj – U modernom dobu, tehnološki napredak i sveprisutan pristup internetu omogućili su razvoj *Internet of Things* (*IoT*) sistema koji povezuju različite uređaje na internet radi unapređenja svakodnevnog života. Međutim, sa ovim napretkom dolaze i izazovi u vezi sa sigurnošću i privatnošću podataka u ovim sistemima. Ovaj rad istražuje implementaciju *Virtual Private Network* (*VPN*) tehnologije kao ključnog mehanizma za rešavanje bezbednosnih i izazova zaštite privatnosti u *IoT* okruženju. Takođe, pruža praktičan primer implementacije *VPN* servera pomoću *Raspberry Pi* računara.

I. Uvod

U savremenom dobu, koncept *Internet of Things* (*IoT*) je postao ključni element tehnološkog napretka. *IoT* se odnosi na povezivanje različitih uređaja na internet, omogućavajući im da razmenjuju podatke i pružaju automatizaiciju i udobnost u svakodnevnom životu. U ovom okruženju, uređaji poput pametnih kuća, industrijskih sistema i automobila postaju deo globalne mreže. Međutim, ovaj napredak nosi i određene izazove, posebno u pogledu bezbednosti i privatnosti podataka.

Ovaj rad je koncipiran sa ciljem pružanja dubljeg uvida u *Internet of Things* (*IoT*) tehnologiju i njenu povezanost sa *Virtual Private Network* (*VPN*) tehnologijom. Kroz narednih nekoliko poglavlja istražićemo osnovne koncepte, primene i izazove u vezi sa *IoT*-om, kao i ulogu *VPN*-a u obezbeđivanju sigurne i privatne komunikacije u *IoT* ekosistemu.

Prvo poglavlje će poslužiti kao uvod u temu, postavljajući osnovni okvir za ovo istraživanje. Drugo poglavlje će se detaljnije baviti samom *IoT* tehnologijom, treće će objasniti osnove *VPN*-a, dok će četvrto poglavlje analizirati korišćen hardver, *Raspberry Pi* računar. Peto poglavlje će razmotriti portove i mrežnu konfiguraciju, dok će šesto poglavlje doneti praktičnu realizaciju projekta. Na kraju, sedmo poglavlje će ponuditi zaključke i pogled u budućnost ove zanimljive oblasti.

Cilj ovog rada je da doprinese boljem razumevanju kako se *VPN* tehnologija može koristiti za poboljšanje bezbednosti i privatnosti *IoT* sistema, sa posebnim fokusom na primenu na *Raspberry Pi* platformi.

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

II. Internet of Things (IoT)

A. Definicija i koncept IoT - a

Internet stvari (*IoT*) predstavlja koncept u kojem svakodnevni fizički objekti postaju deo globalne mreže povezanih uređaja. Ovaj koncept omogućava objektima da prikupljaju i razmenjuju podatke, čineći ih "pametnim" i sposobnim da efikasnije obavljaju različite zadatke. *IoT* revolucionarno menja način na koji interagujemo sa okolinom i tehnologijom, a takođe ima širok spektar primena u industriji, zdravstvu, transportu i mnogim drugim sektorima [1].

B. Osnovne komponente IoT-a

U cilju boljeg razumevanja *IoT*-a, važno je razmotriti osnovne komponente ovog sistema:

- Senzori;
- Aktuatori;
- Komunikaciona infrastruktura;
- Centralni server / *Cloud*.

C. Primeri primene IoT-a

IoT tehnologija ima široku primenu u različitim oblastima. Neki od primera uključuju:

- Pametne kuće: *IoT* uređaji kao pametni termostati, pametne brave, pametna osvetljenja - omogućavaju korisnicima da kontrolišu svoj dom putem pametnih uređaja ili mobilnih aplikacija.
- Industrija: U industriji *IoT* se koristi za praćenje i upravljanje proizvodnim procesima, optimizaciju logistike i održavanje mašina.
- Zdravstvo: U zdravstvu, *IoT* senzori se koriste za praćenje vitalnih znakova pacijenata, dostavljanje informacija lekarima i poboljšanje dijagnostike i lečenja.
- Transport: *IoT* se koristi za praćenje vozila, optimizaciju rute, poboljšanje bezbednosti na putu i razvoj autonomnih vozila.

Ovo poglavlje je postavilo osnovne koncepte i komponente *IoT*-a, a sledeće poglavlje će se baviti pojmom *VPN*-a i njegovim značajem u kontekstu *IoT*-a.

III. VIRTUELNA PRIVATNA MREŽA (VPN) U KONTEKSTU INTERNET STVARI (IOT)

A. Uvod u virtuelne privatne mreže (VPN)

Virtuelne privatne mreže (*VPN*) predstavljaju tehnologiju koja omogućava siguran, šifrovan i privatan prenos podataka između različitih uređaja ili mreža putem javnih ili nesigurnih mreža, kao što je internet. *VPN*-ovi su postali neophodan alat za očuvanje privatnosti i bezbednosti u digitalnom dobu.

B. Funkcija VPN-a u IoT ekosistemu

U kontekstu Internet stvari (*IoT*), *VPN* ima ključnu ulogu u obezbeđivanju bezbednog komuniciranja između *IoT* uređaja, centralnih servera i korisnika. U nastavku je dato nekoliko ključnih funkcija *VPN*-a u *IoT* ekosistemu:

- Zaštita podataka: *IoT* uređaji često prikupljaju osetljive podatke, kao što su informacije o lokaciji, zdravstveni podaci ili poslovne tajne. *VPN* omogućava šifrovanje ovih podataka tokom prenosa, čime se minimizuje rizik od neovlašćenog pristupa.
- Pristup izdaleka: Korisnici i administratori moraju imati siguran pristup *IoT* uređajima i infrastukturi izdaleka. *VPN* omogućava udaljen pristup bezbednim kanalima, što je ključno za dijagnostiku, konfiguraciju i održavanje uređaja.
- Zaštita od napada: *IoT* uređaji su često mete različitih vrsta sajber napada, uključujući hakovanja i prisluškivanja. *VPN* može delovati kao zaštitni omotač, otežavajući napdačima da prate i napadaju uređaje.
- Ograničenje pristupa: *VPN* može pomoći u usmeravanju i ograničavanju pristupa uređajima, tako da samo ovlašćeni korisnici i uređaji imaju pristup *IoT* mreži

C. Popularni protokoli za IoT VPN

Postoji nekoliko *VPN* protokola koji su posebno prilagođeni za *IoT* primene. Neki od njih uključuju:

- *OpenVPN*: Otvorenog koda, *OpenVPN* je popularan izbor zbog svoje sigurnosti i fleksibilnosti.
- *IPSec* (*Internet Protocol Security*): *IPSec* pruža šifrovanje i zaštitu podataka na nivou *IP* sloja i često se koristi za siguran tuneliran prenos podataka između *IoT* uređaja i centralnih servera.
- *WireGuard*: Relativno nov, *WireGuard* je poznat prema svojoj brzini i jednostavnosti konfiguracije, što ga čini popularnim izborom za *IoT* uređaje sa ograničenim resursima.

Ovaj deo rada je dao osnovni pregled uloge *VPN*-a u *IoT* ekosistemu i istakao neke od ključnih protokola koji se koriste. U narednom poglavlju ćemo razmotriti hardver koji je korišćen za realizaciju ovog projekta.

IV. HARDVER ZA REALIZACIJU IOT VPN PROJEKTA

A. Raspberry Pi kao osnovni uređaj

Za realizaciju ovog projekta, za infrastrukturnu osnovu korišćen je *Raspberry Pi*, jednokartični računar male veličine i niskih energetskih zahteva. *Raspberry Pi* je popularan izbor u *IoT* projektima zbog svoje pristupačnosti i široke podrške od strane zajendice.

B. Specifikacije korišćenog Rasberry Pi računara

Za realizaciju ovog projekta korišćen je *Raspberry Pi 4 Model B*, koji nudi značajne resurse za izvođenje *VPN* servera i druge *IoT* funkcionalnosti. U nastavku su date osnovne specifikacije ovog modela:

- Procesor: *Raspberry Pi 4 Model B* poseduje četvorojezgarni procesor ARM Cortex-A72 sa 1,5 GHz, što obezbeđuje dovoljno računarske snage za zahtevne *IoT* aplikacije.
- Radna memorija: Za bolje performanse, korišćen je model sa 4 GB RAM-a, što je dovoljno za pokretanje *VPN* servera i drugih servisa.
- Povezivost: *Raspberry Pi Model B* ima ugrađen *Ethernet* port, *WiFi* podršku i *Bluetooth*, što omogućava različite načine povezivanja sa drugim uređajima u *IoT* mreži.
- *USB* portovi: Ovaj model ima četiri *USB* 3.0 porta koji su korisni za povezivanje dodatnih uređaja, poput spoljnih diskova.

C. Operativni sistem

Kao operativni sistem, korišćen je *Raspbian*, prilagođena verzija *Linux* distribucije *Debian* za *Rasberry Pi*. *Raspbian* pruža podršku za širok spektar aplikacija i alatki potrebnih za postavljanje *VPN* servera i upravljanje *IoT* uređajima.

D. Dodatna oprema

Pored *Raspberry Pi* računara, korišćena je i dodatna oprema kao što su napajanje i mikro *SD* kartica za skladištenje operativnog sistema.

U ovom poglavlju su detaljno razmatrane hardverske komponente koje su korišćene za realizaciju *IoT VPN* projekta. U narednom poglavlju bavićemo se detaljnim objašnjenjem portova i njihove uloge u ovom kontekstu [2].

V. PORTOVI I MREŽNA KONFIGURACIJA

U ovom poglavlju, detaljno ćemo razmatrati ulogu portova i mrežne konfiguracije u okviru našeg *IoT* projekta. Portovi igraju ključnu ulogu u usmeravanju komunikacije između uređaja i omogućavaju identifikaciju aplikacija koje se koriste za razmenu podataka.

A. Portovi u internet komunikaciji

Portovi predstavljaju logičke veze između različitih servisa i aplikacija koje razmenjuju podatke putem interneta. Svaki port je označen celim brojem u opsegu od 0 do 65535. Ovaj opseg se deli na tri glavne kategorije:

- Opšti poznati portovi (0-1023): Ovi portovi se koriste za široko prepoznatljive aplikacije i protokole kao što su *HTTP* (port 80), *SMTP* (port 25) i mnogi drugi.
- Korisnički portovi (1024-49151): Korisnički portovi se koriste za potrebe koje projektanti i kompanije registruju za svoje aplikacije ili servise.
- Dinamički portovi (49152-65535): Ovi portovi su namenjeni za dinamičku alokaciju i računar ih automatski dodeljuje tokom sesije.

B. Prosleđivanje portova

Prosleđivanje portova je ključna tehnika u mrežnoj konfiguraciji koja omogućava pristup računarima unutar privatnih mreža sa računara izvan tih mreža. Na primer, kada želimo da se povežemo sa računarom unutar naše kućne mreže putem *Remote Desktop* aplikacije, šaljemo zahtev na *IP* adresu rutera (*Default Gateway*) zajedno sa odgovarajućim portom (npr. 3389 za *Remote Desktop*). Ruter, kada primi ovaj zahtev, mora biti konfigurisan da prosledi zahtev na odgovarajući računar unutar mreže [3].

C. Konfiguracija portova za IoT VPN

U okviru ovog projekta, konfigurisan je port *forwarding* na uređaju koji se koristi kao *VPN* server (*Raspberry Pi*). Ovo omogućava da usmeravamo saobraćaj sa određenih portova prema odgovarajućim uređajima u našoj *IoT* mreži. Na primer, ako želimo da omogućimo daljinsko upravljanje *IoT* uređajem putem određene aplikacije, konfigurisaćemo ruter da prosleđuje zahteve na određen port prema *IP* adresi tog *IoT* uređaja. Korisnik može da konfiguriše port koji će biti korišćen za *VPN* server. Ovde je važno napomenuti da port može biti promenjen, ali mora ostati unutar opsega dinamičkih portova. Uobičajeno, nema posebne razlike između portova koje možemo odabrati pa je najbolje ostaviti podrazumevan port, osim ako postoji poseban razlog za promenu.

U sledećem poglavlju će biti detaljno opisan proces konfiguracije portova i mrežne infrastrukture koja je korišćena u ovom projektu kako bi bilo osigurano pravilno funkcionisanje *IoT* uređaja preko *VPN* veze.

VI. POSTAVLJANJE LOKALNOG VPN SERVERA KORISTEĆI RASPBERRY PI I PIVPN

U šestom poglavlju ovog projekta, detaljno razmatramo proces postavljanja lokalnog *VPN* servera pomoću *Raspberry Pi* uređaja i *PiVPN* alata. Ova sekcija pruža dublji uvid u sve korake projekta istražujući neophodne komponente, korake realizacije, i demonstrirajući funkcionalnost projekta u praksi.

A. PiVPN alat za postavljanje i upravljanje VPN serverom na Raspberry Pi uređaju

PiVPN alat olakšava transformaciju *Raspberry Pi* uređaja u funkcionalni *VPN* server, pružajući mogućnost sigurnog povezivanja na internet sa udaljenih lokacija. U nastavku je dato nekoliko ključnih karakteristika *PiVPN*-a:

- Podržani protokoli: *PiVPN* podržava dva glavna protokola otvorenog koda: *WireGuard* i *OpenVPN*. Oba protokola omogućavaju siguran prenos podataka putem *VPN*-a.
- Jednostavna instalacija: Instalacija *PiVPN*-a relativno je jednostavna i često se može izvršiti pomoću jedne komande.
- Konfiguracija klijenata: *PiVPN* omogućava lako dodavanje i upravljanje *VPN* klijentima.
- Bezbednost i privatnost: *VPN* server postavljen pomoću *PiVPN*-a pruža sigurnu i privatnu vezu sa internetom.
- Administrativni pristup: *PiVPN* obezbeđuje jednostavan način za administriranje servera, uključujući praćenje aktivnih klijenata i pravljenje rezervnih kopija konfiguracionih datoteka.

B. Instalacija operativnog sistema, PiVPN-a i konfiguracija servera

Prvi korak je instalacija *Raspberry Pi imager* alata, koji pojednostavljuje proces instalacije operativnog sistema na *Raspberry Pi* uređaj. Nakon što je odabran operativni sistem i memorijska kartica, treba prilagoditi osnovna podešavanja, uključujući korisničko ime, šifru računara i mrežna podešavanja. Nakon uspešne instalacije operativnog sistema na memorijsku karticu, potrebno je staviti je u *Raspberry Pi* uređaj i pokrenuti inicijalizaciju operativnog sistema. Upotrebom `ip -a` komande saznajemo *IP* adresu uređaja, koja će biti ključna za daljinsko povezivanje. Nakon ovog koraka potrebno je odlučiti da li želimo da dodelimo statičku ili dinamičku *IP* adresu serveru. Server će podrazumevano imati dinamičku adresu koja je dodeljena *Raspberry Pi* - u prilikom inicijalnog povezivanja putem *Ethernet* kabla. Ukoliko želimo statičku adresu, možemo odabrati opciju "*NO*" i uneti željenu adresu. Ova statička *IP* adresa će odrediti koja će adresa uređaja biti vidljiva na mreži kada korisnici preko *VPN* servera pristupaju internetu. Takođe, ova *IP* adresa će

biti nova *IP* adresa *Raspberry Pi* - a koju ćemo koristiti za dalje povezivanje putem *PuTTY* - a ili drugih alata. Kako bismo instalirali najnovije ažuriranje sistema potrebno je koristiti komande poput `sudo apt update` i `sudo apt upgrade`. Zatim instaliramo *PiVPN* koristeći komandu `curl -L https://install.pivpn.io | bash` [4].

C. Odabir protokola

Nakon inicijalnih podešavanja, sledeći korak je odabir protokola za tunelovanje između *OpenVPN* i *WireGuard*. Oba protokola obavljaju istu osnovnu funkciju, ali se razlikuju u načinu enkripcije i performansama. *OpenVPN* koristi *SSL* protokol za enkripciju, dok *WireGuard* pruža brže performanse i manju potrošnju energije. *WireGuard* kosirti *Noise protocol framework* i niz drugih algoritama za bezbednu enkipricju. U ovom primeru izabrana je *WireGuard* opcija zbog njihovih prednosti i prateće aplikacije koja olakšava korišćenje *VPN* servera [4].

D. Dodavanje VPN klijenata

Sledeći korak je dodavanje *VPN* klijenata koji će koristiti *VPN* server. Klijenti mogu biti različiti uređaji koji imaju mogućnost povezivanja na internet. Dodavanje klijenata se vrši putem terminala koristeći komandu `pivpn -a`. Svaki klijent se identifikuje unesenim imenom. Za svakog klijenta se generišu posebni konfiguracioni fajlovi. U *IoT* sistemima, dobra praksa je da svaki uređaj bude jedan klijent kako bi se obezbedila bolja izolacija i bezbednost sistema [4].

E. Prosleđivanje porta

Nakon uspešne konfiguracije servera, potrebno je omogućiti komunikaciju sa internetom putem odabranog porta. Da bi zahtevi i paketi sa interneta prolazili kroz *VPN* server, neophodno je konfigurisati ruter da prosleđuje seobraćaj sa tog porta ka *Raspberry Pi* uređaju. Postupak konfiguracije može varirati u zavisnosti od modela rutera ali se svodi na omogućavanje saobraćaja prema unapred odabranom portu [4].

F. Administrativni pristup

PiVPN pruža podršku za administrativni pristup *VPN* serveru, posebno za one koji su odgovorni za upravljanje i održavanje servera. Ovaj pristup omogućava različite privilegije kao što su pregled korisničkih podataka i upravljanje klijentima. Za pregled trenutno povezanih klijenata i njihovih informacija, koristi se komanda `pivpn -c`. Takođe je moguće kreiranje rezervnih kopija servera i to se postiže komandom `pivpn -bk`. Dodatno, komanda `pivpn -d` omogućava pokretanje *Wizard* - a za debagiranje podešavanja u slučaju da klijent nije u mogućnosti da se poveže sa *VPN* serverom iz nekog razloga. Sve navedene komande, zajedno sa drugim komandama koje pruža *PiVPN*, olakšavaju administraciju *VPN* servera i održavanje sigurne i pouzdane *VPN* infrastructure [4].

VII. ZAKLJUČAK

Internet of Things predstavlja sveprisutnu i brzo rastuću oblast koja obuhvata širok spektar uređaja i aplikacija. U ovom kontekstu, sigurnost i privatnost podataka postaju ključni izazovi. Implementacija *VPN* rešenja na *Raspberry Pi* uređaju omogućava siguran i zaštićen kanal za komunikaciju između *IoT* uređaja i centralnog servera. Ovaj pristup pomaže u očuvanju integriteta i poverljivosti podataka koji se razmenjuju između uređaja, čime se smanjuje rizik od napada i neovlašćenog pristupa. Kroz istraživanje *PiVPN*-a, prepoznajemo značajnu ulogu koju *VPN* tehnologija igra u *IoT* ekosistemu. Bezbednost i privatnost podataka postaju prioriteti za kompanije i pojedince koji se bave *IoT* uređajima. Implementacija *VPN* rešenja na *Raspberry Pi* uređaju pruža praktičan način da se ovi izazovi prevaziđu. U budućnosti, očekuje se dalji razvoj *VPN* tehnologije i njenih primena u *IoT* sektoru. Sa sve većim brojem *IoT* uređaja i sveobuhvatnim prihvatanjem ovih tehnologija, potreba za sigurnim i skalabilnim *VPN* rešenjima će rasti. Takođe, očekuje se da će se pojaviti inovativne aplikacije koje će iskoristiti *VPN* za dodatnu zaštitu *IoT* uređaja i podataka.

ZAHVALNICA

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

LITERATURA

[1] IoT Technologies Explained: History, Examples, Risks & Future, av.at 20.9.2023, https://www.visionofhumanity.org/what-is-the-internet-of-things/

[2] Derek Molloy, “Raspberry Pi”, New York, 2016.

[3] What Is Port Forwarding? - A Beginner's Guide, av. at 20.9.2023., https://learn.g2.com/port-forwarding

[4] PiVPN / WireGuard Complete Setup 2022, av. at 20.9.2023. https://www.crosstalksolutions.com/pivpn-wireguard-complete-setup-2022/

Razvoj sigurnosnog sistema zaključavanja uz pomoć Arduino okruženja i 4x4 *Keypad*-a

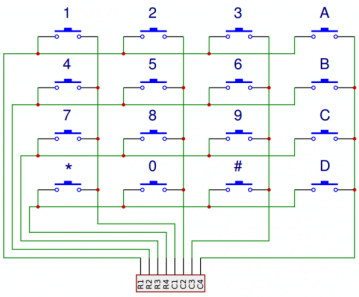
Petar Zdravković

Sadržaj – U ovom radu je opisan rad sigurnosnog sistema koji se deaktivira unosom adekvatne lozinke, a u slučaju neispravnog unosa, sistem aktivira alarm koji vlasnika obaveštava o pokušaju provale. Rad je realizovan u simulatoru.

I. Uvod

Kroz istoriju društva čovek je imao potrebu da zaštiti svoj dom, kao i svoje dragocenosti, te dolazi najpre do razvoja vrata, a zatim i prvih brava. Brava [1] je mehanička ili elektronska naprava koja služi za zaključavanje vrata i koja se otključava fizičkim objektom (ključ, kartica…), unošenjem tajne informacije ili kombinacije ova dva. Od podupiranja grane ili motke na vratima koje se prvo javlja u Asirskom carstvu, preko prvih drvenih brava u starom Egiptu, prvih metalnih brava u starom Rimu, kroz razvoj tehnologije u savremenom dobu pojavljuju se magnetne brave, sistemi koji se otključavaju šifrom, otiskom prsta, glasom i slično. Jedan takav sistem opisan je u ovom radu, koji koristi lozinku kao metod otključavanja, prikaz iste se vrši na LCD displeju, a sam unos se vrši na *4x4 Matrix Membrane Keypad*-u.

Ova komponenta [2] funkcioniše na sledeći način: ispod svakog tastera nalazi se membranski prekidač. Svaki prekidač u jednom redu povezan je sa ostalim prekidačima u tom redu putem vodova ispod kontaktnih tačaka. Svaki prekidač u jednoj koloni povezan je na isti način - jedna strana prekidača povezana je sa svim ostalim prekidačima u toj koloni putem vodova (slika 1). Svaki red i kolona izvode se na jedan pin, što čini ukupno 8 pinova na tastaturi od 4x4. Pritiskanjem tastera zatvara se prekidač između voda u koloni i voda u redu, omogućavajući struji da teče između pina kolone i pina reda.



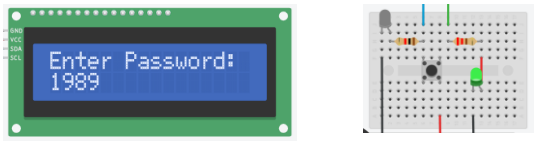
Slika 1. Šema 4x4 *Keypad*-a.

P. Zdravković je učenik četvrtog razreda prirodno-matematičkog smera Gimnazije „Bora Stanković“, Vožda Karađorđa 27, 18000 Niš. E-mail: petarzdravkovic73@gmail.com

II. Princip rada sistema

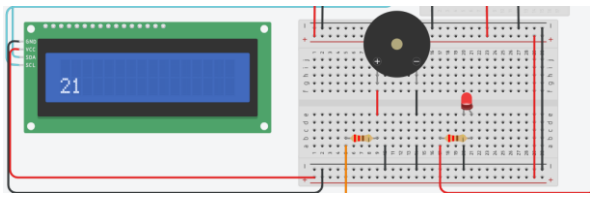
Nakon pokretanja, sistem od nas zahteva da unesemo određenu lozinku. Sama dužina lozinke je u kodu unapred definisana, u ovom slučaju 4 karaktera. Ova dužina lozinke daje 16^4, odnosno oko 65.5 hiljada kombinacija što je dovoljno dobro za potrebe zaključavanja ulaznih vrata kuće/zgrade i dovoljno kratko da se lozinka lako upamti. Naravno, po potrebi, moguće je napraviti dužu lozinku. Nakon što taj definisani broj karaktera bude dostignut proverava se tačnost unete lozinke. Svaki pojedinačni karakter se ispisuje na LCD displeju.

Postoje dve mogućnosti. Prvo, ako je uneta šifra tačna, sistem nas o tome obaveštava porukom na displeju i otključava „bravu” (uključuje se se zelena LED dioda) (slika 2).



Slika 2. Tačno uneta lozinka.

Druga mogućnost jeste da je uneta lozinka netačna. U tom slučaju nas sistem u vidu poruke na displeju o tome obaveštava i otpočinje tajmer u trajanju od 30 sekundi. U tom vremenskom intervalu potrebno je uneti odgovarajući deaktivacioni kod koji će stopirati tajmer i resetovati ceo sistem. U slučaju da tajmer odbroji do nule, a odgovarajući deaktivacioni kod nije unet dolazi do aktivacije alarma u vidu piezo zujalice (slika 3).



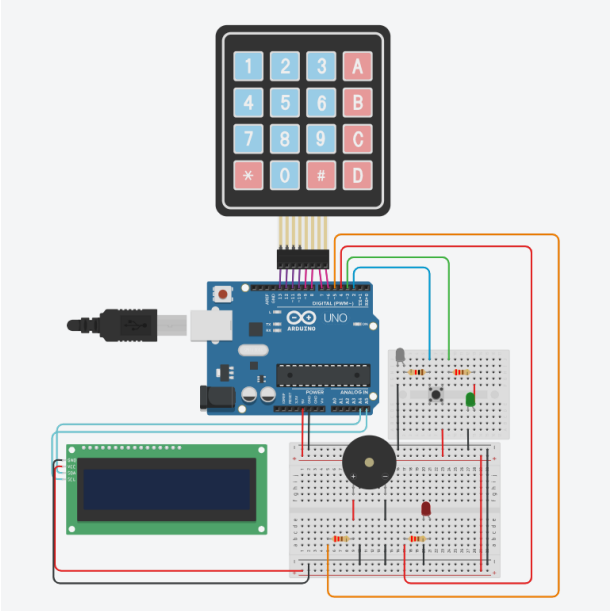
Slika 3. Netačno uneta lozinka (odbrojavanje tajmera).

III. Realizacija sistema

A. Električna šema sistema

Električna šema (slika 4) je skicirana u logičkom simulatoru Tinkercad [3]. Na šemi se mogu uočiti dve

centralne komponente, *4x4 keypad* I *LCD 16x2 (I2C) display* [4]. Ovaj displej je korišćen iz razloga što zahteva manji broj pinova na Arduinu (samo 2) od klasičnog 16x2 displeja koji zahteva čak 6 pinova, što je previše s obzirom na to da je 8 pinova već iskorišćeno za *4x4 Keypad*. Obe navedene komponente su adekvatno povezane na *Arduino Uno* mikrokontroler i preko njih se vrši unos i prikaz podataka. Osim dela za unos podataka postoje dve celine koje dobijaju signal u zavisnosti od tačnosti unete lozinke. Prvu celinu koja se aktivira u slučaju tačno unete lozinke čine zelena LED dioda koja predstavlja otključanu bravu, kao i mehanički taster za resetovanje sistema nakon tačno unete šifre, odnosno ponovnog zaključavanja. Naravno, obe navedene komponente su povezane na *Arduino*. Poslednju celinu čine crvena LED dioda koja se pali prilikom netačno unete lozinke, kao i piezo zujalica koja se oglašava nakon isteka tajmera na LCD displeju.



Slika 4. Električna šema sistema.

B. Kod za Arduino okruženje

U nastavku su prikazani ključni delovi koda ispisanog takođe u Tinkercad simulatoru. U programu je implementiran prethodno opisan način rada sistema. Propratnim tekstom su pojašnjeni određeni delovi koda.

```
//Definisanje podataka potrebnih za unos lozinke
char Data[Password_Length];
char Master[Password_Length] = "1989";
byte data_count = 0, master_count = 0;
char Dtry[Deact_Length];
//Definisanje podataka potrebnih za unos
deaktivacionog koda
char Dmaster[Deact_Length] = "13";
int t = 30;
char customKey;
const byte ROWS = 4;
const byte COLS = 4;
```

```
//Provera da li je taster pritisnut, čuvanje
podataka sa pritisnutog tastera u Data i njegovo
štampanje
customKey = customKeypad.getKey();
  if (customKey) {
Data[data_count] = customKey;
lcd.setCursor(data_count, 1);
    lcd.print(Data[data_count]);
    data_count++;                    }
//Obavljanje određene radnje u zavisnosti od
tačnosti lozinke
    if (!strcmp(Data, Master)) {
      lcd.print("Correct");
      status = HIGH;
      digitalWrite(signalPin, status);
    } else {
      lcd.print("Incorrect");
      digitalWrite(errorPin, HIGH);
      delay(2000);
      lcd.clear();
      lcd.print("Deactivate code");
      delay(1000);
      lcd.clear();
      data_count = 0;

      while (t >= 0) {
        lcd.setCursor(0, 1);
        if (t < 10) {
          lcd.print("0");
          lcd.setCursor(1, 1);
        }

        //tajmer
        lcd.print(t);
        t = t - 1;
        delay(1000);
        if(checkDeactivationCode()){
          check = true;
          break;
        }
      }
```

IV. ZAKLJUČAK

U ovom radu je prikazan sistem sigurnosnog sistema zaključavanja koji se otključava unošenjem odgovarajuće lozinke na *4x4 Keypad*-u. Rad je realizovan u logičkom simulatoru i još uvek nije implementiran u realnom sistemu. Naravno da bi se mogao primeniti u realnom sigurnosnom sistemu potrebne su određene modifikacije i nadogradnje. Pre svega zamena zelene diode za određeni sistem koji će zaključati vrata, mehanički ili magnetni. Nakon toga ovakav sistem se može koristiti za zaključavanje vrata, ormarića, sefa i sl.

LITERATURA

[1] Kratka istorija razvoja brave: http://bravarnonstop.rs/kratka-istorija-razvoja-brave/
[2] Tehnička dokumentacija za *4x4 Matrix Membrane Keypad* dostupno na: https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/
[3] Logički simulator Tinkercad: https://www.tinkercad.com/
[4] Tehnička dokumentacija za *LCD 16x2 (I2C) display* dostupno na: http://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

Sistem automatskog zalivanja biljaka

Uroš Smadulović, Luka Minić

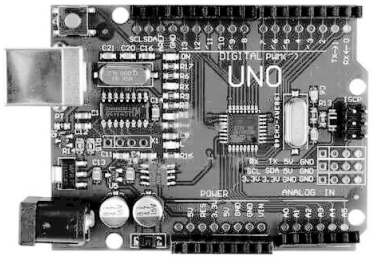
Sadržaj - Zalivanje biljaka može biti prilično naporno i zahtevno, posebno kada se radi o većem broju biljaka koje zahtevaju redovnu negu. Automatsko zalivanje biljaka može biti rešenje za ovaj problem, jer omogućava precizno navodnjavanje i uštedu vremena. U ovom radu opisuje se projekt automatskog zalivanja biljaka koji koristi mikrokontroler, pumpu za vodu i senzor za vlagu. Cilj ovog projekta je bio da se olakša proces navodnjavanja biljaka i da se osigura da biljke uvek dobijaju optimalnu količinu vode.U ovom radu ćemo detaljno opisati dizajn i implementaciju ovog projekta, kao i rezultate koje smo dobili.

I. Princip rada

Automatsko zalivanje biljaka je proces koji uključuje kontrolu vode u biljnom uzgoju. To može biti izazovno jer prekomerna ili nedovoljna količina vode može oštetiti biljke i smanjiti njihovu produktivnost. Korišćenje automatizacije za održavanje optimalne vlažnosti tla i biljaka može pomoći u prevenciji ovih problema.Za ovaj projekt smo koristili *Arduino Uno* [1] mikrokontroler, koji je odličan izbor za jednostavne i srednje složene projekte, kao što je ovaj. Mikrokontroler je povezan sa senzorom vlage [2] i pumpom za vodu [3], kako bi se u realnom vremenu očitavale vrednosti i po potrebi zalivalo biljke. Pored toga, programski kod je napisan u *Arduino IDE* [4], besplatnom i otvorenom izvoru, koji pruža mnoge korisne funkcije i alate za razvoj programa.

II. Praktična realizacija

A. Arduino Uno mikrokontroler



Slika 1. Arduino Uno mikrokontroler.

Arduino Uno [1] mikrokontroler (slika 1) je mikrokontroler koji se često koristi u projektima kao što je ovaj. To je mali računar koji se može programirati da izvršava različite zadatke. Arduino Uno se sastoji od mikrokontrolera ATmega328P, serijskog porta, USB porta, 14 digitalnih ulazno/izlaznih pinova, 6 analognih ulaznih pinova i kristala od 16 MHz.

U. Smadulović i L. Minić, učenici II razreda Tehničke škole u Zaječaru, smera Informacione tehnologije, Zaječar, Srbija. E-mail: smadulovicuros@gmail.com

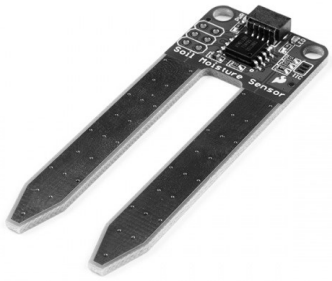
B. Pumpa za vodu



Slika 2. Pumpa za vodu.

Pumpa za vodu, prikazana na slici 2, se koristi za transportovanje vode do biljaka [2]. U ovom projektu se koristi mala pumpa koja može lako da se poveže na mikrokontroler. Kada je vreme da se biljka zalije, mikrokontroler pokreće pumpu za vodu. Voda se zatim transportuje do biljke i tako biljka dobija potrebnu količinu vode. Sastoji se od malog elektromotra i „propelera“ koji gura vodu kroz odredjenu cev.

C. Senzor vlage

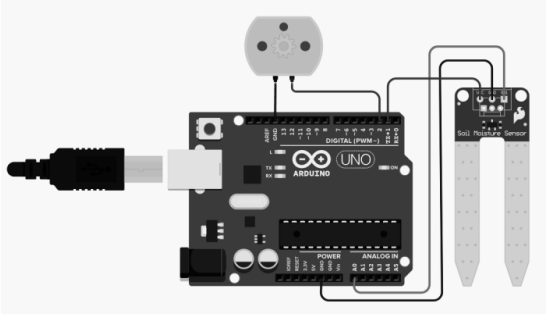


Slika 3. Senzor vlage.

Senzori za praćenje vlažnosti zemljišta [2] se često koriste u različitim aplikacijama i konstruišu se pomoću dva metalna provodnika zakopana u tlu (slika 3). Ovi provodnici oblikuju elektrode koje su postavljene na određenoj dubini u zemljištu. Kada zemljište postane vlažno, njegova električna otpornost se menja. Ta promena otpornosti se zatim meri i registruje. Senzori za vlažnost zemljišta funkcionišu tako što merenjem ove otpornosti daju indikaciju o trenutnom nivou vlažnosti. Ovi senzori često koriste analogni ili digitalni izlaz kako bi preneli te

podatke mikrokontroleru ili drugim uređajima koji upravljaju procesom zalivanja.

D. Opis rada



Slika 4. Električna šema.

Sistem radi tako što mikrokontroler u određenom intervalu proverava da li je zemlja vlazna. Zatim, ukoliko je zemlja suva, uključuje pumpu. Interval između dva zalivanja se podešava na osnovu potreba biljke. Na električnoj šemi (slika 4) mozemo videti da je negativni kraj senzora za vlagu povezan na pin GND, pozitivni kraj je povezan na pin 1, a S pin koji šalje podatke senzora je povezan na A0 pin, pošto je to analogni signal. Masa pumpe za vodu je povezana na GND, a pozitivan kraj je povezan na pin 2, time što su pozitivni krajevi vezani za određene pinove, umesto na direktan napon, možemo da kontrolišemo njihovo ulkjučenje i isključenje odvojeno od mikrokontrolera.

E. Kratak opis blokovskog dijagrama koda

Poćinjemo tako što unosimo „promenljivu“, odnosno analognu vrednost pina A0 koja nam daje vrednost senzora vlage (slika 5). Zatim proveravamo da li vrednost senzora odgovara optimalnoj vrednosti (vrednost se prostire od 0 do 850, a optimalna je ako je veća od 750), ukoliko nije optimalna,uključujemo pumpu, ukoliko jeste, isključujemo pumpu i čekamo određeni interval da ne bi smo proveravali svakog trenutka i time nepotrebno trošili struju.

F. Kod

```
{
  Serial.begin(9600);
  pinMode(2, OUTPUT); // pin 2 je izlazni - kontrolise
pumpu
  pinMode(1, OUTPUT); // pin 3 je izlazni - kontrolise
senzor
}
void loop()
{
  digitalWrite(1,HIGH); // uključujemo senzor vlage
  delay(10); //dajemo mu 10 milisekundi da se pokrene
  Serial.println(analogRead(A0));
  while(analogRead(A0) < 750) {//ulazimo u petlju sve dok
vrednost nije optimalna
   digitalWrite(2,HIGH); // uključujemo  pumpu
  }
  digitalWrite(2,LOW); // isključujemo pumpu
  digitalWrite(1,LOW);// isključujemo senzor
  delay(H*60*60*1000); // cekamo odredjeni interval
}
```



Slika 6. Izgled završenog projekta.

III. ZAKLJUČAK

Automatsko zalivanje biljaka pomoću Arduino Uno mikrokontrolera, pumpe za vodu i senzora za vlažnost zemlje je koristan projekat za sve one koji vode računa o biljkama i žele da se pobrinu da imaju dovoljno vode. Ovaj sistem je jednostavan za montiranje i programiranje, a omogućava efikasno upravljanje zalivanjem biljaka. Sistem se može nadograditi dodavanjem drugih senzora i uređaja, što ga čini fleksibilnim i prilagodljivim za različite potrebe

Testiranje sistema pokazalo je da se automatsko zalivanje biljaka može izvesti sa ovim sistemom na efikasan način. Sistem radi pouzdano i precizno, a upravljanje njime je jednostavno I fleksibilno. Ovo čini sistem korisnim za upotrebu u kućnoj upotrebi ili u manjim vrtovima.

ZAHVALNICA

Autori se zahvaljuju mentoru, Dejanu Nikoliću, dipl.ing. elektrotehnike, profesor Tehničke škole u Zaječaru.

LITERATURA

[1] Arduino Uno, Dostupno na https://docs.arduino.cc/hardware/uno-rev3
[2] Senzor vlage, Dostupno na: https://projecthub.arduino.cc/nikolaiapalis/simple-soil-moisture-sensor-ec23c7
[3] Pumpa za vodu, Dostupno na: https://www.crompton.co.in/blogs/pumps/water-pump-working-and-types/
[4] Arduino IDE, Dostupno na https://www.arduino.cc/en/software

Projektovanje *Cut the Rope* mašine korišćenjem razvojnog okruženja sa mikrokontrolerom ARM arhitekture

Marija Prijić, Beba Mladenović

Sadržaj – U ovom radu je, uz pomoću *Bluepill* razvojnog okruženja, projektovana arkadna igra *Cut the Rope Machine* (u nekim izvorima pronađena i pod nazivima *The String Cutting Machine*, *The Barber Cut Machine* ili *Cut ur Prize Machine*). Jedan taster simulira uključivanje mašine, tj. ubacivanje novčića. Ako je taster pritisnut, preko UART- a se šalje poruka „Ubačen novčić“ i igra počinje. Potenciometar služi za pozicioniranje “makaza“ iznad igračke. Drugi taster ima funkciju „*Cut*“, odnosno pokušaj presecanja konca koji drži igračku. Opseg potenciometra je podeljen na delove „promašaj“ i „pogodak“. Ako je vrednost potenciometra u opsegu „promašaj“, preko UART- a se ispisuje „Promašili ste. Pokušajte ponovo.“ i uključuje se crvena boja na diodi. Da bi se dobila igračka, potrebno je pogoditi konac tri puta tokom kojih se preko UART- a pojavljuje poruka „Pogodili ste.“ i dioda svetli zelenom bojom. Kada se treći put pogodi, uključuje se plava boja na diodi i izlazi poruka „Čestitamo, dobili ste igračku“. Aktiviranje DC motora ilustruje pokušaj presecanja konca. Posle svakog pokušaja je potrebno ponovo ubaciti novčič (pritisnuti taster).

I. Uvod

Cut the Rope mašina se prvi put pojavljuje u japanskim izvorima marta 2009. godine, na AMOA, tj. *Amusement Expo*, sajmu koji se svake godine održava u Las Vegasu, gde proizvođači mogu da predstave svoje *coin-activated arcade* mašine, odnosno igrice koje se viđaju na karnevalima ili festivalima, a koje se aktiviraju ubacivanjem novčića. Tu spadaju igre na sreću, igre bazirane na umeću korisnika, džuboksi, video igrice, i mnoge druge. Primer *Cut the Rope* mašine [1] je prikazan na slici 1.

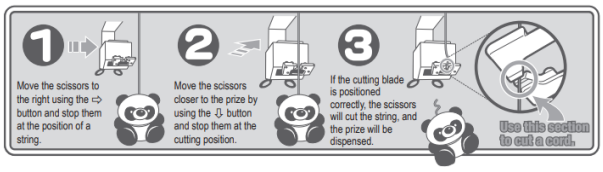
Koncipirana je kao igrica sa 10 igračaka obešenih o tanke konce, po pet sa leve i desne strane. Nakon što igrač ubaci novčić, džojstikom pomera “makaze” (slika 1) [2] levo-desno kako bi ih pozicionirao i presekao konac koji drži jednu od igračaka. *Cut the Rope* mašina je pojednostavljena varijacija poznatije igrice pod nazivom *The Claw Machine*, koja ima “kandžu” koju igrač pozicionira iznad igračke koju želi da izvuče. Glavna razlika je u tome što *The Claw Machine* ima mogućnost pomeranja kandže u četiri smera, za razliku od *Cut the Rope* mašine koja ima samo dva smera, levo i desno.

M. Prijić i B. Mladenović su studenti osnovnih akademskih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marija.prijic@elfak.rs, beba.mladenovic@elfak.rs



Slika 1. *Cut the Rope* mašina [1] i izgled makaza na mašini [2].

Nakon što igrač uspešno preseče konac, igračka pada, i igrač može da je preuzme kroz otvor u kućištu. Ako mu pokušaj ne uspe, mora da ubaci novčić da bi pokušao ponovo. Instrukcije za korišćenje mašine [3] su prikazane na slici 2. Pored plišanih ili plastičnih igračaka, o konac mogu da budu obešene različite vrste nagrada, poput poklon kartica ili zaštitnih maski za mobilne telefone.

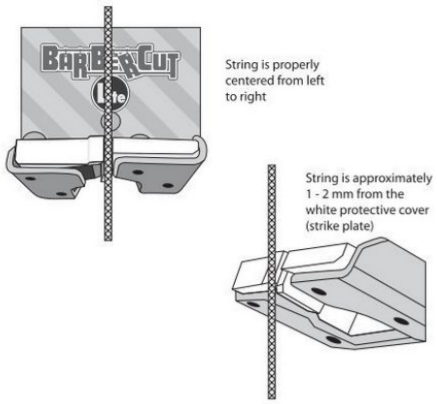


Slika 2. Instrukcije za korišćenje mašine, preuzete iz [3].

Izazov ove igrice je preseći konac koji drži igračku. Nakon ubacivanja novčića za početak igre, korisnik džojstikom pomera “makaze” levo-desno iznad igračke, pokušavajući da ih pozicionira u liniji sa koncem. Kada je zadovoljan pozicijom sečiva, korisnik pritiska taster za sečenje. Ilustracija idealne pozicije makazica u odnosu na konac [3] je data na slici 3.

U ovom radu biće prikazano kako se može projektovati replika ovog sistema korišćenjem proto ploče, različitih mikroelektronskih komponenata i ARM

mikrokontrolera, za koji je program pisan korišćenjem alata *STM32CubeMX* i *Keil Microvision*.



Slika 3. Ilustracija idealne pozicije makazica u odnosu na konac [3].

II. HARDVERSKA REALIZACIJA

A. Korišćene komponente za realizaciju projekta

U realizaciji projekta su korišćene sledeće hardverske komponente: ARM mikrokontroler, potenciometar, integrisano kolo L293D, DC motor, RGB dioda, otpornici, dva tastera, UART-USB konvertor, proto ploča i žice za međusobno spajanje komponenata.

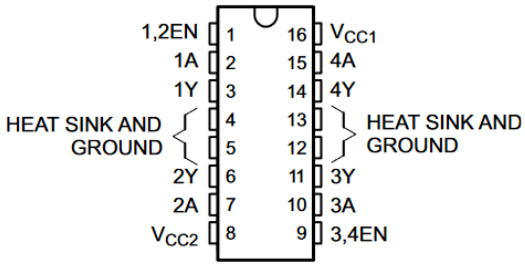
L293D je *half bridge* drajver induktivnih opterećenja kao što su releji, solenoidi, DC i bipolarni stepper motor, itd. U našem primeru se ovo kolo koristi za upravljanje DC motora koji je povezan na pinove 3 i 6 ove komponente. Pin 1 (*enable* pin) je povezan na mikrokontroler, kao i pinovi 2 i 7, pin 4 je povezan za masu, a pin 16 za napajanje od 5 V. Simbol L293D [4] drajvera sa rasporedom pinova je prikazan na slici 4.

Taster 1 i taster 2 su jednim krajem povezani na napajanje od 3.3 V, a drugim preko otpornika od 10 kΩ na masu. Oba tastera su u *pull-down* konfiguraciji. Tasteri su zatim povezani na dva susedna pina mikrokontrolera, što omogućava njihovo očitavanje. Jedan taster simulira ubacivanje novčića, a drugi simulira pokušaj presecanja konca.

Integrisana RGB dioda ima četiri priključka, tri anode (za Red, Green i Blue) i jednu katodu. Anode su povezane preko otpornika od 220 Ω na tri pina mikrokontrolera, a katoda je povezana na masu. Krajnji izvodi potenciometara su povezani na napajanje i masu, a centralni izvod na jedan od analognih ulaza mikrokontrolera.

B. Povezivanje sa mikrokontrolerom

Centralni deo sistema je ARM razvojno okruženje *Bluepill* sa STM32F103 mikrokontrolerom. Za komunikaciju između mikrokontrolera i računara koristi se UART-USB konvertorsko kolo (FTDI32 drajver). Poruke poslate do računara prikazuju se u softverskom emulatoru terminala *RealTerm*. Drajver FTDI32 komunicira sa mikrokontrolerom preko UART protokola. Pin PA2 predstavlja TX liniju mikrokontrolera koja se povezuje se RX pinom drajvera, dok se pin PA3 mikrokontrolera, koji predstavlja RX USART- a, povezuje sa TX pinom drajvera. Za indikaciju uspeha pogotka koristi se RGB dioda čija se boja kontroliše preko PWM signala generisanim tajmerom TIM2. Katoda RGB diode je povezana na masu dok su anode plave, crvene i zelene boje povezane na pinove PA15, PB3 i PB10, respektivno. Na pin PA0 je povezan potenciometar od 10 kΩ koji simulira džojstik *Cut the Rope* mašine. Taster za unos novčića je povezan na pin PA4 u *pull-down* konfiguraciji sa otpornikom od 10 kΩ. Slično, na pin PA5 povezan je taster 2 koji služi za pokretanje motora, tj. makazica. Dozvola rada drajvera L293D vrši se preko pina PA6 koji se ponaša kao PWM generator (signal se generiše preko kanala 1 tajmera TIM3). Kontrolni signali L293D šalju se sa pinova PA8 i PA9. Za projektovanje električne šeme sistema korišćen je softverski paket *Altium*. Električna šema prikazana je na slici 5.



Slika 4. Raspored pinova kola L293D [4].

III. SOFTVERSKA REALIZACIJA

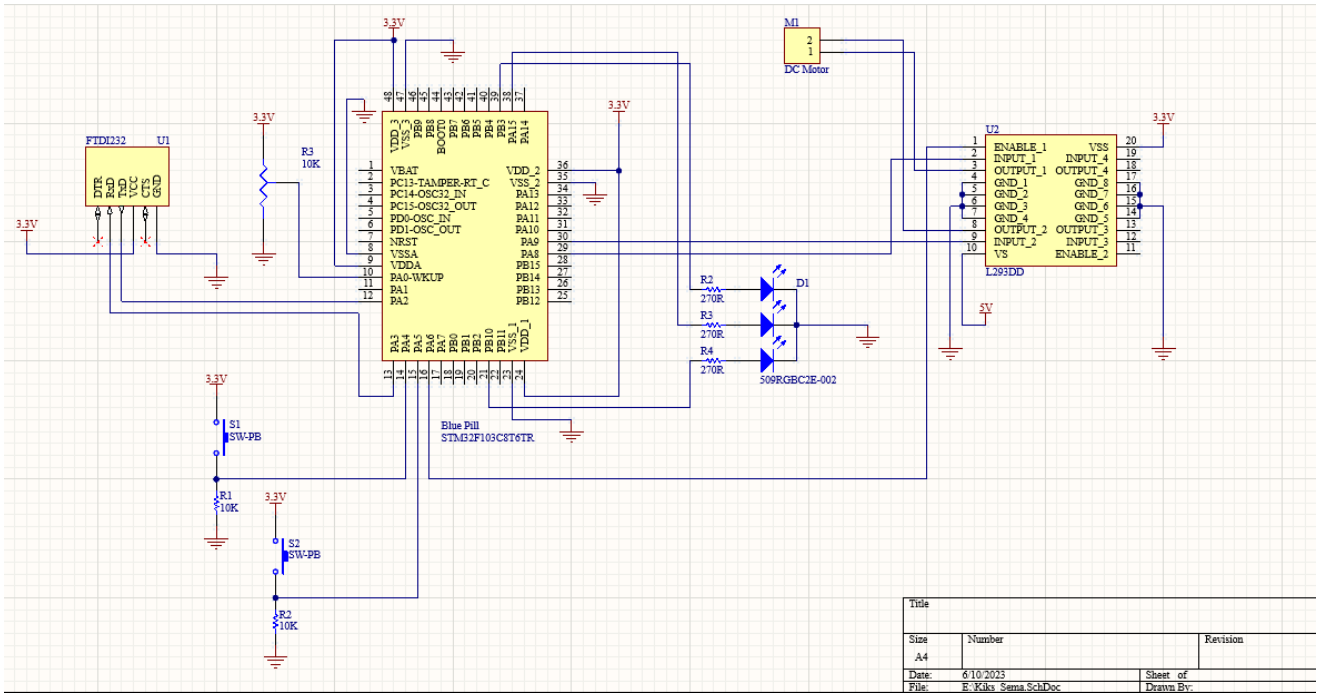
Za softversku realizaciju korišćena je ARM (*Advanced RISC Machine*) arhitektura, i u okviru nje okruženja *Keil* i *STM32CubeMX*, kao i softver *RealTerm*.

A. STM32CubeMX

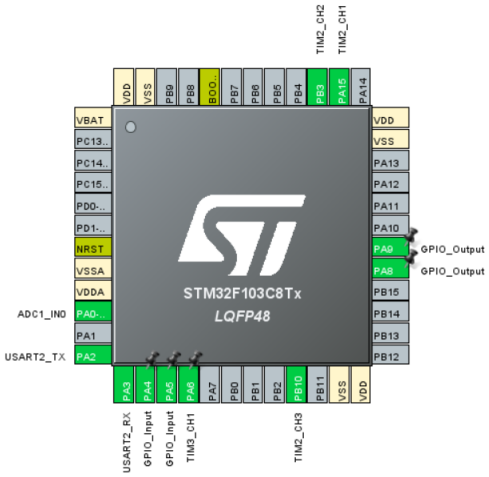
Ovaj softver omogućava podešavanje pinova mikrokontrolera, odabir serijskih protokola i periferijskih jedinica za rad. Konkretna podešavanja su uključivala podešavanje različitih pinova kao ulaznih ili izlaznih, podešavanje UART- a, AD konverzije, tajmera, i drugo. *CubeMX* u skladu sa podešavanjima generiše projekat u jeziku C za softverski alat *Keil*. Upotrebljeni pinovi mikrokontrolera su prikazani na slici 6.

Kako bismo omogućili prikaz podataka u softverskom alatu *RealTerm*, prvo treba da se uključi USART komunikacija. Ovo se radi tako što u sekciji *Connectivity* postavi USART2 da radi u *Asynchronous* režimu. Asinhroni režim nema taktni signal za sinhronizaciju izlaznih bitova sa prijemnim uređajem. Ovaj režim ima

brzinu prenosa od 115200 Bits/s, dužinu reči od 8 bitova i jedan stop bit. Nakon ovih podešavanja pinovi PA2 i PA3 su označeni za prenos podataka.



Slika 5. Električna šema realizovanog prototipa.



Slika 6. Prikaz upotrebljenih pinova mikrokontrolera.

Boja svetlosti RGB diode dobija se kombinacijom boja kojom svetle sve tri diode. Boju svake od LED, moguće je podesiti korišćenjem PWM (*Pulse Width Modulation*). U sekciji *Timers* podešeno je *PWM Generation CH1* za *Channel 1*. Ponavlja se isti postupak za kanale 2 i 3. Pinovi za *Channel 1*, *Channel 2*, i *Channel 3* su pinovi PA15, PB3, i PB10, respektivno. Glavna podešavanja PWM signala se nalaze u *Counter Settings*. Budući da je frekvencija internog takta (HCLK) 8 MHz, neophodno je postaviti vrednost *Prescaler*- a na 80-1 [5]. U vrednost se uključuje '-1' jer tajmer počinje odbrojavanje od 0. Ovim je spuštena vrednost oscilatora na 100 kHz. Svaka boja u RGB diodi je predstavljena vrednošću između 0 i 255, pri čemu 0 predstavlja odsustvo bilo kakve boje, a 255 predstavlja maksimalni intenzitet te boje. Ovo omogućava ukupno 16.777.216 mogućih kombinacija boja [6]. Kako bismo prilagodili amplitudu PWM signala, za vrednost *Counter Period*- a se postavlja 255.

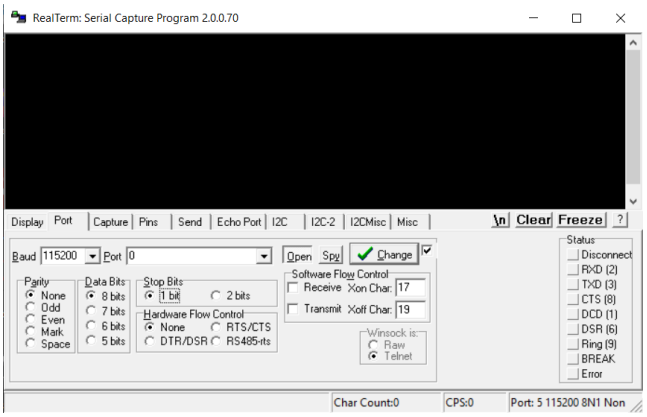
Kako bi očitali i prikazali vrednost otpornosti potenciometra neophodno je izvršiti analogno-digitalnu konverziju. Odabirom *ADC1* u sekciji *Analog* otvara se *ADC1 Mode and Configuration*. U *Mode* opciji se čekira *IN0* i time se PA0 postavlja na *Analog GPIO mode*. Dodatno, u podešavanjima uključuje se kontinuirani mod konverzije (*Continuous Conversion Mode*). Sledeća podešavanja odnose se na taster 1 i taster 2. Da bi se očitalo kada je taster pritisnut pinovi PA4 i PA5 se postavljaju kao *GPIO Input*.

Na kraju se vrše podešavanja za drajver L293D koji kontroliše pokretanje motora. Uključenje tajmera TIM3 se radi na isti način kao i prethodno pomenut TIM2. Međutim, za drajver je potreban samo jedan kanal (*Channel 1*) koji će služiti za dozvolu rada kola. Naponski nivo PWM signala koji se šalje drajveru ponaša se kao "ON/OFF" prekidač DC motora. DC motor je uključen kada je PWM signal u "*HIGH*" stanju, a isključen kada je signal u "*LOW*" stanju. Duty-cycle signala određuje brzinu motora; pošto je ona konstantna, brzina motora je takođe konstantna. Podešavanja za *Counter Settings* su: *Prescaler* 8000-1 i *Counter Period* 100-1. U polju *Mode* koje se nalazi u

podešavanjima za *PWM Generation Channel 1* potrebno je izabrati vrednost *PWM mode 1*. Inicijalni faktor ispune se podešava na 25 u polju *Pulse*. Za pokretanje i zaustavljanje motora neophodno je imati pinove koji kontrolišu njegov rad. Pinovi PA8 i PA9 su izabrani kao ovi kontrolni pinovi i zbog toga se oni postavljaju kao *GPIO Output*.

B. RealTerm

Korišćen je emulator terminala *RealTerm* (slika 7), povezan preko UART- a. Važna podešavanja ovog softvera su: brzina prenosa 115200 Bits/s i port za prenos podataka. Uloga *RealTerm*- a je ispis komandi koje se zadaju sistemu pritiskom na tastere. Njegova funkcija je prikaz poruka „Promašili ste“, „Ubačen novčić“, „Još jedan pokušaj“, i slično. Takođe u realnom vremenu prikazuje vrednosti otpornosti potenciometra.



Slika 7. RealTerm emulator terminala.

C. Keil Microvision

Softverski alat *Keil* služi za pisanje i editovanje programskog koda i uprogramiravanje izvršnog koda u mikrokontroler. Kod je dobijen klikom na *Generate Code* opciju nakon završetka podešavanja u *CubeMX*- u. Prvi korak u pisanju koda je inicijalizacija tajmera i AD konverzije, ujedno sa deklaracijom promenljivih koje će biti korišćene u programu. Dodatno napisana je komanda koja PWM signale tajmera TIM2 postavlja na 0, tj. osigurava da RGB ne svetli (slika 8).

```
HAL_ADC_Start(&hadc1); //inicijalizacija AD konverzije

//pocetak generacije PWM signala tajmera 3
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);

//pocetak generacije PWM signala tajmera 2
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_3);

htim2.Instance -> CCR3 = 0; //RGB ne svetli
htim2.Instance -> CCR1 = 0;
htim2.Instance -> CCR2 = 0;
```

Slika 8. Inicijalizacije tajmera i ADC- a.

Nakon ovih naredbi kreće glavni deo programa, odnosno beskonačna petlja, gde se pišu naredbe za primarnu funkcionalnost sistema. Na slici 10 je prikazana *if* struktura koja ispituje da li je taster pritisnut, tj. da li je „ubačen novčic“. Ako je taster pritisnut na *RealTerm*- u se ispisuje poruka “Ubacen je novcic“ i *enableFleg* postaje *true* (slika 9).

```
if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4)==1) //pokretanje igre pritiskom tastera
{
  htim2.Instance -> CCR3 = 0;
  htim2.Instance -> CCR1 = 0;
  htim2.Instance -> CCR2 = 0;
  HAL_UART_Transmit(&huart2, "Ubacen je novcic\r\n", strlen("Ubacen je novcic\r\n"), 10);
  enableFleg=true;
  HAL_Delay(1000);
}
```

Slika 9. Pokretanje igrice.

U sledećoj *if* strukturi ispituje se da li je *enableFleg true*, ako jeste igra počinje. Prva naredba u strukturi stavlja pročitanu vrednost potenciometra, dobijenom AD konverzijom, u predefinisanu promenljivu *pot*. Pomeranjem potenciometra igrač pomera poziciju “makazica“, vrednost *pot* obeležava tu poziciju. Ta vrednost se zatim konvertuje u string i prikazuje na terminalu. Čitanje vrednosti potenciometra je prikazano na slici 10.

```
if(enableFleg==true) //pocetak pokusaja
{
  pot = HAL_ADC_GetValue(&hadc1); //citanje vrednosti potenciometra
  sprintf(bafer, "Pot = %d \r\n", pot); //konverzija promenljive pot

  //prikaz vrednosti pot na terminalu
  HAL_UART_Transmit(&huart2, (uint8_t*)bafer, strlen(bafer), 10);
```

Slika 10. Čitanje vrednosti potenciometra.

Slično prvoj *if* strukturi, treća ispituje da li je taster pritisnut. Ako jeste, to znači da je igrač, tj. korisnik, izabrao svoju konačnu poziciju “makazica“ i završava svoj pokušaj. DC motor se uključuje narednom naredbom i simulira sečenje konca “makazicama“. Nakon 300 ms DC motor se isključuje. Uključenje i isključenje motora prikazano je na slici 11.

```
if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5)==1)
{
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1); //pokretanje DC motora
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 1);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
  HAL_Delay(300);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0); //zaustavljanje DC motora
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 0);
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
```

Slika 11. Uključenje i isključenje motora.

Pretpostavljajući da je korisnik uspeo da pogodi konac, definisano rangom *if*- a, ispituje se stanje oštećenog konca. Ako je igrač prvi ili drugi put pogodio konac na terminalu se javlja poruka „Pogodili ste.“ i RGB dioda svetli zelenom bojom. Ukoliko je korisnik prethodna dva puta pogodio konac nagrada pada i igra se završava. Na terminalu se ispisuje „Cestitamo! Dobili ste igracku!!“ i RGB dioda svetli plavom bojom. Rang za pogodak je prikazan na slici 12. Međutim, ako igrač nije pogodio rang za pogodak, program ide u *else* deo *if* strukture (slika 13).

Na terminalu se ispisuje poruka „Promasili ste. Pokusajte ponovo.“, a RGB dioda svetli crvenom bojom. Nakon toga *enableFleg* se vraća na *false* i *while* petlja se ponavlja.

```
if(pot>=0 && pot<=500 || pot>=2000 && pot<=2500) //rang za uspesno presecanje kabla
{
  if(cut<2)
  {
    htim2.Instance -> CCR1 = 0; //kanal za plavu boju
    htim2.Instance -> CCR3 = 0; //kanal za crvenu boju
    htim2.Instance -> CCR2 = 255; //kanal za zelenu boju
    cut++; //inkrementiranje broja ostecenja kabla
    HAL_UART_Transmit(&huart2, "Pogodili ste.\r\n", strlen("Pogodili ste.\r\n"), 10);
  }
  else
  {
    htim2.Instance -> CCR3 = 0;
    htim2.Instance -> CCR1 = 255;
    htim2.Instance -> CCR2 = 0;
    cut=0; //resetovanje broja ostecenja za sledeceg igraca
    HAL_UART_Transmit(&huart2, "Cestitamo! Dobili ste igracku!!\r\n", \
    strlen("Cestitamo! Dobili ste igracku!!\r\n"), 10);
  }
}
```

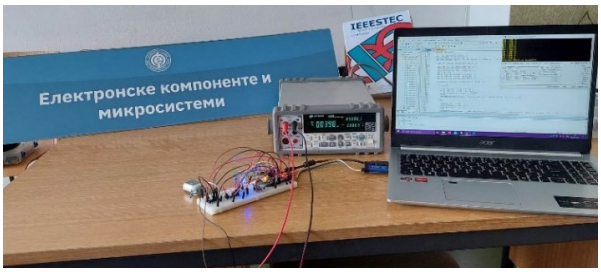
Slika 12. Rang za pogodak.

```
else //promasaj kabla
{
  htim2.Instance -> CCR3 = 255;
  htim2.Instance -> CCR1 = 0;
  htim2.Instance -> CCR2 = 0;
  HAL_UART_Transmit(&huart2, "Promasili ste. Pokusajte ponovo.\r\n", \
  strlen("Promasili ste. Pokusajte ponovo.\r\n"), 10);
}
enableFleg=false; //kraj pokusaja
```

Slika 13. Promašaj konca.

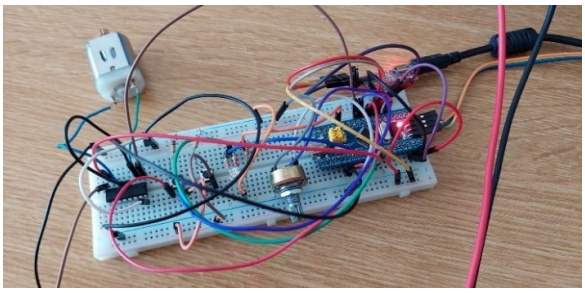
IV. PRAKTIČNA REALIZACIJA

Praktična realizacija ovog sistema prikazana je na slici 14.



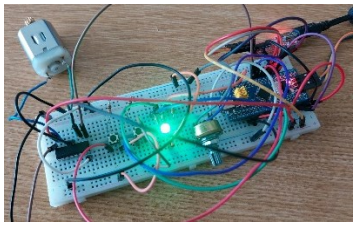
Slika 14. Praktična realizacija sistema.

Na slici 15 je prikazana praktična realizacija električne šeme sa slike 5.



Slika 15. Hardverska realizacija električne šeme.

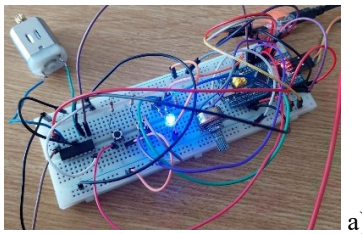
Indikacija prvog ili drugog pogotka konca (zelena boja RGB diode) prikazana je na slici 16 a), a prateća poruka na *RealTerm*- u je prikazana na slici 16 b).



a) b)

Slika 16. a) Indikacija prvog ili drugog pogotka, b) poruka o uspehu pogotka.

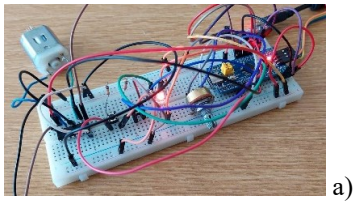
RGB dioda koja svetli plavom bojom, označavajući osvajanje nagrade, i poruka o uspehu prikazana je na slici 17.



a) b)

Slika 17. a) Indikacija za osvojenu nagradu, b) poruka čestitke o osvojenoj nagradi.

Indikacija promašaja (crvena boja RGB diode) prikazana je na slici 18 a), a prateća poruka na *RealTerm*- u je prikazana na slici 18 b).



a) b)

Slika 18. a) Indikacija promašaja, b) poruka o neuspehu pogotka.

V. ZAKLJUČAK

U ovom radu prikazana je realizacija mehanizma *arcade* mašine kojom se ubacivanjem novčića korisniku omogućava da preseče konac i dobije igračku koja je o njega okačena. Izrađene su šeme i plan realizacije, algoritam, a zatim je uređaj realizovan na proto-ploči uz pomoć mikrokontrolera i različitih mikroelektronskih komponenata čija je uloga detaljno objašnjena. Korišćeni su softverski alati *Keil Microvision* i *STM32CubeMX*. Za prikaz realizovanih akcija korišćen je softver *RealTerm*.

Ovaj sistem se može modifikovati na različite načine. Jedan primer bi bio dodavanje još nekoliko opsega potenciometra, kako bi se igraču otežalo pogađanje povoljnih vrednosti otpornosti. Takođe, RGB dioda može da se zameni klasičnom crvenom, plavom i zelenom diodom, a i zadatak može da se olakša tako što bi samo jedan pogodak značio da je igrač dobio igračku. Sa

softverske strane, moguće unapređenje bi bilo smanjenje ponavljanja koda uz pomoć funkcija.

ZAHVALNICA

Autori rada se zahvaljuju dr Milošu Marjanoviću na izdvojenom vremenu, sugestijama i pomoći pri izradi ovog projekta. Takođe se zahvaljujemo Elektronskom fakultetu na omogućenim komponentama i izvoru napajanja koji su korišćeni u izradi ovog projekta.

LITERATURA

[1] Cut ur Prize NF-P29 Barber Cut Machine for Sale. Dostupno na: https://www.neofuns.com/barber-cut-machine/

[2] BarberCut Lite. Dostupno na: https://primetimeamusements.com/product/barber-cut-lite/

[3] Barber Cut Lite Operators Manual - BMI Gaming, 2014. Dostupno na: https://www.yumpu.com/en/document/view/31128855/barber-cut-lite-operators-manual-bmi-gaming/9

[4] Tehnička specifikacija za L293D. Dostupno na: https://www.ti.com/lit/ds/symlink/l293d.pdf?ts=1686305995651&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FL293D

[5] ShawnHymel, "Getting Started with STM32 - Timers and Timer Interrupts". Dostupno na: https://www.digikey.com/en/maker/projects/getting-started-with-stm32-timers-and-timer-interrupts/d08e6493cefa486fb1e79c43c0b08cc6

[6] Jerry Mo, "How Many Colors Can a RGB LED Wall Washer Light Display", 2022. Dostupno na: https://www.linkedin.com/pulse/how-many-colors-can-rgb-led-wall-washer-light-display-jerry

Automatizacija procesa pakovanja čaja primenom SCADA sistema

Isidora Jovanović

Sadržaj – Glavni fokus ovog istraživanja je implementacija automatizovanog sistema za pakovanje čaja, uz istovremeno nadgledanje i upravljanje putem SCADA sistema. Sam proces pakovanja čaja sastoji se iz dve faze. Prvo se mleveni čaj pakuje u filter kesice, a zatim se te kesice smeštaju u aroma omotače. Za automatizaciju ovog procesa korišćen je program Citect SCADA verzija 7.10.

I. Uvod

Rad koji će u nastavku biti izložen je deo diplomskog rada autora [1], u kojem je prikazana automatizacija sistema za pakovanje čaja primenom SCADA sistema.

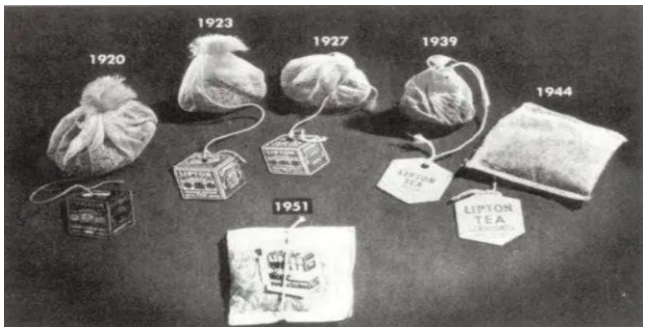
SCADA (eng. Supervisory Control and Data Acquisition) predstavlja sistem procesne automatizacije koji se koristi za prikupljanje informacija pomoću senzora koji se nalaze na udaljenim lokacijama, prenos podataka do centralne jedinice, obradu i prikaz tih podataka kao i njihovo korišćenje u svrhu nadgledanja ili upravljanja procesom [2]. Glavni zadaci SCADA sistema su da [3]:

- vrši nadzor i uspostavi upravljanje nad sistemom
- osigura očekivano ponašanje sistema
- redukuje radnu snagu primenom automatizacije
- skladišti podatke o ponašanju sistema
- obezbedi sistem uzbunjivanja koji će omogućiti evidentiranje nepravilnog funkcionisanja sistema sa centralne lokacije.

II. Proces pakovanja čaja

A. Istorija Čaja

Čaj se prvi put pominje u zapisima na području Kine koji datiraju još iz 59. godine p.n.e., mada postoje naznake da su ljudi koristili napitak od suvog lišća i vrele vode i pre toga. Početkom 20. veka pojavljuje se čaj pakovan u vrećici od svile. Kako zbog velike potražnje više nije bilo isplativo koristiti svilu za izradu vrećica, prešlo se na upotrebu gaze, zatim na papir od manilske konoplje, dok 40-ih godina 20. veka nije otkriven i patentiran filter papir koji se koristi i danas. Izgled kesica čaja kroz istoriju dat je na Slici 1, dok je na Slici 2 prikazan izgled kesice čaja danas. Filter papir nema miris ili ukus, ne rastvara se u vodi i ne menja se pod uticajem temperature, pa ni na koji način ne utiče na ukus čaja. Vrećice su ručno ušivane sve do pojave prve mašine za pakovanje čaja koju je izumeo Adolf Rahlbold 1929. godine [4].



Slika 1. Izgled kesica čaja kroz istoriju [4].



Slika 2. Kesica čaja danas; filter kesica sa končićem i vinjetom (levo) i filter kesica u aroma omotaču (desno).

B. Proces Pakovanja

U daljem tekstu biće opisan proces pakovanja mlevenog čaja, prvo u filter kesice a kasnije i pakovanje kesica u aroma omotač. Pre uključenja mašine potrebno je razvući filter papir preko odgovarajućeg kalupa kako bi zauzeo oblik levka. Takođe je potrebno razvući traku za vinjetu do valjka koji je povlači i dovesti konac do kuke koja će ga razvući do filter kesice. Vinjeta je etiketa koja se nalazi na kraju končića i služi za lakše korišćenje kesice. Kada se uključi mašina, valjak koji povlači traku za vinjetu kreće da rotira i povlači je kroz levkasti kalup koji je presavija po sredini sve dok odgovarajuća dužina trake za vinjetu ne stigne do prese. Zajedno sa valjkom i kuka kreće da razvlači konac do mesta gde će se spojiti sa filter kesicom. Za to vreme cilindri koji pokreću filter papir su ga postavili u odgovarajući položaj ispod dozera za čaj, otvara se ventil i iz rezervoara se sipa određena količina mlevenog

I. Jovanović je student master studija na modulu Automatsko upravljanje i saradnik u nastavi pri Katedri za teorijsku elektrotehniku, na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: isidora.jovanovic@elfak.ni.ac.rs

čaja. Cilindri sa grejačima lepe konac za filter kesicu onda kada senzor registruje da se kuka nalazi u odgovarajućem položaju ispred grejača. Isti senzor pokreće i presu koja lepi presavijenu traku za vinjetu i konac na drugoj strani, kao i grejače koji zatvaraju filter kesicu i sekače koji je odsecaju. Odsečenu filter kesicu preuzima hvataljka i prenosi je do levka gde otpočinje druga faza procesa.

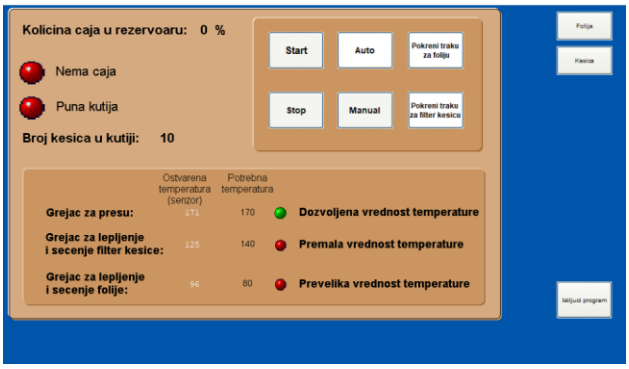
Druga faza procesa predstavlja pakovanje filter kesica u aroma omotače. Aroma omotač je folija koja služi da zaštiti kesicu od spoljnjeg uticaja i očuva aromu i svežinu čaja. Foliju sa kotura odmotavaju i presavijaju cilindri koji je povlače ispod levka onda kada hvataljka dođe u položaj iznad levka. Bočni cilindri lepe foliju stvarajući „džepove u koje upada kesica kada je hvataljka ispusti. Nakon toga se aktiviraju sekači koji odsecaju foliju i upakovana kesica čaja pada na pokretnu traku koja je prenosi do kutije. Pre nego što upadne u kutiju, upakovanu kesicu registruje senzor koji šalje impulse brojaču koji pamti broj kesica koje se nalaze u kutiji.

III. Realizacija

Program koji je korišćen za realizaciju je Citect SCADA, verzija 7.10. Prikazane su dve faze pakovanja čaja: punjenje filter kesice čajem i njeno pakovanje u celofaniranu kesicu koju zovemo aroma omotač. Ove dve faze sadrže i proces sečenja i lepljenja filter papira radi formiranja filter kesice, kao i proces sečenja i lepljenja aroma omotača.

A. Grafičke strane

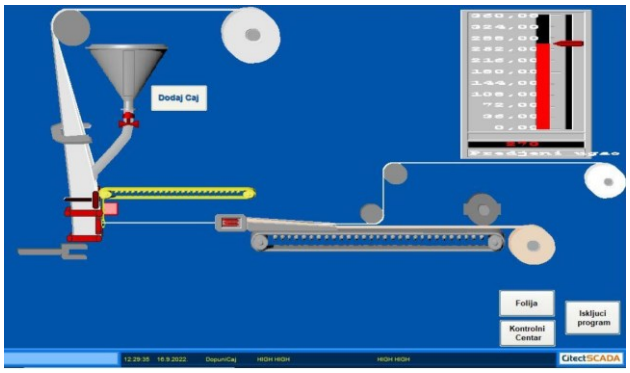
Za potrebe simulacije ovih procesa formirane su tri grafičke strane. Prva grafička strana nazvana je „Kontrolni Centar“ i njen izgled je prikazan na Slici 3. Sva podešavanja programa i definicije tagova data su u [1].



Slika 3. Izgled strane „Kontrolni Centar“ u Run režimu.

Pritiskom na taster „Start“ pokreće se mašina i omogućava odabir između automatskog i ručnog načina rada pomoću tastera „Auto“ i „Manual“. Taster „Stop“ zaustavlja izvršenje procesa ali ne prekida Run režim. „Pokreni traku za filter kesicu“ poziva funkciju *pokreniTraku()*. „Pokreni traku za foliju“ poziva funkciju *pokreniFoliju()*. Tasteri „Folija“ i „Kesica“ vode do grafičkih strana čija imena nose. Takođe na kontrolnom panelu se nalaze i led i numerički indikatori koji daju informacije o količini čaja u rezervoaru, broju kesica u kutiji kao i o vrednosti temperature grejača i njenom dozvoljenom opsegu.

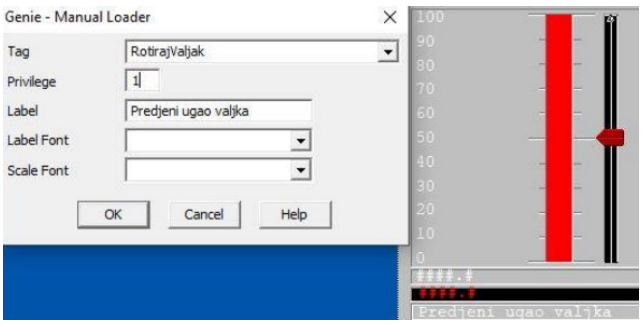
„Kesica“ je naziv grafičke strane na kojoj je prikazan proces punjenja filter kesice čajem kao i njeno lepljenje i sečenje, Slika 4. Ovaj deo mašine sadrži pokretnu traku preko koje se pomoću valjka pomera traka za vinjetu, kuku koja povlači konac, hvataljku koja prenosi filter kesicu i skup cilindara preko kojih je razvučen filter papir tako da se preklapa po sredini. Pozivom funkcije *pokreniTraku()* cilindri počinju da rotiraju, povlače filter papir i odmotavaju ga sa kotura dok istovremeno lepe slobodne strane preklopljenog papira. Na taj način filter papir dobija oblik levka, otvara se ventil i sipa se unapred određena količina čaja. Funkcija *pokreniTraku()* pokreće i valjak na pokretnoj traci, koji počinje da rotira i da svojim pljosnatim delom povlači traku za vinjetu i na taj način je odmotava sa kotura. Traka za vinjetu se pomera duž pokretne trake i dolazi do suženja u obliku levka koje je preklapa po sredini. Na izlazu iz levkastog suženja nalazi se presa koja pritiska i lepi traku po sredini i na taj način se dobija vinjeta. Za isti vremenski period kuka povlači konac i odmotava ga sa kotura sve dok ga ne dovede u krajnji položaj što registruje određeni senzor. U tom trenutku je konac razvučen između levkastog suženja, tako da se nalazi između preklopljenih strana trake za vinjetu, i krajnjeg položaja u kojem se nalaze sekač i grejači. Kada senzor registruje dolazak kuke u krajnji položaj uključuje se presa na jednom kraju koja lepi vinjetu i konac, i sekači i grejači na drugom kraju koji lepe konac za jednu stranu filter papira i istovremeno ga seku. Na taj način se dobija jedna filter kesica.



Slika 4. Izgled strane „Kesica“ u Run režimu.

Ceo proces podešen je u skladu sa pređenim uglom valjka na pokretnoj traci. Jedna filter kesica dobija se dok valjak pređe ugao od 90 stepeni. Dok valjak prelazi sledećih 90 stepeni, kuka se vraća u početni položaj na izlazu levkastog suženja i hvataljka prenosi isečenu filter kesicu do sledećeg levka, gde se vrši pakovanje u celofanirane kesice. Opisani proces se ponavlja iz početka

dok valjak rotira za još 180 stepeni. Za vreme za koje valjak izvrši jednu punu rotaciju napune se, iseku i prenesu dve filter kesice. Proces koji je opisan do sada predstavlja automatski režim rada i izvršava se kada korisnik uključi mašinu tako što pritisne taster „Start“ a zatim taster „Auto“ pa taster „Pokreni traku za filter kesicu“. Međutim, postoji i ručni režim rada koji se omogućava pritiskom na taster „Manual“. Ručni režim rada se koristi ukoliko je došlo do nekog kvara na mašini ili razdešavanja parametara koji omogućavaju automatski režim. Tada radnik koristi klizač (Slika 5) kojim zadaje pređeni ugao valjka koji povlači traku za vinjetu. Raspon vrednosti za klizač na slici 5 je od 0 do 100, međutim kada se pokrene Run režim, raspon vrednosti postaje od 0 do 360.



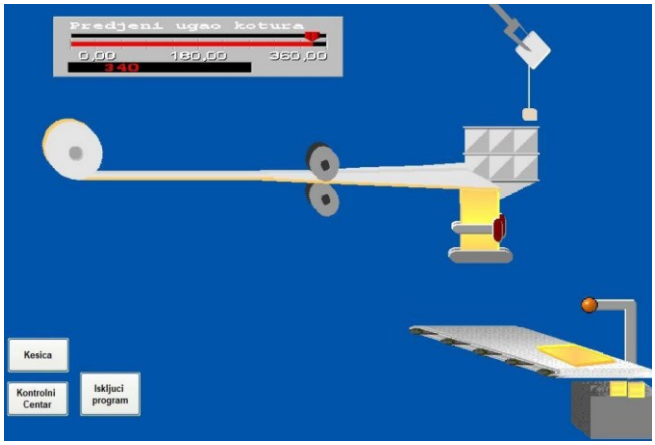
Slika 5. Klizač za ručno zadavanje pređenog ugla valjka.

Tasteri „Folija“ i „Kontrolni Centar“ koji se nalaze na grafičkoj strani „Kesica“ vode do grafičkih strana čija imena nose dok taster „Dodaj Caj“ simulira dopunu rezervoara.

Na grafičkoj strani „Folija“, Slika 6, prikazan je proces pakovanja filter kesice u aroma omotač kao i transport upakovane kesice do kutije. Deo mašine koji obavlja ovaj zadatak sadrži: kotur na koji je namotan aroma omotač (folija), cilindre koji guraju foliju, levak, bočne cilindre sa grejačima, sekač, pokretnu traku, kutiju, senzor za registrovanje dolaska upakovane kesice do kutije i klizač za ručni režim.

Kada hvataljka dođe u položaj iznad levka pokreću se cilindri koji povlače aroma omotač preko levka, bočni cilindri sa grejačem ga povlače ka pokretnoj traci dok istovremeno lepe slobodne krajeve omotača stvarajući „džep“ u koji se kroz levak spušta filter kesica nakon što je hvataljka ispusti. Cilindri sve vreme povlače traku tako da se odmah nakon upada kesice u „džep“ aktivira sekač koji odseca aroma omotač i dobija se upakovana kesica. Tako upakovana kesica pada na pokretnu traku koja je odnosi do kutije. Automatski režim rada omogućuje se tako što se pritisne taster „Start“, a zatim taster „Auto“ pa taster „Pokreni traku za foliju“. Taster „Pokreni traku za foliju“ poziva funkciju *pokreniFoliju()* koja izaziva rotaciju cilindara koji povlače omotač. Za vreme jedne rotacije cilindara i sekača dobijamo dve upakovane kesice.

Bitno je naglasiti da opisane faze pakovanja nisu nezavisne jedna od druge. Deo mašine za pakovanje u omotače ne počinje sa radom dok hvataljka ne dođe u zadati položaj. Takođe, oba dela mašine rade sve dok u rezervoaru ima čaja, osim ako radnik ne pritisne dugme „Stop“. Kada se rezervoar isprazni, mašina prestaje sa radom, oglašava se alarm koji obaveštava radnika da je potrebno dopuniti rezervoar. Radnik isključuje mašinu, puni rezervoar i ponovo je pokreće.



Slika 6. Izgled grafičke strane „Folija“ u Run režimu.

B. Cicode datoteke

Cicode datoteke predstavljaju skup korisničkih funkcija koje omogućavaju lakše upravljanje procesima. U ovom radu korišćene su dve funkcije čiji su kodovi prikazani na Slikama 7 i 8.

```
FUNCTION pokreniTraku()
KolicinaCaja=0;
Grejac=170;
Grejac1=140;
WHILE (KolicinaCaja<100 AND Start=1)DO
  i = 0;
  RotirajValjak=0;
  WHILE (i<36 AND Start=1)DO
  IF (Start=1 AND Auto=1) THEN
  SkloniPresu=0;
  RotirajValjak=RotirajValjak+10;
  Sleep(0.5);
  i=i+1;
  IF (RotirajValjak=60 OR RotirajValjak= 240) THEN
  KolicinaCaja=KolicinaCaja+5;
  Sleep(0.5);
  KolicinaCaja=KolicinaCaja+5;
  END

  IF (RotirajValjak=90 OR RotirajValjak=270) THEN
  SkloniPresu=1;
  Sleep(1);
  END

  IF RotirajValjak=360 THEN
  RotirajValjak=0;
  END
  END
  END
  END
  END
```

Slika 7. Kod funkcije *pokreniTraku()*.

```
FUNCTION pokreniFoliju()
BrojacKesica=0;
Grejac2=80;
Kesica_u_kutiji=0;
Kesica_u_kutijil=0;
Ukljuci_sekac=1;
Sleep(1);
Ukljuci_sekac=0;
RotirajValjak1=0;
WHILE (KolicinaCaja<=100 AND BrojacKesica<10) DO
j=0;

IF (RotirajValjak=180 OR RotirajValjak=360) THEN
WHILE (j<18 AND Start=1) DO
IF (Auto=1 AND Start=1) THEN
j=j+1;
Ukljuci_sekac=0;
RotirajValjak1=RotirajValjak1+10;
Sleep(0.5);

IF (RotirajValjak1=90 OR RotirajValjak1=270) THEN
Ukljuci_sekac=1;
Sleep(1);
END

 IF (RotirajValjak1=90 OR RotirajValjak1=270) THEN
 Ukljuci_sekac=1;
 Sleep(1);
 END

 IF RotirajValjak1=180 THEN
 Kesica_u_kutiji=1;
 BrojacKesica=BrojacKesica+1;
 END
 IF RotirajValjak1=360 THEN
 Kesica_u_kutijil=1;
 BrojacKesica=BrojacKesica+1;
 RotirajValjak1=0;
 END

 END
 END
 END
 END
 END
```

Slika 8. Kod funkcije *pokreniFoliju()*.

IV. Zaključak

U ovom radu prikazan je primer automatizacije procesa pakovanja čaja primenom SCADA sistema. Može se primetiti da je jedan program, kao što je Citect SCADA, primenljiv i za procese koji se ne odvijaju na velikoj udaljenosti. Pakovanje čaja odvija se na jednoj lokaciji, ali je za nadgledanje i kontrolu rada mašine koja se koristi u tu svrhu Citect SCADA dobro rešenje.

Na taj način, mašina može raditi samostalno, dok je zadatak radnika sveden na nadgledanje procesa putem indikatora na kontrolnoj tabli, koja ne mora biti smeštena u istoj prostoriji kao i sama mašina.

Zahvalnica

Autor se zahvaljuje prof. dr Saši S. Nikoliću na savetima i pomoći prilikom realizacije ovog projekta.

Literatura

[1] I. Jovanović, „Pakovanje čaja primenom SCADA sistema“, Elektronski fakultet u Nišu, 2022.
[2] https://www.automatika.rs/baza-znanja/teorija-upravljanja/uvod-u-scadu.html
[3] https://automatika.elfak.ni.ac.rs/oglasna-tabla/nastavni-materijal-menu#scada-sistemi
[4] https://cheese-gtch.ru/bs/recipes/istoriya-izobreteniya-chainogo-paketika-kak-poyavilsya-chai-v-paketikah.html

Sušara za sušenje šljiva

Bata Anđelković, Lazar Pešić, Tamara Mitrović, Igor Trajković

Sadržaj - U ovom radu opisan je jedan sklop za automatsku regulaciju temperature i vlažnosti vazduha u komori za sušenje šljiva kako bi se dobio najbolji kvalitet proizvoda.

I. UVOD

U novije vreme, javila se potreba za primenom novih tehnologija. Savremene tehnologije objedinjuju niz savremenih uređaja kao što su programabilni kontroleri, računari i tako dalje.Jedan od takvih sklopova koji objedinjuje niz pametnih uređaja je sušara za sušenje šljivakoja ima mogućnost automatske regulacije temperature i vlažnosti vazduha koja je potrebna kako bi se dobila što kvalitetnija suva šljiva, upotrebu savremenih sklopova ograničava njihova cena na svetskom tržištu, ali vremenom cene padaju, tako da će upotreba ovih elemenata biti sve rasprostranjenija.

Kao osnova projekta upotrebljen je PLC FATEK FBS 24 MAR kontroler, a pored ove osnove upotrebljava se računar i operator panel HMI TG465-MT. Kodovi za PLC su pisani u *WinProladder*-u, a za HMI u programu*Ttouchwin edit tool*. Parametri sistema koji se mere pomoću senzora RS-WS-N01-MW su temperatura i vlažnost vazduha. Sušara se može isključiti ili uključiti opciono lokalno pritiskom na taster ili HMI uređajem. U svakom trenutku se mogu videti parametri sistema na HMI uređaju i da li je ventilator u uključenom ili isključenom stanju.

II. REALIZACIJA UREĐAJA

PLC FATEK FBS 24 MAR je kontroler namenjen za automatizaciju mašina i procesa upravljanja u industriji. U svom sastavu kontroler poseduje četrnaest digitalnih ulaza na koje se dovodi taster za lokalno uključenje i isključenje sušare i deset relejnih izlaza na koje su vezani releji koji neposredno ili putem kontaktora, uključuju i isključuju strujna kola.

Kod se ubacuje sa računara u kontroler na jedan od više poznatih načina. Kontroler komunicira sa HMI uređajem RS232 komunikacionim protokolom. HMI uređaj se programira pomoću računara USB kablom. Na Slici 1. je prikazan PLC FATEK FBS 24 MAR kontroler.



Slika 1. PLC FATEK.

Za lokalno uključenje i isključenje sušare koriste se tasteri, koji se vezuju na logičke ulaze kontrolera. Taster je instalacioni prekidač koji zatvara ili otvara strujno kolo samo dok se deluje pritiskom na njega. Napomena u našem projektu tasteri nisu upotrebljeni već se sušara uključuje ili isključuje pomoću tač panela. Na Slici 2. je prikazan taster.



Slika 2. Taster.

Za uključenje i isključenje trofaznih potrošača koriste se elektromagnetni prekidači (kontaktori) Slika 3. Umesto kontaktora, moguće je koristiti i beskontaktne releje (solid-state reley). Naš sklop procesom sušenje šljiva koristi kontaktore zbog visoke cene beskontaktnih releja.

B. Anđelković, L. Pešić i T. Mitrović su učenici Tehničke škole “Rade Metalac” iz Leskovca, Dimitrija Tucovića bb,16000 Leskovac, Srbija. E-mail: bataandjelkovic12@gmail.com

I. Trajković profesor elektrotehničke grupe predmeta u TŠ “Rade Metalac”, Leskovac, Srbija. E-mail: itrajkovic7@gmail.com



Slika 3. Kontaktor.

Jedan od načina upravljanja sušarom je uz pomoć HMI uređaja. U našem slučaju upotrebljen je TG465-MT koji omogućava jednostavno povezivanje, praćenje i kontrolu procesa sušenja, on je prikazan na Slici 4.



Slika 4. TG465-MT.

TG465-MT vezan je za računardok se programira pomoću USB kablom, nakon završenog programiranja prekida se veza sa računarom i HMI uređajvezujemosa PLC-om RS232 komunikacionim protokolom.

Za zaštitu provodnika i kablova od struje, preopterećenja i kratkih spojeva upotrebljavaju se osigurači koji se postavljaju na početak svakog strujnog kola, Slika 5.



Slika 5. Automatski prekidač-osigurač.

Koriste se automatski osigurači tipa C određene nazivne struje koji služe za zaštitu provodnika i kablova od prekomernih struja.

Upotrebljen je senzor temperature i vlažnosti, slika 6. koji je pouzdan senzor za opštu i industrijsku namenu, koji radi u opsegu između 5-30VDC, može meriti od -40 – 80°C +/-0.3°C, vlagu 0% – 100%RH +/-2%RH. Njegovo povezivanje vrši se direktno na komunikacionoj kartici FBS-CB55, Slika 7, ona je povezana preko komunikacijskog porta RS485na FATEK PLC RS485.



Slika 6. Senzor temperature i vlažnosti.



Slika 7. FBS-CB55 –komunikaciona kartica.

Kako bi naša sušara efikasno dostigla zadatu temperaturu koristimo grejač, Slika 8, čija je snaga određena veličinom sušare,da bi se izjednačila temperatura i izbacio vlažan vazduh potreban nam je ventilator ili turbina za prinudnu cirkulaciju vazduha kako bi temperatura bila izjednačena i šljiva bila najboljeg kvaliteta. Zbog isparenja vode iz šljive dobija se kondenzacija u sušari, nju izbacujemo efikasno van sušare preko klapne sa smanjenje vlage u vazduhu, odnosno senzor detektuje povećanu vlagu u vazduhu, uključuje se ventilator, otvaraju se klapne i vlažan vazduh izlazni prinudno pod dejstvom ventilatora.

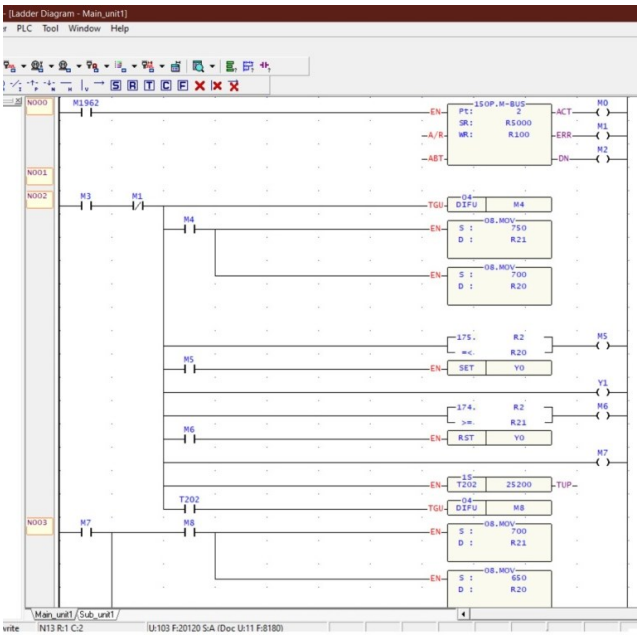


Slika 8. Grejač.

Projekat je započet povezivanjem najpre ulaznog tastera na digitalni ulaz kontrolera. Fatekov kontroler ima četrnaest digitalnih ulaza koji su označeni oznakama X0-X13i deset digitalnih izlaza koji su označeni oznakama Y0-Y9. Zatim se na kontroler dovodi i izvor napona 220V240V AC.

Drugi korak u izradi projekta je postavljanje komunikacione kartice FBS-CB55 na PLC preko RS485 modula. Senzor temperature i vlažnosti povezujemo na FBS-CB55.

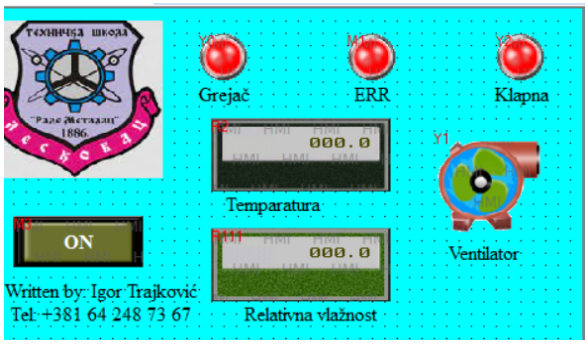
Treći korak je povezivanje računara i kontrolera radi upisivanja koda serijskim RS 232 protokolom. Kod je pisan jezikom lestvičastog dijagrama, u našem slučaju *WinProladder*-u, Slika 9.



Slika 9. Kod u WinProladder-u.

Rezultati merenja smeštaju se u registre kontrolera, a zatim se proveravaju na niz unapred zadatih uslova, a na HMI uređaju se može videti koji uređaj je aktivan u datom trenutku.

Četvrti korak je prekidanje komunikacije između računara i kontrolera i povezivanje kontrolera sa HMITG465-MTRS 232 komunikacionim protokolom, uspostavljanje komunikacije između računara i HMI uređaja kako bi se ubacio program u HMI. Potom prekidanje komunikacije između računara i HMI i povezivanje kontrolera sa HMI. HMI se programira korišćenjem programskog paketa *TouchWin edit tool*. Na Slici 10. se može videti kod programa *TouchWin edit tool*.



Slika 10. Glavni meni HMI uređaja.

Nakon pisanja koda, on se upisuje u HMI uređaj, ali se USB komunikacija sa računarom ne prekida.



Slika 10. Prikaz pokrenutog programa na HMI uređaju.

III. ZAKLJUČAK

Ovaj projekat je osmišljen i razrađen od strane učenika i realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti osnovno znanje iz automatike, programabilnih kontrolera, električnih instalacija i praktičnog rada sa elektro materijalima.

U radu je prikazan sklop koji je potvrdio da automatizacija procesa sušenja šljive značajno poboljšava efikasnost sistema. Upotreba preciznih senzora i naprednih algoritama upravljanja omogućava optimalnu kontrolu faktora sušenja kao što sutemperatura, relativna vlažnost i protok vazduha. Rezultati istraživanja su pokazali da se automatski sistem može prilagoditi različitim uslovima i vrstama šljive čime se postiže konstantan kvalitet sušenih proizvoda. Pored toga, automatizacija smanjuje potrebu za ljudskim nadzorom, minimizira troškove energije i povećava produktivnost. Ovaj rad ilustruje znperspektive automatizacije u poljoprivredi, pružajući naučnu osnovu za primenu ovih tehnoloških rešenja.

LITERATURA

[1] Upravljanje u realnom vremenu, dr Vera Petrović (http://www.ricum.edu.rs/files/3_7/viser/URV_prirucnik.compressed.pdf)

[2] Programibilni logički kontroleri (http://es.elfak.ni.ac.rs/mps/materijal/5-PLC.pdf)

[3] Vikipedija – Relej (https://sr.wikipedia.org/wiki/Relej)

[4] Električni osigurač- Vikipedija (https://www.adafruit.com/product/386)

[5] FBs-PLC/Manual download/FATEK AUTOMATION CORP. (http://www.fatek.com/en/technical.php?act=manual&catId=11).

Rukavica za prevođenje znakovnog jezika

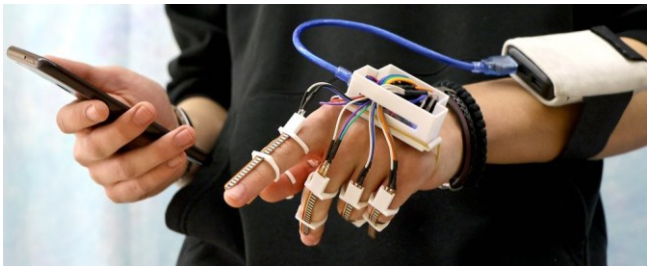
Nikola Ranđelović, Luka Rakić, Ivan Ćirković

Sadržaj – U ovom radu je prikazana softverska i hardverska realizacija rukavice za prevođenje znakovnog jezika koja koristi Arduino mikrokontoler i raznovrsne senzore, kako bi prevela pokret šake u slova.

I. Uvod

Kao zajednica, komunikaciju često uzimamo zdravo za gotovo. Koristimo je bez napora u svakodnevnom životu da bismo izrazili svoje misli i emocije, povezali se sa drugima i pristupili informacijama i uslugama. Međutim, za određene ljude, komunikacija može biti veliki izazov. Znakovni jezik je osnovno sredstvo komunikacije za mnoge u zajednici gluvih i nagluvih, kao i za ljude koji zbog fizičkih povreda ne mogu da govore, ali ne mogu svi da razumeju ili protumače znakovni jezik.

Ovo može dovesti do barijera u komunikaciji i poteškoća u pristupu informacijama i uslugama. Da bi se rešio ovaj izazov, razvijena je Arduino[1] rukavica za prevođenje znakovnog jezika. Ova rukavica ima za cilj da prevede znakovni jezik u tekst u realnom vremenu, olakšavajući osobama koje koriste znakovni jezik da komuniciraju sa drugima koji ga ne razumeju.



Slika 1. Praktična realizacija rukavice.

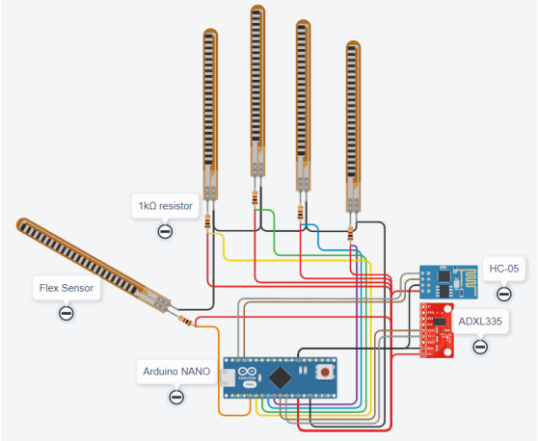
II. Princip rada i realizacija

Arduino rukavica za prevođenje znakovnog jezika je dizajnirana da prevede pokrete ruku u tekst koristeći flex senzore[2] i ADXL335 akcelerometar[3].

Flex senzori se koriste za praćenje savijanja prstiju i palca, dok se akcelerometar koristi za praćenje orijentacije i kretanja ruke, nakon čega se podaci šalju na aplikaciju na telefonu putem HC-05[4] bluetooth modula.

Za realizaciju ovog projekta, bilo nam je potrebno mesto za skladištenje ovih komponenti. Dizajnirali smo kućište za komponente u OnShape CAD[5] softveru, nakon čega smo model izvezli iz OnShape-a u .obj formatu, a potom ga uvezli u Prusa Slicer[6] i pripremili ga za 3D štampu. Ujedno smo kreirali i prstenje koje se koristi za postavljanje flex senzora na prste.

Flex senzori su softverski kalibrisani da uzmu u obzir individualne razlike u veličini ruke i dužine prstiju. Žice koje se koriste za povezivanje Arduino Nano[7] ploče i flex senzora su zalemljene za komponente. Između Arduino ploče i jednog od dva pina flex senzora se nalazi po jedan otpornik od 1 kΩ. Veze između HC-05 modula, akcelerometra i Arduino-a ostvarene su kratkospojnicima. Šema se može videti na slici 2.



Slika 2. Šema rukavice.

Hardverske komponente uključuju flex senzore, akcelerometar, žice, Bluetooth modul, otpornike i samo kućište. Flex senzori su pričvršćeni za prste, dok je akcelerometar pričvršćen za zadnji deo šake.

III. Komponente

1. Arduino Nano R3 – ATMega328P mikrokontroler na štampanoj ploči, ima 14 digitalnih ulazno-izlaznih pinova i 8 analognih ulaza.



Slika 3. Arudino Nano R3.

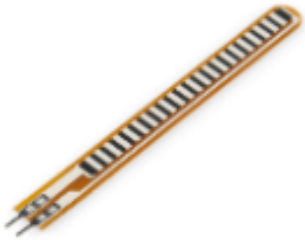
N. Ranđelović, L. Rakić i I. Ćirković su učenici četvrtog razreda na smeru Elektrotehničar informacionih tehnologija, u Elektrotehničkoj školi „Mija Stanimirović“ u Nišu, Bulevar Svetog cara Konstantina br 82-84, 18000 Niš. E-mail: netrunners@duck.com

2. HC-05 Bluetooth modul – Modul za bežičnu komunikaciju na Arduino platformi. Omogućava bežični prenos podataka do 10 m.



Slika 4. HC-05.

3. Flex senzor - Flex senzori su elektronski senzori koji se koriste za detektovanje savijanja objekata, u našem slučaju, prstiju. Ovi senzori koriste promenu otpora prilikom savijanja, a očitavaju se putem analognih ulaznih pinova na Arduino-u.



Slika 5. Flex Senzor.

4. ADXL335 Akcelerometar – ADXL335 je troosni analogni akcelerometar koji se koristi za merenje ubrzanja i nagiba. U našem slučaju, određuje ugao pod kojim je šaka. Ovaj senzor koristi MEMS tehnologiju za detekciju i merenje ubrzanja.



Slika 6. ADXL335 Akcelerometar.

5. Otpornici od 1 kΩ – Otpornici su jedni od najčešće korišćenih komponenti u elektronici. Otpornost od 1 kΩ se često koristi za ograničavanje struje ili smanjenje napona u kolu.



Slika 7. Otpornik.

IV. Rezultati testiranja i primene

Nakon sastavljanja i programiranja kola, sproveli smo testiranje, probajući gestikulacije ruku iz nakovnog jezika, kako bismo utvrdili funkcionalnost našeg uređaja.

Otkrili smo da je uređaj sposoban da prepozna i precizno prevede širok spektar gestikulacija ruku u sva slova američkog znakovnog jezika. Celokupni rezultati testiranja su ukazali na to da je rukavica efikasna, pouzdana i korisna kao uređaj za prevođenje gestikulacija ruku u tekst u realnom vremenu.

A. Realna primena i testiranje

Imali smo priliku da testiramo ovaj uređaj i uz pomoć gradske „Organizacije gluvih i nagluvih“ u Nišu, nakon čega smo saznali za prednosti i mane naše rukavice i formirali plan za buduću nadogradnju.

Zaključili smo da bi rukavica bila daleko efikasnija i brža ukoliko bi mogla da prevodi cele reči, pošto je trenutna funkcionalnost ograničena samo na prevođenje slova. Takođe smo zaključili da je podrška za više jezika od velikog značaja. Trenutna verzija može da prevodi samo američki znakovni jezik, odnosno slova američkog znakovnog jezika, što zapravo i predstavlja najveću manu uređaja.

Sem toga, kroz dalje testove smo zaključili da je rukavica izuzetno precizna i efikasna, retko pravi greške, ali je potrebno navikavanje na rukavicu i na njen način rada kako bi se ostvarili pozitivni rezultati kakve smo mi ostvarili. Glavna prednost je njena brzina i lako navikavanje na način korišćenja.

B. Budući planovi

S obzirom da je još jedan nedostatak rukavice manjak pratećeg softvera, planiramo da napravimo aplikaciju za mobilne telefone koja bi omogućila bolju kontrolu nad rukavicom, kao i odabir različitih znakovnih jezika.

V. Zaključak

Ovim projektom smo razvili rukavicu za prevođenje znakovnog jezika u tekst, koristeći Arduino platformu, flex senzore i akcelerometar.

Ovaj uređaj predstavlja korak ka razvoju tehnologije koja može pomoći uklanjanju barijera u komunikaciji za osobe sa oštećenim sluhom. Iako su sadašnji rezultati izuzetno dobri, u budućnosti, ovaj uređaj se može dalje razvijati i unapređivati, kako bi postao još efikasniji i

precizniji u prepoznavanju kompleksnih gestikulacija ruku i prevođenja istih.

ZAHVALNICA

Zahvaljujemo se upravi škole i mentoru, profesorki Ivani Miladinović, koji su nam obezbedili sredstva i uslove za rad, kao i profesorki Jeleni Rajković za pomoć oko nabavke modula iz inostranstva.

LITERATURA

[1] Arduino IDE - https://www.arduino.cc/en/main/software/
[2] Flex senzor - https://tinyurl.com/FlexSensor
[3] ADXL335 - https://tinyurl.com/ADXL335
[4] HC-05 Bluetooth - https://tinyurl.com/HC-05-BT
[5] OnShape CAD - https://www.onshape.com/en/
[6] Prusa Slicer - https://www.prusa3d.com/page/prusaslicer_424/
[7] Arduino Nano - https://docs.arduino.cc/hardware/nano

Multifunkcionalno pametno vozilo

Strahinja Srnić, Milos Radulović

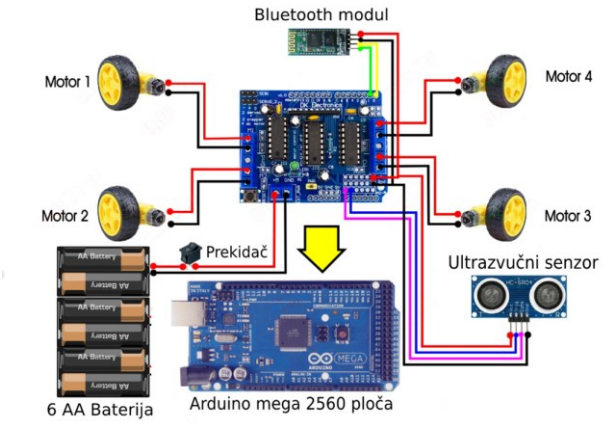
Sadržaj - Rad podrazumeva funkcionalnu maketu Multifunkcionalnog Pametnog Vozila koristeći Arduino Mega platformu i motor šild kontroler. Sistem bi trebao omogućiti manuelno i autonomno kretanje vozila preko Bluetooth modula uz korišćenje besplatne android aplikacije.

I. Uvod

Cilj ovog rada je demonstracija primene automobila sa kombinovanim upravljanjem koristeći Arduino mikrokontrolere. Upravljanje se vrši korišćenjem 4 zasebna DC motora koja su povezana putem arduinovog „motor shield“ kontrolera, koji omogućava nezavisnu rotaciju sva 4 točka.

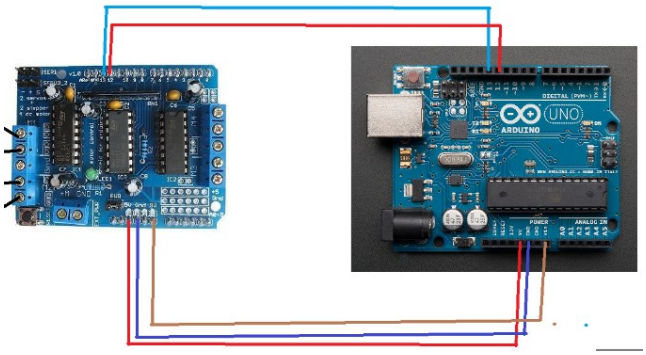
II. Princip rada

Naponski signal iz baterija (od 5V i 9V) je povezan na Arduino i motor šild, a upravljanje je rešeno preko blutut senzora i Android aplikacije (manuelno), a automatsko upravljanje preko ultrazvučnog senzora (automatsko-mapiranje). Električna šema prikazna je na slici 1, dok slika 2 prikazuje šemu povezivanja Arduina i motor šilda.



Slika 1. Električna šema vozila.

Sastavni delovi našeg vozila su:
- Arduino mega
- Arduino motor kontroler
- Bluetooth modul
- Ultra zvučni senzor
- 4 DC motora
- TT gear motor prenosnik
- Skelet i konstrukcija
- Pištalica.



Slika 2. Šema povezivanja motor šilda.

Arduino mega (slika 3) je napredniji model arduino uno-a, suštinska razlika je u broju analognih i digitalnih pinova, i u brzini integrisanog kola. Za razliku od arduino uno-a, arduino mega poseduje i dodatne RX i TX pinove koji su neophodni za uspostavljanje podatkovne veze sa Bluetooth modulom.



Slika 3. Arduino Mega.

Arduino motor kontroler (slika 4) je uređaj koji se koristi za kontrolu motora pomoću Arduinovog mikrokontrolera. On omogućava tačnu kontrolu brzine i smera rotacije motora, što ga čini idealnim za razne projekte automatizacije, robotičke tehnologije i sl.

Arduino Bluetooth modul (slika 5) je dodatak za Arduino mikrokontroler koji omogućava bežičnu komunikaciju sa drugim uređajima preko Bluetooth protokola. Ova funkcionalnost omogućava Arduinu da prima i šalje podatke na daljinu.

Arduino ultrazvučni senzor sa slike 6 (takođe poznat kao HC-SR04) se koristi za merenje udaljenosti do

S. Srnić i M. Radulović su učenici Mašinsko- saobraćajne škole iz Čačka, Srbija. E-mail: milenkovic87@yahoo.com.

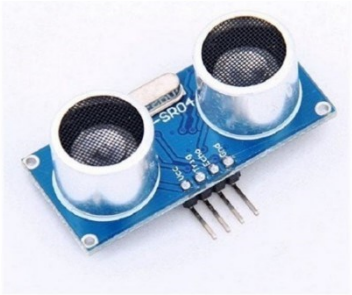
predmeta u okruženju. Radi na principu ultrazvučnog signala koji se šalje i onda se vraća kad se sudari sa preprekom. Korišćenjem četiri pina, dva za slanje signala (Trig i Echo) i dva za napajanje (Vcc i GND), senzor može meriti udaljenosti u rasponu od 2cm do 400 cm.



Slika 4. Arduino motor kontroler.



Slika 5. Arduino Bluetooth modul.



Slika 6. Arduino ultrazvučni senzor.

DC motori (slika 7) su jednostavni i ekonomični tip motora koji se često koriste u projektima automatizacije sa Arduinom. Oni rade na direktnom naponu (DC) i kontrolišu se pomoću Arduinovog mikrokontrolera putem motor kontrolera. DC motori su brzi i jednostavni za korišćenje, što ih čini idealnim za projekte kao što su roboti, mobilne platforme i sl.

TT gear motor prenosnik (slika 8 i 9) ovaj tip prenosnika se koristi za pojačanje snage, koje se ostvaruje regulacijom brzine motora. Prenosni odnos okji se ostvaruje je 1:48 (na svakih 48 obrtaja motora, točak će se okrenuti jedanput).

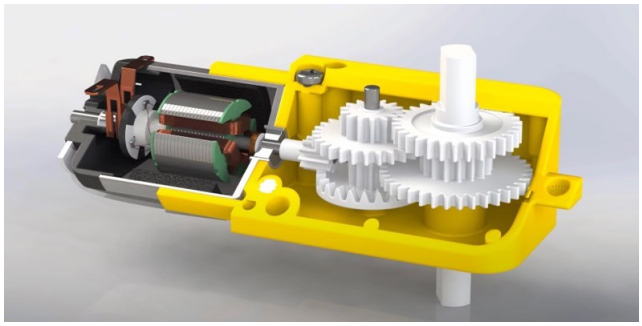
Konstrukcija je prvobitno bila napravljena u softveru SolidWorks gde je nakon toga prebačen u .stl format pa u .gcode i odštampan na 3D štampaču. Konstrukcija se sastoji iz 7 delova (slike 10 i 11). Korišćen je ISO standard, a delovi su povezani M3 40, M3 30, M3 15, M3 8 vijcima i M3 navrtkama.



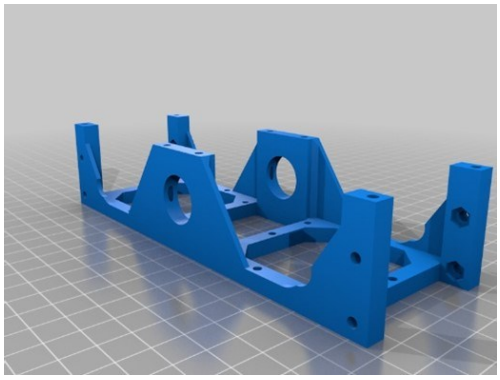
Slika 7. DC motor.



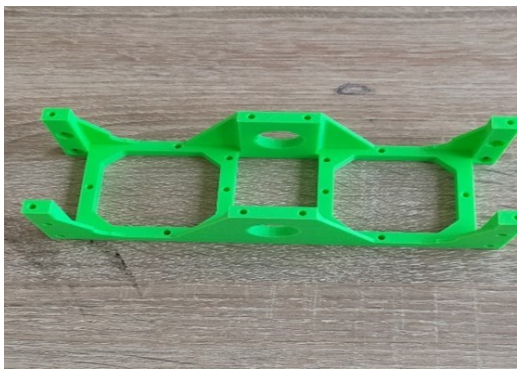
Slika 8. TT gear motor prenosnik.



Slika 9. TT gear motor prenosnik - presek.



Slika 10. 3D model.



Slika 11. Skeletna konstrukcija.

III. SOFTVERSKA REALIZACIJA

Ideja nam je bila da napravimo multifunkcionalna vozila sa jednom referentnom osnovom : kombinacija arduinove elektronike i 3D elemenata. Model koji prezentujemo je prototip osnovnog modela. Ostali modeli:

- Auto za lokalni prenos robe i namirnica.
- Auto sa robotskom rukom
- Auto za nepristupačne terene.
- Pametna kosilica
- Dron-auto (max. nosivost ~680g).
- eSIM_card WiFi auto za velike udaljenosti.

Kod za motore prikazan je na slici 12.

KOD ZA MOTORE:

```
#include<AF_MOTOR.h>
AF_DCMOTOR1(1);
AF_DCMOTOR2(1);
AF_DCMOTOR3(1);
AF_DCMOTOR4(1);
void forward()
{
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
}
```

Slika 12. Kod za motore.

Ultrasonic sensor je podešen da na određenoj udaljenosti prepozna prepreku, zaustavi vozilo i zarotira ga u pravcu u kome u blizini nema prepreka. Kod je dat u nastavku.

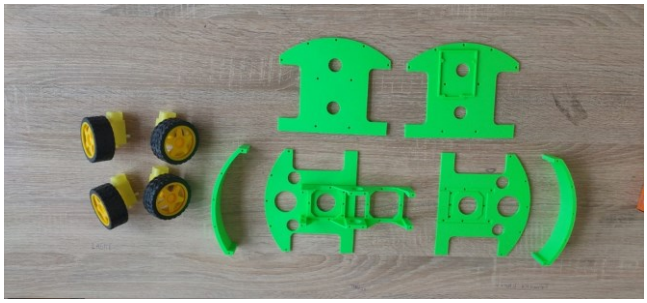
```
#include<SR04.h>
#define TRIG_PIN 12 #define ECHO_PIN 11
SR04 sr04 = SR04(ECHO_PIN, TRIG_PIN);
void setup() {
Serial.begin(9600);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
delay(1000);
}
```

Bluetotth modul, koji je stavljen na auto, je povezan sa Bluetooth android aplikacijom na telefonu, koji prima signal pokretnosti i omogućava da se preko viruelnog dzojstika upravlja multifunkcionalnim autom (manuelno). Kod je dat u nastavku.

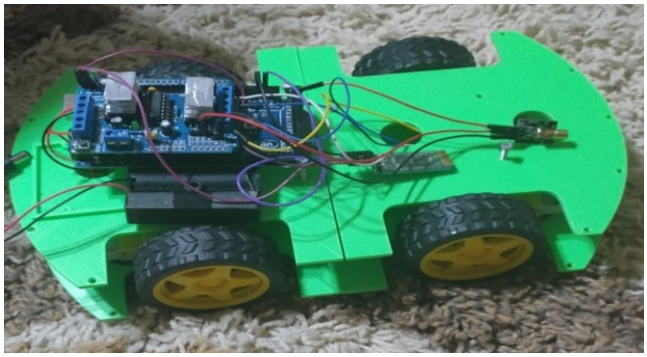
```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(19, 18); // RX | TX
void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);
}
void loop () {
  if (BTSerial.available()) {
    char data = BTSerial.read();
    Serial.write(data);
  }
  if (Serial.available()) {
    char data = Serial.read();
BTSerial.write(data)}
```

IV. PRAKTIČNA REALIZACIJA

Na slikama 13-14 prikazani su detalji praktične realizacije.



Slika 13. 3D štampani delovi.



Slika 14. Praktična realizacija modela.

V. ZAKLJUČAK

Korišćenje našeg poluautomatizovanog Sistema i robota bi se ubrzao i olakšao lokalni transport dobara čak i na nepristupačnim terenima zbog mogućnosti modulacije samih robota.

ZAHVALNICA

Hteli bismo da se zahvalimo svim profesorima koji su nam pomogli u ovom radu.

LITERATURA

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

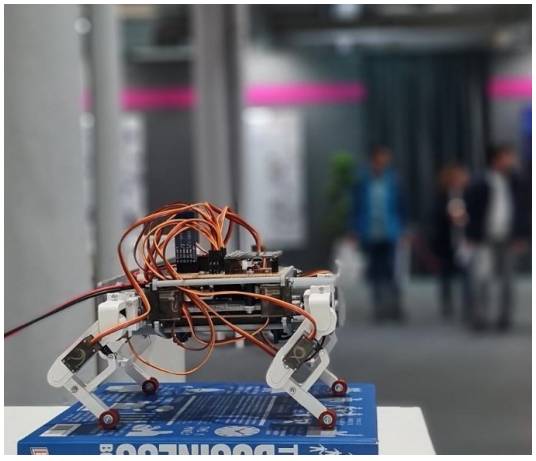
Održavanje ravnoteže četvoronožnog robota

Aleksandar Pavlovski

Sadržaj – U ovom radu prikazan je sistem za održavanje ravnoteže četvoronožnog robota baziran na PD regulatoru. Pored samog sistema su grafički prikazani i odzivi na različite pobudne signale.

I. UVOD

Četvoronožni roboti postaju sve popularniji iz godine u godinu i pronalaze sve više primena u industriji, a i u svakodnevnom životu ljudi. Najveću ulogu u industriji pronalaze tamo gde je izvršenje nekog zadatka opasno za čoveka, a nepogodno za dronove ili robote sa točkovima. To su najčešće kompleksni zadaci na nepristupačnom terenu. U svakodnevnom životu, ovi roboti pronalaze razne primene zbog njihovog atraktivnog izgleda i kretanja poput četvoronožnih životinja. Da bi ovi roboti mogli da izvršavaju zadatke, izuzetno je bitno da poseduju mogućnost održavanja ravnoteže. Bez ove mogućnosti, prelazak preko nepristupačnog terena bi bio izuzetno otežan, a samim tim i izvršenje zadatka bi bilo neuspešno ili jako neprecizno. Iz tog razloga, jedna od osnovnih funkcija četvoronožnog robota bi trebalo da bude upravo održavanje ravnoteže. Četvoronožni robot sa slike 1 ima 3 stepena slobode po nozi, dakle, 12 ukupno i baziran je na *Arduino Mega 2560* razvojnom sistemu. Održavanje ravnoteže se vrši uz pomoć PD regulacije, a informacije o uglovima naginjanja na osnovu kojih se računaju signali se dobijaju sa žiroskopa.

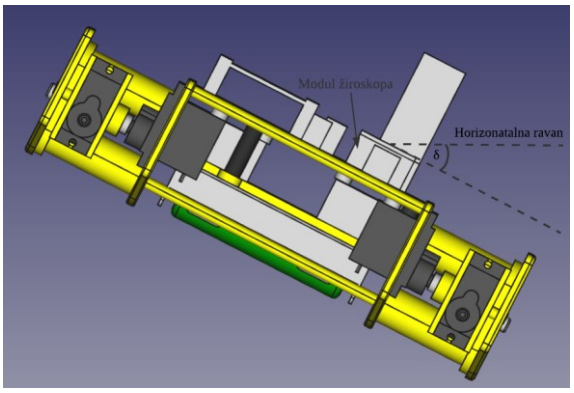


Slika 1. Četvoronožni robot na 65. medjunarodnom sajmu tehnike u Beogradu [1].

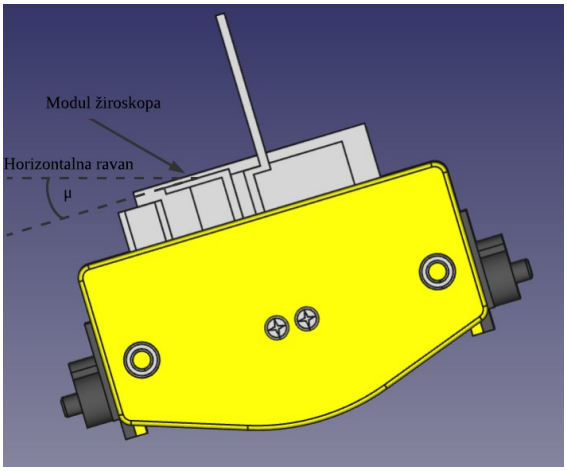
A. Pavlovski je student na modulu Upravljanje sistemima, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.pavlovski@elfak.rs

II. OSNOVNI PRINICIP RADA

Željeni uglovi naginjanja robota δ i μ, prikazni na slikama 2 i 3, su jednaki nuli jer je u interesu da robot održava horizontalan položaj tela u svakom trenutku. Trenutni uglovi robota δ i μ se dobijaju na osnovu informacija sa žiroskopa. Računa se greška, to jest, razlika željene pozicije i stvarne pozicije koja se dovodi na ulaz PD regulatora. Signal PD regulatora predstavlja vrednost za koju treba korigovati parametre h ekstremiteta. Parametar h predstavlja rastojanje izmedju podloge i zgloba koji povezuje ekstremitet i telo robota. Parametar h, kao i ostali parametri su detaljnije objašnjeni u [1].

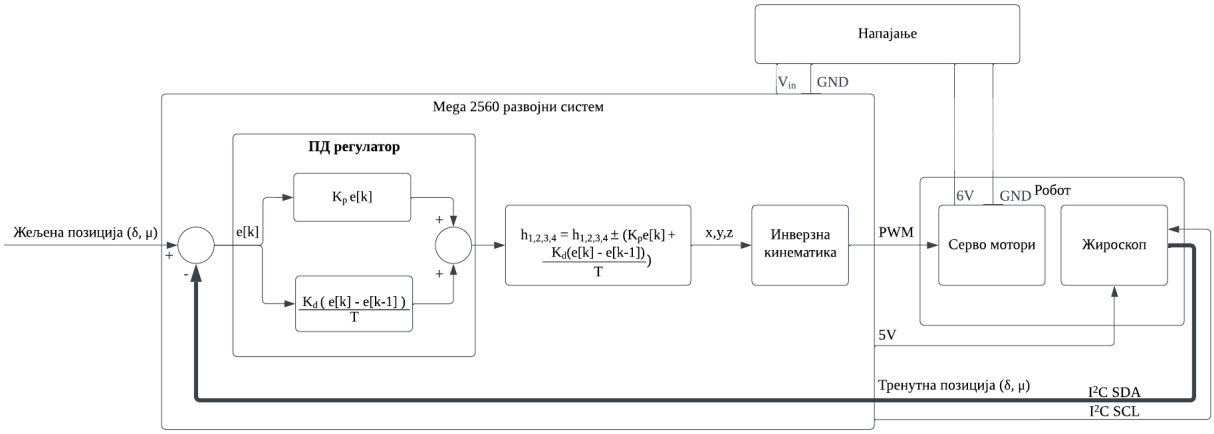


Slika 2. Ugao δ [1].



Slika 3. Ugao μ [1].

Računaju se dva signala greške, jedan signal za ugao δ i drugi signal za ugao μ. Na osnovu signala greške se dobijaju P i D parametri koji direktno utiču na parametre h ekstremiteta.

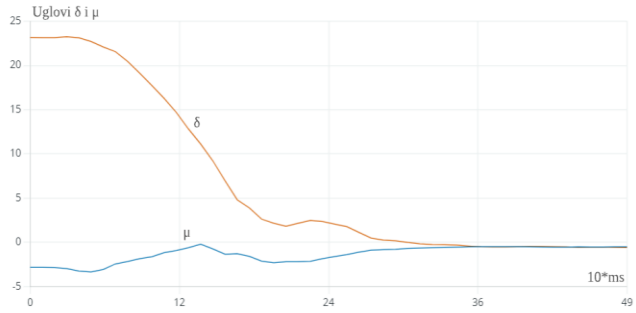


Slika 4. Šema povratne sprege [1].

Nakon dobijenih novih parametara h se izračunavaju x, y i z koordinate vrha svakog ekstremiteta i nakon rešavanja inverzne kinematike se šalju PWM signali servo motorima. Na slici 4 je prikazana šema povratne sprege sistema [1].

III. ODZIV SISTEMA

Na slikama 5 i 6 su prikazani odzivi sistema. Kako su grafici snimljeni u Arduino okruženju gde ne postoji mogućnost modifikacije x i y ose, na x osi se nalazi broj ispisa na serijskom portu. Uz pomoć funkcije millis() je određeno vreme potrebno od jednog ispisa do drugog i utvrđeno je da je u ovom konkretno slučaju to vreme 10 milisekundi što znači da su na x osi označene desetine milisekundi. Na graficima se primećuje da sistem ima brz odziv i da ulazi u ustaljeni režim za manje od pola sekunde [1].



Slika 5. Odziv sistema, vrši se samo regulacija ugla δ [1].

Parametri pojačanja P i D su eksperimentalno određeni na osnovu snimanja odziva i teorijskog znanja o PID regulaciji. Vrednosti određenih parametara su:

- P = 0.07
- I = 0
- D = 0.002

Navedeno je i integralno dejstvo, međutim, njegovo pojačanje je jednako nuli iz razloga što se pokazalo da integralno dejstvo unosi kašnjenje u sistem. Na graficima se jasno vidi da postoji greška u ustaljenom režimu od približno -1° što se može zanemariti [1].



Slika 6. Odziv sistema, vrši se regulacija oba ugla [1].

IV. ZAKLJUČAK

Postignuti su dobri rezultati održavanja ravnoteže jednostavnom primenom PD regulatora. Treba napomenuti da ovakav algoritam vrši održavanje ravnoteže isključivo na kosoj ravni, a može se iskoristiti u daljem projektovanju koračanja po neravnom terenu gde vrhovi ekstremiteta ne formiraju ravan. Zainteresovani čitaoci mogu uživo da vide mehatronički sistem prikazan na slici 1 u Laboratoriji za robotiku Elektronskog fakulteta u Nišu. Ukoliko je čitaoc zainteresovan da učestvuje u razvoju projekta, poziva se da kontaktira autora ovog rada.

LITERATURA

[1] A. Pavlovski, Diplomski rad „Praktična realizacija četvoronožnog robota i projektovanje i implementacija sistema upravljanja“.

Sistem za pomoć pri paralelnom parkiranju

Lazar Lapadatović, Sofija Đorđević, Milutin Milenković

Sadržaj - Ovaj rad ima za cilj da predstavi sistem koji služi kao pomoć pri paralelnom parkiranju vozila. Osnovu sistema čine: *Arduino* [1], senzor za daljinu (HC-SR04) [2], protoploča [3], Wi-Fi modul [4] i LED diode. Programiranje sistema se vrši u softverskom paketu *Arduino IDE* [5].

I. Uvod

Sistem za pomoć pri paralelnom parkiranju je pomoćno sredstvo koje obezbeđuje bezbedno i lako izvođenje paralelnog parkiranja.

Ovaj sistem ima primenu u svakodnevnom životu zbog sve veće gužve u saobraćaju i poteškoća pri parkiranju vozila a sigurni smo da bi smanjio broj sudara i šteta pri izvođenju paralelnog parkiranja.

Na ideju za ovaj rad smo došli tako što smo često čuli žalbe kako iskusnih, tako i manje iskusnih vozača na problem sa paralelnim parkiranjem, zato smo odlučili da napravimo ovaj sistem koji bi pomogao svim vozačima u izvođenju radnje paralelnog parkiranja.

Prednosti ovog sistema što nije potrebno uložiti mnogo novca za izradu i lako ga je postaviti na putničko vozilo.

II. Princip rada i praktična realizacija

A. Uvod u sistem

Sistem za pomoć pri paralelnom parkiranju funkcioniše tako što dok se automobil kreće određenom brzinom, senzorom za daljinu registruje dovoljan prostor za parkiranje vozila a paralelno se meri i vreme u sekundama praznog mesta. Merenje vremena praznog mesta se vrši od početka praznog mesta do trenutka dok automobil ne naiđe na početak drugog parkiranog vozila. Parking mesto se računa formulom za računanje pređenog puta:

$$s = v \cdot t. \tag{1}$$

U zavisnosti od dužine automobila određuje se da li ima dovoljno prostora da se izvrši paralelno parkiranje. Podaci koji se mere i obrađuju šalju pomoću Wi-Fi modula na *web* server.

L. Lapadatović, S. Đorđević i M. Milenković su učenici četvrte godine smera Elektrotehničar Informacionih Tehnologija u Tehničkoj školi, Zaječar, Kneginje Ljubice 3-5, 19000 Zaječar, Srbija, E-mail: lazar.lapadatovicsk@gmail.com, sofijadj.tehnicka@gmail.com, milutin778@gmail.com.

B. Predstavljanje Komponenata i Priprema za Izradu

Arduino Uno [6] je platforma otvorenog koda. Hardver se sastoji od jednostavnog otvorenog hardverskog dizajna *Arduino* ploče sa Atmel AVR procesorom i pratećim ulazno-izlaznim elementima, tačnije, na sebi poseduje mikrokontroler. Softver se sastoji od razvojnog okruženja koje čine standardni kompajler i *bootloader* koji se nalazi na samoj ploči.

Arduino ploču čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama. koje omogućavaju programiranje i povezivanje sa drugom elektronikom.

Na slici 1 prikazan je *Arduino Uno* koji je korišćen u razvoju ovog projekta.



Slika. 1. Arduino Uno.

Ultrazvučni senzor za udaljenost HC-SR04 pruža funkciju beskontaktnog merenja od 2 cm do 400 cm sa preciznošću koja može da dostigne i do 3 mm. Senzor je težine 8.7 g i radi pri ultrazvučnoj frekvenciji od 40 KHz. Senzor radi na naponu od 5 V. Dimenzije senzora su: 45.5 x 20 x 15.5 mm. Senzor sadrži četiri pina: VCC, GND, TRIG i ECHO. Ovaj senzor smo izabrali zbog niske cene i velike preciznosti u merenju. Na slici 2 je prikazan ultrazvučni senzor udaljenosti.



Slika. 2. Ultrazvučni senzor udaljenosti HC-SR04.

Ultrazvučni senzor udaljenosti služi da izmeri da li ima prostora za parkiranje vozila sa strane. Merna jedinica je u centimetrima.

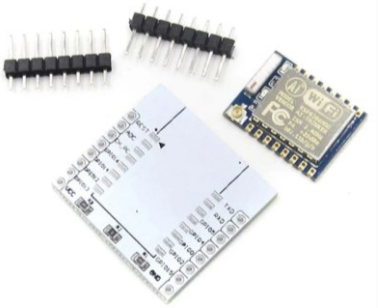
Protoploča je ploča koja sadrži 830 rupa koje su podeljene u 10 kolona i 63 reda. Svi su standardno

razmaknute 0.254 cm. Dva seta od 5 kolona su odvojena za 0.762 cm. Ploča prihvata veličine žice u rasponu od 20-29 AWG. Pomoću protoploče smo povezali sve senzore i ona omogućuje pravilno funkcionisanje sistema. Na slici 3 je prikazana protoploča.



Slika 3. Protoploča.

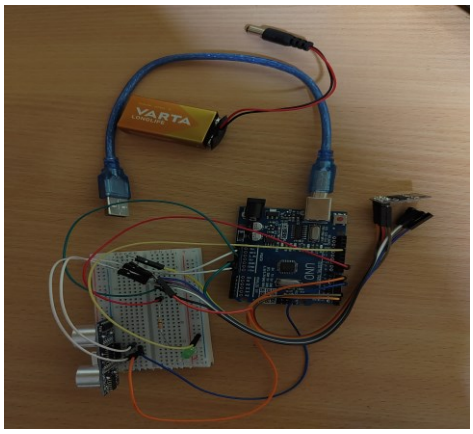
Wi-Fi modul je baziran na ESP8266 SoC integrisanim TCP/IP protokolom. TTL serijski komunikacioni interfejs i parametri mogu biti podešeni preko AT komandi. Radni napon komunikacionog interfejsa je 3.3 V. Tip antene je PCB antena. Podržava WPA i WPA2 sigurnosne modove. Wi-Fi modul je prikazan na slici 4:



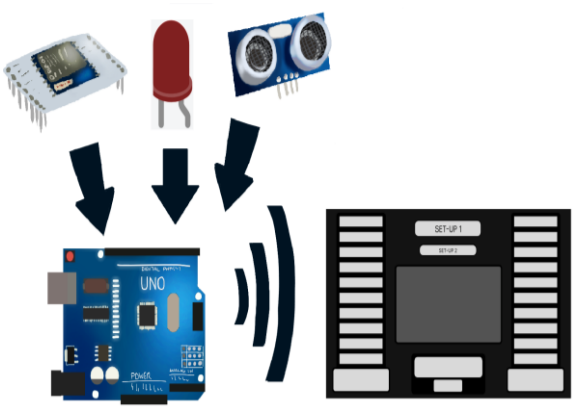
Slika 4. Wi-Fi modul.

C. Izrada sistema

Projekat je započet povezivanjem ultrazvučnog senzora za udaljenost HC-SR04 i Wi-Fi modula sa Arduinom Unom preko protoploče. Prikaz povezanih komponenti je na slici 5, dok je slika blok šeme na slici 6.



Slika 5. Slika povezanih komponenti.



Slika 6. Blok šema.

D. Programiranje

Posle povezivanja, kod je napisan u softveru za *Arduino IDE* koji se preko USB kabla učitava u *Arduino*. Dokle god sa strane vozila nema dovoljno prostora za parkiranje neće svetleti indikator koji je u našem slučaju led dioda. U trenutku kada vozio naiđe na dovoljan prostore led dioda se aktivira. Na slici 7. je prikazan deo koda.

HC-SR04 senzor za daljinu.ino

```
const int TRIG_PIN = 6; // Arduino pin connected to Ultrasonic Sensor's TRIG pin
const int ECHO_PIN = 7; // Arduino pin connected to Ultrasonic Sensor's ECHO pin
const int LED_PIN  = 9; // Arduino pin connected to LED's pin
const int DISTANCE_THRESHOLD = 10; // centimeters
// variables will change:
float duration_us, distance_cm, duzina_mesta, predjeni_put = 0;
float brzina = 2.77; // m/s
unsigned long pocetak_vremena, kraj_vremena;
void setup() {
  Serial.begin (9600);       // initialize serial port
  pinMode(TRIG_PIN, OUTPUT); // set arduino pin to output mode
  pinMode(ECHO_PIN, INPUT);  // set arduino pin to input mode
  pinMode(LED_PIN, OUTPUT);  // set arduino pin to output mode
}
void loop() {
  // generate 10-microsecond pulse to TRIG pin
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // measure duration of pulse from ECHO pin
  duration_us = pulseIn(ECHO_PIN, HIGH);
  // calculate the distance
  distance_cm = 0.017 * duration_us;
  // check if distance is greater than threshold
  if(distance_cm > DISTANCE_THRESHOLD) {
    digitalWrite(LED_PIN, HIGH); // turn on LED
    // print the value to Serial Monitor
    Serial.print("distance: ");
    Serial.print(distance_cm);
    Serial.println(" cm");
    // start timing
    pocetak_vremena = millis();
```

Slika 7. Deo koda.

Za prikupljena informacija o pređenom putu od strane Arduino senzora koristi se $_POST metoda [7], takođe i za slanje podatka na internet stranicu, napravljenu za izračunavanje pređenog puta formulom:

$$s = v \cdot t. \tag{2}$$

Metoda $_POST je super globalna promenljiva u programskom jeziku PHP [8], koja služi za prikupljanje informacija koje su prethodno poslate HTML formom [9].

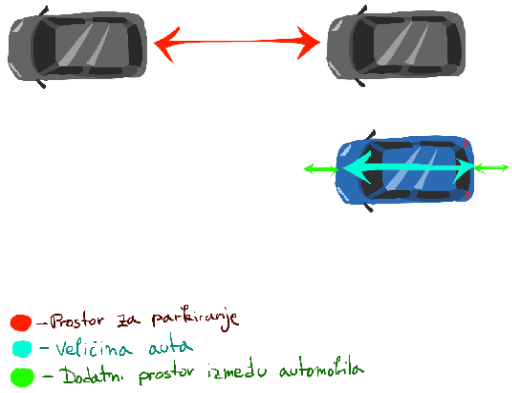
Metoda $_POST je takođe korišćena i za prenošenje promenljivih vrednosti.

Kada *Arduino* senzor počne sa merenjem rastojanja između dva već parkirana vozila, na specijalnoj internet stranici, koju ćemo u nastavku zvati *developer stranica*, funkcija sa JavaScript intervalom [10] na svakih 200 ms slati zahtev u kom će „*pitati*" da li je merenje gotovo.

Ako je odgovor na zahtev *false*, procedura će se ponoviti i interval će se pozvati nakon dodatnih 200 ms.

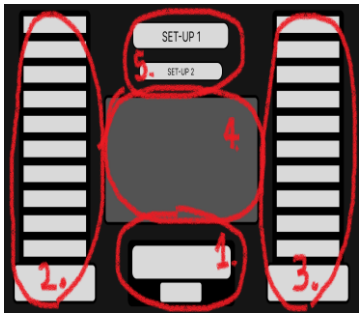
U slučaju da je merenje završeno i odgovor na postavljeni zahtev je *true*, interval će se završiti i finalna informacija o rastojanju će biti poslata *developer stranici*.

Uzimajući u obzir da je dužina automobila unapred definisana, do podatka da li će parkiranje biti moguće, moguće je doći formulom: *Pp* - (*Va* + *PiA*), gde je *Pp* prostor za parkiranje, *Va* je dužina vozila i *PiA* je dodatni prostor između vozila. Na slici 8 je vizualna prezentacija događaja.



Slika 8. Vizuelna prezentacija događaja.

Izračunata informacija se potom upisuje u tekstualni dokument funkcijom *fread()* [11] omogućavajući lokalni pristup informaciji. *Web* stranica, koja će kasnije biti nazvana i *glavna stranica*, „čita" podatak iz tekstualnog dokumenta i tu informaciju vizuelno prikazuje. Informacija o mogućnosti parkiranja se prikazuje na web stranici u vidu slike u *.SVG* formatu [12] koji je prethodno dizajniran u programu za dizajn interfejsa *Figma* [13].



Slika 9. Skelet *web* stranice.

Na slici su prikazani delovi skeleta *web* stranice. U prvom delu se prikazuje mogućnost za uspešno izvršavanje parkinga pomoću boja i izračunate površine prostora za parking pomoću senzora, dok se u drugom delu prikazuje brzinomer za vozilo.

U trećem delu se prikazuje brojač obrtaja za vozilo, a u četvrtom se prikazuje ekran koji pokazuje željene podatke na osnovu odabranog setupa. U petom delu se prikazuje displej za biranje željenog setupa u zavisnosti vozača koji upravlja vozilom.

III. Rezultat

Rezultat ovog rada je funkcionalni sistem koji omogućuje bezbedno izvršavanje paralelnog parkiranja. Sistem će u mnogim slučajevima pomoći svim vozačima i tako smanjiti broj nesrećnih slučajeva koji su posledica lošeg izvršenja paralelnog parkiranja, pogotovo zbog toga što je u današnje vreme gužva u saobraćaju česta pojava.

Ovaj sistem se dobro pokazao u simulaciji izvršenja radnje paralelnog parkiranja.

IV. Zaključak

Moguće je uz pomoć dostupnih komponenti na tržištu napraviti sistem za pomoć pri paralelnom parkiranju koji bi u praksi bio pomoć lošijim vozačima. Bilo je potrebno iztražiti moguće metode i primeniti znanje stečeno ne samo u okviru redovne nastave iz računarskih i elektro predmeta. Autori rada su zadovoljni krajnjim proizvodom.

Zahvalnica

Projekat ne bi bio moguć bez podrške lokalne softverske firme NextBlink doo koja je materijalno i mentorskim radom pomogla realizaciju ovog projekta.

Literatura

[1] Tehnička dokumentacija za Arduino. Dostupno na: https://www.arduino.cc/
[2] Tehnička dokumentacija za ultrazvučni senzor za daljinu. Dostupno na: https://www.sparkfun.com/products/15569#:~:text=This%20is%20the%20HC%2DSR04,receiver%20and%20a%20control%20circuit
[3] Tehnička dokumentacija za protoploču. Dostupno na: https://www.sparkfun.com/products/12615
[4] Tehnička dokumentacija za Wi-Fi modul. Dostupno na: https://www.mikroprinc.com/sr/proizvod/arduino-wifi-modul
[5] Softverski alat Arduino IDE. Dostupno na: https://www.arduino.cc/en/software
[6] Tehnička dokumentacija za okruženje Arduino Uno. Dostupno na: https://docs.arduino.cc/hardware/uno-rev3
[7] Opis $_POST metode. Dostupno na: https://www.w3schools.com/php/php_superglobals_post.asp
[8] Tehnička dokumentacija za PHP. Dostupno na: https://www.php.net

[9] Tehnička dokumentacija za HTML format. Dostupno na: https://www.w3schools.com/html/html_forms.asp
[10] Tehnička dokumentacija za JavaScript interval. Dostupno na: https://www.w3schools.com/JSREF/met_win_setinterval.asp
[11] Opis funkcije fread(). Dostupno na: https://www.w3schools.com/php/func_filesystem_fread.asp
[12] Opis .SVG formata. Dostupno na: https://en.wikipedia.org/wiki/SVG
[13] Softverski alat .SVG Figma. Dostupno na: https://www.figma.com/

Aplikacija za učenje srpskog jezika „*Serbiapp*“

Aleksa Milić, Dragana Andrejević

Sadržaj – Aplikacija “*Serbiapp*” prikazana u ovom radu koristi se za učenje srpskog jezika. Autor je dizajnirao i implementirao aplikaciju sa ciljem da olakša učenje i na jednostavan i zanimljiv način premosti brojne poteškoće sa kojima se susreću svi oni koji žele da nauče srpski jezik. Aplikacija je kreirana kao projektni zadatak iz predmeta Objektno orijentisano programiranje. Jednostavnim kretanjem kroz aplikaciju korisnik se susreće sa gramatičkim pravilima, primerima i interaktivnim pričama, koje tokom progresije postaju sve teže. Pomoć korisniku pružena je u vidu “hintova” koji prikazuju prevod reči. Nakon završene lekcije, reči ulaze u rečnik korsnika, koji odatle može odabrati koje će od njih dodatno vežbati po “*flashcards*” sistemu.

I. Uvod

Tehnološke inovacije i ubrzani napredak tehnologije doveli su do razvoja učenja na daljinu putem interneta. Ovakav vid nastave zasniva se na korišćenju savremene računarske i kominukacione tehnologije. Komunikacija, kao i razmena različitih materijala za učenje, između učesnika u obrazovnom procesu odvija se putem interneta, korišćenjem različitih platformi [1]. Softverske platforme za učenje jezika pružaju lingvističke informacije, najčešće su organizovane po modulima i mogu imati procedure za procenu jezičkih kompetencija i veština nakon završetka svakog modula [2].

Razvojem tehnologija, naročito računara i interneta, a poslednje decenije i mobilne telefonije, menjali su se i načini na koje su se savremene tehnologije primenjivale u nastavi jezika. Tradicionalni način učenja jezika kroz neposrednu komunikaciju nastavnika i učenika ne treba menjati, ali se on, svakako, može dopuniti i unaprediti korišćenjem alata i aplikacija za prevođenje na mreži i mobilnim uređajima [3].

Poznata je činjenica da je srpski jezik veoma specifičan i težak za učenje, naročito kada govorimo o gramatici. Mnogim strancima je problem da savladaju ćirilično pismo, ali i gramatiku prilikom učenja srpskog jezika. Velike poteškoće pri bržem i lakšem učenju predstavljaju četvoroakcentni sistem, sedam padeža, veliki broj dijalekata, kao i ostali segmenti gramatike. Sa druge strane, postoji veliki broj aplikacija za učenje jezika, ali one uglavnom koriste generičke fraze i ne oslikavaju na pravi način reči koje se koriste u svakodnevnoj komunikaciji. Autor ovog rada je kreirao aplikaciju “*Serbiapp*” sa ciljem da se olakša učenje srpskog jezika.

A. Milić i D. Andrejević, Gimnazija u Leskovcu, 16000 Leskovac, Srbija, E-mail: milicaleksaleskovac@gmail.com, dragana.a@elfak.rs.

II. Aplikacija „*Serbiapp*“

A. Pregled literature

Google Translate je jedna od najpopularnijih aplikacija za prevođenje. Kompanija Google je razvila ovu aplikaciju u aprilu 2006. godine za prevođenje reči, rečenica, pasusa i veb stranica sa jednog jezika na drugi. *Google* prevodilac nudi funkcije unosa teksta, glasovni režim i skeniranje pomoću mobilne kamere. Lak pristup i brz prevod su pozitivne strane, dok su gramatičke greške i nedostatak alata za lekturu uočeni kao osnovni nedostaci [4].

Pregledom literature i aplikacija, trenutno dostupnih na internetu, za učenje jezika može se istaći da su većina aplikacija, koje su dizajnirane za učenje srpskog jezika, automatizovane aplikacije. One funkcionišu na principu kreiranja primera na engleskom jeziku, a zatim se ti primeri (reči, fraze, rečenice) prevode automatski na različite jezike, ne vodeći računa o lingvističkim i gramatičkim specifičnostima svakog jezika ponaosob. Takođe, fraze koje koriste su generičke i ne oslikavaju na pravi način komunikaciju između ljudi na dnevnom nivou.

Brojne su prednosti korišćenja softverskih aplikacija za učenje jezika. Jedna od najvažnijih jeste fleksibilnost. Ona se prvenstveno odnosi na raspored vremena, odnosno časova, jer korisnik sam određuje u koje vreme će da uči. Potom, tu je fleksibilnost koja se odnosi na lokaciju, aplikaciju je moguće koristiti sa različitih uređaja, važno je samo da postoji pristup internetu [5]. Korisnik može da prekine sa korišćenjem aplikacije i nakon nekog vremena ponovo nastavi sa učenjem tamo gde je stao.

B. Osnovne karakteristike aplikacije

Aplikacija prati strukturu lekcija koja je prisutna u velikom broju knjiga za učenje engleskog, kao i drugih stranih jezika. Konkretnije rečeno, lekciju čine gramatička pravila, primeri i priča.

Svaka lekcija prelazi određenu temu, koja može da se odnosi na vokabular ili gramatiku. Od vokabulara prisutan je veliki broj svakodnevnih reči i fraza poput pozdrava, pitanja, hrane, brojeva i boja. Iz gramatike se obrađuju osnovni koncepti kao što su lične zamenice, rod, broj, padež (nominativ i akuzativ), vremena (prošlo i sadašnje), pridevi i nekoliko glagola. Zbog toga što je ovo prototip aplikacije, ne sadrži dovoljno lekcija kako bi se svaka od ovih gramatičkih tema objasnila u celosti. Najosnovniji koncept poput množine reči zahteva znanje glasovnih promena koje je teško objasniti i maternjim govornicima

jezika. Za potpuno učenje gramatike potreban je ogroman broj lekcija, ne samo da bi se objasnili brojni koncepti gramatike srpskog jezika, već i kako bi se kroz primere podsvesno upijala njihova primena.

Kao što je već rečeno, svaka lekcija podeljena je u 3 sekcije. To su gramatika, primeri i priča. Gramatika predstavlja uvod u lekciju, prelazeći koncepte u njoj uz objašnjene žasto su takvi kakvi jesu. Zatim sledi sekcija primera koja se prelazi kroz 3 ponavljanja, gde u svakom primeri postepeno postaju zahtevniji. Neki od primera zahtevaju prevod reči sa engleskog na srpski jezik uz slike, prevod rečenice sa srpskog na engleski jezik pomoću datih reči, sklapanje rečenice na osnovu audio snimka pomoću datih reči, povezivanje srpskih i engleskih reči izmešanih u 2 reda, prevod rečenice sa srpskog na engleski jezik kucanjem, kao i još nekoliko sličnih primera koji predstavljaju manje varijacije navedenih, poput menjanja smera prevoda. Konačno, nakon odrađenih primera korisnik dobija mogućnost da svoje znanje primeni kroz interaktivnu priču na kraju svake lekcije. Za razliku od primera, priče nemaju šablon i svaka je na neki način unikatna.

Jedna od važnih karakteristika aplikacije prikazane u ovom radu je sistem pamćenja reči nakon lekcija uz njihovo obnavljanje po "*flashcards*" sistemu [6]. Nakon završetka lekcije vokabular reči se dopunjuje novim rečima. Korisnik je u mogućnosti da od datih reči izabere one koje želi da dodatno vežba. Jedna po jedna od odabranih reči pojavljuju se na ekranu na srpskom jeziku, a klikom na datu reč, prikazano na slici 1. otkriva se njen prevod na engleskom jeziku. Na taj način korisnik aplikacije je u mogućnosti da sam testira svoje znanje bez neke vrste ocenjivanja. Neke postojeće aplikacije za učenje jezika su kreirane isključivo za ovakav način učenja, a kod veoma malog broja aplikacija je ova funkcionalnost dostupna kao jedna od više opcija.



Slika 1. Izgled kartice pre i nakon klika od strane korsnika.

Aplikacija je kreirana korišćenjem softverskog alata, okruženja, za kreiranje programa u programskom jeziku C#, *Visual Studio* [7]. Kompletan programski kod aplikacije napisan je u programskom jeziku C#. Uprkos nekim limitima koje *Visual Studio* ima, poput pozadinske boje elemenata koja nikada ne može biti istinski transparentna, težnje ka elementima pravougaonog oblika, poteškoća za highlight slika i brojnih drugih, autor ovog rada uspeo je da vrlo kreativno i estetski implementira sve funkcionalnosti aplikacije koje su neophodne za učenje srpskog jezika, sa ciljem da ona bude što jednostavnija za korišćenje, ali i vizuelno dopadljiva korisnicima. Neka od rešenja navedenih problema su crtanje transparentne bitmape za highlight tastere i pravljenje oblih elemenata spajanjem više crtanih grafika u jednu.

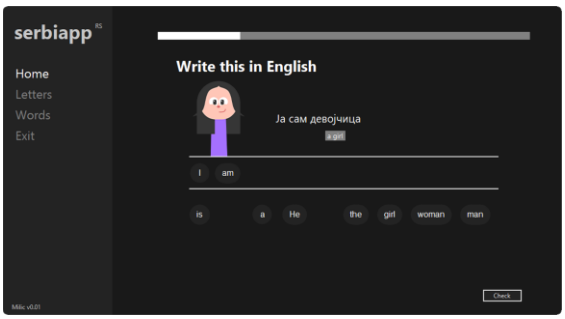
Aplikacija sadrži stranicu na kojoj se nalaze sva slova srpske azbuke, kao i mogućnost čitanja svakog slova. Na slici 2. prikazan je izgled stranice na kojoj se nalazi interaktivan prikaz slova azbuke uz ekvivalente zvukove napisane pravilima izgovora engleskog jezika. Pritiskom na bilo koje od slova pušta se audio fajl izgovora datog slova. Za snimanje datih audio fajlova korišćen je program *Audacity* [8].



Slika 2. Izgled stranice na kojoj se nalaze slova azbuke.

Značajna karakteristika aplikacije je prikazivanje "hintova" odnosno prevoda reči, koje daju bolji vizuelni efekat ali i značajno pojednostavljuju upotrebu aplikacije.

Na slici 3. prikazan je izgled stranice na kojoj se nalaze materijali za učenje jezika, odnosno rečenica na srpskom i reči na engleskom od kojih korisnik kreira prevedenu rečenicu. Tekst prate slike koje dodatno pružaju pomoć prilikom učenja jezika i imaju vizuelni efekat kako bi korisnik što lakše mogao da upamti izraze. Takođe, na stranici se nalazi i statusna linija koja se popunjava prilikom svakog prelaska na naredni nivo, odnosno sledeću rečenicu, koja je složenija od prethodne.



Slika 3. Izgled stranice na kojoj se nalaze lekcije.

Prelaskom miša preko reči koje su na srpskom jeziku na ekranu se prikazuje "hint" sa tekstom na engleskom jeziku. Implementacija ovog svojstva zahteva podelu

rečenice na reči oko kojih se stavljaju nevidljive granice njihove dužine i širine. Prilikom prelaska miša poziva se funkcija koja određuje prevod date reči.

Završetak svake lekcije predstavljaju priče. One su vrhunac lekcija. Na praktičan način prelaze gradivo koje je učeno pre njih kao vid finalne provere. Svaka je sačinjena od jedinstvenih grafika i primera za testiranje koji nisu pravljeni po šablonu, već zavise od konteksta same priče. Priče prate turistu koji putujući kroz Srbiju uči jezik i upoznaje se sa kulturom države na autentičan način. Zbog kratkog broja lekcija prototipa nalazi se samo u Beogradu. Kroz kadrove njegovog putovanja korisnici su izloženi svakodnevnim događajima i čestim vidovima interakcija.

Na slici 4. prikazan je meni restorana. Turista, nakon određenog dijaloga odlazi u restoran u kome je dat zadatak da se na osnovu slika i teksta u meniju popune prazna mesta. Ovakav deo priče nalazi se posle lekcije koja je posvećenja širenju vokabulara, odnosno reči vezane za hranu.



Slika 4. Izgled menija iz interaktivne priče.

U prvom planu aplikacije jeste vizuelni dizajn. Za izradu grafika autor je koristio program vektorske grafike, *Adobe Illustrator* [9], što se ogleda kroz jednostavan i minimalistički stil aplikacije, kao i program rasterske grafike, *Adobe Photoshop* [10]. Osim ovih programa putem funkcija koda, omogućenih od strane biblioteka programskog jezika C# urađeni su određeni elementi.

III. Zaključak i pravci budućeg razvoja

Aplikacija "*Serbiapp*" je još uvek u fazi razvoja i nije komercijalno dostupna za preuzimanje i korišćenje. Autor je implementirao za potrebe projektnog rada iz predmeta Objektno orijentisano programiranje. Aplikacija se može koristiti za učenje srpskog jezika. Interaktivna je, jednostavna za korišćenje i pruža nekoliko zanimljivih funkcionalnosti koje je izdvajaju od ostalih aplikacija trenutno dostupnih za učenje jezika.

U narednim verzijama, koje će predstavljati nadogradnju postojeće aplikacije autor ima u planu da implementira nove funkcionalnosti. Jedna od njih je kreiranje grafa koji bi pružao informaciju o efikasnosti načina učenja. Drugi pravac budućeg istraživanja je kreiranje grafa prilagođavanja težine programa na osnovu uspeha korisnika, što bi predstavljalo veoma dobru povratnu informaciju korisniku o njegovom napredovanju i ostvarenju ishoda učenja.

Takođe, naredne verije aplikacije imaće posebnu funkcionalnost u vidu korišćenja *Google API*-ja za zadatak u kome je potrebno da se mikrofonom izgovori rečenica na srpskom jeziku. Korisnik izgovorom rečenice iz datog zadatka kreira audio fajl koji se putem *Google Cloud API* servisa pretvara u reči i proverava nivo podudarnosti izgovorene rečenice sa rečenicom iz primera. Ako nivo podudarnosti prelazi određeni stepen prihvata se kao tačan zadatak. Osnovna sintaksička pravila jezika i pravilan izgovor sastavni su deo učenja jezika. Zbog toga autor smatra da će ovo biti jedna od važnih tema za buduća istraživanja.

Najvažniji, ali i najteži izazov za buduće verzije aplikacije predstavlja dodatak novog sadržaja. Lekcije moraju biti međusobno povezane, a njihov hronološki redosled smislen. Svaka lekcija zahteva nove primere, grafike i kreativne načine objašnjenja složenih tema na jednostavan način.

Literatura

[1] V. Nejković and D. Andrejević, "The Impact of the Use of Clicker Tools in Academic Classroom Environments", *J. Teach. English Specif. Acad. Purp.*, vol. 10, no. 3, 2022, pp. 407-420.
[2] J. Šafranj, M. Katić, and J. Zivlak, "Internet platforme za učenje stranog jezika", *Trend*, 2021, no.T2, pp 213-217.
[3] G. Medvedev, "Google Translate in Teaching English", *J. Teach. English Specif. Acad. Purp.*, vol. 4, no. 1, 2016, pp 181-193.
[4] H. A. Fahim, "Relationship between Gender Difference and Students' Perceptions of Google Translate Use and Accuracy along with Difficulties", Babylon, Iraq, 2021.
[5] L. Pticina, "Onlajn podučavanje srpskog kao stranog jezika – prednosti i nedostaci", 2019.
[6] https://en.wikipedia.org/wiki/Flashcard
[7] https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/?view=vs-2022
[8] https://www.audacityteam.org
[9] https://www.adobe.com/products/illustrator.html
[10] https://www.adobe.com/products/photoshop.html

Uređaj za zaštitu električnih potrošača u domaćinstvu

Nemanja Živković

Sadržaj – Uređaj služi za zaštitu električnih potrošača u domaćinstvu od previsokog napona gradske mreže koja može da nastane usled posledica atmosferskih pražnjenja. Princip rada uređaja baziran je na podacima o vremenskoj prognozi koje uređaj prikuplja sa sajta za meteorološku prognozu za određenu geografsku lokaciju, na osnovu kojih odlučuje da li električne potrošače treba isključile iz napajanja kako bi iste zaštitio.

I. Uvod

Atmosferska pražnjenja mogu da nanesu velike štete po električne potrošače i električne instalacije u domaćinstvu. Gromobranske instalacije pak pružaju drugu vrstu zaštite od atmosferskih pražnjenja i ne mogu da zaštite električne potrošače u domaćinstvu od indukovanog napona na liniji distributivne mreže od razvodnog postrojenja do domaćinstva. Ručno isključivanje potrošača za napajanja zahteva fizički posao koji nije ni moguć ukoliko se u domaćinstvu ne nalaze osobe koje bi isključili sve potrošače sa napajanja.

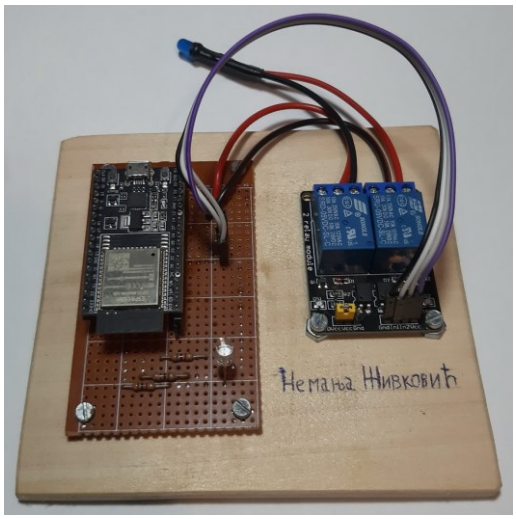
II. Princip rada

Princip rada uređaja baziran je na vremenskoj prognozi koje uređaj prikuplja sa meteorološkog sajta[1] preko lokalne Wi-Fi mreže. Na osnovu prikupljenih podataka i zadatog algoritma uređaj procenjuje da li postoji verovatnoća da dođe do atmosferskih pražnjenja za zadatu geografsku lokaciju i na osnovu toga odlučuje da li će električne potrošače u domaćinstvu isključiti sa napajanja. Potrebne podatke za rad uređaja, poput lokacije i šifre za Wi-Fi konekciju unosi korisnik. Jednom uneti podaci ostaju sačuvani i nakon nestanka napajanja čime se izbegava potreba za njihovim ponovnim unošenjem.

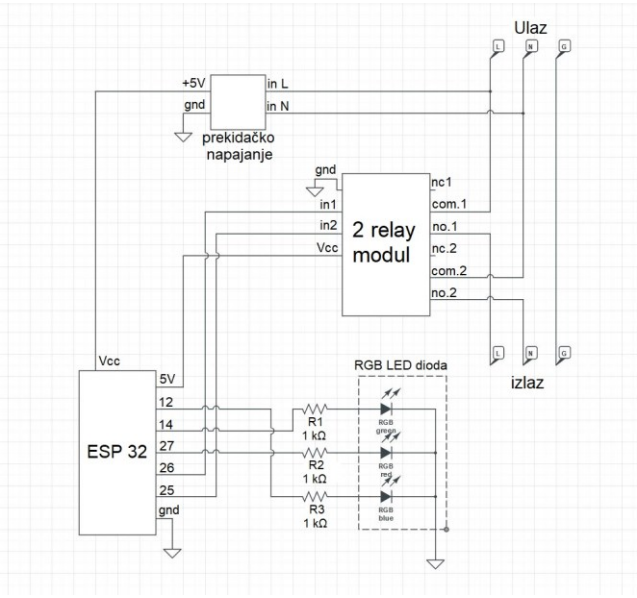
III. Hardver uređaja

Hardver uređaja (slika 1) baziran je na mikrokontroleru ESP 32[2]. Upravljanje napajanjem potrošača realizovano je preko releja, a o režimu rada uređaja korisnik se obaveštava pomoću RGB led diode. Uređaj se napaja preko prekidačkog napajanja sa napona gradske mreže (slika 2).

N. Živković, učenik Tehničke škole u Loznici, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, e-mail: nemaya015@gmail.com



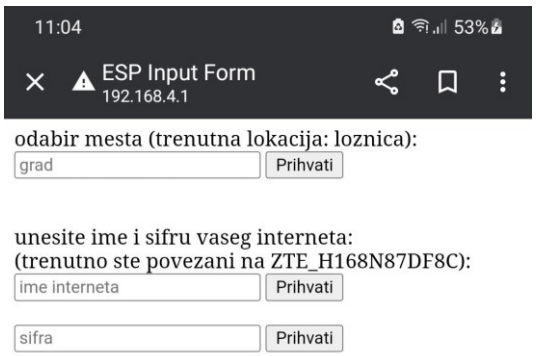
Slika 1. Izgled uređaja.



Slika 2. Električna šema uređaja.

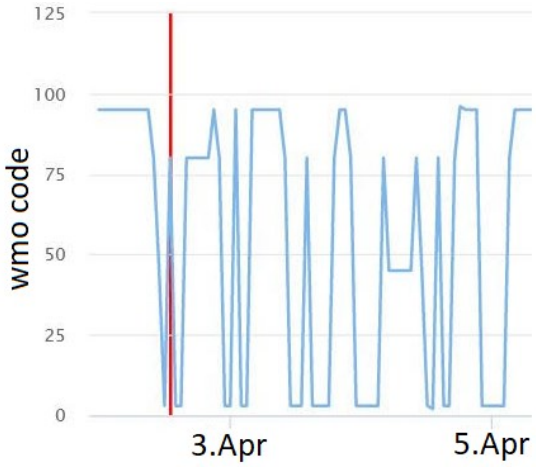
IV. Inicijalizacija i opis rada uređaja

Inicijalizacija uređaja, odnosno, unos podataka potrebnih za rad uređaja vrši se preko internet stranice koju kreira mikrokontroler ESP 32 na svojoj mreži, na adresi 192.168.4.1. Unos podataka vrši sa drugim uređajem poput laptopa, mobilnog telefona, tableta ili drugih uređaja sa Wi-Fi interfejsom, bez instaliranja novih aplikacija i fizičke veze između uređaja (slika 3).



Slika 3. Pristup web-stranici uređaja.

Na osnovu podataka unetih od strane korisnika uređaj će se povezati na Wi-Fi mrežu pomoću podataka koje je korisnik prethodno uneo. Kada je uređaj povezan na mrežu RGB led dioda će emitovati zelenu boju. Ako uređaj iz nekih razloga nije uspeo da se ostvari internet konekciju, RGB led dioda će emitovati crvenu boju. Ukoliko se internet konekcija prekine tokom rada uređaj će ugasiti potrošač i pokušavaće da ponovo ostvari konekciju. Podatke o vremenskoj prognozi uređaj dobija sa sajta "Open-meteo"[1]. Na osnovu podataka o geografskoj širini i dužini lokacije gde će uređaj biti upotrebljen. Podaci o vremenskoj prognozi uređaju se šalju u vidu JSON[3] formata podataka, a uređaj analizira dobijene podatke u WMO[4] kodu.



Slika 4. Meteorološki podaci u WMO kodu sa lažnim signalima.

Na slici 4 prikazan je WMO kod za određenu geografsku lokaciju i predstojeći vremenski period. Svaka vrednost na vertikalnoj osi predastavljenog grafika ima značenje određene vremenske prognoze. Analizirajući dobije podatke uređaj zaključije da li će i u kom vremenskom peruodu doći do vremenskih nepogoda. Preko releja uređaj uključuje ili isključuje uređaje sa napajanja kako bi iste zaštitio od potencijalnih oštećenja izazvanih nadolazećim vremenskim neprilikama. Kako dobijeni podaci vremenske prognoze nisu sasvim tačni, odnosno dobijeni podaci mogu i da odstupaju od stvarnih meteoroloških prilika, uređaj obavlja i softversko filtriranje podataka tako što će reagovati tek na višestruke i uzastopne iste vremenske prognoze.

Uređaji u domaćinstvu se na napajanje priključuju preko uređaja.

Za programiranje mikrokontrolera ESP 32 koji nema svoj softver za programiranje može se koristiti bilo koje programsko okruženje sa potrebnim bibliotekama. Zbog lako dostupnih dokumentacija u datom slučaju mikrokontroler je programiran pomoću programa Arduino IDE[5]. Kako bi mikokontroler ESP 32 radio i kao access point i kao klijent drugom access point-u korišćena je biblioteka <WiFi.h>[6]. Za kreiranje internet stranice mikrokontrolera i unos potrebnih podataka, za rad uređaja korišćena je biblioteka <AsyncTCP.h>[7] i <ESPAsyncWebServer.h>[8].

V. Zaključak

Atmosferska pražnjenja su relativno česta pojava, naročito u letnjim mesecima. Energija koja se oslobađa usled atmosferskih pražnjenja je ogromna a njene posledice na uređaje u domaćinstvu mogu biti i takve da dovedu do trajnog otkazivanja uređaja. Sa datim uređajem se, uz relativno nisku cenu i relativno dobru pouzdanost, pruža zaštita uređaja u domaćinstvu, naročito u situacijama kada u domaćinstvu nema osoba koje mogu da, umesto uređaja, isključe sve uređaje sa napajanja.

Zahvalnica

Zahvalnost na realizaciji ovog istraživačkog rada uputio bih Tehničkoj školi u Loznici i mom mentoru, profesoru dipl. inž. elek. Tomislavu Mijiću na savetima i pomoći prilkom izrade i realizacije rada.

Literatura

[1] https://open-meteo.com
[2] https://en.wikipedia.org/wiki/ESP32
[3] https://www.json.org/json-en.html
[4] https://www.weather.gov/bmx/wmocodes
[5] https://sr.wikipedia.org/sr-el/Arduino
[6] https://github.com/arduino-libraries/WiFi
[7] https://github.com/me-no-dev/AsyncTCP
[8] https://github.com/me-no-dev/ESPAsyncWebServer

Vizuelno-zvučno unapređen sistem bezbednosti

Mihajlo Radivojević, David Petrović, Lazar Cvetković, Lazar Živković

Sadržaj – Naš projekat podrazumeva postojanje adekvatne vizuelno-zvučne signalizacije za dve grupe korisnika. Model našeg sistema predstavlja objedinjen koncept Ustanova za osobe oštećenog vida/sluha. Sa željom da im omogućimo brzu i bezbednu evakuaciju u slučaju poplave i požara, kreirali smo poseban sistem signalizacije.

I. UVOD

Cilj savremenog društva je integracija svakog pojedinca kako u smislu normalnog funkcionisanja, tako i u smislu bezbednosti. Po statističkom izveštaju Ministarstva za rad , zapošljavanje, boračka i socijalna pitanja Republike Srbije u našoj zemlji postoji 74 ustanove socijalne zaštite, gde spadaju i ustanove i domovi za rad sa osobama sa posebnim potrebama. Veliki broj takvh ustanova naveo nas je na ideju o unapređenom sistemu bezbednosti, posebno u Ustanovama za osobe oštećenog vida i sluha. Na osnovu istraživanja u Nišu, u Međuopštinskoj organizaciji slepih , kao i u Humanitarnom centru za sluh-HCZS, uvideli smo potrebu za unapređenim i adekvatnim sistemom bezbednosti, posebno u vanrednim okolnostima . To je sistem koji se ugrađuje u prostorije u kojima svakodnevno borave navedena lica. Putem senzora, sistem se aktivira u vanrednim okolnostima na prisustvo vode ili dima u prostoriji. Vizuelno-zvučnom signalizacijom pomoću LED-a, odnosno zujalice, sistem ima mogućnost brzog i efikasnog navođenja korisnika ka bezbednoj i brzoj evakuaciji.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Model sistema je dizajniran tako da su sa jedne strane korisnici sa oslabljenom funkcijom vida, a sa druge korisnici sa oslabljenom funkcijom sluha. Na ulazu u objekat se nalaze dva ultra-zvučna senzora, koji registruju broj osoba prilikom ulaza i izlaza.

Ovi senzori su postavljeni i u prostorijama koje se nalaze na drugoj strani modela, a predviđene su za zajednički boravak određenih grupa korisnika. Ultra-zvučni senzori u tim prostorijama se aktiviraju jedino u vanrednim okolnostima. Na slici 1 je označena staza za slepe i ta strana modela predstavlja deo za slabovide, dok druga strana obuhvata deo za osobe oštećenog sluha. U svakoj od navedenih prostorija se nalaze senzori za detekciju dima i vode. U slučaju aktiviranja nekog od tih senzora, aktivira se i celokupan sistem. Na strani modela predviđenoj za osobe oštećenog vida, aktiviraju se zvučnici, koji se nalaze na izlazima iz tih prostorija. Intenzitet zvuka se povećava od prostorija u kojima borave, do izlaza za evakuaciju. Tom prilikom se korisnici bliži glavnom ulazu, zvučnim signalom usmeravaju ka tim vratima za evakuaciju i očitava se broj osoba koje su napustile objekat. Korisnici koji su udaljeni od glavnog ulaza, usmeravaju se ka izlaznim vratima iz zajedničke prostorije. U tom trenutku se ta vrata i otvaraju, pri čemu se registruje broj osoba koje su napustile objekat. Druga strana modela je projektovana za osobe oštećenog sluha. U slučaju aktiviranja nekog od senzora dima ili vode, aktivira se i signalizacija za ove korisnike. Sa unutrašnje strane prostorija za ove korisnike nalaze se trepćuća crvena svetla upozoravajući na opasnost, dok se sa spoljne strane nalaze LED svetla u obliku strelica koja pokazuju pravac evakuacije. Korisnike u prostorijama bližim glavnom ulazu LED svetla usmeravaju ka tim vratima i istovremeno se očitava broj osoba koje su napustile objekat . Korisnici koji su u delu bližem zajedničkim prostorijama, strelice navode ka izlaznim vratima koja se u tom momentu otvaraju. Pritom, ultra-zvučni senzor registruje broj osoba koje su napustile objekat. Na ovaj način ne dolazi do paničnog kretanja svih učesnika u vanrednim okolnostima, već se evakuacija kontrolisano odvija.



Slika 1. Model gotovog projekta.

III. KOMPONENTE

1. Arduino Uno(2) (slika 2) – razvojno okruženje koje koristi ATmega328P mikrokontroler. Radni napon je 5 V, ima 14 digitalnih I/O pinova, 6 analognih ulaza.

M. Radivojević, D. Petrović, L. Cvetković i L. Živković su studenti prve godine Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ivana.bog75@gmail.com



Slika 2. Arduino Uno.

2. Ultra-zvučni senzor (slika 3) - ovo je modul za merenje udaljenosti ultra-zvučnog senzora od prepreke i predstavlja beskontaktni modul. Ima visoku tačnost merenja i njegova slepa oblast(senzor ne čita precizno) je blizu (do 2cm).



Slika 3. Ultra-Zvučni senzor.

3. LED (slika 4) – svetleća dioda, čija vrednost struje ne bi trebalo da prelazi 20mA, dok je idealna vrednost struje između 16 i 18 mA. Jačinu struje kontrolišemo pomoću otpornika, koji vezujemo redno. Za naš sistem koristimo LED crvene boje.



Slika 4. LED diode.

4. Servo motor (slika 5) - je komponenta koja može rotirati svoju ruku(obično između 0 i 180 stepeni). Koristi se da kontroliše ugaonu poziciju objekta.



Slika 5. Servo motor.

5. Zujalica (slika 6) - stvara zujanje ili zvučni signal. Složenije zujalice uključuju sklop oscilatora i piezoelektrični element ili zujalica u jednom paketu, tako da sve što trebate učiniti je da promenite napon i dobićete zvuk zujanja.



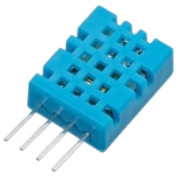
Slika 6. Zujalica.

6. Senzor za detekciju vode (slika 7) – radni napon senzora je od 3,3 do 5V. Radna struja ovog senzora ne bi trebala biti veća od 20mA. Pomoću ovog senzora moguće je izmeriti i nivo tečnosti u određenom sudu.



Slika 7. Senzor za detekciju vode.

7. Senzor temperature (slika 8) - Arduino senzor za temperaturu obično se sastoji od termistora, elektronske komponente, koja menja svoj otpor u zavisnosti od temperature. Kada se termistor zagreje, otpor se smanjuje, a kada se ohladi, otpor se povećava. Ova promenjiva otpornost meri se pomoću analognog ulaza na Arduinu, a zatim se preko programskog koda pretvara u vrednosti temperature u Celzijusima.



Slika 8. Senzor temperature.

IV. Rezultati testiranja i primene

Radi realizacije ovakvog sistema, upoznali smo se sa konceptom evakuacije u Savezu slepih u Nišu i u Humanitarnom centru za sluh u Pasi Poljani, gde smo i testirali naš projekat. U fazi testiranja radili smo sa po 5 osoba iz obe organizacije i improvizovali smo naš sistem. Za zvučnu signalizaciju koristili smo zvuk sirene sa telefona, a za vizuelnu signalizaciju baterijske lampe koje smo postavili na kontrolnim mestima. Celokupan proces se odvijao efikasno i brzo. Svi ispitanici u ovom procesu imali su jasno definisan

cilj i pokazali izuzetnu saradnju. Vreme potrebno za evakuaciju u obema ustanovama bilo je kraće od minuta, što ukazuje na opravdanost instaliranja ovakvog sistema.

V. ZAKLJUČAK

Ovim projektom želimo da unapredimo standardni način evakuacije za osobe sa posebnim potrebama. Prednost ovog projekta je efikasnije i brže reagovanje u vanrednim situacijama. U zavisnosti od toga koliki je prostor , koje su specijalne potrebe korisnika, projekat je podložan promenama. U cilju unapređivanja sistema postoji dosta mogućnosti. Jedna od njih je i postojanje posebne baze podataka sa korisnicima usluga , kako bi se prilikom evakuacije registrovali konkretni korisnici. Ovaj model sistema imao bi primenu u staračkim domovima, u Ustanovama za osobe sa posebnim potrebama i dr. Smatramo da je neophodno imati ovakav sistem ne samo u specijalizovanim ustanovama , već i u ostalim ustanovama i organizacijama , kako bi se postiglo efikasno reagovanje svih učesnika. Cilj našeg projekta je jednako pružanje sigurnosti u uslovima svakodnevnog života.

ZAHVALNICA

Autori rada zahvaljuju se Elektrotehničkoj školi „Mija Stanimirović“, koji su nam obezbedili sredstva i uslove za rad, kao i mentoru Ivani Bogdanović, nastavniku u ETŠ „Mija Stanimirović“ na pruženoj podršci. Takođe se zahvaljujemo upravi Saveza slepih u Nišu, kao i upravi Humanitarnog centra za sluh u Pasi Poljani na izuzetnoj saradnji i razumevanju.

LITERATURA

[1] Preuzeto sa stranice https://www.minrzs.gov.rs/sr/aktuelnosti/vesti/informacija-na-da02-05-2020-ustanove-socijalne-zastite-za-smestaj-korisnika-i-organizacije-socijalne-zastite-za-smestaj-korisnika
[2] Sajt Saveza slepih Niš http://mosssnis.org.rs/
[3] Sajt Humanitarnog centra za sluh https://www.hczs.org/
[4] Slike preuzete sa https://www.arduino.cc .

Udaljena OET laboratorija – Pločasti kondenzator

Nikola Petrović, Vladimir Petrović

Sadržaj – U radu je prikazano izvođenje laboratorijske vežbe predmeta Osnove elektrotehnike I u udaljenoj OET laboratoriji, kroz primer određivanja nepoznate dielektrične konstante materijala pločastog kondenzatora. Koriste se dva kondenzatora izrađena od različitog materijala za izradu štampanih ploča, sa ciljem da se izračuna njihova relativna dielektrična konstanta.

I. UVOD

Učenici elektrotehničkih škola se u prvoj godini učenja kroz predmet Osnove elektrotehnike I upoznaju sa pojmovima: polarizacija dielektrika, kondenzatori i kapacitivnost kondenzatora. U izlaganju ovih tema za primer se obično uzima slučaj pločastog kondenzatora, zbog jednostavnosti konstrukcije i načina izračunavanja kapacitivnosti.

Udaljene laboratorije omogućavaju izvođenje eksperimenata i laboratorijskih vežbi bez fizičkog prisustva, korišćenjenjem savremenih informacionih tehnologija [1] [2].

Prikazana laboratorijska vežba omogućava jednostavnu proveru znanja iz navedene oblasti, kroz određivanje relativne dielektrične konstante nepoznatog materijala pločastog kondenzatora.

Oznake, merne jedinice i sl. u ovom radu jednake su oznakama koje se koriste u zvaničnom udžbeniku [3].

II. PLOČASTI KONDENZATOR

Formula za računanje kapacitivnosti pločastog kondenzatora (slika 1) je jednostavna i glasi:

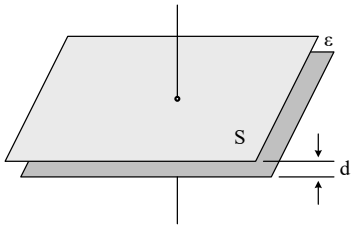
$$C = \varepsilon\,(S/d) = \varepsilon_o\,\varepsilon_r\,(S/d) \qquad (1)$$

gde je:

C	kapacitivnost pločastog kondenzatora
ε	dielektrična konstanta
ε_o	diel. konstanta vakuuma ($8{,}85 \cdot 10^{-12}$ C^2/Nm^2)
ε_r	relativna dielektrična konstanta materijala
S	površina ploča
d	rastojanje između ploča

N. Petrović je učenik IV razreda smera Elektrotehničar informacionih tehnologija ETŠ „Mihajlo Pupin“ u Novom Sadu, Futoška 17, e-mail: nikolainformatika000@gmail.com

V. Petrović je učenik IV razreda smera Elektrotehničar informacionih tehnologija ETŠ „Mihajlo Pupin“ u Novom Sadu, Futoška 17, e-mail: vlada.petrovic300@gmail.com



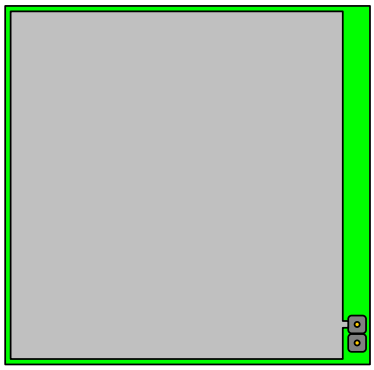
Slika 1. Pločasti kondenzator.

Zadatak vežbe je da se merenjem parametara iz formule (1) odredi relativna dielektrična konstanta ε_r nepoznatog materijala pločastog kondenzatora.

Potrebno je izmeriti kapacitivnost pločastog kondenzatora C, površinu ploča S i rastojanje između ploča d (debljinu materijala). Zatim se računskim putem iz formule (1) dobija relativna dielektrična konstanta ε_r nepoznatog materijala.

III. KOMPONENTE

Za realizaciju laboratorijske vežbe koriste se materijali za izradu štampanih ploča, koji su lako dostupni (u ovom slučaju dva komada dvoslojnih štampanih ploča, jedan od pertinaksa, a drugi od fiberglasa, FR-4, slika 2).

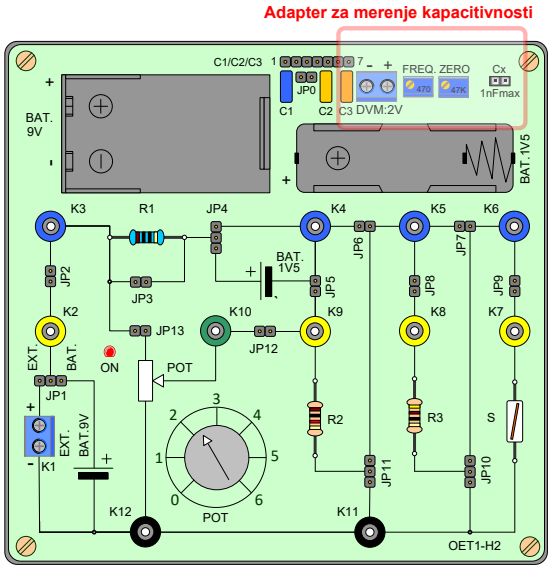


Slika 2. Pločasti kondenzator formiran na štampanoj ploči.

Bakarni slojevi dvoslojne štampane ploče koriste se kao ploče kondenzatora (debljina bakra na ploči je tipično 35 μm, tako da debljina "ploča" ovakvog kondenzatora ne utiče na rezultat merenja rastojanja između ploča d). Površina S ovih bakarnih delova ploče može biti proizvoljna, ali je radi jednostavnosti merenja odabrana površina kvadrata ivice 10 cm. Standardna debljina d korišćenog materijala je 1.6 mm (mogu se koristiti i ostale standardne debljine 0.8 mm, 1 mm, 1.8 mm, 2.3 mm ili 3.2 mm, ako su na raspolaganju).

Merenje kapacitivnosti C ovog pločastog kondenzatora može se obaviti odgovarajućim instrumentom (postoji kod pojedinih modela univerzalnih instrumenata ili kao zaseban instrument – LC metar).

Za potrebe udaljene laboratorije ovde se koristi adapter za merenje kapacitivnosti [4], koji je sastavni deo školske makete za Osnove elektrotehnike I, slika 3.



Slika 3. Adapter za merenje kapacitivnosti na školskoj maketi.

Adapter je kalibrisan tako da na izlazu daje jednosmeran napon direktno proporcionalan kapacitetu koji se meri (na primer, za C = 246 pF, napon na izlazu adaptera je 246 mV).

Merni opseg adaptera za merenje kapacitivnosti kondenzatora je 1 nF, odnosno 1000 pF, što odgovara praktičnim vrednostima kapacitivnosti korišćenih pločastoh kondenzatora (reda 300 pF).

IV. Praktična realizacija

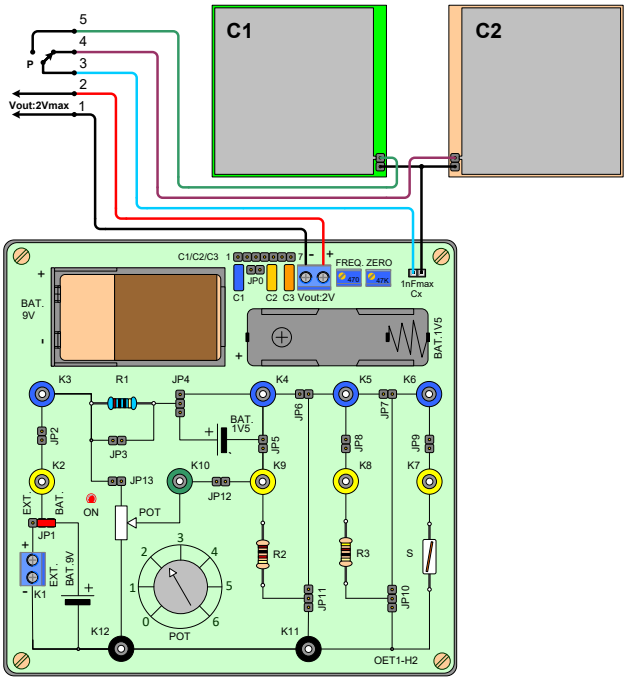
Za potrebe merenja relativne dielektrične konstanta ε_r nepoznatog materijala korišćenjem mogućnosti udaljene laboratorije koriste se dva pločasta kondenzatora (jedan izrađen od pertinaksa, a drugi od fiberglasa, FR-4).

Dimenzije ploča (bakarnog sloja) pločastih kondenzatora su 10 cm x 10 cm. Debljina materijala je 1.6 mm.

Merenje kapacitivnosti pločastog kondenzatora *C1* i *C2* obavlja se uz pomoć adaptera za merenje kapacitivnosti i preklopnika *P* (položaj 5 za merenje *C1*, odnosno položaj 4 za *C2* – umesto preklopnika udaljena laboratorija koristi relejne izlaze), slike 4 i 5.

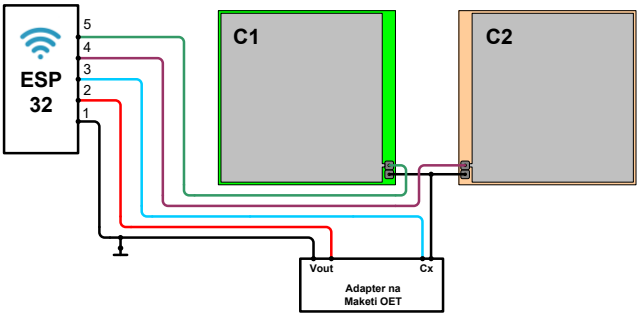
Rezultat merenja kapacitivnosti C (jednosmerni napon sa izlaza adaptera za merenje kapacitivnosti) šalje se u analogno-digitalni konvertor na dalju obradu.

Korišćenje releja za odvojeno uključivanje *C1* i *C2* daje dodatnu mogućnost merenja kapacitivnosti paralelne veze *C1* i *C2* (aktivna oba relejna izlaza), što predstavlja uvod u narednu oblast (vezivanje kondenzatora – redna, paralelna i mešovita veza).



Slika 4. Princip merenja kapacitivnosti pločastog kondenzatora *C1* i *C2*.

V. Hardverska realizacija



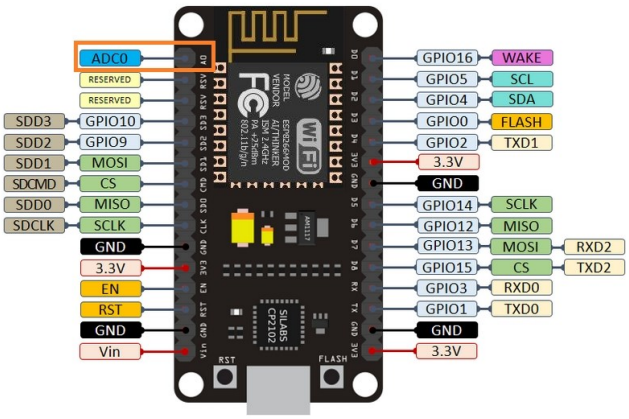
Slika 5. Povezivanje ESP32 modula sa maketom OET.

Oznake na slici 5:

1 GND (*Ground*)
2 Napon sa adaptera za ADC
3 Cx ulaz za merenje kapaciteta na adapteru makete
4 Rele 2 (spaja izlaze 3 i 4 - Cx meri kapacitet C2)
5 Rele 1 (spaja izlaze 3 i 5 - Cx meri kapacitet C1)

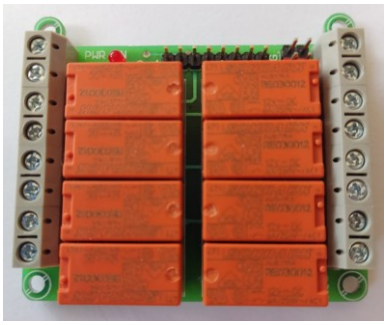
Za upravljanje radom makete, merenje kapacitivnosti i komunikaciju (Wi-Fi) koristi se modul ESP32, slika 6 [5]. On je povezan sa *RealTime FireBase* bazom podataka.

Iz ove baze ESP32 čita trenutna stanja releja (komande za upravljanje radom makete) i šalje povratne informacije o izmerenim naponima (u ovom slučaju napon u mV proporcionalan izmerenoj kapacitivnosti).



Slika 6. ESP 32.

ESP32 ima dodatnu relejnu karticu za upravljanje (slika 7), dok za merenje napona koristi interni AD konvertor rezolucije 12 bita (ulazni napon u opsegu 0 do 3,3 V).



Slika 7. Relejna kartica udaljene laboratorije.

Rezultate AD konverzije (izmereni naponi, u slučaju vežbe Pločasti kondenzator to je izmerena kapacitivnost kondenzatora) šalju se u bazu podataka, odakle ih *web* sajt Udaljene laboratorije čita i prikazuje korisniku.

VI. SOFTVERSKA REALIZACIJA UDALJENE LABORATORIJE

Sajt udaljene laboratorije [6] je realizovan u *Next.js framework*-u, koji je hostovan preko *vercel*-a.

Sajt je povezan na *Firebase Realtime Database* bazu podataka iz koje čita, ali i u koju upisuje podatke.

Prijavljivanje učenika na sajt je realizovano sa *GoogleAuth* autentifikacijom, gde se učenici na sajt prijavljuju putem mail-a.

Zakazivanje termina se upisuje u *Firebase*-ovu *Firestore* bazu podataka, gde se upisuje datum, vreme i *Google user id* učenika. (edited).

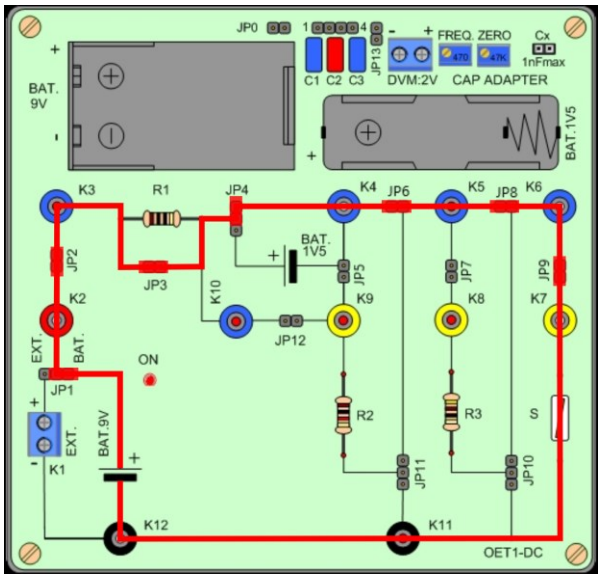


Slika 8. Izgled *web* sajta Udaljene laboratorije.

Osim merenja kapaciteta pločastog kondenzatora, udaljena laboratorija trenutno sadrži i laboratorijsku vežbu – Povezivanje električnog kola sa sijalicom, koja demonstrira i osobinu pozitivnog temperaturnog koeficijenta (PTC) užarenog vlakna same sijalice.

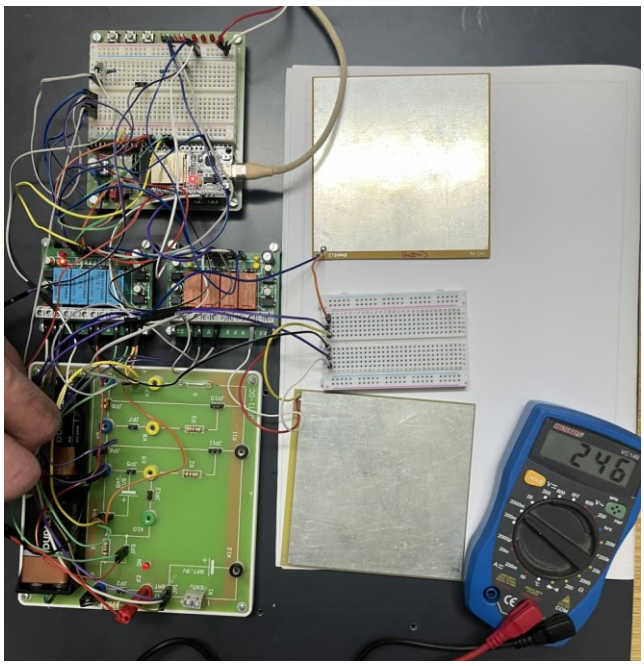
Prilikom spajanja električnog kola korišćenjem udaljene laboratorije učenici imaju pomoć u vidu animiranog prikaza (crvena linija) koji olakšava realizaciju laboratorijske vežbe, kao u primeru na slici 9.

Osim toga, udaljena laboratorija ima i odgovarajuću kameru koja korisniku vizuelno prikazuje aktiviranje komponenti na maketi (rad sijalice, led za aktiviranje releja i sl.)



Slika 9. Primer animacija pri povezivanju električnog kola putem udaljene laboratorije.

U planu je pokrivanje svih laboratorijskih vežbi koje sadrži maketa za OET (ukupno 12).



Slika 10. Testiranje makete za OET – vežba Pločasti kondenzator (voltmetar prikazuje 246 mV, odnosno izmerena kapacitivnost je 246 pF). Vide se i oba pločasta kondenzatora (iznad i ispod protoboard ploče u sredini slike).

VII. Izvođenje vežbe

Tekst i zadatak laboratorijske vežbe učenik preuzima sa sajta [6]. U tekstu su navedeni podaci o površini ploča *S* i debljini materijala pločastih kondenzatora.

Tabela I
Kapacitivnost pločastog kondenzatora

Kapacitivnost	Izmereno (pF)	Rastojanje između ploča d (mm)	Površina ploča S (m^2)	Diel. Konstanta ε	Relativna diel. konstanta εr
C1					
C2					

Zatim se izdaje komanda za merenje kapacitivnosti *C1* i rezultat se upisuje u Tabelu I. Postupak se ponavlja i za kapacitivnost *C2* (ako pločasti kondenzator nije priključen na adapter, merenje se ne vrši). Na osnovu dobijenih vrednosti, učenik računa relativnu dielektričnu konstantu za oba pločasta kondenzatora.

Ako je učenik zakazao termin za izvođenje laboratorijske vežbe, merenje može da obavi ne samo pomoću računara, nego i pomoću mobilnog telefona.

VIII. Zaključak

U radu su prikazane mogućnosti udaljene laboratorije za izvođenje jedne vežbe iz predmeta Osnove elektrotehnike I za prvu godinu elektrotehničke škole.

U uslovima epidemije (korona i sl.), zatim za učenike koji su odsutni ili su propustili vežbu zbog bolesti, za učenike koji slušaju nastavu na daljinu, pa i za škole koje nemaju adekvatno opremljenu laboratoriju, koncept udaljene laboratorije ima velike prednosti.

Ipak, konkretno ova laboratorijska vežba pokazala je i jedan mali nedostatak pristupa sa udaljene lokacije. Za vreme rada u laboratoriji učenici pri merenju dobijaju različite rezultate jer rukama dodiruju ploče kondenzatora prilikom merenja. Na taj način praktično se može demonstrirati i objasniti postojanje parazitnog kapaciteta ljudskog tela i njegov uticaj na tačnost merenja (kod udaljene laboratorije moguće tu pojavu samo simulirati).

Zahvalnica

Autori zahvaljuju mentorima Milanu Vukobratu i Ušan Branislavu na pomoći i podršci u izradi rada.

Literatura

[1] M.Vukobrat, B.Ušan, PiLab – Razvoj laboratorije za udaljene eksperimente, Prezentacija na međunarodnom skupu "Utrip prihodnosti", 2018, Kranj, Slovenija. Dostupno na: http://up.sckr.si/2018/Zbornik_prispevkov_UP_2018.pdf
[2] NeReLa project, 2013, Univerzitet u Kragujevcu. Dostupno na: http://www.nerela.kg.ac.rs/
[3] Grupa autora: Osnove elektrotehnike 1 za I razred elektrotehničke škole, 2008, Zavod za udžbenike, Beograd.
[4] M. Vukobrat, „Maketa OET1-DC za laboratorijske vežbe iz OET“, ETŠ „Mihajlo Pupin“, Novi Sad, Interno uputstvo.
[5] ESP32 - AioT Solution Provider. Dostupno na: https://www.espressif.com/
[6] Sajt Udaljene laboratorije (demo verzija) dostupan na adresi: https://maketa.vercel.app/

Igrica za testiranje refleksa realizovana korišćenjem digitalnih kola

Lazar Arsić

Sadržaj - U današnje vreme deca sve više gube na svojoj refleksivnosti. Zbog toga se javila ideja da se realizuje sistem, u vidu igrice, koji će ujedno povećati brzinu refleksa dece ali i poslužiti kao dobar izvor zabave.

I. Uvod

Na početku je potrebno definisati pravila po kojima igrica funkcioniše. Kada sudija pritisne svoj taster on daje dozvolu igračima da pritisnu svoje tastere i da dobiju poene. Indikator da je sudija dao dozvolu jeste LE dioda koja počinje da svetli. Prvi igrač koji pritisne taster dobija poen. Ovde pre svega važno da igrač ima dobru moć zapažanja i „brze prste“. Nakon što neko od dvojice igrača pritisne svoj taster, njegov brojač će se inkrementirati i stanje brojača se prikazuje na 7-segmentnom displeju. Pobednik je onaj koji prvi stigne do 5 osvojenih partija. U svakoj partiji igrač treba da osvoji 6 poena kako bi pobedio u toj partiji. Kada igrač osvoji 6 poena u trenutnoj partiji, brojači poena u trenutnoj partiji i jednog i drugog igrača se resetuju a stanje brojača za prikaz rezultata u parijama se inkrementira za jedan, i to brojač igrača koji je pobedio u toj partiji. Takođe igrač ne sme da ima ni previse „brze prste“ jer onda može da izgubi poen. Naime, ako igrač pritisne taster pre nego što sudija da dozvolu, stanje brojača rezultata u trenutnoj partiji će se smanjiti za jedan.

II. Realizacija

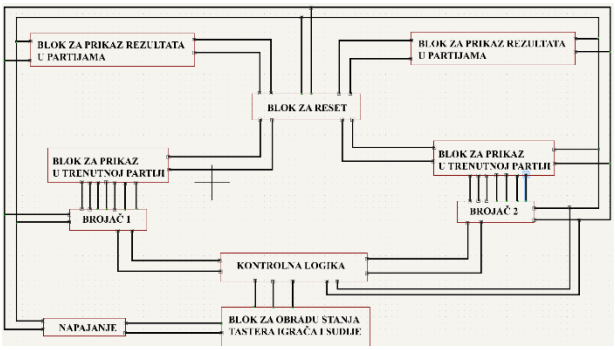
Sama igrica se sastoji iz nekoliko blokova i to:

-Blok za obradu stanja tastera igrača i sudije
-Blok kontrolne logike
-Blok brojača poena u trenutnoj partiji
-Blok za prikaz rezultata u trenutnoj partiji
-Blok za prikaz rezultata u partijama
-Blok za reset brojača i reset celog sistema
-Blok napajanja

Pri realizaciji korišćen je softver Logisim [1] simulator za simulaciju rada blokova i same igrice.

Na slici 1. prikazana je blok šema igrice na kojoj se mogu videti svi blokovi od kojih se igrica sastoji. Neki od blokova su identični i ima ih po dva jer igru igraju dva igrača. U nastavku biće opisani blokovi i njihove funkcije.

L. Arsić je student Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: lazar.arsic@elfak.rs

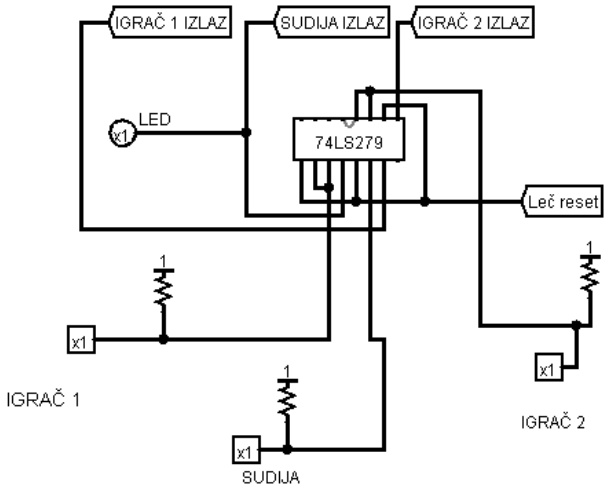


Slika 1. Blok šema igrice.

A. Blok za obradu stanja tastera igrača i sudije

Ovaj blok se sastoji iz dva podbloka, bloka za detekciju pritiska tastera i bloka za resetovanje samih lečeva.

B. Blok za detekciju pritiska taster



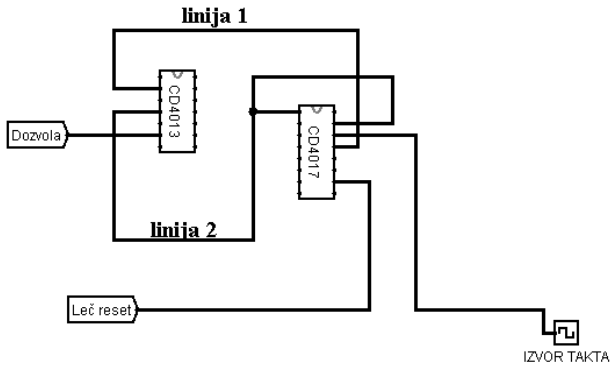
Slika 2. Šema bloka za obradu stanja taster igrača i sudije.

Glavna komponenta ovog bloka jeste IC kolo 74LS279 [2], kolo koje sadrži 4 S-R leča. S-R lečevi se koriste kako bi se eliminisao tzv. *ripple* odnosno poskakivanje napona kada se pritisne taster (*ripple* nastaje zbog nesavršenosti komponenata). Ulazi ovog kola su *active-low*. Šema bloka data je na slici 2.

Kada se pritisne neki od tastera setuje se neki od izlaza kola 74LS279 i to u zavisnosti koji je taster pritisnut. Kako

su ulazi ovog kola *active-low*, tasteri Sudija, Igrač 1 i Igrač 2 su vezani u *pull-up* konfiguraciji.

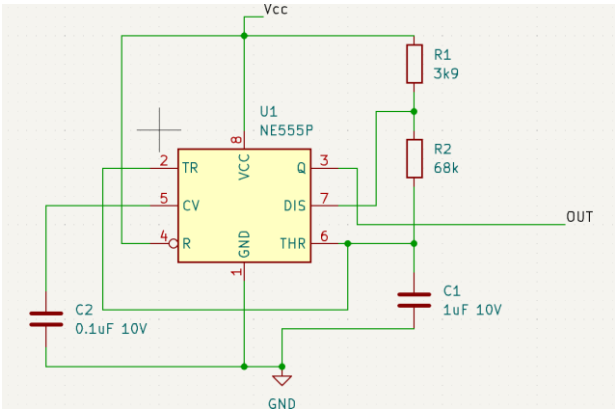
C. Blok za reset SR lečeva



Slika 3. Šema bloka za reset SR lečeva.

Na slici 3. prikazana je šema bloka za reset S-R lečeva. Komponente koje su iskorišćene su CD4013 [3], dvostruki D flip-flop sa mogućnosti asinhronog setovanja i resetovanja, i CD4017 [4], dekadni Džonsonov brojač. Kako ovo zapravo radi? Kada se pritisne taster IGRAČ1, IGRAČ2 ili taster za reset celog sistema, na pin SET D flip-flopa (pin DOZVOLA na slici 3) se dovede logička jedinica koja setuje Q izlaz a resetuje Q komplement flip-flopa. Pin Q kom. flip-flopa je povezan na pin clk.inh kola CD4017 (linija 1 na slici 3), kada se tu dovede logička nula brojač kreće da broji. Izlaz Q4 predstavlja izvor reseta za S-R lečeve (pin Leč reset na slici 3), dok izlaz Q5 resetuje sam brojač i flip-flop (linija 2 na slici 3) tako da je sada na clk.inh pin kola CD4017 logička jedinica pa ovo kolo prestaje da broji.

Izvor takta je realizovan uz pomoć popularnog tajmerskog kola NE555 [5] i to u konfiguraciji astabilnog multivibratora. Šema je prikazana na slici 4.



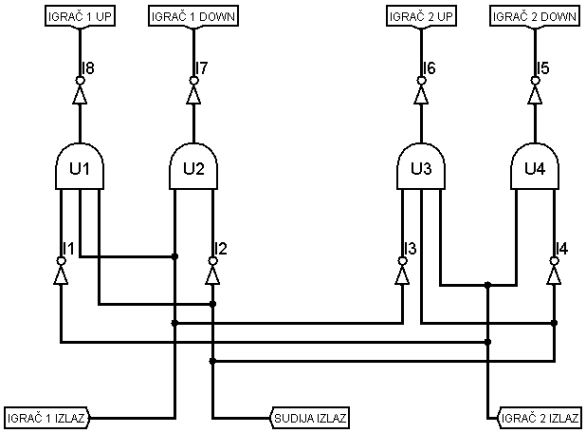
Slika 4. Šema izvora taktnog signala.

Frekvenciju taktnog signala (signal je pravougaonog oblika) određuju otpornici R1 i R2 i kondenzator C1. Formula za računanje frekvencije data je izrazom 1. Frekvencija koja se dobija za vrednosti komponenata kao na šemi iznosi približno 10 Hz.

$$F = \frac{1.44}{(R_1 + 2R_2) * C_1} \quad (1)$$

D. Blok kontrolne logike

Ovo je najbitniji deo celog sistema jer on donosi odluku o tome koji igrač je prvi pritisnuo taster i da li je to po pravilima.

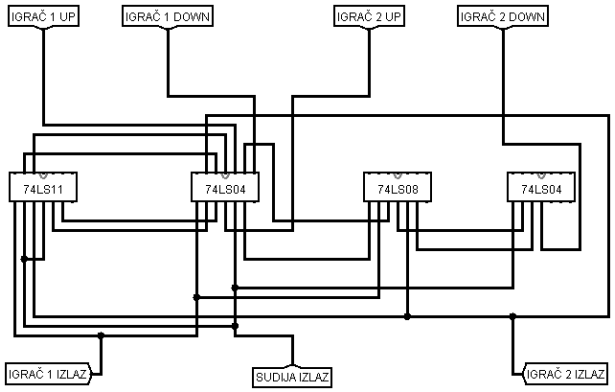


Slika 5. Šema kontrolne logike.

Na slici 5. data je šema povezivanja bloka kontrolne logike. Kao što se može videti ona nije mnogo komplikovana ali je najbitniji deo. Kako radi, kada sudija pritisne taster, izlaz SUDIJA IZLAZ (slika 5.) prelazi u stanje logičke jedinice i daje dozvolu da igrači mogu da pritisnu svoje tastere. Isto tako on preko invertora I2 i I4 blokira rad kola obeleženih sa U2 i U4, i tako onemogućava dekrementiranje brojača. U slučaju da Igrač 1 prvi pritisne taster, on će preko I3 da na ulaz U3 dovede logičku nulu i tako spreči inkrementiranje brojača Igrač 2. Izlaz kola U1 postaje logička jedinica odnosno IGRAČ 1 UP. U slučaju da Igrač 2 prvi pritisne taster dešava se potpuno isto stim da sada on preko invertora I1 blokira igrača 1. Praktično invertori I1 i I3 igraju glavnu ulogu, jer igrač koji prvi pritisne taster on preko invertora blokira drugog igrača.

Međutim ako sudija nije pritisnuo svoj taster, a igrač pritisne svoj, dolazi do dekrementiranja njegovog brojača jer, pošto sudija nije pritisnuo svoj taster na SUDIJA IZLAZ je logička nula i na taj način se blokira rad kola U1 i U3. Ali preko invertora I2 i I4 na ulaze kola U2 i U4 se dovode logičke jedinice pa će izlaz da zavisi od stanja drugog ulaza. Ako se tu dovede logička jedinica doćiće do dekrementiranja brojača igrača koji je pritisnuo taster pre sudije.

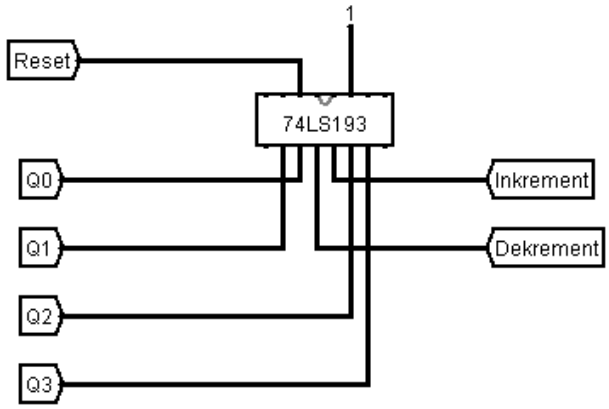
Za praktičnu realizaciju korišćena su kola 74LS11 [6] (trostruko troulazno I kolo), 74LS08 [7] (četvorostruko dvoulazno I kolo) i 74LS04 [8] (u sebi sadrži 6 invertora). Šema sa realnim kolima data je na slici 6.



Slika 6. Šema kontrolne logike sa realnim kolima.

E. Blok brojača poena u trenutnoj partiji

Ovaj blok služi za čuvanje trenutnog stanja u trenutnoj partiji. Komponenta uz pomoć koje je ovaj blok realizovan jeste 74LS193 [9], asinhroni binarni UP/DOWN brojač sa BCD izlazom. Sistem sadrži dva ovakva brojača, za svakog igrača po jedan. Šema bloka prikazana je na slici 7.

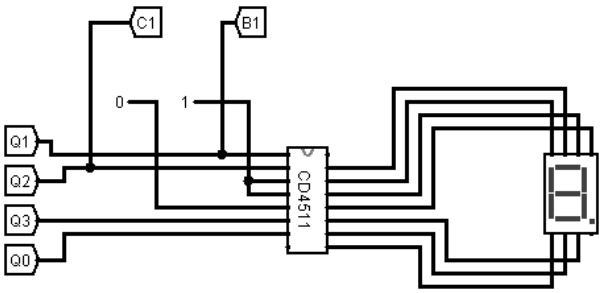


Slika 7. Šema bloka brojača u trenutnoj partiji.

Na pin, na koji je vezana labela Inkrement, se dovodi linija IGRAČ ½ UP i služi za inkrementiranje brojača u slučaju da igrač regularno pritisne taster. Na pin, na koji je vezana labela Dekrement, se dovodi linija IGRAČ ½ DOWN i služi za dekrementiranje brojača u slučaju da igrač pritisne taster pre sudije.

F. Blok za prikaz rezultata u trenutnoj partiji

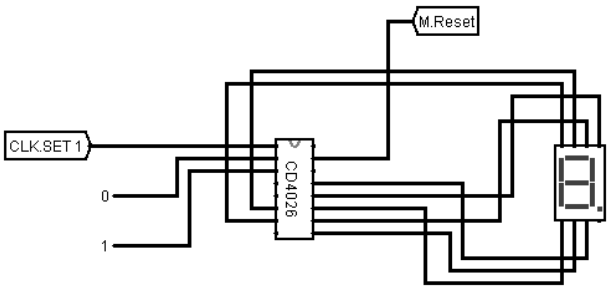
Glavna komponenta ovog bloka jeste kolo CD4511 [10] BCD u 7-segment dekoder za 7-segmentne displeje sa zajedničkom katodom jer su izlazi *active-high*. Na slici 8. je prikazana šema ovog bloka. Pinovi Q0-Q3 dolaze iz bloka brojača i predstavljaju BCD kod trenutnog stanja brojača i njihova uloga će biti objašnjena u tom delu. Sistem sadrži dva ovakva bloka jer se nadovezuju na blokove brojača. Kao što se sa šeme može videti postoje dva pina B1 i C1 koji se vode u blok za reset celog kola.



Slika 8. Šema bloka za prikaz rezultata u trenutnoj partiji.

G. Blok za prikaz rezultata u partijama

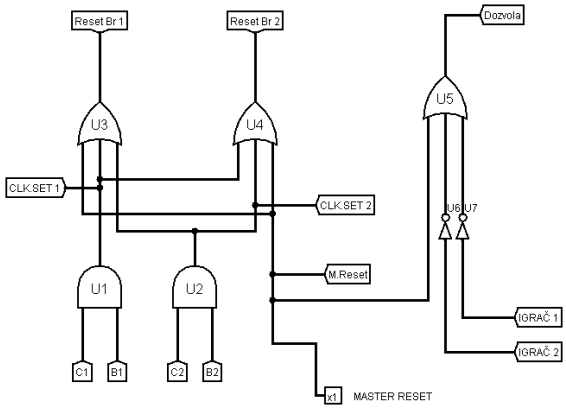
U sklopu šeme povezana su dva ovakva bloka. Kolo uz pomoć kog je realizovan ovaj blok jeste kolo CD4026 [11]. To je dekadni Džonsonov brojač koji u sebi sadrži i konvertor za 7-segmentni displej. Ovi brojač čuva rezultat u partijama, odnosno broj partija u kojima je igrač pobedio. Na slici 9 data je šema ovog bloka.



Slika 9. Šema bloka za prikaz rezultata u partijama.

Pinovi M.Reset i CLK.SET 1 će detaljnije biti objašnjeni u bloku za reset sistema. Konstanta 0 je povezana na pin 2 brojača kako bi on uvek imao dozvolu za brojanje, a konstanta 1 na pin 3 kako bi bilo moguće prikazivanje cifara na displeju.

H. Blok za reset brojača i reset celog sistema



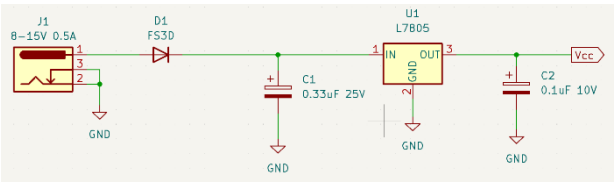
Slika 10. Šema bloka za reset.

Shodno pravilima definisanim u uvodnom delu, kada jedan od igrača dođe do 6 osvojenih poena njegov brojač kao i brojač drugog igrača treba da se resetuju i da krenu da broje od nule i da se pritom promeni stanje u partijama odnosno da se inkrementira vrednost brojača igrača koji je osvojio 6 poena. Kako to radi, izlaz bloka brojača se uvek može predstaviti na sledeći način –DCBA- (izlaz je BCD kod). Brojač tokom brojanja prolazi kroz sledeća stanja: 0000 (BCD kod za nulu), 0001 (jedinica), 0010 (dvojka), 0011 (trojka), 0100 (četvorka), 0101 (petica), 0110(šestica). Kada dođe u stanje 6 kod na izlazu je 0110, mogu se primetiti dve jedinice na mestima B i C, kojih nema u prethodnim stanjima tako da se može reći da kada se tu nađu dve jedinice da je brojač došao u stanje 6. Te dve jedinice su iskorišćene za reset brojača Igrača 1 i Igrača 2, i za takt signal brojača poena u partijama. Šema ovog bloka data je na slici 10. Signali B i C (izlazi Q1 i Q2 bloka brojača) se vode u jedno I kolo, čiji je izlaz u stanju logičke jedinice samo kada su oba ulaza u stanju logičke jedinice, njegov izlaz se zatim vodi u jedno troulazno ILI kolo koje ima funkciju „sabirača“ reset signala za brojač. Takođe njegov izlaz se vodi i na CLK ulaz brojača u partijama kao bi se njegova vrednost inkrementirala za 1. Isto važi i za deo kola Igrača 2.

Funkcija Master Reset – ova funkcija služi da resetuje ceo, sistem i da sve brojače vrati u početno stanje. Ova funkcija se izvršava pritiskom na taster MASTER RESET. Ovaj taster je vezan u *pull-down* konfiguraciji što znači da kada se pritisne on dovodi jedinicu na ulaze kola U3 i U4, kao i na pinove M.Reset.

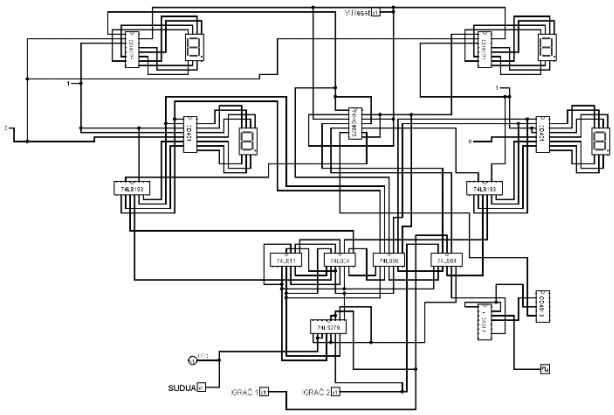
Kolo U5 zajedno sa invertorskim kolima U6 i U7 ima funkciju obezbeđivanja dozvole rada bloka za reset S-R lečeva. S-R lečevi treba da se resetuju kada neko od igrača pritisne taster ili kada se pritisne taster MASTER RESET. Invertori U6 i U7 su tu postavljeni jer su tasteri IGRAČ 1 i IGRAČ 2 vezani u *pull-up* konfiguraciji.

I. Blok napajanja



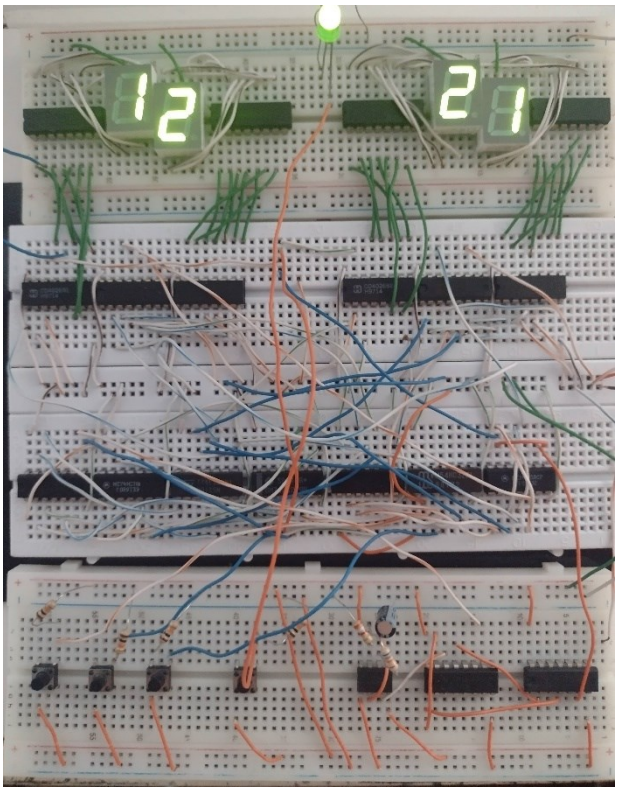
Slika 11. Šema bloka napajanja.

Šema napajanja je prikazana na slici 11. Glavna komponenta je redni stabilizator napona L7805 [12]. Ovom kolu su dodata dva kondenzatora za stabilniji rad samog stabilizatora, njihova vrednost preuzeta je iz tehničke dokumentacije stabilizatora. Na šemi se može videti i dioda FS3D [13] čija je uloga zaštita od suprotnog polariteta.



Slika 12. Kompletna šema igrice.

J. Praktična realizacija



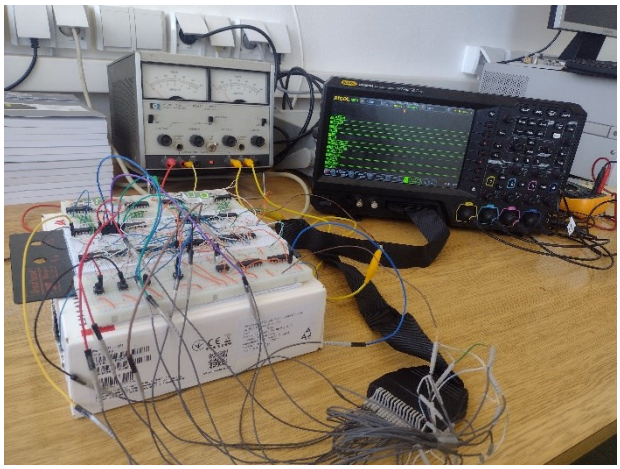
Slika 13. Praktična realizacija igrice.

Prilikom realizacije dodate su otporne mreže 4116R [14] kako bi se zaštitili displeji. Na slici 13. dat je izgled prototipa igrice realizovanog na proto ploči.

K. Testiranje

Nakon praktične realizacije na red je došlo testiranje funkcionalnosti same igrice. Testiranje je urađeno pomoću logičkog analizatora marke RIGOL model MSO5104 [15]. Testiranjem je utvrđeno da prototip zadovoljava pravila koja

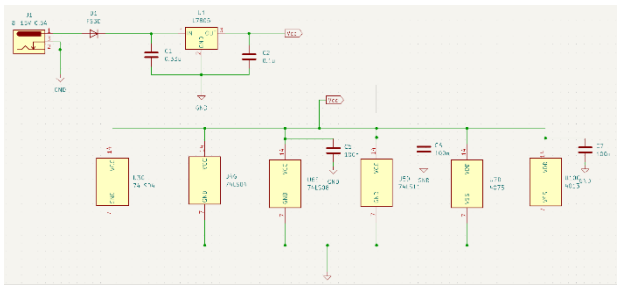
su definisana u uvodnom delu. Na slici 14 je prikazan deo testiranja.



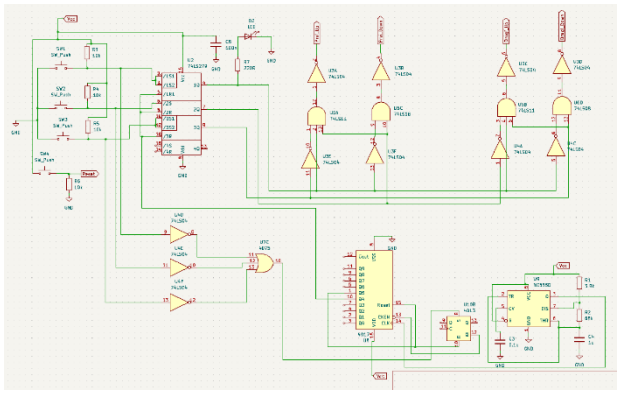
Slika 14. Testiranje.

L. PCB

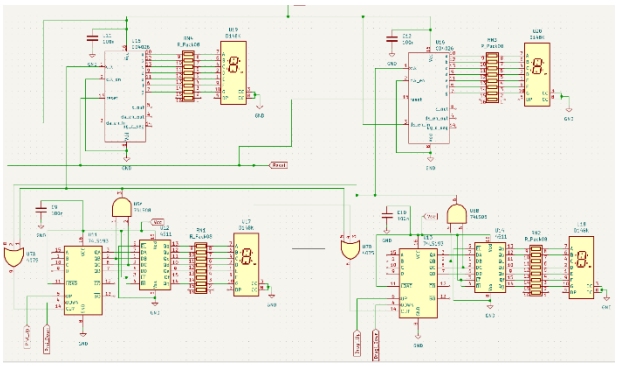
Nakon što je testiranjem utvrđenja funkcionalnost igrice za igricu je kreirana i štampana pločica (PCB). PCB je projektovan u softverskom alatu *KiCad* [16]. Igrica je podeljena u nekoliko celina i to napajanje, kontrolna logika i prikaz. Na slikama 15, 16, 17 prikazane su šeme ovih celina, respektivno.



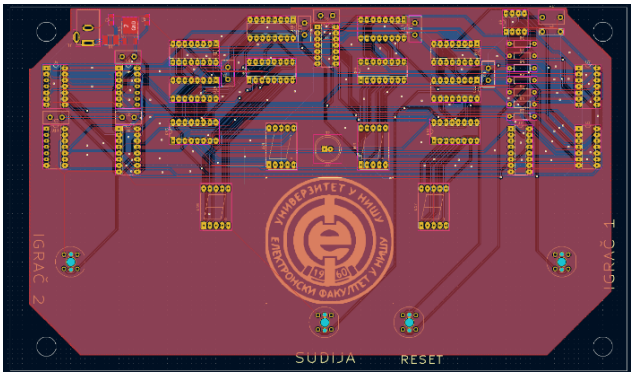
Slika 15. Šema celine napajanje.



Slika 16. Šema celine kontrolna logika.

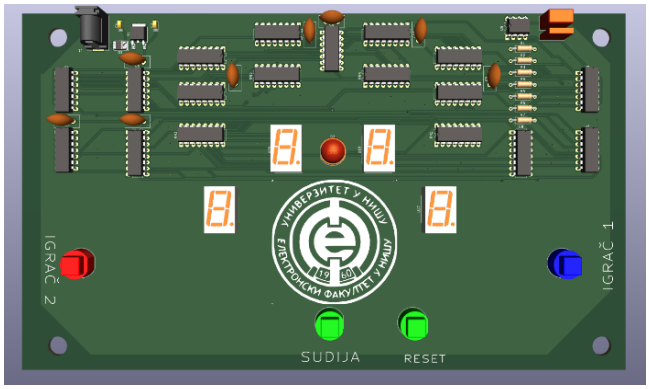


Slika 17. Šema celine prikaz.



Slika 18. Izrutirana štampana ploča.

Na slici 18 prikazana je izrutirana štamapana pločica. Na slici 19 je prikazan 3D model štamapane pločice. Kao što se može videti na ploči se nalaze 4 7-segmentna displeja, led dioda tasteri za oba igrača i sudiju kao i taster za reset celog sistema, a kako je igrica realizovana na elektronskom fakultetu dodat je i logo fakulteta.



Slika 19. 3D model štampane pločice.

III. ZAKLJUČAK

U ovom radu je opisan postupak realizacije igrice koja služi za testiranje brzine refleksa ali isto tako može poslužiti kao dobar izvor zabave. Na početku su blokovi igrice a i sama igrica kao celina realizovani u simulatoru, kada je

utvrđeno da to funkcioniše onda se prešlo na praktičnu realizaciju samog prototipa. Nakon praktične realizacije na red je došlo testiranje. Testiranjem je utvrđeno da prototip ispunjava sve kriterijume koji su unapred bili definisani. Nakon toga je definisana konačna verzija igrice za koju je kreirana štampana ploča (PCB).

ZAHVALNICA

Za pomoć tokom realizacije same igrice autor se zahvaljuje prof. dr Danijelu Dankoviću. Takođe se zahvaljuje mast. inž. Nikoli Mitroviću, asistentu pri Katedri za mikroelektroniku na Elektronskom fakultetu, i prof. dr Aneti Prijić za pomoć tokom testiranja.

LITERATURA

[1] Softverski alat Logisim. Dostupno na: https://sourceforge.net/projects/circuit/
[2] Tehnička dokumentacija za kolo 74LS279. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51065/FAIRCHILD/74LS279.html
[3] Tehnička dokumentacija za kolo CD4013. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/50840/FAIRCHILD/CD4013.html
[4] Tehnička dokumentacija za kolo CD4017. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/26855/TI/CD4017.html
[5] Tehnička dokumentacija za kolo NE555. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/355583/TI/NE555.html
[6] Tehnička dokumentacija za kolo 74LS11. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51027/FAIRCHILD/74LS11.html
[7] Tehnička dokumentacija za kolo 74LS08. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51024/FAIRCHILD/74LS08.html
[8] Tehnička dokumentacija za kolo 74LS04. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51022/FAIRCHILD/74LS04.html
[9] Tehnička dokumentacija za kolo 74LS193. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51049/FAIRCHILD/74LS193.html
[10] Tehnička dokumentacija za kolo CD4511. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/26905/TI/CD4511.html
[11] Tehnička dokumentacija za kolo CD4026. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/26860/TI/CD4026B.html
[12] Tehnička dokumentacija za kolo L7805. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/22615/STMICROELECTRONICS/L7805.html
[13] Tehnička dokumentacija za diodu FS3D. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/157159/ETC1/FS3D.html
[14] Tehnička dokumentacija za otpornu mrežu 4116R. Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/554322/BOURNS/4116R.html
[15] RIGOL-MSO5104, dostupno na: https://www.rigolna.com/products/digital-oscilloscopes/MSO5000/
[16] Softverski alat KiCad. Dostupno na: https://www.kicad.org/

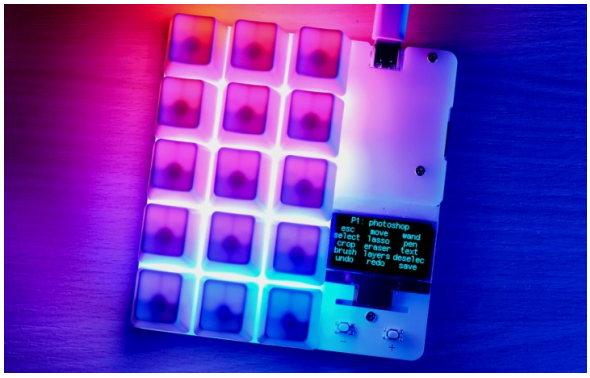
Makro programibilna tastatura (Macro KeyPad)

Nikola Bojinović

Sadržaj - U ovom radu je prikazan Macro KeyPad (makro programibila tastatura), koji služi kako bi ubrzao rad na računaru i povećala produktivnost. Pomoću ove tastature malih dimenzija, jednim prtiskom tastera postiže se mnogo više od otvaranja programa, do samog slanja završenih dokumenata. Sve ovo je postugnuto korišćenjem platforme realizovane *Raspberry Pi Zero* mikrokontrolerom i *CircuitPython* i *KMK Firmware*.

I. Uvod

Najvažnije pitanje izbora Makro (programabilne) tastature (*Macro keypad*) je koliko tastera treba da sadrži. Potrebno je odlučiti za koliko makroa treba da se ima pristup u bilo kom trenutku. Odredi se maksimalni broj makroa, a ako se koristi istovremeno manji broj u tom slučaju, možete dobiti manji Makro KeyPad i osloniti se na njegove slojeve da biste dobili sve komande koje su vam potrebne (slika 1).



Slika 1. Tastatura sa RGB osvetljenjem i OLED displejom.

Ali ako nam treba maksimalni broj makroa sve vreme, trebalo bi razmisliti o nabavci tastature sa dovoljno (ili više) tastera, ili korišćenje softverskog rešenja u vidu slojeva (*layers*). (slika 2.)

II. Princip rada Makro programibilne tastature

Razvijeni Macro keypad - programibilna tastatura funkcioniše pomoću mikrokontrolera tako što se prenese izvorni kod (razvijen u CircuitPython-u) u njegovu memoriju.

N. Bojinović je učenik IV razreda, obrazovni profil: Elektrotehničar računara, Srednja elektrotehnička škola „Mihajlo Pupin“ – Futoška 17, NoviSad, Srbija. E-mail: bojinovicnikola12@gmail.com



Slika 2. Tastatura bez RGB pozadinskog osvetljenja.

Korišćenjem već postojećeg i poznatog *KMK firmware-a* u ovoj kategoriji, može se isporgramirati da tasteri vrše željenu funkciju, bez ikakvih ograničenja.



Slika 3. Naš model Macro Tastature.

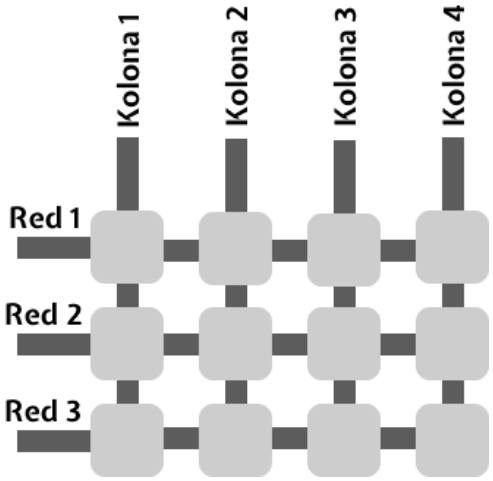
Naš uređaj u konačnoj prototip varijanti će imati 12 (dvanaest) tastera za broj makroa koje pokriva (manji od broja tastera), rotirajuće enkodere za dodatne funkcije (npr. regulisanje jačine tona na zvučniku) i adresabilne RGB svetleće diode za označavanje pojedinog sloja makroa (kada se pređe broj makroa u odnosu na broj tastera)

Nedostatak našeg uređaja u odnosu na konkurente u istom rangu je da ima manji broj tastera, ali ima dodatnu funkcionalnu dugmad (rotirajuće enkodere) što neki modeli nemaju, a manjih je dimenzija za smeštaj u radnom prostoru

oko računara. Za obeležavanje slojeva za pojedinu grupu makroa ima RGB LED diode (nemaju svi konkurenti), a ujedno je efektnog izgleda pri aktiviranju diode.
Delovi su:
- Taktikal svičevi (mehanički tasteri)
- Gornji poklopac sa rupama za tastere i enkodere
- Donji poklopac uz lak mehanizam skidanja od gornjeg dela
- Mikrokontroler Raspberry Pi Zero sa RP2040
- Rotirajući enkoderi
- RGB (*Red-Green-Blue*) svetleće diode
- USB kabl
- Softver (Izvorni kod i *firmware*)

Tastere (*switch*-eve) smo povezali u *Matrix* principu, ondosno u principu matrica, kako bi bili što ekonomičniji u korišćenju pinova i kako bi bili u mogućnosti da povežemo što više tastera mogućih. Naša tastatura koristi ukupno 12 tastera, odnosno 3 reda po 4 tastera. (slika 4)

Mehanički taster (*Switch*) je taster koji se koristi kod raznih novih *„mehaničkih tastatura“*, zahteva manju silu pritiska, čuje se glasniji i jasan zvuk pritiska, te je praktičniji za korišćenje u ovom slučaju.



Slika 4. Šema Matrix povezivanja svičeva.

Adresne RGB LED (svetleće diode) služe kako bi vizualnim efektom korisniku dali do znanja na kojem tačno sloju radi. Sloj (*layer*) je softversko rešenje manjka tastera. *Primer: Imamo 12 tastera, a treba nam 24, sloj omogućava da svaki taster ima 2 funkcije, na 1. sloju (CTRL + C), na 2. sloju (CTRL + V)*. Svaki sloj će biti prezentovan jednom bojom, a RGB LED-ovi će svetleti u toj boji zavisno od sloja koji je aktivan. Isto tako daje i lep vizuelni efekat na estetici samog proizvoda. Nulti prototip je za sada bez ovih komponenti (Slika 5.)

U našoj verziji 1. Taster će otvarati GitHub stranicu, 2.Taster će otvarati Gmail aplikaciju, 3.Taster Arduino IDE, itd. Prvi red će biti za otvaranje aplikacije i *Web* stranica, a već 2. i 3. red za odredjene funkcije
(CTRL + C), (CTRL + V), itd.



Slika 5. Naš model Macro Tastature.

III. Zaključak

Opšte karakteristike uređaja su: da je prenosiv, jednostavan za korišćenje, malih dimenzija. Upotrebljiv je praktično za razne aktivnosti svih korisnika računara (programiranje, obiman unos teksta, intezivan rad na računaru u kratkom vremenskom periodu i dr.)

Rešenje je bazirano na hardversko/softverskom mikrokontrolerskom okruženju, optimizovano u oblasti primene elektronskih i drugih (neelektričnih) komponenti. Obuhvata prilagođeni, razvijeni programski kod. Prototip kućišta je projektovan i izrađen u 3D štampi, a u pripremi je izrada varijanta transparentnog.

Zahvalnica

Autori se posebno zahvaljuju Novici Gutoviću (mentor), rukovodioc sekcije „Arduino“ u ETŠ „Mihajlo Pupin“- Novi Sad, na intenzivnoj tehničkoj podršci i veoma uspešnoj saradnji.

Takođe, autori se zahvaljuju direktoru Ermidinu Elezoviću i Upravi ETŠ „Mihajlo Pupin“- Novi Sad na logističkoj podršci.

Literatura

[1] Ocreb originalna ideja. GitHub repository, dostupno na: https://github.com/sb-ocr/ocreeb-12/tree/main
[2] Ocreb originalni video o ideji. Dostupno na: https://www.youtube.com/watch?v=P_oSLBZABGA
[3] *KMK Firmware* dokumentacija. Dostupno na: https://kmkfw.io/docs/
[4] *CircuitPython* dokumentacija. Dostupno na: https://docs.circuitpython.org/en/8.2.x/README.html
[5] *GitHub repository* našeg modela. Dostupno na: https://github.com/DzoniCodec06/Macro-Key-Pad-RP2040
[6] Razni ralizovani projekti u *Arduino* okruženju dostupno u: „ARDUINO“ UNO - 45 projekata za početnike i stručnjake“, Bert van Dam, InfoElektronika, 2016.

Pregled konvergencije proširene stvarnosti sa sveprisutnim učenjem radi unapređenja obrazovanja

Kristina Petrović

Sadržaj – Kao rezultat ogromnih promena koje je donela digitalna revolucija, sada postoji više načina nego ikad da se učenici angažuju i podstaknu na učenje osim onog tradicionalnog u učionici. Obezbeđivanjem pristupa resursima za učenje u bilo koje vreme i bilo gde, platforme za e-učenje su već promenile način obrazovanja. U ovom radu je obrađen pojam i princip sveprisutnog učenja, šta ono obuhvata i kakve promene donosi u odnosu na e-učenje. Kako je proširena realnost (PR) mlada tehnologija koja se koristi u sveprisutnom učenju, postavlja se pitanje, na koji način, ona, kao tehnologija koja kombinuje stvarne i virtuelne predmete u trodimenzionalnom svetu, može imati uticaja na obrazovanje. U daljem toku rada, fokus je stavljen na to kakav je značaj PR-a u današnjem obrazovanju učenika, koje su prednosti, a koji izazovi sa kojima se susreće, kolika je stopa njenog korišćenja u obrazovnom kontekstu i kakav je uticaj na trajnost informacija kod učenika koji ih usvajaju na ovakav način [1].

I. UVOD

Velike izazove u obrazovanju, uzrokovala je prekomerna upotreba pametnih telefona, a kod mladih su sve učestaliji i poremećaji u procesu učenja. Sa druge strane, ovakvo zapažanje je podstaklo na razmatranje koji su to načini da se proces učenja unapredi, a ujedno motivišu i zainteresuju učenici [2]. U cilju da se zadovolje različite potrebe učenika pri učenju i da bi se ispunila obrazovna očekivanja, nove aplikacije se aktivno uključuju u ovaj segment života. Kao rezultat, učenje unapređeno tehnologijom može biti ključna komponenta 21. veka [3].

Obezbeđivanjem pristupa resursima za učenje u bilo koje vreme, bilo gde, platforme za e-učenje su već promenile način obrazovanja. Korišćenje multimedijalnih resursa, učešće u online razgovorima/predavanjima i polaganju (delova) ispita, bile su inovacije koje je sa sobom donelo e-učenje, što je bilo od koristi pogotovo tokom pandemije virusa COVID-19 [4]. Ovakvim formatima e-učenja obično nedostaju interaktivni elementi.

Tehnologija proširene realnosti (PR) predstavlja potencijalan način za premošćavanje izloženog nedostatka. PR pridodaje virtuelne komponente stvarnom svetu, što rezultira stvaranjem dinamičkog i zanimljivog okruženja za učenje. Korišćenjem proširene realnosti, učenici mogu da istražuju virtuelne predmete, igraju se simulacijama i savladavaju komplikovane ideje na konkretniji i dinamičniji način, poboljšavajući svoje razumevanje i pamćenje nastavnog materijala [5].

U ovom radu, će biti pregledana relevantna literatura, postojeće studije slučaja kao i empirijski eksperimenti kako bi se istražile karakteristike ove tehnologije u kontekstu učenja. Bitno je napomenuti da učenje potpomognuto savremenom tehnologijom nije samo po sebi dovoljno da bude uspešnije od tradicionalnog pristupa, već je neophodna dodatna pedagoška sprega između tehnologije i učenika.

K. Petrović studentkinja master studija, na smeru Computer Science, Politecnico di Milano, Piazza Leonardo da Vinci, 32 02 23991 Milano, Italy, E-mail: kristina.petrovic@mail.polimi.it

II. POJAM SVEPRISUTNOG UČENJA

Usvajanje znanja kroz knjige, eksperimente, prošla iskustva i uz pomoć nastavnika, karakteriše tradicionalni način prihvatanja znanja. U 21. veku sa procvatom informatike, komunikacionih i elektronskih tehnologija, modernizovan je i način nastave i učenja. Sada su knjige i predavanja pretvorena u e-knjige, pfd-ove (prenosne formate dokumenata), vebinare, prezentacije, itd.

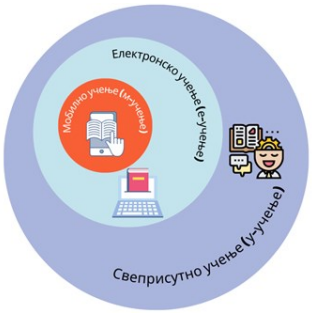
Ovaj tip obrazovnog sistema koji se naziva sistemom e-učenja, osim prethono navedenih mehanizama unapređenja tradicionalnog učenja u vidu dostupnih word/pfd dokumenata i web sadržaja podrazumeva i kontinualne ili striming medijume i autorizovani nastavni sadržaj u vidu kurseva. E-učenje promoviše sticanje informacija, koje su prisutne širom sveta na jednom mestu. Njegov podskup, sistem m-učenja (mobilno učenje), prikuplja informacije bilo gde, u bilo koje vreme. Kao što znamo, mobilni uređaji su elektronski uređaji i spadaju u kategoriju sistema za e-učenje, ali pošto rade na različitim platformama i zahtevaju posebnu vrstu protokola i aplikacija, posebno su klasifikovani. Tehnološki napredak nas je sada osnažio da učimo svuda, svaki put i bilo šta u realnom vremenu. Sveprisutno učenje se može objasniti formulom „s-učenje = e-učenje + m-učenje”, ili integracija m-učenja u okruženje za e-učenje će oblikovati s-učenje [6,7]. Na slici (Slika 1.) se može videti grafička reprezentacija odnosa sveprisutnog učenja, e-učenja i m-učenja.

Sveprisutno okruženje za učenje je kombinacija različitih postavki koje podržavaju učenje korišćenjem tehnologija koje mogu da podstaknu autentično, interaktivno, efektivno iskustvo učenja i u skladu sa stilom učenika bez da je obavezan vremenom i mestom.

Sveprisutno okruženje za učenje je sistem koji je svestan konteksta i može znati/proceniti sposobnosti i predznanje učenika, pa u skladu sa time, mogu ponuditi personalizovane usluge. Zbog toga učenici mogu da steknu

znanja, veštine i rešavaju probleme dok su u interakciji sa stvarnim svetom kroz autentične i interaktivne scenarije. Učenicima je omogućeno da uče u pravo vreme, na pravom mestu, sa pravim alatima i pravim sadržajem. Kao glavne karakteristike sveprisutnog učenja se navode:

1) pristupačnost
2) dostupnost izloženih nastavnih aktivnosti i materijala
3) **svest o kontekstu**
4) interaktivnost,
5) personalizacija
6) fleksibilnost



Slika 1. Grafička reprezentacija odnosa sveprisutnog učenja, e-učenja i m-učenja.

Svest o kontekstu je ovde najširi pojam koji bi po mišljenju nekih mogla da zameni mnoge stavke koje su prethodno navedene. Naime, svest o kontekstu podržava učenike da mogu da pristupe resursima za učenje, sadržaju ili interaktivnoj aktivnosti zasnovanim na njihovoj lokaciji, vremenu i aktivnostima kojima se bave. Svest o kontekstu podržava individualno učenje i nudi mogućnost kreiranja personalizovanog plana i programa rada za izabranu stručnu oblast, testiranje novostečenog znanja, procena naučenog i prilagođenje danjeg rada na osnovu napretka studenta [7].

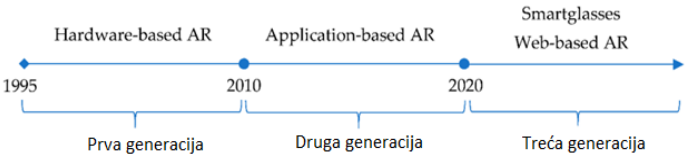
III. PROŠIRENA REALNOST

A. Definicija i taksonomija

Termin proširena realnost (PR) se odnosi na direktan pogled na fizičko, stvarno okruženje koje je poboljšano dodavanjem virtuelnih, kompijuterski generisanih informacija. PR kombinuje stvarne i virtuelne predmete u trodimenzionalnom svetu obezbeđujući interaktivnost [2]. Brojne oblasti, uključujući poslovanje, zabavu, zdravstvenu zaštitu i turizam, imaju koristi od upotrebe ove tehnologije. Stručnjaci predviđaju da će bolje, jeftinije i šire dostupne aplikacije proširene realnosti biti aktuelne u bliskoj budućnosti i da je ovo samo početak. I obrazovanje je imalo koristi od ove tehnologije. Prva aplikacija proširene stvarnosti postoji nešto više od dvadeset i pet godina. Od tada, različiti obrazovni konteksti, oblasti proučavanja i nivoi obrazovanja, uspešno su integrisali aplikacije proširene stvarnosti (PR), koje su se pokazale korisnim za učenike na više načina [8].

B. Tri generacije proširene realnosti u obrazovanju

Predložene su tri generacije aplikacija proširene stvarnosti u obrazovanju (Slika 2.) na osnovu izučavanja njihove istorije [8].



Slika 2. Tri generacije PR u obrazovanju [8].

Prvu generaciju proširene realnosti u obrazovanju (PGPRO) su obeležili skupi i kompleksni sistemi proširene stvarnosti zasnovani na prenosivim, head-mounted i heads-up ekranima. Visoka cena PGPRO je bila uslovljena skupim hardverom, potrebe za posebnim načinom za održavanje istog i zahtevanja specijalizovanih znanja iz programiranja za kreiranje sadržaja [9].

Pošto su postojala druga obrazovna tehnološka rešenja koja su nudila mogućnosti slične onima iz PR edukativnih aplikacija, većina obrazovnih instituciaja nije mogla da ih kupi i nije se potrudila da to učini. Upotrebljivost je bila značajan element koji je uticao i na PGPRO aplikacije.

Upotrebljivost je termin koji opisuje koliko često korisnik stupa u interakciju sa aplikacijom, a ujedno je povezan sa tim koliko je aplikacija jednostavna za korišćenje. Ove aplikacije su predstavljale složene sisteme koji su korisnike stavljale pod preveliko opterećenje. Shodno tome, učenje je bilo ometeno kognitivnim preopterećenjem u pokušaju da se shvate aplikacije [10]. Zbog visoke cene, a malog korišćenja, PGPRO aplikacije nisu bile popularne u obrazovanju, ni zanimljive za dalju primenu van te sfere.

Druga generacija proširene realnosti u obrazovanju (DGPRO), koja se proteže od 2010. do 2019. godine, poznata je kao proširena realnost zasnovana na aplikacijama, sa naglaskom da se iskustvo proširene realnosti ostvaruje na osnovu softvera, pre nego hadrvera. Broj PR aplikacija za obrazovanje značajno se povećao od početka ove generacije. Ova „epidemija“ nastala je kao rezultat prevazilaženja izazova koji su postojali kod prve generacije (PGPRO). Najpre, dostupnost PR softvera na mobilnim uređajima povećala je pristup tehnologiji i smanjila potrebu za kupovinom skupog hardvera. Pored toga, razvoj game engine-a, GARS-ova i framework-a za PR aplikacije, učinilo je kreiranje ovakvih aplikacija isplativijim i jednostavnijim. Predstavljanje Google Smart naočara 2014. godine i Pokémon Go 2016. godine su dodatno potpomogli komercijalizaciji ove generacije [8].

Treća generacija proširene realnosti u obrazovanju, koja počinje od 2020. godine zasnovana je na

specijalizovanim uređajima proširene realnosti kao što su pametne naočare i PR zasnovana na web-u. Ovakve web aplikacije, predstavljaju „lek“ za otpor pojedinaca da koriste aplikacije za pametne telefone kako bi iskusili proširenu stvarnost. Korisnici pametnih telefona bi trebalo da preuzmu posebnu aplikaciju da bi uživali u PR iskustvu. Ipak, neki korisnici nakon nekoliko upotreba ovakve aplikacije brišu i to je uglavnom uslovljeno potrebom za memorijskim prostorom na telefonu, dok se drugi i ne trude da ovakve aplikacije skidaju na svoje pametne uređaje. Shodno tome, od 2017. godine prioritet je da se korisnicima omogući PR iskustvo putem vebPR-a, ukidajući potrebu za instalacijom posebnih aplikacija. Iako trenutno manje efikasan od PR-a zasnovanog na aplikacijama, webPR ima veliki potencijal da PR iskustvo dovede na viši nivo u budućnosti [8].

C. Zajedničko delovanje proširene realnosti (PR) i veštačke inteligencije (VI)

Simultaneous Localization and Mapping (*SLAM*), tip konvencionalne metode računarskog vida, je izvorno korišćen u PR softveru. Za mapiranje i praćenje okoline, sistemi bazirani na *SLAM* algoritmu upoređuju vizuelne informacije koje se dobijaju između različitih frame-ova kamere. Međutim, da bi se ponudile funkcionalnosti veće verodostojnosti, savremene PR aplikacije se oslanjaju na duboko učenje. Neke od aplikacija u kojima zajednički deluju PR i VI [11]:

1. Uočavanje i identifikacija objekata (npr. *Ikea place* aplikacija)
2. Prepoznavanje i prevođenje teksta (npr. *Google Translate*)
3. Automatsko prepoznavanje govora (npr. *Panda Sticker* aplikacija)

IV. KONCEPTI TEHNOLOGIJA PROŠIRENE REALNOSTI (PR) I VIRTUELNE REALNOSTI (VR)

Iako su PR i VR tehnologije usko povezane, one zapravo nude različite pristupe u interakciji sa stvarnošću i virtuelnošću [11]. Spajanjem digitalnih informacija sa onim iz stvarnog sveta, tehnologija proširene realnosti (PR) omogućava korisnicima da ostvare kontakt sa virtuelnim objektima i istovremeno posmatraju realni svet (često preko digitalne kamere na mobilnom telefonu ili tabletu). Za razliku od toga, VR tehnologija izmešta korisnika iz realnog sveta u virtuelni sa ciljem da se potpuno izloži takvom okruženju. Drugim rečima, dok PR omogućava korisnicima da komuniciraju i sa virtuelnim stvarima i realnim objektima u realnom svetu, VR potpuno psihološki uranja u virtuelno okruženje. Kao rezultat toga, postoje značajne razlike u mogućnostima PR-a i VR-a.

Cilj PR-a prema Azumi [12] je da stvori besprekornu koegzistenciju stvarnih i virtuelnih objekata unutar istog prostora, ostavljajući korisniku utisak da virtuelni objekti dopunjuju stvarnost, a ne da u potpunosti zamenjuju pogled na stvarni svet. Takođe, konstatuje se za PR da:

1. pametni telefoni mogu da pruže iskustvo proširene realnosti
2. neprimetno kombinuje virtuelne i fizičke aspekte
3. nudi interakciju u realnom vremenu
4. precizno postavlja 3D objekte

Tri suštinske komponente VR-a su sledeće:

1. VR od opreme zahteva poseban headset i često u kombinaciji sa još nekim ulaznim uređajima
2. korisnik istražuje okruženje u kome se nalazi vizuelno, i to uz pomoć head-mounted displeja, pomerajući svoje telo u prostoru
3. kako bi bilo moguće tzv. uranjanje korisnika u virtuelnu realnost, potrebno je obezbediti ubedljivi virtuelni svet koji bi korisnik bez sumnje prihvatio kao novu realnost
4. interaktivnost se meri nivoom angažovanja korisnika sa virtuelnim okruženjem i često se postiže korišćenjem senzora i ulaznih uređaja kao što su džojstici ili tastature

Na slici ispod je prikazana ilustracija prethodno pomenutih suštinskih razlika između PR i VR (Slika 3.)



Slika 3. Ilustracija razlika između VR i PR [13].

V. POGLED NA PR IZ PERSPEKTIVE NEURONAUKA

Kako bi se bolje razumela biološka strana čoveka koja objašnjava zašto čovek ima benefite od ovakvog načina učenja pre nego od tradicionalnog, bilo je potrebno pretražiti dostupnu literaturu neuronauka. Naime, radnje koje se aktiviraju u čoveku prilikom razumevanja sveta oko sebe podrazumevaju: čulne, vizuelne, perceptivne, motoričke, empatične i emotivne aspekte. Posebna pažnja je posvećena proučavanju vizuelnog stila učenja radi procene njegovog obrazovnog uticaja u smislu potkrepljivanja sticanja znanja.

Istraživanje vizuelnog stila učenja je dobilo poseban fokus kako bi se procenio njegov obrazovni uticaj u smislu jačanja usvajanja informacija. Zaista, vid predstavlja dominantni senzorni sistem kod čoveka, koji zauzima

između 20 i 30% površine moždane kore kod ljudi, što ga čini glavnim čulnim organom. Ljudi razvijaju vizuelno-neverbalne komponente brže od verbalnih jer su svakodnevno izloženi širokom spektru vizuelnih reprezentacija, uključujući digitalne grafikone, mape, znakove, filmove, dijagrame, ilustracije itd. Prema studijama Ricolatija, Sinigalje, Kolinsa i Gazanige, čovek ima tendenciju da se seti oko 10% onoga što je čuo, 20% onoga što je pročitao i 80% onoga što je video.

Veza između vizuelnog i prostornog, identifikovana je u studijama neuronauka koje tvrde da vizuelno-narativne reprezentacije kojima nedostaju vizuelno-prostorne, šematske i sekvencijalne reference onemogućavaju stvaranje mentalnog modela pitanja koje treba rešiti, dok su dinamičke šematski-vizuelno-prostorne reprezentacije dovoljne za sposobnost rešavanja problema.

Nekoliko studija je, međutim, pokazalo da grafičko-vizuelni komunikacija može dovesti do dva tipa međuzavisnih rizika: dekorativizam i kognitivno preopterećenje. Dekorativizam odnosi se na uvođenje slika koje nisu direktno povezane sa suštinom sadržaja pisanog teksta ili lekcije, a koji može da izazove kognitivno preopterećenje dodajući komponentu odvlačenja pažnje. Odavde sklonost da verbalni opis prethodi i čak utiče na vizuelno čitanje, do te mere da je u različitim školskim udžbenicima prilično ograničen prostor rezervisan za slike [5].

PR može da obezbedi pristup i primenu svih modula učenja (vizuelni, slušni, modul čitanja/pisanja, kinestetički/taktilni) kako bi svojim korisnicima omogućilo najkvalitetniji način pristupa informacijama. U osnovi PR aplikacije jesu vizuelnog tipa, međutim to nije jedina njihova mogućnost. Ovakve aplikacije mogu da budu obogaćene propratnim tekstom, čulnim efektima za dočaravanje nekog okruženja, objekta, bića ili jednostavno za prenošenje informacija o određenom pojmu koji bi inače bio korisniku na uvid u formi teksta [14].



Slika 4. Primeri PR aplikacija iz oblasti hemije, medicine i matematike [15].

Jedan od osnovnih ciljeva PR aplikacija u kontekstu obrazovanja je da omogući simulaciju ili jednostavno interaktivnu grafičku reprezentaciju pojma o kome se uči. Na prethodnoj slici (Slika 4.) su predstavljeni primeri ovakvih aplikacija iz oblasti hemije i medicine i matematike.

VI. PREDNOSTI PR-A (U PROCESU UČENJA)

U obrazovanju se uočavaju mnogobrojne prednosti PR-a, posebno u oblastima koje zahtevaju zamišljanje i apstrakciju, daju ovakav empirijski dokaz. Na primer, [16] je detaljno objasnio kako su prednosti specijalizovane vizuelizacije koje je omogućila PR poboljšale kapacitet učenika da razumeju i rešavaju probleme iz stvarnog sveta. Jedna od prednosti korišćenja proširene stvarnosti u obrazovnim okruženjima to što poboljšava vizuelizaciju čak i u izazovnim akademskim kontekstima, kao što su hemija i biologija. U većini slučajeva, ključne komponente vizuelno-prostornih sposobnosti u situacijama e-učenja oslanjale su se na ilustracije ili verbalne opise profesora. No, u praksi se pokazalo da takva zavisnost od crteža nastavnika i usmenih predavanja može negativno uticati na ishode učenja i performanse učenika [16,17].

Dostupni materijali za učenje, bilo gde, bilo kada. Fizički modeli, posteri, pisana uputstva i papirni udžbenici mogli bi biti zamenjeni proširenom stvarnošću. Pruža prenosive, jeftinije obrazovne resurse. Kao rezultat toga, obrazovanje postaje dostupnije i prenosivo [17].

Specijalni uređaji nisu potrebni. Za razliku od virtuelne stvarnosti, proširenoj stvarnosti nije potrebna nikakva skupa oprema. Većina ciljne populacije može odmah da koristi PR tehnologiju jer 73% adolescenata trenutno poseduje pametni telefon. Za učenike starije životne dobi ovaj procenat se procenjuje da je još veći.

Veća kognitivna angažovanost i zainteresovanost učenika. PR učenje koje je interaktivno i organizovano u neku ruku u vidu igrica može biti od velike koristi učenicima [17]. To čini učenje prijatnim i lakim i drži učenike zainteresovanim tokom čitavog časa. Uz pomoć grupe od 30 studenata medicine, [18] je pokazao kako PR može povećati interesovanje učenika za učenje anatomije i zadržavanje informacija, istovremeno podstičući interaktivno okruženje za učenje.

Povećana želja za međusobnu kooperaciju. Aplikacije koje koriste proširenu stvarnost pružaju obilje šansi da se časovi učine interaktivnim, raznovrsnim i daju mnogo prilika da studenti aktivno učestvuju kroz diskusije ili međusobnu kooperaciju. Veštine timskog rada unapređuju se kroz interaktivne časove koji istovremeno uključuju sve učenike u proces učenja [17].

Brži i efikasniji proces učenja. Kroz vizuelizaciju i potpunu predanost predmetu, proširena stvarnost u obrazovanju pomaže učenicima da postignu superiorne rezultate. Bez sumnje, *slika govori hiljadu reči*, baš kao što je i rekao kineski filozof i socijalni reformator Konfučije. Učenici tako mogu posmatrati nešto svojim očima, u stvarnosti, umesto da čitaju o tome u teoriji [17].

Praktično obrazovanje pruža podršku za kinestetičko učenje. Osim obrazovanja, upotreba PR-a u profesionalnoj obuci može biti prilično korisna. Na primer, tačna

reprodukcija uslova na terenu u realnosti može pomoći da se ovladaju praktičnim veštinama potrebnim za obavljanje određenog posla. Kinestetičko učenje je opisano u radu [19] kao vrsta učenja kada učenici učestvuju u fizičkim aktivnostima umesto da slušaju predavanja ili gledaju kako predavači demonstriraju. Interakcije studenata medicine sa kompjuterski generisanom 3D lobanjom i njihovo učenje ispitali su u svojoj studiji od strane [18] da bi videli kako je uključivanje PR-a uticalo na ove aktivnosti. Rezultati istraživanja impliciraju da je jedna od najosnovnijih obrazovnih prednosti proširene stvarnosti njena mogućnost da olakša kinestetičko učenje.

Univerzalno primenljiv na sve stepene obrazovanja. PR nije ograničen na jedan slučaj upotrebe ili oblast primene, bilo da se radi o obrazovnim igrama za malu decu ili obuci na poslu [17].

VII. Izazovi sa kojima se PR suočava

Prvi problem je problem pristupačnosti, koji pogađa one koji imaju neku vrstu invaliditeta. Dizajn aplikacije koji je čini upotrebljivom za sve, bez obzira na specifične potrebe, naziva se „pristupačnim“. Učenici sa oštećenjima bilo koje vrste, kao što su slepilo ili oštećenje vida, gluvoća ili oštećenje sluha, intelektualni invaliditet, ili fizički invaliditet, nazivaju se osobama sa posebnim potrebama u obrazovanju [8]. Većini aplikacija proširene stvarnosti za obrazovanje još uvek nedostaju karakteristike pristupačnosti, uprkos nekim pokušajima i dokazanim prednostima PR za obrazovanje sa posebnim potrebama [36].

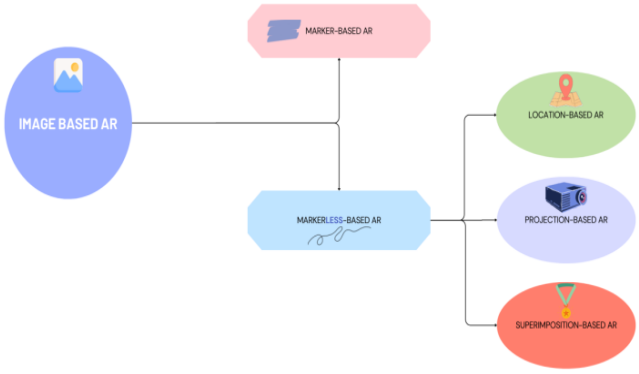
Druga briga je upotrebljivost, koja se odnosi na to koliko je jednostavno koristiti PR aplikacije. U stvari, nekoliko istraživanja je pokazalo da je teškoća korišćenja PR sistema najčešće navedena prepreka PR-a u obrazovanju [8,17,20]. Čak i ako je upotrebljivost DGPRO aplikacija znatno poboljšana u odnosu na PGPRO aplikacije, određene okolnosti utiču na to koliko dobro se korisnici snalaze sa PR aplikacijama. Prema [5], PR uključuje nekoliko čula i zahteva od učenika da rade brojne aktivnosti odjednom, što može opteretiti njihovu pažnju i učiniti PR sisteme manje korisnim. Drugim rečima dolazi do kognitivnog plavljenja velikom količinom sadržaja.

Treći problem ima veze sa difuzijom i uzrokovan je stvarima kao što su primena na više platformi i zahtev za preuzimanje i instalaciju. Većini PR aplikacija nedostaje podrška za više platformi i kreirane su da se koriste na jednoj, što utiče na to koliko se široko koriste. Kao rezultat, kako bi aplikacije bile multiplatformske, programeri PR aplikacija moraju da prođu kroz brojne razvojne cikluse, što povećava vreme proizvodnje i troškove.

Nastavno osoblje često nema potrebnu obuku u rukovanju i pedagoškoj primeni aplikacija proširene realnosti.

VIII. Tipovi aplikacija proširene realnosti

Podela PR aplikacija je predstavljena grafički na sledećoj ilustraciji (Slika 5.).

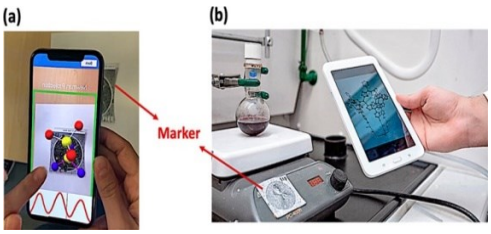


Slika 5. Grafička reprezentacija podele PR aplikacija.

A. *Marker-based PR*

Mobilne aplikacije koje koriste PR zasnovan na markerima mogu da preklope objekte koji se prepoznaju preko kamere iz naše okoline, virtuelnim objektima. Ovo se čini tako što se vizuelni podaci koje kamere prikupljaju, analiziraju i donosi se odluka gde se postavlja marker. Virtuelna stavka nestaje kada kamera „izgubi iz vida“, tj. više ne može da detektuje marker (na primer kada korisnik pomera telefon), a ponovo se pojavljuje kada je marker ponovo vidljiv kameri [21]. Za slučaj marker-based PR aplikacija, predefinisana 2D slika, QR kod ili 3D objekat mogu da posluže kao marker, kako bi se odredio položaj virtuelnog objekta [22]. Algoritmi otkrivanja i praćenja karakteristika, kao što su *SIFT* (*Scale-Invariant Feature Transform*) ili *SURF* (*Speeded-Up Robust Features*), često se koriste u PR baziranoj na markerima da bi se pomoglo u identifikaciji i praćenju određenih markera ili obrazaca u okruženju.

U radu [22] PR aplikacija zasnovana na markerima (*ARchemy*) je korišćena u oblasti hemije, pri čemu je korisnik morao da usmeri svoju kameru ka markeru da bi video posebnu animaciju mehanizma hemijske reakcije ili novonastavu molekularnu strukturu. Ovo je studentima bilo posebno od pomoći kako bi u potpunosti shvatili sve procese koji se dešavaju u jednoj reakciji, kao i da vide konačnu strukturu svoj sintetizovanog proizvoda.



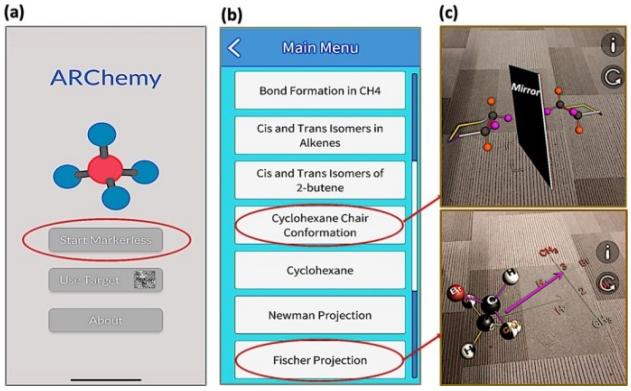
Slika 6. Primer PR aplikacije (*ARchemy*) zasnovane na markerima iz oblasti hemije [22].

B. *Markerless-based PR*

PR bez fizičkih markera, kao što je QR kod ili 2D slika, poznat je kao PR bez markera. Umesto toga, prati okruženje korisnika i locira virtuelni sadržaj koristeći podatke zasnovane na lokaciji sa uređaja kao što su GPS i akcelerometri. Ovo eliminiše potrebu da korisnik odštampa prepoznatljiv oblik, sliku ili bar kod da bi video PR [22]. Program tada može prekriti virtuelni sadržaj u skladu sa prostornim vezama i orijentacijom objekata i površina u vidnom polju korisnika [21].

Zapravo, kod *markerless* PR-a se koriste posebni, složeniji algoritmi računarskog vida i tehnike kojima se prate i detektuju objetki realnog sveta. Jedan od takvih algoritama je i *SLAM* (*Simultaneous Localization and Mapping*) algoritam koji je najčešće u upotrebi. *SLAM* je tehnika za dobijanje 3D strukture nepoznatog okruženja i kretanja senzora u okruženju. Pored primene u robotici, za onlajn 3D modeliranje, koristi se i u vizualizaciji kod proširene realnosti. U prvim *SLAM* algoritmima, integrisano je mnogo različitih tipova senzora kao što su laserski senzori dometa, rotacioni enkoderi, inercijalni senzori, GPS i kamere. Poslednjih godina se aktivno raspravlja o *SLAM*-u koji koristi samo kamere jer je konfiguracija senzora jednostavna, a tehničke poteškoće veće od ostalih, pa se drugačije zove *vSLAM*.

U kontekstu obrazovanja, PR bez markera može se lakše koristiti u okruženjima za učenje kako bi se učenici izložili stvarnim scenarijima i naučili ih osnovnim laboratorijskim tehnikama. Na primer, uspešno je korišćen kao alat za titraciju gde studenti mogu da koriste ovu tehnologiju da vežbaju svoje veštine titracije u okviru PR aplikacije bez potrebe za hemikalijama. Kada su takvi resursi dostupni učenicima, oni ne samo da olakšavaju njihovo iskustvo učenja, već i smanjuju rizike i bezbednosne izazove koji mogu nastati kada učenici rukuju opasnim hemikalijama. Primer takve jedne aplikacije je prethodno pomenuta *ARchemy* (Slika 7.).

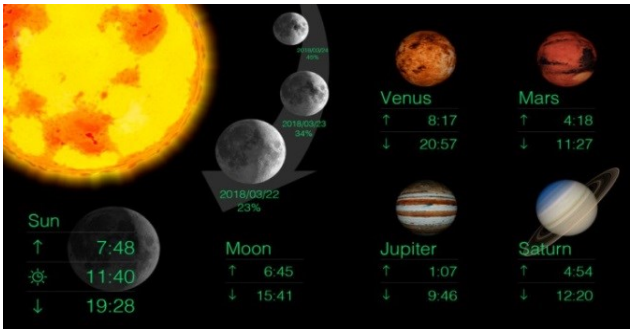


Slika 7. *Markerless* PR aplikacija *ARchemy*: (a) PR početna strana, (b) deo menija, (c) vizuelizacija 3D objekta postavljanjem kamere bilo gde u prostoru [22].

C. *Location-based PR*

Proširena stvarnost zasnovana na lokaciji (PR) je vrsta proširene stvarnosti bez markera koja koristi geografske podatke za predstavljanje digitalnih vizuelnih prikaza na određenim mestima [21]. Ovo omogućava stavljanje virtuelnih objekata u stvarni svet kroz virtuelni pejzaž koji postoji na ekranu mobilnog uređaja.

Primer jedne ovakve aplikacije koja može da se koristi u edukativnom kontekstu je *Star Walk*. To je *real-time* aplikacija za posmatranje zvezda u realnom vremenu, pri čemu sadrži interaktivnu mapu noćnog neba. Ovakva aplikacija namenjena je svim uzrastima, jednostavna je za upotrebu i poučna. *Star Walk* aplikacija zahteva da korisnik pomera svoj mobilni uređaj kako bi pružila tačnu reprezentaciju rasporeda nebeskih tela na noćnom nebu. Na ovaj način se mogu saznati zanimljive činjenice o nebeskim telima, dodirom na njihova imena. Takođe, još jedna od mogućnosti aplikacije je da se „iskoristi vremeplov“ kako bi se videla prošla i buduća pozicija nebeskog objekta. Moguće je i uključiti obaveštenja kako bi korisnik bio u toku sa spektakularnim astronomskim događajima, uključujući planarne opozicije, konjukcije, pomračenje Sunca i Meseca, itd [23].



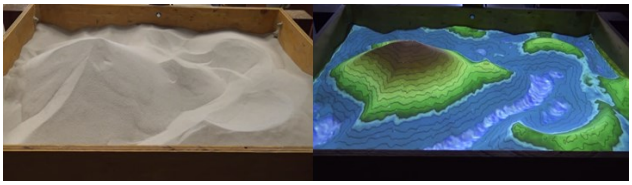
Slika 8. Star Walk aplikacija [23].

D. *Projection-based PR*

Projekciona PR, često poznata kao prostorna PR, je sredstvo za prikazivanje digitalnih informacija unutar fiksnog konteksta. Fokusira se na prikazivanje virtuelnih stvari unutar ili na stvarnom prostoru korisnika. To je jedan od najosnovnijih tipova PR u kojem se svetlost projektuje na površinu. Interakcija se odvija fizičkim dodirivanjem projektovane površine. Korisnik nije vezan ni za jedan uređaj u projekciji PR-a pošto su virtuelne stavke ugrađene direktno u okruženje.

PR pristupi zasnovani na projekciji mogu se koristiti za stvaranje iluzija u pogledu dubine, lokacije i orijentacije objekta. Primer jedne takve primene se može videti na slikama koje slede u nastavku rada (Slika 9.). Demonstracija je nazvana *Augmented Reality Sandbox*, a zasniva se na tome da je ispod projektora postavljen beli pesak koji je visoko reflektivan, tako da kada se svetlost spusti do njega, ona se odbija. Pomeranjem postavnjenog

peska nastaju nove topografske linije tako da prikaz deluje kao da je projekcija neke slike, ali dodatno je moguće biti interaktivan sa ovakvom postavkom. *Augmented Reality Sandbox* se često koriste u obrazovnim okruženjima ili muzejima za podučavanje koncepata koji se odnose na topografiju, hidrologiju, eroziju ili modeliranje pejzaža. Takođe se mogu koristiti za interaktivnu igru ili u umetničke svrhe [21,24].



Slika 9. *Augmented Reality Sandbox* primer projection-based PR-a u edukaciji [24].

E. Superimposition-based PR

Proširena stvarnost zasnovana na superponovanju je tehnologija koja zamenjuje neke ili sve atribute originalne stvari sa novododatim atributima koji su superponirani na isti objekat proučavanja [25].

Kada je učenje upareno sa fizičkim posetama u istorijskim mestima, tehnologije proširene stvarnosti zasnovane na superponovanju jednog dana mogu imati izuzetne rezultate na obrazovanje. Uz korišćenje savremene tehnologije, učenici mogu da posete istorijska mesta, ratišta i muzeje, da upoznaju karakteristike starih građevina, istražuju ratne događaje i ishode i da proučavaju neobične muzejske postave [26].

Da bi aplikacija zamenila originalni prikaz stavke poboljšanim, prepoznavanje objekta je od suštinskog značaja. Ograničenje ove tehnike je zahtev za potpunim prepoznavanjem objekta. Identifikacija objekta može biti samo delimično uspešna zbog tehnološkog kvara u zavisnosti od veličine, boje i okruženja položaja objekta. Paradoks ograničenja neograničenog kretanja je još jedan nedostatak ove vrste PR tehnologije. Korisnici mogu da dobiju nedoslednu proširenu sliku ili možda ne dobiju proširene povratne informacije ako kamera uređaja nije u stanju da identifikuje sve elemente stavke kada je skenira iz različitih uglova [26]. Na slici (Slika 10.) može se videti primer jedne *superimposition-based* PR aplikacije.



Slika 10. Primer superponirane PR aplikacije prilikom obilaska spomenika tokom časa istorije [26].

VIII. UTICAJ PR-A NA DUGOTRAJNOST PAMĆENJA NAUČENOG

U radu [27] vršila se pilot studija nad studentima četvrtog razreda (ukupno 39 njih), jedne obrazovne institucije u Španiji. Ovo istraživanje je raspolagalo sa dve grupe studenata, jedna kontrolna grupa studenata i jedna eksperimentalna, odnosno grupa studenata koja je primenjivala tradicionalan način praćenja nastave i druga koja je primenjivala PR softver. Kontrolna grupa studenata je učila po principu da profesor predaje, učenici slušaju, hvataju beleške, gledaju slike ili snimke vezane za temu koja se obrađuje na času. Eksperimentalna grupa je bila izložena usvajanju znanja uz pomoć pedagoga/profesora i PR softvera. Cilj istraživanja je bio da se evaluira uticaj ovog PR pomoćnog alata na sticanje i zadržavanje znanja.Početna hipoteza je bila da PR alati obezbeđuju efikasnije učenje. Ako je to tako, to znači da bi i ocene trebalo da budu više u odnosu na one koje se ostvaruju tradicionalnim načinom učenja.

Da bi se demonstrirala ova hipoteza učenici i profesori su izabrali dva predmeta slične težine, u ovom slučaju: „Probavni sistem“ i „Promene u prošlom veku“. Predmet „Probavni sistem“ je u toku semestra bio predavan uz pomoć PR alata, dok je predmet „Promene u prošlom veku“ bio predavan tradicionalnom metodom. Razlog zašto su se izabrala dva predmeta slične složenosti se objašnjava time što je ova obrazovna institucija privatna i roditelji studenata odlučuju u velikoj meri o načinu sprovođenja nastave. U želji da nijedan student iz neke kontrolne grupe ne bude „oštećen“ za neko obrazovno iskustvo, ovo je bilo potrebno da se učini kako bi se istraživanje dalje sprovodilo. Rezultati studije: Primenjena su tri testiranja studenata za svaki čas. Prvo čim se nastava završi, drugo nakon dve nedelje (bez prethodne najave) i treće nakon četiri nedelje (takođe bez prethodne najave). Test se sastojao od 23 pitanja a ocene su išle u rasponu od 0 do 10. Ocene učenika su bile analizirane da bi se procenilo poboljšanje u učenju. Statistička procena ocena sa testova izvršenih odmah nakon predavanja utvrđuje da nema nikakve bitne razlike između ova dva načina učenja. Ocene sa testova održanih nakon dve nedelje od predavanja donose razlike. Naime, sada postoji statistički dokaz da su ocene dobijene na testovima iz oblasti koja je bila predavana pomoću PR alata značajno bolje. Konačno, treća procena je izvršena četiri nedelje nakon inicijalne za obe nastavne jedinice. Još jednom je dokazano da su ocene dobijene na nastavnoj jedinici koja je predavana uz pomoć PR -a bolje od onih koje su dobijene na jedinici kod koje se primenio tradicionalni pristup. Studija je takođe merila da li je motivisanost učenika veća kod nastavne jedinice koja koristi PR alat i to je potvrđeno upitnikom koji su studenti popunjavali.

IX. Zaključak

Ovaj rad sugeriše koliko je napredovanje tehnologije poslednjih godina donelo promena u sferi obrazovanja. Na kraju, bitno je sumirati ključne stavke.

Koje su dobre strane i izazovi sa kojima se PR susreće danas u obrazovnom kontekstu? U kontekstu proširene realnosti kao prednosti se ističu: poboljšanje kapaciteta učenika da razumeju probleme iz stvarnog sveta, dostupnost materijala za učenje bilo gde i bilo kada, posebni uređaji za praćenje ovakve nastave nisu neophodni, veća kognitivna angažovanost i zainteresovanost učenika, povećana želja za međusobnu kooperaciju, brži i efikasniji proces učenja, pruža podršku za kinestetičko učenje, primenjiv na sve stepene obrazovanja. Kao izazovi se prepoznaju: problem pristupačnosti za učenike sa posebnim potrebama (nekim aplikacijama), upotrebljivost koja se odnosi na to koliko je komplikovano rukovati aplikacijama proširene realnosti, mogućnost kognitivnog plavljenja, zahtev za instalacijom aplikacija i zauzimanje neke količine memorijskog prostora, neadekvatna ili nikakva obuka nastavnog osoblja za rad sa ovim aplikacijama u svojoj nastavi, otpor nastavnika prema tehnološkim inovacijama.

Kakva je dugotrajnost znanja stečenog tehnikom PR-a? Istraživanje koje je bilo razmatrano je poredilo učinkovitost u pogledu dugotrajnosti učenje na tradicionalan način i primenom proširene realnosti. Rezultati su pokazali da je zaista, znanje stečeno na časovima na kojima se primenjivala proširena realnost čvršće ostalo učenicima u sećanju i da je dugotrajnije.

Zahvalnica

Autor rada se posebno zahvaljuje prof. dr Emiliji Živanović na izdvojenom vremenu, sugestijama i pomoći pri pisanju rada.

Literatura

[1] Petrović K., "Pregled konvergencije proširene stvarnosti i ChatGPT-a sa sveprisutnim učenjem radi unapređenja obrazovanja", 2023, diplomski rad.

[2] Wyss C., Bührer W., Furrer F., Degonda A., and Hiss J. A, "Innovative Teacher Education with the Augmented Reality Device Microsoft HoloLens—Results of an Exploratory Study and Pedagogical Considerations", *Multimodal Technol. Interact*., vol. 5, no. 8, p. 45, 2021.

[3] Lampropoulos G., Keramopoulos E., Diamantaras K., and Evangelidis, G., "Augmented Reality and Virtual Reality in Education: Public Perspectives, Sentiments, Attitudes, and Discourses", Educ. Sci., vol. 12, no. 11, p. 798, 2022.

[4] Nagy A, "The Impact of E-Learning. In: Bruck P.A., Karssen Z., Buchholz A., Zerfass A. (eds) E-Content," pp 79-96, Berlin, Heidelberg, Springer, 2005.

[5] Luigini A. and Panciroli C., "Graphics", *Interdisciplinary journal on image, imagery and imagination*, no. 2, 2020.

[6] Mishra S., Yadav M., & Choudhary K., "Ubiquitous learning: Future of e-learning", *Proc. First International Conference on Emerging Trends in Engineering and Applied, Conference*, 2013, pp. 1-4.

[7] Suartama I. K., Setyosari P., Sulthoni S., and Ulfa S., "Development of Ubiquitous Learning Environment Based on Moodle Learning Management System", *International Journal Od Interactive Mobine Technologies*, pp 182–204, 2020.

[8] Garzón J., "An Overview of Twenty-Five Years of Augmented Reality in Education", *Multimodal Technol. Interact*., vol. 5, no.7, p 37, 2021.

[9] Lamberti F., Manuri F., Sanna A., Paravati G., Pezzolla P., and Montuschi P., "Challenges, Opportunities, and Future Trends of Emerging Techniques for Augmented Reality-Based Maintenance", *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 4, pp 411–421, 2014.

[10] Bower M., Howe C., McCredie N., Robinson, A., and Grover D., "Augmented Reality in education – cases, places and potentials", *Educational Media International*, vol. 51, no. 1, pp 1–15, 2014.

[11] Cevikbas M., Bulut N., and Kaiser G., "Exploring the Benefits and Drawbacks of AR and VR Technologies for Learners of Mathematics" *Recent Developments. Systems*, vol. 11, no. 5, p. 244, 2023.

[12] Azuma R. T., "A Survey of Augmented Reality", *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[13] Understanding Virtual Reality and Augmented Reality, 2018, https://www.youtube.com/watch?v=vz0UUVDt2ps.

[14] Prithishkumar J., "Understanding your student: using the VARK model", *Journal of Postgraduate Medicine*, vol. 60, no. 2, pp. 183-186, 2018.

[15] Chemical reaction using augmented reality | Envision AR, 2019, https://www.youtube.com/watch?v=Kp5ubU2Y07Y.

[16] Gavish N., Gutiérrez T., Webel S., Rodríguez, J., Peveri M., Bockholt U., and Tecchia F., "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks", *Interactive Learning Environments*, vol. 23, no. 6, 2013, pp. 778–798.

[17] Shalimov A., "Augmented Reality in Education", *The Hottest EdTech Trend. Eastern Peak - Technology Consulting and Development Company*, 2023.

[18] Chien C., Chen C. H., and Jeng T., Semantic Scholar. "An Interactive Augmented Reality System for Learning Anatomy Structure" *Semantic Scholar*, 2021.

[19] Bos A. S., Herpich F., Kuhn I., Guarese R. L. M., Tarouco L. M. R., Zaro M. A., Pizzato M., and Wives L., "Educational Technology and Its Contributions in Students' Focus and Attention Regarding Augmented Reality Environments and the Use of Sensors", *Journal of Educational Computing Research*, vol. 57, no. 7, 2019.

[20] Learner-Centered Teaching, Teaching and Learning STEM. https://www.engr.ncsu.edu/stem-resources/legacy-site/learner-centered.

[21] Abdinejad M., Ferrag C., Qorbani H. S., and Dalili S. "Developing a Simple and Cost-Effective Markerless Augmented Reality Tool for Chemistry Educatio", *Journal of Chemical Education*, vol. 98, no. 5, 2021, pp. 1783–1788.

[22] Taketomi T., Uchiyama H., and Ikeda S., "Visual SLAM algorithms: a survey from 2010 to 2016", *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, 2017.

[23] How to Build an Augmented Reality Sandbox. 2017, Bealsscience. https://www.bealsscience.com/post/2017/06/07/augmented reality-sandbox-will blow-your-mind.

[24] Types of AR and ndash; Digital Promise. (n.d.). Digital Promise. https://digitalpromise.org/initiative/360-story-lab/360-production-guide/investigate/augmented-reality/getting-started-with-ar/types-of-ar/.

[25] Poghosyan S., "Learning-Oriented Augmented Reality Technology", *European Science Review*, vol. 18, 2019, pp. 42–46.

[26] Perez-Lopez D., and Contero M. (n.d.). "ERIC - EJ1018026 - Delivering Educational Multimedia Contents through an Augmented Reality Application: A Case Study on Its Impact on Knowledge Acquisition and Retention", *Turkish Online Journal of Educational Technology – TOJET,* vol. 12, no. 4, 2013, pp. 19-28.

[27] Wittman J. W., The Forgetting Curve. Course Hero, 2020, https://www.coursehero.com/file/53418964/forgetting-curvepdf/.

Arduino Visor – robotska glava

Stefan Pisarov

Sadržaj - U ovom radu prikazujemo sklop zličitih komponenti u jednu repliku ljudske glave sa funkcijama sluha, vida, govora kao i mogućnostima za izvršavanje jednostavnih umnih zadataka. Korišćenjem „Arduino“ hardversko-softverske razvojne platforme kao i „Raspberry Pi 4“ mikro računara omogućena je potpuna kontrola rada ovog projekta.

I. Uvod

U današnjem ubrzanom tehnološkom svetu, interakcija između ljudi i mašina postaje sve više kompleksnija, ali u isto vreme i sve zanimljivija. Naš projekat „Arduino Visor“ predstavlja specijalnu kreaciju koja simbolizuje ljudsku glavu, pored slobode kretanja, poseduje i mogućnost govora, sluha kao i vida. Kombinujući različite tehnologije u oblasti robotike stvorili smo inteligentnu robotsku glavu koja je sposobna da nadmaši dosadašnju interakciju između ljudi i mašina. Opremljen mikrofonom koji simulira ljudsko čulo sluha, kamerom koja simulira ljudsko oko kao i zvučnicima za simuliranje govora, naš robot ima sposobnost da stvori percepciju na svet koji je bio rezervisan samo za ljudska bića. Njegove sposobnosti govora omogućavaju mu da komunicira sa ljudima na prirodan i intuitivan način, dok njegov vid omogućava prepoznavanje raznih objekata i lica u realnom vremenu. U narednim delovima rada, prikazaćemo korake kako smo razvili ovu jedinstvenu robotsku glavu, uključujući i tehničke izazove sa kojima smo se suočili tokom izrade.

II. Princip rada robotske glave

A. Kontrola pokreta

Robotska glava je opremljena sa pet servo motora koji omogućavaju precizno i prirodno kretanje glave.

Dva servo motora postavljena su unutar glave i omogućavaju glavi da se pomera po x i y osama odnosno levo-desno i gore-dole (Slika 1.), pružajući celokupnoj glavi potpunu slobodu pokreta. Dodatna dva servo motora smeštena su unutar lobanje i služe za kontrolisanje pokreta očnih jabučica u različitim pravcima (Slika 3.), stvarajući iluziju fokusa i pažnje. Još jedan servo motor smešten je u donjoj vilici i omogućava realistično kretanje donje vilice (Slika 2.), omogućavajući realistično imitiranje govora [1].



Slika 1. MG996R Servo motor.



Slika 2. HS-805BB Servo motor.



Slika 3. MG90S Servo motor.

B. Kontrola pokreta

Dva zvučnika su direktno smeštena unutar ušiju robota (Slika 4.). Ovi zvučnici su ključni za simulaciju ljudskog govora. Kombinujući precizno pozicioniranje vilice sa sofisticiranim algoritmima obrade zvuka, naš robot može reprodukovati različite tonalitete i nijanse, stvarajući realističan i raznolik govor [5].

S. Pisarov je učenik IV razreda na smeru Elektrotehničar računara, Srednja elektrotehnička škola “Mihajlo Pupin”, Futoška 17, 21000 Novi Sad, Srbija, E-mail: pisarov.stefann@gmail.com



Slika 4. Zvučnik.

C. Vizuelna percepcija

Kamera, smeštena unutar desnog oka robota [2], predstavlja čulo vida našeg robota. Ova kamera koristi kvalitetan senzor kako bi beležila slike visoke rezolucije. Pomoću algoritma za prepoznavanje objekata, lica i pokreta, robot može precizno identifikovati i pratiti objekte u svom vidnom polju [4].



Slika 5. Microsoft Lifecam HD-3000.

D. Napajanje

Tokom izrade projekta primećeno je da motori previse troše obične alkalne baterije zbog čega je dodatno jedno posebno napajanje koje napaja samo motore kako bi ih održavalo u optimalnom radnom nivou.



Slika 6. Napajanje D V/12 Ah.

E. Programiranje i komunikacije

Svi aspekti vezani za vid, sluh i govor su programirani u programskom jeziku Python na Raspberry Pi 4 računaru. Raspberry Pi 4 funkcioniše kao mozak sistema, interpretirajući podatke sa kamere i mikrofona. Ključna uloga Raspberry Pi-a je analiza i obrada podataka. Nakon analize, informacije se šalju na Arduino Mega platformu putem serijske komunikacije [3].



Slika 7. Raspberry Pi 4.

F. Interakcija sa Arduino Mega mikrokontrolerom

Arduino Mega, kao glavna hardversko-softverska platforma, prima podatke od Raspberry Pi-a i kontroliše pokrete servo motora i druge elektronske komponente unutar glave. Ovaj deo sistema odgovoran je za sinhronizaciju svih pokreta u stvarnom vremenu, obezbeđujući da glava deluje fluidno i prirodno.



Slika 8. Arduino Mega.

Ovaj robot može da pronađe primenu u raznim granama edukacije počevši od robotike i mehatronike. Može da posluži kao eksponat na sajmovima i drugim skupovima vezanim za robotiku i vestacku inteligenciju, eksponiranjem robota postoje velike mogućnosti da će učenici, studenti i mnogi drugi zaljubljenici u tehnologiju probuditi interesovanje prema oblastima kao što su robotika, veštačka inteligencija i programiranje. Zbog mogućnosti komunikacije verujemo da bi ovaj robot posebno privukao pažnju mlađim generacijama i zagolicao njihovu maštu i želju za upoznavanjem sa ovakvom vrstom tehnologije.

III. ZAKLJUČAK

Učestvujući u ovom projektu, razvili smo duboko razumevanje kompleksnosti veštačke inteligencije i robotske tehnologije. Kroz uspešnu implementaciju našeg robota, proširili smo granice onoga što je moguće u domenu interakcije između ljudi i robota. Učili smo iz

izazova sa kojima smo se susreli tokom izrade projekta i pronalazili mnoga inovativna rešenja, demonstrirajući našu sposobnost da prevaziđemo tehničke prepreke.

Ovaj projekat ne samo da ima trenutnu praktičnu vrednost u domenu robotike, već i šire značenje. On nam pokazuje da tehnologija ne mora biti odvojena od ljudskosti, umesto toga, može služiti kao produžena ruka ljudskih sposobnosti, poboljšavajući naše svakodnevne živote, ali i produbljujući naše razumevanje veštačke inteligencije. Ovaj rad treba da otvori vrata za buduće istraživače da nastave našu ili započnu neku svoju novu misiju, istražujući nove načine kako tehnologija može poboljšati naš svet.

ZAHVALNICA

Posebno se zahvaljujem Novici Gutoviću (mentor), rukovodiocu sekcije „Arduino“ u ETŠ „Mihajlo Pupin“ Novi Sad, na intenzivnoj tehničkoj podršci i veoma uspešnoj saradnji. Takođe, zahvaljujem se direktoru Ermedinu Elezoviću i Upravi ETŠ „Mihajlo Pupin“- Novi Sad na logističkoj podršci.

LITARUTERA

[1] Servo motori uputstvo, Dostupno na: https://lastminuteengineers.com/servo-motor-arduino-tutorial/.
[2] Uputstvo za rastavljanje Microsoft Lifecam HD-3000, Dostupno na: https://www.youtube.com/watch?v=lOU6rBUK0Do.
[3] Komunikacija između Arduina i Raspberry Pi-a, Dostupno na: https://medium.com/@araffin/simple-and-robust-computer-arduino-serial-communication-f91b95596788.
[4] Praćenje objekata korišćenjem Pythona i kamere, Dostupno na: https://automaticaddison.com/real-time-object-tracking-using-opencv-and-a-webcam/.
[5] https://github.com/Serbian-AI-Society/SrbAI.

Agrotemp

Relja Sokolov, Milica Krstić, Stefan Tošić, Jovan Kostić, Nemanja Mitov, Bojan Ćirić

Sadržaj – Ovaj rad predstavlja opis sistema za regulisanje temperature, vlažnosti vazduha i automatskog navodnjavanja plastenika malih dimenzija. Korišćenjem savremenih informacionih tehnologija.

I. Uvod

Agrotemp je proizvod koji služi za zadovoljavanje potreba malih poljoprivrednika koji poseduju manje plastenike. Ovaj sistem je dizajniran tako da pruža jednostavno i efikasno rešenje za upravljanje uslovima u plastenicima i pomaže u povećanju prinosa useva.

Agrotemp proizvodi su povezani sa *Internet of Things* (IoT) tehnologijom, što znači da korisnici mogu pristupiti proizvodu sa bilo kog mesta i u bilo koje vreme preko interneta. Ovo omogućava korisnicima da prate i kontrolišu uslove u svojim plastenicima iz udobnosti svojih domova, ili dok su na putu, putem web aplikacije koju pruža Agrotemp. Ova funkcionalnost pruža korisnicima fleksibilnost i olakšava upravljanje svojim usevima, što je od ključne važnosti za efikasno poljoprivredno poslovanje.

II. Princip rada i praktična realizacija

A. Princip rada sistema

Agrotemp proizvod funkcioniše tako što senzori ugrađeni u plastenik prate parametre kao što su temperatura i vlažnost vazduha, a prikupljeni podaci se prenose u bazu podataka. Korisnici mogu lako pristupiti aplikaciji preko interneta i videti stvarno vreme i prošle podatke o uslovima u svom plasteniku, kao i upravljati sistemom za regulisanje temperature i navodnjavanja.

B. Hardver

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

SG90 *MicroServo* [2] - Servo može da se rotira za približno 180 stepeni (90 u svakom smeru). Može se koristiti bilo koji servo kod, hardver ili biblioteka za kontrolu. Ekonomičan je i jednostavan za korišćenje, s vrlo niskom potrošnjom energije. Njega koristimo kako bi otvarali i zatvarali vrata plastenika.



Slika 1. SG90 *MicroServo*.

BMP280 [3] - digitalni senzor za precizno merenje barometarskog pritiska i temperature. BMP280 koristi naprednu MEMS tehnologiju.



Slika 2. BMP280.

ESP8266 [4] - sistem sa jednim čipom (SoC). Posebnost ovog 32-bitnog mikrokontrolera sa RISC arhitekturom je TCP/IP stek koji je implementiran veoma blizu hardvera. Svi ESP8266 moduli su stoga opremljeni WiFi konekcijom.



Slika 3. ESP8266.

Relej –moduo koji koristimo za uključivanje i isključivanje pumpe za navodnjavanje.

R. Sokolov je učenik Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: sokolov.relja.201@tsp.edu.rs
M. Krstić je učenica Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: krstic.milica.201@tsp.edu.rs
S. Tošić je učenik Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: tosic.stefan.201@tsp.edu.rs
J. Kostić je učenik Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: kostic.jovan.201@tsp.edu.rs
N. Mitov je učenik Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: mitov.nemanja.201@tsp.edu.rs
B. Ćirić je profesor Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija. E-mail: bojan.ciric@tsp.edu.rs



Slika 4. Relej.

C. Funkcionisanje SG90 MicroServo i senzora

SG90 *MicroServo* je preko kontrolera povezan na Internet odakle dobija informacije o rotaciji, odnosno otvaranju okvira za provetravanje. Ovi podaci se sa kašnjenjem od 60s prikazuju na web aplikaciji.

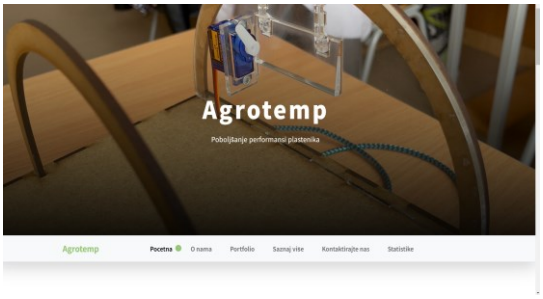
Podaci koji se upisuju u bazi su temperatura, pritisak, vlažnost vazduha, osvetljenje, kao i to da li su vrata otvorena ili zatvorena, da li je navodnjavanje uključeno i da li su sijalica i ventilator isključeni ili ne. Ako je neka od ovih komponenata uključena ili ako su vrata otvorena, za njih se u bazu upisuje vrednost 1, dok se u suprotnom upisuje 0.

D. Korisnički interfejs i vizualizacija podataka

Korisnički interfejs je deo sistema koji je rađen kao klasična web aplikacija. Iskoristili smo *HTML*, *CSS* i *JavaScript* kako bi stilizovali stranicu [1].

Korisnički interfejs se sastoji od više web strana. Na početnoj strani se nalazi video naše makete plastenika, kao i informacije o nama i kontakt forma. Na strani "Statistike" možete videti parametre plastenika i kontrolisati njegove komponente. Na slici 5 možete videti početnu stranu web aplikacije.

Podaci koji se prikazuju stižu sa *backend* strane. U tom delu se podaci čitaju iz baze podataka koju ažurira SG90 *MicroServo*. Svi podaci se prosleđuju po *Request Response* modelu klijentskoj strani, tj. korisničkom interfejsu.



Slika 5. Korisnički interfejs.

III. Moguće primene sistema

Ciljana grupa su nam mali poljoprivrednici, i bilo ko ko može da priušti naš sistem, koji nije skup, odnosno mali proizvođači ga mogu priuštiti uz subvencije.

Sistem je realizovan da omogućava automatsko navodnjavanje useva u plasteniku i pomaže u održavanju optimalne glasnost zemljišta za rast biljaka. Ovakav sistem za navodnjavanje smanjuje rizik od preteranog ili nedovoljnog zalivanja, što može negativno uticati na zdravlje biljaka i prinose.

IV. Moguće nadogradnje sistema

Proširenje proizvodnje i korišćenje obnovljivih izvora energije su važni ciljevi za Agrotemp kompaniju u budućnosti. Povećanje proizvodnje će nam omogućiti da pomognemo više zemljoradnicima koji imaju manje plastenike, kao i da se približimo tržištima na kojima ima većih poljoprivrednih gazdinstava.

Takođe, Agrotemp će nastaviti da koristi solarne panele za proizvodnju električne energije. Korišćenje obnovljivih izvora energije smanjuje uticaj na okolinu, doprinosi održivosti i efikasnosti sistema, te čini proizvodnju energije dugoročno održivom.

V. Zaključak

Sistem plastenika 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. Zaključno sa svim navedenim ovaj sistem trenutno ima dobru praktičnu primenu i potencijalnu komercijalnu primenu.

Zahvalnica

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

Literatura

[1] Naša web aplikacija -https://agrotemp.cirasoft.net/index.html
[2] Robojax SG90 MicroServo datasheet - https://robojax.com/learn/arduino/robojax-servo-sg90_datasheet.pdf
[3] Bosch Sensortec BMP280 datasheet - https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/
[4] B. Puljko "Primjena ESP8266 komunikacijskog modula u udaljenom upravljanju", 2017, pp. 9

Algorithms for Drone Positioning

Bensaid Djalal Eddine

Abstract - This paper presents a basic theoretical study on how to control a quadcopter drone, some positioning algorithms and a practical realization of a flight controller module with a transmission unit based on Arduino.

I. Introduction

Due to the fast development of automation and control technologies, the unmanned machines are becoming the most interesting industry worldwide, due to their efficiency compared to the same manned machines of their kind, and because: of their low impact on the environment, they reduce the loss of the "human factor", they are flexible and easy to use, they consume less energy, they are cheap to build and to repair, and in some cases, they can reach the most places which are very difficult for humans to reach, some of those machines are: rockets (Falcon1) [1], satellites (Voyager1) [2], deep submarines (Remus) [3], unmanned aerial vehicles (UAV) [4].

The inspiration of unmanned vehicles came in 1898 by the inventor and the electrical engineer Nikola Tesla, he demonstrated his remote-control boat for the first time at Madison Square Garden [5]. After that, based on his research the world started to think about bigger machines during and after World War II, and the need for unmanned aircraft vehicles had increased. during the COVID-19 pandemic the drones had an important role: reducing the infection rate in the world, and delivering food and necessaries of life to the people inside their homes during the quarantine.

II. Background and Related Work

The drone like any autonomous machine is a system based on the discrete closed-loop configuration, the most important components of any drone are the Inertial Measurement Unit (IMU), the Microcontroller, the RF-transceiver, the motors, the propellers and the frame. [6] all these components will be attached to the frame, and the propellers will be attached to the motors.

The RF-transceiver can be set to receive the proper motion data from a transmission unit, or as a transmitter and a receiver periodically, if we want to collect some data from other sensors on the drone and send it to the transmission unit.

The data received from the RF-transceiver will be the reference signal to control the drone, the IMU generally is a 6-axis measurement unit that senses the angular velocity in three axes and also the acceleration in three axes, the microcontroller will do all the calculations to get the appropriate process to control the motors and moves the drone as desired.

B. D. Eddine is a Master student at the Department of Electronics, Faculty of Science and Technology, University of Batna-2, 53, Constantine Road, Fesdis, Batna 05078, Algeria. Email: Bensaid.djalaleddine.2001@gmail.com.

III. The Theory Ulti-Rotors (Multi-Copters)

For more understanding, a simple quad-copter model will be demonstrated, this model has four motors, and each motor tilts from the other by 90 degrees. for the drone motion, there are three translation directions (X, Y and Z) and three rotation directions (Yaw, Pitch and Roll) [7].

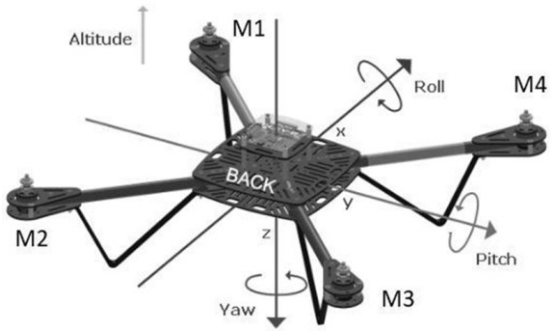


Fig. 1. Representing the axes of translation and rotation in the space [8].

To make the drone rise off the ground, the four motors should increase their speeds, and we will call this command: Altitude for example [9].

For the rest rotations and translations, the speed of the motors should change in every pair of motors (changing the speed of two motors at the same time), one pair of motors should increase its speed while the other should decrease it, the motors M1-M3 will always rotate in the same direction(same as the direction of the yaw axis for example) and in the opposite direction of M2-M4, the engineers insist on this configuration because of the motion that will be generated while actuating the motors, other configurations will make the drone unstable harder to control [7].

If we increase the speed of the motors M2 and M3 and decrease the speed of M1 and M4, we will tilt the drone in the opposite direction of the pitch axis with certain degree, and the vector of the forces generated by the propellers will tilt also, and this will move the drone in the direction of the X axis, and to move it in the opposite direction of the X axis, we should tilt the drone in the direction of the Pitch axis by increasing the speed of M1 and M4, and decreasing the speed of M2 and M3. And we will call this command: P.

Moving the drone in the direction of the Y axis is similar, tilt the drone in the direction of the roll axis by increasing the speed of the motors M1 and M2, and decreasing the speed of the motors M3 and M4. If we do

the opposite, we will tilt the drone in the opposite direction of the Roll axis and move it in the opposite direction of the Y-axis. We will call this command: R.

If we increase the speed of M1 and M3 and decrease the speed of M2 and M4, the drone will rotate in same direction of yaw axis due to the friction between the motor's shaft and the motor's brush, causing the motor support to turn, and because the motor is mounted on the frame, the frame will turn in the direction of the yaw axis. doing the opposite will rotate the drone in the opposite direction of the Yaw axis. And we will call this command: Y.

Depending on those commands, every motion will be quantified to a real variable to make a mathematical approach for the microcontroller to analyze the data. These four variables were mentioned before as motion commands and they are: P, R, Y and A (abbreviation of: Pitch, Roll, Yaw and Altitude respectively), their sign defines the direction of motion ("+"in the same direction of the axis, "-" in the opposite direction of the axis) and their absolute value defines the degree of rotation for the yaw pitch and roll, and the distance from the surface for Altitude. [7]

Each motor M_i on the drone will rotate with a specific speed $_i$, which can be quantified by the variables mentioned before.

$$V_1 = Altitude - Pitch + Roll - Yaw \quad (1)$$

$$V_2 = Altitude + Pitch + Roll + Yaw \quad (2)$$

$$V_3 = Altitude + Pitch - Roll - Yaw \quad (3)$$

$$V_4 = Altitude - Pitch - Roll + Yaw \quad (4)$$

The equations (1), (2), (3) and (4) represents the mathematical approach to control the speed of the motors [10], the values of the variables mentioned before will not control the speed directly, but they will be optimized using an appropriate controller the stabilize the drone.

IV. The Drone System Design

The main blocks of the drone system are: the sensors, the RF-transceiver, the motors, the IMU and the Microcontroller. Together they form the flight controller module of the drone. The so-called flight controller algorithm is the algorithm that has been programmed in the microcontroller, to stabilize and control the drone.

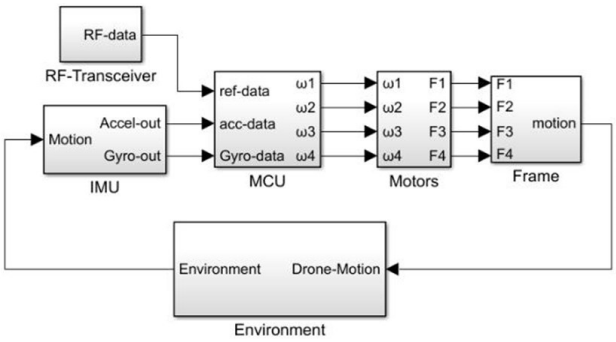


Fig. 2. The drone system blocks with the environment impact [11].

The frame is included in the system design because it impacts the motion, depending on its dimensions, its weight and the materials which it was made of. those factors play a big role on the frame flexibility, strength, vibration absorption and the aerodynamic of the drone.

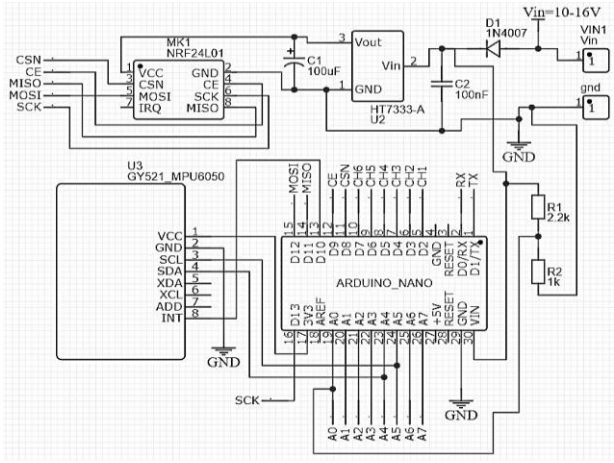


Fig. 3. The schematics of the designed flight controller module based on Arduino using EasyEDA.

The realized flight controller module contains an Arduino Nano module (ATmega328p), a MPU-6050 sensor their measurements are coded by 7ADCs in 16 Bits (one ADC for one axis), it has also an embedded digital motion processor that can process the measurements will be connected to the Arduino pins A4 (SDA) and A5 (SCK) to communicate with it. The NRF24L01 is a common transceiver, it has very important key features like: auto-data handling, payload size of 32 Bytes, 2.4 GHz ISM band, 126RF communication channels and enhanced ShockBurst data link layer. With a good antenna, the range of this transceiver can reach 1.1 km. This transceiver uses the SPI protocol [14] and it will be connected to the Arduino pins D11, D12 and D13, with another two additional pins which are: D9 (CE) to enable the transceiver and D8 (CSN) to select it. The pins D4 to D7 on the Arduino (see the Fig. 3.) are used as PWM channels (CH1 to CH4) to control the motors. The resistors R1 and R2 are used as a voltage divider to read the drone's battery voltage, the diode D1 is used to protect the circuit against the reverse current, and the rest of the pins was connected to an array of channels to control other actuators in the future.

The transmission unit in the Fig. 4. has also an NRF24L01 transceiver, and uses all the pins of the Arduino to maintain the best control, and it includes 2joysticks to control the drone motion: Altitude, Yaw, Pitch and Roll. a potentiometer for the PID gain tuning and another one to set the maximum motion velocity of the drone (velocity of the translations and the rotation on the Yaw-axis). The 6 push buttons and the 2 switches in the transmission unit will be used to control other actuators in the future, except the main motors of the drone. The transmission unit will send the data by setting the NRF24L01 on this circuit into transmission mode and specifying the address of the receiver.

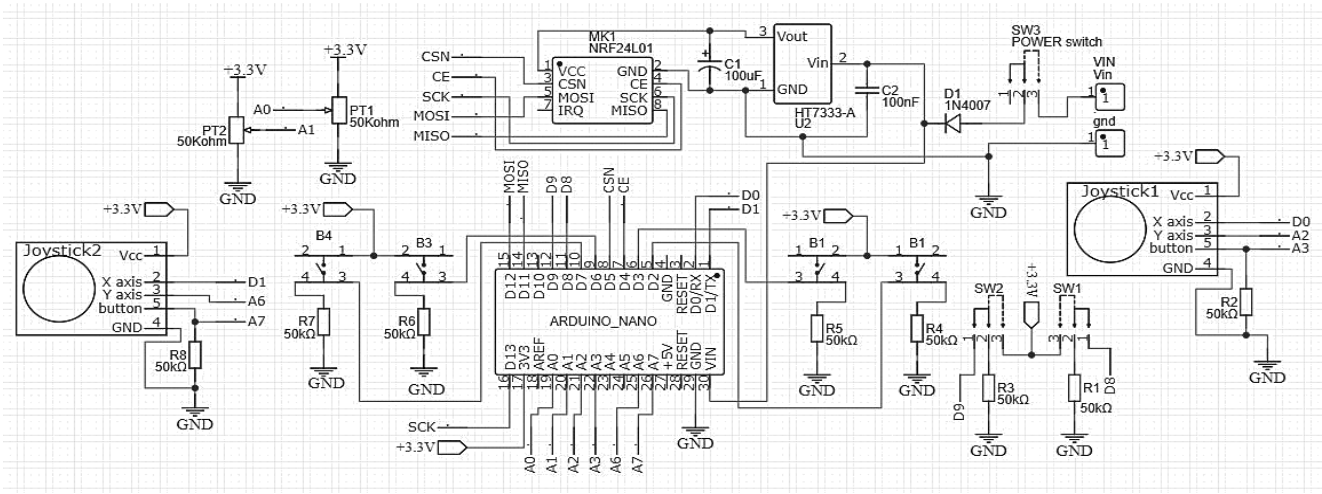


Fig. 4. The schematics of the designed transmission unit based on Arduino using EasyEDA.

V. Positioning Algorithms

The flight controller algorithm can be separated into two sub-algorithms, the first is the Motion algorithm that deals with the IMU, this algorithm will collect the IMU measurement, filter it, integrate it and do all the necessary calculation to get a useful data about the three rotations and sometimes the three translations or only the Z-axis translation, the second algorithm represents the command algorithm which controls the actuators of the drone.

A. Algorithm depends on angle

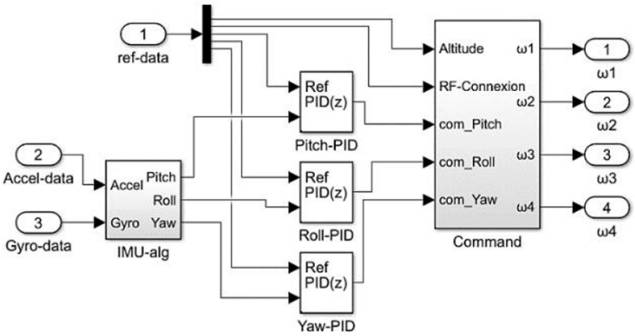


Fig. 5. Flight controller block diagram based on the angle control [7].

There are many flight controller algorithms, the simplest one uses a PID-controllers that depends on the angle error to control the three angles of the quadcopter and stabilize its response [8], this algorithm can be used only for a drone with ideal structure or to tune the PID-gains, without taking in considering any imperfections in the motors or the frame, if we apply this algorithm, the drone may move with a specific speed because of a static error caused by the imperfections mentioned before. to solve this problem, another data can be sent from the transmission unit to reduce the error by adding a constant command, or by using better algorithms.

B. Algorithm depends on velocity

The algorithm that depends on the speed solves the static error problem, but it needs more accurate IMU and more complicated calculations [10], and that will affect the time of the control loop. This algorithm is limited by the performance of the microcontroller, and it uses the angle control algorithm next to it, while the motion algorithm will give another data which are the velocity Vx and Vy.

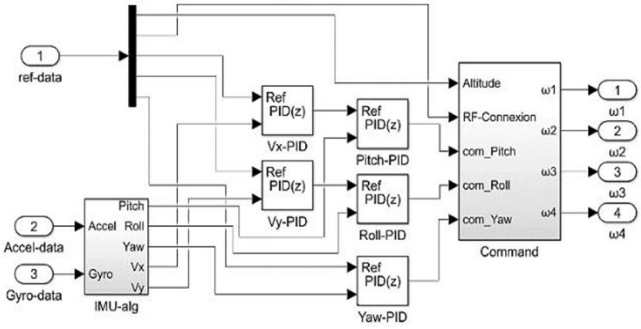


Fig. 6. Flight controller block diagram based on the velocity control [10]-[11].

C. Algorithm depends on position

The algorithm that depends on position is more complicated than the algorithm that depends on the speed, because it needs more accurate and more intelligent IMU and a microcontroller with better performance. This algorithm represents the best choice for a total autonomous flights, and with it the drone can go to a specific point on the space or follow a trajectory without any interference from transmission unit, the position on this algorithm still relative, but it can use a GPS module for further applications [9].

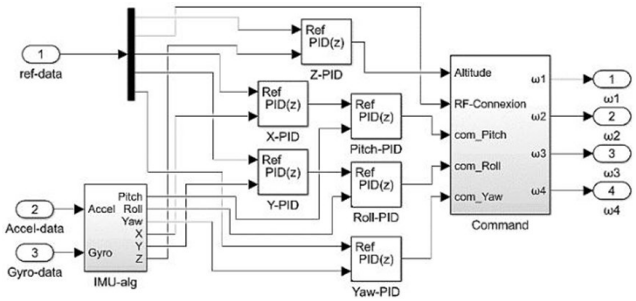


Fig. 7. Flight controller block diagram based on the position control [15]-[16].

VI. Conclusion

The drones are the next generation of the aircraft vehicles, nowadays they represent the start of a new era of fully autonomous vehicles, more features can be added to the drone based on the algorithms mentioned before, the first feature is the auto tuning by using and algorithm to initialize and optimize the PID controller gains of the drone, adding a GPS and compass blocks to achieve the absolute positioning in the space, adding barometer to control the altitude, use ultrasound sensors to control the path and avoid the obstacles during the flight.

Acknowledgement

This work was done during the participation on the "ERASMUS+" mobility program at the Faculty of Electronic Engineering, University of Nis, the spring semester, scholar year: 2022-2023.

References

[1] B. Bjelde, P. Capozzoli and G. Shotwell, "The SpaceX Falcon1 Launch Vehicle Flight 3 Results, Future Developments and Falcon 9 Evolution", p. 1,2,6, 2008.

[2] R. L. Heacock, "The Voyager Space Craft", *Proceedings of the Institution of Mechanical Engineers*, vol. 194, no. 1, pp. 3,4,7, 1980.

[3] T. E. Allen, A. H. Buss and S. M. Sanchez, "Assessing obstacle location accuracy in the REMUS unmanned underwater vehicle", *Proc. of the 2004 Winter Simulation Conference*, 2004, p. 948.

[4] P. G. Fahlstrom, T. J. Gleason and M. H. Sadraey, *Introduction to UAV Systems*, 2022, Wiley, p. 17.

[5] J. Vujić, A. Marinčić, M. Ercegovac and B. Milovanović, "Nikola Tesla: 145 Years of visionary ideas", *Proc. of 5th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service* (TELSIKS), 2001, pp. 323-326.

[6] N. Ahmad, R. Ariffin Raja Ghazilla and N. M. Khairi. "Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications", *International Journal of Signal Processing Systems*, vol. 1, no. 2, p. 256, 2013.

[7] P. Wang, Z. Man, Z. Cao and J. Zheng, "Dynamics modelling and linear control of quadcopter", *Proc. of the International Conference on Advanced Mechatronic Systems*, 2016, pp. 498-501.

[8] Transition and rotation of drone. Available at: https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcT5gSCV_HBlKOf_kIrEOJbr_c4Uet14SOV7VibLw2CsQbvnhkBY

[9] V. Praveen and A. S. Pillai, "Modeling and simulation of quadcopter using PID controller", *International Science Press*, pp. 7151-7154, 2016.

[10] B. Sumantri, N. Tamami, B. Kurniawan and Y. Birawa Nuraga, "Development of a Low-cost Embedded Flight Controller for Quadcopter", *Proc. of International Electronics Symposium*, 2020, pp. 233-235.

[11] D. Li, "Modeling, controlling, and flight testing of a small quadcopter", p. 7,16,19,23,29.

[12] ATmega 328P datasheet. Available at: https://pdf1.alldatasheet.com/ datasheet-pdf/view/313656/ATMEL/ATmega328P.html

[13] MPU6050 datasheet. Available at: https://pdf1.alldatasheet.com/datasheet-pdf/view/517744/ETC1/MPU-6050.html, and https://pdf1.alldatasheet.com/datasheet-pdf/view/1132807/TDK/MPU-6050.html

[14] NRF24L01 datasheet. Available at: https://pdf1.alldatasheet.com/datasheet-pdf/view/1489848/NORDIC/NRF24L01.html

[15] H. Jin Kim and D. H. Shim, "A flight control system for aerial robots: algorithms and experiments", *Control Engineering Practice*, vol. 11, no. 12, pp. 1389-1400, 2003.

[16] J. M. Roberts, P. I. Corke and G. Buskey, "Low-cost Flight control system for a small autonomous helicopter", Proc. of the 2003 International Conference on Robotics and Automation, 2003, p. 547.

Pametni sistem za detekciju pokreta zasnovan na Arduino platformi

Stefan Korica

Sadržaj – U ovom radu su predstavljene funkcionalnosti PAJ7620 senzora pokreta na praktičnom primeru sistema za detekciju pokreta zasnovanog na *Arduino Uno* platformi. Takođe je predstavljena i komunikacija između *Arduino* mikrokontrolera i računara preko serijskog porta na primeru C# aplikacije.

I. UVOD

Praćenje i prepoznavanje pokreta je bitna funkcionalnost za mnoge sisteme, naročito tzv. *smart home* sisteme u kojima se različitim pokretima mogu izdavati naredbe modulima sistema. Senzori za detekciju pokreta se stoga mogu koristiti u pametnim prekidačima za svetlo [1], u prekidačima na industrijskim mašinama, kao brojači prisutnih ljudi u zatvorenom prostoru. i sl. Ovakvi prekidači mogu imati višestruke prednosti u odnosu na tradicionalne: lakši su za upotrebu, higijenski prihvatljiviji, dugotrajniji itd.

Pri razvoju sistema za detekciju pokreta naročito je važno obratiti pažnju na brzinu detekcije (prespor sistem će učiniti sistem neefikasnim) kao i na smanjenje grešaka u detekciji pokreta (poput prepoznavanja pogrešnog pokreta ili izostanka detekcije u potpunosti).

Ovaj projekat je zasnovan na *Arduino* platformi i PAJ7620 senzoru. Projekat služi kao prikaz funkcionalnosti ovog senzora u kombinaciji sa odgovarajućim mikrokontrolerom, a koje funkcionalnosti bi se mogle koristiti u praktičnim proizvodima.

Rad se sastoji od četiri poglavlja: uvoda, prikaza korišćenih komponenata, principa rada, i zaključka.

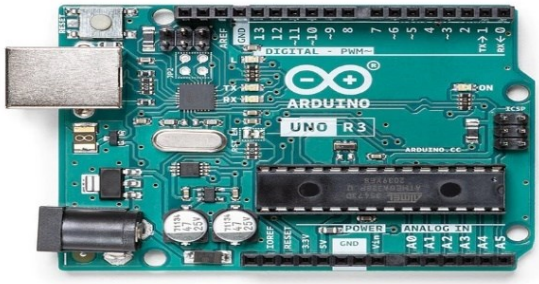
II. KORIŠĆENE KOMPONENTE

A. Arduino Uno R3

Arduino je jedna od najpopularnijih porodica mikrokontrolera, razvijena 2006. godine u Italiji. Ono što ju je učinilo naročito popularnom jeste to što je čitav Arduino projekat *open source* što znači da svako može koristiti Arduino pločice u svojim projektima [2]. Za pisanje programa koje Arduino mikrokontroler izvršava koristi se Arduino IDE razvojno okruženje, a programi se pišu u modifikovanom C++ programskom jeziku.

S. Korica je učenik smera Elektrotehničar računara, Školski Centar „Nikola Tesla" Vršac, Sterijina 40-44, 26300 Vršac, Srbija. Email: koricastefan5@gmail.com.

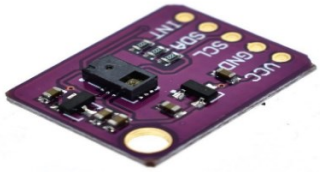
U ovom radu je korišćena *Arduino Uno* ploča sa ATMEGA328P procesorom [3], prikazana na slici 1.



Slika 1. Arduino UNO R3 mikrokontroler.

B. PAJ7620 senzor pokreta

PAJ7620 je senzor namenjen detekciji pokreta. Njegovu osnovu čini IR 940 nm LED svetlo i nekoliko optičkih senzora. Nakon što detektuje pokret, on se putem A/D konvertora prebacuje i digitalni oblik a zatim se finalni rezultat u obliku „flag"-a šalje mikrokontroleru [3]. Sam senzor može prepoznati 9 pokreta: gore, dole, levo, desno, napred, nazad, „clockwise", „counter-clockwise" i „wave". Senzor pouzdano detektuje pokrete na razdaljini od oko 20cm. Senzor je izuzetno lak za povezivanje i ima 4 pina: VCC i GND su naponski pinovi (ulazni napon i uzemljenje) dok se za komunikaciju putem I2C protokola koriste SDA i SCL pinovi [4]. Za korišćenje senzora neophodno je instalirati odgovarajuću biblioteku u Arduino IDE pre pisanja programa, o čemu će biti više reči u poglavlju „realizacija projekta".

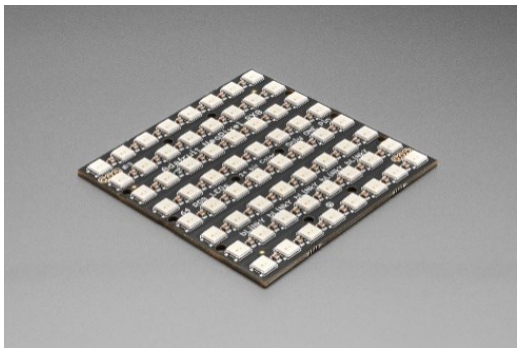


Slika 2. PAJ7620 senzor pokreta.

C. Adafruit NeoPixel 8x8 LED matrica

Adafruit NeoPixel matrica će se u ovom projektu koristiti za animirani prikaz detektovanih pokreta. Matrica

sadrži 64 LED diode raspoređenih u formi kvadrata 8x8. Sama matrica koristi tri pina za povezivanje: VCC i GND pinovi za napajanje te DIN pin kojim se matrica povezuje sa Arduino mikrokontrolerom. Matrica takođe posetuje i dodatni set od 3 pina na koji se može prikačiti ista takva matrica [4]. U zavisnosti od jačine osvetljenja, svaka dioda koristi do 60 mA struje (odnosno do 3.84 A struje za ceo panel), pa se za napajanje jedne matrice može koristiti i 5 V izlaz iz samog Arduina, dok je za više matrica u nizu potrebno koristiti eksterne izvore napajanja poput 9(5) V baterije [5]. Za upravljanje matricom preko Arduina potrebno je instalirati zasebnu NeoPixel biblioteku, pomoću koje se može kontrolisati boja svake pojedinačne diode u RGB opsegu, kao i jačina osvetljenja.



Slika 3. Adafruit NeoPixel 8x8 LED matrica.

III. Princip rada

A. Opis funkcionalnosti

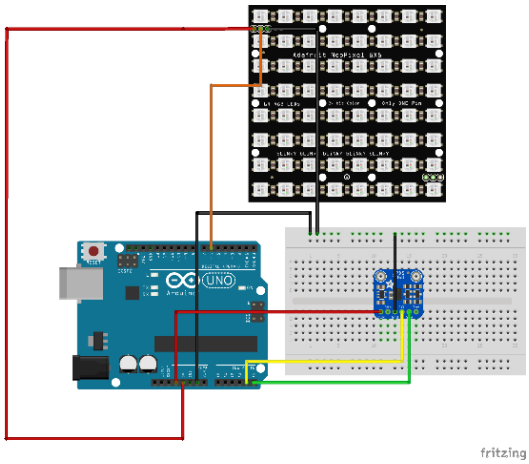
Kada korisnik načini određeni pokret rukom, njega prvo detektuje senzor pokreta. Optički senzori u okviru tog modula šalju analogni izlaz, koji se zatim u samom modulu pretvara u digitalni. Digitalni izlaz se zatim, u obliku „flag-a“ šalje u Arduino mikrokontroler.

Mikrokontroler zatim na osnovu primljenog podatka pokreće odgovarajuću animaciju na LED matrici, a takođe šalje i tekstualni opis pokreta preko serijskog izlaza ka računaru gde se taj tekstualni opis može prikazati putem C# aplikacije.

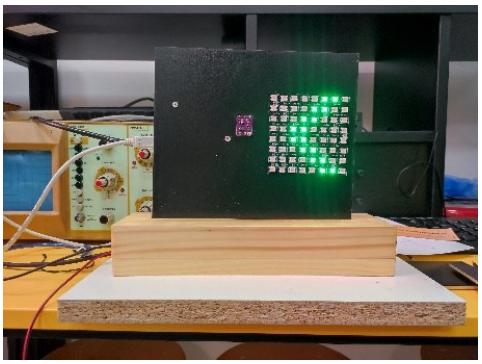
B. Fizičko povezivanje komponenata

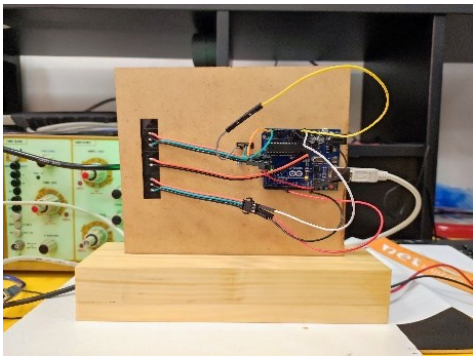
Projekat je prvo povezan isključivo „jumper“ žicama preko proto ploče, kako bi se proverio rad uređaja i otklonile eventualne greške. Povezivanje je učinjeno na sledeći način: iz Arduino mikrokontrolera je doveden GND direktno na protoploču. VCC je doveden iz 5 V pina ka LED matrici, odnosno iz 3.3 V izlaza ka senzoru pokreta. Senzor pokreta je sa Arduinom povezan preko pinova A4 i A5 koji služe za I2C komunikaciju, dok je LED matrica sa mikrokontrolerom povezana preko pina 6.

Nakon provere ispravnosti sklopljenog sistema pristupilo se učvršćivanju komponenata. Praktična realizacija komponenata prikazana je na slikama 5a i 5b.



Slika 4. Šema povezivanja komponenti u programu Fritzing.





Slika 5. a. Finalni proizvod (prednja strana). b. Finalni proizvod (zadnja strana).

C. Kodovi korišćeni u projektu

Kod korišćen u projektu može se podeliti na dve celine:

1) Kod za Arduino mikrokontroler,
2) Kod za Windows program.

Kod za Arduino mikrokontroler

Na početku Arduino koda neophodno je uvesti biblioteke i definisati LED matricu koju ćemo koristiti za prikaz animacija:

```
#include <Wire.h>
#include "paj7620.h"
#include <Adafruit_NeoPixel.h>

#define LED_PIN    6

#define LED_COUNT 64

Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB +
NEO_KHZ800);
```

Zatim se u `setup` funkciji otvara serijski port i inicijalizuje se senzor pokreta i LED matrica:

```
void setup() {
  Serial.begin(9600);
  Serial.println("\nPAJ7620U2 TEST DEMO: Detekcija
pokreta");

  int error = paj7620Init();
  if (error)
  {
    Serial.print("GREŠKA:");
    Serial.println(error);
  }
  else
  {
    Serial.println("INIT OK");
  }
  Serial.println("Molim unesite pokret:\n");
strip.begin();
strip.show();
strip.setBrightness(50);
}
```

Detekcija pokreta radi se uz pomoć `switch` petlje, gde je svaki „flag“ odnosno pokret zaseban `case`. Svaki `case` poziva odgovarajuću funkciju za prikaz animacije i šalje opis pokreta na serijski port:

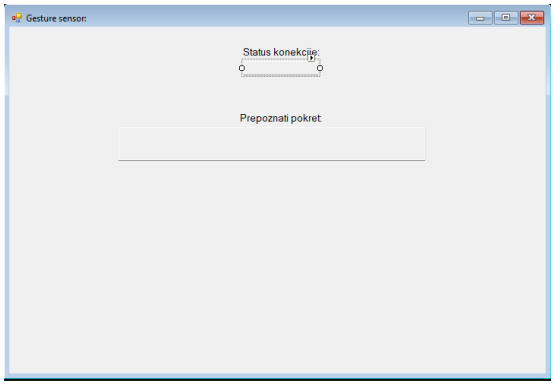
```
switch (data){
      case GES_RIGHT_FLAG:
          Serial.println("Desno");
          right(strip.Color(0,   255,   0),50);
        break;
      case GES_LEFT_FLAG:
          Serial.println("Levo");
          left(strip.Color(0,   255,   0),50);
        break;
      case GES_UP_FLAG:
          Serial.println("Gore");
          up(strip.Color(0,   255,   0),50);
        break;
      case GES_DOWN_FLAG:
          Serial.println("Dole");
          down(strip.Color(0,   255,   0),50);
        break;
}
```

Animacije se vrše uz pomoć nizova i `for` petlje. Svaki niz sadrži redne brojeve piksela koje se osvetljavaju u jednoj statičnoj slici. Zatim se u for petlji svaka dioda podešava na zadatu boju komandom `strip.setColor`, a na kraju se osvetljene diode prikazuju komandom `strip.show`:

```
int niz1[17] = {2, 3, 11, 12, 20, 21, 25, 26, 37, 38,
42, 43, 51, 52, 60, 61};
int niz2[17] = {3, 4, 10, 11, 21, 22, 24, 25, 38, 39,
41, 42, 52, 53, 59, 60};
for(int i=0; i<16; i++) {
 strip.setPixelColor(niz1[i], color);
}
  strip.show();
  delay(100);
  strip.clear();
for(int i=0; i<16; i++){
    strip.setPixelColor(niz2[i], color);
  }
  strip.show();
  delay(100);
  strip.clear();
```

Kod za Windows program

Postoji više načina kako se može očitati izlaz koji Arduino šalje preko svog serijskog porta. Jedan od načina na koji je to moguće uraditi je preko C# programa, čiji interfejs je prikazan na slici 6.



Slika 6. Izgled korisničkog interfejsa programa.

Na početku programa neophodno je definisati serijski port i otvoriti ga, te definisati interval tajmera:

```
public Form1()
{
    InitializeComponent();
    timer1.Interval =500;
    timer1.Start();
    serialPort = new SerialPort("COM3", 9600);

    try
    {
        serialPort.Open();
    }
    catch
    {
        textBox2.BackColor = Color.Red;
        textBox2.Text = "neuspesno";
    }
}
```

Nakon toga se pokušava očitavanje podataka sa serijskog porta svaki put kada tajmer zada signal (tick). U slučaju uspešnog očitavanja, primljeni podatak u vidu

string-a prikazujemo u **textBox1**, dok se u **textBox2** ispisuje da je konekcija uspešna:

```
private void timer1_Tick(object sender, EventArgs
e)
{
    bool izlaz = false;
    string pokret = string.Empty;

    while (!izlaz)
    {
        pokret = serialPort.ReadLine();
        if (!string.IsNullOrEmpty(pokret))
        izlaz = true;
        {
            izlaz = true;
        }
    }

    if (izlaz)
    {
        textBox2.Text = "uspesno";
        textBox2.BackColor = Color.LawnGreen;
        textBox1.Text = pokret;

    }
}
```

IV. Zaključak

Projektovani sistem je vrlo štedljiv u pogledu potrošnje električne energije, tako da nije potreban eksterni izvor napajanja, već se sklop napaja putem USB kabla kojim se Arduino povezuje sa računarom. Takođe, sistem je izrazito pouzdan – na 100 pokušaja, senzor je svega 5% pokreta detektovao pogrešno, i to su uglavnom bili komplikovaniji pokreti (poput pokreta u smeru kazaljke na satu ili obrnuto od njega).

Ovakav sistem može imati primenu u različitim sferama života. Na primer, može se koristiti kao beskontaktni prekidač svetla, ventilatora i sl. ili za brojanje ljudi u nekom zatvorenom prostoru (što je naročito korisno u situacijama kada je broj ljudi u tom prostoru ograničen, na primer za vreme epidemije virusa i sl.). Takođe, ovakav sistem se može koristiti u domaćinstvima, javnim mestima/institucijama, kao i u industrijskim kompleksima.

Sistem se može dodatno unaprediti uvođenjem beskontaktne konekcije sa računarom ili mobilnim telefonom posredstvom *Bluetooth* tehnologije.

Zahvalnica

Autor se zahvaljuje profesorima-mentorima Milanu Njegomiru i Nenadu Cvetkoviću na savetima i sugestijama u toku izrade rada.

Literatura

[1] https://www.electronicsforu.com/electronics-projects/automatic-contactless-switch-for-smart-home
[2] https://en.wikipedia.org/wiki/Arduino
[3] https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf
[4] https://www.epsglobal.com/Media-Library/EPSGlobal/Products/files/pixart/PAJ7620F2.pdf?ext=.pdf
[5] https://www.adafruit.com/product/1487

Pristup dubokog učenja za predviđanje hidroloških vremenskih serija sa modelom ekstremnog mašinskog učenja

Miljan G. Jeremić

Sadržaj - Prediction is a significant undertaking for resources planning and management. Several traditional and the deep learning models have been applied to simulate the complex nature of the system. The aim of this research is to apply a deep learning model to simulate the complex nature of the hydrological system. In this paper, the Extreme Learning Machine (ELM) methodology is investigated to predict one step ahead for solving reference evapotranspiration (ET0) measured for the City of Niš, Serbia.The performance analysis of the ELM model is also presented. The ELM model performed superfast as compared to traditional/deep learning architecture, gives comparable performance with fast execution rate. In numerical terms, the superiority of ELM over other model (Convolution Neural Network – 1D-CNN) is more expressive. It can be concluded that the ELM model would be an alternative to the 1D-CNN model for highly accurate streamflow forecasting in many regions of the world.

I. PREGLED MODELA KORIŠĆENOG U ISTRAŽIVANJU

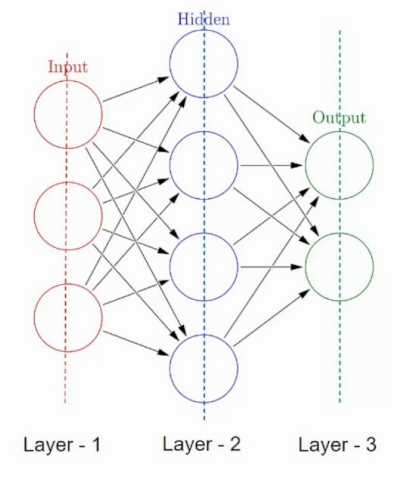
Ekstremno mašinsko učenje (ELM) je tehnika mašinskog učenja koja se koristi za zadatke klasifikacije ili regresije [1].

Ekstremno mašinsko učenje (ELM) čini neuronske mreže sa prosleđivanjem unapred (SLFN) sposobnim za brže učenje u poređenju sa tehnikama učenja zasnovanim na gradijentu. To je kao klasična neuronska mreža sa jednim skrivenim slojem bez procesa učenja. Ovaj tip neuronske mreže ne vrši iterativno podešavanje, koje je brže sa boljim performansama generalizacije od mreža obučenih metodom povratnog širenja [2].

Mašina za ekstremno učenje (ELM) je jedan od algoritama neuronske mreže koji koristi pojedinačne skrivene slojeve da prosleđuje vrednosti unapred. Topologija ELM mreže ima tri sloja: ulazni sloj, skriveni sloj i izlazni sloj kao što je prikazano na Slici 1.

Algoritam za ekstremno mašinsko učenje je jedan od najefikasnijih algoritama mašinskog učenja u neuronskim mrežama. Zbog neiterativnog treninga, svi parametri se postavljaju jednom. Ovo rezultira treningom velike brzine. Njegova primena je laka za razumevanje i može se koristiti za rešavanje složenih problema.

M. G. Jeremic, Knjaževačka gimnazija, Knjaževac, miljan.jeremic@gmail.com



Slika 1. Neuronska mreža sa skrivenim slojevima.

II. MATEMATIČKA NOTACIJA PREDSTAVLJENOG MODELA

A. ELM model

Za ELM tehniku imamo sledeći opis aktivnosti:
Training set

$$\{(x_i, t_i) || x_i \in R^d, t_i \in R^m, i=1, ..., n\} \quad (1)$$

- Funkcija skrivenog izlaza čvora H (*w, b, x*)
- Broj skrivenih čvorova L.

ELM može biti primenjena u 3 prosta koraka:

1. Nasumično se dodeljuju skriveni parametri čvora (*w, b*)
2. Izračunava se izlazna matrica skrivenog sloja X.
3. Izračunava se izlazna težina β

Linearna regresija je statistička metoda za modeliranje linearne veze između zavisne promenljive Y (tj. one koju želimo da predvidimo) i jedne ili više nezavisnih promenljivih (X).

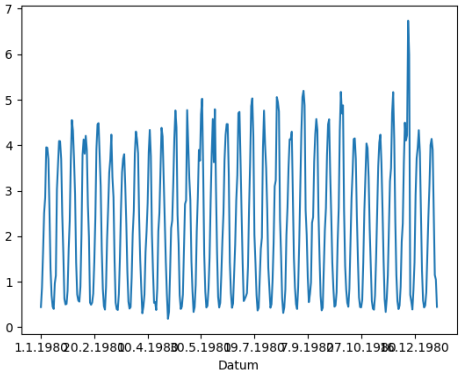
Grafovi reziduala se koriste za traženje osnovnih obrazaca u rezidualima koji ukazuju na to da model ima problem. Kada se koristi funkcija plot(), prvi grafikon predstavlja odnos reziduala prema odgovarajućim

podešenim vrednostima prikaza i pokazuje da li postoje nelinearni obrasci. Za ispravnu linearnu regresiju, podaci moraju biti linearni, pa će se testirati da li je ovaj uslov ispunjen.

III. Opis posmatranih serija podataka

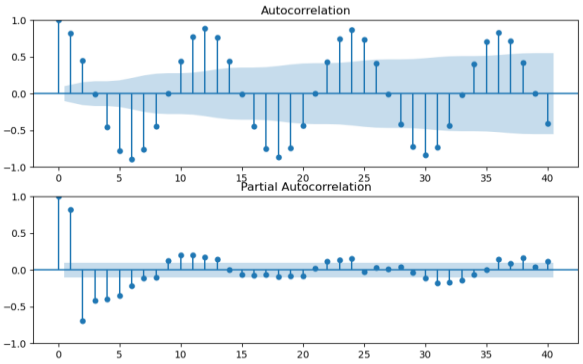
Skup podataka je upisan u datoteci o referentnoj evapotranspiracija u Excel fajlu u CSV format. Datoteka sadrži 372 zapisa sa 10 kolona, gde jedna od njih predstavlja referentnu evapotranspiraciju merenu za stanicu u Nišu u oznaci ET0 [3]. Skup podataka za treniranje (train set) i testiranje (test set) mreže podeljen je u nasumičnom odnosu 80:20%. Nakon štampanja vrednosti dela prvog skupa podataka (Dataset), možemo videti sledeći ispis prvih 5 vrednosti u skupu podataka i grafički prikaz vrednosti referentne evapotranspiracije na Slici 2.

	Vrednost ET0
0	0.441223
1	0.866711
2	1.639383
3	2.507400
4	2.848985



Slika 2. Prikaz serije podataka.

Dijagram vrednosti autokorekcije i delimične autokorelacije serije podataka u oba modela prikazan je na Slici 3 [9].



Slika. 3. Prikaz vrednosti ACF i PACF korelacionih funkcija.

A. ELM odel opisa podataka

Ovde će biti prikazan opis vrednosti referentne evapotranspiracije ET0 [4]. Prikazana iz serija podataka za ELM model:

	ET0
count	75.000000
mean	1.280418
std	1.591741
min	-0.666446
25%	-0.296320
50%	1.175866
75%	2.660529
max	5.737676

Pogled statističkih vrednosti referentne evapotranspiracije koristeći programski jezik Python izgleda ovako:

	ET0
min	-0.666446
max	5.737676
median	1.175866
skew	0.303705

Autokorelacione, ACF, i parcijalne autokorelacione vrednosti, PACF, izračunate primenom prethodnih koraka izgledaju ovako:

```
[ 1.         0.78858043  0.39702449  0.00966969 -0.35967167 -0.64363142
 -0.72375602 -0.60485872 -0.35656612 -0.00716138  0.33908231  0.57340375
  0.64046235  0.56465039  0.35760673  0.03930548 -0.29906944 -0.52602835
 -0.59258217 -0.51969602 -0.30164495 -0.01263757  0.26807426  0.45976616
  0.52001322  0.45478161  0.30966207  0.08434253 -0.17764435 -0.38343344
 -0.48518263 -0.46463821 -0.33227911 -0.11788161  0.10657593  0.28982309
  0.3855381   0.37140856  0.26385609  0.08919863 -0.09398137]

[ 1.         0.78858043  0.39702449  0.00966969 -0.35967167 -0.64363142
 -0.72375602 -0.60485872 -0.35656612 -0.00716138  0.33908231  0.57340375
  0.64046235  0.56465039  0.35760673  0.03930548 -0.29906944 -0.52602835
 -0.59258217 -0.51969602 -0.30164495 -0.01263757  0.26807426  0.45976616
  0.52001322  0.45478161  0.30966207  0.08434253 -0.17764435 -0.38343344
 -0.48518263 -0.46463821 -0.33227911 -0.11788161  0.10657593  0.28982309
  0.3855381   0.37140856  0.26385609  0.08919863 -0.09398137]
```

B. Prikaz mera ACF i PACF dobijenih u modelima

Mere vezane za ELM model se opisuju određenim greškama [5]. One su prikazane u sledećoj tabeli [5]: Ove mere predstavljene u tabeli su sledeće: RMSE – srednja kvadratna greška je uobičajena mera razlike između vrednosti (vrednosti uzorka ili populacije) predviđenih modelom ili procenjivačem i posmatranih vrednosti. MAE - srednja apsolutna greška, mera grešaka između uparenih posmatranja koja izražavaju istu pojavu. Rho - Spearmanov koeficijent.

TABELA I
VREDNOSTI GREŠAKA U ELM MODELU

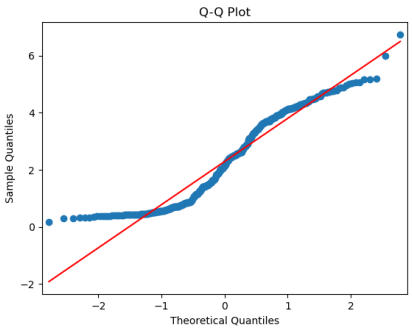
Model	RMSE	MAE
ELM	- 0.6558262673572601	1.5512129

Za prikaz vrednosti Spirmanov koeficijenta Rho dobija se sledeći ispis:

```
[[ 1.         0.         0.07561275 ... -0.02902272 -0.03482726
  -0.07545906]
 [ 0.         1.         0.24097135 ... -0.03759894  0.02255937
   0.05263852]
 [ 0.07561275  0.24097135  1.         ... -0.0019339   0.06067625
   0.01377907]
 ...
 [-0.02902272 -0.03759894 -0.0019339  ...  1.         -0.00269542
  -0.00269542]
 [-0.03482726  0.02255937  0.06067625 ... -0.00269542  1.
  -0.00269542]
 [-0.07545906  0.05263852  0.01377907 ... -0.00269542 -
0.00269542
   1.        ]]
```

C. Q-Q model diagrama

U statistici, Q-Q (kvantil-kvantil) grafovi igraju veoma vitalnu ulogu u grafičkoj analizi i upoređivanju dve distribucije verovatnoće crtanjem njihovih kvantila jedan protiv drugog [6]. Ako su dve distribucije koje se upoređuju potpuno jednake, tada će tačke na tački Q-Q savršeno ležati na pravoj liniji y = x. Prikazan je dijagram za ELM model na Slici 4 .



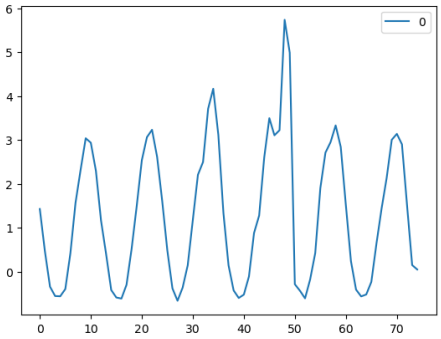
Slika. 4. Prikaz Q-Q grafa.

Veličina ostatka je dužina vertikalne linije od tačke do tačke u kojoj zadovoljava liniju regresije [7].

Prikaz prvih 5 preostalih vrednosti veličine ostatka izgledaju ovako:

```
          0
0 -4.648255
1 -11.724967
2   0.913462
3  -7.687643
4  -9.017658
```

Grafički prikaz preostalih vrednosti je prikazan na Slici 5 [8].

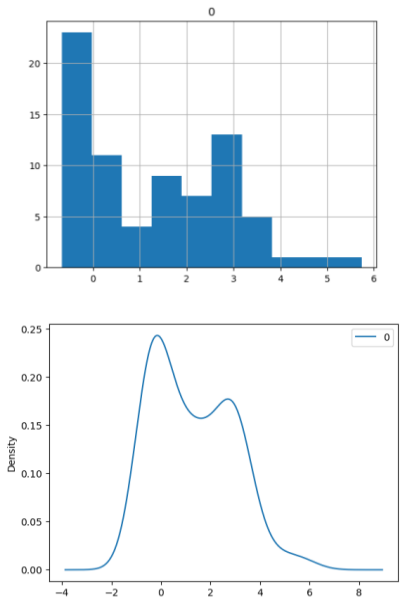


Slika 5. Grafička reprezentacija rezidualnih vrednosti.

Opis preostalih rezidualnih vrednosti je takođe dat u numeričkom obliku [9]:

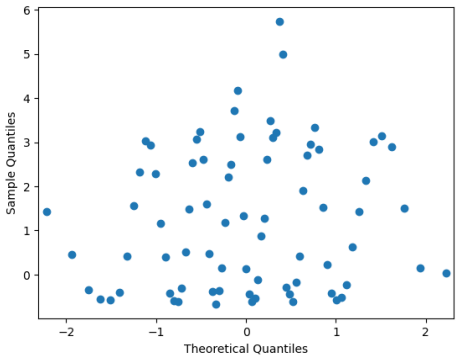
```
                0
count  1241.000000
mean     -0.068408
std       8.557295
min     -65.977423
25%      -3.905593
50%      -0.333333
75%       3.343750
max      67.354167
```

Grafički prikaz rezidualnih vrednosti u vidu histograma prikazan je na Slici 6.



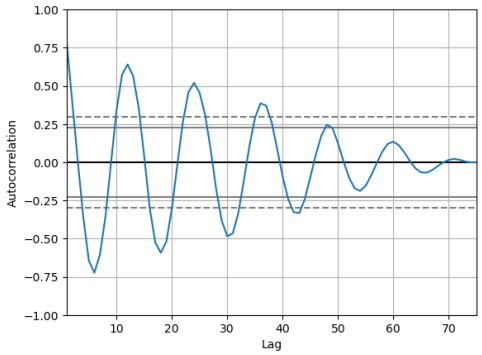
Slika. 6. Grafička reprezentacija rezidualnih vrednosti u obliku histograma.

Q-Q dijagram prikaza preostale vrednosti je prikazan na slici 7.



Slika. 7. Q-Q diagram rezidualnih vrednosti.

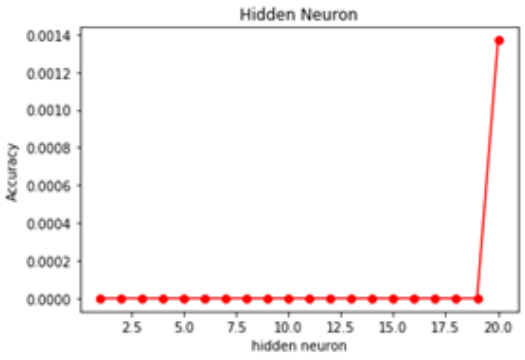
Autokorelacioni dijagram rezidualnih vrednosti je prikazan na Slici 8.



Slika. 8. Grafička reprezentacija reziduala autokorelacije.

IV. TAČNOST MODELA

Provera tačnosti modela korišćenjem stabala odlučivanja, korišćenjem Weka softvera koji prikazuje skup podataka u .arff formatu daje sledeći prikaz za ELM model: na osnovu modela sa 25 skrivenih neurona dobijen je sledeći prikaz tačnosti modela prikazan na Slici 9.



Slika. 9. Grafička reprezentacija mere skrivenih neurona. Accuracy: 0.0013679890560875513,.The best hidden neuron: 20. Mean Accuracy Training: 0.0017656012176560122. Mean Accuracy Testing: 0.0013698630136986301

V. ZAKLJUČAK

Na osnovu analize meteoroloških podataka za period od 1980. do 2010. godine vidi se da su u period od 1990 do 2004. Vrednosti klimatskih podataka ekstremniji, sa ne velikim odstupanjem od proseka. Izvedena analiza pokazuje neregularnost javljanja sušnih i kišnih godina, što pokazuje da je klima u Srbiji humidnog tipa.

Na osnovu svega navedenog, ELM model mašinskog učenja se pokazao kao adekvatan za uočeni problem u odnosu na druge modele (regresija ili konvolucione neuronske mreže). ELM se pokazao kao lakši za pisanje programa u programskom jeziku Python.

LITERATURA

[1] Ian H. Witten, Eibe Frank, Data Mining, *Practical Machine Learning Tools and Techniques*, Elsevier, 2005.

[2] Igor Ševo, *Specijalizovana neuronska mreža za klasifikaciju i segmentaciju aero-snimaka*, doktorska disertacija, Banja Luka, 2020.

[3] Jinyoung Rheea, Jungho Im, "Meteorological drought forecasting for ungauged areas based on machine learning: Using long-range climate forecast and remote sensing data", *Agricultural and Forest Meteorology,* 2017, vol. 237–238 pp. 105–122.

[4] I. Goodfellow, Y. Bengio i A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*, The MIT Press, 2016.

[5] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, C. Suen, A. Coates, A. Maas, A. Hannun, B. Huval, T. Wang i S. Tandon, „*Multi-Layer Neural Network*," UFLDL Tutorial, Dostupno na: http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/.

[6] Artificial Neural Networks/Activation Functions, Dostupno na: https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions#/media/File:ArtificialNeuronModel_english.png.

[7] Masinsko ucenje je zabavno, Dostupno na: https://startit.rs/masinsko-ucenje-je-zabavno/.

[8] Build an Extreme Learning Machine in Python Dostupno na: https://towardsdatascience.com/build-an-extreme-learning-machine-in-python-91d1e8958599.

[9] A Gentle Introduction to Autocorrelation and Partial Autocorrelation, Dostupno na: https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/.

Unapređenje učenja studenata kroz implementaciju sistemskog rešenja za zadatke uz korišćenje OCR tehnologije i ChatGPT-a

Dajana Milošević, Stefan Đokić

Sadržaj - U današnjem digitalnom dobu, učenje i rešavanje zadataka postaju sve veći izazovi za studente. Ovaj rad predstavlja inovativan sistemski pristup rešavanju ovog problema. U ovu svrhu je implementirana aplikacija koja koristi Optical Character Recognition (OCR) tehnologiju za prepoznavanje teksta sa slike zadataka, a zatim se koristi ChatGPT sa naprednim jezičkim modelom kako bi se dobila rešenja i objašnjenja za te zadatke. Ovaj rad razmatra prednosti i ograničenja ovako implementiranog sistema, pruža dublji uvid u implementaciju, i razmatra potencijalna unapređenja sistema.

I. UVOD

U današnjem obrazovnom okruženju, studenti se suočavaju sa brojnim izazovima koji otežavaju njihov proces učenja i rešavanja zadataka. Ovi izazovi obuhvataju širok spektar problema koji utiču na efikasnost i razumevanje studenata. Neki od ključnih izazova u ovom kontekstu uključuju:

1. Nedostatak vremena, studenti često imaju gust raspored sa predavanjima što može predstavljati ozbiljnu prepreku za posvećivanje dovoljno pažnje svakom zadatku koji treba analizirati i rešiti.
2. Nedostatak resursa, studenti se često suočavaju sa problemom nedostatka pristupa odgovarajućim resursima ili finansijskim ograničenjima koji se odnose na nabavku potrebne literature i sadržaja.
3. Složenost zadataka, neki zadaci i problemi koji se sreću u oblastima kao što su matematika, programiranje i naučne discipline mogu biti izuzetno složeni i zahtevaju duboko razumevanje koncepata.
4. Poteškoće u razumevanju, gde studenti imaju teškoće u razumevanju uputstava za zadatke ili u interpretaciji problema.

D. Milošević, master inženjer elektrotehnike i računarstva, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dajana.milosevic@gmail.com.

S. Đokić, student završne godine osnovnih studija na modulu Računarstvo i informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefan.djokic1995@gmail.com.

U cilju prevazilaženja ovih izazova, u ovom radu istražujemo implementaciju sistema koji omogućava studentima brži pristup rešenjima i objašnjenjima za zadatke. Korišćenjem tehnologija kao što su OCR (Optical Character Recognition) i ChatGPT, pružamo mogućnost podrške studentima u njihovom učenju i rešavanju zadataka. Na ovaj način je studentima omogućeno unapređenje iskustva u razumevanju ključnih koncepata, bez obzira na izazove sa kojima se suočavaju.

II. KORIŠĆENE TEHNOLOGIJE

Tehnologije koje su korišćene za implementaciju sistema jesu .NET Core, kao osnovna platforma za razvoj aplikacije, kao i Python skripta koju koristimo za integraciju sa OCR tehnologijom za prepoznavanje teksta sa slike, kao i integracija sa ChatGPT-om.

A. Optical Character Recognition (OCR)

Optical Character Recognition (OCR) je ključna tehnologija u implementaciji sistema je omogućava prepoznavanje i ekstrakciju teksta sa slike. OCR funkcioniše tako što analizira sliku piksel po piksel i pokušava da prepozna slova, brojeve i simbole [1-3]. Neki od bitnijih aspekata koje bi trebalo izdvojiti prilikom analize ove tehnologije jesu:

1. Algoritmi za prepoznavanje teksta koriste tehnike kao što su detekcija kontura, segmentacija teksta i klasifikacija znakova kako bi prepoznali tekst sa slike [4, 5].
2. Kvalitet slika igra ključnu ulogu u tačnosti OCR-a. Slike sa lošom rezolucijom ili oštećenim teksom mogu dovesti do grešaka u prepoznavanju. U ovu svrhu se koriste filteri za uklanjanje šuma i poboljšanje kontrasta [6].
3. OCR sistem takođe treba da se nosi sa jezičkim preprekama, kao što su fontovi, jezici i stilovi pisanja. To zahteva upotrebu jezičkih modela kako bi se prepoznali specifični znakovi i simboli [9].
4. Greške i korekcije: Važno je napomenuti da nijedan OCR sistem nije savršen i da mesta za grešku prilikom prepoznavanja teksta sa slike uvek ima. U ovu svrhu, poboljšanja tačnosti sistema,

implementirani su mehanizmi za detekciju i korekciju ovih grešaka [7, 8].

B. Integracija sa ChatGPT

Integracija sa ChatGPT-om je ključni deo našeg sistema koji omogućava da prepoznati tekst sa slike bude poslat ChatGPT-u na dalju obradu. Ovaj deo integracije funkcioniše na sledeći način:

1. Pretvaranje teksta u ChatGPT format: Nakon što OCR prepozna tekst sa slike, potrebno ga je formatirati tako da bude pogodan za dalju obradu od strane ChatGPT-a. Tekst se može podeliti na odgovarajuće ulazne rečenice ili segmente koji će biti razumljivi ChatGPT-u.
2. Slanje upita ChatGPT - u: Nakon formatiranja teksta, šaljemo ga kao upit ChatGPT-u. Upit može sadržati i dodatne informacije kao što su instrukcije za rešavanje zadatka ili zahtev za objašnjenjem.
3. Obrada rezultata: Nakon što ChatGPT generiše odgovor, on će se sastojati od dva ključna aspekta: rešenje zadatka i objašnjenje. Rešenje može biti programski kod ili odgovarajući odgovor na matematički ili logički zadatak, dok objašnjenje predstavlja dublje razumevanje kako se došlo do tog rešenja.
4. Povratni odgovor studentu: Konačni rezultat, koji uključuje rešenje i objašnjenje, šalje se nazad korisniku ili studentu putem aplikacije.

Integracija sa ChatGPT-om igra ključnu ulogu u našem sistemu jer omogućava personalizovane odgovore na pitanja i pruža dublje razumevanje zadatka. Važno je napomenuti da kvalitet integracije direktno utiče na korisničko iskustvo i tačnost sistema. Osim toga, dobra integracija sa ChatGPT-om takođe može značajno unaprediti efikasnost sistema, omogućavajući brže i preciznije odgovore na veliki broj različitih upita, što dodatno povećava vrednost koju naša platforma pruža korisnicima. Takođe, kontinuirano unapređivanje integracije i praćenje povratnih informacija korisnika ključno je za održavanje visokog standarda učinkovitosti i zadovoljstva korisnika.

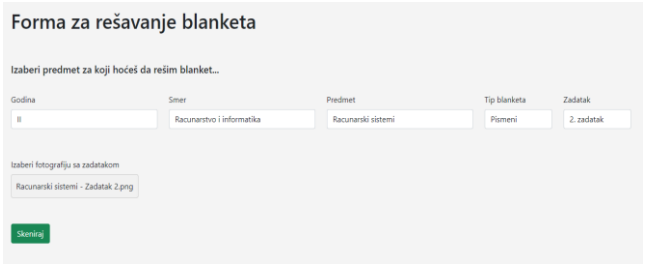
III. IMPLEMENTACIJA SISTEMA

Sistem koji je implementiran u ovom naučno-istraživačkom radu predstavlja aplikaciju razvijenu u .NET Core okruženju, sa dodatnom integracijom Python skripte koja koristi Tesseract OCR biblioteku [14]. Ova aplikacija je namenjena studentima sa ciljem da im olakša proces rešavanja zadataka sa slika, pružajući im precizna rešenja i objašnjenja uz pomoć ChatGPT modela. Evo detaljnijeg opisa ključnih koraka i opisa najznačajnijih funkcionalnosti ove aplikacije:

Konfiguracija studijske godine i smera: Prilikom pokretanja aplikacije, korisnicima se pruža mogućnost da odaberu svoju studijsku godinu (I, II, III, IV, V) i smer (npr. računarstvo i informatika, telekomunikacije, upravljanje sistemima). Ovi izbori personalizuju iskustvo korisnika i omogućavaju aplikaciji da pruži relevantne opcije za predmete i tip blanketa.

Odabir predmeta i tipa blanketa: Nakon što korisnici definišu svoju godinu i smer, sledeći korak je izbor konkretnog predmeta povezanog sa zadatkom. Lista dostupnih predmeta se dinamički generiše u zavisnosti od prethodnih izbora, pružajući korisnicima opciju da tačno specifikuju zadatke koje žele rešiti. Takođe, korisnici mogu izabrati tip blanketa (usmeni ili pismeni) kako bi dalje usmerili aplikaciju.

Unos rednog broja zadatka: Svaki zadatak ima svoj jedinstven redni broj, što omogućava precizno identifikovanje zadatka koji korisnik želi rešiti. Unos rednog broja je bitan za dobijanje tačnog rešenja. U nastavku je dat primer forme koja je vidljiva korisniku nakon pokretanja aplikacije:



Slika 1. Forma za rešavanje blanketa.

Učitavanje slike: Klikom na dugme "Izaberi sliku," korisnik može učitati sliku sa zadatkom sa svog računara. Ova slika će biti korišćena za izdvajanje teksta koji će biti analiziran. U nastavku je dat deo koda koji vrši učitavanje izabranog fajla, kao i svih svojstava unetih u okviru aplikacije od strane korisnika.

```
[HttpPost]
0 references
public async Task<ActionResult> Upload(IFormFile file, string yearId, string moduleId,
    string subjectId, string typeId, string taskId)
{
    if (file == null || file.Length == 0)
        return Json(new { status = "fail", message = "File is empty!" });

    ChosenBlanket.Year = yearId;
    ChosenBlanket.Module = moduleId;
    ChosenBlanket.Subject = subjectId;
    ChosenBlanket.Type = typeId;
    ChosenBlanket.Task = taskId;

    string fileName = $"{yearId}_{moduleId}_{subjectId}_{typeId}_{taskId}_{file.FileName}";

    string filePath = Path.Combine(PhotosPath, fileName);

    using (var fileStream = new FileStream(filePath, FileMode.Create))
    {
        await file.CopyToAsync(fileStream);
    }

    string text = ReadTextFromImage(PythonScriptPath, filePath);

    string fixedText = FixText(text);

    return Json(new { status = "success", message = fixedText });
}
```

Slika 2. Učitavanje fajla sa slikom teksta zadatka.

Izvršavanje OCR-a: Nakon što je slika učitana, aplikacija pozadinski poziva Python skriptu koja koristi.

Tesseract OCR biblioteku. Tesseract je odgovoran za prepoznavanje teksta sa slike i izdvajanje tog teksta kao podataka koji će dalje biti korišćeni [13]. U nastavku je dat deo koda koji poziva python skriptu u ovu svrhu:

```
1 reference
public string ReadTextFromImage(string scriptPath, string argument)
{
    ProcessStartInfo start = new ProcessStartInfo();
    start.FileName = "C:\\Users\\Stefan Djokic\\AppData\\Local\\Programs\\Python\\Python312\\python.exe";
    start.Arguments = $"\"{scriptPath}\" \"{argument}\"";
    start.UseShellExecute = false;
    start.CreateNoWindow = true;
    start.RedirectStandardOutput = true;
    start.RedirectStandardError = true;

    using (Process process = Process.Start(start))
    {
        using (StreamReader reader = process.StandardOutput)
        {
            string stderr = process.StandardError.ReadToEnd();
            string result = reader.ReadToEnd();

            return result;
        }
    }
}
```

Slika 3. Deo koda koji poziva python skriptu.

Dodatne napomene za preciznost: Korisnici mogu dodati dodatne napomene ili smernice za rešavanje zadatka kako bi osigurali tačna i relevantna rešenja od strane aplikacije. Ove napomene su od suštinskog značaja za dobijanje što preciznijih rezultata.

Korekcija teksta: Ukoliko postoji greška u prepoznavanju teksta sa slike, korisnik može izvršiti korekciju kako bi osigurao tačnost informacija koje će biti prosleđene ChatGPT modelu [12]. Tekst dobijen korišćenjem OCR-a se šalje ponovo ChatGPT-u kako bi ispravio sve sintaksne greške i vratio ga nazad korisniku u finalnom obliku nakon ispravke svih grešaka. Deo koda koji prati i implementira prethodno opisano ponašanje je dat u nastavku:

```
1 reference
public string FixText(string text)
{
    string question = "Piši na srpskom jeziku. Nađi i ispravi sve sintaksne greške u sledećem tekstu. " +
        "Ukoliko smatraš da postoji semantička greška ili da nedostaje neki deo teksta, pokušaj i to da ispraviš. " +
        "Odgovori samo sa ispravljenim tekstom. Tekst je: ->  ";

    string prompt = $"{question} {text}";

    var openAIAPI = new OpenAIAPI("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");

    var completions = openAIAPI.Completions.CreateCompletionAsync(
        prompt: prompt,
        model: "text-davinci-003",
        max_tokens: 3000,
        temperature: 0.5f
        );

    var completionsResult = completions.Result.Completions;

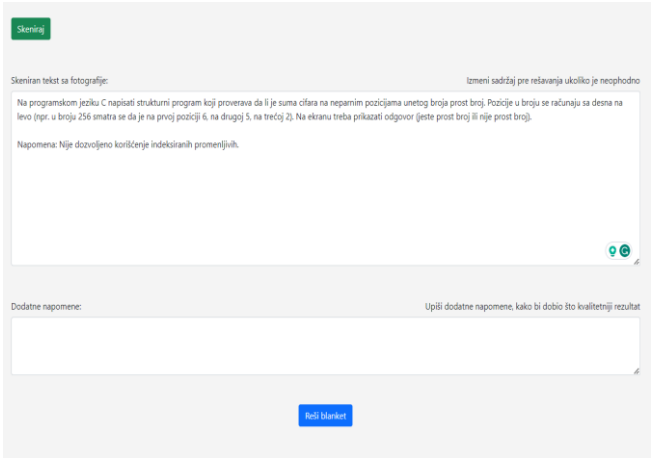
    string fixedText = string.Empty;

    if (completionsResult is not null && completionsResult.Count > 0)
    {
        fixedText = completionsResult[0].Text;
    }

    return fixedText;
}
```

Slika 4. Deo koda koji vrši ispravku teksta dobijenog putem OCR-a.

Korisniku se prikazuje tekst, kao što je na sledećoj slici. Korisnik može, po želji, navesti neke dodatne napomene koje mogu služiti kao smernica prilikom rešavanja zadatka.



Slika 5. Tekst dobijen putem OCR-a.

Upit ChatGPT-u za rešenje: Klikom na dugme "Reši blanket," aplikacija šalje upit ChatGPT modelu sa svim informacijama koje su unete. Ovo uključuje godinu, smer, predmet, tip blanketa, redni broj zadatka, prepoznat tekst sa slike i dodatne napomene. ChatGPT model generiše odgovor koji se prikazuje u odgovarajućem polju aplikacije. U nastavku je dat deo koda koji koristi integraciju sa ChatGPT - om i generiše odgovor odnosno rešenje skeniranog zadatka i objašnjenje postupka za rešavanje zadatka.

```
[HttpPost]
0 references
public ActionResult SolveProblem(QuestionModel question)
{
    string solve = "Reši ovaj zadatak, napisi kod u programskom jeziku koji je dat u zadatku i daj objašnjenje. Zadatak je: -> ";
    string additional = $"Dodatne napomene: {question}";

    var prompt = $"{solve} {question.OriginalText} {additional}";

    var openAIAPI = new OpenAIAPI("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");

    var completions = openAIAPI.Completions.CreateCompletionAsync(
        prompt: prompt,
        model: "text-davinci-003",
        max_tokens: 3000,
        temperature: 0.5f
        );

    var completionsResult = completions.Result.Completions;

    string solution = string.Empty;

    if (completionsResult is not null && completionsResult.Count > 0)
    {
        solution = completionsResult[0].Text;
    }

    BlanketModel blanketModel = new BlanketModel
    {
        Year = ChosenBlanket.Year,
        Module = ChosenBlanket.Module,
        Subject = ChosenBlanket.Subject,
        Type = ChosenBlanket.Type,
        Task = ChosenBlanket.Task,
        Solution = solution
    };

    _blanketService.AddBlanket(blanketModel);

    return Json(new { status = "success", message = solution });
}
```

Slika 6. Deo koda odgovoran za generisanje rešenje zadatka.

Uz pomoć integracije sa ChatGPT - om generiše se odgovor, odnosno kod koji je rešenje unetog zadatka, kao i objašnjenje koje prati implementaciju rešenja zadatka. U nastavku je dat primer rešenja zadatka, koji je dobijen na osnovu prethodno unetih informacija:

Rešenje zadatka

```
int main()
{
    int n, sum = 0;  // deklarisemo promenljive n i sum

    printf("Unesite broj: ");
    scanf("%d", &n);  // unosimo broj

    while (n != 0)  // petlja radi dok se broj ne svede na 0
    {
        int c = n % 10;  // c je cifra sa najmanjim znakom
        if (c % 2 != 0)  // proveravamo da li je cifra na neparnoj poziciji
            sum += c;  // ako jeste, dodajemo je na sumu
        n /= 10;  // umanjujemo broj za jedan znak
    }

    int i;
    for (i = 2; i < sum; i++)  // prolazimo kroz sve brojeve od 2 do sume
    {
        if (sum % i == 0)  // proveravamo da li je suma deljiva sa nekim brojem
            break;  // ako jeste, prekidamo petlju
    }
```

Rešenje zadatka

```
    int i;
    for (i = 2; i < sum; i++)  // prolazimo kroz sve brojeve od 2 do sume
    {
        if (sum % i == 0)  // proveravamo da li je suma deljiva sa nekim brojem
            break;  // ako jeste, prekidamo petlju
    }
    if (i == sum)  // ako je i jednako sumi, to znaci da nije deljiva sa nijednim brojem
        printf("Jeste prost broj\n");
    else  // ako nije, znaci da je deljiva sa nekim brojem
        printf("Nije prost broj\n");

    return 0;
}

Ovaj program proverava da li je suma cifara na neparnim pozicijama unetog broja
prost broj. Prvo se unosi broj, a zatim se u petlji računa suma cifara na neparnim
pozicijama. Nakon toga, proverava se da li je suma deljiva sa nekim brojem od 2
do same sume. Ako nije, znači da je prost broj. Ako jeste, znači da nije prost broj.
Na kraju se na ekranu ispisuje odgovor.
*/
```

Slika 7. Rešenje zadatka.

Čuvanje zadataka u bazi podataka: Svaki zadatak može biti sačuvan u bazi podataka kako bi se omogućila kasnija referenca i pristup prethodno rešenim zadacima. U nastavku je dat deo koda koji dodaje novi blanket u bazu. Blanket se dodaje sa svojstvima koje je korisnik prethodno izabrao u okviru aplikacije prilikom dodavanja novog blanketa.

```
public void AddBlanket(BlanketModel blanketModel)
{
    Blanket blanket = new Blanket
    {
        Year = blanketModel.Year,
        Module = blanketModel.Module,
        Subject = blanketModel.Subject,
        Type = blanketModel.Type
    };

    _context.Blankets.Add(blanket);
    _context.SaveChanges();

    BlanketTask blanketTask = new BlanketTask
    {
        BlanketId = blanket.Id,
        Solution = blanketModel.Solution,
        Task = blanketModel.Task
    };

    _context.BlanketTasks.Add(blanketTask);
    _context.SaveChanges();
}
```

Slika 8. Deo koda za dodavanje novog blanketa u bazu podataka.

Aplikacija vrši proveru da li postoji tekst sa istim kombinacijama godine, smera, predmeta i rednog broja zadatka kako bi sprečila dupliciranje zadataka. U nastavku je dat deo koda koji proverava da li blanket sa unetim svojstvima već postoji u okviru baze podataka.

```
public BlanketModel ReturnIfExist(BlanketModel model)
{
    var existBlanket = _context.Blankets.FirstOrDefault(x => x.Year.ToLower() == model.Year.ToLower() &&
                                                             x.Module.ToLower() == model.Module.ToLower() &&
                                                             x.Subject.ToLower() == model.Subject.ToLower() &&
                                                             x.Type.ToLower() == model.Type.ToLower());

    if (existBlanket == null) return null;

    var existBlanketTask = _context.BlanketTasks.FirstOrDefault(x => x.BlanketId == existBlanket.Id && x.Task.ToLower() == model.Task.ToLower());

    if (existBlanketTask == null) return null;

    return new BlanketModel
    {
        Id = existBlanket.Id,
        Solution = existBlanketTask.Solution
    };
}
```

Slika 9. Deo koda koji vrši proveru da li blanket sa unetim svojstvima već postoji u bazi.

Ovaj sistem omogućava studentima brzo i efikasno rešavanje zadataka sa slika, dobijanje tačnih rešenja od strane ChatGPT-a i praćenje prethodno rešenih zadataka radi kasnijeg pristupa. Cilj sistema je olakšati proces učenja i rešavanja zadataka, pružajući precizna rešenja i objašnjenja na zahtev korisnika.

IV. Prednosti i nedostaci sistema

A. Prednosti sistema

Prednosti ovako implementiranog sistema su mnogobrojni, a neki od njih jesu i sledeći:

1. Brz pristup rešenjima: Opisan sistem omogućava studentima brz pristup rešenjima zadatka. Ovo je posebno korisno za učenike koji žele da provere svoje radove ili steknu bolje razumevanje konkretnih problema.
2. Objašnjenje zadataka: Pored rešenja, sistem pruža i objašnjenja kako se do tih rešenja došlo. Ovo je od suštinskog značaja za razvoj razumevanja koncepata i algoritama, čime se studentima pomaže da postanu bolji prilikom rešavanja problema.
3. Personalizacija: Integracija sa ChatGPT-om omogućava personalizovane odgovore na pitanja i zadatke. Svaki student može dobiti odgovore prilagođene njihovom specifičnim potrebama i nivou razumevanja.
4. Dostupnost: Sistem je dostupan 24/7, što znači da studenti mogu dobiti pomoć u bilo koje vreme, bez obzira na radno vreme ili dostupnost profesora.
5. Poboljšano razumevanje: Kroz objašnjenja i interakciju sa ChatGPT-om, studenti mogu razviti dublje razumevanje gradiva, što može doprineti njihovom dugoročnom uspehu u učenju.

B. Potencijalni problemi *i rešenja*

Prilikom korišćenja prethodno opisanih tehnologija mogu se javiti neki od potencijalnih problema. Neke od ključnih tačaka koje treba razmotriti uključuju:

1. Tačnost OCR - a: kako bi se obezbedila visoka tačnost prepoznavanja teksta sa slike, moraju se koristiti visoko kvalitetni OCR algoritmi i tehnike. Greške u ovom koraku mogu dovesti do netačnih rešenja i objašnjenja.
2. Netačni odgovori ChatGPT-a: ChatGPT nije savršen i može generiati netačne odgovore. Važno je imati mehanizme za verifikaciju i korekciju odgovora kako bi se osigurala tačnost [10, 11].
3. Bezbednost podataka: Zaštita privatnosti i bezbednost podataka korisnika je od suštinskog značaja. Implementiramo odgovarajuće mere kako bismo osigurali da osetljivi podaci ne budu ugroženi tokom prenosa i obrade.
4. Vreme odgovora: Brzina obrade i odgovora je važna, posebno u obrazovom kontekstu. Treba razmotriti optimizacije kako bi se osiguralo da studenti dobiju brze odgovore.
5. Personalizacija: Prilagođenje odgovora ChatGPT-a individualnim potrebama i nivou razumevanja svakog studenta može biti izazovna, ali klučna za uspeh sistema.
6. Zavisnost o tehnologiji: Sistem zavisi od tehnologija kao što su internet konekcija , OCR i ChatGPT, što može dovesti do problema ako ove tehnologije nisu dostupne.

C. Mogućnosti za unapređenje

Kod sistema implementiranog na ovaj način, i pored mnogih prednosti koje on pruža, svakako ima prostora i za buduća unapređenja, a neka od njih su sledeća:

1. Poboljšanje OCR tačnosti: Unapređenje tačnosti prepoznavanja teksta sa slike kroz korišćenje naprednijih OCR algoritama i tehnika za uklanjanje šuma sa slika.
2. Obogaćivanje ChatGPT-a: Integracija sa ChatGPT-om može se poboljšati dodatnim obukama modela kako bi generisao preciznije i informativnije odgovore.
3. Višejezična podrška: Proširenje sistema da podrži više jezika kako bi se omogućila pomoć studentima iz različitih kulturnih i jezičkih konteksta.
4. Automatizacija kvaliteta slika: Implementacija alata koji automatski poboljšavaju kvalitet slika pre nego što budu poslate na OCR analizu.
5. Razvoj aplikacije za mobilne uređaje: Kreiranje mobilne aplikacije omogućava korisnicima lakši pristup sistemu putem pametnih uređaja.

Uvođenje ovih unapređenja može pomoći da se prevaziđu nedostaci sistema i pruži bolje iskustvo učenja studentima, čineći učenje i rešavanje zadataka efikasnijim i korisnijim.

V. ZAKLJUČAK

Implementacija sistema za rešavanje zadataka pomoću Optical Character Recognition (OCR) tehnologije i integracije sa ChatGPT predstavlja inovativan pristup rešavanju problema sa kojima se studenti suočavaju prilikom učenja i rešavanju zadataka. Ovaj sistem donosi brojne prednosti u smislu brzog pristupa rešenjima, personalizovanih odgovora i unapređenog razumevanja koncepata. Kroz analizu prednosti, nedostataka i mogućnosti za unapređenje, možemo izvući nekoliko ključnih zaključaka.

Sistem pruža brz i jednostavan pristup rešenjima za zadatke, čime se studentima štedi vreme i olakšava učenje. Personalizovani odgovori od strane ChatGPT-a omogućavaju studentima bolje razumevanje koncepata i pomažu u razvoju veština rešavanja problema. Ovo je posebno korisno u obrazovnim okruženjima gde su resursi ograničeni.

Međutim, kao i svaki sistem, postoje i određeni nedostaci koje se moraju uzeti u obzir. Tačnost OCR tehnologije i ChatGPT-a mogu varirati, što može dovesti do netačnih rešenja i odgovora. Osim toga, zavisnost o tehnologiji može predstavljati izazov u situacijama gde tehnologija nije dostupna.

Unapređenje sistema, kao što su poboljšanje tačnosti OCR-a, obogaćivanje ChatGPT-a i višejezična podrška, pružaju put ka stvaranju efikasnijeg i korisnijeg sistema za učenje i rešavanje zadataka.

U celini, ovaj sistem predstavlja korak napred u digitalizaciji obrazovanja i pruža studentima dodatne resurse za uspešno učenje i rešavanje zadataka. Kroz dalji razvoj i unapređenja, ovakvi sistemi mogu igrati značajnu ulogu u podršci obrazovnom procesu i pomoći studentima da postignu bolje rezultate u svom učenju.

LITERATURA

[1] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing Machines," arXiv preprint arXiv:1410.5401, 2014.
[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
[3] X. Chen and A. Gupta, "Webly Supervised Learning of Convolutional Networks," in Proceedings of the IEEE International Conference on Computer Vision, 2015.
[4] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," Neural Computation, vol. 1, no. 2, pp. 270-280, 1989.
[5] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016.
[6] M. Mozafari, "Understanding, Optimizing, and Deploying Deep Learning," ACM SIGMOD Record, vol. 46, no. 1, pp. 50-60, 2017.

[7] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.

[8] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," OpenAI, 2018.

[11] S. G. Reed and N. de Freitas, "Neural Programmer-Interpreters," arXiv preprint arXiv:1511.06279, 2015.

[12] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems 30, 2017.

[13] Ray Smith, "An Overview of the Tesseract OCR Engine," in Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[14] D. Zamparelli, "Tesseract Unleashed: A Tribute to Open Source OCR," in Proceedings of the First International Workshop on Multilingual OCR, 2011.

Verifikacija RTL modela rutera sa jednim ulaznim i tri izlazna kanala UVM metodologijom

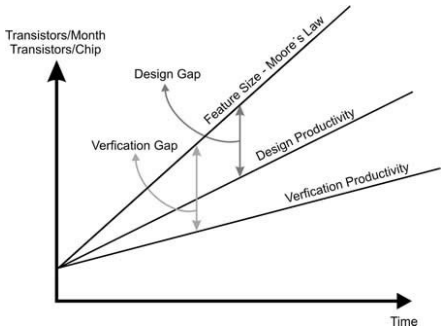
Marko Antonijević

Sadržaj - Cilj ovog rada je da se predstavi metodologija verifikacije RTL modela rutera sa jednim ulazom i tri izlaza korišćenjem UVM metodologije i jezika za opis hardvera *Sysetmverilog*. Objašnjen je rad RTL modela rutera, na osnovu specifikacije je kreirano okruženje za verifikaciju. Kreiran je *testplan*, određeni scenariji testova i promenljive za praćenje pokrivenosti testiranja. Na kraju su predstavljeni rezultati testiranja.

I. UVOD

Izražen napredak tehnologije poluprovodnika dovodi do znatnog usložnjavanja *SoC* dizajna. Ovi kompleksni dizajni koji imaju puno primena unutar različitih industrija integrišu mnoštvo hardverskih i softverskih komponenti na jedan čip čime postižu izvanrednu moć računanja i svestranost. Ovu povećanu kompleksnost prati i problem osiguravanja tačne funkcionalnosti.

Iz ovih razloga se proces verifikacije izdvaja kao najvažniji korak u dizajnu *SoC*-ja, jer je usmeren na rano uočavanje potencijalnih grešaka u dizajnu i garantovanje pouzdanosti sistema. Verifikacija testira dizajn sistema i time osigurava da su ključne funkcionalnosti dizajna tačno implementirane pre nego što se krene u proces proizvodnje u velikim serijama. U proseku, verifikacija zauzima preko 70 % razvoja *SoC*-ja. što je čini najvažnijim procesom u dizajnu i razvoju. Mogućnosti koje otvara sve veći broj tranzistora koje je moguće proizvesti na parčetu silicijuma rastu u skladu sa Murovom hipotezom (slika 1).



Slika 1. Rast mogućnosti proizvodnje, dizajna i verifikacije sa vremenom.

M. Antonijević student doktorskih studija, studijski program Elektronika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: markoanton@elfak.ni.ac.rs.

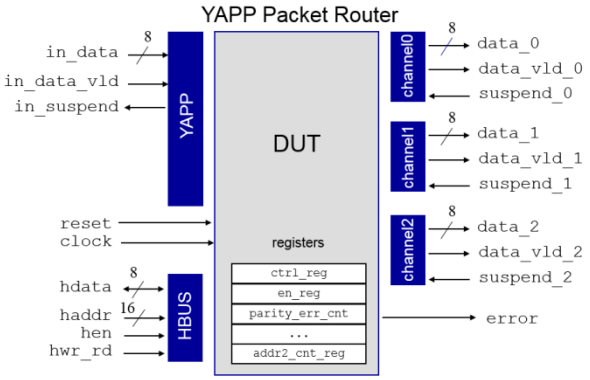
i prevazilaze ono što je moguće dizajnirati (*design gap*), dok mogućnosti dizajna prevazilaze ono što je moguće verifikovati sa visokom pouzdanošću (*verification* gap) [1]. Verifikacija predstavlja usko grlo razvoja *SoC*-ja upravo iz razloga što je sve teže pokriti sve mogućnosti ulaza testiranjem i predvideti sve scenarije koje je potrebno testirati.

II. SPECIFIKACIJA DUT-A

Ruteri su važne komponente digitalnih sistema. Oni su odgovorni za povezivanje različitih komponenti digitalnog sistema tako što omogućavaju efikasan prenos podataka. Njihova tačna implementacija izuzetno je važna jer omogućava optimizaciju performansi i funkcionalnosti digitalnih sistema i osiguravaju da se podaci prenose tačno i visokom brzinom.

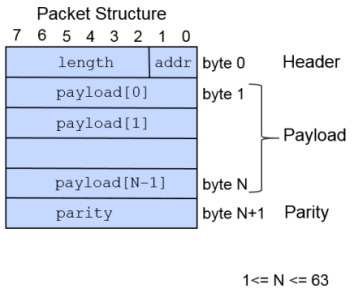
RTL modela rutera (*YAPP*) koji je tema ovog rada radi na sledeći način (slika 2). *YAPP router* prima pakete podataka na *YAPP* interfejsu. Ove podatke sa ulaza ruter šalje na jedan od tri izlazna kanala (*Channel* 0, *Channel* 1 ili *Channel* 2) zavisno od adrese koju ima ulazni paket. Rad rutera se podešava konfiguracijom odgovarajućih registara kojima se pristupa preko *HBUS* interfejsa.

Na slici 2. prikazan je dijagram YAPP *router*-a.



Slika 2. Blok šema rutera sa interfejsima rutera i ulaznim i izlaznim signalima.

Paket koji se šalje na ruter je sekvenca bajtova (slika 3). Prvi bajt ove sekvence se sastoji od 2-bitnog polja adrese i 6-bitnog polja dužine.



Slika 3. Struktura paketa podatka koji se šalju na ruter.

Polje adrese se koristi da se izabere na koji od mogućih izlaza će paket biti preusmeren, pri čemu je adresa 3 nedozvoljena. *Length* polje određuje dužinu odnosno broj bajtova podataka (*payload*). Ostali bajtovi čine *payload* odnosno bajtove podataka pri čemu je poslednji bajt - bajt parnosti. Paket može imati *payload* minimalne dužine 1 a maksimalne dužine 63 bajta. Bajt parnosti je bajt *bitwise* parnosti koje se računa nad *header*-om i bajtovima *data*.

RTL model rutera je napravljen tako da treba da odbaci pakete koji se šalju na neispravnu adresu (3) i da u slučaju netačno izračunate parnosti podigne *error* signal na logičku jedinicu. Svi paketi koji se šalju na ulaz rutera, šalju se preko *YAPP* interfejsa dok za svaki izlaz postoji odgovarajući *Channel* interfejs. Ruter sadrži unutrašnje registre koji čuvaju informacije vezane za konfiguraciju (slika 4). Ovim registrima se pristupa preko *HBUS* interfejsa

address	register	reset	field	field name	policy	description
0x1000	ctrl_reg	0x3f	5:0	maxpktsize	RW	Maximum packet length
			7:6		RW	Unused
0x1001	en_reg	0x01	0	router_en	RW	Router enable
			1	parity_err_cnt_en	RW	Parity error count enable
			2	oversized_pkt_cnt_en	RW	Length error count enable
			3	[reserved]	RW	Not implemented
			4	addr0_cnt_en	RW	Address 0 packet count enable
			5	addr1_cnt_en	RW	Address 1 packet count enable
			6	addr2_cnt_en	RW	Address 2 packet count enable
			7	addr3_cnt_en	RW	Address 3 packet count enable
0x1004	parity_err_cnt_reg	0x00	7:0		RO	Packet parity error count
0x1005	oversized_pkt_cnt_reg	0x00	7:0		RO	Packet length error count
0x1006	addr3_cnt_reg	0x00	7:0		RO	Address 3 packet count
0x1009	addr0_cnt_reg	0x00	7:0		RO	Address 0 packet count
0x100a	addr1_cnt_reg	0x00	7:0		RO	Address 1 packet count
0x100b	addr2_cnt_reg	0x00	7:0		RO	Address 2 packet count

Slika. 4. Slika registara rutera iz specifikacije sa njihovim adresama, veličinom, i pravima pristupa i opisom.

Ukoliko je dužina ulaznog paketa veća od vrednposti *maxpktsize* polja u *ctrl_reg* registru ruter će zanemariti čitav paket.

Router_en polje *en_reg* kontroliše uključenje i isključenje rutera. Isključen *router* zanemaruje sve pakete. Brojači *router*-a se aktiviraju individualnim bitovima en_reg.

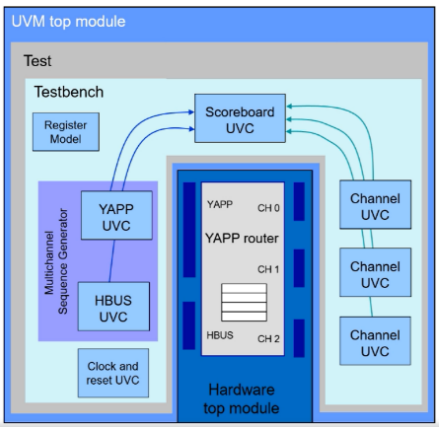
Brojači rutera su definisani na sledeći način:

- *parity_err_cnt_reg* se inkrementira kad se detektuje bit sa netačnom parnošću.
- *oversized_pkt_cnt_reg* se inkrementira kada se primi paket sa veličinom većom od maxpktsize.
- *addr3_cnt_reg* inkrementira se kada se paket sa nedozvoljenom adresom 3 primi.
- *addr0_cnt_reg* inkrementira se kada se paket sa adresom 0 primi, na isti način rade *addr1_cnt_reg I addr2_cnt_reg*.

III. Okruženje za verifikaciju

Okruženje za verifikaciju je pisano u programskom jeziku SystemVerilog, pri čemu je korišćena UVM metodologija. Korišćenje UVM metodolgoije sa *SystemVerilog*-om omogućava efikasnu i efektivnu verifikaciju i pruža određene prednosti. Te prednosti se ogledaju u Modularnosti, *Constrained-random* testiranju i *coverage-driven* verifikaciji kao i korišćenju *checkera* [2]. UVM metodologija omogućava korišćenje modularne arhitekture i kreiranje komponenti kao što su *driver*, *monitor*, *Scoreboard* i slično koje se lako se mogu ponovo koristiti, kao i odvajanje pobude od samog okruženja.

Okruženje za verifikaciju ovog RTL modela sastoji se od 7 UVC-jeva koji su povezani sa RTL modelom i regmodela koji služi za verifikaciju registara RTL modela (slika 5). UVC je modularni blok u UVM metodologiji koji verifikuje određeni deo *DUT*-a. U slučaju ovog projekta svaki UVC je vezan sa po jednim interfejsom i on verifikuje protokol za komunikaciju tog interfejsa. Osim UVC interfejsa instanciran je i *Scoreboard* UVC koji služi za poređenje podataka sa ulaznih kanala i sa izlaza koji služi da proveri da li *RTL* model zaista odbacuje pakete koji nisu ispravni. Takođe radni takt se daje UVC-ju preko posebnog UVC-ja. Povezivanje UVC-jeva sa RTL modelom prikazano je na slici ispod.



Slika 5. Šema povezivanja UVC-jeva sa interfejsima rutera.

Ovi UVC-jevi se instanciraju i konfigurišu unutar *testbench* klase. Ova klasa je instancirana unutar test klase koja kontroliše pobudu koji ide na svaki UVC.

YAPP UVC je povezan na ulazni kanal RTL modela. On šalje signale na YAPP interfejs prema YAPP protokolu korišćenjem *driver* klase. *Monitor* klasa skuplja *coverage* i osigurava da je slanje signala u skladu sa protokolom (korišćenjem *checker*-a). Sve transakcije koje su poslate na YAPP UVC i snimljene u *monitor*-u šalje i prema *Scoreboard* UVC-ju gde će se porediti sa paketima sa izlaza (*Channel UVC*-jeva).

Channel UVC je povezan na izlazne kanale RTL modela. Ovaj UVC šalje signale na *Channel* interfejs prema protokolu, monitorše transakcije koje se odvijaju preko *Channel* interfejsa, proverava da li su po protokolu za izlazne signale i šalje izlazne transakcije u *Scoreboard* na poređenje. S obzirom da postoje tri izlazna kanala, za svaku adresu po jedan, za svaki od izlaznih kanala postoji poseban *Channel* UVC.

HBUS UVC služi za programiranje registara DUT-a. Ovaj UVC monitoriše transakcije koje se šalju na *HBUS* interfejsu i kontroliše da li su u skladu sa *HBUS* protokolom. Pre nego što se pošalju YAPP paket transakcije na DUT, potrebno je da se u vrednosti njegovih registara upišu vrednosti koje utiču na ponašanje RTL modela. Sinhronizaciju upisa na *HBUS* i YAPP UVC obavlja *Multichannel* sekvencer.

Scoreboard UVC je komponenta koja ima ulogu da osigura da DUT ispravno radi tako što proverava da li su se svi paketi koji su očekivani na izlazu zaista i pojavili na izlazu. *Scoreboard* UVC je povezan sa jedne strane sa YAPP UVC-jem i *HBUS* UVC-jem na ulazu a sa druge sa *Channel* UVC-jevima na izlazu i podeljen je na dva klase. Prva klasa je *router reference* koja proverava ispravnost paketa koji mu je prosleđen sa YAPP UVC-ja. Ova klasa čuva vrednosti upisane u registre modela i proverava da li paket treba da se pojavi na izlazu (ako je RTL model uključen i ako je dužina paketa manja od vrednosti upisane u *maxpktsize* registar). Druga – *router_scoreboard* povezana je sa *router_reference* klasom. U njoj se vrši poređenje ispravnih paketa koje dobija sa *router_reference* onima koji se prosleđuju sa *Channel* UVC-jeva. Ovo osigurava da *Scoreboard* zaista odbacuje neispravne pakete kao i da se svaki ispravni paket zaista prosledio na zadatu adresu.

Registri se verifikuju korišćenjem *regmodel*-a. U *testbench*-u se pravi model registara gde su njihove ključne karakteristike kao što su veličina, reset vrednosti i pristup (*Read Only, Write Only* ili *Read Write*). Registiri u modelu su duplikati registara iz DUT-a i svaki registar u *regmodel*-u odgovara stvarnom registru u dizajnu.

Clock_and_reset UVC služi za resetovanje na vrednosti registara i ostalih signala na početne (*default*) i generisanje *clk* signala za izvršavanje. Samo generisanje *clk* signala je odvojeno unutar posebnog modula kako bi se postigla modularnost i omogućilo njegovo lakše ponovno korišćenje.

IV. TESTIRANJE RTL-MODELA

Testiranje modela se radi na osnovu *testplan*-a. To je dokument koji se piše na početku procesa verifikacije, unutar kog su navedene ključne tačke RTL modela koje je potrebno proveriti, testovi koji to proveravaju i metrika pokrivenosti testiranja. On olakšava identifikaciju potencijalnih problema u ranoj fazi dizajna, čime se smanjuje šansa da se pojavi greška u konačnom proizvodu. Na osnovu plana se biraju inženjeri koji će obavljati posao, alati i okruženje koje će koristiti da bi se verifikacija uradila na najefikasniji mogući način. Takođe, na osnovu plana može da se proceni koje će vreme biti potrebno za verifikaciju.

Ključne karakteristike YAPP RTL modela koje je potrebno proveriti su:

- Prosleđivanje ulaznih paketa na izlaze i odbacivanje paketa koji su poslati na adresu 3.

Test koji proverava ovu stavku je *all_adress_test* koji ima sledeći scenario:

Prvo se na *HBUS* UVC-ju programiraju registri rutera tako da se uključi ruter i tako da se upiše maksimalna vrednost u registar za maksimalnu dužinu paketa. Zatim se u deset iteracija šalju randomizovane transakcija redom na adrese 0, 1, 2, 3. Ove transakcije bi trebalo da se pojave na odgovarajućim *Channel* interfejsima osim transakcija koje su poslate na adresu 3

- Provera funkcionalnosti *max_pkt_size* registra (odbacivanje paketa sa kraćom dužinom)

Test koji proverava ovu stavku *je max_pkt_size* test i ima sledeći scenario:

U max_pkt_size registar se prvo upisuje randomizovana vrednost u rasponu od 2 do 62 preko *HBUS* UVC-ja. Zatim se na YAPP UVC šalje 5 transakcija sa manjom dužinom od one upisane u registar i 5 sa većom dužinom. Prvih 5 transakcija treba da budu uspešno prosleđene na izlaze dok će se drugih 5 odbaciti.

- Proveravanje parnosti paketa (podizanje *error* signala za paketa sa pogrešnom parnošću)

Test koji to proverava je *YAPP_parity_test* i ima sledeći scenario:

Ovaj test prvo upisuje maksimalnu vrednost u *max_pkt* registar preko *HBUS* UVC-ja kako bi se izbeglo odbacivanje zbog dužine transakcije. Zatim se u pet iteracija naizmenično šalju transakcije sa tačnom i netačnom parnošću. Cilj je da se proveri da se nakon svakog pojavljivanja paketa netačne parnosti na ulazu podigne *error* signal i da se nakon toga simulacija nastavi normalno, odnosno da se paketi pojavljuju na izlazu.

- Provera reset vrednosti testa

Potrebno je proveriti da li reset vrednosti registara odgovaraju onima koje su navedene u specifikaciji.

Test koji ovo proverava je *register_reset_test*

- Provera prava pristupa registrima

Potrebno je proveriti da li svi registri zaista imaju prava za čitanje i upis kao što je određeno specifikacijom.

Test koji ovo proverava je *reg_access_test.*

- Provera funkcionalnosti *cnt* registara

Upis u odgovarajuće vrednosti u *en_reg* omogućava rad *parrity_err_cnt_reg, oversized_pk_cnt_reg*, *addr_cnt_reg* registara. U ove registre RTL model upisuje broj paketa koji su odbačeni zbog parnosti, dužine i broj paketa koji su poslati na adrese. Potrebno je proveriti da se tačne vrednosti zaista upisuju u registre kao i da *en* registar zaista kontroliše da li se upisuju vrednosti u registar.

Na osnovu stavki koje je potrebno ispitati određene su promenljive za metriku pokrivenosti testiranja (*coverage*) kago bi se osiguralo da su različite vrednosti pogođene. Dužina paketa je podeljena u 5 *bin*-ova. Ekstremne vrednosti za minimum i maksimum dužine paketa imaju posebne *bin*-ove (Tabela I), dok su ostale vrednosti podeljene u tri grupe. Promenljive za adresu su pokrivene sa dva *bin*-a za legalne i nelegalne adrese. Na kraju je promenljiva za parnost pokrivena sa *bin*-ovima za tačnu i netačnu parnost. Na kraju se radi *cross* svih gore navedenih vrednosti.

TABELA I
NAZIV I VREDNOSTI ODABRANIH BINOVA

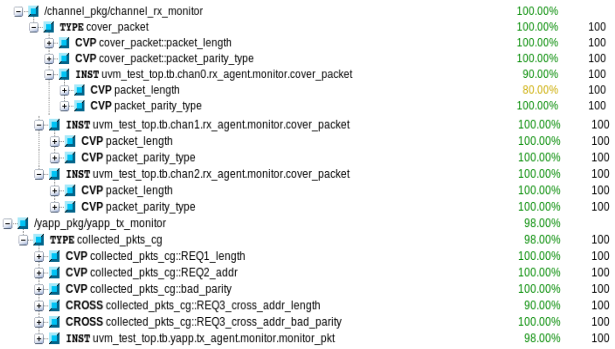
Naziv bina	Vrednosti koje obuhvata
Dužina paketa	
Min	1
Small	2-10
Medium	11-24
Large	25-62
Max	63
Adresa	
Legalna	0-2
Nelegalna	3
Parnost	
Bad	Tačna parnsot
Good	Netačna parnost

Na sličan način odabrani su *bin*-ovi za pokrivenost registara. Adrese registara pokrivene su tako što e*nable* registar i max_pkt_size registar imaju posebne binove, dok ostali registiri imaju zajednički bin. Takođe je pokriven je tip transakcije sa *bin*-om za upis i *bin*-om za čitanje.

V. REZULTATI TESTIRANJA

Za pokretanje testova korišćen je Mentor *QuestaSim* simulator. *QuestaSim* ima podršku za UVM i za *SystemVerilog*.

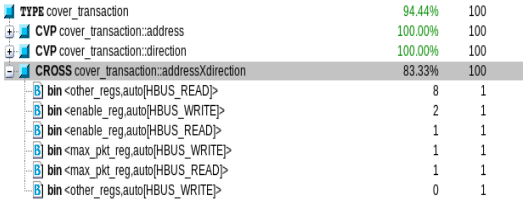
Simulator takođe omogućava prikaz signala na *waveform*-u i skupljanje *coverage*. S obzirom da *QuestaSim* nema mogućnost skupljanja kumulativnog coverage za regresiju: *all_adress_test, max_pkt_size* i *YAPP_parity* test su objedinjeni u jedan test. Na slici 6. su prikazani rezultati *coverage* za ovaj objedinjeni test.



Slika 6. Rezultati pokrivenosti na *YAPP* i *Channel* UVC-jevima.

Sa slike se vidi da su na ulazu (*yapp*) pogođeni svi *bin*-ovi za dužinu, sve moguće vrednosti parnosti, sve *cross* vrednosti adrese i parnosti, dok nisu pogođene sve *cross* vrednosti adrese i dužine. Razlog je što je ukupan broj transackija bio 300, što nije bilo dovoljno da se pogode sve moguće kombinacije. Za očekivati je da bi sa većim brojem transakcija ili puštanjem testa u regresiji bile pogođene sve kombinacije. S obzirom da se transakcije sa ulaza prosleđuju na 3 izlaza ukoliko im je adresa dozvoljena i dužina manja od vrednosti *max_pkt_size* registra, rezultait na izalzu su očekivano lošiji. Tako na izalzu *Channel0* nisu pogođene sve vrednosti dužine. Sa većim brojem puštanja testova u regresiji, za očekivati je da i ovi *bin*-ovi budu pogođeni.

Na slici 7. je prikazana je pokrivenost registara nakon pokretanja *register_function* testa. Sa slike se uočava da su pokriveni svi definisani *bin*-ovi osim cross *bin*-a za upis u registre koji nisu *max_pkt_size* i *router_en*. Ovo je iz razloga što iz *count* registri služe samo za čitanje vrednosti koje upisuje sam RTL model i u njih nema upisa, prema scenariju testa.



Slika 7. Rezultati pokrivenosti na *HBUS* UVC-ju.

Ovi rezultati pokazuju visoku pokrivenost mogućih kombinacija testiranja, što uz činjenicu da nema *uvm_error* ispisa u *log* fajlovima znači da je potvrđen ispravan rad rutera.

VI. ZAKLJUČAK

Ovaj rad predstavio je metodologiju verifikacije rutera korišćenjem UVM biblioteke i programskog jezika SystemVerilog. Analizom ključnih stavki specifikacije napravljeno je okruženje za verifikaciju i *testplan* po kome je testiranje vršeno. Dobijeni rezultati testiranja pokazuju visoku pokrivenost mogućih kombinacija. Oni se mogu

dodatno poboljšati korišćenjem alata koji omogućavaju sakupljanje metrike za čitavu regresiju (*vManager*) što bi bio dalji pravac istraživanja. Korišćenje modularnih komponenti koje UVM metodologija pruža, omogućava da se u slučaju dodavanja neke funkcionalnosti menja samo komponenta na koju utiče nova funkcionalnost dok ostatak okruženja ostaje isti. Na primer u slučaju da se menja protokol koji se koristi za komunikaciju na ulazu i izlazu rutera. To znači da ovaj model može da se koristi kao osnova za kreiranje kompleksnijih modela rutera koji se u praksi koriste čime se štedi na vremenu potrebnom za razvoj okruženja za verifikaciju.

LITERATURA

[1] A. Molina and O. Cadenas, "Functional verification: approaches and challenges," Latin American applied research, vol. 37, no. 1, pp. 65-69, 2007

[2] C Spear, G Tumbush, "SystemVerilog for Verification: A Guide to Learning the Testbench Language Features" 2012

[3] A. Smith and B. Johnson, "UVM Essentials: A Practical Approach to UVM for System Verilog Verification" Piscataway, NJ: IEEE, 2016.

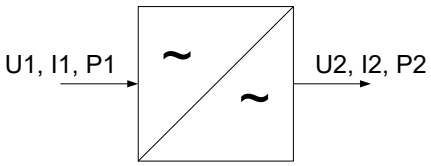
Praktična realizacija naizmeničnog pretvarača sa trijakom primenom integrisanog kola TCA785

Ognjen Bunjac, Vid Vuković

Sadržaj - U ovom radu prezentovana je praktična realizacija naizmeničnog pretvarača sa trijakom za čije upravljanje se koristi integrisano kolo TCA785 proizvođača SIEMENS. Tehnika upravljanja koja je primenjena je jednostrana fazna regulacija. Nakon hardverske realizacije elektronskog kola pretvarača, pretvarač je testiran pri omskom i omsko induktivnom opterećenju i rezultati testiranja prikazani su u radu.

I. NAIZMENIČNI PRETVARAČ SA TRIJAKOM I JEDNOSTRANOM FAZNOM REGULACIJOM

Naizmenični pretvarači sa faznom regulacijom su uređaji energetske elektronike koji služe za pretvaranje naizmeničnih napona i struja jednog nivoa u naizmenični napon i struju drugog nivoa [1]. Pri tome mislimo na efektivne vrednosti naizmeničnih veličina. Opšti simbol naizmeničnog pretvarača prikazan je na slici 1.

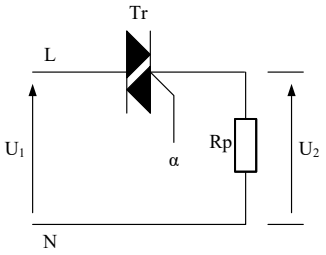


Slika 1. Opšti simbol naizmeničnog pretvarača [1].

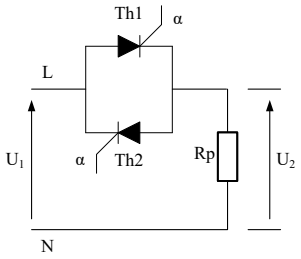
Drugi naziv za ovu vrstu uređaja je AC/AC pretvarači ili AC/AC konvertori. Naizmenični pretvarači rade na principu prekidanja i podešavanja naizmeničnog napona i struje, krajnji cilj je regulacija električne energije (snage) koju izvor predaje prijemniku (potrošaču).

Kod naizmeničnih pretvarača sa jednostranom faznom regulacijom koriste se trijaci za opterećenja manje snage i antiparalelna veza tiristora za uređaje većih snaga. Na slici 2 prikazana je principijelna šema naizmeničnih pretvarača sa jednostranom faznom regulacijom i trijakom, dok je na slici 3 prikazana principijelna šema naizmeničnog pretvarača sa jednostranom faznom regulacijom i antiparalelnom vezom tiristora.

O. Bunjac i V. Vuković su učenici Elektrotehničke škole „Mihajlo Pupin“, Novi Sad, Futoška 17, smer elektrotehničar procesnog upravljanja, E-mail: bunjaco05@gmail.com



Slika 2. Principijelna šema naizmeničnog pretvarača sa jednostranom faznom regulacijom i trijakom [1].



Slika 3. Principijelna šema naizmeničnog pretvarača sa jednostranom faznom regulacijom i antiparalelnom vezom tiristora [1].

Ova tehnika upravljanja naziva se tehnika jednostrane fazne regulacije zato što se ugao uključenja poluprovodničke komponente podešava samo sa jedne strane poluperiode naizmeničnog napona. Tom prilikom potrebno je naglasiti da upravljačko integrisano kolo (TCA780, TCA785, U208B ili neko slično) radi na principu detektovanja prolaska napona izvora napajanja kroz nulu, i od tog trenutka meri vreme i određuje trenutak kada će proslediti pobudni impuls na gejt komponente.

Efektivna vrednost napona na potrošaču zavisi od ugla uključenja komponente i može se odrediti pomoću sledećeg obrazca (1) [1].

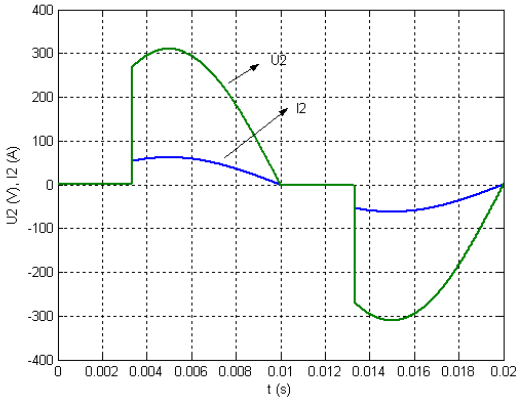
$$U_2 = U_1\sqrt{\frac{1}{\pi}\left(\pi - \alpha + \frac{\sin 2\alpha}{2}\right)} \tag{1}$$

Pri čemu je: U_1-efektivna vrednost ulaznog napona pretvarača (napona električne mreže, izvora napajanja), U_2-je efektivna vrednost napona na potrošaču a $\boldsymbol{\alpha}$-je ugao uključenja komponente.

Ulazni napon U_1 određuje električna mreža na koju je pretvarač priključen i tokom rada ulazni napon je

konstantne efektivne vrednosti i u jednofaznom sistemu iznosi U_1=230 V. Na osnovu izraza (1), vidi se da efektivna vrednost izlaznog napona U_2 zavisi od vrednosti ugla uključenja trijaka. Ta zavisnost je složena trigonometrijska funkcija definisana izrazom (1). Izraz (1) ima tu osobinu da povećanjem ugla paljena tiristora α efektivna vrednost napona na potrošaču se smanjuje i obrnuto. Ako je ugao uključenja trijaka α=0º ulazni i izlazni napon su jednaki U_2=U_1=230 V, ako je ugao uključenja trijaka α=180º, efektivna vrednost napona na potrošaču je U_2=0 V.

Na slici 4 prikazani su karakteristični vremenski dijagrami napona i struje jednofaznog naizmeničnog pretvarača sa jednostranom faznom regulacijom za ugao uključenja α=60º i omsko opterećenje koji su dobijeni simulacijom u *Matlab/Simulink*-u.



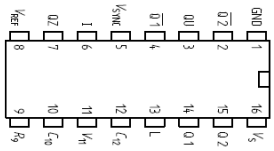
Slika 4. Vremenski dijagram napona i struje za ugao uključenja trijaka α=60º i omsko opterećenje.

II. INTEGRISANO KOLO TCA785

Integrisano kolo TCA785 korišćeno u praktičnoj realizaciji proizvodi kompanija SIEMENS i namenski se koristi za faznu kontrolu napona u jednofaznom sistemu napajanja [2]. Osnovne karakteristike integrisanog kola su:

- pouzdano prepoznavanje prolaza napona kroz nulu
- veliki obim praktične primene
- može se koristiti u trofaznim kolima
- izlazna struja 250 mA
- veliki temperaturni opseg

Na slici 5 prikazan je izgled integrisanog kola TCA785 sa obeleženim pinovima [2].



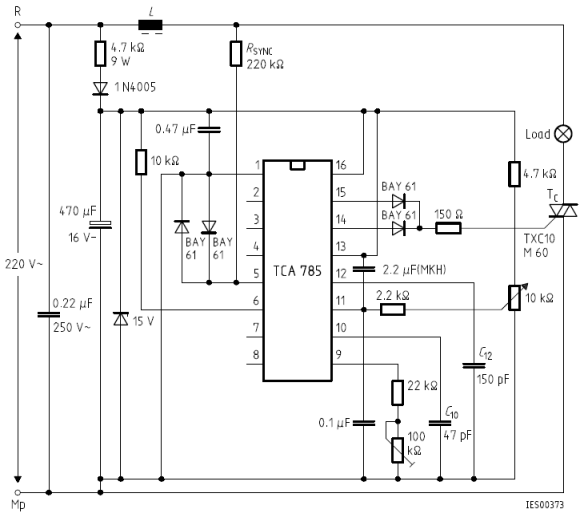
Slika 5. Izgled integrisanog kola TCA785 [2].

U Tabeli I data su značenja pojedinih pinova integrisanog kola. Značenje i namena pinova je veoma važna da bi se izvršilo pravilno povezivanje integrisanog kola TCA785[2].

TABELA I
ZNAČENJE I NAMENA POJEDINIH PINOVA INTEGRISANOG KOLA TCA 785 [2]

1	GND	Ground
2	$\overline{Q2}$	Output 2 inverted
3	Q U	Output U
4	$\overline{Q1}$	Output 1 inverted
5	V_{SYNC}	Synchronous voltage
6	I	Inhibit
7	Q Z	Output Z
8	V_{REF}	Stabilized voltage
9	R_9	Ramp resistance
10	C_{10}	Ramp capacitance
11	V_{11}	Control voltage
12	C_{12}	Pulse extension
13	L	Long pulse
14	Q 1	Output 1
15	Q 2	Output 2
16	V_S	Supply voltage

Na slici 6 prikazana je električna šema proizvođača integrisanog kola TCA785 za jednofazne pretvarače sa jednostranom faznom regulacijom [2].



Slika 6. Električna šema naizmeničnog pretvarača sa trijakom i upravljačkim kolom TCA785[2].

Nakon analize električne šeme pristupljeno je izradi elektronske pločice u odgovarajućem softveru (*Express*

PCB) za projektovanje elektronskih pločica, kontrola i priprema za izradu elektronske pločice.

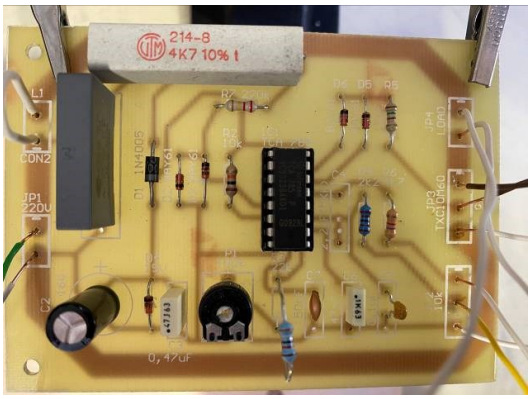
III. PRAKTIČNA REALIZACIJA PRETVARAČA SA JEDNOSTRANOM FAZNOM REALIZACIJOM I TESTIRANJE

Praktična realizacija je sprovedena u nekoliko koraka. Nakon izrade elektronske pločice neophodno je prvo izvršiti vizuelnu kontrolu pločice i kontrolu neprekidnosti provodnih linija. Zatim se pristupa postavljanju i lemljenju komponenti čije su vrednosti naznačene na slici 6. Temperatura lemilice treba biti pažljivo podešena da se prevelikom temperaturom ne bi pojavilo odvajanje linija od podloge. Samo integrisano kolo postavljeno je na odgovarajuće podnožje i nije zalemljeno direktno na pločicu. To je urađeno zato što u slučaju kvara integrisanog kola jednostavno možemo izvršiti zamenu pokvarenog integrisanog kola ispravnim. Na slici 7 prikazan je postupak lemljenja elemenata i provodnika na štampanu pločicu od strane autora rada.

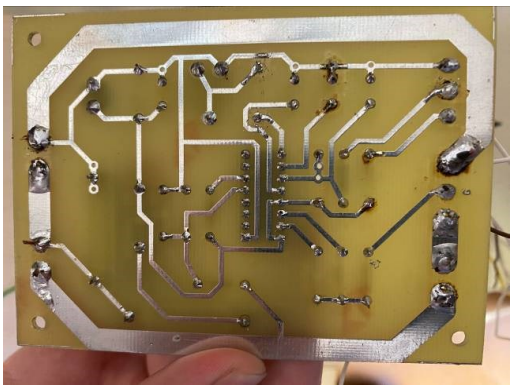


Slika 7. Lemljenje komponenti na štampanu pločicu pretvarača.

Na slici 8 prikazan je izgled gornje strane a na slici 9 izgled donje strane štampane pločice sa postavljenim komponentama.



Slika 8. Izgled gornje strane štampane pločice sa postavljenim komponentama.



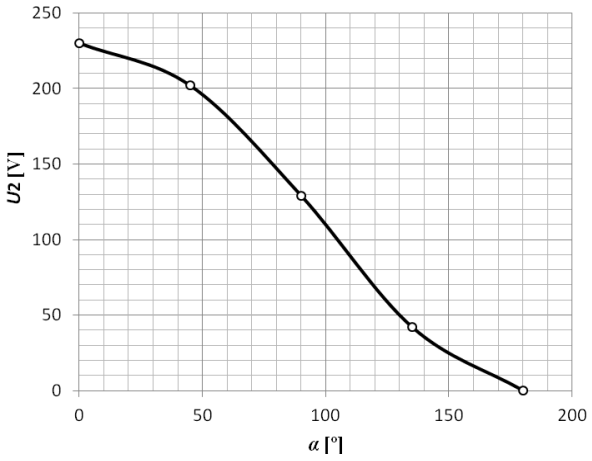
Slika 9. Izgled donje strane štampane pločice.

Nakon postavljanja komponenti vrši se testiranje rada elektronskog kola. Testiranje rada se zasniva na snimanju regulacione karakteristike [3]. Regulaciona karakteristika predstavlja zavisnost izlaznog napona U_2 u funkciji ugla paljenja trijaka i upravo je data izrazom (1). Testiranje je vršeno za omsko opterećenje a rezultati testiranja su prikazani u Tabeli II.

TABELA II
REGULACIONA KARAKTERISTIKA PRETVARAČA SA JEDNOSTRANOM FAZNOM REGULACIJOM

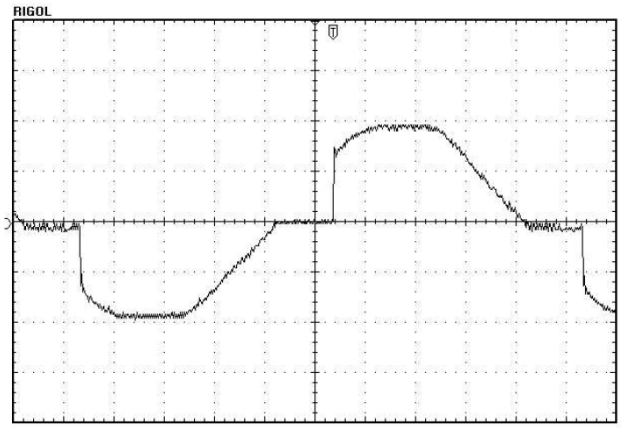
α [°]	0°	45°	90°	135°	180°
U_2 [V]	230	202	129	42	0

Na slici 10 prikazan je izgled regulacione karakteristike naizmeničnog pretvarača sa jednostranom faznom regulacijom i omskim opterećenjem [3].

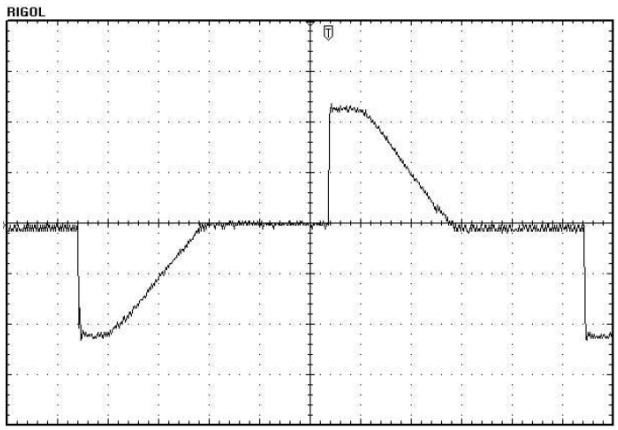


Slika 10. Regulaciona karakteristika naizmeničnog pretvarača sa jednostranom faznom regulacijom.

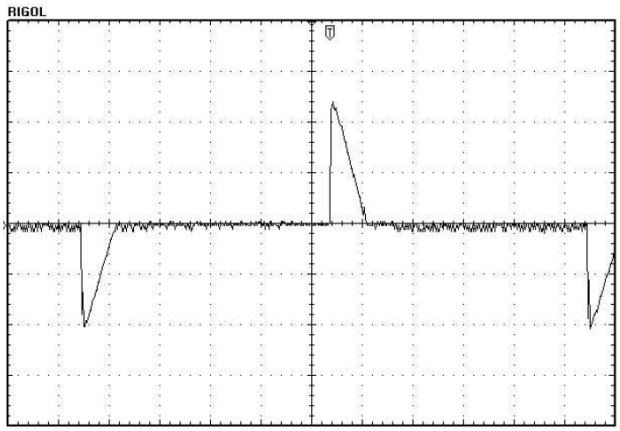
Na slikama 11, 12 i 13 prikazani su vremenski dijagrami za karakteristične uglove uključenja trijaka [3]. Slike 11-14 snimljene digitalnim osciloskopom RIGOL DS1052E i ogovarajućim softverom (*Ultrascope*) [4].



Slika 11. Vremenski dijagram napona na potrošaču za uključenja trijaka $\alpha = 45^{o}$.

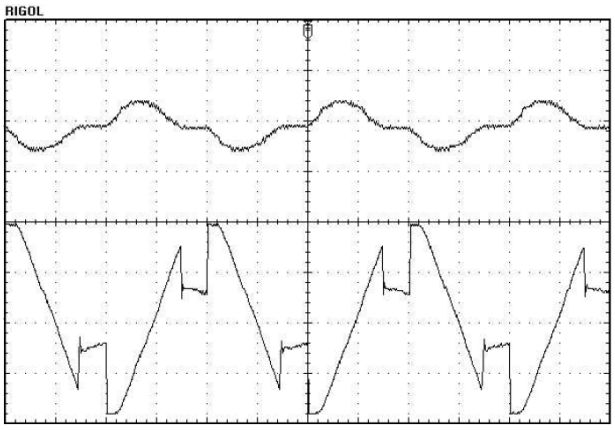


Slika 12. Vremenski dijagram napona na potrošaču za ugao uključenja trijaka $\alpha = 90^{o}$.



Slika 13. Vremenski dijagram napona na potrošaču za ugao uključenja trijaka $\alpha = 150^{o}$.

Na slici 14 prikazani su vremenski dijagrami napona i struje naizmeničnog pretvarača sa jednostranom faznom regulacijom za omsko-induktivnom opterećenjem (jednofazni asinhroni motor male snage) [3].



Slika 14. Vremenski dijagram napona i struje potrošača za omsko-induktivno opterećenje $\alpha \approx 26^{o}$.

IV. Zaključak

U ovom radu prikazan je primer praktične realizacije naizmeničnog pretvarača sa jednostranom faznom regulacijom primenom upravljačkog integrisanog kola TCA785. Rezultati testiranja u potpunosti odgovaraju teorijskim principima koji su izučavani iz energetske elektronike. Ova vrsta pretvarača ima veliku primenu u praksi počev od regulacije osvetljenja sijalica sa užarenim vlaknom, do regulacije temperature jednofaznih grejača pa do regulacije brzine jednofaznih asinhronih motora. Ograničenje se ogleda u nelinearnoj regulacionoj karakteristici jer ne postoji prporcija između ugla paljenja komponente i efektivne vrednosti napona na potrošaču.

Zahvalnica

Ovom prilikom želimo da se zahvalimo našem profesoru mentoru Skoko Saši na pomoći pri realizaciji ovog rada.

Literatura

[1] D. Martinović, Z. Pendić, J. Menart, Ž. Janda, *Energetska elektronika*, Zavod za udžbenike, Beograd, 2007
[2] Datasheet za TCA 785. Dostupno na https://datasheetspdf.com/datasheet/TCA785.html.
[3] S. Skoko, P. Kolarski, *Energetska elektronika-praktikum laboratorijskih vežbi*, Novi Sad, Elektrotehnička škola "Mihajlo Pupin", 2018.
[4] Softver za osciloskop RIGOL DS1052E. Dostupan na https://www.rigol.eu/

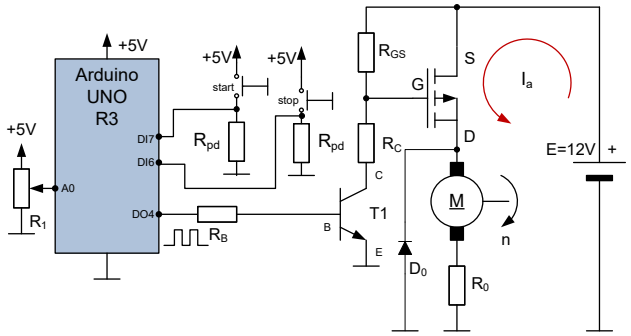
Praktična realizacija estimatora brzine u pogonu motora jednosmerne struje sa jednokvadrantnim čoperom

Dušan Novković, Nikola Šurlan, Mateja Dudaš

Sadržaj- U ovom radu predstavljena je praktična realizacija mikrokontrolerski upravljanog pogona motora jednosmerne struje sa jednokvadrantnim čoperom. Za potrebe procene brzine obrtanja vratila motora razvijen estimator brzine motora jednosmerne struje čiji se rad zasniva na filtriranju i merenju jednosmernog napona i struje motora. U završnom delu rada prikazani su rezulatai testiranja rada pogona i estimatora brzine.

I. MOTOR JEDNOSMERNE STRUJE NAPAJAN IZ JEDNOKVADRANTNOG ČOPERA

Motori jednosmerne struje imaju veliku primenu u mehatronskim sistemima. U velikom broju praktičnih primena postoji potreba za promenom i podešavanjem brzine motora jednosmerne struje. U tu svrhu se koriste različite vrste pretvarača energetske elektronike [1] . Jedna od jednostvanijih aplikacija je sa jednnokvadrantnim čoperom koja je razvijena za potrebe ovog projekta je prikazana na slici 1 [1-2].

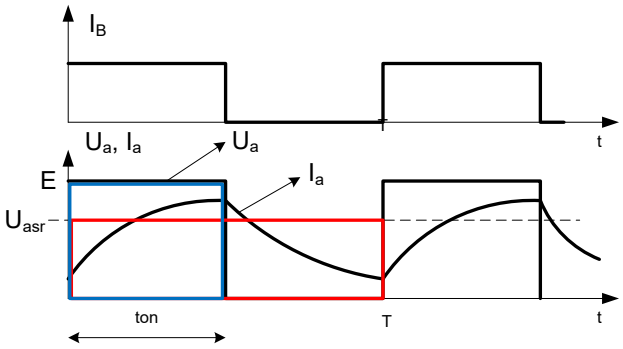


Slika 1. Principski izgled mikrokontrolerski upravljanog jednokvadrantnog čopera.

Tranzistorskim pretvaračem se upravlja primenom PWM algoritma upravljanja čiji se faktor ispune zadaje potenciometrom R_1. Tasteri start i stop služe za pokretanje i zaustavljanje rada pretvarača. Otpornici R_{pd} su pull-down otpornici za tastere. Tranzistor T_1 služi kao pobudni za snažni p-kanalni mosfet koji uključuje i isključuje motor.

Otpornik R_0 je noskoomski kalibrisani otpornik u funkciji strujnog šanta pomoću koga merimo struju motora. Kao izvor napajanja koristimo standardni izvor napona E=12V. Jednosmerni motor je napona 12V i snage 8W.

PWM tehnika upravljanja je tehnika impulsno-širinske modulacije čiji su karakteristični vremenski dijagrami prikazani na slici 2 [1].



Slika 2. Vremenski dijagrami pri PWM upravljanju jednosmernim motorom.

Ako faktor ispune ***a*** definisan je kao vreme trajanja uključenosti napona t_{on} i periode T, tada se može pisati

$$a = \frac{t_{on}}{T} \tag{1}$$

Do srednje vrednosti napona na motoru dolazi se na osnovu volt-sekundnog balansa kada se izjednače površina plavog i crvenog pravougaonika sa Sl. 2.

$$\begin{aligned} E \cdot t_{on} &= U_{asr} \cdot T \\ U_{asr} &= E \frac{t_{on}}{T} = E \cdot a \end{aligned} \tag{2}$$

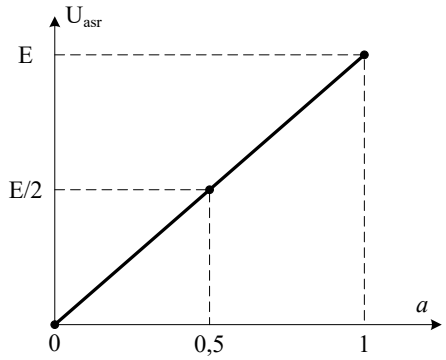
Na osnovu jednačina (2) se zaključuje da je srednja vrednost napona na motoru U_{asr} direktno srazmerna faktoru ispune a. Ako je perioda T konstantna, a vreme t_{on} se menja, menjaće se i vrednost faktora ispune, a na taj način se menja i srednja vrednost napona na motoru, što će uticati i na promenu brzine obrtanja motora n. Krakteristične vrednosti promene osnovnih veličina date su u Tabeli I.

Iz Tabele I i saglasno izrazu (2) može se uočiti da je promena srednje vrednosti napona na motoru linearna funkcija faktora ispune čiji je izgled dat na slici 3.

D. Novković, N. Šurlan i M. Dudaš su učenici Elektrotehničke škole "Mihajlo Pupin", Novi Sad, Futoška 17, smer tehničar mehatronike-4. razred, E-mail: dusan.n.game@gmail.com

TABELA I
PROMENA OSNOVNIH VELIČINA ZA RAZLIČITE VREDNOSTI FAKTORA ISPUNE PWM UPRAVLJAČKOG SIGNALA

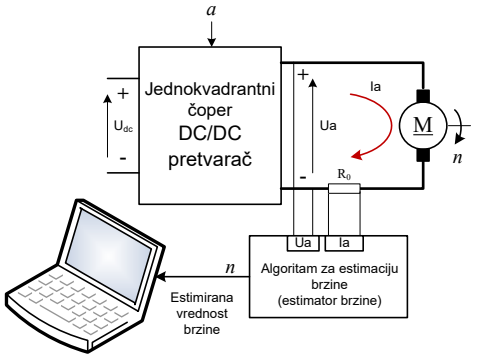
a	0	0,5	1
t_{on}	0	T/2	T
U_{asr}	0	E/2	E



Slika 3. Promena srednje vrednosti napona na motoru u zavisnosti od promene faktora ispune.

II. SINTEZA ESTIMATORA BRZINE JEDNOSMERNOG MOTORA

Estimatori su posebni uređaji koji se koriste u regulisanim električnim pogonima za određivanje neke nepoznate veličine [3] . U ovom radu biće izvršena sinteza estimatora brzine koji kao ulazne veličine koristi izmereni napon i struju motora. Principski izgled blok dijagrama pogona sa estimatorom prikazan je na slici 4.



Slika 4. Principski izgled blok dijagrama pogona sa čoperom i estimatorom jednosmernog pogona.

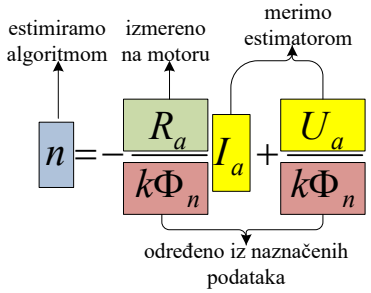
Do algoritma za procenu (određivanje) brzine obrtanja dolazi se na osnovu osnovnih naponskih jednačina motora jednosmerne struje koje su date izratom (3).

$$U_a = R_a \cdot I_a + E_a$$
$$E_a = k\Phi_n \cdot n \tag{3}$$

Kombinacijom jednačina koje su date u sistemu (3) dolazi se do izraza za brzinu obrtanja motora koji je dat jednačinom (4):

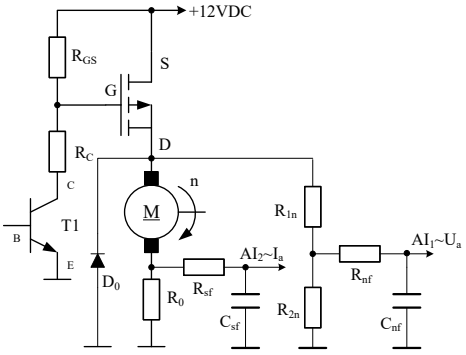
$$n = -\frac{R_a}{k\Phi_n} I_a + \frac{U_a}{k\Phi_n} \tag{4}$$

Pri čemu je: n-estimirana vrednost brzine obrtanja motora, R_a-otpornost armaturnog kola koju određujemo pripremnim merenjem IU metodom, I_a-struja armatirnog kola motora koju merimo estimatorom, U_a-napon armaturnog kola koga merimo estimatorom, $k\Phi_n$-konstanta koju određujemo iz naznačenih podataka motora za nazivni režim rada, s obzirom da je motor sa permanentnim magnetma na statoru član $k\Phi_n$ se ne menja tokom rada motora pa je u estimatoru konstantan. Na Sl.5 prikazan je ilustrativno pristup određivanju parametara za sintezu algoritma estimatora.



Slika 5. Pristup određivanja parametara za sintezu algritma estimatora.

Da bi mogli da izvršimo sintezu algoritma estimatora neophodno je izmeriti napon i struju motora. Za potrebe merenja pristupljeno je filtriranju napona i struje motora. Za potrebe merenja napona formiran je pre filtarskog kola napona naponski razdelnik (R_{1n},R_{2n}) kako bi se obezbedilo da pri maksimalnom naponu motora od 12V napon na analognom ulazu mikrokontrolera ne prelazi 5V. Principski izgled prilagodnog kola za merenje napona i struje sa kolima za filtriranje prikazan je na slici 6.

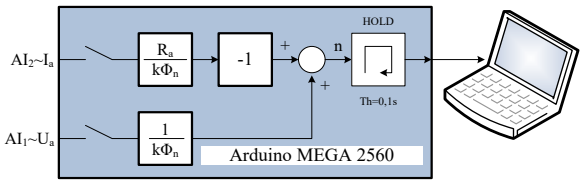


Slika 6. Principski izgled prilagodnog kola za merenje napona i struje sa kolima za filtriranje.

Prilikom izbora elemenata RC kola za filtriranje napona i struje primenjeni su uslovi dati izrazom (5).

$$\tau_n = R_{nf} \cdot C_{nf} \geq 3T$$
$$\tau_s = R_{sf} \cdot C_{sf} \geq 3T \quad (5)$$

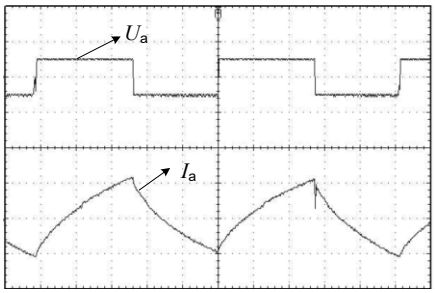
Signali AI_1 i AI_2 koji odgovaraju signalu napona U_a i struje I_a (respektivno) dovedeni su na analogne ulaze mikrokontrolerske kartice estimataora koja je realizovana sa Ardunino MEGA 2560 [3]. Principski izgled implementiranog algoritma prikazan je na slici 7.



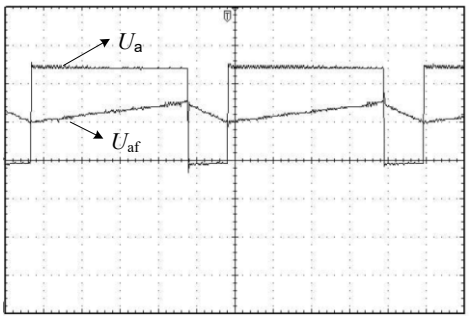
Slika 7. Principski igled implementiranog algoritma u kirokontrolerskoj kartici Arduino MEGA2560.

III. Rezultati testiranja pogona sa estimatorom brzine

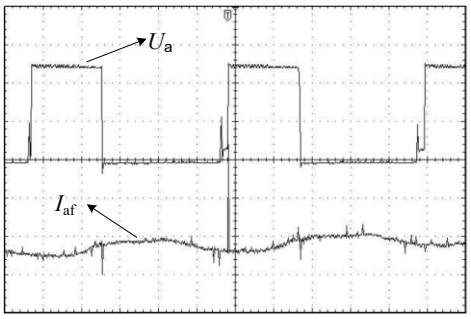
Na slikama 8-10 prikazani su karakteristični signali snimljeni u toku testiranja rada pogona sa estimatorom snimljeni digitalnim osciloskopom (softver Ultrascope for DS1000E series) [4].



Slika 8. Vremenski dijagrami napona U_a i struje motora I_a.



Slika 9. Vremenski dijagrami napona U_a i filtriranog napona U_{af}.



Slika 10. Vremenski dijagrami napona U_a i filtrirane struje I_{af}.

Na slikama 11-13 prikazanai su dijagrami estimirane brzine obrtanja koja se prikazuje u serijal ploteru na ekranu računara koji je povezan na karticu estimatora za različite sekvence upravljanja (Arduino serijal ploter).



Slika 11. Dijagram estimirane brzine n, prikazane u serijal ploteru na računaru - 1. snimljena sekvenca.



Slika 12. Dijagram estimirane brzine n, prikazane u serijal ploteru na računaru - 2. snimljena sekvenca.



Slika 13. Dijagram estimirane brzine n, prikazane u serijal ploteru na računaru - 3. snimljena sekvenca.

Važno je naglasiti da je završnim koracima neophodno sprovesti i finalnu kalibraciju i dodatno podešavanje rada estimatora. Izmerene vrednosti brzine obrtanja neophodno je uporediti sa vrednošću brzine koja je dobijena nekom drugom mernom metodom.

U ovom slučaju poređena je estimirana vrednost brzine sa laserskim merenjem stvarne brzine obrtanja primenom odgovarajućeg tahometra.

Dalji pravci rada u ovoj oblasti predstavljaju sintezu regulatora brzine na osnovu podatka o estimiranoj brzini obrtanja pogona.

Takođe bi bilo korisno na vratilo motora postaviti odgovarajući davač brzine (tahogenerator) i uporedno prikazati signale tahogeneratora i estimeatora brzine.

Posebno bi bilo analizirati rad estimatora na odskočnu pobudu referentne brzine.

Dugotrajni rad pogona izaziva zagrevanje motora. Porast temperature utiče na promenu otpornosti armaturnog kola što za posledicu ima negativan uticaj na rad estimatora i odstupanje estimirane i stvarne vrednosti brzine obrtanja. Razlog tome je što se podatak o otpornosti armature direktno nalazi u matematičkom izrazu po kome estimator procenjuje brzinu (4). Eksperimentalnim testiranjima ovaj efekat je uočen ali rešavanje uočenog problema izlazi iz okvira ovoga rada i može biti predmet nekog budućeg istraživanja.

IV. ZAKLJUČAK

U ovom radu prikazana je primer praktične realizacije jednosmernog pogona sa jednokvadrantnim čoperom i estimatorom brzine. Prilikom sinteze estimatora brzine potrebno je izvršiti dodatnu kalibraciju i uštimavanje algoritma u cilju tačnosti procene brzine. Tokom dugotrajnog rada pogona uočeno je da se motor greje što utiče na promenu vrednosti R_a a što utiče na tačnost procene brzine. Ovaj problem trebao bi da bude predmet nekih daljih istraživanja na ovu temu i usvršavanja rada estimatora brzine. Primena estimatora u regulisanim električnim pogonima posebno je važna jer se na taj način smanjuje broj senzora (davača) u pogonu a na taj način smanjuje i investiciona cena pogona.

ZAHVALNOST

Ovom prilikom želimo da se zahvalimo našem profesoru mentoru Skoko Saši na pomoći, savetima i sugestijama pri realizaciji ovog praktičnog rada.

LITERATURA

[1] R. Radetić, *Tranzistorski pretvarači*, Infoelektronika, Bor, 2006.
[2] https://docs.arduino.cc/hardware/uno-rev3
[3] V. Vučković, *Električni pogoni*, Elektrotehnički fakultet, Beograd, 1997.
[3] https://docs.arduino.cc/hardware/mega-2560
[4] S. Skoko, *Električni pogoni i oprema u mehatronici-praktikum laboratorijskih vežbi-skripta*, Novi Sad, Elektrotehnička škola "Mihajlo Pupin", 2022/23

Merenje parametara okoline pomoću *Arduino Mega* u *LabView* okruženju i slanje podataka na *ThingSpeak* platformu

Anđela Jovanović

Sadržaj – Rad se bavi korišćenjem Arduino platforme za merenje temperature okoline i količine svetlosti gde se *Arduino* razvojno okruženje programira u *LabView*-u. Kroz ovaj softverski paket, programiranje mikrokontrolera se vrši grafički uz pomoć blokova koji su deo *LabView* biblioteka. Merene podatke koje dobija sa *Arduino Mega* mikrokontrolera korisnik vidi na ekranu ali i fizički pomoću svetlosne indikacije. Takođe, osim merenja datih parametara biće data mogućnost da korisnik po želji sačuva podatke kao *Excel* fajl i da ih u svakom trenutku prikaže grafički. Podaci se šalju i na *ThingSpeak* platformu.

I. Uvod

Trenutno jedna od najpopularnijih mikrokontrolerskih pločica kako za manje i jednostavne projekte iz oblasti elektronike, tako i za složeniju upotrebu je *Arduino Mega*. Za njeno programiranje, kao i za programiranje ostalih *Arduino* razvojnih okruženja, razvijeno je okruženje *IDE* (*Integrated Development Environment*). Upotreba ovog alata podrazumeva poznavanje osnova *C* i *C++* jezika.

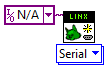
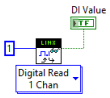
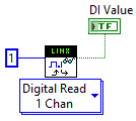
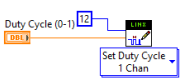
Međutim, mnogi drugi softverski paketi pored svojih bazičnih funkcija omogućavaju povezivanje i rad sa *Arduinom*, tako da korisnici mogu da biraju one pakete sa kojima su najviše upoznati i na taj način olakšaju sebi samo programiranje. Takvo jedno okruženje je i *LabView*, u kome će biti prikazan rad sa hardverskim komponentama kao što su senzori, tasteri, diode.

II. Softver

Sam softverski deo se radi u posebnom porozoru koji se zove *block-diagram* a programiranje se vrši pomoću blokova i na taj način je izrada koda intuitivna i olakšana. Korišćena verzija je *LabView 2023 Community Edition*.

Paket poseduje mogućnosti standardnih programskih jezika kao što su petlje, grananje, alati za rad sa fajlovima itd. Ono što ovaj programski paket čini izuzetno moćnim je što se kroz svaku novu verziju softvera povećava broj *add-on* alata koji su upravo i omogućili da se pomoću *LabView*-a može vršiti programiranje *Arduino* razvojnog okruženja na drugačiji način u odnosu na standardno programiranje [1].

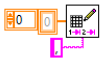
TABLE I
LabView blokovi korišćeni za rad sa *Arduinom*

	*Open Connection* - Služi za otvaranje konekcije sa *Arduinom* i postavljanje određenog *COM* porta na kome je *Arduino* pločica povezana.
	*Close Connection* - Služi za zatvaranje konekcije.
	*Digital Read* - Služi za očitavanje digitalnog ulaza (tastera npr.). Na ulaz se dovodi broj pina sa koga se želi očitavanje a na izlazu može postojati indikator stanja (*true/false*).
	*Analog Read* - Služi za očitavanje analognog ulaza (analognog senzora temperature npr.). Na ulaz se dovodi broj pina sa koga se želi očitavanje a na izlazu može postojati indikator stanja (brojna vrednost).
	*Digital Write*- Služi za upisivanje na digitalni izlaz (za paljenje i gašenje diode npr.). Na ulaz se dovode broj pina na koji se želi izvršiti upis kao i vrednost za upisivanje (*true/false*).
	*PWM Set Duty Cicle* - Služi za postavljanje *PWM* (*Pulse With Modulation*) iz opsega vrednosti 0-1 na pinu koji podržava *PWM* signal. Na ulaz se dovode broj pina na

A. Jovanović je student master studija na modulu Računsrsko upravljanje sistemima i merna tehnika i saradnik u nastavi na katedri za Automatiku na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: andjela.jovanovic@elfak.ni.ac.rs

	koji se želi izvršiti upis kao i vrednost za upisivanje.
	*Photocell* - Namenjen posebno za iščitavanje vrednosti sa *LDR* senzora. Na ulaz se dovodi broj pina na koji se želi izvršiti upis.

TABLE II
OSTALI KORIŠĆENI BLOKOVI

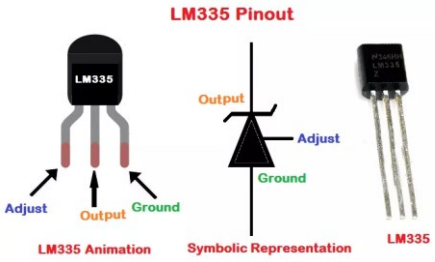
	*Open Handle* – otvara http konekciju sa serverom
	*GET* – postavlja podatke na server
	*Close Handle* – zatvara http konekciju
	*Write Delimited Spreadsheet* – vrši upis podataka sa ulaza u fajl
	*Read Delimited Spreadsheet* – vrši čitanje podataka iz fajla

III. HARDVER

Kako je zamišljeno da postavka ima edukativan karakter, odnosno da služi kao pokazni model za rad sa *Arduino* mikrokontrolerom, potrebne komponente za realizaciju su date u nastavku.

A. LM335 senzor

Ovo je analogni temperaturni senzor visoke preciznosti. Ovaj senzor je u osnovi integrisano kolo koje radi sa opsegom struje od 400 μA – 5 mA sa otpornošću manjom od 1 Ω. Radi kao Zener dioda sa dva izvoda. Jeftin je i lako dostupan na tržištu. Dobre osobine su mu širok temperaturni opseg isplativost, mala dinamička otpornost kao i tačnost do 1 °C. Raspon temperature koju meri je od -40 °C – 100 °C [2].



Slika 1. LM335 senzor.

B. LDR senzor osvetljaja

Fotootpornik, poznat kao *Light Dependent Resistor* (*LDR*) je svetlosno osetljiva komponenta koja se koristi za indikaciju količine svetlosti. U mračnom okruženju, otpornost je veoma visoka, ali kada se senzor izloži nekom izvoru svetlosti, otpornost se naglo smanjuje u zavisnosti od količine svetlosti. [3] Senzor kao izlaz daje relativnu vrednost osvetljenja, odnosno izlazna veličina nema jedinicu već samo izražava analognu brojnu vrednost u opsegu od 0-1023.



Slika 2. *LDR* senzor [3].

C. Taster

Taster je vrsta prekidača koji spaja ili prekida dva ili više provodnika samo onda kada je pritisnut od stane korisnika. Nakon prestanka delovanja, uz pomoć opruge se vraća u početni položaj [4].



Slika 3. Taster [5].

D. LED

LED su vrsta poluprovodničkih dioda koje emituju svetlost kada su direktno polarisane odnosno kad kroz njih teče struja. Mogu se naći u različitim bojama i veličinama [6].

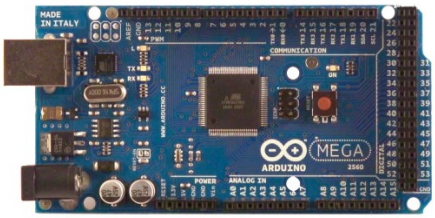


Slika 4. *LED* [6].

E. Arduino Mega 2560

Ova pločica sadrži 16 analognih i 54 digitalna pina, što je čini izuzetno pogodnom za rad sa velikim brojem hardverskih uređaja. Na digitalne pinove se mogu dovesti ili sa njih pročitati jedna od dve vrednosti: 0 ili 1. Na analogne pinove se može dovoditi analogni napon iz intervala 0 V – 5 V koji se u okviru desetobitnog *A/D* konverotra u samoj pločici prevodi u digitalni oblik iz intervala 0-1023. Postoje i pinovi na koje je moguće upisati

PWM signal, odnosno signal kod koga se u okviru jedne periode može menjati trajanje logičke 1 i logičke 0. Izvedeni su i pinovi za napajanje od 5 V i 3.3 V kao i *GND* pinovi [7].



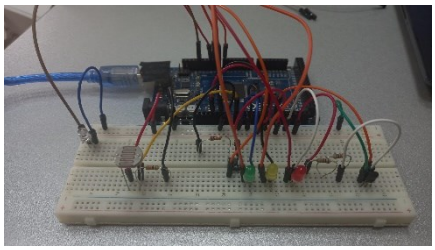
Slika 5. Arduino Mega 2560 pčočica [7].

IV. THING SPEAK

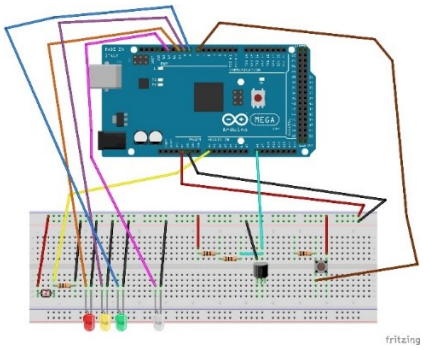
Ovo je platforma otvorenog koda koja omogućava komunikaciju uređaja preko interneta. Koristi se kod mnogih *IoT* aplikacija a razvijena je od *Math Works* tima. Korisnik kreira kanal na kome želi da se prikažu rezultati u vidu grafika, kojima se pristupa preko dobijenog *ApiKey*-a. Ima jednostavan interfejs koji omogućava lakše skladištenje, obradu i vizuelizaciju podataka. Podaci se mogu slati na određena polja u kanalu, a pristup kanalima može biti privatan ili javan. Podaci koji se skladište mogu dolaziti sa senzora, kao što je ovde to slučaj, i oni se upisuju na *ThingSpeak* u intevalima od po jedne sekunde. Samo slanje se naješće vrši tako što se u odgovarajućem softveru otvara internet konekcija ka stranici na kojoj se nalaze kreirani kanali i specificiraju se kanal i polje (putem linka u kome se navode *ApiKey*, ime kanala i redni broj polja).

V. SOFTVERSKA I HARDVERSKA REALIZACIJA

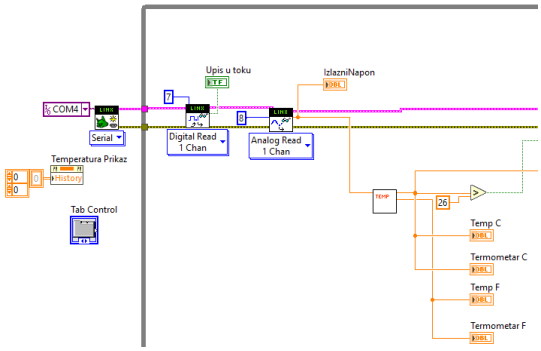
Izgled sistema u realnosti dat je na slici 6. Šema povezivanja u *Fritzing*-u data je na slici 7. Na slikama 8, 9 i 10 data je realizacija u *LabView*-u. Korisnik pokreće program i na front-panelu dobija rezultate merenja, dok se na platformi prikazuje grafik izmerenih vrednosti dok je pritisnut taster, a podaci se upisuju u kreirani fajl.



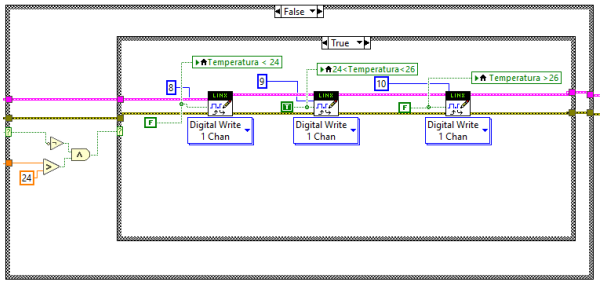
Slika 6. Fizička realizacija.



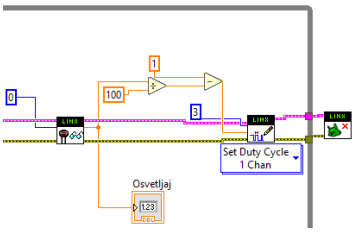
Slika 7. Šema povezivanja.



Slika 8. Otvaranje komunikacije, očitavanje i obrada podataka.



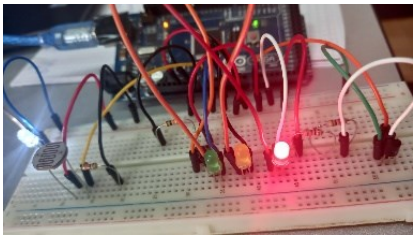
Slika 9. Paljenje odgovarajućih dioda zavisno od temperature.



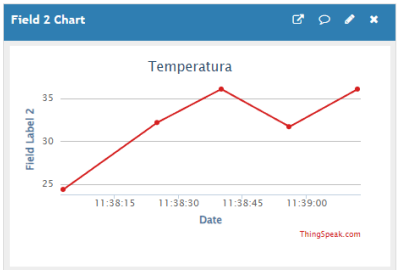
Slika 10. Čitanje osvetljaja, kontrola osvetljaja diode i zatvaranje komunikacije.



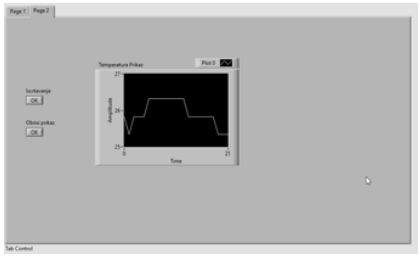
Slika 11. Prikaz na front panelu nakon pokretanja.



Slika 12. Izgled u realnosti.



Slika 13. Prikaz podataka na platformi [8].

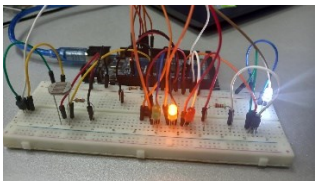


Slika 14. Iscrtavanje ranije snimljenih podataka.

Još jedan od mogućih scenarija kada je temperatura između 24 i 26 stepeni je dat na slikama 15. i 16. Tada je na snazi žuto upozorenje što se vidi kako na front panelu tako i na maketi.



Slika 15. Prikaz na front panelu za temperaturu između 24 i 26 stepeni.



Slika 16. Žuto upozorenje na temperaturu.

VI. Zaključak

LabView je moćan softverski alat koji omogućava i rad sa mikrokontrolerskim pločicama kao što je *Arduino*, bez gubljenja na mogućnostima, čak i sa dodatnim funkcionalnostima u odnosu na standardna okruženja.

Zahvalnica

Posebna zahvalnost članovima laboratorije za robotiku Elektronskog fakulteta u Nišu kao i profesorima Bobanu Veseliću, Jeleni Jovanović i Danijelu Dankoviću.

Literatura

[1] LabView - http://www.ed.rs/ed/softver/labview/opste.html.
[2] https://www.theengineeringprojects.com/2017/08/introduction-to-lm335.html.
[3] Fotootpornik - https://eepower.com/resistor-guide/resistor-types/photo-resistor/#.
[4] Taster - https://sr.wikipedia.org/sr-ec/%D0%A2%D0%B0%D1%81%D1%82%D0%B5%D1%80.
[5] Taster - https://www.gartner.rs/prekidaci-i-tasteri/tasteri-i-mikroprekidaci/mini-taster-za-st--plocu-dts62r.
[6] LED - https://my.cytron.io/p-led-3mm-green.
[7] Arduino - https://repository.unair.ac.id/55279/4/FV.OSI.42-16%20Bah%20%20r-3.pdf.
[8] MatLab -https://www.mathworks.com/.

High Frequency PWM Generator and MOSFET Bootstrapping Circuit

Luka Đorđević

Abstract In this paper a method of generating PWM, a MOSFET driver for both high and low side switching, as well as their advantages and limitations will be shown and explained. The main applications of these circuits would be switching mode power supplies and electric motor drivers. Both simulations of the circuit and laboratory tests of a prototype have been performed.

I. Introduction

PWM has a wide range of applications as the simplest method of digital to analog conversion. Switching mode power supplies, electric motor drivers, and dimming LED lights are some of its most common uses, the area of interest for this paper being power electronics. The PWM generator that will be covered is simple and will work the best with a feedback loop instead of a duty cycle based on a calculated estimate. In this paper the term "high frequency" refers to a range from a few hundred kilohertz to two megahertz.

II. PWM Generator

A. The Oscillator Circuit

Using a very simple circuit such as the one shown in Fig.1, we get a square wave with a duty cycle of approximately 50% (in my measurements 48%) on the output of the NE555 IC.

The way it works is simple: in one half cycle the output pin is charging the capacitor C1 until it reaches ⅔ of Vcc, triggering the internal voltage comparator connected to the trigger(TRIG) pin. That's when the second half cycle begins, during which the output pin is discharging the capacitor C1 until it reaches ⅓ of Vcc when the threshold(THRS) pin's internal voltage comparator switches the state of our output to Vcc once again.

Knowing how the circuit works we can use the formulas for capacitor charging and discharging in an RC circuit to figure out how to calculate what our RC time constant needs to be for a given frequency. Once the equations are solved we can derive Eq. 1 from them:

L.Đorđević, student of High School of Electrical Engineering "Nikola Tesla" and participant of the Petnica science center's seminar of electronics. Kraljice Natalije 31, 1100, Belgrade, Serbia, E-mail: lukadjordjevic8335@gmail.com

$$f = \frac{1}{2 \cdot 0.693 \cdot 2RC} \quad (1)$$

A view of the output voltage in a simulation is shown in Fig.2, and a picture of what it looks like on an oscilloscope is shown in Fig.3.

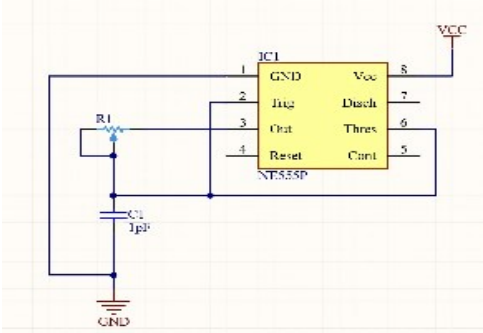


Fig. 1. Simple NE555 oscillator circuit.

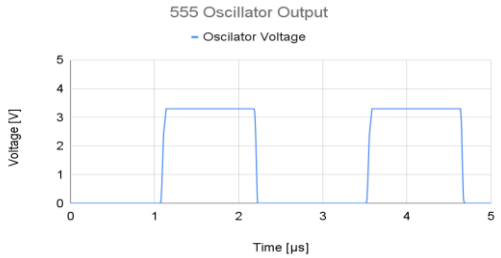


Fig. 2.Output pin waveform from the simulation.

Since we don't have ideal components in real life it's best to use a trimmer potentiometer as a rheostat instead of a fixed value resistor as R1 for fine-tuning the oscillation frequency.

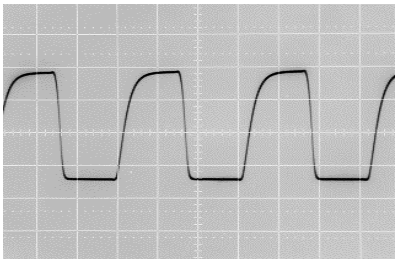


Fig. 3. Output pin waveform as seen on an oscilloscope with a time base of 0.5µS/div and a vertical sensitivity of 1V/div.

B. PWM

To turn our oscillations into a PWM signal with an easily adjustable duty cycle we will use the fact that our RC circuit that's a part of the oscillator works as a rough passive integrator, and one voltage comparator, in this case an LM311. The LM311 voltage comparator has an internal BJT and can be used to switch a higher current, and is in our case configured as an inverter. The circuit is shown in Fig.4.

The waveform of the voltage on C1 follows the rising and falling exponential functions, and by changing the voltage we compare it to we can in theory change the duty cycle of the comparator's output. The voltage on C1 never actually reaches 0, so we use C2 to eliminate the DC component, and R4 to compensate for negative voltage as can be seen in Fig.5.

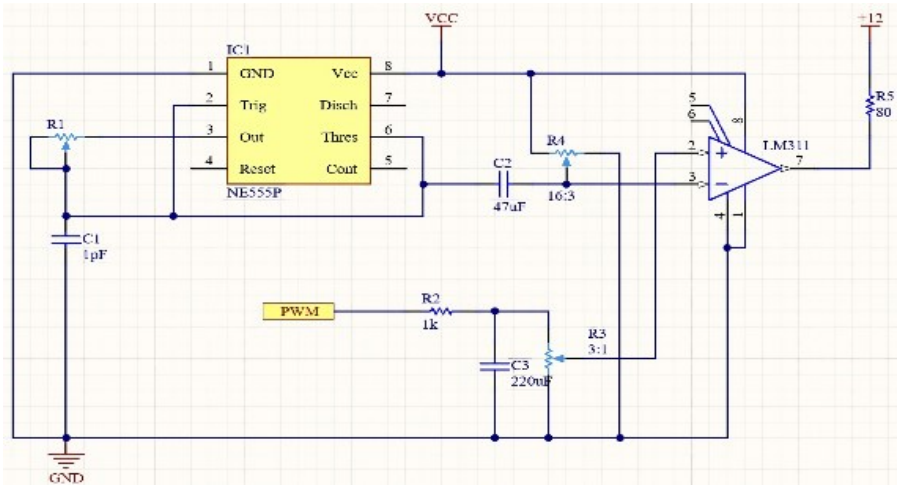


Fig. 4. Complete circuit for generating a high frequency PWM.

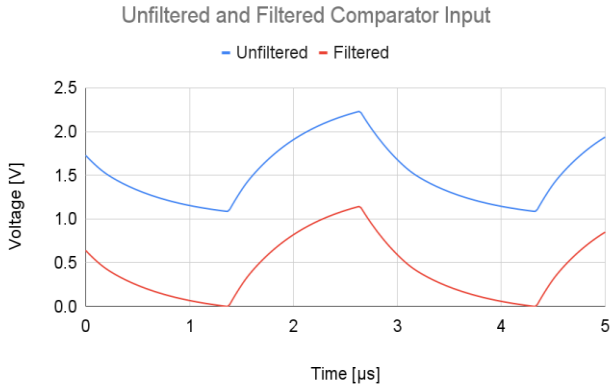


Fig. 5. The same waveform before (above) and after passing through the filter capacitor C2 (below).

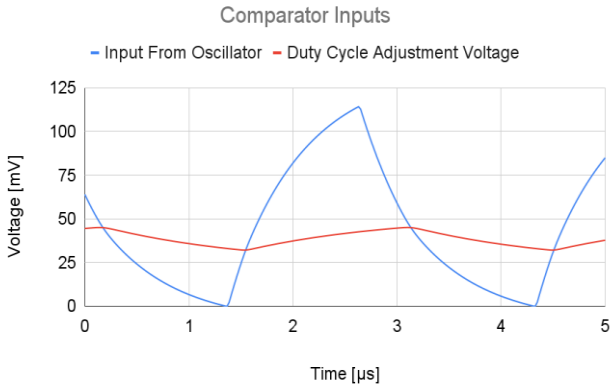


Fig. 6. Microcontroller PWM average and the oscillator output voltage.

Using R2 and C3 as another, more effective passive integrating circuit we can turn our microcontroller's low frequency PWM into a low ripple DC voltage. Using the potentiometer R3 we are reducing our 3.3V max voltage to the peak voltage on the output of our oscillator. By doing this we can keep the full resolution of our microcontroller's PWM, which is now upscaled in frequency. The average of the microcontroller's PWM and the oscillator's output can be seen on Fig.6. The high frequency PWM output can be seen on Fig.7.

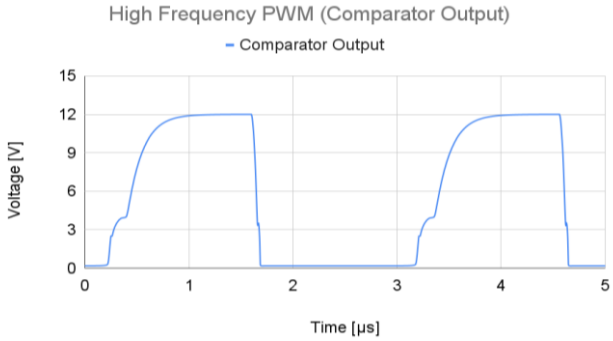


Fig.7. High Frequency PWM Output.

III. Introduction to MOSFET Bootstrapping

Bootstrapping a MOSFET refers to making a circuit that will drive an N-channel MOSFET as a high side switch. To understand the need for bootstrapping mosfets, one must first understand how they work. This paper will only cover the necessary basics.

In Fig.8 we have the symbols of an N-channel MOSFET, and a P-channel MOSFET [1].

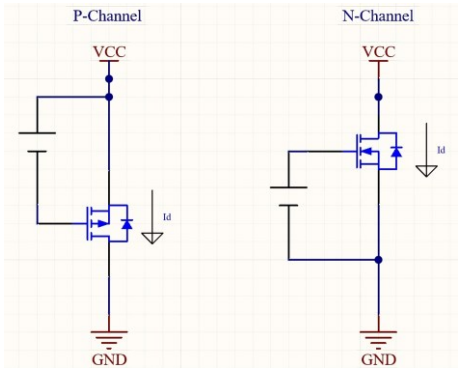


Fig. 8. Difference in Gate-Source voltage polarity between N-channel (left) and P-channel (right) MOSFETS.

What we can see is that for an N-channel MOSFET current flows from Drain to Source, and the Gate-Source Voltage has to be greater than the threshold voltage for it to begin conducting, making it convenient for low side switching (Fig.9). Inversely, a P-channel MOSFET's current flows from Source to Drain, and the Gate-Source voltage needs to be lower than the threshold voltage for it to begin conducting, making it convenient for high side switching (Fig.9).

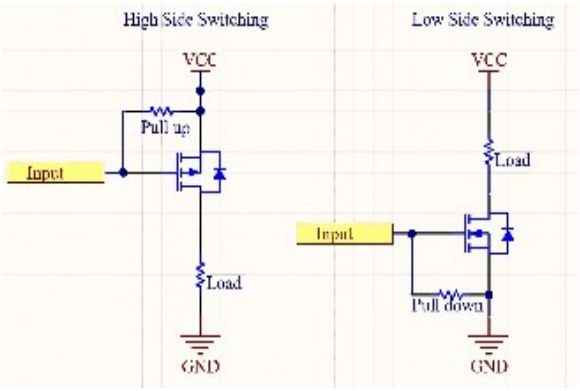


Fig. 9. Examples of low side switching (on the left), and high side switching (on the right).

P-channel MOSFETs have a higher internal resistance, which is why we want to use N-channel mosfets both as high and low side switches in high power applications. We run into an issue with high side switches though. Once the MOSFET starts conducting the voltage on the load, and by extension the Source of the N-channel MOSFET becomes equal to the voltage of our power source. Since the Gate-Source voltage needs to be positive on an N-channel MOSFET, it means that for an N-channel MOSFET we need to supply the gate with a voltage higher than the voltage of our power source. That is what bootstrapping is for.

IV. CMOS Bootstrapping Circuit

Compared to a basic single transistor inverter, a CMOS inverter has one key advantage: it doesn't have to constantly sink current to send out a negative voltage.
Due to the MOSFET's parasitic capacitance, driving one at high frequencies requires large peak currents, but it doesn't require high continuous currents, making a CMOS driver ideal.

The circuit is shown in Fig.10, the input and output waveforms are shown in Fig.11. The bootstrap circuit combined with the PWM generator is shown in Fig.12. The capacitor C4 supplies the circuit with a voltage higher than that of our power source, it charges while Q3 isn't conducting. In the other part of the cycle when Q3 is conducting, the voltage on the Source of Q2 is equal to the sum of the load voltage and the capacitor voltage, which is high enough to keep the Q3 powered.

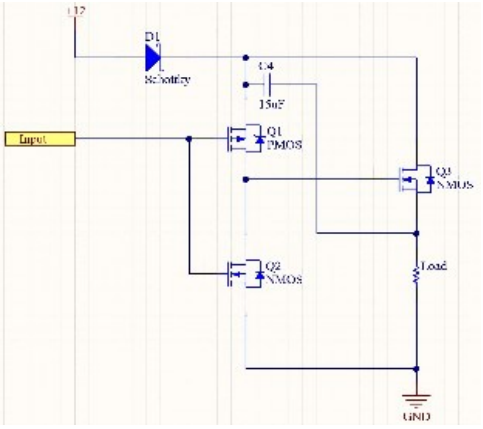


Fig. 10. Bootstrapping circuit powering the Q3 MOSFET high side switch.

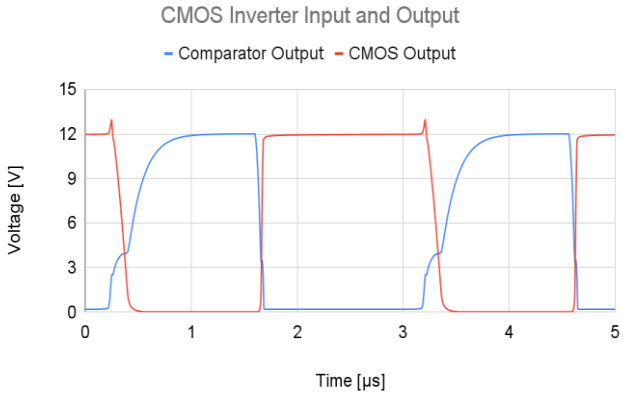


Fig.11. Input(wider) and output waveforms of the inverting circuit.

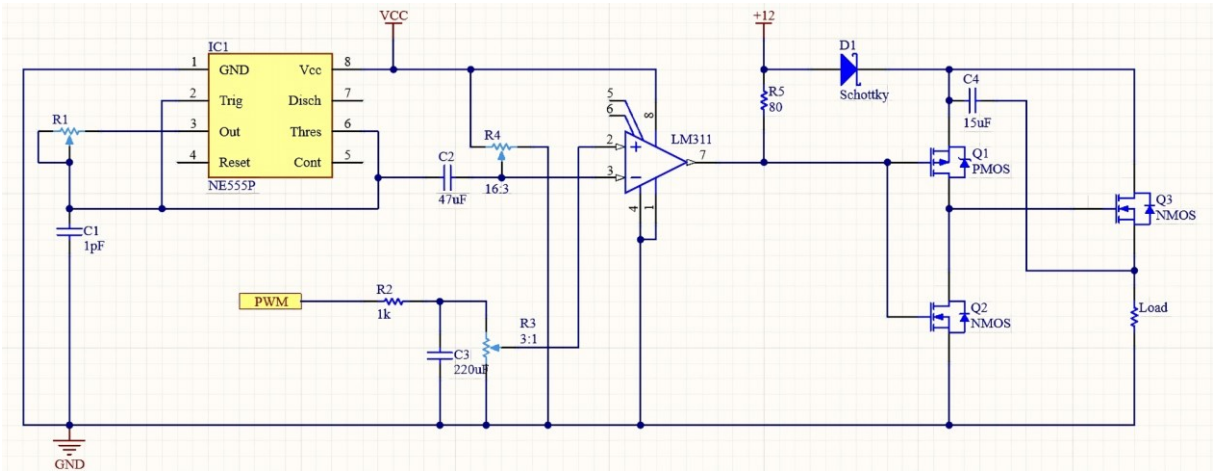


Fig.12. MOSFET driver being controlled by the PWM generator.

V. Conclusion

In this paper I covered two circuits, one simple PWM generator and one MOSFET driver. Both of those were designed with the aim of being used in DC DC converters for switching mode power supplies.

The main pros of these designs are that they can be easier to make for a prototype out of readily available parts, and that they are great for deepening your understanding of driving mosfets at high frequencies. The cons are that they're not as compact as a dedicated IC, and have more points of failure due to the amount of discrete components used in them.

Acknowledgements

I would like to express my thanks to Stevan Golubović who has helped me equip myself both with better instruments and with the knowledge required to use them.

References

[1] A. Prijić, D. Danković, Z. Prijić, "Uvod u Poluprovodničke komponente i njihovu primenu" ("Introduction to Semiconductor components and their applications"), Faculty of Electronic Engineering in the University of Niš, 2020.

Istraživanje haosa u Čua oscilatoru

Jovana Janjatović

Sadržaj – U radu je analizirana pojava haosa u Čua oscilatoru. Postojanje haosa dokazano je kvalitativnim, kvantitativnim i eksperimentalnim pokazateljima determinističkog haosa. Ovim radom objašnjeno je ponašanje haosa u najjednostavnijem kolu – Čua oscilatoru. Takođe, opisan je proces proizvodnje štampane ploče koja je upotrebljena za lakše eksperimentalno ispitivanje haotičnog ponašanja u ovom kolu.

I. UVOD

Haos (grč. Χάος - „upražnjen prostor, velika praznina“) je neuređeno ili nepravilno menjanje stanja nekog sistema. Razlikujemo dve vrste haosa:

- deterministički haos - javlja se u sistemima koji su matematički determinisani.
- nedeterministički haos - javlja se u sistemima koji nisu potpuno matematički određeni.

Haos koji se javlja u električnim kolima svrstava se u deterministički haos.

Osnovna karakteristika svih haotičnih sistema jeste da su nelinearni. Takođe, da bi se unutar sistema koji opisujemo mogla ispoljavati haotična svojstva, po teoremi Poenkare-Bendiksona, neophodno je da su jednačine stanja tog kola opisane diferencijalnim jednačinama, najmanje drugog reda ili sistemom od tri nezavisne diferencijalne jednačine prvog reda. Ono što karakteriše sve haotične sisteme jeste osetljivost na početne uslove („efekat leptira“).

Da bi se haotično ponašanje ispoljilo u nekom električnom kolu, u kolu mora postojati najmanje:

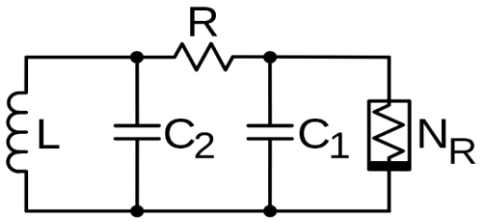
1. jedan nelinearni element
2. jedan ili više otpornika
3. tri ili više elemenata za skladištenje energije.

II. OPIS KOLA

Čua oscilator predstavlja najjednostavnije električno kolo koje ispunjava sva tri uslova za postojanje haosa. Njega je konstruisao Leon Čua u proleće, 1983. godine. Ideja je nastala iz potrebe da se osmisli laboratorijski sistem koji u realnosti može zadovoljiti Lorencove jednačine. Pored računarske simulacije, Leon Čua je eksperimentalno i matematički dokazao postojanje haotičnog ponašanja.

A. Šema kola - Čua oscilator

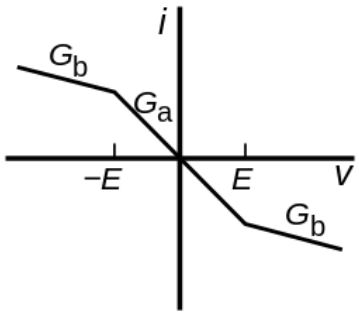
Na slici 1 uočavamo od čega se sastoji električno kolo. Na prikazanoj šemi, R predstavlja otpornik promenljive otpornosti (potenciometar), L je induktivni element, N_R je Čua dioda, a elementi C1 i C2 su kondenzatori.



Slika 1. Šema Čua oscilatora.

B. Čua dioda

Čua dioda je nelinearan element u ovom kolu. To je nelinearan otpornik sa dva terminala i parcijalno linearnom strujno-naponskom karakteristikom koja se može videti na slici 2. Kada se posmatra strujno-naponska karakteristika nekih nelinearnih elemenata, ona se razdvaja na linearne segmente, pa se isti problem svodi na linearnu analizu. U strujno-naponskoj karakteristici Čua diode mogu se posmatrati tri linearna segmenta.



Slika 2. Strujno-naponska karakteristika Čua diode.

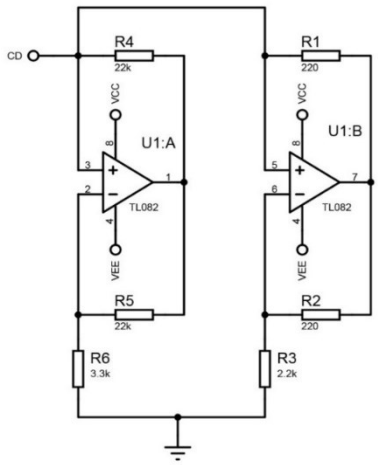
Struja koja prolazi kroz ovaj element, može se napisati u funkciji od napona na njenim krajevima: $i_L = f(v_1)$. Ta zavisnost prikazana je u formuli 1:

$$f(v1) = \begin{cases} m_1 x + m_0 - m_1, x > 1 \\ m_0 x, -1 \le x \le 1 \\ m_1 x - m_0 + m_1, x < -1 \end{cases} \quad (1)$$

gde je $m_0 = G_a/G$, $m_1 = G_b/G$, $G = 1/R$.

Postoji nekoliko načina realizacije Čua diode. Implementacija Čua diode koja je korišćena u ovom radu predstavljena je na slici 3. Ona se sastoji od dva operaciona pojačivača i šest linearnih otpornika.

J. Janjatović, učenica četvrte godine Šabačke gimnazije, Masarikova 13, 15000 Šabac. Polaznica Istraživačke stanice Petnica, seminar elektronike. Republika Srbija. E-mail: jovana.janjatovic.05@gmail.com



Slika 3. Implementacija Čua diode.

C. Jednačine stanja kola

Jednačine stanja su jednačine kola po strujama kalema, naponima kondenzatora i pobudama, napisane u dogovorenom obliku (Košijeva normalna forma). U ovim jednačinama, sa leve strane je prvi izvod napona kondenzatora i struje kalema. Sa desne strane jednakosti su algebarski članovi struja kalemova i napona kodenzatora.

$$\frac{dv1}{dt} = 2 * \frac{Vc2-Vc2}{Rc1} - \frac{iL}{C1} \quad (1)$$

$$\frac{dv1}{dt} = \frac{iL}{C2} - \frac{Vc2-Vc1}{Rc1} \quad (2)$$

$$\frac{diL}{dt} = -\frac{Vc2}{L} \quad (3)$$

Kada se uvedu smene, jednačine (1), (2) i (3), dobijaju sledeći oblik:

$$\dot{x} = \alpha(y-x-f(x)) \quad (4)$$
$$\dot{y} = x-y+z \quad (5)$$
$$\dot{z} = -\beta y \quad (6)$$

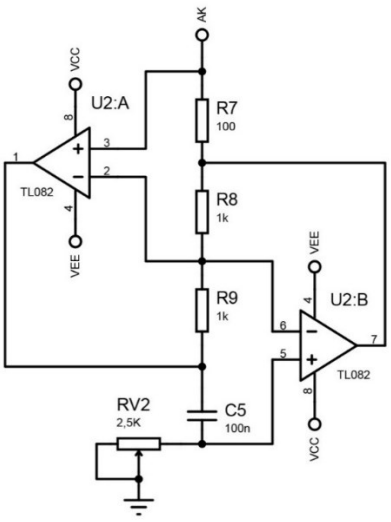
gde formula 4 opisuje promenu napona na kondenzatoru C1; α je parametar koji zavisi od odabira kapacitivnosti kondenzatora koji ulaze u kolo. Analogno, formula 5 opisuje promenu napona na kondenzatoru kapacitivnosti C2. Formula 6 predstavlja promenu struje na krajevima kalemova (u ovom slučaju, kola sa induktivnom osobinom).

D. Iduktivni element u kolu

Prilikom postizanja precizne i stalne vrednosti induktivnosti kalema može se naići na problem. Zato, u ovom kolu umesto kalema koristimo žirator, odnosno, kolo koje ima induktivnu osobinu. Prikaz ovog kola (nazivanog još i Antonio Inductance-Simulation kolo) sa vrednostima odgovarajućih komponenti vidi se na slici 4. Induktivnost ovog kola se dobija po formuli 7:

$$L = \frac{C*R7*R9*R10}{R8} \quad (7)$$

gde R10 predstavlja potenciometar kako bi mogli da menjamo vrednost induktivnosti ovog kola.

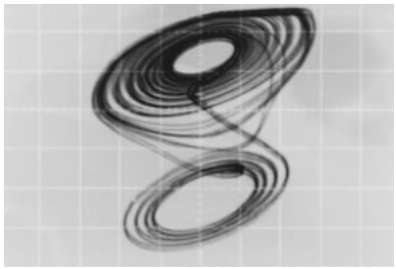


Slika 4. Antonio Inductance-Simulation kolo.

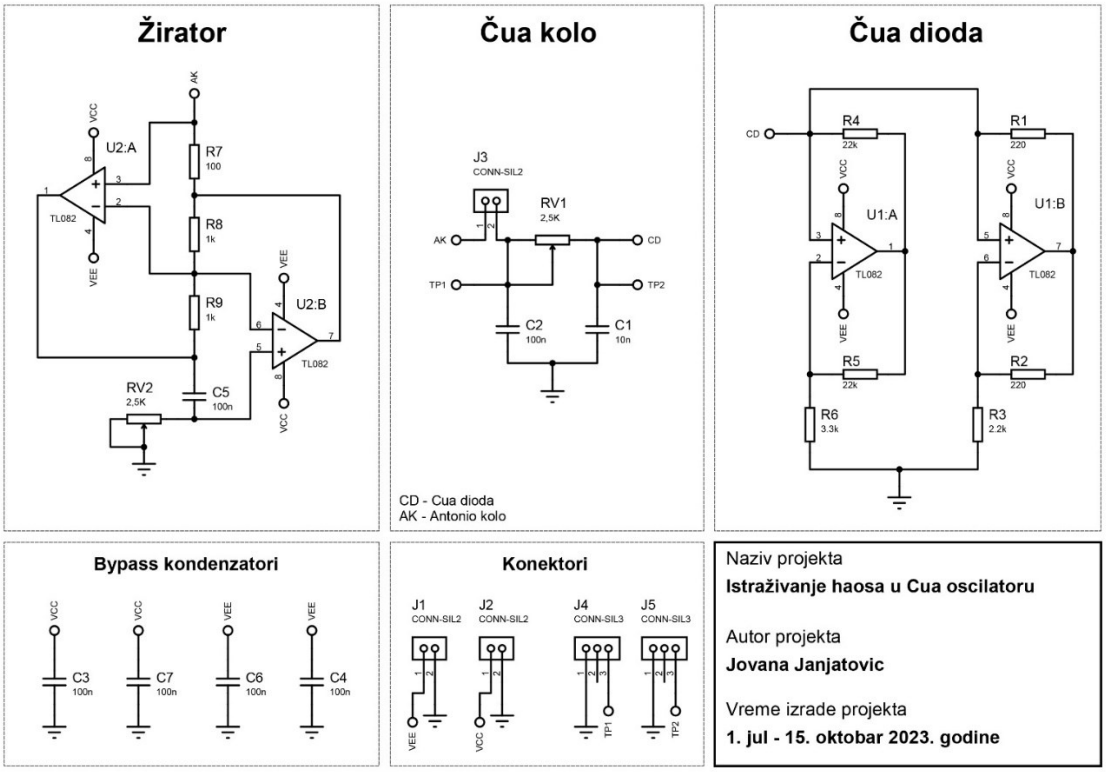
III. FAZNI DIJAGRAMI

Figura u faznom prostoru po kojoj se „kreće“ sistem naziva se atraktor. Čudni atraktor (koji se može videti na slici 5), za razliku od običnih, predstavlja geometrijski objekat koji ima fraktalnu dimenziju. Jedan od kriterijuma koji je neophodan da postoji, kada definišemo sistem koji je ušao u haotično stanje, jeste da je dimenzija njegovog atraktora fraktalna (necelobrojna).

Za posmatranje signala koji se javlja tokom oscilovanja, korišćen je merni uređaj osciloskop. Na X i Y osi prikazani su promenljive vrednosti napona na kondenzatorima C_1 i C_2. Zbog kašnjenja između trenutka merenja i prikaza na ekranu (koje se javlja prilikom korišćenja digitalnog osciloskopa) za prikazivanje haosa u električnom kolu, korišćen je analogni osciloskop.



Slika 5. Uočen čudni atraktor na analognom osciloskopu.



Slika 6. Šema kola.

IV. PROIZVODNJA ŠTAMPANE PLOČE

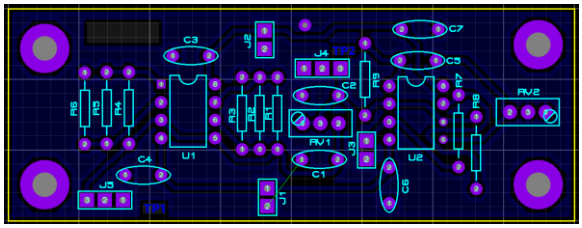
Štampana ploča (PCB – *Printed Circuit Board*), predstavlja elektronsku komponentu koja omogućava ostvarivanje veze između ostalih elektronskih komponenti koje su uključene u njen sklop. Štampana ploča čija je izrada opisana u ovom odeljku je jednoslojna. Ona sadrži komponente poređane sa jedne strane, a rute kojima su one povezane sa druge.

A. Šema štampane ploče

Dizajn štampane ploče počinje od šematskog prikaza kola (slika 6). Tačke CD i AK predstavljaju skraćenice za Čua diodu i Antonio kolo. Kondenzatori sa oznakama C3-C7 se nalaze na pinovima za napajanje na kućištu operacionog pojačavača (integrisanog kola TL082) i oni su *Bypass kondenzatori*. Njihova uloga u ovom kolu je stabilizacija napona. Oni su postavljeni u blizini pinova za napajanje integrisanog kola.

B. Dizajn štampane ploče

Nakon sastavljanja šeme kola, sa odgovarajućim vrednostima komponenata, osmišljen je raspored komponenti na štampanoj ploči dimenzija 80x30. Tokom dizajniranja izgleda ploče (sekcija *PCB Layout*), komponente su nekoliko puta postavljane, tako da se nađe njihova optimalna kombinacija. Nakon raspoređivanja komponenti, potrebno ih je povezati. Kako je u pitanju jednoslojna ploča, a sve komponente tipa THT (Through Hole Technology), taj proces je bio otežan. Finalni izgled ploče, može se videti na slici 7.



Slika 7. PCB Layout.

C. Proizvodnja štampane ploče

Za proizvodnju štampane ploče, izvršen je fotopostupak:

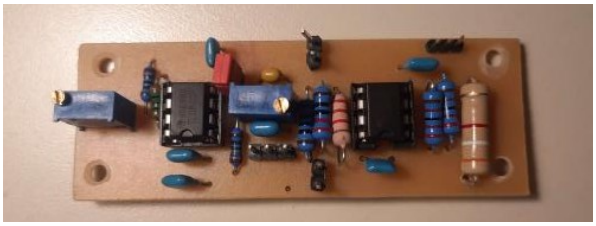
1. Štampanje layout-a na paus papir i postavljanje tog papira na pertinaks foto-pozitiv ploču odgovarajućih dimenzija.
2. Osvetljavanje ploče neonskim lampama, ultraljubičastim zračenjem.
3. Razvijanje ploče pomoću rastvora natrijum-hidroksida (NaOH).
4. Ecovanje (nagrizanje) ploče – za ecovanje je korišćen rastvor sone kiseline (HCl) i hidrogena (H_2O_2).

5. Nakon ecovanja ploče, ostatak nerazvijenog foto-rezista uklonjen je acetonom.
6. Bušenje rupa, postavljanje i lemljenje komponenata.

Izgled ploče nakon ovih koraka može se videti na slici 8, a finalni izgled štampane ploče sa komponentama, nalazi se na slici 9.



Slika 8. Izgled ploče nakon fotopostupka.



Slika 9. Finalni izgled štampane ploče sa komponentama.

V. Zaključak

Čua kolo je jedno od najpopularnijih dinamičkih sistema korišćenih za istraživanje dinamičkih ponašanja u haotičnim stanjima. Haotična kola, poput ovog, primenu nalaze u različitim naukama (fizika, biologija, ekonomija). Neka dalja istraživanja koja se nadovezuju na ovu temu, upravo bi mogla da ispituju ponašanja i primenu sinhronizovanog haosa u različitim sredinama.

Zahvalnica

Autor ovog rada se zahvaljuje Istraživačkoj stanici Petnica, na korišćenju laboratorija za elektroniku i hemiju, tokom fizičke realizacije projekta. Posebna zahvalnost rukovodiocu seminara elektronike, Stevanu Goluboviću, za ukazanu pomoć i podršku tokom izrade ovog projekta.

Literatura

[1] L. Fortuna, M. Frasca, M. Gabriella-Xibilia „Chua's circuit implementations, Yesterday, Today and Tomorrow“, Singapore 2009.
[2] N. Ojkić, „Deterministički haos i analiza nekih nelinearnih sistema“, diplomski rad, Univerzitet u Novom Sadu: Prirodno - matematički fakultet – departman za fiziku, Novi sad, 2006.
[3] The Antoniou Inductance - Simulation Circuit Derivation Valentin Siderskiy.
[4] D. Danković "Projektovanje štampanih ploča" Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, 2017.
[5] M. Etinski, T. Šarkić, „Istraživanje haosa u električnom kolu“, *Petničke sveske 49,* Radovi polaznika obrazovnih programa '99.

Pametna invalidska kolica

Filip Goldberger

Sadržaj – U ovom radu je opisan princip rada projekta „Pametna invalidska kolica“ (PIK), kao i komponente i alati korišćeni za njegovu izradu.

I. Uvod

PIK je *IoT* (*Internet of Things*) projekat razvijen sa ciljem da pomogne osobama sa invaliditetom, da koriste automatizovana kolica kojima se može upravljati putem interneta.

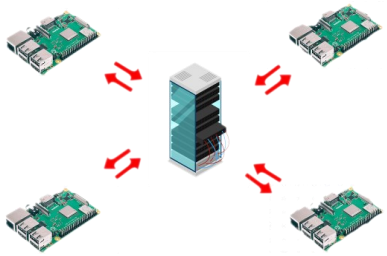
Projekat je nastao iz interesovanja studenta za robotiku, kako serverske strane za upravljanje, tako i softverske i hardverske strane za pravljenje skena prostora i njegovog naknadnog korišćenja za pronalaženje putanje do određene pozicije.

Glavne komponente korišćene za simulaciju invalidskih kolica su centralni server (PC računar), mikroračunar *Raspberry Pi* [1] i *TF Luna Lidar* [2].

II. Princip rada

Princip rada zasnovan je na konstantnoj komunikaciji između centralnog servera i mikroračunara preko interneta (slika 1). Centralni server prikuplja, prikazuje i čuva podatke koje kontroleri šalju i odgovara na iste sa odgovarajućom porukom. U isto vreme je u mogućnosti da preuzme kontrolu nad mikroračunarom u slučaju nužde. Pregled prikupljenih podataka i demo projekta mogu se naći na sajtu “*Pametni Šor*” [3].

Centralni server i mikroračunar imaju kod napisan u *C++* programskom jeziku i koriste *socket* [4] biblioteku za povezivanje preko interneta i slanja *UDP* [5] paketa.



Slika 1. Šema komunikacije četiri raspberry pi okruženja sa serverom.

F. Goldberger je student druge godine smera Računarstvo i automatika, Fakultet tehničkih nauka u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija, E-mail: filip.gold159@gmail.com

Mikroračunar preko senzora prikuplja podatke o trenutnoj udaljenosti kolica od prepreka i zidova prostorije i šalje ih na server. Na serveru se izračunava pozicija kolica, određuju potrebna pomeranja i šalju se komande motorima preko mikroračunara. Komunikacija između servera i mikroračunara je preko računarske mreže odnosno interneta.

III. Implementacija

A. Hardverska implementacija

Glavna komponenta upravljačkog dela projekta PIK je *Raspberry Pi* okruženje.



Slika 2. Izgled *Raspberry Pi* mikroračunara bez korišćenih senzora.

Raspberry Pi (slika 2) je relativno jeftin i svestran računar koji je i razvio *Raspberry Pi*. Namenski je dizajniran za obrazovanje, hobiste i profesionalce. Ovaj kompaktan uređaj podržava različite operativne sisteme i programske jezike, čineći ga idealnim alatom za učenje programiranja, izradu *DIY* projekata i rešavanje raznovrsnih zadataka, od kućne automatizacije i *IoT* aplikacija do pametnih invalidskih kolica. *Raspberry Pi* ima aktivnu zajednicu korisnika i pruža pristupačno rešenje za mnoge računarske potrebe.

Mikroračunar ima više funkcija u projektu *PIK*. Najpre koristi *LiDAR* (slika 3) senzor za pravljenje 2D skena prostora (zbog trenutnih mogućnosti korišćen je servo motor za simulaciju *RPLiDAR*-a [6]).



Slika 3. *TF Luna LiDAR*.

Izmerene vrednosti šalju se serveru na obradu i čuvanje. Na mikroračunar su povezani motori i enkoderi (slika 4) radi kretanja u prostoru, određivanja trenutne lokacije u prostoru i radi prikaza prostora.



Slika 4. Enkoder za računanje broja obrtaja motora i DC motor sa točkićem.

Enkoder [7] za motor koristi se za izračunavanje koliko su se kolica pomerila u nekom smeru od prvobitnog položaja. U obzir treba uzeti broj obrtaja („klikova“) enkodera kao i obim točka preko čega se računa koliko su se kolica zapravo pomerila (slika 5).

```
#define OBIM 0.25
#define METAR_TO_PIXEL 50
#define MAX_TICKS 360

int pomeraj(){
    int pixels, ticks;

    ticks = getTicks();
    pixels = ticks / MAX_TICKS * OBIM * METAR_TO_PIXEL;

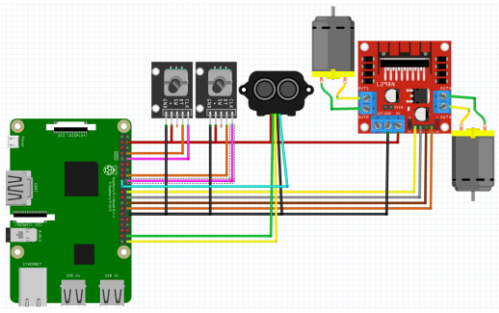
    return pixels;
}
```

Slika 5. Izračunavanje pređenog puta u pikselima preko enkodera.

B. Softverska implementacija

Korišćen je *C++* programski jezik uz biblioteku socket za slanje podataka preko interneta. Komunikacija između servera i mikroračunara odvija se preko *UDP* protokola za slanje paketa zbog bržeg slanja i prijema podataka, iako ova vrsta komunikacije nije najpouzdanija.

Paketi koje šalje mikroračunar sastoje se iz udaljenosti koje meri *LIDAR*, kao i ugla iz kog meri (0 – 180) i x i y vrednosti koje označavaju koliko su se kolica pomerila od prvobitnog položaja.

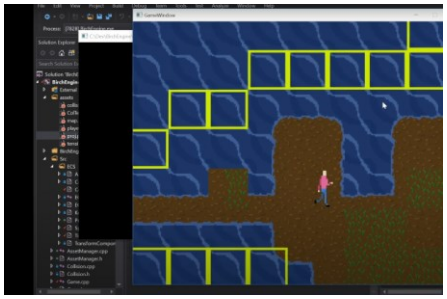


Slika 6. Šematski prikaz sistema.

Softverska implementacija na serverskoj strani uključuje grafički prikaz dobijenih podataka sa mikroračunara i kontrolisanje kretanja istog. Server svoje pakete šalje na adresu odakle su stigli paketi sa izmerenim vrednostima senzora i sastoje se od potvrđivanja preuzimanja komandi, kao i pravca kretanja. U slučaju prekida komunikacije, mikroračunar preuzima kontrolu i pali SOS signal.

C. Grafički interfejs

Za grafički interfejs na serverskoj strani zadužena je popularna *C++* biblioteka za razvoj multimedijlnih aplikacija *SDL* (*Simple DirectMedia Layer*). Ona omogućava jednostavan pristup hardverskim funkcijama, kao što su grafika, zvuk i ulaz, čime olakšava izradu igara, aplikacija za obradu slike, video editora i sličnih programa. *SDL* je platformski nezavisan, što znači da može raditi na različitim operativnim sistemima.



Slika 7. Primer upotrebe SDL biblioteke.

Pre upotrebe *SDL* biblioteke potrebno ju je inicijalizovati i proslediti joj neke početne vrednosti radi kreiranja prozora, kao što su veličina, pozicija, ime i još neka opciona podešavanja (slika 8).

```
#include "main.hpp"
#include "server.hpp"

const char* title   = "Test App";
const int   width   = 640;
const int   height  = 480;
const int   flags   = SDL_WINDOW_SHOWN;

int main(int argc, char** argv){
    if(SDL_Init(SDL_INIT_VIDEO) < 0){
        std::cout << "Error initializing SDL" << std:endl;
        return -1;
    }

    window = SDL_CreateWindow(
            title, SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOST_CENTERED,
            width, height, flags
        );

    if(window == nullptr){
        std::cout << "Error creating window" << std::endl;
        return -2;
    }
}
```

Slika 8. Inicijalizivanje *SDL* biblioteke i kreiranje novog prozora za crtanje.

IV. Zaključak

A. Prednosti sistema

Hardversko-softverska rešenja su potpuno primenljiva na realnom sistemu i cilj je da se to realizuje na realnom sistemu, invalidskim kolicima.

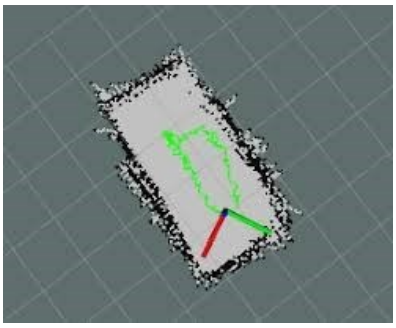
Prednosti sistema su olakšano kretanje invalida, kretanje po prostorijama i van istih, i praćenje i kontrolisanje u slučajevima nužde.

Unapređenja predviđena za sledeći period:

- dodavanje *UPS* [8] uređaja na mikroračunar omogućujući mu dodatno napajanje u slučaju gubljenja glavnog napajanja radi slanja SOS signala,
- *web* pristup i pregled podataka pomoću *NodeRed* [9] alata,
- *MariaDB* [10] baza podataka radi čuvanja informacija i istorije senzora,
- pravljenje sekcija na 2D mapi radi dodavanja predefinisanih položaja mikroračunara radi lakšeg i bezbednijeg kretanja po prostoriji.

B. Mane sistema

Zbog kratkog vremenskog roka izrade, projekat nije dostigao svoj pun potencijal, ali njegova izrada i unapređenja su u toku i napredovanje se može videti na *web* portalu "*Pametni šor*" [3].



Slika 9. Izgled *RAW* podataka sa senzora pre obrade podataka.

Sistem je trenutno zavistan od servera radi izračunavanja i pravljenje 2D prostorije, ali nastojimo ka tome da taj deo implementiramo u mikroračunar (*Raspberry Pi*) jer je on dovoljno moćan da ih vrši. Ovim bi postigli nezavisnost kretanja kolica od servera i mogućnost dodavanja malog displeja na kolica kako bi korisnik mogao da vidi gde se trenutno nalazi i da izabere poziciju gde bi hteo da ide.

ZAHVALNICA

Zahvaljujem se mentorima sekcije Ušan Branislavu, Pavlović Jovi, Skala Vladimiru i Vukobrat Milanu, kao i sadašnjim i bivšim članovima sekcije *„PiLab"* na sugestijama i podršci pri izradi ovog projekta.

LITERATURA

[1] Raspberry Pi. Dostupno na: https://www.raspberrypi.com/.
[2] TF Luna Lidar. Dostupno na: https://www.waveshare.com/wiki/TFLuna_LiDAR_RangeSensor.
[3] Pametni Šor. Dostupno na: https://pametnisor.pilab021.com/2023/10/16/pik-pametna invalidska-kolica/.
[4] Socket programiranje C/C+. Dostupno na: https://www.geeksforgeeks.org/socket-programming-cc/.
[5] UDPServer Client implementation in C++. Dostupno na: https://www.geeksforgeeks.org/udp-server-client-implementation-c/.
[6] RPLiDAR. Dostupno na: https://www.waveshare.com/wiki/RPLIDAR_A1.
[7] Enkoder, Arduino Tutorials, Electrical Engineering. Dostupno na: https://howtomechatronics.com/tutorials/arduino/rotary encoder-works-use-arduino/.
[8] UPS uređaj, Tech Target, Data Center: https://www.techtarget.com/searchdatacenter/definition/uninterruptible-power-supply.
[9] NodeRed alat. Dostupno na: https://nodered.org/.
[10] SQL baza podataka. Dostupno na: https://mariadb.org/.

Merenje isparljivih organskih jedinjenja

Mihajlo Marković, Stefan Karać, Dušan Petrović, Filip Goldberger

Sadržaj – U radu je prikazan sistem za merenje isparljivih organskih jedinjenja zasnovan na digitalnom senzoru povezanom na mikrokontroler. Razultati merenja prikazuju se na lokalnom displeju i na serveru.

I. Uvod

Isparljiva organska jedinjenja (eng. *Volatile organic compounds* – VOC) emituju se svakodnevno u vidu gasova iz raznih proizvoda ili predmeta. U domaćinstvu primeri ovih materija su razne hemikalije i proizvodi, u radnoj sredini to su boje, lakovi i predmeti od plastike, a u životnoj sredini derivati nafte. Izlaganje ovim isparenjima može imati ozbiljne posledice po zdravlje ljudi: kratkotrajne (kašalj, iritacija, alergije) ili dugotrajne (bolesti jetre i bubrega, kancer). Za merenje koncentracije ustanovljen je VOC indeks [1] i razvijeni su odgovarajući senzori. U radu je opisan sistem sa senzorom za merenje VOC, *Sensirion SGP40* [2] koji daje podatke merenja direktno proporcionalne VOC indeksu. Senzor je povezan na mikrokontroler tako da se rezultati merenja prikupljaju i prikazuju lokalno i šalju na server radi dalje obrade čuvanja i prikaza. Rezultati merenja su dostupni na *NodeRED* [3] serveru. Za razmenu podataka se koristi protokol *MQTT* [4]. *NodeRED* server i MQTT broker su instalirani na mikroračunaru.

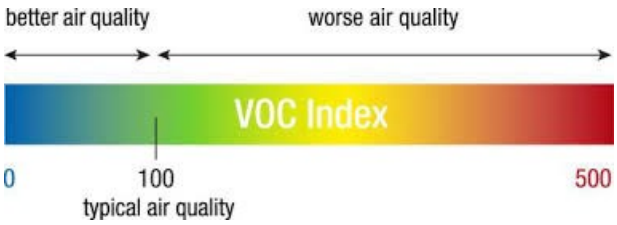
II. Princip rada

Merenje VOC indeksa podložno je uticaju ambijentalne temperature, vlažnosti vazduha i koncentracije štetnih gasova, pa proizvođači razvijaju posebne algoritme za pouzdano merenje. Skala VOC indeksa prikazana je na Slici 1.

U radu se koristi SGP40, digitalni senzor za merenje koncentracije VOC u zatvorenom prostoru, koji kompenzuje uticaj temperature i vlažnosti na rezultate merenja i omogućava dobijanje rezultata prema skali VOC indeksa. Za povezivanje sa mikrokontrolerom *ESP32* [5] koristi se I2C protokol (Slika 2).

M. Marković je učenik 4. razreda, a D. Petrović je učenik 2. razreda elektrotehničke škole „Mihajlo Pupin“, Futoška 17, 21000 Novi Sad, Srbija. E-mail: mihajlomarkovicets@gmail.com, anjadusan4@gmail.com

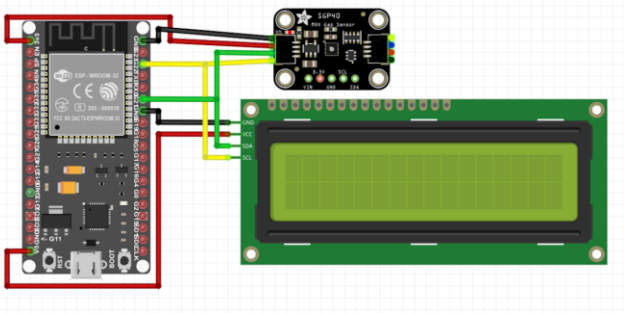
S. Karać je student prve godine, a F. Goldberger student druge godine studijskog programa Računarstvo i automatika, Fakulteta tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića, 21000 Novi Sad, Srbija. E-mail: stefankarac123@gmail.com, filip.gold159@gmail.com



Slika 1. Skala VOC indeksa.

MQTT je jednostavan komunikacioni protokol baziran na sistemu objave i pretplate (*publish – subscribe*). Svako očitavanje senzora se objavljuje na određenu temu. Uređaji koji žele da prime rezultate merenja se „pretplate“ na tu temu i dobijaju odgovarajuće rezultate merenja preko MQTT brokera. U radu se koristi broker *Mosquitto* [6]. Prednost ovakvog pristupa je što se omogućava jednostavna i istovremena distribucija poruka do svih zainteresovanih korisnika.

Mikrokontroler ESP32 očitava rezultat merenja VOC indeksa, prikazuje na LCD i šalje MQTT brokeru. Broker prosleđuje podatke *NodeRED* serveru.



Slika 2. Povezivanje ESP32 sa SGP40 putem I2C protokola.

Za tumačenje rezultata merenja koristi se Tabela I.

TABELA I
VREDNOST VOC INDEKSA I ODGOVARAJUĆI KVALITET VAZDUHA PREDSTAVLJENI RAZLIČITIM BOJAMA

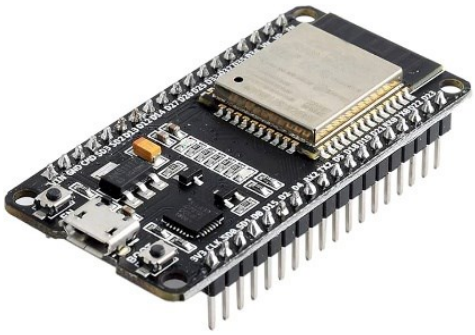
Vrednosti indeksa	Kvalitet vazduha
0 do 50	Dobar
51 do 100	Podnošljiv
101 do 150	nezdrav za osetljive grupe
151 do 200	zagađen (nezdrav)
201 do 300	veoma zagađen
301 do 500	rizičan

ESP32 povezan je sa LCD na kojem se prikazuju trenutne vrednosti VOC indeksa.

III. IMPLEMENTACIJA

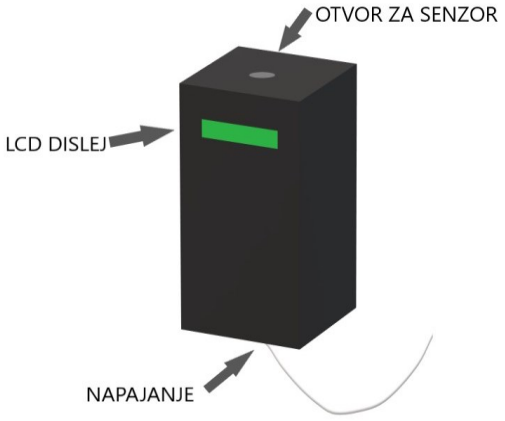
A. Hardverska implementacija

Za razvoj projekta korišćen je ESP32 WROOM (Slika 3), jeftina ali pouzdana mikrokontrolerska platforma sa integrisanim Wi-Fi i dual-mode Blututom. Poseduje dva interfejsa I2C, dva interfejsa I2S, četiri SPI, CANbus, PWM.



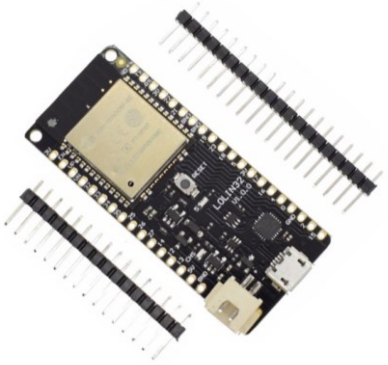
Slika 3. ESP32-WROOM.

Konačno rešenje za mobilnu verziju uređaja biće u kutiji (Slika 4). U toj verziji senzor SGP40 će biti u kombinaciji sa *ESP32 Wemos LOLIN32 Lite* (Slika 5) razvojnim okruženjem i konektorom za baterijsko napajanje.



Slika 4. Idejno rešenje za kutiju mobilnog uređaja.

ESP32 Wemos LOLIN32 Lite ima tridesetdvobitni dvojezgarni procesor koji radi na 240 MHz i 4 MB memorije. Može se programirati sa različitim jezicima, kao što su *microPython*, LUA, C/C++. Poseduje izvod za baterijsko napajanje i predstavlja odličan mikrokontroler za IoT rešenja.



Slika 5. ESP32 Wemos LOLIN32 Lite.

SGP40 (Slika 6) je digitalni gasni senzor firme Sensirion dizajniran za jednostavnu integraciju u uređaje za prečišćavanje vazduha i ventilacione sisteme sa kontrolom na zahtev. Zasnovan na CMOSens® tehnologiji, SGP40, predstavlja kompletan senzorski sistem na jednom čipu. Ima digitalni I2C interfejs za povezivanje sa mikrokontrolerom. Senzor karakteriše automatska kompenzacija uticaja temperature i vlage na rezultate merenja u zatvorenom prostoru. Da bi rezultati merenja bili što precizniji, potrebno je na sirove podatke iz senzora primeniti Senzirionov „*Gas Index Alghorithm*“ u mikrokontroleru. Kao rezultat dobijaju se rezultati merenja prema skali VOC indeksa.



Slika 6. SGP40.

Mikroračunar koji se koristi, *Raspberry Pi 3* [7] (Slika. 7) je single-board kompjuter veličine kreditne kartice. Procesor je četvorojezgarni ARM Cortex-A53, učestanosti 1200 MHz sa memorijom od 1 GB. Za čuvanje podataka koristi se MicroSDHC kartica. Operativni system je *Raspbian*.

B. Softverska implementacija

Program je napisan u jeziku C, a kao razvojna platforma korišćen je *Arduino-IDE*. Na slici 8 prikazan je primer koda za komunikaciju mikrokontrolera preko MQTT protokola.



Slika 7. Rapberry Pi.

```
void connectToWifi() {
  Serial.println("Connecting to Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}
void onWifiConnect(const WiFiEventStationModeGotIP& event) {
  Serial.println("Connected to Wi-Fi.");
  connectToMqtt();
}
void onWifiDisconnect(const WiFiEventStationModeDisconnected& event) {
  Serial.println("Disconnected from Wi-Fi.");
  mqttReconnectTimer.detach();
  wifiReconnectTimer.once(2, connectToWifi);
}
void connectToMqtt() {
  Serial.println("Connecting to MQTT...");
  mqttClient.connect();
}
void onMqttConnect(bool sessionPresent) {
  Serial.println("Connected to MQTT.");
  Serial.print("Session present: ");
  Serial.println(sessionPresent);
}
void onMqttDisconnect(AsyncMqttClientDisconnectReason reason) {
  Serial.println("Disconnected from MQTT.");

  if (WiFi.isConnected()) {
    mqttReconnectTimer.once(2, connectToMqtt);
  }
}
void onMqttPublish(uint16_t packetId) {
  Serial.print("Publish acknowledged.");
  Serial.print("  packetId: ");
  Serial.println(packetId);
}
```

Slika 8. Komunikacija preko MQTT protokola.

U server NodeRED potrebni nodovi prevučeni su na tablu, konfigurisani i povezuani. Funkcije koje nisu bile definisane, pravljene su sa „*function node*“ a kod je pisan u *JavaScript*-u.

Na slici 9 je prikaz merenja VOC indeksa prilikom testiranja uređaja.

Detalji o projektu [8] i rezultati merenja u realnom vremenu [9] mogu se videti preko navedenih URL adresa.



Slika 9. Prikaz merenja VOC indeksa.

III. Zaključak

Željena funkcionalnost sistema je postignuta. Uređaj za merenje VOC indeksa uspešno radi a podaci se objavljuju. Server prihvata podatke, čuva ih i predstavlja u zahtevanom formatu. U planu je nadogradnja sistema dodavanjem ventilacionog dela koji bi se aktivirao u slučaju povećanog nivoa VOC indeksa.

Zahvalnica

Zahvaljujemo se mentorima Ušan Branislavu i Vukobrat Milanu na smernicama i sugestijama pri izradi ovog rada.

Literatura

[1] https://www.pressac.com/insights/what-are-volatile-organic-compounds-vocs-and-why-should-you-be-monitoring-them/
[2] https://www.sensirion.com/products/catalog/SGP40
[3] https://nodered.org/about/
[4] https://mqtt.org/
[5] https://www.espressif.com/en/products/socs/esp32
[6] https://mosquitto.org/
[7] https://www.raspberrypi.com/products/raspberry-pi-3-model-b/
[8] https://pametnisor.pilab021.com/2023/10/16/voc-merenje-isparljivih-organskih-jedinjenja/
[9] http://91.187.148.150:12880/ui/#!/0?socketid=9PV4ZcTfKMPxY-O7AAAP
[10] https://evocs.org/o-voc-u/
[11] https://epa.org.me/wp-content/uploads/2022/04/VOC-fl.pdf

Automatizacija kreiranja modela registarskih blokova pri verifikaciji ASIC čipova UVM metodologijom

Bogdan Šipetić, Novak Radivojević

Sadržaj - U ovom radu biće prikazan automatizovani postupak kreiranja registarskih modela radi detaljne verifikacije registarskih blokova koji su sastavni deo svakog ASIC čipa. Automatizacija je postignuta pomoću *Python* skripte *gen_ral.py* koja na osnovu podataka o registrima datim u *.xls* dokumentu kreira kompletan model registarskog bloka. Dobijeni model se sastoji iz niza povezanih *SystemVerilog* klasa koje opisuju hijerarhiju registarskih blokova i bazirane su na postojećim klasama koje pruža UVM metodologija. Po kreiranju, UVM testovima se ispituje ispravnost rada registara na osnovu poređenja njihovih odziva i stanja sa očekivanim ponašanjem modela. Ovim postupkom se značajno skraćuje vreme procesa verifikacije registarskih i memorijskih blokova, kao i drugih sličnih modula. Obzirom da skripta uzima *.xls* dokument kao ulazni parametar, ovakav vid generisanja modela olakšava ažuriranje kako dokumentacije, tako i samog koda verifikacionog okruženja.

I. VERIFIKACIJA REGISTARSKIH POLJA

U svakom ASIC (*Application Specific Integrated Circuit*) čipu, polje registara (*register file*) ima glavnu ulogu u upravljanju rada čitavog sistema. Zbog svoje namene i konstrukcije, registarska polja predstavljaju pogodan medijum preko kog korisnici mogu podešavati sistem, pokretati razne aktivnosti koje sistem pruža i posmatrati rezultate tih aktivnosti. Upotreba registara kod ovakvih čipova je najčešće omogućena specijalizovanim softverskim interfejsa gde korisnik može zadavati komande upisa/čitanja i time postaviti sistem u željeni režim rada.

Registri u ASIC čipovima se prema nameni najčešće dele na konfiguracione, statusne i komandne [1]. Pomoću konfiguracionih registara vrši se podešavanje sistema i njegovih periferija i određuje se način na koji će se neke aktivnosti sistema odvijati. Statusni registri služe kao uvid u trenutno stanje jednog elementa sistema, daju rezultat neke operacije ili informacije o dešavanjima u sistemu (prekidi/*interrupts*, pojava greške/*error reporting* itd.). Komandni registri najčešće služe za iniciranje neke operacije sistema nakon što je obavljena konfiguracija.

Pošto od ispravnosti rada registarskog polja zavisi i ispravnost čitavog sistema, tokom verifikacije ASIC čipova posebna pažnja je posvećena registrima. Zarad uspešne verifikacije, potrebno je da testiranje dizajna bude zasnovano na realnom načinu upotrebe sistema. Kako su registri najčešće i jedini način pomoću kog korisnik ostvaruje kontrolu, tako se i većina verifikacionih testova sastoji iz simuliranja upisa i čitanja u određene registre. U testovima se podešavaju i iniciraju aktivnosti koje test cilja, a potom se čitanjem iz registara proverava status sistema i dobijeni rezultati se porede sa očekivanim [2]. U tu svrhu, vrši se kreiranje modela registarskih polja koji na visokom nivou oponašaju funkcionalnosti realnih registara u dizajnu i služe kao referenca prilikom provere rezultata. Modeli su korisni pošto pored grešaka u elementima digitalnog dizajna, mogu da ukažu i na greške u samom registarskom bloku.

Kao implementacija univerzalne metodologije verifikacije (UVM – *Universal Verification Methodology*), u velikoj je upotrebi biblioteka klasa napisanih na jeziku za dizajn i verifikaciju *SystemVerilog* koje podržavaju kreiranje raznih delova verifikacionog okruženja. Podskup ove biblioteke koji se naziva UVM *Register Abstraction Layer* (UVM RAL) pruža API (*Application Programming Interface*) koji sadrži klase i metode za implementaciju registarskih modela [3]. Modelovanjem registara i integracijom modela u verifikaciono okruženje, omogućava se efikasna verifikacija pristupa i manipulacije realnim registrima dizajna. Pomoću UVM RAL-a moguće je na nivou klase opisati svaki registar pojedinačno, a potom grupisati sve registre u više hijerarhijske strukture kao što su blokovi i fajlovi. Zbog svoje strukture i pristupačnosti, RAL olakšava interoperabilnost između različitih alata i metodologija, a samim tim i timova inženjera koji rade u okviru istog projekta.

Problem vezan za registarske blokove koji se javlja kako kod inženjera verifikacije, tako i kod inženjera digitalnog dizajna je to što moderni digitalni sistemi mogu zbirno imati i po nekoliko stotina registara u svim svojim blokovima. Rad svih registara mora biti ispitan i zbog toga nije pogodno ručno kreirati svaki registar i njegov model ponaosob [4]. Pritom, svi registri imaju sličnu strukturu i osobine tako da se može lako uočiti šablon po kom se registri i njihovi modeli kreiraju. Zbog toga je ovo idealan primer problema koji se može rešiti automatizacijom [5].

Glavni princip kod ovakvog vida automatizacije se sastoji iz kreiranja liste registara u nekom jednostavnom obliku (npr. *xls* fajl) koji bi bio čitljiviji od koda i koji se može jednostavno ažurirati, a potom napisati skriptu na odgovarajućem jeziku koja će učitati ovaj fajl, njegovu

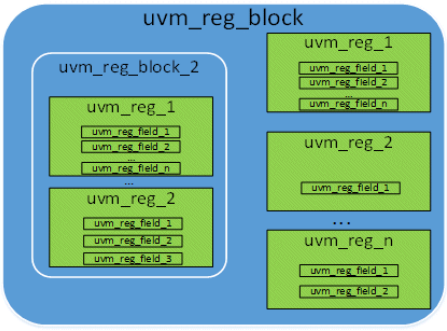
B. Šipetić je master inženjer elektrotehnike i računarstva na studijskom programu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: bogdan.bogi.sipetic@gmail.com

N. Radivojević je asistent pri Katedri za elektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: novak.radivojevic@elfak.ni.ac.com

sadržinu prevesti u oblik *SystemVerilog* klasa a potom dobijene klase sačuvati u novom fajlu. U ovom radu, skripta će biti napisana na jeziku *Python*, a lista registara će biti sadržana u *Excel* tabeli.

II. Osobine i organizacija UVM RAL-a

Organizacija i struktura RAL modela nastala je po uzoru na način na koji digitalni dizajneri vrše strukturu i raspoređivanje registara po blokovima u digitalnim sistemima [6]. Primer modela jedne ovakve strukture dat je na slici 1. Tipičan dizajn sadrži više blokova, gde svaki blok sadrži jedan podskup registara iz celokupnog polja relevantnih za njegove aktivnosti. Registri u okviru jednog polja su iste dužine bitova, i u svakom registru, bitovi su podeljeni na uzastopne grupe (polja registra). Svako polje nosi određeni podatak kojim se ili vrši kontrola nekog elementa hardvera ili koji oslikava trenutno stanje nekog elementa. Svako polje u registru može imati različita prava pristupa (*access policy*) u zavisnosti od namene. Najčešće korišćeni načini pristupa su dozvola čitanja i upisa (*read-write,* RW), samo čitanje (*read-only,* RO) i samo upis (*write-only*, WO) [1].



Slika 1. Primer modela registara sa različitim nivoima hijerarhije.

U skladu sa hijerarhijom blokova u dizajnu, i klase modela su hijerarhijski raspoređene, gde svaka od njih predstavlja „kontejner“ klase u kojima se instanciraju objekti klasa iz njima nižeg nivoa u hijerarhiji.

Glavni elementi RAL-a su:

- Model bloka registara (klasa *uvm_regblock*);
- Fajl registara (klasa uvm_*reg_file*);
- Model registra (klasa *uvm_reg*);
- Model polja registra (klasa *uvm_reg_field*).

Ove klase predstavljaju bazu za kreiranje modela. Za potrebe nekog modela, nijedna od ovih klasa se ne koristi direktno (sem *uvm_reg_field*), već se kreiraju nove klase koje ih nasleđuju i čiji se sadržaj prilagođava strukturi dizajna.

Pomoću klase *uvm_regblock* modeluje se skup svih registara unutar jedne kompletne komponente dizajna. Blok registara uvek stoji na vrhu hijerarhije modela i može instancirati registre, i modele memorije ukoliko postoje. U nekim slučajevima, blokovi mogu instancirati i druge pod-blokove radi veće enkapsulacije. Pri uspostavljanju registarskih blokova, vrši se izgradnja i mapiranje adresa za pojedinačne registre.

Primer koda jednog modela bloka je dat na slici 2. U definiciji klase, deklarisano je nekoliko polja koja predstavljaju buduće instance registara. Definisana je i metoda *build* koja definiše proces izgradnje modela. Ova metoda postoji u svim klasama RAL modela i poziva se rekurzivno, od višeg nivoa hijerarhije ka nižim, tako da jedan poziv ove metode sa vrha hijerarhije inicira kreiranje čitavog modela.

```
1  class example_reg_block extends uvm_reg_block;
2      `uvm_object_utils(example_reg_block)
3      reg_type_1 reg_1;
4      reg_type_2 reg_2;
5      ...
6      reg_type_n reg_n;
7
8      virtual function void build();
9          default_map = create_map("default_map", 'h0, 4,
10                                   UVM_LITTLE_ENDIAN, 1);
11         reg_1 = reg_type_1::type_id::create("reg_1", get_full_name());
12         reg_1.configure(this, null, "");
13         reg_1.build();
14         default_map.add_reg(reg_1, 'h0, "RW");
15         reg_2 = reg_type_2::type_id::create("reg_2", get_full_name());
16         reg_2.configure(this, null, "");
17         reg_2.build();
18         default_map.add_reg(reg_2, 'h4, "RW");
19         ...
20         reg_n = reg_type_n::type_id::create("reg_n", get_full_name());
21         reg_n.configure(this, null, "");
22         reg_n.build();
23         default_map.add_reg(reg_1, 'h4*num_of_regs, "RW");
24     endfunction
25 endclass
```

Slika 2. Primer modela bloka.

U implementaciji *build* metode registarskih blokova potrebno je izvršiti mapiranje adresa i instanciranje i konfiguraciju svih registara. Adresne mape se kreiraju i konfigurišu pomoću objekata ugrađene UVM klase *uvm_reg_map*. Ovaj objekat sadrži informacije o određenoj mapi kao što su početna adresa u prostoru, dužina adresne magistrale i format slaganja bitova koji može biti LSB ili MSB (*Least*, odnosno *Most Significant Bit*). Svaka klasa tipa *uvm_reg_block* sadrži jedno predefinisano polje *default_map* pomoću kog korisnik može izvršiti mapiranje, a po potrebi, mogu se kreirati i dodatni objekti ovog tipa u slučaju da model ima više adresnih prostora. Svaki registar u klasi se instancira i dodaje u adresnu mapu pozivom *add_reg* nad objektom mape, pri čemu se navode osobine registra – ime promenljive koja ga predstavlja, njegova podrazumevana vrednost, pravo pristupa i dr.

Registarskim blokovima mogu se modelovati blokovi dizajna koji sadrže jedan ili više različitih komunikacionih interfejsa za pristup pojedinim pod-blokovima sistema. U slučaju više interfejsa, moguće je u istoj klasi izvršiti više različitih mapiranja adresa registarskih polja za pojedinačne pod-blokove.

Fajl registara predstavlja enkapsulaciju skupa više modela registara koji odgovaraju jednoj jedinice dizajna. Po nameni su slični modelima registarskih blokova, a razlika je u tome što se fajlovi uobičajeno upotrebljavaju ukoliko dizajn sadrži više identičnih kopija instanci neke jedinice koje su deo nekog većeg bloka. Time se postiže bolja iskorišćenost koda, pošto je dovoljno instancirati samo niz registarskih fajlova koji će imati onoliko

elemenata koliko postoji kopija date jedinice, umesto instanciranja ili pravljenja posebnih klasa za svaku kopiju ponaosob. Fajlovi se mogu instancirati unutar blokova i drugih fajlova. Ukoliko ne postoje duplikati jedinica, nije obavezno koristiti fajlove registara već se hijerarhija može ostvariti samo pomoću modela blokova i registara.

Sami registri se modeluju korišćenjem klase *uvm_reg*, gde se za svaki registar definiše posebna klasa koja nasleđuje ovu osnovnu klasu. Primer koda jednog modela registra je dat na slici 3. Definicija jedne registarske klase sadrži promenljive tipa *uvm_reg_field* koje predstavljaju kontrolna, statusna ili komandna polja svakog registra. U telu *build* metode, vrši se konfiguracija svakog polja koja obuhvata dužinu registra u bitovima, slaganje bitova, tip pristupa polju (RW, RO, WO), itd.

```
class example_reg extends uvm_reg;
    `uvm_object_utils(example_reg)
    rand uvm_reg_field field_1;
    rand uvm_reg_field field_2;
    ...
    rand uvm_reg_field field_n;

    virtual function void build();
        this.field_1 = uvm_reg_field::type_id::create("field_1");
        this.field_1.configure(this, 4, 1, "RW", 0, 1'h0, 1, 1, 1);
        this.field_1.set_compare();

        this.field_2 = uvm_reg_field::type_id::create("field_1");
        this.field_2.configure(this, 8, 5, "RW", 0, 1'h0, 1, 1, 1);
        this.field_2.set_compare();
        // ...
    endfunction
endclass
```

Slika 3. Primer modela registra

Definisanje klasa modela svih registara i njihovo grupisanje u blokove vrši se najčešće u okviru jednog *SystemVerilog* fajla. Registarski model se vrlo jednostavno integriše u verifikaciono okruženje instanciranjem objekta klase vrhovnog bloka i pozivom metode *build* u fazi uspostavljanja okruženja (*build_phase*). Iz okruženja, model se može učiniti dostupnim i drugim verifikacionim komponentama kao što su *scoreboard*-i ili objekti za analizu pokrivenosti testiranja (*coverage*).

III. AUTOMATSKO GENERISANJE

Iako RAL model značajno olakšava proces verifikacije, način na koji je struktuiran ga čini jako teškim za rukovanje ukoliko korisnik sam definiše klase. Ovo je posebno izraženo pri verifikaciji digitalnih sistema koji sadrže veliki broj registara. Ručno kreiranje koda modela je podložno širokom spektru problema:

- Očigledno, pisanje RAL koda za svaki pojedinačni registar bilo bi dugotrajan proces. Verifikacione metodologije teže ka tome da se što manje vremena u verifikaciji uloži u uspostavljanje i integraciju okruženja, a što više u pisanju testova i analizu njihovih rezultata.
- Model se kreira na osnovu specifikacije registara. Pri tome se podaci iz specifikacija prevode u klase modela. Tu veoma često dolazi do raznih grešaka kao što su previdi u sintaksi, nedoslednosti ili pogrešno uneta imena i svojstva registarskih polja.
- Svaka promena u specifikaciji oslikava se i na dizajn i na model. Ažuriranje registara, dodavanje novih ili izbacivanje registara može biti zahtevno.

U strukturi koda za modelovanje blokova klasama izvedenih od *uvm_reg_block*, kao i koda za modelovanje registara pomoću *uvm_reg*, mogu se uočiti šabloni. Registarski blokovi instanciraju sve registre i dodeljuju im odgovarajuće adrese u mapi. Svaki model registra deklariše polja koja ga čine i njegova svojstva i prava pristupa. Ovakva uniformnost u strukturi klasa daje dovoljno dobro definisana pravila za automatsko generisanje ovakvog koda.

U ovom radu, *Python* skripta koja je nazvana *build_ral.py* će kao ulazni parametar uzimati *Excel* tabelu u kojoj se nalazi specifikacija svih registara koji će biti poređani po vrstama, koristeći podatke zadate u tabeli, generisaće tekst u formatu *SystemVerilog* klasa a potom taj tekst sačuvati u zasebnom fajlu koji će kasnije moći da se ubaci u verifikaciono okruženje.

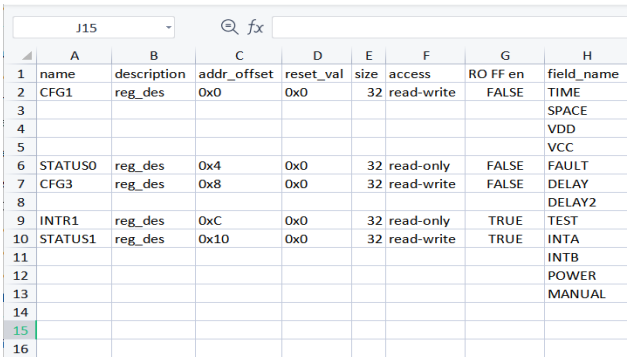
Pisanje specifikacije registara je jedini deo ovog procesa koji se mora odraditi ručno. Međutim, tabele su mnogo lakše za održavanje i ažuriranje od *SystemVerilog* koda pošto nemaju glomaznost strukture RAL klasa. Štaviše, od ovakvog pristupa pogodnosti imaju dizajn inženjeri pošto takođe postoje skripte koje se koriste za generisanje registara u dizajnu. Ovim se proces ažuriranja takođe olakšava pošto je sada dovoljno izvršiti sve promene samo u specifikaciji, a pokretanjem skripti će se automatski ažurirati i modeli i dizajn.

Specifikacija mora biti zadata u strogo određenom formatu kako bi skripta mogla precizno da očita sve vrednosti od značaja. Format specifikacije koji će ova skripta koristiti je dat u Tabeli 1, gde je definisano šta svaka kolona *.xls* fajla treba da sadrži. Na slici 4 se može videti jedan deo primera specifikacije koji sadrži opise registara *CFG1*, *STATUS0*, *CFG3*, *INTR1* i *STATUS1*. Primeri su ilustrativni i nisu korišćeni u realnom projektu.

TABELA I
FORMAT SPECIFIKACIJE

Kolona	Značenje
A	Ime registra
B	Opis registra
C	Adresni ofset registra
D	Podrazumevana vrednost (u *reset* stanju)
E	Veličina registra u bitovima
F	Pravo pristupa registru
G	Reset pri čitanju (*Clear-on-read* registar)
H	Imena polja u registrima
I	Opis polja
J	Ofset polja od adrese registra
K	Podrazumevana vrednost polja
L	Maska reseta
M	Širina polja

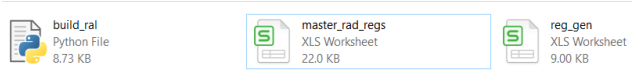
N	Indikator da li je polje *volatile* ili ne
O	Pravo pristupa polju



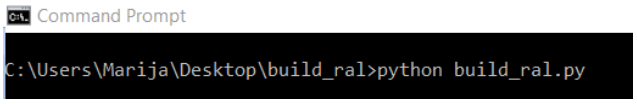
	A	B	C	D	E	F	G	H
1	name	description	addr_offset	reset_val	size	access	RO FF en	field_name
2	CFG1	reg_des	0x0	0x0	32	read-write	FALSE	TIME
3								SPACE
4								VDD
5								VCC
6	STATUS0	reg_des	0x4	0x0	32	read-only	FALSE	FAULT
7	CFG3	reg_des	0x8	0x0	32	read-write	FALSE	DELAY
8								DELAY2
9	INTR1	reg_des	0xC	0x0	32	read-only	TRUE	TEST
10	STATUS1	reg_des	0x10	0x0	32	read-write	TRUE	INTA
11								INTB
12								POWER
13								MANUAL
14								
15								
16								

Slika 4. Odlomak iz tabele specifikacija registara.

Na slici 5, prikazana je struktura foldera u kojoj se nalazi *build_ral.py* skripta. Potrebno je da se sa skriptom u istom folderu nalazi i *.xls* fajl koji se procesira. Skripta se može pokrenuti iz komandne linije kao na slici 6. U komandi kao ulazni parametar je potrebno navesti željeni *.xls* fajl. Ako ovaj fajl nije naveden ili je navedeno pogrešno ime, postoji rezervni fajl *reg_gen.xls* sa kojim će skripta inače raditi.



Slika 5. Struktura radnog foldera skripte.



Slika 6. Komanda za pokretanje skripte.

Nakon pokretanja skripte, dobija se kod u novom fajlu. Primer dobijenog koda za registar *CFG3* dat je na slici 7. Ovaj registar prema specifikaciji treba da sadrži dva polja – *DELAY* i *DELAY2*, dužine je 32 a pravo pristupa je *read-write* (RW). Iz koda sa slike, vidi se da su sve ove stavke obuhvaćene. Skripta je takođe dodala i metode i direktive koji podrazumevano postoje u svakoj klasi u UVM-u.

```
class rdb_CFG3_c extends uvm_reg;
    `uvm_object_utils (rdb_CFG3_c)
    rand uvm_reg_field rdb_DELAY;
    rand uvm_reg_field rdb_DELAY2;

    function new (string name = "rdb_CFG3_c");
        super.new(name,32,UVM_NO_COVERAGE);
    endfunction

    virtual function void build();
        this.rdb_DELAY =
        uvm_reg_field::type_id::create(rdb_DELAY");
        this.rdb_DELAY.configure(this,10,0,"RW",0,1'h0,1,1,1);
        this.rdb_DELAY.set_compare();

        this.rdb_DELAY2 = uvm_reg_field::type_id::create("rdb_DELAY2");
        this.rdb_DELAY2.configure(this,16,16,"RW",0,1'h0,1,1,1);
        this.rdb_DELAY2.set_compare();
    endfunction
endclass
```

Slika 7. Primer rezultujućeg koda jednog registra.

IV. ZAKLJUČAK

Generalno gledano, UVM RAL poboljšava produktivnost, održivost i prenosivost okruženja za verifikaciju hardvera. Podstiče primenu najboljih praksi, standardizaciju i apstrakciju, omogućavajući efikasniju i efektivniju verifikaciju složenih hardverskih dizajna.

Automatizacija procesa generisanja RAL koda koristeći *Python* omogućava uštedu vremena i smanjenje grešaka koje mogu nastati prilikom ručnog kreiranja i održavanja RAL koda. Ovaj pristup omogućava brzo prilagođavanje promenama u specifikacijama registara i promoviše doslednost u različitim projektima. Takođe, pruža fleksibilnu osnovu za dalje prilagođavanje i proširenje generisanog RAL koda prema potrebama.

LITERATURA

[1] B. Šipetić, „Primena registarskih modela u toku verifikacije ASIC čipova UVM metodologijom“, Elektronski fakultet Niš, master rad, 2023.

[2] M. Andrejević Stošović, N. Radivojević, O. Stojanović, „Funkcionalna Verifikacija“, Elektronski fakultet u Nišu, 2023.

[3] Accellera, „UVM Class Reference Manual 1.2“, accellera.org

[4] M. Patel, B. Vora, „UVM RAL Model: Usage and Application“, internet portal *Design&Reuse*

[5] S. Rosenberg, „Register this! Experiences Applying UVM Registers“.

[6] UVM Overview, Verification Guide. Dostupno na: https://verificationguide.com/uvm-ral/uvm-ral-overview/

Programiranje igre Moj broj

Pavle Stanković

Sadržaj – U ovom radu prikazan je proces izrade aplikacije čiji cilj je rešavanje igre Moj broj iz popularnog kviza „Slagalica“. Kod je napisan u programskom jeziku C++ koristeći razne metode o kojima je pisano u nastavku rada. Ovaj kod nalazi primenu u programima koji za cilj imaju rešavanje igre Moj broj, u *softveru* koji se koristi u kvizu, ali i u svim aritmetičkim zadacima koji zahtevaju pravljenje ispisa. Takođe može se koristiti i u drugim kvizovima kao što je engleski kviz „*Countdown*“.

I. Uvod

Kviz „Slagalica“ [1] je popularna televizijska emisija koja testira znanje takmičara u različitim oblastima. U ovoj uzbudljivoj igri, takmičari se takmiče u rešavanju raznih zagonetki, rebusa, matematičkih problema i mnogih drugih izazova. Cilj je da se brzo i tačno odgovori na postavljeno pitanje i osvoji što više poena.

Igra „Moj broj“ je druga po redu igra u najpopularnijem domaćem TV kvizu „Slagalica“. Cilj ove igre je da se od 6 ponuđenih brojeva, korišćenjem osnovnih matematičkih operacija („+“, „-“ , „*“, „/“) i zagrada, dođe do što približnijeg broja traženom broju. Svi brojevi se biraju nasumično kada takmičar izgovori reč *stop*. Ponuđeni brojevi se sastoje od 4 jednocifrenih brojeva, jednog broja iz skupa {10, 15, 20, 25} i jednog broja iz skupa {25, 50, 75, 100}. Svaki od ovih brojeva može se iskoristiti samo jednom ili nijednom.

Cilj ovog rada je izrada programa koji će od ponuđenih brojeva sastaviti izraz koji nakon izračunavanja daje traženi broj, ili ispisuje da to nije moguće. Konkretno ovaj program ima mogućnost unosa broja ponuđenih elemenata, te se može koristiti za proizvoljan broj cifara, a ne samo 6 kao u standardnom kvizu.

Program je napisan u programskom jeziku C++ u vidu konzolne aplikacije. Korisnik u njoj može unositi razne kombinacije brojeva, nevezano sa pravilima kviza. Aplikacija prestaje sa radom kada se unese cifra 0 u ulaz.

II. Metoda rada programa

U ovom poglavlju detaljnije su opisane metode i znanja koje se koriste za rešavanje datog problema.

A. Rešenje problema primenom grube sile

Rešenje koje intiutivno dolazi pri izradi ovog programa bazira se na algoritmu grube sile. Gruba sila je jedan od najosnovnijih algoritamskih pristupa koji se koristi u različitim oblastima računarstva. Ova metoda se odlikuje iscrpnim pregledom svih mogućnosti kako bi se pronašlo traženo rešenje. U kontekstu programiranja igara, grubom silom se obično isprobavaju sve moguće kombinacije ili akcije kako bi se pronašlo optimalno ili zadovoljavajuće rešenje. Iako može biti vremenski zahtevna i neefikasna u slučaju velikih skupova podataka, metoda grube sile je jednostavna za implementaciju i često se koristi kao početna tačka za razumevanje problema i generisanje ranih prototipa rešenja. Prednost grube sile je pružanje tačnog rezultata jer obuhvata sve moguće kombinacije, a mana je vremenska neefikasnost, te ovaj metod nije adekvatan za dati problem.

Prednost kod rešavanja problema metodom grube sile je ta što se lako može primetiti moguća optimizacija, i to iskoristiti kao ideju za dalje građenje algoritma. Primetimo sledeće – svako rešenje će se obraditi više puta, jer prilikom operacija sabiranja i množenja važi komutativnost. Komutativnost je svojstvo pri kojem mesto ne menja vrednost izraza. U našem konkretnom primeru bitno je primetiti sledeće: a+b=b+a i a*b=b*a.

Ovo zapažanje nas vodi do toga da treba primeniti algoritam odsecanja.

Odsecanje je tehnika koja se koristi u programiranju radi izbegavanja nepotrebnog izvršavanja operacija za koje smo sigurni da ne vode do rešenja. Sve mogućnosti koje se dobiju grubom silom možemo zamisliti kao stablo sa kojeg odsecamo čvorove koji zbog komutativnosti imaju istu vrednost korišćenjem istih brojeva. Takođe još jedno bitno odsecanje možemo uraditi kada nađemo rešenje. Ukoliko nam je cilj isključivo da dođemo do tačnog rešenja jednačine kada dođemo do istog možemo prekinuti traženje rešenja. Nema potrebe za daljim radom programa jer nam je potrebno samo jedno. U nekim primerima možemo doći do rešenja na više desetina načina, a nama je potrebno samo jedno, stoga možemo prekinuti izvršavanje programa i izbeći nepotrebno vremensko opterećenje.

Treba primeniti i dva bitna odsecanja, odsecanje deljenja pri kojima se ostatak pri deljenju razlikuje od nule i odsecanje oduzimanja većeg broja od manjeg.

B. Dinamičko programiranje

Primetimo da primenom metode grube sile vrednost nekih izraza računamo više puta. To možemo izbeći tako što ćemo, kada jednom izračunamo izraz, zapamtiti njegovu vrednost, i kasnije ga koristimo za dalje rešavanje problema. Ovaj proces se zove memoizacija i na njemu se zasniva dinamičko programiranje.

P. Stanković, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: pavlestankovic696@gmail.com

Dinamičko programiranje je algoritamska tehnika koja se koristi za rešavanje problema tako što se problem deli na manje, jednostavnije potprobleme i rešenja tih potproblema se pamte i koriste kasnije prilikom rešavanja većeg problema. Ova tehnika je efikasna kada se problem može rešiti kroz kombinaciju optimalnih rešenja manjih potproblema.

U našem programu dinamičko programiranje možemo iskoristiti tako što ćemo sve izraze pamtiti i pamtiti njihovu vrednost. Kada kasnije bude trebalo da se taj izraz iskoristi mi ćemo samo da „nalepimo“ taj izraz a iz memorije ćemo izvući vrednost istog. Primetimo da se ovim opterećuje memorija, ali nama je glavni cilj vremenska efikasnost te će nam ovaj algoritam značiti u daljem rešavanju.

C. Podeli pa vladaj

Podeli pa vladaj je algoritamska strategija koja se koristi za rešavanje složenih problema tako što se problem deli na manje, jednostavnije potprobleme, rešavaju se nezavisno, a zatim se kombinuju rešenja potproblema kako bi se dobio konačan rezultat.

Ova strategija se sastoji iz tri koraka: podela, vladavina i kombinacija. Prvo, problem se deli na manje potprobleme koji su slični originalnom problemu, ali manjih dimenzija. Zatim se svaki potproblem rešava rekurzivno, obično tako što se primenjuje isti algoritam na svaki potproblem zasebno. Na kraju, rešenja potproblema se kombinuju kako bi se dobio konačan rezultat.

Podeli pa vladaj je posebno efikasna strategija za rešavanje problema koji se mogu lako podeliti na manje potprobleme i za koje se može lako izvršiti kombinacija rešenja potproblema.

Primena strategije podeli pa vladaj u algoritmu za rešavanje igre „Moj broj“ donosi značajne koristi u efikasnosti rešavanja problema. Algoritam se može organizovati u skladu s ovom strategijom tako što se problem deli na manje potprobleme koji se rešavaju nezavisno, a zatim se kombinuju rešenja potproblema kako bi se dobio konačan rezultat.

U konkretnom slučaju, podela se može postići tako što se generišu sve moguće kombinacije brojeva i operacija koje se koriste u izračunavanju ciljnog broja. Ove kombinacije predstavljaju manje potprobleme koji se mogu rešavati nezavisno. Zatim se za svaki potproblem izračunava vrednost izraza koristeći dinamičko programiranje, gde se memorisane vrednosti koriste za ubrzavanje izračunavanja. Primena pristupa podeli pa vladaj u ovom algoritmu omogućava efikasno rešavanje igre „Moj broj“ tako što se problem razbija na manje potprobleme i rešenja tih potproblema se kombinuju.

III. OBJAŠNJENJE ALGORITMA

U kodu programa koriste se opisane tehnike, a neke od njih su dinamičko programiranje, podeli pa vladaj i gruba sila. Program se izvršava u funkciji „*main*“ u kojoj se poziva funkcija „*ucitaj*“ , koja ima strukturu bool. Struktura bool je vrsta podataka koja može imati samo dve vrednosti, „0“ ili „1“. U slučaju kada je vrednost jednaka „0“ računar smatra to kao netačan podatak, dok u suprotnom smatra da je tačan. Petlja *while* u funkciji „*main*“ ispituje vrednost koja se prva unela u funkciji „*ucitaj*“ i ukoliko ta vrednost nije jednaka nuli on će pozvati funkciju „*resi*“. Funkcija „*ucitaj*“ pomaže korisniku da unese sve brojeve ispisivajući mu na koji način to treba da uradi, i obaveštavajući ga da ukoliko unese „0“ rad programa se završava.

Funkcija „*resi*“ služi za rešavanje zadatog problema. U njoj se nalazi petlja *do-while*. Ona je prilično slična petlji *while*, u kojoj se stvari unutar petlje izvršavaju sve dok je uslov ispunjen, osim što se *do-while* petlja izvršava barem jednom bez obzira na uslov. Kod postavljanja ovakvih petlji bitno je da budemo pažljivi prilikom postavljanja uslova za izlaz iz petlje, jer u suprotnom petlja može da se izvršava beskonačan broj puta. Uslov za ovu petlju nam je bibliotetička funkcija *next_permutation*, koja za atribute ima iteratore na početak i kraj globalnog niza u kojem su ponuđeni brojevi. Ova funkcija najefikasnije pronalazi sledeću permutaciju brojeva menjajući im redosled.

Pre same *do-while* petlje sortira se niz bibliotetičkom funkcijom *sort*, jer je tako prirodnije posmatrati rad programa.

Pri svakom prolazu kroz petlju varijabli *mp* dodeljuju se vrednosti mape koja je definisana kao funkcija „*sve_vrednosti*“. Funkcija „*sve_vrednosti*“ pravi sve mogućnosti koristeći tehniku podeli pa vladaj. Problemi se razbijaju na manje probleme, na manji broj brojeva i rešavaju se njihovi izrazi. U mapi *mp* se računaju sve mogućnosti sa ovim operacijama, bez ponovnog računanja izraza.

U ovoj fukciji se nalazi *for* petlja koja poziva funkciju „*sve_vrednosti*“ za sve vrednosti sa leve i sa desne strane izraza. Sve moguće vrednosti sa leve strane izraza čuvamo u mapi *mpA*, dok sve moguće vrednosti desne strane izraza čuvamo u mapi *mpB*. Njihove vrednosti su sačuvane kao ključ u mapi, dok su izrazi sačuvani kao vrednost. Računamo sve moguće vrednosti dobijene operacijama za svaka dva izraza dobijena ugnežđenom petljom. Pri svakom računanju ključa mape kao vrednost dajemo funkciju objedini.

Funkcija objedini služi za objedinjavanje dva izraza u jednom, u kojem se između njih nalazi operand, a oko njih zagrade. Njena jedina funkcija je vraćanje vrednosti izraza dobijenog spajajući dva izraza.

Na kraju u funkciji „*resi*” ispitujemo da li smo dobili rešenje koje smo tražili, i ukoliko je odgovor da prekidamo dalje traženje i ispisujemo izraz koje to rešenje u mapi ima kao vrednost. Kada se ispitaju sve permutacije za koje tražimo rešenje, ukoliko nismo našli rešenje aplikacija ispisuje da nema rešenja za dati izraz.

IV. REZULTATI TESTIRANJA

U narednom poglavlju date su realne slike iz igre „Moj broj“, testiran kod na tim zadatim vrednostima na sajtu „*repl.it*“ i upoređivana vremenska složenost.



Slika. 1. Primer zadatih brojeva u kvizu „Slagalica“. Foto: Youtube printscreen [2]

Rešenje: (((1 * 2) + ((7 + 100) * 10)) / 4). Vreme izvršavanja programa: 0.006676 s



Slika. 2. Primer zadatih brojeva u kvizu „*Countdown*“. Foto: Youtube printscreen [3]

Rešenje: ((2 + ((3 + 9) * (5 + 50))) - 4). Vreme izvršavanja programa: 0.01006 s

V. ZAKLJUČAK

U zaključku, ovaj kod predstavlja implementaciju rešenja za igru „Moj broj“ koristeći strategiju podeli pa vladaj. Program omogućava korisniku unos brojeva i ciljnog broja, a zatim koristi podelu problema na manje potprobleme kako bi generisao sve moguće kombinacije brojeva i izračunao rezultate korišćenjem osnovnih aritmetičkih operacija.

Korišćenjem rekurzivne funkcije i mapiranja vrednosti, program pretražuje sve moguće kombinacije brojeva i izračunava njihove vrednosti. Ako se pronađe kombinacija koja daje traženi ciljni broj, rezultat se prikazuje korisniku. U suprotnom, ukoliko nijedna kombinacija ne zadovoljava ciljni broj, ispisuje se poruka „Ne postoji rešenje!“.

Ovaj kod demonstrira primenu podeli pa vladaj strategije u rešavanju problema koji zahteva kombinatornu analizu i pretragu svih mogućih kombinacija. Kroz deobu problema na manje potprobleme i kombinovanje njihovih rešenja, program uspešno generiše i proverava sve moguće kombinacije i pronalazi rešenje ako postoji.

Ovaj kod nalazi primenu u programima koji za cilj imaju rešavanje igre „Moj broj“, u softveru koji se koristi u kvizu, ali i u svim aritmetičkim zadacima koji zahtevaju pravljenje ispisa. Takođe može se koristiti i u drugim kvizovima kao što je engleski kviz „*Countdown*“ koji pomoću zadatih brojeva dolazi do nule sličnim principom kao u domaćoj igri „Moj broj“.

Ukupno gledano, ovaj kod pruža osnovu za rešavanje igre „Moj broj“ kroz primenu podeli pa vladaj strategije i demonstrira važnost razumevanja strategija rešavanja problema u programiranju.

LITERATURA

[1] „O kvizu TV Slagalica“ , Wikipedia, 2018. [Online]. Dostupno na: https://sh.wikipedia.org/wiki/TV_Slagalica
[2] Primer zadatih brojeva u kvizu „Slagalica“. Dostupno na: https://www.youtube.com/watch?v=vD42DkvF8xg&t=361s&ab_channel=TVBiseri
[3] Primer zadatih brojeva u kvizu „*Countdown*“. Dostupno na: https://www.youtube.com/watch?v=WslHCziT4xM&t=19s&ab_channel=BjoernGumboldt

Generisanje građevina u računarskoj igri „Minecraft” korišćenjem Shap-e modela

Marko Milenković

Sadržaj - Napravljena je praktična implementacija Shap-e modela kao nadogradnja računarske igre „Minecraft”. Konačni rezultati su izgenerisane građevine na osnovu tekstualnog upita, sa ciljem da se smanji količina vremena koju profesionalne ekipe igrača troše na građevinske projekte u ovoj računarskoj igri. Primenjeno je uzorkovanje STF funkcije za pravljenje mreže voxel-a. Nakon toga je urađeno smanjeno uzorkovanje da bi se uklonile anomalije i praznine. Za sva računanja i poređenja vezana za boje i teksture je korišćen CIE lab prostor boja zbog boljeg preklapanja sa ljudskom percepcijom boja i linearnosti u poređenju sa RGB prostorom boja. Da bi se napravilo mapiranje boja na blokove korišćen je median cut algoritam za pronalaženje boje koja najbolje predstavlja blok. Ta predstavljajuća boja se koristi za računanje euklidske distance boja u TF od Shap-e objekta i blokova u igri, time dobijajući podudarno predstavljanje boja Shap-e generisanog objekta unutar igre korišćenjem Minecraft blokova. Konačni rezultat je zadovoljavajući, uzevši u obzir ograničenja Shap-e generativnog modela, i pokazana je praktična primena Shap-e modela.

I. Uvod

Minecraft je poznata računarska igra. Spada u kategoriju *voxel* igara [1]. Igra nema predodređen početak i kraj i pruža igraču mnogo različitih načina igranja igre, pa se zato naziva *sandbox* igrom.

Vremenom su nastale ekipe igrača koje su krenule isključivo da se bave građenjem struktura, okruženja i mapa. Rastom popularnosti igre rasla je i veličina projekata. Građevine su se pravile isključivo ručno ili uz asistenciju alatki. Nakon korišćenja tih alatki je još uvek bilo potrebno ručno izmeniti detalje. Takav pristup je doveo do toga da je potrebno izuzetno mnogo vremena da se napravi neka struktura, pogotovo ako je velikih razmera.

Cilj istraživanja je da se napravi alatka kojom bi se ubrzalo ili automatizovalo pravljenje struktura. Time bi se igračima olakšao rad i smanjila bi se količina vremena potrebna za realizaciju projekata. Uz razvoj veštačke inteligencije su se kao moguće rešenje pokazali uslovno generativni algoritmi.

Za generisanje objekata korišćen je OpenAI uslovni generativni model za 3D objekte *Shap-e* [2]. Shap-e generiše skup parametara implicitno zadate funkcije [3] koja se može predstaviti kao model ili *neural radiance field* (NeRF) [4]. U ovom radu je korišćen teksturirani model.

M. Milenković, student 1. godine na Fakultetu tehničkih nauka, Trg Dositeja Obradovića 6, 21102, Novi Sad, Srbija/polaznik programa Računarstva u Istraživačkoj stanici Petnica, Selo Petnica, 14104 Valjevo, Srbija, E-mail: mmarkomile@gmail.com

Shap-e model je izabran jer u poređenju sa drugim uslovnim generativnim modelima [5][6][7][8] brzo izvršava generaciju, a pruža zadovoljavajuće rezultate. U poređenju sa *Point-e* modelom [8], koji ima sličnu primenu, brži je i proizvodi slične ili bolje rezultate.

Shap-e proizvodi SDF (*signed distance function*) [9] i TF (*texture field*) [10] koji se koriste u uzorkovanju funkcije da bi se napravila teksturizovana mreža *voxel*-a.

Da bi se boje objekta koji Shap-e izgeneriše uvezle u Minecraft, potrebno je napraviti mapiranje boja na blokove. Najjednostavniji način da se ovo uradi je uporediti euklidske distance RGB [11] vrednosti i prosečne boje teksture svakog bloka u Minecraft-u. Međutim, ovaj pristup ne naglašava preciznost i ne poklapa se dovoljno sa ljudskom percepcijom sličnosti boja. Zato je korišćen CIE lab [12] prostor boja [13] za mapiranje boja na blokove i *median cut algoritam* [14] za pronalaženje predstavljajućih boja blokova. Uveden je i parametar standardne devijacije [15] nad teksturama blokova. Standardnom devijacijom može se predstaviti koliko je neki blok šaren. Šareni blokovi su neželjeni jer nemaju jasno definisanu boju koja ih predstavlja.

II. Priprema za obradu

A. Eliminacija nepoželjnih blokova

Da bi objekti izgledali prihvatljivo u Minecraft-u, potrebno je eliminisati određene grupe blokova iz selekcije. Zbog toga su uvedeni različiti kriterijumi: standardna devijacija boja, providnost i da li je taj blok dekorativan.

Pojedinačni blokovi su takođe eliminisani. To su problematični blokovi kao što su vrata (koja se rasprostiru preko 2 koordinate), korali (jer izgube boju kada su na suvom) i blokovi sa gravitacijom.

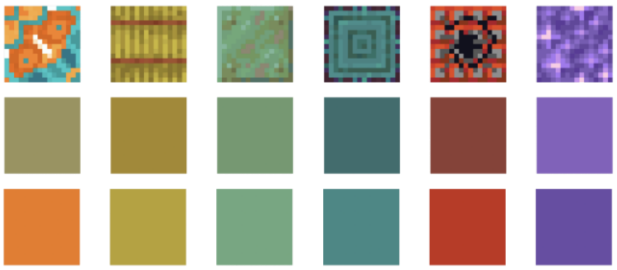
Polublokovi, zidovi i stepenice nisu ubačeni u selekciju. Za sada je u Minecraft-u prilično ograničena raznovrsnost takvih blokova i postoji mali broj različitih boja i materijala za njih. Zbog toga nisu jako značajni.

B. Mapiranje blokova na boje

Mapiranje blokova na boje i kasnije upoređivanje boja na izgenerisanoj TF sa blokovima koji najbliže odgovaraju je jedan od najbitnijih koraka. Korišćen je CIE lab prostor boja.

Iako računanje prosečne boje teksture bloka može u određenim uslovima pokazati zadovoljavajuće rezultate, to

nije dobar način da se traži boja koja najbolje predstavlja neku sliku. *Median cut* algoritam može da pronađe boju koja najbolje predstavlja sliku. Vidi se značajna razlika kada gledamo predstavljajuću boju šarenih blokova datoj u primeru prikazanom na slici 1.



Slika 1. Neki primeri blokova (prvi red), njihovih predstavljajućih boja dobijenih prostim prosekom (drugi red) i korišćenjem *median cut* algoritma (treći red).

Posterizacija slike je pretvaranje slike sa velikim brojem tonova u sintetičku sliku sa izraženim ravnim površinama i manjim brojem tonova. *Median cut* algoritam je jedan od algoritama za posterizaciju koji pronalazi ograničenu paletu boja kojom se posterizuje slika tako da najviše liči na original. Algoritam je iskorišćen da se odabere najučestalija boja od dobijenih boja u paleti teksture nekog bloka, da bi se dobila boja koja najbolje predstavlja taj blok. Da bi se mapirao svaki blok, algoritam se pokreće na svakoj od tekstura blokova koji se koriste pojedinačno. Od parova predstavljajućih boja i naziva blokova se pravi mapiranje koje se dalje koristi za upoređivanje sa bojama iz TF.

Gradijenti između boja imaju znatno blaži prelaz u CIE lab prostoru boja u poređenju sa RGB prostorom boja i više se poklapaju sa ljudskom percepcijom. Ovo se može videti na grafičkom prikazu gradijenata boja (slika 2).

Zbog toga je za svako računanje i poređenje korišćen CIE lab prostor boja. Prebacivanje je odrađeno korišćenjem *skimage* biblioteke koja prebacuje sRGB boje u CIE XYZ (koji je derivat CIE RGB prostora boja) prostor boja [16][17]. Postupak počinje normalizacijom boja u vrednosti od 0 do 1 (deljenjem sa 255). Zatim se vrši konverzija sRGB→XYZ.

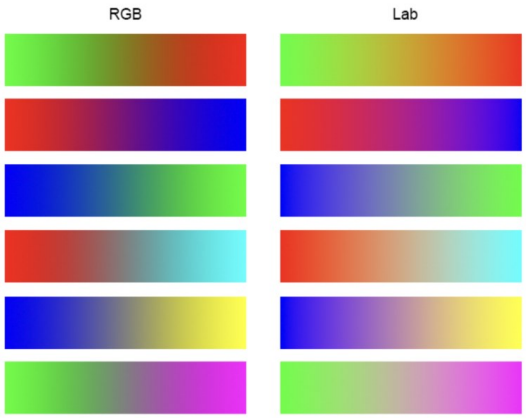
CIE XYZ je potrebno prebaciti u CIE lab prostor boja. Da bi se to odradilo mora se definisati *illuminant* (osvetljivač) [18]. Osvetljivač predstavlja teoretski izvor svetlosti i time daje osnovu za poređenje slika. U ovom konkretnom slučaju korišćen je uobičajeni osvetljivač D_{65} jer predstavlja dnevnu svetlost u podne (6500K). Kada je izabran osvetljivač, moguće je odraditi konverziju CIE XYZ→CIE lab.

III. Obrada

A. Shap-e generacija

Shap-e model je obučen da na osnovu tekstualnog upita koji je zadat na prirodnom jeziku izgeneriše objekat koji je opisan u upitu. S obzirom da je skup podataka koji je korišćen za obučavanje Shap-e modela uglavnom sačinjen iz objekata koji su poprilično jednostavni, najbolje rezultate daje kada su i tekstualni upiti jednostavni.

Iz izlazne vrednosti modela se izdvajaju SDF i TF korišćenjem funkcija predstavljenih uz Shap-e [2]. Da bi se SDF i TF iskoristile u Minecraft-u potrebno je od njih napraviti mrežu *voxel*-a.



Slika 2. Linearna interpolacija od leve do desne boje u RGB i CIE lab prostorima boja.

B. Uzorkovanje funkcije

Da bi se dobila mreža *voxel*-a, odrađeno je uzorkovanje funkcije. Pravi se binarna trodimenzionalna matrica, gde je svaki element indikator postojanja *voxel*-a na datoj koordinati iz SDF. Ako je vrednost u SDF na datim koordinatama >=0 onda se postavlja na 1.

C. Smanjeno uzorkovanje

Zbog pojave veoma tankih elemenata u objektima, ponekad se može desiti da nastanu rupe u konačnom objektu kada se on prebaci iz SDF u mrežu *voxel*-a. Jedan od metoda rešavanja ovog problema je smanjeno uzorkovanje sa veće rezolucije.

Uzorak iz SDF je uzet u 3 puta većoj rezoluciji od konačne (željene) rezolucije, zatim je urađeno sažimanje pomoću srednje vrednosti [19] nad tom matricom sa dimenzijama jezgra 3x3x3. Posledica je trodimenzionalna matrica željenih dimenzija. Ovako je smanjena količina nepravilnosti u konačnom rezultatu uz minimalan uticaj na performanse.

Smanjeno uzorkovanje primenjeno nad teksturama ne utiče previše na konačan rezultat, pa iz tog razloga nije rađeno nad teksturama.

D. Postavljanje blokova u Minecraft-u

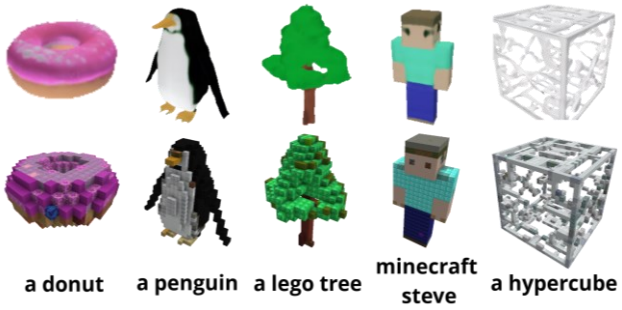
Poslednji korak je postavljanje blokova u Minecraft-u. Ovo je implementirano jednostavnom Python skriptom

koja se povezuje na Minecraft server korišćenjem RCON port-a i poziva komandu „/setblock X Y Z minecraft:block”. X, Y i Z su koordinate gde se postavlja blok, a *minecraft:block* je blok koji treba da se postavi na osnovu vrednosti boje na tim koordinatima. Da bi se našao blok koji će da se postavi na osnovu boje iz TF na datim koordinatima, računamo euklidsku distancu za svaki od blokova. Koristi se blok čija predstavljajuća boja ima najmanju euklidsku distancu do proizvoljne boje iz TF.

IV. Rezultati i diskusija

A. Evaluacija rezultata

Za evaluaciju rezultata su upoređene izlazne slike Shap-e modela i odgovarajuća građevina u Minecraft-u. Može se videti izrazita sličnost (slika 3).



Slika 3. Prikaz različitih primera kao Shap-e objekat (prvi red) i u Minecraft-u (drugi red) zajedno sa upitima.

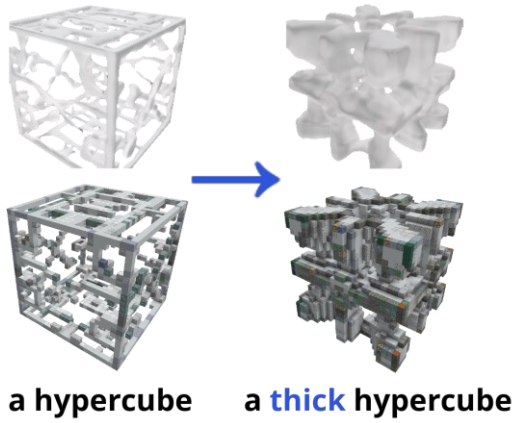
Vidi se da u kompleksnim strukturama kao što je „hypercube” nastaju problemi u generaciji Shap-e modela.

Nekada je potrebno prilagoditi upit da bi objekat izgledao kao što je zamišljeno i da bi se adekvatno prebacio u Minecraft. Može biti korisno dodavati pojmove kao „lego” (slika 3) kako bi izgled izgenerisanog objekta bio bolje prilagođen za prebacivanje u Minecraft. Iz *hypercube* primera može se lako zaključiti da postoji nekakav problem, pa je jedno od mogućih rešenja sakriveno u menjanju upita. Za primer razlike pri menjanju upita je dodat pridev „thick”, koji značajno menja rezultat, što se vidi na slici 4.

Razlika između korišćenja RGB prostora boja i CIE lab prostora boja je veoma značajna. Može se videti da je CIE lab prostor boja mnogo sličniji onome što bi se očekivalo i nema velikih skokova boje od svetlije do tamnije strane spektra (slika 5).

Sličnost građevina i objekata iz modela je zadovoljavajuća. Primećuje se sličnost boja i nema ozbiljnih artefakta na jednostavnim upitima.

Vreme potrebno da Shap-e izgeneriše model se kreće oko 20-30 sekundi na Nvidia Tesla T4 grafičkoj. Ovakvi rezultati se mogu smatrati dovoljno dobrim za praktičnu upotrebu u igri.



Slika 4. Razlika pri menjanju upita.



Slika 5. Poređenje rezultata kada se koristi RGB i kada se koristi CIE lab prostor boja.

V. Ograničenja i budući rad

Trenutni uslovno generativni modeli za generisanje trodimenzionalnih objekata imaju brojna ograničenja. Moguće ih je podeliti na „brze” [8] i „spore” [5][6][7]. Brzi modeli još uvek podržavaju samo jednostavne upite i ne raspoznaju komplikovanije opise i strukture. Iako modeli razumeju jednostavne opise, ograničeno podržavaju kompoziciju pojmova. Spori modeli nisu praktični za primene u igrama ili u *real-time* scenarijima, kada je potrebno da se brzo vizuelizuje rezultat.

U ovom radu nisu razmatrani nepravilni blokovi (polublokovi, stepenice, zidovi i dekorativni blokovi), ali bi bilo moguće implementirati ih u budućem radu uz nekakav sistem pravila i marching cubes algoritmom umesto uzorkovanja funkcije. Marching cubes algoritam bi mogao biti prilagođen Minecraft blokovima. Ovo bi takođe omogućilo jednostavno dodavanje podrške za nove blokove u budućnosti.

VI. Zaključak

Shap-e algoritam pruža zadovoljavajuće rezultate kada se njegovi izlazni objekti pretvore u Minecraft građevine. Napravljen je način da se ubrzaju građevinski projekti u Minecraft-u i da se igračima olakša građenje određenih objekata. Pokazano je da savremeni uslovno generativni modeli pored svojih ograničenja mogu da naprave zadovoljavajuće rezultate u kratkom vremenskom intervalu i pronađu praktičnu primenu.

LITERATURA

[1] "Tutorials/units of Measure: Distance," Minecraft Wiki, https://minecraft.wiki/w/Tutorials/Units_of_measure#Distance (pristupljeno 14. oktobra 2023).

[2] Jun, H. and Nichol, A., 2023. Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463.

[3] Krantz, S.G. and Parks, H.R., 2002. The implicit function theorem: history, theory, and applications. Springer Science & Business Media.

[4] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R. and Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), pp.99-106.

[5] Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y. and Lin, T.Y., 2023. Magic3d: High-resolution text-to-3d content creation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 300-309.

[6] Poole, B., Jain, A., Barron, J.T. and Mildenhall, B., 2022. Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988.

[7] Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z. and Fidler, S., 2022. Get3d: A generative model of high quality 3d textured shapes learned from images. Advances In Neural Information Processing Systems, 35, pp.31841-31854.

[8] Nichol, A., Jun, H., Dhariwal, P., Mishkin, P. and Chen, M., 2022. Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751.

[9] Chan, T. and Zhu, W., 2005, June. Level set based shape prior segmentation. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, pp. 1164-1170.

[10] Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T. and Geiger, A., 2019. Texture fields: Learning texture representations in function space. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4531-4540.

[11] Süsstrunk, S., Buckley, R. and Swen, S., 1999. Standard RGB color spaces. In Proc. IS&T;/SID 7th Color Imaging Conference (Vol. 7, No. CONF, pp. 127-134).

[12] "CIE 1976 (L *a*b*) colour space," u CIE Colorimetry 15 (Third ed.), CIE, 2004, ISBN 3-901-906-33-9.

[13] Joblove, G.H. and Greenberg, D., 1978, August. Color spaces for computer graphics. In Proceedings of the 5th annual conference on Computer graphics and interactive techniques (pp. 20-25).

[14] Heckbert, P., 1982. Color image quantization for frame buffer display. ACM Siggraph Computer Graphics, 16(3), pp.297-307.

[15] Bland, J.M. and Altman, D.G., 1996. Measurement error. BMJ: British medical journal, 312(7047), p.1654.

[16] International Commission on Illumination. (1932). Commission internationale de l'éclairage huitième session cambridge--septembre 1931. University Press.

[17] T S. & J G. (1931). The c.i.e. colorimetric standards and their use. Transactions of the Optical Society 73–134. https://doi.org/10.1088/1475-4878/33/3/301.

[18] Noboru, O. and Robertson, A.R., 2005. 3.9: Standard and Supplementary Illuminants, Colorimetry. Wiley Hoboken, NJ, USA. https://doi. org/10, 1002(04700), p.94745.

[19] "Papers with code - average pooling explained," Papers With Code, https://paperswithcode.com/method/average-pooling (pristupljeno 15. oktobra 2023).

Uporedna analiza algoritama za generisanje realističnih pahuljica

Emilija Sredojević

Sadržaj - U ovom radu prikazuju se tri metoda za generisanje pahuljice gde se akcenat stavlja na raznolikost pahuljica i sličnost sa pahuljicama iz prirode. Prvi metod koristi algoritam za generisanje Kohove pahuljice sa slučajno odabranim uglom i položajem generatora. Drugi metod su L - sistemi sa definisanim pravilima i ograničenjima za generisanje nasumičnog prepisivačkog pravila. Aksiom se nasumično bira iz skupa definisanih aksioma. Treći metod je Rejterov model za ćelijski automat sa slučajno odabranim vrednostima konstanti difuzije, količine vodene pare u okolini i količine vode koja se dodaje određenom skupu ćelija.

I. Uvod

Pahuljice su grupe snežnih kristala koje nastaju u atmosferi od čestica leda. Svoj šestougaoni oblik duguju prostornom položaju veza vodonika sa kiseonikom. Put koji pahuljica pređe od postanka do pada na zemlju praćen je različitim fizičkim pojavama i procesima koji utiču na njen oblik. Mali varijeteti na tom putu mogu dovesti do znatnih razlika među pahuljicama [1,2].

Ideja ovog projekta je da se prikaže osobina jedinstvenosti pahuljica kroz tri metoda za generisanje: korišćenjem slučajno odabranog generatora za Kohovu pahuljicu, Rejterov model za ćelijski automat (*Clifford A. Reiter*) i L - sistemi (*Lindenmayer Systems*).

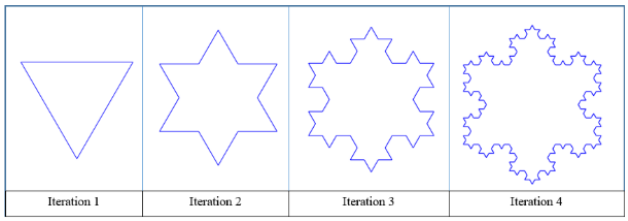
II. Metod

A. Generisanje Kohove pahuljice sa slučajno odabranim generatorom

Ovaj metod se zasniva na modifikaciji algoritma za generisanje Kohove pahuljice uvođenjem nasumičnog odabiranja ugla i položaja generatora. Generisanje obične Kohove pahuljice započinje jednakostraničnim trouglom gde svaka stranica predstavlja inicijator. Svaki inicijator menja se generatorom i tako od jednog inicijatora nastaju četiri nova, dužine trećine početnog inicijatora (slika 1). Postupak se nastavlja na isti način nad inicijatorima [3].

Modifikacija se vrši nad dvanaestinom pahuljice kako bi ona zadržala svoju prirodnu simetričnost. Nasumično se bira položaj generatora i ugao generatora.

Nasumično odabrani ugao generatora pripada skupu [25°, 120°] - [-120°, -25°]. Položaj generatora zavisi od znaka ugla. Ako je ugao negativan, generator će biti okrenut ka unutrašnjosti pahuljice, u suprotnom ka spoljašnjosti.



Slika 1. Koraci pri generisanju Kohove pahuljice.

B. Generisanje pahuljice upotrebom L - sistema

L - sistem je prepisivački model. L - sistemi su izvorno korišćeni za modelovanje ponašanja ćelija biljaka i za opisivanje razvoja jednostavnih višećelijskih organizama poput kvasca i algi [4]. Svaki L - sistem sastoji se od alfabeta, aksioma (početni niz simbola) i pravila [5]. Alfabet je skup svih simbola [6]. Osnovni alfabet korišćen prilikom izrade ovog projekta je sledeći:

- ‘F’ - crtaj napred za dužinu d,
- ‘f’ - pomeri se napred bez crtanja za dužinu d,
- ‘+’ - rotiraj se u levo za ugao φ,
- ‘-’ - rotiraj se u desno za ugao φ,
- ‘[‘ - prebaci trenutno stanje crtanja na stek,
- ‘]’ - uzmi sa steka trenutno stanje crtanja.

Osnovni alfabet proširen je dodatnim simbolima koji doprinose što većoj različitosti pravila. Dodatni simboli su ‘G’ i ‘H’. Prilikom pretvaranja simbola u grafički prikaz ova dva simbola se tumače isto kao simbol ‘F’.

Pravila definišu u koji će se niz simbola preslikati svaki od simbola u alfabetu. Simboli f, [,], + i - uvek će se preslikavati u sami sebe, pa se pravila definišu nad simbolima F, G i H. Prilikom svake iteracije svaki od simbola F, G i H preslikaće se u niz simbola određenim za njih odgovarajućim pravilima.

Kako bi se dobile nasumične pahuljice, vrši se randomizacija pravila i delimična randomizacija aksioma.

Radi što veće raznolikosti pahuljica početni aksiom se bira iz što većeg skupa. Nažalost, potpuna randomizacija aksioma često uzrokuje da pahuljice ne liče na pahuljice iz prirode. Zbog toga je pronađen skup ispavnih aksioma iz koga se za svaku generisanu pahuljicu bira jedan. Uočena je pravilnost da svaki aksiom mora biti oblika [X]*[X]*[X]*[X]*[X]*[X] gde je X neprazan skup simbola iz alfabeta i za njega važi X ∈ { F, FG, fF, ---fff+++F, ---

E. Sredojević je učenica 2. razreda Užičke gimnazije, trg Svetog Save 6, 31000 Užice, Srbija, E-mail: emasredojevic8@gmail.com.

fff+fff---F, GfF, FG, FfG, fff+fff-F} i * ∈ { +, - }. Aksiom mora biti ovog oblika zbog šestostrane simetričnosti pahuljice.

Da bi ovako generisane pahuljice sačuvale osobine iz prirode, postavljaju se ograničenja nad randomizacijom pravila. Ograničenja i pravila garantuju šestostranu simetriju i što realističniji izgled pahuljice.

Osim randomizacije simbola unutar pravila vršena je i randomizacija broja pravila i dužine svakog od njih. Generiše se jedno ili dva pravila. Veći broj pravila bi značajno uticao na brzinu algoritma. Broj znakova u pravilu nalazi se u intervalu [5, 25], osim u slučaju kada je za realizaciju pravila 1) potrebno više simbola. Broj simbola među uglastim zagradama takođe je ograničen i iznosi 9. Ovo ograničenje garantuje da će otvorena zagradi morati i da se zatvori.

Različiti simboli se u pravilu pojavljuju različit broj puta. Najviše ima simbola za crtanje (F, G i H), pa onda simbola za upravljanje stekom i tako dalje. Simboli su odabirani po određenoj raspodeli.

Raspodela korišćena u ovom radu je dobijena izračunavanjem procenta svakog simbola u već postojećim pravilima za pahuljicu i to su vrednosti prikazane u Tabeli I (u procentima):

TABELA I
RASPODELA SIMBOLA U PROCENTIMA

'F'	'G'	'H'	'f'	'['	']'	'+'	'-'
10	8	2	4	18	18	20	20

Verovatnoća zavisi i od definisanih ograničenja i pravila jer se neki simboli ne mogu koristiti u određenim uslovima.

C. Generisanje pahuljice korišćenjem Rejterovog modela

Ovaj model se koristi fizičkim procesima koji utiču na pahuljicu u toku njenog nastanka [7]. Mreža koju koristi sastoji se od pravilnih šestouglova, a jedan šestougao predstavlja ćeliju. Svaka ćelija z opisana je vrednošću količine vode koja se u njoj nalazi u trenutku t, što označavamo sa $s_t(z)$. U zavisnosti od vrednosti $s_t(z)$ ćelije mogu biti:

- zaleđene – ako je $s_t(z) \geq 1$;
- granične – ako ćelija nije zaleđena, ali ima zaleđenog komšiju;
- nereceptivne – ako ćelija nije ni zaleđena niti ima zaleđenog komšiju.

Zaleđene i granične ćelije zajedno se nazivaju receptivne.

Rejterov model započinje jednim kristalom leda $s_0(0) = 1$ sa početnom ćelijom 0. Za sve ostale ćelije $s_0(z) = \beta$, gde je β fiksna konstanta količine vodene pare u okolini.

Ćelije se opisuju i vrednostima $u_t(z)$ i $v_t(z)$ za trenutak t. Vrednost $u_t(z)$ označava količinu vode koja učestvuje, a $v_t(z)$ količinu vode koja ne učestvuje u difuziji. Za receptivne ćelije vrednost $u_t(z)$ u trenutku t biće 0, a $v_t(z)$ će biti jednaka $s_t(z)$. Za nereceptivne, obrnuto. Definišemo fiksne konstante γ – konstanta koja označava količinu vode koja se dodaje na $v_t(z)$, i α – koeficijent difuzije. Postoje dva načina da ćelija dobije vodu i njih definišu sledeća pravila:

- Pravilo konstantne adicije – za svaku ćeliju z količina vode koja ne učestvuje u difuziji konstantno se dodaje, tj.:

$$v_t^+(z) = v_t^-(z) + \gamma \quad (1)$$

- Pravilo difuzije – ovim pravilom definiše se količina vode koja učestvuje u difuziji posle svakog izvršenog koraka; koristi se diskretan oblik jednačine difuzije:

$$u_t^+(z) = u_t^-(z) + \frac{\alpha}{2}(\overline{u_t^-(z)} - u_t^-(z)) \quad (2)$$

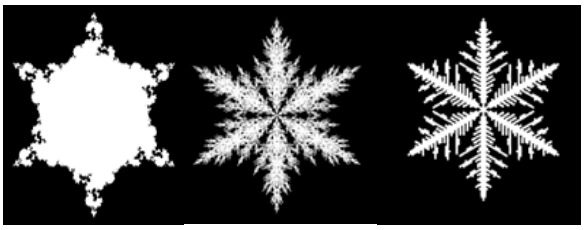
Kod ovog modela koncentrišemo se samo na dvanaestinu pahuljice kako bismo sačuvali njenu simetriju. Možemo da obrađujemo sve one tačke koje su najviše 30° udaljene od glavne koordinatne ose.

Da bi ćelija postala zaleđena mora da prođe kroz dva stanja. Na početku je nereceptivna i gubi vodenu paru preko difuzije. Kada postane granična sakuplja vodu sve dok ne postane zaleđena gore opisanim pravilima (difuzija i adicija). Zaleđene ćelije više ne mogu da sakupljaju vodu preko difuzije [8].

Za randomizaciju ovog modela dovoljno je da se randomizuju konstante α, β i γ. Vrednost za α se nasumično bira iz intervala [0.5, 2], za β iz intervala [0, 1], a za γ iz intervala [0.01, 0.4].

III. REZULTATI

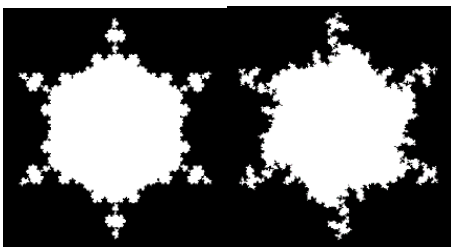
Na slici 2 redom su prikazani rezultati i to prvo pahuljica generisana modifikovanim algoritmom za Kohovu pahuljicu, zatim pahuljica generisana L – sistemima in a kraju pahuljica generisana Rejterovim modelom.



Slika 2. Pahuljica generisana modifikovanim algoritmom za Kohovu pahuljicu, pahuljica generisana L - sistemima i pahuljica generisana Rejterovim modelom.

A. Kohova pahuljica

Izgled Kohove pahuljice zavisi od nasumično odabranog ugla i položaja generatora. Ako se ugao ne bi birao nasumično već mu se dodelila fiksna vrednost od 60°, pahuljice bi više podsećale na pahuljice iz prirode, ali bi raznolikost pahuljica bila manja (slika 3). Pahuljice sa nasumično odabranim uglom manje podsećaju na prave, ali su pahuljice različitije.



Slika 3. Prva slika je pahuljica sa uglom 60°; druga sllika je pahuljica sa nasumično odabranim uglom.

B. L - sistemi

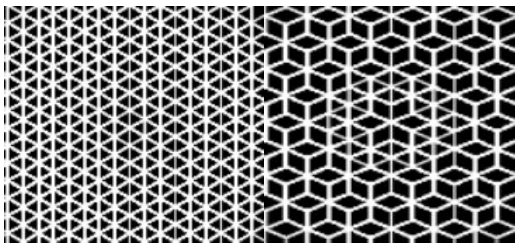
Ograničenja i pravila koja su uočena i korišćena u ovom radu su sledeća:

1) Ako u izgenerisanom pravilu postoji skup simbola oblika [X], gde je *X* skup simbola koji u sebi sadrži bar jedan od znakova '+' ili '-', odmah nakon njega u pravilu se mora nalaziti skup simbola oblika [X'], gde je *X'* skup simbola nastao tako što se svaki od simbola '+' ili '-' u *X* zamenio njemu suprotnom simbolu.
2) Ugao φ može biti vrednosti 15°, 30° ili 60°. Ako je ugao $\varphi = 15°$, broj simbola (pluseva ili minusa) između svakog para [X] u aksiomu biće jednak $\frac{60°}{15°} = 4$, a ako je ugao $\varphi = 30°$, taj broj će biti 2.
3) U pravilu simboli '+' i '-' ne smeju se naći jedan pored drugog.
4) Znakovi '+' i '-' u pravilu mogu se nalaziti samo između uglastih zagrada i nikada van njih.
5) Među uglastim zagradama mora se nalaziti neprazan skup simbola.
6) Broj nezatvorenih uglastih zagrada u jednom trenutku može biti najviše 3.
7) Pravilo mora da sadrži simbole '+' i '-'.

Uzroci za definisanje pravila:

1) Ako u jednom paru uglastih zagrada postoji neki od simbola '+' ili '-'to znači da će se grana pahuljice račvati u levu ili desnu stranu. Ovo pravilo je neophodno da bi se račvanje odvijalo na obe strane.
2) Pahuljica mora u sebi sadržati uglove čija vrednost deli 60 zbog simetrije pahuljce.
3) Svaki od uglova može se opisati samo jednim simbolom, pa je nepotrebno da se dva različita simbola nađu jedan pored drugog. Primer: '++-+' je jednako kao '++'.
4) Kada se simboli '+' i '-'nalaze među zagradama obezbeđeno je da će se pahuljice račvati na obe strane podjednako, u suprotnom nije. Još jedan od problema opisan je u sledećem primeru: Ako se u pravilu nađe skup simbola 'F+F-F+F' van zagrada, grafički prikaz ovakvog dela pravila bio bi kriva linija što nije svojstveno pahuljci.
5) Ako se među zagradama ne nalazi ništa onda one ne vrše nikakvu funkciju.
6) Što ima više uglastih zagrada u pravilu, pravilo je kompleksnije i duže. Dugačka pravila često daju pahuljice koje su guste i ne podsećaju na pahuljice iz prirode, a vreme za izvršavanje algoritma se značajno povećava.
7) Bez simbola '+' i '-' u pahuljici ne postoji račvanje kraka što nije osobina pahuljica iz prirode.

Izgled pahuljice značajno zavisi od slučajnog odabira ugla. Pahuljice čiji je ugao 60° najviše međusobno liče. Slika 4 prikazuje šare koje ima najveći broj pahuljica sa ovim uglom bez obzira na pravila i aksiom.



Slika 4. Tipične šare za pahuljice sa uglom od 60°.

Pahuljice sa uglovima 15° i 30° međusobno nisu toliko slične. Problem koji se javlja kod nekih pahuljica čiji je ugao 15° je što su previše guste ili zbijene, mada to kod većine pahuljica nije slučaj.

C. Rejterov modeli

Rejterovim modelom dobijaju se pahuljice koje su najviše nalik na one iz prirode. Na izgled pahuljice utiču vrednosti konstanti α, β i γ. Odnose ovih konstanti i rezultat njihovih varijeteta obrađeni su u radu Kliforda Rejtera [9].

D. Analiza ankete

Vršena je anketa nad 226 ispitanika. Ispitivano je koliko odabrane pahuljice generisane nekim od 3 metoda liče na pahuljice u prirodi, koliko se međusobno razlikuju i koji od metoda daje pahuljice koje najviše liče na prave. Rezultati ankete predstavljeni su u Tabeli II:

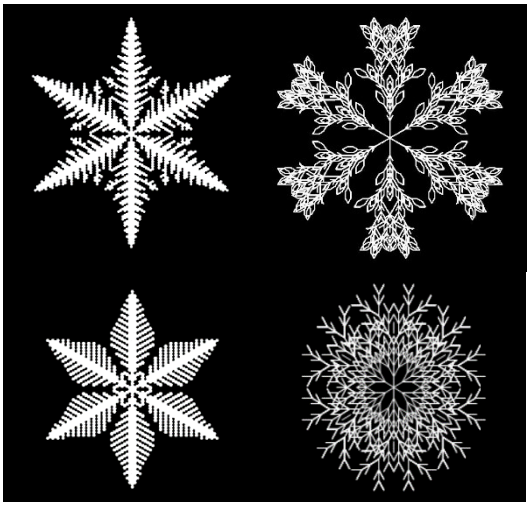
TABELA II
OCENE ISPITANIKA PO METODIMA

	L - sistemi	Rejterov model	Kohova pahuljica
Prosečna ocena koliko pahuljica liči na pahuljice iz prirode (ocena ima vrednost 1-10)	3,66	5,38	3,79
Metod kojim se generišu pahuljice koje najviše liče na one u prirodi (broj glasova u procentima)	13,72%	66,37%	19,91%
Metod kojim se generišu pahuljice koje se ispitanicima najviše dopadaju (broj glasova u procentima)	40,71%	51,77%	7,52%

Ispitanici su poređali metode po različitosti generisanih pahuljica na sledeći način:

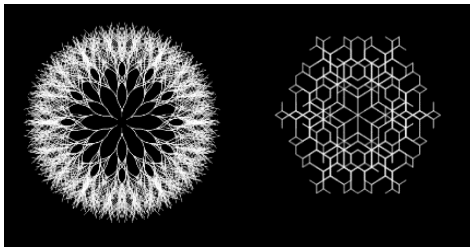
1. velika međusobna različitost: L – sistemi,
2. srednja međusobna različitost: Rejterov model,
3. mala međusobna različitost: modifikacija algoritma za Kohovu pahuljicu.

Analizom rezultata ankete izdvojene su četiri najbolje i dve najlošije ocenjene pahuljice (slike 5 i 6), kao i pahuljica sa najvećom devijacijom ocena (slika 7).

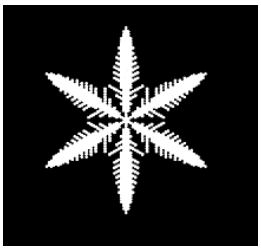


Slika 5. Četiri najbolje ocenjene pahuljice.

Ovi rezultati mogu da koriste za dalje proučavanje i unapređivanje sva tri modela. Informacije o najlošije ocenjenim pahuljicama mogu da pomognu u unapređivanju slučajno generisanih pravila kod L - sistema i u definisanju novih ograničenja radi što realističnijeg izgleda pahuljice.



Slika 6. Dve najlošije ocenjene pahuljice.



Slika 7. Pahuljica sa najvećom devijacijom ocena.

IV. ZAKLJUČAK

Rejterov model se pokazao kao najbolji za generisanje pahuljice od tri istraživana modela. Pahuljice generisane ovim metodom su najviše nalik pahuljicama iz prirode i ne postoje slučajevi koji mnogo odstupaju od izgleda pravih. Pahuljice generisane L - sistemima su najrazličitije, ali postoje slučajevi koji nemaju karakteristike pravih pahuljica. Pahuljice generisane modifikovanim algoritmom za Kohovu pahuljicu daju slične pahuljice koje zbog nasumičnog ugla ne ispunjavaju šestostranu simetričnost pravih pahuljica.

LITERATURA

[1] Libbrecht, Kenneth, "The snowflake", *Voyageur Press*, 2003.
[2] Libbrecht, Kenneth G. "Morphogenesis on ice: The physics of snow crystals." *Engineering and Science 64.1 (2001): 10-19*.
[3] Lewis G. Mason, Ronald R. Rautenberg, Michael J. Zyda, "Computer Graphics Interactive Workshop for Two-Dimensional Fractals", 1987, 109 pp.
[4] Prusinkiewicz, Przemyslaw, and James Hanan. *Lindenmayer systems, fractals, and plants*. Vol. 79. Springer Science & Business Media, 2013.
[5] Pavel Hazdra, Miloš Mazánek, "L-System Tool for Generating Fractal Antenna Structures with Ability to Export into EM Simulators", June 2006, 4p.
[6] Vladim Malyshev, "Random Grammars", June 1997, 36pp.
[7] Shanti. M, E. G. Rajan, "Spatio-Temporal Modeling of Snowflake Crystals Using Packard's Cellular Automata", Dec 2012, 6p.
[8] Jessica Li, "On the Modeling of Snowflake Growth Using Hexagonal Automata", 2014, 25pp.
[9] Reiter, Clifford A. "A local cellular model for snow crystal growth." *Chaos, Solitons & Fractals* 23.4 (2005): 1111-1119.

Distribuirani Girvan-Newman algoritam za detekciju zajednica

Ivan Bokor

Sadržaj - U ovom radu predstavljena je distribuirana verzija Girvan-Newman algoritma za detekciju zajednica. Glavna ideja je bila ubrzati algoritam kako bi se mogao koristiti i za veće grafove. Za testiranje su korišćeni distribuirani sistemi koji su sadržali 5, 7 i 13 računara ali veličina sistema može biti proizvoljna. Rezultati pokazuju da se vreme izvršavanja algoritma smanjuje povećanjem broja računara u sistemu. Optimalna veličina nije pronađena, što predstavlja temu za dalje istraživanje. Algoritam je testiran nad računarski izgenerisanim grafom upotrebom stohastičkog blokovskog modela.

I. UVOD

Grafovi predstavljaju matematičku strukturu koja se koristi za modelovanje veza između nekih objekata. Ti objekti se u grafu predstavljaju čvorovima i ukoliko između dva objekta postoji posmatrana veza, onda se čvorovi koji odgovaraju tim objektima povezuju. Veza između dva čvora u grafu se naziva ivica ili grana. Grafovi su izuzetno pogodni za reprezentaciju raznih sistema i mreža. Analizom ovakvih sistema se mogu bolje razumeti njihove osobine [1]. Modelovanjem realnih mreža primećeno je da se čvorovi često nalaze u grupama koje imaju osobinu da je broj ivica između čvorova iste grupe znatno veći od broja ivica između čvorova različitih grupa [2]. Te grupe se nazivaju zajednice i u mnogim mrežama čvorovi koji su u istoj zajednici imaju neke zajedničke osobine i/ili igraju slične uloge u toj mreži [1]. Na primer, ukoliko bismo modelovali prijateljstva u okviru neke škole, zajednice u tom grafu bi odgovarale grupama učenika koji se međusobno najviše druže. Detekcija ovakvih zajednica u mrežama može pružiti mnogo korisnih informacija o toj mreži. Jedan od algoritama za detekciju zajednica jeste Girvan-Newman algoritam [1]. Algoritam iterativno pronalazi i briše „najcentralniju" ivicu u grafu i izvršava se dok god postoje neobrisane ivice. Najcentralnija ivica je ivica kroz koju je povezano najviše čvorova.

Svakim brisanjem dobija se novi graf za koji se evaluira njegova struktura zajednica (*eng. community structure*) računanjem funkcije koja predstavlja evaluaciju isturenosti zajednica u tom dobijenom grafu. Ta funkcija se zove modularnost [2]. Pronađene zajednice predstavljaju komponente grafa sa najvećom vrednošću modularnosti. Ovaj algoritam se pokazao kao izuzetno efikasan, međutim zbog vremenske složenosti koja asimptotski pripada skupu $O(nm^2)$, gde su n i m respektivno broj čvorova i broj ivica u grafu, primenjiv je na grafove koji imaju do 10000 čvorova [2]. U problemima koji se susreću u realnosti je često potrebno obrađivati ogromne grafove sa nekoliko stotina hiljada čvorova. Cilj ovog rada je implementacija navedenog algoritma na distribuiranom sistemu i ispitati da li se time smanjuje vreme izvršavanja čime bi se omogućilo obrađivanje većih grafova.

I. Bokor, učenik 4. razreda Zemunske gimnazije, Gradski park 1, 11080, Beograd, Srbija / polaznik programa računarstva u Istraživačkoj stanici Petnica, Selo Petnica, 14104 Valjevo, Srbija, E-mail: ivanbokor17@gmail.com

II. METOD

A. Distribuirani Girvan-Newman algoritam

Algoritam se može podeliti na dva dela. Prvi deo je računanje vrednosti centralnosti za ivice u grafu i brisanje one ivice sa najvećom vrednošću a drugi deo je evaluacija grafa sa obrisanom ivicom računanjem funkcije modularnosti. U svom radu, Newman i Girvan su predstavili 3 metrike kojima su predstavljali centralnost ivice. U ovom radu će biti korišćena ona koja je pokazala najbolju tačnost, a to je metrika definisana pomoću najkraćih puteva (shortest path betweenness). Po toj metrici, centralnost ivice predstavlja broj najkraćih puteva između dva čvora u grafu koji prolaze kroz tu ivicu (ova metrika je definisana u [3]).

Za računanje centralnosti ivica iskorišćen je algoritam predstavljen u [1]. Funkcija modularnosti, ili samo modularnost, koja predstavlja ocenu isturenosti zajednica u nekom grafu [2]. Najčešće se zajednice u grafu ne znaju unapred i funkcija modularnosti je zato jako korisna jer omogućava da se oceni podela do koje je došao algoritam. Pošto nasumični graf (Erdős–Rényi model) uopšte nema zajednice, ideja je da se zajednice u grafu mogu pronaći poređenjem gustine ivica u nekom podgrafu tog grafa i vrednosti gustine ivica koja bi se dobila da je taj podgraf povezan nasumično [1]. Postoji nekoliko načina za računanje modularnosti, a u ovom radu je korišćena sledeća formula:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m}\right) \delta(c_i, c_j) \qquad (1)$$

gde je m broj ivica, A_{ij} matrica susedstva grafa, k_i stepen čvora i (respektivno k_j predstavlja stepen čvora j) a $\delta(c_i, c_j)$ funkcija koja vraća 1 ako su argumenti jednaki a 0 u suprotnom (c_i i c_j zapravo predstavljaju zajednice dodeljene čvorovima i i j). Modularnost se označava slovom

Q. Iz izraza (1) sledi da modularnost može imati vrednosti između 0 i 1, ali je za većinu grafova ta vrednost između 0.3 i 0.7 [2]. Algoritam se može elegantnije predstaviti na sledeći način: ako se graf u kome se traže zajednice označi sa G, za njega se u k-toj iteraciji pronađe ivica sa najvećom vrednošću centralnosti e_k, onda se ta ivica izbriše iz grafa G i izračuna se vrednost modularnosti grafa sa obrisanom ivicom Q_k. Ovaj postupak se ponavlja dok god postoje ivice u grafu G, što znači da će ukupan broj iteracija biti $|E|$ gde je E skup ivica grafa G. Pritom, računanje centralnosti ivica je potpuno nezavisno od računanja modularnosti. Umesto izvršavanja ovih računanja jedno za drugim, prva modifikacija je da se ona izvršavaju paralelno. Problem koji se javlja paralelizovanjem je taj što će se ova računanja izvršavati različitim brzinama, pa može doći do netačnih rezultata. Na primer, ako se računanje centralnosti izvršava brže nego računanje modularnosti, može se desiti da se modularnost ne izračuna za svaku obrisanu ivicu. Ovaj problem se može rešiti uvođenjem FIFO strukture u koju će se upisivati ivice sa najvećim vrednostima centralnosti. Pošto će takva struktura sačuvati redosled ivica, pre računanja modularnosti potrebno je samo obrisati ivicu koja se nalazi na vrhu te strukture. Jedini uslov sada jeste da postoje dve kopije grafa, po jedna za oba dela algoritma kako ne bi došlo do nekonzistentnosti. Kako bi izvršavanja ovih računanja bila još brža, ideja je razdvojiti ih na različite računare, odnosno izvršavati algoritam na distribuiranom sistemu. Time se svakako ispunjava i uslov da oba dela imaju svoju kopiju grafa, pošto se svaki deo izvršava na zasebnom računaru. Struktura u koju se upisuju ivice će se nalaziti na nekom računaru na kojem se ne izvršava nikakvo računanje. Taj računar zapravo predstavlja glavni računar u sistemu. Njegov zadatak je da na osnovu rezultata do kojih dolaze ostali računari odredi koji graf ima najveću vrednost modularnosti, pošto komponente tog grafa predstavljaju zajednice koje je algoritam pronašao. Kako bi se vreme izvršavanja algoritma još smanjilo, računanje centralnosti odnosno modularnosti se ne mora vršiti na samo jednom računaru, već se može razdvojiti i na više njih. U tom slučaju se svakom računaru, na kome se računanje izvršava, dodeljuje interval čvorova koje on treba da obradi. Konkretno, ako je broj čvorova u grafu N, onda se svakom računaru dodeljuje $K = \lfloor \frac{N}{C} \rfloor$ čvorova, gde je C broj računara u klasteru. Pošto je velika mogućnost da N nije deljivo sa C, onda će određeni broj računara (konkretno $N \bmod C$) obrađivati $K + 1$ čvorova. Interval $[1, N]$ se deli na C međusobno disjunktnih podintervala dužina K i $K + 1$. i-ti računar u klasteru obrađuje i-ti interval.

B. Implementacija distribuiranog sistema

Distribuirani sistem koji izvršava navedeni algoritam se sastoji od nekoliko računara i svakom računaru je dodeljena jedna od tri uloge. Jedan računar u tom sistemu ima ulogu glavnog računara. Njegov zadatak je da inicijalizuje ceo sistem, da čuva FIFO strukturu sa ivicama (kao što je pomenuto u prošlom odeljku), da pamti rešenja do kojih dolaze ostali računari, kako bi na osnovu njih mogao odrediti konačan rezultat algoritma i da koordiniše celim sistemom. Većina računara u sistemu ima ulogu radnika - na njima se izvršavaju računanja centralnosti ivica ili modularnosti. Na jednom računaru se izvršava samo jedno računanje, pa su ovi računari podeljeni u dva klastera: klaster računara koji računaju centralnost ivica i klaster računara koji računaju modularnost. Broj računara u klasteru može biti proizvoljan i ne mora se u oba klastera nalaziti isti broj računara. Kada se u klasteru nalazi više od jednog računara, tada svaki od njih daje rezultate samo za onaj interval čvorova koji mu je dodeljen. Da bi se dobili rezultati za ceo graf potrebno je agregirati rezultate svih radnika. To je zadatak računara koji imaju ulogu vođe klastera. Svaki klaster ima jednog vođu i on je odgovoran za komunikaciju sa glavnim računarom u sistemu. Vođa klastera glavnom računaru šalje rezultate do kojih je taj klaster došao a od glavnog računara dobija potrebne podatke za računanje. Konkretno, samo klaster koji računa modularnost dobija podatke od glavnog računara i taj podatak je ivica koju treba obrisati. Klaster za računanje centralnosti ivica samo šalje rezultate glavnom računaru, jer za njegovo izvršavanje nisu potrebni nikakvi ulazni podaci. Pošto je ukupan broj iteracija algoritma jednak kardinailtetu skupa ivica, ne bi bilo efikasno da glavni računar za svaku iteraciju pamti ceo graf i njegov modularnost. Efikasnije je pamtiti samo ivicu koja je u toj iteraciji obrisana. Traženi graf se onda dobija brisanjem ivica od prve iteracije do iteracije za koju je izračunata najveća vrednost modularnosti. Ceo postupak izvršavanja algoritma se može predstaviti sledećim koracima:

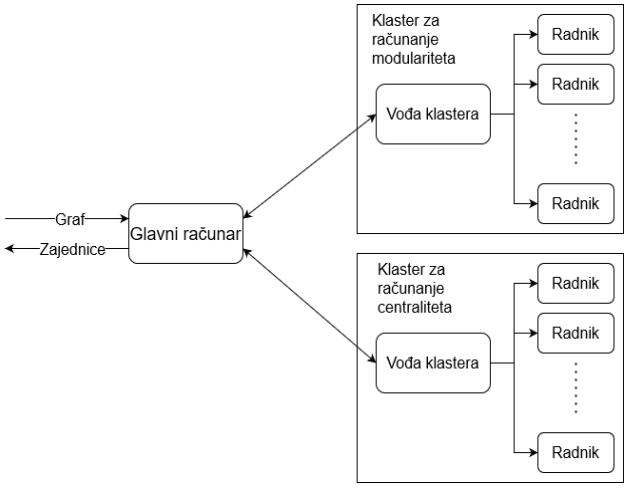
1. Glavni računar inicijalizuje sistem (ovde spada uspostavljanje komunikacije između računara i dodeljivanje uloga računarima)
2. Glavni računar daje signal vođama klastera da počnu sa izvršavanjem
3. Ukoliko je vođa klastera koji računa centralnost ivica poslao neke rezultate, glavni računar ih dodaje u strukturu
4. Ako struktura nije prazna, uzima se ivica sa njenog vrha i ona se šalje klasteru za računanje modularnosti
5. Dok nisu obrisane sve ivice ponavljaju se koraci 2.-4.
6. Nakon što su obrisane sve ivice, glavni računar konstruiše graf sa najvećom modularnošću (na osnovu redosleda brisanja ivica)
7. Glavni računar ispisuje komponente grafa sa najvećom modularnošću, što ujedno predstavlja i rezultat algoritma

Ceo sistem je implementiran u programskom jeziku C++ i korišćen je ZooKeeper, alat sa propratnom bibliotekom. Apache ZooKeeper predstavlja softverski alat za koordinaciju distribuiranih sistema [4]. Koordinacija se vrši pomoću takozvanog ZooKeeper stabla (skraćeno ZK stablo). ZK stablu imaju pristup svi računari povezani na ZooKeeper i mogu na njemu kreirati i brisati čvorove. U svaki čvor se može upisati neki podatak koji mogu pročitati svi povezani

računari. U konkretnoj implementaciji, za svaki računar postoji jedan čvor u ZK stablu i preko njega se vrši koordinacija. Na primer, kako bi računari razmenili svoje IP adrese, svaki računar u svoj čvor upiše svoju IP adresu i onda kada nekom računaru treba adresa nekog drugog računara, on samo treba da pročita šta je upisano u odgovarajući čvor (čvor koji odgovara tom drugom računaru). Sistem je šematski prikazan na slici 1.

C. Generisanje grafova

Performanse algoritma evaluirane su nad računarski generisanim grafovima. Grafovi su generisani pomoću stohastičkog blokovskog modela [5]. Model uzima sledeće parametre: broj čvorova u grafu, podela čvorova u zajednice (tako da je svaki čvor u tačno jednoj zajednici) i matrica verovatnoća. Skup ivica se onda dobija tako što se nasumično povezuju čvorovi iz različitih zajednica: verovatnoća da čvorovi iz zajednice i i j budu povezani je data u i-tom redu i j-toj koloni matrice verovatnoća. Korišćenjem ovog modela generisan je graf koji je korišćen za evaluaciju performansi algoritma.



Slika 1. Šematski prikaz distribuiranog sistema.

III. REZULTATI

Kako bi se ispitala tačnost algoritma korišćen je Zaharijev karate klub [6]. Pošto je za taj graf unapred poznata podela na zajednice. Nad ovim grafom je algoritam ostvarivao tačnost od 97%. Pošto je Zaharijev graf poprilično mali, za glavno testiranje je korišćen graf generisan na osnovu stohastičkog blok modela. Taj graf je sadržao 1200 čvorova i ~8200 ivica. Variran je broj računara u klasterima za računanje centralnosti ivica i modularnosti (broj računara u klasterima je međusobno jednak). Korišćeni su klasteri koji su sadržali 1, 2 i 5 računara. Za svaku veličinu klastera izmereno je vreme potrebno da algoritam odredi zajednice nad pomenutim grafom (vreme je izraženo u sekundama). Dobijeni rezultati su prikazani u tabeli 1. Rezultati u tabeli predstavljaju srednje vrednosti merenja. Merenja su izvršena dva puta. Iz tabele se vidi da broj računara direktno utiče na vreme izvršavanja.

TABELA I
REZULTATI TESTIRANJA

Broj računara u klasteru	Vreme izvršavanja [s]
1	218302
2	110030
5	43556.4

IV. ZAKLJUČAK

Na osnovu rezultata se može zaključiti da većim sistemima treba manje vremena za pronalaženje zajednica. Međutim, postavlja se pitanje da li je sistem uvek efikasniji sa većim brojem računara, ili postoji neka vrednost za koju sistem daje optimalne rezultate. Pretpostavka da ova vrednost postoji zasnovana je na tome što komunikacija između računara zahteva određeno vreme i što je više računara u sistemu, više vremena se troši na komunikaciju. Može se desiti da za neki veliki sistem računari radnici većinu vremena provedu čekajući na potrebne podatke od vođa klastera umesto da računaju vrednosti centralnosti, odnosno modularnosti. Ovu pretpostavku svakako treba ispitati i to može biti tema daljih istraživanja. Takođe, nije ispitano kako se sistem ponaša kada su klasteri različitih veličina i to predstavlja još jednu temu za dalje istraživanje. Prilikom testiranja sistema primećeno je da grafovi koji se generišu pomoću stohastičkog blok modela često nemaju onoliko isturenu strukturu zajednica koliko se očekuje sa zadanim verovatnoćama. Zbog toga bi trebalo ispitati performanse ovog sistema na većim, realnim, grafovima.

LITERATURA

[1] Fortunato, S. 2009. Community detection in graphs.
[2] Newman, M.E.J., Girvan, M. 2004. Finding and evaluating community structure in networks.
[3] Anthonisse, J.M. 1971. The rush in a directed graph. Technical Report BN 9/71, Stichting Mathematisch Centrum, Amsterdam
[4] Hunt, P., Konar, M., Junqueira, P.F., Reed, B. 2010. ZooKeeper: Wait-free coordination for Internet-scale systems
[5] Holland, P. W., Laskey, K. B., Leinhardt, S. 1983. Stochastic blockmodels: First steps. Social networks, 5(2), 109-137
[6] Zachary W. W., 1977. An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological research. 33(4), str. 452-473.

Segmentacija slika primenom dekompozicije tenzora i nenegativne faktorizacije matrica

Andrija Malbaša

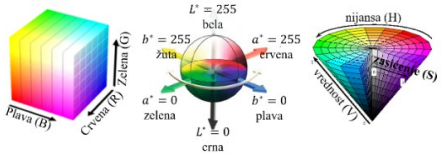
Sadržaj – Analizirani su i testirani algoritamski pristupi u problemu segmentacije slika u boji. Paradigme od interesa bazirane su na dekompoziciji tenzora i nenegativnoj faktorizaciji matrica. Korišćena su tri prostora boja. Prvi metod baziran na faktorizaciji tenzora je primenjen samo na CIELab prostor boja slike. Drugi metod baziran na NMF faktorizaciji testiran je na RGB, HSV i CIELab prostorima boja. Pre samog postupka segmentacije slike su podvrgnute pretprocesiranju. Sprovedeni koraci pripreme slika su birani specifično za odgovarajući metod. Konačni rezultati su upoređeni sa tradicionalnim postupkom segmentacije, K-means klasterizacijom.

I. Uvod

Segmentacija slika je proces deljenja slike na više segmenata ili regiona, gde svaki segment predstavlja različit objekat, tj. deo slike. Ovo je jedan od najvažnijih koraka u obradi i analizi slika. Podrazumeva odvajanje glavnog dela slike od pozadine ili grupisanje regiona piksela na osnovu sličnosti boja i oblika sa ciljem dalje obrade istaknutih regiona. To može biti u svrhe identifikacije, labeliranja ili neke statističke analize, itd. Segmentacija slika svoje primene nalazi u oblastima kao što su računarski vid i analiza medicinskih slika: lociranje tumora i drugih patologija, dijagnostika, simulacija virtuelne hirurgije. U ovom radu analizirane su metode za segmentaciju slika u boji bazirane na faktorizaciji matrica i tenzora. Dobijene informacije iz matrica faktora prosleđene su *k-means* algoritmu za odgovarajuće grupisanje.

A. Karakteristike tri prostora boja

Za predstavljanje slika na računaru i njihovu obradu korišćena su tri standardna prostora boja RGB, CIELab i HSV, slika 1.



Slika 1. Šematski prikazi prostora boja RGB, CIELab i HSV.

A. Malbaša je student master studija na modulu Veštačka inteligencija i mašinsko učenje na Elektronskom fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: malbasa@elfak.rs

Jedan od standarda za predstavljanje slike u boji na računarima je RGB model. Spada u aditivne modele boja. Osnovne boje u RGB sistemu su crvena, zelena i plava. Njihovim dodavanjem (mešanjem) dobija se široka paleta boja i nijansi. Ovakav model dizajniran je u skladu sa načinom generisanja boja na elektronskim uređajima, tj. nekadašnjim aktivnim elementima televizora i monitora.

Druga dva modela koja se koriste u radu dizajnirana su u Međunarodnoj komisiji za osvetljenje (CIE, 1976.) i od strane istraživača u oblasti kompjuterske grafike davnih 70-tih. CIELab model daje osvetljenje i hromatske informacije slike uz pomoć 3 koordinate. U implementaciji su korišćene sledeće vrednosti:

- L^* koordinata opisuje osvetljenje i ima opseg vrednosti od 0 (crno) do 255 (belo).
- a^* koordinata opisuje čiste boje od zelene do crvene. 0 predstavljaju zelenu boju, a 255 crvenu.
- b^* koordinatom se opisuju čiste boje žute i plave. Opseg vrednosti je od 0 za plavu, do 255 za žutu boju.

Bitna karakteristika CIELab modela je njegova uniformnost sa načinom ljudskog doživljaja bliskih boja. Time se Euklidovo rastojanje dve boje CIELab sistema slaže sa perceptualnom razlikom u ljudskom vizuelnom sistemu. Euklidovo rastojanje igra važnu ulogu u algoritamskim postupcima koji su kombinovani u literaturi iz oblasti segmentacije slika, pa i u ovom radu. Shodno tome je CIELab sistem korišćen u eksperimentima.

HSV sistem dizajniran je da modeluje izgled boja pod različitim osvetljajem. Za različite boje HSV sistem koristi radijalne isečke u koordinatnom konusu (ili cilindru) kojim se modelira. Centralna osa konusa rezervisana je za nijanse sive kao neutralne. Udaljavanjem od centralne ose meri se zasićenost bojom. HSV model boja je zbog toga model u skladu sa fizičkim karakteristikama vidljivog spektra, ali ne i ljudskim načinom percepcije. Upravo iz tog razloga veoma često ovakvi modeli daju rezultate i informacije koje ljudima nisu vidljive.

B. Dekompozicija matrica NMF

Non-negative matrix factorization (NMF ili NNMF) je niz algoritamskih rešenja za aproksimaciju matrica oblika

$$D \approx WH \tag{1}$$

gde su sve tri matrice D, W i H sa nenegativnim elementima. Matrice faktori W i H biraju se da budu niskog ranga i predstavljaju dobru aproksimaciju dekompozicije punog

ranga matrice D. Rang matrica W i H je parametar koji se definiše pri pozivu NMF iterativnog algoritma. Mera kvaliteta aproksimacije $D \approx WH$ iskazuje se nekom od matričnih normi. U radu je korišćena Frobenijusova norma. NMF je postala popularan pristup nakon što su Li i Sung predstavili jednostavne i efikasne algoritme izračunavanja [1], [2]. Koristi se u analizi podataka zbog svoje interpretacije u postupcima klasterizacije [3], [4]. S obzirom da NMF daje samo aproksimaciju matrice D, prisutan je određeni gubitak informacija. D je u kontekstu ovog rada matrica obeležja smeštenih duž vrsta. Manji skup izvedenih baznih obeležja nalazi se u vrstama matrice H. Vrste matrice W predstavljaju koordinate obeležja u odnosu na novu bazu. One su nosioci informacija za razvrstavanje sadržaja slike.

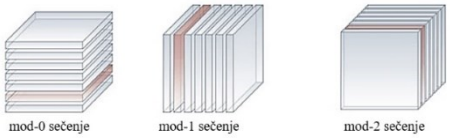
C. Taker dekompozicija tenzora

Dekompozicija matrica, kao što su SVD, PCA, NMF, zauzimaju važno mesto u računarstvu i informacionim tehnologijama. Predstavljaju algoritme velike pouzdanosti i odličnih performansi. Njihova geometrijska i algebarska interpretacija su u potpunosti istražene. Zbog toga je primena dekompozicija matrica veoma široka i stalno se pronalaze nove mogućnosti. Srećemo ih u postupcima kompresije slika i otklanjanja šuma, u sistemima za davanje preporuka, itd. Sa druge strane, dekompozicije tenzora su relativno nova i nepoznata oblast [5]. To se odnosi kako na matematičku teoriju koja stoji iza njih, tako i na domen njihove primene. Neke od novijih oblasti primene su obrada signala i statističko učenje. Algoritmi dekompozicija tenzora se i dalje istražuju za njihovu tačniju i efikasniju implementaciju na računarima.

Slično rastavljanju matrice na vektore vrsta i vektore kolona, trodimenzionalni tenzor može da se posmatra kroz tri skupa vektora koje zovemo vlakna, slika 2. Osim rastavljanja na skupove vektora trodimenzionalni tenzori mogu da se posmatraju i kroz skupove matrica. To zovemo sečenja tenzora, slika 3.



Slika 2. Tri skupa vlakana trodimenzionalnog tenzora [6].



Slika 3. Tri skupa matrica trodimenzionalnog tenzora [6].

Taker dekompozicija rastavlja tenzor $X \in \mathbb{R}^{I \times J \times K}$ u proizvod po tri moda oblika:

$$X \approx G \times_0 U_0 \times_1 U_1 \times_2 U_2 \qquad (2)$$

gde je $G \in \mathbb{R}^{P \times Q \times R}$ manji trodimenzionalni tenzor koji nazivamo jezgro-tenzor. Njegove elemente zovemo singularne vrednosti 0, 1 i 2-mod sečenja tenzora X. Matrice $U_0 \in \mathbb{R}^{I \times P}, U_1 \in \mathbb{R}^{J \times Q}, U_2 \in \mathbb{R}^{K \times R}$ su matrice ortonormiranih kolona i igraju ulogu singularnih vektora ili glavnih komponenti za svaki od skupova vlakana. Oznakom $\times_i$ zapisan je i-mod tenzorski proizvod. Dimenzije P, Q i R su tada 0, 1 i 2-rang tenzora X, tj. dimenzije vektorskih prostora vlakana mod-0, 1 i 2. To su parametri kojima se utiče na kvalitet rezultata. Primetimo da i Taker dekompozicija predstavlja aproksimaciju polaznog tenzora X. Njom se pomoću manjeg broja komponenti postiže aproksimacija polaznog skupa informacija. Greška se meri u smislu najmanjih kvadrata. Ova dekompozicija se često koristi u svrhe kompresije slika, tj. redukcije dimenzionalnosti podataka baš kao i SVD i PCA.

D. Pomoćni algoritmi u postupku segmentacije

Pored navedenih dekompozicija u implementaciji su korišćeni *K-means* i SLIC (*Simple Linear Iterative Clustering*) algoritam [7], [8], [9]. *K-means* algoritam je jedan od najpopularnijih postupaka ne-hijerarhijske particije skupa podataka. Zbog svoje jednostavnosti i brzine izvršenja koristi se samostalno ili u kombinaciji sa drugim načinima procesiranja podataka u svrhe klasterizacije. Jedan od ključnih parametara u pozivu *K-means* algoritma jeste K – broj traženih klastera. Kvalitet dobijenog rezultata u velikoj meri zavisi od njega.

SLIC algoritam generiše super piksele klasterovanjem piksela na osnovu njihovih međusobnih sličnosti i bliskosti u ravni slike. Koristi tri ulazna parametra kojima može da se utiče na kvalitet rezultata. Klasterovanje se vrši u petodimenzionalnom $labxy$ prostoru, gde je lab vektor boja jednog piksela u CIELab prostoru, a xy je pozicija tog piksela unutar slike. Rezultat primene SLIC algoritma je prekomerna segmentacija slike koja daljom obradom daje segmentacije manjeg obima.

II. IMPLEMENTACIJA METODA SEGMENTACIJE

U radovima autora Cirillo et al. i Achanta et al. predloženi su metodi segmentacije uz pomoć Takerove dekompozicije tenzora i nenegativne faktorizacije matrica [10], [11]. Po ugledu na njih sprovedena su istraživanja za potrebe ovog rada.

A. Segmentacija slike u boji Taker dekompozicijom tenzora

Slika se najpre konvertuje iz RGB u CIELab prostor boja, a potom se predstavlja u obliku tenzora histograma boja. Dimenzija tenzora je $I \times J \times Q$, gde I, J i Q predstavljaju broj nijansi sivih kanala L^*, a^* i b^*, redom. Iz tenzora boja se konstruišu dva tenzora X_s i X_d kako bi se udružili osvetljenje i boje na sledeći način:

$$x_d(i,j,k) = \begin{cases} 0, & a(i) < 128 \text{ ili } b(i) < 128 \text{ ili } L < \delta \\ \frac{L(k)}{\rho^2(i,j,k)}[a(i) - b(i)], & \text{inače} \end{cases} \quad (3)$$

$$x_s(i,j,k) = \begin{cases} 0, & a(i) < 128 \text{ ili } b(i) < 128 \text{ ili } L < \delta \\ \frac{L(k)}{\rho^2(i,j,k)}[a(i) + b(i)], & \text{inače} \end{cases} \quad (4)$$

gde $i \in L, j \in a, k \in b$, $\rho(i,j,k)$ je norma CIELab vektora čije su koordinate i, j i k, a δ je neki unapred određeni prag (255/11 u ovom radu). Na taj način se eliminišu mračniji, zeleni i plavi regioni slike i smanjuje dimenzionalnost tenzora. Tenzori X_s i X_d se mogu da predstave u obliku:

$$X = Y + E, \quad (5)$$

gde je tenzor Y dobijen rekonstrukcijom Takerove dekompozicije tenzora X, a E je greška aproksimacije. Prilikom Takerove dekompozicije tenzora X_s i X_d neophodno je odrediti rangove odgovarajućih jezgro-tenzora. Tenzori nad kojima se vrši dekompozicija imaju 3 moda. Gornja granica ranga tenzora je [256, 256, 256] kada sve vrednosti osvetljenja i boja definišu sliku, dok je donja granica [1, 1, 1], gde samo po jedna vrednost za osvetljenje i boje predstavljaju sliku. Određeno je da rang jezgro-tenzora bude količina nijansi sive svake CIELab komponente koje definišu sliku. U ovom radu, pozadinske vrednosti tenzora su postavljene na 0, čime se postiže dodatna redukcija dimenzionalnosti. Ovo eliminiše dosta mračne regione, i pojedine nijanse a^* i b^* komponente slike iz konačnog tenzora. Tenzori dobijeni Taker dekompozicijom u ovom radu su ranga [200, 71, 26]. Rekonkstruisani tenzori postaju ulaz u algoritam za klasterovanje koji vrše konačnu segmentaciju slike.

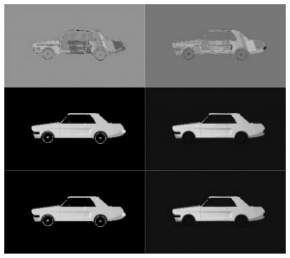
B. Segmentacija slike primenom nenegativne faktorizacije matrica

Slika se konvertuje u odgovarajući prostor boja i iz nje se izdvajaju super pikseli SLIC algoritmom opisanim u odeljku I.*D.*, po ugledu na rad Achanta et al. [11]. Za svaki od dobijenih super piksela se kreira 87-dimenzionalni vektor obeležja izdvajanjem informacija o boji, konturama i teksturi. Matrica obeležja, koja je dimenzija $broj_superpiksela \times 87$, razlaže se na matrice W i H uz pomoć NMF metode. Matrica W predstavlja matricu koordinata obeležja super piksela i na osnovu nje se uz pomoć algoritma klasterovanja oni dalje pregrupišu da daju izdvojene segmente.

III. Sprovedeni eksperimenti i dobijeni rezultati

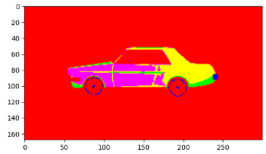
A. Segmentacija slike u boji Taker dekompozicijom tenzora

Slika [12] je prebačena u CIELab prostor boja i zamućena Gausovim 5x5 filterom. Kreirani su X_d i X_s tenzori boja kako je opisano u sekciji II.*A*. ovog rada. Na X_d i X_s tenzore je primenjena Taker dekompozicija sa 3-rangom [200, 71, 26]. Rekonstrukcijom na osnovu dobijene dekompozicije su kreirani tenzori Y_d i Y_s. Dobijeni tenzori, u zavisnosti od složenosti slike mogu značajno da poboljšaju kontrast tenzora X_d i X_s nakon eliminacija grešaka E, slika 4. Tenzori Y_s i Y_d se prosleđuju *K-Means* algoritmu za klasterovanje.



Slika 4. Kolona levo: tenzori $X_s - Y_s, X_s$ *i* Y_s, kolona desno: tenzori $X_d - Y_d, X_d$ *i* Y_d.

Na slici 5 prikazani su rezultati segmentacije slike sa *K*=5 klastera tj. segmenata.

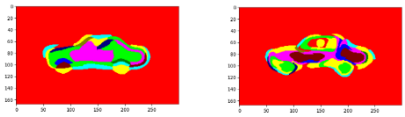


Slika 5. Slika segmentirana Taker dekompozicijom.

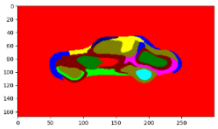
B. Segmentacija slike primenom nenegativne faktorizacije matrica

Eksperimenti su izvršeni nad 3 prostora boja: HSV, RGB i CIELab. U sva tri slučaja je slika najpre zamućena Gausovim filterom 5x5 kako bi se otklonio šum iz slike. Nakon toga je slika podeljena na super piksele primenom SLIC algoritma. Inicijalna podela sadržala je oko 200 super piksela. Iz segmentiranih super piksela se izvlače obeležja. Obeležja se dele na obeležja boje, teksture i oblika. Za obeležja boje izabrane su kumulativne sume histograma 3 kanala CIELab-a malog broja podeonih segmenata, kao i srednje vrednosti i standardne devijacije svakog od kanala. Za obeležja teksture je korišćena *Gray level co-occurrence matrix* (GLCM) [14]. GLCM kvantifikuje pojavljivanje vrednosti parova piksela određenih intenziteta kod unapred specificiranih prostornih veza (udaljenost i pomeraj). U eksperimentima je određeno da se posmatraju susedni pikseli u pomerajima od 0, 45, 90 i 135 stepeni. Iz GLCM su izvučeni energija, homogenost, kontrast, korelacija i nesličnost, koji predstavljaju obeležja teksture super piksela. Korišćena je i binarna informacija o konveksnosti super piksela kao obeležje njihovog oblika. Ovako kreirana matrica obeležja svih superpiksela se u nastavku razlaže na W i H matrice postupkom NMF. W je matrica dimenzija $broj_superpiksela \times 20$. Ona se prosleđuje u *K-means*

algoritam za klasterovanje. Time se vrši konačna segmentacija na $K = 10$ klastera tj. segmenata. Eksperimenti su pokazali da opisani postupak daje najbolje rezultate za HSV prostor slika, kod koga je prednji deo slike jasno odvojen od pozadine, bez potrebe za dodatnim procesiranjem. Kod RGB varijante je bilo neophodno povećati broj segmenata na 500, takođe izdvojiti konturu prednjeg dela slike uz pomoć OpenCV funkcija *findContour* i *drawContour*. Funkciji *findContour* je prosleđena pragovana *Grayscale* verzija početne slike i izlazi funkcije *drawContour* kojom se modifikuje početna slika. Rezultati su malo gori od HSV verzije slike, ali uz mnogo više pretprocesiranja.



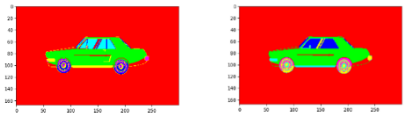
Slike 6 i 7. NMF segmentacija HSV i RGB slike, redom.



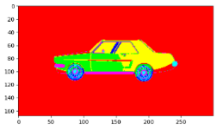
Slika 8. NMF segmentacija CIELab slike.

C. K-Means segmentacija

Kod *K-Means* segmentacije je slika raskloplјena po modu 2 i predata *K-Means* algoritmu na klasterovanje za segmentaciju na 10 regiona.



Slike 9 i 10. *K-Means* segmentacija HSV i RGB slike, redom.



Slika 11. *K-Means* segmentacija CIELab slike.

V. Zaključak

Opisani metodi daju najbolje rezultate kada je u slici prednji deo jasno odvojen od pozadine. Kod fotografija, oni će skoro uvek jasno izdvojiti glavne regione. Međutim, ukoliko fotografija sadrži previše detalja, moguće je da pozadina i/ili prednji deo budu izdeljeni na dodatne, neželjene regione. *K-means* algoritam pokazuje dobre karakteristike u primeni nad slikama sa malim brojem elemenata homogenih u boji. Zbog svoje jednostavnosti i brze primene on svakako predstavlja veoma važnu kariku u opisanim postupcima. Ipak, njegova primena u kompleksnijim situacijama mora da sadrži kompleksniju transformaciju slike unutar nekog prostora podataka. U planu je istraživanje nenegativne faktorizacije tenzora koja bi mogli da posluži u postupku segmentacije slika po analogiji sa NMF, kao i drugih načina primene spomenutih faktorizacija u postupcima segmentacije.

Zahvalnica

Autor rada se zahvaljuje prof. dr Jovani Džunić, sa Elektronskog fakulteta u Nišu na pruženoj pomoći tokom izrade ovog rada.

Literatura

[1] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization. Nature. 401 (6755) 788–791, 1999.

[2] D.D. Lee, H.S. Seung, Algorithms for Non-negative Matrix Factorization, Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press. 556–562, 2001.

[3] C. Ding, X. He, H.D. Simon, On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering, Proc. SIAM Int'l Conf. Data Mining, 606-610, 2005.

[4] C. Ding, Y. Li, W. Peng, On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing, Computational Statistics & Data Analysis. 52 (8) 3913–3927, 2008.

[5] T.G. Kolda, B.W. Bader, "Tensor Decompositions and Applications", SIAM REVIEW, 51 (3) 455-500, 2009.

[6] J. Džunić, "*Matematički metodi u mašinskom učenju*", Elektronski fakultet, Univerzitet u Nišu, Niš, 2023.

[7] Dokumentacija K-means implementacije u sklearn.cluster modulu Python biblioteke, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (posećen 14.10.2023.)

[8] Dokumentacija za SLIC algoritam implementaciju u modulu skimage.segmentation Python biblioteke, https://scikit-image.org/docs/dev/api/skimage.segmentation.html#skimage.segmentation.slic

[9] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11) 2274-2282, 2012. [11]

[10] M.D. Cirillo, R. Mirdell, F. Sjöberg, T.D. Pham, Tensor Decomposition for Color Image Segmentation of Burn Wounds. Sci Rep., 1;9(1):3291, 2019.

[11] V.-H. Duong, Y.-S. Lee, B.-T. Pham, P. T. Bao and J.-C. Wang, NMF-based image segmentation, 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, Taiwan, 1-2, 2016.

[12] Slika automobila https://www.linearity.io/blog/how-to-draw-a-car/ (posećen 14.10.2023.)

[13] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika, 31 (3) 279–311, 1966.

[14] GLCM texture features https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_glcm.html (posećen 14.10.2023.

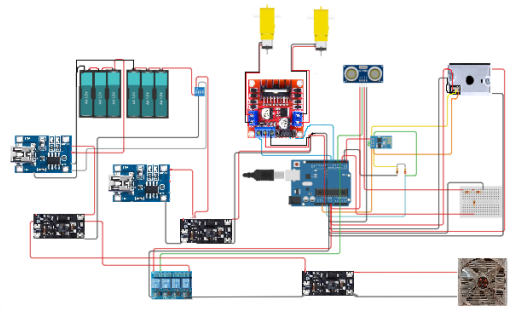
ROBO-Clean: robot za prečišćavanje vazduha

Teodor Jovanović

Sadržaj - Ovaj rad ima za cilj da prikaže realizaciju sistema za prečišćavanje zagađenog vazduha. Osnovu robota sadrže sledeće komponente: mikročip ESP8266, Cubic PM1006k senzor za detekciju čestica, Dual H-Bridge L298N, HC-SR04 ultrazvučni senzor za merenje rastojanja, relej, dva BMS modula (*Battery Management System*), dva step-up modula (kola za podizanje napona), ventilator sa filterom, dve litijum-jonske baterije i dva Arduino DC motora. Program je izrađen u softverskom paketu *Arduino IDE*.

I. Uvod

Zagađen vazduh predstavlja veliki problem u svetu, a i u Srbiji. Beograd je 2. marta bio na crnoj listi 10 najzagađenijih gradova na svetu. Izduvni gasovi iz fabrika i prevoznih sredstava pune vazduh azot-dioksidom i drugim česticama, a najopasnije od njih su PM2.5 čestice, koje su 20 puta manje od zrna peska. Zbog toga sa lakoćom dospeva ne samo u pluća i krvotok, već i u mozak. Kada dopru do mozga, one stvaraju plak između nervnih ćelija. Ovakve čestice uzrokuju bolesti poput demencije, bolesti disanja, a čak i infarkt. Zbog ovog problema, konstruisan je robot : *ROBO-Clean*, kako bi pokazao realizaciju sistema za prečišćavanje vazduha. Evo za primer sajt koji prikazuje zagađenost vazduha u celom Beogradu : www.beoeko.com.



Slika 1. Osnovna šema.

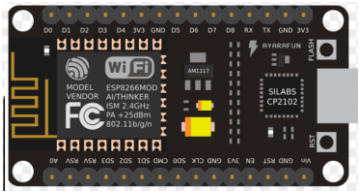
II. Princip rada robota

Robot radi na principu prikupljanja podataka iz svog okruženja uz pomoć Cubic PM1006k senzora za detekciju čestica (koji meri čistoću vazduha) i CH-SR04 ultrazvučnog senzora (koji detektuje i meri dimenzije prostora u kome se robot nalazi). Ovi podaci prosleđeni su mikrokontroleru robota ESP8266, koji zatim u odnosu na njih robotu daje komande da ugasi motore i upali ventilator sa filterom (kada je vazduh zagađen), upali motore i ugasi ventilator sa filterom (kada vazduh nije zagađen) ili da upali motore da se kreću unazad 0.8 sekundi, da se rotira ka levo 0.8 sekundi i upali motore da se kreću ka napred (kada robot naiđe na prepreku). Robot takođe, uz pomoć WiFi modula na mikrokontroleru, šalje podatke o zagađenosti vazduha na bilo koji uređaj koji je u njegovoj blizini i povezan na internet.

T. Jovanović je učenik prve godine oglednog društveno-jezičkog smera Šeste beogradske gimnazije, Milana Rakića 33, 11000 Beograd, Srbija, Email: teodor.jovanovic@hotmail.com.

A. Predstavljanje komponenata i priprema za izradu

ESP8266 WiFi modul je SOC (sistem na čipu) sa integrisanim TCP/IP protokolom koji svakom mikrokontroleru može dati pristup WiFi mreži. ESP8266 uz WiFi modul poseduje i mikrokontrolerske sposobnosti. Njegov procesor je *Tensilikin L106*, 32-bitni mikroprocesor, koji radi na 80 ili 160 MHz. Napaja se na 3.3 V DC. Na slici 2 prikazan je ESP8266 mikročip.



*S*lika 2. ESP8266.

Senzor zagađenosti vazduha (Cubic PM1006k senzor za detekciju čestica) je senzor koji meri zagađenost vazduha. On se sastoji od malog ventilatora i kutijice. Mali ventilator ima zadatak da uvlači i izvlači vazduh iz kutijice. U kutijici se nalazi jedno iskrivljeno ogledalo i izvor infracrvenog svetla. Kada je robot upaljen, infracrveno svetlo će se prostirati kroz vazduh u kutijici dok ne dođe do iskrivljenog ogledala koje se nalazi ispred izvora te svetlosti. Ako je vazduh čist, infracrveno svetlo će doći do ogledala, odbiti se i otići negde odakle se neće vratiti ka izvoru. Ako je vazduh zagađen, infracrveno svetlo će naleteti na neke čestice zagađenosti na svom putu, odbiće se o njih i vratiti ka izvoru. Koliko svetlosti se vratilo i za koje vreme ovaj senzor računa i pretvara u informaciju koliko je vazduh zagađen.

Senzor zagađenosti vazduha napravljen je tako da može da prepozna od PM1 do PM10 čestica. PM čestice su suspendovane čestice dimenzija reda veličine mikrometra. Najveći zagađivači u vazduhu su PM2.5 čestice, zbog čega je program po kome se računa koliko je vazduh zagađen izmenjen. Senzor zagađenja u vazduhu nalazi se na slici 3.



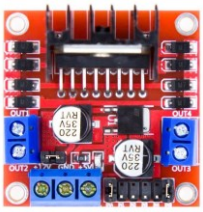
Slika 3. Senzor zagađenja vazduha (Cubic PM1006k senzor za detekciju čestica).

Ultrazvučni senzor za merenje razdaljine (CH-SR04) je senzor koji daje informacije o prostoru. Pošto je uperen ka napred, on robotu daje informaciju o tome da li se ispred robota nalazi neka prepreka. On to radi tako što šalje ultrazvuk u prostor i čeka da se odbije o neku prepreku, kako bi mu se vratio. Kada se ultrazvuk vrati, program računa za koje vreme je zvuk putovao kroz prostor, i uz pomoć brzine zvuka računa razdaljinu od neke prepreke. Ako robot dođe previše blizu neke prepreke, ultrazvučni senzor upozoriće ga da promeni smer motora u nazad za 0.8 sekundi, da se rotira ka levo 0.8 sekundi i da na kraju krene da se kreće napred. Ovom akcijom robot se okreće 180 stepeni od prepreke. Ultrazvučni senzor nalazi se na slici 4.



Slika 4. Ultrazvučni senzor (HC-SR04 ultrazvučni senzor za merenje rastojanja).

Dual H-Bridge (L298N) je modul zaslužan za kontrolu motora robota, individualno jedan od drugog. To znači da motori ne moraju da rotiraju u istom smeru, što robotu daje sposobnost da umesto da se kreće samo napred-nazad, da može da se rotira levo i desno. Dual H-Bridge napaja se na 5 V DC. Dual H-Bridge nalazi se na slici 5.



Slika 5. Dual H-Bridge L298N.

Na slici 6. prikazan je jedan Arduino DC motor koji služi za kretanje robota.



Slika 6. Arduino DC motor.

Na slici 7. prikazan je BMS (*Battery Management System*). On se brine o tome da se baterije robota ne isprazne, da puni baterije i da ne prepuni baterije strujom.



Slika 7. BMS (*Battery Management System*).

Na slici 8 nalazi se kolo za podizanje napona (*step-up* modul). Ono služi da podiže napon na 5 V kako bi komponente poput Dual H-Bridge- a i ventilatora sa filterom mogle da rade, jer zahtevaju 5 V.



Slika 8. *Step-up* modul.

Relej služi za paljenje ventilatora sa filterom. Oni se nalaze na slici 9.



Slika 9. Relej i ventilator sa filterom.

B. Izrada sistema

Sve komponente na robotu su povezane na ESP8266. Senzor zagađenja vazduha koristi pinove TX, RX, G i Vin na ESP8266. Senzor zagađenja vazduha povezuje svoju naponsku žicu na Vin i GND žicu na G pin od ESP8266. Ultrazvučni senzor povezuje svoje pinove GND na G, Vcc na Vin, Echo na D1 i Trig na D4 od ESP8266. Dual H-Bridge povezuje svoje pinove o motorima IN1 na D6, IN2 na D5, IN3 na D7 i IN4 na D8 od ESP8266. Dual H-Bridge povezuje ostale pinove 5 V na RST, GND na G i 12 V na step up 1. Motor A je povezan sa svoje naponske žice na OUT1 i GND žice na OUT2 na Dual H-Bridge- u. Motor B je povezan na svoje naponske žice na OUT3 i GND žice na OUT4 na Dual H-Bridge- u. Relej povezuje svoje pinove Vcc na Vin, IN3 na D2 i GND na G od ESP8266. Relej takođe povezuje svoje pinove za izvor energije od 5 V na step-up 1 i pin za korišćenje te energije za naponsku žicu ventilatora sa filterom. Baterija 1 povezuje naponsku žicu

na prekidač 1 i GND žicu na step-up. Baterija 2 povezuje naponsku žicu na BMS 2 i GND žicu na BMS 2. Prekidač 1 naponsku žicu povezuje povezuje sa drugim krajem step-up-a 1, odakle GND i naponska žica od baterije 1 dolaze do BMS- a 1. Iz BMS- a 2 naponska žica odlazi u prekidač 2, a odatle u step-up 2, a od njega u pin RST na ESP8266. Druga žica iz step-up- a 2 odlazi u G pin od ESP8266.

Linija koda za očitavanje zagađenosti u vazduhu:

```
pm1006.read_pm25(&zagadj);
```

Linija koda za očitavanje razdaljine od prepreke:

```
//Očitavanje rastojanja
digitalWrite(pin_EhoPrednji_trig, LOW);
delayMicroseconds(2);
//Postavlja TrigPin na HIGH stanje za 10 mikro sekundi
digitalWrite(pin_EhoPrednji_trig, HIGH);
delayMicroseconds(10);
digitalWrite(pin_EhoPrednji_trig, LOW);
//Čita EchoPin I vraća vreme putovanja zvučnog talasa u mikro sekundama
duration = pulseIn(pin_EhoPrednji_echo, HIGH);
//Računanje rastojanja
distance = duration * 0.034 / 2;
//Prikazivanje rastojanja na serialni monitor
DPRINT("Distance: ");
DPRINTLN(distance);

float rastojanje_napred = distance;
```

Definicija granice zagađenosti:

```
#define granica_zagadjenosti 100
```

Linija koda za WiFi modul:

```
void handle_OnConnect() {
  server.send(200, "text/html", SendHTML(zagadj));
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}

String SendHTML(int zagadjenje){
  String ptr = "<!DOCTYPE html> <html>\n";
  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
  ptr +="<title>LED Control</title>\n";
  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
  ptr +=".button {display: block;width: 80px;background-color: #1abc9c;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";
  ptr +=".button-on {background-color: #1abc9c;}\n";
  ptr +=".button-on:active {background-color: #16a085;}\n";
  ptr +=".button-off {background-color: #34495e;}\n";
  ptr +=".button-off:active {background-color: #2c3e50;}\n";
  ptr +="p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
  ptr +="</style>\n";
  ptr +="</head>\n";
  ptr +="<body>\n";
  ptr +="<h1>ESP8266 Web Server</h1>\n";
  ptr +="<h3>Using Access Point(AP) Mode</h3>\n";

  ptr +="<p>Trenutno ocitano: ";
  ptr += String(zagadjenje);
  ptr += " mikrograma PM2.5 cestica.</p>\n";

  ptr +="</body>\n";
  ptr +="</html>\n";
  return ptr;
}

  WiFi.softAP(ssid, password);
  delay(100);

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
  server.begin();
}

void loop() {
  server.handleClient();
```

Linija koda sa posebnom akcijom:

```
//Stani i uključi ventilator ako je zagađeno
  if(granica_zagadjenosti>100) {
```

```
  digitalWrite(relej_ventilator, LOW);
  stani();
 } else  {
  digitalWrite(relej_ventilator, HIGH);
  //Nastavi u istom smeru do prepreke, pa promeni smer
  if(rastojanje_napred < 30 && rastojanje_napred>0){
     idiNazad();
     delay(800);
     idiLevo();
     delay(800);
     idiNapred();
  }else
    idiNapred();
 }

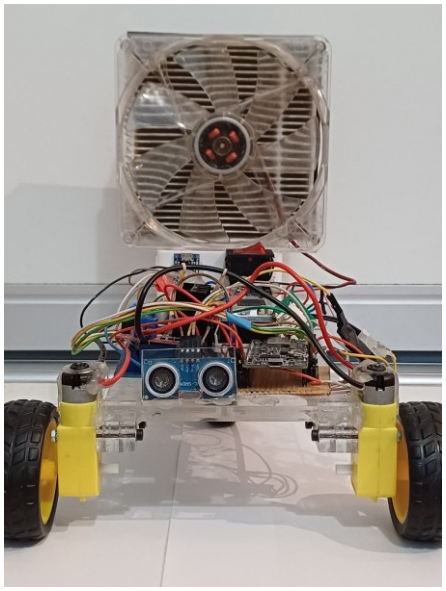
 delay(20);
 DPRINTLN(millis());
}
```

III. Zaključak

Ovaj robot predstavlja realizaciju rešenja problema o zagađenosti vazduha. Robot može da se kreće do mesta gde je zagađen vazduh i da ga prečisti. Kada završi svoj zadatak, robot nastavlja dalje. Pored toga što predstavlja rešenje problema o zagađenosti vazduha, *ROBO-Clean* predstavlja i inspiraciju za dublje pokretanje teme i nalaženja rešenja.

Ovaj projekat pruža mogućnost za dalja istraživanja. Da bi dobili sistem koji je dobra osnova za prilagođavanje i proširavanje kapaciteta i koji uz odgovarajuća poboljšanja može služiti za regulisanje različitih parametara (na primer, tačnije računanje PM čestica i elemenata u vazduhu). Od ovakve polazne tačke lako je napraviti sistem koji će biti mnogo efikasniji i praktičniji od sadašnjih rešenja za prečišćavanje vazduha. Logičku osnovu za ovakvu primenu smo napravili, a kod realizacije bi samo trebalo prilagoditi dimenzije i rešenja prečišćavanja vazduha na drugačiji način. Autor rada je zadovoljan sa krajnjim prozvodom.

Realizovan robot prikazan je na slici 10.



Slika 10. Izgled robota.

Zahvalnica

Autor rada zahvaljuje se Šestoj beogradskoj gimnaziji radi pružanja tehničkih i materijalnih sredstava. Takođe velika zahvalnost i profesorki računarstva i informatike Gordani Ćeranić na pruženoj podršci.

Literatura

[1] ESP8266. Dostupno na: https://en.wikipedia.org/wiki/ESP8266)
[2] Senzor zagađenja vazduha (*Cubic PM1006k particle sensor*). Dostupno na: https://preciscivacvazduha.rs/pm-cestice-sve-sto-treba-da-znate
[3] Dual H-Bridge L298N. Dostupno na : https://dronebotworkshop.com/dc-motors-l298n-h-bridge/
[4] Tehnička dokumentacija za step-up module. Dostupno na : https://www.smart-prototyping.com/Mini-DC-DC-Step-up-Power-Module
[5] Tehnička dokumentacija BMS. Dostupno na : https://www.servovision.com/Battery/Batterry%20LithiumFerrumPhosphate/Large%20LiFe%20BMS.html
[6] Relej. Dostupno na : https://sh.wikipedia.org/wiki/Relej
[7] Zagađenje u vazduhu Beograda. Dostpno na : (https://www.beoeko.com)

Projektovanje pasivne LED matrice korišćenjem *Constant Current* LED drajvera

Sava Dinić

Sadržaj – Usled sve većeg razvoja tehnike, tržište se više ne fokusira samo na funkcionalost, već i na izgled uređaja. Shodno tome, primena LE dioda, čije su primarne funkcije u vizuelnom domenu, se povećava. Cilj ovog rada je projektovanje matrice LE dioda u konfiguraciji 4x4 i njeno osposobljavanje za rad u realnim uslovima. U radu su prikazani teorijski osnovi LED displeja, a zatim i način funkcionisanja *Constant Current* LED drajvera (LED drajvera sa konstantnom strujom na izlazu). Nakon toga date su i neke osnove LED matrica. Najvažnije komponente sistema koje su upotrebljene u ovom radu su posebno objašnjene. Na kraju, najviše pažnje posvećeno je praktičnoj realizaciji, gde je akcenat na kodu, šemi projekta i krajnjim rezultatatima.

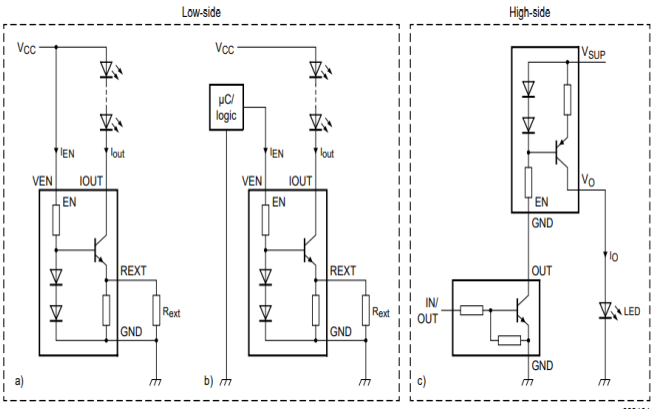
I. Uvod

LED rasveta predstavlja tehnologiju koja omogućava svetlosnu emisiju pomoću dioda koje se nazivaju LED (Light-Emmiting Diode). Zbog brojnih prednosti LED-a, u koje spadaju mala potrošnja električne energije i velika pouzdanost u radu, odnosno dugotrajnost, LED rasveta je postala dominantna u oblasti osvetljavanja prostorija. LED displej predstavlja tip video ekrana sastavljen od LE diode. Tipičan LED ekran se sastoji od mnoštva LED panela, koji se ponekad koriste i kao oblik osvetljenja, ne samo za prikaz. Postoje dve klase LED panela: konvencionalni LED paneli, koji koriste diskretne LE diode, i paneli za površinsku montažu (*SMD*). Konvencionalni paneli su češći, posebno za velike spoljne ekrane. U ovom tipu LED panela, skup crvenih, zelenih i plavih LE dioda su grupisani zajedno tako da formiraju brojne piksele kvadratnih oblika. *SMD* tehnologija se obično koristi za izradu LED displeja koji se koristi u zatvorenom prostoru. Ovi LED displeji, takođe, sadrže crvene, zelene i plave LE diode za prikaz na ekranu. Glavne praktične razlike između njih i konvencionalnih LED panela su smanjena maksimalna udaljenost gledanja kod *SMD*-a, kao i blago smanjena osvetljenost. Ovo čini *SMD* pogodnijom LED tehnologijom za unutrašnje aplikacije. LED displeji imaju mnogo primena, koje se mogu videti ispred izloga, na stadionima, aerodromima, elektronskim uređajima (televizori, monitori, telefoni). Neki od tipova LED displeja su: LED displeji za prikaz teksta i slika, video, digitalni i rešetkasti LED displeji [1].

S. Dinić je student master akademskih studija na studijskom programu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail:savadinic@elfak.rs

II. Koncept *Constant Current* LED drajvera

Iako su za LE diode velike snage potrebne složene šeme drajvera i kompleksni sistem za termalnu zaštitu, velika većina aplikacija koje sadrže LED drajvere koristi LE diode sa strujama daleko ispod 500 mA. U tom slučaju diode se mogu pokretati veoma jednostavno pomoću kola koja sadrže *Constant Current* LED drajvere, koji rade kao prost linearni regulator. Slika 1 prikazuje unutrašnje topologije, odnosno konfiguracije tri različita *Constant Current* LED drajvera. [2] Takav drajver se u unutrašnjosti sastoji od bipolarnog tranzistora, dve diode i dva otpornika. Na Slici 1 a) i b) NPN BJT rade kao *low-side* drajveri, dok se na Slici 1 c) PNP bipolarni tranzistor se koristi kod *high-side* drajvera. Jedan od otpornika definiše minimalnu izlaznu struju, a drugi obezbeđuje ulazni napon i igra važnu ulogu za enable pin. I *low-side* i *high-side* drajveri imaju enable pin. Razlika je ta što je *low-side* drajverima potreban poseban potencijal da bi bili omogućeni. Drajveri na Slici 1 a) zahtevaju napon u opsegu napona napajanja da bi ih omogućili (uključili). Drajver na Slici 1 b) može biti omogućen na mnogo nižem naponu od 3,3 V uz pomoć dodatnog mikrokontrolera ili logičkog uređaja. *High-side Constant Current* LED drajver sadrži enable pin koji je povezan na GND, tako da se LED može isključiti diskonektovanjem ovog pina sa GND-a. U praksi, ovo se izvodi pomoću *RET*-a (specifično kućište bipolarnog tranzistora, gde se osim bipolarnog tranzistora nalaze i dva otpornika), kao što je ilustrovano na Slici 1 c).



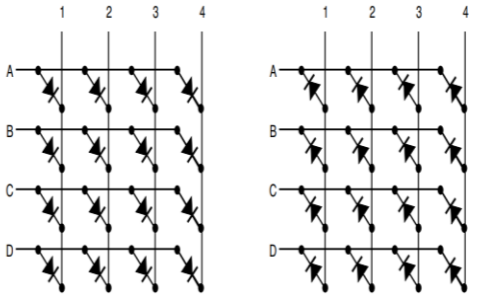
Slika 1. Konfiguracije LED drajvera. a) i b) Low-side), c) High-side.

Izlazna struja *Constant Current* LED drajvera se podešava eksternim (spoljašnjim) otpornikom R_{ext}. Neki tipovi drajvera su podešeni na uobičajene fiksne struje pomoću unutrašnjeg otpornika. U aplikacijama gde se eksterni otpornik može povezati, on će biti paralelan sa unutrašnjim otpornikom, čime se smanjuje efektivna otpornost. *Low-side Constant Current* drajveri koji koriste NPN tranzistor imaju unutrašnji otpornik od 95 Ω. Treba obratiti pažnju da *Constant Current* LED drajveri unose strujni drift (strujno odstupanje) u zavisnosti od temperature i napona, koje u praksi ne iznosi više od 5 %. Minimalni pad napona na *Constant Current* LED drajveru je oko 1,4 V. Ispod ovog napona, linearna regulacija ne funkcioniše pravilno. Iznad njega, pad napona na ovom drajveru se automatski prilagođava da bi se dobila željena izlazna struja. Kod *low-side* i *high-side* konfiguracija, izlazni napon se uvek izračunava kao $V_{OUT} = V_{CC} - V_{LED}$, gde je V_{LED} napon na LE diodama, a V_{CC} napon napajanja. Ako su poznati željena struja pogona i pad napona na LE diodama, minimalni mogući napon napajanja može se izračunati sabiranjem izlaznog napona na drajveru sa naponom na LE diodama. Dakle, dozvoljeni radni opseg je:

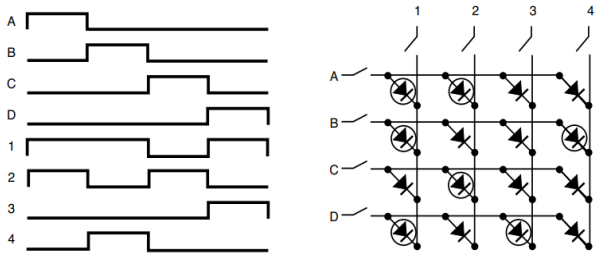
$$V_{LED} + 1.4\ V < V_{CC} \tag{1}$$

III. LED MATRICE

Relativno je jednostavno da se pokrenu nekoliko LE dioda pojedinačno. Međutim, kako se broj LED dioda povećava, količina resursa potrebnih za rad ovih LED dioda raste do nivoa kojim se ne može upravljati. Kao takve, LE diode su često raspoređene u matrice, [3] kako bi se omogućilo efikasnije korišćenje datih resursa. Osnovni princip u strukturi LED-a u obliku matrice je taj da se svaka LE dioda može adresirati (pozvati) jasnim navođenjem njene lokacije preko redova i kolona u kome se ona nalazi. Na primer, prva gornja leva LE dioda, na Slici 2, se adresira kao (A,1), tj. red A, a kolona 1. Ako bi, na primer, bilo potrebno da se uključi treća dioda u drugom redu, onda bi u tom slučaju ta LE dioda bila adresirana kao (B,3), tj. red B, a kolona 3. Ovaj metod adresiranja takođe ukazuje na protok električne struje. Da bi se uključila LED (A,1), struja teče od zajedničke anode (ili katode) reda A do zajedničke anode (ili katode) kolone 1. Slika 2 prikazuje dve različite konfiguracije. Razlika između ovih konfiguracija je u metodi koja se koristi za uključivanje LE dioda. U konfiguraciji gde je svaki od redova od A do D povezan na zajedničku anodu, kolone koje su na slici prikazane, od 1 do 4 su povezane preko konfiguracija sa zajedničkom katodom. U obrnutom slučaju, kada su redovi u konfiguraciji sa zajedničkom katodom, kolone su u konfiguraciji sa zajedničkom anodom. Multipleksiranje je tehnika koja se koristi za rad sa LED matricama. Multipleksiranje omogućuje rad sa pojedinačnim diodama samo jednog reda LED matrice u trenutku vremena. Za ilustrovanje koncepta multipleksiranja koristi se konfiguracija redova sa zajedničkom anodom. Struktura koja podseća na stepenice (A do D) pokazuje vremensku raspodelu multipleksiranja, koja je upotrebljena na Slici 3. Samo jedan red se nalazi pod naponom u određenom vremenskom trenutku. Tokom perioda u kome je dati red pod naponom, željene LE diode se uključuju preko odgovarajućih kolona.



Slika 2. LED matrice u konfiguraciji redova sa zajedničkom anodom (levo) i zajedničkom katodom (desno).

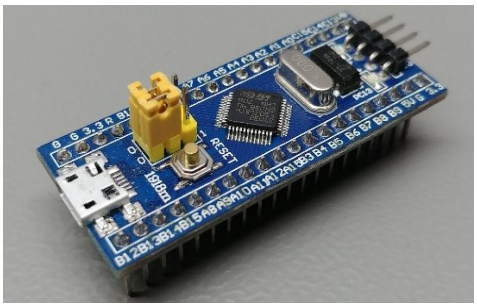


Slika 3. Multipleksiranje LED matrice.

IV. KORIŠĆENE KOMPONENTE ZA REALIZACIJU PROJEKTA

A. Blue pill

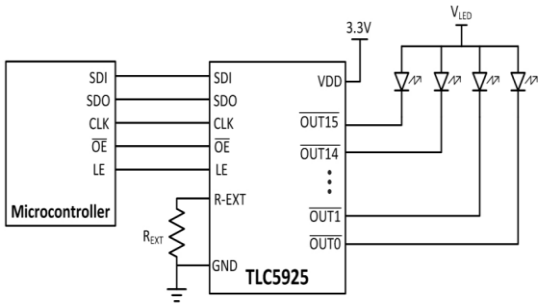
Blue pill (Slika 4) predstavlja razvojno okruženje koje omogućava korisnicima izradu prototipova uz pomoć STM32F103C8T6 mikrokontrolera, [4] koji koristi *ARM Cortex-M3* procesno jezgro sa radnim taktom frekvencije do 72 MHz. Napaja se naponima od 2 do 3.6 V i podržava serijsku UART, SPI i I2C komunikaciju.



Slika 4. Blue pill razvojno okruženje.

B. TLC5925 Constant Current drajver

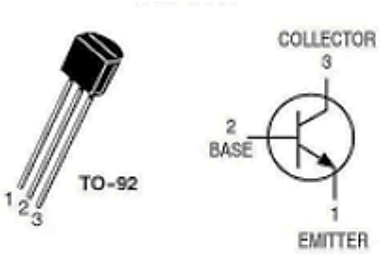
TLC5925 *Constant Current* drajver [5] je dizajniran za LED displeje i aplikacije za LED rasvetu (Slika 5). Kolo TLC5925 sadrži 16-bitni *shift* registar (registar pomeranja) i *latch* za podatke, koji konvertuju serijske ulazne podatke u paralelni izlazni format. Na TLC5925 izlaznom stepenu, 16 portova sa regulisanom strujom obezbeđuju uniformnu struju za pokretanje LE dioda. TLC5925 pruža veliku fleksibilnost i odlične performanse, a koristi se u konstruisanju sistema za aplikacije sa LED displejima. Korisnici mogu podesiti izlaznu struju u opsegu od 3 mA do 45 mA preko eksternog otpornika, R_{ext}, koji daje fleksibilnost u kontroli intenziteta osvetljenja LE dioda. Visoka frekvencija takta (*clock*-a) koja iznosi 30 MHz, takođe zadovoljava sistemske zahteve koji se odnose na brzi prenos podataka velikog obima. Kolo se napaja naponima od 3.3 do 5 V.



Slika 5. Tipična šema povezivanja TLC5925 LED drajvera sa mikrokontrolerom.

C. PN2222 BJT

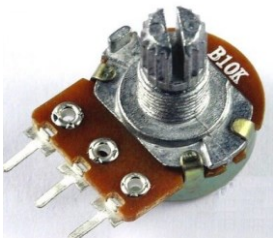
PN2222 [6] je popularan NPN bipolarni tranzistor dostupan u *TO-92* kućištu (Slika 6). On predstavlja tranzistor opšte namene i može se koristiti u širokom spektru aplikacija u svrhe prekidanja i pojačavanja signala. Ovaj tranzistor ima dobre tehničke karakteristike i nisku cenu. Maksimalna kolektorska struja na njemu iznosi 600 mA, što je jedinstveno, jer većina dostupnih tranzistora u *TO-92* kućištu ima maksimalnu kolektorsku struju od 100 mA do 200 mA.



Slika 6. PN2222 bipolarni NPN tranzistor.

D. B10K linearni potenciometar

Predstavlja linearni potenciometar oznake B10K (Slika 7), [7] čija je maksimalna otpornost 10 kΩ. Koristi se za fino podešavanje struje na izlazima TLC5925 drajvera.



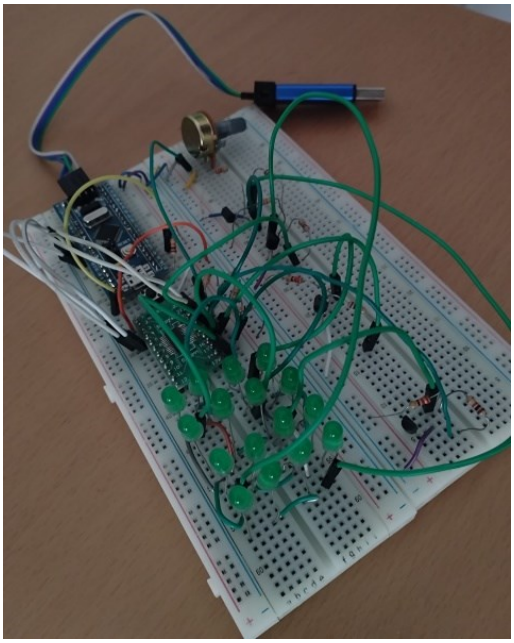
Slika 7. B10K linearni potenciometar.

Pored ovih komponenata upotrebljene su: pet otpornika od 10 kΩ, četiri otpornika od po 220 Ω i 1 kΩ, taster u *pull-up* konfiguraciji, kao i 16 zelenih LE dioda.

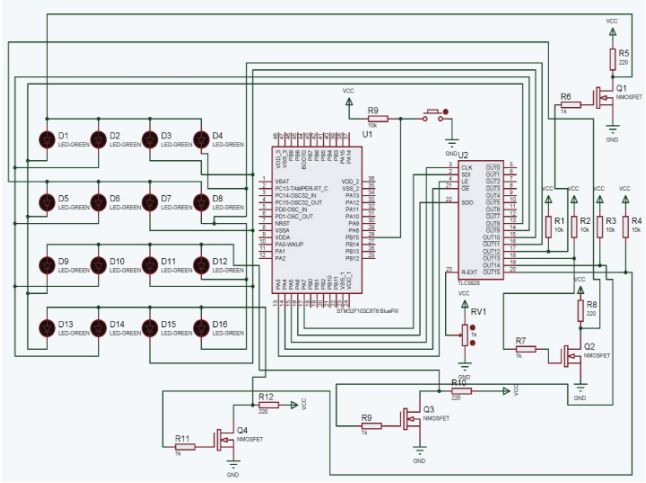
V. OPIS ŠEME SISTEMA

Šema praktične realizacije prikazana je na Slici 8, dok je električna šema prikazana na Slici 9. Posmatrajući električnu šemu može se primetiti da je glavni deo sistema *STM32F103C8T6 - Blue Pill* mikrokontroler, koji je preko pinova PA3 do PA7 povezan sa karakterističnim pinovima TLC5925 drajvera: LE, OE, CLK, SDO, SDI, respektivno. Pin R-EXT je povezan na linearni potenciometar otpornosti od 10 kΩ, preko koga se kontroliše intezitet struje na izlazu. Pin mikrokontrolera PB15 je konfigurisan kao ulazni pin i povezan sa jednim krajem tastera i otpornikom od 10 kΩ u *pull-up* konfiguraciji, pri čemu je drugi kraj tastera povezan na masu. Kada je taster pritisnut, mikrokontroler očitava nisku vrednost signala (*LOW*) i na LED matrici 4x4 se menja svetlosni prikaz LE dioda. Izlazi TLC5925 drajvera OUT8, OUT9, OUT10 i OUT11 su direktno povezani na kolone LED matrice, koje su u konfiguraciji sa zajedničkim katodama. Da bi se uključile LE diode, napon na anodi mora biti veći u odnosu na napon na katodi za minimalno pad napona na diodi. Da bi se to ostvarilo, potrebno je imati kolo koje će signal logičke nule (nizak naponski nivo), koji daju aktivni izlazi TLC5925 drajvera (*active LOW*), konvertovati u signal logičke jedinice, odnosno visok naponski nivo. Za te potrebe se koriste četiri NPN bipolarna tranzistora, pri čemu su izlazni signali drajvera povezani na baze tranzistora, preko otpornika otpornosti 1 kΩ. Emitori tranzistora su povezani na masu, dok su kolektori povezani preko otpornika od 220 Ω na napajanje. Izlazi sa kolektora ovih tranzistora se povezuju linijama veze na redove dioda sa zajedničkim anodama, pri čemu će, ukoliko je određeni izlaz aktivan, biti omogućen protok struje kroz anode LE dioda u tom redu. Da bi željene diode u izabranom redu emitovale svetlost, potrebno je, takođe, da budu aktivni izlazi koji su povezani na kolone zajedničkih katoda u

kojima se nalaze te diode. Još jedna stavka koju treba napomenuti je uloga otpornika otpornosti 10 kΩ na izlazima drajvera OUT12, OUT13, OUT14 i OUT15. Njihov cilj je da izlaze TLC5925 drajvera, koji su u *open drain* konfiguraciji, postave u stanje visoke impedanse. Koncept sa *pull-up* otpornicima se uvodi iz razloga da bi linija veze, koja povezuje izlaz drajvera i bazu BJT-a, bila plivajuća (eng. "*floating*") u slučaju da je izlaz neaktivan. U tom slučaju baza tranzistora ne očitava nikakav signal. Kada se izlaz aktivira, odnosno pošalje signal niskog logičkog nivoa kroz liniju veze, on se prebacuje u stanje niske impedanse i linija veze postaje aktivna, odnosno baza tranzistora očitava signal koji je poslat sa izlaza, gde se dalje dešava inverzija signala sa niskog na visoki logički nivo, koji se dalje šalje redovima LED matrice sa zajedničkim anodama.



Slika 8. Šema praktične realizacije projekta na proto ploči.

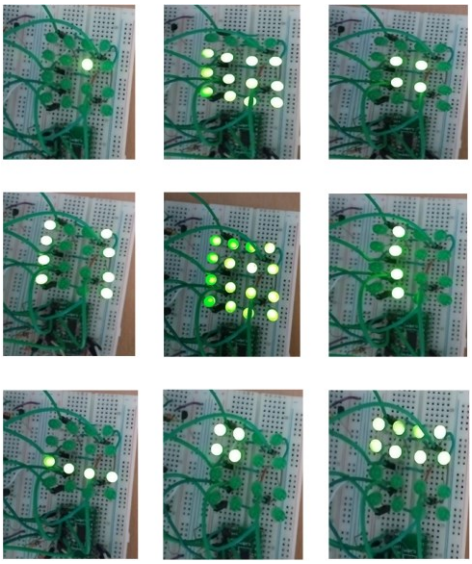


Slika 9. Šema projekta u Proteusu.

VI. IMPLEMENTACIJA PROGRAMSKOG KODA

Na Slici 10 su date proizvoljno izabrane kombinacije LE dioda u matrici, dobijene prosleđivanjem programskog koda, napisanog u *Keil MDK* okruženju, na mikrokontroler. Da bi se uključila treća dioda u drugom redu LED matrice, potrebno je da budu uključeni drugi red i treća kolona. To znači da moraju da se aktiviraju izlazi OUT13 i OUT10. Pošto drajver TLC5925 podržava SPI komunikaciju, potrebno je da mu se proslede dva niza osmobitnih podataka sa mikrokontrolera. Prvi niz, neka se zove pod[1], mora proslediti sve nule (00000000 u binarnom formatu) izlazima drajvera, zbog toga što se izlazi od OUT0 do OUT7 ne koriste. U kodu bi to bilo predstavljeno kao pod[1]=0x00 (u heksadecimalnom formatu). Drugi niz, pod[0] mora proslediti odgovarajuću kombinaciju izlazima drajvera, koja u konkretnom slučaju uključuje drugi red i treću kolonu. U binarnom formatu je potrebno poslati niz binarnih vrednosti 00100100 izlazu drajvera, što bi se u heksadecimalnom formatu prevelo kao pod[0]=0x24. Viša nibla u navedenom binarnom formatu (0010) se odnosi na red koji se uključuje, dok niža nibla (0100) predstavlja kolonu koja se uključuje. Prosleđivanjem ova dva niza podataka do drajvera se aktiviraju izlazi OUT13 i OUT10, pri čemu se uključuje drugi red i treća kolona, usled čega odgovarajuća LE dioda počinje da emituje svetlost.

U slučaju da treba da se uključe sve LE diode prvog i drugog reda istovremeno, potrebno je poslati niz binarnih podataka 00111111, to bi u heksadecimalnom format bilo 0x3F. Da bi se uključila prva i poslednja kolona istovremeno, niz binarnih podataka bi iznosio 11111001, što bi u heksadecimalnom formatu bilo 0xF9. Ako je potrebno da se uključe sve LE diode u matrici, u binarnom formatu niz koji se prosleđuje bi iznosio 11111111, dok bi u heksadecimalnom formatu iznosio 0xFF.



Slika 10. Različite kombinacije LE dioda u LED matrici dobijene prosleđivanjem koda na mikrokontroler.

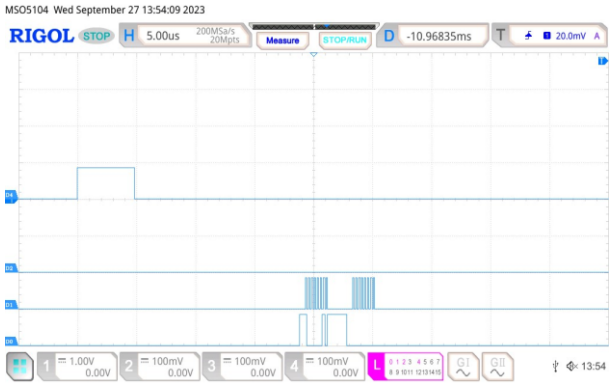
VII. ANALIZA SIGNALA PREKO LOGIČKOG ANALIZATORA - OSCILOSKOPA

Osciloskop koji je korišćen za analizu signala je *Rigol MSO5104* (Slika 11). On predstavlja osciloskop sa 4 kanala koji ima širinu propusnog opsega od 100 MHz. Ovaj osciloskop je opremljen 9-inčnim kapacitivnim ekranom osetljivim na dodir, što olakšava njegovo korišćenje. Radi prenosa izmerenih podataka, moguće je povezati *MS05104* sa računarom preko različitih interfejsa, kao što su *LAN*, *USB* ili direktno sa monitorom putem *HDMI* konekcije. [8]



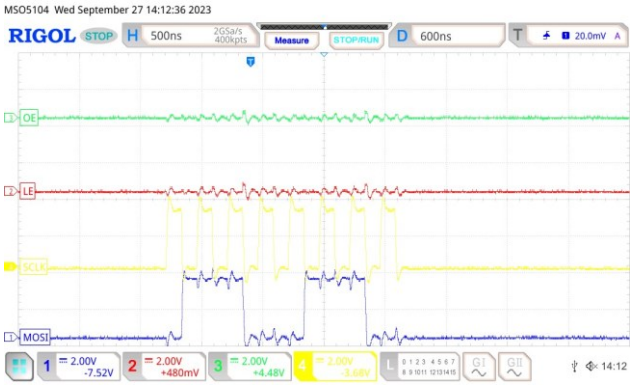
Slika 11. Izgled Rigol MSO5104 osciloskopa.

Na Slici 12 prikazan je proces slanja podataka preko TLC5925 drajvera, usnimljen osciloskopom. Sa slike se može videti da se signal na kanalu 4 osciloskopa označen kao D4, koji se dobija sondom povezanom sa kanala 4 na pin LE drajvera, sa logičke nule podiže na logičku jedinicu i ponovo vraća na logičku nulu, što označava početak prenosa podataka. Sonda predstavlja uređaj za povezivanje osciloskopa koja mu omogućava da prima i prikazuje električni signal. Ujedno se može primetiti da je kanal 2 osciloskopa označen sa D2, koji je povezan na pin OE takođe sondom, sve vreme na logičkoj nuli, pri čemu je omogućen prenos podataka. Bitno je napomenuti da je u toku povezivanja, sonda svakog kanala osciloskopa jednim krajem povezana na masu.



Slika 12. Proces slanja podataka usnimljen preko osciloskopa.

Na Slici 13, kanal 1 je sondom povezan na SDI pin drajvera i očitava signal sa njega. MOSI pin mikrokontrolera je povezan na SDI pin TLC5925 drajvera, pri čemu on sadrži informacije o aktivnim i neaktivnim izlazima koje se dalje prosleđuju drajveru. Informacije su u vidu pravougaone povorke impulsa, koje se upoređuju sa uzlaznom ivicom taktnog signala (CLK), koji je prikazan na kanalu 2 osciloskopa. Ako je u trenutku okidanja taktnog signala, na signalu pina MOSI (SDI) visok naponski nivo, uključiće se odgovarajući izlaz drajvera. Niz osmobitnih podataka koji se u ovom slučaju prosleđuje drajveru je 01100110, tj. uključiće se drugi i treći red, kao i druga i treća kolona, što će rezultirati da četiri središnje LE diode u matrici emituju svetlost.



Slika 13. Osmobitni podatak 01100110 koji se prosleđuje drajveru.

VIII. ZAKLJUČAK

Iz svega prethodno opisanog, može se zaključiti da je ovaj rad dao doprinos razumevanju i praktičnoj primeni *Constant Current* LED drajvera u svetlosnim aplikacijama. Implementirani sistem omogućava brzu i tačnu promenu svetlosnih efekata koji daju LE diode u matrici 4x4, čineći ga korisnim za različite primene, od kojih su neke upotreba u reklamnim displejima, ambijentalnom osvetljenju, kao i u drugim svetlosnim aplikacijama.

ZAHVALNICA

Rad je realizovan kao diplomski rad iz predmeta Optoelektronika, na modulu Elektronske komponente i mikrosistemi. Autor se zahvaljuje Vesni Paunović i Nikoli Mitroviću na pruženom znanju i pomoći.

LITERATURA

[1] Tipovi LED displeja i njihova uloga, dostupno na: https://www.unilumin.com/blog/types-of-led-display.html
[2] Osnovne informacije o *Constant Current* LED drajverima, dostupni na:https://assets.nexperia.com/documents/application-note/AN90026.pdf

[3] LED matrice i funkcija multipleksiranja, dostupno na: https://docs.broadcom.com/doc/AV02-3697EN?ICID=I-CT-TECH-RES-CLA-SEP_21-0
[4] Osnovne informacije i primene *STM32F103C8T6-Blue-Pill* mikrokontrolera, koje su dostupne na: https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html
[5] Tehnička dokumentacija TLC5925 LED drajvera, ujedno sa slikama, dostupna na: https://www.ti.com/lit/gpn/tlc5925
[6] PN2222 NPN bipolarni tranzistor, dostupno na: https://www.componentsinfo.com/pn2222/
[7] B10K potenciometar i njegove glavne karakteristike: https://datasheetspdf.com/pdf/866974/ALPHA/RV24AF-10-40R1-B10K/1
[8] Tehnička dokumentacija Rigol MSO5104 osciloskopa, dostupna na: https://www.allaboutcircuits.com/electronic-components/datasheet/MSO5104--Rigol/

Realizacija uređaja za testiranje brzine refleksnih reakcija

Milica Petrušijević

Sadržaj – Ovaj rad detaljno istražuje strukturu i funkcionalnost uređaja za testiranje brzine refleksnih reakcija. Ključna komponenta ovog uređaja je NE555 integrisano kolo, čija šema i uloga u okviru sistema su pažljivo analizirane. Pored toga, u sklopu NE555 kola, koristi se dekadni brojač CD4026, koji obavlja ulogu dekodera i takođe je podvrgnut detaljnom opisu.U radu je pružen dubok uvid u funkcionisanje čitavog kola, uz dodatak šeme uređaja i PCB layout-a, koji su kreirani korišćenjem programa Altium Designer. Glavni fokus ovog istraživanja bio je na samoj realizaciji uređaja za merenje brzine refleksnih reakcija, istovremeno pružajući detaljan pregled strukture i funkcionalnosti sistema.Kao dodatak analizi, struktura uređaja je smeštena u kutiju i prikazana je kao 3D model u programu Solid Edge.

I. Uvod

Brzina refleksnih reakcija predstavlja ključnu karakteristiku ljudske percepcije i sposobnosti za brzu i preciznu reakciju na spoljne stimulanse.

U ovom radu, fokus je na realizaciji merača brzine refleksnih reakcija koji koristi NE555 integrisano kolo u astabilnom režimu kao generator taktnih signala.NE555 je dobro poznato i široko korišćeno integrisano kolo koje se često koristi kao astabilni multivibrator za generisanje kvadratnih signala. CD4026 dekoderi s druge strane, omogućavaju brojanje i obradu tih signala, dok sedmosegmentni displeji pružaju vizuelni prikaz merenih vrednosti.

U nastavku ovog rada, detaljno će biti razmotren princip rada ovog sistema za merenje brzine refleksnih reakcija, uključujući električne karakteristike NE555 i CD4026 kola, kao i tehničke detalje vezane za povezivanje ovih komponenti.

II. Opis korišćenih komponenata

A. NE555 kolo

Integrisano kolo NE555 [1] je jedno od najčešće upotrebljavanih integrisanih kola. Može raditi u tri različita moda: monostabilnom, astabilnom i bistabilnom modu rada. Sačinjen je od četiri dela gde prvi deo čini tri otpornika spojena u nizu, svaki sa vrednošću od 5 kΩ, zbog čega se i naziva 555 tajmerom. Ovi otpornici formiraju naponski razdelnik, pa se napon sa delova razdelnika prosleđuje drugim delovima kola [2].

M. Petrušijević, student master studija Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milica.petrusijevic@elfak.rs

Drugu celinu čini komparatorski blok, sačinjen od dva komparatora. Izlazi oba komparatora povezani su na ulaz flip flopa. SR flip-flop čini treću celinu kola. Ovo kolo sadrži dva ulaza i dva izlaza. Ulazi flip-flopa su obeleženi sa S (set) i R (reset), a izlazi sa Q i sa Q (komplement izlazu Q). Četvrtu celinu kola NE555 čini bipolarni trazistor. Emitor je povezan sa masom, kolektor je povezan sa pinom Discharge a baza je povezana sa pinom Q flip-flopa, tranzistor ima ulogu prekidača, što znači da radi u oblasti zasićenja (kada je prekidač otvoren) i oblasti zakočenja (kada je prekidač zatvoren). Izlaz flip-flopa određuje da li će prekidač biti otvoren ili ne.

B. CD4026 brojač

CD4026 [3] je integrisano kolo brojača sa dekodiranim izlazima za upravljanje sedmosegmentnim LED displejom sa zajedničkom katodom. Zasnovan je na CMOS tehnologiji i radi od samo 3 V (minimum), do 18 V maksimalno, a apsolutni maksimum je 20 V. CD4026 ima izlaze za upravljanje svakim segmentom sedmosegmentnog displeja. Ovi izlazi se povezuju sa odgovarajućim segmentima displeja radi prikazivanja brojeva. Sadrži interni brojač koji se povećava svaki put kad primi impulse na CLK ulazu [4].

C. Sedmosegmentni displej

Sedmosegmentni displej [5] predstavlja ključnu komponentu u elektronskim uređajima za prikazivanje brojčanih informacija.Ovaj tip displeja sastoji se od sedam segmentnih LED (*Light Emitting Diode*) elemenata koji su organizovani tako da mogu prikazati različite cifre i neke osnovne simbole. Osnovna podela sedmosegmentnih displejeva je na sedmosegmentne displeje sa zajedničkom anodom i sedmosegmentne displeje sa zajedničkom katodom kojima se za aktivaciju dovodi napon ili masa [6] [7].

III. Funkcionalnost i opis rada uređaja

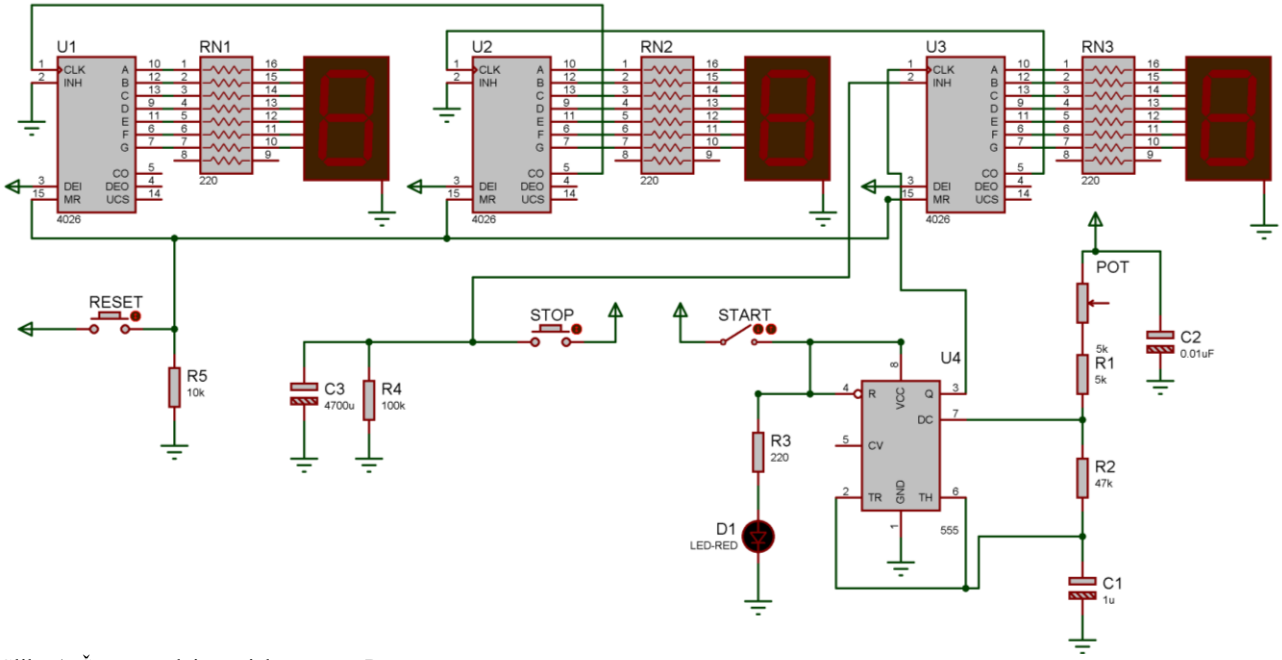
U ovom radu projektovan je reakciometar odnosno tajmer za merenje brzine refleksnih reakcija koji radi po principu štoperice. Princip rada se može se opisati na sledeći način: U celom postupku učestvuju dve osobe, ispitivač (doktor) i ispitanik (pacijent). Priključenjem adaptera u utičnicu uređaj se startuje i prikazuje 00.0 (u sekundama). Kada doktor pritisne taster START, uređaj

kreće da broji i aktivira se LED dioda. Ispitanik odnosno pacijent ima zadatak da čim ugleda aktiviranu LED diodu pritisne taster STOP. Pri pritisku tastera STOP, brojanje se zaustavlja i očitava se vreme koje je isteklo od aktiviranja. Očitano vreme naziva se vreme refleksne reakcije. Postoji mogućnost resetovanja pritiskom tastera RESET, gde se brojčanik vraća na nulu odnosno 00.0 s i ceo postupak se može ponoviti.

Za ovaj princip rada iskorišćen je NE555 tajmer u astabilnom režimu, pri čemu daje pravougaone signale na izlazu i ima ulogu da daje taktne signale ovom sistema. Na izlaz NE555 tajmera kaskadno su vezana tri CD4026 dekadna brojača koji su preko otporne mreže vezani za sedmosegmentne displeje. CD4026 reaguje na prednju ivicu signala i svaki put kad do njega dođe nova ivica, interni brojač se inkrementira odnosno poveća za jedan. Sedmosegmentni displej za to vreme preko CD4026 prikazuje te brojeve. Kad prvi brojač izbroji do 9 sledećom prednjom ivicom koja mu dolazi aktivira drugi brojač odnosno on broji do 9 i sledeći impuls prosleđuje brojaču koji je kaskadno vezan za njega a on se vraća na nulu i ponovo kreće svoj krug od 9 inkrementa gde 10-ti prebacuje susednom brojaču [8].

IV. PROJEKTOVANJE UREĐAJA

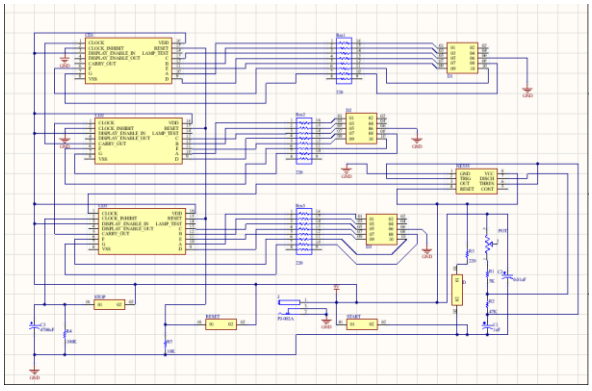
Na slici 1 prikazana je projektovana šema u okruženju Proteus 8 Professional. Korišćene komponente su:
NE555 kolo, CD4026B brojač, otporna mreža 220Ω, otpornici R1 = 5 kΩ, R2 = 47 kΩ, R3 = 220 Ω, R4 = 100 kΩ, R5 = 10 kΩ, kondenzatori C1 = 1 μF, C2 = 0.01 μF, C3 = 4700 μF, potenciometar POT = 5 kΩ, crvena LED dioda D1, tasteri STOP i RESET, prekidač START.



Slika 1. Šema uređaja projektovana u Proteusu.

A. Projektovanje PCB pločice uređaja u programu ALTIUM DESIGNER

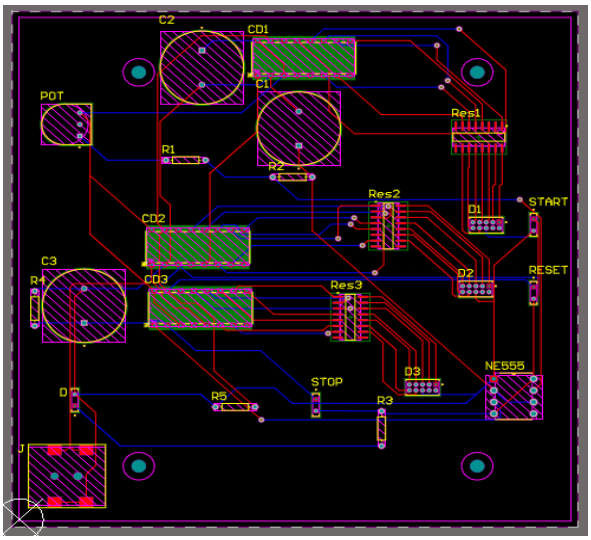
Proces izrade PCB pločice uređaja započinje kreiranjem šeme u okviru dela Altium Schematic. Ovaj korak uključuje uvoz modela komponenata koje će se kasnije postaviti na PCB pločicu. Pored toga, u ovom delu se definišu i povezivanja između komponenata, što je ključno za uspešno rutiranje tragova na PCB pločici, na slici 2 vidi se prikaz šeme na kojoj su komponente adekvatno povezane.



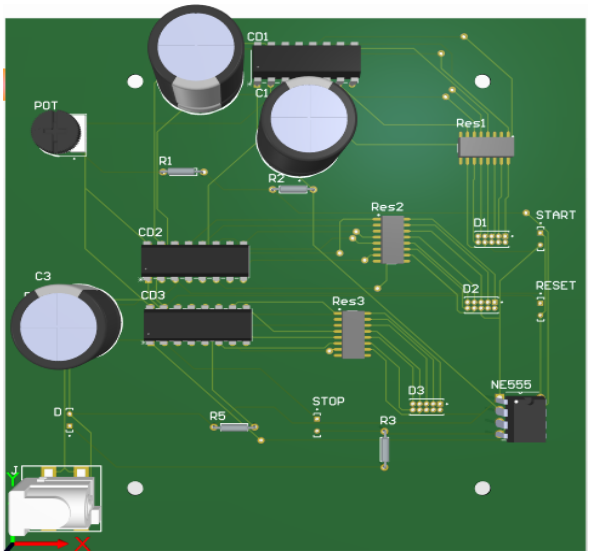
Slika 2. Prikaz šeme uređaja u programu Altium Designer.

B. Raspoređivanje komponenti na ploču u 2D i 3D režimu

Nakon što je uspešno izrađena električna šema uređaja, prelazi se na sledeći korak u dizajnu - kreiranje PCB (*Printed Circuit Board*) pločice. U okviru PCB dokumenta, prvo je potrebno uvesti samu pločicu na kojoj će se smestiti sve komponente, a takođe i definisati kako će komponente biti raspoređene na njoj. Prilikom ovog procesa, ključno je da komponente koje se uvoze imaju odgovarajući Footprint, odnosno šablon za postavljanje na pločicu. Osim toga, za potpunu preciznost i vizualizaciju dizajna formiraju se 2D i 3D model pločice koji su predstavljeni na slici 3 i slici 4. U delu Components pronalaze se odgovarajuće biblioteke i iz nje se uvozi određene 2D i 3D modeli kao i footprint-ovi za komponente u suprotnom se mogu preuzeti preko online biblioteka.



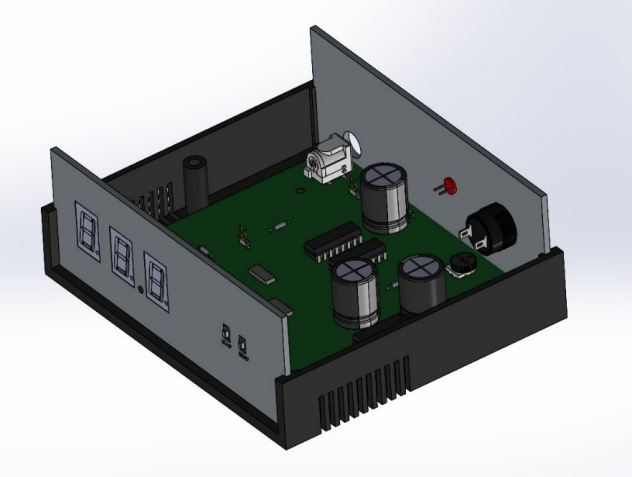
Slika 3. 2D prikaz pločice nakon rutiranja.



Slika 4. 3D prikaz pločice nakon rutiranja.

C. Smeštanje pločice u kutiju u programu SOLID EDGE

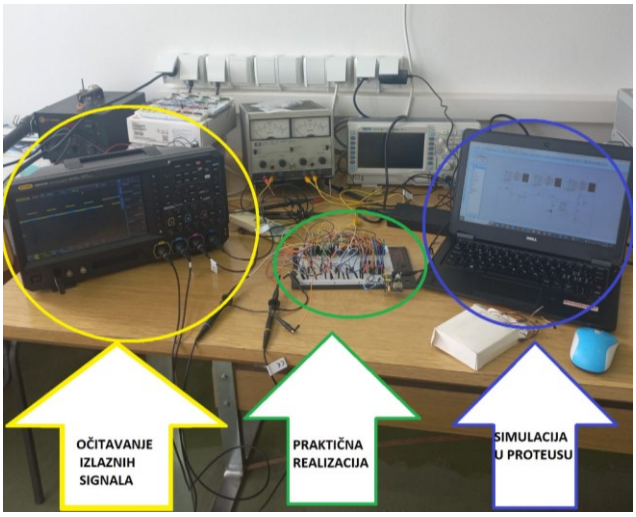
Nakon uspešnog završetka procesa rutiranja, provere veza i šeme u programu Altium Designer, sledeći korak u razvoju ovog uređaja je prenošenje dizajna u Solid Edge. Uvoz STEP fajla iz Altium Designer-a u Solid Edge omogućava detaljno modeliranje i montažu, pružajući alate za precizno pozicioniranje i integraciju svih komponenata u stvarnom 3D okruženju. Krajnji proizvod bez gornjeg dela kutije, radi bolje preglednosti, je prikazan na slici 5.



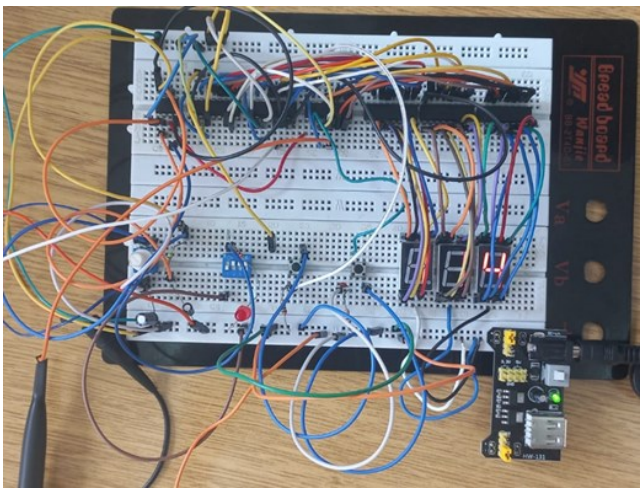
Slika 5. Prikaz celog uređaja bez gornjeg dela.

D. Praktična realizacija uređaja na protoploči

Praktičnom realizacijom i očitavanjem signala vrši se provera rada uređaja. Na slici 6 je prikazano radno okruženje u kome se sa protopločice (slika 7) očitavaju izlazni signali na osciloskopu (slikama 8 i 9).

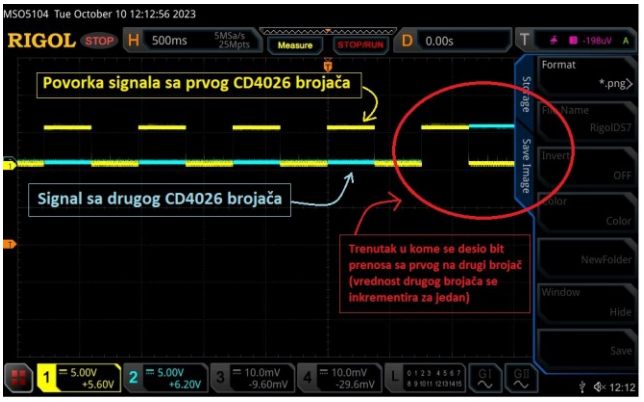


Slika 6. Izgled radnog okruženja.



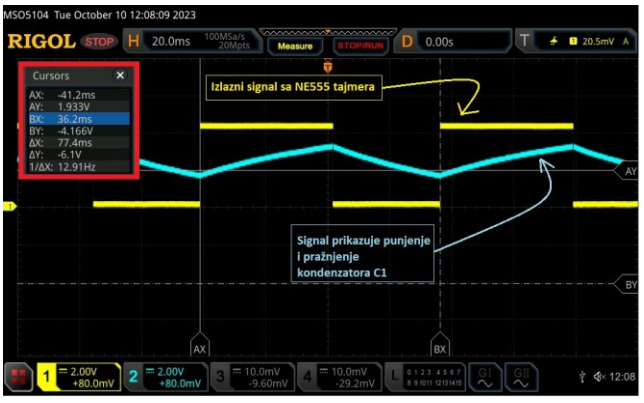
Slika 7. Šema projekta povezana na protoploču.

Na slici 8 prikazan je trenutak u kome se dešava bit prenosa sa prvog na drugi brojač i plavi signal prelazi sa logičke nule na logičku jedinicu.



Slika 8. Očitavanje signala sa prvog i drugog brojača.

Na slici 9 može se videti izlaznog signal sa NE555 kola i kondenzatora C1.



Slika 9. Očitavanje signala sa NE555 kola i kondenzatora C1.

V. Zaključak

Kroz analizu strukture i funkcionalnosti uređaja i realizovanju makete, istraživanje je omogućilo dublje razumevanje kako ovaj uređaji funkcioniše i kako se može unaprediti. Navedene su neke od mana, kao što su ograničen broj stimulansa i fizička veličina uređaja. Dodavanje različitih tipova stimulansa, dodavanje softvera za analizu, integracija nekih senzora i poboljšanje dizajna uređaja su neki od mogućih koraka ka budućim poboljšanjima. Sa aspekta praktične primene ovaj uređaj ima veoma veliki značaj i svakako je potrebno nastaviti istraživanja u ovoj oblasti.

Zahvalnica

Zahvalila bih se profesoru Danijelu Dankoviću na otvorenosti za sva pitanja tokom izrade ovog rada kao i asistentu Nikoli Mitroviću za pomoć pri praktičnoj realizaciji. Ovaj rad predstavlja deo istraživanja iz diplomskog rada Milice Petrušijević pod nazivom „Realizacija uređaja za testiranje brzine refleksnih reakcija“ odbranjenog na Elektronskom fakultetu u Nišu, na modulu Elektronske komponente i mikrosistemi.

Literatura

[1] N. Mitrović, D. Danković, Z. Prijić „Praktikum za analognu mikroelektroniku“ 109-124 može se pristupiti preko linka https://mikro.elfak.ni.ac.rs/wp-content/uploads/Analogna-mikroelektronika-praktikum-v3.pdf
[2] Tehnicka dokumentacija za NE555 tajmer https://www.ti.com/lit/ds/symlink/se555.pdf
[3] Tehnicka dokumentacija za CD4026 https://www.ti.com/lit/ds/symlink/cd4026b.pdf
[4] Materijal o CD4026 dekoderu https://www.petervis.com/GCSE_Design_and_Technology_Electronic_Products/4026-counter-circuits/4026-counter-circuit.html
[5] Tehnicka dokumentacija za sedmosegmentni led displej https://www.alldatasheet.com/datasheet-pdf/pdf/159535/KINGBRIGHT/SC56-11SRWA.html
[6] Materijal u vezi sedmosegmentnog displeja- https://components101.com/displays/7-segment-display-pinout-working-datasheet
[7] M.Đorđević “Praktikum za optoelektroniku“ 53-56 može se pristupiti preko linka https://mikro.elfak.ni.ac.rs/wp-content/uploads/Praktikum-za-Optoelektroniku2020.pdf
[8] Rad iskorišćen kao ugledni primer https://ieee.elfak.ni.ac.rs/wp-content/uploads/2021/03/2018.pdf - Digitalna štoperica- - Stefan Nikolić

Razvoj meteorološkog sistema za prevenciju od bujica

Dimitrije Anđelković

Sadržaj - U radu biće predstavljen deo razvoja meteorološkog sistema za rano upozoravanje i prevenciju od bujičnih poplava u Srbiji. Poplave predstavljaju ozbiljnu pretnju po ljudske živote i imovinu kada se njima ne upravlja efikasno. Sistem koji je razvijen, dizajniran je tako da koristi tehnologiju praćenja parametara u realnom vremenu i ranog upozoravanja za predviđanje bujičnih poplava i upozoravanje zajednica u oblastima visokog rizika. Takođe tabelarno će biti predstavljeni rezultati merenja sa četiri različite lokacije u okolini Niša.

I. UVOD

Poplava je prirodna pojava koja se dešava kada se voda izlije iz svojih uobičajenih korita i prekrije obližnju zemlju. Izlivanje vode može da se dogodi kada padne velika količina kiše, kada se otapaju snegovi u planinama ili kada se nivo vode u rekama i jezerima poveća usred nekontrolisanih antropogenih delovanja. Poplave mogu da predstavljaju opasnost po ljudske živote i dovode do oštećenja imovine i infrastrukture. U nekim slučajevima, poplave mogu da budu i epidemiološke opasnosti, zbog potencijalne kontaminacije vode.

Poplave jedino dovode do štete kada čovek pogrešno koristi vodu i zemljište. Što se intenzivnije i neprikladnije koristi plavna zona, to je veći potencijal za štetu, a u uslovima poplava veća je stvarna šteta. Iako su poplave prirodna pojava, njihove posledice mogu biti katastrofalne kada utiču na ljude i infrastrukturu.

Republika Srbija je stalno u opasnosti od pojave poplava. U 2014. godini Srbiju su pogodile razorne poplave koje su odnele mnoge živote, raselile hiljade ljudi i izazvale ogromnu štetu. Poplave 2014. godine su događaj koji predstavlja osnovnu motivaciju razvoja meteorološkog sistema za prevenciju od bujica i bujičnih poplava.

Značaj rada i samog istraživanja je u tome da ako se meteorološka stanica za monitoring i ranu najavu adekvatno implementira i koristi, može:

- spasiti ljudske živote,
- poboljšati komunikaciju u vanrednim uslovima,
- povećati vreme reagovanja stanovnika pre nailaska bujične poplave,
- dokumentovati promene vodostaja u svakom trenutku,
- poboljšati monitoring vodotokova,
- proširiti postojeću bazu podataka,
- smanjiti potencijalnu materijalnu štetu.

D. Anđelković je student master akademskih studija na modulu Elektronika i mikrosistemi, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dimitrijeandjelkovic@gmail.com

Meteorološki sistemi za monitoring i prevenciju od bujičnih poplava spadaju u sisteme i tehnologije najnovijih generacija, zbog toga se može naći više podeljenih mišljenja po stavu generalizacije arhitekture meteoroloških stanica i njihovoj implementaciji u sisteme za monitoring. Generalno prihvaćen stav po nekim autorima je da arhitektura meteoroloških stanica i njihova implementacija u sisteme za monitoring i ranu najavu bujičnih poplava može biti podeljena u četiri celine. A to su:

- senzorni deo sistema, koji služi za prikupljanje podataka iz okoline,
- transferni deo sistema, koji služi za prenos prikupljenih podataka,
- baza podataka u kojoj se prikupljeni podaci čuvaju i obrađuju,
- grupa softverskih alata koja služi za vizuelizaciju, analizu i potencijalnu ranu najavu poplava.

II. GENERALNO O METEOROLOŠKOM SISTEMU ZA PREVENCIJU OD BUJICA

Zbog opširnosti i kompleksnosti celog sistema u ovom radu biće predstavljen samo konceptualni dokaz o radu Sistema, kao i prikupljena merenja sa terena.

Dokaz o radu meteorološkog sistema za prevenciju od bujičnih poplava u ovom radu podrazumeva tabelarni prikaz prikupljenih parametara iz okoline sa različitih lokacija. Za sada sistem prikuplja podatke poput parametara temperature, vlažnosti vazduha, pritiska, količine osvetljenja, vlažnost zemljišta i različitih GPS parametara kao što su geografska širina, geografska dužina, nadmorska visina, broj satelita u vidokrugu senzora za određivanje pozicije sistema. Sistem ima mogućnost jednostavne obrade i transformacije prikupljenih podataka iz okoline i slanje tih podataka na trenutni lokalni *web server* gde je odrađena i vizuelizacija nekih podataka, kao što su temperatura i vlažnost vazduha.

Sama meteorološka stanica za monitoring se lako gradi, jeftina je za servisiranje i laka za održavanje, pri tom može da pruži ogromne benefite.

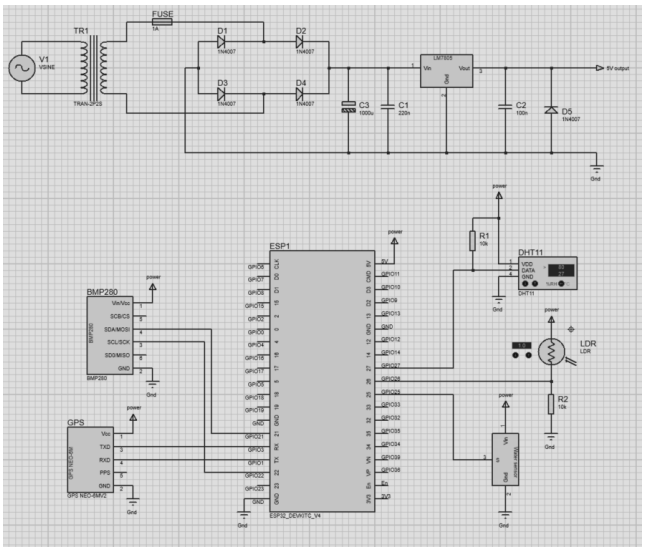
Generalno gledano na slici 1 prikazana je kompletna električna šema uređaja. Kao što se može primetiti, električna šema se sastoji iz dva glavna bloka, a to su blok za napajanje i blok koji sadrži mikrokontroler i senzore.

Razvojno okruženje korišćeno za realizaciju projekta poznato je pod nazivom *ESP32_devkitC_V4* [1]. Postoji više načina napajanja navedenog razvojnog okruženja. Prilikom realizacije projekta odlučeno je da projektovani sistem može jednostavno da se priključi na mrežni napon od 220 V, kakav postoji u svim domaćinstvima. Priključivanje na

mrežni napon možda nije najpraktičniji pristup pri korišćenju uređaja na terenu, ali je sasvim dovoljan pristup u demonstrativne svrhe.

Prilikom napajanja uređaja, jako je bitno koristiti samo jedan izvor napajanja istovremeno. Na primer, ako koristimo bateriju od 5 V i u istom trenutku pokušamo da priključimo uređaj pomoću USB- a radi programiranja, sigurno će doći do oštećenja razvojnog okruženja, a vrlo verovatno i do računara koji se koristi za programiranje.

Programiranje *ESP32_devkitC_V4* razvojnog okruženja se vrši pomoću USB priključka, dok je sam kod pisan i razvijan uz pomoć integrisanog razvojnog okruženja *Arduino IDE* (*Arduino Integrated Development Environment*). U ovom radu neće biti prikaza koda, kao ni kreiranog privremenog *web servera* za povezivanje ESP32 okruženja za vizuelizaciju izmerenih parametara.



Slika 1. Šematski prikaz celog sistema.

III. SISTEM NAPAJANJA

U svetu povezanom tehnologijom, postoji suštinski faktor koji često prolazi neprimećeno, a to je sistem napajanja. Ova kompleksna mreža električne infrastrukture čini osnovu modernog sveta u kome živimo.

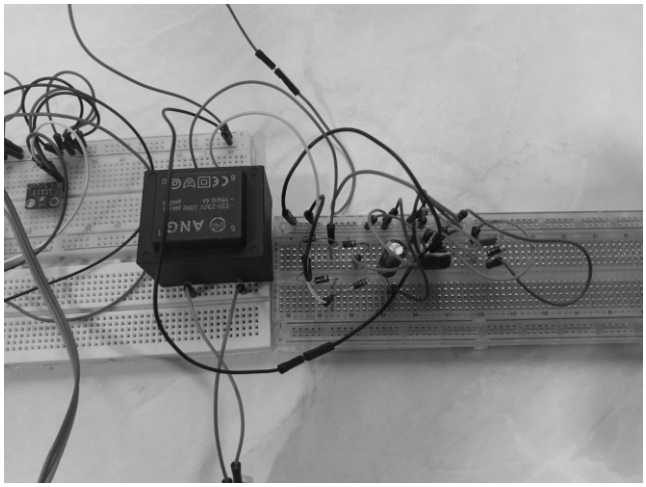
Sistem za napajanje je pretvarač koji od naizmeničnog napona pravi jednosmerni regulisani napon, slično kao punjač za mobilne telefone. Sistem za napajanje u ovom slučaju prevodi naizmenični napon od 220 V ili 230 V u jednosmerni regulisani napon od 5 V. Regulisani izlaz sistema za napajanje se dalje dovodi na napajanje senzora i napajanje razvojnog okruženja.

Potrebne komponente za realizaciju sistema za napajanje su:

- silazni transformator sa 230 V na 12 V,
- osigurač jačine 1 A,
- četiri 1N4007 PN diode za realizaciju ispravljača napona (Grecov spoj) [2-3],
- kondenzator 0.1 μF,
- integrisano kolo L7805 kao regulator napona,
- kondenzator 0.22 μF,
- jedna 1N4007 PN dioda za sprečavanje naponskih premašenja,
- elektrolitski kondenzator od 1000 μF.

Gledano sa leve strane na slici 1 naizmenično mrežno napajanje se prvo dovodi do transformatora, koji služi za spuštanje napona sa 230 V na 12 V. Primar kalema je povezan na mrežni napon, dok se sekundar dovodi do Grecovog spoja. Osigurač od 1 A (preporučuje se 600 mA) postavljen je između transformatora i Grecovog spoja. Osigurač će ograničiti struju koju kolo maksimalno može da povuče na 1 A. Nakon Grecovog spoja, dobija se ispravljen jednosmerni napon koji se stabilizuje pomoću elektrolitskog kondenzatora C_3 od 1000 μF. Ovako stabilisan signal se dovodi na integrisano kolo L7805, koje će na izlazu dati jednosmeran regulisan napon od 5 V. Regulisan jednosmeran napon od 5 V se kasnije koristi za napajanje svih senzora i razvojnog okruženja.

Slika 2 predstavlja grub prikaz realizovanog sistema za napajanje. U planu je prebaciti komponente sa protoploče na PCB (engl. *Printed Circuit Board*) ploču i koristiti kombinaciju SMD (engl. *Surface Mount Device*) i TH (engl. *Throught Hole*) komponenata.



Slika 2. Izgled realizovanog sistema za napajanje.

IV. RAZVOJNO OKRUŽENJE *ESP32_DEVKITC_V4*

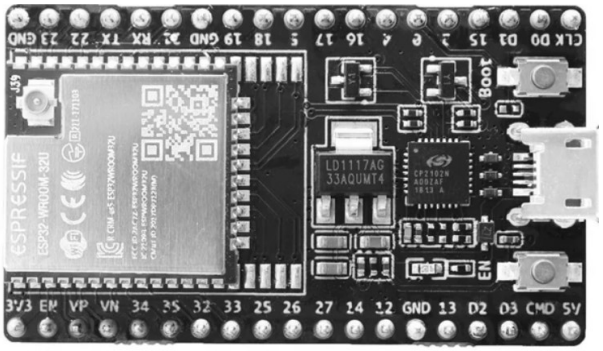
ESP32 je razvojno okruženje koje pruža široke mogućnosti u upravljanju i komunikaciji sa različitim elektronskim komponentama [4]. ESP32 je poznat po svojoj snažnoj procesorskoj jedinici, velikom broju perifernih interfejsa, ugrađenom Wi-Fi i Bluetooth podrškom. ESP32 se koristi u mnogim projektima u oblasti interneta stvari (engl. IoT - *Internet of Things*), robotike, senzornih aplikacija i uprošćenom upravljanju elektronike. ESP32 razvojna okruženja pružaju mogućnost spajanja i kontrole različitih senzora, aktuatora i drugih elektronskih

komponenata, što čini ovaj mikrokontroler idealnim za razvoj i testiranje raznovrsnih ideja.

ESP32_devkitC_V4 razvojni modul može imati široku primenu u meteorološkim sistemima, kao i sistemima za monitoring i ranu najavu. Mogućnost komunikacije preko Wi-Fi i Bluetooth veze, uz snažnu procesorsku jedinicu, omogućava ovom modulu da prikuplja i obrađuje podatke sa različitih senzora i komunicira sa drugim uređajima.

U meteorološkim sistemima i sistemima za monitoring i ranu najavu, *ESP32_devkitC_V4* modul se može koristiti za prikupljanje podataka o okolini, kao što su temperatura, vlažnost vazduha, vlažnost zemljišta, osvetljenost, itd. Nakon što se podaci prikupe, modul ih može analizirati i proslediti u adekvatnom formatu na centralnu upravljačku stanicu ili *web server* radi dodatne obrade, prikupljanja ili prikaza na korisničkom interfejsu. Ovakve karakteristike sistema omogućavaju operatorima sistema da brzo i efikasno primete promene u okruženju i reaguju u skladu sa njima.

Za sisteme rane najave, *ESP32_devkitC_V4* modul može detektovati i obraditi signale ili alarme iz različitih izvora. Na osnovu detekcije, modul može generisati upozorenja ili aktivirati odgovarajuće akcije ili servise, kao što su aktiviranje zvukovnog alarma ili obaveštavanje odgovornih lica putem poruka ili elektronske pošte.



Slika 3. Prikaz *ESP32_devkitC_V4* razvojnog modula.

V. SENZOR VLAGE I TEMPERATURE *DHT11*

U ovom radu, biće predstavljen kratak opis samo jednog senzora, radi sticanja ideje o čemu je sve potrebno voditi računa prilikom izbora senzora radi upotrebe u određeni sistem [5]. Neki od faktora prilikom izbora su:

- brzina odziva,
- ponovljivost,
- preciznost,
- dugoročna stabilnost,
- zamenljivost,
- sposobnost oporavka od kondenzacije,
- otpornost na fizičke i hemijske zagađivače,
- pakovanje,
- veličina senzora,
- isplativost.

Senzor temperature i vlažnosti *DHT11* sadrži temperaturni senzor i senzor vlažnosti vazduha sa kalibrisanim digitalnim izlazom signala. Koristeći ekskluzivnu tehniku akvizicije digitalnog signala i tehnologiju senzora temperature i vlažnosti, *DHT11* osigurava visoku pouzdanost i odličnu dugoročnu stabilnost. *DHT11* senzor uključuje komponentu za merenje vlažnosti vazduha otpornog tipa i komponentu za merenje temperature NTC. Komponente za merenje vlažnosti vazduha i temperature se povezuju sa 8 - bitnim mikrokontrolerom visokih performansi, nudeći odličan kvalitet, brz odziv, otpornost na šumove i jako bitno senzorsku ekonomičnost.



Slika 4. Senzor *DHT11*.

Svaka *DHT11* komponenta je kalibrisana u laboratoriji, što je izuzetno važno zbog same kalibracije elementa koji služi za merenje vlažnosti vazduha [6].

VI. TABELA MERENJA

U tabeli merenja prikazani su rezultati prikupljeni sa terena. Tabela se sastoji od prikaza četiri lokacije: Svrljig, Suvi Do, Čegar i Crveni pevac (lokacija u Nišu).

Prikupljeni podaci o ambijentalnim parametrima su provereni sa stvarnim podacima korišćenjem rezultata drugih meteoroloških stanica, kao i podataka koji su bili dostupni na sajtu Republičkog Hidrometeorološkog Zavoda. Dok su prikupljeni podaci o lokaciji i nadmorskoj visini upoređeni sa rezultatima koje pruža *GoogleMaps*. Prilikom upoređivanja prikupljenih i stvarnih parametra, izvodi se zaključak da je sistem veoma precizan i tačan sa minimalnim odstupanjima, koja mogu nastati usled baždarenja i kalibracije instrumenata.

U Tabeli I. može se primetiti da se neki mereni parametri ponavljaju, kao što su temperatura i nadmorska visina. Temperaturu mogu meriti senzor *DHT11* i senzor *BMP280*, dok nadmorsku visinu mogu meriti GPS modul i *BMP280* senzor [7,8].

Merenja temperature i nadmorske visine ukazuju na značajnost upotrebe senzora za merenje one veličine za koju su primarno dizajnirani, kao i na upotrebu kvalitetnijih senzora prilikom merenja.

BMP280 senzor je senzor visokog kvaliteta i ima široku primenu u mobilnoj industriji. Samim tim, *BMP280* senzor pruža tačnija merenja temperature od *DHT11* senzora, kao i tačnije i preciznije merenje nadmorske visine od GPS modula.

TABELA I
TABELA SA MERENJIMA

Senzor	Veličina merenja	Svrljig	Suvi Do	Čegar	Crveni pevac
„**DHT11**”	Temperatura:		32.0°C	31.6°C	29.1°C
	Vlažnost vazduha:		41.00%	35.00%	44%
„**LDR**”	Približna osvetljenost u „Lux”:		187835.64	130163.12	26027.01
„**WaterSensor**”	Vlažnost zemljišta:		23%	14%	18%
„**BMP280**”	Temperatura:	24°C	32.58°C	30.16°C	30.54°C
	Pritisak:	97006.78 Pa	98969.80Pa	97715.47Pa	99569.15Pa
	Nadmorska visina:	365.89m	197.96m	304.95m	147.22m
„**GPS-NEO-6M**”	Status očitavanja:	A	A	A	A
	Geografska širina:	43° 24.74877' N	43.331623 °N	43.365097 °N	43.319393 °N
	Geografska dužina:	022° 06.89557' E	21.972717 °E	21.943077 °E	21.905567 °E
	Nadmorska visina:	416.4m	265.4m	364.0m	332m
	Mod rada:	Automatic	Automatic	Automatic	Automatic
	Tip fiksiranja:	3D fix	3D fix	3D fix	3D fix
	Broj satelita u vidokrugu:	6	13	12	11

VII. ZAKLJUČAK

Primena sistema ranog upozorenja predstavlja značajan korak ka ublažavanju uticaja erozije i bujičnih poplava. Pružajući pravovremene i tačne informacije, sistem može pomoći u spašavanju života, zaštiti infrastrukture i osiguravanju održivog razvoja zajednica u područjima podložnim opasnostima. Cilj rada je da inspiriše dalja istraživanja i razvoj u ovoj oblasti, što će dovesti do usvajanja efikasnih mera za upravljanje prirodnim opasnostima. Oblasti za poboljšanje, potencijalno dalje istraživanje i dalje unapređivanje meteorološkog sistema za monitoring, rano upozoravanje i prevenciju od bujica predloženog u ovom radu su upotreba raznovrsnijih i kvalitetnijih senzora. Na primer, senzor za merenje brzine vetra, senzor za merenje količine padavina, kvalitetniji senzor za merenje vlažnosti i temperature zemljišta. Takođe, velika oblast za istraživanje i poboljšanje ovog sistema je upotreba solarne energije ili upotreba energije vetra za rad sistema. Na taj način bi se ovaj sistem načinio nezavisnim od mreže za napajanje. Sledeća oblast poboljšanja sistema je korišćenje naprednije softverske tehnologije, kao što je čuvanje podataka na serverima, izrada adekvatne web platforme, na kojoj će svako moći da pristupi i da dobija informacije o trenutnom stanju na terenu, takođe i implementacija veštačke inteligencije (AI) i mašinskog učenja. Uključujući ove alate, sistem bi postao precizniji, efikasniji i brži u otkrivanju potencijalnih opasnosti i pružanju upozorenja u realnom vremenu. Pored toga, sistem bi mogao da se integriše sa drugim platformama, kao što su društveni mediji, kako bi se dosegla šira publika i povećala svest javnosti o potencijalnim rizicima. Uz korišćenje karata, tj. mapa, i implementacije CAP protokola (engl. *Common Alerting Protocol*) [9], sistem bi dobio pravu realnu i konkurentnu vrednost. Još jedno potencijalno poboljšanje moglo bi biti proširenje oblasti pokrivenosti meteorološkog sistema ranog upozoravanja. Dok se sistem ukratko predstavljen u radu fokusira na konkretan uređaj i opis samog problema, sam sistem bi se mogao prilagoditi i implementirati u drugim područjima sklonim eroziji i poplavama. Ovo bi zahtevalo dodatna ulaganja u infrastrukturu, kao što su senzori i uređaji za praćenje, kao i saradnju između lokalnih i nacionalnih vlasti kako bi se obezbedila efikasnost sistema, kao i ulaganje u dalji razvoj projekta i proširenje tima koji bi bio zadužen za razvoj, održavanje i nadgledanje sistema.

LITERATURA

[1] Tehnička dokumentacija za ESP32-WROOM-32D/U. Dostupno na: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

[2] A. Prijic, „Poluprovodničke komponente”, predavanja 2023. Dostupno na: https://mikro.elfak.ni.ac.rs/predmeti/poluprovodnicke-komponente/

[3] A. Prijic, D. Danković, Z. Prijić, „Uvod u poluprovodničke komponente i njihovu primenu”, drugo izmenjeno i dopunjeno elektronsko izdanje, 2020., ISBN 978-86-6125-225-9.

[4] Tehnička dokumentacija za ESP32. Dostupno na: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[5] Lj. Vračar, „Senzori i pretvarači”, predavanja, 2023. Dostupno na: https://mikro.elfak.ni.ac.rs/predmeti/senzori-i-pretvaraci/.

[6] Tehnička dokumentacija za DHT11. Dostupno na: https://www.alldatasheet.com/datasheet-pdf/pdf/1132088/ETC2/DHT11.html

[7] Tehnička dokumentacija za BMP280. Dostupno na: https://www.alldatasheet.com/datasheet-pdf/pdf/1132069/BOSCH/BMP280.html.

[8] Tehnička dokumentacija za NEO-6, u-blox 6 GPS Modules. Dostupno na: https://www.digikey.com/htmldatasheets/production/2757457/0/0/1/neo-6.html.

[9] E. Christian, “Introducing the Common Alerting Protocol (CAP)”, 2001.

Primena kola TCRT5000 za merenje udaljenosti

Nenad Radivojević, Sandra Stanković, Lana Tasić

Sadržaj - U ovom radu prikazan je princip rada komponente TCRT5000, kao i njene karakteristike, sa detaljnijim fokusom na njenu primenu za mogućnost merenja udaljenosti u određenom opsegu. Mogućnost merenja udaljenosti u određenom opsegu je eksperimentalno proverena i prikazana tabelarno i grafički.

I. UVOD

Opto senzore, kao i ostale optoelektronske komponente proučava grana elektronike koja se zove optoelektronika. Ova oblast proučava svetlost kao fizičku pojavu, komponente kojom ona može da se konvertuje u električni signal, meri ili detektuje, kao i optimizaciju samih tih komponenti. U opto senzore spadaju fotootpronik, fotodioda, fototranzistor, refliktivni i transimitivni opto senzori, kao i mnogi drugi. Većina opto senzora koristi fotodiodu ili fototranzistor (u zavisnosti od primene) kao glavnu komponentu. Jedna od tih komponenata je i predmet ovog rada, senzor TCRT5000.

II. PRINCIP RADA KOMPONENTE

Reflektivni opto senzori rade na principu refleksije svetlosti. Sastoje se od jedne LE (*Light Emitting*) diode i jedne komponente za detektovanje svetlosti, koja može biti fotodioda ili fototranzistor. Konkretno u slučaju opto senzora TCRT5000 su u pitanju fototranzistor i infracrvena LE dioda (slika 1).

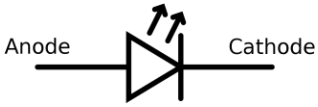


Slika 1. Opto senzor TCRT5000.

Između ove dve komponente nalazi se crna neprozirna prepreka koja onemogućava direktnu spregu između infracrvene diode i fototranzistora. Naime, svetlost koju emituje dioda treba da se odbije od neke prepreke i da na taj način omogući proticanje struje kroz fototranzistor. Detaljniji opis rada infracrvene diode i fototranzistora biće prikazan u nastavku.

N. Radivojević, S. Stanković, L. Tasić su studenti četvrte godine Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nenad.radivojevic@elfak.rs, s.sandra@elfak.rs, lana.tasic@elfak.rs.

Infracrvena dioda (*Infrared diode*) pripada grupi LE dioda i za razliku od standardnih LE dioda, infracrvena dioda emituje svetlost talasne dužine u infracrvenom spektru koji ljudsko oko ne može da detektuje. Standardni napon vođenja ove diode iznosi 1.3 V, što može malo da se razlikuje u zavisnosti od tehnologije izrade.



Slika 2. Simbol infracrvene diode.

Fototranzistor je u osnovi bipolarni tranzistor. Razlikuje se po tome što je mesto za izvod za bazu projektovano kao fotosenzitivna površina, pa struja baze nastaje kao fotostruja, usled upadne svetlosti. Tranzistor se polariše tako da se kolektor nalazi na višem potencijalu u odnosu na emitor, a struja će proteći kroz tranzistor kada na bazu deluje svetlosno zračenje i tada dolazi do fotogeneracije nosilaca. Struja će teći od kolektora ka bazi. U slučaju kola TCRT5000, fototranzistor je projektovan tako da bude osetljiv samo na talasne dužine u infracrvenom spektru, što znači da fototranzistor neće biti pobuđivan ukoliko na njega deluje spoljašnje osvetljenje.



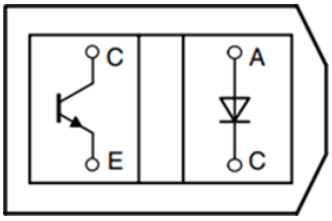
Slika 3. Simbol fototranzistora.

III. OSNOVNO KOLO KOMPONENTE I NJENO POVEZIVANJE

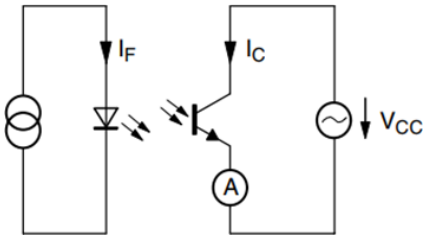
Simbol ovog senzora prikazan je na slici 4. Ovaj senzor ima četiri izvoda, što se vidi sa slike simbola. To su izvodi za kolektor i emitor fototranzistora i izvodi za anodu i katodu infracrvene diode. Na simbolu se takođe primećuje nepravilnost, odnosno zasečena strana, koja služi za određivanje pinova.

Osnovno kolo ove komponente prikazano je na slici 5. Na izgled deluje da infracrvena dioda pobuđuje fototranzistor direktno, međutim, komponenta sadrži crnu neprozirnu prepreku između infracrvene diode i fototranzistora, kao što je već napomenuto. Dakle, fototranzistor će biti pobuđen samo ukoliko se nalazi neka prepreka ispred komponente i to udaljena u određenom

opsegu. Ova komponenta može meriti udaljenost prepreke u opsegu od 0.2 mm do 15 mm.

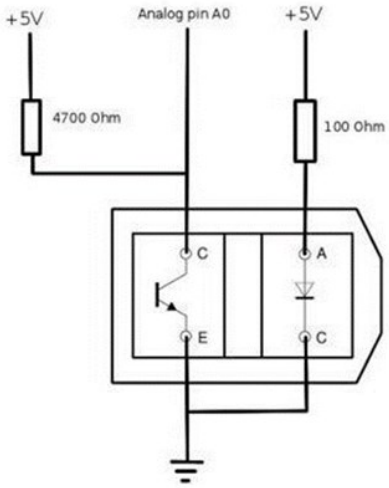


Slika 4. Simbol komponente TCRT5000.



Slika 5. Osnovno kolo komponente TCRT5000.

Senzori se najčešće povezuju na neki mikrokontroler čijim programiranjem je moguće manipulisati izlaznim veličinama koje nam senzor pokazuje. Senzor TCRT5000 je naponski senzor, odnosno izlazna veličina ovog senzora predstavlja napon. Dakle, merenjem napona je moguće odrediti rastojanje prepreke u odnosu na senzor pomoću zavisnosti koju ćemo dobiti eksperimentalno.



Slika 6. Povezivanje senzora sa mikrokontrolerom.

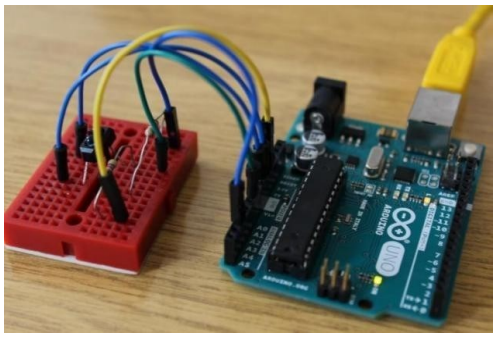
Na slici 6. prikazan je način povezivanja ovog senzora sa mikrokontrolerom. Kolektor fototranzistora povezan je preko otpornika na napon od 5 V na mikrokontroleru. Na isti način je povezana i anoda infracrvene diode. Emitor fototranzistora i katoda infracrvene diode povezani su na masu. Izlaz se čita sa kolektora fototranzistora i ovo je analogni pin.

IV. EKSPERIMENTALNO ISPITIVANJE KARAKTERISTIKA SENZORA

Tokom ispitivanja karakteristika senzora, odnosno tokom merenja vrednosti napona koju senzor daje na osnovu udaljenosti od objekta korišćene su sledeće komponente:

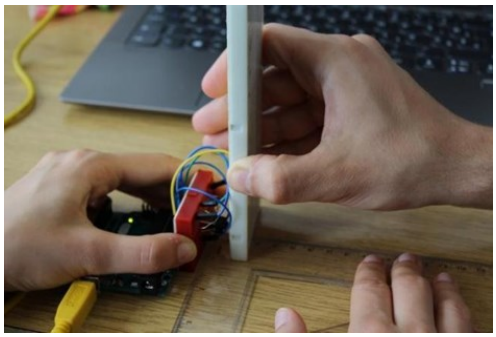
- Arduino Uno razvojno okruženje,
- Opto senzor TCRT5000,
- Neprozirna kutija crne boje,
- Proto ploča, koja je korišćena kao objekat,
- Osciloskop,
- Gumica za brisanje, koja je korišćena kao objekat tokom merenja ispod kutije,
- Otpornici različitih otpornosti (220 Ω, 270 Ω, 390 Ω).

Vrednosti koje su izmerene dobijene su putem odgovarajućeg koda, koji je ubačen u mikrokontroler razvojnog okruženja, koji u ovom slučaju ima funkciju voltmetra i u kome je vrednost napona koju senzor daje pretvorena u odgovarajuće nivoe uz pomoć analogno-digitalne konverzije.



Slika 7. Povezano okruženje.

Merenje je vršeno za različite udaljenosti objekta (proto ploče) od senzora i to od 1 mm do 7 mm razdaljine za svaku vrednost otpornika. Udaljenost objekta je tokom svakog merenja proveravana lenjirom.



Slika 8. Merenje.

Rezultati dobijeni prilikom ovih merenja su dobijeni u digitalnim vrednostima i prikazani u tabeli I.

TABELA I
REZULTATI MERENJA U DIGITALNIM VREDNOSTIMA NAPONA

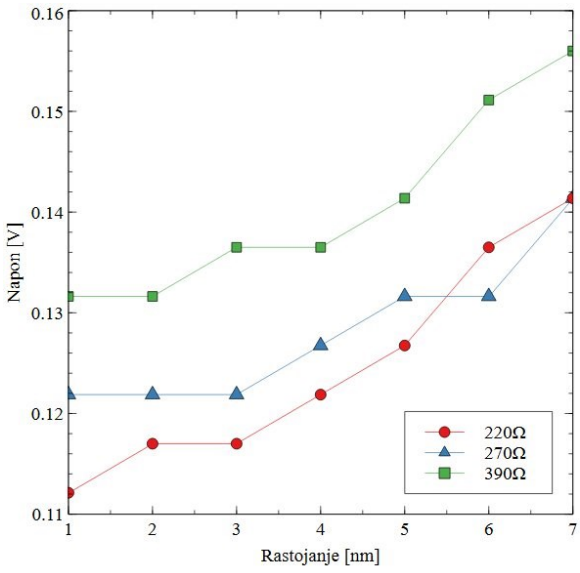
Rastojanje [nm] /Otpornost	220 Ω	270 Ω	390 Ω
1	23	25	27
2	24	25	27
3	24	25	28
4	25	26	28
5	26	27	29
6	28	27	31
7	29	29	32

Nakon toga su rezultati iz tabele preračunati radi konvertovanja digitalnog signala u analogni zbog povećanja preciznosti i ti rezultati su upisani u tabelu II.

TABELA II
REZULTATI MERENJA U ANALOGNIM VREDNOSTIMA NAPONA

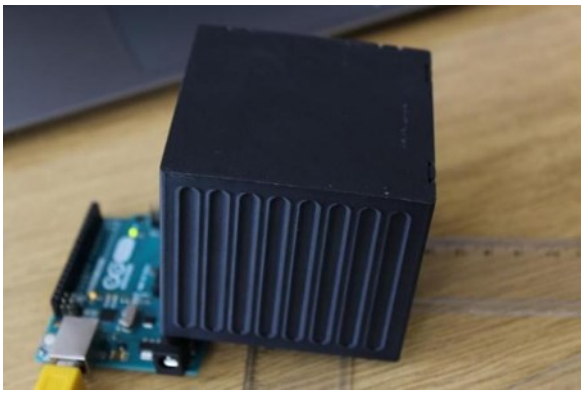
Rastojanje [nm] /Otpornost	220 Ω	270 Ω	390 Ω
1	0.112	0.122	0.132
2	0.117	0.122	0.132
3	0.117	0.122	0.137
4	0.122	0.127	0.137
5	0.127	0.132	0.141
6	0.136	0.132	0.151
7	0.141	0.141	0.156

Zatim su vrednosti iz tabele broj II prenesene na grafik koji prikazuje različite zavisnosti izlaznog napona od rastojanja i to za različite vrednosti otpornika. Napon je prikazan na x osi i izražen u jedinici volt - [V], a rastojanje je prikazano na y osi i izraženo je u nanometrima - [nm].



Slika 9. Zavisnost izlaznog napona od rastojanja. Grafik je nacrtan pomoću softverskog alata Veusz.

Rezultatima iz tabela I i II, kao i analizom grafika, dolazi se do zaključka da se sa povećanjem otpornosti povećava i izlazni napon senzora, odnosno grafik se pomera na gore po y-osi za veće vrednosti otpornosti.



Slika 10. Merenje sa neprozračnom kutijom

Nakon izvršenih merenja pristupljeno je merenjima, ali unutar neprozirne kutije i pokazano je da su rezultati identični rezultatima koji su dobijeni pri merenjima bez neprozirne kutije. Na taj način je pokazano da na opto senzor TCRT5000 ne utiče spoljašna svetlost, kao što je navedeno u tehničkoj dokumentaciji.

Na kraju je bilo potrebno proveriti dobijene rezultate izlaznog napona pomoću osciloskopa, kako bi utvrdili ispravnost koda. To je odrađeno za granične slučajeve rastojanja korišćenjem otpornosti od 220 Ω. Dobijeni rezultati na osciloskopu se približno poklapaju sa rezultatima dobijenim pomoću mikrokontrolera, sa malim odstupanjem zbog mogućih grešaka pri merenju. Izlaz osciloskopa je prikazan na slikama 11a i 11b. Na slici 11a prikazan je izlazni napon bez prepreke, dok je na slici 11b prikazan izlazni napon sa preprekom udaljenom 1 mm. Izlazni napon za prepreku udaljenu 7 mm gotovo je identičan, zbog male razlike u naponima (a) bez prepreke, b) sa preprekom udaljenom 7 mm).



Slika 11. Izlazni napon prikazan na osciloskopu.

V. ZAKLJUČAK

Opto senzor TCRT5000 može da se koristi za merenje malih rastojanja, međutim ukoliko bi želeli da izmerimo neka veća rastojanja, to ne bi bilo moguće. Postoje drugi senzori, kao što su ultrazvučni pretvarači ili LIDAR (Light Detection and Ranging) koji preciznije mere rastojanja od objekta. Za korišćenje opto senzora TCRT5000 za merenje udaljenosti objekta neophodno je poznavanje rada ovog kola, kao i komponenti od kojih se sastoji.

ZAHVALNICA

Zahvaljujemo se profesorki dr Emiliji Živanović i predmetnom asistentu master inž. Nikoli Mitroviću na pruženoj pomoći i podršci i izdvojenom vremenu prilikom pisanja ovog rada. Nadamo se budućim saradnjama.

LITERATURA

[1] TCRT5000 Datasheet, Vishey.
[2] Z. Prijic, A. Prijić, D. Danković, „Uvod u poluprovodničke komponente i njihovu primenu" Elektronski fakultet, Univerzitet u Nišu, 2020.
[3] E. Živanović, „Praktikum laboratorijskih vežbi sa primerima zadataka iz predmeta Osnovi optike", Elektronski fakultet, Univerzitet u Nišu, 2021.
[4] Lj. Vračar, „Senzori i pretvarači" slajdovi sa predavanja, Elektronski fakultet, Univerzitet u Nišu, 2023.
[5] The Arduino Projects Book.

TCAD projektovanje i simulacija bipolarnog tranzistora

Sara Savić

Sadržaj - U ovom radu analiziraju se električne karakteristike bipolarnog tranzistora korišćenjem TCAD alata za projektovanje i simulaciju u zavisnosti od različitih parametara. Ispituje se uticaj strukture na električne karakteristike, kao i uticaj spoljašnjih faktora, poput temperature.

I. Uvod

Bipolarni tranzistor je poluprovodnička komponenta sa tri izvoda (emitor, baza i kolektor). Bipolarni tranzistor je strujno-kontrolisana komponenta, što znači da se njen rad zasniva na ulaznoj (baznoj) struji. Reč bipolarni u nazivu ove komponente upućuje na to da u provođenju struje učestvuju i elektroni i šupljine, dakle obe vrste nosilaca naelektrisanja. Bipolarni tranzistor, u zavisnosti od režima rada i pojačanja, na izlazu daje kolektorsku (izlaznu) struju. Bipolarni transistor čine dva p-n spoja, od kojih je jedan kolektor-bazni (kolektorski spoj), a drugi emitor-bazni (emitorski spoj). Na osnovu tehnologije izrade postoje dva tipa bipolarnog tranzistora i to su: npn i pnp. Emitorska struja je uvek određena kao zbir kolektorske i bazne struje, dok odnos kolektorske i bazne struje predstavlja strujno pojačanje tranzistora. Osnovna primena bipolarnog tranzistora jeste kao prekidač i kao pojačavač [1,2]. Ove godine obeležava se 75 godina bipolarnog tranzistora [3].

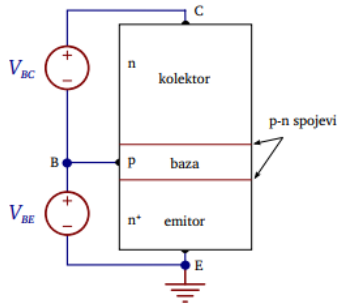
Ulazno (bazno) kolo određuje vrednost bazne struje, tako da se za svaku konkretnu vrednost bazne struje može dobiti kriva (strujno-naponska karakteristika) koja prikazuje zavisnost izlazne (kolektorske) struje od napona između kolektora i emitora. Iscrtavanjem zavisnosti $I_C=f(V_{CE})$ za različite vrednosti I_B dobija se skup (familija) izlaznih strujno-naponskih karakteristika bipolarnog tranzistora. Direktna polarizacija p-n spoja baza-emitor za posledicu ima eksponencijalnu zavisnost kako bazne tako i kolektorske struje od napona između baze i emitora. Zavisnosti $\log(I_C)=f(V_{BE})$ i $\log(I_B)=f(V_{BE})$ čine takozvanu Gummel-ovu karakteristiku. Gummel-ova karakteristika je kombinovana karakteristika struje baze i struje kolektora (u logaritamskoj razmeri) u funkciji napona između baze i emitora [4]. Odnos ovih karakteristika daje strujno pojačanje tranzistora, tako da je ono veće što je veći razmak između karakteristika.

Formiranju strukture predstoji ispisivanje tehnološkog niza bipolarnog tranzistora u Athena-i. Za ovu svrhu, postoje predefinisane naredbe koje oponašaju tehnološke procese koji se koriste u proizvodnji poluprovodničkih komponenata. Svaka naredba sadrži spisak obaveznih i neobaveznih parametara potrebnih za dosledno predstavljanje konačnog ishoda procesa. Spisak naredbi sa njihovim objašnjenjem, parametrima i primerima moguće je pronaći u zvaničnom priručniku za korišćenje Athena-e u okviru alata Silvaco, dato u [5]. Za simulaciju električnih karakteristika u okviru Silvaco alata koristi se simulator pod nazivom Atlas. Atlas je softver za simulaciju poluprovodničkih komponenti u jednoj, dve i tri dimenzije. Koristi predefinisane modele za izračunavanje složenih matematičkih jednačina [6]. Struktura formirana u Athena-i simulira se u Atlas-u.

S. Savić, student master akademskih studija na studijskom programu Elektronika i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sarassavic@elfak.rs.

II. Tehnološka realizacija bipolarnog tranzistora kao diskretne komponente

Strukturu bipolarnog tranzistora čine dva p-n spoja postavljena „leđa o leđa“. Bipolarni tranzistor se normalno polariše tako da mu je spoj baza-emitor polarisan direktno, a spoj baza-kolektor inverzno, kako bi se omogućio protok struje od kolektora ka emitoru, pri čemu se intenzitet protoka kontroliše preko baze. Kada govorimo o npn bipolarnom tranzistoru, emitor je jako dopirana oblast n-tipa, baza je oblast p-tipa, dok je kolektor umereno do slabo dopirana oblast n-tipa. Struktura npn bipolarnog tranzistora ilustrovana je na slici 1, preuzetoj iz [2].

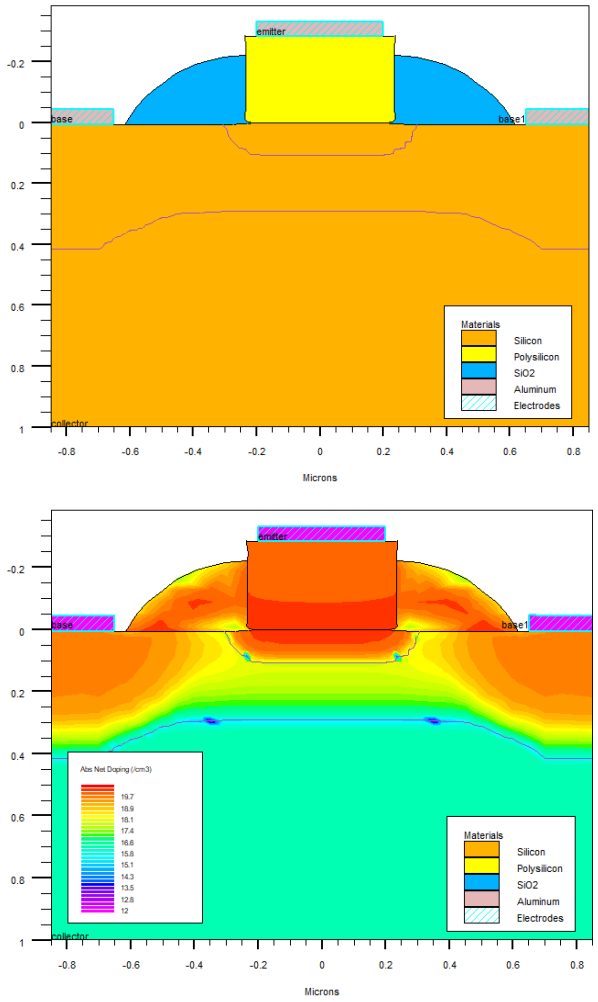


Slika 1. Struktura i polarizacija npn bipolarnog tranzistora.

Tehnološki niz za realizaciju bipolarnog tranzistora kao diskretne komponente čine složeni tehnološki procesi poput oksidacije, difuzije, jonske implantacije, selektivnog nagrizanja i mnogih drugih procesa. Svaki od ovih procesa zahteva definisanje karakterističnih parametara. Na primer, kada se simulira difuzija potrebno je u okviru parametara za ovaj proces navesti vreme trajanja difuzije, temperaturu na kojoj se ona odvija, kao i ambijent, a ukoliko neki od parametara nije definisan onda on ima predefinisanu uobičajenu vrednost. Takođe, koncentracija i vrsta

donorskih primesa mogu uticati na električne karakteristike komponente, pa je tako potrebno voditi računa o tome kako primese utiču na krajnje ponašanje komponente.

Struktura bipolarnog tranzistora (slika 2) dobijena je tehnološkim nizom u Athena-i u okviru Silvaco TCAD alata. Početna struktura zasniva se na primeru iz baze podataka programa Silvaco [7]. Na osnovu datog primera, izmenom određenih parametara, pokazano je kako izmene u strukturi mogu uticati na poluprovodničku komponentu i njene karakteristike, konkretno pokazano na primeru bipolarnog tranzistora.



Slika 2. Kompletna struktura formiranog bipolarnog tranzistora u TCAD alatu.

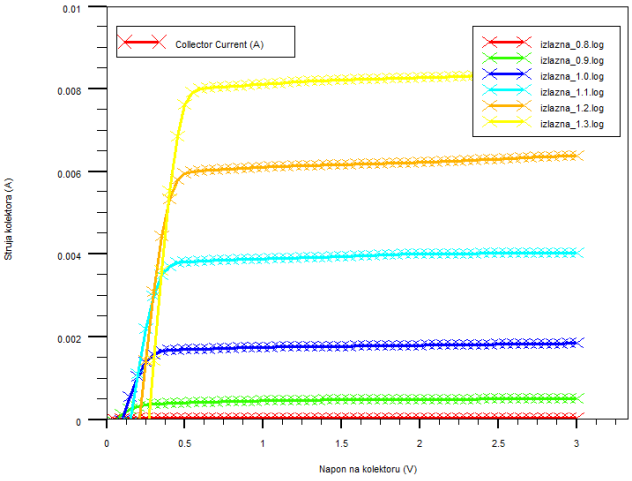
Na slici 2 prikazana je raspodela koncentracija primesa u strukturi. Jasno se uočavaju formirani p-n spojevi: emitor-baza i baza-kolektor. Emitor je od polisilicijuma i to je najjače dopirana oblast. Dva kontakta za bazu koriste se kako bi se struja ravnomerno raspodelila kroz strukturu. Po pravilu, metal za kontakte ne bi trebalo nanositi direkno preko p-n spoja, kako ne bi došlo do kratkospajanja p i n oblasti. Ovaj problem izbegnut je postojanjem oksida (spacer-a).

III. REZULTATI SIMULACIJA I ANALIZA

Za simulaciju električnih karakteristika u Atlas-u najpre se definišu kontakti na koje se dovodi napon, zatim se definiše model simulacije, onda matematički metod izračunavanja jednačina i određuje se početno rešenje simulacije. Nakon toga zadaju se parametri struja ili napona koji se menjaju, a izračunavaju se vrednosti struja/napona koji zavise od tih promena. Rezultati simulacije upisuju se u fajl koji se otvara pre same naredbe za rešavanje numeričkih jednačina. Na taj način dobija se karakteristika prikazana u TonyPlot-u.

A. Izlazne karakteristike

Na slici 3 prikazan je skup izlaznih karakteristika bipolarnog tranzistora za sledeće vrednosti napona na bazi: 0.8 V, 0.9 V, 1 V, 1.1 V, 1.2 V i 1.3 V, pri čemu se napon na kolektoru menja od 0 V do 3 V, sa korakom od 0.05 V. Na datom skupu izlaznih karakteristika mogu se uočiti i oblasti rada tranzistora. S obzirom da je napon vođenja ovog tranzistora između 0.8 V i 0.9 V, može se videti da za napon na bazi od 0.8 V tranzistor se nalazi u zakočenju, a struja kolektora je približna nuli, dok za vrednosti napona na bazi iznad 1 V struja kolektora postaje značajna i povećava se sa porastom napona na bazi.



Slika 3. Simulacija izlaznih karakteristika npn bipolarnog tranzistora.

B. Gummel-ova karakteristika

Gummel-ova karakteristika podrazumeva zavisnost struje baze i struje kolektora od napona na spoju baza-emitor, pri konstantnoj vrednosti napona između kolektora i emitora. Prema tome, za simulaciju karakteristike, naredbom solve, napon kolektor-emitor postavlja se najpre na fiksnu vrednost (V_{CE}=5 V). Zatim se otvara fajl u kome će se upisivati rezultati simulacije. Nakon toga, za svaku vrednost napona na bazi od 0.1 V do 1 V sa korakom od 0.05 V računaju se vrednosti struja i upisuju u otvoreni fajl.

Na slici 4 prikazan je kod u Atlas-u (u programu DeckBuild) za simulaciju Gummel-ove karakteristike bipolarnog tranzistora. Dobijena karakteristika prikazana je na slici 5.

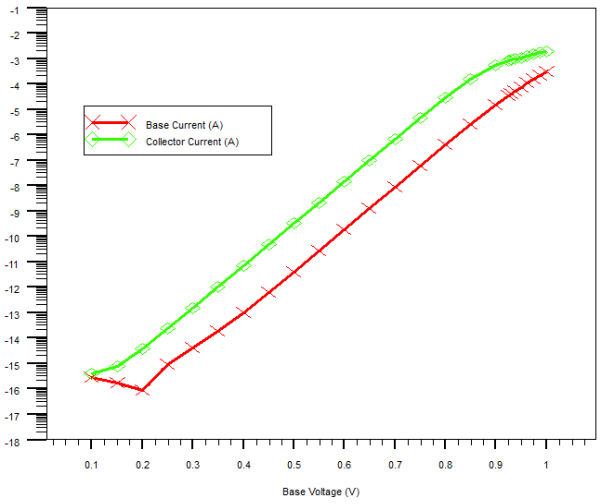
```
go atlas

material taun0=5e-6 taup0=5e-6
contact name=emitter n.poly surf.rec
contact name=base1 common=base short

models bipolar print
method newton autonr trap
solve init
solve vcollector=0 vstep=0.5 vfinal=5 name=collector
log outf=gummel.log master
solve vbase=0.1 vstep=0.05 vfinal=1.0 name=base

tonyplot gummel.log -set gummel.set
```

Slika 4. Kod u Atlas-u za simulaciju Gummel-ove karakteristike.

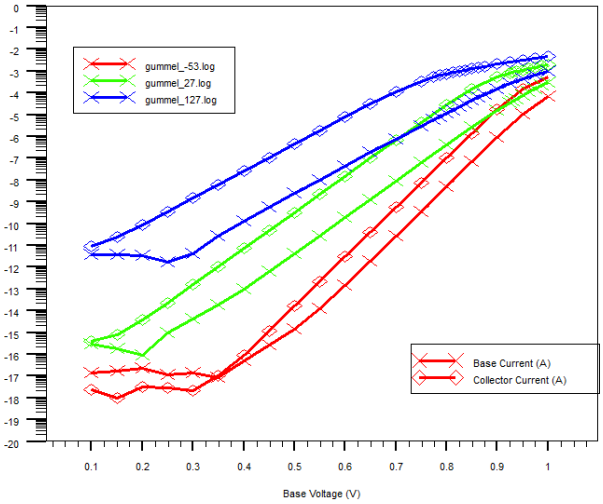


Slika 5. Simulacija Gummel-ove karakteristike.

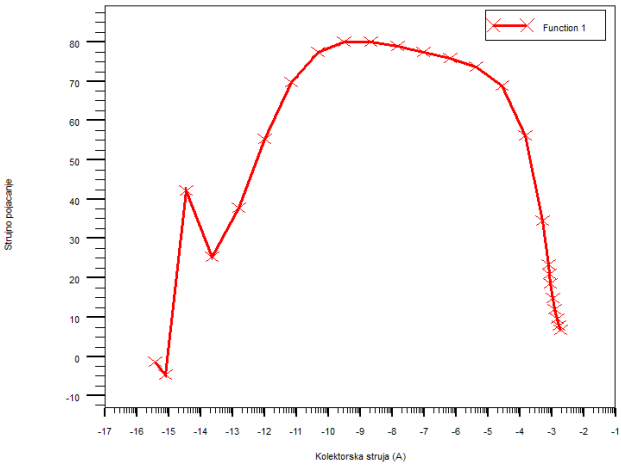
Ponavljanjem simulacije Gummel-ove karakteristike pri temperaturi od 400K (127°C) i pri temperaturi od 220K (-53°C) dobija se zavisnost ove karakteristike od temperature. Na slici 6 prikazani su dobijeni rezultati preklapanjem karakteristika za tri karakteristične temperature. Analizom dobijenih rezultata potvrđuje se teorijska zavisnost strujnog pojačanja od temperature. S obzirom da je strujno pojačanje određeno rastojanjem između karakteristika bazne i kolektorske struje, uočava se da je to rastojanje veće sa povećanjem temperature.

C. Strujno pojačanje

Karakteristika zavisnosti strujnog pojačanja od napona na kolektoru $\beta=f(\log(I_C))$ izvodi se iz Gummel-ove karakteristike. Strujno pojačanje dobija se kao odnos struje kolektora i struje baze. Na slici 7 prikazana je zavisnost strujnog pojačanja simuliranog bipolarnog tranzistora u funkciji kolektorske struje.



Slika 6. Simulacija Gummel-ove karakteristike za različite vrednosti temperature.

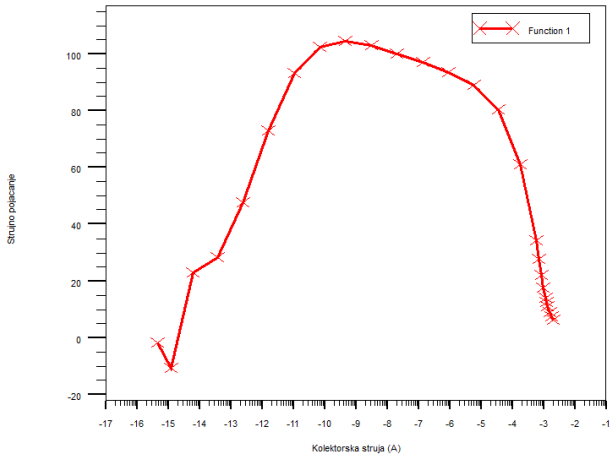


Slika 7. Simulacija strujnog pojačanja bipolarnog tranzistora.

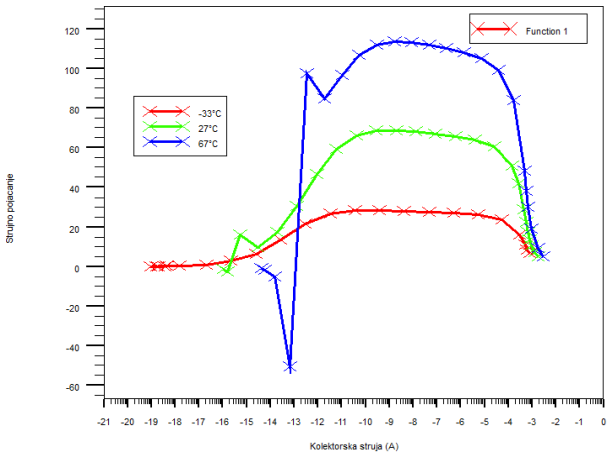
S obzirom da se prvobitno dobija karakteristika na kojoj je prisutan pik (slika 7), pristupa se izmenama u tehnološkom nizu radi eliminisanja pika i ujedno povećanja strujnog pojačanja tranzistora. U te svrhe, odrađena je simulacija sa izmenama u dozi implantacije pri formiranju baze. Međutim, manipulacija širine oblasti baze nije doprinela poboljšanju karakteristika. Zbog toga, sledeći korak je povećanje dopiranja emitora, što za posledicu ima smanjenje oblasti baze. Kao rezultat dobija se karakteristika strujnog pojačanja bez pika, a strujno pojačanje je ujedno i povećano (slika 8).

Strujno pojačanje direktno zavisi od temperature. Na slici 9 prikazane su zavisnosti strujnog pojačanja od kolektorske struje za tri različite vrednosti temperature. Simulacija je odrađena sa početnim parametrima dopiranja kolektora, baze i emitora. Napon između kolekotra i emitora je 1 V, a napon između baze i emitora se menja od

0 V do 1 V. Prikazane su istovremeno karakteristike pri sobnoj temperaturi od 300 K (27 °C), pri većoj temperaturi od 340 K (67 °C) kao i pri nižoj temperaturi koja iznosi 240 K (-33 °C). Dobijeni rezultati potvrđuju teorijsku zavisnost, dakle sa porastom temperature raste pojačanje.



Slika 8. Simulacija eliminacije pika strujnog pojačanja.

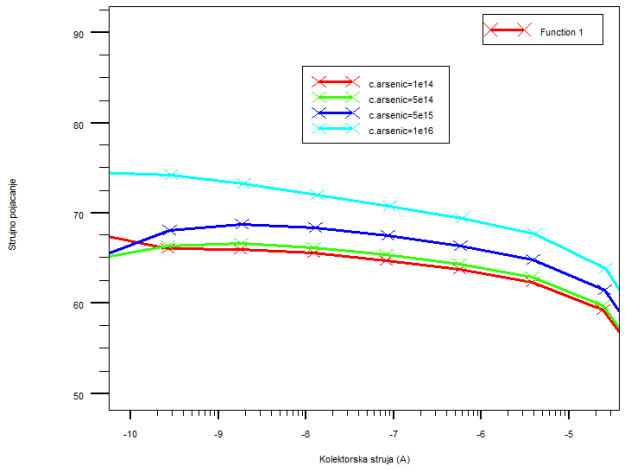


Slika 9. Simulacija strujnog pojačanja za različite vrednosti temperature.

Promena koncentracije primesa u supstratu takođe utiče na strujno pojačanje tranzistora. Promenom koncentracije arsena u naredbi za inicijalizaciju supstrata i preklapanjem dobijenih karaktristika strujnog pojačanja za svaku od vrednosti dobija se zavisnost prikazana na slici 10. Zaključuje se da sa porastom nivoa dopiranja supstrata raste i strujno projačanje, pri čemu taj porast nije drastičan, ali ipak u manjoj meri postoji. Važno je naglasiti da oblast kolektora treba, u svakom slučaju, da ostane najniže dopirana oblast kako bi bipolarni tranzistor efikasno obavljao svoju funkciju.

IV. Zaključak

Bipolarni tranzistor je elektronska komponenta koja ima široku primenu u svetu elektronike. Zbog toga je od važnosti razumevanje principa rada i optimizovano projektovanje ove komponente. Radi efikasnije proizvodnje bipolarnog tranzistora, neophodno je uzeti u obzir uticaj različitih faktora na električne karakteristike gotove komponente.



Slika 10. Simulacija uticaja nivoa dopiranja supstrata na strujno pojačanje bipolarnog tranzistora.

Prikazane su izlazne karakteristike za različite bazne struje i zavisnosti kolektorske i bazne struje od napona na bazi. Parametri tehnoloških procesa treba da budu podešeni tako da oblast emitora bude najjače dopirana, a oblast kolektora najslabije dopirana. Bipolarni tranzistor treba da ima strujno pojačanje što manje zavisno od struje kolektora i što manju struju curenja. Da bi se to postiglo oblast baze treba da bude što uža, što se može postići podešavanjem tehnoloških parametara. Treba istaći da sa porastom temperature, pojačanje bipolarnog tranzistora raste.

Zahvalnica

Rad je realizovan po ugledu na simulacije i rezultate dobijene prilikom izrade mog diplomskog rada. Posebno se zahvaljujem mentoru, prof. dr Vesni Paunović, na pruženom znanju i sugestijama tokom pisanja rada. Takođe, veliko hvala Milošu Marjanoviću na izdvojenom vremenu, saradnji i pomoći.

Literatura

[1] A. Prijić, Poluprovodničke komponente, slajdovi sa predavanja na temu: Bipolarni tranzistor. Elektronski fakultet, Univerzitet u Nišu, 2020.

[2] A. Prijić, D. Danković, i Z. Prijić, *Uvod u poluprovodničke komponente i njihovu primenu* (drugo, izmenjeno i dopunjeno elektronsko izdanje). Elektronski fakultet u Nišu, 2020.

[3] 75 godina bipolarnog tranzistora. Dostupno na: https://eds.ieee.org/about-eds/75th-anniversary-of-the-transistor#:~:text=The%20Electron%20Devices%20Society%20has,75th%20Anniversary%20of%20the%20Transistor.

[4] M. Marjanović, Modeliranje i simulacija mikroelektronskih komponenata i kola, vežbe na računaru. Elektronski fakultet, Univerzitet u Nišu, 2019.
[5] SILVACO International, ATHENA User's Manual, 2007.
[6] SILVACO International, ATLAS User's Manual, device simulation software, 2004.
[7] Silvaco primer simulacije bipolarnog tranzistora. Dostupno na: https://silvaco.com/examples/tcad/CMOS_and_BiCMOS/Bipolar_and_Diode/Bip_Dio_ex03/

Projektovanje i realizacija prenosivog ECG uređaja baziranog na STM32 platformi

Marina Miladinović

Sadržaj – Ovaj rad proučava osnove na kojima se zasniva rad ECG-a. Određuje osnosvni koncept metodologije potrebne za realizaciju. Predstavlja upoznavanje sa senzorom i mikrokontrolerom korišćenim pri izradi. Prikazuje idejno rešenje i potrebne komponente.

I. Uvod

Danas živimo u vremenu koje nam donosi neizbežan stres, bilo da je u pitanju tempo modernog života, zahtevi posla ili konstantna povezanost putem tehnologija. Pored našeg mentalnog zdravlja stres najviše utiče na naše srce. Redovni pregledi i određeni testovi mogu sprečiti komplikacije. Jedan od najpoudanijih testova za otkrivanje srčanih bolesti, rutinske preglede i za dijagnostiku različitih kardiovaskularnih stanja je ECG test koji se izvodi na ECG uređaju. Elektrokardiogram (Electrocardiogram - ECG) je grafički zapis električne aktivnosti srca. Elektrokardiografija je brza, bezbolna i neinvazivna metoda kojom se registruje električna aktivnost srca [1].

II. Princip rada ECG-a

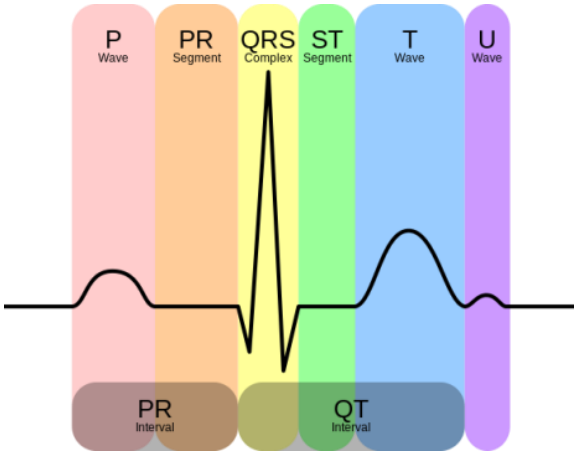
Princip rada elektrokardiografa zasniva se na fizičkom zakonu. Srce se ponaša kao električni dipol zato što ima delove sa pozitivnim i negativnim naelektrisanjem. Kada se srce stegne (usko kontrahuje i opušta), stvara se električni signal. Električni potencijal je ključna fizička veličina koja se meri tokom ECG-a. On predstavlja razliku u električnom naponu između različitih tačaka na telu. Elektrode postavljene na površini kože registruju ove električne potencijale.

ECG uređaj koristi elektrode postavljene na određenim tačkama na telu pacijenta. U želji da se što bolje sagleda električna aktivnost srca, uvedeni su unipolarni odvodi. Unipolarnih ekstremitetnih odvoda ima tri, a snimaju ih eksplorantne elektrode vezane za desnu ruku (aVR), levu ruku (aVL) i levu nogu (aVF) [2].

Na rezultujućem ECG grafikonu koji se može videti na slici 1, amplitude signala (visina talasa) i vreme između različitih tačaka na grafikonu (intervali) igraju ključnu ulogu u dijagnostici srčanih problema. Promene u ovim vrednostima mogu ukazivati na različite srčane probleme [3].

M. Miladinović je student Elektronskog fakulteta, Aleksandra Medvedeva 14, 18000 Niš, Srbija, e-mail: personagrata@elfak.rs

Fizičari koriste koncept vektora da bi opisali putanju električnih impulsa u srcu. Ovo se obično prikazuje kao vektorski dijagram, koji ilustruje kretanje električnih impulsa i kako to utiče na ECG talase.



Slika 1. Normalni sinusni ritam srca predstavljen na grafikonu.

III. Mikrokontroler STM32F750n8

Pri izboru mikrokontrolera za ovaj rad akcenat je bio na brzini obrade, složenosti urađenih zadataka, složenosti perifernih kola, troškova proizvodnje i velike potrošnje energije. STM32F750n8 uređaji su zasnovani na ARM Cortex-M7 visokih performansi gde 32-bitno RISC jezgro radi na frekvenciji do 216 MHz [4]. STM32F750n8 uređaji uključuju ugrađene memorije velike brzine. Svi uređaji nude tri 12-bitna ADC-a, dva DAC-a, RTC male snage, trinaest 16-bitnih tajmera opšte namene uključujući dva PWM tajmera za kontrolu motora i jedan tajmer male snage dostupan u stop režimu, dva 32-bitna tajmera opšte namene, pravi generator slučajnih brojeva (RNG) i ćeliju za kriptografsko ubrzanje [4].

Takt je osnovni signal koji određuje brzinu rada mikrokontrolera. Izražava se u hercima, a brži taktovi znače brži rad procesora. Poretanje sistema se odnosi na upotrebu taktnog signala za sinhronizaciju svih operacija unutar MCU-a. Pokretanje sistema često definiše centralni oscilator, PLL kolo, ili neki drugi izvor takta [4].

Oscilator je izvor takta za sistem. Kada je potrebno da se ostvari komunikacija MCU-a sa drugim spoljašnjim čipom neophodno je da posedujemo unutrašnji blok koji

podržava određene komunikacione protokole. Unutrašnji blok je poznat kao digitalni *Crossbar*. *Crossbar* pruža mogućnost programeru da u procesu inicijalizacije ostvari kontrolu nad konfiguracijom pinova koji su u početku pinovi opšte namene ili GPIO pinovi [4].

Na određene pinove MCU-a koji su predefinisani za tu namenu ili odabranim putem *Crossbar*-a kao analogni ulazi mogu da se dovedu analogni signali određenog naponskog nivoa. Ovakav signal se konvertuje u digitalnu reč određenog broja bitova pomoću analogno-digitalnog konvertora koji je ugrađen u samom mikrokontroleru [4].

Prekidni sistem omogućava različitim izvorima prekida da ukoliko su dozvoljeni lokalno i globalno, mogu da prekinu tekuće izvršenje programa [4].

Serijska komunikacija označava proces slanja podataka jedan po jedan bit, sekvencijalno, preko magistrale ili komunikacionog kanala [4].

IV. ECG CLICK

ECG Click je kompaktna ploča koja prihvata električne signale kao posledicu aktivnosti srca koje kuca putem elektroda zalepljenih za kožu. Ovo je kolo za prilagođenje biometrijskog tipa sa analognim interfejsom [8]. Projektovan je u sedam blokova (kolo za zaštitu, kolo pretpojačavača, dva visokopropusna filtera, kolo pojačavača, niskopropusni filter i DRL kolo) [5]. ECG Click ima jedan kratkospojnik i potenciometar za podešavanje izlaznog napona da odgovara ulaznom naponu nivoa ADC-a.

ECG Click se može podeliti u 7 blokova:

- Kolo za zaštitu - pruža ESD zaštitu (štiti pločicu), prenaponsku i prekostrujnu zaštitu. Pored zaštite, ulazni blok filtrira i sprečava da radio talasi „uđu“ u pretpojačavač [6].
- Pretpojačavač – implementira se kroz tri operaciona pojačavača konfigurisana kao instrumentacioni pojačavač (IA – pojačava razliku napona između „+“ i „-“ elektrode) koja na svom izlazu obezbeđuje “single-end” 1 signal [6].
- Visokopropusni filter (High Pass Filter) – treba da eliminiše DC komponentu signala. To je pasivni RC filter (prvog reda) [6].
- Pojačavač – pruža dodatno pojačanje koje se može podesiti pomoću potenciometra VP1. Ovo je da bi se analogni izlaz mogao prilagoditi opsegu ulaznog napona ADCa. Pojačavač je implementiran korišćenjem operacionog pojačavača u neinvertujućoj konfiguraciji [6].
- Visokopropusni filter (High Pass Filter) – treba da eliminiše DC komponentu signala ovog puta posle pojačavača. To je pasivni RC filter (prvog reda) [6].
- Niskopropusni filter (Low Pass Filter) – trebalo bi da ograniči opseg frekvencije na 60Hz. To je aktivni filter trećeg reda [6].
- DRL kolo (Driven Right Leg) – ovaj tip kola se često dodaje pojačavačima signala da bi se smanjio “common mode” interferencije. Pojačavač za bio signale, kao što je ECG, mere veoma male električne signale koje emituje telo (mereno u mV ili čak μV). Nažalost, ljudsko telo takođe može delovati kao antena koja hvata elektromagnetne smetnje (naročito šum od 50/60 Hz iz električnih vodova). Ova smetnja može prikriti biološke signale, što ih čini teškim za merenje. DRL kolo koristi se da eliminiše šum smetnji tako što ga aktivno poništava. To je stepen selektivnog pojačala koji pomera fazu signala za 180° (invertovanje) i vraća ga ispitanicima radi otkazivanja [6].

V. SOFTVERSKI ALATI KORIŠĆENI PRI IZRADI RADA

Hardver i softver su dve neophodne komponente sistema i rade zajedno kako bi omogućili funkcionalnost računara i drugih elektronskih uređaja. Bez softvera, hardver bi bio beskoristan, a bez hardvera softver ne bi imao platformu na kojoj bi se izvršavao. Kako bi dobili ono što je i cilj ovog rada potrebno je instalirati i podesiti sledeće programe: STM32Cube MX, STM32Cube IDE, STM32Cube drajveri i MikroPlot.

A. STM32Cube MX

Koristimo za inicijalizaciju pinova i izradu osnove programa koji ćemo menjati shodno potrebama. Dok je dizajnerima ploča glavna briga da optimizuju konfiguraciju pinova mikrokontrolera za njihov izgled ploče i da ispune zahteve aplikacije (izbor režima rada perifernih uređaja), programeri “embedded” sistema su više zainteresovani za razvoj novih aplikacija za određeni ciljni uređaj i migraciju postojećih dizajna na različite mikrokontrolere. STM32Cube MX je razvijen u okviru STM32Cube inicijative čija je svrha da ispuni ključne zahteve kupaca kako bi se maksimizirala ponovna upotreba softvera i minimiziralo vreme za kreiranje ciljnog sistema [7].

B. STM32Cube IDE

STM32Cube IDE je napredna C/C++ razvojna platforma sa perifernom konfiguracijom, generisanjem koda, kompilacijom koda i funkcijama za otklanjanje grešaka za STM32 mikrokontrolere i mikroprocesore. STM32Cube IDE generator koda inicijalizuje samo potrebne periferne taktove, što zauzvrat štedi energiju.Preuzima informacije o čipu sa ST servera i prikazuje karakteristike čipa, blok dijagram, dokumente i resurse i listu sa podacima na jednom mestu.

C. STM32Cube drajveri

Flešovanje programa na STM32 radi izvršavanja istog neophodno je instalirati dodatne alate poput drajvera bez kojih ne bi smo mogli koristiti procesor mikrokontrolera.

D. MikroPlot

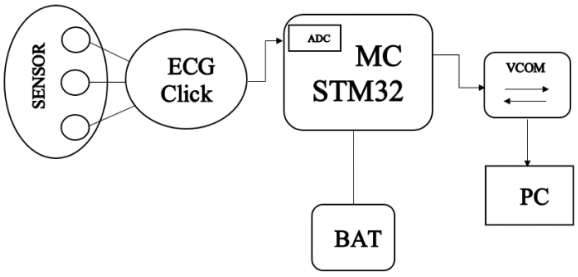
Aplikaciju koristimo da podatke upisane u mikrokontroler od strane ECG Click pločice pročitamo i pretvorimo u ECG dijagram prikazan u nastavku.

VI. REALIZACIJA ECG-A NA STM32 PLATFORMI

U ovom radu dizajniran je prenosivi ECG uređaj, koji kombinuje ”embedded„ tehnologija obrade digitalnih signala i tehnologiju akvizicije signala u jednom. Realizacija je podeljena na softversku i hrdversku realizaciju.

A. Hardverska realizacija

Ceo sistem se sastoji od šest modula prikazanih na blok dijagramu.

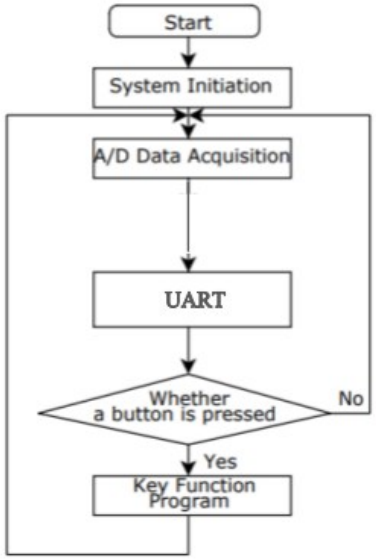


Slika 2. Blok dijagram hardverske realizacije.

Signal pulsa osobe čiji se ECG meri prikuplja Sensor, a zatim signal otkucaja srca se prenosi u ECG akviziciono kolo ECG Click. Ovo kolo zatim izvršava niz operacija kao što su pred pojačanje, glavno pojačanje, visoko i nisko propusno filtriranje... Nakon niza ovih operacija dobiće se usklađen ECG signal. Sistem zatim šalje ECG signal na STM32 AD konvertor za AD konverziju koja je potrebna. Podaci iz mikrokontrolera se preko VCOM-a (Virtual COM Port) šalju na PC i dalje obrađuju. Napajanje kola je preko jedne eksterne baterije (BAT). Pri dizajnu kola za napajanje uglavnom se uzima u obzir koji tip uređaja za napajanje treba koristiti, ulazni i izlazni napon, izlaznu struju i kontrolno stanje [8].

B. Softverska realizacija

Softverska realizacija ima funkciju da upravlja hardverom i obavi prijem i obradu podataka, gde obrada uglavnom uključuje prijem, prikaz i čuvanje podataka. Ona se sastoji od dva dela. Prvi deo se odnosi na aplikacijiu softvera na nivou STM32. Dok se drugi deo odnosi na obradu i prikaz podataka na samom kraju lanca. Blok dijagram prikazuje logiku softverske realizacije korišćene pri izradi rada.



Slika 3. Blok dijagram softverske realizacije.

Na početku je potrebno inicijalizovati sistem. Kod na mikrokontroleru je prilično jednostavan: imamo tajmer čiji je prekid podešen na svakih 3,9 ms. Kada se pritisne dugme, tajmer i prekidi se inicijalizuju i čitanje može da počne [9]. MCU šalje signal "START" da obavesti računar da se pripremi za prihvatanje podataka. Kada računar prepozna signal "START", on će obrisati prethodni grafikon (ako postoji) i pripremiti se za novo merenje[9]. Kada se aktivira prekid tajmera, uhvatićemo vrednost analognog pina, kao i vreme našeg merenja [9]. Svaki uzorak ECG podataka će pratiti ovu proceduru 15 sekundi. Kada prođe 15 sekundi, MCU će onemogućiti prekide i skočiti u beskonačnu while petlju.

VII. ZAKLJUČAK

Inovacije u medicinskim tehnologijama igraju ključnu ulogu u pružanju bolje zdravstvene nege. Jedna od takvih inovacija je predstavljena kroz ovaj rad na teorijskoj osnovi. Prenosivi elektrokardiogram (ECG) uređaj baziran na STM32 platformi, je uređaj koji omogućava brzu i efikasnu dijagnozu srčanih problema.

LITERATURA

[1] Pojam elektrokardiograma. https://intervita.rs/sta-je-ekg-elektrokardiogram/
[2] Kako radi elektrokardiogram i čemu služi https://www.stetoskop.info/pregledi/elektrokardiogram-ekg

[3] Normalni sinusni ritam srca https://www.martinloren.com/news/ecg-with-stm32-and-ad8232/
[4] Osnovne karakteristike mikrokontrolera https://www.st.com/en/microcontrollers-microprocessors/stm32f750n8.html
[5] Osnovne karakteristike ECG Click modula https://download.mikroe.com/documents/add-on-boards/click/ecg/ecg-click-schematic-v100.pdf
[6] Detaljniji opis ECG Click modula https://media.digikey.com/pdf/Data%20Sheets/MikroElektronika%20PDFs/ECG_click_web.pdf
[7] Opis programa STM32Cube MX https://www.st.com/resource/en/user_manual/um1718-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf
[8] Hardverska realizacija https://www.jstage.jst.go.jp/article/jaalr/3/2/3_3/_pdf
[9] Softverska realizcija https://www.mikroe.com/blog/ecg-click-mikroplot-complete-solution-human-heart-data-analysis

Realizacija 3D skenera korišćenjem *Raspberry Pi* 4 platforme

Milutin Dinić

Sadržaj - U ovom radu je prikazana realizacija 3D skenera korišćenjem Raspberry Pi 4 razvojnog sistema. Takođe je objašnjen ceo postupak kreiranja samog 3D objekta.

I. UVOD

3D skeniranje je postupak kojim se kreiraju digitalne informacije o stvarnim dimenzijama fizičkih objekata u simulacionom okruženju. Postoji nekoliko metoda za merenje stavki u 3D koje uključuju laserske skenere, svetlosne skenere, mašine za merenje koordinata i komercijalne skenere. Tehnologije 3D skeniranja generalno prikupljaju sirove podatke kao informacije o oblacima tačaka i pretvaraju ih u formate prilagođene korisniku. Sistemi za kontakt merenje, kao što je mašina za merenje koordinata, mere samo jednu tačku podataka istovremeno, što ih čini dugotrajnim i repetitivnim. S druge strane, kod 3D laserskih skeniranja, laser se kreće po površini mete, prikuplja podatke, a skeniranje se završava brzo u roku od nekoliko sati ili dana u zavisnosti od veličine.

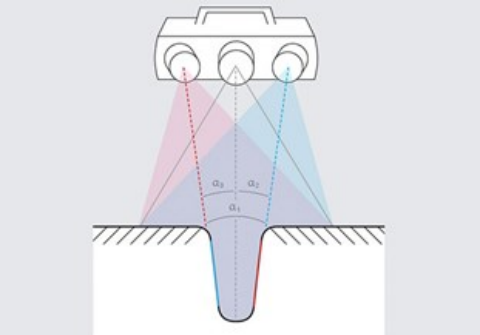
II. NAČIN RADA 3D SKENERA

3D skener radi sa strukturisanim svetlom koristeći principe triangulacije. Laserska tačka ili linija se projektuje na objekat sa laserskog uređaja i senzor (obično uređaj sa naelektrisanjem ili uređaj osetljiv na položaj) meri rastojanje do površine. Senzor precizno pomerajući uzorak (šablon svetlosti) preko površine dela subjekta, a dve kamere snimaju geometriju površine na osnovu izobličenja uzorka, izračunavajući 3D koordinate na osnovu merenja. Uzorak se projektuje na subjekt pomoću LCD projektora ili drugog stabilnog izvora svetlosti. Kamera, blago pomerena od projektora šablona, posmatra oblik šablona i izračunava rastojanje svake tačke u vidnom polju. 3D skener prikuplja i obrađuje milione X-Y-Z tačaka podataka u „oblaku tačaka“, stvarajući detaljnu digitalnu repliku objekta. Pošto su rastojanja i uglovi senzora i kamere poznati, primenjuje se princip triangulacije, dajući tačne, pouzdane rezultate merenja.

Slika 1 prikazuje kako strukturisani svetlosni 3D skeneri rade na principu triangulacije. Izvor svetlosti se nalazi između dve kamere postavljene pod uglom jedna prema drugoj, projektujući uzorak koji nalikuje prugama na površinu fizičkog objekta.

M. Dinić je sa smera Elektronika, Elektronskog fakulteta u Nišu, Srbija, E-mail: milutin.dinic@elfak.rs

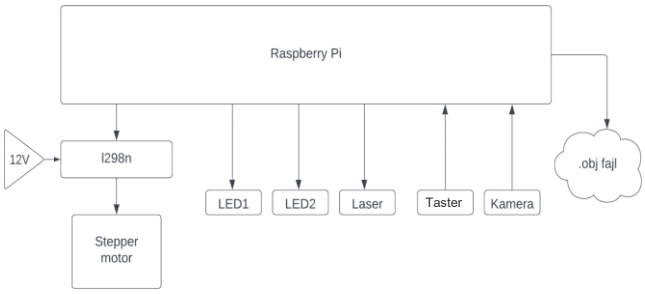
Pruge se menjaju po veličini i pravcu tokom prikupljanja podataka. Senzorne kamere posmatraju kontrast duž ivica pruga i dodeljuju tim pikselima X-Y-Z koordinate, brzo prikupljajući precizne, jasne podatke skeniranja koji sadrže vrlo malo teksture. Softver zatim transformiše tačke podataka u vidljive podatke koji čine tačnu, digitalnu repliku fizičkog dela.



Slika 1. Princip triangulacije.

III. MODEL LASERSKOG 3D SKENERA

Osnovni cilj ovog rada je da prikaže proces projektovanja i razvoj 3D skenera. Opšti blok dijagram laserskog 3D skenera prikazan je na slici 2.



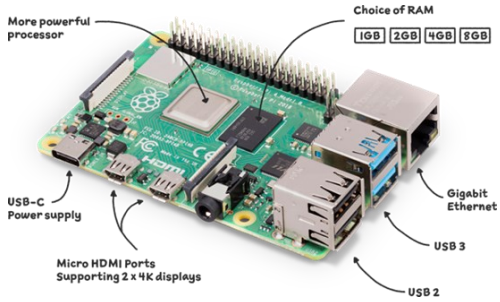
Slika 2. Logika iza projekta.

Centralna komponenta u ovom dizajnu je linearni laser koji se projektuje na vertikalni presek objekata. Ova projekcija se može snimiti kamerom (*PiCam*), podesiti perspektiva da bude uvek jednaka, a zatim pustiti sliku kroz filter. Nakon toga sledi obrada filtrirane slike. U obradi slike, rastojanje između svakog segmenta linije od centra objekta može se prikupiti. U radijalnim koordinatama, ova slika bi dala i r i z komponente. Treća dimenzija, Θ (teta), se zatim postiže rotiranjem objekta na novi presek.

IV. Opis *Raspberry Pi* 4 sistema

A. Hardver RPi 4

„*Raspberry Pi* 4“ („RPi 4“) je računar na jednoj ploči koji je izašao na tržište 2019. godine i od tada je postao izuzetno popularan zbog činjenice da je bio veliki korak u odnosu na prethodni model, "*Raspberry Pi* 3 B+". Sam uređaj sadrži *Broadcom* BCM2711 sa 4 64-bitna procesorska jezgra ARM Cortex-A72 tipa sa radnom frekvencijom od 1.5 GHz. Pored toga, ima radnu memoriju od 1, 2, 4 ili 8 GB, podršku za dva monitora preko dva mikro HDMI porta, dva USB 2 porta, 2 USB 3 porta, USB C port za napajanje, čitač za SD karticu na kojoj se nalazi operativni sistem RPi-a i jedan gigabitni ethernet port za povezivanje sa internetom. Izgled RPi-a može da se vidi na slici 3. Moguće je da se *Raspberry* poveže na internet bežičnim putem kao i da vrši prenos podataka upotrebom *Bluetooth*-a ali je za potrebno vršiti rekonfiguraciju prilikom inicijalizacije operativnog sistema na njemu.



Slika 3. *Raspberry Pi 4*.

B. Softver RPi 4

Za projekte vezane za IoT nudi se nekoliko opcija: *Raspbian* OS, *Raspberry* OS *Lite, Windows* 10 *IoT core, Ubuntu mate, Risc* OS, i drugi. Za ovaj projekat je korišćen „*Raspberry Pi* OS 32-BIT” i to „*Full*“ verzija koji je stavljen na SD karticu i kao takav se konfigurisao na samom RPi-u. „*Full*“ verzija nam nudi „GUI“ koji ima aplikativni program koji podržava „Python3“ programski jezik koji je i korišćen. Za instalaciju operativnog sistema je korišćen „*Raspberry Pi Imager*“ učitan sa oficijalnog sajta.

V. Komponente korišćene za projekat

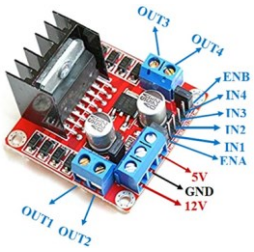
A. NEMA-17 Stepper motor

„*Stepper* motor“ je vrsta električnog motora koji se koristi za precizno pozicioniranje i kontrolu pokreta. Ova vrsta motora koristi elektromagnete kako bi se postigao precizan korak-pokret. *Stepper* motor se sastoji od rotora (unutrašnjeg dela) i statora (spoljnog dela). Rotirajući deo ima magnete, dok stator sadrži elektromagnete. Elektromagneti statora se napajaju strujom kako bi generisali magnetska polja. Rotirajući deo motora reaguje na ta polja i okreće se u koracima. Glavna prednost *stepper* motora je to da omogućava precizno i ponovljivo pozicioniranje. Svaki korak rotacije je unapred određen, što znači da se motor može koristiti za vrlo precizne zadatke, kao što su pokretanje preciznih mehanizama, pozicioniranje instrumenata ili kontrola 3D štampača. Izgled motora možemo videti na slici 4.



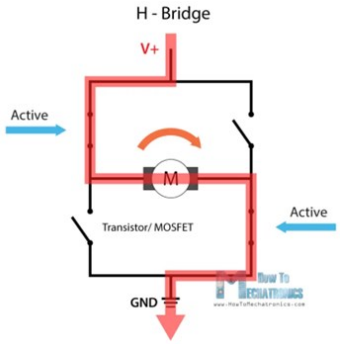
Slika 4. NEMA-17 *Stepper* Motor.

Da bismo pokrenuli motor potreban nam je drajver motora. To je integrisano kolo koje služi da obezbedi odgovarajući napon i struju motoru i da kontroliše njegove pinove. U našem slučaju je to 'l298n' integrisano kolo. Postavlja se kao sprega između motora i razvojnog sistema. Može da obezbedi do 2 ampera struje po kanalu. Izgled kola možemo videti na slici 5.



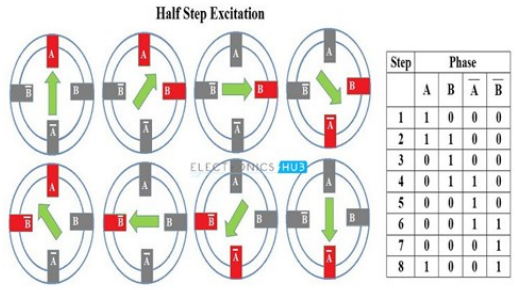
Slika 5. l298n drajver *stepper* motora.

'l298n' drajver radi na principu H-mosta. To je konfiguracija 4 tranzistora, povezanih u latinično H, što omogućuje da struja protiče u oba smera kroz motor. Time se postiže kontrola rotacije ali i kontrola smera. Primer H-mosta možemo videti na slici 6.



Slika 6. H-most.

Sama kontrola motora se radi korišćenjem tehnike polu koraka (*'Half-Step'*). Tehnika polu koraka je tehnika kojom se umesto da polarizujemo jednu po jednu fazu, koristimo dve faze u isto vreme. Ovim postupkom dobijamo veću rezoluciju koraka, rezolucija je duplirana, što nam daje tečnije pomeranje samog motora. Tehnika polu koraka je objašnjena na slici 7.

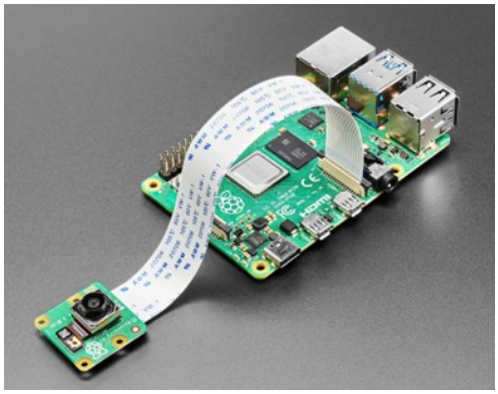


Step	Phase			
	A	B	$\overline{A}$	$\overline{B}$
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Slika 7. Tehnika polu koraka.

B. Raspberry Pi Camera Module 3

„*Raspberry Pi* kamera module 3“ je kompaktna kamera namenjena za korišćenje isključivo sa RPi razvojnim sistemima. Ona nam omogućava da slikamo i snimamo direktno na razvojnom sistemu i da to integrišemo u naše projekte. Na slici 8 možemo videti njen izgled i načina povezivanja.

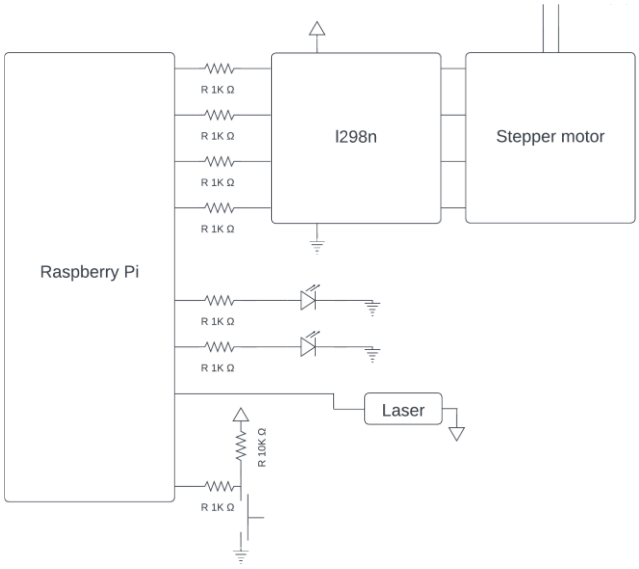


Slika 8. *Raspberry Pi* Camera Module 3.

VI. Povezivanje hardverskih komponenti

U cilju implementacije svih komponenata i omogućavanja njihovog zajedničkog rada neophodno je najpre ih povezati sa samim razvojnim sistemom. Povezivanje komponenata je prikazano na dijagramu na slici 9.

Kao što možemo videti na slici, kod povezivanja 'l298n' drajvera sa razvojnim sistemom korišćeni su otpornici kako bi obezbedili sam razvojni sistem. Sve komponente su povezane na odgovarajuće RPi pinove razvojnog sistema.



Slika 9. Dijagram povezivanja.

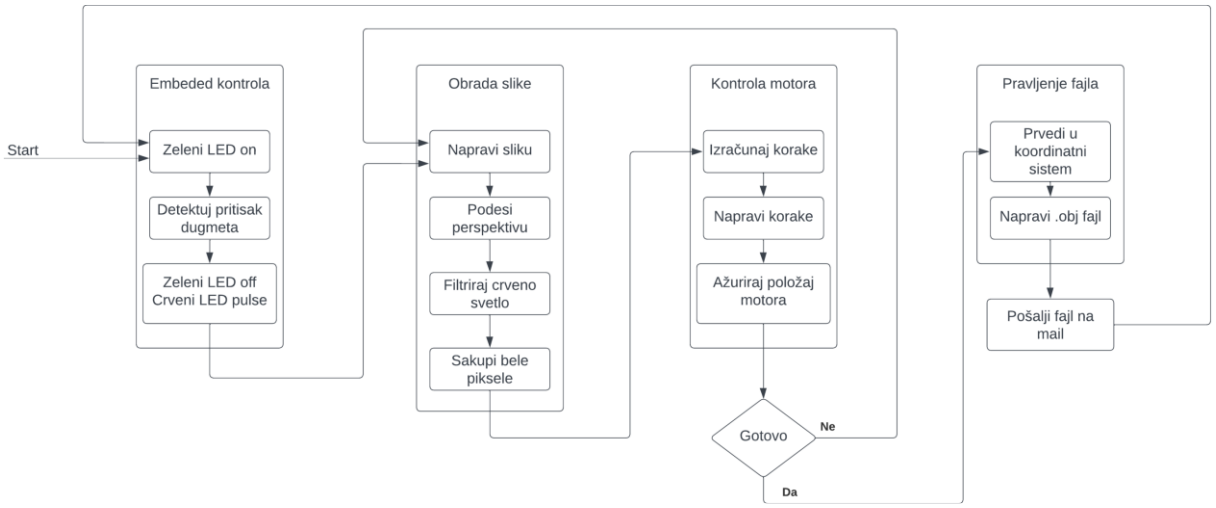
Drajver 'l928n' je povezan na razvojni sistem samo zbog kontrole. Postoji mogućnost povezivanja i napajanja ali obzirom na to da RPi može da obezbedi samo 0.8 A struje, nije bilo pametno uzeti napajanje za drajver od njega. Napajanje je dovedeno eksterno i ono obezbeđuje potreban napon i struju za pokretanje drajvera odnosno samog *stepper* motora.

Taster je povezan u takozvanom *pull-up* režimu rada. Postoje dva režima rada *pull-up* i *pull-down*. Razlika između njih je ta da kod *pull-up* režima kada taster nije pritisnut on drži visok nivo, odnosno ima visok napon na njemu, a kada se pritisne ima nisku vrednost. Kod *pull-down* je to suprotno. Da bismo izabrali željeni režim, samo je potrebno da otpornik od 10 KΩ stavimo na stranu napona, odnosno na stranu mase.

VII. Projektovanje softvera

Softver koji upravlja sistemom 3D skenera je projektovan da podrži snimanje određenog broja 3D skeniranja. Funkcionisanje softvera, odnosno algoritam, predstavljen je šemom na slici 10.

Na samom početku sistem se podiže, fajl '.bashrc' automatski uloguje sistem i pokrene naš program. Statusni LED se pali i čeka se pritisak tastera. Kada se on dogodi, statusni LED se menja i ulazimo u petlju obrade slike i kontrole motora.



Slika 10. Algoritam rada softvera.

Kada se završe sve iteracije, kreira se fajl koji se putem mejla šalje korisniku, odnosno nama. Program izlazi iz petlje, menja se statusni LED i očekuje se ponovni pritisak tastera.

A. Embeded kontrola

Embeded kontrola softvera nam omogućava da pokrenemo rad celog sistema bez korišćenja monitora ili bilo kojih drugih periferija. U samu embeded kontrolu su uključene i LED diode koje nam jasno prikazuju u kojoj je fazi sam proces. Zelena LED dioda svetli kada je sistem spreman za rad, odnosno, čeka se pritisak tastera koji će pokrenuti proces obrade slike. Kada se detektuje pritisak tastera, zelena dioda se isključuje i crvena počinje da pulsira. Za to se koristi klasična PWM modulacija.

B. Obrada slike

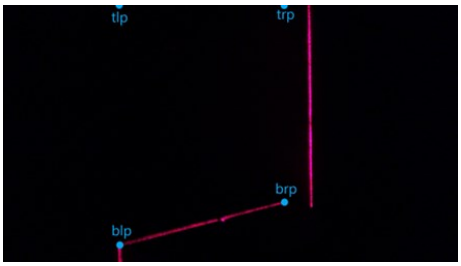
Da bismo uopšte mogli da dođemo do korisnih informacija iz same slike, moramo je prvo filtrirati. Ali pre samog filtriranja moramo prvo rešiti problem perspektive.

Naime, kada gledamo sliku iz nekog ugla, kao u našem slučaju, nemoguće je tačno i konstantno odrediti visinu objekta. Na slici 11. su prikazana dva latinična slova 'L', jedno pravilno i jedno okrenuto za 180°. U oba slučaja je visina ista, ali zbog perspektive nam se čini da su visine različite, odnosno da je slovo bliže nama više od ovog što se nalazi dalje

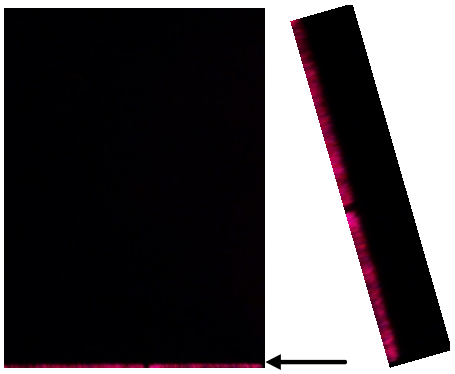
Da bismo rešili ovaj problem koristićemo gotov kod iz *openCV* biblioteke. On uzima 4 određene tačke i odseca to parče slike, samim tim praveći iz trapezoidnog oblika u pravougaoni oblik sliku koja će biti konstantna u svakoj iteraciji. Na slikama 12 i 13 možemo videti sliku pre i posle transformacije.



Slika 11. Problem perspektive.



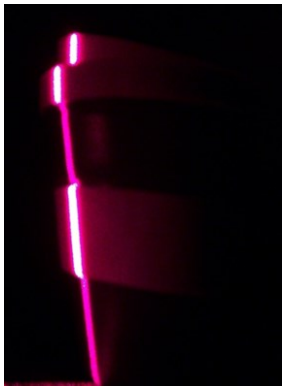
Slika 12. Originalna slika.



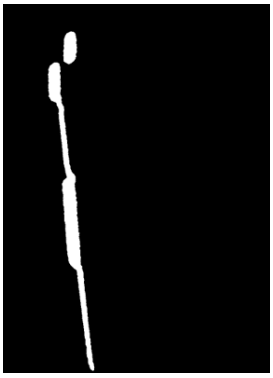
Slika 13. Transformisana slika.

Sledeći korak je filtriranje slike. To smo radili korišćenjem *inRange()* funkcije *openCV* biblioteke.

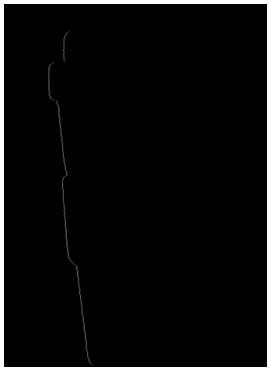
Podešavamo granicu po kojoj se prikuplja svetlost tako da prikupljena svetlost ostane samo na objektu. Tu se javlja problem prirode objekata koji se skenira. Tamniji objekti zahtevaju manju, dok svetliji veću granicu. Tako da moramo naći neku srednju vrednost koja će zadovoljiti potrebe svih materijala, odnosno boja. Tako dobijenu sliku propustamo kroz petlju kojom se izdvaja samo svaki prvi piksel sa leve strane u svakom redu i tako dobijena slika se čuva. Na slikama 14, 15 i 16 možemo videti izgled originalne slike, izgled filtrirane slike i izgled slike sa prikupljenim pikselima.



Slika 14. Originalna slika.



Slika 15. Filtrirana slika.



Slika 16. Izdvojeni pikseli.

C. Kontrola motora

Nakon uspešne obrade jedne slike, neophodno je rotirati objekat kako bismo nastavili skeniranje iz drugog ugla. Da bismo to uradili, neophodno je dobro kontrolisati *stepper* motor koji se nalazi ispod objekta. Za to je korišćena promenljiva koja prati stanje, odnosno poziciju motora, i na određen način ga pokreće, vodeći računa koji se ulazi motora koriste. Rad motora je opisan u delu IV.A.

D. Generisanje fajla

Da bismo napravili mrežu („*mesh*“) od obrađenih slika, moramo prvo da prevedemo svaki beli piksel obrađene slike u teme („*vertex*“) u 3D prostoru. Zbog toga što smo sakupljali slike u delovima sa cilindričnom simetrijom, logično je bilo da se sakupljaju polarne koordinate, tako da je visina predstavljala Z-osu, udaljenost od centra platforme koja se okreće R-osu, i rotacija motora, odnosno razmak između delova, Θ-osu.

Međutim, za pravljenje 3D objekta, možemo koristiti samo podatke predstavljene u dekartovom koordinatnom sistemu. Zbog toga je neophodno prevesti podatke iz polarne u dekartov koordinatni sistem.

Kada dobijemo željeni oblik podataka, oni se čuvaju u listu koja se čuva u još jednu listu koja sadrži listu temena za svako skenirano parče. Kada su sve slike procesirane i prevedene u temena koja su u željenom formatu, biramo koja ćemo temena zapravo predstaviti u finalnoj mreži. Čuvamo gornje i donje teme i biramo, u zavisnosti od rezolucije skeniranja, broj temena koja ćemo koristiti. Takođe, zbog toga što nisu sve liste koje smo kreirali iste, moramo da nađemo najmanju i ostale prilagodimo njoj. Tako dobijamo uniformni oblik na kraju kreiranja.

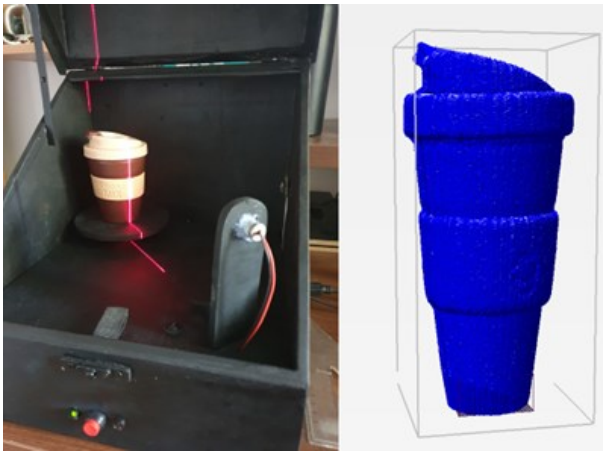
Kada su sve liste, za svako parče, napravljene, ostaje nam samo da ih upišemo u fajl koji će imati standardnu ekstenziju .obj koja je pogodna za pregled i modifikaciju, kao i štampanje korišćenjem 3D štampača.

VIII. REZULTATI

Skenirani objekti imaju približan oblik. Nisu baš identični, ali je sličnost velika. Možemo ih pregledati korišćenjem različitih programa. U našem slučaju je korišćen *Repetier* zbog toga što ima mogućnost da spoji delove koje nemaju definisane tačke (vrh i dno objekta).

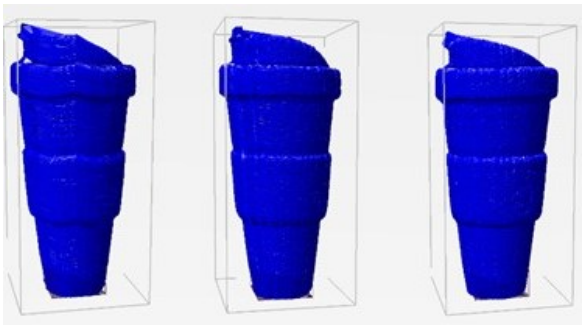
Jedna od bitnih karakteristika sistema je konstantnost. Svaki put kad skeniramo isti objekat dobijamo sličan, ako ne i identičan izgled objekata. Rezultate skeniranja možemo videti na slici 17.

Korišćenjem promenljive *numItt* možemo fino podešavati koliko ćemo imati delova objekta koje ćemo prevoditi u 3D objekat.



Slika 17. Fizički objekat i njegov skenirani izgled.

Pošto imamo 400 koraka motora, možemo izabrati ΔΘ da nam određuje uglovnu rezoluciju. Po difoltu, rezolucija je postavljena na 40 što nam daje 400/40=10 koraka pri svakoj slici, odnosno svakom parčetu. Ovaj broj je izabran najviše zbog vremena skeniranja koje traje oko 1 minut, a rezultat je zadovoljavajuć. Rezolucija se može povećavati čime se dobija kvalitetniji izgled skeniranog objekta, ali vreme potrebno za skeniranje raste. Rezultate različitih rezolucija možemo videti na slici 18.



Slika 18. Rezolucije 25, 100 i 200 respektivno.

Uprkos odličnim rezultatima skeniranja, postoje određena ograničenja. Prvo, korišćenje lasera i način na koji svetlo interaguje sa objektima različitih materijala. Većinu objekata koji su providni ili imaju reflektivnu površinu nije moguće skenirati. Dobijeni skenirani objekat ima mnogo nedostataka i ne liči na originalni. Takođe, objekti sa oštrim ivicama bivaju izobličeni. Zbog prirode svetla koje pada na objekat, neki deo objekta može da zakloni drugi deo objekta i da ga kamera uopšte ne vidi. Problem predstavljaju i neka udubljenja u objektu koja bivaju predviđena i nepostojeća u finalnom skeniranom objektu (dešava se kod kompleksnijih objekata). Ali ovo su problemi sa kojima se suočavaju i mnogi drugi, sofisticiraniji skeneri.

IX. Zaključak

Inspiracija za ovaj projekat došla je iz prethodnog iskustva sa 3D štampanjem i brzim prototipovima. Mogućnost brzog skeniranja postojećih objekata može ih učiniti lakim za reprodukciju i olakšava dizajniranje drugih delova oko njih. Cena ovih skenera je često prilično skupa.

U ovom radu je predstavljena realizacija laserskog 3D skenera zasnovanog na Raspberry Pi platformi, koji predstavlja jedan izvodljiv ugrađeni sistem u našim uslovima. Na samom kraju može se konstatovati da je rezultat skeniranja zadovoljavajući. Skenirani objekat se može popraviti naknadno u softveru za obradu objekata i može se reprodukovati. Dobra strana svega je što je skener finansijski pristupačan i što ne koristi previše energije i resursa, odnosno efikasan je. Može naći primenu u obrazovnom procesu studenata, a može se koristiti i u istraživačke svrhe.

Literatura

[1] Adrian Rosebrock, 4 Point OpenCV getPerspective Transform Example, 2014, https://pyimagesearch.com

[2] Getting started with your Raspberry Pi, https://www.raspberrypi.com/documentation/computers/raspberry-pi.html, 2023

[3] Raspberry Pi Stepper Motor Control using L298N, https://www.electronicshub.org/raspberry-pi-stepper-motor-control/, 2023

[4] Wavefront OBJ File Format, https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml, 2023

[5] Cartesian coordinates, https://mathinsight.org/cartesian_coordinates, 2023

[6] The Picamera2 Library, 2023, https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf

[7] L298N Dual H-Bridge Motor Driver, http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf, 2023

[8] L298N Motor Driver – Arduino Interface, How It Works, Codes, Schematics, 2023, https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

[9] Kanishk Godiyal, Guide to NEMA 17 Stepper Motor Dimensions, Wiring Pinout, 2022, https://www.etechnophiles.com/guide-to-nema-17-stepper-motor-dimensions-wiring-pinout/

[10] OpenCV AI Competition, 2023, https://opencv.org/

[11] NumPy - The fundamental package for scientific computing with Python, https://numpy.org/, 2023

[12] SMTP protocol client, https://docs.python.org/3/library/smtplib.html, 2023

Verifikacija AMBA APB protokola primenom UVM metodologije

Vladimir Rudić

Sadržaj - Cilj ovog rada je da se predstavi proces projektovanja verifikacione komponente koja omogućava verifikovanje APB protokola. Za rad je korišćen jedan od poznatijih jezika za verifikaciju, *System Verilog*, kao i UVM metodologija koja je razvijena za isti taj jezik. Opisane su sve vrste operacija, svi signali APB protokola kao i svaka komponenta sistema. Rad prati detaljan opis za svaku od komponenata koja je značajna u procesu kreiranja verifikacione komponente.

I. UVOD

Testiranje funkcionalnosti čipa pre same proizvodnje istog je možda i najbitniji korak u procesu projektovanja nekog sistema. Ukoliko čip dođe na tržište bez detaljne verifikacije, postoji velika šansa da uređaj ne radi po specifikaciji, pa čak i da uopšte ne radi. Ovakvi čipovi i sistemi, koji su često delovi većih sistema ili uređaja, moraju biti povučeni iz prodaje, čime su gubici za kompaniju koja proizvodi čip, kao i kompanija koje koriste proizvedeni čip, ogromni. Zbog toga je jako bitno da se pre proizvodnje samog čipa šansa da dođe do pogrešnog funkcionisanja uređaja dovede do minimuma [1].

U ovom radu će biti objašnjen postupak projektovanja verifikacione komponente za APB komunikacioni protokol, kao i objašnjenje svake komponente unutar verifikacionog okruženja. Upoznajemo se sa APB prokolom, njegovim signalima i mogućim operacijama. Nakon ovoga, detaljno je opisana univerzalna verifikaciona komponenta i njeni sastavni najbitniji delovi. Na kraju imamo rezultate samih testova.

II. APB PROTOKOL

AMBA je set standardizovanih interfejsa i protokola razvijen od strane kompanije ARM za poboljšanje komunikacije između različitih komponenti u ugrađenim sistemima. AMBA se često koristi u projektovanju složenih čipova i sistemskih rešenja [2]. AMBA APB (eng. *Advanced Peripheral Bus*) predstavlja deo AMBA arhitekture koji se fokusira na povezivanje perifernih komponenti sa glavnim sistemskim magistralama. Ovaj protokol je osmišljen da olakša komunikaciju između procesora i raznovrsnih periferija u ugrađenim sistemima [4].

Ključna karakteristika APB protokola je njegova jednostavnost i fleksibilnost, što ga čini idealnim za implementaciju na različitim tipovima perifernih uređaja. Ovaj protokol se zasniva na jednom master-slejv modelu, gde master uređaji (najčešće procesori) iniciraju transakcije prema APB periferijama (slejv uređaji). Transakcije se realizuju putem jednostavnih kontrolnih signala i sekvencijalnih operacija koje omogućavaju čitanje i pisanje podataka iz periferija [4].

Signali APB protokola su:

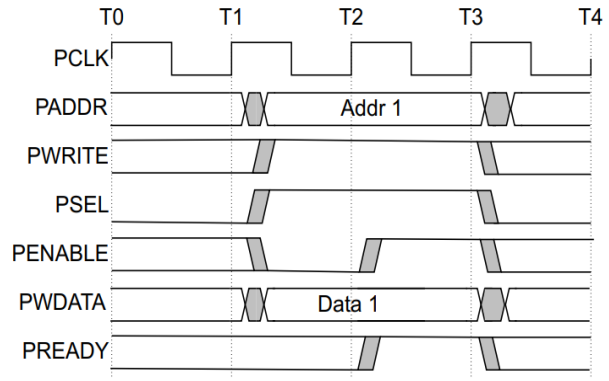
- PCLK - globalni APB clock,
- PRESETn - APB asinhroni reset, active LOW,
- PADRR - APB magistrala adrese,
- PSELx - slejv je izabran i indikator početka transfera podatka,
- PENABLE - indikator drugog ciklusa APB transfera,
- PWRITE - HIGH kada je upis, LOW kada je čitanje,
- PWDATA - podatak koji se upisuje,
- PSTRB - kontrolni signal,
- PPROT - signal za informacije podatka,
- PREADY - signal koji se koristi da produži transfer od strane slejva,
- PRDATA - podatak koji se čita od strane slejva,
- PSLVERR - indikator greške od strane slejva.

Operacije APB protokola mogu biti upis ili čitanje, bez čekanja ili sa čekanjem. Same operacije i pravilan tok signala ćemo posmatrati na primeru upisa bez čekanja i čitanju sa čekanjem da bismo prošli kroz sve moguće varijante.

Faza postavljanja (eng. *Setup phase*) za operaciju pisanja se dešava kada je PSEL aktivan, odnosno u trenutku T1 na slici 1. U tom trenutku PADDR, PWRITE, PWDATA, PSTRB i PPROT moraju biti validni. Vidimo da je u tom trenutku PWRITE na logičkoj jedinici čime inicira upisivanje, a na PWDATA busu imamo podatak koji želimo da upišemo. Jedan ciklus takta kasnije u trenutku T2, PENABLE se aktivira, označavajući početak faze pristupa (eng. *Access phase*). Na uzlaznoj ivici PCLK, Slejv uređaj aktivira PREADY kako bi signalizirao da će podaci za pisanje biti prihvaćeni u trenutku T3. PADDR, PWDATA i drugi kontrolni signali poput PPROT i PSTRB moraju ostati stabilni dok se prenos ne završi. Na kraju prenosa,

V. Rudć je diplomirani inženjer elektrotehnike i računarstva na modulu Elektronske komponente i mikrosistemi sa Katedre za mikroelektroniku, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: vladimirrudic@elfak.rs.

PENABLE se deaktivira. PSEL takođe prestaje da važi, osim ako postoji drugi prenos prema istom perifernom uređaju [4].

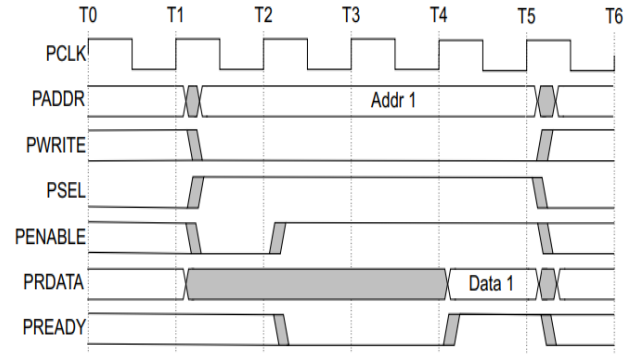


Slika 1. Upis bez čekanja.

Operacije sa čekanjem su vrlo slične kao bez čekanja. Zapravo faze postavljanja i faze pristupa su identične osim toga što slejv produžava sam transfer držeći PREADY na nuli dodatan broj clock ciklusa. Kod čitanja signali PADDR, PWRITE, PSEL i PENABLE su isti kao kod upisa, odnosno faze postavljanja i faze pristupa su identične, osim toga što PWRITE ima vrednost nule i time inicira čitanje. Takođe, vrednost PWDATA nam nije od značaja i slejv obezbeđuje podatke na PRDATA, dok je PREADY aktivan [4]. Sledeći signali ostaju nepromenjeni dok PREADY ostaje na nuli:

- signal adrese, PADDR,
- signal operacije, PWRITE,
- signal selekcije, PSELx,
- signal dozvole, PENABLE,
- signal za pisanje podataka, PWDATA,
- signal za kontrolu, PSTRB,
- signal tipa zaštite, PPROT.

PREADY može imati bilo koju vrednost kada je PENABLE nizak [4]. Na slici 2 data je operacija čitanja sa čekanjem.

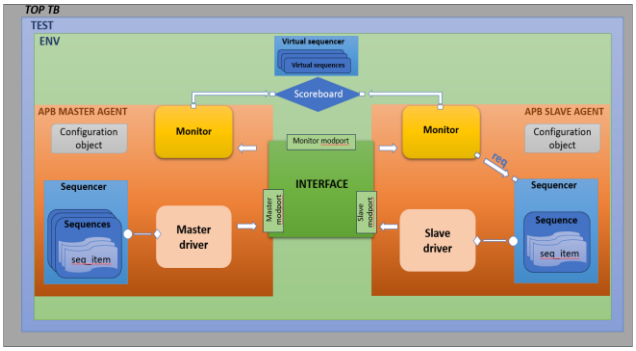


Slika 2. Čitanje sa čekanjem.

III. ARHITEKTURA SISTEMA

Koristeći UVM, arhitektura je standardizovana i sastoji se od određenih komponenata, koje omogućavaju prvenstveno modularnost, laku ponovnu upotrebu i laku modifikaciju verifikacionog okruženja. Sve ove komponente su enkapsulirane u agentu, koji se kasnije uključuje u okruženje [1,3].

Na slici 3 dat je izgled sistema realizovanog u ovom projektu. Arhitektura okruženja sastoji se od dva agenta u aktivnom režimu: master i slejv, interfejsa, *skorborda* i virtuelnog sekvencera i virtuelnih sekvenci. Svaki od ovih agenata sastoji se od sekvencera, *drivera*, monitora i konfiguracionog objekta. Sekvenceri i driveri povezani su sa *seq_item_export* i *seq_item_port*. *Skorbord* komponenta povezana je sa monitorima mastera i slejva putem njihovih TLM portova. Interfejs ima 3 modporta: monitor modport za monitor, master modport za master *driver* i slejv modport za slejv *driver*. Interfejs pored sih signala APB protokola u sebi sadrži *protocol checkere* u vidu *assertion*- a. Agent slejva je reaktivan [1].



Slika 3. Arhitektura sistema.

Za samu verifikaciju od ključnog značaja su komponente: konfiguraciona klasa, transakcija, sekvence, *coverage*, *protocol checkeri*, kao i testovi i sami rezultati.

A. Konfiguraciona klasa

Master i slejv agenti imaju instanciranu konfiguracionu klasu (eng. *Configuration object*) unutar sebe, odnosno *apb_cfg*. U sebi ima dve opcije:

•is_active_agent - Da li je agent aktivan ili pasivan,

•always_ready - Promenljiva koja se koristi samo za *always_ready_test*, tokom koje je *"pready"* signal na jedinici tokom cele simulacije

B. Transakcija

Transakciju (eng. *Transaction*) predstavlja klasa *apb_item*. Kao i sve ostale klase registruje se u fabriku pomoću *uvm* makroa. Ova klasa je objekat tako da se registruje kao objekat. Drugačije se nazivaju *item*- i. Promenljive unutar transakcija su u nekim slučajevima

vezani za signale u interfejsu, a u nekim slučajevima su promenljive od kojih zavisi drajvovanje. Uvek su tipa *rand* kako bi bile nasumične. Takođe, unutar transakcija se pišu i *constrein*t- i, odnosno promenljivama zadajemo opsege vrednosti ili verovatnoću raspodele unapred zadatih vrednosti [1]. Promenljive su:

• operation - upis ili čitanje,
• paddr - adresni bas,
• strobe – strobe,
• transaction_delay - kašnjenje između transakcija,
• data - podatak koji će biti pwdata ili prdata u zavisnosti od operacije,
• protection - bitovi za pprot signal,
• response_time - kašnjenje APB transfera. Slejv produžava transfer tako što drži PREADY na nuli,
• slave_error - greška prilikom transfera.

C. Sekvence

Sekvence (eng. *Sequence*) u verifikaciji možemo opisati kao skup različitih akcija koje izvršavaju čekanje nekih događaja ili uslova, generišu transakcije na određeni način i prosleđuju te generisane transakcije u određenom trenutku drajveru preko sekvencera. Uz pomoć sekvenci možemo kreirati sve scenarije koji su od interesa [1,3]. U master biblioteci imamo sledeće sekvence:

• apb_master_write_seq - operacija upisa sa ostalim nasumičnim poljima,
• apb_master_read_seq - operacija čitanja sa ostalim nasumičnim poljima,
• apb_master_random_seq – sva nasumična polja sa master strane,
• apb_master_back_to_back_seq - transaction_delay=0 i ostala sva nasumična polja sa master strane.

Slejv agent je reaktivan, tako da ima samo jednu sekvencu, odnosno *apb_slave_response_seq*. Njena konstukcija je nešto komplikovanija u odnosu na master sekvence. Njen *body task* je uvek aktivan i nalazi se u beskonačnoj petlji. Ona čeka određeni zahtev koji dolazi od slejv monitora preko sekvencera do sekvence. Nakon prepoznavanja određenog zahteva, na osnovu toga da li je zahtev za upis ili čitanje i on zadaje određenu operaciju transakciji [5].

D. Coverage

Funkcionalni *coverage* predstavlja stepen pokrivenosti funkcionalnosti dizajna koje se proveravaju u okviru testova. Bitno je napisati kod *coverage* dobro, kako se određene funkcije dizajna ne bi predvidele. U okviru *coverage*- a pišemo *covergroup*- e u okviru kojih imamo *coverpoint*- e. *Coverpoint*- i predstavljaju promenljive čija se vrednost proverava. Ta vrednost ili opsezi vrednosti se zadaju u okviru *bin*- ova. Smatra se da je bin pokriven kada promenljiva ima vrednost iz njegovog opsega. Sakupljanje *coverage*- a se okida ili nekim *eventom* ili *sample()* funkcijom. U ovom projektu *coverage* je pisan u monitoru i tu se i sakuplja korišćenjem *sample()* funkcije. Posmatraćemo primer *coverpointa* za adresu. Ako se pogodi vrednost singala iz jednog opsega, može se smatrati da će dizajn funckionisati ispravno i za sve druge vrednosti iz tog opsega:

bins_10_buckets[10]={[32`h0000_0000:32`hffff_ffff}];

U ovom slučaju, opseg od 32`h0000_0000 do 32`hffff_ffff će biti podeljen na 10 manjih opsega, i proveravaće se vrednost signala za svaki od tih opsega. Zbog dodatne sigurnosti proverava se i da li je bit na svakoj poziciji imao vrednost 0 ili 1 korišćenjem ključne reči *wildcard*. Na pozicijama na kojima je napisan "?" se ne proverava vrednost signala, proverava se samo na pozicijama gde je napisana 0 ili 1.

```
coverpoint item.addr{
wildcard bins address1_0
={32'b????_????_????_????_????_????_????_???1};
wildcard bins address0_0
={32'b????_????_????_????_????_????_????_???0};
wildcard bins address1_1
={32'b????_????_????_????_????_????_????_??1?};
wildcard bins address0_1
={32'b????_????_????_????_????_????_????_??0?};
.
.
.
wildcard bins address1_31
={32'b1???_????_????_????_????_????_????_????};
wildcard bins address0_31
={32'b0???_????_????_????_????_????_????_????};
}
```

E. Protokol čekeri

Protokol čekeri su jedni od najbitnijih delova verifikacione komponente. Njima se ispituje ispravnost protokola i proverava samo verifikaciono okruženje. Protokol čekeri su realizovani u interfejsu u vidu asercija (eng. *assertions*). Čekeri u ovom projektu su:

• access_phase_checker - PENABLE mora da se podigne na jedinicu jedan clk posle psel,
• assynchronous_reset_checker - signali moraju da imaju vrednost nule prilikom reseta,
• strobe_during_read_checker – PSTROBE mora da bude nula tokom operacije čitanja,
• slave_error_checker - PSLVERR jedino može da bude aktivan samo kada su i psel, penable and pready na jedinici,
• prdata_must_be_one_after_penable_checker – PREADY mora da se podigne na jedinicu do kraja simulacije,
• x_values_signals_checker - signali ne smeju da imaju nepoznate vrednosti (X/Z),

• stable_signals_during_transfer_checker – kontrolni signali moraju da budu stabilni tokom transfera,

• penable_can_be_one_when_psel_is_one_checker - PENABLE jedino može da bude jedinica kada je i PSEL jedinica,

• prdata_during_write_operation_checker - PRDATA ne sme da se menja tokom operacije upisa.

IV. TESTOVI I REZULTATI

Test (eng. *Test*) predstavlja poslednji korak u pisanju konkretnog stimulusa za naš DUT. Testovima se simuliraju specifične situacije i proverava se ispravnost uređaja koji se verifikuje. Mogu biti nasumični ili specifični za konkretno ponašanje uređaja. U ovom projektu imamo jedan osnovni *apb_base_test* u kome su instancirani i kreirani u *build* fazi okruženje i konfiguraciona klasa okruženja. U *run* fazi koristimo metodu *set_drain_time* kojom omogućavamo da nam se simulacija završi kako treba, odnosno da ne ostaju procesi koji nisu završeni skroz. Testove koje smo koristili su:

• apb_write_test - testiranje funkcionalnosti operacije upisa,

• apb_read_test - testiranje funkcionalnosti operacije čitanja,

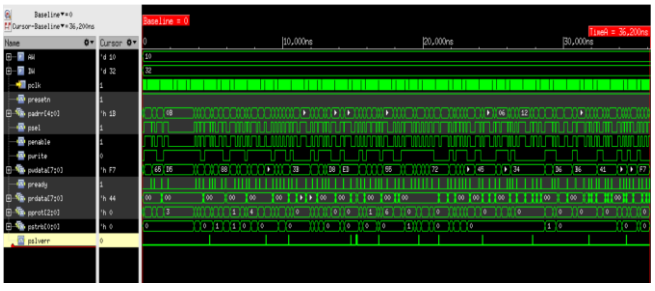
• apb_back_to_back_test - testiranje nasumičnih operacija dok je "psel" aktivan sve vreme tokom cele simulacije,

• apb_always_ready_test - testiranje nasumičnih operacija dok je "pready" aktivan sve vreme tokom cele simulacije,

• apb_random_test - testiranje nasumičnih operacija sa skroz nasumičnih poljima iz transakcije,

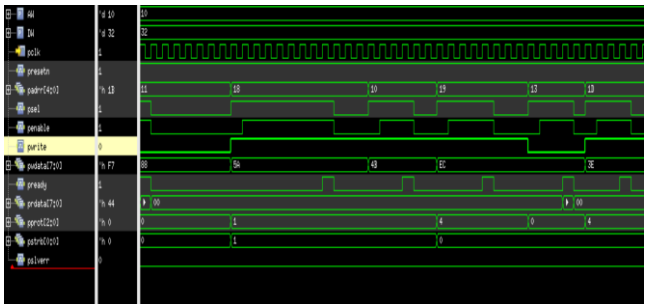
• apb_reset_on_the_fly_test - testiranje nasumičnih operacija sa skroz nasumičnih poljima iz transakcije sa resetom u toku simulacije.

Izgled simulacije na samom *waveform*- u nakon pokretanja *apb_random_testa* je dat na sledećim fotografijama:



Slika 4. Nasumični test.

Možemo videti tri operacije upisa na različitim adresama za redom sa nekim brojem čekanja, potom operaciju čitanja i na kraju još jednu operaciju upisa. Primećujemo da su adrese nasumične, da su operacija čitanja i pisanja nasumične takođe, broj čekanja u taktovima od strane slejv agenta, kao i sami podaci koji se upisuju.



Slika 5. Operacije nasumičnog testa.

V. ZAKLJUČAK

Neprekidnim razvojem tehnologije, konstantno raste i potreba za usavršavanjem i inovativnim metodama, kako iz oblasti elektronike, tako i iz oblasti programiranja, time čineći ovu granu industrije veoma interesantnom i popularnom. Testiranje funkcionalnosti čipa pre same proizvodnje istog je možda i najbitniji korak u procesu projektovanja nekog sistema. Ukoliko čip dođe na tržište bez detaljne verifikacije, postoji velika šansa da uređaj ne radi po specifikaciji, pa čak i da uopšte ne radi. Ovakvi čipovi i sistemi, koji su često delovi većih sistema ili uređaja, moraju biti povučeni iz prodaje, čime su gubici za kompaniju koja proizvodi čip, kao i kompanija koje koriste proizvedeni čip, ogromni. Zbog toga je jako bitno da se pre proizvodnje samog čipa šansa da dođe do pogrešnog funkcionisanja uređaja dovede do minimuma.

ZAHVALNICA

Autor ovog rada se zahvaljuje prof. dr Danijelu Dankoviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji ovog rada. Takođe zahvaljuje se kompaniji Veriest, koja je omogućila korišćenje svih potrebnih alata i dodatnih materijala.

LITERATURA

[1] Verilog Testbench, Chipverify. Dostupno na: https://www.chipverify.com/.

[2] IEEE Standard for System Verilog, "Unified Hardware Design, Specification, and Verification Language", IEEE Computer Society and the IEEE Standards Association Corporate Advisory Group.

[3] Universal Verification Methodology (UVM) 1.2 User's Guide – Accellera. Dostupno na: https://www.accellera.org/images/downloads/standards/uvm/uvm_users_guide_1.2.pdf.

[4] AMBA APB Protocol - ARM. Dostupno na: https://developer.arm.com/documentation/ihi0024/latest/

[5] M. Litterick, J. Montesano, T. Reddy, "Mastering reactive slaves in UVM", 2015.

Korišćenje Arduino razvojnog okruženja u Python-u

Anđela Jovanović

Sadržaj - Kako je *Arduino* razvojno okruženje široko rasprostranjeno u različitim projektima u elektronici, razvijene su i različite metode kojima se ono programira, u zavisnosti od afiniteta samih projektanata. Jedan od načina je programiranje pomoću *Python*-a, odnosno pisanjem koda u *PyCharm* tekst editoru ovog programskog jezika. Cilj ovog rada je prikaz osnovnih funkcionalnosti korišćenja *Arduino Mega 2560* mikrokontrolerske pločice putem *Python*-a, na primeru merenja temperature pomoću *LM35* senzora temperature i postavljanja odgovarajućih svetlosnih indikacija uz pomoć *LED* dioda u zavisnosti od toga da li sistem nije pokrenut, da li je korisnik pokrenuo ili stopirao očitavanje temperature. U toku očitavanja temperature, podaci se šalju na *ThingSpeak* platformu.

I. Uvod

Zbog velikih potreba za korišćenjem *Arduino* platforme, mnogi softverski paketi su omogućili funkcionalnosti za rad sa *Arduinom*. Za potrebe ovog rada, izabran je programski jezik *Python*, čija je sintaksa slična sa *Arduino IDE* okruženjem koje je razvijeno namenski za rad sa *Arduinom*, ali sa tom razlikom što je *Python* danas jedan od glavnih programskih jezika kod modela mašinskog učenja, u robotici itd., tako da je kombinacija mikrokontrolera i različitih metoda koje *Python* omogućava odličan izbor u takvim primenama. Njegove funkcionalnosti će biti prikazane kroz čitanje analognih ulaza sa mikrokontrolera i upisivanje na digitalne izlaze. Naravno, moguće su i sve druge kombinacije kao kod klasičnog programiranja putem *Arduino IDE* razvojnog okruženja. U ovom slučaju je kao parametar izabrana temperatura okoline kao pokazni primer. Kako je često podatke potrebno skladištiti na nekom serveru, to će biti učinjeno pomoću *ThingSpeak* platforme koja će služiti za vizuelizaciju dobijenih mernih podataka.

II. *Python* programski jezik

Python je viši programski jezik opšte namene. Razvio ga je Guido van Rossum u Holandiji. Kroz njega se mogu razviti različite vrste softvera, web aplikacija, igara, modela mašinskog učenja... Podržava raznovrsne programske paradigme kao što su objektno-orijentisano programiranje, strukture, funkcionalno programiranje i veliki broj biblioteka koje se mogu uključiti kako bi se nadogradile njegove mogućnosti. Načela na kojima se zasniva su jednostavnost, čitljivost, eksplicitnost i estetika [1]. Korisnik programira u *Python*-u kroz editor koji se zove *PyCharm*. *PyCharm* je integrisano razvojno okruženje za programiranje u *Python*-u razvijeno od strane *JetBrains*-a, softvreske kompanije koja se bavi upravo razvojem ovakvih okruženja. Dostupno je za sve platforme (*Linux*, *Windows*, *Mac*...). Sadrži alate za analizu koda, otklanjanje grešaka i testiranje [2]. Biblioteka koja omogućava kontrolisanje *Arduino Mega 2560* uz pomoć *Python*-a je *pyfirmata*, zasnovana na *Firmata* protokolu. Da bi bilo moguće korišćenje *Arduino* okruženja kroz *Python*, neophodno je u *Arduino IDE* pronaći primer gotovog koda na sledeći način *File – Examples – Firmata – StandardFirmata* i spustiti ga na pločicu. Nakon toga, moguće je koristiti Arduino kroz *PyCharm* editor uključivanjem biblioteke *pyfirmata* koja sadrži neophodne alate [3].

III. Arduino Mega 2560

Ova verzija Arduino okruženja sadrži ukupno 16 analognih pinova i 54 digitalna pina, što je izuzetno zahvalno za povezivanje sa raznim hardverskim uređajima. Digitalni pinovi omogućavaju slanje i primanje jedne od dve vrednosti: 0 ili 1. Analogni pinovi omogućavaju primanje analognog napona iz opsega 0 V- 5 V. Zatim se taj napon pomoću desetobitnog *A/D* (*Analog/Digital*) konvertora pretvara u vrednosti 0-1023. Pločica je snabdevena pinovima koji omogućavaju prihvatanje *PWM* (*Pulse Width Modulation*) signalima gde se u toku jedne periode smenjuju logičke vrednosti 0 i 1 u određenom trajanju u toku periode (što se može koristiti za kontrolu brzine motora ili količine svetlosti diode). Za napajanje komponenata povezanih za Arduino, koriste se pinovi napona napajanja sa mogućim vrednostima od 5 V i 3,3 V, a postoje i pinovi za obezbeđivanje referentnog potencijala (masa) označeni sa *GND*. Sama pločica napajanje može dobijati putem *USB* kabla kojim se i povezuje sa računarom na kome se vrši programiranje [4].



Slika 1. Arduino Mega 2560 [4].

A. Jovanović je student master studija na smeru Računarsko upravljanje sistemima i merna tehnika i saradnik u nastavi na Katedri za automatiku, Elektronski fakultet u Nišu, Aleksandra Medvedeva br. 14, 18000 Niš, Srbija, e-mail: andjela.jovanovic@elfak.ni.ac.rs

IV. LM35 SENZOR

Ova komponenta služi za analogno merenje temperature u opsegu od -55 °C – 150 °C. Izlazni analogni napon je direktno proporcionalan temperaturi u stepenima Celzijusa. Sam senzor sadrži tri pina: 5 V , *GND* i analogni izlazni pin. Osetljivost ovog senzora je 10 mV/°C. Merenje se vrši sa preciznošću od ±1 °C, koliko iznosi i greška linearnosti. Jeftin je i lako dostupan, tako da je zbog toga naišao na široku upotrebu [5].



Slika 2. *LM35* senzor [5].

V. LED DIODA

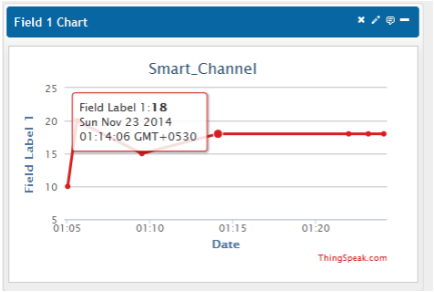
LED (*Light Emitting Diode*) je poluprovodnička komponenta koja emituje svetlost onda kada struja prolazi kroz nju. Ove diode se nalaze u različitim bojama i omogućavaju svetlosnu indikaciju različitih pojava ili stanja sistema. Jednostavne su za upotrebu, sadrže dva pina (jedan *GND* a drugi za povezivanje sa digitalnim pinom *Arduina*). Za ovaj projekat su korišćene tri diode [6].



Slika 3. *LED* indikacija [6].

VI. THINGSPEAK PLATFORMA

Ova internet platforma je zamišljena kao posrednik u komunikaciji između uređaja preko interneta. Razvijena je od strane *MathWorks* tima. Često se koristi u *IoT* (*Internet of Things*) aplikacijama. Upotreba se sastoji u kreiranju kanala u kome može biti više različitih polja. Da bi korisnik mogao da kreira kanale mora biti registrovan sa svojim nalogom (registracija je besplatna). Nakon kreiranja kanala, generiše se *ApiKey* za taj kanal, pomoću koga se, uz redni broj željenog polja na kome se žele slati podaci, vrednosti šalju na grafike. Pristup kanalima može biti privatan ili javan, što se u svakom trenutku može podesiti. Najčešće, podaci koji se šalju na grafike određenih polja potiču sa senzora, kao što je to ovde slučaj. Vremenski interval u kome se podaci primaju može biti podešen (na slici je to 5 sekundi) [7]. Za čuvanje podataka o temperaturi kreiran je kanal „Temperatura" na koji pristižu informacije (na polje sa rednim brojem 2).



Slika 4. Primer jednog polja u *ThingSpeak*-u [7].

VII. PROJEKTNI ZADATAK

Demonstracija korišćenja *Python*-a u programiranju *Arduino* okruženja ogleda se u merenju ambijentalne temperature i prikazivanju izmerene vrednosti na ekranu kao i slanje podataka na *ThingSpeak*. Na ekranu, temperatura se prikazuje u okviru *GUI* (*Graphical User Interface*) prozora, na kome se nalaze i tri dugmeta. Kada se inicijalno pokrene izvršavanje programa, korisnik u okviru *GUI*-ja treba da vidi obaveštenje *SystemOFF*. Klikom na dugme *START* pokreće se izvršavanje programa i korisnik na ekranu vidi vrednosti temperature. Istovremeno, podaci se šalju na ThingSpeak. Status sistema je tada *SystemRUN*, što se prikazuje umesto *SystemOFF*. Kada korisnik želi da pauzira očitavanje temperature, potrebno je da klikne na dugme *STOP*. Tada se čuje kratak zvučni signal i sistem prelazi u stanje *SystemSTOP*. Korisnik može opet da klikne na *START*. Klikom na dugme *WEB*, korisnik biva odveden na stranicu *ThingSpeak*-a, gde, nakon logovanja odgovarajućim mejlom i lozinkom na sajt *MathWorks*-a, može videti grafik sa primljenim temperaturama. Ovde se ogleda još jedna pogodnost *Python*-a: u *PyCharm* editoru lako se mogu praviti forme sa kojima korisnik može da interaguje. Forma je u ovom slučaju jednostavna i laka za korišćenje, ali *GUI* može biti i veoma kompleksan. Ono što korisnik vidi na maketi je žuti svetlosni signal na diodi kada je stanje sistema *SystemOFF*, zeleni svetlosni signal kada je sistem u stanju *SystemRUN* i crveni svetlosni signal kada je sistem u stanju *SystemSTOP*.

VIII. REALIZACIJA KODA

Prva faza u realizaciji ovog jednostavnog sistema je uključivanje neophodnih biblioteka za ugodniji rad sa objektima u *PyCharm*-u. One podržavaju rad sa *Arduino* platformom, internet stranicama, zvučnim signalima, kreiranjem grafičkog okruženja, matematičkim operacijama nad rezultatima itd.

```
import pyfirmata
from pyfirmata import Arduino, util
import serial
import board
import time
import requests
import math
import tkinter as tk
import winsound
import threading
import webbrowser
```

Slika 5. Uključivanje biblioteka u kod.

Zatim sledi postavljanje *COM* porta na kome je *Arduino* povezan na računar i inicijalizacija pločice, kao i definisanje parametara *http* adrese kanala na kome će se podaci slati na *ThingSpeak* (*ApiKey*).

```
board = Arduino('COM9')
iterator = util.Iterator(board)
iterator.start()


url = "https://api.thingspeak.com/update"
api_key = "PXUL62KTL038VO6K"
```

Slika 6. Inicijalizacija pločice i link ka internet platformi.

Ono što je ključno za čitanje/upis pinova je i njihovo definisanje. Za realizaciju zadatka je potreban jedan ulazni analogni pin za čitanje temperature sa senzora kao i tri digitalna izlazna pina za postavljanje svetlosne indikacije na diodama.

```
board.analog[1].mode = pyfirmata.INPUT
board.digital[10].mode = pyfirmata.OUTPUT
board.digital[9].mode = pyfirmata.OUTPUT
board.digital[8].mode = pyfirmata.OUTPUT
```

Slika 7. Inicijalizacija pinova.

Nakon toga, sledi definicija funkcionalnosti dugmića koji će se nalaziti na formi. Potrebno je kreirati funkcije koje će se obavljati nakon pokretanja programa, nakon klika na svako od dugmeta (*START*, *STOP*, *WEB*), funkcije koja će slati podatke na *ThingSpeak* i funkcije koja će se izvršavati pri zatvaranju forme.

```
def send_to_thingspeak(data):
    payload = {'api_key':api_key, 'field2': data}
    response = requests.get(url, params=payload)
    if response.status_code == 200:
        print("Podaci su uspesno poslati")
    else:
        print("Greska")
```

Slika 8. Funkcija za slanje podataka na *ThingSpeak*.

```
def startPROG():
    flag.set(True)
    t = threading.Thread(target=citanje)
    t.start()
    board.digital[8].write(1)
    board.digital[9].write(0)
    board.digital[10].write(0)
```

Slika 9. Funkcija koja se vrši nakon klika na *START*.

```
def citanje():
    while flag.get():
        temp = board.analog[1].read() * 50
        temp = round(temp, 3)
        analogReadLabel.config(text=str(temp), fg='blue')
        statusLabel.config(text="SystemRUN")
        time.sleep(0.1)
        send_to_thingspeak(temp)
```

Slika 10. Funkcija koja vrši očitavanje analognog pina i obradu podataka u temperaturu.

```
def stopPROG():
    flag.set(False)
    statusLabel.config(text="SystemSTOP")
    board.digital[10].write(1)
    board.digital[9].write(0)
    board.digital[8].write(0)
    winsound.Beep(1000, 2000  )
```

Slika 11. Funkcija koja se vrši nakon klika na *STOP*.

```
def on_closing():
    board.digital[8].write(0)
    board.digital[9].write(0)
    stopPROG()
    board.digital[10].write(0)
    win.destroy()
```

Slika 12. Funkcija koja se vrši pri zatvaranju forme.

```
def webPAGE():
    webbrowser.open('https://thingspeak.com/channels')
```

Slika 13. Funkcija koja vodi korisnika na *web* stranicu platforme.

Kreiranje forme se svodi na kreiranje dugmića i labela koje korisnik može da vidi kao i na pridruživanju odgovarajućih funkcionalnosti dugmićima. Biće dati samo primeri kreiranja dugmića.

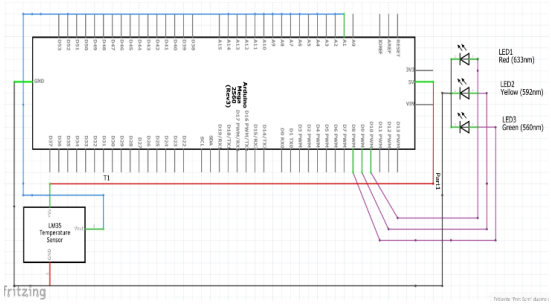
```
startBtn = tk.Button(win, bd=4, bg='#90EE90', text="START", command=startPROG)
startBtn.grid(column=1, row=3)
stopBtn = tk.Button(win, bd=4, bg='#FF6666', text="STOP", command=stopPROG)
stopBtn.grid(column=2, row=3)
stopBtn = tk.Button(win, bd=4, bg='#FF6666', text="STOP", command=stopPROG)
stopBtn.grid(column=3, row=3)
```

Slika 14. Programiranje dugmića.

IX. Fizička realizacija

Povezivanje komponenti je prikazano električnom šemom u *Fritzing*-u, dok će izgled sistema u realnosti biti prikazan kroz rezultate. Po jedan pin svake od dioda (crvene,

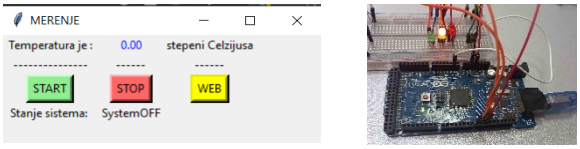
žute i zelene) povezuje se na *GND* mikrokontrolera, dok je drugi pin cvene diode povezan na digitalni pin *D10*, drugi pin žute diode je povezan na digitalni pin *D9* i drugi pin zelene diode je povezan na digitalni pin *D8*. Što se tiče senzora, njegov pin u sredini je povezan na analogni pin *A1* na pločici, dok su preostala dva pina povezana na 5 V i na *GND*.



Slika 15. Povezivanje komponenti.

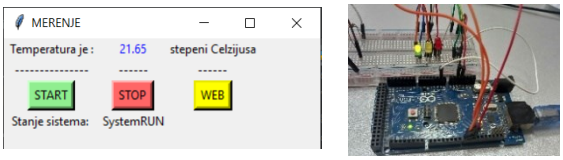
X. Rezultati

Nakon startovanja programa, sistem je zakočen odnosno čeka se da korisnik omogući čitanje temperature.



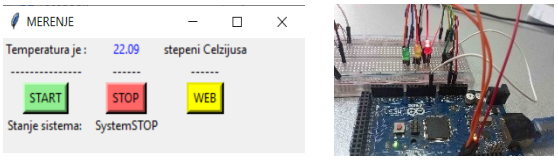
Slika 16. Izgled na ekranu i u realnosti kada sistem nije pokrenut.

Nakon klika na *START*, prikaz je sledeći:



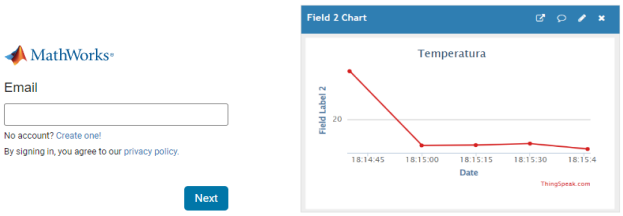
Slika 17. Izgled na ekranu i u realnosti kada je sistem pokrenut.

Nakon klika na *STOP*, prikaz je sledeći:



Slika 18. Izgled na ekranu i u realnosti kada je sistem stopiran.

Nakon klika na dugme *WEB*, korisniku se otvara pretraživač.



Slika 19. Logovanje na platformu i prikaz temperature na grafiku.

XI. Zaključak

Python je izuzetno pogodan za rad sa mikrokontrolerima kao što je *Arduino* i *Raspberry-Pi* mikroračunarima. Nadogradnja u pogledu korišćenja *Arduina* i *Python*-a u odnosu na ovaj demonstrativni primer se ogleda u kreiranju baza podataka koji dolaze sa senzora (kao što je u ovom slučaju temperaturni, a može biti i senzor vlažnosti, udaljenosti itd.) i na osnovu toga vršiti predikciju ponašanja sistema nekim od modela mašinskog učenja. Takođe, kod koji se piše u *Python*-u je intuitivan i logički sličan kodu iz *Arduino IDE* okruženja, tako da i oni koji nisu koristili *Python* mogu sa lakoćom početi programiranje mikrokontrolera u ovom programskom jeziku.

Zahvalnica

Velika zahvalnost za sve vidove podrške u toku pisanja ovog rada članovima Laboratorije za robotiku Elektronskog fakulteta u Nišu kao i profesorima Bobanu Veseliću, Jeleni Jovanović i Danijelu Dankoviću.

Literatura

[1] *Python* - https://en.wikipedia.org/wiki/Python_(programming_language)
[2] *PyCharm* - https://hackr.io/blog/what-is-pycharm
[3] *pyfirmata* - https://roboticsbackend.com/control-arduino-with-python-and-pyfirmata-from-raspberry-pi/
[4] *Arduino Mega 2560* - https://www.mouser.com/catalog/specsheets/ArduinoBoardMega2560.pdf
[5] *LM35* - https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor
[6] *LED* - https://en.wikipedia.org/wiki/Light-emitting_diode
[7] *ThingSpeak* - https://web.iitd.ac.in/~ddz208073/images/ts.pdf

Primena mašinskog učenja za detekciju i klasifikaciju anemije

Emilija Ćojbašić

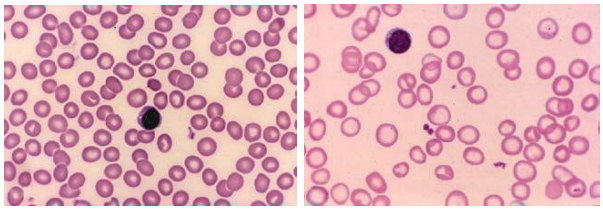
Sadržaj – Poslednjih godina, primena veštačke inteligencije kao mehanizma za podršku u postavljanju dijagnoza različitih bolesti postala je široko rasprostranjena. S obzirom da je anemija najrasprostranjenija bolest krvi na svetu, koja vrlo brzo ispoljava simptome i utiče na kvalitet svakodnevnog života obolelih, rana i korektna dijagnoza tipa anemije je od suštinskog značaja za pravilno lečenje. U ovom radu razvijen je model na bazi veštačke neuronske mreže koji vrši detekciju i klasifikaciju anemije na osnovu parametara krvne slike i biohemije, što može poslužiti kao značajna pomoć medicinskim stručnjacima za postavljanje brze, konzistentne i objektivne dijagnoze, uz uštedu vremena i resursa. Za treniranje i evaluaciju neuro mreže iskorišćeni su podaci iz publikovanog istraživanja, za koje su klinički podaci o preko 15000 pacijenata javno dostupni na sajtu *Kaggle*. Razvijeni model pokazao je tačnost klasifikacije u 96,9% slučajeva, a dobijena tačnost je upoređena sa drugim modelima iz domena veštačke inteligencije publikovanim u pomenutom istraživanju. Dobijeni rezultati potvrđuju primenljivost razvijenog i sličnih modela na bazi mašinskog učenja u kliničkoj praksi.

I. Uvod

Prema definiciji Svetske Zdravstvene Organizacije (SZO), *anemija* je stanje koje se manifestuje smanjenjem broja crvenih krvnih zrnaca ili smanjenjem koncentracije hemoglobina u krvi [1]. Hemoglobin je protein koji igra ključnu ulogu u vezivanju kiseonika i njegovog prenošenja putem krvotoka do svih organa u telu kako bi se omogućilo njihovo normalno funkcionisanje. Anemija može izazvati različite simptome poput umora, slabosti, vrtoglavice i kratkog daha, što utiče na kvalitet života [2].

Anemija je veoma često oboljenje krvi koja dovodi do ozbiljnih zdravstvenih problema. Ako se ne prepozna i ne leči na vreme, može dovesti do značajnih posledica, uključujući teški umor, srčane probleme i potencijalno životno ugrožavajuće komplikacije. Upravo iz ovog razloga je od izuzetnog značaja prepoznavanje ranih simptoma anemije, pravovremeno i tačno postavljanje dijagnoze i adekvatno određivanje tipa anemije.

Razlozi nastanka anemije su mnogobrojni i mogu biti nutritivni i nenutritivni. Nutritivna anemija [3] javlja se kao rezultat nedostatka nutritivnih sastojaka ključnih za proizvodnju crvenih krvnih zrnaca ili hemoglobina. Ukoliko je detektovana vrednost hemoglobina <12 g/dL kod muškaraca, odnosno <11 g/dL kod žena, vrednost gvožđa <20 μg/L, feritina <30 ng/mL, vitamina B12 <200 pg/mL i folata <4 ng/nL, može se potvrditi prisustvo *nutritivne anemije*. Generalno, nutritivna anemija se može podeliti u nekoliko osnovnih kategorija, kao što su: (a) *anemija usled deficita gvožđa*, (b) *anemija usled deficita folata* i (c) *anemija usled deficita vitamina B12.* S druge strane, najčešći faktori nastanka nenutritivne anemije su krvarenje, infekcija, hronične bolesti, infiltracija koštane srži, itd.



Slika 1. Periferni razmaz zdrave osobe (levo) i osobe sa anemijom usled deficita gvožđa (desno) [2].

U cilju detektovanja anemije, potrebno je uraditi hemogram – laboratorijski test koji se koristi za analizu sastava krvi i ocenu različitih komponenti krvi. Neke od ključnih parametara koji se analiziraju hemogramom jesu, između ostalog, informacije o hemoglobinu (HGB), broju crvenih krvnih zrnaca (RBC), srednjem volumenu eritrocita (MCV), srednjoj vrednosti koncentracije hemoglobina eritrocita (MCHC), srednjoj vrednosti količine hemoglobina eritrocita (MCH), širini raspodele crvenih krvnih zrnaca (RDW) i drugi.

Brojne studije posvećene detekciji i klasifikaciji anemije su objavljene širom sveta [4,5], istražujući različite metode i tehnologije za precizno dijagnostifikovanje ovog zdravstvenog stanja. Ovi istraživački napori doprineli su unapređenju medicinske dijagnostike, što ima potencijal da poboljša zdravlje i celokupan kvalitet života pacijenata. Na osnovu tih studija, razvijeni su brojni modeli za detekciju i klasifikaciju kliničke anemije.

Veštačka inteligencija i mašinsko učenje su danas već široko prisutni u dijagnostici i tretmanu različitih oboljenja. Veštačka inteligencija (engl. *Artificial intelligence*, skraćeno AI) [6] 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) [6], 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.

E. Ćojbašić je student prve godine master studija Računarstva i informatike na Elektronskom fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija i studentski član IEEE, E-mail: ecojbasic@elfak.rs

Sprovedeno istraživanje opisano u radu imalo je za cilj da se razvije model mašinskog učenja koji koristi široki spektar inicijalnih laboratorijskih nalaza pacijenta kako bi se povećala stopa tačnosti detekcije i klasifikacije anemije. U radu je prikazano poređenje razvijenog modela tipa veštačke neuronske mreže (ANN) sa modelima zasnovanim na drugim tehnikama iz domena veštačke inteligencije predloženim u publikovanom istraživanju [7], iz koga potiču i klinički podaci o pacijentima korišćeni u ovom radu.

II. METODE

A. *Pacijenti i klinički podaci*

U ovom radu korišćeni su podaci o pacijentima koji su prikupljeni u studiji sprovedenoj na Medicinskom fakultetu Tokat Gaziosmanpaşa univerziteta u Turskoj tokom petogodišnjeg perioda [7], a preuzeti su iz javno dostupne baze sa sajta *Kaggle*. Klinički podaci uključeni u analizu sadržali su 24 parametra krvne slike i biohemije pacijenata. Nazivi parametara, njihovi opisi, jedinice, kao i opseg referentnih vrednosti prikazani su u tabeli 1.

TABELA I
PARAMETRI IZ SKUPA PODATAKA [7]

Parametar	Kratak opis	Jedinica	Ref. vred.
B12	B12	ng/mL	200-505
BA	Bazofili	$10^3/\mu L$	0.01-0.07
EO	Eozinofili	$10^3/\mu L$	0.03-0.59
FERRITE	Gvožđe	ng/mL	30-400
FOLATE	Folati	ng/mL	4-175
GENDER	Ženski/Muški	0-1	
HCT	Hematokrit	%	35-45
HGB	Hemoglobin	g/dL	13.5-16.9
LY	Limfociti	$10^3/\mu L$	1.26-3.31
MCH	Širina raspodele eritrocita	pg	27-32.3
MCHC	Srednja vrednost količine hemoglobina eritrocita	g/dL	32.35
MCV	Srednja vrednost koncentracije hemo-globina eritrocita	fL	81.8-95.5
MO	Monociti	$10^3/\mu L$	0.29-0.95
MPV	Srednji volumen trombocita	fL	9.3-12.1
NE	Neutrofili	$10^3/\mu L$	1.8-6.98
PCT	Trombokrit	K/uL	0.17-0.32
PDW	Širina raspodele trombocita	fL	10.1-16.1
PLT	Trombociti	K/uL	166-308
RBC	Eritrociti	Milyon/	4.44-5.61
RDW	Širina distribucije eritrocita	%	12-13.6
SD	Serumsko gvožđe	µg/dL	20-50
SDTSD	(SD/TSD) * 100	µg/dL	20-50
TSD	Totalno serumsko gvožđe	µg/dL	250-450
WBC	Leukociti	µg/dL	3.91-10.9

Iz inicijalnog skupa podataka najpre su otklonjeni podaci o pacijentima sa nedostajućim parametrima krvne slike, kao i podaci o trudnicama, deci i pacijentima koji boluju od karcinoma. U razmatranje su konačno uzeti parametri krvne slike i biohemije za ukupno 15300 pacijenata, od čega je 10379 (68%) žena i 4921 (32%) muškaraca, što potvrđuje konstataciju da je anemija bolest koja uglavnom pogađa ženski deo populacije. Skup podataka sadrži ukupno 5553 (36%) pacijenata koji imaju potvrđenu dijagnozu da boluju od nekog tipa anemije, i to: 1019 (7%) pacijenata sa nenutritivnom anemijom, 4182 (27%) sa anemijom usled deficita gvožđa, 153 (1%) koji boluju od anemije usled deficita folata i 199 (1%) obolelih od anemije usled deficita vitamina B12. Preostalih 9747 (64%) pacijenata ne boluju od anemije. Ova statistika prikazana je u tabeli 2.

TABELA II
KLASE ANEMIJE U SKUPU PODATAKA O PACIJENTIMA [7]

Klasa anemije	Pacijenata	Udeo [%]
Nema anemije	9747	63,70
Nenutritivna anemija	1019	6,66
Anemija usled deficita gvožđa	4182	27,33
Anemija usled deficita folata	153	1,00
Anemija usled deficita B12	199	1,31

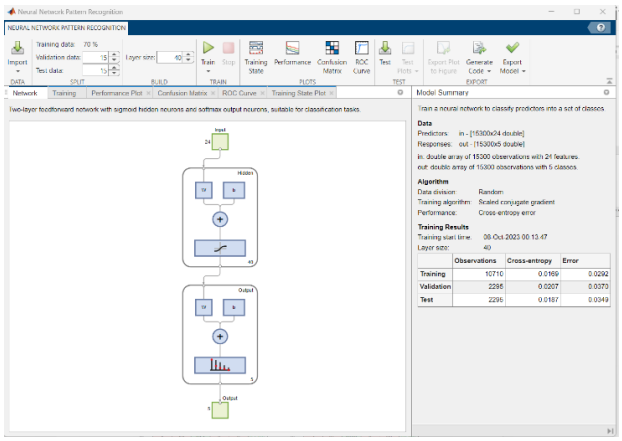
B. *ANN model*

Model za detekciju i klasifikaciju anemije predložen u ovom radu je neuronska mreža sa prostiranjem signala unapred (feedforward), sa sigmoidnom aktivacionom funkcijom [6] kod neurona u skrivenom sloju i softmax aktivacionom funkcijom kod neurona u izlaznom sloju, prikladna za obavljanje zadataka klasifikacije.

Projektovana neuronska mreža je po svojoj strukturi višeslojni perceptron sa tri sloja: ulaznim slojem, skrivenim slojem i izlaznim slojem. U ovom slučaju, neuronska mreža ima 24 ulaza i 5 izlaza, dok je u skrivenom sloju usvojeno 40 neurona nakon testiranja više različitih arhitektura mreže, sa različitim brojem slojeva i neurona u njima. Ulazni parametri obuhvataju različite aspekte laboratorije, uključujući broj crvenih krvnih zrnaca, nivo hemoglobina i hematokrita, parametre biohemije kao što su nivo gvožđa, folata i vitamina B12, kao i ostale pokazatelje navedene u tabeli 1. Veliki broj ulaza je tipična pojava kod neuronskih mreža, jer su one pogodne za obradu velikih količina podataka koje efikasno koriste kako bi preciznije mogle obraditi, naučiti i modelirati složene obrasce i zavisnosti u podacima, što kasnije poboljšava njihovu sposobnost za precizno diferenciranje različitih klasa. Izlazi neuronske mreže, koji su predstavljeni sa 5 neurona u izlaznom sloju, odnose se na dijagnozu anemije i istovremenu klasifikaciju tipa anemije. Veštačka neuronska mreža klasifikuje svakog pacijenta reprezentovanim podacima na ulazu u jednu od pet klasa - u klasu pacijenata kod kojih nije prisutna anemija, kod kojih je prisutna nenutritivna anemija, ili kod

kojih je prisutna anemija usled deficita gvožđa, anemija usled deficita folata ili anemija usled deficita vitamina B12.

ANN model je projektovan i obučavan Matlab softverskim paketom (The MathWorks Inc., USA), verzija 2023a, i njegovim dodatkom za primenu mašinskog učenja Deep Learning Toolbox (ranije Neural Network Toolbox). Struktura opisanog ANN modela koji je predložen u ovom radu kao i grafički interfejs softverskog alata za projektovanje i obučavanje veštačkih neuronskih mreža za probleme klasifikacije (nprtool) sa osnovnim parametrima modela prikazani su na slici 2.



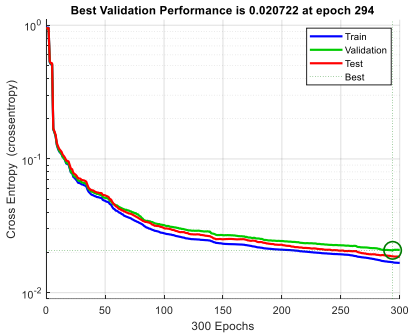
Slika 2. Razvijeni ANN model sa 24 ulaza, 5 izlaza i skrivenim slojem sa 40 neurona.

Raspoloživi podaci o ukupno 15300 pacijenata su nasumično podeljeni na skupove za treniranje, validaciju i testiranje veštačke neuronske mreže. Skup za treniranje, sastavljen od 70% (10710 pacijenata), korišćen je za obučavanje modela kako bi nakon treniranja reprezentovao odnose između ulaznih podataka i ciljnih izlaza neuro mreže. Skup za validaciju, sačinjen od 15% podataka (2295 pacijenata), korišćen je tokom treniranja neuronske mreže kako bi se ocenila sposobnost generalizacije mreže i zaustavio proces treniranja kada generalizacija prestane da se poboljšava. Na kraju, skup za testiranje (15%, 2295 pacijenata) nije imao uticaja na proces obučavanja neuronske mreže, već je služio kao nezavisna mera performansi modela nakon završetka obučavanja.

III. Rezultati

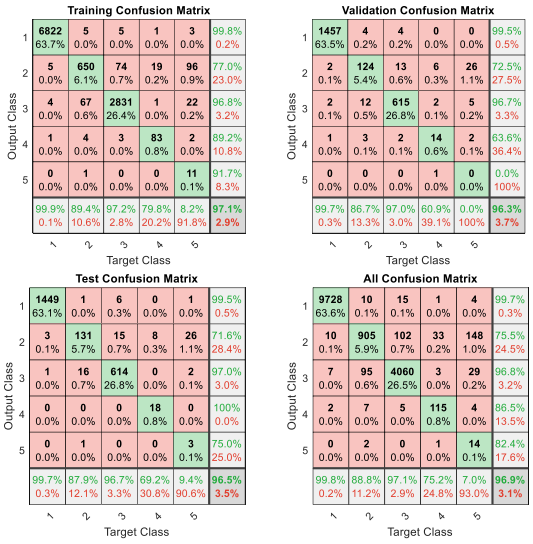
Rezultati procesa obučavanja veštačke neuronske mreže i pad greške tokom obučavanja na skupovima za obučavanje, validaciju i testiranje prikazani su na slici 3. Obučavanje je trajalo 294 epohe obučavanja.

Obučavanje mreže je vršeno referentnim algoritmom za obučavanje veštačkih neuronskih mreža tipa višeslojni perceptron dostupnim u softveru Matlab, iz klase gradijentnih metoda (tzv. *Scaled conjugate gradient backpropagation metod*). Postepeno opadanje greške tokom epoha ukazuje na činjenicu da se tačnost modela poboljšava, odnosno da model postaje sve bolji u minimiziranju greške i prilagođavanju podacima. Obučavanje je prekinuto kada greška na validacionom skupu podataka počinje da raste, čime se sprečava preterano treniranje mreže.



Slika 3. Proces obučavanja veštačke neuronske mreže.

ANN model najveće tačnosti razvijen u ovom radu je tačno klasifikovao 96,9% pacijenata posmatrano na ukupnom skupu podataka. U nezavisnom uzorku za testiranje, veštačka neuronska mreža je tačno klasifikovala pacijente u 96,5% slučajeva. Matrice konfuzije i ROC krive modela neuronske mreže za trening skup, validacioni skup, test skup i ukupni skup podataka prikazani su respektivno na slikama 4 i 5. Mreža koja iskazuje dobijenu tačnost definisana je navedenom strukturom i optimalnim skupom hiperparametara, koji zbog obima nije prikazan u radu, može se relativno lako replicirati opisanom metodologijom.

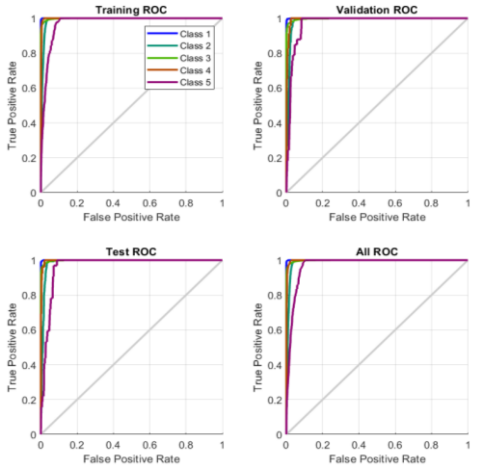


Slika 4. Matrice konfuzije za razvijeni ANN model.

IV. Diskusija

U publikovanoj studiji [7], primenjen je isti opisani skup podataka, pri čemu je razmatrano i upoređeno nekoliko modela iz domena veštačke inteligencije. Rezultati su uključivali pored drugih i primenu metode

potpornih vektora [6] (eng. *support vector machine*, SVM), metodu k najbližih suseda [6] (eng. *k-nearest neighbourghood*, k-NN) i model dubokog učenja - model zasnovan na konvolucionim neuronskim mrežama [6] (engl. *Convolutional Neural Network*, CNN). Kao referentan model u smislu tačnosti predložen je hibridni GA-CNN model kod koga je CNN mreža dodatno optimizovana primenom genetskog algoritma (eng. genetic algorithm, GA). U cilju obučavanja i validacije navedenih modela, opisani skup kliničkih podataka podeljen je u radu [7] na nezavisne podskupove za treniranje (70%) i testiranje (30%). Uporedna ukupna tačnost modela iz istraživanja [7] prikazana je u Tabeli 3, a dostignuta tačnost GA-CNN modela koji je predložen kao najbolji je 98,50%. Sva četiri modela generalno su pokazala dobre rezultate i visoku tačnost klasifikacije, što je potvrdilo njihovu primenljivost u kliničkoj praksi.



Slika 5. ROC krive za razvijeni ANN model.

TABELA III
TAČNOST KLASIFIKACIJE ZA RAZNE MODELE

Model	**Tačnost (%)**
SVM model iz [7]	92,60
k-NN model iz [7]	84,74
CNN model iz [7]	97,12
GA-CNN model iz [7]	98,50
ANN model (ova studija)	96,90

Po dobijenim rezultatima i tačnosti, performanse modela veštačke neuronske mreže koji je razvijen u ovom radu, uporedive su sa performansama konvolucionih neuronskih mreže iz [7] (Tabela 3). Iako je tačnost modela zasnovanog na mašinskom učenju razvijenog u ovom radu neznatno manja od najboljih CNN modela iz [7], može se konstatovati da i ovde predloženi model efikasno koristi informacije koje su sadržane u kliničkim podacima i vrši veoma pouzdanu detekciju anemije kod pacijenta i klasifikaciju u jednu od četiri posmatrana tipa ovog oboljenja. Na osnovu dobijenih rezultata, može se zaključiti da i predloženi ANN model ima potencijal za praktičnu primenu u dijagnostici i klasifikaciji anemije kod pacijenata.

Iako neznatno zaostaje u pogledu tačnosti, predloženi ANN model ima i neke prednosti u odnosu na CNN modele dubokog učenja razvijene u radu [7]. To je pre svega njegova jednostavnost u odnosu na alternativne modele dubokog učenja. Takođe, razvijeni model ima superioran algoritam obučavanja, koji je računarski veoma efikasan, posebno u odnosu na model dubokog učenja koji se optimizuje genetskim algoritmom (GA-CNN). Efikasna softverska implementacija u paketu Matlab koja prilikom obučavanja uzima u obzir i rezultate na verifikacionom skupu kako bi sprečila preterano treniranje mreže (eng. overfitting) zaslužna je za visoku tačnost modela. Razvijena neuronska mreža se zbog jednostavnosti može relativno lako implementirati i drugim softverskim alatima za razvoj neuro mreža, a kao obučeni model se efikasno može i implementirati kao izvršni kod za neposrednu primenu u kliničkoj praksi, prilagođen raznovrsnim u svakodnevnoj praksi dostupnim računarskim i mobilnim uređajima.

V. ZAKLJUČAK

Može se zaključiti da su rezultati dobijeni u ovom radu uz rezultate iz [7] i rezultate iz drugih srodnih studija [8,9] potvrdili primenljivost veštačke inteligencije kod problema dijagnoze i klasifikacije anemije. Korišćenje modela zasnovanih na mašinskom učenju se poslednjih godina sve više popularizuje, pre svega zbog izuzetne sposobnosti da efikasno analiziraju velike količine medicinskih podataka, kao i zbog sposobnosti da prepoznaju suptilne obrasce i složene odnose između različitih faktora povezanih sa anemijom. Ovo ne samo da olakšava medicinskom osoblju preciznu dijagnozu, već i obećava unapređenje pristupa medicinskom odlučivanju i bolji tretman za pacijente koji se suočavaju sa ovim zdravstvenim problemom.

Konačno, kao mogući pravac daljeg istraživanja mogu se razmatrati hibridni neuro-fazi modeli [10] koji bi omogućili da se pored obučavanja korišćenjem kliničkih podataka o pacijentima model mašinskog učenja dogradi i na osnovu dostupnog ekspertskog znanja medicinskih stru;njaka o anemijama. Takođe, moguće je razmatrati savremene tehnike objašnjive veštačke inteligencije [11].

LITERATURA

[1] WHO, C.M., 2011. Haemoglobin concentrations for the diagnosis of anaemia and assessment of severity. *Geneva, Switz World Heal Organ*, pp.1-6.
[2] Kaushansky, K., 2016. *Williams hematology*. McGraw-Hill.
[3] Bhadra, P. and Deb, A., 2020. A review on nutritional anemia. *Indian J. of Natural Sci.*, 10(59), pp.18466-18474.
[4] Saxena R, Chamoli S, Batra M. Clinical Evaluation of Different Types of Anemia. World J Anemia, 2018;2(1):26-30.
[5] Vohra R, Hussain A, Dudyala AK, Pahareeya J, Khan W (2022) Multi-class classification algorithms for the diagnosis

of anemia in an outpatient clinical setting. PLoS ONE 17(7): e0269685. https://doi.org/10.1371/journal.pone.0269685.

[6] Janičić, P., Nikolić, M., 2023, *Veštačka inteligencija - treće izdanje*, Matematički fakultet Univerziteta u Beogradu.

[7] Kilicarslan, S., Celik, M. and Sahin, Ş., 2021. Hybrid models based on genetic algorithm and deep learning algorithms for nutritional Anemia disease classification. *Biomedical Signal Processing and Control*, 63,p.102231.

[8] Berkol, A., Tartan, E. and Ekici, Y., 2020. Anemia diagnosis by using artificial neural networks. *Int. Journal of Multidisciplinary Studies and Innovative Technologies*, 4(1), pp.14-17

[9] Yıldız, T.K., Yurtay, N. and Öneç, B., 2021. Classifying anemia types using artificial learning methods. *Engineering Science and Technology Int. Journal*, 24(1), pp.50-70.

[10] Ćojbašić, E., 2022, *Machine learning as an aid to predicting clinical outcome after stroke*. Proc. of 2022 IEEE 21st Mediterr. Electrotechnical Conf. (MELECON), pp. 24 27.

[11] Saranya, A. and Subhashini, R., 2023. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. *Decision analytics journal*, p.100230.

Bluetooth interfejs za FM radio na bazi Arduina

Milica Milosavljević, Jovan Jordanov

Sadržaj - U ovom projektu se bavimo projektovanjem i realizacijom hardverskog i softverskog interfejsa za *bluetooth* uređaj koji bi kontrolisao FM radio na bazi *Arduina*.

I. Uvod

Zadatak ovog projekta je projektovanje *Bluetooth* interfejsa za FM radio. Cilj projekta jeste hardversko povezivanje modula sa *Arduino* razvojnim sistemom, i realizacija softvera koji omogućava komunikaciju pametnog telefona sa *Arduino* razvojnim sistemom preko *Bluetooth*-a.

Bluetooth je tehnologija pomoću koje se vrši bežični prenos podataka između uređaja koji poseduju istu tehnologiju. Ova tehnologija omogućava komunikaciju između PDA (*personal digital assistent*) uređaja, mobilnih telefona, lap-top računara, PC računara, digitalnih foto aparata i kamera itd. *Bluetooth* uređaj emituje ultra-ljubičaste zrake do drugog *Bluetooth* uređaja i na taj način se odvija komunikacija. *Bluetooth* je bežični standard pre svega namenjen nisikopotrošnim uređajima, sa kratkim dometom (od 10 cm do 100 m) i koji u sebi obavezno sadrže primopredajnik. *Bluetooth* omogućava komunikaciju ovih uređaja, ako su u dometu (maksimum je 100 m), čak iako nisu u istoj prostoriji. *Bluetooth* radi na frekvencijama između 2,402 i 2,480 GHz, ili 2,400 i 2,4835 GHz, uključujući zaštitne opsege širine 2 MHz na donjem kraju i 3,5 MHz na vrhu. *Bluetooth* koristi radio tehnologiju koja se zove frekventno skakanje proširenog spektra. *Bluetooth* deli prenete podatke u pakete i svaki paket prenosi na jedan od 79 određenih *Bluetooth* kanala.

Svaki kanal ima propusni opseg od 1 MHz. Obično izvodi 1600 skokova u sekundi, sa omogućenim adaptivnim skokom frekvencije (AFH). *Bluetooth* LE (*Low Energy*) koristi razmak od 2 MHz, koji može da primi 40 kanala. Danas su u upotrebi više verzija *Bluetooth*-a, od verzije 1.0, pa sve do najnovije verzije 5.3. Naš modul koristi *Bluetooth* verzije 3.0.

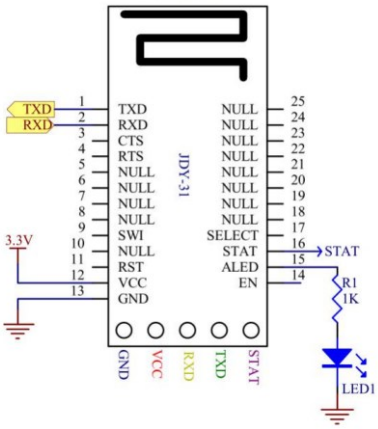
II. Komponente prototipa

Prototip sadrži sledeće komponente: *bluetooth* modul JDY-31, dvostruki prekidač, pomerač nivoa, *Arduino* razvojni system, 5x otpornik od 5.1 kΩ i 2x NPN tranzistor.

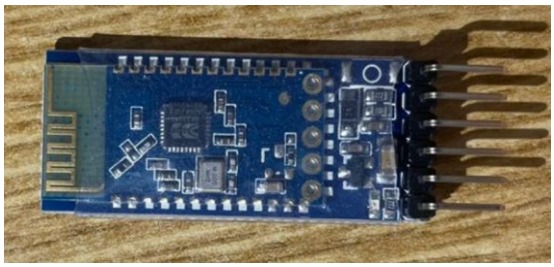
J. Jordanov i M. Milosavljević su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija. E-mail: milicamilo@elfak.rs, jovan.jordanov@elfak.rs

A. Bluetooth modul JDY-31

Za realizaciju Bluetooth interfejsa smo koristili modul JDY-31, koji ima sledeće karakteristike [1]: frekvencija rada: 2.4GHz, komunikacija: UART, napon napajanja: 1,8-3,6 V (preporučeno 3,3 V), pokrivenost: 30 metara, dimenzije modula: 19,6 x 14,94 x 1,8 mm (dužina, visina, debljina), potrošnja struje u stanju mirovanja: 4.7 mA, potrošnja struje nakon povezivanja sa nekim uređajem: 7.3 mA, prenosna snaga: 8 dB, osetljivost na prijemu: -97 dBm, maksimalni prenos podataka: 16 kbita/s (*android*, *windows*).



Slika 1. Šema Bluetooth modula JDY-31 [1].



Slika 2. Gornja strana Bluetooth modula JDY-31.



Slika 3. Donja strana Bluetooth modula JDY-31.

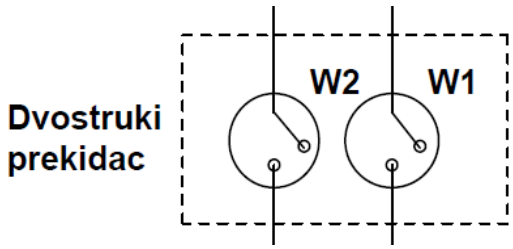
Jedan od problema ovog modula je ta što nije moguće povezivanje sa IOS uređajima.

B. Bluetooth komunikacija

Ovaj *Bluetooth* modul dolazi programiran sa firmverom koji mu omogućava da prenosi izlaz serijskog komunikacionog porta (podrazumevano je 9600 *baud*) preko Bluetooth bežične veze bez konfiguracije softvera. Modul je montiran na zgodnu međuspratu štampanu ploču koja pruža interfejs za napajanje i Tx/Rx serijsku komunikaciju [2].

Dodatno, modul, korišćenjem tekstualnih komandi tipa AT izdatih preko serijskog interfejsa, može biti rekonfigurisan iz svog podrazumevanog *Bluetooth slave* stanja u *Master* stanje. To znači da modul ne može samo da komunicira sa *Bluetooth* glavnim uređajima kao što su računari i pametni telefoni, već i kada je u glavnom režimu može automatski da se upari sa drugim modulom u svojoj podrazumevanoj podređenoj konfiguraciji, omogućavajući serijsku komunikaciju od kraja do kraja preko *Bluetooth-a*.

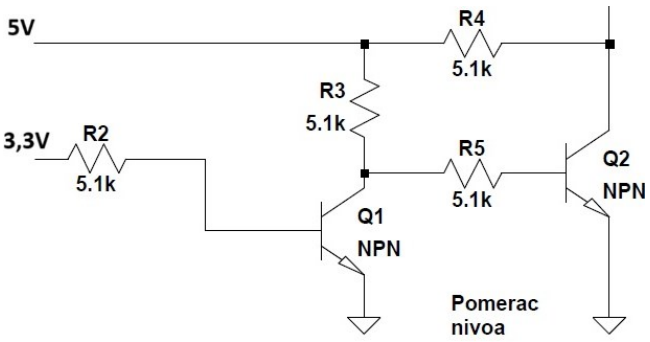
C. Dvostruki prekidač



Slika 4. Šematski prikaz dvostrukog prekidača.

Zbog optimalnog rada *Arduino* razvojnog sistema, dvostrukim prekidačem smo omogućili fizičko odvajanje *Bluetooth* modula od *Arduina*, prekidač W1 prekida RX liniju prema *Arduino* razvojnim sistemom, dok prekidač W2 prekida TX liniju. Ovim smo onemogućili mešanje signala koji dolaze sa računara i sa *Bluetooth* modula.

D. Pomerač nivoa



Slika 5. Šematski prikaz pomerača napona [3].

U elektronici, pomerač nivoa, takođe nazvan pomerač logičkog nivoa ili translator nivoa napona, je kolo koje se koristi za prevođenje signala sa jednog logičkog nivoa ili domena napona u drugi, omogućavajući kompatibilnost između integrisanih kola sa različitim zahtevima napona, kao što su TTL i CMOS.

Moderni sistemi koriste pomerače nivoa da premoste domene između procesora, senzora i drugih kola. Poslednjih godina, tri najčešća logička nivoa su bila 1,8 V, 3,3 V i 5 V, mada se koriste i nivoi iznad i ispod ovih napona.

Pošto se menjači nivoa koriste za rešavanje nekompatibilnosti napona između različitih delova sistema, oni takođe imaju širok spektar primena. Menjači nivoa se široko koriste u povezivanju starih uređaja, kao i u SD karticama, SIM karticama, CF karticama, audio kodecima i UART-ovima.

Arduino razvojni sistem prima i šalje signale koji su na nivou od 5 V, dok bluetooth modul JDY-31 radi sa nivoima signala koji su 3,3 V.

Upotrebom pomerača napona smo omogućili da *Arduino* primi signal od *Bluetooth* modula, i time ostvarimo željenu komunikaciju.

III. PRAKTIČNA REALIZACIJA

Za praktičnu realizaciju ovog projekat prvenstveno smo analizirali *data sheet Bluetooth* modula JDY-31 kako bi ispunili sve parametre i načine povezivanja koje ovaj modul zahteva. Nakon toga smo iscrtalji detalju šemu povezivanja i na osnovu nje smo projektovali vezu na samoj ploči.

U toku samog rada smo nailazili na neke sitnije probleme koje nismo mogli unapred da pretpostavimo samo sa analiziranja *data sheet*-a.

Prvi problem koji se javio u toku testiranja jeste razlika nivoa napona na prijemnoj i predajnoj strani između *Arduina* i samog modula, i to smo rešili upotrebom pomerača napona.

Drugi problem koji se javio jeste problem prilikom programiranja samog *Arduino* razvojnog sistema. Kada smo želeli da "spustimo" naš kod na *Arduino* ploču imali smo problema jer u tom trenutku i *bluetooth* modul i računar koriste iste pinove, i dolazi do mešanja tih signala, što onemogućava upisivanje programa u ROM *Arduina*. Ovaj problem smo rešili ubacivanjem dvostrukog prekidača, koji nam je fizički odvojio modul od *Arduina* i time uz pomoć prekidača imamo tzv. "*programing mode*".

Kod koji smo upisali u *Arduino* razvojni system nam omogućava da sa modula primimo poruke na mikrokontroler na *Arduino* okruženju koje šaljemo sa pametnog uređaja koji je na modul povezan, i na taj način te podatke obrađujemo po našem algoritmu (npr. ukoliko primimo karakter „+" to znači da želimo da pojačamo zvuk).

Sa druge strane naš kod nam omogućava slanje informacija preko serijskog terminal ana sam modul,

odnosno na uređaj koji je *Bluetooth* vezom povezan na modul. Time smo omogućili komunikaciju u oba smera.

```
void setup() {

  mySerial.begin(9600);
  pinMode(13, OUTPUT);
  sendCommand("AT");
  sendCommand("AT");
  sendCommand("AT+ROLE0");
  sendCommand("AT+UUID0xFFE0");
  sendCommand("AT+CHAR0xFFE1");
  sendCommand("AT+NAMEFMRADIO");
  sendCommand("AT+VERSION");

}
```

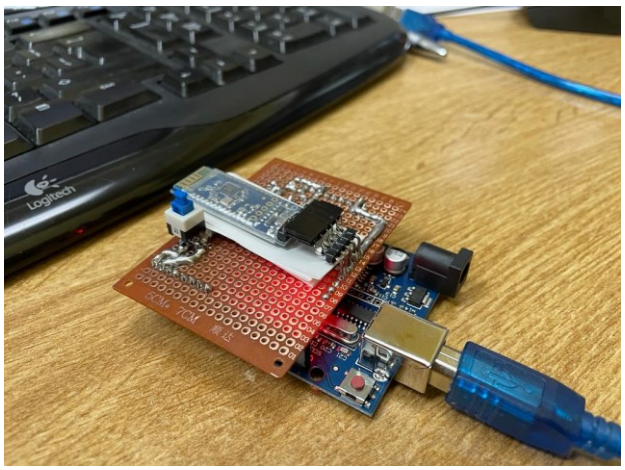
Slika 6. Šematski prikaz pomerača napona [3].

```
void sendCommand(const char * command){
  //Serial.print("Command send :");
 // Serial.println(command);
  Serial.print("\r\n");
  delay(50);
  Serial.print(command);
  Serial.print("\r\n");
  //wait some time
  delay(50);

  char reply[255];
  int i = 0;
  for (int j = 0; j<15; ++j){
    while (Serial.available()) {
      reply[i++] = Serial.read();
    }
    delay(20);
  }
  //end the string
  reply[i] = '\0';
  //Serial.print(reply);
  Serial.println("[");
  Serial.print(reply);
  Serial.println("]");
  if (i>0) {
    Serial.println("Reply successful");
    delay(100);
  }
}
```

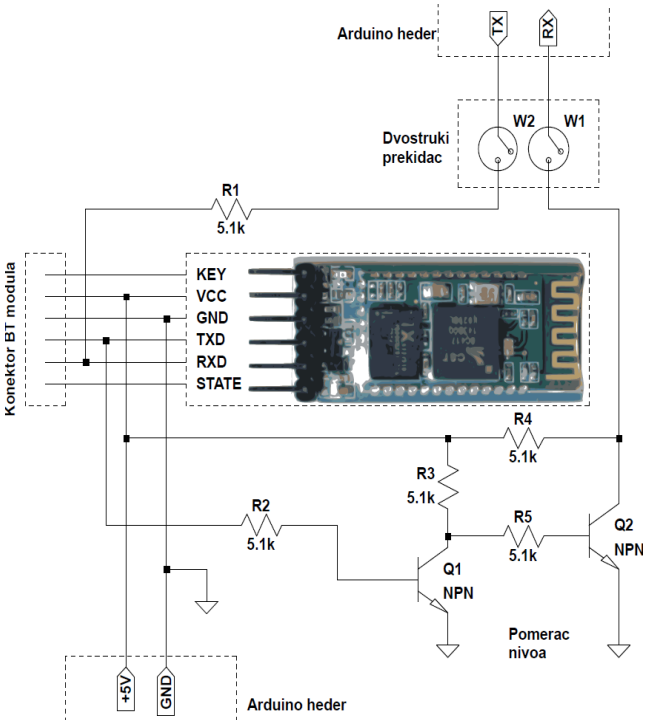
Slika 7. sendCommand funkcija.

Konačan izgled modula sa povezanim komponentama prikazan je na slici 8.



Slika 8. Prikaz modula ugrađenog na *Arduino*.

Blok šema kompletnog projekta prikazana je blokovski na slici 9.



Slika 9. Blok šema kompletnog projekta.

IV. ZAKLJUČAK

Projektovani *bluetooth* interfejs je uspešno realizovan i implementiran u projekat FM Radia. Ovim interfejsom je omogućena kontrola FM Radia sa daljine uz pomoć bilo kog pametnog uređaja koji poseduje bluetooth. Unutar aplikacije je moguće upravljati jačinom zvuka, menjati radio stanice, itd.

LITERATURA

[1] Tehnička specifikacija za Bluetooth modul JDY-31. Dostupno na:http://myosuploads3.banggood.com/products/20190129/20190129043725SKUA87502.pdf

[2] Opis Bluetooth komunikacije. Dostupno na: https://ardubotics.eu/en/communication-control/1796-hc-05-masterslave-bluetooth-module.html

[3] Opis pomerača napona. Dostupno na: https://en.wikipedia.org/wiki/Level_shifter

Manuelni ulazni interfejs FM radija na bazi Arduina

Aleksandar Milosavljević, Veljko Panić, Mihajlo Radonjić

Sadržaj - U ovom radu se bavimo projektovanjem i realizacijom hardverskog i softverskog interfejsa za manuelni ulazni interfejs koji bi kontrolisao FM radio na bazi Arduina. Cilj projekta jeste hardversko povezivanje modula sa Arduino razvojnim sistemom, i realizacija softvera koji omogućava manuelnu kontrolu FM radija. Za realizaciju ulaznog interfejsa pored Arduina Uno, korišćeni su prioritetni i rotacioni koder, kao i tasteri i odgovarajući otpornici. Na kraju realizacije, dobijen je interfejs, pomoću kog je moguće fizički kontrolisati FM radio i ručno upravljati frekvencijama, a samim tim i radio stanicama.

I. UVOD

Radio je naziv za bežični prenos i detekciju komunikacionih signala elektromagnetnih talasa čije su frekvencije niže od frekvencije vidljive svetlosti. FM radio koristi modulaciju frekvencije u opsegu je od 87.5 MHz do 108.0 MHz, što je mnogo veći opseg frekvencija od AM radio uređaja. Opseg daljine za FM prenos je ograničeniji od AM - obično manje od 100 milja. Nas zadatak u ovom projektu je da konstruišemo manuelni ulazni interfejs tj. sve ono čime ćemo fizicki kontrolisati FM radio .

Kontrolu radija ćemo vršiti pomoću 8 tastera od kojih će svaki imati svoju funkciju, i jednog rotacionog kodera. Prva četiri tastera će služiti za menjanje kao i čuvanje radio stanica dok ce ostala četiri služiti za promenu frekvencije (pretragu radio stanica) odnosno podešavanje jačine zvuka. Softverski smo odradili i da se radio stanice pamte držanjem tastera dok samo jednim kratkim pritiskom tastera tražimo već upamćenu radio stanicu koja je na njemu.

Pomoću rotacinog kodera ćemo takođe moći menjati frekvenciju i volume kao i uključivati i isključivati naš FM radio. Ovaj manuelni interfejs ćemo integrisati u kućište FM radio prijemnika. Takođe kontrola radio prijemnika je omogućena i preko aplikacije koju možemo pokrenuti preko mobilnog telefona ili nekog drugog „pametnog“ uređaja.

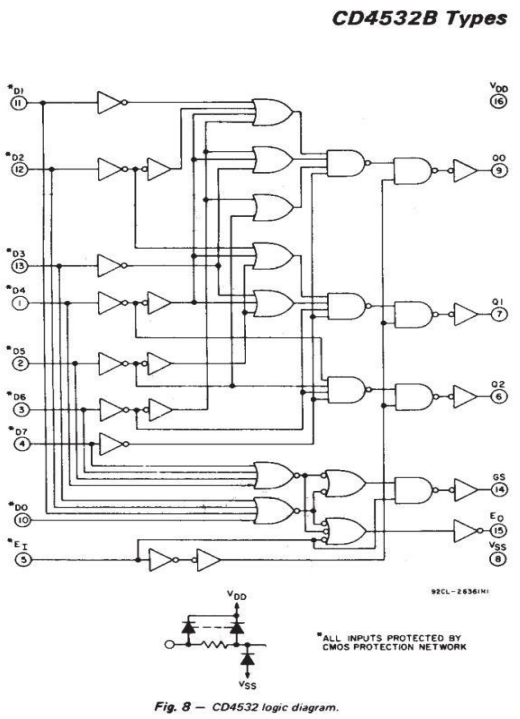
II. KOMPONENTE PROTOTIPA

Prototip sadrži sledeće komponente:

1. Proiritetni koder
2. Tasteri
3. Rotacioni koder
4. Otpornici
5. Arduino razvojni sistem

A. Milosavljević, V. Panić i M. Radonjić su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandarm@elfak.rs, veljkopanic@elfak.rs, mihajlo.radonjić@elfak.rs)

Kodiranje je suprotna operaciju od dekodiranja i označava postupak dogovorenog jednoznačnog pridruživanja skupa ulaznih binarnih cifri odgovarajućem binarnom nizu - kodu. Kombinaciona mreža koja obavlja ovu operaciju naziva se koder (eng. *encoder*).



Slika 1. Prioritetni koder.

Koder je kombinaciona mreža koja na izlazu daje n bitni binarni kod zavisno od m aktiviranih ulaza, 0<=m<=2n-1.

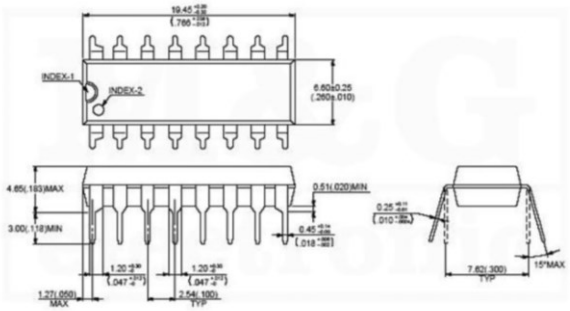
Potpuni koder 8/3 ima 8 ulaza i 3 izlaza.U datom trenutku samo jedan od ulaza kodera može biti aktivan (imati signal 1). U tom trenutku, u zavisnosti od toga koji je ulaz aktivan, na izlazu se generiše binarna kombinacija bitova koja odgovara rednom broju aktivnog ulaza. Razlog iz kod smo mi koristili prioritetni koder je upravo taj sto nam je bilo potrebno da osam različitih ulaza svedemo na tri izlaza zbog ograničenosti Arduino pinova koje smo imali na raspolaganju.

Koder je kombinaciona mreža koja na izlazu daje *n*

bitni binarni kod zavisno od *m* aktiviranih ulaza, 0<=*m*<=2*n*-1.

Potpuni koder 8/3 ima 8 ulaza i 3 izlaza. U datom trenutku samo jedan od ulaza kodera može biti aktivan (imati signal 1). U tom trenutku, u zavisnosti od toga koji je ulaz aktivan, na izlazu se generiše binarna kombinacija bitova koja odgovara rednom broju aktivnog ulaza. Razlog iz kod smo mi koristili prioritetni koder je upravo taj što nam je bilo potrebno da osam različitih ulaza svedemo na tri izlaza zbog ograničenosti Arduino pinova koje smo imali na raspolaganju.

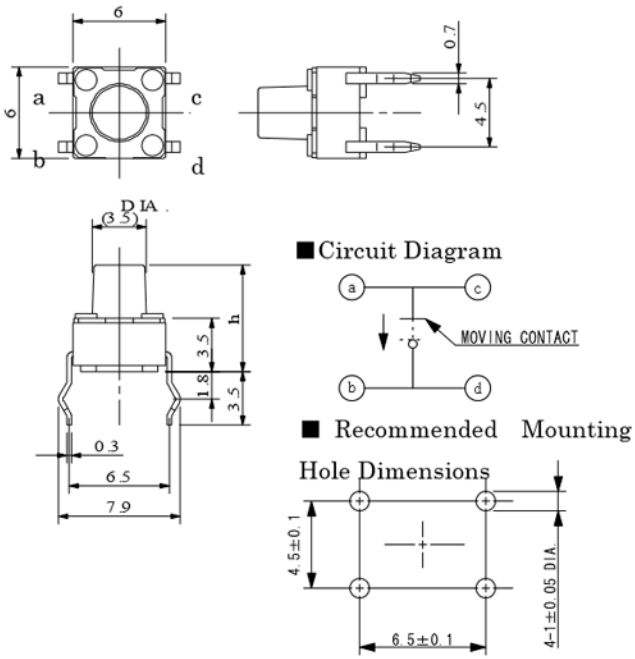
Na slici 2 prikazan je dvostruki linijski paket (dual in-line package), koji može biti od različitog materijala (plastični, keramički,...). To je paket elektronskih komponenti sa pravougaonim kućištem i dva paralelna reda električnih spojnih pinova. Paket može biti montiran kroz otvor (*through-hole technology*) na štampanu ploču (PCB-*printed circuit board*) ili umetnut u utičnicu. Uglavnom se koriste kod integrisanih kola.



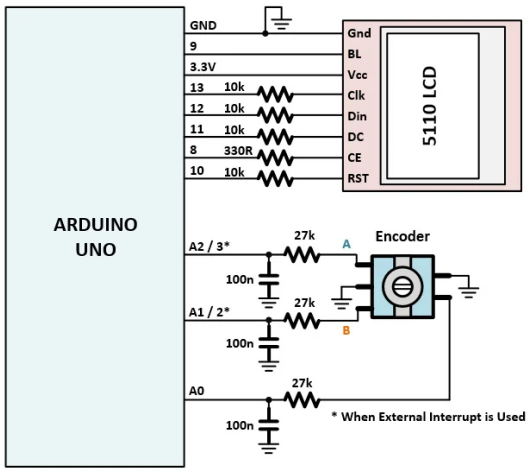
Slika 2. Dvostruki linijski paket.

Na slici 3, dat je prikaz šeme tastera. Taster ili dugme, je jednostavan mehanizam prekidača za kontrolu nekog aspekta mašine ili procesa. Uglavnom su napravljena od tvrdog materijala i površina je pogodna da čovek može da upotrebi prst ili šaku. Dugmad su najčešće tipa *baised switch*, gde održavaju svoj postavljeni položaj koji se jednom aktivira. Ovakva dugmad sadrže mehanizam koji ih dovode u drugi položaj kad ga operater pusti. Imaju izrazito široku upotrebu kod tastatura, kalkulatora, kuhinjskih aparata.

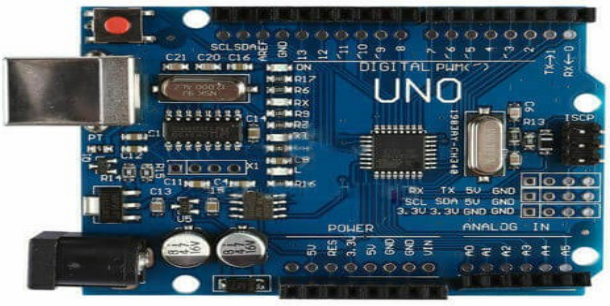
Na slici 4 je prikazana šema rotacionog enkodera povezanog na Arduino. Drugi naziv za rotacioni enkoder jeste i enkoder osovine. On predstavlja elektromehanički uređaj koji pretvara ugaoni položaj ili kretanje vratila ili osovine u analogne ili digitalne izlazne signale. Postoje dva tipa rotacionih enkodera, to su: apsolutni i inkrementalni. Izlaz apsolutnog enkodera pokazuje trenutnu poziciju osovine, što ga čini pretvaračem ugla. Izlaz inkrementalnog enkodera daje informacije o kretanju osovine, koje se obično obrađuje negde drugde u informacije kao što su položaj, brzina i rastojanje. Imaju široku upotrebu kod aplikacija za praćenje i kontrolu kao i kod mehaničkih sistema za robotiku, industrijsku kontrolu i slično.



Slika 3. Šeme tastera.



Slika 4. Rotacioni koder povezan na Arduino.



Slika 5. Arduino Uno.

Arduino je kompanija za hardver i softver otvorenog koda, projektna i korisnička zajednica koja dizajnira i proizvodi mikrokontrolere sa jednom pločom i setove za mikrokontrolere za izgradnju digitalnih uređaja. Arduino ploče su komercijalno dostupne sa zvanične veb stranice ili preko ovlašćenih distributera.

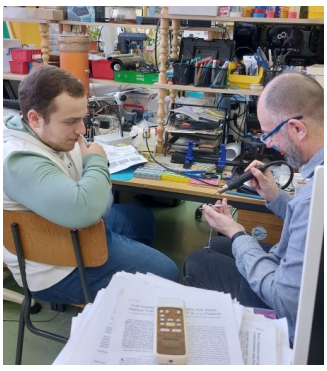
Dizajn ploče Arduino koristi razne mikroprocesore i kontrolere. Ploče su opremljene setovima digitalnih i analognih ulazno/izlaznih (I/O) pinova koji se mogu povezati sa raznim pločama za proširenje ('štitovi') i sa drugim elektronskim kolima. Ploče imaju serijske komunikacione interfejse, uključujući USB-*Universal Serial Bus* na nekim modelima, koji se takođe koriste za učitavanje programa sa ličnih računara. Mikrokontroleri se mogu programirati pomoću programskih jezika C i C ++, koristeći standardni programski interfejs aplikacije (API - Application programming interface) koji je poznat i kao „jezik Arduino“. Pored upotrebe tradicionalnih kompajlera, projekat Arduino pruža integrisano razvojno okruženje (IDE-Integrated development environment) i alat za komandnu liniju (arduino-cli) razvijen u programu Go.

Na slici 6 je prikazan Arduino UNO sa svojim komponentama. Komponente su sledeće:

1. Napajanje (USB / Barrel Jack)
2. Pinovi (5V, 3.3V, GND, analogni, digitalni, PWM, AREF)
3. Reset dugme
4. LED indikator napajanja
5. TX i RX LED diode
6. Glavno integrisano kolo

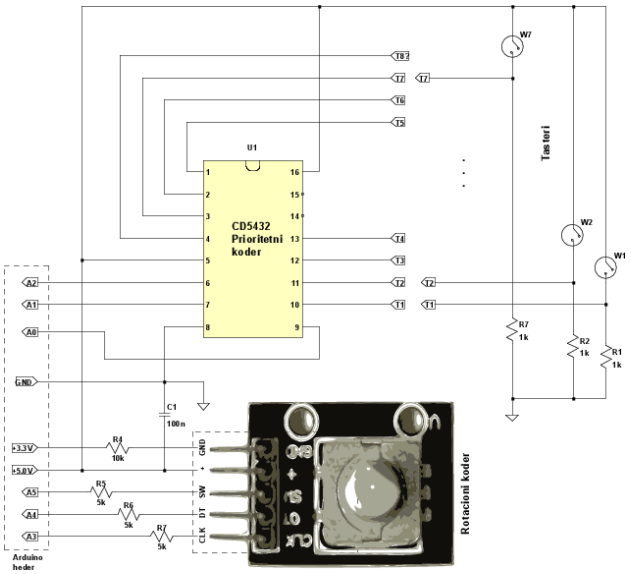
III. REALIZACIJA

Kada smo krenuli u realizaciju ovog projekta prva stavka nam je bila kako da zbog zauzetosti ostalih pinova, 8 ulaza sa tastera svedemo na samo 3 izlaza koja smo imali na raspolaganju. Dosli smo do ideje da koristimo prioritetni koder 8/3 tip CD4532B i da svedemo to na prosto binarno kodiranje koje smo učili iz digitalne elektronike. Prema tome, analiziramo smo *data sheet* ovog prioritetnog kodera ne bi li ispoštovali sve načine povezivanja koje ovaj modul zahteva.



Slika 6. Rad u laboratoriji.

Sledeći korak bio je da adekvatno zalemimo sve tastere i otpornike od 10 kHz na raster ploču, i da onda vodeći računa o pinovima sa *data sheet*-a, sve te tastere povežemo na prioritetni koder. Posle toga, smo povezali beskonačni rotacioni koder za raster ploču i onda povezali sve komponente na Arduino.



Slika 7. Realizovani ulazni interfejs.

Nakon toga smo krenuli sa pisanjem koda i programiranjem Arduina.

```
void manual() {
  // put your main code here, to run repeatedly:
  //Serial.println("Zdravo");
  //delay(50);
  int a=digitalRead(A0);
  //Serial.print("A0:");
  //Serial.println(a);
  int b=digitalRead(A1);
  //Serial.print("A1:");
  //Serial.println(b);
  int c=digitalRead(A2);
  //Serial.print("A2:");
  //Serial.println(c);
  if (a != 0 || b != 0 || c != 0) {
    Serial.print(a);
    Serial.print(b);
    Serial.print(c);
    Serial.println();
    if (a == 1 && b == 0 && c == 1) {
      Serial.println("Memorija 1");
    }
    else if (a == 0 && b == 0 && c == 1){
      Serial.println("Memorija 2");
    }
    else if (a == 1 && b == 0 && c == 0){
      Serial.println("Memorija 3");
    }
```

```
   else if (a == 0 && b == 1 && c == 0){
    Serial.println("Vol+");
   }
   else if (a == 0 && b == 1 && c == 1){
    Serial.println("Frekvencija+");
   }
   else if (a == 1 && b == 1 && c == 1){
    Serial.println("Vol-");
   }
   else if (a == 1 && b == 1 && c == 0){
    Serial.println("Frekvencija-");
   }
  }
  // a3 = clk
  // a4 = dt
  // a5 = switch
  trenutno_stanje = digitalRead(A3);
  int a4stanje = digitalRead(A4);
  if(trenutno_stanje!=novo_stanje){
   if(a4stanje!=trenutno_stanje){
    pol+=10;
   }else{
    pol-=10;
   }
   Serial.print("Pozicija: ");
   Serial.println(pol);
  }
  novo_stanje=trenutno_stanje;

}
```

Najveći zadatak je bio programiranje dela koda koji je zadužen za rotacioni koder i njegovo testiranje, jer u početku nismo mogli naći odgovarajući kod i prilagoditi ga da služi našem modulu. Kod testiranja, koder tokom okretanja nije davao negativne vrednosti, kada je bio okrenut na tu stranu i nije odgovarajuće brzo reagovao na promene položaja. Ispostavilo se da je problem bio u lošem lemljenju i povezivanju tog dela kola, što smo ubrzo rešili.

IV. Zaključak

U ovom radu smo uspešno realizovali i implementirali manualni ulazni interfejs FM radija na bazi Arduina, kao deo većeg projekta realizacije FM radia. Ovim interfejsom je moguće fizički kontrolisati FM radio i ručno upravljati frekvencijama i stanicama koje on poseduje. Radio uspešno menja i pamti stanice, i reaguje na menjanje frekvencija.

Zahvalnica

Predmet Telekomunikacije, na smeru Upravaljanje sistemima, Elektronskog fakulteta Univerziteta u Nišu je ispit na 5. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri laboratoriji za Telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

Literatura

[1] Arduino Uno vikipedija: https://en.wikipedia.org/wiki/Arduino_Uno
[2] Otpornik vikipedija: https://sh.wikipedia.org/wiki/Otpornik.
[3] Arduino tutorijal: https://www.circuitbasics.com/arduino-7-segment-display-tutorial 2.

Sedmosegmentni displej za FM radio prijemnik

Katarina Milić, Miljan Nedeljković

Sadržaj - U ovom projektu se bavimo projektovanjem i regalizacijom hardvera i softvera sedmosegmentnog displeja za FM radio prijemnik na bazi Arduina. Potrebno je realizovati jednostavan način prikazivanja frekvencije kako bi se omogućilo njeno podešavanje. Zbog jednostavnosti hardverske realizacije smo se odlučili za sedmosegmentni displej, dok je kompleksnost prikazivanja prebačena na softver koji izvršava mikrokontroler. Realizovani prototip je pokazao odlične karakteristike, kao i fleksibilnost u pogledu prikazivanja različitih numeričkih informacija.

I. UVOD

Zadatak ovog projekta je projektovanje sedmosegmentnog displeja za FM radio. U ovom projektu ćemo se baviti osmišljanjem hardvera, a zatim i realizacijom softvera displeja u razvojnom okruženju Arduino. Sedmosegmentni displej [1] je tip elektronske komponente koji služi za prikazivanje decimalnih cifara. Prvi prototip displeja je korišćen početkom 20. veka, tačnije 1903. kada je Karl Kinsli osmislio metodu za telegrafsko prenošenje slova i brojeva i štampanja na traci u segmentiranom formatu. Korišćeni su i za prikazivanje biranog telefonskog broja tokom prelaska sa ručnog na automatsko telefonsko biranje. 1970-ih displej je proširio svoju upotrebu usled pojave LE dioda.

Sedmosegmentni displeji mogu da koriste displej sa tečnim kristalom (LCD), dioda koja emituje svetlost (LE dioda) za svaki segment, elektrohromski displej ili druge tehnike za generisanje ili kontrolu svetlosti kao što je gasno pražnjenje sa hladnom katodom, itd. Iako golim okom sve cifre sedmosegmentnog displeja izgledaju upaljeno, zapravo samo jedna cifra svetli u bilo kom trenutku na multipleksiranom displeju. Cifra se menja dovoljno velikom brzinom da ljudsko oko ne može da vidi treptanje (na ranijim uređajima to je bilo vidljivo perifernim vidom).

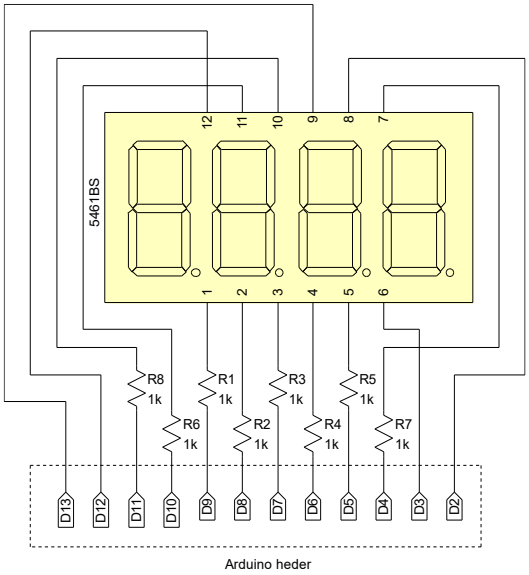
Prototip koji smo realizovali sadrži sledeće komponente:

- sedmosegmentni displej sa četiri cifre koji koristi LE diode za emitovanje svetlosti,
- otpornike (osam komada) od 1kΩ,
- raster ploču
- *Arduino* razvojni sistem.

II. KOMPONENTE U REALIZACIJI DISPLEJA

Šema koju smo koristili za realizaciju ovog projekta prikazana je na slici 1. Na toj slici se može videti povezivanje otpornika i pinova za svaku diodu.

K. Milić i M. Nedeljković su sa Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: katarina.milic@elfak.rs



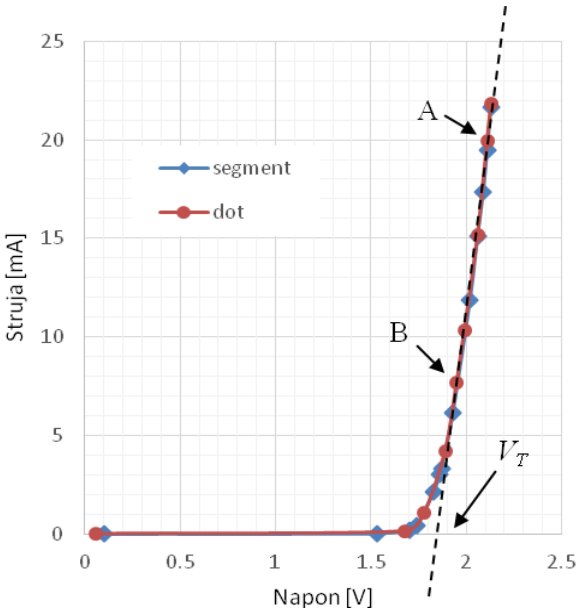
Slika. 1. Šema povezivanja pinova i dioda.

Osnovne specifikacije sedmosegmentnog displeja koji je korišćen, su:

Model: 5461BS
Veličina: 0,56-inchEmittingcolor:Red(Ultra-Bright)
Mod: Common-Anode(CA)
Cifre: 4-Digit
Kategorija: LED7-SegmentDisplay
Proizvođač: XLITXTechnology

Otpornik [2] je pasivna elektronska komponenta koja pruža otpor električnoj struji stvarajući pad napona između priključaka. U otporniku uložena električna energija većim delom nepovratno prelazi drugi vid energije, najčešće u toplotu ali može i svetlosnu, mehaničku itd. U ovom projektu koristili smo osam otponika radi ograničavanja struje kroz segmente displeja. Iz strujno-naponske karakteristike koje smo odredili možemo zaključiti koliki nam je otpornik potreban.

Najpre smo obavili seriju merenja strujno naponskih karakteristika segmenata. Razlog za ovo je sumnja da se njihove karakteristike razlikuju usled različite veličine segmenata. Na primer, segment decimalne tačke je mnogo manjih dimenzija nego segment uspravne lijine u cifri jedinice. Za ovo merenje smo koristili izvor konstantne struje koji je bio dostupan u laboratoriji. Nakon svake promene stuje kroz LE diode, zapisali smo izmerene vrednosti napona i struje [3] na krajevima LE diode. Na taj način smo dobili grafik prikazan na slici 2.



Slika. 2. Strujno-naponska karakteristika LE diode.

TABELA I
MERENJA KARAKTERISTIKA LE DIODE NA OSNOVU KOJE SMO NACRTALI GRAFIK

V [V] - napon	I [mA] - struja
0,10	0
1,74	0,42
1,83	2,12
1,86	3,03
1,87	6,12
1,93	11,84
2,02	15,08
2,06	17,34
2,09	19,46
2,11	21,65

Izračunavanje praga LE diode smo uradili na sledeći način, na dobijenom grafiku strujno naponske-karakteristike smo povukli pravu kroz dve proizvoljne tačke na ustaljenom stanju krive grafika i tako dobili presek sa x-osom tj. osom koja nam predstavlja napon. Dobijena vrednost preseka prave i naponske ose jeste prag LE diode. Najpre smo izabrali dve tačke na strujno-naponskoj karakteristici: A = (20 mA, 2,1 V) i B = (8 mA, 1,95 V). Jednačina prava kroz ove dve tačke se može zapisati kao:

$$i(u) = g_d u + i_0, \tag{1}$$

gde je g_d dinamička provodnost diode, a i_0 predstavlja konstantu. Dinamička provodnost predstavlja nagib tangente na karakteristici i može se odrediti iz izabrane dve tačke kao:

$$g_d = \frac{i_A - i_B}{u_A - u_B} = \frac{(20-8)mA}{(2.1-1.95)V} = 80mS. \tag{2}$$

Dinamička otpornost LE diode je recipročna vrednost dinamičke provodnosti:

$$r_d = \frac{1}{g_d} = 12.5\,\Omega \tag{3}$$

Jednačina prave se može zapisati i na drugi način, koristeći napon praga V_T:

$$i(u) = g_d\,(u - V_T). \tag{4}$$

Odavde se može izračunati napon praga kao:

$$V_T = u_A - \frac{i_A}{g_d} = u_A - r_d i_A, \tag{5}$$

odnosno:

$$V_T = u_B - r_d i_B \tag{6}$$

Iz podataka koje smo izabrali za tačke A i B se dobija da prag diode iznosi:

$$V_T = 1.85V \tag{7}$$

Da bismo odredili vrednost otpornosti koja nam je potrebna, moramo najpre da odlučimo koliku vrednost struje kroz diodu ćemo postaviti. U ovome je potrebno da razmotrimo koliku struju naš sistem može da podrži na svojim pinovima, maksimalnu struju koja je dozvoljena za izabrani kontroler, kao i činjenicu da li je sjajnost diode dovoljna za našu primenu sa izabranom strujom. Imajući sve ovo na umu, odlučili smo da fiksiramo struju segmenta na $I_D = 3$ mA. Odatle proizilazi da nam je potreban otpornik od:

$$R = \frac{V_{CC} - V_T}{I_D} = \frac{(5-1.85)\,V}{3\,mA} = 1.05\,k\Omega \tag{8}$$

U laboratoriji smo pronašli najbližu vrednost otpornika od 1 kΩ koju ćemo koristiti za ovu svrhu. Pored toga, zaključili smo da svi segmenti imaju identičnu strujno-naponsku karakteristiku koja ne zavisi od veličine samog segmenta.

Iskoristili smo raster ploču za ovaj projekat, tako što smo na njoj zalemili sedmosegmentni displej, žicu za povezivanje pinova sa Arduinom i otpornike.

Arduino je fizičko – računarska platforma (razvojni sistem) otvorenog koda. Hardver se sastoji od jednostavnog otvorenog hardverskog dizajna arduino ploče sa Atmel AVR procesorom i pratećim ulazno-izlaznim elementima, tačnije, na sebi poseduje mikrokontroler. Softver se sastoji od razvojnog okruženja koje čine standardni kompajler i bootloader koji se nalazi na samoj ploči.

Arduino hardver se programira koristeći programski jezik zasnovan na *Wiring* jeziku. U osnovi je sličan C++ programskom jeziku sa izvesnim pojednostavljenjima i izmenama. Bitan aspekt *Arduino* projekta je standardizovan raspored konektora koji omogućava lako povezivanje sa dodatnim modulima, poznatijim kao štitovi.

Arduino integrisano razvojno okruženje je aplikacija napisana u *Java* programskom jeziku. Kreirano je tako da uvede u programiranje učenike, studente, i ostale početnike koji nisu upoznati sa načinom razvoja softvera. Sastoji se od uređivača koda sa mogućnostima kao što su označavanje koda, uparivanje zagrada, automatsko uvlačenje linija. Ovaj uređivač može da prevede kod a zatim ga i prebaci u čip jednom komandom.

Arduino integrisano razvojno okruženje dolazi sa C/C++ bibliotekom zvanom „*Wiring*“ koja čini uobičajene izlazno-ulazne operacije veoma jednostavnim.



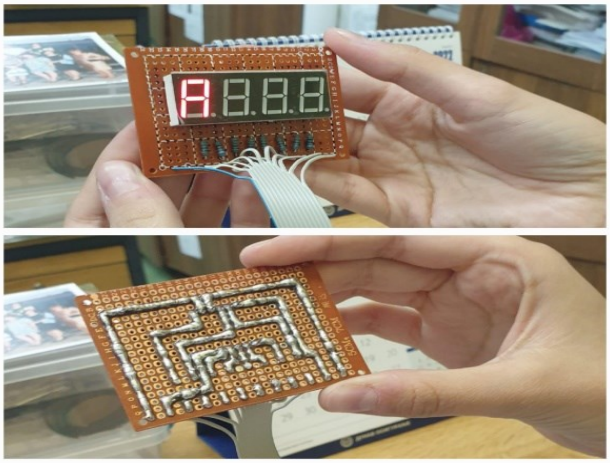
Slika 3. Strujno-naponska karakteristika LE diode.

Za realizaciju ovog projketa koristili smo *Arduino Uno* ploču. Ona je bazirana na *Microchip Atmega328P* mikrokontroleru koji je proizveden 2010. godine od strane *Arduina*.

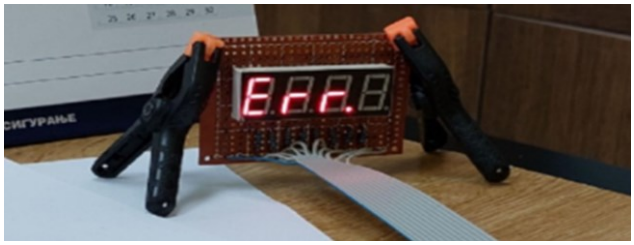
III. PRAKTIČNA REALIZACIJA

Za realizaciju ovog projekta koristili smo raster ploču dimenzije 7x9 cm i rupica veličine 2,54 mm, na kojoj smo lemili kalajem komponente. Prvo smo zalemili sedmosegmentni displej sa četiri cifre, nakon toga smo merili strujno-naponsku karakteristiku i tako odredili koje ćemo otpornike koristiti. Koristeći konstantni strujni izvor kao i multimetar, uspeli smo da dobijemo potrebne podatke za tu karakteristiku. Odlučili smo se za otpornike R = 1 kΩ i iskoristili 8 komada zbog 8 pinova koji se nalaze na displeju. Iz priložene šeme (slika 1) videli smo kako ih treba pravilno povezati.

Koristeći tehničku dokumentaciju [4] za pribavljanje neophodinih informacija utvrdili smo koji od pinova su anode a to smo potvrdili eksperimentalnim putem, što nas je dovelo do zaključka da su pinovi 13, 12, 3, 2 oni koji šalju signal za uključivanje odgovarajuće cifre. Na multipleksiranom displeju samo jedna cifra svetli u bilo kom trenutku koja se menja velikom brzinom i ljudsko oko ne može da primeti treptanje. Nakon toga bilo je potrebno utvrditi koji pin uključuje odgovarajući segment cifre. Improvizacijom starog kabla koji smo blankirali i izolovali sa termoskupljajućim bužirom, uspešno smo napravili 12 pinova koje smo povezali na *Arduino*. Tu smo završili hardverski deo projekta i prešli smo na softverski.



Slika 4. Konačni izgled displeja.



Slika 5. Testiranje koda i prikazivanje greške.

U *Arduino* okruženju nakon definisanja svih potrebnih pinova, odredili smo sve potrebne cifre od 0 do 9, kao i tačku. Pored cifara ubacili smo i tri karaktera (Err.), koji bi se ispisivali ukoliko dođe do neke greške. Naš cilj je bio da napravimo funkcije za pojačavanje zvuka FM radia kao i prikaz i memorisanje frekvencija radio stanica. Granica je bila od 87 MHz do 106 MHz, ako bi se granica premašila ispisivalo bi se Err. Minimalna jačina zvuka je 0 a maksimalna 15.

Prilikom testiranja, utvrdili smo da je naš kod funkcionalan i da je spreman za dalju upotrebu naših kolega. Tokom glavnog testa, kada smo sve povezali, jos jednom smo potvrdili da sve funkcionise kako treba, stanice i jacina zvuka se menja do određenih granica koje smo zadali, kvalitet zvuka je bio zadovoljavajući, čak iznenađujuće dobar.

```
void setup() {
        // ZAJEDNICKE ANODE:
        pinMode(12, OUTPUT);
        pinMode(13, OUTPUT);
        pinMode(2, OUTPUT);
        pinMode(3, OUTPUT);
        // KATODE SEGMENATA:
        pinMode(10, OUTPUT);
        pinMode(5, OUTPUT);
        pinMode(6, OUTPUT);
        pinMode(7, OUTPUT);
        pinMode(4, OUTPUT);
        pinMode(11, OUTPUT);
        pinMode(9, OUTPUT);
        pinMode(8, OUTPUT);
}
void zero(){
        digitalWrite(10, LOW);
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
        digitalWrite(7, LOW);
        digitalWrite(4, LOW);
        digitalWrite(11, LOW);
        digitalWrite(9, LOW);
        digitalWrite(8, HIGH);
}
```

Slika 6. Deo koda koji inicijalizuje pinove i ispisuje nulu.

```
void loop() {
        // ovde treba postaviti segment 1
        digitalWrite(12, HIGH);
        delay(d);
        digitalWrite(12, LOW);
        // ovde treba postaviti segment 2
        digitalWrite(13, HIGH);
        delay(d);
        digitalWrite(13, LOW);
        // ovde treba postaviti segment 3
        digitalWrite(2, HIGH);
        delay(d);
        digitalWrite(2, LOW);
        // ovde treba postaviti segment 4
        digitalWrite(3, HIGH);
        delay(d);
        digitalWrite(3, LOW);
}
```

Slika 7. Deo koda koji prikazuje kako smo multipleksirali cifre.

IV. ZAKLJUČAK

U ovom projektu cilj je bio realizovati hardverski i softverski deo sedmosegmentnog displeja koji se bazira na Arduino Uno [5] okruženju. Bili smo podeljeni u 5 grupa, svako je imao svoje zaduženje u cilju sklapanja FM Radia, što je uspešno realizovano. Primarni deo naše grupe je bio kako smisliti i napraviti sedmosegmentni displej koji će prikazivati frekvenciju radio stanice koju antena trenutno registruje, međutim tokom realizacije dodali smo jos par modifikacija i kozmetički ulepšali i usavršili. Tako da se na kraju sedmosegmentni displej koristi i za prikazivanje frekvencije radio stanica i memorisanja istih i za prikazivanje jačine zvuka radia.

LITERATURA

[1] Opis okruženja Arduino Uno. Dostupno na: https://en.wikipedia.org/wiki/Arduino_Uno

[2] Opis sedmosegmentnog displeja. Dostupno na: https://en.wikipedia.org/wiki/Seven-segment_display

[3] Opis otpornika. Dostupno na: https://sh.wikipedia.org/wiki/Otpornik

[4] Tehnička specifikacija za sedmosegmentni displej. Dostupno na: http://www.xlitx.com/datasheet/5461BS.pdf

[5] Uputstvo za povezivanje Arduino okruženja sa sedmosegmentnim displejem. Dostupno na: https://www.circuitbasics.com/arduino-7-segment-display-tutorial/.

[6] Realizacija strujno-naponske karakteristike LE diode. Dostupno na: https://fizika.pmf.kg.ac.rs/pages/srednje%20skole/skolski-eksperiment-iz-fizike/2020IU%20karakteristika%20LED%20diode%202.pdf

Interfejs za FM prijemnik sa Arduinom

Dimitrije Stojanović, Natalija Stojanović

Sadržaj - U ovom projektu se bavimo projektovanjem i realizacijom hardverskog i softverskog interfejsa za FM radio prijemnik koji će biti u mogućnosti da se podesi na odgovarajuću noseću frekvenciju i da služi za prijem FM stereo signala.

I. Uvod

Zadatak ovog projekta je projektovanje interfejsa FM prijemnika na bazi *Arduina*. Da bismo ispunili zadatak potrebno je ostvariti hardversko povezivanje modula sa *Arduino*-vim razvojim sistemom i realizacija softvera za kontrolu noseće frekvencije. FM prijemnik je vrsta radio prijemnika koji prima radio signale u FM opsegu frekvencija, obično od 87.5 do 108 MHz. Koristi se za prijem FM radio signala, obično sa lokalne radio stanice. Antena prima dolazni radio signal i pretvara ga u električni signal.

Ovo se zatim pojačava od strane pojačavača, podešava na željenu frekvenciju od strane tjunera i demodulira demodulatorom tako da se audio informacije mogu izdvojiti. Konačno, audio signal se pojačava audio pojačalom, a zatim se šalje na izlazni uređaj kao što su zvučnici ili slušalice. Moderni FM prijemnici se često ugrađuju uređaje kao što su stereo uređaji za automobile i mobilni telefoni, a dostupni su i kao samostalne jedinice.

II. FM Prijemnik RDA5807M modul

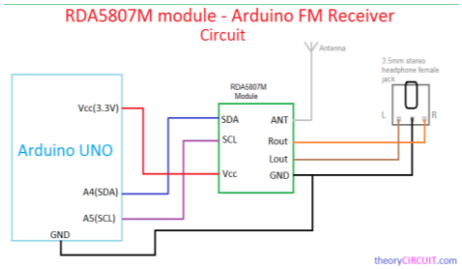
Komponente koje smo koristili u toku izrade prototipa su: FM prjemnik modul RDA5807M, 2x otpornik od 5.1 kΩ (5% tolerancije), Arduino razvojni sistem (Uno), antena, 2x pinski muski utikači(10 pinova,8 pinova).

Pri prikupljanju literature za FM prijemnik na bazi *Arduina* karakteristike od interesa su nam bile: isplatljivost, mala potrošnja energije, laka implementacija, kvalitetan zvuk, male dimenzije.

Sve navedene karateristike je obuhvatao RDA5807M. RDA5807M je najeftiniji čip na tržištu. Mali broj pinova i prosta funkcionalnost ga odličnim za početnike inžinjerske hardvesrke realizacije. Prednost korišćenja njega jeste puno realizovanih softverskih kodova sa gotovim funkcijama korišćeni za taj čip. Male dimenzije ovog čipa omogućavaju da se prijemnik koristi u gotovo svim realizacijama FM radio prijemnika na bazi *Arduina*.

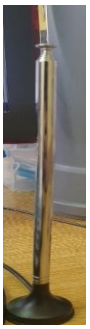
Dimenzije RDA5807M su (5.10, 3.10, 1.10) cm. Opseg frekvencija koje podrzava nas izabrani čip su 50 – 115 MHz. Oscilator na kome radi nas čip ima frekvenciju 32.768 KHz. U čipu su uključene sve osnovne kontrole zvuka. Podržava I2C komunikaciju(sa 2 žice). RDA5870M radi u opsegu 2.7-3.3 V. Mogućnost menjanja izmedju mono/stereo zvuka.

D. Stojanović i N. Stojanović su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dimitrijestoanovic@elfak.rs, natalijastojanovic@elfak.rs



Slika 1. Elektronska šema projekta.

Antena se obično sastoji od dugačke, ravne metalne šipke ili žice. Antena je suštinski deo svakog FM radija, jer je odgovorna za prijem radio signala. Antena hvata elektromagnetne talase koji sadrže audio informacije radio stanice i prenosi ih na radio prijemnik, koji zatim pojačava signal i proizvodi zvuk koji slušalac čuje. Antena se obično montira na otvorenom, bilo na krovu zgrade ili na stubu, a najčešće je povezana sa radio prijemnikom preko koaksijalnog kabla.



Slika 2. Štap antena korišćena u projektu.

III. Hardverska i softverska realizacija

Rad na projektu je izgledao kao na fotografijama koje slede (slika 3).

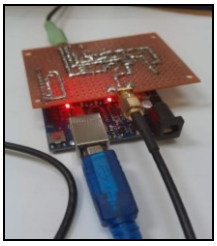
Povezivanje svih komponenti našeg modula ostvareni su lemljenjem. Ovim pristupom mi smo poboljlšali kompaktnost i uprošćenost našeg modela. Lemljeni deo modula poboljšava pouzdanost tog dela modula prilikom daljeg njegovog korišćenja ili transporta. Svaku zalemljenu

vezu na PCB tabli smo proverili unimetrom da su one zaista dobro povezane i da se zalemljene veze ne prispajaju sa drugim zameljenim vezama. Na sledećoj slici uokviren je mod na multimetru koji se koristi za proveravanje zalemljenih veza.



Slika 3. Rad u laboratoriji.

Red pinova 1x10 i 1x8 (muških) iskorišćeni su za povezivanje našeg FM prijemnika sa *Arudiono*-vim razvojnim sistemom (Uno). Prednost je kompaktnost i stabilnost veze izmedju *Arduino*-vog razvojnog sistema i našeg povezanog modula



Slika 4. Završni izgled realizovanog prototipa.

Koraci u pronalasku koda koji je bio potreban za realizaciju našeg modula su: pronalazak koda na *GitHub*-u, tako što na *web* stranicu *GitHuba*-a u labelu „Istraži“ na vrhu stranice, unošenje ključnih reči u traku za pretragu, kao što je „FM radio antena“, „RDA5807M“, iz predstavljene liste izvornih kodova izabrali smo onu koja najviše obećava, i na kraju analizom izabranog koda uzelimo smo delove koda koje su nam potrebne za rad našeg modula.

```
#include <Arduino.h>
#include <Wire.h>
#include <radio.h>
#include <RDA5807M.h>

// ----- Fixed settings here. -----

#define FIX_BAND     RADIO_BAND_FM   ///< The band that will be tuned by this sketch is FM.
#define FIX_STATION  9560
#define FIX_VOLUME   5

RDA5807M radio;    // Create an instance of Class for RDA5807M chip

/// Setup a FM only radio configuration
/// with some debugging on the Serial port
void setup() {
  // open the Serial port
  Serial.begin(9600);
  Serial.println("Radio...");
  delay(200);

  // Initialize the Radio
  radio.init();

  // Enable information to the Serial port
  radio.debugEnable();
```

Slika 5. Deo koda za uključivanje funkcije za promenu frekvencije.

Izabrani deo koda je uključivao funkcije za promenu frekvencije na koju želimo da se priključimo, kao i kontrolu jačine zvuka od 0-15. Deo koda kojim ovo omogućavamo prikazan je na slici 5. Crvenim kvadratom su označene ime promenljive za jačine zvuka i trenutne frekvencije. Kao što vidimo pri pokretanju modula po *default*-u je jačina zvuka 5, a frekvencija 9560 Hz.Softverski alat koji smo koristili za ovaj kod je *Arduino IDE 2.0*.

```
  // Set all radio setting to the fixed values.
  radio.setBandFrequency(FIX_BAND, FIX_STATION);
  radio.setVolume(FIX_VOLUME);
  radio.setMono(false);
  radio.setMute(false);
} // setup


/// show the current chip data every 3 seconds.
void loop() {
  char s[12];
  radio.formatFrequency(s, sizeof(s));
  Serial.print("Station:");
  Serial.println(s);

  Serial.print("Radio:");
  radio.debugRadioInfo();

  Serial.print("Audio:");
  radio.debugAudioInfo();

  delay(3000);
} // loop
```

Slika 6. Deo koda za uključivanje funkcije za promenu frekvencije.

IV. Zaključak

Zadatak projekta bio je da se realizuje hardver i softverski deo FM prijemnik za primenu u FM radiju na bazi *Arduina*. Izvršena je analiza zadatka i obavljeno je projektovanje prototipa našeg modula. Odabrali smo čip za naš modul prema našim željenim karakteristikama. Pri hardverskoj realizaciji urađena je validnost zalemljenih veza pomoću multimetra, urađena je provera validnost svih izabranih komponenti. Softverska realizacija je urađena tako što smo pronašli kod sa već napravljenim funkcijama na *GitHub*-u i usvojili smo funkcije koje nama odgovaraju. Finalizacija našeg prototipa urađena je tako što smo softver otpremili u realizovani hardver i testirali ga na licu mesta.

Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta iz ispita Telekomunikacije (V semestar osnovnih akademskih studija), na smeru Upravljanje sistemima, Elektronskog fakulteta Univerziteta u Nišu. Hardverska realizacija projekta izvedena je u Laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

Literatura

[1] Theory circuit. Dostupno na: https://theorycircuit.com/arduino-rda5807m-fm-receiver.
[2] Datasheet za RDA5807M_C82537. Dostupno na: https://datasheet.lcsc.com/szlcsc/RDA5807M_C82537.pdf.
[3] Tehnička dokumentacija za RDA5807, Texas Instruments. Dostupno na: https://www.ti.com/product/RDA5807M.
[4] Arduino IDE 2. Dostupno na: https://docs.arduino.cc/software/ide-v2.

Analiza pojačavača malih signala realizovanog korišćenjem bipolarnog tranzistora u konfiguraciji sa zajedničkim emitorom

Stefan Nikolić

Sadržaj – U ovom radu biće prikazana analiza pojačavačkog kola realizovanog korišćenjem bipolarnog tranzistora u konfiguraciji sa zajedničkim emitorom, uporedni pregled rezultata dobijenih preko *LTspice* simulacije i eksperimentalnih merenja, kao i analiza promene naponskog pojačanja sa amplitudom ulaznog napona $V_{IN(P-P)}$, promeni frekvencije ulaznog signala f_{IN}, promeni otpornosti R_4 u kolu emitora i priključivanju otpotnosti opterećenja R_L na izlaz datog kola.

I. Uvod

Bipolarni tranzistor (BJT – *Bipolar Junction Transistor*) je aktivna elektronska komponenta sa tri izvoda. BJT je napravljen od tri različito dopirana poluprovodnička dela: emitorski, bazni i kolektorski. Ovi delovi su, redom, P tipa, N tipa i P tipa u PNP tranzistorima i N tipa, P tipa i N tipa u NPN tranzistorima. Svaki poluprovodnički deo je priključen na označen izvod: emitor E, baza B i kolektor C. Bipolarni tranzistor poseduje 4 režima rada od kojih su nama od značaja samo 2: prekidački režim i pojačavački režim rada [1]. Ostala dva režima su zakočenje i režim inverzne polarizacije.

Prvi bipolarni tranzistori izrađeni od germanijuma pojavili su se 1948. godine. Zbog te činjenice, 2023. godine obeležava se 75. godišnjica bipolarnog tranzistora. Danas, upotreba bipolarnih tranzistora je potisnuta u korist CMOS tehnologije u dizajnu integrisanih kola. Ipak, bipolarni tranzistori ostaju komponente koje se koriste za analogna kola, bilo diskretna ili integrisana. Zbog svoje temperaturne osetljivosti, bipolarni tranzistor može da se koristi za merenje temperature. Njegova nelinearna karakteristika se takođe može iskoristiti da računa logaritme.

U ovom radu biće urađena analiza pojačavačkog kola bipolarnog tranzistora u konfiguraciji sa zajedničkim emitorom. Analiza je sprovedena eksperimentalnim putem i simulacijom korišćenjem programa *LTspice* [2].

II. Kolo pojačavača sa zajedničkim emitorom

Da bi bipolarni tranzistor radio kao pojačavač, potrebno je da u provodnom stanju bude u normalnoj aktivnoj oblasti rada [1]. Naponsko pojačanje za male signale predstavlja odnos napona na izlazu i ulazu kola. Napon na ulazu kola je napon koji se dovodi na priključak baze, a napon na izlazu je napon koji se očitava sa priključka kolektora datog tranzistora Izlazni signal će biti fazno pomeren za 180° u odnosu na ulazni. U ovom radu biće korišćena polarizacija bipolarnog tranzistora preko razdelnika napona. Električna šema sa vrednostima elemenata u kolu prikazana je na slici 1. U ovom slučaju napon na bazi tranzistora je:

$$V_B = \frac{R_2}{R_1+R_2} V_1 = 1{,}75\ V, \quad (1)$$

dok se za struju emitora može napisati:

$$I_E = \frac{V_B - V_{BE}}{R_4+R_5} = 1{,}05\ mA, \quad (2)$$

kako je $\beta = \frac{I_C}{I_B} \gg 1$, onda je $I_C \approx I_E = 1{,}05\ mA$. Napon između kolektora i emitora je tada:

$$V_{CE} = V_1 - (R_3 + R_4 + R_5) I_C = 4{,}015\ V. \quad (3)$$

Kondenzator C_1 prespaja naizmenični signal na masu, odvajajući ga od jednosmernog signala na emitoru. Međutim, kada bi ovaj kondenzator prespojio ukupnu otpornost u emitoru naponsko pojačanje bilo bi zavisno od temperature. Da bi se postigao kompromis između temperaturne stabilnosti i vrednosti naponskog pojačanja, otpornost u emitoru treba razdeliti [1]. Kako je $V_t = 26\ mV$, na osnovu [1] efektivna otpornost emitora je:

$$r_e \approx \frac{V_t}{I_E} = 24{,}8\ \Omega, \quad (4)$$

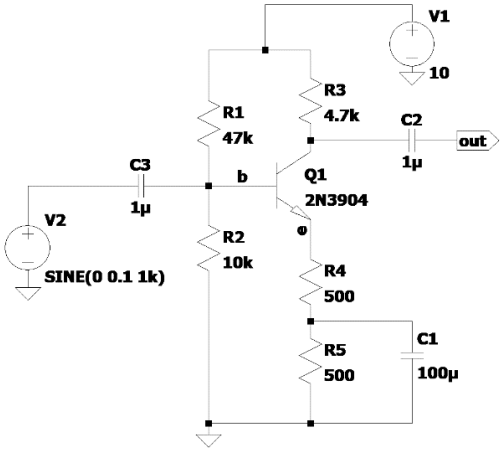
tako da se za naponsko pojačanje dobija:

$$A_v \approx -\frac{R_3}{R_4} \approx -9{,}4. \quad (5)$$

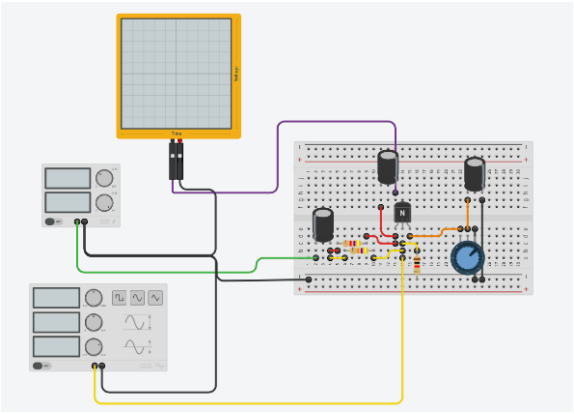
U praktičnoj realizaciji korišćen je NPN bipolarni tranzistor 2N3904 [3]. Na slici 2 prikazana je šema povezivanja komponenata u kolo pojačavača za male signale na protoploči sa ostalom laboratorijskom mernom opremom (izvorom napajanja, signal generatorom i osciloskopom). Simulacija u programu *LTspice* napravljena je na osnovu

S. Nikolić je student druge godine na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet Niš, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefannikolic@elfak.rs

šeme sa slike 1. Urađena je vremenska simulacija kola u trajanju od 2,5 ms.



Slika 1.Električna šema pojačavača sa zajednički emitorom polarizovanog preko naponskog razdelnika.



Slika 2. Šema povezivanja komponenata u kolo pojačavača za male signale sa laboratorijskom mernom opremom.

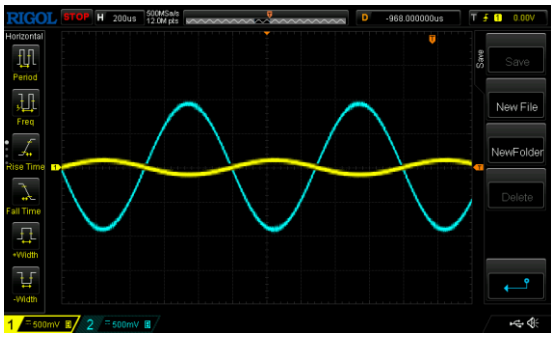
III. Rezultati i diskusija

A. Analiza naponskog pojačanja

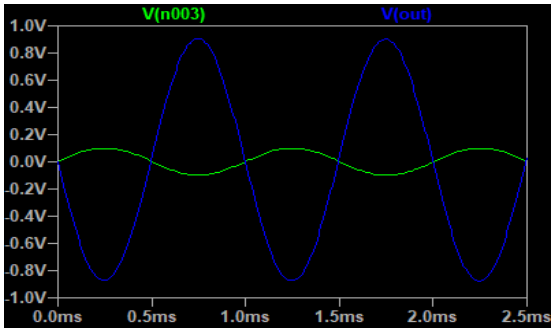
Na slici 3 prikazana je zavisnost izlaznog napona u tački *out* (slika 1) i ulaznog napona u funkciji vremena, dok je na slici 4 prikazan rezultat simulacije istog kola u programu *LTspice*, pri čemu je korišćen SPICE model tranzistora 2N3904. Može se zaključiti da je izlazni signal pojačan i fazno pomeren za 180°. Na osnovu formule:

$$Av = \frac{Vout(p-p)}{Vin(p-p)} \qquad (6)$$

gde je Av naponsko pojačanje signala, $V_{OUT(P-P)}$ je *peak – to – peak* napon izlaznog signala i $V_{IN(P-P)}$ je *peak – to – peak* napon ulaznog signala, određeno pojačanje je 9. Dobijena vrednost malo odstupa od analitički određene vrednosti jednačinom (5) zbog toga što se u analitičkom izrazu razmatra idealni bipolarni tranzistor. Rezultati eksperimenta i simulacije se dobro slažu, s tim da se u eksperimentalnim zavisnostima uočavaju signali šuma.



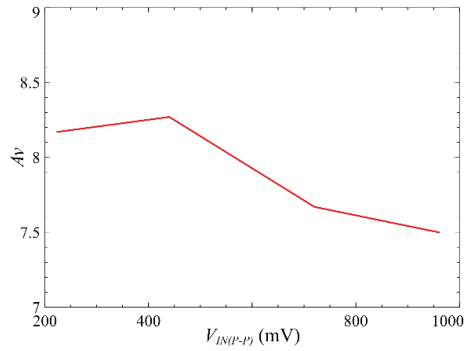
Slika 3. Grafik funkcije pojačanja signala izmeren preko osciloskopa. Žuta boja predstavlja ulazni napon, plava – izlazni.



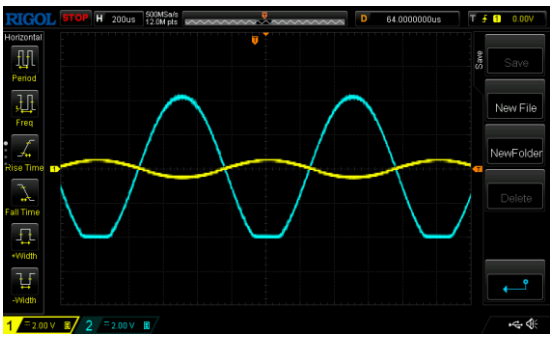
Slika 4. Grafik funkcije pojačanja signala izmeren preko *Spice* simulacije. Zelena boja predstavlja ulazni napon, plava – izlazni.

B. Promena pojačanja sa promenom amplitude $V_{IN(P-P)}$

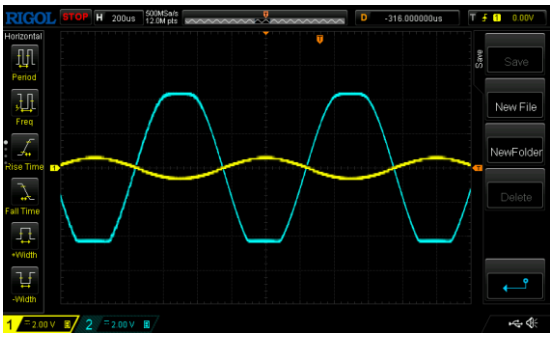
Promenom amplitude $V_{IN(P-P)}$ ulaznog signala od 200 mVpp do 1 Vpp eksperimentalno je pokazano da pojačanje ostaje približno isto (slika 5). Blagi pad pojačanja uslovljava jačanje šumova na većim amplitudama ulaznog signala. Pri vrednostima $V_{IN(P-P)}$ preko 1 V (slika 6) na osciloskopu se uočava izobličenje izlaznog signala zbog ulaska tranzistora u zasićenje, a pri vrednostima preko 1.2 V (slika 7) dodatno se vidi izobličenje zbog ulaska tranzistora u oblast zakočenja.



Slika 5. Pojačanje signala u odnosu na amplitudu $V_{IN(P-P)}$.



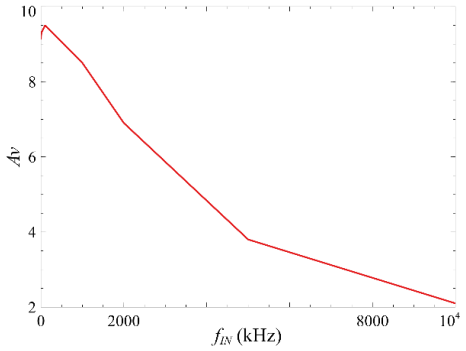
Slika 6. Grafik funkcije pojačanja izmeren na osciloskopu, gde se vidi izobličenje negativne poluperiode izlaznog signala usled ulaska tranzistora u zasićenje.



Slika 7. Grafik funkcije pojačanja izmeren na osciloskopu, gde se vidi izobličenje negativne poluperiode izlaznog signala used ulaska tranzistora u zasićenje i izobličenja pozitivne poluperiode usled ulazka u zakočenje.

C. Promena pojačanja sa promenom frekvencije f_{IN}

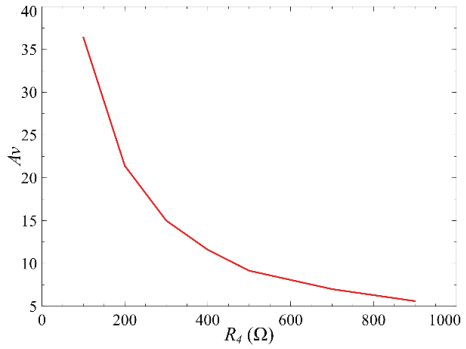
Pri promeni frekvencije ulaznog signala (slika 8) može se uočiti da za signale reda veličine kHz pojačanje ostaje približno isto, dok za signale reda veličine MHz pojačanje se drastično smanjuje.



Slika 8. Pojačanje signala zavisno od frekvencije ulaznog signala f_{IN}.

D. Promena pojačanja sa promenom otpornosti u emitoru

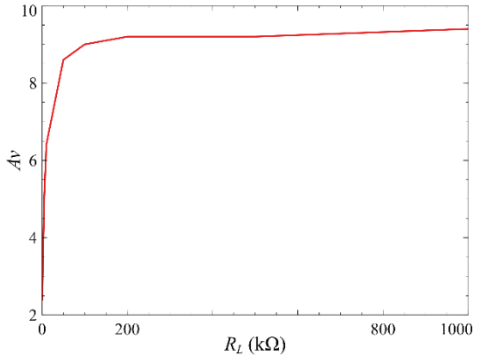
Pri promeni otpornosti potenciometra, koji je na slici 1 predstavljen otpornicima R_4 /R_5 (slika 9), tako da se deo otpornosti potenciometra označen sa R4 menja od 100 Ω do 900 Ω, određena je vrednost naponskog pojačanja. Može se uočiti da se za male vrednosti dobija veliko pojačanje, dok pri povećanju vrednosti te otpornosti dolazi do značajnog smanjenja vrednosti pojačanja.



Slika 9. Promena pojačanje signala u odnosu na otpornost R_4.

E. Promena pojačanja sa promenom otpornosti opterećenja

Pri priključivanju dekadne kutije na izlaz pojačavača radi testiranja rada pod opterećenjem (slika 10) može se videti da za male vrednosti otpornosti tog opterećenja pojačanje je malih vrednosti, ali kako se povećava otpornost opterećenja uočava se da je za vrednost od 100 kΩ pojačanje je dostiglo vrednosti od ~9 puta.



Slika 10. Promena pojačanja u odnosu na otpornost opterećenja.

IV. ZAKLJUČAK

Bipolarni tranzistor ima veoma značajnu ulogu u elektronici i mikroetronici zbog svojih karakteristika kao pojačavača i prekidača. Podešavanjem svih elemenata u kolu može se dobiti željena vrednost pojačanja u izradi pojačavača malih signala. Na primeru datom u ovom radu urađena je analiza kola korišćenjem realnog tranzistora i simulacijom. Pokazano je da za veće vrednosti ulaznog signala, usled ulaska tranzistora u oblast zasićenja ili zakočenja, dolazi do izobličenja signala. Pojačanje opada sa porastom frekvencije ulaznog signala jer tranzistor ne stiže da prati promene tog signala. Otpornost u kolu emitora se prespaja kondenzatorom sa ciljem temperaturne stabilnosti

vrednosti pojačanja. Što je veća vrednost ne prespojenog dela, pojačanje je manje. Pojačanje pojačavača raste sa porastom vrednosti opterećenja.

ZAHVALNICA

Autor se zahvaljuje Katedri za mikroelektroniku na pruženoj pomoći, a posebno asistentu Milošu Marjanoviću na mentorstvu i sugestijama u izradi ovog projekta.

LITERATURA

[1] A. Prijić, D. Danković, Z. Prijić, *„Uvod u poluprovodničke komponente i njihovu primenu”*, Elektronski fakultet Niš, 2020. Dostupno na: https://mikro.elfak.ni.ac.rs/wp-content/uploads/Poluprovodnici-CD-1.pdf.

[2] LTspice simulator. Dostupno na: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html.

[3] 2N3904 Datasheet. Dostupno na: https://www.onsemi.com/pdf/datasheet/2n3903-d.

Primena kamere u autoindustriji

Veljko Radosavljević, Ksenija Petrović

Sadržaj - Život modernog čoveka u savremenom svetu nije moguće zamisliti bez motornog vozila. Prevozna sredstva postala su simbol vremena i savremenog života. Pojava automobila dovela je do enormnih promena u saobraćaju. Od identifikacije korisnika pa do samih sistema provere i računanja brzine, tehnologija savremenih vozila razvija se svakodnebno. Rezultat mnogih projekata upravo je najbolji pokazatelj inovacija ovog sektora.

I. Uvod

Pored velikih prednosti u brzini, komforu i lakšem savladavanju kako kratkih tako i dužih rastojanja, putnicima, ali i samim vozačima, potreban je istaknut osećaj bezbednosti i sigurnosti.

Ova težnja izazvala je rast i razvoj naprednijih tehnologija unutar sistema bezbednosti, provere, i održavanja sigurnosti u toku kretanja ali i tokom mirovanja vozila. Od bezbednosnih pojaseva do sistema autonomne vožnje, praćenja i indentifikovanja putnika, inovacije ovakvog tipa dovele su do revolucije automobilske industrije na različitim nivoima [1], [2].

II. Tehnologija i sistem za "Prepoznavanje lica"

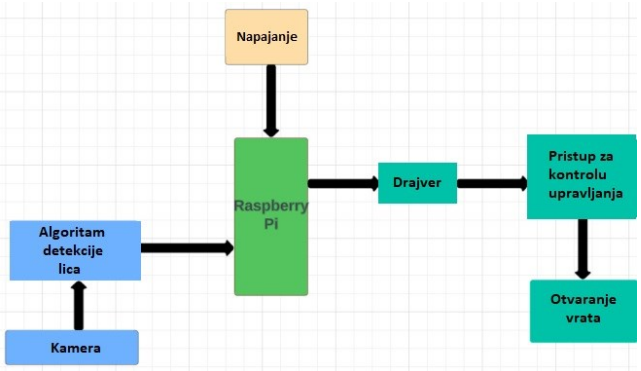
"Prepoznavanje lica" je tehnologija koja omogućava identifikaciju ili verifikaciju identiteta osoba na osnovu njihovih karakterističnih crta lica. Ova tehnologija se koristi u različitim aplikacijama, uključujući sigurnost, upravljanje identitetom, praćenje prisustva i još mnogo toga.

Osnovni koraci u procesu prepoznavanja lica uključuju sledeće:

- Detekcija lica: Prvi korak je detekcija lica na slici ili u video snimku. Algoritmi za detekciju lica pretražuju sliku i označavaju regione koji sadrže lica.
- Izdvajanje karakteristika: Nakon što je izvršena detekcija lica, karakteristike lica se izdvajaju iz slike. To uključuje analizu oblika, veličine i položaja karakterističnih tačaka kao što su nos, oči, usta i drugi detalji.
- Poređenje sa referencama: Izdvojene karakteristike se zatim upoređuju sa referentnim podacima ili šablonima koji su unapred registrovani u sistemu.
- Donošenje odluke: Na osnovu rezultata poređenja, sistem za prepoznavanje lica donosi odluku o identitetu osobe. Ako karakteristike lica odgovaraju nekom registrovanom identitetu sa dovoljno visokom verovatnoćom, osoba se prepoznaje.
- Odgovor sistema: Na kraju, sistem može izdati odgovor, na primer, o imenu osobe koja je prepoznata ili može preduzeti određene akcije, kao što je otvaranje vrata ili davanje pristupa sistemu.

„Prepoznavanje lica" koristi napredne tehnike obrade slike i računarskog vida, uključujući veštačku inteligenciju i duboko učenje.

Na slici je prikazana hardverska realizacija sistema za nadzor i detekciju osoba unutar vozila. Za srce sistema je iskorišćen jedan od najpopularnijih razvojnih okruženja Raspberry Pi 4 [3], [4].



Slika 1. Sistem za nadzor i detekciju osoba unutar vozila.

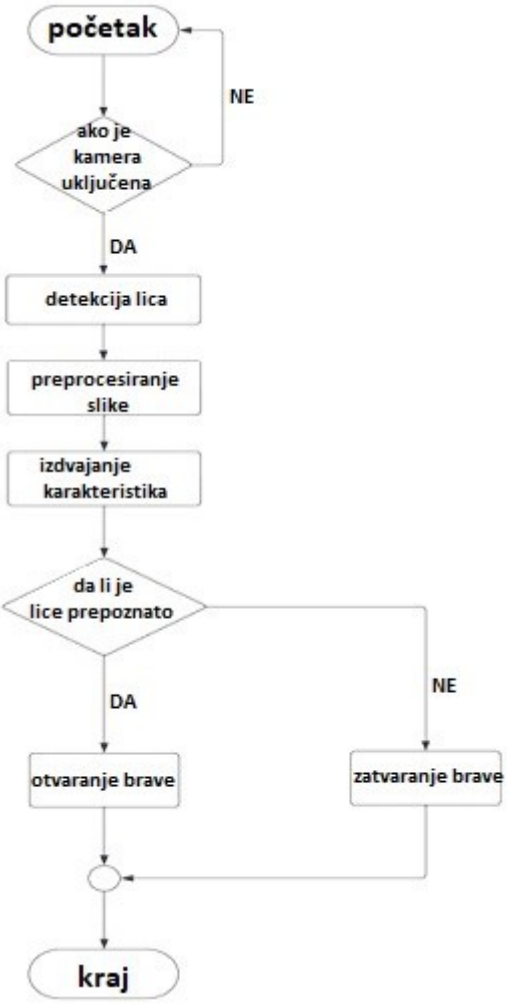
III. Implementacija sistema za "Prepoznavanje lica"

Algoritam prikazan na slici 2, koji se koristi pri implementaciji projekta poseduje sledeće bitne korake:

- Utvrđivanje da je priključena kamera uključena i osposobljena za rad, ako imamo pozitivan odgovor prelazimo u sledeću fazu, ako nije onda čekamo i ispitujemo ponovo.
- Sledeći korak bio bi detekcija lica, ako dobijemo pozitivan odgovor da je kamera uključena pristupamo ovom koraku, nakon čega automatski ulazimo u sledeću fazu.
- Preprocesiranje slike, kao naredna faza, obrađuje sliku i primenjuje različite tehnike pri transformisanju, u ovom slučaju u pitanju je LBPH (*Local Binary Pattern Histogram*).
- Kada se obradi u potpunosti slika, detektuju detalji ulazimo u sledeću fazu, izdvajanje karakteristika, gde se grupišu upečatljivi pikseli i dalje upoređuju sa slikama iz zadatih baza podataka
- Ako je lice detektovano i prepoznato u

V. Radosavljević i K. Petrović su sa modula Elektronika, Elektronskog fakulteta u Nišu, Srbija, E-mail: veljko.radosavljevic@elfak.rs; ksenija.keka@elfak.rs

specificiranoj bazi slika aktiviramo hardverski sklop priključen na sistem, npr. otvaramo bravu ili je zatvaramo [5].



Slika 2. Algoritam za implementaciju sistema za prepoznavanje lica.

IV. Rezultati i analize sistema za "Prepoznavanje lica"

A. Visoka tačnost prepoznavanja

Ovo je postignuto upotrebom metode LBPH (*Local Binary Pattern Histogram*) za prepoznavanje lica, što je omogućilo dobro prepoznavanje čak i uz varijacije u osvetljenju i pozama.

B. Prosečno vreme prepoznavanja

U okviru prihvatljivih vremenskih ograničenja za praktičnu primenu. Ključna za efikasnost sistema i zadovoljava potrebe.

C. Brzina otvaranja vrata

Nakon uspešnog prepoznavanja lica, brzina otvaranja sigurnosnih vrata bila je zadovoljavajuća.

V. Metodologija VASCAR-a za određivanje brzine objekta

Računar prosečne brzine (VASCAR – *Visual Average Speed Computer And Recorder*) je tehnološki uređaj koji koristi računarski program za merenje brzine.

Ova metodologija se zasniva na tome koliko je vremena potrebno da vozilo putuje između dve fiksno postavljene tačke. To radi tako što pređenu udaljenost podeli sa vremenom koje je bilo potrebno da se pređe to rastojanje, što daje prosečnu brzinu vožnje tokom tog rastojanja (slika 3).

Ovi uređaji se često nalaze u policijskim patrolnim automobilima, a s obzirom da ne koriste elektromagnetne talase, radar detektori ih ne mogu otkriti [6].



Slika 3. Metodologija VASCAR-a.

Postoje dve mogućnosti za realizaciju ovakvog sistema, a to su:

a) Računar ima već zadatu razdaljinu unetu u sistem na osnovu koje startuje samostalno TAJMER (*TIMER*) i nakon prelaska te putanje, isto tako automatski zaustavlja TAJMER (*TIMER*).

b) Računar nema zadatu razdaljinu, što iziskuje da ljudski faktor bude taj koji će startovati TAJMER (*TIMER*) i uneti DISTANCU (*DISTANCE*).

VI. Realizacija i implementacija sistema za određivanje brzine

Idejno rešenje projekta sistema za određivanje brzine vozila zasnovano je na implementaciji jedinstvenog sklopa programa koji koriste softvere za obradu slika (*Image Processing*) i kompjutersku viziju (*Computer Vision*) pri proračunima. Algoritam za realizaciju i implementaciju projekta podeljen je na nekoliko neizostavnih faza sa pojedinačnim zadacima za izvršenje:

I faza: Prikupljanje podataka - snimanje video snimaka, tj. hvatanje slika (*Video/Picture Input*).

II faza: Mikroračunar RPi procesira slike - izdvajanje frejmova i dalje ih obrađuje (*Raspberry Pi Input*).

III faza: Vrši se maskiranje frejmova kako bi se dalje mogli obrađivati segmenti (*Masking*).

IV faza: Zatim se izvučeni region centrira da se dobije tačan položaj vozila u različitim vremenskim periodama (*Centre*).

V faza: Računa se brzina i slika se ona slika koja se dalje čuva na veb serveru razvojnog sistema kako bi se moglo pristupiti slikama bilo kada.

Algoritam za realizaciju sistema za određivanje brzine vozila, kao i način računanja brzine, prikazani su na slici 4.

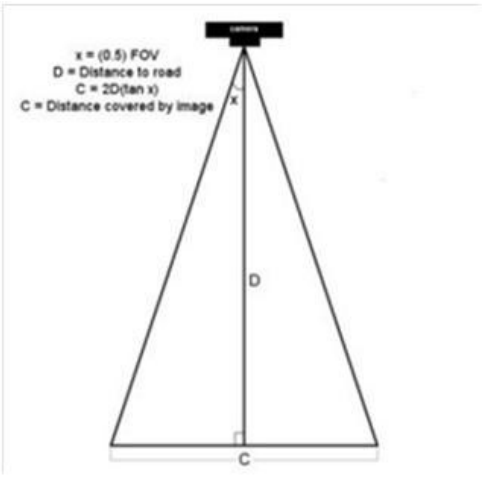


Slika 4. Algoritam za implementaciju sistema za određivanje brzine.

Pomoću trigonometrijskih formula, kao što je prikazano na slici 5, može se relativno precizno odrediti svaki parametar koji je neophodan za izračunavanje brzine objekta. Korišćene oznake su sledeće

- FOV (*Field Of View*) – opseg vidnog polja kamere koji zadaje proizvođač
- D (*Distance*) – razdaljina koju korisnik definiše
- X (polovina ugla) – polovina ugla kojom kamera zahvata vidno polje
- C (Horizontalna distanca) – računa se preko jednačine 2 * D * tan(x), čime se dobija slična vrednost opsega vidnog polja kamere

Kompletan algoritam na osnovu koga je realizovan model VASCAR-a prikazan je na slici 6.



Slika 5. Određivanje parametara neophodnih za izračunavanje brzine objekta.

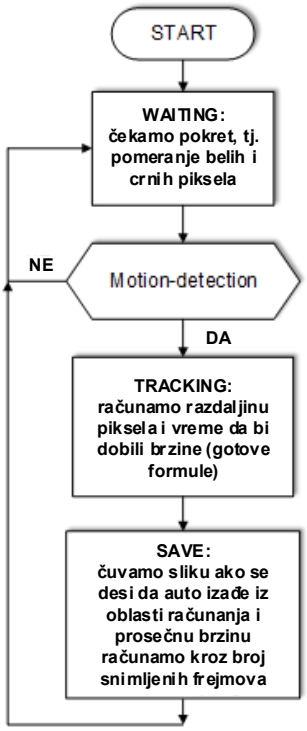


Slika 6. Algoritam modela VASCAR.

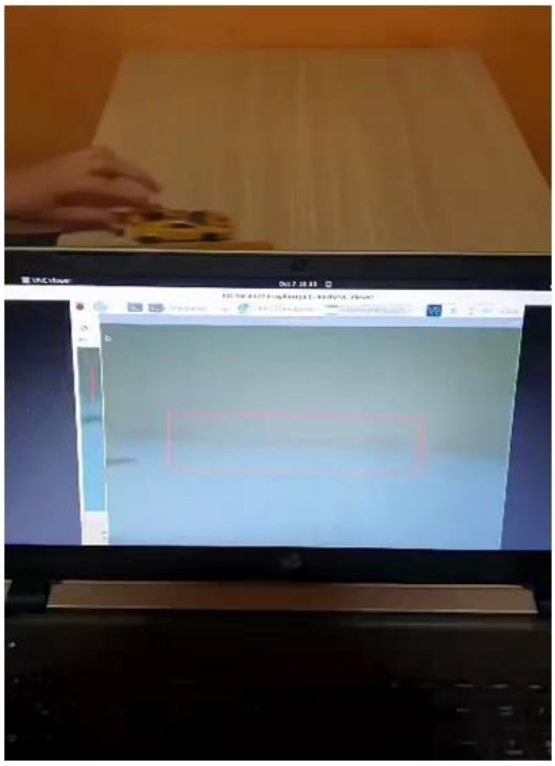
Pomoću jednostavnije verzije konačnog automata sa opcionim brojem koraka koji se uvek može nadograditi, realizovan je sistem za računanje prosečne brzine automobila na putevima, kao što je prikazano na slici 7. Kroz sliku je objašnjena funkcionalnost svakog od slučajeva u kome se može naći program. Naravno i u ovom slučaju je za osnovu projekta uzet najpopularniji mikroračunar Raspberry Pi 4 sa 2GB RAM-a [7], [8].

VII. REZULTAT RAČUNANJA BRZINE U SISTEMU

Na slici 8 je prikazano radno okruženje u kome je izvršena simulacija mrenja brzine automobila. Relativno korektan proračun od 2.6 m/s dobijen je implementacijom ovakvog sistema u kućnim uslovima. Pametno praćenje promene brzine automobila i bezbednija vožnja postaju sve veći izazov i pokreću brzo učenje i prilagođavanje glomaznijim i zahtevnijim sistemima koji su ne samo softverski već i hardverski složeniji.



Slika 7. Konačni automat sistema za računanje prosečne brzine automobila.



Slika 8. Simulacija merenja brzine automobila.

VIII. ZAKLJUČAK

Sigurnost pri vožnji zauvek će ostati oblast sabraćaja koja će se razvijati u različitim smerovima, ne samo pri zaštiti putnika, već i pasivnih osmatralaca i samih vozača čije odluke mogu biti presudne.

U ovom radu su razvijeni i implementirani sistemi za prepoznavanje lica i merenje brzine u realnom vremenu pomoću Raspberry Pi platforme i kamere. Istaživanje i sam projekat izrađen je u svrhe napredovanja svesti čovečanstvu o tome koliko je u stvari bitno biti siguran, bezbedan i apsolutno obezbeđen u vozilu, pri kretanju ili mirovanju.

Na kraju ovog istraživanja, može se doći do zaključka da sa napretkom tehnologije i sistemima zasnovanim na procesorima i kamerama primenjivim u svim vidovima saobraćaja, budućnost koja je pred nama sve više biva svetlija i bezbednija. Svojstveno, pomak u sferi automobilizma se svakodnevno pomera korak po korak, i nikada neće stati, sve dok se barem jedan točak okreće.

LITERATURA

[1] IoT Design Pro, "Face Recognition Door Lock System using Raspberry Pi", 2020, https://iotdesignpro.com/projects/face-recognition-door-lock-system-using-raspberry-pi

[2] Nwukor Frances Nkem, "Face Recognition Door Lock System Using Raspberry Pi", Global Scientific Journals, GSJ, Volume 10, Issue 8, August 2022

[3] Muhammad Aqib, "How to Create a Facial Recognition Door Lock With Raspberry Pi", 2019, https://maker.pro/raspberry-pi/projects/how-to-create-a-facial-recognition-door-lock-with-raspberry-pi

[4] K. V. Usha Ramani, Imroze Jahan, Mr. P. Anvesh, "Face Recognition Door Lock System using Raspberry Pi", Intern. Journal for Research in Appl. Sci.& Eng. Techn. (IJRASET), Vol. 9, Issue VI, 2021

[5] Adrian Rosebrock, "What is face recognition?", 2021, https://pyimagesearch.com/2021/05/01/what-is-face-recognition/

[6] Nave Law Firm, "Vehicle Speed Measurement Methods", https://www.nysdwi.com/dui-foundation/drunkdriving/trafficviolations/speedmeasurement/, 2023

[7] D. Dharini, R. Srividhya, „Vehicle Speed Detection using Raspberry Pi", Intern. Journal of Innov. Techn. and Explor. Eng. (IJITEE), Vol. 8, Issue 11, 2019

[8] M. A. Jahangir, et al, „A Secure Traffic Police Remote Sensing Approach via a Deep Learning-Based Low-Altitude Vehicle Speed Detector through UAVs in Smart Cites: Algorithm, Implementation and Evaluation", Future Transportation, vol. 3, no. 1: 189-209, 2023

Izlazni pojačavač za FM radio

Katarina Stanisavljević, Uroš Jović

Sadržaj - Rad se bavi projektovanjem i realizacijom izlaznog pojavača za stereo FM radio.

I. UVOD

Zadatak ovog projekta je realizacija izlaznog pojačavača za stereo FM radio.

Pojačavač je uređaj koji povećava amplitudu neke fizičke veličine, obično napona ili struje. Termin se može koristiti i za pojačavače sile, pritiska, i drugih veličina. Svima je zajedničko da imaju određeni propusni opseg i pojačanje. Obično postoji kompromis između propusnog opsega i pojačanja, zbog praktičnih osobina komponenti, i zbog nastojanja da se šum svede na minimum. Pojačivači audio snage su prisutni u svim vrstama zvučnih sistema, uključujući pojačanje zvuka, sisteme za razglas i kućne audio sisteme i pojačavača za muzičke instrumente. To je poslednja elektronska faza u tipičnom lancu za reprodukciju zvuka pre nego što se signal pošalje na zvučnike.

Pojačavači snage se klasifikuju prema tome koliki deo periode čini vremenski interval u toku koga tranzistor vodi, odnosno pojačava signal. Najšire zastupljeni pojačavači snage su pojačavači u klasama: A, B, AB, C. Pojačavači u klasama A, B i su širokopojasni pojačavači dok klasa C predstavlja selektivni pojačavač.

A. Klasa A

Ovde se koristi 100% ulaznog signala (ugao provodnosti $\theta=2\pi$ ili 360º, tj, aktivni element radi u svom lineranom rangu čitavo vreme). Tamo gde se ne razmatra efikasnost, većina linearnih pojačavača malih signala definisana je kao klasa A, što znači da su izlazni uređaji uvek u provodnoj oblasti. Klasa A pojačavača se najčešće koristi u fazama malog dizajna ili za niskoprotočne aplikacije (poput slušalica za korišćenje tokom vožnje). Za ovaj rad se koristila klasa A jer je najjeftinija, efikasna i ovaj pojačavač radi na većim naponima. Najveća moguća vrednost stepena iskorišćenja pojačavača snage koji rade u klasi A dobija se kada je najveća amplituda ulaznog signala.

Pojačavače snage u klasi A karakteriše:

-Vrlo mala izobličenja

- Velika disipacija snage na aktivnom elementu (idealno 50% u najpovoljnijem realnom slučaju oko 60% od ukupne uložene snage).

- Koriste se za relativno male snage do 1W

K.Stanisavljević i U. Jović su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14,18000 Niš, Srbija (e-mail:urosjovic@elfak.rs, katarina.stanisavljevic74@gmail.com)

B. Klasa B

Koristi se 50% ulaznog signala ($\theta=180$ º ili π, tj, aktivni element radi u svom linearnom opsegu pola vremena i u drugoj polovini uređaj je manje ili više vremena isključen). U većem delu Klase B postoje dva izlazna uređaja od kojih svaki alternativno vrši provođenje (push-pull) za egzaktno 180 stepeni (ili pola ciklusa) ulaznog signala, a selektivni RF pojačavači takođe mogu biti implementirani korišćenjem pojedinačnog aktivnog elementa.

Ovi pojačavači su izloženi poprečnoj distorziji ako predaja od jednog aktivnog elementa na drugi nije perfektna, pošto su dva komplementarna tranzistora (tj. jedan PNP, jedan NPN) povezana kao dva emitora sa njihovim bazama i zajedničkim emiterskim terminalom, što zahteva promenu osnovnog napona u oblasti gde su oba uređaja isključena. Kada je signal pozitivan, vodi tranzistor QN (NPN tipa) i njegova izlazna struja teče preko otpornika Rp. Tranzistor QP (PNP tipa) je zakočen.

C. Klasa AB

Pojačavač klase AB karakteriše

• manja korisna snaga

• manji stepen iskorišćenja

• manja izobličenja nego pojačavaču klasi B.

Klasa AB je po karakteristikama praktično kompromis između klase A i klase B.

• Korisna snaga u klasi AB manja je nego u klasi B jer je redukovano dinamičko područje promene ulaznog, a time i izlaznog signala.

• Stepen iskorišćenja u klasi AB manji je nego u klasi B, jer teče jednosmerna struja i u odsustvu ulaznog signala, tako da uvek postoji disipacija na tranzistoru.

• Široka primena u audio pojačavačima.

D. Klasa C

Pojačavač u klasi C se može koristiti za pojačanje signala sa uskim frekvencijskim opsegom i stalne amplitude. Primenjuju se za pojačanje frekvencijski modulisanog signala. Pobudni signal treba da ima veću amplitudu jer je kod ovih pojačavača značajno umanjen i osnovni harmonik.

II. KOMPONENTE I KARAKTERISTIKE PROTOTIPA

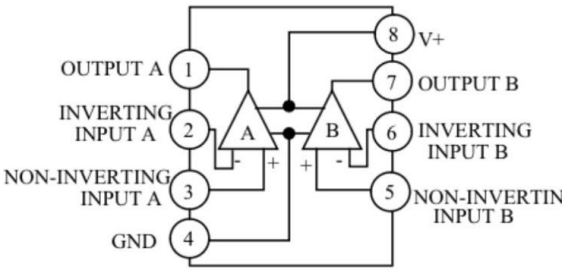
Prototip sadrži sledeće komponente:

1. Na ulazu se nalazi kondenzator koji predstavlja model parazitne ulazne kapacitivnosti
2. Slede otpornici.
3. Integrisano kolo
4. NPN tranzistor
5. Potenciometar

Otpornici: Otpornici su obično napravljeni od materijala sa visokom otpornošću, kao što su ugljenik, metal ili keramika. Dolaze u različitim oblicima i veličinama, a karakteriše ih vrednost otpora, koja se meri u omima (Ω). Vrednost otpora otpornika određuje koliko će smanjiti protok struje u kolu. Korišćeni su otpornici jačine od 270 Ω, 22 Ω i 1k Ω.

Kondenzator: Kondenzator je pasivna električna komponenta koja skladišti energiju u električnom polju između dve provodne ploče razdvojene dielektričnim materijalom. Kondenzatori dolaze u različitim oblicima i veličinama i koriste se u širokom spektru elektronskih kola. Obično se koriste za filtriranje šuma i neželjenih signala iz kola, za skladištenje i pražnjenje energije i za stabilizaciju nivoa napona u kolu. Kapacitet kondenzatora se meri u faradima (F) i određuje se veličinom i razmakom ploča, kao i dielektričnom konstantom materijala između ploča. Kondenzatori se takođe mogu okarakterisati njihovom ocenom napona, koja određuje maksimalni napon koji kondenzator može da izdrži pre nego što se pokvari. Postoji mnogo različitih tipova kondenzatora, uključujući keramičke, elektrolitičke, tantalske i filmske kondenzatore, svaki sa svojim jedinstvenim svojstvima i primenom. Kondenzatore koje smo koristili su jacine 10µF i 100 µF.

LM358N se sastoji od dva nezavisna operaciona pojačavača s visokim pojačanjem, interno kompenzovanom frekvencijom, posebno dizajniranom da radi iz jednog izvora napajanja u širokom opsegu napona. Odvod napajanja male snage je nezavisan od veličine napona napajanja.



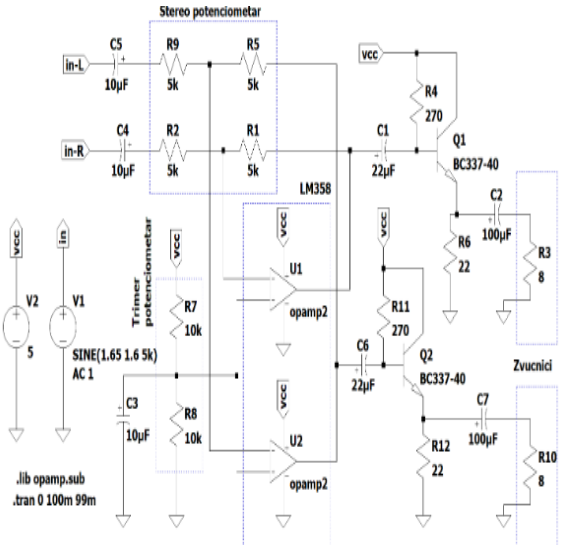
Slika 1. LM358.

NPN tranzistor: NPN tranzistor je tip tranzistora bipolarnog spoja koji ima tri dopirana poluprovodnička regiona: tanki bazni region p-tipa između dva regiona n-tipa, emitora i kolektora. "N" u "NPN" se odnosi na činjenicu da su većinski nosioci u oblastima emitora i kolektora negativno naelektrisani elektroni. NPN tranzistor se obično koristi kao elektronski prekidač ili pojačavač u kolima. Kada se na bazu primeni dovoljna količina struje, veća struja teče od kolektora do emitora, omogućavajući tranzistoru da deluje kao prekidač za kontrolu drugih komponenti u kolu. U aplikaciji za pojačavanje, mali signal primenjen na bazu tranzistora može se pojačati u veći signal na kolektoru, čineći NPN tranzistor korisnim za pojačavanje signala male snage u elektronskim uređajima.

BC237 je NPN BJT tranzistor opšte namene koji se može koristiti za prebacivanje kao i za pojačanje. Pošto je maksimalna struja kolektora tranzistora 100 mA, može se koristiti za pokretanje i kontrolu velikog broja opterećenja u kolima sa manje od 100 mA. Maksimalna disipacija kolektora od 500mV i maksimalno pojačanje jednosmerne struje do 800, sa druge strane, čine ga odličnim za audio i drugo elektronsko pojačanje signala.

Potenciometar: Potenciometar je vrsta promenljivog otpornika koji se može koristiti za podešavanje otpora u kolu. Potenciometar se obično sastoji od otpornog elementa, kao što je namotaj žice ili film od ugljenika ili metala, sa pokretnim kontaktom ili brisačem koji se može podesiti po dužini elementa. Kako se brisač pomera, otpor između dva terminala potenciometra se menja, omogućavajući mu da funkcioniše kao promenljivi delilac napona. Potenciometri se obično koriste u elektronskim kolima za kontrolu nivoa signala ili za podešavanje pojačanja pojačavača. Takođe se koriste u kontrolnim tablama i drugim korisničkim interfejsima za podešavanje podešavanja kao što su jačina zvuka, osvetljenost i temperatura. Potenciometri dolaze u različitim veličinama i stilovima, u rasponu od malih trim lonaca koji se koriste za fina podešavanja do velikih lonaca montiranih na panelu koji se koriste za industrijsku primenu. Oni su suštinska komponenta u mnogim elektronskim kolima i široko se koriste u raznim aplikacijama.

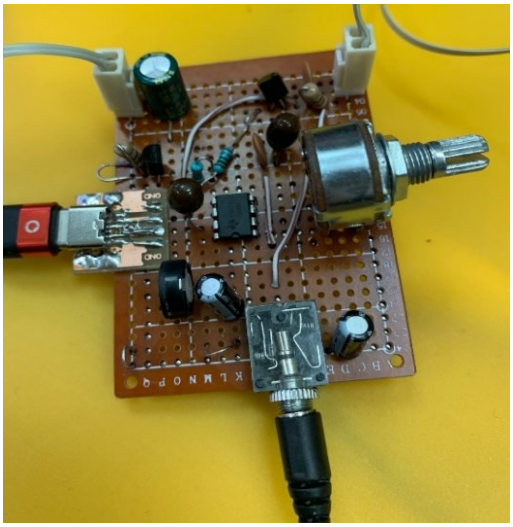
III. SIMULACIJA I PPRAKTIČNA REALIZACIJA



Slika 2. Šema kola.

Simulacija kola izvedena je u grafičkom simulacionom okruženju LT Spice. Kompletna šema koja je projektovana prikazana je na slici 1. Na osnovu šeme se vidi da se na ulazu kola nalaze kondenzatori, zatim slede otpornici, dalje ide operacioni pojačavač koji može da pojača i manipuliše ulaznim signalom na različite načine. Nakon toga se nalazi NPN tranzistor i na izlazu zvučnik.

Sa ovako definisanim sistemom, rezultati simulacije su prikazani na slici. Svi aktivni elementi u kolu su smatrani idealnim, što znači da imaju beskonačnu ulaznu otpornost i beskonačan propusni opseg.



Slika 3. Izgled prototipa na raster ploči.

IV. Rezultati merenja

Nakon realizacije pristupa se testiranju prototipa, tako što se meri jačina frekvencije zvuka prikazano u tabeli 1.

Frekvencija je mera broja ciklusa periodičnog talasnog oblika koji se javljaju u datom vremenskom periodu. Jedinica frekvencije je Herc (Hz), koja se definiše kao jedan ciklus u sekundi. U kontekstu zvučnih talasa, frekvencija određuje visinu zvuka. Više frekvencije odgovaraju višim visinama, dok niže frekvencije odgovaraju nižim visinama. Na primer, zvižduk visokog tona proizvodi zvučne talase visoke frekvencije, dok bas gitara niskog tona proizvodi zvučne talase niske frekvencije. Na primer, radio talasi, mikrotalasi, infracrveno zračenje, vidljiva svetlost, ultraljubičasto zračenje, rendgenski zraci i gama zraci imaju različite frekvencije i talasne dužine, što određuje njihova svojstva i primenu. Frekvencija je važan koncept u mnogim oblastima, uključujući fiziku, inženjering i telekomunikacije. Koristi se u raznim aplikacijama, kao što su radio komunikacija i obrada signala. Opseg frekvencija sa kojima smo merili zvuk je od 20 Hz do 22kHz.

Kod ovakvog prototipa najniža frekvencija koja se čuje koju zvučnik reprodukuje je 200Hz a najvisa je 13kHz. Primećujemo da Uout1 raste do vrednosti od otprilike 1000 mV, nakon čega opada, Uout2 raste do vrednosti od otprilike 300 mV, nakon čega opada dok je Uin konstantno.

TABELA I
MERENJE FREKVENCIJE

F[Hz]	Uin[mV]	Uout1[mV]	Uout2[mV]
20	544	168	40
40	544	208	56
70	544	280	64
153	544	472	80
188	544	536	96
221	544	584	112
344	544	736	152
645	544	880	200
982	544	936	176
1501	544	960	184
2615	544	984	200
5009	544	1020	224
8037	544	1030	248
12052	544	1050	288
15368	544	544	320
17931	544	96	160
18576	544	88	152
22000	544	72	80

V. Zaključak

Zadatak projekta bio je da se realizuje izlazni pojačavač za stereo FM radio. Izvršena je analiza zadatka i obavljeno je projektovanje prototipa čija je validnost ispitana korišćenjem simulatora. Prototip je zatim praktično realizovan korišćenjem raster ploče i komponenti, i njegovo funkcionisanje je potvrđeno konkretnim merenjima na multimetru.

U radu je utvrdjeno je da se zvuk čuje tek na frekvenciji od 200 Hz pa sve do 13 kHz nakon čega zvuk postaje nečujan za ljudsko uho.

Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta na ispitu Telekomunikacije, na smeru Upravljanje sistemima, Elektronskog fakulteta Univerziteta u Nišu. Ispit je na 5. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

Literatura

[1] Ben Duncan, “High Performance Audio Power Amplifiers for music performance and reproduction”, Newnes, 1996. https://jntukucen.ac.in/ebook_files/135.pdf

[2] Kataloški podaci integrisanog kola LM328N. https://pdf1.alldatasheet.com/datasheet-pdf/view/447322/TGS/LM358N.

[3] Kataloški podaci tranzistora BC237B. https://www.utmel.com/components/bc237-transistor-datasheet-pdf-pinout-bc547-vs-bc237?id=1096#cat2

[4] Douglas Self, “Audio Power Amplifier Design Handbook”, fifth edition, Elsevier Ltd. 2009. https://d1.amobbs.com/bbs_upload782111/files_29/ourdev_554203.pdf

[5] Robert Lacoste, “Amplifier Classes From A to H”, Circuit cellar, 2013. https://circuitcellar.com/wp-content/uploads/2019/10/2013-12-015-Lacoste.pdf

[6] Jingjie Sun, Yingjun Chen, “Audio Power Amplifier Design”, Faculty of engineering and sustainable development, 2011. http://hig.diva-portal.org/smash/get/diva2:429678/FULLTEXT01

Realizacija jednostavnog optičkog skenera

Ana Cvetanović, Jana Milošević, Natalija Deljanin, Marko Đorđević, Milan Stojanović

Sadržaj – Cilj ovog rada je da se opiše realizacija uređaja koji će detektovati ispisani tekst na listu papira i taj tekst dalje prikazati na ekranu računara. Najpre je prikazana praktična realizacija optičkog detektora, koji je dalje uz primenu Arduino UNO okruženja iskorišćen za detekciju crnih slova na belom papiru i ispisivanje tih slova na ekranu računara. Sistem za detekciju i ispisivanje teksta sastoji se od LE dioda, fototranzistora, otpornika, komparatora, Arduino UNO razvojnog okruženja, dva plastična lenjira i raster ploče. Pokazali smo da realizovani sistem pomoću 16 piksela može uspešno da obavlja ovaj zadatak.

I. Uvod

Cilj telekomunikacija jeste da se informacija prenese iz jedne u drugu udaljenu tačku posredstvom elektromagnetnih talasa. Informacije koje se prenose, utiskuju se u signal, koji su zapravo nosilac informacija. Signali se zatim prenose kroz određeni medijum od predajnika do prijemnika. Ovaj medijum može biti bakarna parica, koaksijalni kabl, optičko vlakno, slobodni prostor itd. [1]. Svetlost se može koristiti kao nosilac informacija. Tada se radi o optičkim telekomunikacionim sistemima, gde se koristi nosilac visoke frekvencije (reda veličine 100 THz). Kao izvor svetlosti u optičkim telekomunikacionim sistemima mogu se koristiti LE diode (LED – *Light-Emitting Diode*) [2]. Optički signali mogu se koristiti i za zapis informacija na odgovarajući nosač informacija, kao i za skeniranje teksta ili slika.

Do ideje za realizaciju ovog projekta smo došli nakon odslušanih predavanja i zanimljivog rada u laboratoriji. Želeli smo da osmislimo nešto novo što bi bilo korisno i u skladu sa našim dosadašnjim znanjem. S obzirom na to da je skener, koji nam je bio inspiracija za rad, digitalni uređaj koji ima više piksela i čiji način rada nije nam bio poznat, kod detektora optičkih signala imamo manje piksela i njegov način rada nam je bio relativno poznat. Želeli smo da pojednostavimo realizaciju uređaja, što smo postigli smanjenjem broja piksela, a razlika u odnosu na skener je to što nemamo RGB (RGB - *Red Green Blue*) detekciju.

Sistem detektuje slova koja su ispisana na papiru na sledeći način: LE diode emituju svetlost koja prolazi kroz papir na kome su ispisana slova. Ispod papira se nalaze fototranzistori koji detektuju količinu svetlosti koja je prošla kroz papir. U zavisnosti od toga da li svetlost koju emituje LE dioda pada na beli deo papira ili na deo papira na kome je ispisano slovo, fototranzistori detektuju veću ili manju količinu svetlosti. Slika, koja je detektovana pomoću fototranzistora, ispisuje se na ekranu računara.

Rad je organizovan na sledeći način. U drugom delu rada kratko su opisane komponente koje su korišćene u radu. Princip realizacije sistema opisan je u trećem delu, dok su u četvrtom delu navedeni osnovni zaključci do kojih smo došli tokom rada. Na kraju je dat spisak korišćene literature.

A. Cvetanović i M. Đorđević su studenti na Katedri za mikroelektroniku, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anacvetanovic@elfak.rs; markodjordjevic@elfak.rs.

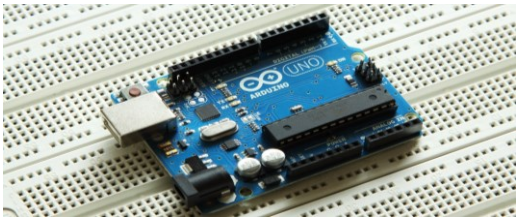
J. Milošević, N. Deljanin i M. Stojanović su studenti na Katedri za elektroniku, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: jana.milosevic@elfak.rs; natalija.deljanin@elfak.rs; milancestojanovic@elfak.rs..

II. Korišćene komponente

U ovom delu, dat je spisak komponenata korišćenih u radu, kao i njihov sažet opis. Osnovne komponente koje smo koristili su: ploča mikrokontrolera, integrisano kolo koje ima ulogu komparatora, LE dioda, fototranzistor i otpornici.

A. Arduino UNO

Arduino UNO (slika 1) je štampana ploča, koja se sastoji iz mikrokontrolera i prateće elektronike [3]. Sastoji se od 14 digitalnih ulazno/izlaznih pinova od kojih su 6 analognih ulaza, jedan pin je za USB priključak, a jedan priključak je za napajanje, jedan pin je dugme za resetovanje i jedan je kristal kvarca.



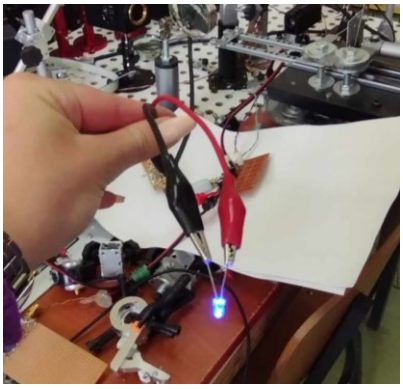
Slika 1. Arduiono UNO.

B. Komparator

Komparator je komponenta koja vrši upoređivanje dva ulazna napona i daje binarni signal sa informacijom koji od napona je veći. Ako je neinvertujući (+) ulaz veći od invertujućeg (-) onda je nivo izlaznog signala visok. Ukoliko je invertujući ulaz veći od neinvertujućeg, onda je naponski nivo na izlazu nizak.

C. LE diode

LE diode propuštaju struju u jednom smeru, a blokiraju u suprotnom. U zavisnosti od materijala od kog je napravljena, LE dioda će emitovati svetlost određene talasne dužine [4]. Funkcionisanje LE diode ilustrovano je na slici 2.



Slika 2. Ilustracija rada LE diode.

D. Fototranzistori

Fototranzistor (slika 3) je poluprovodnička komponenta kod koje baza detektuje svetlost i konvertuje je u struju koja prolazi kroz kolektor i emitor [5]. Kod ove vrste tranzistora imamo dva spoljašnja izvoda, jer je baza zatvorena u kućištu. Dakle, fototranzistor pretvara svetlosni u električni signal.



Slika 3. Fototranzistor.

E. Otpornici

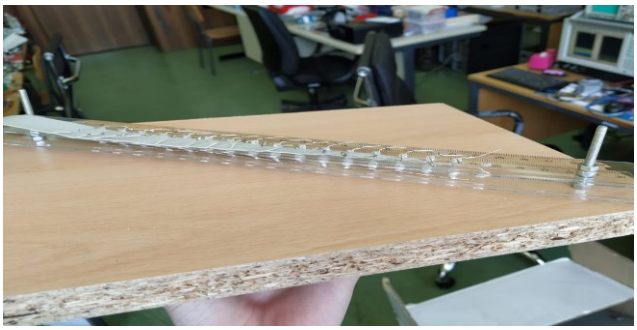
Otpornici su pasivne komponente koje vrše ograničenje protoka struje. U ovom projektu korišćeni su otpornici od 1 kΩ (*through-hole* montaža) i 39 0Ω (SMD - *Surface Mounted Device*).

III. PRINCIP REALIZACIJE

U ovom delu, dajemo detaljan opis realizacije sistema. Naglašavamo da je sistem realizovan od dostupnih komponenata na tržištu čija cena je bila vrlo pristupačna.

Za izradu ploča detektora koriste se plastični lenjiri (videti sliku 4). Na lenjir, koji predstavlja gornju ploču, postavljene su LE diode koje emituju svetlost ka papiru. Na papiru je ispisan tekst crnom bojom, koji se detektuje. Diode su povezane u konfiguraciji sa zajedničkom anodom. To znači da su anode povezane na napajanje, a katode su povezane na masu. Redno sa njima su povezani otpornici od 390 Ω da bi ograničili protok struje kroz diode.

Donji lenjir je pričvršćen za drvenu ploču da bi ceo sistem bio stabilan. Na donju ploču montirani su fototranzistori koji detektuju svetlost, koja prolazi kroz papir. Uloga fototranzistora je da detektuje svetlost na takav način da sistem utvrdi da li se detektuje svetli ili tamni deo papira.

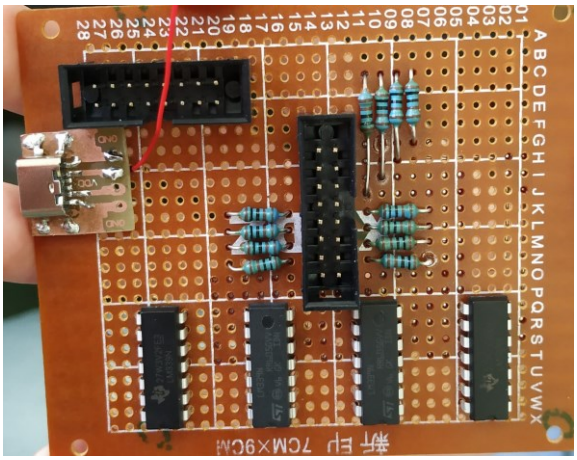


Slika 4. Osnovna konstrukcija sistema.

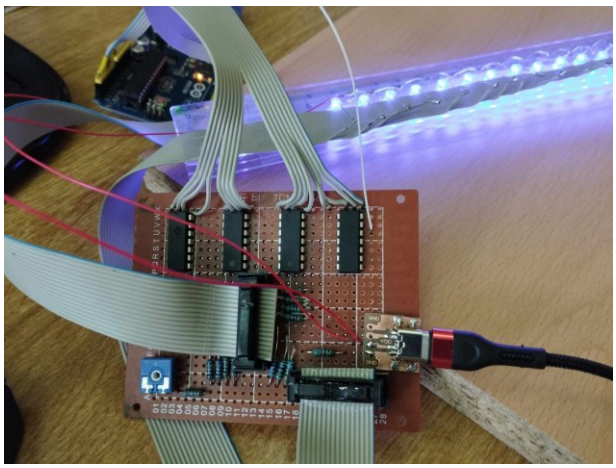
Kolektori su svi kratkospojeni i na njih se dovodi napajanje. Emitor svakog tranzistora i jedan kraj otpornika od 1 kΩ povezani su na neinvertujući ulaz komparatora [6], a drugi kraj otpornika je vezan na masu.

Komparatori su povezani na raster ploči, na kojoj imamo i otpornike i konektore.

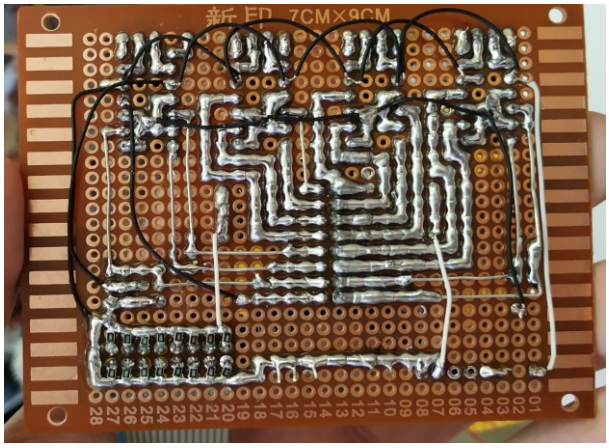
Izlazi komparatora su povezani sa digitalnim pinovima na Arduin-u. Arduino je preko USB porta povezan sa računarom. Na izlaze komparatora se dobija jedan od dva moguća nivoa signala koji se dalje prenose na Arduino nakon čega imamo prikaz na ekranu računara. Na slici 5 prikazana su integrisana kola komparatora na raster ploči.



Slika 5. Integrisana kola komparatora na raster ploči.

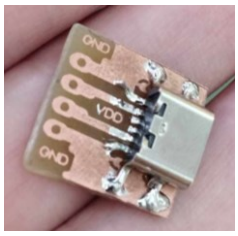


Slika 6. Raster ploča povezana sa ostalim delovima sistema.



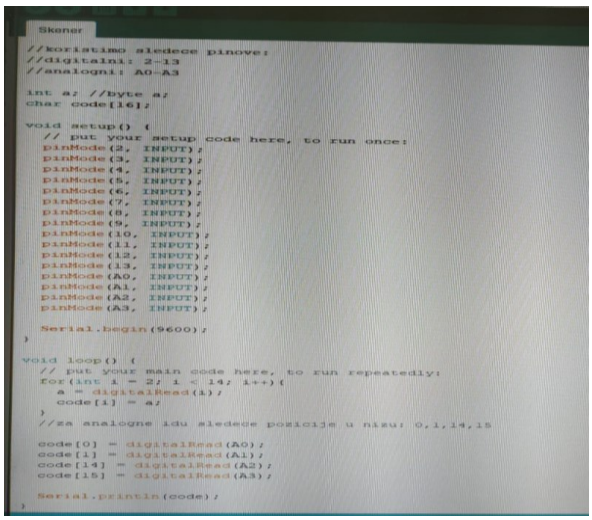
Slika 7. Prikaz lemova i veza na poleđini raster ploče.

Za napajanje dioda koristi se USB konektor koji daje napon od 5 V.



Slika 8. USB konektor.

Da bi se obezbedio prikaz na ekranu, neophodno je bilo napisati program za Arduino (slika 9) koji će to omogućiti na osnovu signala koji se dovode na pinove. Na osnovu programiranog koda sistem detektuje dovedeni nivo signala. Digitalni pinovi, koji se koriste, označeni su brojevima od 2 do 13, a analogni pinovi, koji će se vršiti funkciju digitalnih pinova, su od A0 do A3.



```
Skener

//koristimo sledece pinove:
//digitalni: 2-13
//analogni: A0-A3

int a; //byte a;
char code[16];

void setup() {
  // put your setup code here, to run once:
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
  pinMode(11, INPUT);
  pinMode(12, INPUT);
  pinMode(13, INPUT);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);

  Serial.begin(9600);
}

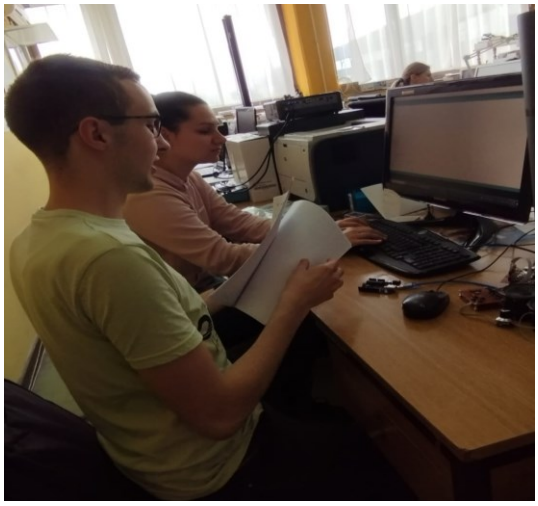
void loop() {
  // put your main code here, to run repeatedly:
  for(int i = 2; i < 14; i++){
    a = digitalRead(i);
    code[i] = a;
  }
  //za analogne idu sledece pozicije u nizu: 0,1,14,15

  code[0] = digitalRead(A0);
  code[1] = digitalRead(A1);
  code[14] = digitalRead(A2);
  code[15] = digitalRead(A3);

  Serial.println(code);
}
```

Slika 9. Arduino Source kod.

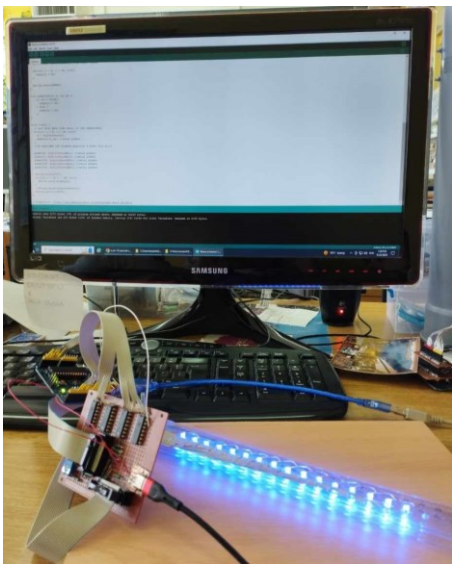
U kodu se prvo vrši deklaracija promenljive *a* koja predstavlja signal koji se dovodi (*low* ili *high*). Nakon toga se deklarišu ulazni pinovi od kojih su 12 digitalni i 4 analogna koji se ponašaju kao digitalni pinovi. Prolazi se kroz petlju koja vrši iščitavanje unetog signala. Vršimo deklaraciju analognih pinova koji su deklarisani da rade kao digitalni. Na kraju imamo ispis dobijenog signala. Vrši se iscrtavanje na ekranu računara.



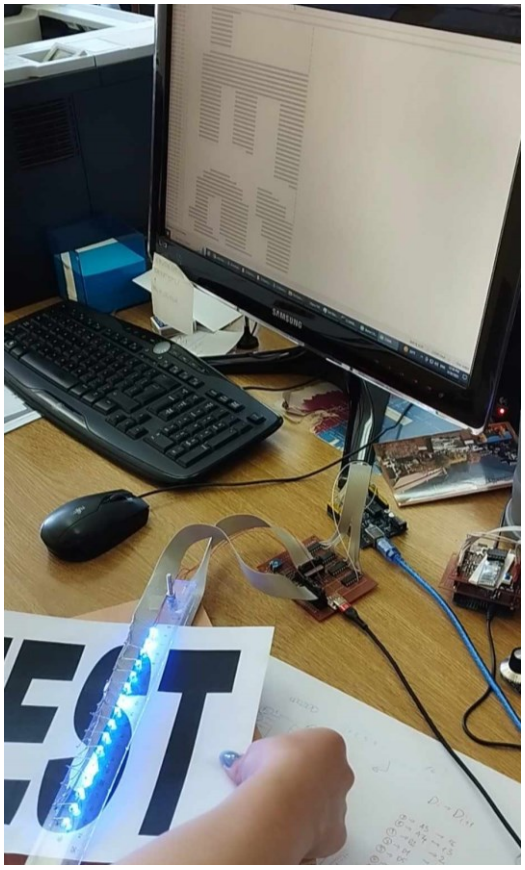
Slika 10. Detalj tokom realizacije projekta.

Kako su izlazi u konfiguraciji sa otvorenim kolektorom, neophodno je bilo da se u kodu pinovi mikrokontrolera postave kao ulazni pinovi. Neophodno je da se ti pinovi postave u konfiguraciji sa unutrašnjim *pull-up* otpornicima.

Na slici 11 prikazan je tok realizacije programskog koda za Arduino, a na slici 12 prikazani su detalji rada realizovanog sistema.



Slika 11. Pisanje Arduino programskog koda.



Slika 12. Demonstracija rada realizovanog sistema.

IV. Zaključak

Realizovali smo sistem za detekciju slova u crnoj boji na belom papiru i ispisivanje detektovanih slova na ekranu računara. Sistem je realizovan na raster ploči. Realizacija ovog projekta je uspešno izvršena i sve funkcioniše po pretpostavci koju smo imali pre početka izrade. U toku realizacije projekta, uverili smo se da svaka komponenta obavlja svoju funkciju. Naglašavamo da je sistem realizovan od komponenata koje mogu da se nađu po pristupačnoj ceni, i da je potpuno funkcionalan onako kako je opisano u prethodnom delu rada.

Zahvalnica

Zahvaljujemo se profesoru Goranu T. Đorđeviću koji nam je omogućio izradu ovog projekta. Podstakao nas je da razmišljamo kao pravi inženjeri. Predložio nam je rad u laboratoriji kao deo ispita i na taj način smo naučili puno korisnih stvari i da je moguća realizacija svega što naučimo. Uvideli smo da sve što smo do sada teorijski naučili, moguće da realizujemo u praksi i da sve to bude veoma interesantno. Posebna zahvalnica profesorima Dejanu Miliću i Nenadu Miloševiću, koji su nam takođe pomogli prilikom izrade projekta i podstakli nas da puno vremena provedemo u laboratoriji. Zahvaljujemo se Katedri za telekomunikacije Elektronskog fakulteta u Nišu u čijoj laboratoriji, broj 304, smo realizovali ovaj projekat.

Literatura

[1] G. T. Đorđević, *Predavanja iz predmeta Telekomunikacije*, 2022/23.
[2] D. N. Milić, *Optičke telekomunikacije*, Elektronski fakultet u Nišu, Niš, 2016.
[3] Tehnička specifikacija za Arduino UNO. Dostupno na: https://docs.arduino.cc/hardware/uno-rev3.
[4] Tehnička specifikacija za LE diodu. Dostupno na: https://www.mgelectronic.rs/ProductFilesDownload?Id=3969.
[5] Tehnička specifikacija za fototranzistor. Dostupno na: https://www.mgelectronic.rs/ProductFilesDownload?Id=8045.
[6] Tehnička specifikacija za integrisano kola komparatora. Dostupno na: https://www.mgelectronic.rs/ProductFilesDownload?Id=6445.
[7] Google search for "LED as light sensor".
[8] Make - How to use LEDs to detect light.
[9] Mitsubishi Electric Research Laboratories (www.merl.com) - Very Low-Cost Sensing and Communication Using Bidirectional LEDs.
[10] Google search for "photodiode light intensity stackexchange".
[11] Electronics.StackExchange.com - Photodiode circuit for measuring light intensity.
[12] Electronics.StackExchange.com - How do I connect a photodiode? - shows a voltage divider photoconductive mode application technique instead, which must be read with an analog to digital converter (ADC).
[13] Learning About Electronics - What is a Photodiode?
[14] Read how to measure light level through the ADC using an LED and an opamp here.

IoT sistem za nadzor košnica

Mihajlo Milovanović

Sadržaj - Nadgledanje i praćenje košnica pčelarima pružaju informacije koje im mogu pomoći pri upravljanju pčelinjim društvima. U ovom radu biće detaljno predstavljen razvoj IoT sistema za nadzor košnica koji je realizovan uz pomoć mikrokontrolera i različitih vrsta senzora. Upotrebom ovog sistema smanjuje se potreba za manuelnim pregledom košnice i omogućava se brže reagovanje pčelara u slučaju nepredvidjenih situacija.

I. UVOD

Pčele su najvažniji oprašivači na svetu. Više od trećine svetskih useva zavisi isključivo od njihovog oprašivanja. Kako se broj pčelinjih društava drastično smanjuje, proteklih godina sve više pažnje posvećuje se pitanju izumiranja pčela i opasnostima koje ono sa sobom nosi. Ugroženost pčela i održivo pčelarastvo postale su veoma važne teme širom sveta. Rešenja kao što su urbano pčelarstvo, građenje specijalizovanih pčelinjih bašti, usvajanje zakona o zabrani upotrebe različitih pesticida u poljoprivredi samo su neke od strategija pomoću kojih se društvo suprotstavlja ovom problemu. Naravno, i tehnološka indrustrija pozabavila se ovim temom.

Među prvim i zasigurno jedno od najzanimljivijih pristupa koji dolaze iz sveta tehnologije jesu sistemi za nadgledanje i praćenje pčelinjih društava, odnosno košnica [1-4]. Prikupljanje najrazličitijih podataka uz pomoć električnih senzora i dostavljanje istih pčelarima u realnom vremenu napravilo je veliki pomak i poboljšanje u pčelarstvu. Istvremeno postavljen je temelj za nebrojano istraživanja i realizacija kreativnih proizvoda čiji se broj s godinama samo povećava.

Podatke koji se prikupljaju iz košnica A. Zacepins je u svom radu [1] podelio na tri grupe:

1. Podaci koji se tiču pčelinjaka (meteorološki parametri, video osmatranja itd.).
2. Podaci koji se tiču pčelinjeg društva (temperatura i vlažnost unutar košnice, prisustvo različitih gasova, zvuk, vibracije određenih delova košnice, težina itd.).
3. Podaci koji se tiču samih pčela (broj pčela koje ulaze/izlaze iz košnice, trenutni broj pčela u košnici itd.).

M. Milovanović je student doktorskih studija na modulu Elektronika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mihajlo.milovanovic@elfak.rs

U ovom radu biće opisan razvoj IoT (*Internet of Things*) sistema koji prikuplja podatke o temperaturi, vlažnosti, težini jedne košnice kao i njenoj trenutnoj lokaciji. Primena ovakve vrste sistema značajno smanjuje broj manuelnih pregleda košnice, što za posledicu ima povećanje produktivnosti i brzine reagovanja u slučaju nepredviđenih situacija.

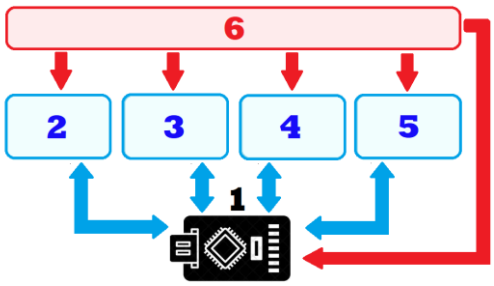
II. REALIZACIJA SISTEMA

Sistem se može podeliti na šest funkcionalnih jedinica. (1) Centralna procesorska jedinica, (2) blok za prenos podataka, (3) blok za merenje težine, (4) blok za merenje temperature i vlažnosti vazduha izvan košnice, (5) GPS modul i (6) napajanje. Blok dijagram sistema prikazan je na slici 1.

Kao centralna procesorska jedinica korišćen je mikrokontroler Arduino Nano ATmega328P [5]. Upravo ovaj kontroler zadužen je za rad i kontrolu čitavog sistema. Firmware je pisan u Arduino IDE razvojnom okruženju koristeći C programski jezik.

Blok za prenos podataka sastoji se iz tri dela. Prvi i centralni deo tog sistema jeste kompaktni GSM/GPRS modul, SIM800L [6]. U centru ovog modula nalazi se integrisano kolo SIM800L. Radni napon ovog integrisanog kola kreće se od 3.4 do 4.4V. Minimalna brzina prenosa (*baud rate*) koju podržava ovaj modul iznosi 1200 b/s, dok je maksimalna 115200 b/s. Drugi deo sistema jeste GSM antena, postavljena par centimetra od PCB ploče radi smanjenja negativnih uticaja i postizanja boljih performansi. Najveća mana ovog module jeste njegova potrošnja, koja zavisno od stanja u kome se nalazi može biti prilično velika. U toku slanja, struja može da dosegne vrednosti do 2000 mA. Da bi se nadomestio ovaj problem, između dva prenosa podataka modul SIM800L nalazi se u režimu rada sa minimalnom potrošnjom (*sleep mode*). Kada se nalazi u ovom režimu vrednost struje ne prelazi 0.7 mA. S obzirom da čitav sistem koristi napajanje od 5V, dok je radni napon SIM800L modula između 3.4 i 4.4V, neophodan je naponski regultar. U tu svrhu iskorišćen je naponski regulator LM2596 koji ujedno predstavlja treći i poslednji deo bloka za prenos podataka.

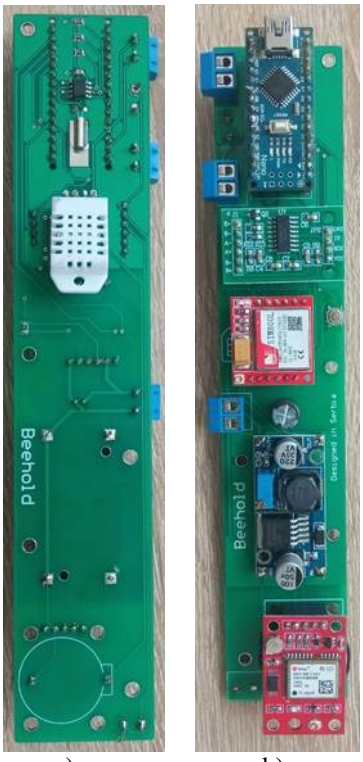
Sistem za merenje težine bazira se na četiri identične merne ćelije postavljene u svakom od četiri ugla sistema. Opseg svake od ovih ćelija iznosi 50kg. Razlog zbog koga su postavljene četiri ćelije jeste taj što, prema dosadašnjem iskustvu pčelara, težina jedne košnice u retkim slučajevima može da dostigne čak 200 kg. Sva četiri senzora u sprezi su sa A/D konvertorom HX711 [7] koji izmerene vrednosti u digitalnom formatu prosleđuje mikrokotroleru.



Slika 1. Blok dijagram sistema. (1) Centralna procesorska jedinica (Atmega328P), (2) Blok za prenos podataka (SIM800l), (3) Blok za merenje težine, (4) Blok za merenje temperature i vlažnosti, (5) GPS modul, (6) Napajanje.

Za prikupljanje informacija o temperaturi i vlažnosti vazduha postoje različiti pristupi. U predstavljenom sistemu koristi se DHT22 [8], senzor vlažnosti i temperature koji na svom izlazu daje digitalne podatke. Osobine kao što su kompaktnost, mala potrošnja, kao i stabilnost podataka pri prenosu na velike razdaljine čine ovaj senzor savršen za ovakvu primenu.

Za nalaženje lokacije koristi se NEO-6M GPS modul [9]. Ovaj modul predstavlja veoma pogodno rešenje za sisteme koji koriste baterijsko napanje i kod kojih je veoma važno voditi računa o karakteristikama kao što su potrošnja i dimenzije [10]. Takođe, radi minimizacije smetnji, GPS modulu pridodata je eksterna antena, koja se postavlja na određenoj udaljenosti od PCB ploče.



a) b)

Slika 2. Praktična realizacija hardvera. a) Donja i b) Gornja strana PCB ploče.

Važan funkcionalni deo hardvera predstavlja napajanje. Opisani uređaj napaja se uz pomoć fotonaponskog sistema. Jedinica za napajanje sastoji se od standardnih komponenti: solarnog panela, kontrolera punjenja i baterije. Razloga za ovakvu vrstu napajanja ima dva. Naime, pčelari najčešće svoje košnice postavljaju u ruralna područja gde standardno napajanje nije dostupno, a tokom sezone, pčelari su primorani da svoje košnice premeštaju i po nekoliko puta, pri čemu udaljenost između lokacija može iznositi i nekoliko stotina kilometara.

Kako je pri radu ovakvog sistema vremenska komponenta veoma važna, sistemu je pridodato integrisano kolo DS1307 (*real time clock*) [11]. Njegova uloga jeste upravo obezbeđivanje rada u realnom vremenu. Da bi se obezbedio konstantan rad digitalnog sata čak i u slučaju nepredviđenog i iznenadnog gubitka napajanja čitavog sistema, koristi se posebna litijum jonska baterija (2023).

Nakon merenja, podaci se šalju na server pomoću mobilne mreže. Na osnovu podataka koje dobije, server formira poruke i izveštaje koje prosleđuje web aplikaciji. Uspostavljanje konekcije sa serverom i slanje podataka izvršava SIM800L uz pomoć AT komandi. Svaki aktiviran uređaj ima svoj jedinstveni identifikacioni broj (ID).

Ceo sistem, osim jedinice za napajenje smešten je u dva odvojena kućišta međusobno povezana kablom koja se postavljaju ispod košnice (slika 3).

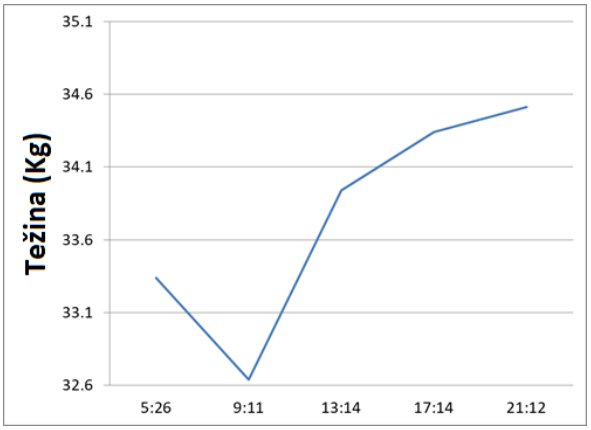


Slika 3. Praktična realizacija čitavog sistema.

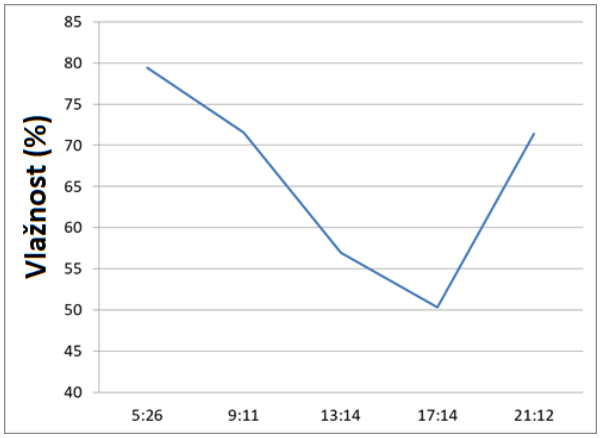
III. REZULTATI

Sistem je projektovan tako da poseduje mogućnost rada u četiri različita režima. U prvom režimu, sistem prikuplja i podatke šalje jednom dnevno (21:00h). Ovaj režim rada pogodan je za zimske uslove, odnosno van pčelarske sezone. Pčele su u ovom period godine mirne i stanje u košnici se veoma sporo menja, tako da je pčelarima dovoljno da informacije dobijaju jednom dnevno. Naravno, ovakav režim rada ima najmanju potrošnju, što je takođe velika prednost tokom zimskih meseci, kada je broj sunčanih dana mali. U drugom režimu rada, sistem prikuplja podatke pet puta u toku dana(05:00h, 09:00h, 13:00h, 17:00h, 21:00h) a šalje samo jednom(*21:00h*). Razlog za postojanje ovakvog režima

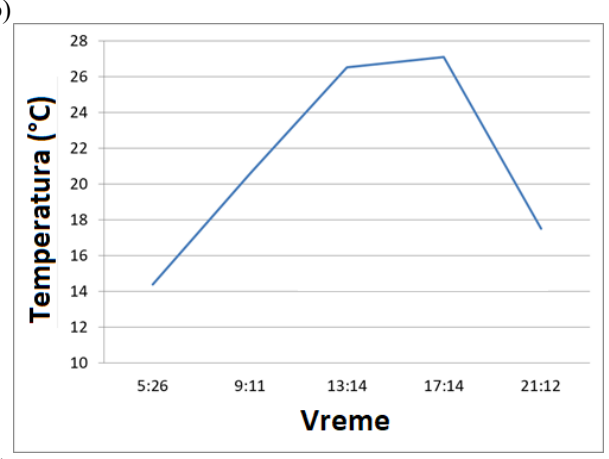
rada je taj što pčelarima ponekad nije potrebno praćenje košnice u realnom vremenu, već im je dovoljno da na kraju dana dobiju izveštaj o promenama u toku tog dana. Treći režim rada je vrlo sličan drugom, sa tom razlikom što se slanje podataka vrši dva puta dnevno(13:00h, 21:00h).



a)

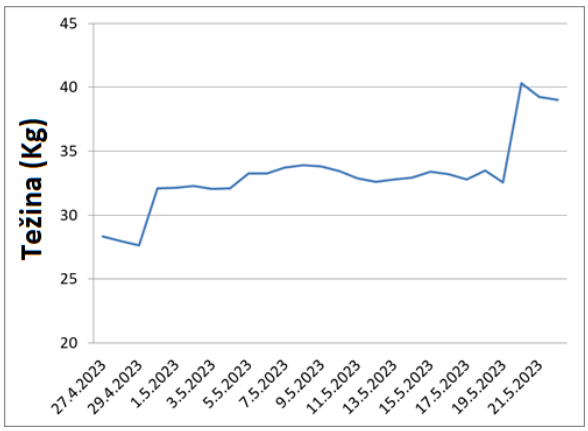


b)

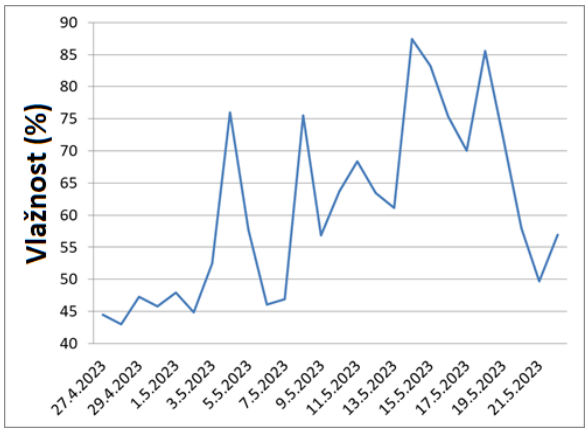


c)

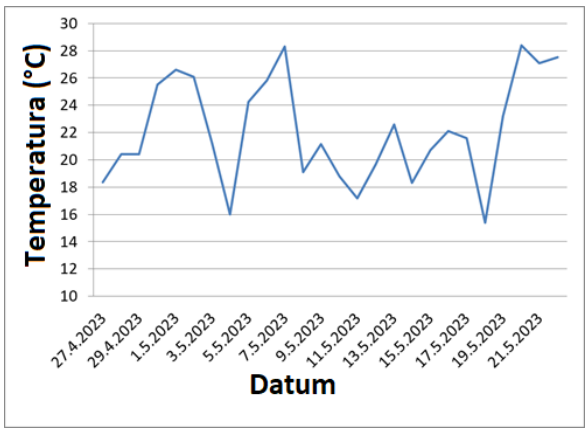
Slika 4. Eksperimentalni rezultati praćenja košnice tokom jednog dana. a) Težina, b) Vlažnost i c) Temperatura



a)



b)



c)

Slika 5. Eksperimentalni rezultati praćenja košnice tokom jednog meseca. a) Težina, b) Vlažnost i c) Temperatura

U četvrtom režimu rada uređaj šalje podatke nakon svakog očitavanja, i taj proces se ponavlja pet puta dnevno (05:00h, 09:00h, 13:00h, 17:00h, 21:00h). Naravno, u ovom režimu rada, potrošnja je najveća, ali zato se korisniku pruža najširi spektar informacija i samim tim, potreba za manuelnom inspekcijom košnica je svedena na minimum.

Na slici 4 prikazani su rezultati uređaja u četvrtom režimu rada (pet slanja u toku dana), na dan 17. 5. 2023.

Prva stvar na koju treba obratiti pažnju na ovoj slici jeste temperatura. Uz pomoć dobijenih podataka može se zaključiti da je 17. Maj bio normalan prolećni dan, bez ikakvih nepredviđenih i ekstremnih promena u temperaturi. Takođe, dobijeni podaci koji se tiču vlažnosti vazduha, pokazuju da nikakvih padavina toga dana nije bilo. Na kraju, posmatrajući grafik koji pokazuje zavisnost težine od vremena može se videti da je razlika u težini na početku i na kraju dana +1500g. Uzrok padu težine između 05:26h i 09:10h jeste izlazak pčela iz košnice.

Rad sistema u četvrtom režimu i posmatranje podataka prikupljenih tokom jednodnevnog praćenja košnice na žalost daju ograničene informacije. Na slici 5 prikazani su rezultati uređaja u prvom režimu rada. U periodu od skoro mesec dana, od kraja aprila do kraja maja, podaci su se prikupljali i slali svakog dana u 21:00h. Tokom ovog perioda takođe je vršena i manuelna inspekcija košnice od strane iskusnog pčelara kako bi se potvrdila validnost dobijenih informacija.

Ono što se analizom dobijenih podataka može zaključiti jeste to da je period tokom kojih su vršena merenja bio veoma nestabilan u pogledu vremenskih prilika. Bilo je dosta hladnih i kišovitih dana. Upravo ovo je razlog zbog koga grafik težine ne pokazuje očekivane rezultate. Zbog loših vremenskih uslova produktivnost pčela nije bila na zavidnom nivou pa tako ni proizvodnja meda nije dostigla željene reultate. Nagli poremećaj u težini koji se uočava 20. maja jeste rezultat fizičkog delovanja pčelara.

IV. Zaključak

U ovom trenutku postoje nekoliko aktivnih uređaja instaliranih u preko deset pčelinjaka širom Srbije. Uređaji prikupljaju podatke i uredno korisnike obaveštavaju o trenutnom stanju njihovih košnica u zavisnosti od izbora režima rada.

Testiranje ovih uređaja pokazalo je da se korišćenjem ovakvih sistema zaista smanjuje potreba za manuelnim pregledom košnica, kao i to da pčelari prateći informacije koje im pristižu mogu na vreme da reaguju i preduzmu potrebne mere ukoliko u košnici dođe do nekih nepredviđenih okolnosti. Takođe, u nekim delovima zemlje u kojima se nalaze instalirani uređaji, vreme je bilo prilično nestabilno.Ovim testiranjem utvrđeno je da loši vremenski uslovi kao što su kiša ili niska temperature neće uticati na rad uređaja.

Zahvalnica

Autor rada se zahvaljuje kompaniji Beehold iz Niša na pružanju materijalne i tehničke podrške. Takođe, zahvalnost duguje prof. dr Danijelu Dankoviću i asistentu Nikoli Mitroviću na nesebičnoj pomoći i usmeravanju u toku izrade projekta.

Literatura

[1] A. Zacepins, V. Brusbardis, J. Meitalovs, and E. Stalidzans, “Challenges in the development of Precision Beekeeping”, Biosystems Engineering, 2015, vol. 130, pp. 60-71

[2] A. Kviesis and A. Zacepins, “System Architectures for Real time Bee Colony Temperature Monitoring”, Procedia Computer Science, 2015, vol. 43, pp. 86-94.

[3] A. Zacepins and J. Meitalovs, “Implementation of multi-node temperature measurement system for bee colonies online monitoring”, Proc. of the 2014 15th International Carpathian Control Conference (ICCC)., 2014, pp. 695-698.

[4] A. Zabasta, A. Zhiravetska, N. Kunicina, and K. Kondratjevs, “Technical Implementation of IoT Concept for Bee Colony Monitoring”, Proc. of 2019 8th Mediterranean Conference on Embedded Computing (MECO)., 2019, pp. 1-4.

[5] ATmega328P datasheet, Atmel, 2015. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[6] SIM800l datasheet, SIMcom, 2013. Dostupno na: https://www.makerhero.com/img/files/download/Datasheet_SIM800L.pdf

[7] HX711 datasheet, Avia Electronics, 2011. Dostupno na: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

[8] DHT22 datasheet, Aosong Electronics Co., 2018. Dostupno na: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[9] NEO-6M module datasheet, U-blox, 2011. Dostupno na: https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

[10] N. Mitrović, S. Veljković, Z. Prijić, and D. Danković, “Comparison of the performance of the different GPS receivers in practical applications”, Proc. of IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2022, pp. 11-16.

[11] DS1307 datasheet, Maxim Integrated, 2015.

Projektovanje i realizacija rezonantnog LLC pretvarača

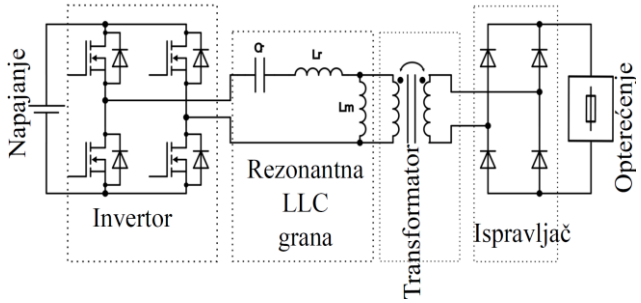
Vojin Vešković

Sadržaj – U ovom radu je prikazano projektovanje i realizacija rezonantnog LLC pretvarača sa svim svojim osnovnim i pomoćnim komponentama. Izabrana topologija LLC pretvarača obuhvata ulazni mostni invertor, rezonantnu LLC granu i izlazni ispravljač sa srednjom tačkom na transformatoru. Takođe u radu je prikazan detaljan postupak projektovanja visokofrekventnih magnetnih komponenti elektroenergetskih pretvarača, i dat je proračun očekivanog zagrevanja. Na kraju rada prikazani su rezultati dobijeni eksperimentalnom verifikacijom realizovanog pretvarača.

I. Uvod

Smanjenje gubitaka, povećanje efikasnosti i poboljšanje performansi svih električnih uređaja, uključujući i elektroenergetske pretvarače, jedan je od glavnih pravaca istraživanja u energetskoj elektronici. Rezonantne topologije elektroenergetskih pretvarača poseduju rezonantno kolo. Postojanje pomenutog rezonantnog kola omogućava promenu i deformaciju napona i struja na prekidačkim komponentama pretvarača, čime se mogu postići režimi rada kojima se smanjuju gubici na prekidačima. Režimi rada koji omogućavaju „meku komutaciju“ (*soft switching*) su prekidanje napona u nuli (*zero voltage switching - ZVS*) [1] i rad pri prekidanju struje u nuli (*zero current switching - ZCS*). Prvi režim rada omogućava teorijsko anuliranje gubitaka na prekidačima pri uključenju, dok drugi režim omogućava smanjenje gubitaka pri isključenju prekidača.

Rezonantni LLC pretvarač služi za pretvaranje jednosmernog napona napajanja u jednosmerni (DC-DC).



Slika 1. Blok šema rezonantnog LLC pretvarača.

Prekidački gubici pri uključenju su dominantni u odnosu na gubitke pri isključenju, odakle se zaključuje da ZCS režim nije pogodan za rad pri visokim frekvencijama zbog povećanih gubitaka. Upravo osobina rada u ZVS režimu povećava efikasnost LLC pretvarača.

Još jedna osobina navedenih rezonantnih pretvarača je mogućnost promene pojačanja pretvarača promenom frekvencije rada. Ovo pokazuje da se rezonantni pretvarači, pa samim tim i LLC pretvarač frekventno kontrolišu. Još jedna prednost ove vrste pretvarača je postojanje galvanske izolacije. Na slici 1. prikazana je blok šema navedenog pretvarača.

II. Projektovanje rezonantnog LLC pretvarača

Postupak projektovanja rezonantnih pretvarača može se podeliti na pet koraka:

a. Odabir topologije na osnovu ulaznih parametara, zahteva i snage.
b. Proračun parametara rezonantne grane.
c. Simulacija energetskog dela pretvarača.
d. Projektovanje parametara rezonantne grane.
e. Odabir prekidačkih i pomoćnih komponenti.

A. Odabir topologije

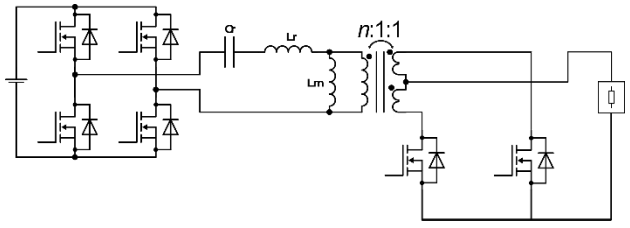
Energetski zahtevi pretvarača su sledeći:

- Ulazni napon pretvarača: $U_{in} = 47 \div 49V$,
- Izlazni napon pretvarača: $U_{out} = 48V$,
- Snaga pretvarača: $P = 300W$.

Na osnovu prethodno izloženih zahteva, bira se odgovarajuća topologija i frekvencija pretvarača (što veća ostvariva). Odabrano je:

- Topologija na mostnim invertorom (*full-bridge* LLC),
- Transformator sa srednjom tačkom na sekundaru sa prenosnim odnosom 1:1:1,
- Sinhrono (tranzistorsko) ispravljanje na izlazu, i
- Rezonantna frekvencija: $f = 460kHz$.

Šema energetskog dela pretvarača sa odabranom topologijom prikazana je na slici 2.



Slika 2. Izabrana topologija LLC pretvarača.

V. Vešković je student master studija Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: vojin.veskovic@elfak.rs

Uzimajući u obzir da se radi o pretvaraču čije je ukupno naponsko pojačanje jednako jedinici, te znajući činjenicu da je rad ove vrste pretvarača pri rezonantnoj frekvenciji stabilan pri svim opterećenjima, jednostavnije je projektovati navedeni pretvarač za rad u otvorenoj petlji pri rezonantnoj frekvenciji.

B. Proračun parametara rezonantne grane

Proračun i konačni odabir rezonantnih komponenti je najbitniji i najsloženiji deo projektovanja rezonantnih pretvarača. Proračun rezonantnih komponenti, zavisi od:

- odabira topologije ulaznog invertora,
- ulaznog i izlaznog napona kao i snage pretvarača,
- odabira prekidačke učestanosti, i
- odabira topologije transformatora i ispravljačkog dela pretvarača.

Postoji više načina proračuna parametara rezonantne grane koji se razlikuju po stepenu složenosti i preciznosti izračunavanja. Jedan od najčešće korišćenih proračuna obuhvata aproksimaciju prvog hramonika (*First harmonic approximation method - FHA*) [2]. Postupak je vrlo složen i podrazumeva iterativni račun. Nakon proračuna i provere izvodljivosti realizacije proračunatih parametara, odabrane su sledeće konačne vrednosti parametara:

$$C_r = 141nF, \quad (1)$$

$$L_r = 800nH, \quad (2)$$

$$L_n = 7, \quad (3)$$

$$L_m = 5.7\mu H \text{ i} \quad (4)$$

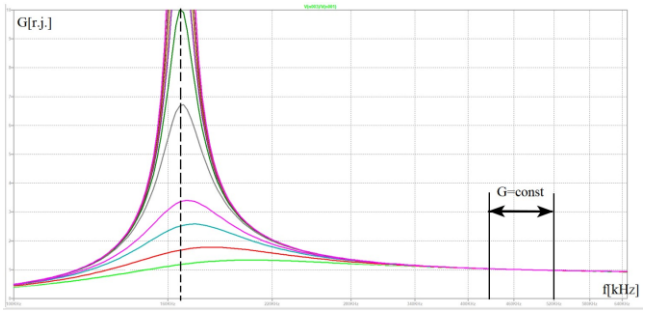
$$f_r = 459kHz, \quad (5)$$

gde je C_r je kapacitivnost rezonantnog kondenzatora, L_r induktivnost prigušnice, L_n parametar koji se bira i predstavlja odnos magnetizacione induktivnosti transformatora L_m i pomenute induktivnosti prigušnice, dok je f_r rezonantna frekvencija. Dobijene parametre treba verifikovati u simulacijama koje će biti prikazane u daljem delu rada.

C. Simulacija pretvarača

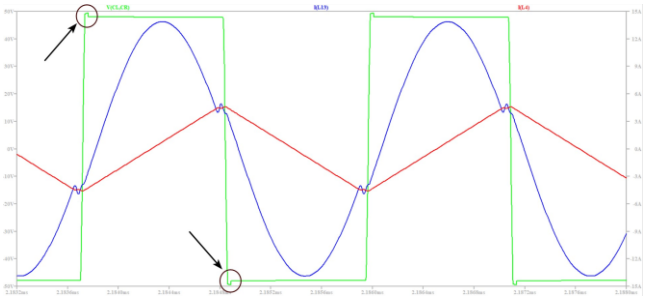
Nakon što su odabrani parametri rezonantne grane, neophodno je proveriti naponsko pojačanje rezonantnog kola u opsegu opterećenja od minimalnog do maksimalnog. Pomenuto je da je prenosni odnos transformatora jednak jedinici, kao i da je na ulazu mostni invertor, koji će raditi sa faktorom ispune od 50% (*duty cycle*), što implicira da će u ovom slučaju ukupno pojačanje pretvarača biti jednako pojačanju rezonantnog kola. Naponsko pojačanje se može proveriti računski ili simulacijom, u ovom slučaju je provera obavljena simulacijom u programu *LTSpice*, koji ima mogućnost analize kola u frekvencijskom domenu.

Normalizovano naponsko pojačanje rezonantnog kola G, u odnosu na frekenciju prekidanja rezonantnog kola f prikazano je na slici 3. Na grafiku je zelenom bojom prikazano pojačanje pri minimalnom opterećenju, dok je ljubičastom bojom prikazano maksimalno opterećenje koje je od glavnog interesa. S obzirom da je odabrano da pretvarač u ovom slučaju radi u otvorenoj petlji, bitno je primetiti da rezonantna kriva ima pojačanje jednako jedinici u širem opsegu frekvencija (opseg obeležen sa G=const). Ta osobina je dobijena odabirom velike vrednosti parametra L_n.



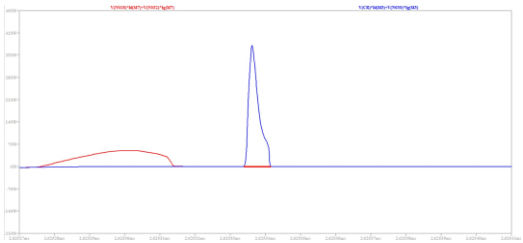
Slika 3. Grafik naponskog pojačanja pretvarača.

Treba pomenuti da pretvarač treba da radi na frekvencijama višim od niže rezonantne frekvencije, odnosno frekvencije na kojoj je dominantna magnetizaciona induktivnost (frekvencija obeležena isprekidanom linijom), budući da je to opseg frekvencija pri kojima je zadovoljen ZVS. Sledeći korak je simulacija energetskog dela pretvarača (slika 4).



Slika 4. Napon na tranzistorskom mostu (zeleno), struja rezonantne grane (plavo), magnetizaciona struja (crveno).

Na slici 4 su prikazani talasni oblici struje rezonantne grane, magnetizacione struje transformatora, kao i napona na tranzistorskom mostu, pri maksimalnom opterećenju. Bitno je primetiti obeležene delove talasnog oblika tranzistorskog mosta, odnosno trenutak kada provede bodi dioda tranzistora. Taj trenutak pokazuje da tranzistori rade u ZVS režimu. Talasni oblik rezonantne struje je sinusoidalan i to je još jedna osobina rezonantnih pretvarača, dok je magnetizaciona struja transformatora trougaonog oblika.



Slika 5. Prekidački gubici tranzistora u mostnom invertoru.

Na slici 5 prikazani su talasni oblici prekidačkih gubitaka pri uključenju na istim tranzistorima pri istim frekvencijama i radu u ZVS režimu LLC pretvarača (crveno), i običnog full bridge pretvarača (plavo) koji radi u režimu tvrde komutacije (*hard switching*). Primećuje se drastično smanjenje prekidačkih gubitaka LLC pretvarača.

D. Projektovanje parametara rezonantne grane

Projektovanje parametara rezonantne grane je najzahtevniji i najvažniji deo projektovanja ove vrste pretvarača [3]. Rezonantna grana se sastoji od rezonantnog kondenzatora, rezonantne prigušnice i transformatora. Na navedenim frekvencijama se kao rezonantni kondenzatori koriste keramički kondenzatori. Glavne osobine na koje treba obratiti pažnju pri odabiru kondenzatora su struja naprezanja, radni napon i frekvencija. Na osnovu toga odabran je kondenzator proizvođača *KEMET,* kapacitivnosti 47 nF. Vezivanjem tri ovakva kondenzatora u paraleli dobija se potrebna rezonantna kapacitivnost.

Projektovanje magnetnih komponenti u energetskoj elektronici, jedan je od najzahtevnijih zadataka. Pri projektovanju distributivnih transformatora mrežne frekvencije mogu se zanemariti mnogi parazitni efekti, koji su dominantni na visokim frekvencijama. Među njima su najdominantniji skin efekat, koji je na ovoj frekvenciji dubine 97.19 μm, i efekat blizine.

Postupak projektovanja rezonantnog transformatora se može podeliti na dva dela, na projektovanje feritnog jezgra, i projektovanje namotaja [4]. Jedan od najlakših načina za odabir veličine feritnog jezgra jeste metod proizvoda površina (*area product method*). Proračun proizvoda površina u suštini pokazuje da proizvod efektivne vrednosti površine poprečnog preseka feritnog jezgra (A_e), i površine prostora za namotaje u feritu (W_a), treba da bude veći od proračunate vrednosti proizvoda površina. Za minimalnu vrednost proizvoda površina jezgra transformatora se dobija:

$$A_p = \frac{U \times ton \times 2 \times Irms}{\Delta B \times J \times k} = 1920 m^4 \, . \qquad (6)$$

Gde je za gustinu struje *J*, uzeto 4.5 $\frac{A}{m}$, za koeficijent *k* (koeficijent ispune namotajima) uzeto 0.5, dok je za vrednost promene magnetne indukcije uzeta standardna vrednost od ΔB=100 mT, dok su ostali elementi preuzeti sa simulacije. Odabrano je feritno planarno jezgro male termičke otpornosti EELP 32 materijala n49. Za vrednost magnetne indukcije ovog jezgra, ukoliko odnos broja namotaja primara i sekundara bude 4:4:4, i kada se doda veštački vazdušni zazor u sva tri stuba ferita od 0.25 mm dobija se:

$$\Delta B \; = \; 49.3 \, mT. \qquad (7)$$

Dalje se sa grafika karakterističnih gubitaka u feritu u odnosu na indukciju i frekvenciju dobija vrednost očekivanih gubitaka od:

$$P_v = 91 \frac{kW}{m3}. \qquad (8)$$

Nakon toga treba pristupiti formiranju namotaja transformatora, budući da se radi o planarnom niskoprofilnom feritnom jezgru jedna od mogućnosti je umetanje namotaja u vidu tankih štampanih ploča, što je i odabrano u ovom slučaju. Namotaji se mogu projektovati u programskom paketu *Altium*. Pri odabiru rasporeda namotaja od velike važnosti je obratiti pažnju na uticaj skin efekta i efekta blizine. Da bi se anulirao uticaj skin efekta neophodno je da debljina provodnika bude manja od dvostruke dubine skin efekta (u ovom slučaju je odabrano 0.14 mm), dok se efekat blizine smanjuje pravilnim rasporedom namotaja primara i sekundara, odnosno neophodno je koristiti *interleave* metodu prilikom postavljanja namotaja. U programskom paketu *FEMM* se može rešiti ovaj problem. Nakon simulacije dobijena je vrednost povećanja naizmenične otpornosti namotaja u odnosu na jednosmernu otpornost:

$$R_{ac} \approx 2 \times R_{dc} = 2 \times 0.06\Omega = 0.12\Omega \qquad (9)$$

Sada se za gubitke u bakru očekuje vrednost od:

$$P_{cu} = 1.44 \text{ W}. \qquad (10)$$

Odakle se za očekivano zagrevanje transformatora iznad ambijentalne temperature dobija vrednost od:

$$\Delta t = R_{th} \times \left(P_{fe} + P_{cu}\right) = 13.6 \frac{^\circ C}{W} = 1.93 = 26 ^\circ C \qquad (11)$$

Na sličan način kao kod transformatora, metodom proizvoda površina se bira jezgro rezonantne prigušnice. Jednačina je u ovom slučaju malo drugačija, i za minimalnu vrednost proizvoda površina dobija se [5]:

$$A_p = \frac{L[nH] \times Ipk \times Irms}{Bmax \times J \times k} = 854 m^4 \qquad (12)$$

Na osnovu toga se bira jezgro RM8 materijala 3f3. Prigušnica će biti namotana sa tri navojka licnaste žice 105x0.08 mm, paralelno vezana i međusobno uvijena. Sada se za vrednost magnetne indukcije prigušnice dobija vrednost od:

$$\Delta B \; = \; 43.5 \, mT, \qquad (13)$$

dok se za specifične gubitke u bakru dobija vrednost od:

$$P_v = 90 \frac{kW}{m2}. \qquad (14)$$

Cilj uvijanja licnaste žice je smanjenje efekta blizine, dok je skin efekat zanemarljiv s obzirom da je debljina jedne licne 0.08 mm. Očekivano je da sada naizmenična otpornost namotaja bude 3 puta veća u odnosu na jednosmernu otpornost namotaja. Zagrevanje prigušnice iznad ambijentalne temperature računa se na sledeći način:

$$\Delta t = Rth \times (Pfe + Pcu) = 23.4\,\frac{C^\circ}{W} \times 1.02W, \qquad (15)$$

odakle se konačno dobija:

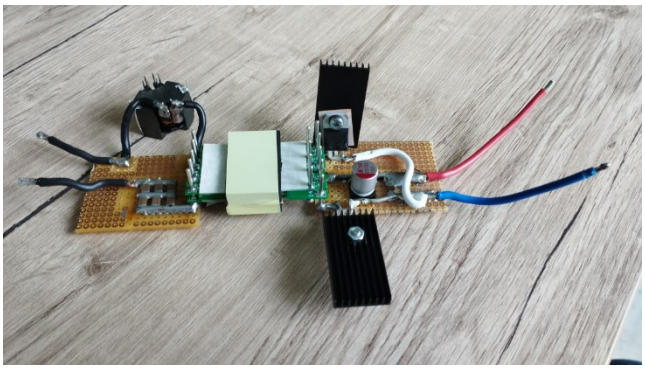
$$\Delta t = 23.87°C. \qquad (16)$$

E. Odabir prekidačkih komponenti

Tranzistori se biraju u odnosu na struju i napon opterećenja, da imaju što manju moguću otpornost provođenja i što manje vreme uključenja i isključenja. U tu svrhu su za ulazni tranzistorski most odabrani tranzistori proizvođača *Toshiba* TPH5R60APL, dok su za sinhrono ispravljanje odabrani tranzistori iz serije TPH9R00CQH, s obzirom da ti prekidači imaju duplo veći radni napon (ispravljač sa srednjom tačkom na transformatoru).

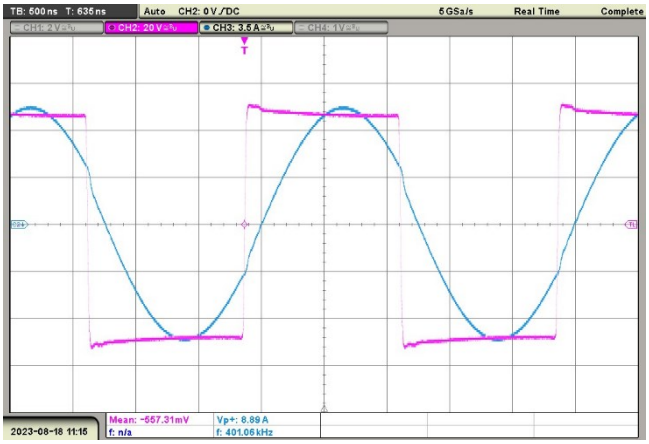
III. Eksperimentalna realizacija energetskog dela pretvarača

Izgled eksperimentalne postavke prikazan je na slici 6.



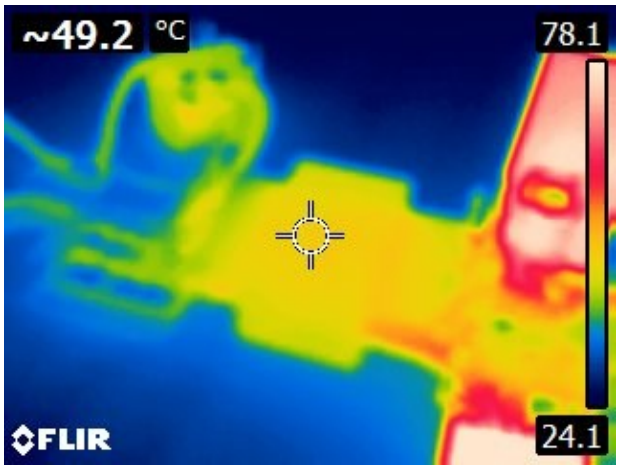
Slika 6. Eksperimentalna postavka.

Usled tehničkih mogućnosti izabrano je da se rezonantna grana pretvarača veže sa tranzistorskim mostom pomoću provodnika, koji su uneli izvesnu parazitnu induktivnost, čime je smanjena rezonantna frekvencija samog kola. Takođe na izlazu su vezane diode umesto sinhronog tranzistorskog ispravljača, s obzirom da je gotovo nemoguće vezati navedene tranzistore na ovakvu probnu ploču. Dodatni gubici koji su dobijeni usled korišćenja dioda umesto tranzistora, mogu aproksimativno modelovati ukupne gubitke pretvarača u pomoćnim kolima i napajanjima. Tako da efikasnost eksperimentalnog kola predstavlja približnu očekivanu efikasnost celog pretvarača, sa svim pomoćnim kolima.



Slika 7. Napon na tranzistorskom mostu (ljubičasto), struja rezonantne grane (plavo).

Na slici 7 prikazani su napon na tranzistorskom mostu i struja rezonantnog kola pri maksimalnom opterećenju. Ulazni i izlazni naponi su u ovom slučaju jednaki i iznose 48 V. Upoređivanjem slika 7 i 4, zaključuje se da pretvarač radi u režimu ZVS, kao i da su prikazani naponi i struje očekivanih oblika i vrednosti. Termalnom kamerom je snimano zagrevanje transformatora i prigušnice. Na slici 8 prikazano je zagrevanje transformatora u ustaljenom stanju i pri maksimalnom opterećenju.



Slika 8. Zagrevanje transformatora.

Temperatura ambijenta u datom trenutku iznosila je 25°C. Zaključuje se da je vrednost dobijena proračunom iz jednačine (11) gotovo identična stvarnom zagrevanju transformatora. Sa druge strane prigušnica se zagrejala 20.9°C iznad temperature ambijenta, što je približno jednako proračunatoj vrednosti iz jednačine (16).

IV. Zaključak

U ovom radu prikazan je pretvarač koji radi pri konstantnom ulaznom i izlaznom naponu, što je pogodno za

rad u otvorenoj petlji pri rezonantnoj frekvenciji. Međutim, projektovani pretvarač se zatvaranjem petlje može efikasno koristiti u slučaju potrebe za promenljivim ulaznim i izlaznim naponima. U radu je, takođe, pokazano da se simulaciono i računski može efikasno odrediti zagrevanje magnetnih komponenti. Realizovani pretvarač ima efikasnost od 97.07%, što pokazuje pogodnost korišćenja ovog tipa pretvarača.

ZAHVALNICA

Zahvaljujem se doc. dr Igoru Jovanoviću na sugestijama, nesebičnoj pomoći i podršci tokom izrade ovog rada.

LITERATURA

[1] F. C. Lee, "Zero-voltage switching tehnique in DC/DC converters," 1986.

[2] P. Smeets, "First Harmonic Approximation," 2017.

[3] B. Lu, W. Liu, Y. Liang, F. C. Lee and J. D. V. Wyk, "Optimal Design Methodology for LLC Resonant Converter," 2006.

[4] J. Zhang, Analysis and design of high frequency gapped transformers and planar transformers in LLC resonant converters, Galway: National University of Ireland, 2015

[5] M. K. Kazimierczuk and H. Sekiya, "Design of AC Resonant Inductors Using Area Product Method," in *ECCE*, 2009.

Kontrola prava pristupa pomoću razvojnog sistema Arduino

Strahinja Stošić

Sadržaj - Životni standardi današnjice nažalost teraju ljude na „lakšu" i uglavnom bolju zaradu. Kao što svi znamo to su krađe. Da bi se to sperčilo, domaćini kuća i vlasnici raznih firmi pribegavaju metodama postavljanja različitih bezbednosnih sistema. Na osnovu toga moj rad se zasniva na jednom primeru bezbednosnog sistema koji sam ja nazvao „Kontrola prava pristupa".

I. Uvod

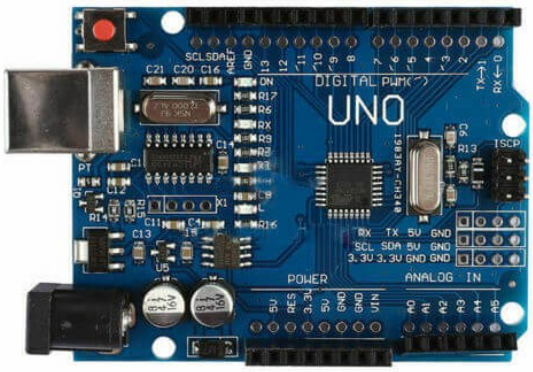
U suštini moj rad nije teško objasniti tako da se dosta toga shvati kroz sam sadržaj, ali ću još malo da ga pojasnim. Osnovna svrha sistema je da kad se dođe do određenog broja pokušasja unošenja šifre i ti pokušaji budu pogrešni dolazi do blokiranja sistema, kada se to desi sistem nije upotrebljiv sve dok se ne resetuje na fabrička podešavanja. Sam sistem je programabilan pa korisnik unosi šifru koju on želi, te se naravno sam sistem može nadograđivati softverski kao i hardverski na izuzetno lak način. Sistem koji zasniva ovaj rad jeste sam razvojni sistem Arduino. Ovaj rad se sastoji iz pet poglavlja. Prvo poglavlje predstavlja uvod, drugo poglavlje sadrži opis komponenata koje su korišćene u ovom radu, u trećem poglavlju se nalazi opis realizovanog sistema, četvrto poglavlje prikazuje šemu sistema, peto poglavlje daje opis makete izrađene za ovaj rad i poslednje poglavlje predstavlja zaključak.

II. Izbor komponenata

A. Arduino

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

S. Stošić je student prve godine na Elektronksom fakultetu u Nišu, Aleksandra Medvedeva 14, 18000 Niš, E-mail: strahinja.stosic@elfak.rs



Slika 1. Arduino Uno R3.

B. Potenciometar

Potenciometar je promenljiv otpornik koji funkcioniše kao razdelnk napona [2]. Potenciometri omogućavaju „biranje" otpora, odnosno kontinualno podešavanje otpora od praktično 0 oma do najveće naznačene vrednosti koja određuje opseg. Većina potenciometara obeležena je gornjom vrednošću (npr. 10k, 50k, 100k, 1M). Opseg potenciometra je samo približan. Ako na potenciometru nema oznake, može se upotrebiti multimetar da se odredi opseg. Vrednost otpora obično se podešava okretanjem osovine ili pomeranjem klizača. Promena električnog otpora može biti linearna (ravna) ili logaritamska. Potenciometri su uglavnom obrtnog tipa, dok se potenciometri s klizačem najčešće koriste kod stereo uređaja.



Slika 2. Potenciometar.

C. LCD 16x2 Displej

LCD (engl. *Liquid-crystal display* (LCD)) [3] je ekran zasnovan na tehnologiji tečnih kristala. Danas se najčešće koriste u LCD monitorima u obliku aktivnih TFT LCD ekrana. TFT LCD ekran sastoji se od određenog broja piksela, koji su poređani ispred nekog svetlosnog izvora (danas najčešće LED-e, a donedavno CCFL cevi). Troši relativno malo električne energije, te zauzima malo prostora. Tečne kristale otkrio je još 1888. godine austrijski botaničar F. Reintzer, kada je proučavao materiju aster cholesteryl benzoate. Topljenjem te materije dobio je mutnu tečnost koja se hlađenjem bistrila te se na kraju kristalisala. Međutim, tek je 1968. godine pronađena materija koja je na sobnoj temperaturi imala ova svojstva.

Tečne kristale je krajem 19. veka prvi pronašao austrijski botaničar Friedrich Reinitzer, a sam termin "tečni kristal" smislio je malo kasnije nemački fizičar Otto Lehmann.



Slika 3. LCD 16x2 Display.

D. LCD Dioda

LED, odnosno svetleća dioda (engl. LED; *light-emiting diode*) je posebna vrsta poluprovodničke diode koja emituje svetlost kada je propusno polarisana, tj. kada kroz nju teče struja [4]. Fotoni svetla se emituje prilikom rekombinacije para elektron-šupljina. Svetleća dioda se sastoji od čipa napravljenog od poluprovodnog materijala koji je dopiran nečistoćama kako bi se napravio p-n spoj. Kao i kod običnih dioda, električna struja teče od p-strane ili anode ka n-strani ili katodi, ali ne i u suprotnom smeru.



Slika 4. LCD Dioda.

E. Taster

Dugme je jednostavan mehanizam prekidača za kontrolu nekog aspekta mašine ili procesa [5]. Dugmad su obično napravljena od tvrdog materijala, obično plastike ili metala. Površina je obično ravna ili oblikovana tako da se prilagodi ljudskom prstu ili šaci, tako da se lako pritisne ili gurne. Dugmad su najčešće prekidači sa pristrasnošću, iako je mnogim dugmadima bez pristrasnosti i dalje potrebna opruga da bi se vratili u svoje stanje bez pritiska.



Slika 5. Taster.

F. Otpornik

Otpornik je pasivna elektronska komponenta koja pruža otpor električnoj struji, stvarajući pad napona između priključaka [6]. U otporniku uložena električna energija većim delom nepovratno prelazi u drugi vid energije, najčešće u toplotnu, ali može i u svetlosnu, mehaničku itd. Otpornici su uobičajeni elementi električnih i elektronskih kola, mogu biti sastavljeni od različitih materijala i imati različite oblike. Mogu biti stalni ili promenljivi.



Slika 6. Otpornik.

III. PRINCIP RADA SISTEMA

A. Unošenje šifre

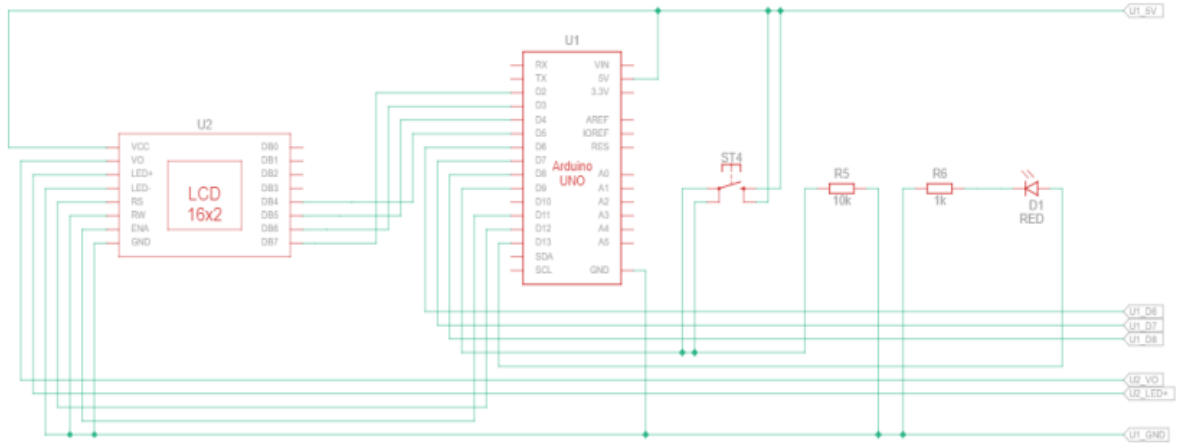
Ovaj deo je vema jednostavan. Tasteri su povezani preko proto borda uz pomoć provodnika na sam Arduino. Tasteri su podešeni na silazni signal što znači da kada se izvrši pritisak na sam taster on pošalje signal. U sistemu ne postoji nijedan taster koji bi vraćao šifru na sami početak to jest „clear" taster, već njegovu funkciju izvršava bilo koji od četiri tastera jer kada se pritisne bilo koji taster peti put, a pri tome je sama šifra netačna sistem, vraća pointer na početak i šifra koja je bila uneta pre se briše.

B. Prikaz šifre

Šifra se prikazuje na LCD 16x2 displeju. Prikaz šifre naravno nije u brojevima već u nekim isprogramiranim „predodređeni" znakovi, u našem slučaju to je "*". Takođe sistem ima potenciometar koji je izuzetno važan. Pitate se zašto? Zato što uz pomoć potenciometra kontrolišem osvetljenje samog LCD displeja, to je veoma bitno zbog vidljivosti u različitim vremenskim uslovima.

IV. ŠEMA

Na šemi koja je skinuta sa mog Tinkercad naloga [7] možemo videti kako to ovaj rad izgleda kao električna šema. Većinu pinova može da bira sam korisnik(programer), dok je odredjeni broj ima svoju predodređenu namenu (npr. VCC i GND).
Šema možda igleda malo neuredno ali je potpuno ispravna te ju je laše razumeti ako se pored same šeme ima i hardverski prikazan rad.



Slika 6. Električna šema.

V. ZAKLJUČAK

Iako sam se potrudio da ovaj sistem bude efikasan, da se lako nadograđuje, voleo bih da se što manje koristi. Takođe ako dođe do toga da treba da se upotrebi bilo bi veoma produktivno.

ZAHVALNICA

Zahvaljujem se kolegama koji su mi pomogli da smislenije uradim ovaj rad.

LITERATURA

[1] Arduino, https://en.wikipedia.org/wiki/Arduino_Uno
[2] Potenciometar, https://sh.wikipedia.org/wiki/Potenciometar
[3] LCD Display, https://www.techtarget.com/whatis/definition/LCD-liquid-crystal-display
[4] LCD Dioda, https://en.wikipedia.org/wiki/Light-emitting_diode
[5] Taster, https://en.wikipedia.org/wiki/Push-button
[6] Otpornik, https://sh.wikipedia.org/wiki/Otpornik
[7] Tinkercad, https://www.tinkercad.com/things/1vi5u338dvF-prava-pristupa/editel?sharecode=Z7FkQQy5h0QlneN1GdTyv58pJrQP-IzNhvEioqMSZ5w

Kontrola roletni korišćenjem servo motora u zavisnosti od izabranog moda rada

Marina Tasić, Jelena Milenković

Sadržaj – U radu je realizovan prototip sistema koji uz pomoć *Nucleo* razvojnog okruženja i ARM mikrokontrolera, korišćenjem servo motora, kontroliše roletne u zavisnosti od izabranog moda rada. Sistem se sastoji iz fotootpornika za merenje osvetljenosti, potenciometra za podešavanje željene osvetljenosti i tastera kojim korisnik bira mod rada. Podešenu i izmerenu osvetljenost, kao i mod rada prikazujemo na OLED displeju. Ako je izabran mod 1, servo motor menja položaj tako da se postigne željena osvetljenost. Ako je izabran mod 2, servo motor je konstantno u jednom od krajnjih položaja ako je osvetljenost ispod podešene granice ili u suprotnom položaju ako je osvetljenost iznad podešene granice.

I. UVOD U KOMPONENTE SNAGE

Komponentu snage, odnosno snažnu komponentu predstavlja svaka elektronska komponenta koja pouzdano može da vodi struju jaču od 1 A. Otkrićem bipolarnog tranzistora, 1947. godine počinje dominacija tehnologije bipolarnih komponenata snage. Njihovim unapređivanjem došlo je do razvijanja veoma snažnih komponenata koje su u stanju da vode struju preko 10 kA i izdrže napone iznad 4 kV.

U međuvremenu, 70-ih godina prošlog veka, razvijen je i MOS tranzistor snage. Osnovna razlika MOS tranzistora u odnosu na bipolarne komponente snage je to što je MOS naponski kontrolisana komponenta visoke ulazne impedanse. Time se obezbeđuje jednostavnost upravljačkih kola, a samim tim i smanjenje cena snažnih elektronskih sistema, što obezbeđuje značajnu primenu u savremenim aplikacijama, kao što su uređaji u domaćinstvu, automobilska elektronika, itd [1].

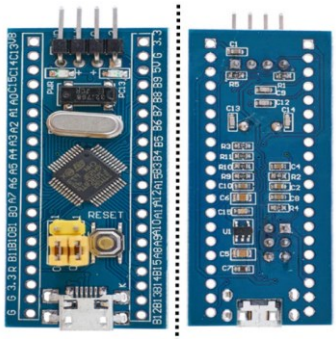
U ovom radu biće predstavljen sistem za kontrolu roletne pomoću servo motora koji se upravlja postizanjem željene osvetljenosti na fotootporniku okretanjem potenciometra. Potenciometar funkcioniše kao razdelnik napona. Servo motor je rotacioni aktuator važan u preciznoj kontroli ugaone pozicije, brzine i ubrzanja. Povezan je sa OLED displejom koji služi za dobijanje povratne informacije o njegovom položaju, kao i o izmerenoj osvetljenosti fotootpornika.

M. Tasić i J. Milenković su studentkinje osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. Email: marinatasic@elfak.rs, jelapn@elfak.rs.

II. KORIŠĆENE KOMPONENTE ZA REALIZACIJU PROJEKTA

A. Nucleo/STM32F103 Blue Pill

Nucleo je razvojno okruženje koje omogućava korisnicima izradu prototipova uz pomoć STM32 mikrokontrolera. Međutim, prilikom izrade projekta naišle smo na problem nestašice ovog okruženja na tržištu, pa smo umesto njega koristile *Blue Pill* [2]. *Blue Pill* je razvojna integrisana ploča koja je predstavljena kao jeftinija i alternativna verzija koja se koristi za STM mikrokontrolere.



Slika 1. Blue Pill.

B. Fotootpornik

Fotootpornik (*light dependent resistor* – LDR) je otpornik čija se otpornost smanjuje sa povećanjem intenziteta svetlosti. Zbog te osobine da njegova električna otpornost zavisi od upadne svetlosti, nailazi primenu kao elektronski senzor. Ovaj fotootpornik je jednostavan za upotrebu, ali ima mane u nedovoljnoj preciznosti i sporoj reakciji pri nagloj promeni osvetljenja. U ovom radu koristiće se fotootpornik prikazan na Slici 2. Dimenzije fotootpornika su 32mm dužine, 5mm širine i 4mm visine [3].



Slika 2. Fotootpornik.

C. Potenciometar

Potenciometar, odnosno promenljivi otpornik, omogućava kontinualno podešavanje otpornosti od najmanje (tj. gotovo od nule) okretanjem na desno do 4096Ω (2^{12}) okretanjem na levo.

D. Servo motor

Termin *servo motor* se koristi za označavanje motora pogodnog za upotrebu u sistemu upravljanja zatvorenom petljom [4]. Koristeći povratnu informaciju o položaju, servo motor (na Slici 3) kontroliše svoje kretanje i konačnu poziciju. Ulaz za njegovu kontrolu je signal.



Slika 3. Servo motor.

E. OLED displej

OLED (organic light-emitting diode) emituje svetlost koja se sastoji od organskih komponenti. U okviru projekta, OLED displej korišćen je u ispisivanju, gde je očitavanje informacija od presudnog značaja za korisnika [5]. Jednostavnost njegovog povezivanja, ne zahtevanje pozadinskog osvetljenja i manja debljina su prednosti kojima se odlikuje, ali je i njegova cena na tržištu i dalje mana.

U ovom projektu, korišćen je OLED SSD1306 (Slika 4), koji zbog svojih malih gabarita predstavlja moderniji tip displeja. Modul korišćenog displeja je 0.96 OLED I2C 128x64.



Slika 4. OLED SSD1306 displej.

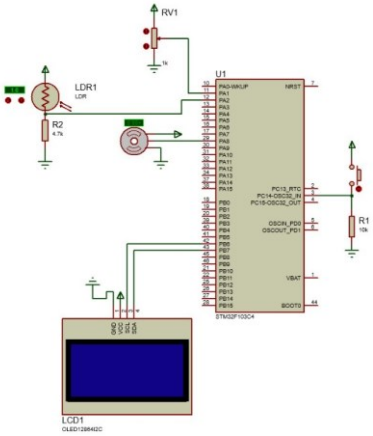
F. PRINCIP RADA

U početnom režimu rada, predviđeno je da se uz pomoć kontrolisanja potenciometra, servo motor vrti do određene željene vrednosti osvetljenosti fotootpornika, uz 2 potencijalna scenarija (osvetljavanje fotootpornika i zamračivanje fotootpornika). Nakon što postigne željenu vrednost, servo motor će se zaustaviti, a zatim vratiti u početni položaj uz ispisivanje vrednosti osvetljenosti i "moda1" na OLED displeju.

U drugom režimu rada, servo motor će promeniti položaj 2 puta, jednom kada je iznad zadate granice i drugi put kada je ispod, uz ispisivanje vrednosti osvetljenosti i "moda 2" na OLED displeju.

III. Šema sistema

Za projektovanje električne šeme korišćen je softverski paket *Proteus*. Električna šema prikazana je na Slici 5. Centralni deo sistema je razvojno okruženje *Blue Pill* sa STM32F103 mikrokontrolerom. Na razvojnom okruženju, preko pina PA1 povezan je potenciometar, koji je njegovim drugim krajem vezan za masu. Fotootpornik (LDR) koji je povezan na razvojno okruženje preko pina PA2 je u konfiguraciji sa otpornikom povezanim na masu. Preko pina PC14 je povezan taster, čijim pritiskom se menjaju režimi rada, odnosno dolazi do smenjivanja moda 1 i moda 2 na ekranu. Servo motor je povezan na pin PA8, drugi kraj mu je na napajanju, a treći na masi. OLED SSD1306 se povezuje na razvojno okruženje preko 4 pina, od kojih je jedan za VCC, a drugi za GND. SDA i SCL vezani su za pinove PB6 i PB7 respektivno (SDA – *serial data* i SCL – *serial clock*). Za komunikaciju je korišćen I2C protokol.



Slika 5. Električna šema sistema.

IV. Podešavanja u Cube Mx-u

CubeMx (Slika 6) je grafički alat koji omogućava vrlo lako konfigurisanje STM32 mikrokontrolera i generisanje kompletnog C projekta za: IAR, KEIL, AC6, ATOLLIC, itd.

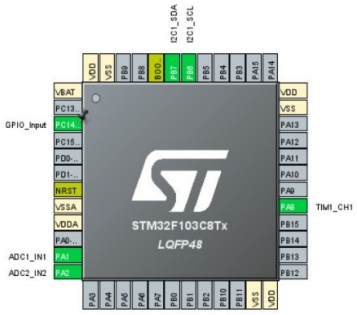
Da bi se obezbedio rad displeja, I2C komunikacija je realizovana na sledeći način: *Pinout & Configuration* > *Connectivity* > *I2C1* > *Mode* i I2C se bira iz padajućeg menija. U *Parametar Settings* podešen je *Fast Mode* sa

frekvencijom od 400 kHz jer OLED SSD1306 modul ne može da radi na nižoj frekvenciji. Pinovi koji označavaju I2C komunikaciju na mikrokontroleru su PB6 (za SCL) i PB7 (za SDA).

Kada je u pitanju AD konverzija, za ADC1 je potrebno u *Parametar Settings* u *Continuous Conversion Mode* označiti *Enable*, dok podešavanja za ADC2 ostaju difoltna. Pinovi koji označavaju ADC su PA1 za ADC1, odnosno PA2 za ADC2.

Naredna podešavanja se tiču tajmera TIM1, koji će generisati PWM potreban za menjanje brzine okretanja motora, gde je u *TIM1 Mode and Configuration* > *Mode* > *Channel 1* izabran PWM *Generation* CH1. U *Parametar Settings* je neophodno postaviti vrednost prescalera na 160-1, a *ARR* na 1000-1. Na mikrokontroleru, TIM1 označava pin PA8.

Podešavanjem pina PC14 (GPIO_Input) na koji će biti povezan taster, završena su podešavanja u Cube MX-u.



Slika 6. Podešavanja u Cube MX-u.

V. KOD U *KEIL-U*

Keil MDK je kompletno okruženje za razvoj softvera za niz ARM Cortex-M baziranih mikrokontrolerskih uređaja.

Na samom početku projekta, uvoze se potrebne biblioteke pomoću kojih će se ispisati kod. Sledeći deo koda odnosi se na deklaraciju promenljivih. Deklarišu se promenljive *mod*, kojom se označava početno stanje sistema, kao i *pomoćna* koja će služiti da se vrši promena iz moda 1 u mod 2. Promenljive *buffer1* i *buffer2* su tipa char sa maksimalnim brojem karaktera 20, dok je *dtc* (*duty cycle*) tipa uint16_t, a *foto* (za fotootpornik), *pot* (za potenciometar) i *taster* tipa uint32_t. Promenljive *foto* i *pot* će služiti za dalje određivanje granica do kojih će se vrteti servo motor.

Glavni deo koda zapisan je u *USER CODE BEGIN 3*. Na Slici 7 se vide naredbe za ispisivanje vrednosti osvetljenosti koje se dobijaju osvetljivanjem i zamračivanjem fotootpornika i osvetljenosti koja se dobija okretanjem potenciometra. Podešene su pozicije teksta koji će se ispisivati na OLED displeju, kao i font i boja slova, u dva reda, jedan ispod drugog. Iskorišćena je i naredba za brisanje teksta kako bi se prilikom prelaska iz jednog režima rada u drugi očitavale nove vrednosti.

Pomoću petlje ispisan je prelazak iz jednog režima rada u drugi pritiskom tastera. Tako se iz moda 1 na OLED displeju prelazi u mod 2, da bi se sledećim pritiskom taster iz moda 2 direktno vraćalo u mod 1.

```
125     /* USER CODE BEGIN 3 */
126
127
128     //rezim krajnji polozaj
129
130     foto=HAL_ADC_GetValue(&hadc2);
131     SSD1306_GotoXY(0,0);
132     SSD1306_Puts("E=", &Font_7x10,SSD1306_COLOR_WHITE);
133     sprintf(buffer1,"%d",foto);
134     SSD1306_Puts(buffer1, &Font_7x10,SSD1306_COLOR_WHITE);
135     SSD1306_UpdateScreen();
136
137
138     pot=HAL_ADC_GetValue(&hadc1);
139     SSD1306_GotoXY(0,12);
140     SSD1306_Puts("Pot=", &Font_7x10,SSD1306_COLOR_WHITE);
141     sprintf(buffer2,"%d",pot);
142     SSD1306_Puts(buffer2, &Font_7x10,SSD1306_COLOR_WHITE);
143     SSD1306_UpdateScreen();
144    HAL_Delay(300);
145    SSD1306_Clear();
146
147    if(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_14) && pomocna){
148       mod=!mod;
149      pomocna=false;
150    }
151    else if(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_14)== GPIO_PIN_RESET){
152     pomocna=true;
153    }
```

Slika 7. Glavni deo koda.

U nastavku smo obezbedili ispisivanje “mod 1” i “mod 2” na OLED displeju (Slika 8) radi lakšeg raspoznavanja dešavanja scenarija, podešavajući i za te reči odgovarajuće pozicije, boje i fontove. Što se tiče prvog slučaja kada se servo motor vrti do željene vrednosti osvetljenosti, potrebno je postaviti odgovarajuće granice do kojih on može doći i tu se zaustaviti pre nego što se vrati u početni položaj. Kako nam je od značaja osvetljenost fotootpornika, granice se postavljaju u opsegu od ±50 u odnosu na osvetljenost postignutu okretanjem potenciometra.

U suprotnom, prelazi se u mod 2, koji će imati dve promene položaja servo motora, ispod i iznad zadate granice, sa 50% i 100% *Duty Cycle*-om.

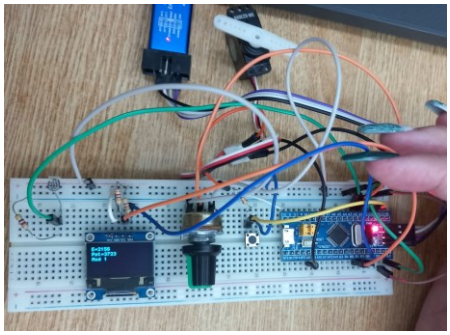
```
if(mod==true){
   SSD1306_GotoXY(0,24);
   SSD1306_Puts("Mod 1", &Font_7x10,SSD1306_COLOR_WHITE);
   SSD1306_UpdateScreen();
   htim1.Instance->CCR1=dtc;
  if(foto+50 > pot || foto-50 > pot){
    htim1.Instance->CCR1=dtc;
  }
  else{
     dtc=dtc+10;
     if(dtc==100){
       dtc=0;
     }
  }

}
else{
  SSD1306_GotoXY(0,24);
   SSD1306_Puts("Mod 2", &Font_7x10,SSD1306_COLOR_WHITE);
   SSD1306_UpdateScreen();
  if(foto<pot){
    htim1.Instance->CCR1=50;
  }
  else{
     htim1.Instance->CCR1=100;
  }
}
```

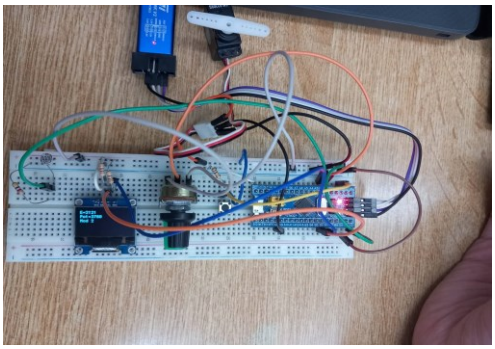
Slika 8. Definisanje moda 1 i moda 2.

VI. Praktična realizacija

Praktična realizacija ovog sistema prikazana je na Slici 9. Ukoliko u modu 1 osvetlimo fotootpornik tako da osvetljenost fotootpornika bude približna onoj koja se podešava potenciometrom, servo motor će prestati da se vrti. Isto će se desiti kada potenciometar podešavamo okretanjem u suprotnom smeru, a fotootpornik zamračimo. To znači da se njegovo kretanje vrši sve dok ne postigne određeni cilj, za razliku od moda 2 gde je potenciometar u mestu sve vreme, osim u dva slučaja, kad se nađe iznad i ispod dozvoljene granice.



Slika 9a. Mod 1.



Slika 9b. Mod 2.

VII. Zaključak

Ovaj projekat može se unaprediti na mnogo načina, zavisno od volje, zamisli i inovativnosti kreatora. To se može postići zadavanjem različitih granica okretanja servo motora, kao i ispitanih vrednosti željenih osvetljenosti. OLED displej mogu zameniti i neke druge vrste displeja, drugačijih rezolucija, boja i fontova. Takođe, mogli bi se popraviti problemi estetske prirode poput jakog treperenja korišćenjem naredbe za kašnjenje manje od 1ms. Izmene u kodu ne bi iziskivale previse vremena, postigle bi se na jednostavan način – dodavanjem naredbi u zavisnosti od primene.

Zahvalnica

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

Literatura

[1] I. Manić „*Komponente i kola snage*” Niš. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Komponente-i-kola-snage_predavanja.pdf

[2] Tehnička specifikacija za Blue Pill. Dostupno na: https://components101.com/sites/default/files/component_datasheet/STM32F103C8T6-Blue-Pill-Development-Board-Datasheet.pdf

[3] Tehnička specifikacija za Fotootpornik. Dostupno na: https://datasheetspdf.com/pdf/756925/SENBA/GL5516/1

[4] Tehnička specifikacija za Servo motor. Dostupno na: https://datasheetspdf.com/pdf/791970/TowerPro/SG90/1

[5] Tehnička specifikacija za OLED SSD1306 model. Dostupno na: https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf

Analiza probojnog napona gasnih odvodnika *Littelfuse* CG2300 pri različitim eksperimentalnim uslovima

Nikola Stojiljković

Sadržaj – U ovom radu predstavljena je analiza probojnog napona komercijalnih gasnih odvodnika *Littelfuse* sa radnim naponom od 300 V. Određivane su vrednosti statičkog probojnog napona pri različitim eksperimentalnim uslovima za 10 gasnih odvodnika prenapna. Najpre je analiza vršena pri normalnim atmosferskim uslovima a zatim u prisustvu gama zračenja iz izvora 226/88 Ra.

I. Uvod

Razvoj elektronike i računarske tehnike, koji se posebno ogleda u sve većoj minijaturizaciji, odnosno u sve većem stepenu integracije elektronskih komponenata aktuelizovao je problem njihove zaštite od prenapona. Naime, otpornost na pojavu prenapona je značajno smanjena minijaturizacijom elektronskih komponenata. Zato je veoma bitno posvetiti što veću pažnju problemu prenaponske zaštite komponenata. Gasni odvodnici prenapona su jedni od najboljih i najčešće primenjivanih nelinearnih elemenata za zaštitu od prenapona na niskonaponskom nivou.

Prenapon nastaje kada potencijal jedne tačke nekog voda, uređaja ili komponente postane veći od dozvoljenog, u odnosu na drugu tačku u kolu ili tačku nultog potencijala. U zavisnosti od stepena prekoračenja, prenapon može dovesti do privremenog ili trajnog poremećaja u radu uređaja, oštećenja uređaja, čak i ugroziti bezbednost osobe koja rukuje uređajem u trenutku nastanka prenapona. Prenaponi mogu biti impulsni ili trajni. Nastaju direktno usled komutacijskih procesa (poput uključivanja ili isključivanja uređaja, promena režima rada elektromotora i slično), elektrostatičkog pražnjenja i atmosferskog pražnjenja unutar uređaja ili komponente, ili unutar mreže na koju su uređaji priključeni; ili indirektno, kao posledica interakcije provodnika (žičanih struktura komponente ili uređaja) sa elektromagnetnim impulsom [1]. Atmosferska pražnjenja (munje i gromovi) su najopasniji izvor prenapona, jer se ne može uticati na uzrok njegovog nastanka.

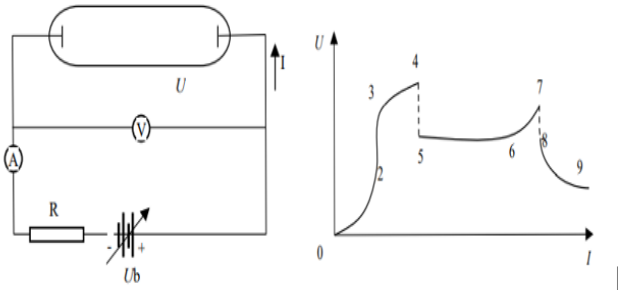
II. Gasni odvodnici prenapona

Gasni odvodnici prenapona (eng. *Gas Filled Surge Arresters* – GFSA) su komponente za prenaponsku zaštitu koje se sastoje od dve ili tri elektrode u simetričnoj konfiguraciji, zatopljene u keramičko ili stakleno kućište ispunjeno izolacionim medijumom – plemenitim gasom (najčešće argon) sa određenim primesama. Pri nastanku prenapona dolazi do jonizacije gasa i posledičnog naglog pada električnog otpora komponente, koji omogućava bezbedno odvođenje prenapona mimo osetljivih delova mreže ili uređaja. Gasne odvodnike prenapona odlikuje velika izdržljivost i sposobnost odvođenja vrlo velikih struja (do 60 kA, za pojedine komponente), ali se suočavaju sa malom brzinom reagovanja u odnosu na druge tipove odvodnika prenapona, i problemom gašenja odvodnika u impulsnom režimu. objašnjenjima, kako bi preciznije objasnilli sam princip rada.

A. Princip rada gasnih odvodnika prenapona

Na primeru elektronske cevi punjenje plemenitim gasom (pritiska 0.000001 bara do 0.08 bara) koji je obično neon, argon, helijum ili njihova smeša, može se analizirati princip rada gasnog odvodnika prenapona. Nastanak električnog pražnjenja u gasu može se objasniti korišćenjem Taunzendove teorije.

Na slici 1. prikazana je UI karakteristika cevi punjene inertnim gasom kao modela gasnog odvodnika prenapona i šema za snimanje karakteristike. Deo karakteristike od tačke 0 – 4 predstavlja oblast Taunzendovog pražnjenja. U tački 4 dovoljan broj elektrona dostiže jonizacionu energiju za samoodržavanje procesa, napon na cevi opada što pokazuje da je za održavanje procesa jonizacije gasa dovoljan manji napon nego za njegovo započinjanje.



Slika 1. UI karakteristika cevi punjene inertnim gasom i šema za njeno snimanje.

Napon u tački 4 se zove napon paljenja cevi, a cev se osvetli svetlošću koja je različita za različite vrste gasa, smeše gasa, pritiska gasa i rastojanja elektroda. Proces između tačaka 5 i 6 naziva se procesom tinjavog pražnjenja. U tački 7 počinje proces lučnog pražnjenja kada

N. Stojiljković je student master akademskih studija na studijskom programu Elektronika i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stojiljkovicn@elfak.rs.

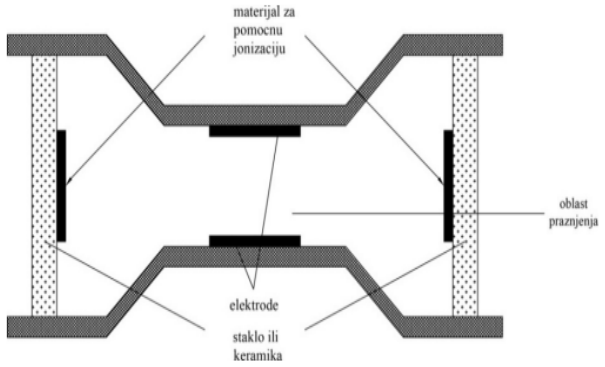
napon na cevi opadne do vrednosti jonizacionog potencijala za dati gas koji može biti 15 V – 20 V pri strujama od nekoliko stotina ili hiljada ampera. Problem u kolima prenaponske zaštite predstavlja činjenica da je za održavanje lučnog pražnjenja dovoljna struja intenziteta 0.1 A, pa se gašenje cevi može postići jedino smanjenjem struje na zanemarljive vrednosti ili njenim prekidanjem [2].

B. Konstrukcija gasnih odvodnika prenapona

Gasni odvodnik prenapona sastoji se od dve identične metalne elektrode zatopljene u keramičko ili stakleno kućište. Međuelektrodno rastojanje je reda veličine 1 mm. Na unutrašnjem zidu kućišta se u nekim slučajevima nanose radioaktivni izotopi radi povećanja brzine odziva (α-zračenje). Izolacioni medijum je plemeniti gas (neon, helijum, argon) ili smeša plemenitih gasova na pritisku od 0.01 bar do 1 bar.

Kućište odvodnika mora biti od neporoznog materijala pošto plemeniti gasovi kao što je poznato, imaju popunjenu poslednju atomsku orbitu i oni se javljaju isključivo u atomskoj formi, malih dimenzija i lako difunduju kroz porozne sudove [3].

Karakterističan presek jednog odvodnika prenapona dat je na slici 2.



Slika 2. Poprečni presek gasnog odvodnika prenapona.

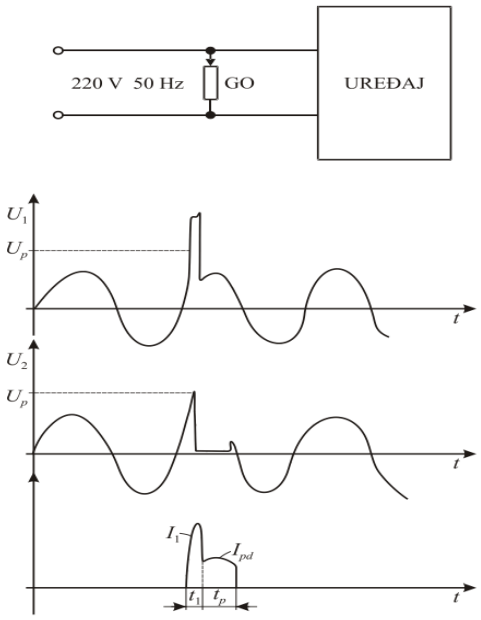
C. Karakteristike gasnih odvodnika

Otpornost gasnog odvodnika pre reagovanja je veoma velika i nepromenljiva za razliku od varistora. Pri naponima nižim od napona reagovanja otpornost gasnih odvodnika je 108 Ω do 1010 Ω, dok je u stanju rada odnosno pri lučnom pražnjenju reda 0.1 Ω. Sa aspekta otpornosti odvodnik se u normalnom radu može zanemariti, odnosno ne unosi nikakvo slabljenje ili prigušenje bez obzira na veličinu impedanse.

Mogućnost provođenja velikih impulsnih struja i velikih naizmeničnih struja izazvanih naponom distributivne mreže su bitne karakteristike gasnih odvodnika prenapona. Karakteristike provođenja impulsnih struja daju se za talasni oblik (8/20) μs, a mogućnost provođenja naizmeničnih struja izazvanih spojem sa distributivnom mrežom za period vođenja od 1 s sa ponavljanjem 10 puta sa razmacima od po 3 minuta. Odnos ovih struja je oko 1000. Kod izbora gasnih odvodnika za zaštitu od impulsnih smetnji koje dolaze preko mrežnog napajanja treba obratiti pažnju na struju produženog delovanja (*follow-on current*).

Na slici 3. prikazano je takvo kolo gde gasni odvodnik čiji je napon paljenja iznad mrežnog napona treba da zaštiti uređaj od prenaponskih impulsa koji se mogu pojaviti zajedno sa mrežnim naponom. Struja produženog delovanja je posledica paljenja odvodnika impulsnim prenaponom i traje sve dok se ne postignu uslovi za gašenje odvodnika i mada manja po intenzitetu može predstavljati značajno opterećenje za gasni odvodnik [4].

Za gasni odvodnik prenapona karakteristična je pojava starenja (ireverzibilnosti) karakteristika u toku eksploatacije uzrokovana promenom na površinama elektroda. Do ove promene na površinama elektroda dolazi usled proboja međuelektrodnog prostora odvodnika.



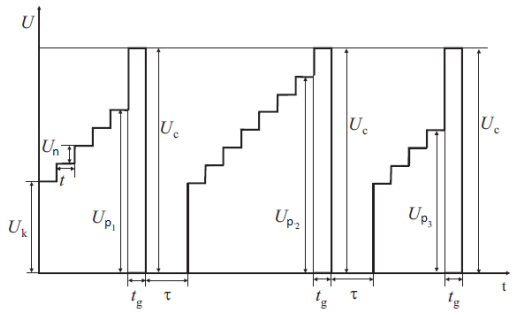
Slika 3. Zaštita uređaja od prenapona u distributivnoj mreži: U_1 – napon distributivne mreže sa prenaponskim impulsom; U_2 – napon na gasnom odvodniku; I_2 - struja kroz gasni odvodnik; I_1 - struja usled prenaponskog impulsa; I_{pd} – struja produženog delovanja.

III. PROBOJNI NAPON I NAČIN ODREĐIVANJA STATIČKOG PROBOJNOG NAPONA DINAMIČKOM METODOM

Probojni napon se najčešće određuje u uslovima stepenastog porasta napona na elektrodama gasne komponente tj. diskretizovanom dinamičkom metodom. Najpre se na gasnu komponentu dovodi fiksni napon U_k koji je dosta manji od očekivanog probojnog napona U_p. Zatim napon povećavamo za unapred zadati naponski step

U_n pri čemu se takođe zadaje i vreme trajanja naponskog stepa. Odnos $k = U_n / t_n$ predstavlja brzinu porasta napona. Kada nastane proboj naponom U_c se obezbeđuje zadata vrednost struje pražnjena i_g za vreme pražnjena t_g kako bi se u cevi uspostavila stacionarna koncentracija naelektrisanih i pobuđenih čestica. Zatim se napon na elektrodama gasne komponente isključuje za vreme τ koje se naziva vreme relaksacije i ono se takođe unapred zadaje.

Važno je takođe istaći da je za što tačnije određivanje probojnog napona U_p potrebno održavati konstantne parametre kao što su naponski step U_n, vreme između dva naponska stepa t_n, struju pražnjenja i_g i vreme relaksacije τ. τ je vreme od momenta isključenja napona U_c na gasnu komponentu do ponovnog priključenja napona U_k [5]. Ovaj postupak prikazan je na slici 4.



Slika 4. Stepenasti porast napona na gasnoj komponenti za seriju od tri merenja.

IV. Postavka merne instrumentacije

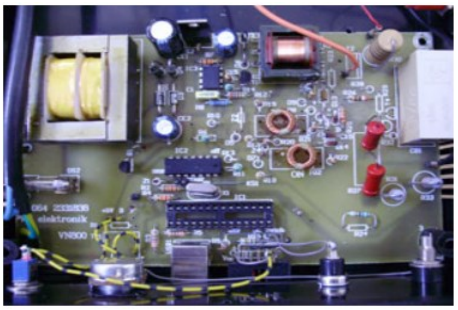
Laboratorija za vakuumsku i gasnu tehniku Elektronskog fakulteta, Univerziteta u Nišu, je osnovana sa ciljem da se istražuju fundamentalni procesi u niskotemperaturnoj plazmi koji se javljaju pre proboja, tokom pražnjenja i posle prestanka pražnjenja. Takođe, u ovoj laboratoriji se vrši karakterizacija gasnih komponenata kao što su gasni odvodnici prenapona, tinjalice, gasni detektori jonizujućeg zračenja, gasni izvori svetlosti.

U Laboratoriji za vakuumsku i gasnu tehniku uporedo sa eksperimentalnim istraživanjima vrše se i projektovanja i realizacije mernih sistema koji su neophodni za merenje pojedinih parametara gasnih komponenata i sistema.



Slika 5. Izgled sistema za merenje i akviziciju podataka probojnog napona gasnih komponenata.

Na slici 5. prikazana je postavka merne instrumentacije za akviziciju podataka probojnog napona gasnih odvodnika, dok je na slici 6. predstavljeno integrisano kolo generatora stepenastih signala odnosno deo sistema za merenje i akviziciju podataka probojnog napona gasnih komponenata.

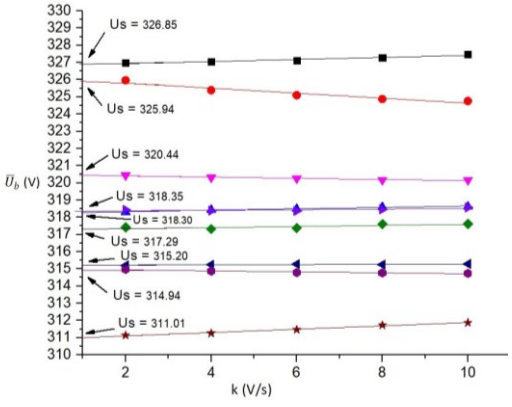


Slika 6. Integrisano kolo generatora stepenastih signala.

V. Prikaz dobijenih rezultata

Nakon izvršenih merenja, kao što se i može videti slika 7, eksperimentalni rezultati su sprovedeni na gasnim odvodnicima komercijalnog tipa dobro odražavaju linearnu zavisnost $\overline{U_b}$ od k, odnosno, možemo videti veoma dobro fitovanje pravom linijom. Srednja vrednost probojnog napona dobijena je iz skupova podataka koji sadrži po 100 izmerenih vrednosti napona za svaku zadatu vrednost brzine porasta napona od 2, 4, 6, 8, 10 V/s. Procenjene vrednosti statičkog probojnog napona razlikuju se od komponente do komponente, ali su u opsegu mogućih vrednosti datih u tehničkoj dokumentaciji [6].

U tabeli 1 može se videti vrednosti za statički probojni napon korišćenih komercijalnih gasnih odvodnika. Može se primetiti variranje napona od komponente do komponente. Upravo to nam pokazuje koliko je važna procena vrednosti statičkog probojnog napona pre bilo kog uključivanja gasnih odvodnika u složena elektronska kola i sisteme, kao i značaj ove eksperimentalne analize.



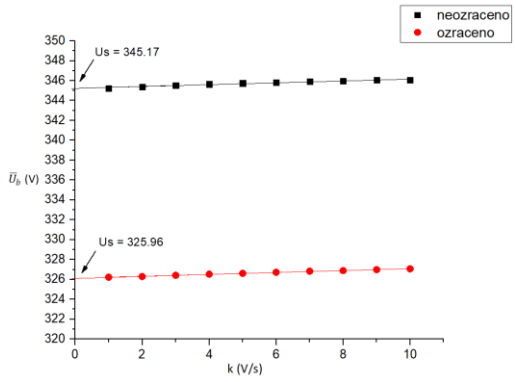
Slika 7. Procenjene vrednosti statičkog probojnog napona za 9 komercijalnih gasnih odvodnika CG2300.

TABELA I
VREDNOSTI STATIČKOG PROBOJNOG NAPONA ZA KORIŠĆENE GASNE ODVODNIKE

Gasni odvodnik	U_s (V)
1	326.85
2	325.94
3	318.35
4	320.44
5	317.29
6	318.30
7	315.20
8	314.94
9	311.01

Vrlo često potrebno je analizirati uticaj zračenja na probojni napon gasnih odvodnika. Za detaljniju analizu svakako je bolje odraditi veći broj ponavljanja kada su u pitanju merenja. Dok je u prethodnom delu analiza probojnog napona vršena na 100 merenja za svaku parnu brzinu porasta napona od 2 V/s do 10 V/s, sada su vršena 1000 merenja i to za svaku brzinu porasta napona od 1 V/s do 10 V/s. Kako je u ovom delu veći akcenat stavljen na uticaj zračenja odabran je jedan gasni odvodnik prenapona. Najpre mu je procenjen statički probojni napon pri normalnim atmosferskim uslovima a nakon toga je izložen uticaju gama zračenja i procenjen probojni napon u tako dobijenim novim uslovima.

Ono što se jasno može uočiti na slici 8. je da dolazi do smanjenja vrednosti statičkog probojnog napona nakon što se gasni odvodnik prenapona izloži izvoru gama zračenja, što je direktna posledica povećanja elektronskog prinosa u međuelektrodnom prostoru izazvano fotonima gama zračenja.



Slika 8. Vrednosti statičkog probojnog napona za gasni odvodnik pre i nakon izlaganja izvoru gama zračenja.

VII. ZAKLJUČAK

Cilj daljih istraživanja je da se ista merenja izvrše sa sličnim gasnim odvodnicima radi upoređivanja rezultata i pri novim eksperimentalnim uslovima, u prisustvu drugih izvora zračenja i tome slično.

Ovde je od značaja bilo prikazati značaj primene dinamičke metode za procenu statičkog probojnog napona i značaj njegovog određivanja. Utvrđeno je da se za svaku od korišćenih komponenata dobijala drugačija vrednost, ali je po tehničkoj dokumentaciji komponente potvrđena stabilnost napona i pouzdanost ovog tipa gasnih odvodnika.

S obzirom na široku primenu gasnih odvodnika prenapona potrebno je izvršiti što veći broj eksperimentalnih istraživanja kako bi se sa većom tačnošću utvrdila vrednost probojnog napona gasnog odvodnika koji će se kasnije koristiti.

ZAHVALNICA

Autor rada se srdačno zahvaljuje profesorima i saradnicima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu.

Posebna zahvalnost prof. dr Emiliji Živanović i asistentu Sandri Veljković koje su obezbedile sav potreban materijal za izradu ovog projekta kao i sugestije koje su uputile tokom izrade istog.

LITERATURA

[1] Momčilo M. Pejović Milić M. Pejović, Električni proboj gasova merni sistemi i eksperimentalna istraživanja, Niš: Elektronski fakultet u Nišu, 2009.

[2] Emilija Živanović, Sandra Veljković, Marija Živković and Milić Pejović, "Reliability of Various Type of Gas-filled Surge Arresters Under DC Discharge", vol. 11, pp. 2447, 2022.

[3] Marija Živković, N. Dimitrijević and E. Živanović, "Statistical Analysis of Breakdown Voltage of CITEL Gas- filled Surge Arrester," 2021 IEEE 32nd International Conference on Microelectronics (MIEL), Nis, Serbia, pp. 111-114, 2021.

[4] Marija Živković, "Pouzdanost EPCOS gasnog odvodnika prenapona", Proceeding of the 13th Student projects conference, IEEESTEC, Niš, Serbia, 2020.

[5] Dragan V. Brajović, "Optimizacija dinamičkih karakteristika gasnih odvodnika na niskonaponskom nivou", doktorska disertacija, Beograd 2014.

[6] Littlefuse datasheet, Dostupno na: https://eu.mouser.com/datasheet/2/240/media-3321228.pdf

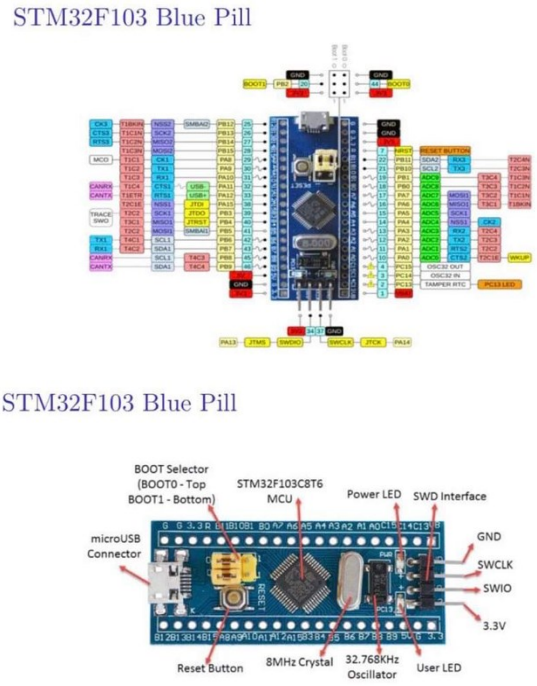
Kontrola servo motora korišćenjem ARM mikrokontrolera

Emilija Kotlo, Marta Stamenković

Sadržaj – U ovom radu dat je primer praktične realizacije mikrokontrolerskog sistema za kontrolu pozicije servo motora.. Za realizaciju korišćena su dva tastera koji kontrolišu okretanje servo motora. Pritiskom tastera 1 servo motor se okreće u smeru kazaljke na satu a pritiskom tastera 2 servo motor se okreće u smeru suprotnom od kazaljke na satu za 10°. Okretanje servo motora vrši se u opsegu od -30° do +30° u odnosu na njegovu početnu poziciju. Grafička interpretacija pozicije obezbeđena je pomoću 7 LE dioda.

I. Uvod

Okruženje koje je korišćeno u ovom zadatku je prikazano na slikama 1a i 1b.



Slika 1. a. Razvojno okruženje i raspored pinova, b. Razvojno okruženje i raspored komponenata.

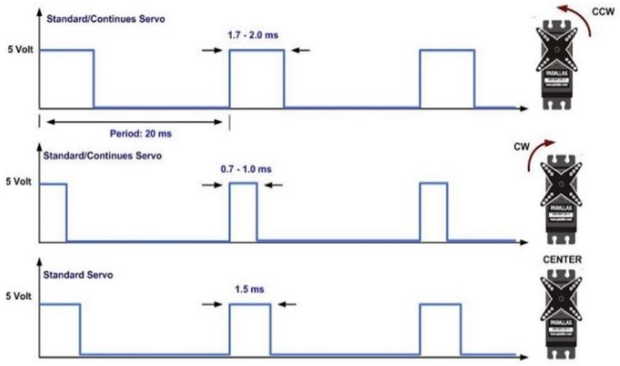
U ovom projektu korišćen je servo motor čija je frekvencija kontrolnog signala 50Hz, pa se odatle zaključuje da je perioda njegovog signala 20 ms. Servo motor kontroliše se pomoću PWM (Pulse Width Modulation) signala, odnosno modulacijom širine kontrolnog signala. Da bi servo motor bio u poziciji od 180° potrebno je da logička jedinica traje 2 ms a logička nula 18 ms. Faktor ispune signala u ovom slučaju je 10%. Za slučaj kada je servo motor u poziciji od 90°, potrebno je da u toku svake periode kontrolnog signala, signal bude 1.5 ms u stanju logičke jedinice. Takođe, da bi servo motor zauzeo poziciju od 0°, potrebno je da kontrolni signal bude u stanju logičke jedinice 1 ms, a 19 ms u stanju logičke nule. Na slici 2 je prikazana pozicija servo motora za odredjenu vrednost širine impulsa.

Ovaj servo motor kontroliše se pomoću Blue Pill razvojnog okruženja sa ARM mikrokontrolerom. Napaja se jednosmernim naponom vrednosti 5V. Da bi se podesila konfiguracija pinova ovog razvojnog okruženja koristi se program *CubeMX*, a za programiranje mikrokontrolera koristi se program *Keil*. Program u Keilu pisan je u programskom jeziku C.

E. Kotlo i M. Stamenković su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: emilijakotlo@elfak.rs, marta.stamenkovic@elfak.rs.

II. Konfiguracija tajmera mikrokontrolera

Upravljanje servo motora realizovano je generisanjem odgovarajućeg PWM signala. Za to se koristi tajmer mikrokontrolera. Da bi bilo moguće podešavanje tajmera u CubeMX-u, potrebno je odrediti preskaler i vrednosti ARR (Auto-Reload Register) registra [2]. Preskaler predstavlja delilac učestanosti. Inkrementacija sadržaja registra tajmera realizovana je pomoću registra ARR. Inkrementiranje sadržaja dovodi do prekida (interrupta). Za određivanje ovih vrednosti korišćene su sledeće relacije i unapred zadate vrednosti: T = 20ms zato što je frekvencija servo motora f = 50Hz. Faktor ispune određuje se za ugao od 90°.



Slika 2. Pozicija servo motora za određenu vrednost širine impulsa.

Pomoću faktora ispune nalazi se vrednost ARR regisra da bi vrednost Pulse bila ceo broj odnosno 75. Frekvenciju preskalera računamo iz formula:

$$f = 50Hz \rightarrow T = 20ms \quad (1)$$

$$DTC = \frac{1.5ms}{20ms}100\% = 7.5\% \quad (2)$$

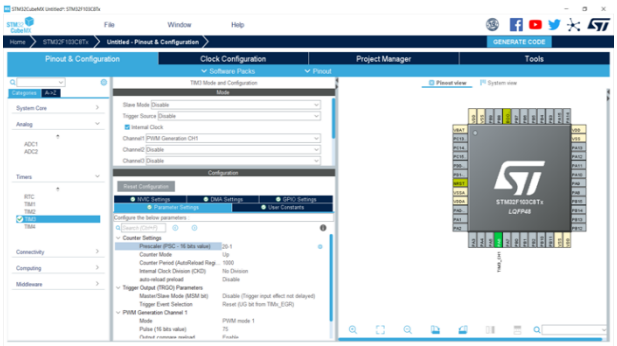
$$7.5 = \frac{Pulsse}{ARR}100 \quad (3)$$

$$T = 20ms = \frac{ARR}{fprescaler} = \frac{1000}{50*10^3} \rightarrow fprescaler = 50kHz \quad (4)$$

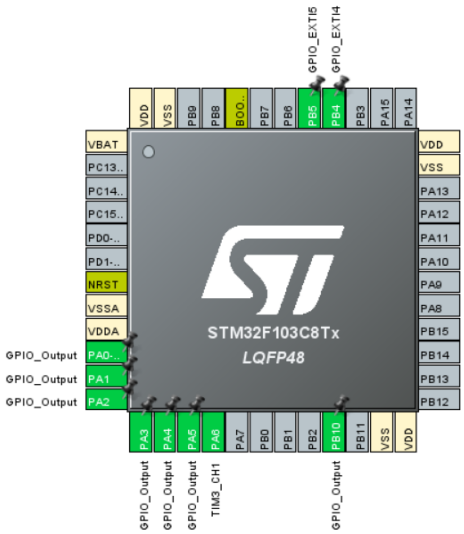
$$\frac{ftim}{prescaler} = fprescaler; ftim = 1MHz \rightarrow prescaler = 20 \quad (5)$$

Vrednost preskalera dobija se količnikom frekvencije preskalera i tajmera koja je podešena na 1 MHz.

Podešavanja za tajmer u program CubeMX prikazana su na slici 3. Ovim podešavanjima, prilikom generisanja koda predefinisane su vrednosti pojedinih pinova. Pinovi koji su korišćeni za realizaciju prikazani su na slici 4. To su: PA0, PA1, PA2, PA3, PA4 PA5 i PB10 koji kontrolišu LE diode, pinovi PB4 i PB5 očitavaju tastere, a pin PA6 služi za komunikaciju sa tajmerom.



Slika 3. Podešavanja parametara tajmera.

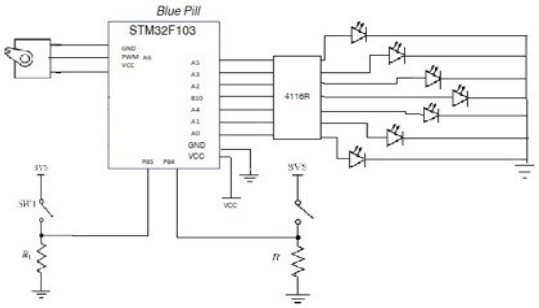


Slika 4. Korišćeni pinovi mikrokontrolera.

III. PROGRAMIRANJE MIKROKONTROLERA

Kao što je na početku naglašeno, rad ovog programa zasnovan je na pritisku tastera koji su povezani na pinove PB4 i PB5. Pomoću ovih pinova menja se smer kretanja servo motora i smer kojim se LE diode uključuju. U delu programa namenjenom za deklaraciju promenljve uvedena je promenljiva *uint32_t smer=75*. Ova promenljiva ima vrednost 75 jer je tolika vrednost *Pulse-a*, da bi se servo motor našao u položaju od 90 steoeni. Ukoliko smanjujemo ili povećavamo vrednost promenljive smer, servo motor se pokreće ulevo ili udesno u zadatom opsegu, od -30° do +30°.

Blok šema našeg zadatka prikazana je na slici 5, nakon toga napisan je kod za njeno izvršenje:



Slika 5. Električna šema zadatka.

U delu USER CODE BEGIN 2 koji predstavlja deo programa za kodove koji se izvršavaju samo jednom prilikom pokretanja programa, deklariše se tajmer TIM3 i generiše PWM signal preko kanala CHANNEL_1. Uključuje se LE diodu na pinu PB10 koji odgovara poziciji od 90°. Podešavanje je prikazano na slici 6.

```
90  /* USER CODE BEGIN 2 */
91  HAL_TIM_Base_Start(&htim3);
92  HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
93  __HAL_TIM_SetCompare(&htim3,TIM_CHANNEL_1,75); //ravnotezni polozaj
94  HAL_GPIO_WritePin(GPIOB,GPIO_PIN_10,1);
95  /* USER CODE END 2 */
```

Slika 6. Podešavanje parametara za tajmer u kodu.

Potom se prelazi na deo koda u kom se nalazi beskonačna petlja koja se sve vreme prilikom izvršenja koda ponavlja. U ovom delu koristi se funkcija *Set_Compare* koja služi za podešavanje vrednosti *duty cycle* na određenu vrednost i u zavisnosti od toga da li je povećavamo ili smanjujemo, menja se smer okretanja servo motora. Od argumenata redom imamo: pointer za handler na tajmer (&htim3), kanal na kome se aktivira PWM signal i vrednost promenljive smer.

Naredbom if(HAL_GPIO_ReadPin(GPIOB, GPIO_P_IN_4)==GPIO_PIN_RESET && smer<93) ispituje se da li je pritisnut levi taster i da li je vrednost promenljive smer manja od 93 jer ta vrednost odgovara pomeraju od 30° udesno. Ukoliko je taster pritisnut vrednost promenljive smer povećava se za 6 zato što ta vrednost

odgovara pomeraju od 10°. Ta vrednost dobijena je tako što krajnja leva i desna pozicija servo motora (gde je razlika u uglovima 180°) odgovaraju vrednostima promenljive *smer* od 25 i 125. Proporcijom je izračunata vrednost 6 koja se koristi u prethodno prikazanom kodu i ona mora da se inkrementira ili dekrementira u zavisnosti od toga koji je taster od dva moguća pritisnut. Određena LE dioda uključuje se u zavisnosti od toga koja je trenutna vrednost promenljive smer i u istom trenutnku se isključuje LE diode koja je do tada svetlela.

Naredbom else_if(HAL_GPIO_PeadPin(GPIOB, GPIO_PIN_5)==GPIO_PIN_RESET && smer>57) ispituje se da li je pritisnut desni taster i proces koji se primenjuje za levi taster je isti i kod desnog. Jedina razlika je u tome što se ovde ispituje je da li promenljiva smer ima vrednost veću od 57. Vredost od 57 odgovara pomeraju servo motora od 30° ulevo. U ovom slučaju vrednost promenljive taster se smanjuje za 6.

Deo programa koji je neophodan za realizaciju ovog projekta:

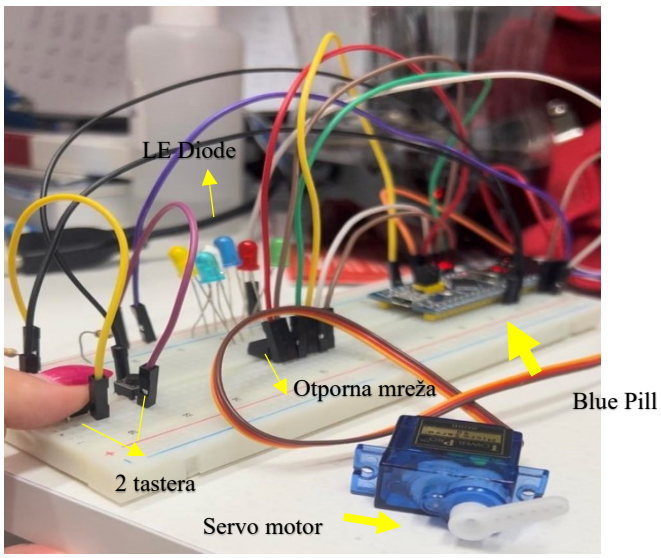
```
while (1)
{
  if(HAL_GPIO_ReadPin (GPIOB,GPIO_PIN_4 )==GPIO_PIN_RESET && smer<93)
  {

    //servo
    smer=smer+6;
    __HAL_TIM_SetCompare(&htim3, TIM_CHANNEL_1, smer);
    HAL_Delay(200);
    if(smer==57){
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
    }
    if(smer==63){
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
    }
    if(smer==69){
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 , 1);
    }
    if(smer==75){
      HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 , 0);
      HAL_GPIO_WritePin(GPIOB , GPIO_PIN_0, 1);
    }
    if(smer==81){
      HAL_GPIO_WritePin(GPIOB , GPIO_PIN_0, 0);
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 1);
    }
    if(smer==87){
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 0);
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
    }
   if(smer==93){
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 0);
      HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, 1);
    }

}
```

IV. Zaključak

Pomoću razvojnog okruženja Blue Pill-a i odgovarajućih dodatnih komponenata na protoploči realizovan je i testiran ovaj primer rada servo motora. Ovaj primer može biti sastavni deo složenijih sistema koji koriste servo motore u njihovom radu. Sistem može biti korišćen za kontrolu potrošača koji se upravljaju PWM signalima. Na slici 7 prikazan je završni proizvod našeg projekta.



Slika 7. Završni izgled projekta.

Zahvalnica

Autori rada zahvaljuju se profesorima i saradnicima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu. Posebnu zahvalnost dugujemo asistentima Milošu Marjanoviću i Milanu Stojanoviću koji su nam obezbedili potrebnu opremu i materijal za izradu ovog projekta kao i za znanje koje su nam prižili prilikom trajanja kursa Komponente i kola snage.

Literatura

[1] Teoretski i praktični deo PWM signala: I. Manić „*Komponente i kola snage*" Niš. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Komponente-i-kola-snage_predavanja.pdf
[2] Praktična realizacija i princip rada servo motora: https://www.automatika.rs/baza-znanja/mehatronika/rc-servo-motori.html?fbclid=IwAR3qh0Hsqm0QVGNs8WJV3DNvvQ6c41oMSCi2Mxpgb4j3DKMZ5UzMM8JS9sM
[3] Teorija o digitalnoj konverziji: https://mikro.elfak.ni.ac.rs/wp-content/uploads/Analogno-digitalna-konverzija-nastavak.pdf?fbclid=IwAR3cCDS8USP-8xGTy2mq3wTPIE2LIONGgS40HNfHFYr6b_q_kwcYhXefWVE
[4] Uputstvo za rad i korišćenje programa Keil: https://www.keil.com/_errors/e404.asp?aspxerrorpath=/support/man_arm.htm
[5] Datasheet STM32F103 Blue pill okruženja sa ARM mikrokontrolerom: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf?fbclid=IwAR14XXpz8o-YGrudkOl7qS_8vnHA3r84HNxH__VltCK6wDMG0Ag0t65Opak
[6] Tehnička dokumentacija za mikrokontroler STM32F103RBTx

Projektovanje i implementacija aplikacije za vizualizaciju topologije lokalnih računarskih mreža

Vukašin Dobromirović

Sadržaj – U ovom radu je prikazana aplikacija pomoću koje je moguće vizualizovati računarske mreže u cilju boljeg razumevanja.

I. UVOD

U današnjem digitalnom dobu, mrežna infrastruktura predstavlja ključnu stavku za funkcionisanje modernih organizacija i preduzeća. Upravljanje mrežom je postalo neophodno kako bi se osigurala stabilnost, bezbednost i efikasnost komunikacije između različitih uređaja i servera. U tom kontekstu, praćenje topologije mreže i identifikacija povezanih uređaja je od suštinskog značaja.

Izazov koji se nameće u ovoj oblasti je kako efikasno prikupiti informacije o uređajima na mreži, identifikovati njihove veze i konfiguraciju, te na kraju, prikazati ove podatke na način koji je pregledan i informativan. Tradicionalno, ovakvo prikupljanje informacija zahteva ručno interagovanje sa svakim uređajem, što može biti vremenski zahtevno i podložno greškama.

U cilju rešavanja ovog problema, ovaj rad predstaviće aplikaciju koja omogućava virtualizaciju topologije lokalnih racunarskih mreža. Ovaj program integriše različite biblioteke i tehnologije kako bi efikasno obavio ovaj zadatak. Kroz analizu i prikazivanje topologije mreže, ovaj alat omogućava administratorima da bolje razumeju kako su uređaji povezani, identifikuju potencijalne probleme i brže reaguju na potrebe za održavanjem mreže.

II. PREGLED TEHNIKA NEOPHODNIH ZA IMPLEMENTACIJU APLIKACIJE

A. Cisco operativni system (Cisco IOS)

Cisco [1] operativni sistem, ili skraćeno Cisco IOS [2], je operativni sistem koji se koristi na Cisco mrežnim uređajima kao što su ruteri i svičevi. Ovaj operativni sistem je ključan za upravljanje mrežom i obezbeđuje korisnicima širok spektar funkcionalnosti za konfiguraciju i upravljanje uređajima. U implementaciji programa, Cisco OS je osnova za komunikaciju sa Cisco uređajima putem SSH protokola.

V. Dobromirović, student treće godine osnovnih studija na modulu Računarstvo i informatika Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: vukasindobromirovic@elfak.rs

B. CDP komanda

CDP (*Cisco Discovery Protocol*) [3] je Cisco-ov protokol za otkrivanje susednih uređaja u mreži. Ovaj protokol omogućava uređajima da razmenjuju informacije o svojim identitetima, povezanim interfejsima i drugim relevantnim podacima. Aplikacija projektovana u ovom radu koristi CDP komandu "*show cdp neighbors detail*" kako bi prikupila informacije o susednim uređajima, interfejsima i IP adresama.

C. SSH protocol (Secure Shell)

SSH (*Secure Shell*) [4] je protokol za sigurnu komunikaciju i daljinsko upravljanje sa udaljenim uređajima u mrežnom okruženju. Osnovna svrha SSH protokola je omogućavanje bezbedne razmene podataka između klijenta (npr. računar i administrator) i udaljenih servera ili uređaja, kao što su svičevi, ruteri ili serveri. Prilikom povezivanja sa uređajima, SSH omogućava autentikaciju korisnika i uspostavljanje sigurne veze. To znači da se informacije kao što su korisnička imena, lozinke i prikupljeni podaci šifriraju kako bi se sprečilo prisluškivanje ili neovlašćeni pristup. Osim toga, SSH pruža mehanizme za autentikaciju i zaštitu integriteta podataka.

D. BFS algoritam (Breadth-First Search)

BFS (*Breadth-First Search*) [5] je algoritam za pretraživanje ili kretanje kroz stablo ili graf strukturu podataka. U kontekstu aplikacije, BFS se koristi za skeniranje mreže. Počevši od ciljnog uređaja (*gateway*) [6], BFS se širi na susedne uređaje i prikuplja informacije o njihovim povezanim uređajima i interfejsima. Ovaj algoritam omogućava temeljno istraživanje mreže i identifikaciju svih povezanih uređaja.

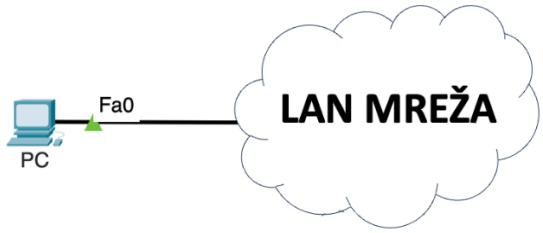
Pregled ovih ključnih tehnika i tehnologija omogućava bolje razumevanje implementacije i kako program efikasno prikuplja informacije o mreži. Ovaj pregled će takođe postaviti temelje za kasnije delove rada koji će se baviti detaljima implementacije i praktičnom primenom programa.

III. PROJEKTOVANJE APLIKACIJE

A. Princip ulazne tačke

Program je od značaja za sve mrežne administratore u cilju boljeg razumevanja i upravljanja mrežom. Veza se

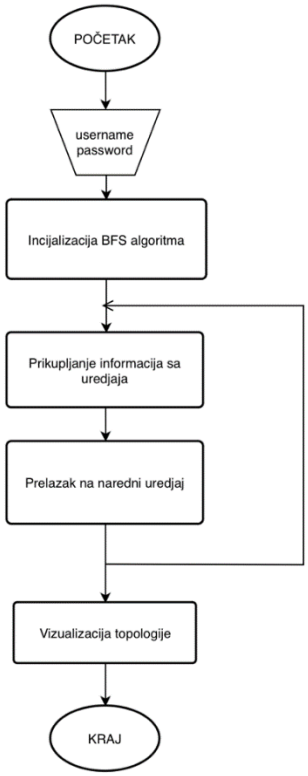
uspostavlja sa gejtvejom, koji predstavlja ulaznu tačku na mrežu. Na slici 1 se može videti slikovit opis. Nakon što je veza sa gejtvejom uspostavljena, CDP (*Cisco Discovery Protocol*) naredba se koristi za postepeno otkrivanje uređaja i ostatka mreže. Na ovaj način, mrežna mapa se sistematski gradi putem programa, omogućavajući administratorima da steknu dublji uvid u topologiju mreže i identifikaciju uređaja i resursa koji čine deo lokalne mreže.



Slika 1. Princip povezivanja na ulaznoj tački.

B. Algoritam aplikacije

Na slici 2 je prikazan algoritam rada predložene aplikacije. Na početku izvršenja korisnik zadaje korisničko ime i šifru za ssh pristup uređajima. Podrazumeva se da administrator ima iste kredencijale za pristup svim uređajima.



Slika 2. Algoritam aplikacije.

Nakon unosa potrebnih podataka poziva se funkcija *get_default_gateway* pomoću koje se saznaje ip adresa gejtveja. Inicijalizuje se BFS (*Breadth- First Search*) algoritam i za početni čvor se uzima adresa gejtveja. BFS algoritam se primenjuje na mrežu kako bi se sistematski otkrivali uređaji i njihove veze. Za svaki pronađeni uređaj se izvršava CDP komanda kako bi se saznale dodatne informacije o susedima i interfejsima. Informacije o svakom uređaju, uključujući IP adrese, interfejse i veze, prikupljaju se tokom izvršavanja BFS algoritma. Ove informacije se beleže i čuvaju za kasniju analizu. Nakon završetka skeniranja i prikupljanja informacija, razmatra se način vizualizacije topologije mreže. Odgovarajuće biblioteke za crtanje grafova ili druge tehnike koriste se za grafički prikaz rezultata. Algoritam se testira na različitim mrežama kako bi se osiguralo da ispravno funkcioniše i da zadovoljava očekivanja.

IV. KREIRANJE APLIKACIJE

U nastavku će biti predstavljene određene funkcije i biblioteke koje su ključne za aplikaciju.

A. Potrebne biblioteke kao pomoć u izradi aplikacije

Na početku biće upotrebljeni neki standardni paketi u programskom jeziku Python. Paramiko [7] je *python* biblioteka koja omogućava implementaciju SSH protokola. Netmiko [8] je *python* biblioteka koja se koristi za automatizaciju mrežnih uređaja. Omogućava programerima da se povežu sa različitim mrežnim uređajima. *Networkx* [9] je *Python* biblioteka za analizu i manipulaciju grafova i mreža. Koristi se za proučavanje i vizualizaciju odnosa između entiteta u mreži, kao što su uređaji ili čvorovi, i njihove veze ili grane. *Matplotlib* [10] je *Python* biblioteka za crtanje grafova i dijagrama. Koristi se za vizualizaciju podataka i rezultata, uključujući grafove mreža, histograme, dijagrame rasipanja i mnoge druge vrste grafičkih prikaza. Na slici 3 je prikazan isečak koda kojim se uključuju u aplikaciju već pomenute biblioteke.

```
import paramiko
from netmiko import ConnectHandler
import networkx as nx
import matplotlib.pyplot as plt
```

Slika 3. Potrebni paketi u programskom jeziku *Python*.

B. Funkcija za čitanje podrazumevane adrese uređaja sa koga se pokreće aplikacija

Funkcija *get_default_gateway* ima za cilj da pronađe podrazumevani gejtvej odnosno ip adresu rutera mreže na kojoj je povezan uređaj sa koga se pokreće aplikacija. Kod funkcije se može videti na slici 4.

```
def get_default_gateway():
    try:
        output = os.popen("route
        print").read()
        gateway_pattern = r"0\.0\.0\.
        0\s+0\.0\.0\.0\s+(\d+\.\d+\.\d
        +\.\d+)\s+"
        match = re.search
        (gateway_pattern, output)
        if match:
            return match.group(1)
    except Exception as e:
        print(f"Greška: {e}")
    return None
```

Slika 4. Funkcija *get_default_gateway.*

C. Funkcija za uspostavljanje veze sa uređajem i prikupljanje informacija o samom uredjaju

Funkcija *gather_device_info* ima zadatak da prikupi informacije o udaljenom mrežnom uređaju putem SSH protokola. Funkcija koristi biblioteke *paramiko* i *netmiko* koje su objašnjene na početku poglavlja. Kod funkcije se može videti na slici 5.

```
def gather_device_info(device_ip, username, password,
enable_password):
    # Parametri za SSH konekciju
    ssh_params = {
        'device_type': 'cisco_ios',
        'ip': device_ip, 'username': username,
        'password': password, 'secret' : enable_password
    }
    try:
        ssh_session = ConnectHandler(**ssh_params)
        output = ssh_session.send_command('show cdp
        neighbors detail')
        ssh_session.enable()
        shrun = ssh_session.send_command('show
        running-config')
        ssh_session.disconnect()

        return output,shrun
    except Exception as e:
        print(f"Greška pri povezivanju ili izvršavanju
        komande na uređaju {device_ip}: {str(e)}")
```

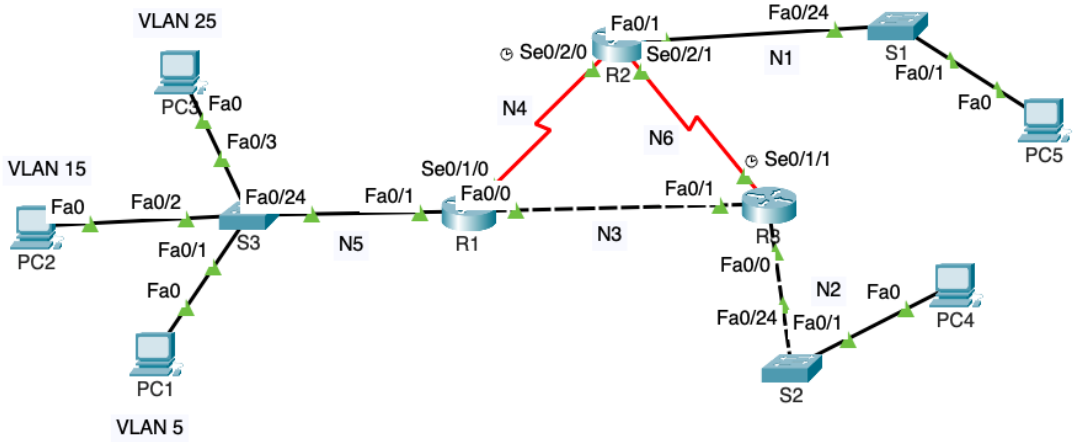
Slika 5. Funkcija *gather_device_info*.

D. Funkcija za iscrtavanje topologije

Funkcija *draw_custom_graph* se koristi za iscrtavanje grafa koji prikazuje mrežnu topologiju na osnovu podataka o čvorovima (uređajima) i granama (vezama) između njih. Ova funkcija koristi biblioteku *networkx* za manipulaciju grafovima i matplotlib za crtanje grafičkog prikaza. Detaljan kod možemo videte na slici 6.

```
def draw_custom_graph(nodes, edges, ip_addresses=None):
    G = nx.DiGraph()
    G.add_edges_from(edges)
    node_labels = {}
    for node in nodes:
        label = node
        if ip_addresses and node in ip_addresses:
            label += f'\n{ip_addresses[node]}'
        node_labels[node] = label
    edge_colours = ['black']
    pos = nx.spring_layout(G)
    node_font = {'fontname': 'Arial', 'size': 12}
    nx.draw_networkx_nodes(G, pos, cmap=plt.get_cmap
    ('jet'), node_color='lightblue', node_size=500)
    nx.draw_networkx_labels(G, pos, labels=node_labels,
    font_color='black', font_size=8, font_weight='bold',
                            font_family=node_font
                            ['fontname'])
    nx.draw_networkx_edges(G, pos, arrows=True,
    edge_color=edge_colours)
    plt.show()
```

Slika 6. Funkcija za iscrtavanje topologije.



Slika 7. Mreža nad kojom je izvršeno testiranje.

V. SLUČAJEVI UPOTREBE

Aplikacija za vizualizaciju topologije lokalnih računarskih mreža je realizovana i testirana na Elektronskom fakultetu u okviru Cisco akademije [11] u laboratoriji 527. Za kreiranje aplikacije korišćen je Python programski jezik i korišćeni su sledeći mrežni uređaji: *Cisco 2811* ruteri i *Cisco Catalyst 2960* svičevi, prikazani na slikama 8 i 9.



Slika 8. Cisco Catalyst 2960.



Slika 9. Cisco 2811.

A. Primer mreže na kojoj je izvršeno testiranje

Aplikacija će biti testirana na topologiji prikazanoj na slici 7. Uočavamo šest mreža označenim simbolima N1, N2 redom do N6. Svaka mreža ima određeni opseg adresa. Za N1 je to 172.16.1.0/24, za N2 je 172.16.2.0/24, za N3 je 192.168.3.0/30, za N4 je 192.168.4.0/30, za N6 je 172.16.6.0/24. Što se tiče mreže N6 tu se koristi *Router-on-stick* [13] tehnika. Na mreži se mogu primetiti VLAN 5, VLAN 15, VLAN 25, svaki redom u opsegu 10.0.5.0, 10.0.15.0 i 10.0.25.0 sa istom maskom /24.

B. Izvršenje aplikacije i analiza izlaznih podataka

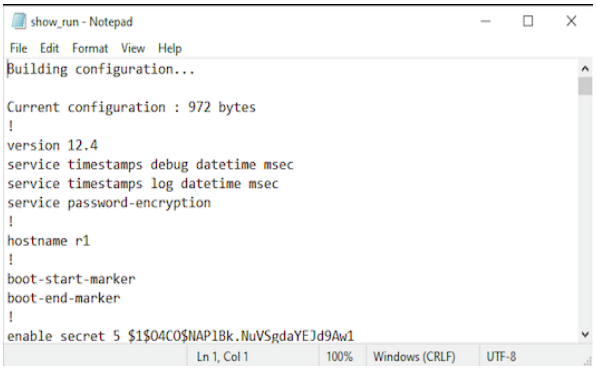
Nakon pokretanja aplikacije od korisnika se očekuje da unese podatke kao što su *username*, *password* i *password privileged mode* [12]. U međuvremenu, aplikacija pokreće prethodno opisanu funkciju *get_default_gateway* i saopštava ip adresu rutera odakle počinje skeniranje. U našem primeru je to ruter R1 pošto je aplikacija pokrenuta sa računara označenim PC1. Potrebno je zatim pritisnuti taster *Enter* i aplikacija pokreće skeniranje mreže.

Nakon završetka aplikacije u konzoli se ispisuju rezultati. Za svaki uređaj, bilo to da je svič ili ruter, dobijamo detaljne informacije kako je povezan sa svojim susedom, odnosno preko kog interfejsa. Konzolni izlaz aplikacije je prikazan na slici 10.

```
Uređaj čiji je hostname "r1" i ip adresa "10.0.5.1" je:
Povezan preko interfejsa "FastEthernet0/1" sa uređajem čiji je hostname "s3".
Povezan preko interfejsa "Serial0/1/0" sa uređajem čiji je hostname "r2".
Povezan preko interfejsa "FastEthernet0/0" sa uređajem čiji je hostname "r3".
Uređaj čiji je hostname "s3" i ip adresa "10.0.5.6" je:
Povezan preko interfejsa "FastEthernet0/24" sa uređajem čiji je hostname "r1".
Uređaj čiji je hostname "r2" i ip adresa "192.168.4.2" je:
Povezan preko interfejsa "Serial0/2/1" sa uređajem čiji je hostname "r3".
Povezan preko interfejsa "Serial0/2/0" sa uređajem čiji je hostname "r1".
Povezan preko interfejsa "FastEthernet0/1" sa uređajem čiji je hostname "S1".
Uređaj čiji je hostname "r3" i ip adresa "192.168.3.2" je:
Povezan preko interfejsa "Serial0/1/1" sa uređajem čiji je hostname "r2".
Povezan preko interfejsa "FastEthernet0/1" sa uređajem čiji je hostname "r1".
Povezan preko interfejsa "FastEthernet0/0" sa uređajem čiji je hostname "S2".
Uređaj čiji je hostname "S1" i ip adresa "172.16.1.100" je:
Povezan preko interfejsa "FastEthernet0/24" sa uređajem čiji je hostname "r2".
Uređaj čiji je hostname "S2" i ip adresa "172.16.2.100" je:
Povezan preko interfejsa "FastEthernet0/24" sa uređajem čiji je hostname "r3".
```

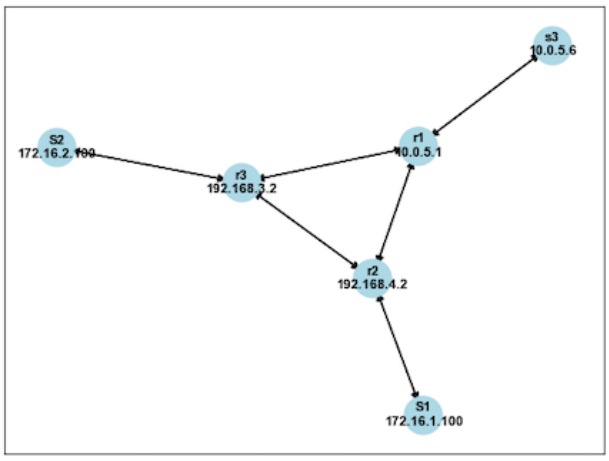
Slika 10. Konzolni prikaz nakon završetka aplikacije.

Takođe, može se uočiti kreirani fajl po nazivom *show_run.txt.* Fajl sadrži detaljni opis svakog uređaja na kome je bio izvršen skok. Na slici 11 je prikazan i sam fajl.



Slika 11. Fajl sa detaljnom konfiguracijom uređaja.

Treći izlazni podatak je vizualizacija mreže kreirane prethodno opisanom fukcijom *draw_custom_graph.* Uočava se povezanost koja je identična konzolnom prikazu. Topologija je prikazana na slici 12.



Slika 12. Vizualizacija mreže.

VI. ZAKLJUČAK

Mreže postaju sve kompleksnije što čini važnim razumevanje kako su uređaji međusobno povezani i kako se saobraćaj kreće kroz mrežu. Ova aplikacija pruža sredstva za brzo i efikasno otkrivanje uređaja u mreži, analizu njihovih povezanosti i prikazivanje tih informacija u obliku grafičkog prikaza. Time se olakšava dijagnostika problema, optimizacija mreže i bolje razumevanje njenih karakteristika. U budućnosti, aplikacija se može proširiti i dodatno unaprediti dodavanjem naprednih funkcionalnosti, kao što su analiza sigurnosti mreže, automatsko konfigurisanje uređaja ili integracija sa drugim alatima za upravljanje mrežom. Ovaj projekat služi kao temelj za dalje istraživanje i razvoj u oblasti mrežnih tehnologija.

ZAHVALNICA

Autor se zahvaljuje prof. dr Vladimiru Ćiriću na mentorstvu i podršci i prof. dr Emiliji Živanović na pomoći pri pisanju rada. Takođe, autor se zahvaljuje i Cisco akademiji na korišćenju laboratorije 527.

LITERATURA

[1] Cisco: Networking, Cloud, and Cybersecurity Solutions n.d. Poslednji pristpu 27.10.2023. https://www.cisco.com
[2] Cisco Networking Academy, Cisco. Poslednji pristup 27.10.2023. https://lms.netacad.com
[3] Cisco n.d. *Understanding and configuring CDP*. Poslednji pristup 27.10.2023. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/cdp.pdf
[4] Cisco Networking Academy, Cisco. Poslednji pristup 27.10.2023. https://lms.netacad.com
[5] https://en.wikipedia.org/wiki/Breadth-first_search
[6] Wikipedia: The free Encyclopedia n.d. Poslednji pristup 27.10.2023. https://en.wikipedia.org/wiki/Default_gateway
[7] Paramiko n.d. Poslednji pristup 27.10.2023. https://pypi.org/project/paramiko/
[8] Kirk Byers [2023]. netmiko: Kolekcija informacija korišćenih u projektu. Github. https://github.com/ktbyers/netmiko
[9] NetworkX – Network Analysis in Python 2014. Poslednji pristup 27.10.2023. https://networkx.org.
[10] Matplotlib: Visualisation with Python 2012. Poslednji pristup 27.10.2023. https://matplotlib.org.
[11] Cisco networking academy n.d. Poslednji pristup 27.10.2023. https://www.netacad.com
[12] Controlling Switch Access with Passwords and Privilege Levels, Cisco. Poslednji pristup 27.10.2023. https://www.cisco.com/en/US/docs/
[13] Wikipedia: The free Encyclopedia n.d. Poslednji pristup 27.10.2023. https://en.wikipedia.org/wiki/Router_on_a_stick

Praktična realizacija signal generatora pomoću analognih komponenata

Kristina Mišić, Vuk Zlatković

Sadržaj – U ovom radu biće prikazana konstrukcija blokova koji učestvuju u izgradnji signal generatora. Kola su bazirana na primeni operacionog pojačavača LM2902 [7]. Signal generator može da generiše signale oblika sinusoide, trougla, testere, četvrtke. Prvo je određena simulacija rada tih blokova, a zatim i praktična realizacija.

I. Uvod

Signal generator predstavlja elektronski instrument koji služi za generisanje signale koji služe kao pobuda uređaja koji se testira, za dizajniranje, popravke elektronskih uređaja.

Projektovani/opisani generator signala se sastoji iz sledećih blokova:

- naponski regulator simetričnog napajanja [1][2];
- oscilator sa vinovim mostom [1];
- relaksacioni oscilator (testerasti signal) [1];
- naponski kontrolisani generator signala (četvrtasti i trougaoni signal) [1];

Ovi blokovi su dodati sa ciljem generisanja podesivih sinusnih, testerastih, trougaonih i četvrtastih signala.

U poglavlju II biće opisan naponski regulator simetričnog napajanja, u III poglavlju biće opisan oscilator sa vinovim mostom, u IV poglavlju biće opisan relaksacioni oscilator, a u V poglavlju biće opisan naponski kontrolisani oscilator (VCO).

II. Naponski regulator simetričnog napajanja

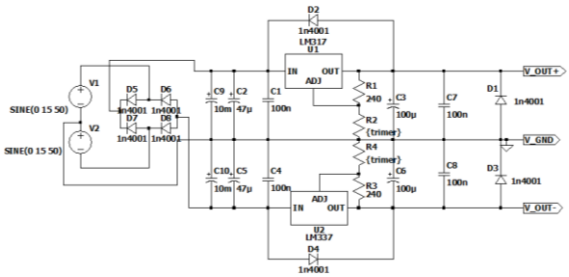
Za regulisanje pozitivnog napona se koristi integrisano kolo LM317T [5] (slika 1.), a za regulisanje negativnog napona se koristi kolo LM337[6] (slika 1.). Ovde će biti objašnjeno samo kako je kolo LM317T primenjeno za regulaciju pozitivnog napona, zato što je primena kola LM337 identična [1][2].

U sklopu kola za regulaciju (slika 1.) nalaze se filterski blokovi koji se sastoje od *bypass* kondenzatora (C_1 i C_7 od 100 nF) koji služe za eliminaciju neželjenih signala visoke učestanosti. Elektrolitski kondenzatori (*C2* od 10 mF, *C3* od 100 µF, *C9* od 47 µF) su veće kapacitivnosti i oni pored svoje filterske uloge služe da kompenzuju kratkotrajne propade i eventualne prekide mrežnog napona.

Kolo LM317T poseduje tri izvoda (IN, OUT i ADJUSTABLE), preko kojih reguliše napon na sledeći način: Regulator je projektovan tako da se između izlaza OUT i ADJ, na otporniku R_1, uvek pojavljuje referentni napon V_{REF}. Izlazni napon je određen relacijom (1):

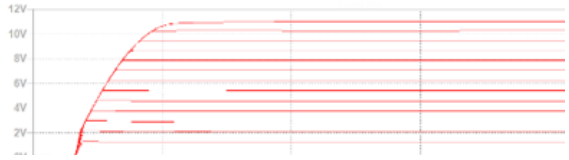
$$V_{OUT} = V_{REF} * (1 + \frac{R_2}{R_1}) \quad (1)$$

Tipično, V_{REF} = 1,25 V i predstavlja temperaturno nezavisnu naponsku referencu (*bandgap voltage reference*), a vrednost otpornosti R_1 je zadata u tehničkoj dokumentaciji.



Slika 1. Šema naponskog regulatora simetričnog napajanja.

Dioda D_1 ima ulogu da zaštiti kolo regulacije napona od napona inverzne polarizacije iz potrošača. Dioda D_2 sprečava pojavu inverznog napona na regulatoru: ako na ulazu regulatora dođe do kratkog spoja, izlazni kondenzator C_3 će se prazniti preko diode D_2, a ne kroz regulator.



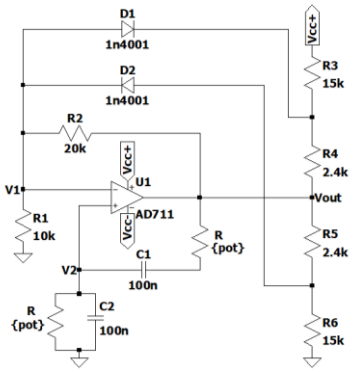
Slika 2. Zavisnost pozitivnog i negativnog izlaznog napona (V_OUT+) od vrednosti otpornosti R_2.

Otpornici R_2 i R_4 su u praktičnoj realizaciji zamenjeni stereo potenciometrom nominalne vrednosti od 5 kΩ i prilagođenjem njegove otpornosti se istovremeno podešava i pozitivni i negativni napon (slika 2).

III. Oscilator sa vinovim mostom

U projektovanom kolu se koristi oscilator sa vinovim mostom (slika 3) za generaciju sinusnog signala [1].

K. Mišić i V. Zlatković su studenti 4. godine na modulu Elektronske komponente i mikrosistemi na Elektronskom fakultetu; Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: kristina.misic@elfak.rs, vuk.zlatkovic@elfak.rs



Slika 3. Šema oscilatora sa Vinovim mostom.

Kolo ima negativnu i pozitivnu povratnu spregu. Kolo negativne povratne sprege se može posmatrati kao neinvertujuća konfiguracija:

$$Vout = \left(1 + \frac{R_2}{R_1}\right) V1. \tag{2}$$

Kolo pozitivne povratne sprege predstavlja RC invertor faze, čiji je ulazni napon V_{out}:

$$\frac{V1}{Vout} = \frac{1}{3+j\left(\omega RC - \frac{1}{\omega RC}\right)}. \tag{3}$$

Prema teoriji pojačavača, kolo se može apstrahovati kao pojačavač sa pozitivnom povratnom spregom kod koga je $A = \frac{V_{out}}{V_1}$ i $B = \frac{V_1}{V_{out}}$. Pojačanje u petlji je:

$$AB = \frac{1+\frac{R2}{R1}}{3+j\left(\omega RC - \frac{1}{\omega RC}\right)}. \tag{4}$$

S druge strane, prema datim definicijama je $AB = 1$. Na učestanosti:

$$f_r = \frac{1}{2\pi RC}, \tag{5}$$

je fazni pomeraj 0°, a pojačanje u petlji je:

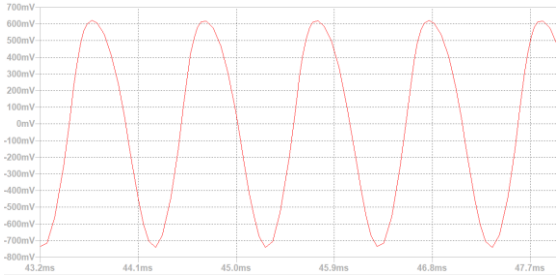
$$AB = \frac{1+\frac{R2}{R1}}{3}. \tag{6}$$

Kada je $\frac{R_2}{R_1} = 2$, tada je $AB = 1$. Time je ispunjen *Barkhausen*- ov kriterijum. Komponente oko operacionog pojačavača čine Vinov most u ravnoteži.

Oscilator sa Vinovim mostom generiše signal oblika sinusa (slika 4) učestanosti f_r [5].Oscilacije nastaju odmah nakon dovođenja napajanja. U praksi, zbog toga što komponente nisu idealne, teško je postići ravnotežu mosta. Ako je $AB < 1$ negativna povratna sprega će prevladati nad pozitivnom, pa će kolo biti u stabilnom stanju. Ako je $AB > 1$ dominiraće pozitivna povratna sprega, pa će kolo oscilovati sa sve većom amplitudom, sve do napona zasićenja operacionog pojačavača. Zbog toga je potrebno kontrolisati amplitudu, tako da je prilikom dovođenja napajanja $\frac{R_2}{R_1} \gtrsim 2$, kako bi se osiguralo nastajanje oscilacija, a zatim bi ovaj odnos trebalo da se smanjuje tako da teži vrednosti $\frac{R_2}{R_1} \lesssim 2$, kako bi se sprečio prekomerni porast amplitude. Kada bude $\frac{R_2}{R_1} = 2$, kolo će se automatski stabilisati sa konstantnom amplitudom.

Kontrola amplitude se vrši pomoću dioda D_1 i D_2 na sledeći način. Po dovođenju napajanja je $|V_{out}| \gtrsim 0\text{V}$, diode su zakočene, pa R_3 nema uticaja na kolo povratne sprege. Tada je $\frac{R_2}{R_1} = 20$, $\frac{3}{10} = 2$, $03 \gtrsim 2$, čime se obezbeđuje nastajanje oscilacija i postepeni porast amplitude V_{out}.

Usled porasta amplitude V_{out} diode će početi da provode (naizmenično, svaka u odgovarajućoj poluperiodi), pa bi se u krajnjem slučaju umesto R_2 pojavila paralelna veza $R_2e = R_2 \parallel R_3$, tako da je $R_2e/R_1 \lesssim 2$. Kolo se pre toga automatski stabilizuje kada je $R2e/R1 = 2$, čime se sprečava dalji porast amplitude (slika 4).

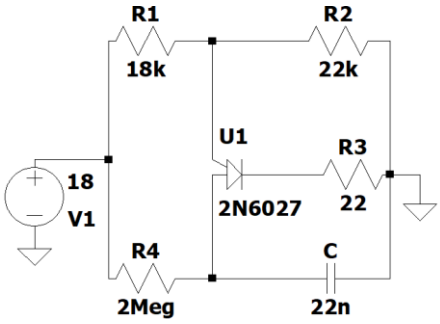


Slika 4. Prikaz izlaznog signala oscilatora sa Vinovim mostom sa kontrolisanom amplitudom.

IV. RELAKSACIONI OSCILATOR

Za generisanje testerastog signala u projektovanom kolu se koristi relaksacioni oscilator sa PUT – om [1].

Glavne komponente relaksacionog oscilatora su PUT 2n6027 [8] (programabilni jednospojni tranzistor) i kondenzator C (slika 5).



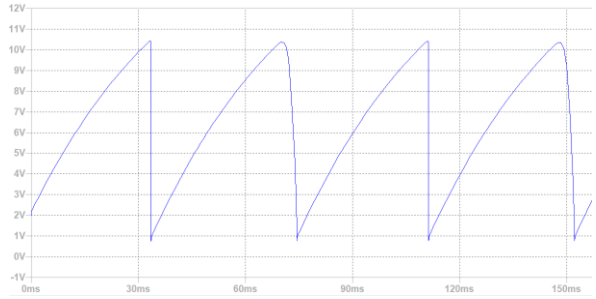
Slika 5. Šema relaksacionog oscilatora sa PUT – om.

Programabilni jednospojni tranzistor je tiristorska struktura sa tri elektrode koje se nazivaju anoda (A), katoda (K) i gejt (G). Gejt je uvek pozitivno polarisan u odnosu na katodu. Kada napon na anodi bude iznad napona na gejtu za vrednost ugradjenog napona pn spoja, tranzistor provodi struju. Po dovođenju napajanja V_{gg}, kondenzator C_1 počinje da se puni preko otpornika R_4. Kada se napuni do vrednosti V_p (napon programiranja PUT- a), PUT (programabilni jednospojni tranzistor) se uključuje.

$$V_p = V_s + V_d \tag{7}$$

Kondenzator se naglo prazni kroz PUT, preko otpornika R_3. Kad se kondenzator isprazni, PUT se isključuje, kondenzator se ponovo puni preko R_4 i ciklus se ponavlja.

Dakle, relaksacioni oscilator koristimo za generisanje signala oblika testere (slika 6).

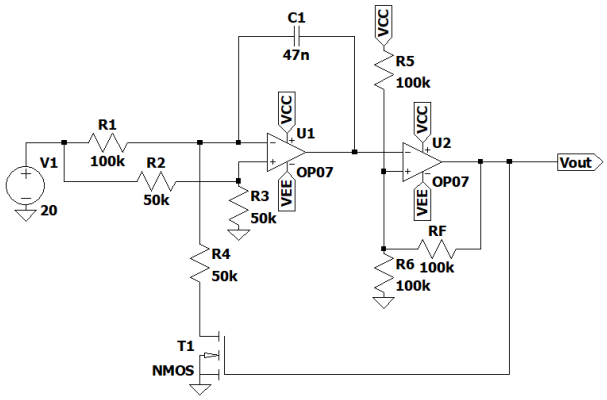


Slika 6. Prikaz izlaznog signala relaksacionog oscilatora.

Promena napona na kondenzatoru (slika 6): napon na kondenzatoru C eksponencijalno raste, a zatim naglo opada.

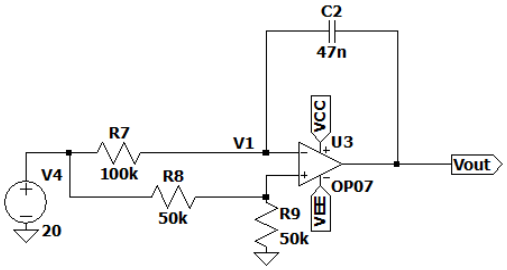
V. NAPONSKI KONTROLISANI OSCILATOR (VCO)

Za generisanje četvrtastog i trougaonog signala se u projektovanom kolu koristi naponski kontrolisani oscilator [1]. Učestanost oscilacija se kontroliše ulaznim naponom. Oscilator se sastoji od integratora, Šmitovog kola i tranzistora T1(IRF 510 [4]) koji radi kao prekidač (slika 7).



Slika 7. Šema naponski kontrolisanog oscilatora.

Kada je tranzistor zakočen, ekvivalentno kolo integratora je (slika 8) :



Slika 8. Ekvivalentna šema integratorskog bloka kada je MOS tranzistor zakočen.

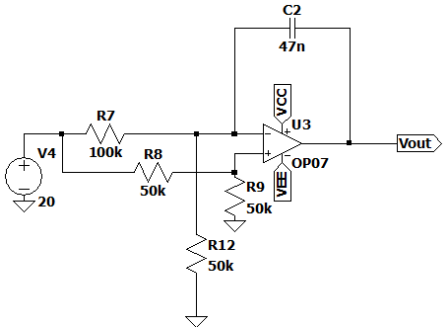
Struja kojom se puni kondenzator, po definiciji:

$$i_{C2} = C2\frac{d(V1-Vout)}{dt}. \tag{8}$$

Na izlazu integratora dobija se napon V_{out} koji opada sa vremenom do – VSAT.

$$Vout = -\frac{V4}{4RC}t + C_{int}. \tag{9}$$

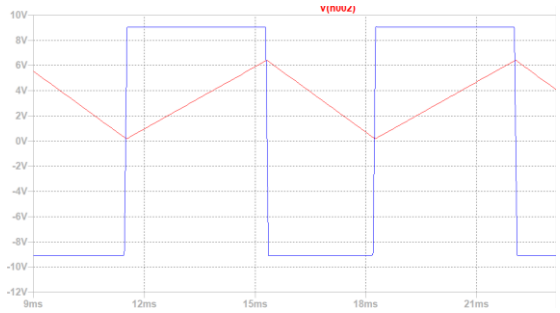
C_{int} je konstanta integracije. Njen fizički smisao predstavljen je naponom V_2, pa je C_{int}=V_2=Vin/2. Kad je T_1 uključen, ekvivalentno kolo integratora je (slika 9) :



Slika 9. Ekvivalenta šema integratorskog bloka naponski kontrolisanog oscilatora kada je MOS tranzistor u zasićenju.

Smatra se da je napon između sorsa i drejna tranzistora zanemarljiv. Na izlazu integratora dobijamo *Vtriangle* koji raste do napona zasićenja +VSAT.

Na ovaj način dobija se trougaoni signal (slika 10). Trougaoni signal se dovodi na ulaz Šmitovog kola, koje radi kao komparator. Napon na izlazu operacionog pojačavača V_{out} može biti pozitivni ili negativni napon zasićenja operacionog pojačavača (slika 11). Komparator će menjati stanje na izlazu kada napon na invertujućem ulazu $V_{triangle}$ dostigne napon na neinvertujućem ulazu operacionog pojačavača. Na taj način dobićemo četvrtasti signal (povorku impulsa).

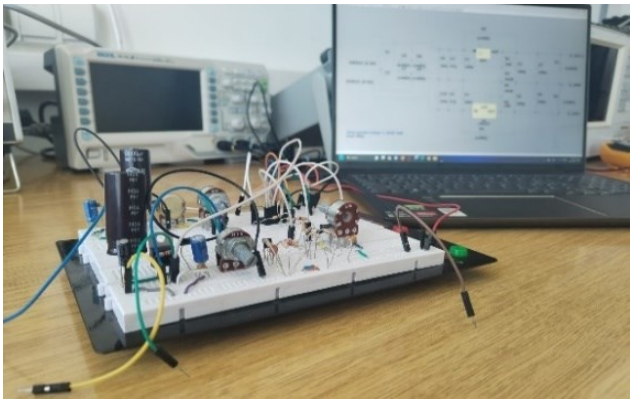


Slika 11. Prikaz izlaza trougaonog signala i četvrtastog sa naponski kontrolisanog oscilatora.

Od amplitude ulaznog napona zavisi strmina generisanih trouglova na izlazu integratora, a sama strmina trouglova ce određivati napone pri kojima će Šmitovo kolo da okida (brže ili sporije) (slika 11).

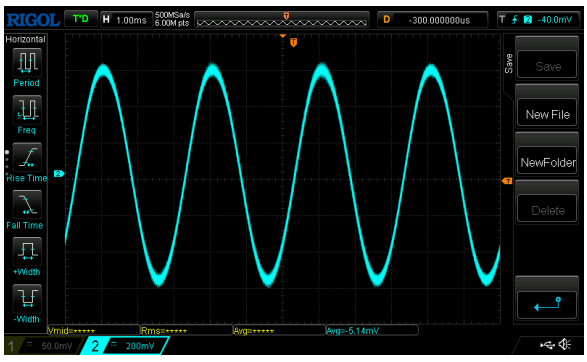
VI. Praktična realizacija

Opisane šeme (slike 1, 3, 5 i 7) su praktično realizovane kao što je prikazano na slici 12:

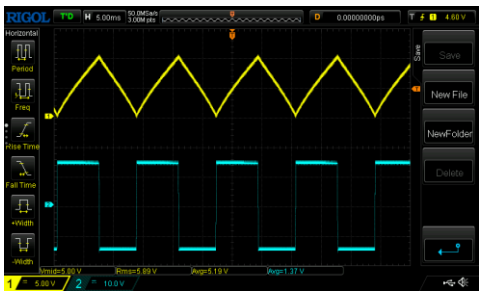


Slika 12. Praktična realizacija signal generatora pomoću analognih komponenata.

Testiranjem realizovanih šema dobijeni su sledeći rezultati prikazani na ekranu osciloskopa (slika 13 i 14).



Slika 13. Snimak ekrana osciloskopa na kome se vidi izlazni signal oscilatora sa vinovim mostom.



Slika 14. Snimak ekrana osciloskopa na kome se vide trougaoni i četvrtasti signal sa izlaza naponski kontrolisanog oscilatora.

VII. Zaključak

Opisani projekat je praktično konstruisan i testiran korišćenjem odgovarajućih komponenata na protoploči i propratnih instrumenata (osciloskop RIGOL DS1102 [9]). Na slici 14 je prikazana ova realizacija. Ovako konstruisan signal generator ima široku primenu pri dizajniranju, testiranju i popravci elektronskih uređaja.

Zahvalnica

Autori ovog rada se srdačno zahvaljuju profesorima i asistentima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu. Posebnu zahvalnost dugujemo asistentu Nikoli Mitroviću koji nam je obezbedio potreban materijal za izradu ovog projekta, kao i sugestije koje nam je uputio tokom izrade istog.

Literatura

[1] Z. Prijić, Analogna Mikroelektronika, (Slajdovi: 372 – 440) predavanja. Dostupno na:https://mikro.elfak.ni.ac.rs/wp- cont ent/uploads/2017/10/Literatura/Analogna/Analogna_Opamps. pdf.

[2] Nikola Mitrović, Danijel Danković, Zoran Prijić, *Laboratorijski praktikum iz analogne mikroelektronike* (Stranice: 1 – 14), Elektronski fakultet, Niš, 2022.

[3] Z. Prijić, Analogna Mikroelektronika, (Slajdovi: 15 – 38) predavanja.Dostupno na:https://mikro.elfak.ni.ac.rs/wp- conte nt/uploads/2017/10/Literatura/Analogna/Analogna_lin_pred.p df.

[4] Power Transistor IRF 510 datasheet. Dostupno na linku: https://www.vishay.com/docs/91015/irf510.pdf.

[5] LM317 3-Terminal Adjustable Regulator datasheet. Dostupno na linku: https://www.ti.com/lit/ds/slvs044y/slvs044y.pdf?HQ S=slvs044-aaj&ts=1698993507405.

[6] LM337 3-Terminal Adjustable Negative Regulator datasheet. Dostupan na linku: https://www.ti.com/lit/ds/symlink/lm137.p df.

[7] Quadruple Operational Amplifier LM2902 datasheet, dostupan na linku: https://www.ti.com/lit/ds/symlink/lm2902-n.pdf

[8] Programmable Unijunction Transistor dataheet. Dostupno na linku: https://www.onsemi.com/download/data- sheet/pdf/2n6 027-d.pdf.

[9] Rigol DS1102 Oscilloscope Datasheet. Dostupno na linku: https://cdn- shop.adafruit.com/datasheets/Data+Sheet+DS100 0E.pdf.

Praktična realizacija signal generatora korišćenjem ICL8038 kola

Kristina Mišić, Vuk Zlatković

Sadržaj – U ovom radu biće prikazana primena kola ICL8038 kao generator signala i njegove performanse će biti upoređene sa alternativnim načinima realizacije istog.

I. UVOD

ICL8038 [1] je monolitno integrisano kolo sposobno da proizvede različite talasne oblike, visoke preciznosti. Prilikom korišćenja ne može se koristiti samostalno već sa minimalno spoljnih komponenti, koje pored omogućavanja rada kola služe i za podešavanje karakteristika generisanih signala.

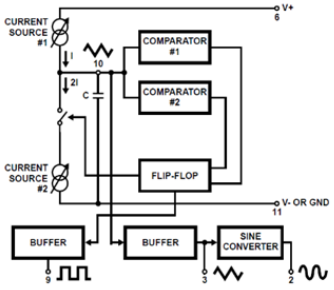
Frekvencija se može podešavati od 0.001 Hz do 300 kHz korišćenjem otpornika i kondenzatora, a frekvencijska modulacija se može postići preko napona dovedenog na komponentu.

Integrisano kolo ICL8038 može da funkcioniše u širokim opsezima napona i temperatura bez izobličenja signala. Opšte karakteristike:

- Opseg napajanja (od V- do V+) je 36 V.
- Nominalna struja kroz kolo 25 mA.
- Stabilan rad na opsegu temperatura od 0°C do 70°C.

II. GENERISANJE RAZLIČITIH TIPOVA SIGNALA

Pomoću ovog IC moguće je generisati različite tipove signala: pravougaoni signal (povorka impulsa), trougaoni signal i sinusni signal.



Slika 1. Funkcionalni dijagram integrisanog kola ICL8038.

K. Mišić i V. Zlatković su studenti 4. godine na modulu Elektronske komponente i mikrosistemi na Elektronskom fakultetu; Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: kristina.misic@elfak.rs vuk.zlatkovic@elfak.rs

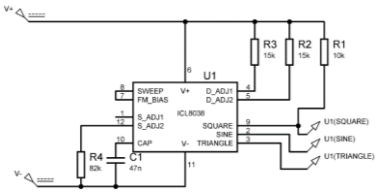
III. PRINCIP RADA

Unutrašnji kondenzator *C* se puni i prazni pomoću dva izvora struje. Drugi strujni izvor se uključuje i isključuje pomoću flip-flopa, dok je prvi strujni izvor neprekidno uključen (slika 1). Pod pretpostavkom da je drugi strujni izvor isključen, kondenzator *C* je napunjen strujom, napon na kondenzatoru raste linearno sa vremenom. Kada ovaj napon dostigne naponski nivo prvog komparatora (podešen na 2/3 napona napajanja), flip-flop se aktivira, *C* počinje da se prazni i napon na njemu opada linearno sa vremenom. Kada dostigne naponski nivo drugog komparatora (podešen na 1/3 napona napajanja), flip-flop se vraća u prvobitno stanje i ciklus se ponavlja.

Tako se preko kondenzatora C generiše trougaoni signal (izlaz na pinu broj 3), preko flip-flopa pravougaoni (izlaz na pinu broj 9) i preko sinusnog konvertera sinusni signal (izlaz na pinu broj 2) (slika 3 i 4).

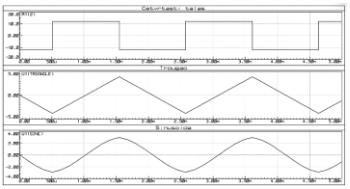
IV. SIMULACIJA U *PROTEUS* - U

Rad integrisanog kola simuliran je u *Proteus*-u.



Slika 2. Šema kola na kome je testiran ICL8038 u *Proteus*-u.

Dobijeni signali su prikazani na slici koja sledi.

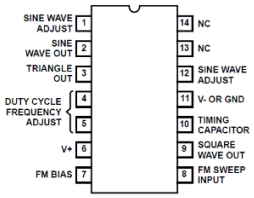


Slika 3. Rezultat simulacije izlaznih signala integrisanog gola ICL8038.

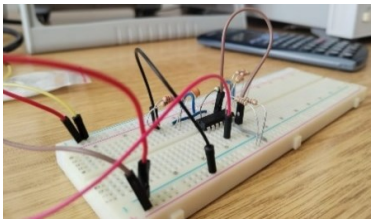
V. PRAKTIČNA REALIZACIJA

Prilikom praktične izrade ovog projekta korišćeno je

ICL8038 kolo kao i propratne spoljne komponente (otpornici (1x10 kΩ, 2x15 kΩ i 1x82 kΩ) i kondenzator (1x47 nF)) za podešavanje učestanosti i faktora ispune(Slika 2).

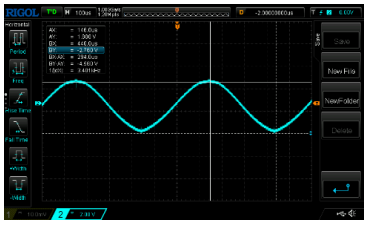


Slika 4. Pinout diagram ICL8038.

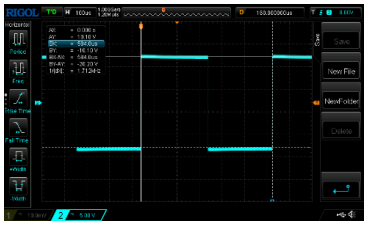


Slika 5. Praktična realizacija signal generatora sa integrisanim kolom ICL8038.

Pomoću osciloskopa *Rigol DS 1102* [3], usnimljeni su generisani signali (slike 6 i 7).



Slika 6. Snimak ekrana osciloskopa na kome se vidi generisani sinusni signal.



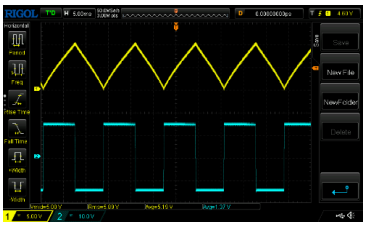
Slika 7. Snimak ekrana osciloskopa na kome se vidi generisani pravougaoni signal.

VI. Poređenje Sa Drugim Rešenjima

Talasne oblike moguće je generisati i drugim metodama: pomoću analognog kola koje se sastoji od više blokova: naponski kontrolisan oscilataor - VCO (za dobijanje trougaonog i pravougaonog signala), oscilator sa Vinovim mostom (za dobijanje sinusnog signala), relaksacioni oscilator (za dobijanje testerastog signala) ili pomoću *Arduino* razvojnog okruženja.

Korišćenje sinusnog generatora ICL8038 je jednostavnije u odnosu na gore navedene metode jer je monolitno integrisano kolo (sve komponente su izrađene na jednom čipu), zahteva jedno napajanje, jeftinije u odnosu na razvojno okruženje *Arduino*, takođe ovo kolo je namenjeno isključivo generisanju signala, te je nije potrebno programirati već se može odmah iskoristiti za svoju namenjenu svrhu.

Talasni oblici dobijeni korišćenjem ICL8038 su jasno definisani i praktično nemaju šumove u poređenju sa realizacijom koja koristi diskretne komponente. Glavni razlog za ovaj slučaj je to što je ICL8038 kolo kompletno integrisani te su maksimalno smanjeni uticaji elektromagnetnih smetnji. Ovo je najlakše uočiti kada se porede snimci sa osciloskopa jedne i druge realizacije generatora signala.



Slika 8. Snimak ekrana osciloskopa na kome se vide trougaoni i četvrtasti signal sa izlaza naponski kontrolisanog oscilatora.

VII. Zaključak

Opisani projekat je praktično konstruisan i testiran korišćenjem odgovarajućih komponenata na protoploči i propratnih instrumenata (osciloskop RIGOL DS1102) [3]. Na slici 5 je prikazana ova realizacija. Ovako konstruisan signal generator ima široku primenu pri dizajniranju, testiranju i popravci elektronskih uređaja.

Zahvalnica

Autori ovog rada se srdačno zahvaljuju profesorima i asistentima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu. Posebna zahvalnost dugujemo asistentu Nikoli Mitroviću koji nam je obezbedio potreban materijal za izradu ovog projekta kao i sugestije koje nam je uputio tokom izrade istog.

Literatura

[1] ICL8038 Precision Waveform Generator/Voltage Controlled Oscillator510 datasheet. Dostupno na linku: https://www.renesas.com/us/en/document/dst/icl8038-datasheet.

[2] Z. Prijić, Analogna Mikroelektronika, (Slajdovi: 372 – 440) predavanja. Dostupno na:https://mikro.elfak.ni.ac.rs/wp- content/uploads/2017/10/Literatura/Analogna/Analogna_Opamps.pdf.

[3] Rigol DS1102 Oscilloscope Datasheet. Dostupno na linku: https://cdn- shop.adafruit.com/datasheets/Data+Sheet+DS1000E.pdf.

Ionizing Radiation Dosimeter Calibrated as Geiger Counter Based on Arduino Microcontroller and BG51 Sensor

Sandra Miljković

Abstract - This paper is focused on developing a dosimetry sensor based on semiconductor devices and Arduino microcontroller, which are widespread and accessible to a wide mass of people. Reading results from sensor, controlling the audio signal when detecting the ionizing radiation, calculating the equivalent dose rate, sending data to the LCD screen, and controlling the display is all done with microcontroller. BG51 sensor is integrated on a click-board, which makes it easily adaptable.

I. Introduction

Electromagnetic radiation, in classical physics, is the flow of energy at the universal speed of light through free space or through a material medium in the form of the electric and magnetic fields that make up electromagnetic waves such as radio waves, visible light, and gamma rays. In such a wave, time-varying electric and magnetic fields are mutually linked with each other at right angles and perpendicular to the direction of motion. An electromagnetic wave is characterized by its intensity and the frequency ν of the time variation of the electric and magnetic fields [1]. According to modern quantum theory, electromagnetic radiation is the flow of photons through space. Electromagnetic radiation exhibits a multitude of phenomena as it interacts with charged particles in atoms, molecules, and larger objects of matter [1]. According to the effect on the material, radiation can be divided into two groups: ionizing radiation and non-ionizing radiation. Ionizing radiation is a fast electron beam which has enough energy to ionize medium through whom it is spread [2]. Furthermore, ionizing radiation can also be divided into two groups - it can be corpuscular or electromagnetic in nature. Corpuscular radiation is a microsystem characterized by mass - like alpha or beta particles, protons, nucleons, and systems with more than one sort of particle. Electromagnetic radiation consists of photons which can be interpreted as waves or particles. Full electromagnetic spectrum is shown on figure [3].

Ionizing radiation can be the result of natural sources, usually of terrestrial or cosmic origin; or it can be produced by artificial sources. The environment near to the radiation source should be observed and there is a need for measuring the radiation levels. It is all done using different kinds of detectors. Detectors are divided into active and passive depending on whether they use power or not; another criteria is whether they collect information on the total amount of radiation over time or provide it immediately; while according to their aggregate state, working substances are divided into gas, liquid and solid. We measure two general phenomena when we discuss radiation. We measure "activity" and "exposure". Activity is basically just how much radiation is coming out of something, whether it's particles or waves. Exposure measures the effect of radiation on substances that absorb it [4]. Most often measured unit is radiation exposure - Radiation exposure is expressed in several ways to account for the different levels of harm caused by different forms of radiation and the different sensitivity of body tissues; radiation activity - measured in an international (SI) unit called a becquerel (Bq). The becquerel counts how many particles or photons (in the case of wave radiation) is being emitted per second by a source. The device used for measurement is often the familiar Geiger counter. Absorbed dose is equivalent to the energy "deposited" in a kilogram of a substance by the radiation, equivalent dose - which relates the absorbed dose in human tissue to the effective biological damage of the radiation, and effective dose - the probability of a harmful effect from radiation exposure depends on what part or parts of the body are exposed, as shown on figure [5].

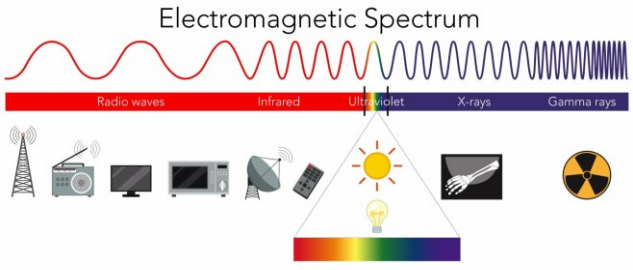


Fig. 1 Electromagnetic spectrum [3].

An ionizing detector controlled with Arduino Uno microcontroller opensource platform is described in [4]. Authors used an Arduino microcontroller platform and BG51 sensor integrated on a click board to achieve a dosimetry sensor. Arduino controls the data acquired from

S. Miljković is with the Department of Microelectronics, Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: sandra.miljkovic@elfak.rs

sensor, sound signalization when radiation is detected, and calculates the equivalent dose rate. System communicates with the PC using serial communication [4].

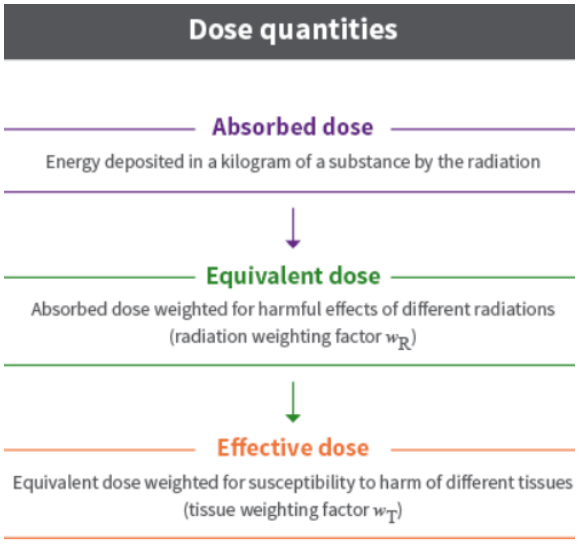


Fig. 2. Dose quantities [5].

This paper is based on work previously published in [4]. System is upgraded, and calibrated in laboratory, to perform like a Gaiger counter. Sound signalization, data acquisition and calculation of the equivalent dose rate are same as in [4]. The results are displayed on an LCD display, along with sound signalization and warnings when the dose is higher than the predetermined one.

II. Radiation Sensor BG51

The function of the BG51 radiation sensor is based on an array of customized PIN diodes. The integrated pulse discriminator with a temperature compensated threshold level provides true TTL signal output. The BG51 is capable of detecting beta radiation (electrons), gamma radiation (photons) and X-rays. The performance of the BG51 solid state sensor, in combination with high immunity to electrostatic fields make it a good choice for new state-of-the-art designs as well as for upgrading existing designs [6].

On figure 3. radiation dose equivalent rate for Cs-137 and Co-60 (mSv/h) depending on counts per minute is shown. The response is linear in log-log scale, and it is claimed that the response is linear for the wide spectrum of temperature scale (from -30°C to 60°C). The sensor is integrated on a click board, manufactured by the company Mikroelektronika from Belgrade, and it has external pins for various applications.

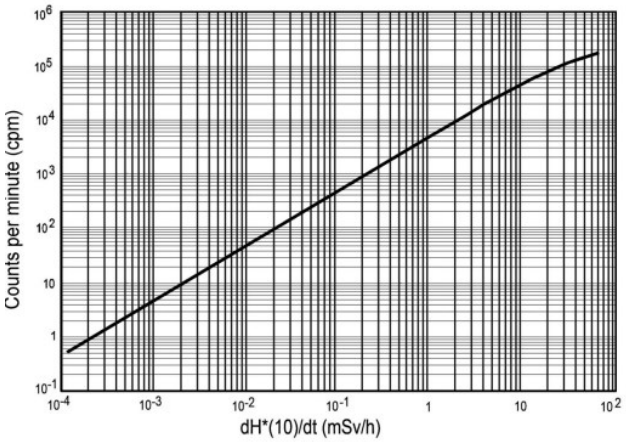


Fig. 3. Radiation dose equivalent rate for Cs-137 and Co-60 (mSv/h).

III. Practical Realization

Block scheme of connecting the peripheral units with Arduino Uno is given in figure 4. Output pin of BG51 sensor module (INT) is connected to pin no.3 of the Arduino Uno. Piezo buzzer indebted for audio signalization is connected to pin no.7. LCD display is connected to pins 12, 11, 6, 5, 4, 2.

Program written using Arduino Software (IDE) is given on figure 4. The program for Arduino microcontroller is just an upgraded version of the one described in [4]. Firstly, the pins are deffined; pin for piezo buzzer, pin for sensor and library for LCD display is included. After defining variables, pins are assigned to the LCD display. In *void setup ()* function display is initialized, along with piezo buzzer.

In *void loop ()* function the counter is setted, and the equation for calculating the equivalent dose ratio. Significant modifications are made in this function by adding disclamer for values higher than predicted ones. It is implemented through if loop. Results are, in contrary to ones in [4], displayed on LCD screen, which is also defined in void loop () function. A warning for adjusting the calibration factor is also given since this is a sensor that can change its original settings over time. The practical connection together with the results of printing on the display are given below. The practical connection is achieved by the results printed on the display, as shown on figure 5. The display is connected to the arduino following the instructions from the datasheet of the display.

A. Experimental setting

The aim of this work is to calibrate the sensor to behave like a commercial Geiger counter. In this experiment, the Inspector EXP+ by Keison products was used. For this purpose, an experiment was conducted with a radium source, as shown on figure 6.

```
#include <LiquidCrystal.h>

#define buzzerPin 7
#define sensorPin 2

volatile int counter = 0;
unsigned long time;
unsigned long dosage;
unsigned long previousTime = 0;
float radiation = 0.0;
const float calibrationFactor = 22.67;
int maxCounter = 10;
float threshold = 25.0;
bool alertActive = false;
unsigned long calibrationInterval = 24 * 60 * 60 * 1000;
unsigned long lastCalibrationTime = 0;

LiquidCrystal lcd(12, 11, 6, 5, 4, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Measurement... ");
  pinMode(buzzerPin, OUTPUT);
  digitalWrite(buzzerPin,LOW);

  attachInterrupt(digitalPinToInterrupt(3), detectionISR, RISING);
  delay(5000);

  lastCalibrationTime = millis();
}

void loop() {
  time = millis();

  if (counter >= maxCounter) {
     dosage = time - previousTime;
     previousTime = time;
     counter = 0;
     radiation = float(maxCounter) * 60.0 * 1000.0 / float(dosage) / calibrationFactor;
      if (radiation >= threshold) {
       alertActive = true;
       digitalWrite(buzzerPin, HIGH);

      } else {
       alertActive = false;
       digitalWrite(buzzerPin, LOW);
      }
  }

  lcd.setCursor(0, 0);
  lcd.print("Radiation: ");
  lcd.print(radiation);
  lcd.println(" uSv/h");
  lcd.setCursor(0, 1);
  lcd.print("Warning: ");
  lcd.print(alertActive ? "Active" : "Not active");
  lcd.print("   ");

  if (time - lastCalibrationTime >= calibrationInterval) {
     lcd.print("Calibrate");
     lastCalibrationTime = time;
  }
}

void detectionISR() {
  counter++;
}
```

Fig. 4. Arduino code for radiation sensor BG51 and LCD display.

Measurements were made in a period of 30 seconds, because for doses less than 1.75mR/h, the commercial dosimeter gives the mean value calculated for that period [7]. The radiation dose emitted by the source was first measured with a commercial dosimeter and then with a BG51 sensor. A commercial dosimeter measures in mR/h, but using the user manual [7], these values are converted into mSv/h. The obtained values are given in Table 1.

TABLE I
RESULTS OF MEASURING THE RADIATION DOSE WITH GEIGER COUNTER AND BG51 SENSOR

	Geiger counter	BG51
1	40.89 µSv/h	240.75 µSv/h
2	41.52 µSv/h	204.42 µSv/h
3	40.06 µSv/h	207.37 µSv/h
4	39.74 µSv/h	235.29 µSv/h
5	38.54 µSv/h	379.51 µSv/h

The values from Table 1. were used to obtain the calibration factor. Based on five measurements, the average value for both the dosimeter and the sensor was calculated, which was used to obtain the calibration factor- calculated to be 22.67.

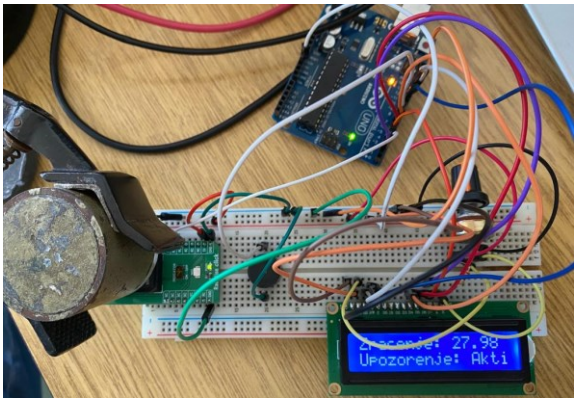


Fig. 5. Practical implementation of dosimeter based on BG51 sensor and Arduino Microcontroller.



Fig. 6. Experimental setup for measuring the radiation dose.

IV. Results and Discussion

The described radiation dosimeter is connected to the oscilloscope after being connected to the breadboard to record the detected pulses of ionizing radiation. The figures present the oscilloscope screen showing the detected pulses for a period of 30 seconds (figure 7.), as well as a separate image showing the appearance of one detected pulse (figure 8.). In Table 2. are the results of measurements after the calibration factor is set to the obtained value.

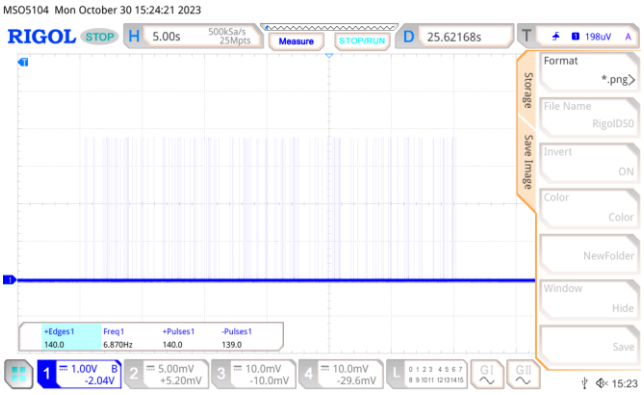


Fig. 7. Pulses detected in period of 30 seconds.

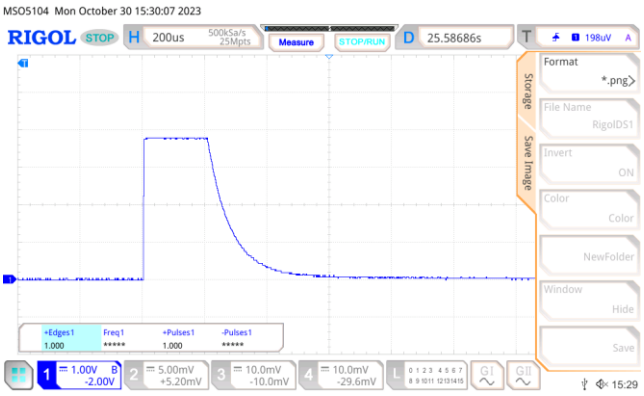


Fig.8. One isolated pulse.

TABLE II
Results of measuring the radiation dose with Geiger counter and BG51 sensor after the calibration

	Geiger counter	BG51
1	30.2 µSv/h	23.67 µSv/h
2	30.2 µSv/h	27.77 µSv/h
3	27.7 µSv/h	24.92 µSv/h
4	28.49 µSv/h	28.37 µSv/h
5	27.7 µSv/h	29.28 µSv/h

V. Conclusion

This paper describes a posibility of using radiation sensor BG51 as Gaiger counter. Prototipe consists of Arduino Uno board, BG51 radiation sensor, audio signalization made out of piezo buzzer and LCD display used to show the results. In addition to results, on display are also shown warnings for doses higher than recomended ones, and also reminders for additional calibration of the sensor. Further improvements can be made to the code, so that it calculates the mean value for a certain time and displays it.

Acknowledgement

I would like to express my sincere gratitude to Assistant Miloš Marjanović for helping with the experiment and giving useful advice during this research.

References

[1] Electromagnetic radiation. Available at: https://www.britannica.com/science/electromagnetic-radiation
[2] Goran Ristić, “Dosimetry and dosimeters lectures”, Faculty of Electronic Engineering Nis, pp. 1-7, 2016.
[3] Electromagnetic spectrum. Available at: https://online-learning-college.com/knowledge-hub/gcses/gcse-physics-help/electromagnetic-spectrum/
[4] Milos Marjanović, Emilija Živanović, Danijel Danković, Goran Ristić “Ionizing radiation dosimeter based on Arduino microcontroller and BG51 sensor” 15th IEEESTEC Student Projects Conference, Nis, Serbia 2022.
[5] Units of ionizing radiation measurements. Available at https://www.arpansa.gov.au/understanding-radiation/what-is-radiation/radiation/measurement
[6] Radiation Sensor BG51, datasheet. Available at: https://www.teviso.com/file/pdf/bg51-data-specification.pdf
[7] Inspector EXP+ User Manual. Available at: https://site.jjstech.com/pdf/SE-International/InspectorPlus&Inspector-EXPPlus_Operation-Manual.pdf

Praktična realizacija sistema za merenje visokih temperatura korišćenjem MAX6675 kola

Milutin Dinić

Sadržaj – U ovom radu je opisano projektovanje sistema za merenje visokih temperatura korišćenjem termopara i kola MAX6675 koje služi za obradu signala sa termopara. Dat je detaljan opis fizičkih procesa, korišćenih komponenata i realizovanog kola. Izvršeno je testiranje projektovanog sistema, kako u laboratorijskim, tako i u realnim radnim uslovima. Dat je zaključak oko prednosti i mana realizovanog sistema.

I. Uvod

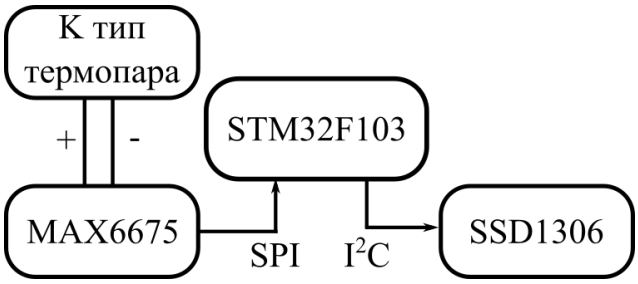
Merenje temperature jedan je od zadataka sa kojim su se istraživači susretali od definisanja tog termina. Temperatura je fizička veličina koja određuje stepen zagrejanosti nekog tela [1]. Vremenom, realizovani su različiti načini za merenje temperature. Razvojem mikroelektronskih komponenata, merenje temperature postalo je sastavni deo većine komercijalnih sistema.

Merenje viših temperature, odnosno temperatura preko 300 °C najčešće ne može biti adekvatno odrađeno standardnim elektronskim komponentama. Za takve primene potrebno je promeniti koncept. Zbog toga, u ovakvim primenama merenje visokih temperatura postiže se korišćenjem termopara.

Termopar predstavlja zatvoreno kolo sastavljeno od dva različita metala, najčešće realizovanih u obliku žica, čiji su krajevi spojeni. U literaturi se može naći i pod imenom termoelement ili termospreg. Jedan spoj metalnih žica izlaže se visokoj temperaturi (onoj koja se meri), dok se drugi spoj žica drži na referentnoj temperaturi (sobna temperatura, a u nekim primenama i temperatura od 0 ºC). Princip rada termopara bazira se na Zibekovom efektu [2]. Postoje različiti tipovi para (J, K, itd.) koji se razlikuju prema tome koja dva metala sačinjavaju termopar. Različiti materijali omogućavaju različite merne opsege termopara.

II. Projektovanje kola za merenje temperature

Projektovano kolo za merenje visokih temperatura sastoji se od termopara i kola MAX6675 za obradu signala sa termopara kao centralnih elemenata. Prateće kolo pojačava signal sa termopara i omogućava komunikaciju sa mikrokontrolerom ili drugim mernim uređajem. Za obradu podataka sa kola MAX6675 iskorišćen je mikrokontroler STM32F103 koji je montiran na odgovarajuće razvojno okruženje pod imenom Blue Pill. Rezultati merenja temperature mogu se očitati sa OLED displeja sa SSD1306 kontrolerom koji je povezan sa mikrokontrolerom. Blok šema mernog uređaja prikazana je na slici 1.



Slika. 1. Blok šema principa rada projektovanog sistema.

A. Termopar i kolo MAX6675

Za realizaciju sistema iskorišćen je K tip termopara, izgrađen od nikl-hroma kao prvog metala i nikl-aluminijuma kao drugog metala, komercijalno dostupan u obliku žice na čijem vrhu se nalazi telo oblika šrafa [3]. Dužina žica je 0,5 m. Korišćeni termopar prikazan je na slici 2.

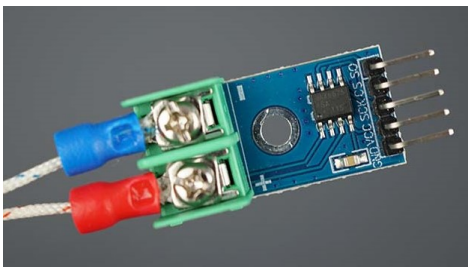


Slika. 2. Izgled korišćenog termopara.

Kolo MAX6675, prikazano je na slici 3, ima 8 pinova i na pločici se nalazi u SOIC-8 SMD kućištu [4]. Sama šema korišćenog kola prikazana je na slici 4.

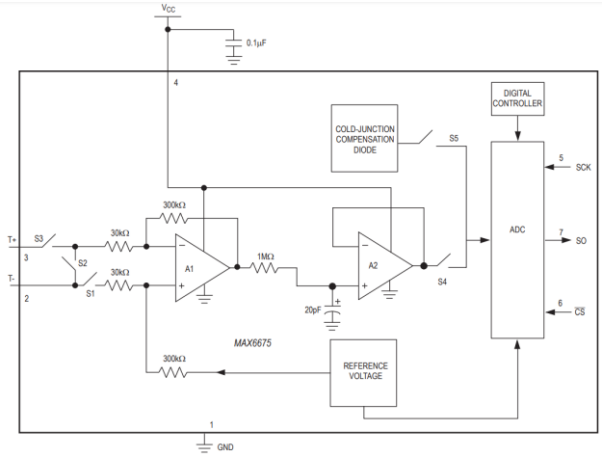
Kao što se može videti na slici 4, kolo MAX6675 ima 8 pinova, dva služe za napajanje i masu (pinovi 1 i 4), dva služe za ulaze termopara (pinovi 2 i 3, označeni sa T+ i T-), tri pina služe za komunikaciju sa mikrokontrolerom ili drugom mernom jedinicom (pinovi 5 (SCK – *Serial Clock*), 6 (CS - *Chip Select*) i 7 (SO – *Signal Output*), korišćenjem SPI protokola za serijsku komunikaciju), dok se pin 8 ne povezuje (NC – *not connected*).

M. Dinić je student pri Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milutin.dinic@elfak.ni.ac.rs



Slika. 3. Izgled pločice za povezivanje termopara sa kolom MAX6675.

Kao što je prikazano na slici 4, između pinova 4 i 1, odnosno između pinova za napajanje i masu, preporučuje se povezivanje *decoupling* kondenzatora, koji se može videti i na pločici prikazanoj na slici 3.

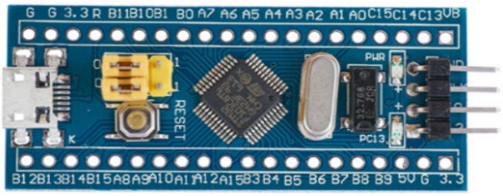


Slika. 4. Blok šema rada kola MAX6675 (preuzeto iz tehničke specifikacije za kolo MAX6675 [5]).

Kolo se sastoji od dvostepenog pojačavača realizovanog korišćenjem operacionog pojačavača, gde se izlaz prvog operacionog pojačavača, označenog na slici 4 sa *A1* (*amplifier one*) dovodi na neinvertujući ulaz drugog operacionog pojačavača, označenog sa *A2* (*amplifier two*). Izlaz sa drugog operacionog pojačavača se prosleđuje na analogno digitalni konvertor, pa se dalja obrada i očitavanje podatka vrši nad digitalnim signalom. Jako bitan deo kola je dioda za temperaturnu kompenzaciju referentnog spoja (*cold-junction compensation diode*) koja se koristi za merenje temperature mernog spoja, koji se u literaturi često naziva i hladan spoj. Ova dioda omogućava da se na izmerenu temperaturu mernog spoja u uticaj uzme i temperatura na kojoj se nalazi referentni spoj (pošto je izmerena temperatura ipak data kao temperaturna razlika dva kraja) i da se na taj način izvrši pouzdanije merenje.

B. Blue Pill okruženje sa mikrokontrolerom STM32F103

Blue Pill je razvojno okruženje sa STM32F103 mikrokontrolerom kompanije STMicroelectronics, koje je bazirano na ARM Cortex-M3 jezgru [5]. *Blue Pill* ima 37 pinova koji predstavljaju pinove 4 porta – portovi A i B (16 pinova), port C (3 pina) i port D (2 pina). Svaki pin ima sposobnost *sink/source* struje od 6 mA. *Pull-up* i *pull-down* konfiguracije mogu biti omogućene na svakom od pinova. Temperaturni radni opseg ovog mikrokontrolera je -40 °C – 105 °C. Razvojno okruženje prikazano je na slici 5.



Slika. 5. Izgled prednje strane STM32 *Blue Pill* razvojnog okruženja.

Pored opisanih osobina, ovaj mikrokontroler je odabran zbog široke dostupnosti i besplatnih, pristupačnih alata za rad i programiranje, kao što su *STM32CubeMX* i *Keil µVision*.

STM32CubeMX razvijen od strane kompanije *STMicrolectronics*. Ovaj softver predstavlja grafički alat koji omogućava veoma laku konfiguraciju STM32 mikrokontrolera i mikroprocesora, kao i generisanje odgovarajućeg inicijalizacionog C koda za jezgro *Arm Cortex-M* ili delimično *Linux* stablo uređaja za *Arm Cortex-A* jezgro, kroz proces korak po korak [5, 6].

Drugi korišćeni softver je *Keil µVision5 IDE* (*Integrated Development Environment*). Ovaj softver kombinuje upravljanje projektima, okruženje za izvršavanje, objekte za pravljenje, uređivanje izvornog koda i otklanjanje grešaka u programu u jednom moćnom okruženju. *Keil µVision* je jednostavan za korišćenje i ubrzava razvoj ugrađenog softvera [5, 6].

C. OLED displej sa SSD1306 kontrolerom

Za prikaz podataka o izmerenoj temperaturi iskorišćen je OLED displej sa SSD1306 kontrolerom. OLED SSD1306 modul predstavlja moderniji tip displeja za primenu u *embedded* sistemima, pre svega zbog svojih malih gabarita [7]. Povezuje se preko samo 4 pina, korišćenjem I2C komunikacije, što rasterećuje mikrokontroler. Koristi SSD1306 drajver koji omogućava veliki broj podešavanja. Korišćeni displej prikazan je na slici 6.

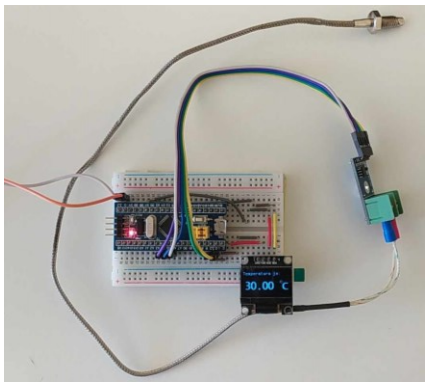


Slika. 6. OLED SSD1306 modul.

Na displeju se mogu bojiti pikseli iz mreže piksela. Na taj način dobija se utisak poruke ili slike. Pikseli su uglavnom jednobojni, mada postoje varijacije displeja koje omogućavaju da pikseli svetle u dve boje.

III. PRAKTIČNA REALIZACIJA

Na osnovu teorijskih razmatranja, opisanog kola i opisanih korišćenih komponenata, sistem za merenje temperature je praktično realizovan na protoploči. Slika praktično realizovanog sistema prikazana je na slici 7.

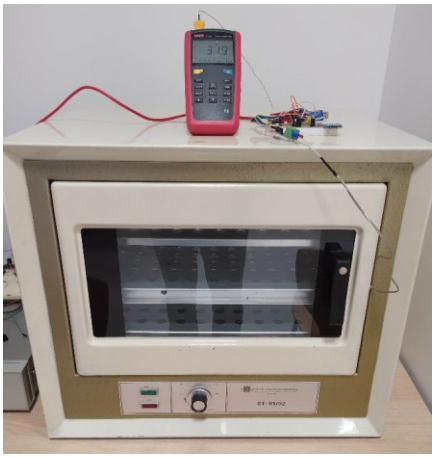


Slika 7. Praktična realizacija sistema na protoploči.

Nakon praktične realizacije, sistem je potrebno testirati i u laboratorijskim i u realnim radnim uslovima.

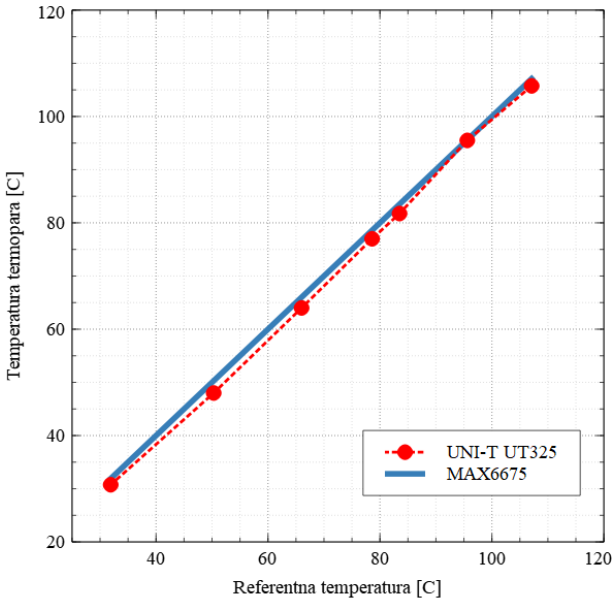
A. Testiranje u laboratorijskim uslovima

Cilj testiranja u laboratorijskim uslovima je osnovna provera rada sistema kao i utvrđivanje potencijalnih grešaka. Za testiranje rada sistema iskorišćena je pećnica. Realizovani sistem i profesionalni termopar sa profesionalnom opremom za merenje stavljeni su u komoru pećnice. Temperatura unutrašnjosti pećnice je podizana i vršena su merenja paralelno preko oba uređaja. Merna postavka prikazana je na slici 8.



Slika 8. Testiranje sistema u laboratorijskim uslovima.

Kao referentni uređaj iskorišćen je drugi K tip termopara povezan na termometar UNI-T UT325 [8]. Rezultati merenja prikazani su i grafički. Na slici 9, rezultati mereni uređajem UNI-T UT325 su prikazani kao referentni (plava linija), a rezultati mereni projektovanim uređajem sa MAX6675 kolom prikazani su crvenom bojom, gde su označene merne tačke.



Slika 9. Grafički prikaz izmerenih rezultata.

Testiranjem u laboratoriji utvrđene su određene merne nepreciznosti koje su korigovane na odgovarajući način. Nakon testiranja u laboratoriji, u kontrolisanim uslovima, potrebno je izvršiti testiranje i u eksperimentalnim uslovima.

B. Testiranje u industrijskoj primeni

Testiranje u industrijskoj primeni je pokazano na primeru merenja temperature unutar ložišta komercijalne peći. Merenje je realizovano u prostorijama kompanije EM-DIP [9]. Termopar je smešten u ložište komercijalne peći T-500. Merenje razvijene temperature unutar ložišta predstavlja deo standardnih testova vezanih za proveru kvaliteta peći odnosno ložišta. Postavljeno okruženje za testiranje prikazano je na slici 10.

Nakon što je utvrđeno da je vatra na potrebnom nivou, vrata ložišta se zatvaraju i kreće merenje temperature. Projektovani sistem sada bez poteškoća prikazuje trocifrenu temperaturu.

Prilikom postavke eksperimenta je jako bilo bitno da se vrh termopara ne nađe u otvorenom plamenu. Temperatura otvorenog plamena je značajnije veća od temperature koja se emituje na zidove ložišta, dok se toplota u slučaju praktične primene emituje sa površine ložišta. Pozicija termopara u okviru ložišta prikazana je na slici 11.

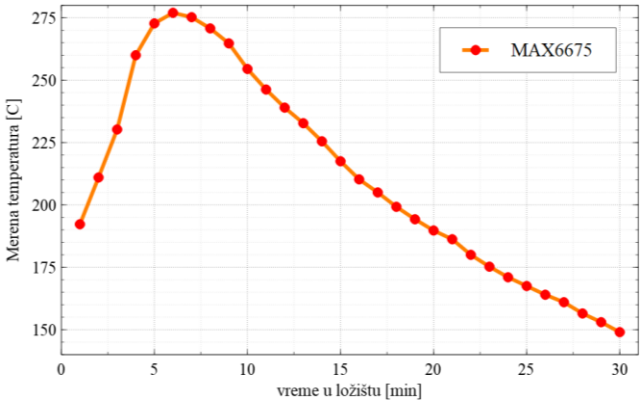


Slika 10. Postavka u prostorijama kompanije (realizovan uređaj i termopar spušten u ložište peći).



Slika. 11. Pozicioniranje termopara unutar ložišta peći.

Prilikom merenja temperature, termopar pokazuje, u skladu sa očekivanjima, temperature koje prelaze vrednost od 250 °C. Merene vrednosti temperature prema vremenu grafički su prikazane na slici 12.



Slika. 12. Izmerena temperatura projektovanim sistemom unutar ložišta.

Testiranjem u realnim uslovima utvrđeno je da sistem pokazuje rezultate koji su u skladu sa očekivanjima, ali da je za visoku preciznost potrebna dodatna kalibracija. Takođe, ovako realizovan sistem nudi mogućnosti integracije sa drugim velikim brojem mernih jedinica koje dale dodatnu informaciju o uslovima pri kojima se vrši odgovarajuće merenje.

IV. Zaključak

U prikazanom radu je pokazan proces projektovanja jednog sistema sa termoparom i pratećim elektronskim kolom, koji ima zadatak da vrši merenje visokih temperatura. Korišćene komponente i princip rada kola MAX6675 je dodatno objašnjen, kao i metod povezivanja komponenata u kolu.

Sistem je praktično realizovan, a zatim i testiran, prvo u laboratorijskim uslovima, a onda i jednoj realnoj industrijskoj primeni. Dobijeni su rezulati koji su odgovarajući, iako je utvrđeno da je za dodatnu preciznost potrebno uvesti dodatne komponente u kolo.

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] M. Pejović, Opšta fizika, Elektronski fakultet Niš, 2006.
[2] V. Paunović i M. Marjanović, predavanja iz predmeta Materijali za elektroniku, Elektronski fakultet Niš, 2022.
[3] Tehnička specifikacija za K tip termopara. Dostupno na: https://www.edata.omron.com.au/eData/TCs/E52_DS.pdf
[4] Tehnička specifikacija za kolo MAX6675. Dostupno na: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX6675.pdf
[5] Tehnička specifikacija za STM32F103. Dostupno na: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf
[6] N. Mitrović, M. Marjanović, D. Danković, Primena ARM mikrokontrolera, Akademska misao, Beograd, 2023.
[7] Tehnička specifikacija za OLED displej. Dostupno na: oled128o064dbpp3n00000.pdf (vishay.com).
[8] UNI-T UT325 specifikacija uređaja. Dostupno na: https://meters.uni-trend.com/product/ut325/
[9] Kompanija EM DIP https://emdip.com/

Implementacija Scrum metodologije u okviru razvoja softvera

Milan Đorđević

Sadržaj - Rad se bavi istraživanjem različitih rešenja kada je u pitanju organizacija članova tima za vreme razvoja softvera ili nekog drugog sličnog proizvoda. Dat je odgovarajući uvod kako bi se uvideo razlog postojanja različitih metodologija za razvoj softvera. Akcenat je stavljen na upoznavanje sa *Scrum* metodologijom, kao jednom od metodologija, ali su predstavljene i druge kako bi bilo moguće uvideti prednosti i mane svake, i napraviti poređenje između njih. Na kraju, rad daje uvid u tok i efikasnost implementacije *Scrum* metodologije u praksi, u okviru kompanije *Better Collective*.

I. Uvod

Na početku bilo kog projekta vezanog za razvoj softvera, tim koji radi na tom projektu mora napraviti odluku od vitalnog značaja, a to je koju metodologiju za razvoj softvera bi trebalo koristiti.

O *Scrum* metodologiji može se razmišljati kao o načinu na koji tim može završiti neki posao tako što se može razložili i raditi na malim delovima, a da pritom u međuvremenu bude prostora za eksperimentisanje i povratne informacije – (*feedback*) koji bi omogućili da tim u toku projekta uči i napreduje [1]. *Scrum* timovima pomaže da kroz kolaboraciju inkrementalno pridodaju na vrednosti nekog produkta. Zasnovan na *agile* - agilnom modelu, Scrum nudi taman dovoljno strukture u okviru tima kako bi se integrisao u njihov način rada, a da usput ponudi i mogućnost optimizacije za specifične potrebe tima. Zbog toga, ova metodologija postala je izuzetno zastupljena u industriji [1, 2].

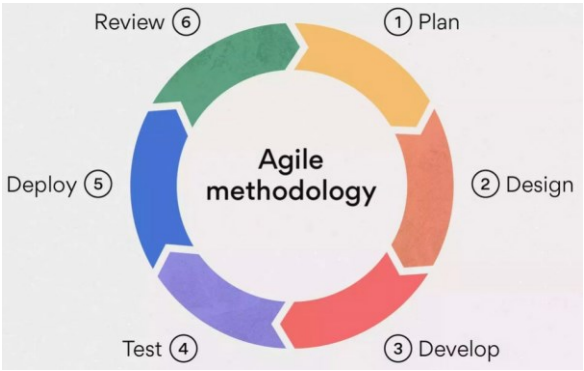
II. Metodologije rukovođenja projektom

Metodologije razvoja softvera predstavljaju okvir za razvoj samog softvera. Primena metodološkog razvoja predstavlja ključnu odluku pri izradi svakog projekta. U okviru rada će biti pokriven agilni metod razvoja softvera pod koji spada i *Scrum*, kao i jedan od tradicionalnih metoda razvoja softvera kao sto je *Vodopad* model radi poređenja.

A. Agile metodologije

Agile je pristup rukovođenju projekta koji je razvijen kao fleksibilniji i efikasniji način da proizvodi dospu na tržište [3]. Sam izraz agile (od eng. *agile*) odnosi se na sposobnost brzog i lakog kretanja. Samim tim, pristup *Agile* omogućava projektnim timovima da se brže prilagode u poređenju sa drugim projektnim metodologijama. Mnogi od današnjih projekata imaju mnogo više nepoznanica nego što bi tradicionalna projektna metodologija mogla adekvatno da izrukuje. Ovakve nesigurnosti postupak definisanja zahteva i mogućnosti adaptacije čine jako izazovnim.

Pristup *Agile* metodologije funkcioniše po principu tekućih sprintova planiranja i izvršavanja zadataka (*taskova*), koji omogućavaju liderima projekta da se kontinualno adaptiraju i da prilagođavaju plan, opseg i projekat softverskog rešenja tokom odmicanja projekta. Baš zbog toga, ovaj pristup zahteva iterativni postupak, koji podrazumeva često i konstantno isporučivanje radnih projekata klijentima. Ovakav pristup osigurava da projektni tim može u kontinuitetu da isporuči konkretne proizvode i rešenja bez kašnjenja koja nastaju promenom ili evolucijom zahteva klijenata. Zbog toga, *Agile* pristup ima visok nivo uključenosti klijenta u tok rada i zahteva česte preglede projekata od strane kako klijenata, tako i celokupnog projektnog tima.



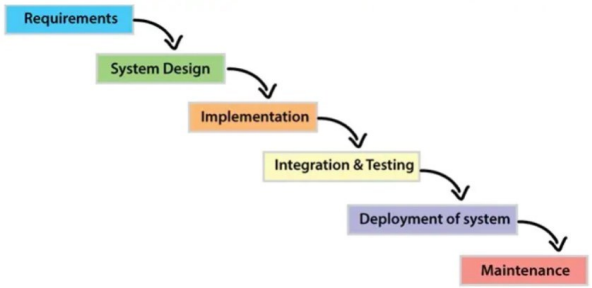
Slika 1. Blok šema *Agile* metodologije.

B. Waterfall metodologije

Vodopad metodologija (ili *model vodopada*) u upravljanju projektima je sekvencijalni i linearni pristup koji se koristi za upravljanje projektima [4]. Sledi strukturirani proces gde se svaka faza projekta završava pre nego što se pređe na sledeću. Metodologija se naziva „vodopad“ jer napredak teče stalno naniže, slično vodopadu. Model vodopada može se koristiti u različitim domenima, uključujući razvoj softvera, inženjering i

M. Đorđević je sa Akademije tehničko-vaspitačkih strukovnih studija, odsek Niš, Univerzitet u Nišu, 18000 Niš, Srbija. E-mail: milan94srb@gmail.com

konstrukciju. Često se koristi u projektima sa striktnim rokom, ograničenim budžetom i fiksnim obimom. Vodopad metodologija prati sekvencijalni pristup upravljanju projektima, koji se sastoji od nekoliko različitih faza. To su sakupljanje zahteva, dizajn, implementacija, testiranje, raspoređivanje i održavanje. Blok šema Vodopad (*Waterfall*) metodologije prikazana je na slici 2.

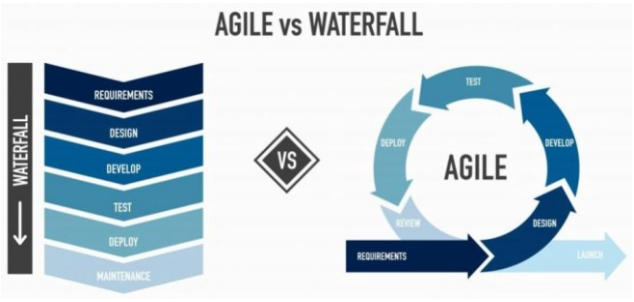


Slika 2. Blok šema Vodopad (*Waterfall*) metodologije.

C. Poređenje metodologija

Waterfall metodologija je linearna forma rukovođenja projektom koja je idealna za projekte kod kojih je krajnji rezultat jasno ustanovljen od početka realizacije projekta. Očekivanja i ishodi svake od faza projekta trebalo bi da budu jasni od početka i ostvareni, kako bi se pristupilo narednoj fazi projekta.

Sa druge strane, *Agile* metodologija je razvijena kao odgovor na rigidnost *Waterfall* metodologije [1, 2]. Kao rezultat, ova metodologija je dosta fluidniji način rukovođenja projektom. Razvoj softvera u nekim slučajevima može trajati i nekoliko godina, i korišćene tehnologije se u tom periodu mogu značajnije promeniti. *Agile* metodologija pokušava da odgovori ovom izazovu i da usvoji metode koji omogućavaju fleksibilnost za usvajanje novih promena kroz realizaciju projekta. U nastavku je data tabela u kojoj su upoređene neke od osnovnih karakteristika navedene dve metodologije za razvoj softvera.



Slika 3. Poređenje *Agile* i *Waterfall* metodologije.

III. SCRUM METODOLOGIJA

Scrum metodologija je tip *Agile* metodologije koja predstavlja jednostavan okvir koji pomaže ljudima, timovima i organizacijama da stvore vrednost putem prilagodljivih rešenja za složene probleme [5]. Bazira se na manifestu za agilan razvoj softvera i propisuje osnovne uloge, predmete (tzv. artefakte) i događaje koje jedan tim treba da koristi u svakodnevnom radu.

Scrum metodologija nema detaljne opise, čak je i sama teorija sa ciljem generalna, jer definiše samo one delove striktno potrebne za implementaciju teorije *Scruma*. Unutar okvira se mogu koristiti različiti procesi, tehnike i metode. Scrum se nadovezuje na postojeće prakse ili ih pak čini nepotrebnim. Na taj način, ova metodologija preispituje relativnu efikasnost trenutnog menadžmenta, okruženja i radnih tehnika, kako bi se mogli unaprediti.

Ova metodologija i njena primena se temelji na kolektivnoj inteligenciji ljudi koji ga koriste. Umesto da ljudima pruža detaljna uputstva, pravila *Scrum* metodologije pruža smernice za njihove odnose i interakcije.

A. Scrum tim

Osnovnu jedinicu *Scruma* čini mali tim ljudi, *Scrum* tim. On se sastoji od jednog *Scrum Mastera*, jednog *Product Ownera* i *Developera*. U *Scrum* timu ne postoje podtimovi niti hijerarhije. To je kohezivna jedinica profesionalaca fokusiranih na jedan cilj u jednom trenutku, *Product Goal* [1, 2, 5].

Scrum timovi su krosfunkcionalni, što znači da članovi imaju sve veštine potrebne da kreiraju vrednosti u događajima *Scruma* (*Sprints*). Takođe su samoupravljani, što znači da interno odlučuju ko radi šta, kada i kako.

Scrum tim je dovoljno mali da ostane spretan, a dovoljno veliki da obavi značajnu količinu posla u *Sprintu*. Obično ima 10 ili manje ljudi. Uopšteno, zaključeno je da manji timovi komuniciraju bolje i da su produktivniji. Ako *Scrum* timovi postanu preveliki, trebaju razmotriti reorganizaciju u više kohezivnih *Scrum* timova, od kojih je svaki fokusiran na isti proizvod. Stoga, potrebni su isti *Product Goal*, *Product Backlog* i *Product Ownera*.

Scrum tim je odgovoran za sve aktivnosti koje se odnose na proizvod, od saradnje sa stejkholderima, verifikacije, održavanja, operacije, eksperimentisanja, istraživanja i razvoja, do bilo čega drugog što bi moglo biti potrebno. *Scrum* tim je struktuiran i osnažen od strane organizacije da sam upravlja svojim poslom. Rad u *Sprintovima* održivim tempom poboljšava fokus i doslednost *Scrum* tima. Celi *Scrum* tim je odgovoran za kreiranje vrednog i korisnog inkrementa u svakom *Sprintu*. *Scrum* definiše tri specifične odgovornosti unutar *Scrum* tima: *Developeri*, *Product Owner* i *Scrum Master*.

B. Developeri u Scrum timu

Developeri su ljudi u *Scrum* timu koji su posvećeni kreiranju bilo kojeg aspekta upotrebljivog inkrementa proizvoda u svakom *Sprintu*. Specifične veštine koje Developeri moraju imati su često raznovrsne i variraće u

zavisnosti od područja posla. Međutim, Developeri su uvek odgovorni za: Kreiranje plana za *Sprint*, *Sprint Backlog*; uvođenje kvaliteta pridržavanjem *Definition of Done*; svakodnevno prilagođavanje vlastitog plana *Sprint Goalu*; i, pozivanje jedni drugih na odgovornost kao stručnjaci.

C. Product Owner u Scrum timu

Product Owner je odgovoran da maksimalno poveća vrednost proizvoda koji je rezultat posla *Scrum* tima. Način na koji se to postiže može značajno varirati među organizacijama, *Scrum* timova i pojedincima. *Product Owner* je takođe odgovoran za efikasno upravljanje *Product Backlogom*, što uključuje: Razvoj i eksplicitno komuniciranje *Product Goala*; kreiranje i jasno komuniciranje stavki *Product Backloga*; određivanje redosleda stavki *Product Backloga*; i, vođenje računa o tome da je *Product Backlog* transparentan i razumljiv.

Product Owner može raditi opisani posao ili može delegirati zaduženja drugima, ali *Product Owner* ostaje odgovoran. Da bi *Product Owneri* bili uspešni, cela organizacija mora poštovati njihove odluke. Ove odluke se odražavaju u sadržaju i redosledu stavki *Product Backloga*, i kroz inspektabilni *Increment* na *Sprint Review* događaju. *Product Owner* je jedna osoba, nije odbor. *Product Owner* može zastupati potrebe mnogih stejkholdera kroz *Product Backlog*. Oni koji žele menjati *Product Backlog* to mogu uraditi tako što će pokušati da uvere *Product Ownera*.

D. Scrum Master u Scrum timu

Scrum Master je odgovoran za uspostavljanje *Scrum* metodologije onako kako je to definisano *Scrum* vodičem [1]. *Scrum* Master to čini pomagajući svima unutar *Scrum* tima i organizacije da razumeju teoriju i praksu *Scruma*.

Scrum Master je odgovoran za efikasnost *Scrum* tima. To postiže pomažući *Scrum* timu da poboljša svoje prakse u sklopu *Scrum* okvira. *Scrum Masteri* su istinske vođe koje služe *Scrum* timu i većim organizacijama. *Scrum Master* služi *Scrum* timu na nekoliko načina, i to: Podučava članove tima o samoupravljanju i krosfunkcionalnosti; pomaže *Scrum* timu da se fokusira na kreiranje inkrementa visoke vrednosti koji ispunjavaju *Definition of Done*; uklanja prepreke ka napretku *Scrum* tima; i, osigurava da se svi *Scrum* događaji održavaju, te da su pozitivne, produktivne i da se dešavaju u sklopu predviđenog vremenskog okvira.

E. Scrum artefakti

Scrum artefakti predstavljaju rad ili vrednost. Namenjeni su povećanju transparentnosti ključnih informacija. Dakle, svi koji imaju uvid u artefakte, poseduju jednaku osnovu za adaptaciju. U svakom artefaktu je sadržana obaveza da se osigura pružanje potrebnih informacija koje će poboljšati transparentnost i fokus na osnovu kojih se napredak može pratiti: Za *Product Backlog* to je *Product Goal*; Za *Sprint Backlog* to je *Sprint Goal*; Za *Increment* to je *Definition of Done*. Ove obaveze postoje kako bi se ojačao empirizam i *Scrum* vrednosti za *Scrum* tim i ostale stejkholdere.

Product Backlog predstavlja uređenu listu onoga što je potrebno za poboljšanje proizvoda koja se stalno menja. To je jedini izvor posla za *Scrum* tim. *Product Goal* opisuje buduće stanje proizvoda koje može poslužiti *Scrum* timu kao cilj za planiranje.

Sprint Backlog se sastoji od *Sprint Goala* (zašto), *Product Backlog* stavki odabranih za *Sprint* (šta), kao i efikasnog plana za isporuku *Incrementa* (kako). *Sprint Backlog* predstavlja plan za Developere koji su oni sami napravili. *Sprint Goal* predstavlja jedinu svrhu *Sprinta*. Iako je *Sprint Goal* obavezujući za Developere, on omogućava fleksibilnost u smislu potrebnog rada za postizanje cilja. *Sprint Goal* također stvara koherentnost i fokus, ohrabrujući *Scrum* tim da radi zajedno, umesto na različitim inicijativama.

Increment predstavlja odskočnu dasku za postizanje *Product Goala*. Svaki *Increment* je dodatak svim prethodnim *Incrementima* i detaljno je proveren, što osigurava da svi *Incrementi* rade zajedno. *Definition of Done* je formalni opis stanja *Incrementa* u kojem se on nalazi onda kada zadovolji kriterije kvaliteta proizvoda. *Increment* nastaje onog trenutka kada stavke iz *Product Backloga* zadovolje *Definition of Done*.

F. Scrum događaji

Scrum obuhvata različite događaje. Svaki događaj je formalna prilika da se pregledaju i prilagode *Scrum* artefakti.

Sprintovi su događaji koji su srce *Scruma*, gde se ideje pretvaraju u vrednost. *Sprintovi* su događaji koje imaju fiksnu dužinu trajanja od jednog meseca ili manje, kako bi se uspostavila doslednost. Novi *Sprint* počinje odmah po zaključivanju prethodnog *Sprinta*. Celokupni posao kojeg je potrebno uraditi kako bi se postigao *Product Goal* se dešava u sklopu *Sprintova*. U sklopu *Sprinta* razlikuju se *Sprint Planing*, *Sprint Review* i *Sprint Retrospective*.

Daily Scrum je događaj čiji cilj jeste pregledati napredak ka *Sprint Goalu* i prilagoditi *Sprint Backlog* ukoliko je potrebno, podešavajući predstojeći isplanirani posao.

IV. CASE STUDY – KOMPANIJA BETTER COLLECTIVE

Implementacija *Scrum* metodologije biće prikazana na konkretnom primeru kompanije *Better Collective*. *Better Collective* je vodeća sportska medijska grupa koja kreira inovativne tehnologije i pouzdane platforme fokusirane na zabavne i odgovorne sadržaje za sportsko klađenje [6].

Zahvaljujući svojim platformama i alatima *Better Collective* povezuje korisnike sa *iGaming* operaterima koji

odgovaraju njihovim potrebama. Portfolio kompanije uključuje: *Action Network*, *VegasInsider.com*, *HLTV.org*, *FUTBIN.com* i *bettingexpert.com*. Transparentnost je jedna od osnovnih vrednosti *Better Collective* i ona je ključna za svaku akciju koja se preduzima.

A. Stanje pre implementacije Scrum metodologije

Scrum kao metodologija za razvoj softvera se u okviru kompanije nije koristila od samog starta. Pre 4-5 godina su tadašnji projektni menadžeri uvideli potrebu da naprave promenu i isprobaju *Scrum* metodologiju kao zamenu za tadašnji način rada. U tom trenutku se koristio *Kanban* kao metodologija, u kombinaciji sa softverom pod imenom „*Trello*“ kao rešenje za rukovođenje projektima, koje nudi dobru integraciju sa *Kanban* metodologijom. Zajedno sa promenom metodologije, kompanija je prešla i na drugo rešenje za rukovođenje projektima u vidu servisa pod imenom „*Jira*“, koja je zamenila „*Trello*“.

B. Uzrok i prva verzija implementacije

Better Collective, kao kompanija koja se između ostalog bavi i marketingom, radi u veoma dinamičnom okruženju. S tim na umu, rukovodstvo kompanije je uvidelo potrebu za čestom i brzom adaptacijom na promene web tehnologija. Takođe, za razliku od mnoštva IT kompanija koje se bave *outsource*-ingom, *Better Collective* često sam pravi softverska rešenja za svoje potrebe, pa je bila potrebna i adaptivnost i brze promene pri pisanju koda. Sve ove potrebe su dovele do ideje i prve implementacije Scrum metodologije u okviru kompanije. *Scrum* se u početku implementirao u okviru projekata po potrebi, i nije se implementirao u celosti već u improvizovanom obliku.

C. Prednosti i mane implementacije

Scrum se i danas aktivno koristi u okviru kompanije, i kroz vreme su uviđene i prednosti i mane ove metodologije. *Scrum* se uglavnom koristi za kompleksnije projekte, kako bi stejkholderima bio omogućen uvid u stanje projekta u svakom trenutku, kao i mogućnost da ostave *feedback* na osnovu koga je moguće adaptirati plan i nastaviti sa projektom u pravom smeru.

Dok se *Scrum* metodologija pokazala kao odlično rešenje u procesu razvoja proizvoda, problemi su nastajali kada je u pitanju održavanje proizvoda. Jedan od problema jeste bilo forsiranje plana za naredne dve nedelje što ponekad nije moguće, i onda mogu nastati „rupe“ u rasporedu koji nije moguće naknadno izmeniti. Implementacija *Scrum* metodologije takođe ostavlja manje prostora za dodatne izmene zahteva na nivou samih zadataka. Još jedan od problema su predstavljali sastanci gde je bilo potrebe korisnicima objasniti i prikazati kako će koristiti proizvod koji je još uvek nedovršen tj. u izradi, pa se često dešavalo da ti prikazi i objašnjenja budu previše apstraktni budućim korisnicima i ne daju pravu sliku.

D. Implementacija Spotify modela

Povećanjem broja zaposlenih u okviru kompanije se povećavao i broj različitih grana rada kao i timova u okviru tih grana. Vremenom su počeli da se javljaju problemi u komunikaciji između timova, sto je sa sobom povelo diskusiju na temu skaliranja *Scrum* metodologije na nivou kompanije. Ideja je bila da bi ljudi, koji su inače bili u čestoj komunikaciji, trebalo da budu na istom mestu tj. u okviru iste logičke celine, pa je kao rešenje ovom problemu na nivou kompanije implementiran „*Spotify*“ model na osnovu koga su zaposleni u kompaniji trenutno organizovani [7].

III. ZAKLJUČAK

Na osnovu teorijskih teza dostupnih u okviru rada, kao i analize praktičnog primera implementacije Scrum metodologije u okviru kompanije *Better Collective*, moguće je zaključiti da metodologija, iako veoma fleksibilna, nije odgovarajuće rešenje za bilo kakav projekat. Raznovrsnost prikazanih metodologija ukazuje da je za postizanje najveće efikasnosti potrebno dobro razumevanje metodologija, ali i dobro razumevanje tima, sredine i naravno iskustvo. Takođe, bitno je napomenuti da *agile* metodologije, kao takve, ne treba pratiti striktno, već ih koristiti kao smernice i adaptirati na odgovarajući način za individualni slučaj.

ZAHVALNICA

Ovaj rad nastao je iz završnog rada na Akademiji tehničko-vaspitačkih strukovnih studija, Odsek Niš. Ovom prilikom bih prvenstveno želeo da se zahvalim svom mentoru dr Milošu Stojanoviću na pomoći oko rada. Takođe bih iskoristio priliku da se zahvalim kompaniji Better Collective i svojim kolegama, kao i kolegama iz struke, na informacijama koje su doprinele na realizaciji ovog rada.

REFERENCES

[1] K. Schwaber and J. Sutherland, “Scrum vodič - Konačni Scrum vodič: Pravila igre”, 2020.
[2] S. Atanasijević, “Životni ciklus razvoja softvera”, 2022.
[3] Agile Methodology Basics [Online]. Dostupno na: https://www.wrike.com/project-management-guide/agile-methodology-basics
[4] J. Ng, Metodologija vodopada [Online]. Dostupno na: https://ahaslides.com/bs/blog/waterfall-methodology
[5] Scrum vs. Kanban vs. Scrumban [Online]. Dostupno na: https://eylean.com/wp-content/uploads/2019/12/Whitepaper-Scrum-vs-Kanban-vs-Scrumban.pdf
[6] Sajt kompanije Better Collective https://bettercollective.com/about/
[7] M. Cruth, Discover the Spotify model [Online]. Dostupno na: https://www.atlassian.com/agile/agile-at-scale/spotify

VTŠFit Aplikacija za prevenciju povreda tokom vežbanja korišćenjem veštačke inteligencije

Aleksa Zdravković, Strahinja Antić, Mihajlo Cekić,
Petar Virijević, Kristina Stevanović, Nikola Vukotić

Sadržaj – U ovom radu opisana je implementacija veštačke inteligencije unutar VTŠFit aplikacije za prevenciju povreda tokom vežbanja sportista. Primenjen je model dubokog učenja – *YOLO Pose* kako bi se detektovale ključne tačke na telu. Uz pomoć tačaka na telu izvršena je analiza ispravnosti vežbe izračunavanjem različitih ključnih uglova. U radu su dati opisi korišćenih tehnologija kao i način funkcionisanja same aplikacije.

I. UVOD

Veštačka inteligencija predstavlja interdisciplinarnu oblast koja spaja računarsku nauku sa obimnim skupovima podataka kako bi omogućila rešavanje različitih problema. Ona obuhvata podoblasti kao što su mašinsko učenje (ML – *Machine Learning*) [1] i duboko učenje (DL – *Deep Learning*) [2], često spominjane u vezi sa ovom oblašću. Ove podoblasti koriste algoritme koji imaju za cilj stvaranje sistema sposobnih za predviđanja ili klasifikacije na osnovu unetih podataka [3]-[5].

Mašinsko učenje je podoblast veštačke inteligencije koja se usmerava na upotrebu podataka i algoritama kako bi oponašala način na koji ljudi uče, postepeno poboljšavajući tačnost svojih predviđanja. Primena mašinskog učenja može se ogledati u situacijama kada je zadat složen zadatak ili problem koji uključuje puno podataka i promenljivih, ali ne postoji jednačina za njegovo rešavanje [6].

Duboko učenje je specijalizovana forma mašinskog učenja. U toku primene algoritama mašinskog učenja, relevantne karakteristike sa slikovnih podataka se ručno izdvajaju, a zatim koriste za kreiranje modela koji kategorizuje slike. Nasuprot tome, u dubokom učenju, karakteristike se automatski izdvajaju iz slika. Duboko učenje izvodi tzv. „učenje od kraja do kraja“ (engl. *End to end learning*) pri čemu se algoritmu prosleđuju sirovi podaci i zadatak koji treba izvršiti, nakon čega se uči kako automatski postići rešenje zadatka.

Svrha korišćenja ovakvih algoritama je višestruka. Oni omogućavaju brže donošenje odluka u kraćem vremenskom periodu uz nižu cenu. Na taj način teži se ka automatizaciji i pojednostavljenju svakodnevnih poslova za pomoć čoveku.

Primena veštačke inteligencije je sve veća. U radu [7] se upoređuje nekoliko tehnika mašinskog učenja za predviđanje prostorno-vremenskog korišćenja resursa od strane aplikacija (na primer mikroarhitektura CPU-a, veličina i brzina memorije, karakteristike ulaznih podataka i sl.). Rad [8] se bavi predviđanjem potencijalnog odustajanja studenata od školovanja na osnovu podataka iz programa za online učenje nad uzorkom od 62.375 studenata. S druge strane u radu [9] je obrađeno istraživanje korišćenjem veštačke inteligencije koje omogućava etiketiranje reči u rečenici na srpskom jeziku na osnovu vrste, podvrste reči i padeža.

U ovom radu je prikazana primena modela veštačke inteligencije koja pomaže korisnicima da izvode osnovne vežbe u teretani pružajući savete za pravilno izvođenje. Na taj način se u odsustvu personalnog trenera, smanjuje rizik od povrede i omogućava praćenje i unapređenje forme tokom vežbanja.

A. Motivacija

Nepravilno izvođenje vežbi sa opterećenjem (tegovima), najčešći su uzrok povreda tokom treninga. Popularne vežbe snage (čučanj i mrtvo dizanje), ukoliko se prilikom izvođenja vežbe opterećenje oslanja kičmu ili kolena, umesto na mišiće, one mogu dovesti do mikrotraume koje nagomilavanjem dovode do povrede. Početnicima bez podrške iskusnog trenera je jako teško da održe pravilnu formu izvođenja vežbe. Ovaj rad ima za cilj da spreči povrede i time omogući ljudima da pravilno treniraju i poboljšaju kvalitet svojih sportskih aktivnosti, a samim tim i kvalitet života.

II. MODELI I BIBLIOTEKE

U razvoju same aplikacije korišćeni su *open-source* razvojni alati. Sve tehnologije i korišćeni modeli su opisani i objašnjena njihova primena u narednim potpoglavljima.

A. Python

Python je jednostavan, čitljiv i veoma popularan programski jezik sa podrškom za različite paradigme u programiranju [10]. Koristi se u oblastima veštačke

A. Zdravković je student druge godine Elektronskog fakulteta Univerziteta u Nišu na modulu Računarstvo i informatika, A. Medvedeva 14, Srbija, e-mail: aleksazdravkovic15@gmail.com

S. Antić, M. Cekić, P. Virijević, K. Stevanović i N. Vukotić sa Katedre za informaciono-komunikacione tehnologije, Akademija tehničko-vaspitačkih strukovnih studija Odsek Niš, A. Medvedeva 20, Srbija, e-mail: nikola.vukotic@akademijanis.edu.rs

inteligencije zbog svoje obimne podrške, brzog razvoja, prenosivosti i sposobnosti za vizualizaciju podataka.

B. Biblioteka OpenCV i Tkinter

OpenCV (engl. *Open Source Computer Vision Library*) je biblioteka otvorenog koda koja uključuje nekoliko stotina algoritama za računarski vid. Poseduje modularnu strukturu, što znači da paket uključuje nekoliko deljenih ili statičkih biblioteka. Dostupni su različiti moduli, kako na primer za procesiranje slika, videa, detekciju objekata, itd. [11].

Tkinter je standardna biblioteka za grafički interfejs (GUI) u Python-u. Koristi se za kreiranje GUI aplikacija sa grafičkim elementima kao što su prozori, dugmad, okviri, unosi teksta i mnogi drugi elementi koji omogućavaju korisnicima da interaguju sa programom. *Tkinter* je deo standardne Python biblioteke, što znači da je dostupan bez potrebe za dodatnim instalacijama [12].

C. YOLO model

YOLO (engl. *You Only Look Once*) je model za detekciju objekata i segmentaciju slika. Objavljen je 2015. godine i ubrzo stekao popularnost zbog svoje brzine i preciznosti [13]. Do sada je objavljeno više verzija ovog modela, a trenutno je aktivna verzija 8 koja je i upotrebljena prilikom realizacije praktičnog dela ovog rada [14]. Dostupni su *YOLOv8* modeli za detekciju, segmentaciju i prepoznavanje tačaka na telu koji su trenirani na *COCO* [15] skupu podataka, a takođe postoje i modeli za klasifikaciju trenirani nad *ImageNet* [16] skupom podataka. Svi modeli su sadržani u poslednjoj objavljenoj verziji [17].

YOLO model se u praksi često koristi kako zbog svoje preciznosti tako i zbog brzine. Postoje ugrađene metode za dotreniravanje ovog modela, a ovaj deo će detaljnije biti opisan u poglavlju III.

D. Cvat

Cvat (engl. *Computer Vision Anotation Tool*) je alat za anotaciju podataka na slikama i video snimcima. Koristi se zbog svoje jednostavnosti i adaptivnosti za različite modele veštačke inteligencije [18]-[19].

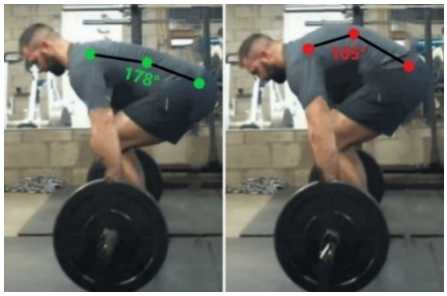
III. TRENIRANJE MODELA

Implementacija modela veštačke inteligencije u jedinstvene projekte uglavnom zahteva „dotreniravanje" modela. Modelu se isporučuju novi podaci sa obeleženim pojedinostima koje su bitne za model. Time se pospešuje njegova predikcija specifičnih podataka za dati projekat.

Npr. za projekat u kojem se na osnovu slike prepoznaje poza u jogi, potrebno je model „hraniti" upravo odgovarajućim slikama sa obeleženim podatkom o pozi.

Za složenije modifikacije izvornog modela, kao što je dodavanje nove tačke na telu čoveka, potrebno je dosta više podataka i postoji potreba za nadograđivanjem osnovnih klasa *YOLO Pose* modela.

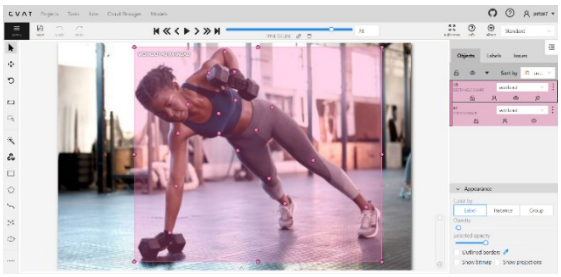
Projekat prepoznavanja pravilnosti forme prilikom izvođenja vežbi zahteva preuređivanje postojećih klasa *YOLO* modela. Uz osnovne tačke koje model pamti, za projekat je bitno da postoje tačke središnjeg dela kičme, koje se koriste u izračunavanju procenta krivljenja kičme. Primer primene tačke na središtu kičme se nalazi na Slici 1.



Slika 1. Pravilno i nepravilno izvedena vežba.

Zbog važnosti središnje tačke na kičmi, u ovom radu izvršeno je dodatno treniranje modela sa dodatom pomenutom tačkom kako bi se predikcija pravilnih vežbi podigla na viši nivo, a i kako bi bila sigurnija po korisnike.

Za anotaciju podataka, koje se daju modelu, koristi se *CVAT*, kao na Slici 2, koji se zbog svoje jednostavnosti svrstava među najpopularnije alate za treniranje modela.



Slika 2. Anotacija tačaka na slici.

Kada se anotira set podataka potreban za treniranje modela potrebno je pokrenuti skriptu sa Slike 3. Dužina treniranja zavisi od količine podataka.

```
from ultralytics import YOLO

model = YOLO("yolov8n-pose.pt")
# Učitamo pretrenirani model
model.train(data="config.yaml", epochs=1)
```

Slika 3. Treniranje modela.

Može se uočiti poziv metode `train` koja uz nove podatke (`data`) vrši dotreniravanje modela u nekoliko iteracija (`epochs`). Dodatno treniranje u ovom radu vršeno je u jednoj epohi, što se može videti u vrednosti drugog parametara metode [20].

Trenutna količina slika treniranih i validiranih u modelu je 103. Sa stanovišta tačnosti, podatak o tački za

kičmu trenutni model nije spreman za široku upotrebu u realizaciji. Za primenu ovako osetljivog podatka koji se koristi u medicinske svrhe, krucijalna je visoka preciznost za koju je potrebno model trenirati sa hiljadama podataka. Te se u vreme pisanja ovog rada za ocenjivanje forme koriste samo osnovne tačke *YOLO Pose* modela.

IV. Implementacija

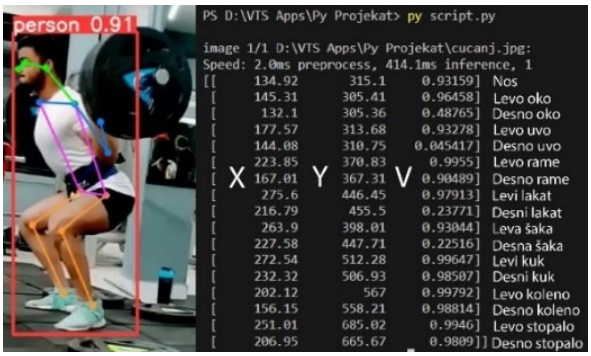
Primena *YOLO Pose* modela na slici kao jedna od najjednostavnijih implementacija, može se naći na Slici 4.

```
from ultralytics import YOLO
# Pre ovoga uraditi pip install ultralytics
model = YOLO('yolov8m-pose.pt')
# Učitamo model
results = model(source='cucanj.jpg', conf = 0.3, save=True)
```

Slika 4. Primer osnovne implementacije.

Rezultat modela dobija lista, svaki član te liste je jedna obrađena sliku odnosno *frame* [20].

Zatim u okviru obrađene slike se pristupa generisanim tačkama korišćenjem `.keypoints` svojstva, u tom svojstvu se nalazi lista od 17 tačaka za svaku bitnu tačku na telu. Tačke mogu biti predstavljene u obliku `[x,y]` ili `[x,y,visibility]`.

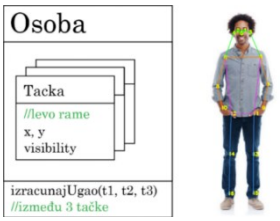


Slika 5. Prikaz rezultata i x, y, visibility komponente tački.

`Visibility` predstavlja sigurnost modela da je tačka vidljiva, tačke sa `visibility` manjom od 0.5 model automatski ne prikazuje u vizuelnoj prezentaciji, to je moguće primetiti na Slici 5. Ovim postupkom može se pristupiti svim koordinatama pojedinačno i njima upravljati u svrhu cilja projekta.

A. Klasifikacija podataka

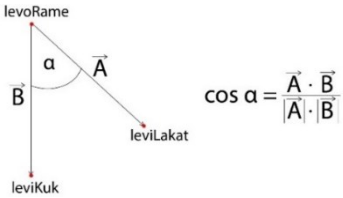
Da bi se lakše pristupalo komponentama koriste se klase. Klasa `Tacka` konstruisana je uz pomoć tri prethodno navedene komponente jedne tačke, klasa `Osoba` sastoji se od 17 tačaka i ima metode za manipulisanje tačkama. Slikoviti prikaz pomenutih klasa se nalazi na Slici 6. Uz pomoć klasa, primer pozivanja koordinate bi izgledao ovako: `osoba.levoRame.x`



Slika 6. Šematski prikaz klasa.

B. Računanje uglova između tačaka

Uglovi između tačaka tela su krucijalni pokazatelji forme izvođenja vežbe, samim tim metoda za izračunavanje uglova je sam ključ ovog projekta. Koordinate tačaka su smeštene u ravni, te se sa njima može manipulisati primenom formula unutar Euklidove geometrije u ravni, kao što je prikazano na Slici 7. Za svaku vežbu, neophodno je definisati tačne uglove kojim se definiše pravilno izvođenje.



Slika 7. Formula za ugao između dva vektora.

V. Realizacija

A. Pravila za vežbe

Pravila i limiti su jedinstveni za svaku vežbu, limiti su postavljeni u skladu sa medicinskim preporukama. Prilikom izvođenja vežbe čučanj, najčešće su povrede kolena i kičme. Upravo zbog toga je najbitnije obratiti pažnju na te delove tela. Zbog toga se prate uglovi između:

1. Ramena–Kičme–Kuka 2. Kuka–Kolena–Stopala

Ugao između kolena, kuka i stopala određuje fazu u jednom ponavljanju. „Faze“ su predstavljene bojom za lakše raspoznavanje. Plava - početni stav, zelena - krajnji stav i crvena - prenisko, mogućnost povrede visoka [21]. Na Slici 8. može se videti primer svih mogućnosti.



Slika 8. Indikatori ispravnog rada vežbe.

B. Uputstvo za upotrebu

Grafički interfejs aplikacije sastoji se iz tri dela. Odabir videa, odabir vežbe i prikaz rezultata. U delu gde se bira

izvor, moguće opcije su: prethodno snimljen video smešten u folderu *videoSnimci*, kao na Slici 9, snimiti novi video korišćenjem *web* kamere i vežbanje uživo preko *web* kamere. Nakon odabrane metode za unos videa, potrebno je odabrati vežbu koja se očitava.

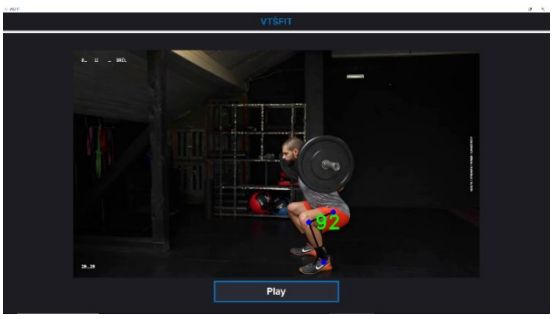


Slika 9. Korisnički interfejs.

Odabir se svodi na šest popularnih vežbi:

1. Čučanj
2. Mrtvo dizanje
3. Sklekovi
4. Trbušnjaci
5. Biceps
6. Letenje

Poslednji prozor prikazuje obrađeni video sa naznačenim greškama i prikazanim upozorenjima, ukoliko postoje. Na Slici 10, prikazan je rezultat ispravno odrađenog dela vežbe za čučanj.



Slika 10. Pravilno odrađen čučanj u aplikaciji.

VI. Zaključak

U ovom radu realizovan je sistem za prevenciju povreda korišćenjem *YOLO Pose* modela, pomoćnih biblioteka u programskoj jeziku Python i pomoćnih alata za anotaciju slika. Implementirani sistem je u stanju da na osnovu postojećeg video snimka, video snimka koji se snima prilikom korišćenja aplikacije, kao i iz snimka sa kamere analizira prethodno izabranu vežbu korisnika i daje mu povratne informacije o tome da li se vežba radi ispravno. Primećena je još jedna veoma važna tačka na telu, na kičmi, koja igra važnu ulogu u praćenju ispravnosti vežbi. U tu svrhu izvršeno je dodatno treniranje modela kako bi se i ova tačka dodatno označavala. Međutim, kako je nova tačka kičme jako bitna i potencijalno opasna, za praktičnu primenu i upotrebu, minimalan procenat tačnosti je 90%. Naredni cilj u okviru ovog projekta jeste dovođenje modela do potrebne sigurnosti, kroz treniranje modela većim brojem uzoraka, u cilju postizanja preciznijih podataka koje su ključ uspeha i praktične primene aplikacije.

Literatura

[1] IBM, "What is machine learning", https://www.ibm.com/topics/machine-learning.

[2] MathWorks, "What is Machine Learning", https://www.mathworks.com/discovery/machine-learning.html

[3] IBM, "What is artificial intelligence (AI)", https://www.ibm.com/topics/artificial-intelligence

[4] N. Laskowski. "Atrificial intelligence (AI)" What is Artificial Intelligence and How Does AI Work? TechTarget. https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence (pristupljeno Okt. 19, 2023.)

[5] Artificial Intelligence: Ai Technology and Deep Learning Systems Explained. (2020). Italija: Self Publisher.

[6] S. Brown. "Machine learning, explained" https://www.mathworks.com/discovery/machine-learning.html (pristupljeno Okt. 15, 2023).

[7] A. Matsunaga and J. A. B. Fortes, "On the Use of Machine Learning to Predict the Time and Resources Consumed by Applications," 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, VIC, Australia, 2010, pp. 495-504, doi: 10.1109/CCGRID.2010.98.

[8] M. Tan and P. Shao, "Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method", Int. J. Emerg. Technol. Learn., vol. 10, no. 1, pp. pp. 11–17, Feb. 2015.

[9] N. Vukotić, S. Petrović and S. Stojković, "Implementation of the Serbian Language POS Taggers Using the NLTK Library" 2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Nis, Serbia, 2023, pp. 151-154.

[10] Python. Dostupno na: https://www.python.org/

[11] OpenCV. Dostupno na: https://pypi.org/project/opencv-python/

[12] Tkinter. Dostupno na: https://docs.python.org/3/library/tkinter.html

[13] G. Jocher, "Ultralytics YOLOv8 Docs" https://docs.ultralytics.com/ (Pristupljeno Sep. 11, 2023.)

[14] H. Muhammad, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection" Machines (jun 2023). https://doi.org/10.3390/machines11070677.

[15] G. Jocher, "COCO - Ultralytics YOLOv8 Docs", https://docs.ultralytics.com/datasets/detect/coco/ (Pristupljeno Sep. 16, 2023.)

[16] G. Jocher, ImageNet - Ultralytics YOLOv8 Docs, https://docs.ultralytics.com/datasets/classify/imagenet/ (Pristupljeno Sep. 15, 2023.)

[17] GitHub – ultralytics, Dostupno na: https://github.com/ultralytics/ultralytics

[18] GitHub – Cvat. https://github.com/opencv/cvat

[19] Cvat.ai. Dostupno na: https://www.cvat.ai/

[20] N. Vukotić, A. Zdravković, S. Antić, M. Cekić (2023). "VTŠFit" GitHub. https://github.com/nikolavukotic/VTS_Apps_Team_AI

[21] Vuk Vukašinović, Srbija, *Čučanj – Pravilno Izvođenje* (maj 2020.). Dostupno na: www.youtube.com/watch?v=WU_ks_hJ7nY&t=241s

Linear Circuit Solver Based on Polymorphism and Automatic Differentiation

Uroš Stojković

Abstract - This paper presents a novel method for analysis of linear electrical circuits in both AC and DC modes, in code. Initially, we utilize polymorphism to precisely determine voltages and currents in tree-like circuit structures. Recognizing its limitation in handling more complex topologies, we introduce a modified, numeric approach employing automatic differentiation. This extended method offers flexibility for solving a broader range of circuit configurations, opening new possibilities for tackling a wider array of circuit problems.

I. Introduction

The classical methods for circuit analysis, commonly taught in introductory electrical engineering courses, traditionally involve transforming the original system of differential equations into a system of algebraic equations. These equations are typically solved using methods such as Gaussian elimination or numerical optimization. Two well-known techniques are nodal and mesh analysis [1,2].

In specific cases where the circuit's topology can be represented as a tree-like structure of series and parallel circuits, the solution process is notably simpler and doesn't necessitate the setup of a complex equation system.

In the first section of this paper, we explore the application of polymorphism, a fundamental concept in object-oriented programming, to exploit this simplified scenario. Subsequently, in the second section, we address the need to generalize our solution method to encompass arbitrary circuit topologies. To achieve this, we return to nodal analysis while introducing automatic differentiation to enable numerical optimization. In the final section, we discuss the advantages and implications of adopting this approach.

II. Related work

Popular software solutions for solving simulating electric circuits include SPICE, LTspice, and PSpice, which use various numerical methods to approximate the solution to circuit equations [5,6,7]. These solutions are comprehensive, widely used in industry, but very sophisticated.

U. Stojkovic is with the Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: uros.stojkovich@elfak.rs

III. Simple Topologies

A. *Theory*

The central insight lies in viewing circuit elements as linear two-terminal components. This perspective is crucial for two reasons. Firstly, it aligns with the nature of simple linear electrical components such as resistors, capacitors, and ideal voltage sources, all of which inherently possess two-terminal characteristics. More complex two-terminal components can be formed by combining primitive elements or other complex components in series or parallel configurations. Effectively, this approach allows us to represent a portion, or at times, an entire circuit, as a tree structure, where inner nodes symbolize series or parallel connections, and leaf nodes represent primitive components.

Secondly, linear two-terminal components can be precisely characterized by their current-voltage characteristic (CVC) [1], which also exhibit linearity and follows a general form:

$$aV + bI = c \tag{1}$$

and thus, can be represented by a tuple *(a, b, c)*. Here, V and I are, in general, complex numbers representing effective voltage and current in either DC or AC mode. Each type of primitive component possesses its unique CVC, and we can establish rules for calculating the CVC of composite components based on their constituent elements. In Table 1, CVCs of primitive two-terminal components are shown (ω stands for angular frequency, j for imaginary unit).

TABLE I
CVCs of Primitive Components

Component Type	*(a, b, c)*
Resistor (R)	(1, -R, 0)
Capacitor (C)	(1, j/(ωC), 0)
Inductor (L)	(1, -jωL, 0)
Impedance (Z)	(1, -Z, 0)
Ideal Voltage Source (E)	(1, 0, E)
Ideal Current Source (J)	(0, 1, J)

To better understand the combination of Current-Voltage Characteristics (CVCs) in series and parallel circuits, we begin with the series connection. When connecting two CVCs in series, if at least one of them is of the form $I = J$, the resulting CVC remains the same.

However, in cases where neither CVC follows this form, a relationship arises between the component parameters:

$$b = b_1 + b_2$$
$$c = c_1 + c_2 \quad (2)$$

This elegant relationship implies that both the Thevenin equivalent impedance and voltage are obtained simply by summing the corresponding parameters.

Conversely, in parallel connections, we encounter several scenarios. If one of the CVCs takes the form $V = E$, the resulting CVC remains unchanged. However, when one CVC is of the form $I = J$, while the other is of the form $V + b_1 I = c_1$, we find the following relationships:

$$b = b_1$$
$$c = c_1 + b_1 J \quad (3)$$

In the most general case where both CVCs are of the form $V + bI = c$, the following relationships apply:

$$b = \frac{b_1 b_2}{b_1 + b_2}$$
$$c = \frac{c_1 b_2 + c_2 b_1}{b_1 + b_2} \quad (4)$$

It is important to note that both series and parallel circuits can consist of more than two components. In such cases, these formulas are iteratively combined to determine the final CVC. Below, we provide an algorithm for calculating the CVC of a series circuit:

1. Initialize *cvc* as (1, 0, 0).
2. For each child node, combine *cvc* in series with CVC of the child node.
3. Return *cvc*.

The same algorithm for parallel circuits is basically identical except that *cvc* is initialized as (0, 1, 0) which is a neutral element for combining in parallel. So, it is evident that to evaluate the CVC of a parent component, we rely on CVCs of its child components. Thus, the computation of CVCs goes from leaf nodes towards the root, following a post-order traversal.

As noted earlier, CVC provides us with means to calculate either voltage across a two-terminal component, given its current, or vice versa. Consequently, to compute voltages and currents across the entire circuit, we must initiate the process by applying either voltage or current to the root component. The choice depends on the nature of the root:

1. Series Circuit – we apply zero voltage, effectively short-circuiting the two terminals.

2. Parallel Circuit – we apply zero current, signifying that the only connection between the terminals exists through its child components.

Once we have determined the current and voltage at the root, we can systematically propagate this information to its children, then to their children, and so forth, following a recursive pattern. Consequently, the computation of voltages and currents flows from the root towards the leaf nodes, adhering to a pre-order traversal approach.

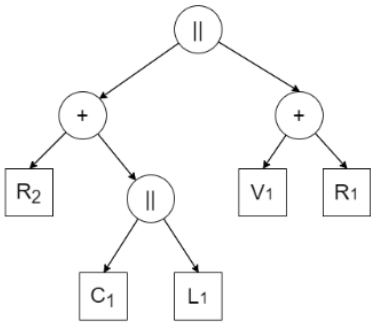


Fig. 1. Tree representation of the circuit on Fig. 2.

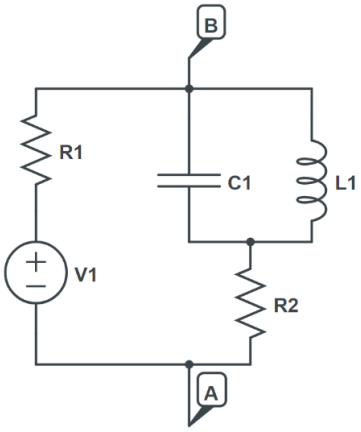


Fig. 2. Example Circuit Schema.

At the end of this section, we provided an example to illustrate how a circuit can be represented as a tree, as shown in Figures 1 and 2. In this example, the reference current direction is from point B to point A.

B. Implementation

In the preceding subsection, we outlined the principles of our approach which can be readily translated to a programming language that supports the object-oriented paradigm. The implementation revolves around two class hierarchies.

The foundation of our implementation is a CVC class, featuring methods like *current_at_voltage()* and *voltage_at_current()*, as well as methods for combination of two CVCs. In our implementation, the latter is done by overloading logical AND and OR operators.

To manage various circuit components, we've established a hierarchy of classes. This hierarchy adheres to the composite pattern [3], consisting of:

1. Abstract Base Class: This base class declares critical methods such as *calc_cvc(), apply_current(), apply_voltage(), in_series_with(), in_parallel_with()*, and more. It serves as the blueprint for all component classes.

2. Intermediate Abstract Classes: We've introduced an extra layer of abstract classes that categorize components into three groups:
    a. Real-Valued Components: These encompass components specified by real values, including resistors, capacitors, and inductors.
    b. Complex-Valued Components: This category encompasses components specified by complex values, such as impedance and ideal voltage/current sources.
    c. Composite Components: These are components specified by their child components, including series and parallel circuits. The logic within these classes revolves around managing the relationships between their constituents.
3. Concrete Classes: These classes represent concrete component types, encompassing both primitive and composite components. They implement the declared functions according to the unique logic associated with each component type. This ensures that specific component behaviors are accurately captured within the framework.

IV. Arbitrary Topologies

The methodology discussed in the previous section works well for circuits represented as trees of series and parallel connections. However, its applicability is constrained when dealing with more intricate topologies, such as bridge circuits, where components are interconnected in more complex ways. Consequently, its scope is somewhat limited.

To extend our method to arbitrary circuits with diverse topologies, we must undergo a fundamental shift in our approach. Our solution entails reverting to nodal analysis, a well-established technique in electrical engineering. In our new method, we view the circuit as a graph composed of branches connecting nodes, where nodes serve as points of junction for three or more branches. Each branch, in turn, comprises one or more components connected in series.

In nodal analysis, we construct a linear system by mandating that the net current flowing out of each node equals zero [2]. However, instead of solving this linear system for exact solutions, our approach leans toward numerical methods. We define our solution as a set of node voltages and quantify the quality of a solution by calculating the sum of squares of net currents over each node, serving as our loss function. The objective is to find a solution that minimizes this loss, ideally reducing it to zero.

$$L = \sum_{v \in V} I_v^2 \tag{5}$$

$$I_v = \sum_{e \in E} B_{ve} I_e \tag{6}$$

In the previous equations V is a set of nodes, E is a set of branches, B is an incidence matrix, and I_e is a current going through branch e. The latter is calculated using the method developed in the previous section.

Since currents depend linearly on node voltages, the loss function takes the form of a quadratic function of node voltages. Moreover, since the loss function is the sum of squares, it is bounded from below by zero, rendering it a convex function. This advantageous property simplifies our optimization problem, allowing us to leverage a broad range of well-established methods for convex optimization.

To iteratively progress toward superior solutions, we need to compute the gradient of the loss function concerning node voltages. Notably, we encounter the challenge of differentiating a real-valued function with respect to complex variables. The key lies in devising a mechanism to store these derivatives in a way that enables their backpropagation through computational graph.

If we denote real and imaginary parts by x and y, respectively, an effective strategy is to store the gradient as follows:

$$\nabla = \frac{\partial}{\partial x} - j\frac{\partial}{\partial y} \tag{7}$$

Here, j denotes the imaginary unit. This formulation allows us to apply the standard chain rule for differentiation, facilitating the propagation of gradients backward through the computational graph. Indeed, if we have two complex variables z and w such that w is a function of z, it can be shown that the following equation hold:

$$\nabla_z \propto \frac{\partial w}{\partial z} \nabla_w \tag{8}$$

As a result, we can efficiently compute the gradients of real-valued functions with respect to complex variables. For updates, we naturally use the complement of this gradient.

To compute gradients, we implement a compact reverse mode automatic differentiation module, conceptually similar to those commonly found in deep learning frameworks like PyTorch [4].

The algorithm for determining the node voltages is as follows:

1. Initialization: Begin by initializing all node voltages to zero. This serves as the initial solution, assuming the circuit contains no active components.
2. Forward Computation: Conduct a forward computation to calculate the loss associated with the current set of node voltages.
3. Check Loss Value: If the computed loss value meets the predetermined satisfactory criteria (very close to zero for an ideal solution), return the current set of node voltages as the solution.
4. Backward Computation: Perform a backward computation to calculate the gradients of the loss function with respect to the node voltages.

5. Update Node Voltages: Utilize the calculated gradients to update the node voltages and proceed to step 2 for another iteration.

Steps 2 to 5 are repeated until the algorithm terminates. Ideally, the loss should decrease with each iteration. At the end of this process, the correct node voltages are determined, enabling the subsequent calculation of current and voltage values for all components within the circuit. Concerning Step 5, many techniques could be used to perform an update including various variants of gradient descent.

While our approach offers a robust framework for circuit analysis, there are a few caveats to discuss:

- Reference Node: One node voltage within the circuit can always be set as a reference, typically fixed to zero volts. This reference node provides a point of comparison for other voltage values in the circuit.
- Ideal Voltage Sources: In instances where a branch comprises solely an ideal voltage source, a unique constraint is introduced. Such a source imposes a specific relationship between the voltages at its terminal nodes. Consequently, it becomes less preferable to consider these voltages independently. Additionally, since the branch contains only an ideal voltage source, the current flowing through it remains undefined. As a result, we treat this current as a new parameter to be determined within the analysis. Thus, overall, the count of parameters we optimize over remains unchanged.

V. Conclusion

In this paper, we have presented a powerful and elegant method for solving linear electric circuits programmatically. Our approach is grounded in the principle of treating various component types uniformly, achieved through a common interface implemented by each component. Moreover, the integration of automatic differentiation empowers us with efficient optimization capabilities, offering a versatile toolset for tackling a broader spectrum of circuit analysis challenges. Notably, this framework's adaptability extends to encompass intriguing problems, such as optimizing the resistance of a resistor R to maximize power dissipation—a task seamlessly accommodated within our methodology.

Through this work, we have laid the foundation for a flexible and efficient approach to circuit analysis, where complex electrical systems can be dissected, understood, and optimized with ease. This method opens the door to a range of possibilities for electrical engineers and students, making it a valuable addition to the field of electrical circuit analysis.

Future work could focus on developing a graphical user interface (GUI), such as a web app, to make the software more accessible to students and other users.

Appendix

To illustrate how to use our method programmatically, we present the following Python code snippet for a simple linear electric circuit. Notice that we inverted the ideal voltage source E1 using the unary operator ~ to adhere to the passive sign convention.

```
from CircuitSolver import *

r1 = Resistor('R1', 200)
r2 = Resistor('R2', 100)
r3 = Resistor('R3', 100)
r4 = Resistor('R4', 50)
r5 = Resistor('R5', 100)
e1 = IdealVoltageSource('E1', 1)
j1 = IdealCurrentSource('J1', 20, SIPrefix.Mili)
j2 = IdealCurrentSource('J2', 10, SIPrefix.Mili)

branches = [
    Branch(1, 4, [j1, r1]),
    Branch(1, 4, [r2]),
    Branch(1, 2, [r3]),
    Branch(2, 3, [r4, ~e1]),
    Branch(2, 3, [r5]),
    Branch(3, 4, [j2])
]

circuit = CircuitSolver(branches=branches)
circuit.solve()
print(f'Current at {r4.Label}: {r4.current} A')

# output: Current at R4: (0.013333333333350026+0j) A
```

Acknowledgement

The foundation of this work was laid by the author during the fourth year of his bachelor's studies as a side project, which was subsequently honed and refined for his bachelor's thesis. The comprehensive and generalized solution presented in this paper, leveraging automatic differentiation, was developed during August 2023. The source code accompanying this paper is available at [8].

References

[1] D. Mitić, "Elektrotehnika 1", Niš, 2007.
[2] D. Mitić, "Elektrotehnika 2", Niš, 2008.
[3] Gamma, Erich and Helm, Richard and Johnson, Ralph and Vlissides, John, "Design Pattern: Elements of Reusable Object-Oriented Software", USA, 1995.
[4] Andrej Karpathy, https://github.com/karpathy/micrograd
[5] Nagel, L. W. (1975). "SPICE2: A computer program for simulation of semiconductor circuits". EECS Department, University of California, Berkeley.
[6] Linear Technology Corporation. (2018). LTspice XVII and SwitchCap XVII User's Guide.
[7] Cadence Design Systems. (2021). PSpice User Guide.
[8] U. Stojković, https://github.com/uros7251

Strukture integrisanih kola na primeru SE555P preciznog tajmera

Miloš Mitrović

Sadržaj - Mnoga otkrića su svojevremeno promenila svet i način ljudskog života, ali otkriće koje je bez pogovora najviše uticalo na nas i iz korena promenilo naše viđenje sveta je otkriće tranzistora i razvoj planarnih tehnologija.

Uprkos ovome, većina ljudi nije upoznata sa procesom projektovanja integrisanih kola. Cilj ovog rada je da predstavi osnovne topografije nekih od najkorišćenijih komponenti u izradi integrisanih kola, a potom da, na osnovu objašnjenih topografija, prikaže njihovu primenu u realnom kolu, u ovom slučaju varijanti NE555 tajmera.

I. Standardna bipolarna tehnologija

Među prvim tehnologijama za izradu integrisanih kola razvijena je standardna bipolarna tehnologija. Prvenstveno razvijena za proizvodnju prostih logičkih kola, najveću primenu pronašla je u izradi analognih kola odličnih performansi. Ova tehnologija je optimizovana za proizvodnju NPN bipolarnih tranzistora, ali dozvoljava kreiranje i ostalih bitnih komponenti, kao što su PNP tranzistori, otpornici, kondenzatori i određene vrste dioda.

Kako bi bilo moguće predstaviti složena integrisana kola, neophodno je najpre objasniti topografiju osnovnih elemenata. Međutim, kako je ovo polje veomo obimno, u nastavku ovog rada biće predstavljene topografije samo onih struktura koje su prisutne u topografiji integrisanog kola od interesa.

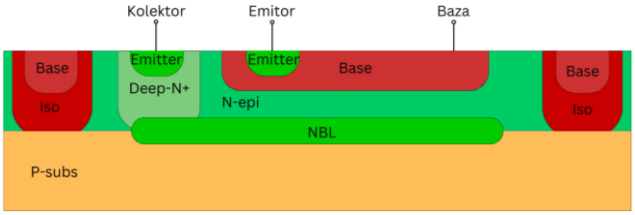
Sve ilustracije topografija u ovom radu prikazane su sa označenim slojevima. Ovi slojevi predstavljaju materijale i difuzije koje se koriste prilikom izrade integrisanih kola. Stoga, imena ovih slojeva ne predstavljaju imena samih elektroda komponenti. Zarad lakšeg razumevanja ovog koncepta, dat je poprečni presek strukture vertikalnog NPN bipolarnog tranzistora, slika 1.

A. Otpornici

Otpornike u standardnoj bipolarnoj tehnologiji moguće je izraditi na više načina, ali svi oni su bazirani na korišćenju difuzija baze i/ili emitora.

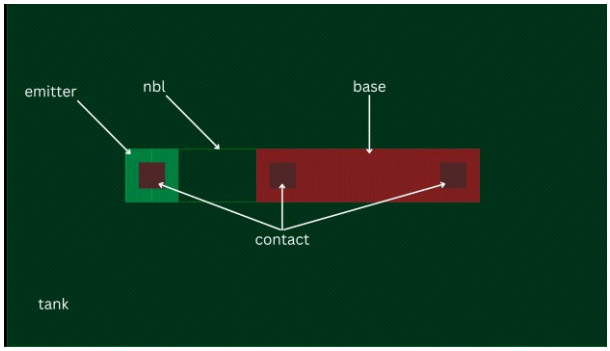
Najzastupljeniji tip čine otpornici od bazne difuzije (*Base Resistors*), slika 2. Ovi otpornici su nudili prosečnu slojnu otpornost od 150 Ω/□ [1].

M. Mitrović, Katedra za mikroelektroniku, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: milos.mitrovic@elfak.rs

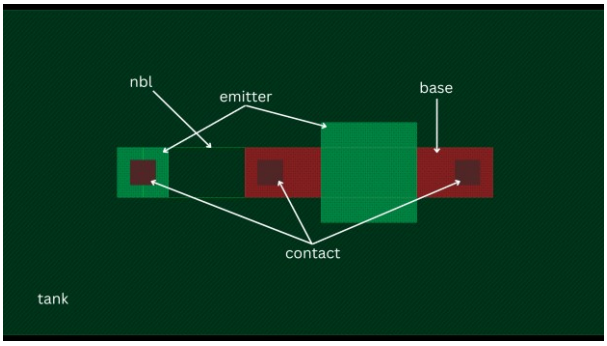


Slika 1. Poprečni presek strukture NPN tranzistora.

Još jedan tip od interesa u ovom radu jesu, priključeni otpornici (*Pinch Resistors)*. Ovi otpornici načinjeni su od kombinacije bazne i emitorske difuzije. Sama topografija je nalik standardnom baznom otporniku, ali je središnji deo ovog otpornika prekriven emitorskom difuzijom, što značajno povećava slojnu otpornost ove vrste otpornika time što smanjuje efektivnu dubinu bazne difuzije, slika 3. Prosečna slojna otpornost ovako konstruisanih otpornika bila je oko 3 kΩ/□ [1]. Struktura priključenog otpornika podseća na strukturu JFET-a.



Slika 2. Topografija baznog otpornika.

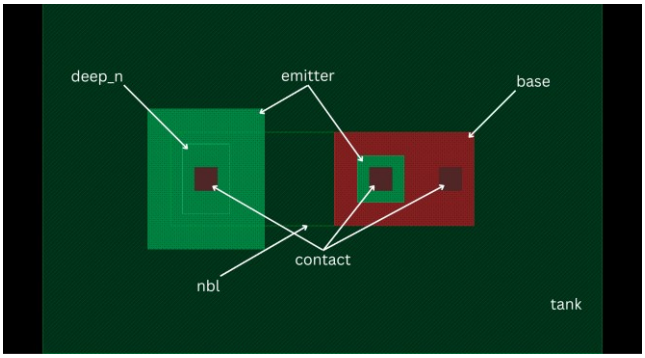


Slika 3. Topografija priključenog otpornika.

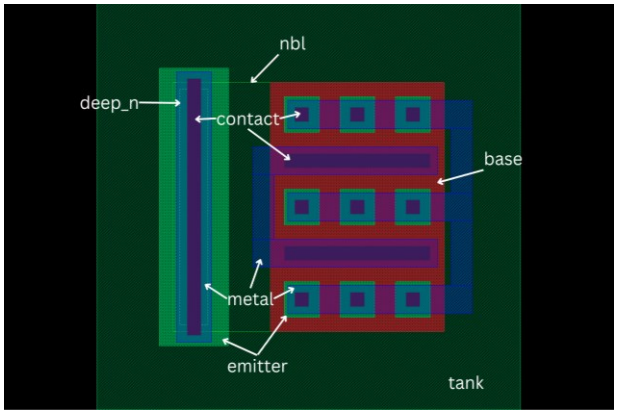
B. Tranzistori

Topografije tranzistora su raznolike, mada se razlikuju po izgledu i svrsi, ovi tranzistori su sačinjeni od istih slojeva (difuzija). Stoga, topografije prikazane u ovom odeljku mogu služiti za prepoznavanje gotovo svih vrsta tranzistora.

Topografija na slici 4, predstavlja vertikalni NPN tranzistor. Ovo je najosnovnija, a ujedno i najkorišćenija topografija bipolarnih tranzistora. Ovaj tranzistor dobio je naziv po smeru kretanja elektrona koji putuju od emitora kroz bazu do kolektora vertikalno.



Slika 4. Topografija lateralnog NPN tranzistora.

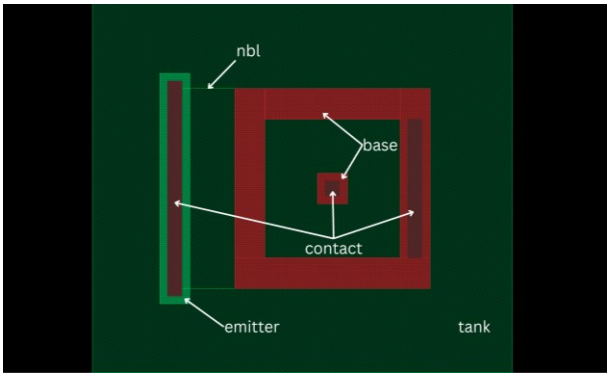


Slika 5. Topografija NPN tranzistora snage.

Na slici 5, moguće je videti topografiju češljastog NPN tranzistora snage (*Interdigitated-emitter Power Transistor)*. Slojevi koji se koriste u izradi tranzistora NPN snage su identični kao i za izradu standardnog vertikalnog NPN tranzistora, ali se strukture značajno razlikuju. Strukture tranzistora snage su optimizovane za protok znatno većih gustina struje. U prethodnim primerima sloj za metalizaciju bio je uklonjen zbog preglednosti, dok je u ovom primeru pogodno ostaviti ovaj sloj kako bi se videla povezanost kontakta emitora i baze.

Mada, slabijih performansi, lateralni PNP tranzistor često je korišćen u topografijama, pogotovo u izvorima konstantne struje (strujna ogledala). Ime ovog tranzistora, kao i kod vertikalnog NPN tranzistora, potiče od smera kretanja nosilaca naelektrisanja, u ovom slučaju šupljina, koje putuju horizontalno (lateralno) od emitora do kolektora. Kao što je to slučaj kod NPN tranzistora, postoje različiti stilovi projektovanja topografija PNP tranzistora.

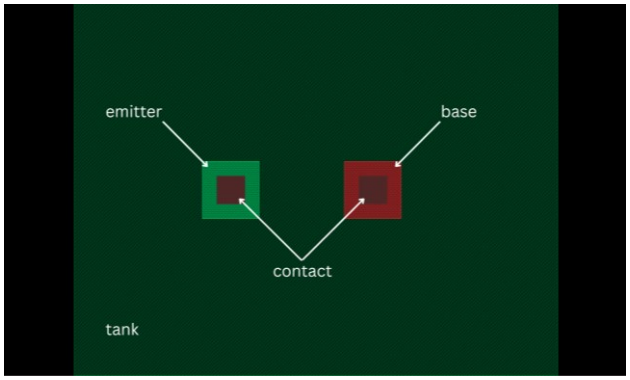
Najkorišćenija topografija PNP tranzistora je topografija sa emitorom kružnog poprečnog preseka, međutim, zarad potreba ovog rada, biće prikazana topografija PNP tranzistora sa emitorom kvadratnog oblika, slika 6. Emitor kružnog poprečnog preseka je uglavnom pogodniji zbog svog odnosa površine i širine. Međutim, pravougaone strukture je lakše verifikovati u procesu projektovanja, stoga se neretko primenjuju ukoliko specifikacije integrisanog kola to dozvoljavaju.



Slika 6. Topografija lateralnog PNP tranzistora.

Zbog toga što se u standardnoj bipolarnoj tehnologiji koristi epitaksijalni sloj N-tipa, a ne postoji mogućnost za epitaksijalni rast silicijuma P-tipa, u ovoj tehnologiji ne postoji vertikalni PNP tranzistor.

Umesto standardnog vertikalnog PNP tranzistora koristi se supstratski PNP tranzistor, koji nudi značajno bolje performanse od lateralnog PNP tranzistora, skoro približne vertikalnom NPN tranzistoru. Međutim, velika mana ovog dizajna je što kolektor ovog tranzistora predstavlja supstrat, po čemu je i dobio ime, zbog čega su kolektori svih supstratskih PNP tranzistora u kolu vezani za isti čvor, što je uglavnom masa. Ova činjenica značajno ograničava njihovu primenu. Topografija supstratskog PNP tranzistora data je na slici 7.



Slika 7. Topografija supstratskog PNP tranzistora.

II. OBRNUTI INŽENJERING SE555P INTEGRISANOG KOLA

Precizni tajmer NE555 predstavlja jedno od najkorišćenijih integrisanih kola na svetu, sa preko milijardu proizvedenih primeraka godišnje [2]. Ovo kolo našlo je svoju primenu u deliteljima učestanosti, detektorima nedostajućih impulsa, oscilatornim kolima i naravno tajmerima. Originalno NE555 integrisano kolo dizajnirano je od strane *Hans Camenzind*-a, 1971 godine, a nešto kasnije, 1972. godine, kompanija *Signetics*, za koju je Hans radio, počela je sa proizvodnjom.

Ubrzo su i ostale velike kompanije dobile prava za proizvodnju ovog integrisanog kola. Jedna od ovih kompanija jeste *Texas Instruments*. Upravo čip ovog proizvođača biće razmatran u ovom odeljku.

SE555P je identično kolu NE555, sa bitnom razlikom koju oznaka ,,S'' predstavlja, a to je dozvoljeni temperaturni opseg. U slučaju čipova sa oznakom ,,N'', ovaj opseg kreće se između 0 °C i +70 °C, što je pogodno za većinu komercijalnih upotreba. Međutim, postoje uslovi u kojima ovo nije dovoljno, kao na primer za potrebe vojne industrije, zbog čega postoje čipovi čiji je temperaturni opseg od -55 °C do +125 °C, a u slučaju 555 tajmera, oni nose oznaku ,,S''.

Poslednje slovo u imenu označava vrstu kućišta. U ovom slučaju ,,P'' predstavlja plastično DIP8 (*Dual-In-line Package*) kućište.

A. Topografija SE555P integrisanog kola

Uz pomoć osnovnih topografija prezentovanih u odeljku I, moguće je obrnutim inženjeringom (*Reverse Engineering*) topografije čipa sa slike 9 dobiti električnu šemu ovog integrisanog kola, slika 10.

Na slici 9, plavim pravougaonicima označena su tri supstratska PNP tranzistora, dok je ostale lateralne PNP tranzistore moguće lako uočiti zato što je velika površina oko emitora i kolektora ovih tranzistora prekrivena slojem metala. Ova struktura naziva se *Field Plate* i služi za zaštitu od formiranja kanala parazitnog PMOS tranzistora. Takođe, lako je uočiti i NPN tranzistore snage na gornjem delu slike 9.

Tri otpornika, uokvirena zelenom bojom, jesu upravo tri otpornika vrednosti 5kΩ, po kojima je ovaj tajmer i dobio ime. Oni su uokvireni zato što je njihova otpornost poznata i na osnovu njihovih dimenzija moguće je odrediti gotovo tačnu slojnu otpornost bazne difuzije korišćene u tehnološkom procesu u kome je izrađeno ovo kolo.

Kako važi relacija:

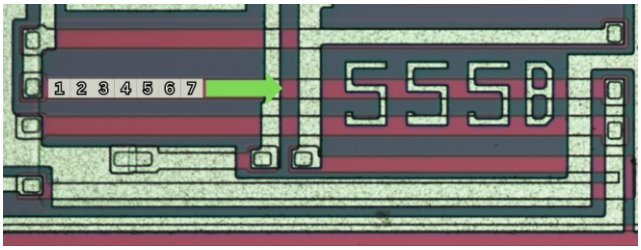
$$R = R_S \cdot \frac{l}{w}, \tag{1}$$

gde je R otpornost otpornika, R_S slojna otpornost materijala i jednaka je količniku specifične električne otpornosti i debljine materijala, l dužina otpornika, a w širina otpornika.

Količnik dužine i širine otpornika, iako bezjedinična veličina, izražava se jedinicom kvadrat (□). Kao što se može videti, odnos dužine i širine otpornika predstavlja broj kvadrata, N. Stoga se prethodna relacija može zapisati u sledećem obliku:

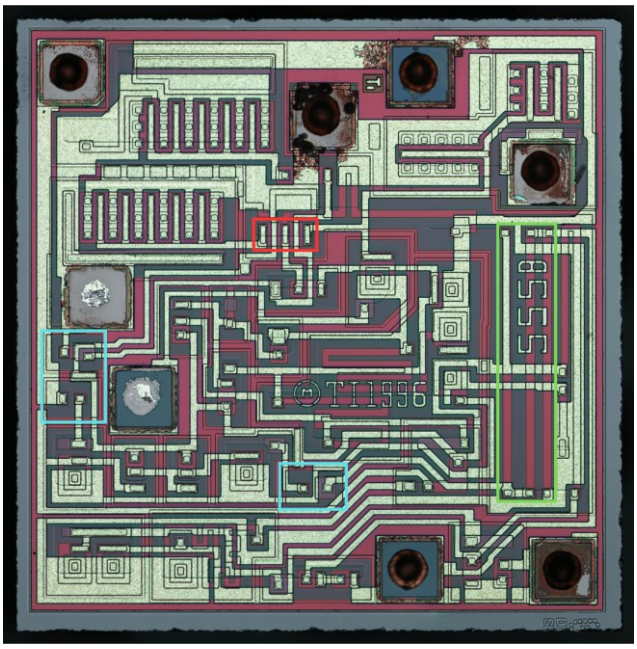
$$R_S = \frac{R}{N} \tag{2}$$

Iz ove relacije, brojanjem kvadrata kao što je prikazano na slici 8, moguće je odrediti slojnu otpornost koja je u ovom slučaju približno jednaka datoj prosečnoj vrednosti iz odeljka I-A.

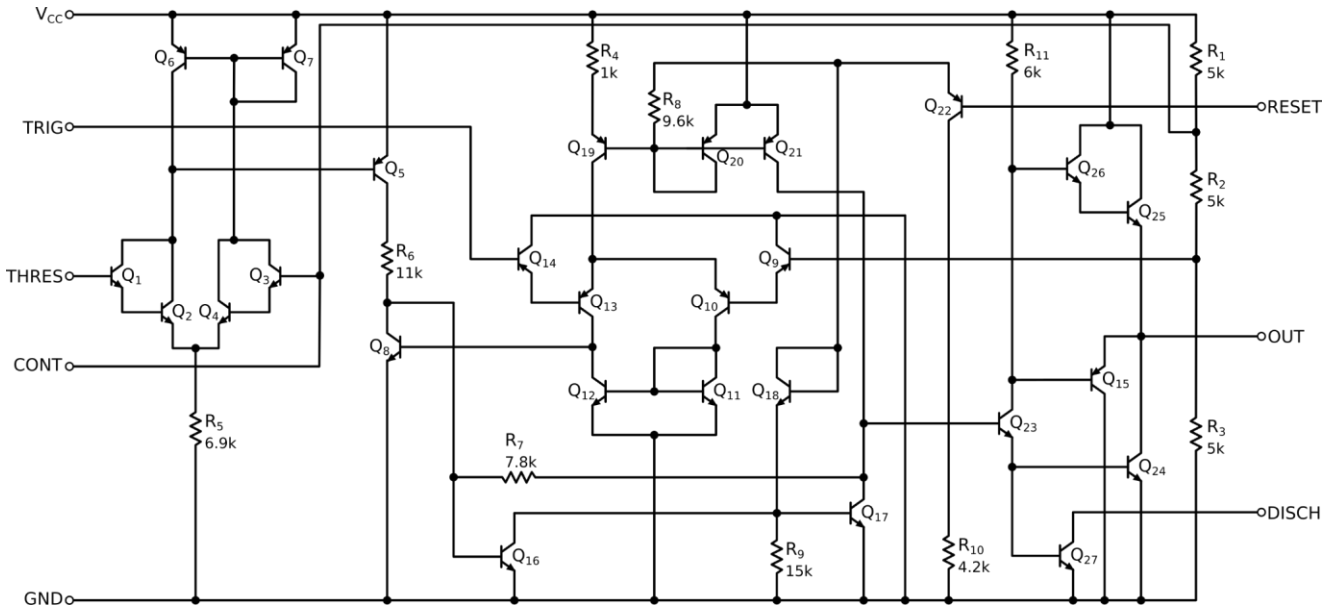


Slika 8. Brojanje kvadrata.

Prilikom računanja otpornosti kontakta, u ovom slučaju uzeta je vrednost jednaka polovini kvadrata. Ovo takođe važi i za kvadrate koji se nalaze na uglovima serpentinastih otpornika. Značajno je napomenuti da različiti autori koriste različite aproksimacije ovih vrednosti.



Slika 9. Topografija SE555P integrisanog kola. [3]



Slika. 10. Električna šema SE555P integrisanog kola.

Struktura uokvirena crvenom bojom, na slici 9, svojim izgledom podseća na bazni otpornik. Međutim, kako je odnos širine i dužine ove strukture znatno manji od odnosa širine i dužine standardnog otpornika, otpornost ove strukture je dosta manja. S obzirom da standardna bipolarna tehnologija koristi samo jedan sloj metala, ove strukture koriste se kao kratkospojnici (*Jumpers*). Ovo značajno doprinosi efikasnosti prilikom rutiranja metalnih linija u procesu projektovanja topografija, uprkos ograničenjima izazvanim postojanjem samo jednog sloja metala.

III. Zaključak

Projektovanje topografija je jako složena, ali neophodna grana elektronike/mikroelektronike, što naročito važi za analogna kola čiji je dizajn teže automatizovati.

Jedan od najboljih načina da se nauče osnove projektovanja topografija, je izvršiti obrnuti inženjering već postojećih čipova. Na ovaj način moguće je videti, na praktičnom primeru, upotrebu određenih projektrantskih pravila i omogućiti bolje razumevanje istih. Takođe, obrnuti inženjering koristan je i u svrhu sticanja iskustva o upotrebi različitih struktura, na primer gde i kada primeniti diode za zaštitu od elektrostatičkog pražnjenja.

U ovom radu su predstavljeni opisi topografija nekih od najčešćih struktura korišćenih u projektovanju topografija. Korišćenjem ovih topografija moguć je obrnuti inženjering velikog broja integrisanih kola.

Literatura

[1] A. Hastings, *The Art of Analog Layout*, 1st ed. Prentice Hall, 2001.

[2] D. Lowe, *Electronics All-in-One For Dummies*, 2nd ed. Wiley, 2017.

[3] Tehnička specifikacija za kolo SE555. Dostupno na: https://zeptobars.com/en/read/Ti-555-SE555P