IEEESTEC

12th 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
Serbian Academy of Sciences and Arts (SASA), Branch Niš
Ministry of Education, Science and Technological Development

ISBN: 978-86-6125-215-0

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
Dušan Vučković

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

Secretary:
Jana Vračar
Aleksandra Stojković
Milan Stojanović
Nikola Mitrović
Jovana Veselinović
Željko Radojković
Anđela Stojiljković
Kristina Avramović
Nikola Radisavljević
Milan Ivković
Marija Živković
Nina Dimitrijević
Anđela Kocić
Sandra Veljković
Milivoje Mihajlović
Ana Sentov
Stefan Popović
Stefana Zogović
Nevena Stanković

Technical Editors:
Miloš Marjanović
Jana Vračar
Danijel Danković
Emilija Živanović

CIP- Каталогизација у публикацији
Народна библиотека Србије

621.3(082)
004(082)

IEEESTEC Student Project Conference (12 ; 2019 ; Niš)
IEEESTEC 12th Student Project Conference, [Niš, 2019] / organized by EESTEC LC, Niš ... [et al.] ; [editors Ninoslav Stojadinović ... [et al.]]. - Niš : Faculty of Electronic Engineering, 2019 (Niš : CopyShop). - 346 str. : ilustr. ; 30 cm

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

ISBN 978-86-6125-215-0

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

COBISS.SR-ID 280825868

Printed by: „CopyShop“, Niš, 2019.
Circulation: 100 copies

Reviewers:

1. Nikola Mitrović
2. Miloš Đorđević
3. Sanja Aleksić
4. Dejan Anđelković
5. Mirjana Perić
6. Milan Stojanović
7. Milivoje Mihajlović
8. Vojkan Davidović
9. Emilija Živanović
10. Miloš Marjanović
11. Ana Vučković
12. Marko Dimitrijević
13. Dragana Živaljević
14. Boban Veselić
15. Ivan Anastasijević
16. Tatjana Nikolić
17. Zlatica Marinković
18. Goran Nikolić
19. Igor Stojanović
20. Nenad Milošević
21. Marija Milijić
22. Branislav Ranđelović
23. Aleksandra Stojković
24. Milutin Petronijević
25. Vladimir Mitić
26. Vladimir Sibinović
27. Stefan Ilić
28. Stefana Zogović
29. Vladimir Ćirić
30. Miona Andrejević Stošović
31. Danijel Danković
32. Vesna Paunović
33. Sandra Veljković
34. Nikola Davidović
35. Anđela Kocić
36. Biljana Stošić
37. Nenad Petrović
38. Tijana Dimitrijević
39. Miodrag Spasić
40. Saša Nikolić
41. Milan Simić
42. Milica Jovanović
43. Miloš Bogdanović
44. Goran Miljković
45. Dragana Jovanović
46. Stevica Cvetković
47. Aneta Prijić
48. Igor Jovanović
49. Filip Filipović
50. Bojan Banković
51. Nikola Stojanović
52. Marija Živković
53. Miljana Milić
54. Ivica Marković
55. Jana Vračar

Table of contents:

Automatsko podizanje rampe na osnovu očitane registarske tablice 1
Smart Gate – uređaj za ograničavanje pristupa prostorijama 5
IoT sistem za akviziciju podataka, merenje temperature, vlažnosti vazduha i intenziteta osvetljenosti 9
Realizacija štampane ploče za projektovani taksimetar 15
Operation of Dynamic Emulator of Multi-Motor Drive with Different Dynamic Emulation Methods 19
LED pahulja kontrolisana Android aplikacijom 25
Analiza I2C protokola korišćenjem logičkog analizatora 33
Primena linearne optimizacije za podršku autonomnih vozila u pametnim gradovima 37
Sistem za detekciju i razmenu vremenskih uslova na putu realizovan korišćenjem Arduino Uno... 41
Pametna kapija 45
Sistem za upravljanje javnim garažnim prostorima 49
Praktična realizacija elektronskog kola trofaznog tranzistorskog invertora 53
Praktična realizacija ispitne stanice za upravljanje trofaznim asinhronim motorom ... 57
Prototip pomerača digitalnog naponskog nivoa 63
Automatizacija mini elektrana 67
Automatizacija staklenika 71
Mehanizam za postizanje bezbednosti objekata 73
Podsetnik za redovnu upotrebu medicinske terapije za stare i senilne osobe 77
Kreiranje univerzalne verifikacione komponente za APB protokol po UVMe metodologiji 79
Pametan novčanik 85
Bežična prezentacija merenja pomoću Android osciloskopa 89
Registracija prometa putnika u vozilima javnog prevoza 93
Teraino 97
Rešavanje lavirinta pomoću infrared senzora na Lego platformi 101
Projekat Centra za promociju nauke “Pretočimo znanje u praktičan rad” ... 105
Projektovanje i realizacija CNC mašine za iscrtavanje 109
Primena OLED komponenti kao izvora svetlosti 113
Alarmni sistem protiv krađe košnica 117
Automatizacija domaćinstva 119
E - papir info panel sa malom potrošnjom energije i prikazom vremenske prognoze 123
Testovi performansi u web aplikacijama 131
Maketa naponskog razdelnika sa potenciometrom 135
Meteorološka stanica realizovana korišćenjem Internet of things (IoT) tehnologije 137
API Benchmark 141
Laserski daljinomer sa opcionim merenjem nivoa nagiba i temperature 143
Take Care – Sistem pametnih uređaja za brigu o zdravlju starijih osoba 147
Modularne laboratorijske vežbe -ModuleClick- 151
Primena stabala odlučivanja korišćenjem Data Mining analize podataka o padavinama primenom C# ... 155
Cane for Visually Impaired People -The Stevie Wonder Model- 163
MK BOX 165
Be Focused! Box 167
Ulazni deo optičkog prijemnika sa visokom impedansom 169
Planiranje kretanja modela Quadrotora u ROS okruženju 173
Maketa naponskog razdelnika 177
Realizacija modela IoT sistema primenom Raspberry Pi 181
Driver for laser with low input impedance 185
Ispitivanje pouzdanosti *Littelfuse* gasnog odvodnika prenapona 189
Realizacija brojača glasova pomoću Arduino Uno okruženja 193
Ispitivanje dinamičkih karakteristika optokaplera 197
Sixth Sense – uređaj za slepe i slabovide osobe zasnovan na Doplerovom efektu 201
Projektovanje okruženja za testiranje komponenata 205
Studije i život na Univerzitetu Balearskih ostrva, Palma de Majorka 209
Razvoj interaktivne aplikacije „blok igre“ u okviru PSoC razvojnog okruženja 213
Uređaj za demonstraciju Lencovog zakona 219
Realizacija gitarskog vakuumskog pretpojačala 221
Projektovanje verifikacione komponente za čip HI-6130 korišćenjem UVM metodologije 227
Kvantni tranzistori sa efektom polja- HEMT(High Electron Mobility Transistor) 233
Izrada sustava za otvaranje elektroničke brave putem RFID senzora 235
Uticaj metode izračunavanja korelacije na performanse algoritama mašinskog učenja 239
Fazna kontrola pomoću TRIAC-a 243

Digitalna mapa kulturno-istorijskih znamenja grada Niša 247
Analogna distorzija V-2 251
Servosistemi za laboratorijske makete bazirani na Arduino Due mikrokontroleru 257
Nova metoda za automatizaciju ispitivanja kondenzatora upotrebom virtuelne instrumentacije 261
Upravljanje asinhronim motorima u prehrambenoj industriji 265
Primena programskog paketa PSCAD za analizu rada odvodnika prenapona 269
Online aplikacija za prikaz aktivacionog sloga i asemblerskog koda C programa 273
Projektovanje i realizacija digitalnog sata WORD CLOCK 279
Razvoj aplikacije za praćenje kretanja simbola na displeju u okviru PSoC razvojnog okruženja 285
Sistem za praćenje temperature preko mobilne mreže baziran na Arduino platformi 291
Projektovanje i realizacija elektronske brave za otključavanje vrata pomoću tastature 297
Govorni časovnik 301
Softverska emulacija UART komunikacije 305
RFID čitač kartica za evidenciju rada zaposlenih 309
Elektronski klavir realizovan pomoću integrisanog kola NE555 313
Realizacija kalkulatora pomoću Arduina 317
Projektovanje i implementacija SMTP protokola sa dokazom uloženog rada klijenta 321
Realizacija digitalnog Real-Time servosistema na Arduino Due mikrokontroleru 325
Sistem za relejnu zaštitu zasnovan na Arduino Uno u okviru podacima-vođene arhitekture pametne mreže 331
Primena programa MATLAB za simulaciju rada višenivovskih pretvarača za obnovljive izvore energije 335
Ispitivanje karakteristika optičkog senzora TCRT5000 339
Primena Arduina i ultrazvučnog senzora u izradi radarskog sistema 343

Automatsko podizanje rampe na osnovu očitane registarske tablice

Jelena Zdravković, Marko Miljković

Sadržaj - U ovom radu je prikazan sistem za brži i jednostavniji pristup praking mestu. Korišćenjem sistema izbegava se nepotrebno zadržavanje prilikom ulaska na parking. Takođe, nepotrebne su i dodatne kartice za ulazak na parking.

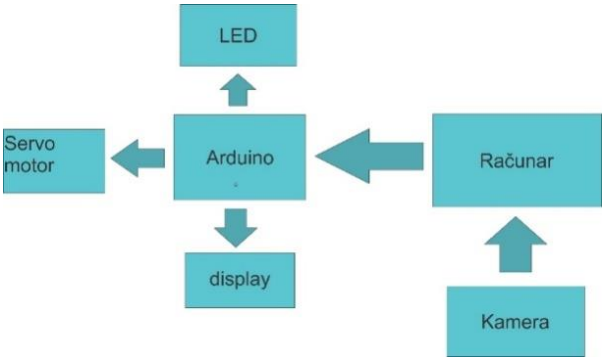
I. UVOD

Parkiranje u gradu je noćna mora svakog vozača. Prilikom ulaska na prostor za parkiranje prvo što vidimo je cena po započetom satu, a odmah zatim „U slučaju gubitka kartice za parkiranje korisnik snosi troškove vađenja nove.”. Na nekim parking mestima ne postoji kartica već se plaćanje obavlja slanjem SMS poruke, što obično traje par minuta, a nervozni vozači trube za to vreme. Korišćenjem ovog sistema ulazak se obavlja za 10-ak sekundi. Takođe, uvek imate rezervisano parking mesto, a izbegnuta je i potreba neprekidnog dežurstva od 24h tokom cele godine. Takođe, nema ni troškova koji prate izadavanje kartice za parking.

II. PRINCIP RADA

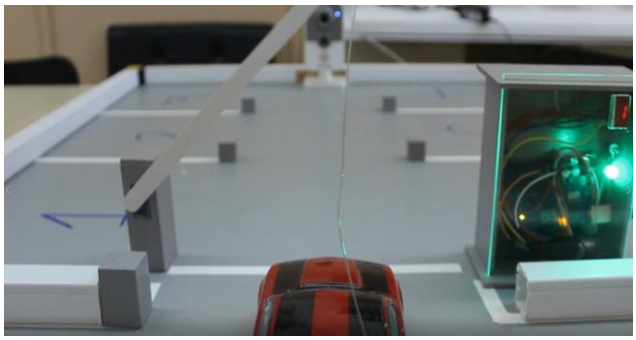
Da bi sistem funkcionisao neophodno je postaviti kameru usmerenu ka ulazu kako bi mogla da detektuje objekat tj. automobil. Nakon toga se slika objekta prenosi do računara gde odgovarajući program izvrši detekciju tablice a njen sadržaj pretvori u tekst. Zatim se tako dobijeni tekst upoređuje sa registarskim tablicama kojima je pristup parkingu dozvoljen, a koje se nalaze u bazi podataka. Ukoliko se tablica nalazi u bazi, šalje se signal Arduinu koji zatim podiže rampu dok istovremeno na displeju ispisuje broj parking mesta koje je rezervisano za auto sa tom tablicom. Svaki automobil ima rezervisano parking mesto na osnovu tablice. Mesto se rezerviše prilikom unosa novog korisnika tako što korisnik prilikom zakupa odabere jedno od slobodnih parking mesta. Prednost izbora je što vozač ne mora da pamti broj parking mesta. Nakon par sekundi rampa se spušta.

Jelena Zdravković i Marko Miljković su učenici četvrte godine Elektrotehničke škole „Mija Stanimirović”, Bulevar Svetog cara Konstantina 82-84, 18000 Niš, Srbija, E-mail: jeciik9@gmail.com, marko01miljkovic@gmail.com

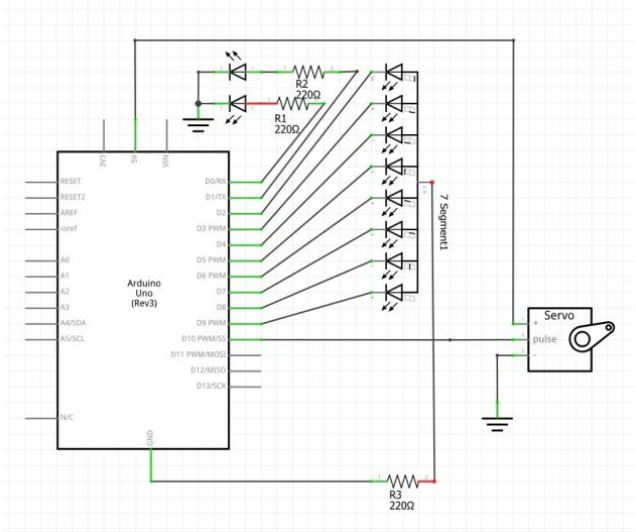


Slika 1. Princip rada sistema.

III. KORIŠĆENI HARDVER



Slika 2. Maketa sistema.



Slika 3. Električna šema sistema.

Za izradu makete sistema korišćene su sledeće komponente:

-Arduino Uno [1] - Mikrokontroler na štampanoj ploči sa radnim naponom od 5 V. Potrebno napajanje je između 7 i 12 V. Arduino Uno raspolaže sa 14 digitalnih I/O pinova, 6 pinova mogu da budu analogni. Brzina radnog takta je 16 MHz.



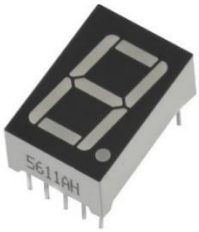
Slika 4. *Arduino Uno.*

-Servo motor SG90 [2] - Radni napon servo motora je od 4.8 do 6 V. U sistemu se koristi za podizanje rampe. Rampa je podignuta kada je motor pomeren za 90°.



Slika 5. Servo motor SG90.

- Sedmosegmentni displej [3] - Sedam LE dioda raspoređenih u obliku broja osam. Služi za ispisivanje rednog broja parking mesta koje je korisnik zakupio.



Slika 6. Sedmosegmentni displej.

- LE dioda [4] - Svetleća dioda. U sistemu su korišćene dve LE diode, zelena i crvena, 5 mm. Struja kroz diode je 22 mA.



Slika 7. LE dioda.

- Kamera [5] - 3D DNR, infracrvena svetlost koja se koristi u noćnim uslovima. Kamera je u osnovi namenjena za video nadzor. Detektuje pokrete i takođe ima ugrađen mikrofon. Rezolucija kamere je 1080p.



Slika 8. IR Cube Network Camera.

IV. SOFTVER I DIZAJN SISTEMA

Za izradu sistema korišćen je Visual Studio [6] i Arduino IDE [7].

Za izradu aplikacije korišćen je C#.

Funkcija ProcessFrame služi za prenos slike sa kamere:

```
private void ProcessFrame(object sender, EventArgs e)
    {
        if (stream != null && stream.Ptr != IntPtr.Zero)
        {
            stream.Retrieve(frame, 0);
            videostream.Image = frame.Bitmap;
        }
    }
```

Deo koda koji služi kako bi isekli tablicu sa slike:

```
    //Iseci sliku tablice
    private Image cropImage(Image img, Rectangle
cropArea)
    {
        var bmpImage = new Bitmap(img);
        return bmpImage.Clone(cropArea,
bmpImage.PixelFormat);
    }
```

Deo koda kojim se šalje naredba Arduinu za podizanje rampe:

```
    if (port.IsOpen)
  {
     string val = tablicatxt.Text.Trim();
     if (whitelist.Items.Contains(val))
        {
         port.WriteLine("H");
         }
}
```

Ukoliko je poslata naredba „H“ koja označava da je ulazak odobren.

Arduino kod za podizanje i spuštanje rampe:

```
Case 'H': {if(a>=0){
          Num_Write(a);
          digitalWrite(11, LOW);
          digitalWrite(12, HIGH);
        for(int i=0; i<90; i++)
        {
          myservo.write(i);
          delay(30);
        }
        delay(6000);
        digitalWrite(11, HIGH);
        digitalWrite(12, LOW);
         for(int i=90; i>=0; i--){
          myservo.write(i);
      delay(30);}
          Num_Write(7);}
        break;}
```

Num_Write služi kako bi se na displeju ispisao broj parking mesta, a zatim se pomoću for petlje podiže rampa za 90°. Nakon 6 sekundi rampa se ponovo vraća u početni položaj. A funkcija *Num_Write* ispisuje ponovo „0" na displeju.

Na slici 9 prikazan je dizajn aplikacije sistema.

Dizajn se sastoji iz dela za konekciju sa kamerom i Arduinom, dela za kontrolu rampe, prozora na kome se emituje video sa kamere, tabele sa svim tablicama svih automobila koji su pokušali da udju na parking kao i tabela sa tablicama kojima je pristup odobren.

Ukoliko se radi o novom korisniku, tj. korisniku koji je uplatio svoje parking mesto i prvi put dolazi na parking, njegovi podaci se dodaju u bazu klikom na "Dodaj na belu listu" gde se pamti podatak o registarskom broju kao i rednom broju parking mesta koje je korisnik izabrao.

Ukoliko korisnik više ne želi da koristi usluge parking površine njegova tablica se uklanja klikom na „Ukloni sa liste".



Slika 9. Dizajn sistema.

V. PREDNOSTI SISTEMA

Prednosti sistema su što omogućava ulazak i izlazak sa parkinga bez zadržavanja. Sistem se može primenjivati na bilo kom ograđenom parkingu (javni parking ili parking u okviru kompleksa zgrada). Unos novih korisnika je jednostavan i brz, obavlja se skeniranjem tablice i ne postoji mogućnost greške. Korisnik ne može doći u situaciju da mu parking nije dostupan jer trenutno nema kod sebe platnu karticu ili mobilni telefon kojim bi platio uslugu parkiranja. Ne postoji mogućnost da vam neko drugi zauzme parking mesto jer se prilikom registracije korisnika rezerviše tačno određeno parking mesto. Ako više osoba koristi isti auto ne moraju da pamte gde je njihovo parking mesto. Sistem radi u svim vremenskim uslovima. Implementacija i održavanje sistema je jednostavno. Sistem je aktivan 24 časa dnevno što predstavlja najbitniju prednost ovog sistema.

VI. UNAPREĐIVANJE SISTEMA

Za unapređivanje sistema postoji dosta mogućnosti. Jedna od njih je dodavanje mesečne, polugodišnje ili godišnje pretplate koja će važiti za sve parking površine koje su povezane u jedinstveni sistem.

Osim vremenske zaštite od prevremenog spuštanja rampe postoji mogućnost dodatnog obezbeđivanja i optičkom zaštitom tj, senzorom. Senzor bi služio da proveri da li je auto prošao pre nego što se rampa spusti.

Može se omogućiti pretplatniku da koristi svoje parking mesto za različite automobile. Ovo je posebno korisno za firme koje raspolažu sa više vozila ali im treba samo jedno parking mesto.

ZAHVALNICA

Veoma smo zahvalni Elektrotehničkoj školi „Mija Stanimirović" koja nam je obezbedila opremu i prostor za rad.

Takođe, želimo da se zahvalimo našem mentoru Lučić Goranu, nastavniku u ETŠ „Mija Stanimirović".

LITERATURA

[1] Bert van Dam, Arduino Uno, Agencija Eho, Rozendal, 2016.
[2] TowerPro SG90 - Micro Servo, https://datasheetspdf.com/pdf/791970/TowerPro/SG90/1
[3] LED7-SegmentDisplay, http://www.xlitx.com/datasheet/5611AH.pdf
[4] https://www.nteinc.com/specs/3000to3099/pdf/nte3024.pdf
[5] http://www.networkcameracritic.com/?p=2236
[6] https://visualstudio.microsoft.com/ -Visual Studio, link
[7] https://www.arduino.cc/en/main/software

Smart Gate – uređaj za ograničavanje pristupa prostorijama

Veljko Milanov, Milan Milojković, Nenad Pavlović, Đorđe Manić, Nemanja Mančić, Bojan Ćirić

Sadržaj – U ovom radu opisan je uređaj za ograničavanje pristupa prostorijama.

I. UVOD

Uređaj "Smart Gate" je nastao iz potrebe naše škole da obezbedi ograničen pristup nekim prostorijama ili radionicama. Da bi se plan i program svih smerova u školi odvijao kako treba, škola poseduje veliku selekciju mašina, alata, računara i nastavnog materijala za stručne predmete. Kada su u pitanju radionice u kojima se nalaze ozbiljne, pa čak i opasne, mašine, brava koja se zaključava običnim ključem nije dovoljno bezbedna.

Iz ovakve problematike potiče ideja da se napravi pametna brava. Ona će omogućiti kontrolisani pristup datim učionicama i radionicama, kao i vođenje evidencije ko sve pristupa određenoj prostoriji.

Dalja implementacija može da bude u vidu vođenja evidencije o dolasku učenika i nastavnika u školu. Na taj način povećavamo bezbednost u školi i efikasnost rada nastavnika. Može se implementirati moduo koji će roditelje obaveštavati o dolasku njihove dece u školu putem SMS-a.

Glavna komponenta ovog uređaja je RF skener kojim se očitava kod sa kartice, tako što očitava frekvencije od 13,56 MHz koje ta kartica emituje. Određena kartica odgovara jednom, i samo jedom korisniku u bazi podataka. Tako, kada RF skener očita karticu i proveri u bazi podataka, tačno zna kome pripada data kartica.

Softverski deo uređaja je izveden u obliku veb aplikacije. To je grafički interfejs koji prikazuje korisniku da li mu je omogućen pristup ili ne. S obzirom da je u pitanju veb aplikacija, moguće postaviti bilo koji uređaj koji ima pristup mreži kao displej. Pristup ovoj veb aplikaciji omogućava se postavljanjem različitih administratorskih privilegija, u zavisnosti od potreba.

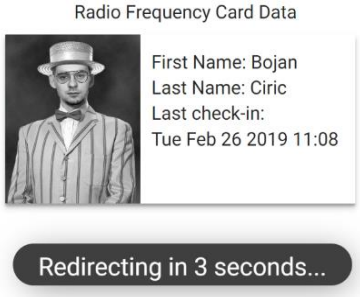
II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

A.Metod rada uređaja

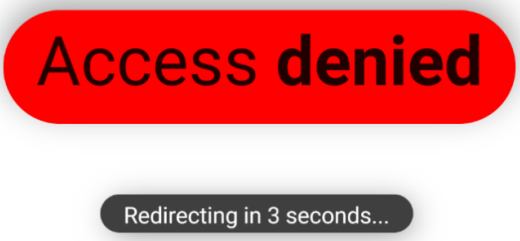
U početku, kontroler koji se koristi za izradu uređaja se nalazi u stanju mirovanja (u ovom stanju, kontroler čeka da se očita kartica). Za to vreme veb aplikacija čeka podatke od servera. Kada RF skener očita karticu, podaci o kartici se šalju na kontroler.

Kada kontroler primi informacije od strane RF skenera, uspostavlja vezu sa serverom na kome se nalazi baza. U bazi se upisuje vreme i datum kada je pokušan pristup, i identifikacioni broj kartice koji je pokušao da pristupi prostoriji. Na serveru se zatim proverava da li je dozvoljen pristup datom korisniku. Ako jeste, informacije o korisniku se šalju veb aplikaciji. Tada veb aplikacija prikazuje informacije o korisniku (slika 1). Takođe, kontroleru se šalje jedinica (1). Kontroler tada zna da treba da otključa bravu.

Ako pristup nije dozvoljen datom korisniku, onda se veb aplikaciji šalje informacija o tome da je pristup nije dodeljen i veb aplikacija onda prikazuje na ekran (slika 2). Takože, kontroleru se šalje nula (0). Kontroler tada zna da ne treba da otključa bravu.



Slika 1. Prikaz na ekranu kada je dodeljen pristup korisniku.



Slika 2. Prikaz na ekranu kada nije dodeljen pristup korisniku.

V. Milanov, M. Milojković, N. Pavlović, Đ. Manić, N. Mančić su učenici Tehničke škole u Pirotu. Takovska 22, 18300 Pirot, E-mail: veljkomilanov28@gmail.com, djordjem01@hotmail.com, milojkovic.milan01@gmail.com

B. Komponente

Na serveru se nalazi i baza podataka gde se nalaze podaci o korisnicima i njihovim interakcijama sa skenerom. Skener takođe ima i napajanje.

I2C [1] – se koristi za povezivanje komponenata koje su povezane kablom. Jednostavnost i fleksibilnos su ključne karakteristike koje čine ovaj interfejs korisnim mnogim aplikacijama.

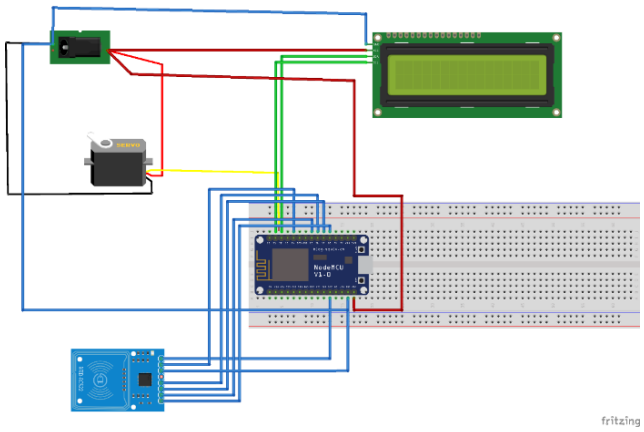
RFID-RC522 – RFID [2] predstavlja akronim termina radio-frekventna identifikacija. RFID koristi elektromagnetno polje za tranfer podataka na kratkim udaljenostima. RFID je koristan da identifikuje ljude, pravi transakcije, i tako dalje. RFID sistem koristi:

- Tagove vezane za predmet koji se identifikuje
- Dvosmerni radio predajnik-prijemnik, čitač koji šalje signal tagu i čita njegov odgovor

ESP8266[3] - WiFi modul - ESP8266 je samostalan SOC (sistem na čipu) sa integrisanim TCP/IP protokolom koji omogućava pristup bilo kom mikrokontroleru na Vašu WiFi mrežu. Koristili smo NodeMCU na 80 MHz.

Displej 20x4 – Koristi se lokalno za prikaz podataka ako ne postoji mogućnost montiranja grafičkog displeja.

C. Šema sistema



Slika 3. Način povezivanja komponenti.

Na ovoj šemi je prikazan servo motor. Neće koristiti pri normalnom radu uređaja, on na šemi predstavlja bravu koja se otključava i zaključava.

Takođe, na ovoj šemi je iskorišćen LCD displej za prikaz podataka, kao jednostavnija varijanta rada uređaja. Umesto tog LCD displeja može da se koristi i displej koji je prethodno opisan.

D. Bitni delovi koda

```
$(document).ready(function () {
  function DoAjax() {
    $.ajax({
      type: 'GET',
      url: "***",
      data: $(this).serialize(),
      dataType: 'text',
      success: function (data) {
        console.log('data response: ' +
data);
        console.log('data length: ' +
data.length);

        if (data == 0) {
          // nista
        }
        else if (data == 1) {
          setTimeout(function () {

document.getElementById('tspLogo').clas
sList.remove('zoomIn');
document.getElementById('rfIDContainer'
).classList.remove('zoomIn');
document.getElementById('tspLogo').clas
sList.remove('zoomOut');
document.getElementById('rfIDContainer'
).classList.remove('zoomOut');
          }, 1000) &&
window.location.replace("./accessDenied
.html");
        }
        else {
          var podaciObject =
JSON.parse(data);
          podaciObject[0].img =
"http://smartschool.cirasoft.net/image.
php?file=" + podaciObject[0].img;
          var podaciObjectStringified =
JSON.stringify(podaciObject);
          localStorage['jsoncache'] =
podaciObjectStringified;
          // data za primer:
[{"fname":"Bojan","lname":"Ciric","img"
:"01.png","laset_checkin":"2018-10-25
09:24:42"}]
window.location.replace("./accessGrante
d.html");
        }
      },
      complete: function (data) {
        // Schedule the next
        setTimeout(DoAjax, 1000);
      },
      error: function (req, response) {
        console.log("Error: " +
response);
      }
    });
  }
  setTimeout(DoAjax, 1000);
});
```

III. REZULTAT

Rezultat našeg rada je dobijanje funkcionalnog sistema koji nam omogućava kontrolu i evidenciju pristupa objektima ili prostorijama gde je uređaj montiran. Korišćenjem ovog uređaja, u školi se postiže veća sigurnost. Uređaj se može postaviti na bilo kojoj bravi, tako da ne mora da se koristi samo za bezbednost u školi, već u bilo kojoj ustanovi gde se ugradi ovaj uređaj.



Slika 4. Smart Gate.

IV. ZAKLJUČAK

Ovaj uređaj može da ima veoma široku upotrebu, u smislu da može da se koristi za zaključavanje bilo kakve prostorije. Prednost ovog uređaja u odnosu na ostale uređaje iste svrhe je to što je veoma jeftin, tj. sve komponente koje su potrebne za izradu ovakvog uređaja su jeftine.

Ovaj uređaj se može unaprediti u budućnosti, tako što će se dodati drugi moduli, ko što je čitač otiska prsta, kao alternativni metod otključavanja. Softverska strana uređaja takođe podleže promenama u smislu da može da se napravi administratorski portal u okviru koga će administratori sistema moći da upravljaju sa nalozima korisnika (dodela, oduzimanje prava pristupa, dodavanje novih korisnika, itd.).

LITERATURA

[1] https://www.i2c-bus.org/
[2] https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/
[3] https://012lab.com/proizvod/wifi-module-esp8266

IoT sistem za akviziciju podataka, merenje temperature, vlažnosti vazduha i intenziteta osvetljenosti

Mihajlo Đorđević

Sadržaj – U ovom radu realizovan je pametni sistem koji je zasnovan na IoT tehnologiji za akviziciju podataka pomoću senzora temperature i vlažnosti vazduha *DHT22* [1] kao i senzora *BH1750* [2] za određivanje intenziteta osvetljenosti korišćenjem *GSM/GPRS Sim800l* [3], modula. Podaci nakon izvršene akvizicije (koja će se sastojati od 10 izvršenih merenja ovih parametara u razmaku od 30 sekundi) biće skladišteni na *ThingSpeak* [4] serveru, pri čemu kanal na serveru treba imati naziv „MEMS“, a polja će nositi nazive Temperatura [C], Vlažnost [%], i Int_Osvetljenosti [lx].

I. UVOD

Termin IoT (*Internet of Things* [5]) je prvi put upotrebljen od strane *MIT Auto-ID* centra 1999. godine i tada se odnosio na viziju sveta gde su svi fizički objekti tagovani sa RFID (*Radio-Frequency Identification*) transponderom sa globalnom jedinstvenom identifikacijom ID. On se odnosi na objekte, stvari i njihove virtuelne predstave u strukturi koja je slična Internetu. U poslednje vreme IoT je postao posebno popularan imajući u vidu niz interesantnih primena: inteligentna proizvodnja, transport, logistika, briga o zdravlju, telemedicina, ekologija (recimo praćenje koncentracije gasova), korišćenje prirodnih resursa, očitavanje pametnih električnih merača, pametne energetske mreže (*smart grid*), pametne kuće, pametni gradovi... Glavne komponente IoT-a su detekcija (osluškivanje) događaja, heterogeni pristup, obrada informacija, primena i servisi, uz sigurnost i privatnost. Uz njega, a nekad i češće, spominju se i termini WSN (*Wireless Sensor Networks*), CPS (*Cyber-Phisycal System*) i M2M (*Machine-to-Machine*).

Može se reći da je IoT globalna mrežna infrastruktura koja spaja fizičke i virtuelne objekte putem korišćenja prikupljenih podataka i mogućnosti komuniciranja. Ova infrastruktura uključuje postojeći i evoluirajući Internet i mreže koje se razvijaju. Ona će ponuditi identifikaciju objekata, korišćenje senzora i mogućnost konekcije na bazi razvoja nezavisnih kooperativnih servisa i aplikacija. Biće okarakterisana visokim stepenom autonomnog prikupljanja podataka, javljanjem o događajima, konektivnošću i interoperabilnošću. Generalno, WSNs, M2M i CPS su slični IoT-u budući da imaju iste komponente. Međutim, razlika je u opsegu primene ova četiri pojma. Ipak treba ukazati da se danas kao sinonimi često koriste M2M i IoT. Takođe treba znati da su definicije M2M, WSNs, CPS i IoT evoluirale sa razvojem tehnologije i novih ideja.

Potencijalne koristi od IoT-a skoro da nemaju granice i odgovarajuće aplikacije menjaju načine kako radimo i živimo, štedeći vreme i resurse i omogućavajući nove pristupe za dalji razvoj i uvođenje inovacija. IoT omogućava privatnim i društvenim organizacijama da upravljaju imovinom, optimizuju performanse i razviju nove modele poslovanja. IoT je vitalni instrument koji omogućava razvijenom društvu da usavrši načine korišćenja energije i optimizuje sve vrste prevoza i transporta. Očigledna je komplementarnost sa računanjem u oblaku (*Cloud computing*), obradom velikih (količina) podataka (*Big Data*) i budućim javnim mobilnim mrežama, kao što je 5G. Uspeh prodora IoT-a zavisiće od razvoja eko-sistema, odgovarajućih pravnih normi i atmosfere poverenja gde su, pored ostalog, važni i aspekti identifikacije, privatnosti i sigurnosti.

U ovom projektu biće dat primer realizacije IoT (*Internet of Things*) pametnog sistema za akviziciju podataka. Za akviziciju podataka tj. skladištenje podataka koristiće se *ThingSpeak* server implementiran na Cloud server razvijen od strane MATLAB kompanije. IoT i Cloud nije isto, Cloud je deo IoT tehnologije i predstavlja medijum za skladištenje podataka. “Pametan” sistem znači da će sistem sam skladištiti podatke na Cloud server bez učešća čoveka. Za slanje podataka na server u bazu podataka biće korišćen GSM Sim800l modul sada konfigurisan kao GPRS. Korišćenje ovog modula ima jednu pogodnost a to je što ne zavisi od Internet konekcije tj. sam modul preko SIM kartice uspostavlja vezu sa Internetom. Za rad GSM Sim800l modula potrebna je struja od 2A iz razloga što Sim800l modul zahteva veliku struju za potrebe slanja SMS-a što ne važi kod Wi-Fi modula (300mA).

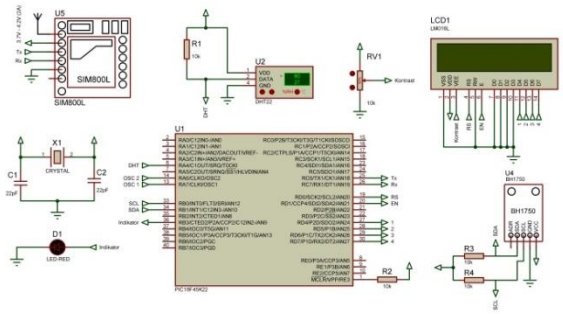
II. ELEKTRIČNA ŠEMA IOT SISTEMA

Kompletna šema IoT sistema za akviziciju podataka predstavljena na Slici 1, projektovana je u programu *Proteus 8 Professional* [6] i biće detaljno objašnjena u nastavku teksta. Realizacija električne šeme izvršena je korišćenjem sledećih komponenata:

Mihajlo Đorđević je student master studija na studijskom programu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: mihajlo628@hotmail.com

A. PIC mikrokontroler PIC18F45K22 [7]

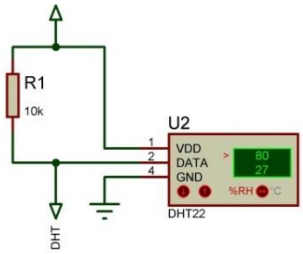
PIC18F45K22 je 40-o pinski mikrokontroler sa 32kB programske memorije i 16 MIPS (milion instrukcija u sekundi). Ovaj mikrokontroler na svom radnom okruženju Ready for Pic ima eksterni kvarcni kristal frekvencije od 8MHz (poseduje interni oscilator od 64MHz). Takođe poseduje 2 UART modula, 2 I²C i 2 SPI modula za komunikaciju sa periferijama, 28 kanala A/D konvertora rezolucije 10 bita, 2 komparatora i napajanje u opsegu od 1,8V do 5,5V.



Slika 1. Ovo je primer kompletne električne šeme IoT sistema za akviziciju podataka.

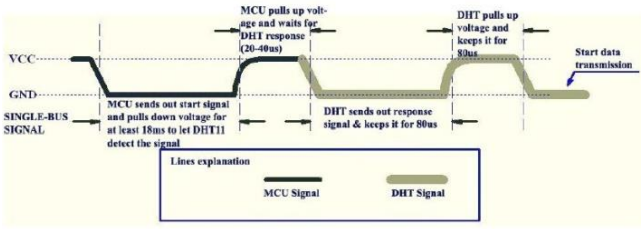
B. Meteosenzor DHT22

Na Slici 2 prikazan je DHT22 digitalni senzor koji pruža mogućnost merenja temperature u opsegu od 0°C do 50°C i relativnu vlažnost vazduha u opsegu od 20% do 90%. Senzor se napaja jednosmernim naponom vrednosti koje su u opsegu od 3V do 5,5V.



Slika 2. Realizacija meteosenzora DHT22 na električnoj šemi.

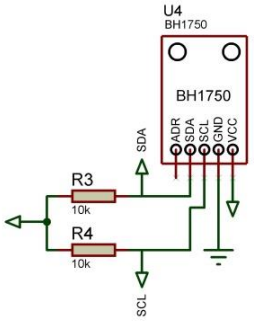
Na Slici 3 prikazan je način inicijalizacije ovog senzora kao i na koji način on uspostavlja komunikaciju. DHT22 senzor koristi jednu liniju za komunikaciju. Najpre je neophodno poslati signal niskog naponskog nivoa (0V) u trajanju od najmanje 18ms. Signal niskog naponskog nivoa šalje se kako bi senzor detektovao signal i izvršio početnu inicijalizaciju. Nakon toga se šalje signal visokog naponskog nivoa (5V) kako bi senzor odgovorio, na odgovor se čeka od 20µs do 40µs. Zatim u trajanju od 80µs prima se odgovor senzora u vidu niskog naponskog nivoa. Na kraju senzor podiže napon na viši naponski nivo (5V) u trajanju od 80µs. Ovime je završena inicijalizacija senzora i omogućeno je merenje i očitavanje rezultata.



Slika 3. Način inicijalizacije i komunikacije sa DHT22 senzorom.

C. Senzor vidljive svetlosti BH1750

Na Slici 4. prikazan je BH1750 digitalni senzor svetlosti koji se kontroliše već realizovanom I²C magistralom. Pogodan je za kontrolu osvetljenja LCD displeja, realizaciju luxmetra, kontrolu osvetljenja u prostorijama... Ima širok opseg detekcije intenziteta osvetljenosti od 0 do 65535Lx pri visokoj rezoluciji, a poseduje i 16-bitni A/D konvertor koji omogućava visoku rezoluciju merenja.



Slika 4. Realizacija senzor vidljive svetlosti BH1750 na električnoj šemi.

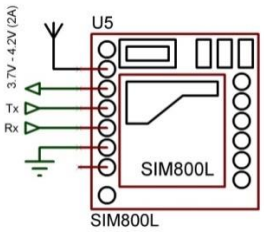
I²C (*Inter-Integrated Circuit*) komunikacija omogućava podršku za komunikaciju sa različitim periferijskim jedinicama u sistemima gde se potreba za njihovom upotrebom javlja povremeno. Fizički I²C magistrala sadrži samo 2 linije signala:

- **SCL (*Serial Clock*) –** služi za prenos taktnog signala.
- **SDA (*Serial Data*) –** služi za prenos podataka

Uređaji priključeni na I²C magistralu dele se u dve grupe: *Master* i *Slave*. Master uređaj generiše taktni signal SCL, on je upravljački, i on može adresirati bilo koji priključeni Slave uređaj sa ciljem predaje ili prijema informacija.

D. GSM/GPRS Sim800l modul

Na Slici 5 predstavljen je jedan od najboljih GSM modula kako zbog mogućnosti kao što su slanje i primanje SMS-a, upućivanje i primanje poziva i GPRS(Internet) komunikacija, tako i zbog malih dimenzija. Ovaj modul poseduje takođe i FM radio što ga izdvaja od ostalih GSM modula na tržištu.



Slika 5. Realizacija GSM/GPRS Sim800l modula na električnoj šemi.

Ovaj modul za komunikaciju sa mikrokontrolerom koristi UART – Tx/Rx serijsku komunikaciju obično pri brzini od 9600bps. Baudrate je moguće menjati softverski. Ovaj modul zahteva napajanje naponom vrednosti od 3,7V do 4,2V i napon iznad ovog opsega može oštetiti ovaj modul. Osim navedenih pinova ovaj modul poseduje i pin za reset(RST) kojim se ovaj modul resetuje ukoliko dođe do neke greške. Sa druge strane ovog modula nalaze se pinovi RING, DTR, MIC+/-, i SPK+/-. Ovi pinovi se mogu koristiti prilikom korišćenja modula za prijem odnosno uput poziva. Pinovi SPK+/- koriste se za FM radio. Led dioda na vrhu modula služi kao indikator struja tj. da li je modul povezan na mrežu ili ne. Dodatna stvar vezana za napajanje modula je potreba za velikom strujom koja pri slanju podataka (SMS,GPRS...) ide i do 2A. Zato je potrebno vezati elektrolitski kondenzator vrednosti 2200µF između VCC i GND.

III. PROGRAMSKI DEO IOT SISTEMA ODRAĐEN U "PROTON IDE" KOMPAJLERU

Prilikom projektovanja električnih šema koje sadrže mikrokontrolere, rad istih je nemoguć ukoliko u mikrokontrolere nije učitan .hex fajl na osnovu koga će raditi mikrokontroler. Heksadecimalni kod se dobija nakon kompajliranja koda koji je napisan u nekom od programskih jezika u nekom od programskih kompajlera. Jedan od takvih programskih jezika je i *Proton Basic* [8]. Pri pisanju novog koda moguće je podesiti zaglavlje u kome se navode informacije samog koda (naziv koda, datum izrade, informacije o autoru...). Pisanje počinje naredbom *Device* kojom navodimo mikrokontroler koji koristimo, zatim sledi naredba *Xtal* koja definiše frekvenciju (radni takt) mikrokontrolera koji koristimo. Nakon toga sledi definisanje promenljivih koje koristimo u kodu. Kao što se vidi na Slici.4 I^2C magistrala je realizovana na pinovima PORTB – B0 i B1 (SCL i SDA). Mikrokontroler PIC18F45K22 poseduje 2 UART modula u sebi kojima se bez softverskog definisanja mogu koristiti I^2C i SPI magistrale. Navedeno je da BH1750 poseduje 16-bitni A/D konvertor tako da je maksimalna vrednost koja se dobija tokom merenja 65535. Iz tog razloga promenljiva u koju smeštamo rezultat merenja ima tip podataka *Word* (0-65535). Sledeći korak jeste definisanje serijske komunikacije za rad sa GSM/GPRS modulom Sim800l. Brzina serijske komunikacije je 9600bps tako da za razliku od RF modula koji za rad zahteva samo Tx ili Rx pin, GSM/GPRS modul Sim800l za serijsku komunikaciju zahteva i Tx i Rx.

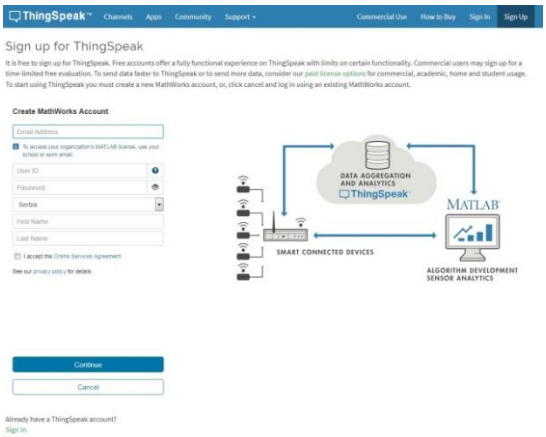
U glavnom delu programa na samom početku izvršena je inicijalizacija senzora, zatim ispitujemo koliko je impulsa primljeno sa pina na koji je povezan DHT22 senzor (da nije došlo do neke greške ili da senzor možda nije lepo povezan). Nakon toga vrši se očitavanje merenja tako što se u promenljivu za temperaturu i vlažnost vazduha upisuju impulsi primljeni iz senzora. Upisivanje se vrši tako da se 15 bitova u promenljivoj *Word* treba popuniti impulsima primljenim iz senzora. Sada treba izvršiti inicijalizaciju I^2C magistrale. Bajt za inicijalizaciju razlikuje se u poslednjem bitu. Razlog tome je da li se vrši upis na I^2C ili se vrši čitanje sa I^2C, 0 je za upis a 1 za čitanje. Nakon inicijalizacije I^2C magistrale primljene podatke merenja treba smestiti u pomenljivu A formata *Word* koja se deli na viši i na niži bajt. Na kraju kada je rezultat smešten u promenljivu A treba tu vrednost podeliti sa 1,2 na osnovu čega se dobija vrednost intenziteta osvetljenosti u luksima (Lx). Nakon izvršenog merenja, da bi se informacija poslala neophodno je konvertovati dobijenu vrednost u promenljivu tipa *String* zbog slanja na server u bazu podataka. To je tako jer GPRS koristi karaktere iz ASCII tabele, i slanje celobrojne vrednosti bi dalo pogrešan podatak u bazi.

Kada je konvertovanje završeno sistem je spreman za slanje podataka na server. Pre slanja treba podesiti modul za rad kao GPRS modul – za komunikaciju sa Internetom. Najpre treba inicijalizovati modul slanjem komande AT. Nakon toga sledi provera da li modul ima konekciju AT+CPIN? (da li treba uneti šifru), AT+CREG (da li se modul registrovao na mrežu), AT+CGATT (da li ima konekcije na GPRS?). Zatim sledi isključivanje komunikacije (reset IP adrese) AT+CIPSHUT kako bi se nastavilo sa podešavanjem modula za slanje podataka. Zatim šaljemo zahtev za odgovor o trenutnom stanju modula AT+CIPSTATUS da bi se on posle podesio kao klijent AT+CIPMUX=0. Zatim treba podesiti parametre za pristupnu tačku (APN) – korisničko ime operatera i šifru koje operater zahteva da bi GPRS komunikacija bila omogućena. Sada šaljemo zahtev modulu da se konektuje na GPRS operatera AT+CIICR a nakon toga zahtev za preuzimanje lokalne IP adrese modula, AT+CIFSR zato što se komunikacija odvija putem TCP/IP protokola. Sledi uspostavljanje konekcije sa *ThingSpeak* serverom putem TCP protokola slanjem:
AT+CIPSTART="TCP","api.thingspeak.com","80". Ovde se figuriše "api.thingspeak.com" jer *ThingSpeak* server je API (aplikaciono programabilni interfejs) a za broj porta navodi se "80". Šaljemo zahtev *ThingSpeak* serveru za prijem podataka putem GET zahteva AT+CIPSEND nakon čega se otvara komunikacija i sledi navođenje onog što šaljemo putem GET zahteva. GET zahtev se u ovom slučaju smešta u istoimenu promenljivu *String*:
GET="GEThttp://api.thingspeak.com/update?api_key=xxxxxxxxxxxxxx&field1=".

Nakon linka *ThingSpeak* servera navodi se *update?* jer *ThingSpeak* čeka nadolazeće podatke i vrši update. Tu je i *api_key* jer se zahteva API ključ kako bi se znalo u koji se kanal smeštaju primljeni podaci (ključ generiše sam *ThingSpeak* server, i sastoji se od slova i brojeva, i jedinstven je za svaki kanal). Kada je GET zahtevu sve dodato treba još dodati konvertovanu vrednost temperature (npr. GET=GET+StrTemperature). Zatim se navodi Ctrl+Z u obliku $1A. I na kraju se komunikacija zatvara sa AT+CIPSHUT čime je završeno slanje podataka. Na samom kraju naš brojač uvećavamo za 1 i vršimo proveru da li su izvršena 10 merenja koliko nam je i potrebno.

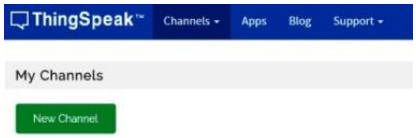
IV. KREIRANJE NALOGA NA THINGSPEAK SERVERU

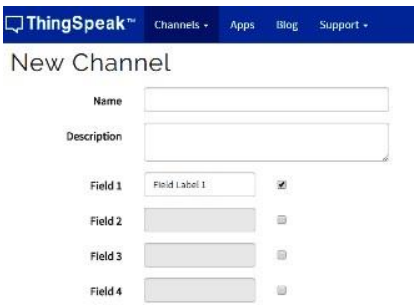
ThingSpeak predstavlja IoT analitičku platformu koja omogućuje da se objedine, vizualizuju i uživo analiziraju tokovi podataka na Cloud-u. *ThingSpeak* pruža trenutnu vizualizaciju podataka koje su naši uređaji objavili na *ThingSpeak*-u. Uz mogućnost izvršavanja MATLAB koda u *ThingSpeak*-u možemo izvršiti *on-line* analizu i obradu podataka u trenutku kada sam podatak dolazi. *ThingSpeak* se često koristi za izradu prototipa i dokaz konceptnih IoT sistema koji zahtevaju analitiku. Na Slici 6 prikazan je izgled servera gde je neophodno popuniti polja kako bi se kreirao nalog na serveru.



Slika 6. Izgled thingspeak.com sajta i faze kreiranje naloga.

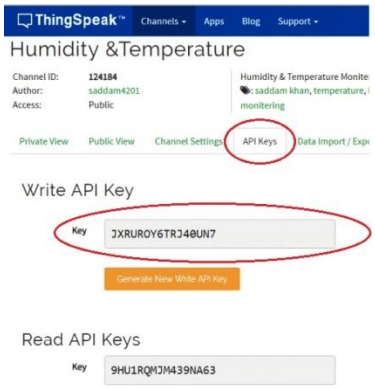
Kada je nalog kreiran i korisnik je prijavljen, moguće je kreirati kanal u koji će biti skladišteni podaci (New Channel). Za kreiranje novog kanala je neophodno popuniti polja koja se zahtevaju od korisnika kao što je prikazano na Slici 7 (naziv kanala, omogućiti broj polja koja se prikazuju na kanalu...).





Slika 7. Odgovarajuća polja na sajtu koja treba popuniti prilikom kreiranja kanala.

Kada je kanal kreiran iz kartica iznad naziva kanala kao što je prikazano na Slici 8, sa oznakom API nalazi se API ključ kreiranog kanala koji se koristi za skladištenje podataka, mogućnost generisanja novog API ključa i API ključ za preuzimanje podataka sa kreiranog kanala.

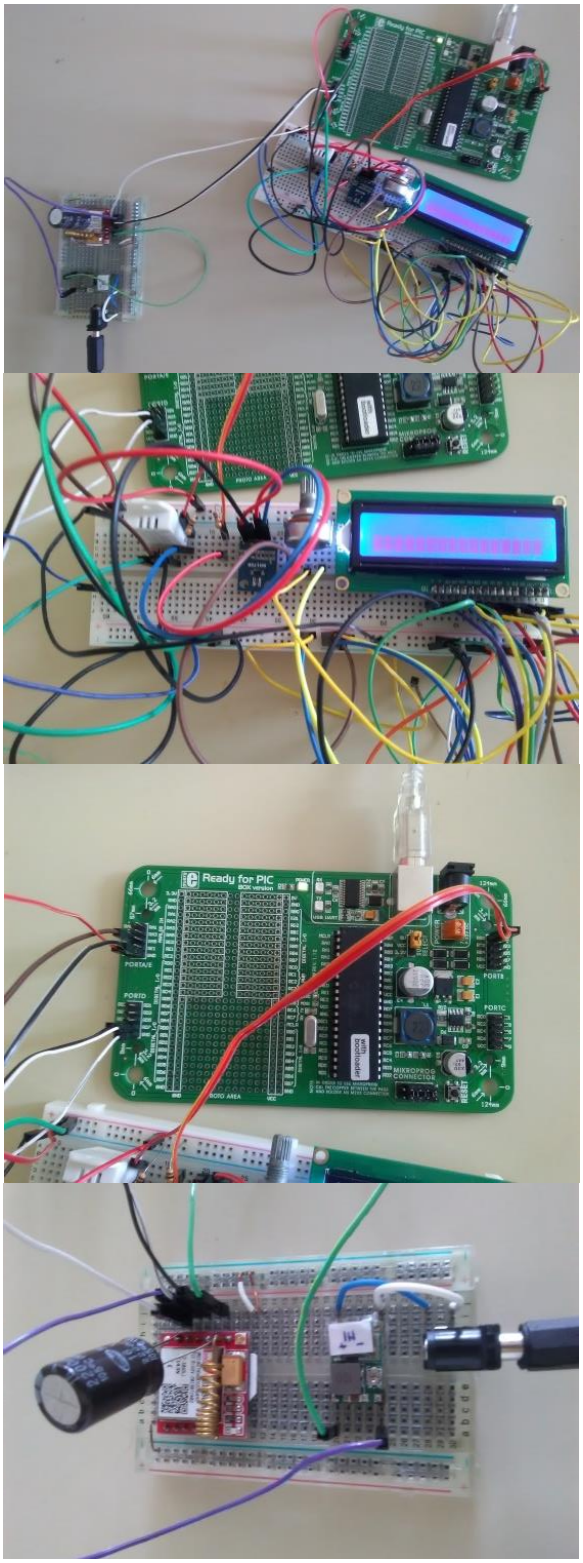


Slika 8. Način dobijanja jedinstvenog API ključa kreiranog kanala.

Podaci koji se skladište na server osim samo vizuelne prezentacije, moguće je preuzeti u tri različita formata **CSV** (Comma Separated Value), **JSON** (JavaScript Object Notation) i **XML** (Extensible Markup Language). Takođe je važno napomenuti da *ThingSpeak* omogućuje do najviše 8 polja na jednom kanalu za prikaz podataka koji su uskladišteni na server.

V. PRAKTIČNA REALIZACIJA IOT SISTEMA I REZULTATI MERENJA PREDSTAVLJENI NA THINGSPEAK SERVERU

Ovaj IoT sistem je uspešno realizovan korišćenjem okruženja *Ready for PIC* firme Mikroelektronika što je i prikazano na Slici 9. *Ready for PIC Board* je najbolje rešenje za brz i jednostavan razvoj mikrokontrolera koji koriste 28- i 40-pinske PIC mikrokontrolere. Dolazi sa PIC18F45K22 mikrokontrolerom spojenim na kristalni oscilator od 8MHz. Sadrži četiri IDC10 pinova(*headers*) za sve dostupne portove mikrokontrolera, USB-UART modul i napajanjem od 3,3V i 5V.



Slika 9. Praktična realizacija IoT sistema uz korišćenje proto pločice.

Kada je sve završeno i kada se podaci šalju, odnosno skladište na *ThingSpeak* server, prikaz uskladištenih vrednosti možemo videti na Slici 10:



Slika 10. Prikaz prikupljenih podataka na ThingSpeak serveru.

VI. ZAKLJUČAK

U ovom radu prikazana je realizacija IoT pametnog sistema za akviziciju podataka pri čemu su podaci na kraju skladišteni i analizirani na *ThingSpeak* serveru. Pored toga data su i objašnjenja o korišćenim komponentama pri realizaciji IoT sistema kao i objašnjenje programa napisanog u *Proton IDE* kompajleru.

Što se tiče samog IoT sistema, zavisnost od Interneta i upotreba mobilnih uređaja raste velikom brzinom. Dostupnost jeftinih procesora i procesora male snage, kao i nedavni razvoj u komunikacionim tehnologijama niske potrošnje omogućavaju da se IoT tehnologija koristi u svakodnevnom životu.

ZAHVALNICA

Autor se zahvaljuje dr Vesni Paunović, dr Danijelu Dankoviću i dipl. inž. Milošu Đorđeviću na izdvojenom vremenu, sugestijama, i pomoći pri realizaciji ovog projekta.

LITERATURA

[1] "DHT22" dostupno na: https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
[2] "BH1750" dostupno na: https://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf
[3] "GSM/GPRS Sim800l" dostupno na: https://lastminuteengineers.com/datasheets/SIM800L-Datasheet.pdf
[4] "ThingSpeak server" dostupno na: thingspeak.com
[5] Dejan D. Drajić, "Uvod u IoT (Internet of Things)", Akademska misao, 2018
[6] "Proteus 8 Professional", dostupno na: https://www.labcenter.com/
[7] "PIC18F45K22" dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/40001412G.pdf
[8] „Proton IDE Compiler" dostupno na: http://www.protonbasic.co.uk/vbdownloads.php?do=download&downloadid=416
[9] Miloš Đorđević, Skripta "Praktikum za laboratorijske vežbe iz predmeta MikroElektroMehaničkiSistemi", Niš 2019.

Realizacija štampane ploče za projektovani taksimetar

Milan Danković

Sadržaj – U ovom radu detaljno je opisana realizacija taksimetra. Taksimetar je projektovan korišćenjem programskog paketa Quartus, a fizički je realizovan na ALTERA FPGA razvojnom okruženju. Realizovanom taksimetru mogu da se dodaju i dodatne funkcionalnosti.

I. UVOD

Kretanje ljudi i robe oduvek predstavlja važan faktor razvoja društvenih i privrednih sistema. Uobičajeni naziv koji se u literaturi sreće za kretanje ljudi i robe je transport, a potiče od latinske reči *transportus*. Međutim, u svakodnevnom životu češće se koristi termin saobraćaj, a on podrazumeva kretanje transportnih sredstava po određenim utvrđenim putanjama. Istorijski gledano prvo se pojavio kopneni saobraćaj, ali sigurno je da vazdušni i vodeni saobraćaj u 21. veku predstavljaju žilu kucavicu svetskog transporta i trgovine [1]. Svima je jasno da je potpuno nemoguće zamisliti današnji svet bez transporta ljudi i robe. Međutim, bitno je istaći da svi ljudi ne raspolažu transportnim sredstvima (automobil, voz, avion, brod…), pa su prinuđeni da za zadovoljavanje svojih potreba koriste neki vid javnog prevoza.

Vrsta javnog prevoza gde putnik iznajmljuje na određeno vreme prevozno sredstvo sa vozačem koji će ga zajedno sa prtljagom prevesti od jedne lokacije do druge naziva se taksi. Pod taksijem se najčešće podrazumeva automobil ali to može biti i avion, čamac, gondola, kamion itd. Taksiti na krovu imaju svetleću oznaku na kojoj piše “Taxi“ (slika 1), u nekim zemljama i broj taksi vozila. Kada je svetlo upaljeno taksi je “slobodan“ i spreman za narednu vožnju.



Slika 1. Prevozna sredstva koja se koriste za taksi prevoz uvek su označena. U nekim državama pored naziva TAXI koristi se i naziv CAB.

Milan Danković je student Softverskog inženjerstva na Elektrotehničkom fakultetu, Univerziteta u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija, E-mail: dankovicmilan@gmail.com

Prema podacima iz literature prvi taksisti su se pojavili početkom 17. veka u Parizu i Londonu. Prvi taksista o kome postoji dokumentacija da je prijavio državnim organima posao taksiranja 1640. godine bio je Nikola Sovaž (engl. *Nicolas Sauvage*). U to doba, prvi gradski taksi se odvijao u fijakerima uz obaveznu konjsku vuču, što se može videti u mnogim filmovima [2, 3].

Beograd nije zaostajao za evropskim prestonicama po pitanju taksi prevoza. Pravi razvoj taksi prevoza u Beogradu bio je tek po završetku Prvog svetskog rata, fijaker je sve ređe saobraćao beogradskim ulicama, dok je njegovo mesto zauzeo pravi auto-taksi. Iz godine u godinu, broj profesionalnih šofera se povećavao, organizovali su se u udruženja i bili su prepoznatljivi (Slika 2) [2-4].



Slika 2. Jedan od prvih profesionalnih šofera u Beogradu (Ivan Majer) [4].

Kao i kod svakog drugog zanimanja, da bi sebi obezbedili platu taksisti su svoje usluge naplaćivali. Uz izvesne razlike, način naplaćivanja taksi usluge je isti, odnosno taksi usluga se plaća iz dva dela:

-“polazak“ - fiksni iznos bez obzira na rastojanje koje će se preći i

-naplata po “pređenom putu“.

Novčani iznos “polaska“ i iznos koji se plaća po “pređenom putu“ zavise od zakonske regulative, doba dana, dana u nedelji, praznika itd. Taksi vožnja se najčešće naplaćuje putem taksimetra, koji je povezan sa elektronikom automobila kojom se meri pređeni put tokom vožnje[1]. Taksimetar je pronašao Vilhelm Brun (engl. Wilhelm Bruhn) 1891. godine. Korišćenjem tehničkih dostignuća današnjice na tržištu se može pronaći ogroman broj različitih taksimetara sa sa vrlo širokim spektrom funkcionalnosti. Bez obzira na to, projektovanje i realizacija taksimetra je izazov za svakog inženjera.

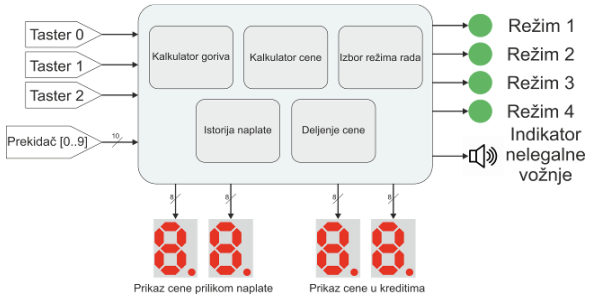
[1] Taksimetar radi i ako se taksi zaglavi u saobraćajnoj gužvi, tada se prebacuje na režim merenja vremena, pa i dalje radi.

II. PRINCIP RADA TAKSIMETARA

Taksimetar je mehanički, a danas najčešće električni uređaj postavljen u taksi vozilima koji služi za izračunavanje cene taksi usluge. Pored osnovne funkcionalnosti većina komercijalnih taksimetara često ima još nekoliko dodatnih funkcionalnosti kao što su:

- izbor režima rada i različitih tarifa,
- izbor početne cene,
- mogućnost podešavanja cene po pređenom kilometru,
- mogućnost podešavanje cene za minut čekanja,
- pamćenje određenog broja vožnji,
- štampanje računa,
- GPS praćenje taksi vozila itd.

Najmoderniji taksimetri imaju i mogućnost detektovanja putnika na sedištu tako da taksista ne može krišom besplatno vršiti prevoz putnika. Takođe mnogi taksimetri imaju mogućnost unosa popusta u procentima da bi se kasnije taj deo cene odbio, a što se realizuje pritiskom na određeni taster. Neke od napred navedenih funkcionalnosti ima taksimetar koji je projektovan i prikazan u ovom radu. To su mogućnost postavljanja startne cene i njenog korigovanja, opciju za izbor tarife, pregled prethodnih 16 naplaćenih računa, ali i neke nove kao što su mogućnost podele računa između više mušterija i režim pauze. Kada se taksimetar uključi taksista je u režimu broj 1 koji označava da je slobodan i to mu govori indikatorska dioda, a taj izlaz možemo dalje da vodimo i do TAXI znaka na krovu automobila. U ovom režimu moguće je podesiti početnu cenu vožnje. Kada putnik uđe u vozilo prelazi se u drugi režim rada u kome se vrši odabir tarife u zavisnosti od koje će se dalje naplaćivati taksi usluga. Treći režim rada označava da vozilo trenutno prevozi putnika ili putnike i tada taksimetar otkucava cenu na osnovu impulsa koji dolazi iz elektronike automobila (vreme i/ili pređeni put). Prilikom završetka vožnje taksista ulazi u četvrti režim iz koga se može direktno vratiti u režim vožnje, može naplatiti uslugu u celosti i vratiti se u prvi režim ili može podeliti trenutnu cenu sa brojem putnika ukoliko ne izlaze svi putnici na datom mestu, zatim tu cenu naplati putnicima koji napuštaju vozilo i nakon toga nastavi prevoz preostalih putnika. Na slici 3 prikazana je grafička ilustracija načina kontrole, funkcionalnih blokova i način prikazivanja funkcionalnosti taksimetra koji je projektovan.

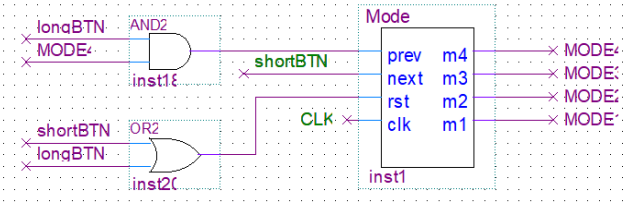


Slika 3. Grafička ilustracija taksimetra. *Jedan kredit, zavisno od podešavanja može da ima različite vrednosti (1 EUR, 10 dinara).

Prikazani su ulazni signali koji kontrolišu rad taksimetra, kao i izlazi taksimetra koji predstavljaju rezultat izvršenih operacija (cena, režim vožnje, zujalica za nelegalnu voznju). Blokovi predstavljaju apstraktne logičke celine koje funkcionišu zajedno i realizuju logiku taksimetra. Tasteri 0-2 služe za upravljanje radom taksimetra. Imaju mogućnost registrovanja kratkog i dugog klika i u zavisnosti od toga vrši se odgovarajuća operacija. Prekidačima zadajemo binarno vrednosti početne cene, cene goriva, cenu za kilometar... Poslednji prekidač služi da u stanju mirovanja prebacimo u režim gledanja memorije ili u stanju 4 prebacimo u režim podele cene za naplatu.

III. PROJEKTOVANJE I TESTIRANJE TAKSIMETRA

Idejno rešenje taksimetra projektovano je u softverskom paketu Altera Quartus II [5]. Altera Quartus II je softverski paket koji nam omogućava projektovanje svih faza FPGA uređaja u konzoli ili kroz grafički interfejs kao što je ilustrovano na slici 4.

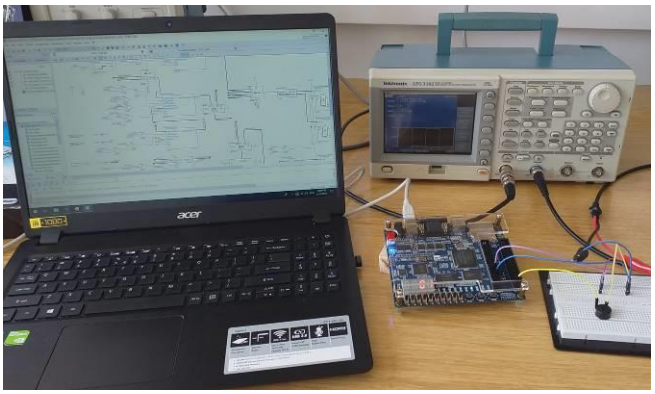


Slika 4. Quartus, deo logičke šeme kojom se vrši izbor režima rada taksimetra.

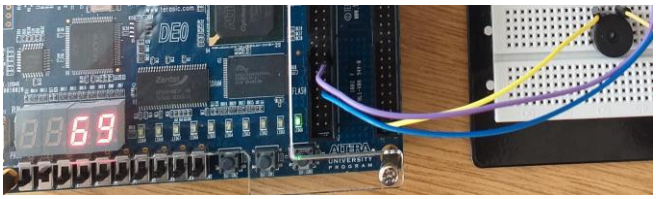
Testiranje taksimetra izvršeno je na Altera DE0 ploči [6] na kojoj se nalazi Altera Cyclone III 3C16 FPGA čip. Altera DE0 ploča je edukativno razvojno okruženje sa velikim asortimanom mogućnosti dizajna FPGA uređaja. Za proveru funkcionisanja taksimetra iskorišćeni su integrisani tasteri, prekidači, diode, četiri sedmosegmentna displeja i GPIO pinovi Altera DE0 ploče. Pored toga, u svrhu realizacije projekta korišćena je i protopločica, otpornici, diode, zujalica i žice za povezivanje. Signal takta za merenje vremena, kao i signal za simulaciju pređenog puta dovođeni su iz signal generatora [7] preko GPIO pinova razvojnog okruženja, a za snimanje signala korišćen je osciloskop [8]. Ovi signali u praksi dovodili bi se iz elektronike automobila. Okruženje u kome je projektovan i testiran taksimetar prikazano je na slici 5.

Prototip taksimetra kompletno je testiran u svim režimima rada. Mogući režimi navedeni su ranije u radu, a potvrda ispravnosti projektovanog taksimetra za neke od objašnjenih režima rada prikazana je na slici 6. Prikazana je cena na kraju vožnje u četvrtom režimu rada, podela cene vožnje između dva putnika, kao i pregled istorije cena prethodnih vožnji. Prototip taksimetra testiran je korišćenjem komponenata povezanih na Altera DE0 ploču. Očigledno je da za testiranje taksimetra iskorišćen samo deo komponenata razvojne ploče. Ideja je napraviti

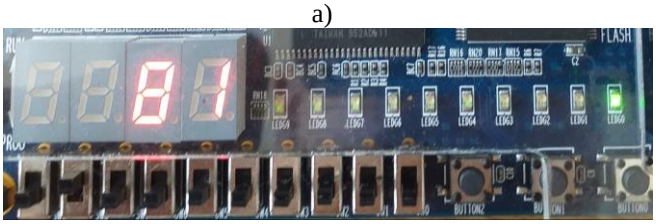
razvojnu ploču na kojoj će se nalaziti samo komponente neophodne za realizaciju taksimetra.



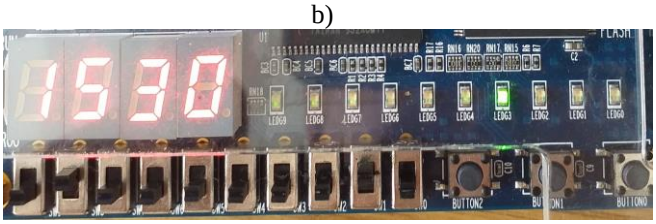
Slika 5. Radno okruženje.



a)



b)



c)

Slika 6. Rezultati testiranja projektovanog taksimetra. a) Pregled istorije cena i korišćenje zujalice, b) Pregled istorije cena, c) Podela cene između dva putnika.

IV. Projektovanje štampane ploče taksimetra korišćenjem Altium Designer-a

Za projektovanje štampane ploče iskorišćen je softverski paket Altium Designer [9]. Altium Designer ima četiri glavne funkcionalnosti: projektovanje električnih šema, 3D projektovanje štampanih ploča, razvoj FPGA uređaja i organizacija verzije proizvoda. Projektovanje štampane ploče taksimetra u Altium Designer-u je izuzetno složen zadatak imajući u vidu da je izvode komponenata sa ploče potrebno spojiti sa izvodima Altera Cyclone III 3C16 FPGA čipa [10]. Imajući u vidu broj pinova (484), za povezivanje svih izvoda na ploči bilo je potrebno koristiti više slojeva. Proces povezivanja izvoda komponenata započet je sa dva sloja, a onda je postepeno povećavan sve do trenutka kada je autoruter uspešno povezao sve veze. Broj potrebnih slojeva za uspešno povezivanje je deset, pri čemu treba napomenuti da je uz povoljniji raspored komponenta verovatno bilo moguće projektovanje uraditi i sa manjim brojem slojeva.

Za sve komponente koje su korišćene pri projektovanju štampane ploče taksimetra obezbeđeni su i 3D modeli (lako dostupni na različitim forumima gde projektanti razmenjuju 3D modele različitih objekata). Izgled štampane ploče taksimetra sa ubačenim 3D modelima komponenata prikazana je na slici 7. Dimenzije štampane ploče su 65 x 125 mm. Na pločici nije prikazan FPGA čip, već samo veze ka njegovim izvodima (linije na slici 7 koje se nalaze van štampane ploče). Povezivanje samog čipa je kompleksan zadatak koji prevazilazi okvire ovog rada, pa je to ostavljeno kao predmet nekog budućeg rada.

V. Zaključak

U ovom radu na početku je dat pregled komercijalnih taksimetara i njihovih funkcionalnosti, a zatim je detaljno opisana realizacija taksimetra osmišljenog od strane autora. U kratkim crtama opisan je programski paket Quartus u kojem je taksimetar projektovan, kao i Altium Designer u kojem je projektovana štampana ploča koja bi fizički podržavala ovako osmišljeni taksimetar. Prikazani su i rezultati testiranja prototipa razvijenog taksimetra na DE0 razvojnoj ploči. Razvijeni taksimetar može se obogatiti dodatnim funkcionalnostima koje poseduju neki najmoderniji komercijalni taksimetri.

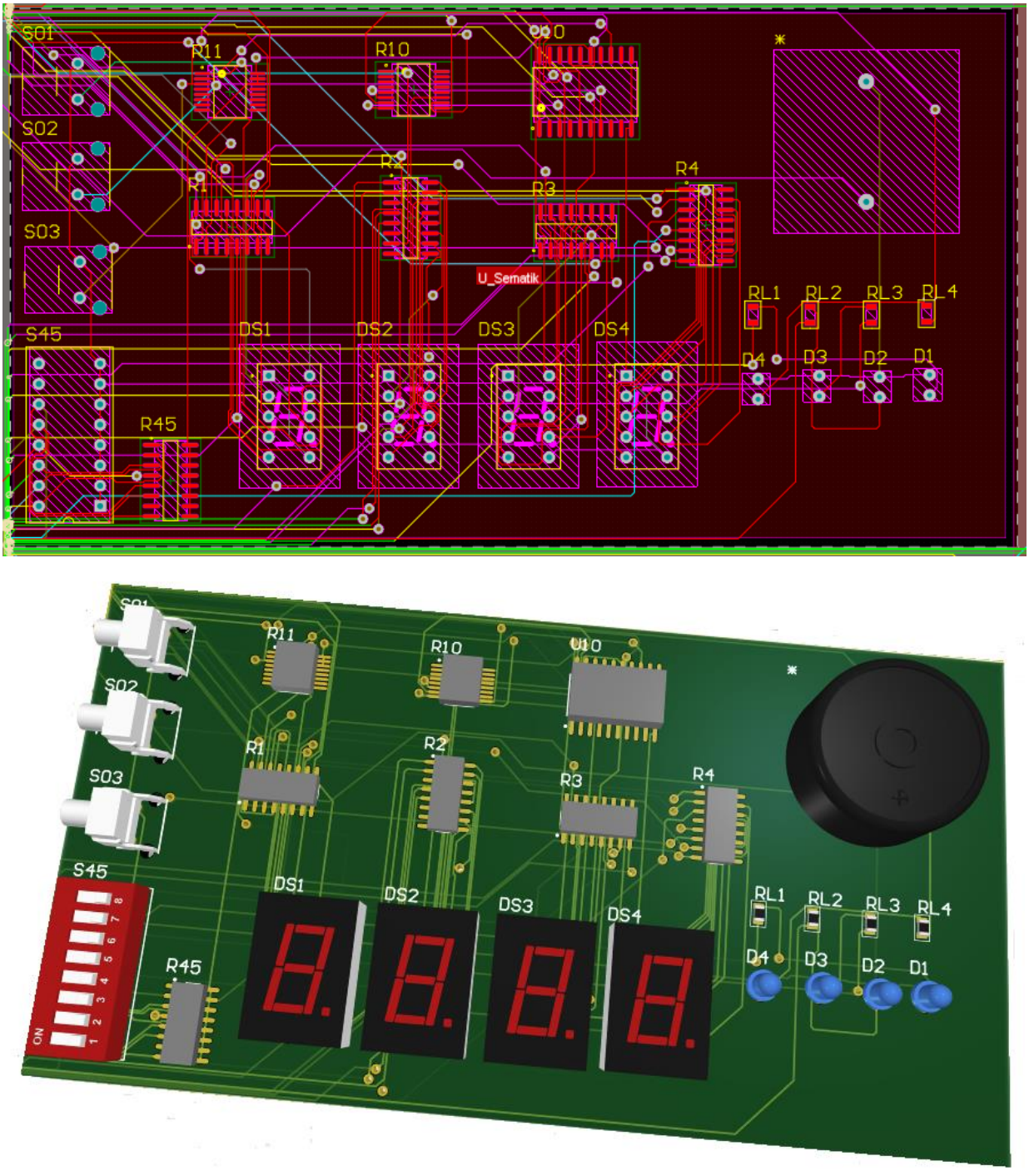
Zahvalnica

Inspiracija za ovaj rad potiče sa predmeta “Praktikum iz osnova računarske tehike” koji se sluša na smeru SI na Elektrotehničkom fakultetu Univerziteta u Beogradu. Ključna znanja vezana za ovaj rad stečena su iz ovog predmeta, a praktičan rad u laboratoriji i projektovanje štampane ploče u Altium Designer-u doprineli su kompletiranju zadatka.

Literatura

[1] K. P. Lipovac, *Bezbednost saobraćaja*, Javno preduzeće Službeni list SRJ, Beobrad, 2008.
[2] Prvi taksisti u Beogradu. Dostupno na: https://www.011info.com/bilo-jednom-u-beogradu/prvi-taksisti-u-beogradu
[3] Istorija taksija. Dostupno na: https://www.thoughtco.com/hailing-history-of-the-taxi-1992541
[4] Oldtajmeri. Dostupno na: http://oldtajmeri.rs/
[5] Quartus II. Dostupno na: https://www.intel.com/content/www/us/en/programmable/downloads/software/quartus-ii-we/120.html

[6] Altera DE0 board. Dostupno na: https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-/board/altera-de0-board.html
[7] Signal generator. Dostupno na: https://www.tek.com/signal-generator/afg3102c
[8] Osciloskop. Dostupno na: https://www.tek.com/oscilloscope/tbs1000-digital-storage-oscilloscope
[9] Altium Designer. Dostupno na: https://www.altium.com/altium-designer/
[10] Altera Cylone III 3C16 FPGA čip. Dostupno na: https://www.intel.com/content/www/us/en/products/programmable/fpga/cyclone-iii.html



Slika 7. Layout i 3D prikaz štampane ploče taksimetra.

Operation of Dynamic Emulator of Multi-Motor Drive with Different Dynamic Emulation Methods

Nikola Krstić

Abstract – Importance of adequate testing of multi-motor drives has been confirmed by their increasing role in technological production processes, which must be efficient and fault-free. One of the best ways to test such drives is to use a dynamic emulator of multi-motor drives, whose operation using various dynamic emulation methods will be considered in this paper. In the theoretical part of the paper, models of used multi-motor drives and methods of dynamic emulation are discussed. While experimental results presented in the end are obtained over the real system of dynamic emulator.

I. Introduction

Most of the products made by humans involve the use of complex technological processes that are realized by multi-motor drives. These technological processes are of great importance because of the quantity and value of the products they produce, and therefore, any interruptions in operation can not be tolerated. To prevent this from happening, motors that are the integral part of multi-motor drives, must be adequately tested. This test involves creating identical working conditions with those that motors have in the real multi-motor drives, for what purpose dynamic emulator of multi-motor drives is used.

The main topic of this paper is the operation of the dynamic emulator with different methods of dynamic emulation, for whose understanding it is necessary first to understand the concept of the multi-motor drive as a central object in this paper and the concept of dynamic emulation as the basic idea of the functioning of the dynamic emulator. For this reason, at the beginning of the work, the concept of two types of multi-motor drives is presented. Models of this drives will be used in computer simulations and experiments for testing the dynamic emulator. Then the paper deals with explaining the basic concepts of dynamic emulation and dynamic emulator in multi-motor drives, with a brief look at static emulation. This term is explained through three methods of dynamic emulation. Explanations of this methods are dominantly related to the design of the controllers that are part of the dynamic emulator control system. Practical part of the paper showing the experimental results obtained over a real dynamic emulator system presenting two different multi-motor drives using three different dynamic emulation methods is given at the end of the paper.

N. Krstić is with the Department of Power Engineering, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikolakrstic@elfak.rs

II. Models of Multi-Motor Drives

For better understanding the concept of dynamic emulation of multi-motor drives, two different models of multi-motor drives [1] will be analysed and emulated. Both of this two models represent a two-motor drive for purpose of physical realization and later possibility of experimental verification. Also the models and operating modes that will be presented and processed in the simulations are adopted according to the rated power and currents of the used motors. Also this models do not represent real systems in practice but rather their reduced approximation, while maintaining and appreciating the main aspects of their operation. The two models considered are:

1. Model of a drive with a conveyor belt
2. Model of a drive with a platform

A. Model of a drive with a conveyor belt

The conveyor belt drive model is a model of a regulated multi-motor drive of a conveyor belt which is driven by two motors. This motors also keep tensile force in the belt for all time in their operation of load transfer.

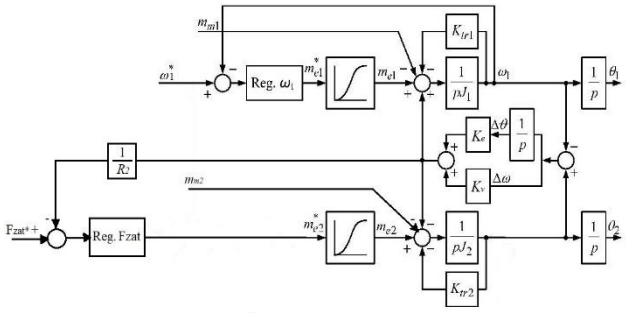


Figure 1. Simplified structural block diagram of a drive with a conveyor belt.

Figure 1 shows the structure of the control system in which the first motor has a speed controller while the speed of the second motor is controlled indirectly through tensile force controller. Mechanical system of the drive is linearized and described with moment of inertia calculated from expression 1 and coefficient of viscous friction calculated from expression 2. This two parameters are the same for both motors considering identical mechanical system of this motors.

$$J_{em} = J_{mot} + ((m_{belt} + m_{load})R_2^2 + J_{cylinder})\frac{1}{red^2} \quad (1)$$

$$B_{em} = B_{mot} + B_{cylinder}\frac{1}{red^2} \quad (2)$$

Where m_{belt}, m_{load} are the masses of the conveyor belt and the transferring load, $J_{mot}, J_{cilinder}$ and $B_{mot}, B_{cilinder}$ are the moment of inertia and viscous friction coefficient of the motor and the cylinder on which the conveyor belt rests, R_2 is the radius of the cylinder and red is reduction ratio. Sizes with *em* in their index, symbolize emulated sizes such as emulated moment of inertia and emulated viscous friction coefficient used for describing the mechanical system of emulated multi-motor drive and designing the speed and tensile force controllers. The used controllers, which include current, speed and tensile force controllers that build multi-motor drive control system are designed using the technical optimum, symmetric optimum and pole adjustment methods respectively.

B. Model of a drive with a platform

The second model, model of a drive with a platform, is the model of a regulated multi-motor drive in which power for moving the platform comes from two electric motors that work together. In this case there is no tensile force controller and motors have just one speed controller which at the output gives required torques for motors according to their rated powers, what can be seen on figure 2.

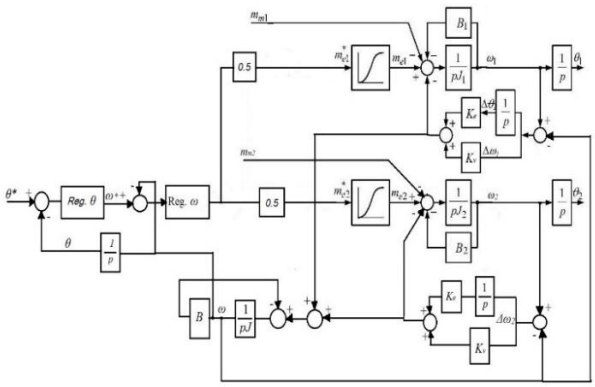


Figure 2. Simplified structural block diagram of a drive with a platform.

As in the previous model, the mechanical system is linearized to calculate the parameters of controllers, and can be determined from expressions 3 and 4 where moment of inertia and coefficient of viscous friction are calculated.

$$J_{em} = J_{mot1} + J_{mot2} + J_{platf}\frac{1}{red^2} \quad (3)$$

$$B_{em} = B_{mot1} + B_{mot2} + 1.2 \cdot B_{platf}\frac{1}{red^2} \quad (4)$$

Where $J_{mot1}, J_{mot2}, J_{platf}$ are moments of inertia and $B_{mot1}, B_{mot2}, B_{platf}$ are viscous friction coefficients of first, second motor and the platform. In this model, the imperfection of the transmission mechanism (air gapes between gears) are taken into account [2].

III. Dynamic Emulation Methods

In order to better understand the dynamic emulation [3], that is, the emulation of multi-motor drives in general, first is necessary to understand the composition and functioning of the dynamic emulator of the multi-motor drive. On figure 3. dynamic emulator for the two-motor drive is shown.

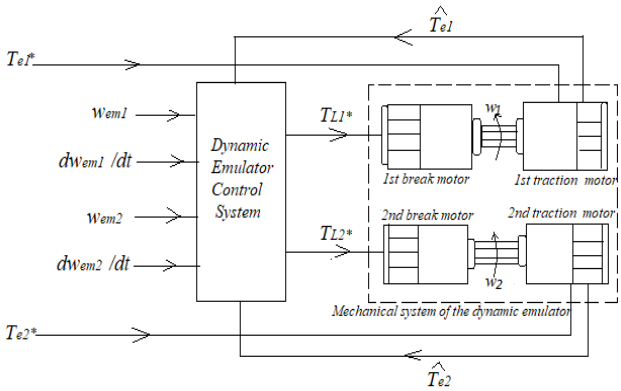


Figure 3. Simplified view of a dual-drive dynamic emulator.

The main idea behind any type of emulation is to achieve the equality between real speed and the emulated one. This is important because if traction motors (ones which are tested) have the same torques as if they are in the multi-motor drive, the load torque in real and testing conditions must be the same, which means same mechanical working conditions for motors. For load torque generation control system of dynamic emulator is used which inputs are estimated value of traction torque, emulated speed and emulated acceleration. Dynamic emulation is a concept that arose from static emulation by overcoming its problems. Static emulation is based on the direct determination of torque for the motor that emulates the load using the mechanical load characteristic and the Newton equation for rotational motion, what is showed by expression 5.

$$T_L = T_{Lem}(\omega) + (J_{em} - J)\frac{d\omega}{dt} + (B_{em} - B)\omega, \quad (5)$$

Where T_L, J, B are load torque, moment of inertia and viscous friction coefficient of dynamic emulator system, while T_{Lem}, J_{em}, B_{em} are mechanical characteristic, moment of inertia and viscous friction coefficient of emulated multi-motor drive. Velocity ω is the same for both of these systems. Disadvantages such as finding the velocity derivative and knowing the analytical expression for mechanical characteristic that are present in expression 5. are overcome by the use of dynamic emulation, that works especially well on highly dynamic systems with complex mechanical characteristic, which include many types of

multi-motor drives. Unlike static, dynamic emulation determines torque for motor that emulates load indirectly, using the control system. In this paper three methods of dynamic emulation are observed:

1. Feedforward compensation method
2. Method with second order structure
3. Method with **PI**-estimator

A. Feedforward compensation method

Feedforward compensation method [4] is based on the use of estimated parameters for the moment of inertia and the viscous friction coefficient of the motors participating in the dynamic emulator to achieve an equality between the actual and the emulated speed and to accelerate the dynamic response. In this method most of the resultant torque (difference between traction and braking torque) is achieved "in advance" by adding its estimated value to the output of the controller, which in this case is used for fine-tuning the resultant torque, thus the speed (figure 4.).

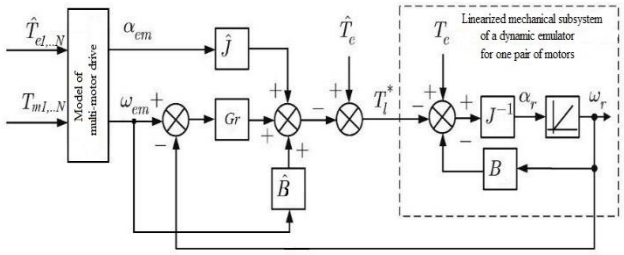


Figure 4. Srtuctural block diagram of a method with feedforward compensation.

In the implementation of this dynamic emulation method, a **PI** controller with transfer function given by expression 6 was used.

$$G_r = \frac{K_r(1 + pT_r)}{pT_r} \quad (6)$$

Where K_r is gain and T_r is a time constant of the controller. Based on the block diagram on figure 4 actual speed can be determined from the expression 7.

$$\omega = \frac{K_r + T_r K_r p + pT_r(\hat{B} + \hat{J}p)}{K_r + K_r T_r p + pT_r(B + Jp)}\omega_{em} - \frac{pT_r}{K_r + K_r T_r p + pT_r(B + Jp)}T_{ext} \quad (7)$$

From expression 7 the characteristic equation is obtain:

$$K_r + (K_r T_r + \hat{B}T_r)p + \hat{J}T_r p^2 = 0 \quad (8)$$

In expression 7 torque T_{ext} is a result of the unmodeled dynamics of the dynamic emulator. The **PI** controller is designed using a pole adjustment method by comparing the characteristic equation and the general form of the second-order equation ($\omega_n^2 + 2\xi\omega_n p + p^2 = 0$). On the basis of which the gain and the time constant of the controller are obtained:

$$K_r = 2\hat{J}\omega_n\xi - \hat{B} \quad (9)$$

$$T_r = \frac{2\xi\omega_n \hat{J} - \hat{B}}{\omega_n^2 \hat{J}} \quad (10)$$

Where $\hat{J}$ and $\hat{B}$ are the estimated values of the moment of inertia and the coefficient of viscous friction of the one pair of motors in the dynamic emulator. While ω_n and ξ are the natural frequency and damping coefficient.

B. Method with second order structure

Method with a second order structure [5] has a two-part structure, the first part is used to calculate an equivalent acceleration, something different from the emulated one to compensate the effect of unmodeled dynamics. While the second part is used to obtain the resultant torque that drives the dynamic emulator, based on the equivalent acceleration and estimated values of parameters of the mechanical system of the dynamic emulator (figure 5). As in the previous method, the **PI** controller with the transfer function given by expression 6 is used. Its parameters are determined based on the pole adjustment method analyzing the characteristic equation.

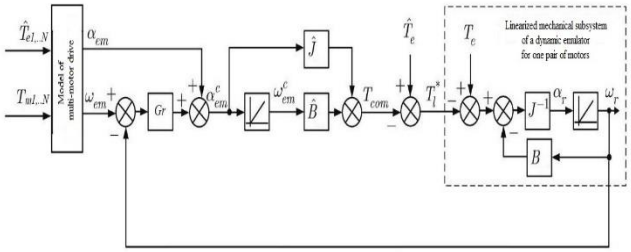


Figure 5. Srtuctural block diagram of a method with second order structure.

Characteristic equation can be obtained from the denominator of the speed transfer function determined by the block diagram on figure 5 and given by the expression:

$$\omega = \omega_{em}\frac{(K_r(1+pT_r)+p^2T_r)(\hat{J}p+\hat{B})}{(K_r(1+pT_r)+p^2T_r)(Jp+B)} - T_{ext}\frac{pT_r}{(K_r(1+pT_r)+p^2T_r)(Jp+B)} \quad (11)$$

Although in this case characteristic equation is a third-order equation, but not considering the known pole ($-\frac{B}{J}$) it becomes the second-order equation on which pole adjustment method can be applied. Also in expression 7 and 11 besides the estimated ones, the actual parameters of the mechanical system (*J, B)* of the dynamic emulator participate, but as only the estimated values of this parameters are known they will be used for finding the gain and the time constant of the controller. In this case

characteristic equation is simple and parameters of the controller can be found using only characteristic parameters of the dynamic response (ω_n, ξ).

$$K_r = 2\omega_n\xi \tag{12}$$

$$T_r = \frac{2\xi}{\omega_n} \tag{13}$$

C. Method with PI-estimator

Method with **PI**-estimator [6] is an upgrade of the previous one in the sense that two controllers are now used instead of one. Both of this controllers are **P** controllers and used speed error at the input. Also second one with a suitable integrator creates **PI**-estimator which at his end gives a torque correction for even better speed tuning of the dynamic emulator (figure 6). Like in the previous methods gains of the controllers will be found using pole adjustment method that will be applied on the characteristic equation.

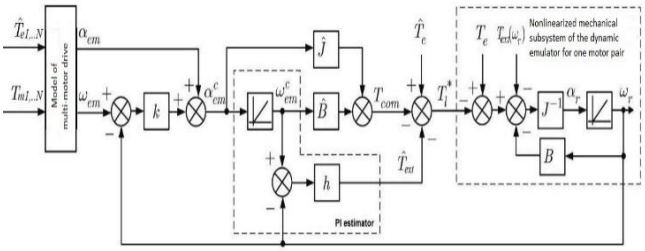


Figure 6. Srtuctural block diagram of a method with PI-estimator.

Based on the structural block diagram showed in the figure 6 speed can be determine by the expression:

$$\omega = \omega_{em}\frac{k(\hat{J}\,p+\hat{B})+h(k+p)+p(\hat{J}\,p+\hat{B})}{k(\hat{J}\,p+\hat{B})+h(k+p)+p(Jp+B)} - T_{ext}\frac{1}{k(\hat{J}\,p+\hat{B})+h(k+p)+p(Jp+B)} \tag{14}$$

In this case characteristic equation is a second-order equation given by expression 15.

$$\frac{k(\hat{B}+h)}{\hat{J}} + \frac{\hat{B}+h+k\,\hat{J}}{\hat{J}}\,p + p^2 = 0 \tag{15}$$

Comparing characteristic equation with a general form of the second-order equation gains of the controllers can be determined from expressions:

$$k = \omega_n\xi \pm \omega_n\sqrt{\xi^2-1} \tag{16}$$

$$h = \frac{\omega_n^2\,\hat{J}}{\omega_n\xi \pm \omega_n\sqrt{\xi^2-1}} - \hat{B} \tag{17}$$

In order to find the value of the gains quadratic equation had to be solved and for that there are two pairs of solutions. This also can be seen from the expressions 16 and 17 taking the plus or minus sign.

IV. Experimental Results

Experiment was implemented using a dSpace platform that translates a Simulink model into code in the **C** programming language and generates the appropriate control signals to the inverters through which the motors in dynamic emulator are fed. Experimental results will be shown for the first pair of motors in the dynamic emulator as the only pair that is physically realized considering limited resources in the laboratory, other pair of motors only exists as a model in Simulink. Both motors in this pair are induction motors that have the same parameters which are given in table I. The experimental results include a representation of the emulated and actual speed of rotation of the motor pair.

TABLE I
Motor parameters

Nominal power (W)	1500
Nominal speed (rpm)	2860
Nominal phase voltage (V)	230
Nominal phase current (A)	3.4
Stator resistance (Ω)	5.45
Stator leakage inductivity (mH)	11.8
Rotor resistance (Ω)	3.18
Rotor leakage inductivity (mH)	11.8
Moment of inertia (kgm^2)	0.0035
Coefficient of viscous friction (Nms/rad)	0.0022

Command signals sent to the inverters, signals received by the dSpace platform and the configuration of the mechanical system of the dynamic emulator are shown on figure 7.

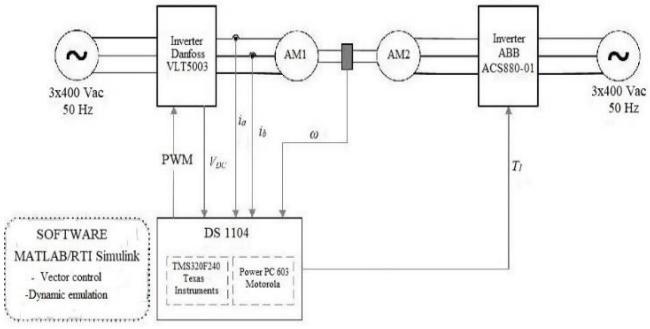


Figure 7. Schematic diagram of experimental facility.

A. Model of a drive with a conveyor belt

In this part of the paper experimental results that refer to the model of a drive with a conveyor belt are shown. They include representation of emulated and actual speed of motor pair in dynamic emulator for all three methods of dynamic emulation described in this paper.

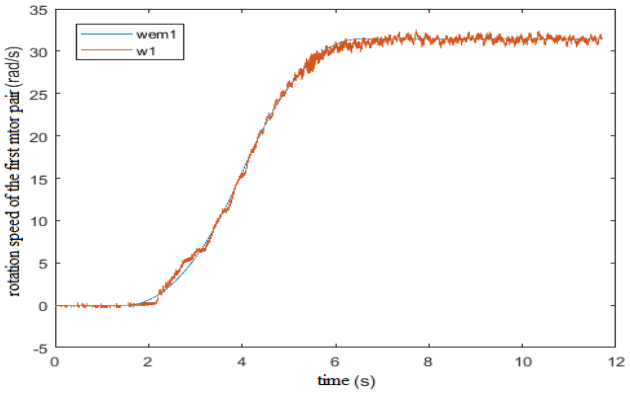


Figure 8. Real and emulated speed, feedforward compensation method.

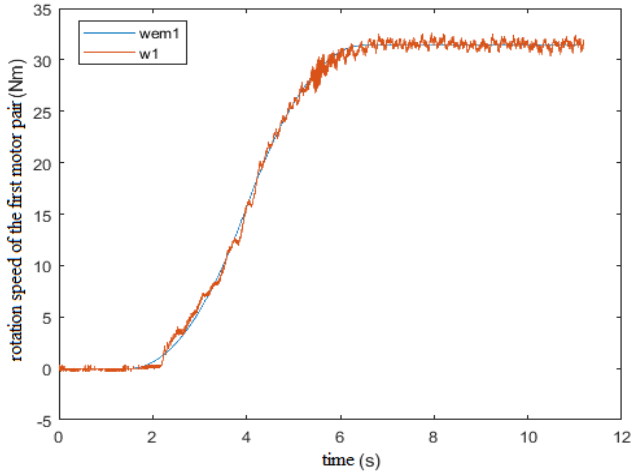


Figure 9. Real and emulated speed, method with second order structure.

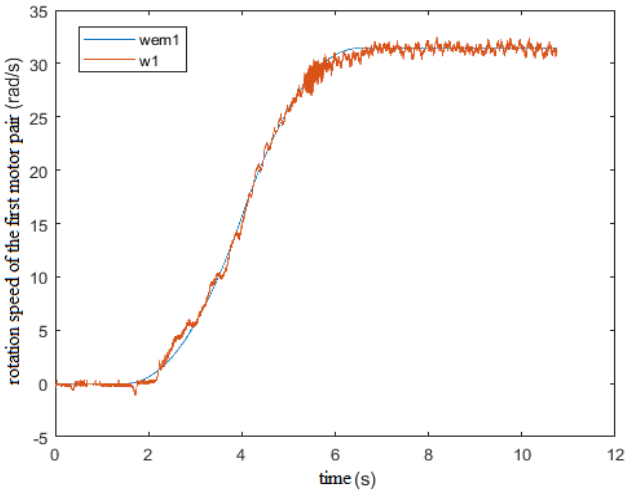


Figure 10. Real and emulated speed, method with PI-estimator.

Figures 8, 9 and 10 show that operation of dynamic emulator using described dynamic emulation methods is at a satisfactory level in every part of technological process. This include the part of the process that tightens the belt, accelerates the belt and transfers the load at a constant speed.

B. Model of a drive with a platform

As in the previous model of multi-motor drive, a comparison of the actual and emulated speed of the motor pair in the dynamic emulator for all three dynamic emulation methods was made here.

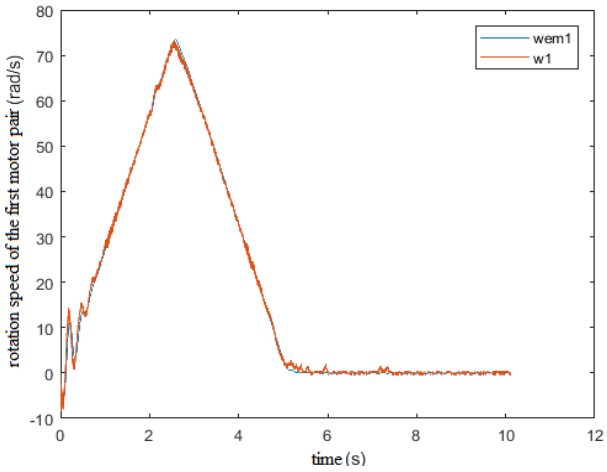


Figure 11. Real and emulated speed, feedforward compensation method.

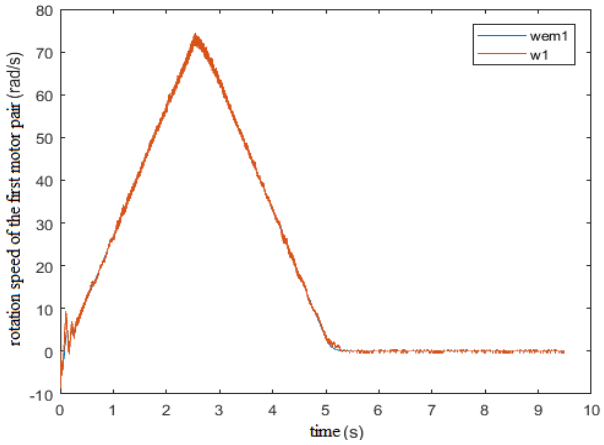


Figure 12. Real and emulated speed, method with second order structure.

Based on the figures 11, 12 and 13 it can be concluded that the emulation of the operation of the multi-motor drive with a platform for all three dynamic emulation methods was efficiently performed. This can be seen in the time of acceleration, deceleration as well as in the moment of slight oscillation of speed at the beginning caused by imperfections in the transmission mechanism.

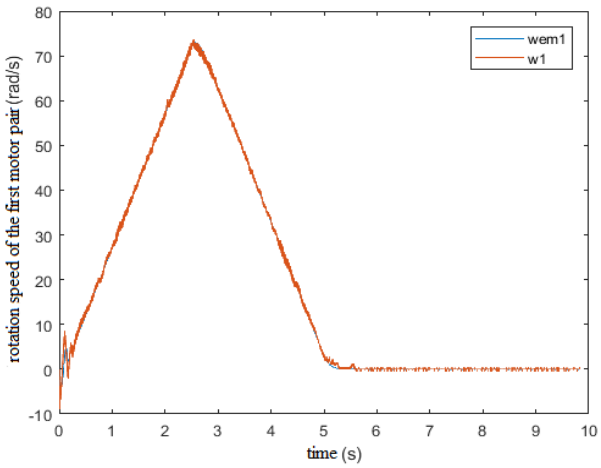


Figure 13. Real and emulated speed, method with **PI**-estimator.

V. Conclusion

Based on the experimental results obtained, it can be concluded that the used dynamic emulation methods are correct and efficient, since the monitoring between the desired, emulated, and the actual speed was achieved at a satisfactory level for both models of multi-motor drives. It should also be noted that the tracking between emulated and actual speed is slightly better with second and third emulation methods, which, unlike the feedforward method have a more complex structure that allows finer adjustments of load torque and thus the speed of rotation.

The presented experimental results are not ideal as some differences between the real and the emulated speed can be observed. This differences are the result of the unmodeled dynamics of the complex system such as dynamic emulator. First of all, that refers to the mechanical system, which in reality is not linear with speed, but is a more complex function. Also there are some imperfections in the estimation of torque and speed and the neglection of time delay in development of braking torque when designing the controllers entering the composition of dynamic emulation methods.

References

[1] Borislav Jeftenić, „Višemotorni električni pogoni”, Elektrotehnički fakultet Beograd, 2011.

[2] Karol Kyslan, Emil Kušnir, Viliam Fedak, Milan Lacko, František Durovsky „Dynamic Emulation of Mechannical Loads with Backlash Based od Rapid Control Prototyping”, 16th International Power Electronics and Motion Control Conference and Exsposition, Antalya, Turkey 2014.

[3] Miran Rodič, „Dinamična emulacija mehanskih bremen” Maribor, december 2004.

[4] Miran Rodič, Karel Jezernik, Mladen Trlep „A Feedforward Approach to the Dynamic Emulation of Mechanical Loads”, 35th Annual IEEE Power Electronics Specialists Conference Germany, 2004.

[5] Karol Kyslan, Viliam Fedak, František Durovsky, Padmanaban Sanjeevikumar, Miran Rodič „Design and Analysis of Torque Control for Load Drive with Dynamic Emulation”, 2017.

[6] Miran Rodič, Karel Jezernik, Mladen Trlep „Dynamic Emulation of Mechanical Loads – Approach Based on Nonlinear Control”, EPE 2005-Dresde

LED pahulja kontrolisana Android aplikacijom

Milan Stojanović, Stevan Živković

Sadržaj - U radu je prikazan postupak projektovanja i realizacije uređaja koji predstavlja deo ukrasne rasvete. Na uređaju se nalaze LE diode raspoređene u obliku pahulje, a uključuju se na osnovu komandi koje zadaje korisnik preko Android aplikacije. Uređaj je baziran na mikrokontroleru PIC18F45K22, Bluetooth modulu HC-05 i LED drajverima TLC5925. Prikazan je postupak projektovanja *hardware*-a i *software*-a uređaja, kao i realizacija kućišta uređaja 3D štampačem.

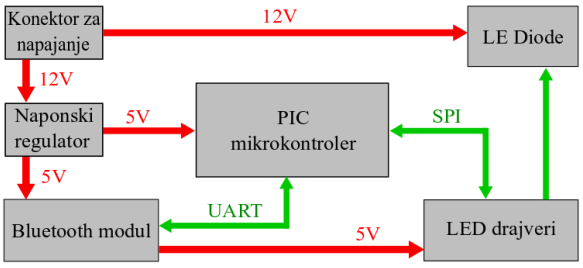
I. Uvod

Razvoj elektronike u mnogome je doprineo promeni načina zivota ljudi svuda u svetu. Ta promena se odnosi kako na uređaji koji olakšavaju svakodnevne poslove, tako i na uređaje koji se koriste u slobodno vreme. U drugu grupu spadaju i uređaji ukrasne rasvete. Tako su vremenom stari praznični ukrasi zamenjeni električnim koji su daleko funkcionalniji i efektniji.

U radu je prikazan uređaj koji se može koristiti kao deo praznične rasvete. Napravljen je od LE dioda koje su raspoređene u obliku pahulje sačinjene od nekoliko segmenata. Kontrola same pahulje vrši se od strane korisnika Android aplikacijom. Pod kontrolom se podrazumeva redosled uključivanja segmenata, brzina promene efekata kao i jačina osvetljenosti.

II. Električna šema uređaja

Projektovani uređaj se može predstaviti kao sklop električnih blokova povezanih odgovarajućim električnim vezama (slika 1).



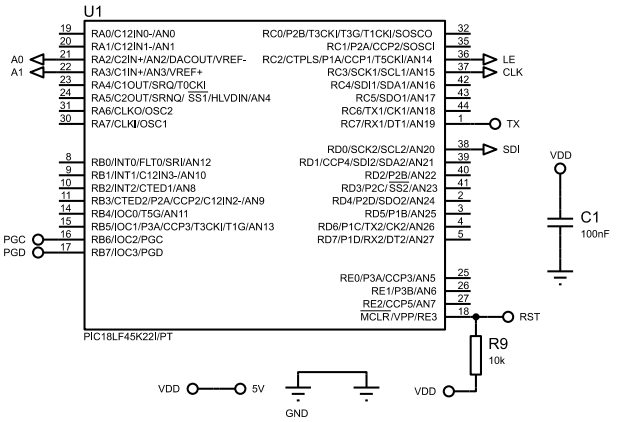
Slika 1. Blok šema projektovanog uređaja.

M. Stojanović, student prve god. Doktorskih akademskih studija na smeru Nanotehnologije i mikrosistemi na Elektronskom fakultetu u Nišu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milan.stojanovic@elfak.ni.ac.rs

S. Živković, student Master akademskih studija na smeru Računarstvo i informatika na Elektronskom fakultetu u Nišu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: Zivkovic_Stevan@yahoo.com

A. Mikrokontroler

Centralna komponenta uređaja je mikrokontroler PIC18F45K22 [1]. Na osnovu upisanog koda, koji je napisan u programskom okruženju Proton IDE, mikrokontroler upravlja ostalim blokovima, tj. komponentama. Na slici 2 prikazana je električna šema bloka sa mikrokontrolerom.



Slika 2. Električna šema bloka sa mikrokontrolerom.

Kao što se može videti, u ovom bloku se pored mikrokontrolera koriste još dve komponente. Keramički *baypass* kondenzator na pinovima napajanja samog mikrokontrolera i otpornik koji sprečava neželjeno restartovanje mikrokontrolera. Ono što treba primetiti na električnoj šemi su labele preko kojih se ostvaruju električne veze mikrokontrolera sa ostatkom uređaja.

Do mikrokontrolera treba dovesti napon napajanja od 5 V i masu koja je zajednička za ceo uređaj. Prilikom projektovanja električne šeme u programu *Proteus* treba obratiti pažnju na skrivene pinove komponenata. Kod mikrokontrolera to su pinovi za napajanje označeni sa VDD i GND. Iz tog razloga su povezane labele 5V i VDD kao i GND sa masom uređaja.

Preko pinova PGD, PGC, RST, uz masu i napajanje vrši se programiranje mikrokontrolera preko PICkit3 uređaja [2].

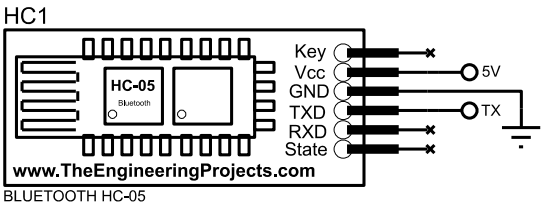
Preko labela LE, CLK, SDI se ostvaruje SPI komunikacija između mikrokontrolera i drajvera. Signali A0 i A1 služe za kontrolu osvetljenja što će biti objašnjeno u nastavku.

Razmena podataka između mikrokontrolera i *Bluetooth* modula vrši se preko UART komunikacije. U ovom slučaju je dovoljno koristiti samo RX pin mikrokontrolera jer se prenos podataka vrši samo u smeru od *Bluetooth* modula ka mikrokontroleru. Na šemi je ovaj

pin povezan sa labelom TX jer je povezan na TX pin *Bluetooth* modula.

B. Bluetooth modul

Da bi komunikacija između uređaja i mobilnog telefona bila moguća neophodno je da uređaj ima *Bluetooth* modul. Pri realizaciji uređaja iskorišćen je modul HC-05 [3]. Ovaj modul se može konfigurisati da radi u širokom opsegu brzine prenosa podataka (6900 bps do 460800 bps), a u samom uređaju se ta komunikacija vrši brzinom od 19200 bps. Na slici 3 prikazan je električni simbol modula HC-05.



Slika 3. Električni simbol *Bluetooth* modula HC-05.

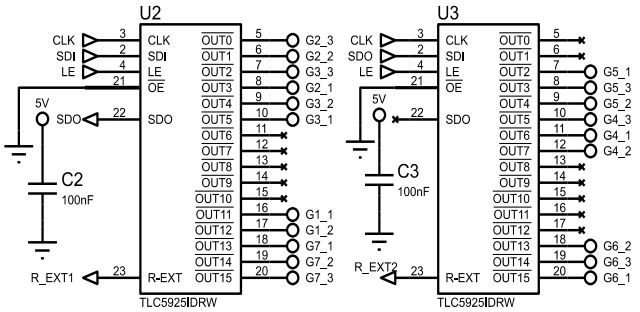
C. LED drajveri

LED drajveri su integrisana kola namenjena za kontrolu LED displeja i uređaja kod kojih se prikaz informacija vrši preko LE dioda. Za realizaciju uređaja koji se opisuje u radu korišćeni su LED drajveri TLC5925 [4]. TLC5925 su u stvari 16-bitni pomerački registri sa serijskim ulazom i paralelnim izlazom. Svaki od izlaza obezbeđuje da kroz diodu (grupu dioda) koja je na njega povezana teče konstantna struja. Posebnu pogodnost predstavlja to što se jačina struje kroz izlaze drajvera može odrediti eksternim otpornikom (R_{ext}) na osnovu jednačine 1:

$$I_{OUT} = \frac{1.21V}{R_{ext}} \cdot 18 \qquad (1)$$

Na taj način se preko jednog otpornika može regulisati jačina svetlosti koju emituju diode.

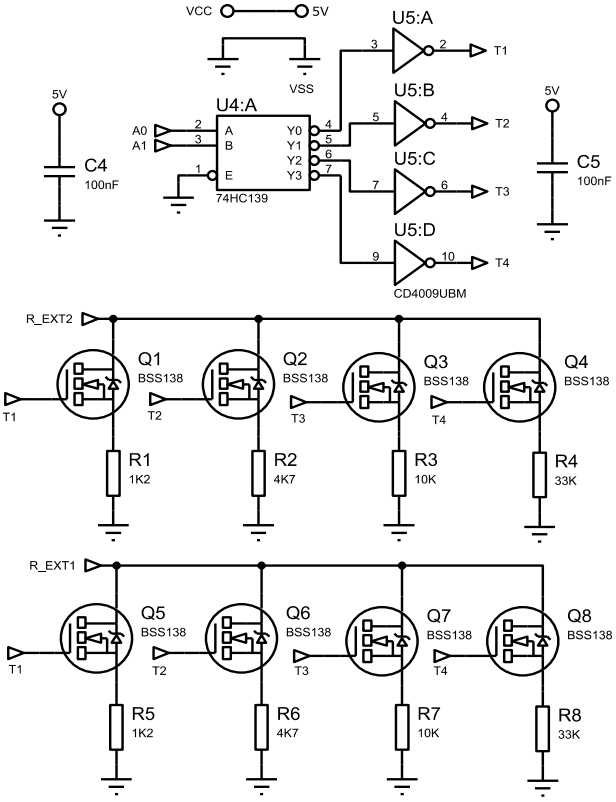
Na slici 4 prikazan je deo električne šeme sa LED drajverima. Kondenzatori C1 i C2 u ovom kolu imaju istu ulogu kao kondenzator C1 prikazan na slici 2.



Slika 4. Deo električne šeme sa LED drajverima.

Kao što je ranije navedeno podaci se sa mikrokontrolera na drajvere prenose preko SPI komunikacije. Oba drajvera imaju zajedničke signalne linije sa mikrokontrolerom. Pri rastućoj ivici CLK signala se podatak na SDI liniji sa mikrokontrolera upisuje u drajver. Prvih 16 bitova se upisuju u drajver U2. Pri upisu sledećih 16 bitova, podatak koji je upisan u dajver U2 preko SDO pina prelazi u drajver U3, a u drajver U2 se upisuje novih 16 bitova. Po završetku upisa se preko signala LE dovodi pozitivan impuls kojim se podaci upisani u oba drajvera prosleđuju na njihov izlaz.

Preko signala R_EXT1 i R_EXT2 se reguliše jačina struje na izlazima drajvera električnim kolom sa slike 5.



Slika 5. Električna šema kola za regulaciju jačine osvetljenosti.

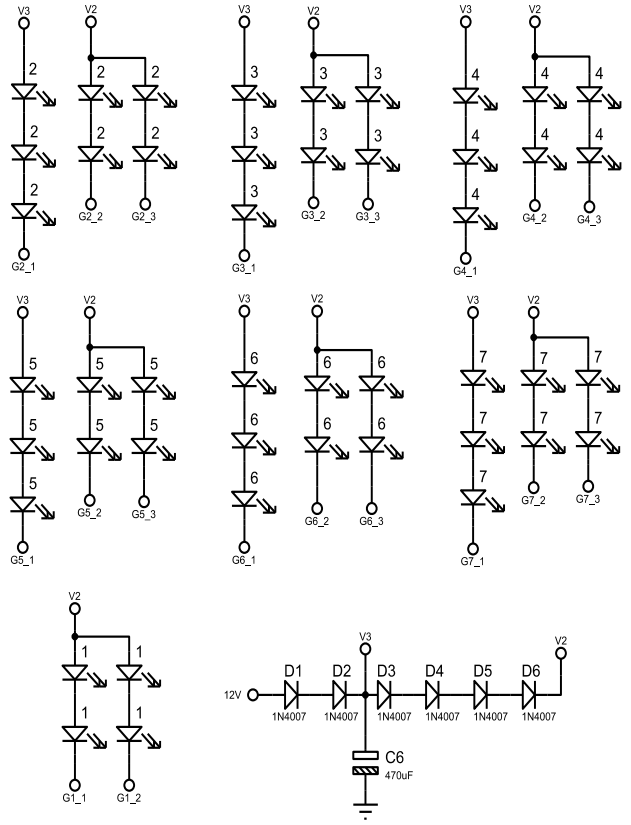
C4 i C5 su *baypass* kondenzatori. Signalima A0 i A1, koji dolaze sa mikrokontrolera, se aktivira jedan od izlaza dekodera. Signal sa aktivnog izlaza dekodera se zatim invertuje i dovodi na *gate* odgovarajućeg para N-kanalnog MOS tranzistora. Na taj način se na pinove za regulaciju jačine izlazne struje povezuje jedan od četiri para otpornika i podešava željeni nivo osvetljenosti.

Iz istog razloga kao kod mikrokontrolera treba povezati labelu VCC na 5V kao i VSS sa masom uređaja.

D. LE diode

Na prvi pogled, može se reći da diode predstavljaju glavni deo uređaja s obzirom da je to deo koji korisniku pruža povratnu informaciju. Uređaj je realizovan sa LE

diodama koje emituju belu svetlost [5]. Diode su raspoređene u grupe koje su povezane na izlaze drajvera formirajući segmente (slika 6). Pahulju čini ukupno sedam segmenata koji istovremeno ili posebno uključivati.



Slika 6. Električna šema i formiranje grupa i segmenata LE dioda.

Labela G1_1 ukazuje na prvu grupu iz prvog segmenta i povezana je na jedan od izlaza drajvera. Kada se u drajver upiše logička jedinica/nula na poziciji koja odgovara pinu na koji je povezana ova labela biće uključena/isključena ova grupa dioda. Sve ostale grupe i segmenti se uključuju/isključuju po istom principu.

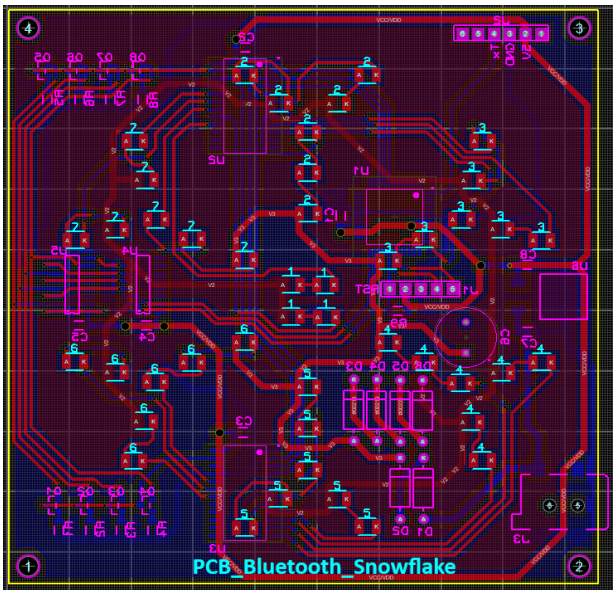
Silicijumskim diodama D1-D6 se ulazni napon dovodi do potrebne vrednosti za napajanje određene grupe. V3 predstavlja napon kojim se napajaju grupe sa tri redno vezane LE diode, a V2 napon napajanja grupa sa dve redno vezane LE diode. Elektrolitski kondenzator C6 služi da održi diode uključenim ukoliko dođe do kratkotrajnih nestabilnosti napona napajanja.

E. Napajanje

Uređaj se napaja preko eksternog napajanja od 12 V. Ovaj napon se koristi za napajanje LE dioda, a potrebnih 5 V za napajanje ostalih komponenata realizovano je preko naponskog regulatora LM7805.

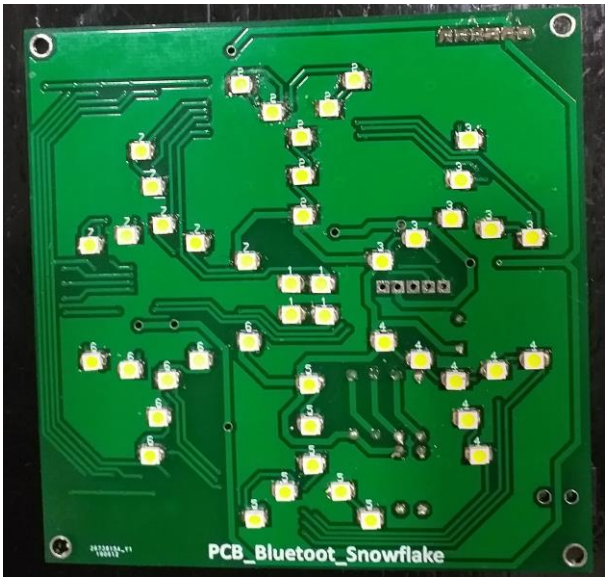
III. PROJEKTOVANJE ŠTAMPANE PLOČE

Nakon realizacije električne šeme izvršeno je raspoređivanje komponenata na štampanoj ploči i rutiranje električnih veza. Na slici 7 prikazan je izgled *layout*-a štampane ploče nakon rutiranja.

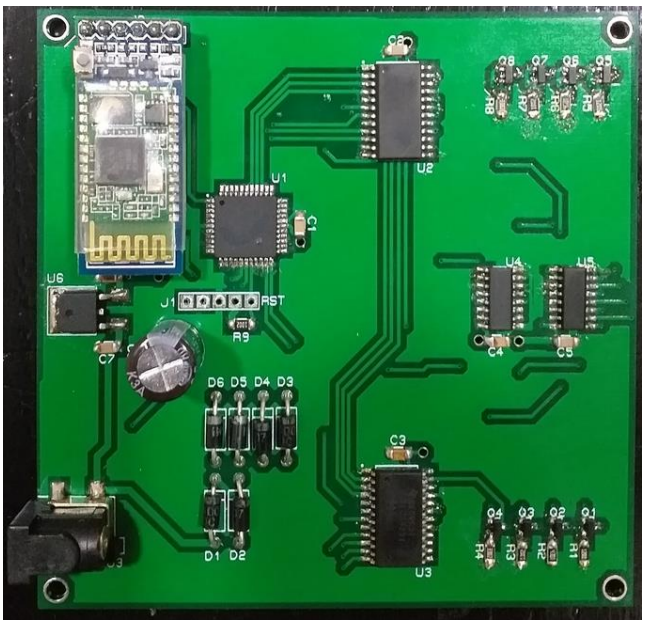


Slika 7. Štampana ploča nakon rutiranja (*layout*).

Na prednjoj (*top*) strani ploče nalaze se LE diode, dok je sva ostala elektronika smeštena na zadnjoj (*bottom*) strani. Po završetku rutiranja GERBER fajlovi su poslati na fabrikaciju, a finalni izgled ploče nakon fabrikacije i lemljenja komponenata prikazan je na slikama 8 (prednja strana) i 9 (zadnja strana).



Slika 8. Završena štampana ploča – prednja strana.



Slika 9. Završena štampana ploča – zadnja strana.

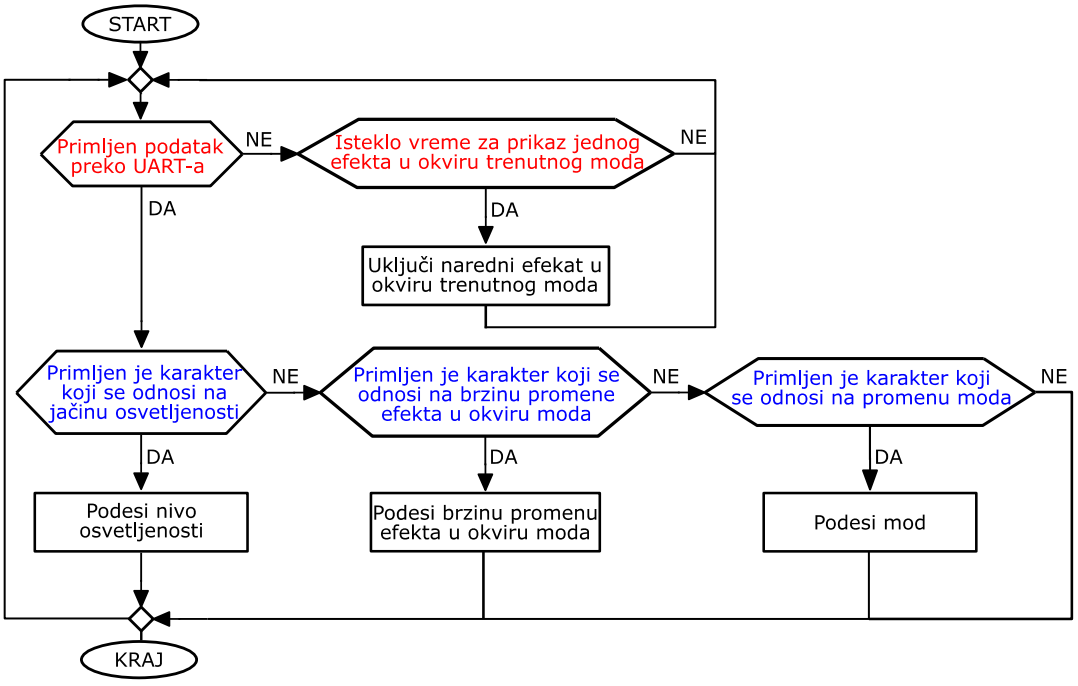
Pored komponenata na ploči su dodate i rupe potrebne da se štampana ploča pričvrsti za kućiste.

IV. ALGORITAM RADA UREĐAJA (MIKROKONTROLERA)

Sva funkcionalnost uređaja određena je mikrokontrolerom, odnosno kodom koji je upisan u njega. Program za mikrokontroler napisan je u *Proton IDE software*-u. Na slici 10 prikazane su glavne celine algoritma koji odgovara programskom kodu mikrokontrolera.

Nakon uključivanja uređaja sve diode su isključene i čeka se prijem podatka sa *Bluetooth* modula preko UART komunikacije. Mikrokontroler podatke sa modula prihvata preko *interrupt* rutine. Na taj način je omogućeno da uređaj u bilo kom trenutku prihvati novi podatak od korisnika. Kada dodje do prijema podatka, preko vrednosti određene promenljive se glavnoj petlji signalizira da je podatak primljen i da treba da pristupi obradi istog.

Druga *interrupt* rutina koja se koristi zasniva se na merenju vremena. Ako je isteklo vreme, postavljeno upisom podataka u odgovarajuće registre mikrokontrolera, preko promenljive se glavnoj petlji signalizira da treba da se izvrši promena efekta u okviru moda.



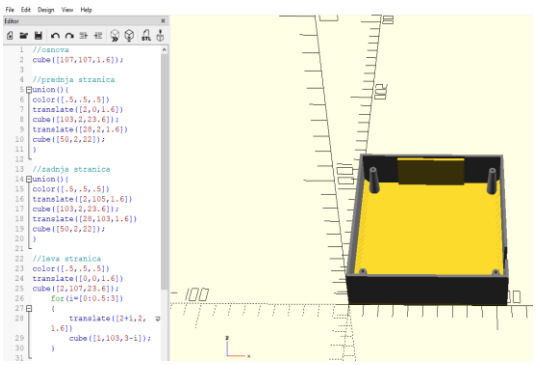
Slika 10. Algoritam programskog koda mikrokontrolera.

Kada je novi podatak primljen ispituje se njegova vrednost i na osnovu nje izvršava jedan od tri dela programa. Prvi se odnosi na promenu jačine osvetljenosti. Drugi je povezan sa vremenskim *interrupt*-om jer se u njemu postavljaju vrednosti koje ukazuju na brzinu ulaska u *interrupt* rutinu, a samim tim i brzina promene efekta u modu. Treći deo koda se izvršava ukoliko je primljen karakter za promenu moda. Mod se odnosi na prikaz, a efekat se odnosi na trenutno stanje prikaza tj. moda. Npr. jedan od modova je uključivanje segmenata s leva na desno pri čemu se prethodno uključeni segment isključuje, a trenutno uključeni segment u tom modu predstavlja efekat.

IV. IZRADA KUĆIŠTA UREĐAJA

Nakon testiranja štampane ploče i *software*-a sledeći korak u realizaciji uređaja je projektovanje kućišta. Poslednjih godina, u razvoju kućišta i mehaničkih delova električnih uređaja, sve veći uticaj imaju 3D štampači.

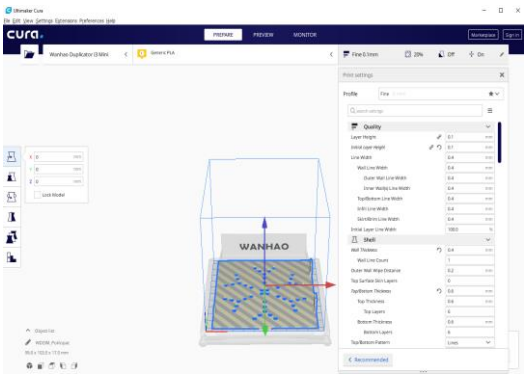
Za projektovanje kućišta uređaja korišćen je *Open SCAD software* [6]. Kučište uređaja se sastoji iz dva dela, osnove sa stranicama i odstojnicima za štampanu ploču i poklopca. Na slici 11 prikazan je izgled pomenutog *software*-a pri projektovanju osnove kućišta.



Slika 11. Projektovanje osnove kućišta u *Open SCAD software*-u.

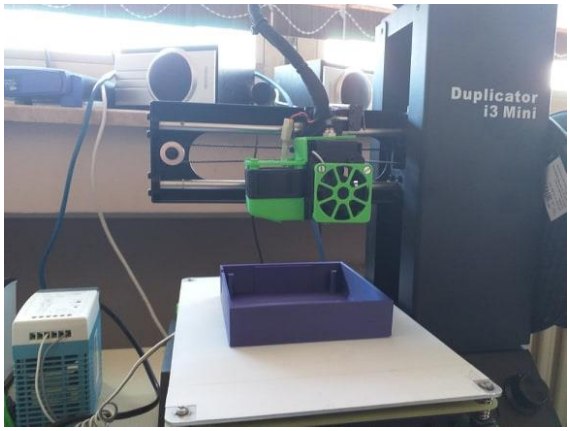
Poklopac kućišta sadrži rupe koje se poklapaju sa rasporedom LE dioda i projektuje se na osnovu *layout*-a štampane ploče. U nekom od software-a za crtanje (*Inscape*) se sa svakom LE diodom centrira krug željenog poluprečnika. Na crtež se potom ostavljuju samo krugovi i linije ivica ploče. Crtež treba sačuvati kao fajl sa ekstenzijom *.dxf* koji se uvozi u *Open SCAD* i tako se dobija poklopac kućišta.

Sledeći korak je da se fajlovi iz *Open SCAD*-a prilagode za 3D štampač. Jedan od software-a za to je *Ultimaker Cura* [7]. U ovom *software*-u (slika 12) se odabira 3D štampač koji se koristi, brzina štampanja, debljina slojeva i ostali parametri za štampanje.

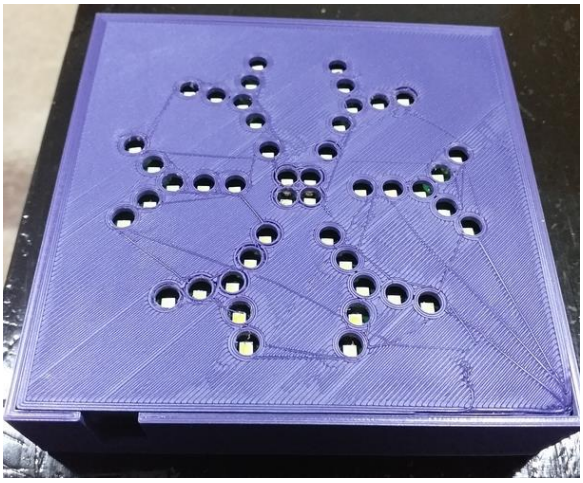


Slika 12. Priprema fajla za 3D štampač (poklopac).

Za izradu kućišta uređaja korišćen je štampač *Wanhao i3 Mini*. Na slici 13 prikazan je proces štampanja kućišta, a na slici 14 uređaj u kućištu.



Slika 13. Štampanje kućišta 3D štampačem *Wanhao i3 Mini*.



Slika 14. Izgled uređaja.

V. REALIZACIJA ANDROID APLIKACIJE

Kao što je navedeno ranije, završni deoprojektovanja uređaja predstavlja korisnička *Android* aplikacija. Aplikacija je razvijena u *Android Studio* okruženju korišćenjem *native* i *third-party* biblioteka. Na slici 15 prikazan je logo realizovane aplikacije.

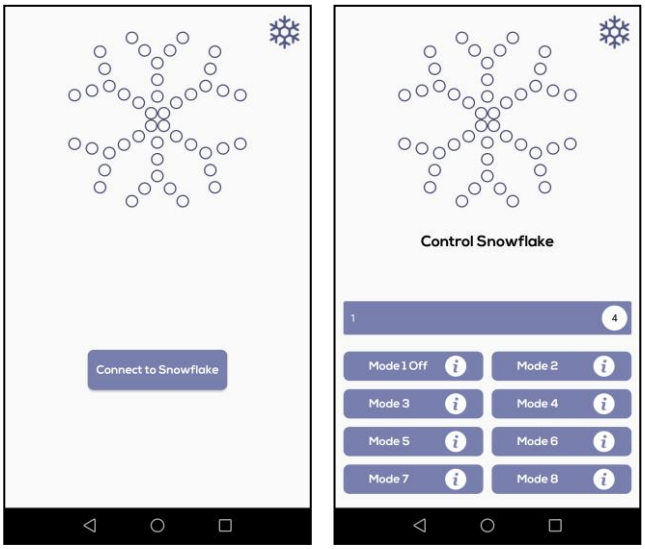


Slika 15. Logo realizovane *Android* aplikacije.

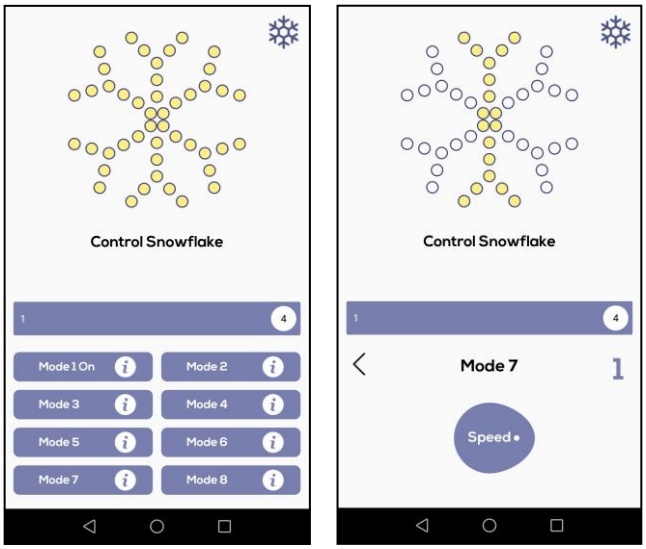
Cilj same aplikacije jeste kontrola hardvera uređaja preko *Bluetooth* komunikacije slanjem pojedinačnih naredbi, kao i prikaz trenutno izabranog moda na ekranu telefona. Pri pokretanju aplikacije korisnik mora da se konektuje na svoj uređaj (levi prozor na slici 16). Nakon uspešne konekcije, pojavljuje se kontrolni panel u kome se

nalaze svi modovi koji se mogu izabrati. Korisnik može u bilo kom trenutku kontrolisati nivo osvetljenosti sistema, a pri odabiru pojedinačnog moda dobija i mogućnost podešavanja brzine prikaza odabranog moda.

Svaki mod ima svoje *info* polje pri čijem kliku korisnik dobija tačan opis tog moda. Na slici 18 dat je primer opisa za **Mod 5.**



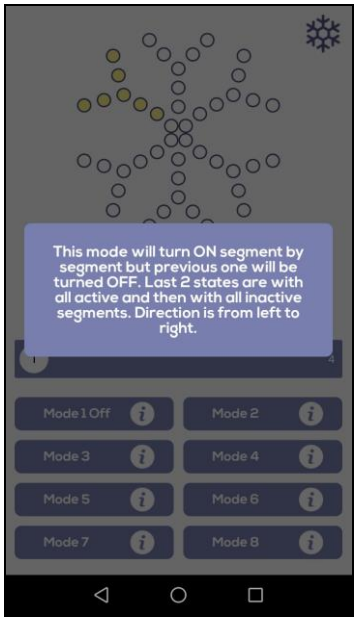
Slika 16. Izgled početnog ekrana pre i nakon konekcije korisnika.



Slika 17. Izgled ekrana aplikacije pri aktiviranom prvom i sedmom modu.

Na desnom prozoru slike 16 prikazan je kontrolni panel sa svim dostupnim modovima i u vrhu panela nalazi se *fluid slider* za podešavanje nivoa osvetljenosti. Dostupni modovi su:

- **Mod 1** - Mod u kome se sve diode mogu uključiti ili isključiti (levi prozor slike 17).
- **Mod 2** - U ovom modu se diode naizmenično uključuju i isključuju, tzv. efekat blinkanja.
- **Mod 3** – U trećem modu se segmenti uključuju jedan za drugim od vrha pahljulje s leva na desno. Na kraju se svi segmenti isključuju.
- **Mod 4** - Ovaj mod je istovetan modu 3 sa suprotnim smerom uključivanja segmenata.
- **Mod 5** - Mod u kome se segmenti uključuju sleva na desno pri čemu se prethodni segment isključuje. Na kraju se svi segmenti uključuju pa isključuju.
- **Mod 6** - Mod 6 je istovetan modu 5 u suprotnom smeru uključivanja segmenata.
- **Mod 7** - Pretposlednji mod prikazuje po dva suprotna segmenta dok je centralni segment uvek uključen. Segmenti se uključuju s leva na desno, a posle svakog para uključenih segmenata sledi period u kome su svi segmenti isključeni (desni prozor slike 17).
- **Mod 8** - Poslednji mod je istovetan mod-u 7 sa suprotnim smerom uključivanja segmenata.



Slika 18. Prikaz ekrana aplikacije sa informacijama o radu moda.

VI. ZAKLJUČAK

Nakon testiranja uređaja, utvrđeno je da uređaj u električnom pogledu funkcioniše bez ikakvih nedostataka. Nema neželjenog grejanja, a prenos podataka sa mobilnog telefona korisnika se na uređaj vrši bez ikakvih problema

(nema kašnjenja niti uticaja šumova što može dovesti do pogrešnog očitavanja primljene komande).

Što se daljeg razvoja tiče, jedna od mogućnosti za unapređenje uređaja je realizacija napajanja pomoću baterija. Na taj način bi se omogućila mobilnost uređaja ali bi se povećale njegove dimenzije.

Realizovano kućište napravljeno je od plastike koja je osetljiva na temperaturne promene. Da bi bilo moguće koristiti uređaj pri širem opsegu temperatura kućište uređaja treba napraviti od materijala koji je otporniji na promenu spoljšnjih uslova.

ZAHVALNICA

Autori se zahvaljuju docentu dr Ljubomiru Vračaru na savetima i pomoći oko realizacije kućišta uređaja. Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026.

LITERATURA

[1] „PIC18(L)F2X/4XK22, 28/40/44-Pin, Low-Power, High-Performance Microcontrollers with XLP Technology”, Microchip. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/40001412G.pdf
[2] „PICkit™ 3 In-Circuit Debugger”, Dostupno na: https://www.microchip.com/Developmenttools/ProductDetails/PG164130
[3] „HC-05, Bluetooth to Serial Port Module”, ITead Studio.
[4] „TLC5925 Low-Power 16-Channel Constant-Current LED Sink Driver”, Datasheet, Texas Instruments.
[5] „PLCC 3528 0.06W White”, EDISON Lighting Design Manufacturing Service, Datasheet
[6] Dostupno na: https://www.openscad.org/
[7] Ultimaker Cura. Dostupno na: https://ultimaker.com/software/ultimaker-cura

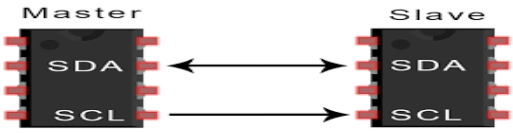
Analiza *I2C* protokola korišćenjem logičkog analizatora

Stefana Zogović, Milan Danković

Sadržaj– Korišćenjem logičkog analizatora vršeno je ispitivanje *I2C* protokola. Rad sadrži teorijske postavke o *I2C* protokolu, kao i objašnjenje i rezultate eksperimenta urađenog u svrhu ispitivanja načina funkcionisanja ovog protokola.

I. UVOD

I2C predstavlja sinhronu serijsku magistralu sa mogućnošću da više *Master*i *Slave* uređaja budu povezani istovremeno ali nema istovremene komunikacije između više od 2 uređaja. *I2C* komunikacija je, poput *SPI* (*Serial Peripheral Interface*) interfejsa, predviđena samo za komunikaciju između uređaja koji se nalaze na malom rastojanju. Poput RS-232, *I2C* koristi samo 2 linije za razmenu informacija.



Slika 1. *I2C* komunikacija.

I2C komunikacija koristi liniju predviđenu za takt (SCL – *Serial Clock Line*) i liniju predviđenu za podatke (SDA – *Serial Data Line*) (slika 1.) Karakteristično za *I2C* jeste to što je to konfiguracija sa OC/OD (*open-collector/open-drain*), što znači da je preko *pull-up* otpornika neophodno obe linije vezati za napajanje. Vrednost ovih otpornika se kreće od 1 kΩ do 10 kΩ, u zavisnosti od broja uređaja vezanih na linije.

Naponi sa kojima *I2C* funkcioniše jesu +5 V i +3.3 V. Za razliku od SPI, *I2C* nije ograničen brojem *Master* i *Slave* uređaja, a mogu se smenjivati između poruka, nakon što je STOP bit poslat.

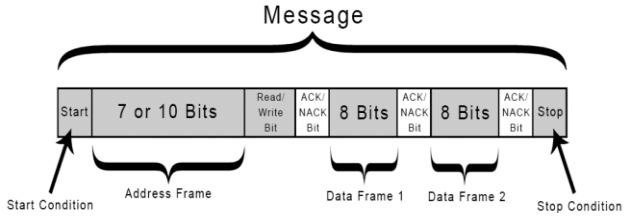
II. *I2C* PROTOKOL

S obzirom da se svi *Master* i *Slave* uređaji vezuju i za SDA i za SCL liniju, bilo je neophodno uvesti određena pravila na osnovu kojih se može razlikovati koje podatke šalju *Master* a koje *Slave* uređaji na zahtev *Master*-a. Ta pravila predstavljaju *I2C* protokol.

I2C protokol definiše sledeća pravila:

- *Master* počinje i završava komunikaciju;
- Podaci se postavljaju na silaznu ivicu a čitaju na uzlaznu ivicu takta;
- Svi podaci koji se šalju su 8-bitni gde se nakon svakog bajta šalje ACK (*acknowledged*) bit.

Master počinje komunikaciju slanjem START bita. Nakon toga sledi 7-bitna adresa *Slave*-a kome se *Master* obraća, R/$\overline{W}$ bit koji, u zavisnosti od toga da li je '0' ili '1', određuje da li će se u registre *Slave* uređaja vršiti upis podataka ili će iz njih podaci biti čitani. Posle R/$\overline{W}$ bita, uređaj koji prima podatke šalje ACK bit kao potvrdu da je uspešno primio podatke koji su mu poslati, a zatim se šalju bajtovi podataka i to tako da se prvo šalje MSB (*Most Significant Bit*) a nakon svakog bajta ACK bit. Ukoliko uređaj koji sluša pošalje NACK (*not acknowledged*), to znači da nije uspeo da primi poruku koja mu je poslata. Nakon poslednjeg poslatog bajta podataka i primljenog ACK bita, *Master* šalje STOP bit čime se komunikacija između *Master* i *Slave* uređaja završava (kao što je prikazano na slici 2.).



Slika 2. Izgled poruke prema *I2C* protokolu.

A. START bit

Kako bi otpočeo komunikaciju, *Master* uređaj drži SCL liniju u stanju HIGH a SDA liniju spušta u stanje LOW (slika 3.). Na ovaj način su svi *Slave* uređaji obavešteni da prenos podataka uskoro počinje. Ukoliko je više *Master* uređaja vezano na magistralu i više njih želi da otpočne komunikaciju, onaj *Master* uređaj koji prvi pošalje START bit preuzima kontrolu nad magistralom.

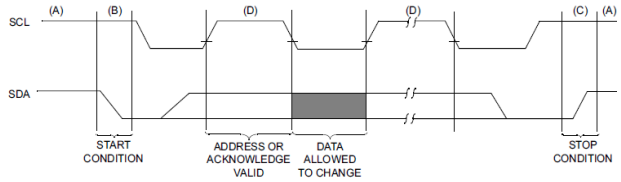
B. Adresa Slave uređaja

Adresa *Slave* uređaja se uvek prva šalje kod svake nove komunikacijske sekvence. Za 7-bitnu adresu prvo se šalje MSB pa ostali bitovi adrese, a zatim slede R/$\overline{W}$ bit koji

Stefana Zogović, diplomirani inženjer Elektrotehnike i računarstva na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, A.Medvedeva 14, 18000 Niš, Srbija, E-mail: stefana.zogovic@elfak.rs

Milan Danković, student druge godine Softverskog inzenjerstva na Elektrotehničkom fakultetu Univerziteta u Beogradu, Bulevar kralja Aleksandra 73, 11000 Beograd, Srbija, E-mail: dankovicmilan@gmail.com

pokazuje da li se obavlja čitanje ('1') ili upis ('0') podataka i na kraju ACK bit potvrde od strane adresiranog *Slave*-a (slika 4.).



Slika 3. Stanja SDA i SCL linija tokom slanja poruke.

Field:	S	I²C address field							R/W'	A
Type	Start	Byte 1								ACK
Bit position in byte X		7	6	5	4	3	2	1	0	
7-bit address pos		7	6	5	4	3	2	1		
Note		MSB						LSB	1 = Read 0 = Write	

Slika 4. Izgled 7-bitnog adresnog paketa.

Kod uređaja sa 10-bitnom adresom, adresa se šalje u vidu 2 adresna paketa. Prvi paket ima oblik 11110xyz, pri čemu su: x – MSB adrese *Slave*-a, y – 9. bit adrese *Slave*-a a z – R/W bit. Nakon ovoga odmah se šalje preostalih 8 bitova adrese *Slave*-a (slika 5.).

Field:	S	10-bit mode indicator					Upper addr		R/W'	A	Lower address field							
Type	Start	Byte 1								ACK	Byte 2							
Bit position in byte X		7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
Bit value		1	1	1	1	0	X	X	X		X	X	X	X	X	X	X	X
10-bit address pos							10	9			8	7	6	5	4	3	2	1
Note		Indicates 10-bit mode					MSB		1 = Read 0 = Write									LSB

Slika 5. Izgled 10-bitnog adresnog paketa.

Takođe, postoje i dve grupe adresa koje su kod *I2C* protokola rezervisane za specijalne funkcije.

Da bi adresa bila validna neophodno je da SDA linija bude stabilna za vreme HIGH perioda SCL linije, a promene na ovoj liniji moraju da se dešavaju samo tokom LOW perioda SCL linije (slika 3.).

C. Paketi podataka

Podaci se smeštaju na SDA liniju od strane *Master* ili adresiranog *Slave* uređaja, u zavisnosti od vrednosti R/$\overline{W}$ bita. Važi da su podaci validni samo ukoliko je SDA linja stabilna za vreme trajanja HIGH perioda SCL linije, a promene se dešavaju tokom LOW perioda linije takta (slika 3.).

D. STOP bit

Kada su svi paketi poslati, *Master* generiše STOP bit. STOP bit je uslovljen prelaskom SDA linije sa LOW na HIGH nakon što SCL linija pređe sa LOW na HIGH i ostane u ovom stanju (slika 3.).

III. OPIS APARATURE

Kako bi praktično potvrdili pravila koja važe za *I2C* protokol, bilo je neophodno u laboratoriji izvesti eksperiment koji će, kao rezultat, na ekranu logičkog analizatora dati poruku poslatu primenom ovog protokola. Za ovaj eksperiment korišćena su dva razvojna okruženja *Arduino Uno* i logički analizator Tektronix TLA5202B [1-5].

A. Razvojno okruženje Arduino Uno

Razvojno okruženje *Arduino Uno* predstavlja *open-source* okruženje sa ATmega328P mikrokontrolerom, razvijeno od strane Arduino.cc. *Open-source* znači da ovo okruženje predstavlja slobodno korišćenje softvera, bez potrebe za licencom.

Karakterisitike *Arduino Uno* okruženja jesu da je to okruženje koje koristi ATmega328P mikrokontroler i radi pri naponu od 5 V. Ulazni napon se kreće u opsegu od 7 V do 20 V, dok DC struja svakog I/O pina iznosi 20 mA. Na ploči se nalaze 14 digitalnih I/O i 6 analognih ulaznih pinova. Poseduje *Flash* memoriju od 32 kB, pri čemu je 0.5 kB iskorišćeno za *bootloader*. Programiranje je moguće vršiti u programu Arduino IDE a zahvaljujući ATmega16 čipu, vrši se konverzija USB signala u serijski signal.

Svaki od pinova ovog razvojnog okruženja poseduje svoju funkciju, kao što je objašnjeno u tehničkoj dokumentaciji [3-5].

B. Logički analizator Tektronix TLA5202B

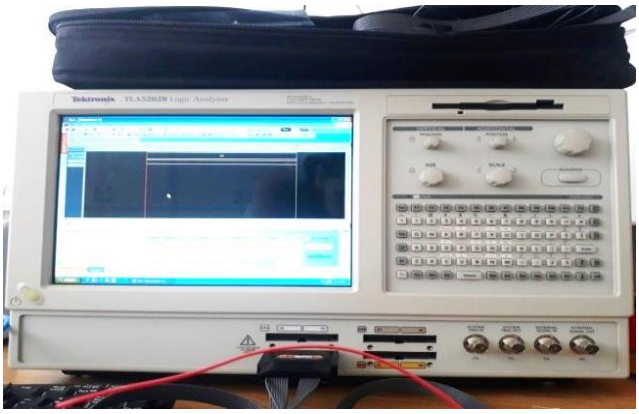
Logički analizator je merni instrument koji služi za analizu logičkih signala u digitalnim elektronskim kolima. Pod analizom se podrazumeva uzimanje uzoraka signala u određenim vremenskim intervalima, skladištenje uzetih uzoraka u memeroiju i prikazivanje memorisanih vrednosti u grafičkom i/ili numeričkom obliku.

TLA5202B serija logičkih analizatora(slika 6.) poseduje 68 kanala, raspodeljenih u 4 sonde. Za ovu seriju logičkih analizatora važi da poseduju mogućnost snimanja 16.000 uzoraka po kanalu kao i da im je brzina uzorkovanja 500 ps. Rezultati snimanja se prikazuju u obliku 0 ili 1, u zavisnosti od toga da li posmatrani signal ima vrednost manju ili veću od definisane vrednosti napona praga. Pored numeričkog prikaza podataka, moguće je i grafički posmatrati oblike signala.

Svaka od sondi poseduje 8 konektora, za 8 različitih signala. Prilikom povezivanja konektora sa sondama neophodno je voditi računa da pin 7 konektora odgovara pinu 7 sonde (slika 7.).

U okviru softvera analizatora moguće je u meniju *Setup* podesiti signale koji se testiraju unošenjem njihovih naziva u određenu kolonu, pri čemu redovi predstavljaju naziv

sonde koja se koristi za merenje signala a kolone služe za određivanjepinova na kojima će se svaki signal pojedinačno testirati.



Slika 6. Logički analizator Tektronix TLA5202B.



Slika7. Pravilno povezana sonda i konektor.

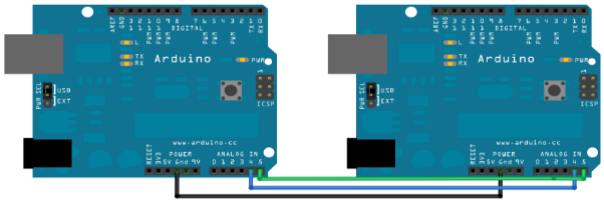
Analizator radi na principu MSB i LSB (*Least Significant Bit*), što znači da će signal koji je prvi označen tretirati kao MSB i njemu dati prioritet pri prikazu rezultata. U okviru ovog menija takođe se vrši podešavanje vrste takta (*internal/external*), vreme uzorkovanja i veličina memorije dostupne za čuvanje podataka. U okviru kartice *Thresholds* vrši se odabir vrednosti napona praga pogodnog za datu komponentu.

U meniju *Trigger* vrši se odabir vrste okidačkog impulsa, odnosno da li će se okidanje vršiti istog momernta kada otpočne merenje, na prednju ili zadnju ivicu takta i slično. Poslednji meni u kome je neophodno vršiti podešavanja jeste meni *Waveform*, gde se biraju signali koji će se prikazivati[1].

IV. Eksperiment i rezultati

Eksperiment je realizovan pomoću 2 *Arduino Uno* okruženja, gde je jedno okruženje predstavljalo *Master* uređaj koji šalje podatke, a drugo *Slave* uređaj koji prima podatke od *Master* uređaja i te podatke ispisuje [6-8].

Da bi mogla da se ostvari *I2C* komunikacija između 2 okruženja bilo je neophodno fizički povezati pinove ovih okruženja (slika 8.), a zatim izvršiti programiranje mikrokontrolera ovih okruženja (kao što je prikazano na slikama9.i 10.).



Slika 8. Prikaz povezivanja 2 Arduino Uno okruženja.

```
/*
KOD ZA MASTER UREĐAJ
*/

#include <Wire.h>

void setup() {
  Wire.begin();              // Pridruživanje I2C magistrali
}

byte x = 0;                  // Početna vrednost broja x

void loop() {
  Wire.beginTransmission(8); // Slanje podatka uređaju sa adresom #8
  Wire.write(x);             // Slanje 1 bajta
  Wire.endTransmission();    // Slanje završeno

  x++;                       // Inkrementiranje podatka x

  delay(500);
}
```

Slika 9. Kodovi koji su korišćeni za programiranje okruženja (*Master* uređaj).

```
/*
KOD ZA SLAVE UREĐAJ
*/

#include <Wire.h>

void setup() {
  Wire.begin(8);                // Pridruživanje I2C magistrali sa adresom #8
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);           // Pokretanje serijskog porta za izlaz
}

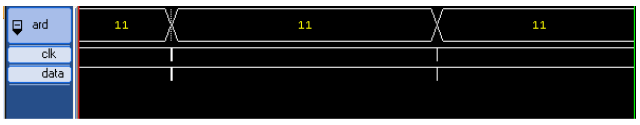
void loop() {
  delay(100);
}

// Funkcija se izvršava svaki put kada primimo podatak iz MASTER uređaja
void receiveEvent(int howMany) {
  int x = Wire.read();          // Primamo podatak tipa byte i čuvamo ga kao int
  Serial.println(x);            // Štampamo podatak koji smo primili
}
```

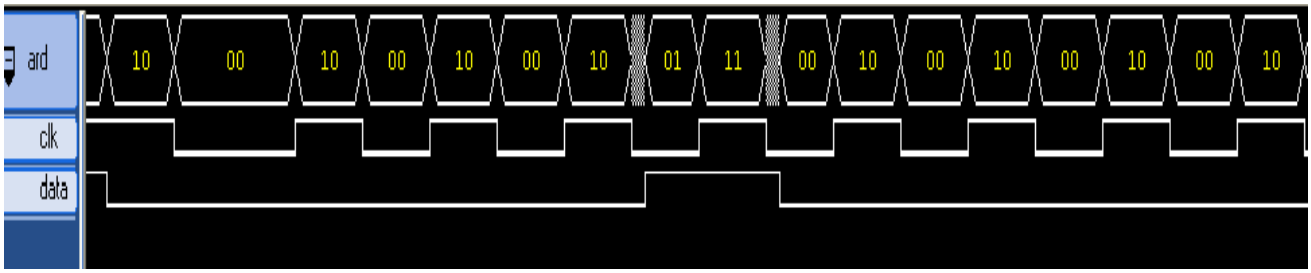
Slika 10. Kodovi koji su korišćeni za programiranje okruženja (*Slave* uređaj).

Rezultati eksperimenta prikazani su na slikama 11.–14. Na slici 11. prikazane su dve snimljene poruke iz komunikacije između *Master* i *Slave* uređaja. Na slici 12. prikazan je početak prve poruke,odnosno prelazak SDA linije sa HIGH na LOW dok je SCL linija i dalje u stanju HIGH, što predstavlja START bit. Nakon toga sledi slanje sedmobitne adrese *Slave* uređaja, pri čemu se uzimaju u obzir samo ona stanja SDA linije kada je SCL linija bila u stanju HIGH. R/$\overline{W}$ bit ima vrednost '0', što znači da se u registre *Slave* uređaja sa datom adresom upisuju podaci a nakon toga sledi ACK bit, kao potvrda da je *Slave* uređaj spreman za prijem podataka (slika 12.). Na slici 13.

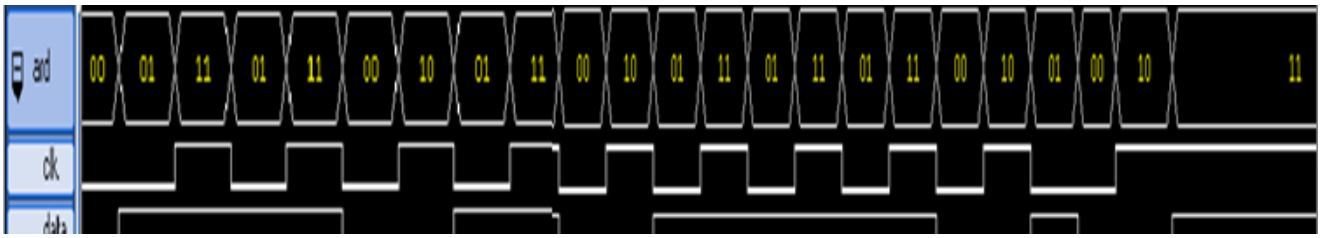
prikazana je prva poruka sa slike 11., gde možemo videti da je u komunikaciji između *Master* i *Slave* uređaja poslata vrednost 215. Na slici 14. prikazana je druga poruka sa slike 11., vidi se da je poslata vrednost 216, a što se uklapa sa logikom programa koji smo upotrebili za analizu podataka.Na slikama 13. i 14.,nakon poruka (brojeva 215 i 216) sledi ACK bit a zatim i STOP bit.



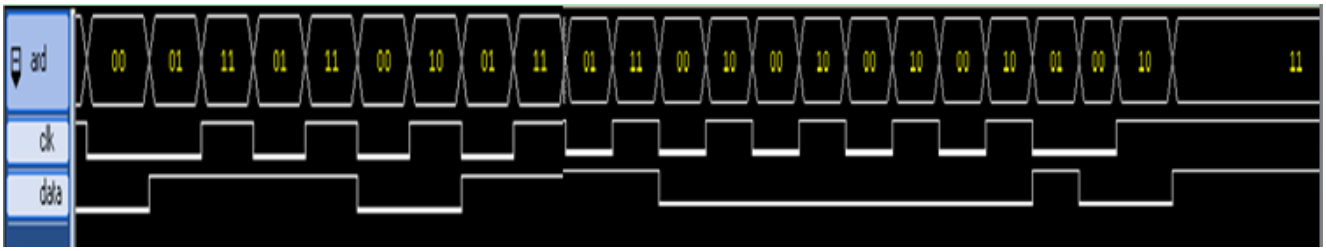
Slika 11. Izgled dobijenih poruka.



Slika 12. START bit i adresa *Slave* uređaja.



Slika 13. Izgled prve poruke i STOP bit.



Slika 14. Izgled druge poruke i STOP bit.

V. ZAKLJUČAK

Eksperimentom je potvrđeno da pravila definisana *I2C* protokolom važe i da poruka poslata ovom magistralom izgledana način propisan pravilima *I2C* protokola.Za ovu proveru iskorišćen je logički analizator Tektronix TLA5202B. Treba napomenuti da je provera izvršena korišćenjem više različitih modula i senzora, ali je u radu demonstrirana na primeru dva*Arduino Uno* okruženja, gde je jedno okruženje predstavljalo *Master* uređaj koji šalje podatke, a drugo *Slave* uređaj koji prima podatke od *Master* uređaja.

ZAHVALNOST

Ovaj rad deo je višenedeljnog istraživanja u laboratorijama Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu. Rad predstavlja deo diplomskog rada Stefane Zogović.

LITERATURA

[1] Analiza SPI komunikacionog protokola pomoću logičkog analizatora, izveštaj, Miloš Marjanović, Aleksandar Ilić, Elektronski fakultet, Niš 2011.
[2] Tektronix Logic Analyzer Family, Quick Start User Manual, http://powersof2.gatech.edu/resources/Instruments/TLAQuick Start.pdf
[3] Introductionto Arduino Uno, https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html
[4] Arduino Uno, https://en.wikipedia.org/wiki/Arduino_Uno
[5] Arduino, https://www.arduino.cc
[6] Master Reader/ Slave Sender, https://www.arduino.cc/en/Tutorial/MasterReader
[7] Wire Library, https://www.arduino.cc/en/Reference/Wire
[8] How to use *I2C* in Arduino: Communication between two Arduino Boards, https://circuitdigest.com/microcontroller-projects/arduino-*I2C*-tutorial-communication-between-two-arduino

Primena linearne optimizacije za podršku autonomnih vozila u pametnim gradovima

Nenad Petrović, Đorđe Kocić

Sadržaj - U ovom radu, razmatra se upotreba tehnika linearne optimizacije da bi se pokrila dva aspekta u okviru arhitekture pametnog grada vezana za autonomna vozila – punjenje baterija i obrada podataka sa senzora. Prezentovan je i alat za simuliranje scenarija pametnih gradova vezanih za autonomna vozila koji koristi rezultate optimizacije.

I. UVOD

Najmodernija tehnologija pametnih gradova je omogućila da autonmna vozila budu svakodnevica. Njihov cilj je da omoguće siguran, pouzdan transport, smanje broj nesreća, eliminišu potrebu za ljudskom interakcijom (parkiranje i obilaženje prepreka) i olakšaju život starijima i osobama sa invaliditetom [1].

Razvoj takvih rešenja stvara mnoge izazove vezane za percepciju, planiranje, kontrolu i etiku [1, 2]. Trebalo bi razmotriti pitanja privatnosti podataka, bezbednosti, prenosa i kašnjenja obrade s obzirom da je neophodno omogućiti komunikaciju velikog broja različitih vozila i drugih uređaja, u službi autonomnih vozila [1, 3]. Vrlo je bitna, takođe, i energetska efikasnost sistema transporta i vozila u okviru pametnih gradova [4].

U ovom radu, razmatra se upotreba tehnika linearne optimizacije da bi se pokrila dva aspekta u okviru arhitekture pametnog grada vezana za autonomna vozila – punjenje baterija vozila i obrada podataka sa senzora. Kao rezultat nude se dva modela linearne optimizacije. Modeliranje i simulacija u oblasti pametnih gradova i autonomnih vozila su veoma bitni jer pružaju siguran i jeftin način za testiranje različitih scenarija bez rizika od štete [5]. Stoga, u radu je opisan i implementiran alat za modeliranje i simulaciju koji koristi pomenute optimizacione modele, kojim se može vršiti testiranje i eksperimentisanje.

II. OSNOVE I SRODNA ISTRAŽIVANJA

A. Linearna optimizacija

Linearno programiranje (linearna optimizacija) je metod za optimizaciju kojim se postiže najbolje rešenje u zavisnosti od uslova (maksimalni profit, minimalna cena), a bazira se na matematičkom modelu koji se sastoji od linearnih jednačina [6]. Preciznije, to je tehnika za optimizaciju linearne funkcije cilja uz linearne jednačine kao ograničenja. Problem prikazan u sledećoj kanoničkoj formi se naziva *linearni program*:

$$\textit{Minimizirati } c^T x \quad (1)$$

$$\textit{pri uslovima } Ax \leq b,\ x \geq 0 \quad (2)$$

Oznake su sledeće: x - vektor promenljivih koje treba odrediti (promenljiva odlučivanja), c,b - vektori poznatih koeficijenata, A - matrica poznatih koeficijenata.

Linearno programiranje se primenjuje u rešavanju raznih problema u različitim oblastima - od ekonomije do tehnike. U arhitekturama pametnih gradova, tehnike optimizacije mogu se primeniti u minimizaciji cene energije, smanjenju emisija CO_2, menadžment otpadnih voda, saobraćaj vozila i optimizacija mreža [7, 8].

U ovom radu, linearno programiranje se primenjuje da bi se rešila dva problema softverskih arhitektura za autonomna vozila. Prvo, upotrebljava se za optimalan raspored autonomnih vozila po punionicama, a takođe se koristi i za smanjenje kašnjenja u obradi podataka sa senzora.

B. AMPL

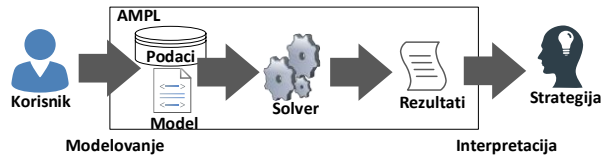
AMPL (*A Modeling Language for Mathematical Programming*) je algebarski jezik za modeliranje matematičkog programiranja, koji koristi tradicionalnu algebarsku notaciju sa ekstenzijama za probleme optimizacije (definisanje ograničenja u grafovima i izlomljena linearizacija). AMPL se koristi jer kombinuje i proširuje mogućnosti sličnih algebarskih jezika za modelovanje (npr. AIMMS, GAMS, LINGO i MPL), što ga čini uopštenijim, ali i dalje ostaje lak za korišćenje zbog toga što je njegova sintaksa [9] prirodna i intuitivna. On ne sadrži svoj mehanizam za rešavanje problema (*solver*), ali pruža tu mogućnost drugim programima kao što su CPLEX, MINOS ili NITRO. Jedna od njegovih glavnih prednosti je ta što pruža interfejs ka ovim solverima, pa nema potrebe poznavati ih detaljno ponaosob. U kontekstu ovog rada, CPLEX se koristi kao solver, jer pokriva linearno programiranje. Njegova implementacija se zaniva na simplex metodu [10], što je široko rasprostranjena tehnika linearnog programiranja.

Na Slici 1 je ilustrovan tok AMPL obrade. Prvo se modeluje konkretan problem koristeći AMPL algebarsku notaciju. Kada se ubace ulazni podaci, problem se rešava

N. Petrović je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: penenadpi@gmail.com

Đ. Kocić je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: seriousdjoka@gmail.com

pomoću odabranog solvera. Konačno, generisani rezultati se tumače i primenjuje odgovarajuća strategija.



Slika 1. AMPL tok obrade.

III. Pregled implementacije

A. Optimizacioni model punjenja baterija

Pretpostavimo da postoji skup stanica za punjenje S koje su vezane na električnu mrežu. Svaka stanica S_j se karakteriše maksimalnim kapacitetom kojim može da puni baterije, $MKap_j$. Svakoj S_j pripisujemo troškove distribucije električne energije označene kao CD_j. Sa druge strane, imamo i skup autonomnih vozila A. Svakom vozilu A_i dodeljujemo maksimalnu napunjenost baterije ($MBat_i$), trenutnu napunjenost ($TBat_i$) i utrošak enegije po km (CV_j). Rastojanja između svakog A_i i S_j su označene kao $R[i,j]$. Za svaku granu dodeljujemo promenljivu odluke $X[i,j]$ koja indicira da li će A_i da se puni na S_j. U tom slučaju promenljiva odluke je $X[i,j]=1$, dok je $X[i,j]=0$ u suprotnom.

Što se tiče funkcije cilja, teži se da se minimizuju ukupni troškovi, uzimajući u obzir i troškove distribucije i troškove puta za datu distancu, data kao :

$$minimize \sum_{i\in A, j\in S} X[i,j]*(CD[j]+CV[i]*R[i,j]) \quad (3)$$

Ali, postoje i pojedina ograničenja. Prvo, auto mora da ima dovoljno baterije da stigne do stanice za punjenje:

$$\sum_{j\in Stanice} X[i,j]*CV[i]*R[i,j] \leq TBat[i],\ i \in A \quad (4)$$

Svaka stanice može da prihvati samo jedan auto:

$$\sum_{i\in A} X[i,j] \leq 1,\ j \in S \quad (5)$$

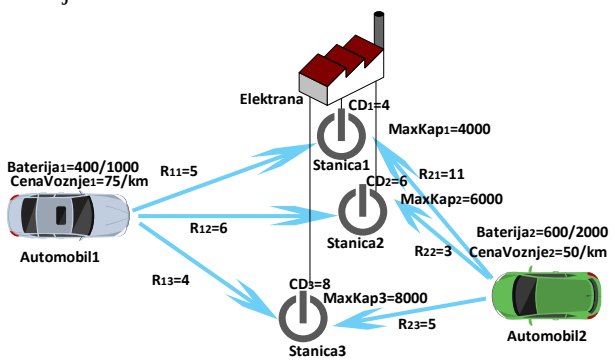
Svaki auto može da se puni na samo jednoj stanici:

$$\sum_{i\in S} X[i,j] = 1,\ i \in A \quad (6)$$

Konačno, stanica mora da pruži dovoljnu snagu automobilu koji se puni, što se izražava kao:

$$MBat[i]*X[i,j] \leq MKap[j], \text{ gde je } i \in A \text{ i } j \in S \quad (7)$$

Ilustracija primera ulaznih podataka za ovaj optimizacioni model je data na Slici 2.



Slika 2. Primer ulaznih podataka za model.

Kompletan AMPL kod za prethodno objašnjenu optimizaciju je dat u Listingu 1. U tom slučaju optimalno rešenje je da će se A_1 napuniti na S_3, a A_2 na S_2.

```
param brA;
param brS;

set A:=1..brA;
set S=1..brStanica;
param MBat{i in A};
param TBat {i in A};
param CV{i in A};
param MKap{j in S};
param CD {j in S};
param R{i in A, j in S};

# deklaracija promenljivih
var X {i in A, j in S} binary;

# funkcija cilja
minimize cost:
      sum{i in A, j in S}
X[i,j]*(CD[j]+R[i,j]*CV[i]);
subject to dovoljno_baterije {i in A}:
      sum{j in S}X[i,j]*CV[i]*R[i,j]<=TBat[i];
subject to jedan_auto_u_trenutku {j in S}:
      sum{i in A}X[i,j]<=1;
subject to jedna_stanica_u_trenutku {i in A}:
      sum{j in S}X[i,j]=1;
subject to punjenje {i in A, j in S}:
      MBat[i]*X[i,j]<=MKap[j];
```

Listing 1 - AMPL kod za punjenje autonomnih vozila.

B. Model minimizacije kašnjenja u prenosu podataka

Za autonomna vozila, od ključne je važnosti da se podaci koji dolaze za senzora obrade u realnom vremenu, pogotovu za tokove koji dolaze sa LIDAR ili kamere. U ovom kontekstu, paradigma *edge computing* nalaže da se obrada obavlja što bliže samim izvorima podataka. Cilj je generisanje pravila za oblikovanje saobraćaja u okviru softverski definisane mreže, koji će obezbediti da je kašnjenje pri prenosu podataka minimalno. U ovom kontekstu, problem se posmatra kao pronalaženje najkraće putanje od automobila do ciljanog servera. U nastavku će biti obrazložen model minimizacije kašnjenja obrade podataka biti obrazložen.

Autonomni automobil je preko $switch_{pocetak}$ povezan na softverski definisanu mrežu, a ciljani server preko $switch_{kraj}$. Potrebno je pronaći najkraću putanju kroz softverski definisanu mrežu od početnog do odredišnog *switch-a*. Oni su povezani za zajednički kontroler softverski-definisane mreže i označeni kao čvorovi, a veze između njih kao potezi u grafu.

Svakom potegu se dodeljuje promenljiva odluke koja određuje da li se poteg od $switch_i$ do $switch_j$ nalazi na najkraćoj putanji ili ne: $Put[i,j]=1$ ako $Poteg[i,j]$ pripada najkraćoj putanji do servera, dok je, u suprotnom $Put[i,j]=0$. Osim toga, svakom potegu se dodeljuje i vrednost kašnjenja po toj vezi, označena kao $Kasnjenje[i,j]$. S obzirom da je glavni cilj minimizacija kašnjenja, funkcija cilja je:

$$minimize \sum_{i,j\in Cvor} Kasnjenje[i,j]Put[i,j] \quad (8)$$

Sledeće ograničenje mora biti zadovoljeno: osim za startni čvor $switch_{pocetak}$ i krajnji $switch_{kraj}$, za svaki čvor *i* na putanji mora da tačno jedan poteg ulazi, ali i tačno jedan izalzi iz njega. Ograničenje je dato jednačinom:

$$\sum_{h,i \in Cvor} Put[h,i] - \sum_{i,j \in Cvor} Put[i,j] = d_i \quad (9)$$

Pri tome, ako je *(i=s)* onda $d_i=1$, a ako *(i=e)* onda $d_i=-1$, dok je u ostalim slučajevima $d_i=0$.

AMPL kod za ovaj model je dat u Listingu 2.

```
param brCvorova;
set Cvorovi:=1..brCvorova;
param Kasnjenje {i in Cvorovi, j in Cvorovi};
param d {i in Cvorovi};

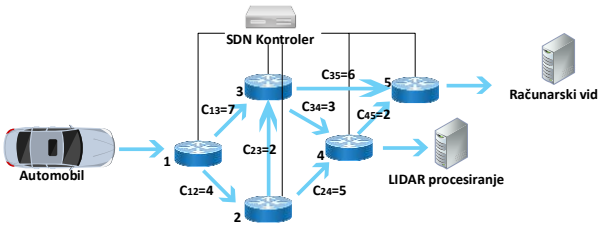
# deklaracija promenljivih
var Put {i in Nodes, j in Nodes} >= 0;

# ciljna funkcija
minimize latency:
       sum{i in Cvorovi, j in Cvorovi} Put[i,j]*Kasnjenje[i,j];

# ogranicenja
subject to conservation {i in Cvorovi}:
      sum{j in Cvorovi}Put[i,j]=d[i]+sum{h in Cvorovi}Put[h,i];
```

Listing 2 – AMPL kod za minimizaciju kašnjenja pri prenosu podataka kroz mrežu.

Ilustracija primera za ulaz ovog linearnog programa je data na Slici 3.



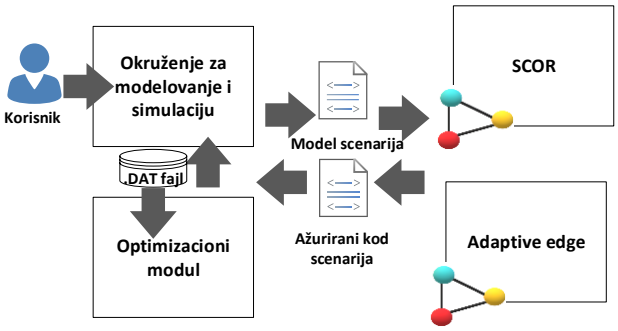
Slika 3. Primer ulaznih podataka za minimizaciju kašnjenja prenosa podataka kroz mrežu.

IV. ALAT ZA MODELOVANJE I SIMULACIJU

U ovom radu je, između ostalog, predstavljen i alat za modelovanje i simulaciju sa korisničkim interfejsom nalik na strateške 2D video igre, koji pruža vizuelnu notaciju za kreiranje scenarija pametnih gradova. Glavni ciljevi ovog softvera jesu da se omogući lako eksperimentisanje u kontekstu pametnih gradova, ali i integracija sa drugim alatima i radnim okvirima koji pružaju još naprednih mogućnosti. Aplikacija se pokreće u web pretraživaču, a implementirana korišćenjem JavaScript, jQuery, AJAX, CSS i HTML tehnologije. Pregled komponenti koje ulaze u sastav alata i njihove interakcije je dat na Slici 4.

Korisnik prvo kreira model scenarija, korišćenjem alata za modelovanje. Svaki kreirani model scenarija može da se izveze kao .DAT fajl, koji AMPL koristi za ulazne podatke procesa optimizacije. Posle toga, kreirani model scenarija se može iskoristiti u okviru alata za simulaciju.

U toku simulacije, neki elementi se mogu kretati, što dovodi do promena u okruženju za simulaciju. Ove promene mogu da okidaju mehanizme za koordinaciju i adaptaciju eksternih radnih okvira, koji mogu zahtevati rezultate procesa optimizacije. U našem slučaju, oslanjamo se na dva postojeća radna okvira za semantikom-vođeno generisanje koda: 1) SCOR – za izbegavanje prepreka i kolizija [11, 12] 2) Adaptive edge [13] – radni okvir koji sadrži funkcionalnosti vezane za generisanje pravila oblikovanja saobraćaja u okviru softverski-definisanih računarskih mreža u edge okruženjima.



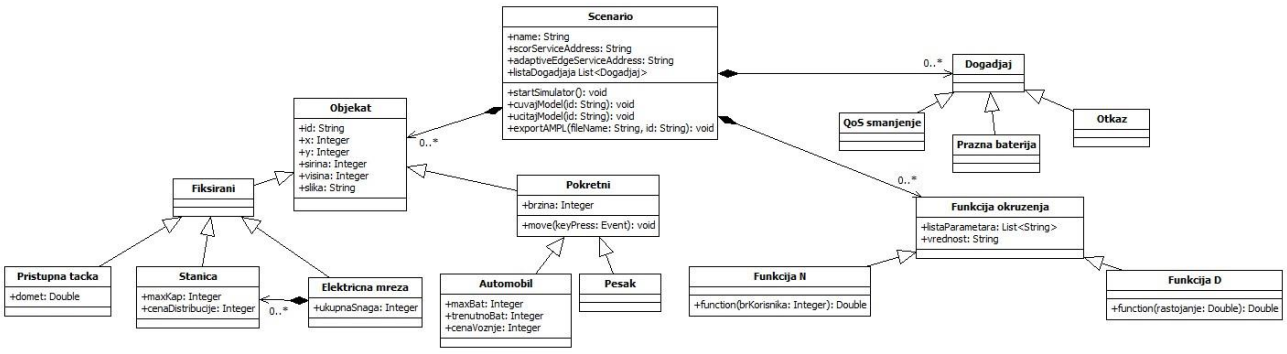
Slika 4. Arhitektura alata za modeliranje i simulaciju.

U okviru prostora alata za modelovanje i simulaciju, korisnici mogu da raspoređuju elemente iz unapred zadate liste elemenata. Svaki od elemenata ima svoju vizuelnu reprezentaciju i poziciju datu koordinatama. Elemente delimo u dva disjunktna podskupa – pokretni i fiksni. Pokretni elementi obuhvataju autonomne automobile i pešake. Alat za simulaciju omogućava njihovo kretanje korišćenjem strelica na tastature, što može da izaziva varijaciju parametara okruženja. Sa druge strane, fiksni elementi nisu pokretni, a tu spadaju prepreke (poput zgrada, kuća), elektrane, stanice za punjenje baterije povezane na električnu mrežu i bežične servisne pristupne tačke. Za svaki pokretni element postoje specifični parametri, kao što su brzina kretanja, maksimalni kapacitet baterije, trenutno stanje baterije ili cena vožnje za automobile izražena u potrošenoj energiji/gorivu po kilometru. Kada govorimo o stanicama za punjenje, moguće je prilagoditi njihove parametre koji se tiču cene distribucije i maksimalan kapacitet koji mogu pružiti za punjenje. Sa druge strane, za bežične pristupne tačke je bitan parametar domet signala. Opis vizuelene notacije korišćene u okviru alata je dat na Slici 5. U toku kretanja, rastojanje do bežične pristupne tačke se menja tako da, u skladu sa tim, variraju i različiti parametri u okruženju.

Recimo, ako je bežična pristupna tačka na lokaciji (x_{pt}, y_{pt}), a autonomni automobil na poziciji (x_a, y_a), onda se posmatrani parametar kvaliteta usluge - *Quality of Service (QoS)* parametar, poput broja prenetih slika sa kamere u sekundi menja po formuli:

$$QoS_{novo} = f_d(rastojanje(a,pt)) * QoS_{staro} \quad (10)$$

U ovoj formuli, QoS_{novo} je novodobijena, modifikovana vrednost QoS_{staro}, dok je f_d opadajuća funkcija rastojanja između automobila i pristupne tačke, normalizovana na interval od 0 do 1.



Slika 5. UML klasni dijagram alata za modelovanje i simulaciju pametnog grada.

Što se tiče promene broja korisnika usluge, vrednost *QoS* parametara varira u skladu sa formulom:

$$QoS_{novo}=f_n(n)*QoS_{staro} \quad (11)$$

U ovom slučaju, n predstavlja broj potrošača, dok je f_n opadajuća funkcija od *n*. Osim toga, karakteristike specifičnog uređaja koji se koristi kao bežična pristupna tačka (poput dometa signala) se uzimaju u obzir.

Trenutno stanje baterije se izračunava po formuli:

$$Baterija_{novo}=Batterija_{staro}-Rastojanje*CenaVoznje \quad (12)$$

Varijacije *QoS* i drugih parametara (poput stanja baterije) izazvane promenom rastojanja i broja korisnika u okolini mogu izazvati okidanje mehanizama za koordinaciju i adaptaciju, što omogućava ispitivanje ponašanja optimizacionih modela pre upotrebe u realnom sistemu, što dovodi do značajnih ušteda vremena i novca.

V. ZAKLJUČAK

Na osnovu inicijalnih testova, može se zaključiti da pristup korišćenja linearne optimizacije za rešavanje problema vezanih za autonomne automobile u pametnim gradovima deluje obećavajuće. U budućnosti planiramo dodavanje novih optimizacionih modela i proširenje postojećih ostalim ograničenjima i aspektima vezanih za *edge computing*, kao što su privatnost, bezbednost i kretanje podataka [13]. Konačno, krajnji cilj je integracija predstavljenih mehanizama sa našim radnim okvirom za podacima-vođene pametne mreže [14].

LITERATURA

[1] I. Yaqoob et al., "Autonomous Driving Cars in Smart Cities: Recent Advances, Requirements, and Challenges", IEEE Network, August 2019, pp. 1-11.

[2] L. Lin, J. J. Zhu, "Path Planning for Autonomous Car Parking", Proceedings of the ASME 2018 Dynamic Systems and Control Conference, 2018, pp. 1-10.

[3] R. Dave, E. S. Boone, K. Roy, "Efficient Data Privacy and Security in Autonomous Cars", Journal of Computer Sciences and Applications, 2019, Vol. 7, No. 1, pp. 31-36. https://doi.org/10.12691/jcsa-7-1-5

[4] J. P. Gouveia, J. Seixas, G. Giannakidis, "Smart City Energy Planning", Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion, 2016, pp. 345-350. https://doi.org/10.1145/2872518.2888617

[5] K. Li, H. Su, J. Cheng and X. Tang, "An Architecture of Autonomous Car Simulation System", CICTP 2019, pp. 5843-5852. https://doi.org/10.1061/9780784482292.502

[6] G. B. Dantzig and M. N. Thapa, Linear Programming 1 - Introduction, Springer, 1997.

[7] N. Mohamed, S. Lazarova-Molnar, J. Al-Jaroodi, "Cloud of Things: Optimizing smart city services", ICMSAO 2017, pp. 1-5, 2017. https://doi.org/10.1109/ICMSAO.2017.7934875

[8] M. Sato, Y. Fukuyama, "Total Optimization of Smart City by Modified Brain Storm", 10th IFAC Symposium on Control of Power and Energy Systems CPES 2018, pp. 13-18 , 2018. https://doi.org/10.1016/j.ifacol.2018.11.670

[9] R. Fourer, D. Gay, B. Kernighan, AMPL: A Modeling Language for Mathematical Programming, Duxbury Press, 2003.

[10] G. B. Dantzig, A. Orden and P. Wolfe, "The generalized simplex method for minimizing a linear form under linear inequality restraints", Pacific Journal of Mathematics, Volume 5, Number 2, 1955, pp. 183-195.

[11] V. Nejkovic, N. Petrovic, N. Milosevic, M. Tosic, "The SCOR ontologies framework for robotics testbed", 2018 26th Telecommunication Forum (TELFOR), pp. 1-4, 2018. https://doi.org/10.1109/telfor.2018.8611841

[12] N. Petrovic, V. Nejkovic, N. Milosevic, M. Tosic, "A Semantic Framework for Design-Time RIoT Device Mission Coordination", 2018 26th Telecommunication Forum (TELFOR), Belgrade, pp. 1-4, 2018. https://doi.org/10.1109/telfor.2018.8611845

[13] N. Petrović, "Model-based Approach for Semantic-driven Deployment of Containerized Applications to Support Future Internet Services and Architectures", Serbian Journal of Electrical Engineering 16 (1), February 2019, pp. 21-44. https://doi.org/10.2298/SJEE1901021P

[14] N. Petrović, Đ. Kocić, "Podacima-vođena arhitektura za prilagodljive energetske mreže zasnovana na IoT uređajima", ETRAN 2019, Silver Lake, Serbia, 2019, pp. 1-6.

Sistem za detekciju i razmenu vremenskih uslova na putu realizovan korišćenjem Arduino Uno razvojnog okruženja

Aleksandar Stoiljković, Lazar Stoiljković, Jelena Zdravković

Sadržaj –U ovom radu će biti predstavljen sistem za automatsku detekciju i razmenu trenutnog stanja vremena (magla, led, kiša) između učesnika u saobraćaju. Sistem je realizovan sa ciljem da poveća svest i upozori vozača o potencijalnim opasnostima i rizicima na putu. Za realizaciju sistema biće korišćeno Arduino Uno razvojno okruženje, kao i neophodni senzori i elektronski elementi, koji će detaljnije biti opisani u nastavku.

I. UVOD

U poslednjih par godina dosta pažnje je posvećeno smanjenju saobraćajnih nezgoda. Uticaj tehnologije u različitim životnim sferama je sve veći, pa se samim tim nameće činjenica da se tehnologija iskoristi i u cilju smanjenja saobraćajnih nezgoda. Osnovna ideja je da se korišćenjem tehnologije na vreme otkrije opasnost na putu i blagovremeno upozori vozač. Fizička komunikacija između dva vozila koja se susreću na putu nije moguća. Sa tim u vezi potrebno je razviti sistem koji bi na neki način omogućio komunikaciju, odnosno razmenu relevantnih informacija za vozača, a da se pri tome vozač ni na koji način ne ometa. Rešenje je vozača upozoriti zvučnim, a ne vizuelnim signalom. Svaki sistem koji će biti razvijen zna trenutno stanje vremena na putu i konstantno prikuplja infromacije i ima za cilj da te informacije prosledi sistemu u vozilu koje mu dolazi u susret kako bi ga obavestilo o potencijalnim opasnostima. Ukoliko se pristigli podaci razlikuju od podataka koje sistem poseduje, sistem obaveštava vozača putem odgovarajućeg zvučnog signala, inače odbacuje pristigli podatak i vraća se u stanje u kome prikuplja i čeka informacije od drugog sistema. Za realizaciju sistema je iskorišćeno *Arduino* razvojna platforma jer je pristupačna, jeftina i pogodna za održavanje.

II. PRINCIP RADA

Princip rada sistema opisan je kroz sledeće korake. U trenutku startovanja automobila startuje se i sistem i tada se sistem nalazi u inicijalnom stanju i počinje sa prikupljanjem informacija o vremenskim uslovima na putu. Bitno je napomenuti da svaki sistem prikuplja i pamti informacije o vremenskim uslovima na putu koji je automobil već prešao, a sa druge strane osluškuje i prima informacije o vremenskim uslovima na putu koji automobil tek treba da pređe kako bi mogao da upozori vozača. Sistem pamti informacije u obliku od po tri bita (za svaku vremensku priliku jedan bit). U inicijalnom stanju sistem ima informaciju da trenutno na predstojećem putu nema ni magle, ni leda, ni kiše (sva tri bita su setovana na 0). Inicijalno stanje je „000“-za put koji je automobil već prešao (pamti se u *sendBuffer*-u) i „000“-za put koji će automobil tek da pređe(pamti se u *receiveBuffer*-u). Sistem konstantno ima mogućnost da primi ili pošalje odgovarajuću informaciju.

Zatim nakon pokretanja, sistem pomoću različitih senzora detektuje određene vrednosti za maglu, kišu i led. Kiša se detektuje pomoću odgovarajućeg senzora za vlagu, led se detektuje pomoću foto senzora. Vrednosti koje su izmerene senzorima sistem poredi sa graničnim vrednostima sa svaku vremensku pojavu. Ukoliko je izmerena vrednost veća od granične vrednosti postavlja odgovarajući bit na ’1’ u *sendBuffer*-u. U slučaju povećane magle na putu vozač aktivira određeno dugme u sistemu čime postavlja bit za maglu na ’1’ u *sendBuffer*-u.

Pored detekcije sistem konstantno osluškuje potencijalne dolazne informacije. Kada primi informaciju od drugog sistema-koje se šalju iz *sendBuffer*-a (u format od tri bita) ispituje se da li se neka informacija razlikuje od one koje poseduje u sopstvenom *receiveBuffer*-u i ako je to slučaj vrši promenu odgovarajućeg bita u *receiveBuffer*-u. Nakon toga sistem proverava da li je neki bit u *receiveBuffer*-u postavljen na ’1’ i u slučaju da jeste obaveštava vozača odgovarajućim zvučnim signalom.

Nakon određenog vremena sistem se resetuje i vraća u inicijalno stanje i nastavlja sa detekcijom uslova na putu i osluškivanjem dolaznih informacija.

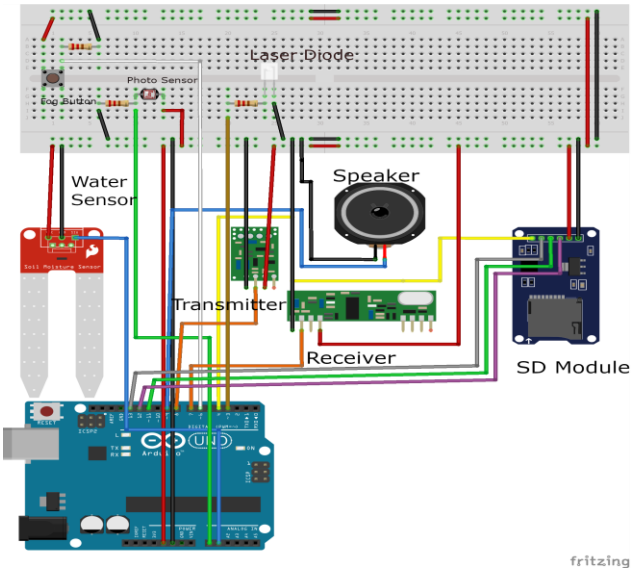
III. ŠEMA I ELEMENTI KOLA

Za realizaciju sistema potrebnu su sledeći elementi:
-Arduino Uno mikrokontolerska ploča
-Senzor za vlagu
-Foto senzor i laser
-Dugme za aktivaciju pojave magle

Aleksandar Stoiljković i Lazar Stoiljković su studenti treće godine na smeru Računarstvo i Informatika, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. Email: aca.stoiljkovic@elfak.rs, lazar.stoiljkovic@elfak.rs

Jelena Zdravković je učenica četvrtog razreda Elektrotehničke škole “Mija Stanimirović” Niš, Bulevar Svetog cara Konstantina 82-84, 18000 Niš, Srbija Email: jeciik9@gmail.com.

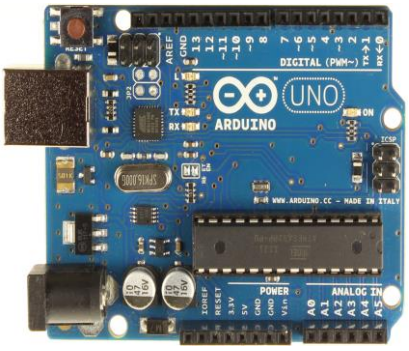
-Transmitter i Receiver
-Modul za SD karticu
-*Speaker*
-Ostali propratni elektronski elementi



Slika 1. Šema sistema.

A. Arduino Uno

Na slici 2 je prikazan izgled *Arduino Uno* mikrokontrolerske ploče. Arduino kao što je prikazano na slici 1 integriše sve ostale komponente i omogućava da sistem funkcioniše kao jedna celina. *ArduinoUno* je razvojno okruženje sa mikrokontrolerom koje se sastoji od skupa ulazno/izlaznih pinova pomoću kojih je omogućeno jednostavno povezivanje sa različitim elektronskim komponentama. Pomoću USB kabla se povezuje na računar i pomoću posebnog programskog jezika se kontroliše rad mikrokontrolera. Celokupna *Arduino* platforma se sastoji od razvojnog okruženja, programskog jezika, koji je prilagođena verzija jezika C i same pločice sa mikrokontrolerom i minimalnim dodatnim delovima da bi se omogućilo napajanje i komunikacija sa računarom.



Slika 2. Arduino Uno.

B. Senzor za vlagu

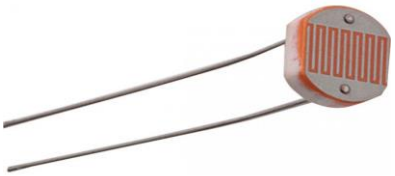
Na slici 3 prikazan je senzor za vlagu. Senzor za vlagu je vodootporan i koristi se za merenje vlažnosti, odnosno za detekciju vode na putu. Izmerena vrednost se šalje na mikrokontroler koji poredi izmerenu vrednost sa graničnom vrednošću i u zavisnosti od rezultata aktivira ili ne bit za kišu.



Slika 3. Senzor za vlagu.

C. Foto senzor

Za detekciju leda na putu sistem koristi odgovarajući foto senzor i laser. Naime usled pojave leda, svetlost koju emituje laser se reflektuje i foto senzor detektuje određenu vrednost. U slučaju da je izmerena vrednost veća od granične aktivira se bit za led na put. Igled foto senzora prikazan je na slici 4.

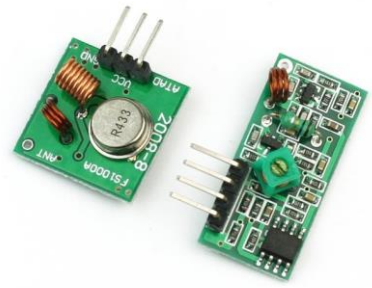


Slika 4. Foto sensor.

C. Transmitter i receiver

Kako bi se obezbedila *full-duplex*, serijska komunikacija(oba sistema-uređaja mogu i da šalju i da primaju podatke) između dva sistema koristi se *receiver* i *transmitter*.Za relativno kratak domet komunikacije (20-30 m), što je slučaj i kod opisanog sistema najpoželjniji mediji koji se koriste su RF(*radio frequency*) *transmitter* i *receiver*. Kako bi mogli da koristimo *transmitter* i *reciver* potrebno je ih povezati na *arduino* mikrokontrolersku ploču i takođe u *arduino* projekat uključiti odgovarajuću

biblioteku za rad sa *receiver*-om i *transmitter*-om. Na slici 5 prikazana su *receiver* i *transmitter*.



Slika 5. Transmitter i receiver.

E. Modul za SD karticu

Zbog činjenice da sistem treba zvučnim signalom da upozorava vozača na eventualne vremenske neprilike potrebno je negde zapamtiti zvučne signale zapamtiti na SD kartici. Povezivanje SD kartice sa arduino pločom ostvaruje se preko modula za SD karticu. Modul za SD karticu prikazan je na slici 6.



Slika 6. Modul za SD karticu.

IV. PROGRAMIRANJE I TESTIRANJE

Osnovna programska struktura *Arduino* razvojnog okruženja je veoma jednostavna. Sastoji se od dve osnovne funkcije: *setup* i *loop*. Funkcija *setup* će se izvršiti samo jednom i to pri pokretanju programa na *Arduino* ploči. U *setup* funkciji se vrši inicijalizacija sistema i navode se početni uslovi rada ploče, definisanje ulazno/izlaznih pinova.

Loop funkcija se izvršava konstantno nakon izvršenja *setup* funkcije beskonačan broj puta. U *loop* segmentu treba opisati rad kola. U opisanom sistemu u *loop* treba definisati slanje i primanje podataka.

Pre definisanja *setup* i *loop* funkcije potrebno je podesiti odgovarajuće pinove, kao i uključiti neophodne biblioteke za serijsku komunikaciju i za rad sa SD modulom, kako bi sistem korektno funkcionisao. Na sledećoj slici prikazan je taj deo koda. Takođe potrebno je izvršiti i inicijalizaciju *sendBuffer*-a i *receiveBuffer*-a.

```
#include "SD.h"
#include "TMRpcm.h"
#include <VirtualWire.h>

/*
    Slot za SD karticu je povezan : CS = 4, MOSI  = 11, MISO = 12, SCK = 13

    Za radio transmitter-a i receiver-a  : Tx = 8, Rx = 7

    fotoSensor = A0, laserska dioda ce konstantno da svetli

    senzor za vlagu na pinu A1

    dugme za maglu na pinu 6
*/

#define musicSpeakerPin 9
#define radioTxPin 7
#define radioRxPin 5
#define buttonPin 6
#define SD_ChipSelectPin 4
#define laserDiodePin 3
#define rainSensorPin A1
#define fotoSensorPin A0
#define radioBitsPerSec 500
#define bufLength 3

TMRpcm memoryCard;
char bufSnd[bufLength]= {'0','0','0'}; // 0 - rain , 1 - ice , 2 - fog
char bufRcv[bufLength]= {'0','0','0'}; // 0 - rain , 1 - ice , 2 - fog
int buttonState = 0;
int fotoSensorValue = 0;
int rainSensorValue = 0;
int countTime = 0;
```

Slika 7. Deo koda u kome se obavlja početno podešavanje okruženja.

```
void setup(){

    // Testiranje
    // Serial.begin(9600);

    if (!SD.begin(SD_ChipSelectPin)) {
        // Testiranje
    //   Serial.println("SD fail");
      return;
    }

    // inicijalizacija fog button-a
    pinMode(buttonPin,INPUT);

    pinMode(laserDiodePin,OUTPUT);
    digitalWrite(laserDiodePin,HIGH);

    // inicijalizacija radio transmitter-a i receiver-a
    vw_set_ptt_inverted(true);
    vw_set_tx_pin(radioTxPin);
    vw_set_rx_pin(radioRxPin);
    vw_setup(radioBitsPerSec);
    vw_rx_start();

    memoryCard.speakerPin = musicSpeakerPin;
    memoryCard.setVolume(5); // moze da ide do 7 ( 5 je idelano )
    memoryCard.quality(1); // 1 je bolji kvalitet 0 je gori
}
```

Slika 8. *Setup* funkcija.

Na slici 8 je prikazana definicija *setup* funkcije. U njoj se vrši inicijalizacija pinova za dugme za signalizaciju magle, kao i pin za laser koji emituje svetlost. Takođe vrši se inicijalizacija *transmitter*-a i *receiver*-a kao i podešavanje i inicijalizacija modula za SD karticu.

U *loop* funkciji se konstantno obavlja preuzimanje vrednosti detektovanih pomoću odgovarajućih senzora kroz procedure *getInputData()*. Takođe pozivaju se i procedure za slanje podataka drugim sistemima, kao i za prijem podataka od drugih sistema. Na sledećoj slici prikazan je taj deo koda koji opisuje *loop* funkciju.

```
189  void loop(){
190      if(countTime<60){
191          getInputData();
192          sendMessage();
193          receiveMessage();
194          //3 puta u minutu obavestava sta se desava
195          if(countTime%20 == 0){
196              callSound();
197          }
198          countTime++;
199      }
200      else{
201          resetData();
202      }
203      delay(100);
204  }
205  // uzima podatke sa svih senzora
206  void getInputData(){
207      readRainSensor();
208      readFotoSensor();
209      readFogButton();
210      delay(100);
211  }
```

Slika 9. *Loop* funkcija.

Možemo videti da je cilj sistema da na svakih 20 sekundi obezbedi da se korisnik upozori zvučnim signalom, pozivom procedure *callSound()*. Nakon 60 sekundi sistem se resetuje i vraća u inicijalno stanje pozivom procedure *callSound()*. Na slici 10 prikazana je realizacija ovih funkcija.

```
133  // nakon 1 minuta resetuj podatke
134  void resetData(){
135      bufSnd[0] = '0';
136      bufSnd[1] = '0';
137      bufSnd[2] = '0';
138      bufRcv[0] = '0';
139      bufRcv[1] = '0';
140      bufRcv[2] = '0';
141      countTime = 0;
142
143      delay(50);
144  }
145
146  // na osnovu podataka iz bufRcv bira zvuk
147  void callSound(){
148      if(bufRcv[0] == '1'){
149          memoryPlay("rainy.wav");
150      }
151      if(bufRcv[1] == '1'){
152          memoryPlay("slippery.wav");
153      }
154      if(bufRcv[2] == '1'){
155          memoryPlay("fog.wav");
156      }
157      delay(200);
158  }
```

Slika 10. *resetData* funkcija i *callSound* funkcija.

Na kraju je opisan deo programa u kome se obrađuju informacije (podaci) koje su prikupili - detektovali senzori. Procedure koje obrađuju taj deo logike predastavljene su na slici 11.

```
92   // ocitava podatke sa foto senzora
93   void readFotoSensor(){
94     fotoSensorValue = analogRead(fotoSensorPin);
95     if(fotoSensorValue > 310){/// neophodna detaljna analiza vrednosti za senzor
96         if(bufSnd[1] == '0'){
97             bufSnd[1] = '1';
98         }
99     }
100  }
101
102  // proverava da li je dugme pritsnuto
103  void readFogButton(){
104      buttonState = digitalRead(buttonPin);
105      if(buttonState == 1){
106          if(bufSnd[2] == '0'){
107              bufSnd[2] = '1';
108          }
109      }
110
111      buttonState = 0;
112  }
113
114  // uzima podatke sa senzora za vlagu
115  void readRainSensor(){
116      rainSensorValue = analogRead(rainSensorPin);
117      if(rainSensorValue > 500){ // neophodna detaljna analiza vrednosti za senzor
118          if(bufSnd[0] == '0'){
119              bufSnd[0] = '1';
120          }
121      }
122
123  }
```

Slika 11. *readFotoSensor*, *readFogButton*, *readRainSensor* funkcije

V. Zaključak

Sistem je razvijen sa idejom da se korišćenje tehnologije uvede u oblast sigurnosti u saobraćaju. Sistem je namenjen svima koji žele da ga ugrade u svoja vozila. Ne postoje nikakva druga ograničenja i zahtevi od strane korisnika koji će da koristi sistem. Neophodno je samo izvršiti instalaciju sistema na vozilu.

Zahvalnica

Autori ovog projekta se zahvaljuju Elektrotehničkoj školi “Mija Stanimirović” koja je obezbedila svu neophodnu opremu za realizaciju ovog sistema. Takođe želimo i da se zahvalimo mentoru Goranu Lučiću za pomoć koju nam je pružio prilikom realizacije ovog sistema.

Literatura

[1] ArduinoIDE dostupno na :
https://www.arduino.cc/en/Main/Software
[2] VirtualWire dostupno na :
http://www.airspayce.com/mikem/arduino/VirtualWire
[3] SD library dostupno na :
https://www.arduino.cc/en/Reference/SD

Pametna kapija

Luka Tomić, Marko Bijelić, Lazar Milovanović, Nemanja Savić, Živko Prolić

Sadržaj – U ovom radu prikazano je upravljanje kapijom preko GSM (*Global System for Mobile Communications*) modula. Upravljanje je omogućeno i u administratorskom modu, putem veb interfejsa. Prikazana su sredstva i alati koji su korišćeni tokom izrade rada.

I. UVOD

Osnovna funkcija sistema je daljinsko upravljanje kapijom (otvaranje/zatvaranje) putem poziva sa mobilnog telefona ili slanjem SMS poruke. Korisnik pozivom broja telefona sistema otvara kapiju, ukoliko je njegov broj registrovan u bazi podataka sistema. U suprotnom, kapija ostaje zatvorena. Sistemom upravlja *Raspberry Pi* [1] mikroračunar. Veb server, na kojem je aktivna aplikacija za administraciju sistema, nalazi se na istom računaru.

II. PRINCIP RADA

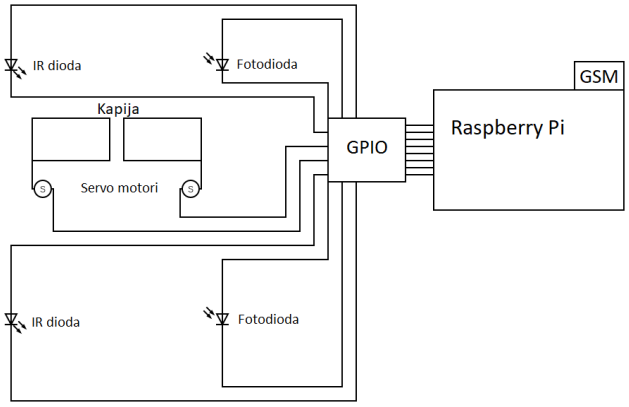
Raspberry Pi , ima fiksnu IP (*Internet Protocol*) adresu u okviru lokalne mreže i na njemu je aktivan veb server (*Flask* [2]).

Otvaranje i zatvaranje se vrši u više korisničkih nivoa. Na primer, korisnici sa najvišim proritetom imaju mogućnost zaključavanja kapije. U tom slučaju je manipulacija kapijom blokirana za korisnike sa nižim prioritetom.

Upravljanje kapijom putem poziva: Prvo se vrši identifikacija broja koji je izvršio poziv. Potom sistem automatski prekida vezu, pamti broj sa kog je upućen poziv i vrši proveru u bazi podataka. Nakon prepoznavanja broja telefona izvršava se proces otvaranja kapije. Na kapiji su postavljeni senzori koji detektuju da li je vozilo ostalo u dometu kapije. U slučaju da jeste, kapija se ne zatvara ili se zaustavlja, ako je zatvaranje počelo. Ukoliko senzori ne očitaju ništa kapija se zatvara nakon 30 sekundi.

Upravljanje kapijom putem SMS poruka: Postoji mogućnost upravljanja kapijom slanjem SMS tekstualne poruke. Opcija nije dostupna korisnicima sa najnižom prioritetom. Korisnici sa ovom dozvolom mogu slati tipske poruke za pojedine akcije. Sistem vrši proveru broja telefona u bazi podataka (kao kod upravljanja putem poziva). Potom izvršava proveru sadržaja poruke da bi sistem znao koju akciju korisnik zahteva. Opcije koje korisnik može da zahteva su otvaranje i zatvaranje kapije, fiksiranje položaja kapije u određenom vremenskom periodu zadatom od strane korisnika(npr. zak30min-zaključavanje kapije na 30 min).

Upravaljanje putem veb aplikacije (administratorsko upravljanje): U slučaju da se aplikaciji pristupa putem interneta unosi se adresa sa WAN (*Wide Area Network*) porta uređaja preko kojeg se ostvaruje internet konekcija sistema. Aplikacija sadrži formu za logovanje koja prosleđuje administratora na stranicu na kojoj pored svih korisničkih opcija, registracije i brisanja korisnika, ima opciju za "trajno" zaključavanje kapije. U tom slučaju ni jedan korisnik ne može da je otvori dok administrator ne izvrši njeno otključavanje. Deo sistema, vidljiv putem veba je i video nadzor. Administrator može obavljati i sve funkcije standardnog korisnika.



Slika 1. Šema sistema.

III. IMPLEMENTACIJA

A. Hardverska implementacija

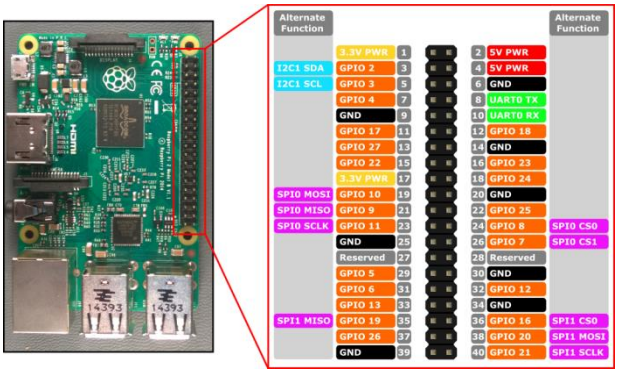
Hardverska osnova sistema je *Raspberry Pi 3*, mikroračunar (Slika 2). *Raspberry Pi* je *single-board* kompjuter veličine kreditne kartice. Koristi se kako na maketi, tako i u konačnoj izradi sistema

Instalirani operativni sistem je *Raspbian*. Koristi 1200 MHz četvorojezgrani *ARM Cortex-A53* procesor. Poseduje memoriju od 1GB. Za skladištenje koristi *MicroSDHC* karticu kao i njegovi prethodnici. Grafiku predstavlja *Broadcom VideoCore IV* ali na višim frekvencijama. Snaga za napajanje ovog modela iznosi 4 W. Moćna karakteristika *Raspberri Pi*-a je red GPIO (*general-purpose input/output*) pinova duž gornje ivice ploče. Četrdesetopinska kontaktna letvica (GPIO pinovi) koja se nalazi na svim trenutnim pločama *Raspberry Pi*-a i pinovi imaju određeni raspored (Slika 2).

Luka Tomić, Marko Bijelić, Lazar Milovanović, Nemanja Savić i Živko Prolić, učenici trećeg razreda smera – Elektrotehničar informacionih tehnologija – *ogled Elektrotehnička škola „Mihajlo Pupin", Futoška,* 17, 2100 Novi Sad, Srbija, E-mail: luka.exe@yahoo.com



Slika 2. Izgled Raspberry Pi 3 [1] mikroračunara.



Slika 3. Raspored GPIO pinova.

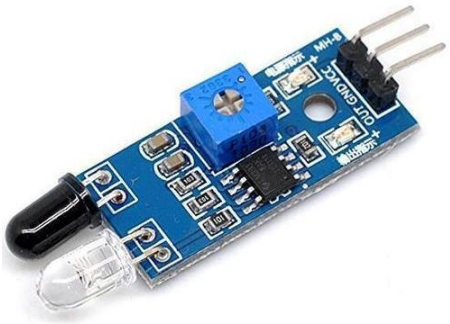
Uz *Raspberry Pi 3* računar, potrebna su 2 servo motora SG90 (Slika 4) koja otvaraju i zatvaraju kapiju na našoj maketi, kao i 2 IR (infracrvena) senzora (Slika 5). U šemi su uključeni i mikroprekidači radi kontrole položaja kapije i dodatne sigurnosti. Ovi senzori i prekidači se koriste isključivo na maketi (Slike 6 i 7), dok će se u realnoj upotrebi koristili namenski motori za otvaranje kapija, kao i potrebni senzori i prekidači.



Slika 4. Servo motor SG90.

Karakteristike servo motora SG90:

- Radni napon: +5V
- Radna brzina: 0.1s/60°
- Moment sile: 2,5kg/cm
- Rotacija: 0° - 180°



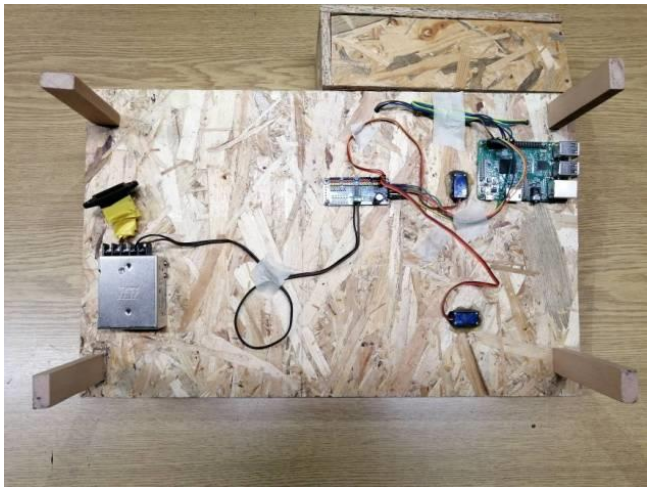
Slika 5. Infracrveni senzor.

Karakteristike infracrvenog senzora:

- Radni udaljenost: 2 cm-30 cm
- Radni napon: 3.3 V – 5 V
- Ugao detekcije: 35°



Slika 6. Izgled makete sa prednje strane.



Slika 7. Izgled makete sa zadnje strane.

B. Softverska implementacija

Softverska implementacija počinje od konfigurisanja rada *Raspberry Pi 3* sa GSM (*Global System for Mobile Telecommunications*) modulom. Instrukcije koje se koriste za kontrolu ovog modula nazivaju se AT komande. (AT naredbe se koriste i za kontrolu dial-up modema za linijski telefonski sistem.) Dial-up modemi, mobilni telefoni i GSM moduli podržavaju zajednički skup standardnih AT komandi[7]. Pored ovog skupa komandi, mobilni telefoni i GSM moduli podržavaju prošireni skup AT komandi. Jedna od takvih komandi je kontrola slanja i prijema SMS poruka. U nastavku je prikazan jednostavan primer koji pokazuje upotrebu AT naredbi za prijem SMS tekstualne poruke. Linije koda koje predstavljaju komande modemu prikazane su podebljane (bold) a odgovori GSM modula italikom.

ATE0
OK

AT
OK

AT+CMGF=1
OK

AT+CNMT=1,2,0,0,0
OK

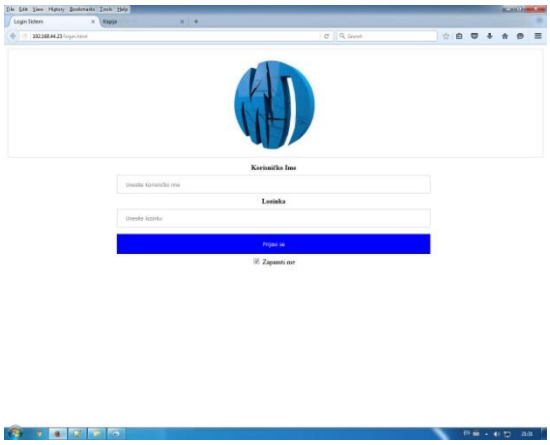
+CMT: “+381631050858”,””,”19/09/20,20:15:00+20”

Na ovaj način ostvaruje se prijem SMS poruka. U slučaju da je broj sa kog je poslata poruka registrovan u sistemu, tekst poruke se uporedjuje sa tekstovima tipskih poruka u bazi podataka. Ako je pronađen istovetan tekst, izvršava se radnja definisana tom porukom. Ukoliko istovetna poruka nije pronađena, kao i u slučaju da broj sa kog je poslata poruka nije registrovan u sistemu, poruka se ignoriše.

Početna forma sajta služi za prijavu korisnika (Slika 8). Na taj način korisnik unosi svoje ime i lozinku. Potom sledi stranica za izbor segmenta kojim admin želi da upravlja (Slika 9). Veb Server je *Flask* veb server. *Flask* je *mikro veb framework* napisan u *Python*-u [3]. Osnovni programski alat je *Python*. Upravljanje servo motorima i senzorima vrši se pomoću GPIO pinova, koji se kontrolišu putem funkcija iz biblioteke *Rpi.GPIO*.

IV. Zaključak

Cilj našeg rada je omogućavanje daljinske kontrole kapije korisnicima. Njima je značajno olakšano upravljanje upotrebom mobilnih te lefona. Prilikom izrade korišćeni su elementi solidnog kvaliteta, ali i pristupačne cene.



Slika 8. Izgled početne forme sajta.



Slika 9. Izgled naredne stranice.

Zahvalnica

Zahvaljujemo se mentorima Branislavu Ušanu, Vladimiru Skali i Draganu Đorđeviću na usmeravanju i sugestijama pri izradi projekta.

Literatura

[1] Raspberry Pi. Dostupno na: https://www.raspberrypi.org
[2] Flask. Dostupno na: http://flask.pocoo.org/docs/1.0/api
[3] Python. Dostupno na: https://www.python.org/doc/
[4] HTML. Dostupno na: https://www.w3schools.com/html/default.asp
[5] CSS. Dostupno na: https://www.w3schools.com/css/default.asp
[6] Java Script. https://www.w3schools.com/js/defa ult.asp
[7] AT komande. Dostupno na: https://www.developershome.com/sms/atCommandsIntro.asp

Sistem za upravljanje javnim garažnim prostorima

Nikola Simić, Marko Milijanović, Aleksandar Krajačević, Ognjen Milojević, Diana Šantavec

Sadržaj – U ovom radu je opisan princip rada sistema za upravljanje garažnim prostorom, kao i komponente i alati korišćeni za izradu ovog rada.

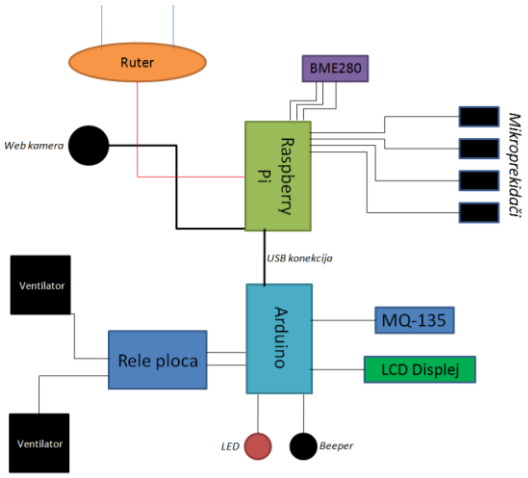
I. UVOD

Zadatak sistema za upravljanje garažnim prostorom je automatizacija upravljanja garažnim prostorima. Prikupljanje informacija sa pojedinačnih garažnih prostora je centralizovano (lokalni sistemi razmenjuju podatke sa centralnim serverom).

Za izradu ovog projekta korišćeni su mikroračunar *Raspberry Pi* [1] i razvojno okruženje *Arduino* [2].

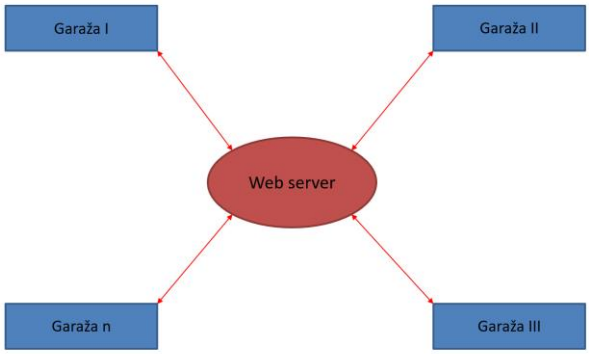
II. PRINCIP RADA

Lokalni garažni sistem (slika1) se sastoji od dve glavne celine: *sistem za merenje i kontrolu štetnih gasova* i *sistem za praćenje i rezervaciju slobodnih mesta*. Svaki garažni sistem razmenjuje svoje podatke sa centralnim web serverom (slika2) kome korisnici mogu da pristupe putem web interfejsa, te dobiju željene informacije i rezervišu parking mesto. Kontrolisanje i nadgedanje sistema se takođe vrši putem web interfejsa.



Slika 1. Blok šema garažnog Sistema.

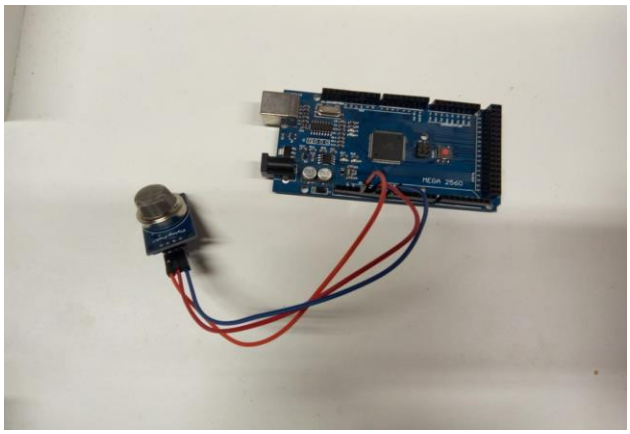
Nikola Simić, Marko Milijanović, Aleksandar Krajačević, Ognjen Milojević i Diana Šantavec, učenici četvrtog razreda smera – Elektrotehničar informacionih tehnologija, Elektrotehnička škola "Mihajlo Pupin", Futoška 17, 21102 Novi Sad, Srbija; Email: nikola7simic@gmail.com



Slika 2. Razmena podataka između garažnih sistema i centralnog server.

A. Sistem za merenje i kontrolu štetnih gasova

Korisćenjem *MQ-135* [3] (slika3) senzora povezanog na *Arduino* proračunava se količina štenih gasova u garaži, prosleđuju se informacije *Raspberry Pi*-u preko *USB* porta i uključuju se alarmi i upozorenja na osnovu očitanih vrednosti.



Slika 3. *MQ135* seznor povezan na *Arduino*.

MQ serija senzora za gas koristi mali grejač sa elektro-hemijskim senzorom. Osetljivi su na niz gasova i koriste se u zatvorenom prostoru na sobnoj temperaturi. Ovaj sensor poseduje 4 pina:

1) Analogni pin (vraća analogne vrednosti koje se očitavaju uz pomoć *Arduina*)
2) Digitalni pin (vraća 0 ili 1)

3) *GND* pin (zemlja)

4) *VCC* pin (napajanje od 5V)

Postoje dva nivoa alarma koji su podeljeni na osnovu toga kolika je količina gasova u ppm (parts per milion):

1) Prvi nivo - povećan nivo štetnih gasova (izmedju 150 i 250 ppm). Na ovom nivou se uključuju ventilatori čija je funkcija da provetravaju garažu u slučaju da nivo štetnih gasova nastavi da raste.

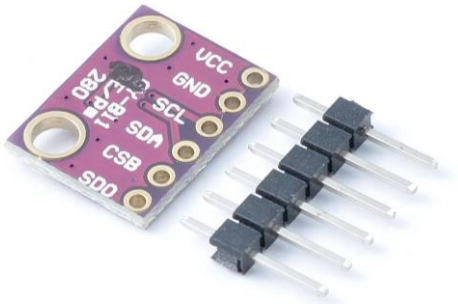
2) Drugi nivo - nivo štetnih gasova je prešao granicu od 250ppm. Na ovom nivou ventilatori nastavljaju da rade, uključuju se crvene diode u garaži, uključuje se LCD sa porukom "Napustite garažu!", "Opasnost od trovanja gasovima!" i uključuje se sirena.

B. Sistem za praćenje i rezervaciju slobodnih mesta

Sistem prati stanje slobodnih i zauzetih mesta u garaži uz pomoć mikroprekidača (na maketi, dok se u realnom sistemu koriste senzori sa preporučenim karaktreristikama za profesionalnu upotrebu) i očitava sledeće parametre sa *BME280* [4] (slika4) senzora: temperaturu, vlažnost vazduha i vazdušni pritisak.

Mikroprekidači povezani na *Raspberry Pi* vraćaju signal u slučaju da su pritisnuti (na maketi ako igračka auto stane na parking mesto) i sistem registruje da je to mesto zauzeto.

Informacije bitne za krajnje korisnike, u realnom vremenu se razmenjuju sa centralnim serverom.



Slika 4. BME280 senzor.

Korisnik koji želi da rezerviše mesto, upisuje svoj e-mail u formu na web aplikaciji i pritiska dugme za rezervaciju. Na unet email stiže potvrda da je mesto rezervisano i generisani QR kod koji se takođe prosleđuje sistemu garaže u kojoj je mesto rezervisano. Po pristizanju do garaže korisnik prislanja svoj telefon sa QR kodom na sekener (na maketi skener nam je web kamera i odgovarajući python kod) koji očitava QR kod, proverava da li je validan i otvara kapiju garaže ako jeste. Podrazumevano vreme do isteka važenja QR koda je 15 minuta (može se promeniti).

Podaci o korisnicima koji su rezervisali mesta se čuvaju u sistemu. Vlasnik garaže redovnim korisnicima omogućuje niz privilegija (niže cene, prioritet pri za rezervaciji u odnosu na ostale korisnike, periodično plaćanje itd.)

C. Web aplikacija

Web aplikacija (slika5) se nalazi na centralnom serveru, razmenjuje podatke sa lokalnim sistemima, prikazuje ih korisnicima i pruža im mogućnost za rezervisanje slobodnih mesta.

Kada korisnik pristupa web aplikaciji nudi mu se opcija da prosledi podatke o svojoj lokaciji kako bi sistem mogao pronaći njemu najbližu garažu sa slobodnim mestom.



Slika 5. Web aplikacija.

III. IMPLEMENTACIJA

A. Hardverska implementacija

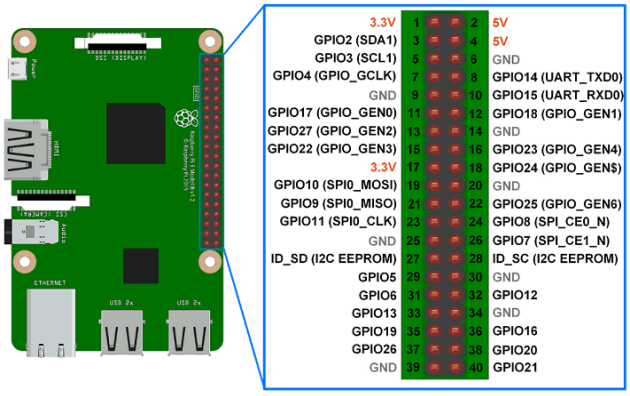
Prva stavka potrebna za implementaciju je mikroračunar *Raspberry Pi 3* [1] (slika6).



Slika 6. *Raspberry Pi 3* model B.

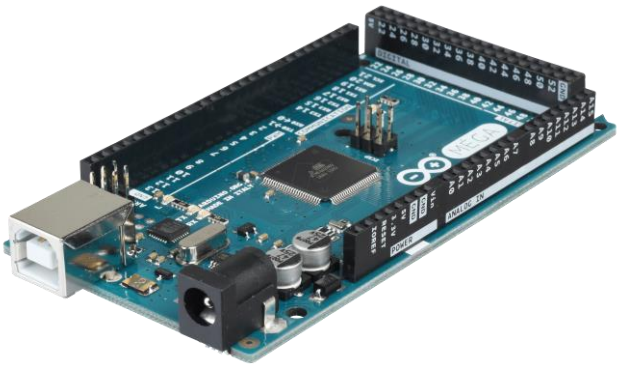
Instarliran je operativni sistem Raspbian. Koristi 1200 MHz quad-core *ARM Cortex-A53* procesor. Poseduje memoriju od 1GB. Za skladištenje koristi *MicroSDHC* slot kao i njegovi prethodnici. Grafiku predstavlja *Broadcom VideoCore IV* ali na većim *clock* frekvencijama. Snaga za napajanje ovog modela iznosi 4.0 W.

Moćna karakteristika *Raspberri Pi*-a je red *GPIO* (*general-purpose input/output*) pinova duž gornje ivice ploče. GPIO pinovi imaju određeni raspored (slika7).



Slika 7. Raspored *GPIO* pinova na *Raspberry Pi 3* ploči.

Sledeća stavka jeste razvojno okruženje *Arduino MEGA 2560* [2] (slika8).



Slika 8. *Arduino Mega 2560.*

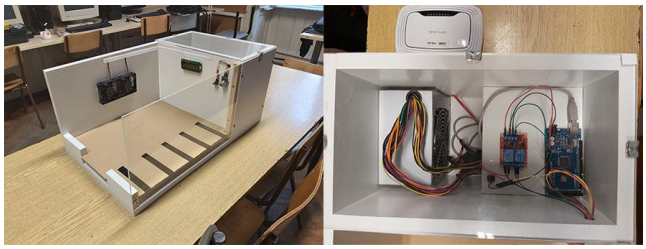
Arduino MEGA je ploča koja poseduje mikrokontroler zasnovan na *ATmega2560*. Ima 54 digitalna ulaza, 4 *UART* (hardverski serijski portovi), kristalni oscillator od 16MHz, *USB* priključak, priključak za napajanje, *ICSP* zaglavlje i taster za resetovanje.

Pored opisanog *MQ-135* senzora u radu se koristi i *BME280* [4].

BME280 je digitalni senzor barometrijskog pritiska i malo je nadograđena verzija *BMP180*. Dostupan je na malom modulu koji omogućava pristup senzoru preko *I2C* (*inter-integrated circuit*) [5] interfejsa. To nam omogućava da ga lako povežemo sa *Raspberry Pi*-em i očitavamo podatke. *BME280* poseduje mogućnost očitavanja temperature, vazdušnog pritiska i vlažnosti vazduha. Raspored pinova na ovom senzoru je sledeći:

1) *VCC* pin (napajanje)
2) *GND* pin (zemlja)
3,4) *SCL* i *SDA* pinovi (koriste se za povezivanje preko *I2C* interfejsa)
5,6) *CSO* i *SDO* pinovi (koriste se za povezivanje preko *SPI* interfejsa)

Pored ovih glavnih komponenata i senzora, tu su i ostale elektronske komponente poput LCD displeja, LED dioda, relea, otpornika, itd.



Slika 9. Maketa garažnog sistema.

B. Softverska implementacija

Softverska implementacija počinje od pisanja koda u programskom jeziku *Python* [6] (slika10) za *Raspberry Pi*. Na *Raspberry Pi*-u se piše kod za:

1) Funkcionalnost *sistema za paraćenje i rezervaciju slobodnih mesta*
2) Funkcionalnost web aplikacije (u kombinaciji sa *JavaScriptom* [7])

Kod za upravljanje *sistemom za merenje i kontrolu štetnih gasova* (koji se nalazi na *Arduinu*) pišemo u programskom jeziku *C* [8] (slika11).

Stranice za web aplikaciju su izradjene korišćenjem *HTML*-a i *CSS*-a.

```
1
2   def Mesta():
3       mesta = ["","","",""]
4
5       if (GPIO.input(pmesto) == 1):
6           mesta[0] = "1"
7       else:
8           mesta[0] = "0"
9
10      if (GPIO.input(pmesto2) == 1):
11          mesta[1] = "1"
12      else:
13          mesta[1] = "0"
14      if (GPIO.input(pmesto3) == 1):
15          mesta[2] = "1"
16      else:
17          mesta[2] = "0"
18      if (GPIO.input(pmesto4) == 1):
19          mesta[3] = "1"
20      else:
21          mesta[3] = "0"
22
23      return mesta
```

Slika 10. Primer koda u *Python*-u (funkcija za čitanje slobodnih i zauzetih mesta).

```
if (gas_current_value >= gas_high_value){
        tone(beeper_port,500);
        lcd.begin(16,2);
        lcd.print("opasnost");
        lcd.setCursor(0,1);
        lcd.print("gasovi");
        digitalWrite(relay_port_1, HIGH);
        digitalWrite(relay_port_2, HIGH);
      }
```

Slika 11. Premer koda u *C*-u (uključivanje drugog nivoa alarma).

Centralni server je standardni web server (Apache, PHP, MySQL), zakupljen kod hosting provajdera. Lokalni sistemi u realnom vremenu prosleđuju informacije o slobodnim parking mestima centralnom serveru. Centralni server lokalnim sistemima prosleđuje rezervacije za parkiranje (sa generisanim QR kodom), putem interneta.

IV. Zaključak

Sistem za upravljanje javnim garažnim prostorom pruža korisnicima niz informacija i omogućuje im jednostavnu rezervaciju parking mesta, čime im štedi dragoceno vreme. Unapređenja sistema, predviđena za naredni period: U slučaju trenutnog prekida rada centralnog servera, korisnicima će (preko rezervnog linka) biti omogućen pristup lokalnom serveru izabrane garaže.

Zahvalnica

Zahvaljujemo se mentorima Bransilavu Ušanu i Jovi Pavloviću na usmeravanju i sugestijama pri izradi ovog rada.

Literatura

[1] Raspberry Pi: https://www.raspberrypi.org
[2] Arduino: https://store.arduino.cc/mega-2560-r3
[3] MQ135: https://playground.arduino.cc/Main/MQGasSensors/
[4] BME280: https://www.raspberrypi-spy.co.uk/2016/07/using-bme280-i2c-temperature-pressure-sensor-in-python/
[5] I2C: https://i2c.info
[6] Python: http://www.ubunturs.org/preuzimanje/piton/zagrljaj_pitona_latinica.pdf
[7] JavaScript: https://www.w3schools.com/js/default.asp
[8] C: https://www.tutorialspoint.com/cprogramming/index. htm

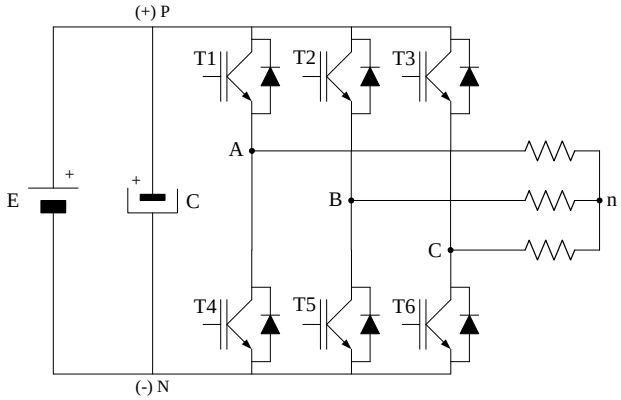
Praktična realizacija elektronskog kola trofaznog tranzistorskog invertora

Jelena Komad, Katarina Đukić

Sadržaj – U ovom radu prezentovana je mogućnost praktične realizacije trofaznog tranzistorkog invertora koji se upravlja mikrokontrolerskom karticom. Prezentovano kolo služi i za testiranje jednostavnijih tehnika upravljanja trofaznim invertorom. U završnom delu rada prikazani su karakteristični talasni oblici jedne faze izlaznog napona, kao i odgovarajući upravljački signali.

I. TROFAZNI TRANZISTORSKI INVERTOR

Trofazni invertori imaju veoma čestu primenu u praksi [1],[2]. Koriste se za regulaciju brzine obrtanja trofaznih asnihronih motora i sihronih motora sa premanentnim magnetima na rotoru [2]. Posebno je važno naglasiti da trofazni invertori imaju veoma veliku primenu u sistemima obnovljivih izvora energije. Najjednostavnija topologija trofaznog invertora sadrži šest tranzistora u mosnom spoju kao što je prikazano na Slici 1.



Slika 1. Principska šema trofaznog naponskog tranzistorskog invertora.

Upravljanje tranzistorima se vrši primenom mikrokontrolera ili digitalnih signalnih procesora u zavisnosti od tehnike upravljanja. Za jednostavnije tehnike upravljanja koriste se mikrokontroleri, a za složenije i zahtevnije tehnike primenjuju se digitalni signalni procesori (DSP). Za pogon trofaznih invertora na tržištu postoje razvijene posebne serije DSP-a pod oznakom *''motor control''*, a jedna od najpoznatijih je serija kompanije *Texas Instruments*.

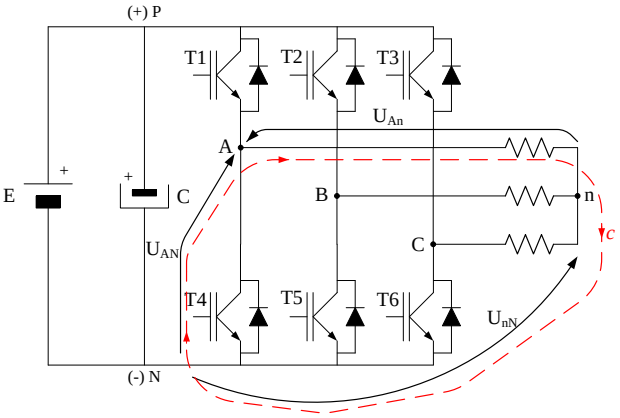
Jelena Komad i Katarina Đukić, smer elektrotehničar procesnog upavljanja, Elektrotehnička škola "Mihajlo Pupin", Futoška 17, 21000 Novi Sad, Srbija, E-mail: jelenakomad@outlook.com

Upravljanje trofaznim invertorima može se realizovati kroz više različitih algoritama. Na ovom mestu biće razmotrena jedna od najjednostavnijih tehnika koja se naziva šestopulsna tehnika ili šestonivojna tehnika. Naziv je dobila jer se napon na fazi trofaznog prijemnika sastoji od šest karakterističnih nivoa od kojih su tri u pozitivnoj,a tri u negativnoj poluperiodi. Zato se invertori koji realizuju ovu tehniku upravljanja nazivaju šestopulsni ili šestonivojni naponski invertori. U planovima nastave i učenja oblast trofaznih invertora se pažljivo izučava kroz teorijsku nastavu premeta Energetska elektronika [2], a osnovni teorijski principi se praktično proveravaju na laboratorijskim vežbama [3].

A. Osnovna analiza trofaznog invertora

Da bi se izvršila analiza rada šestopulsnog invertora neophodno je definisati određene napone na invertoru pomoću kojih će se doći do odgovarajućeg algoritma upravljanja i izraza za fazni i linijski napon prijemnika. Zbog jednostavnosti, usvojićemo da je prijemnik čisto omskog karaktera.

Na Slici 2 naznačeni su određeni naponi U_{AN}, U_{An}, U_{nN} kao i njihovi referentni smerovi. Analogno važi za drugu i treću fazu. Zato će na ovom mestu biti prikazano izvođenje izraza za izlazni napon samo prve faze.



Slika 2. Referentni smerovi napona na principskoj šemi trofaznog naponskog tranzistorskog invertora.

Ako se za naznačenu konturu sa slike 2 napiše naponska jednačina, tada važi:

$$U_{AN} = U_{An} + U_{nN}. \tag{1}$$

Pri čemu je U_{AN} napon između tačke A prve grane trofaznog invertora i negativnog pola jednosmernog međukola, napon U_{An} je između tačke A prve grane trofaznog invertora i zvezdišta potrošača, a napon U_{nN} je napon između zvezdišta potrošača i negativnog pola jednosmernog međukola saglasno slici 2. Iz prethodne jednačine nepohodno je izraziti fazni napon U_{An}, primenom istog postupka za drugu i treću fazu može se pisati:

$$\begin{aligned} U_{An} &= U_{AN} - U_{nN}, \\ U_{Bn} &= U_{BN} - U_{nN}, \\ U_{Cn} &= U_{CN} - U_{nN}. \end{aligned} \tag{2}$$

Za trofazni simetrični sistem važi da je zbir fazora faznih napona u svakom trenutku jednak nuli, odnosno može se pisati $\underline{U}_{An}+\underline{U}_{Bn}+\underline{U}_{Cn}=0$. Uvažavajući prethodno navedenu činjenicu, moguće je definisati napon između negativnog pola jednosmernog kola invertora i zvezdišta trofaznog prijemika.

$$\begin{aligned} &\underbrace{U_{An} + U_{Bn} + U_{Cn}}_{0} = \\ &U_{AN} + U_{BN} + U_{CN} - 3U_{nN} \Rightarrow \\ &U_{nN} = \frac{1}{3}(U_{AN} + U_{BN} + U_{CN}) \end{aligned} \tag{3}$$

Primenom jednačina (2) i (3) i zamenom izraza (3) u izraz (2) fazni naponi trofaznog prijemnika biće definisanani samo preko napona U_{AN}, U_{BN}, U_{CN} koji definišu potencijale tačaka A, B, C prema negativnom polu jednosmernog kola trofaznog invertora.

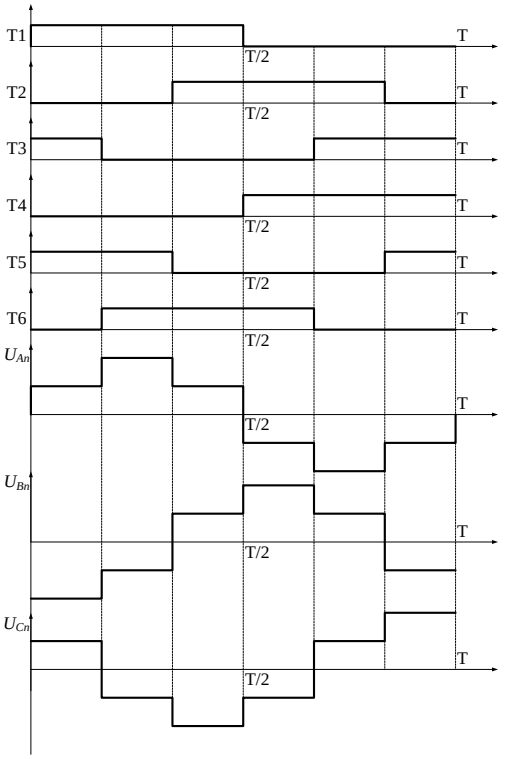
$$\begin{aligned} U_{An} &= U_{AN} - \frac{1}{3}(U_{AN} + U_{BN} + U_{CN}), \\ U_{Bn} &= U_{BN} - \frac{1}{3}(U_{AN} + U_{BN} + U_{CN}), \\ U_{Cn} &= U_{CN} - \frac{1}{3}(U_{AN} + U_{BN} + U_{CN}). \end{aligned} \tag{4}$$

U izrazima (4) neophodno je definisati napone U_{AN}, U_{BN}, U_{CN} . Sa slike 2 može se uočiti da vrednost napona U_{AN} ima vrednost E ako je uključen gornji tranzistor u grani (T1), odnosno ima vrednost 0 ako je uključen donji tranzistor u grani (T4). Identično vredi za drugu i treću fazu. Zato je nepogodno definisati promenljivu S za svaku granu invertora, koja određuje stanje tranzistora.

$$\begin{aligned} U_{AN} &= E \cdot S_A, \\ U_{BN} &= E \cdot S_B, \\ U_{CN} &= E \cdot S_C. \end{aligned} \tag{5}$$

Pri čemu je E napon jednosmernog međukola, a S_A, S_B, S_C su stanja tranzistora u granama invertorskog mosta. Promenljiva S ima vrednost 1 ako je uključen gornji tranzistor u grani, a vrednost 0 ako je uključen donji tranzistor u grani. Na osnovu prethodnih izraza mogu se definisati izrazi za fazne napone trofaznog prijemnika.

$$\begin{aligned} U_{An} &= E \cdot \left[S_A - \frac{1}{3}(S_A + S_B + S_C) \right], \\ U_{Bn} &= E \cdot \left[S_B - \frac{1}{3}(S_A + S_B + S_C) \right], \\ U_{Cn} &= E \cdot \left[S_C - \frac{1}{3}(S_A + S_B + S_C) \right]. \end{aligned} \tag{6}$$



Slika 3. Vremenski dijagrami upravljačkih signala i talasni oblici faznih napona trofaznog prijemnika.

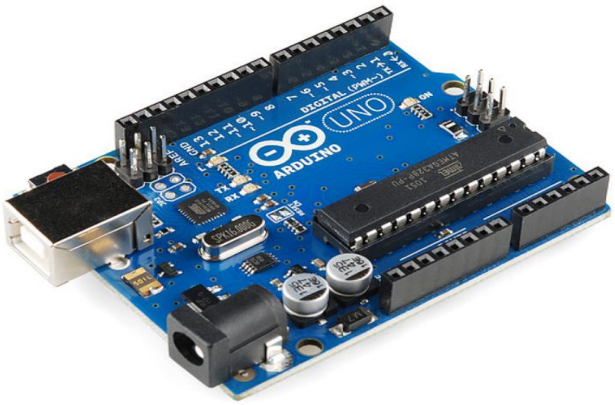
Uključenje tranzistora u invertorskom mostu može da bude realizovano na dva načina, uglom vođenja 180° stepeni i uglom vođenja 120° stepeni.

Na Slici 3 prikazani su pregledno vremenski dijagrami upravljačkih signala tranzistora od T1 do T6 kao i talasni oblici faznih napona trofaznog prijemnika koji je povezan i grane mosnog invertora za ugao vođenja 180° stepeni.

II. MIKROKONTROLERSKA UPRAVLJAČKA KARTICA

Arduino platforma je skup elektronskih i softverskih komponenti koje se mogu jednostavno povezivati u složenije celine. Srce Arduina je mikrokontroler. Mikrokontroler je mali računar na jednom integrisanom sklopu. Arduino okruženje najčešće koristi 8- bitne mikrokontrolere koje proizvodi kompanija ATMEL. Najrasprostranjeniji model je ATMEGA328P koji se koristi na osnovnoj Arduino prototipnoj pločici koja je prikazana na Slici 4. Arduino UNO R3 ploča sadrži 14 digitalnih ulazno/izlaznih pinova (od kojih 6 mogu da se koriste kao PWM izlazi), 6 analognih ulaza, 16 MHz keramički rezonator, USB konektor, priključak za napajanje i dugme za resetovanje. Mikrokontroler se jednostavno povezuje na PC preko USB konektora. Preko USB kabla Arduino će biti napojen sa 5V jednosmernim naponom.

Na Slici 4 prikazan je izgled Arduino upravljačke mikrokontrolerske kartice [4].



Slika 4. Izgled Arduino mikrokontrolerske upravljačke kartice.

III. HARDVERSKA REALIZACIJA I REZULTATI TESTIRANJA

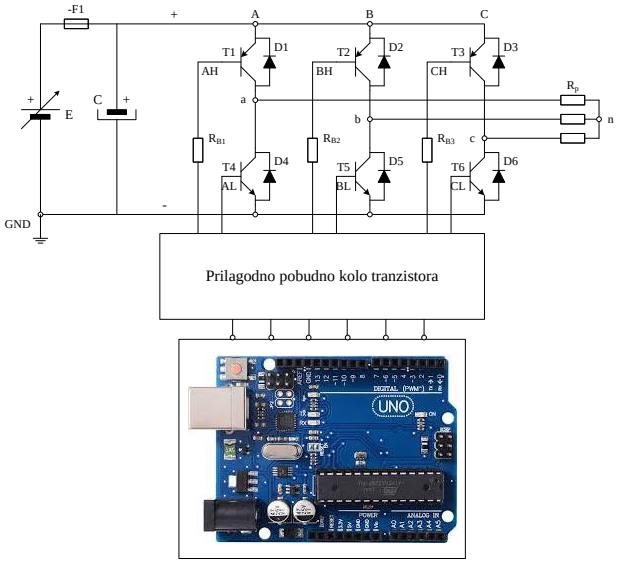
Pre pristupanja hardverskoj realizaciji trofaznog naponskog tranzistorskog invertora neophodno je u potpunosti se upoznati sa osnovnom hardverskom organizacijom elektronskog kola. Zatim je neophodno savladati osnovne elemente mikrokontrolerske Arduino Uno upravljačke kartice i načina programiranja mikrokontrolera. Treći korak podrazumeva primenu softverskog rešenja i testiranje rada elektronskog kola.

Na Slici 5 prikazana je principska šema hardverske realizacije elektronskog kola trofaznog tranzistorskog invertora koja je korišćena u ovom radu. Pre hardverske realizacije pristupljeno je prvo sintezi hardveskog kola invertora na proto pločici sa ciljem testiranja rada elektronskog kola i algoritma upravljanja.

TABELA I
PRIMENJENI TRANZISTORI U INVERTORSKOM MOSTU

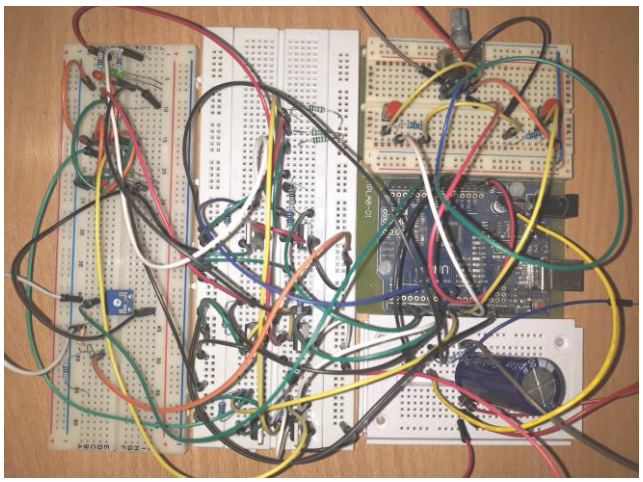
T1, T2, T3	BDX 34C	PNP
T4, T5, T6	BDX 33C	NPN

Na Slici 6 prikazan je izgled elektronskog kola povezanog na proto pločici za testiranje rada elektronskog kola i ispitivanje algoritma upravljanja.

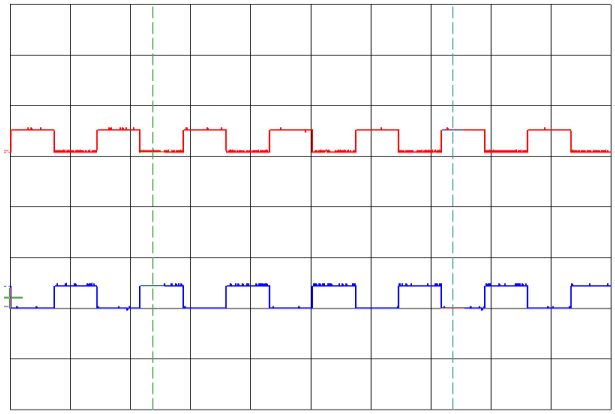


Slika 5. Principska šema hardverske realizacije trofaznog tranzistorskog invertora.

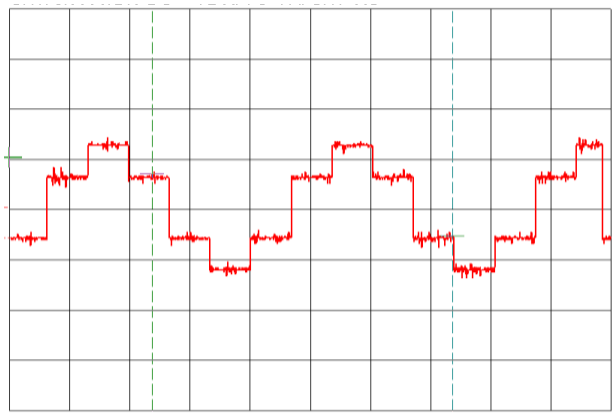
Na Slici 7 snimljeni su talasni oblici prvog i četvrtog tranzistora u prvoj grani invertorskog mosta. Na Slici 7 jasno se uočava da se tranzistori u jednog grani invertorskog mosta naizmenično uključuju, odnosno kada je uključen gornji tranzistor u prvoj grani (T1 uključen, pobuda na AH uključena) donji tranzistor u toj istoj grani je isključen (T4 isključen, pobuda AL isključena). Odnosno, u istoj grani tranzistorskog invertorskog mosta oba tranzistora nikada ne vode istovremeno, saglasno opisanom upravljačkom algoritmu sa Slike 3. Na Slici 8 prikazan je vremenski dijagram prve faze trofaznog prijemnika. Signal je snimljen digitalnim osciloskopom HAMEG 507 čija je sonda priključena u tačku "**a**" prve grane invertorskog mosta, a masa osciloskopa je povezana u zvezdište trofaznog prijemnika "**n**" (Slika 5). Na Slici 9. Prikazan je talasni oblik linijskog napona trofaznog prijemnika. Talasni oblik je snimljen digitalnim osciloskopom čija sonda je priključena u tačke "***a***" i "***b***", prve i druge grane trofaznog invertorskog mosta (Slika 5).



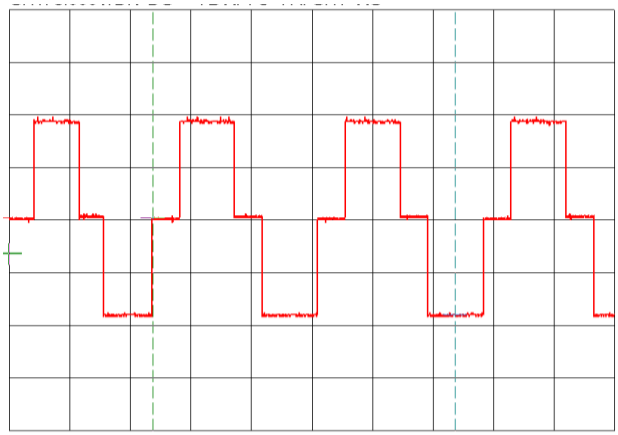
Slika 6. Izgled elektronskog kola za povezanog na proto pločici za testiranje rada elektronskog kola i ispitivanje algoritma upravljanja .



Slika 7. Upravljački signali prvog i četvrtog tranzistora u invertorskom mostu.



Slika 8. Talasni oblik napona prve faze trofaznog prijemnika.



Slika 9. Talasni oblik linijskog napona trofaznog prijemnika.

IV. ZAKLJUČAK

U ovom radu prikazan je jedan način praktične realizacije trofaznog invertora i testiranja najjednostavnije tehnike upravljanja.

Radom je obuhvaćena sinteza teorijskih znanja i praktičnih veština.

Posebna vrednost je u tome što su praktično snimljeni talasni oblici karakterističnih signala u elektronskom kolu odgovaraju teorijskim principima rada razmatranog elektronskog kola.

Može se iskoristiti za testiranje algoritama upravljanja, analizu rada elektronskih kola trofaznih invertora i obuku mlađih generacija učenika iz ove oblasti.

Dalji pravci rada na ovoj temi obuhvatiće izradu malog prototipa, frekventnog pretvarača za edukativnu namenu.

ZAHVALNOST

Na ovom mestu želimo da uputimo zahvalnost našem mentoru Skoko Saši, koji nam je pomogao u savladavanju teorijskih znanja o invertorima i pomogao savetima u praktičnoj realizaciji elektronskog kola.

LITERATURA

[1] R. Opačić, *Elektronika I*, Beograd, Zavod za udžbenike i nastavna sredstva, 1993.

[2] D. Martinović, Z. Pendić, Ž. Janda, J. Menart, *Energetska elektronika*, Beograd, Zavod za udžbenike i nastavna sredstva, 2007.

[3] S. Skoko, P. Kolarski, *Energetska elektronika-praktikum laboratorijskih vežbi*, Novi Sad, Elektrotehnička škola "Mihajlo Pupin", 2018.

[4] https://www.arduino.cc/

Praktična realizacija ispitne stanice za upravljanje trofaznim asinhronim motorom primenom frekventnog pretvarača i PLC-a

Božidar Čičak, Nemanja Eror, Daniel Folkman, Denis Tinjić, Danis Čivović

Sadržaj – U ovom radu prezentovan je jedan način praktične realizacije ispitne stanice za upravljanje trofaznim asinhronim motorom primenom frekventnog pretvarača i programabilnog logičkog kontrolera. Ispitna stanica omogućava snimanje regulacionih karakteristika za U/f upravljanje i testiranje algoritama kada PLC upravlja frekventnim pretvaračem,a pretvarač asinhronim motorom. U završnom delu rada dati su neki od rezultata testiranja.

I. TROFAZNI ASINHRONI MOTOR

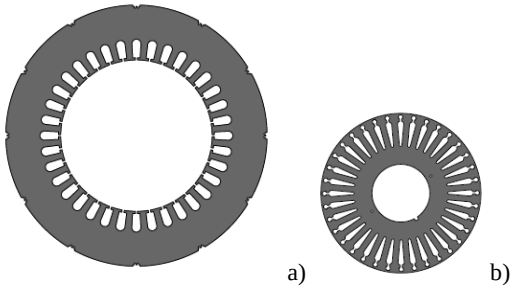
Asinhrone mašine su najrasprostranjenije električne mašine. Uglavnom se koriste kao električni motori, ređe kao generatori [1]. Imaju široku primenu kao elektromotorni pogoni većine proizvodnih mehanizama u industriji, uređajima automatike, kućnih aparata, itd. Proizvode se u širokom opsegu snaga, od delova vata do nekoliko desetina megavata. Široka primena asinhronih motora zasnovana je na njihovoj prostoj konstrukciji, pouzdanosti u radu, dobrim eksploatacionim svojstvima, niskoj ceni i jednostavnom održavanju. Na Slici 1 prikazan je izgled trofaznog asinhronog motora opšte namene [1], [2].



Slika 1. Trofazni asinhroni motor opšte namene.

A. Osnovna konstrukcija trofaznog asinhronog motora

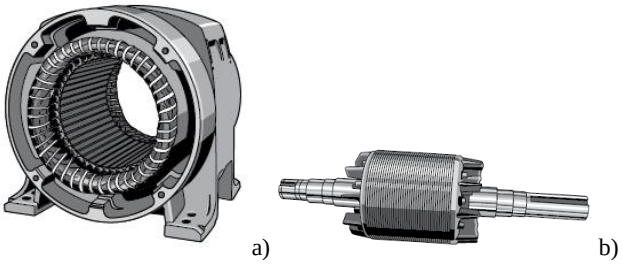
Osnovni delovi asinhrone mašine su: stator (nepokretan deo) i rotor (obrtan deo). Stator je u vidu šupljeg valjka, složenog od limova. Duž valjka, na njegovom unutrašnjem delu, po celom obimu nalaze se žljebovi, a u njima izolovani provodnici, povezani tako da obrazuju višefazni, obično trofazni namotaj. U unutrašnjosti statora asinhrone mašine nalazi se rotor koji se sastoji od naslage limova napresovane na vratilo, na čijem se spoljnjem delu nalaze žljebovi u kojima su provodnici povezani tako da obrazuju namotaj rotora. Radi smanjenja gubitaka usled vrtložnih struja, limovi statora i rotora su uzajamno izolovani. Na Slici 2 prikazan je izgled dinamo limova statora i rotora asinhronog motora [1], [2].



a) b)

Slika 2. Izgled dinamo limova trofaznog asinhronog motora: a) stator, b) rotor.

Rotori se u zavisnosti od tipa namotaja dele na kratkospojene i fazne. Fazni rotor se često naziva i rotor sa prstenovima (kliznokolutni) ili namotani rotor. Na Slici 3 prikazan je izgled statora i rotora asinhronog motora.



a) b)

Slika 3. Osnovni konstrukcioni delovi asinhronog motora: a) stator, b) kratko spojeni (kavezni) rotor.

B. Osnovni principi U/f upravljanja

Iz izraza za sinhronu brzinu vidi se da je ona direktno zavisna od frekvencije napajanja statora (1) [2].

$$n_s = \frac{60\, f_s}{p}\left[\frac{ob}{min}\right] \tag{1}$$

B. Čičak, N. Eror, D. Folkman, D. Tinjić, D. Čivović, tehničar mehatronike, Elektrotehnička škola" Mihajlo Pupin", Futoška 17, 21000 Novi Sad, Srbija, E-mail: bokicicak09@gmail.com

Zato je u ovom slučaju neophodno asinhroni motor napajati iz specijalnog izvora promenljive frekvencije. Promena frekvencije napajanja statora utiče na radne karakteristike motora. Zato se ovaj način ostvaruje istovremenom promenom napona i frekvencije statora, pri čemu postoji direktna srazmera između promene ove dve veličine [2]. Razlog ovakvog načina napajanja motora je što postoji direktna veza između napona i frekvencije statora, i fluksa magnetnog polja u motoru. Ta relacija data je izrazom (2).

$$U_s = k f_s \Phi_s \tag{2}$$

Pri čemu *k* predstavlja konstruktivnu konstantu mašine. Iz izraza (2) uočava se da je neophodno istovremeno vršiti promenu napona i frekvencije statora kako bi vrednost fluksa u mašini za različite frekvencije napajanja ostala konstantna i jednaka nominalnoj vrednosti fluksa u mašini.

$$\frac{U_s}{f_s} = k\Phi_s = const. \Rightarrow \Phi_s = const. \tag{3}$$

Ovakav princip promene brzine obrtanja predstavlja osnovu za tzv. skalarno upravljanje asinhronim motorom (U/f upravljanje) [2]. Uređaji energetske elektronike koji služe za U/f upravljanje nazivaju se frekventni pretvrači.

II. FREKVENTNI PRETVARAČI

Frekventni pretvarač (Slika 4) je uređaj namenjen kontroli brzine elektromotora na naizmeničnu struju pomoću promene frekvencije i napona napajanja motora [3]. Princip rada se ostvaruje tako što se naizmenična struja prvo pretvara u jednosmernu pomoću ispravljača. Zatim se jednosmerna struja pretvara pomoću invertora zasnovanog na prekidačkim kolima u trofaznu naizmeničnu struju nalik sinusoidi. Svi savremeni frekventni pretvarači sadrže u sebi mikroprocesore i pored upravljanja brzinom motora imaju još dosta funkcija. Po svojoj unutrašnjoj strukturi frekventni pretvarači se sastoje od sledećih elemenata: ispravljač, jednosmerno međukolo, tranzistorski trofazni invertor, elektronsko upravljačko kolo. Regulacija brzine skalarnim upravljanjem, primenom frekventnog pretvarača, ostvaruje se u dve zone regulacije.

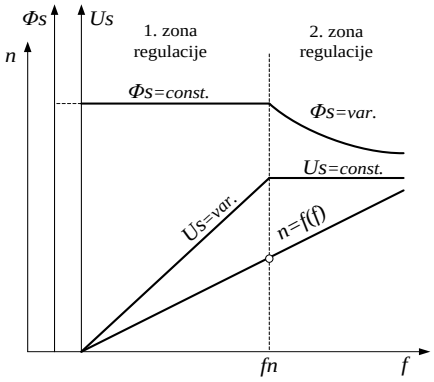
Prva zona regulacije podrazumeva promenu frekvencije od minimalne moguće (1-1,5 Hz, ili manje) do nominalne vrednosti (50 Hz).

Druga zona regulacije brzine ostvaruje se pri frekvencijama napajanja preko nominalne vrednosti. Frekventni pretvarač omogućava da se preko kontrolnog terminala zadaju različite frekvencije napajanja motora. Na taj način možemo vršiti višebrzinsko upravljanje (multi-step speed reference) zadavajući različite vrednosti frekvencija. Na Slici 6 prikazan je izgled kontrolnog terminala a na Slici 7 osnovni princip ožičenja [3].

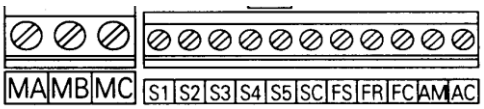
Na Slici 5 prikazan je izgled regulacionih karakteristika skalarnog upravljanja asinhronim motorom [2].



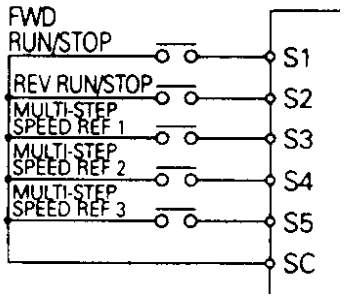
Slika 4. Izgled klasičnog frekventnog pretvarača.



Slika 5. Izgled regulacionih karakteristika za U/f upravljanje.



Slika 6. Osnovni izgled kontrolnog terminala.

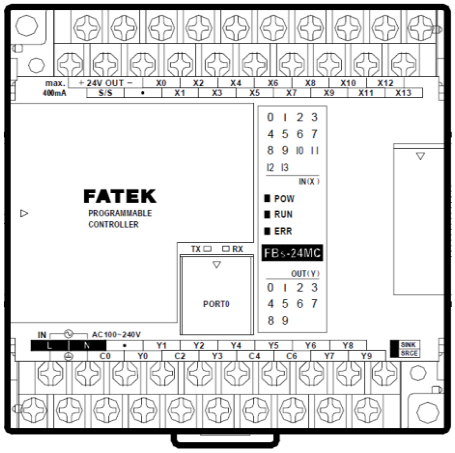


Slika 7. Osnovni princip ožičenja kontrolnog terminala za višebrzinsko upravljanje (multi-step speed reference).

III. PROGRAMABILNI LOGIČKI KONTROLER

PLC je specijalna vrsta računara, čija je namena upravljanje industrijskim procesima.Oni sadrže procesor i memoriju, ali imaju i specijalne ulaze i izlaze po kome se razlikuju od računara i nemaju spoljnu memoriju.

Za praktičnu realizaciju ispitne stanice korišćen je industrijski PLC kompanije FATEK čiji je izgled prikazan na Slici 8 [4].



Slika 8. Izgled prednje ploče programabilnog logičkog kontrolera FATEK.

IV. PRAKTIČNA REALIZACIJA ISPITNE STANICE I REZULTATI TESTIRANJA

Praktična realizacija ispitne stanice je tako osmišljena da omogućava niz praktičnih vežbanja i demonstracija. Na Slici 9 prikazan je izgled ispitne stanice.

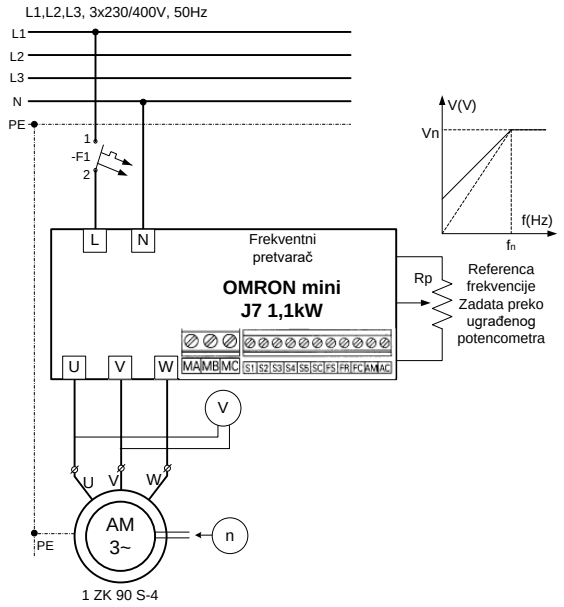


Slika 9. Izgled ispitne stanice za upravljanje trofaznim asinhronim motorom.

Na Slici 10 prikazana je osnovna šema ispitne stanice za snimanje regulacionih karakteristika U/f upravljanja.

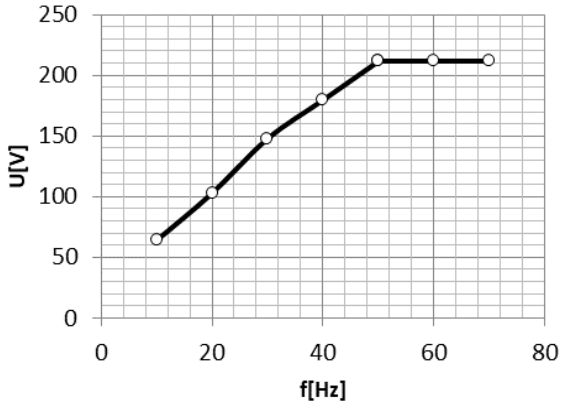
TABELA II
OSNOVNA PROMENA VELIČINA NA MOTORU ZA U/F UPRAVLJANJE

f	[Hz]	10	20	30	40	50	60	70
U	[V]	64	103	148	180	212	212	212
n	[min^{-1}]	295	595	887	1192	1490	1785	2077
$K_{U/f}$	[V/Hz]	6,4	5,2	4,9	4,5	4,2	3,5	3

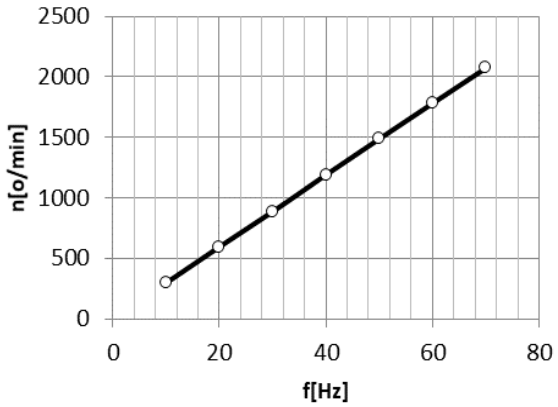


Slika 10. Principska šema ispitne stanice za snimanje regulacionih karakteristika.

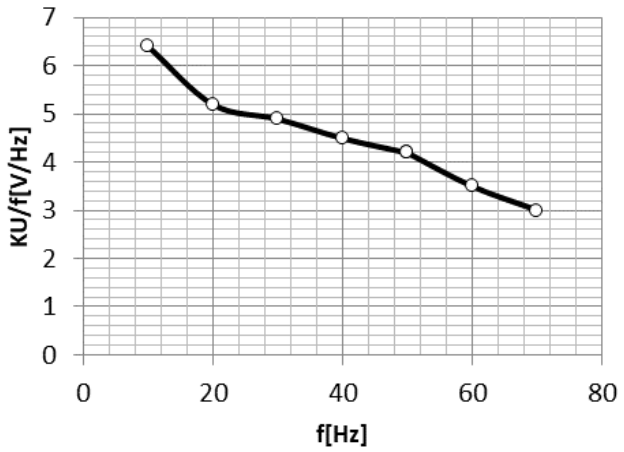
Na Slikama 11, 12, i 13 prikazani su dijagrami regulacionih karakteristika snimljenih na ispitnoj stanici.



Slika 11. Dijagram zavisnosti napona od frekvencije.



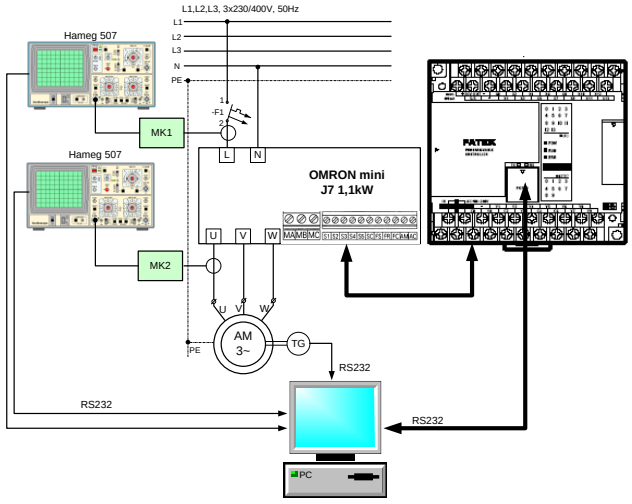
Slika 12. Dijagram zavisnosti brzine od frekvencije.



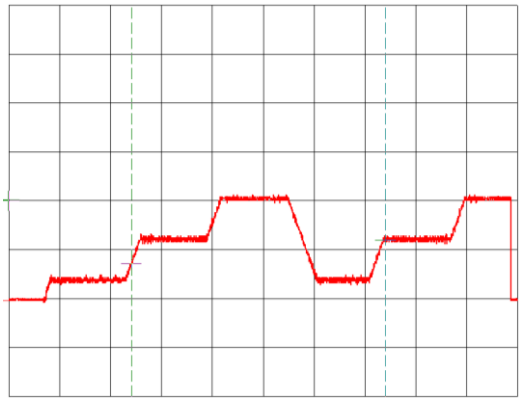
Slika 13. Dijagram zavisnosti koeficijenta U/f od frekvencije.

Na Slici 14 prikazana je principska šema ispitne stanice za testiranje višebrzinskog upravljanja. Na Slici 15 i Slici 16 prikazan je dijagram promene brzine za naznačene reference frekvencije napajanja motora.

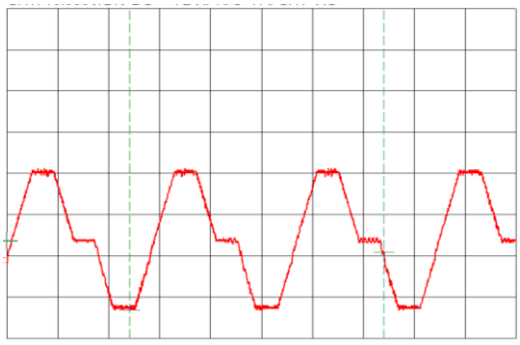
Na Slikama 17 i 18 prikazani su vremenski dijagrami ulazne i izlazne struje frekventnog pretvarača. Snimanje vremenskih dijagrama izvršeno je posebnom strujnom mernom karticom koja je zasnovana na LEM sindi LA 25 NP.



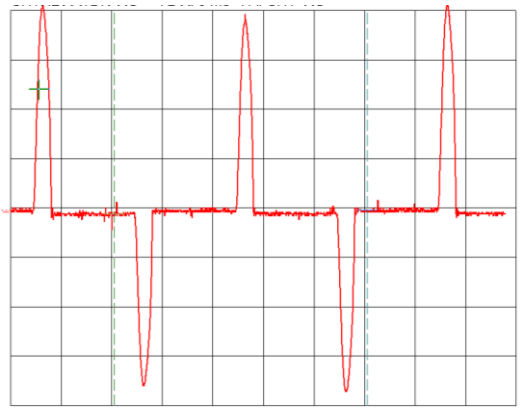
Slika 14. Principska šema za višebrzinsko upravljanje trofaznim asinhronim motorom primenom frekventnog pretvarača i PLC-a.



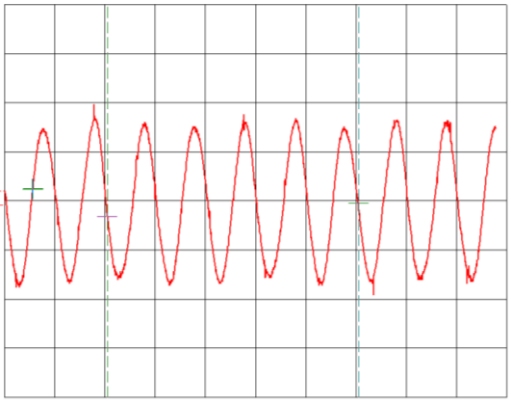
Slika 15. Snimljeni dijagram promene brzine obrtanja za reference frekvencije 10/30/50 Hz.



Slika 16. Snimljeni dijagram promene brzine obrtanja za reference frekvencije 10/-30/50 Hz.



Slika 17. Snimljeni talasni oblik ulazne struje frekventnog pretvarača za 50 [Hz].



Slika 18. Snimljeni vremenski dijagram izlazne struje pretvarača, ulazna struja motora za 50 Hz.

IV. ZAKLJUČAK

U ovom radu prikazana je praktična realizacija ispitne stanice za upravljanje trofaznim asinhronim motorom primenom frekventnog pretvarača i PLC-a.

Ovaj rad se takođe može upotrebiti u daljoj obuci, radu i usavršavanju naših učenika i za razna dalja istraživanja i dokazivanja iz oblasti upravljanja trofaznim asinhronim motorima i PLC-kontrolerima.

ZAHVALNOST

Na ovom mestu želimo da uputimo zahvalnost našem mentoru Skoko Saši i Milici Karabi, koji su nam pomogli u savladavanju teorijskih znanja i prilikom rada na praktičnoj realizaciji ispitne stanice.

LITERATURA

[1] Z. Pendić, J Nikolić, *Električne mašine sa regulacijom elektromotornog pogona*, Beograd, Zavod za udžbenike i nastavna sredstva, 1996.
[2] S. Skoko, *Elektroenergetika*, Beograd, Zavod za udžbenike, 2017.
[3] https://industrial.omron.eu/en/products/frequency-inverters
[4] http://www.fatek.com/en/

Prototip pomerača digitalnog naponskog nivoa

Dušan Kostadinović, Dušan Gidić, Žarko Jovanović

Sadržaj – U ovom radu predstavljena je jedna vrsta pomerača naponskog nivoa koji se bazira na Šmitovom okidnom kolu, sa dodatnim blokom na izlazu, koji obezbeđuje malu izlaznu otpornost, odnosno mogućnost za priključivanje više potrošača. U izradi prototipa korišćene su diskretne komponente radi jednostavnijeg dizajna i mogućnosti podešavanja praga okidanja u kolu. Nakon simulacija u LT spice-u izvršena je realizacija prototipa na raster ploči

I. UVOD

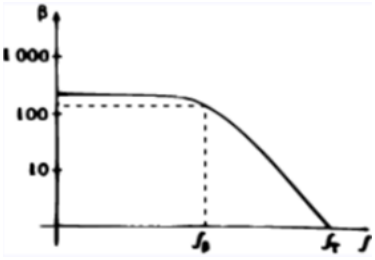
U digitalnim sistemima, postoje dva logička nivoa kojima odgovaraju dva naponska nivoa: visok (HIGH) i nizak (LOW) [1]. Visok nivo je tipična vrednost napona (3.3 V, 5 V,...) koju sistem prepoznaje kao logičku jedinicu. Tradicionalno 5 V je bio podrazumevani logički nivo za mnoge sisteme, međutim kako silicijumski čipovi postaju manji i efikasniji, 3.3 V je sada standard za mnoge sisteme. Problem nastaje kada na pinove sistema koji radi na 3.3 V dovedemo signal čiji je naponski nivo 5 V i obrnuto. U prvom slučaju moze doći do oštećenjauređaja zbog previsokog napona, a u drugom slućaju, napon može biti previše nizak da bi bio prepoznat kao logička jedinica.

Na osnovu navedenih razloga možemo zaključiti da postoji potreba za nekom vrstom adaptera koji ce omogućiti, recimo, da na uređaj čiji je visok naponski nivo 3.3 V priključimo uređaj koji radi na 5 V. U nekim slučajevima potrebno je da na jedan takav adapter priključimo više paralelnih opterećenja, što zahteva malu izlaznu otpornost,odnosno mogućnost da na izlazu dobijemo veću izlaznu struju do 100 mA.

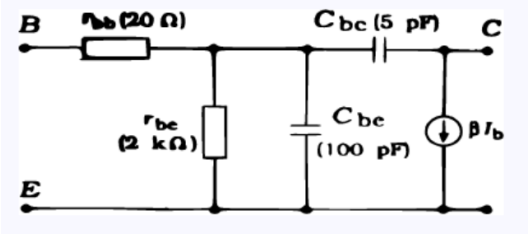
Sve ovo podrazumeva rad na visokim frekvencijama do 10MHz, što dodatno komplikuje projektovanje, pa treba imati to na umu prilikom projektovanja. Poznato je da se osobine tranzistora menjaju sa promenom frekvencije. U praksi je najvažnija promena koeficijenta β koja je prikazana na slici 1.

Na ovoj slici se vidi da da je parametar β konstantan do neke frekvencije, a zatim opada. Na visokim frekvencijama struja baze sve manje teče kroz PN-spoj baza-emitor, a sve više teče kroz difuznu kapacitivnost između baze i emitora. Zbog toga se dobija sve manja kolektorska struja, pa je koeficijent strujnog pojačanja β sve manji. Radi objašnjenja ponašanja tranzistora na visokim frekvencijama, često se daje ekvivalentna šema koja je prikazana na slici 2.

Dušan Kostadinović, Dušan Gidić i Žarko Jovanović su studenti master studija na modulu Telekomunikacije na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija. E-mail: zarko.jovanovic@elfak.rs



Slika 1. Zavisnost parametra β od frekvencije.



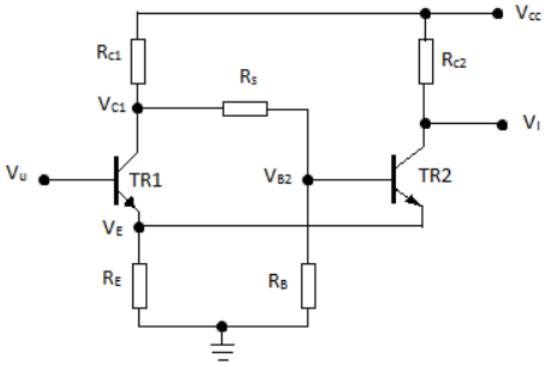
Slika 2. Ekvivalentna šema tranzistora na visokim frekvencijama.

II. IDEJNO REŠENJE

Jedna od glavnih odluka koju smo trebali da donesemo pre početka izrade jeste dali ćemo se odlučiti za korišćenje diskretnih komponenti kao što su bipolarni tranzistori i MOSFET tranzistori, ili ćemo upotrebiti neko integrisano kolo za realizaciju naponskog pomerača logičkog nivoa. Nakon razmatranja više mogućnosti, odlučili smo da koristimo diskretne komponente, odnosno bipolarne tranzistore sa pratećim pasivnim komponentama (kondenzatori i otpornici). Jedna od dobrih ideja bila je da konstruišemo Šmitovo okidno kolo koje na svom izlazu ima dva moguća logička nivoa [2]. Šmitovo okidno kolo je jedna vrsta regenerativnog kola koje se odlikuje velikom brzinom promene stanja, odnosno naponskih nivoa, što odgovara našim potrebama. Pogodnosti realizacije ovakvog kola od diskretnih komponenti jesu jednostavan dizajn, i mogućnost podešavanja praga okidanja u zavisnosti od odabira vrednosti odgovarajućih otpornika. Na slici 3. je data principijelna šema Šmitovog kola, koje je izvedeno kao dvostepeni pojačavač sa emitorskom spregom.

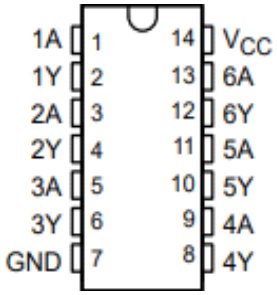
Veza sa izlaza prvog tranzistora ostvarena je preko razdelnika napona RS-RB. Sprega od prvog ka drugom tranzistoru izvedena je preko zajedničkog otpornika Re, koji je postavljen u emitorski dovod oba tranzistora. Za vrlo mali ulazni napon, prvi tranzistor je zakočen, što stvara visok potencijal na bazi drugog tranzistora i čini ga provodnim. Kada ulazni napon premaši napon na bazi

drugog tranzistora, prvi tranzistor će biti u provodnom stanju, a drugi će biti u zakočenju.



Slika 3. Principijalna šema Šmitovog kola.

Šmitovo kolo na svom izlazu ne može da obezbedi veću izlaznu struju zbog velike izlazne otpornosti, pa je za te svrhe, kao dodatni stepen, predviđeno integrisano kolo SN74HCT04N, koje se sastoji od šest nezavisnih invertora. Svaki od invertora može da obezbedi izlaznu struju od 20mA, što paralernim vezivanjem pet invertora na izlaz Šmitovog kola može da obezbedi struju od 100mA. Raspored pinova integrisanog kola SN74HCT04N dat je na slici 4.

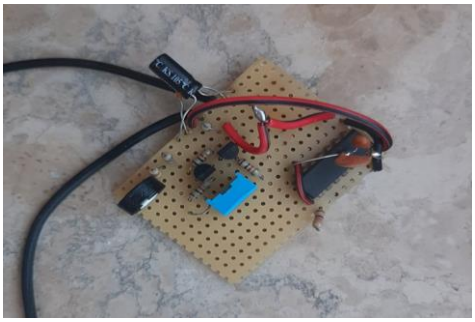


Slika 4. Raspored pinova integrisanog kola SN74HCT04N.

Blok napajanja se sastoji od linearnog stabilizatora napona LM78L05 sa pratećim komponentama. Dioda 1N4007 služi za zaštitu stabilizatora od eventualne inverzne polarizacije. Korišćena su dva elektrolitska kondenzatora kapacitivnosti 10 µF od kojih jedan na izlazu služi za dodatnu filtraciju napona stabilizatora, dok ulazni služi za zaštitu od samooscilovanja stabilizatora.

III. Realizacija prototipa na raster ploči

Za praktičnu realizaciju i testiranje početne ideje, za koju je najpre izvršena simulacija u LT spice-u, korišćena je raster ploča. Simulacije u LT spice-u su pokazale da Šmitovo kolo na većim frekvencijama ne daje očekivane rezultate i da je kolo na većim frekvencijama praktično neupotrebljivo, ali smo hteli da se u to uverimo i praktičnim merenjem [3]. U prvom delu testiranja posmatran je signal na izlazu iz Šmitovog kola, dok je na ulaz dovođen pravougaoni signal iz generatora funkcija, čiji su naponski nivoi 0 i 3V, uz promenu frekvencije od 1KHz do 5MHz. Rezultati merenja su bili slični kao i na simulacijama, tj. signal na izlazu je bio značajno izobličen već nakon 1MHz. U drugom delu testiranja dodali smo integrisano kolo SN74HCT04N, i posmatrali signal sa njegovog izlaza gde smo uočili novu vrstu izobličenja u vidu oscilacija i naglog skoka napona. Nakon testiranja zaključili smo da prva vrsta izobličenja potiče od parazitnih kapacitivnosti tranzistora, odnosno kapacitivnosti spoja baze emitora i baze kolektora, a druga vrsta izobličenja jesu preslušavanja koja se prenose putem vodova za napajanje.



Slika 5. Realizacija na raster ploči.

IV. Decoupling - kondenzatori za odvajanje

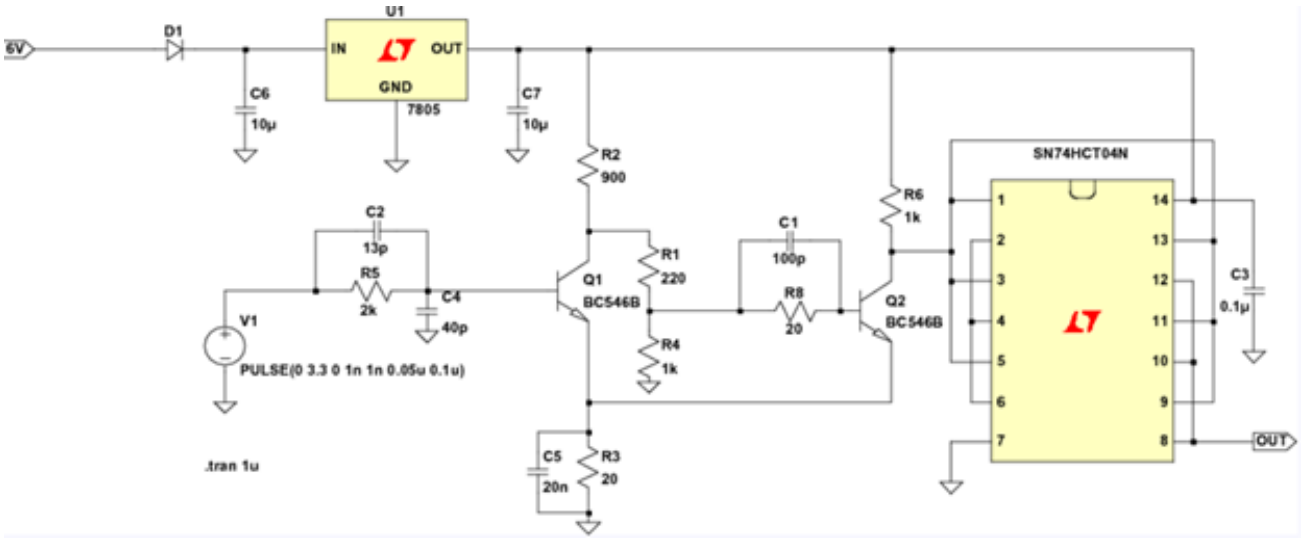
Postoji mnoštvo električnih šumova na tipičnoj električnoj ploči i umesto postojanih 5V na koje računamo, usled različitih šumova, javljaju se odstupanja u vidu skokova i oscilacija ulaznog napona od komponente do komponente.

Kondenzator za razdvajanje (Decoupling) ponaša se kao svojevrsni čuvar energije [4]. Obično se ovakvi kondenzatori nalaze što je moguće bliže integrisanim kolima. Kada je ovakav kondenzator napunjen, njegova uloga je da spreči neželjene promene ulaznog napona napajanja. Kada je "decoupling" kondenzator postavljen, to će obezbediti dve stvari: a) Ako ulazni napon padne, kondenzator će imati dovoljno energije da nadoknadi pad napona i održi napon napajanja stabilnim, b) Ako se ulazni napon poveća, onda će kondenzator apsorbovati višak energije, što čini napon stabilnim.

Na raster ploči dimenzija 50x70 mm pričvršćene su komponente lemljenjem. Prednosti povezivanja komponenata na raster ploči su mehanička čvrstoća i bolji električni spojevi u odnosu na proto-ploču. Izgled povezanih komponenti na raster ploči je predstavljen na slici 5. Na slici 6. je predstavljena tipična upotreba "decoupling" kondenzatora, odnosno postavljanje blizu integrisanog kola.

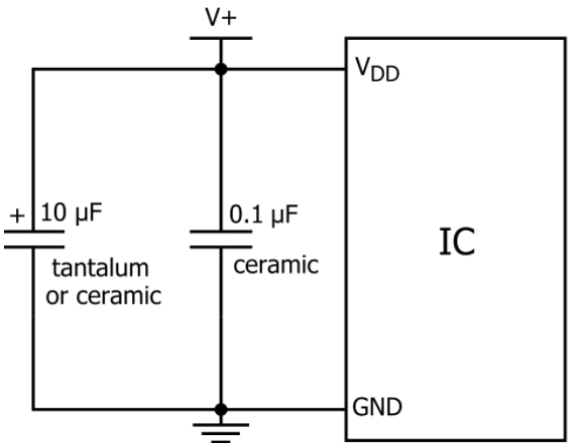
V. Konačna realizacija

Kao i u slučaju testiranja za konačnu realizaciju koristili smo raster ploču većih dimenzija 75x100mm.



Slika 7. Detaljna šema povezivanja

Odlučili smo da za napajanje koristimo najmanji napon koji je bio upotrebljiv i zbog toga smo u konačnoj verziji stavili stabilizator napona koji daje stabilnu vrednost napajanja od 5V. Detaljna šema povezivanja, u skladu sa rasporedom nožica integrisanih kola data je na slici 7.



Slika 6 „Decoupling“.

Na ploči je predviđeno dovoljno mesta za dodavanje nekih komponenata u svrhu kompenzacije, kao što je trimer kondenzator.

Detaljna šema povezivanja uključuje elemente za kompenzaciju, međutim mi ih nismo dodali pri finalnoj realizaciji iz razloga što je za to potrebno duže vremensko testiranje. Na ulaz/izlaz našeg kola dodati su BNC (Bayonet Neill-Concelman) konektori [5], koji se koriste za ostvarivanje veza kod RF signala, analognih i digitalnih video signala. Proizvode se u verzijama od 50 Ω i 75 Ω. U našem slučaju koristili smo BNC konektor od 50 Ω.

VI. Zaključak

U ovom radu prikazana je jedna od mogućih realizacija digitalnog drajvera, kao pomerača naponskog nivoa koji za svoj rad koristi bipolarne tranzistore koji imaju frekventno ograničenje, odnosno dolazi do izobličenja signala pri većim frekvencijama, pa je s tim u vezi potrebno izvršiti frekvencijsku kompenzaciju. Jedno od rešenja bi bilo dodavanje trimer kondenzatora na ulaznom delu tranzistora kako bi preciznim podešavanjem kapacitivnosti izvršili kompenzaciju. Vrednosti trimer kondenzatora su reda pF (tipično 100 pF).

Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Hardverska i softverska realizacija projekta izvedena je pri laboratoriji za Telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

Literatura

[1] VančoLitovski, „Osnovi elektronike“, Elektronski fakultet, 2006.
[2] Wikipedia, „Šmitovo okidno kolo“, dostupno putem interneta na: https://en.wikipedia.org/wiki/Schmitt_trigger
[3] Texas Instruments, „LM340, LM340A and LM7805 Family Wide V IN 1.5-A Fixed Voltage Regulators“, kataloški podaci http://www.ti.com/lit/ds/symlink/lm7805.pdf
[4] Wikipedia, „Decoupling kondenzator“, dostupno putem interneta na: https://en.wikipedia.org/wiki/Decoupling_capacitor
[5] Types of BNC Connectors: https://www.iitk.ac.in/ibc/BNC.pdf

Automatizacija mini elektrana

Nikola Marković, Danijel Stojanović, Igor Trajković

Sadržaj - U ovom radu opisan je jedan sklop za automatskim upravljanjem mini elektranama u skladu sa tehničkom preporukom broj 16 ED Srbije.

I. UVOD

U novije vreme, javila se potreba za primenom novih tehnologija, kako u industriji, tako i u svakodnevnom životu. Savremene tehnologije objedinjuju niz savremenih uređaja kao što su pametni mobilni telefoni, programibilni kontroleri, računari itd... Jedan od takvih sklopova, koji objedinjuje niz pametnih uređaja je i mini elektrana, u kojoj je moguće kontrolisati parametre sistema kako lokalno, tako i daljinski, sa bilo koje tačke na svetu. 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 HMI WEINTEK 8071 IE uređaj. Kodovi za PLC su pisani u WinProladder-u, a za HMI u programu EasyBuilder Pro. Parametri električne mreže se kontrolišu lokalno, daljinski upotrebom HMI uređaja, daljinski pomoću računara upotrebom SCADA aplikacije i daljinski sa neograničene daljine, upotrebom pametnih telefona. Parametri sistema koji se mere su električni napon u sve tri faze, frekvencija, a opciono se može meriti i jačina električne struje. Programski se podešava zaštita koju je propisalo elektrodistributivno preduzeće i to podfrekventna, nadfrekventna, podnaponska i nadnaponska. Elektrana se sa mreže može isključiti lokalno, pritiskom na taster, HMI uređajem, pomoću računara i daljinski, mobilnim telefonom. U svakom trenutku se daljinski mogu pratiti parametri električne mreže i da li je zaštita reagovala ili ne. Opciono, ako elektrana radi u režimu rada van mreže, moguće je isključenje zaštite.

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 elektrane sa mreže, i deset relejnih izlaza na koje su vezani releji koji, neposredno ili putem kontaktora, uključuju i isključuju strujna kola. Elektrana se isključuje sa mreže bez vremenske zadrške, a stanje uključenosti ili isključenosti se može utvrditi na računaru, mobilnom telefonu ili HMI uređaju.

Kod se ubacuje sa računara u kontroler na jedan od više poznatih načina. Kontroler komunicira sa HMI uređajem R232 komunikacionim protokolom. HMI uređaj je umrežen sa računarom LAN mrežom, a računar je povezan na internet WiFi mrežom pomoću koje komunicira sa pametnim telefonom. Na slici 1 je prikazan PLC FATEK FBS 24 MAR.



Slika 1. PLC FATEK.

Za lokalno uključenje i isključenje elektrane sa električne mreže 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. 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 bezkontaktne releje (*solid-state reley*). Naš sklop za automatsko

Nikola Marković i Danijel Stojanović su učenici Tehničke škole "Rade Metalac" iz Leskovca, Dimitrija Tucovića bb, 16000 Leskovac , Srbija

Igor Trajković profesor elektrotehničke grupe predmeta u TŠ "Rade Metalac", Leskovac: itrajkovic7@gmail.com

upravljanje i merenje parametara električne mreže koristi kontaktore zbog visoke cene bezkontaktnih releja.



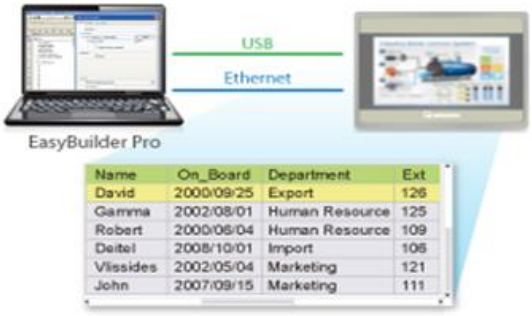
Slika 3. Kontaktor.

Jedan od načina daljinskog upravljanja elektranom je uz pomoć HMI uređaja. U našem slučaju upotrebljen je HMI WEINTEK 8071 IE koji je prikazan na slici 4.



Slika 4. WEINTEK 8071 IE.

HMI WEINTEK 8071 IE je vezan za računar mrežnim kablom, slika 5.



Slika 5. *Ethernet* komunikacija.

Za zaštitu provodnika i kablova od struje preopterećenja i kratkih spojeva upotrebljavaju se osigurači koji se postavljaju na početak svakog strujnog kola slika 6.



Slika 6. Automatski prekidač-osigurač.

U našem konkretnom slučaju upotrebljeni su automatski osigurači S10 nazivne struje 10 A. Osigurači se biraju na osnovu nazivne struje, nazivnog napona mini elektrane i na osnovu njihove prekidne moći.

Za merenje napona i frekvencije koristi se dekodeov uređaj koji komunicira sa kontrolerom MODBUS komunikacionim protokolom i koji je predstavljen na slici 7.



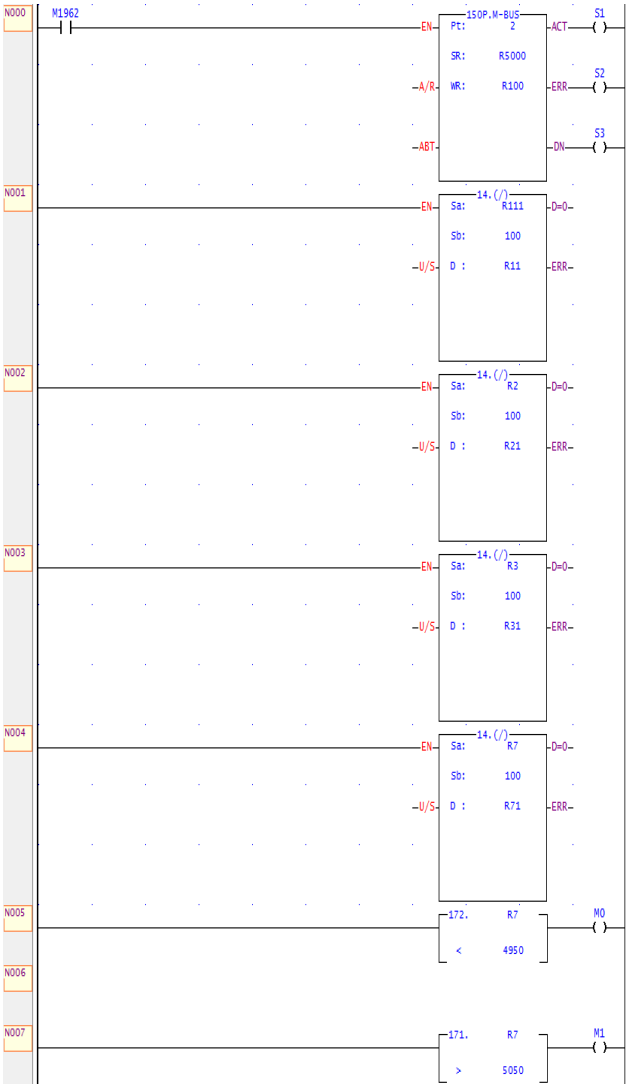
Slika 7. DECODE EMU V3.

Projekat je započet povezivanjem najpre ulaznog tastera na digitalni ulaz kontrolera. Fatekov kontroler ima četrnaest digitalnih ulaza koji su označeni oznakama H0-H13. Zatim se na kontroler dovodi i izvor napona 220V-240V AC koji se obezbeđuje iz sopstvene potrošnje elektrane. Da bi sistem mogao da funkcioniše i pri nestanku

napajanja, potrebno je koristiti UPS uređaj. Napomena: u ovom projektu nije korišćen UPS uređaj.

Drugi korak u izradi projekta je vezivanje kontaktora za isključenje elektrane sa mreže na digitalni izlaz kontrolera. Naš kontroler ima deset digitalnih izlaza koji su označeni oznakama Y0-Y9. Zatim se povezuje DECODE EMU V3 uređaj za provodnike preko osigurača i za PLC sa kojim komunicira RS485 komunikacionim protokolom.

Treći korak je povezivanje računara i kontrolera radi upisivanja koda serijskim RS 232 protokolom. Kod je pisan jezikom lestvičastih dijagrama, u našem slučaju WinProladder-u (slika 8).



Slika 8. Deo koda u *WinProladder*-u.

Na slici 8 je prikazan deo koda u *WinProladder*-u. Kod započinje naredbom M-BUS kojom se uspostavlja komunikacija RS485 sa uređajem za merenje napona i frekvencije DECODE EMU V3. Rezultati merenja smeštaju se u registre kontrolera, a zatim se proveravaju na niz unapred zadatih uslova vezanih za zaštitu elektrane. Kontroler proverava ispunjenost zadatih logičkih uslova, ako neki uslov zaštite nije ispunjen, elektrana se automatski isključuje sa mreže, a na HMI uređaju se može videti koja zaštita je reagovala u datom trenutku.

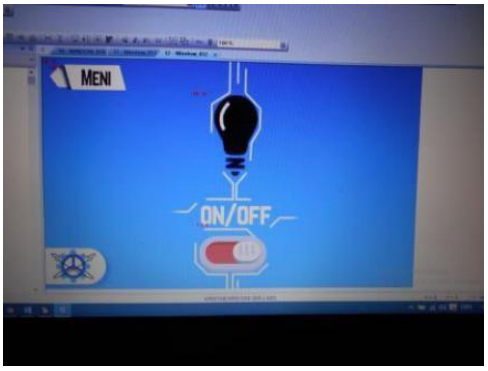
Četvrti korak je prekidanje konekcije između računara i kontrolera i povezivanje kontrolera sa HMI WEINTEK 8071 iE RS 232 komunikacionim protokolom. HMI-u se u osnovnim podešavanjima zada lokalna IP adresa i omogući VNC komunikacija, a zatim se on mrežnim kablom povezuje za računar. HMI uređaj se programira korišćenjem programskog paketa *EasyBuilder Pro*. Na slikama 9, 10, 11, 12 i 13 mogu se videti kodovi u programu *EasyBuilder Pro*.



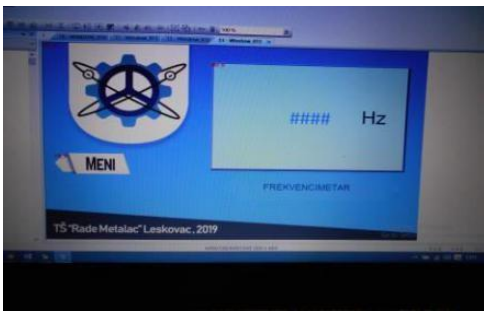
Slika 9. Glavni meni HMI uređaja.



Slika 10. Rezultati merenja faznih napona.



Slika 11. Uključenje i isključenje elektrane.



Slika 12. Merenje frekvencije.



Slika 13. Prikaz reagovanja zaštite.

Nakon pisanja koda, on se upisuje u HMI uređaj, ali se Ethernet komunikacija sa računarom ne prekida. Pomoću programa VNC Viewer moguće je ostvariti komunikaciju između računara i HMI uređaja, tj. zadavati instrukcije kontroleru sa jednog ili drugog uređaja. VNC Viewer-om dobijamo na računaru SCADA aplikaciju za upravljanje datim uređajima ili nekim procesom.

Peti korak je daljinsko upravljanje elektranom, preko mobilnog telefona. Da bi se taj cilj ostvario, neophodno je na računaru i mobilnom telefonu instalirati aplikaciju *Team Viewer*. Zatim se računar mora konektovati na internet mrežu. Ova aplikacija omogućava kontrolu računara mobilnim telefonom sa bilo koje tačke na svetu.

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, programibilnih kontrolera, električnih instalacija i praktičnog rada sa elektro materijalima.

U radu je prikazan sklop za automatizacijom mini elektrana i zaštitu potrošača od: smanjenja napona i frekvencije ispod referentnih vrednosti, povećanje napona i frekvencije, kao i za daljinsko upravljanje elektranom preko interneta. Ovaj rad je praktično realizovan na maketi sa asinhronim motorom i asinhronim generatorom koji predstavljaju motor-generator grupu. Pomoću frekventnog regulatora može se vršiti promena brzine obrtanja asinhronog motora, a samim tim i rotora generatora čime se menja frekvencija i napon indukovane elektromotorne sile. Date promene napona i frekvencije mogu se pratiti na HMI uređaju, kao i reagovanje neke od zaštita. Ovaj rad se može proširiti upotrebom dodatnih modula za merenje i ostalih električnih parametara kao što su: aktivna, reaktivna i prividna snaga, faktor snage, aktivna i reaktivna energija, jačina struje...

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] EasyBuilder Pro-Welcome to Weintek.com (https://www.weintek.com/globalw/download/download.aspx)
[6] FBs-PLC/Manual download/FATEK AUTOMATION CORP. (http://www.fatek.com/en/technical.php?act=manual&catId=11)

Automatizacija staklenika

Jovana Jeličić, Ognjen Galić, Jovan Jeličić

Sadržaj – U ovom radu prikazan je rad iskorišćenja prirodnih resursa za uzgoj biljaka. Cilj nam je maksimalno iskorišćenje prirodnih resursa uz minimalni rad čoveka. Uradili smo sve da za najkraće moguće vreme dobijemo što bolji proizvod i da nam spoljašnji vremenski uslovi ne igraju ulogu.

I. UVOD

Projekat ima kompletno automatizovan proces vegetacije u stakleniku sa ciljem da ljudski faktor obavlja što manje poslova oko održavanja i uzgajanja biljaka. Automatizacijom projekta dobijamo mnogo brže gotov proizvod, a i zdraviji je za čoveka. Ljudi koji su se bavili poljoprivredom tokom svog života imali su veoma veliku svakodnevnu brigu da li će vreme biti dobro, da li će imati kiše, sunca, vetra… Zbog toga su ljudi masovno počeli da se odlučuju za bilo koju vrstu automatizacije svojih zemljišta. Tim inspirisani pokušali smo da napravimo nešto što će sve te brige potisnuti.

II. PRINCIP RADA

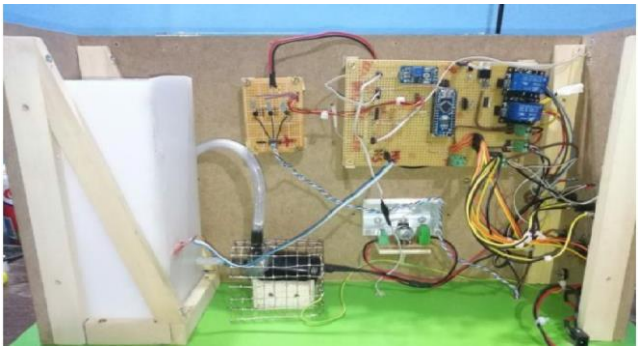
Princip našeg projekta je dug uzgoj i rast biljaka bez obzira na vremenske uslove i godišnja doba, bez upotrebe bilo kakvih hemikalija, štedeći energiju i vreme oko održavanja i uzgajanja biljaka. Koristimo i sistem navodnjavanja kap po kap. Vodu za ovaj sistem u najvećoj količini dobijamo iz sistema koji smo nazvali kišnica. Sistem radi na tome da prikuplja kišu koja je pada i skladišti je u ostavi tj. u rezeorvaru. Najveća količina kišnice nalazi se na krovovima kuća tako da se može na prost način kroz oluk dovesti do rezeorvara. Druga solucija za prikupljanje kišnice su nam podzemne vode. Na dubini od 50 metara provučena je cev sa pumpom koja takodje skladišti vodu u rezeorvar. Sva ta voda po potrebi se ubrizgava u biljke koje se nalaze u nekoj drugoj prostoriji. Za javljanje da li je potrebna voda koristimo senzor za vlažnost zemljišta. Takođe, koristimo solarne panele iz kojih dobijamo dobar procenat električne energije koju koristimo za rad. U zavisnosti kolika je svetlost napolju svetlost u prostoriji gde se gaje biljke se adaptira po njoj što je tamnije svetlost u prostoriji je veća. Taj posao radi senzor osvetljaja. U zavisnosti da li je toplo ili hladno prostorija se zagreva ili hladi kao i posebno lezište gde se sama biljka nalazi.

Na slici 2 je prikazan šematski prikaz projekta. Na njemu su jasno prikazene komponente koje smo koristili.Na levoj strani slike se nalazi rezeorvar za vodu, a na desnoj strani su smestene sve elektronske komponente za automatizaciju projekta.

Jovana Jeličić, Ognjen Galić, Jovan Jeličić, Tehnička skola, Obrenovac, Kralja Petra 12, 15000 Obrenovac, galic.ognjen30@gmail.com, zoranjelicic63@gmail.com, ognjen.galic02@gmail.com



Slika 1. Sakupljanje kišnice.



Slika 2. Praktična realizacija projekta.

III. HIDROPONIKA

Hidroponika je jedan od najmodernijih načina poljoprivredne proizvodnje gde se umesto zemljišta biljke gaje u pesku, a pogodno je i na mestima gde nema zemljišta. Najveća prednost hidroponike je ta što biljke mogu da se gaje u bilo kom kraju sveta. Mogući je uzgoj u bilo kojoj prostoriji čak i u kući. Ovaj način ne daje nikakvo zagadjenje prirodnoj okolini, nema nikakvih prljavština a i pesak se moze koristiti više puta tako da nema velikih ulaganja.Jedino ulaganje je voda koje

mozemo uzeti i iz podzemnih voda ako nemamo u tom trenutku zalihe kišnice.



Slika 3. Gajenje biljaka sistemom hidroponike.

IV. DELOVI ZA IZRADU PROJEKTA

-mikrokontroleri
-solarni paneli
-sistem navodnjavanja (kap po kap) kojicrpi vodu iz sistema za prikupljanje kišnice
-sistem hidroponike objedinjen sa organskom proizvodnjom biljaka

Za prikaz informacija koristimo 16x2 *liquid crystal* displej koji ima nekoliko komandi.on sluzi za prikazivanje menikžja u kome se nalaze sve funkcije.ima sposobnost da kada udjemou meni i jedno vreme ne koristimo ga vrati se na početno stanje, a početno stanje prikazuje ostovne funkcije tj temperaturu vlznost zemljista...ispod njega se nalaze tri dugmetqa pomocu kojih se on navodi.

V. ZAKLJUČAK I PRIMENA

Naš projekat mogu koristiti pojedinci iz ruralnih krajeva koji se bave poljoprivredom i žele da unaprede svoju proizvodnju, restorani, poljoprivredne škole, zadruge, ali takođe i pojedinci koji žive u gradskim sredinama a žele da se bave uzgojom biljaka. Primenom našeg projekta smanjuje se vreme koje ljudi provode uzgajajući biljke, može postati izvor prihoda, takođe velika ušteda novca jer nema hemikalija.... Naš projekat predstavlja celovit projekat u odnosu na druge koji su bazirani na rešavanju jednog problema.

ZAHVALNICA

Autori se zahvaljuju Tehničkoj školi koja je pomogla pri realizaciji ovog projekta kao i profesorima Dušanu Grujiću i Ivanu Milenkoviću.

LITERATURA

[1] Razvoj hardversko/softverske podrške za upravljanje plastenicima, Ivan Stefanović Fakultet tehničkih nauka, Čačak, 2016. Dostupno na: http://www.ftn.kg.ac.rs/download/SIR/SIR%20Ivan%20Stefanovi c%208242015.pdf
[2] Brian W. Evans, Arduino Programming Notebook, 2007, http://playground.arduino.cc/uploads/Main/ arduino_notebook_v1-1.pdf
[3] Trajković, S., Milanovčić, M., Tuneli plastenic i staklenici, http://www.fb.org.rs/en/upload/content/Tuneli_plastenici_i_stakle nici.pdf

Mehanizam za postizanje bezbednosti objekata

Milica Kostić

Sadržaj – Ljudi danas mnogo vremena i energije posvećuju poslovnim obavezama, samim tim veliki deo dana provode van kuće. Kao posledica ovog neminovnog "problema", javlja se sve veći broj provala i krađa kako u stambenim objektima koji se koriste za svakodnevni život, tako i u objektima povremenog boravka, poput vikendica, pa je u ovom radu prezentovana tehnička ideja za sprečavanje i doprinos rešavanju ovog problema.

I. Uvod

Ideja projekta je bila da se postavljanjem ovog mehanizma na ulaznim vratima, prozorima ili sličnim mestima, na kojima je moguće realizovati bilo kakav vid provale za početak usnime/uslikaju izvršioci.

Kasniji rad na ovom projektu može doprineti da se određeni snimci ili slike automatski skladište preko WiFi modula i mreže na određeni stream, gde bi korisnici i vlasnici mogli videti šta se dešava u datom trenutku kada im stigne SMS poruka da su uočene promene u sistemu, tj. da je detektovana bilo kakva promena u okruženju pomoću istog. Takođe, ovaj projekat može pronaći svoju primenu i kod *baby-control* sistema, kao i mnogih drugih sistema koji zahtevaju posmatranje, snimanje/slikanje određenih promena.

II. Princip rada

Za potrebe ovog projekta korišćene su sledeće komponente:

TABELA I
Korišćene komponente za izradu projekta

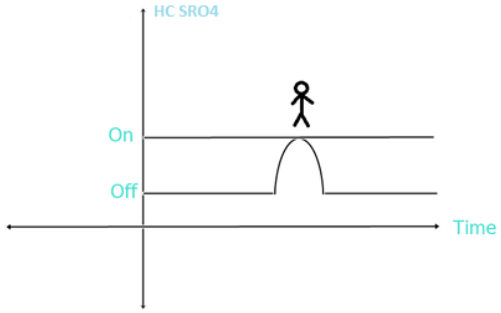
1.	Proto board
2.	2 otpornika od po 10 kΩ
3.	2 otpornika od po 4.8 kΩ
4.	Spojnice
5.	Mikrokontroler – Arduino Uno
6.	Kamera modul OV7670
7.	Senzor pokreta HC SR04
8.	RTC modul DS1307
9.	Arduino IDE okruženje

Princip rada ovog projekta bi bio sledeći: Nakon što dođe do promene na ulazu senzora pokreta, tj. kada senzor pokreta detektuje pokrete na maksimalnom rastojanju od 4 metra od određenog mesta na kome je postavljen, šalje izlazni signal na kamera modul, koji automatski prebacuje kameru u "on" stanje, i fotografije/snimak bivaju napravljene, sve dok se ne neutrališe uzrok okidanja. Ovaj postupak se može realizovati samo u određenim danima, ili u određeno vreme, po potrebi korisnika, a ono se programira pomoću RTC modula (npr. senzor ne šalje obaveštenja o provali i naredbu za slikanje kamera modulu samo ukoliko je u pitanju vikend, iz razlog što su to neradni dani, kada su vlasnici vikendice prisutni u ili pored objekta).

Donekle se može reći da ovaj projekat sadrži tri mini projekta u sebi, tj. svaka od komponenata ovog projekta zasebno može činiti jednu celinu i jedan projekat (kamera modul, senzor pokreta i RTC modul).

A. Senzor pokreta HC SR04

Ovaj senzor predstavlja ultrazvučni senzor rastojanja, koji se koristi za merenje rastojanja od objekta koji miruje ili se kreće do samog senzora. Ovi senzori nude dobru preciznost i jednostavni su za upotrebu. Nemaju smetnji pri radu pri jakom ili slabom osvetljenju. Zvučni talas se prenosi od predajnika i posle odbijanja od objekta vraća eho do prijemnika. Ako je poznata brzina zvuka, rastojanje objekta se može izračunati na bazi vremena koje protekne od trenutka emitovanja do prijema reflektovanog zvuka.

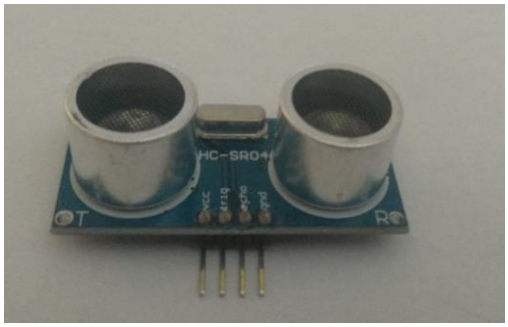


Slika 1. Okidanje senzora pokreta.

Bitne karakteristike ovog modula su da je napajanje potrebno za rad ovog senzora 5 V, izlazni signal koji može dati je 0 ili 5 V [1], ugao u kome je moguće detektovati promenu kod ovog senzora je manji od 15 stepeni, minimalno rastojanje na kom senzor detektuje objekat je 2 cm, maksimalno je 450 cm, i preciznost merenja je 3 mm

Milica Kostić, student doktorskih akademskih studija na modulu Nanotehnologije i mikrosistemi, Katedra za mikorelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milica.kostic@elfak.rs

u optimalnim uslovima. S obzirom da je poznato da je brzina kretanja zvuka 340,26 m/s ili 1236 km/h, a u senzoru je moguće meriti vreme koje je potrebno da se talas, nakon emitovanja odbije od prepreke i vrati do senzora, na taj način je moguće utvrditi rastojanje koje je talas prešao, a samim tim će se znati i rastojanje prepreke od senzora.



Slika 2. Izgled korišćenog senzora pokreta.

Ukratko rečeno, princip rada ovog senzora je sledeći: Modul se aktivira slanjem kontrolnog impulse, dužine najmanje 10μs, zatim automatski šalje osam ultrazvučnih impulsa frekvencije 40 kHz i kada detektuje povratne ultrazvučne impulse, generiše izlazni signal čija je dužina proporcionalna razdaljini [2].

Razdaljina = (Trajanje povratnog impulsa u sekundama) * (brzina zvuka 340 m/s) / 2

B. Kamera modul OV7670

Kamera modul se napaja iz jednog izvora napajanja od 3.3 V. Modul kamere OV7670 je jeftin CMOS modul od 0.3mpx i može dati sliku rezolucije 640 x 480 tj. VGA rezoluciju. Često se može sresti kao deo mobilnih uređaja, PDA, igračaka, nekih proizvoda na baterije (upravo zbog malog napona napajanja koji zahteva), može se koristiti sa Arduinom, APM-om, DSP-om i sl. [3].



Slika 3. Izgled korišćenog kamera modula.

C. Mikrokontroleri

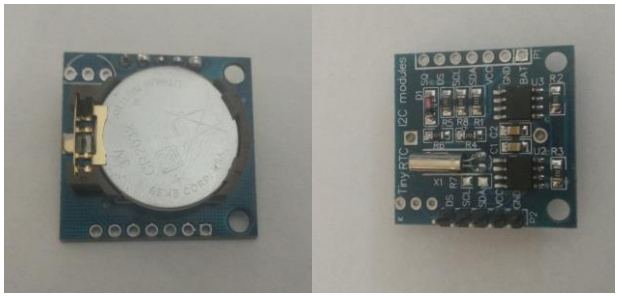
Ako pogledamo uređaje u svetu oko nas, bilo da su to uređaji sa daljinskom kontrolom, roboti, kafe aparati, kućne pekare, saobraćajna signalizacija ili sl. Može se uočiti da u svakom od tih uređaja postoji blok koji kontroliše njegov rad. U današnje vreme upravljanje radom različitih uređaja u čovekovom okruženju najlakše je realizovati na bazi računara. Razvojem poluprovodničkih tehnologija stvoreni su uslovi da danas imamo računare na nivou integrisanih kola u obliku mikroprocesora i mikrokontrolera.

Svakako najpopularniji model mikrokontrolera, što zbog svoje cene, što zbog open source platforme je Arduino. Trenutno postoji vise od 15 različitih modela Arduina, a sve ih je moguće proširiti upotrebom dodatnih modula koji imaju razne funkcije. Ti moduli se nazivaju Shields, i dostupne su različite vrste, poput WiFi modula, LCD ekrana, RFID, NFC… Arduino moduli su dostupni u različitim varijantama, neke od popularnih verzija su Uno, Due, Leonardo, Mega 2560, Nano, Mini i mnogi drugi.

Za realizaciju ovog projekta upotrebljen je konkretno Arduino Uno.

D. RTC (Real Time Clock) DS1307

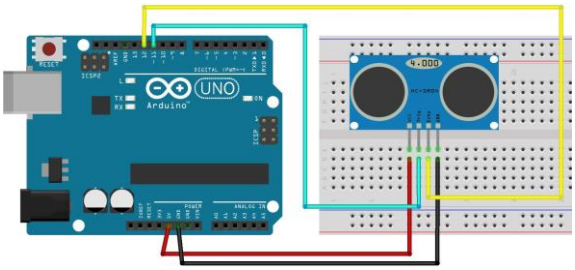
DS1307 modul je sat realnog vremena sa serijskim interfejsom. Ovaj RTC je u potpunosti binarno kodiran decimalni sat, odnosno kalendar, sa malom potrošnjom energije, i sa 56 bajta NV SRAM-a. Pored toga što precizno prikazuje informacije o minutima, satima, sekundama, danu, mesecu i godini, na kraju meseca se automatski prilagođava za mesece sa manje od 31-og dana, uključujući i ispravke vezane za prestupnu godinu. Sat ovog modula radi u 24-časovnom ili 12-časovnom format sa AM/PM indikatorom. DS1307 ima ugrađeni strujni sklop koji detektuje prekide u napajanju i automatski se prebacuje na rezervno napajanje. Operacija merenja vremena se nastavlja sve dok deo iz rezervnog izvora radi (baterija od 3 V) [4].



Slika 4. Izgled korišćenog RTC modula.

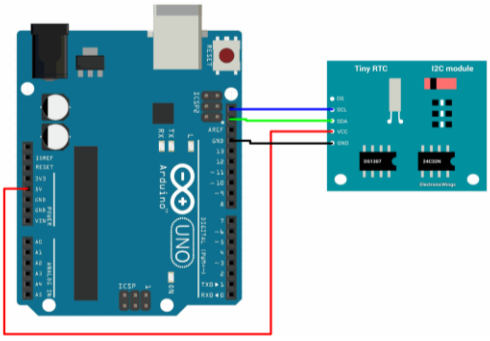
E. Projekat

Povezivanjem senzora pokreta HC SR04 [5] i ispisivanjem odgovarajućeg koda u Arduino okruženju, kao rezultat ovog dela projekta dobijena je na izlazu pri svakom detektovanju promena u okolini senzora, tačna distanca na kojoj se promena dogodila izražena u centimetrima.



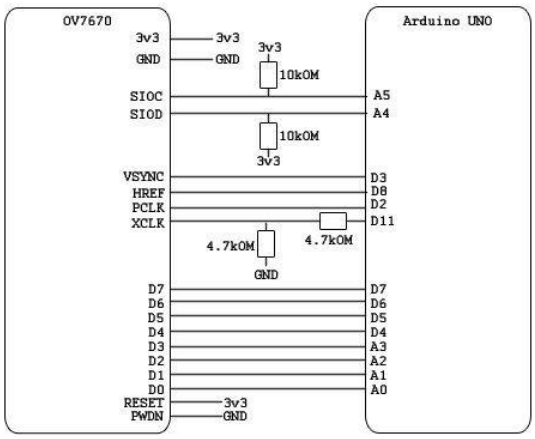
Slika 5. Šema povezivanja Arduino Uno i HC SR04.

Zatim, povezivanjem RTC-a RS1307 na Arduino Uno [6] i ispisivanjem odgovarajućeg koda, na izlazu smo dobili prikaz tačnog vremena (sat, minut i sekund), kao i tačnog datuma (dan, mesec, godina), koji se ispisuju pri svakoj promeni nekog od tih vrednosti (svake sekunde).

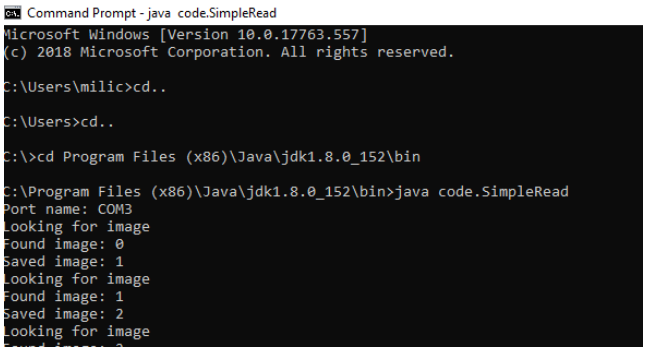


Slika 6. Šema povezivanja Arduina Una i RTC RS1307.

Na kraju, povezivanjem kamera modula OV7670 [7], ispisivanjem opširnog koda, kao i pokretanjem odgovarajućih komandi na računaru, koje služe za očitavanje kamere, pokretanje drajvera, i odgovarajućih biblioteka [8], dobijeno je okidanje kamera modula u određenom vremenskom razmaku koji je definisan u okviru koda kao period okidanja, a fotografije/snimci koji bivaju zabeleženi prosleđuju se i pamte na računaru u okviru foldera na koji je povezan Arduino Uno, i koji je takođe u kodu definisan kao izvorni folder za pamćenje.

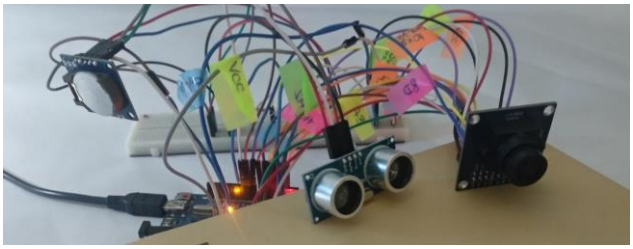


Slika 7. Povezivanje Arduina Una i OV7670 kamera modula.

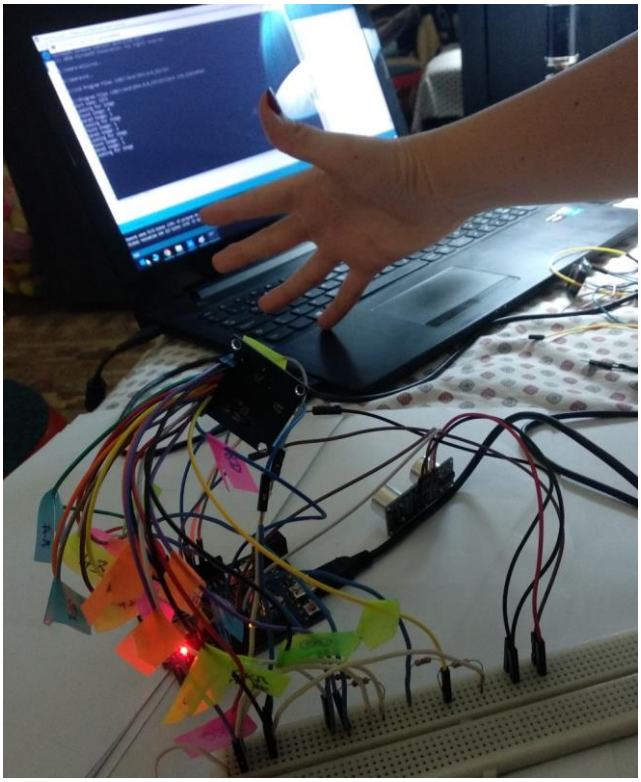


Slika 8. Komande koje pokreću odgovarajuće biblioteke.

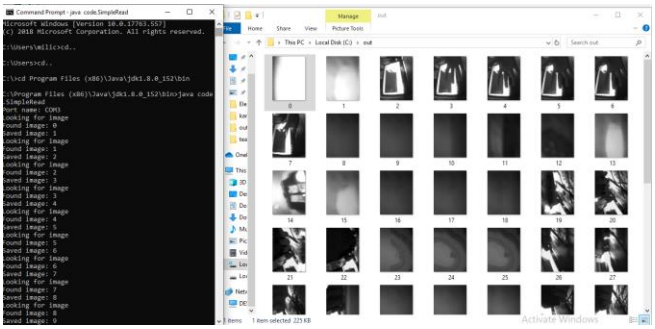
Spajanjem svih ovih funkcionalnih celina u jednu [9], uz minimalnu prepravku koda, koja bi podrazumevala uslove međusobnih uključivanja ovih komponenata dobijeno je sledeće [10][11][12]:



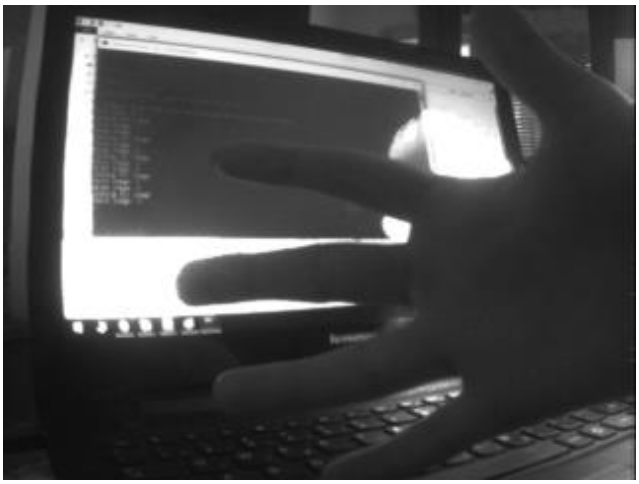
Slika 9. Funkcionalne celine spojene u jedan projekat.



Slika 10. Senzor pokreta detektuje pokret ruke.



Slika 11. Kamera okida i automatski se pamte fotografije na računaru.



Slika 12. Kreirana fotografija.

III. ZAKLJUČAK

Na osnovu svega izloženog u ovom radu, uzimajući u obzir kompaktnost i uspešnost funkcionisanja projekta sa izabranim komponentama, bez nekog značajnijeg kašnjenja projekat ima praktični potencijal za dalje usavršavanje i razvijanje.

LITERATURA

[1] "Sistemi za akviziciju podataka – Ultrazvučni senzori", Elektronski fakultet u Nišu, 2012., prof. Branislav Petrović

[2] http://www.saperel.com/Senzori/HCSR04%20Senzor%20razdaljine

[3] https://www.automatika.rs/projekti/upotreba-ultrazvucnog-senzora-daljine-arduino.html

[4] http://www.otpornik.com/blog/wp-content/uploads/2013/03/HC-SR04.pdf

[5] http://www.otpornik.com/elektronika/komponente/hc-sr04-ultrazvucni-senzor-rastojanja.html

[6] https://create.arduino.cc/projecthub/techmirtz/visual-capturing-with-ov7670-on-arduino-069ebb

[7] https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

[8] https://create.arduino.cc/projecthub/electropeak/interfacing-ds1307-rtc-module-with-arduino-make-a-reminder-08cb61

[9] http://acoptex.com/project/254/basics-project-052a-ov7670-camera-sensor-module-without-fifo-ram-at-acoptexcom/#sthash.NlPRxoIN.Dpf9wYCx.dpbs

[10] https://www.arduino.cc/en/Tutorial/StringComparisonOperators

[11] Datasheet korišćenih komponenata

Podsetnik za redovnu upotrebu medicinske terapije za stare i senilne osobe

Tijana Lazić

Sadržaj - Uređaj je zasnovan na vremenskom alarmu koji u vidu zvučne i svetlosne signalizacije obaveštava osobu da je vremenski trenutak da uzme svoju medicinsku terapiju.

I. UVOD

Zaboravljanje je povremeni ili trajni proces gubitka sećanja (psihofiziološkog sadržaja). Ljudi s godinama počinju da zaboravljaju određene stvari, te se dogodi da se zapitaju i da li su redovno koristili svoju medicinsku terapiju.

Sa napretkom tehnologije, razvijaju se pomagala koja olakšavaju većinu obaveza. Jedno takvo pomagalo predstavljeno je u ovom radu.

Svrha predstavljenog uređaja je da starim i senilnim osobama omogući redovno uzimanje medicinske terapije (tableta ili pilula), a samim tim i izbegavanje neželjenih posledica koje bi nastale neredovnom upotrebom iste.

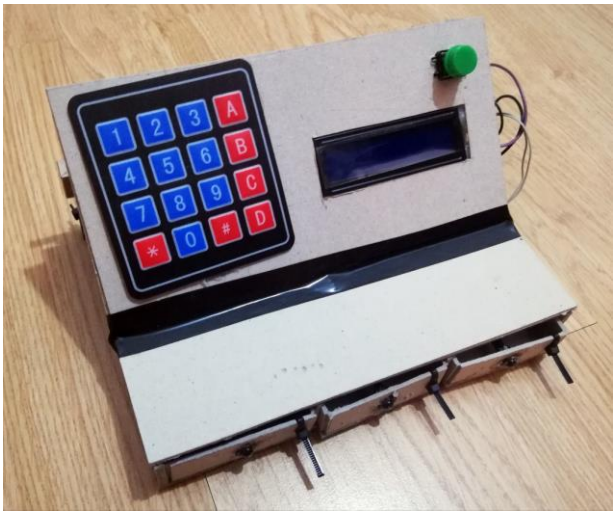
II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Uređaj (slika 1.) se sastoji iz Arduino MEGA 2560, generatora realnog vremena DS1302, LCD displeja 16x2, aktivnog piezo *buzzer*-a, folijarne tastature, tri LE diode i jednog tastera.

Prilikom prvog aktiviranja uređaja**,** korisnik preko folijarne tastature podešava trenutno vreme, kao i vremenske termine kada je potrebno uzeti medicinsku terapiju. Merenje vremena vrši generator realnog vremena DS1302, a Ardiuno MEGA 2560 na osnovu unetih vremenskih termina i trenutnog vremena u određenim vremenskim trenucima obaveštava korisnika kada je potrebno uzeti određenu medicinsku terapiju. Nakon obaveštenja da je vreme za uzimanje medicinske terapije, korisnik je u obavezi da, posle uzimanja iste**,** pritiskom na taster dâ signal mikrokontroleru da je terapija uzeta.

A. Komponente uređaja

- mikrokontroler Arduino MEGA 2560
- LCD displej 16x2
- Aktivni piezo zvučnik
- LE diode
- generator realnog vremena DS1302
- 4x4 folijarna tastatura
- taster



Slika 1. Izgled uređaja.

B. Hardver

Centralni deo uređaja čini Arduino MEGA 2560 mikrokontrolerska ploča zasnovana na mikrokontroleru ATMEGA2560. Merenje vremena obavlja generator realnog vremena DS1302 koji sa mikrokontrolerom komunicira preko I^2C interfejsa. Generator realnog vremena napaja se sa Arduino ploče, a ima i sopstveno baterijsko napajanje koje omogućava merenje vremena u slučaju da ceo uređaj ostane bez napajanja. Preko I^2C interfejsa je povezan i LCD displej koji korisnika obaveštava o trenutnom vremenu (slika 2.), a koristi se i prilikom unosa vremenskih termina za uzimanje medicinske terapije. Folijarna tastatura u vidu matrice tastera 4x4 služi za unos podataka u mikrokontroler**,** kao što su trenutno vreme i termini za uzimanje terapije.

LE diode se fizički nalaze na svakoj od ladica (slika 3.) ponaosob kako bi osobi signalizirale koju ladicu treba da otvori, odnosno koju terapiju treba da uzme (terapiju za ujutro, u toku dana ili za uveče). Aktivni piezo *buzzer* zvučnom signalizacijom obaveštava korisnika da je trenutak za uzimanje medicinske terapije. Piezo *buzzer* i LE diode direktno se preko otpornika pobuđuju sa izlaznih pinova mikrokontrolera.

Tijana Lazić, učenica Tehničke škole u Loznici, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, e-mail: tijanaticalazic02@gmail.com



Slika 2. LCD displej prikazuje trenutno vreme.

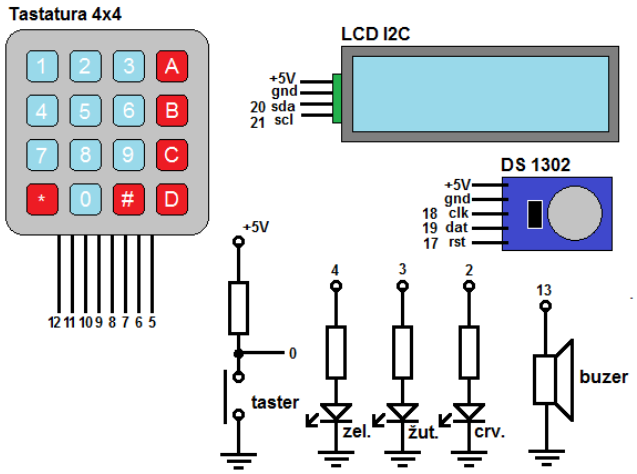
Tačnost merenja vremena sa upotrebljenim generatorom realnog vremena DS1302 za sobne temperaturne uslove iznosi oko 2 sekunde na 24 sata što je dovoljno precizno merenje vremena za ovaj uređaj.

Uređaj se napaja preko USB porta ili posebnog naponskog izvora.



Slika 3. Uključena svetlosna signalizacija na jednoj od ladica.

Električna šema uređaja predstavljena je na slici 4. Označeni brojevi kraj komponenata predstavljaju broj pina na Arduinu MEGA 2560.



Slika 4. Električna šema uređaja.

C. Softver

Za upis programa u mikrokontroler korišćen je Arduino IDE Software. Nakon početne inicijalizacije mikrokontrolera, upisani program u mikrokontroler obavlja stalnu komunikaciju sa generatorom realnog vremena. Kada program utvrdi da je ispunjen jedan od vremenskih trenutaka za uzimanje terapije, program uključuje odgovarajuću LE diodu i aktivira piezo *buzzer*. Tek kada korisnik pritisne taster da je obavešten o tekućoj terapiji, program isključuje LE diodu i piezo *buzzer*. Program ostaje u fazi čekanja do sledećeg vremenskog trenutka za narednu terapiju.

Program omogućava korisniku da podesi trenutno vreme i vremenske termine za uzimanje terapije.

U slučaju nestanka napajanja, program pamti i obaveštava korisnika da, u vremenskom periodu kada je uređaj bio bez napajanja, terapija nije uzeta.

III. Zaključak

Jednostavnavna koncepcija i jasna asocijacija u smislu korišćenja čine ovaj uređaj prihvatljivim za korišćenje kod starih ili senilnih osoba koje nisu u mogućnosti da uvek vode računa o uzimanju svoje medicinske terapije. U odnosu na današnje podsetnike, koji se mogu naći na mobilnim telefonima, računarima i sl., ovaj uređaj predstavlja bolji izbor za navedenu grupu ljudi.

Zahvalnica

Zahvalnost za realizaciju ovog istraživačkog projekta uputila bih Tehničkoj školi Loznica, kao i mojoj porodici na finansijskoj i moralnoj podršci. Posebnu zahvalnost mom mentoru, profesoru, dipl. inž. elek. Tomislavu Mijiću na mnogobrojnim savetima i pomoći u realizaciji projekta.

Literatura

[1] Arduino RTC DS1302 (*https://www.arduino.cc/en/Reference/RTC*)
[2] Arduino LCD I2C (*https://create.arduino.cc/projecthub/Oniichan_is_ded/lcd-i2c-tutorial-664e5a*)
[3] Arduino LED i buzzer (*https://www.instructables.com/id/5-Simple-Button-and-Led-Projects-with-Arduino/*)
[4] Arduino tastatura (*https://www.instructables.com/id/44-Keyboard-For-Arduino-1/*)

Kreiranje univerzalne verifikacione komponente za APB protokol po UVMe metodologiji

Marko Milanović

Sadržaj- U ovom radu biće opisan proces kreiranja jedne univerzalne verifikacione komponente, sam proces verifikacije, kao i kako treba da izgleda struktura verifikacionog okruženja. Pored toga biće opisana metodologija za testiranje univerzalnih verifikacionih komponenata, a takođe će biti reči i o samoj UVM (*Universal Verification Methodology*) metodologiji, kao i *e* jeziku za verifikaciju.

I. UVOD

Svedoci smo ubrzanog razvoja industrije čipova na globalnom nivou. Čipovi postaju sve kompleksniji i pametniji pa samim tim i njihovo testiranje postaje sve veći izazov. Testiranje funkcionalnosti čipa pre njegove izrade je možda i najbitniji korak u procesu projektovanja digitalnog sistema. Kreiranje verifikacionih komponenti, kao i sam proces verifikacije uzima najveći udeo vremena u procesu projektovanja jednog digitalnog sistema.

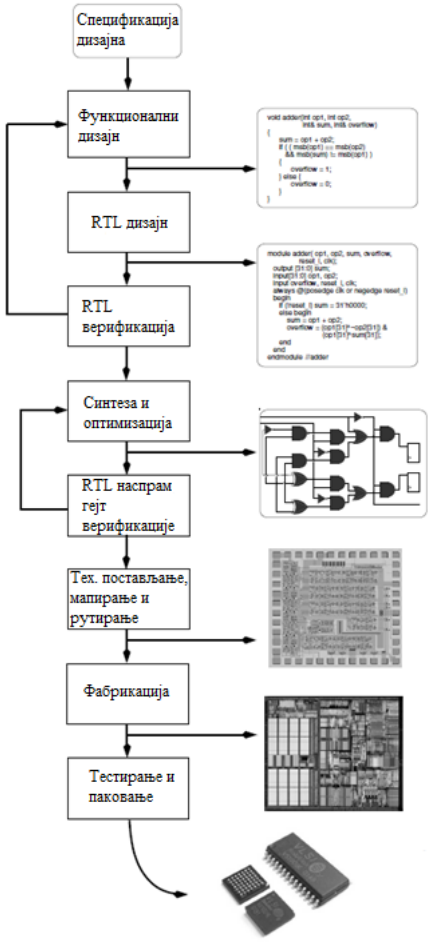
U ovom radu će biti predstavljen i detaljno objašnjen postupak kreiranja verifikacione komponente za APB (*Advanced Peripheral Bus*) protokol. U nastavku su date osnove APB protokola i UVM metodologije, po kojoj je verifikaciona komponenta izrađena. Zatim, je data struktura same verifikacione komponente napisane u *e*-jeziku. Na kraju, dat je detaljan opis verifikacionog okruženja za testiranje verifikacione komponente kao i rezultati testova.

II. TEORIJSKI PREGLED

Proces projektovanja jednog digitalnog sistema je veoma složen proces, koji se sastoji iz više koraka, i koji iziskuje dosta napora, kao i novca da se sprovede u krajnji proizvod. U nastavku biće nešto više reči o procesu dizajna i verifikacije. Svaki od koraka u procesu projektovanja digitalnog sistema može da se radi i u više iteracija, što može dodatno da poveća vreme trajanja samog procesa. Krajnji rezultat je proizvod koji ispunjava sve uslove zadate specifikacijom, u smislu funkcionalnosti, potrošnje energije, cene proizvoda, itd. Na slici 1. može se videti redosled koraka prilikom projektovanja nekog digitalnog sistema. Koraci posle verifikacije dizajna su van okvira ovog rada.

Dizajniranje hardvera predstavlja proces kreiranja modela digitalnog sistema koji je projektovan za izvršavanje funkcionalnosti propisanih specifikacijom.

Marko Milanović je Master inženjer za Elektroniku i mikrosisteme, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E – mail: marko.milanovic@elfak.rs



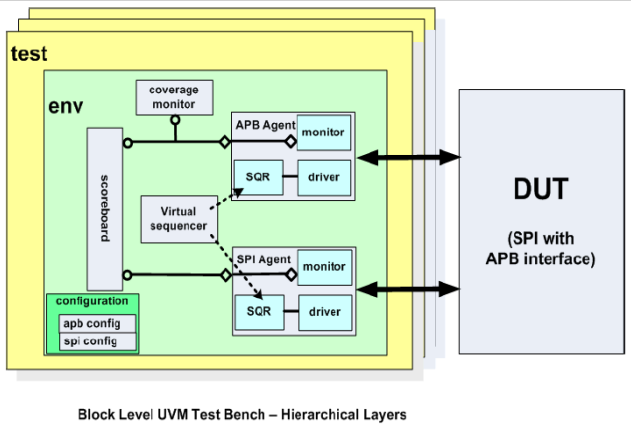
Slika 1. Proces projektovanja digitalnog sistema [1].

Zadatak verifikacionog inženjera jeste da se uveri da uređaj ispunjava svoje zadatke u skladu sa zadatom specifikacijom. Često proces verifikacije počinje u isto vreme kao i proces izrade dizajna. Verifikaciono okruženje predstavlja softverski model ostatka digitalnog sistema u kome će se projektovani digitalni dizajn naći. Za implementaciju verifikacionih okruženja koriste se programski jezici za verifikaciju od kojih su najpoznatiji *e*[i] jezik i *SystemVerilog*. Kako čipovi postaju sve kompleksniji, tako i verifikacija istih iziskuje sve više vremena. Da bi se što više ubrzao proces verifikacije, najveće kompanije iz oblasti dizajna i verifikacije mikročipova, kao što su *Cadence*, *Synopsis*, *Mentor Graphics* i druge, ustanovile su Univerzalnu verifikacionu metodologiju (UVM). UVM standardizuje sam proces verifikacije čime olakšava rad

verifikacionim inženjerima i povećava stepen mogućnosti iskorišćenja istog koda za verifikaciju različitih čipova. Kako bi se postigao ovaj efekat, UVM metodologija uvodi pojam univerzalnih verifikacionih komponenata, u daljem tesktu UVC (*Universal Verification Component*) [2].

A. UVM

UVM se zasniva na velikom broju testova, što usmerenih, što nasumičnih. Svaki test inicijalizuje verifikaciono okruženje (*env* - na slici 2) [3]. Verifikaciono okruženje se uglavnom sastoji od više UVC-ova, komponente Skorbord (engl. *Scoreboard*), Virtualnog sekvencera (engl. *Virtual sequencer*) i konfiguracionog objekta. U skorbord-u je implementiran referentni verifikacioni model dizajna koji se testira kao i provere njegove funkcionalnosti. Skorbord informacije o transakcijama na interfejsima dizajna dobija upravo iz UVC-ova i na osnovu njih kreira očekivano ponašanje dizajna koje kasnije poredi sa stvarnim transakcijama iz dizajna pod testom (*DUT- Design Under Test*). Cilj verifikacionog inženjera jeste da napravi okruženje koje će podržati što više nasumičnih testova, sa različitim konfiguracijama, sa što manje dirigovanih testova. Razlog tome jeste da se pronađe što veći broj grešaka kojih se verifikacioni inženjer kao i dizajner možda neće setiti.



Slika 2. Primer strukture okruženja.

B. UVC

UVC predstavlja konfigurabilno verifikaciono okruženje koje se može koristiti za verifikaciju različitih sistema, a uglavnom se fokusira na određenu arhitekturu ili protokol, u našem slučaju APB protokol. UVC sadrži komponente za generisanje stimulusa, nadgledanje i proveru podataka, sakupljanje pokrivenosti. Osnovne komponente UVC-a su:

- *Drajver* - Drajver predstavlja aktivnu komponentu koja je zadužena, kako to i samo ime kaže, za pobuđivanje (dovođenje odgovarajućih vrednosti na signale interfejsa) interfejsa. Pod pobuđivanjem interfejsa smatra se utiskivanje određenih vrednosti na signale koji se nalaze u okviru zadatog interfejsa. Važno je napomenuti da se nikakvo generisanje podataka ne vrši u drajveru, već se to obavlja u drugim komponentama, a drajver dobija gotovu transakciju koju je neophodno proslediti na interfejs dizajna koji se testira.
- *Monitor* - Monitor predstavlja pasivnu komponentu verifikacionog okruženja. Za monitor kažemo da je pasivna komponenta jer ne vrši pobuđivanje interfejsa kao što je slučaj sa drajverom. Uloga monitora jeste da bude u stanju da prepozna aktivnosti na interfejsu, i da prikuplja informacije od interesa sa interefejsa. Monitor kao komponenta, ne sme da zavisi od drugih komponenata, dok informacije koje monitor skupi mogu da koriste sve druge komponente. Prikupljene informacije monitor najčešće prosleđuje *scoreboard*-u, gde se vrši njihovo poređenje. Monitor kao komponenta može sadržati i čekere (checkers). U slučaju da postoje čekeri u monitoru, to bi trebalo da budu samo protokol čekeri, za interfejs odnosno protokol za koji je taj monitor napravljen.
- *Sekvencer* - predstavlja komponentu koja ima ulogu u kreiranju i randomizaciji transakcija. Ova komponenta zapravo kreira objekat transakcije, randomizuje ga, i kao takvog ga prosledi drajveru. U ovoj komponenti se izvršavaju različite sekvence koje je definisao verifikacioni inženjer ili su ograničeno nasumične.
- *Konfiguracioni objekat* – Konfiguracioni objekat sadrži potrebne informacije o načinu na koji bi trebalo da funkcioniše UVC u specifičnom testu. Neki delovi konfiguracije su statični, tj. ne menjaju se iz testa u test ili u okviru testa već su vezani za sam dizajn, dok su neka polja dinamična, tj. menjaju se iz testa u test ili u okviru jednog testa. Jedna od glavnih konfiguracija samog UVC-a jeste ona koja opisuje da li UVC inicira komunikaciju – master ili odgovara na zahtev za komunikaciju – slejv (engl. Slave).

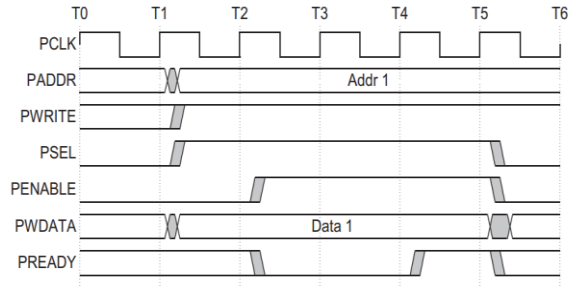
C. APB protokol

APB (*Advanced Peripheral Bus*) [5] protokol je jedan od najkorišćenijih protokola u digitalnim sistemima. Uglavnom se koristi za pristup raznim tipovima memorija i registara kao i izmedju blokova unutar samog sistema koji ne zahtevaju previše veliki protok. Pripada porodici AMBA (Advanced Microcontroller Bus Architecture).

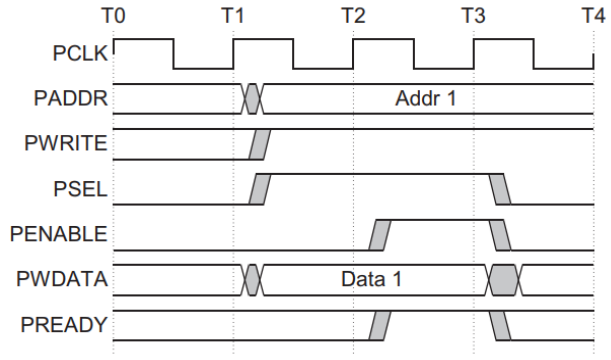
APB protokol sadrži sledeće signale:

- PCLK – Sistemski klok (engl. clock), svi transferi se dešavaju na uzlaznoj ivici ovog signala.
- PRESETn – Reset signal, APB reset signal je aktivan nulom (active-low), ovaj signal se najčešće vezuje direktno na sistemski reset signal.
- PADDR – Adresa. Ovaj signal predstavlja zapravo adresnu magistralu APB protokola. Može biti do 32 bita širine, i drajvovana je od strane komponente koja ima ulogu mastera.
- PPROT – Tip zaštite, protekcije. Ovaj signal označava kakva je vrsta pristupa, pristup normalnog tipa, privilegovanog, ili sigurnog tipa.

- PSEL – Selekt signal. Služi za selektovanje slejv komponente, kojoj je transfer namenjen. Postoji PSEL za svaki slejv.
- PENABLE – Ovaj signal treba da bude na visokom naponskom nivou jedan taktni ciklus nakon PSEL signala.
- PWRITE – Signal koji označava pravac pristupa, odnosno da li je u pitanju upis, ili čitanje podataka. Visok naponski nivo predstavlja upis, dok nizak naponski nivo predstavlja čitanje.
- PWDATA – Podatak koji se upisuje. Ova magistrala se pobuđuje od strane mastera, prilikom write pristupa kada je PWRITE na visokom naponskom nivou. Širina podataka može biti najviše do 32 bita.
- PSTRB – Ovaj signal određuje koji bajt sadrži validne podatke koji će biti upisani u memoriju tokom write transfera. Postoji po jedan PSTRB signal za svaki bajt na magistrali podataka za upis (PWDATA). PSTRB signali moraju biti na niskom naponskom nivou prilikom operacije čitanja.
- PREADY – Ovaj signal slejv koristi kako bi produžio APB transfer.
- PRDATA – Pročitani podatak. Izabrani slejv postavlja podatak na ovu magistralu tokom read pristupa. Širina magistrale može biti do 32 bita.
- PSLVERR – Ovaj signal označava grešku u transferu. Nije neophodno da APB periferije podržavaju postojanje ovog signala. U slučaju da nema podrške za PSLVERR, ovaj signal se vezuje na nizak naponski nivo.



Slika 3. Primer read transfera.



Slika 4. Primer Write transfera.

III. IMPLEMENTACIJA U *E* PROGRAMSKOM JEZIKU

e jezik za verifikaciju (engl. HVL – *Hardware Verification Language*) [6] je osmišljen tako da omogućava kreiranje veoma fleksibilnih verifikacionih okruženja. Kod projektovanja takvih okruženja, posebno se obraća pažnja na mogućnost njihovog ponovnog korišćenja za testiranje više različitih sistema (engl. – *code reusability*). *e* jezik je aspektno-orijentisan programski jezik. Najbitnija karakteristika ove vrste programskih jezika jeste mogućnost dodavanja dodatne funkcionalnosti u već postojeći kod, što se postiže proširivanjem definisanih klasa i implementiranjem novih funkcija na željenim mestima u kodu.

Za rad sa verifikacionim okruženjem kreiranim u *e* jeziku, njegovim kompajliranjem, kao i za lakše nalaženje grešaka koje nastaju tokom projektovanja takvog okruženja, koristi se *Specman* alat. Pošto *Specman* sam po sebi ne sadrži simulator za *VHDL* i *Verilog* programske jezike, mora se koristiti u sprezi sa alatom za simulaciju *HDL*-a.

A. Pregled fajlova UVC za APB protocol

U ovom poglavlju biće prikazan detaljan pregled fajlova neophodnih za kreiranje verifikacione komponente za APB protkol.

- *apb_types.e* - fajl u kom se definišu novi tipovi podataka, specifični za potrebe određenog okruženja (*user defined types*), pored novih tipova podataka često se u ovom fajlu definišu i konstante.

- *apb_signal_map.e* - signal mapa, ili interfejs, predstavlja komponentu koja služi za povezivanje signala iz verifikacionog okruženja sa signalima testbenča. Svi signali iz signal mape definišu se kao portovi. Pored toga neophodno je definisati putanju kroz hijerarhiju dizajna, odnosno *hdl_path()* kako bi *Specman* alat znao pod kojim imenom se odgovarajući port nalazi u okviru hijerarhije testbenča.

- *apb_transaction.e* - transakcija, ili paket, predstavlja enkapsulirani podatak koji sadrži informacije na osnovu kojih *BFM* (*Bus Functional Model*) pobuđuje interfejs.

- *apb_seq_driver.e* - u ovom fajlu kreirana je sekvenca korišćenjem *sequence* makroa, pored toga pozivanjem ovog makroa takođe se vrši i kreiranje drajvera sekvenci. Sekvence predstavljaju strukture koje kreiraju transakcije ili druge sekvence.

-*apb_bfm.e* - *BFM* je deo UVC koji je zadužen za pobuđivanje interfejsa. Prilikom razvoja jedne univerzalne verifikacione komponente neophodno je napraviti *BFM* koji je sposoban da pobudi interfejs i kao master i kao slejv. To znači da u slučaju *DUT*-a koji predstavlja slejv u digitalnom sistemu, *BFM* se mora ponašati kao master, i obratno, pa je stoga na osnovu ovog zahteva neophodno napraviti *BFM* kao komponentu koja će biti u stanju da podrži oba načina rada.

-*apb_monitor.e* - monitor je komponenta u okviru UVC čija osnovna uloga jeste posmatranje signala interfejsa,

kreiranje transakcija sa prikupljenim informacijama, po završetku transfera, i prosleđivanje istih ka *scoreboard*-u (korišćenjem *TLM* (*Transaction Level Modeling*) porta) koji se nalazi u okviru okruženja u kome će agent biti instanciran.

-*apb_checks.e* - u toku razvijanja jedne univerzalne verifikacione komponente posebno se treba obratiti pažnja na definisanje, osmišljavanje i implementaciju čekera jer oni predstavljaju neke od glavnih delova verifikacione komponente. Čekeri treba da budu definisani tako da su u stanju da prepoznaju svaku nepravilnost koja nije u saglasnosti sa protokolom za koji se oni izrađuju.

-*apb_coverage.e*- definisanje, kao i popunjavanje *coverage* je jedna od najbitnijih stavki u procesu verifikacije, pa tako i svaka univerzalna verifikaciona komponenta treba posedovati svoj coverage, kako bi se dokazalo da su testirani svi scenariji, kao i ispravnost rada same komponente.

-*apb_agent_h.e*- ovo je *header* fajl za agent, ovakvi header fajlovi prave se zbog postojanja kružne zavisnosti između agenta i ostalih komponenata agenta kao što su *BFM*, *Monitor*, *Signal_Map* itd. To se dešava zbog postojanja pokazivača na objekat agenta u okviru fajlova *BFM*-a, *Monitor*-a, a sa druge strane u okviru agenta se vrši instanciranje istih, pa je neophodno da oni znaju jedni za druge. To se rešava tako što se prilikom importovanja fajlova u okviru top fajla prvo importuje *header* fajl, pa fajlovi ostalih komponenata, i nakon toga agent. U okviru header fajla nalaze se najosnovnije stvari.

-*apb_agent.e* - kao što je već do sada i rečeno u ovom radu, agent predstavlja komponentu koja enkapsulira sve ostale kompnente UVC (univerzalne verifikacione komponente) u jednu celinu. U zavisnosti od toga kako je konfigurisan, agent može biti aktivan, ili pasivan.

-*apb_agent_cfg.e*- konfiguracioni fajl za univerzalnu verifikacionu komponentu za APB protokol. U ovom fajlu enkapsulirana su sva konfiguraciona polja koja koristi agent. To se radi kako bi se što lakše vršila promena konfiguracije agenta, moguće je izvršiti promenu samo na jednom mestu, dok bi u suprotnom da nema konfiguracionog fajla, promena morala da se vrši u svakom fajlu posebno.

-*apb_seq_lib.e*- biblioteka sekvenci (*sequence library*) predstavlja fajl u kome su smeštene neke od osnovnih sekvenci koje se koriste kako bi se kreirali određeni scenariji. Različiti scenariji, odnosno testovi kreiraju se kombinovanjem više osnovnih sekvenci.

-*apb_top.e*- u okviru ovog fajla vrši se importovanje svih fajlova koje univerzalna verifikaciona komponenta koristi. Prilikom definisanja ovog fajla posebno treba obratiti pažnju na redosled kojim se fajlovi importuju zbog potencijalne kružne zavisnosti koja je objašnjena u delu *apb_agent_h.e*.

IV. METODOLOGIJA ZA ČEKIRANJE UVM UNIVERZALNIH VERIFIKACIONIH KOMPONENATA

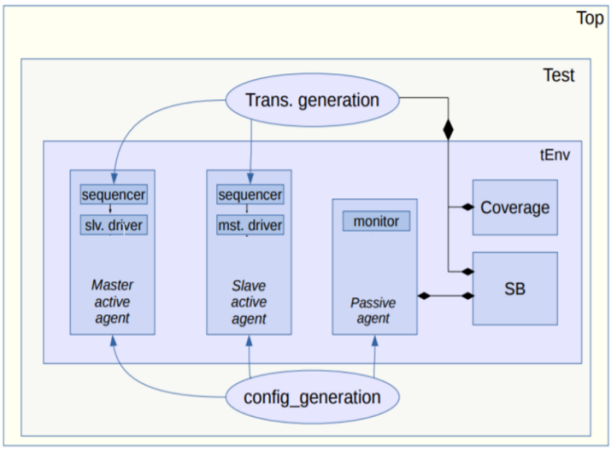
Svaki korisnik univerzalne verifikacione komponente (UVC) tretira istu kao već dokazanu i proverenu komponentu koja omogućava verifikaciju nekog dizajna zajedno sa ostalim komponentama verifikacionog okruženja. U toku planiranja verifikacije, verifikacioni timovi skoro nikada ne planiraju vreme za debagovanje i popravljanje UVC. Često se u praksi dešava da su korisnici primorani da debaguju sam UVC zato što se on ne ponaša kako se očekuje, a to uopšte nije lak posao iz više razloga:

- Neophodno je debagovati tuđi kod
- Fajlovi su enkriptovani
- Korisnik često nije protokol ekspert

Kako bi se izbegle ovakve situacije, oni koji razvijaju UVC moraju obratiti pažnju na čekiranje svih glavnih funkcija UVC. Koje su to stvari koje je neophodno proveriti?

- Postavljanje transakcija na signale interfejsa u skladu sa protokolom
- Nadgledanje signala interfejsa i prijavljivanje transakcija
- Čekiranje protokola
- Reset

Paralelno sa razvojem UVC, potrebno je razvijati i test okuženje (*self-check environment*) u kome će UVC biti testiran. Slika 5 prikazuje strukturu test okruženja preporučenog ovom metodologijom. Ovakva arhitektura testnog okruženja omogućava lako i efikasno proveravanje svih gore navedenih funkcija.



Slika 5. Struktura verifikacionog okruženja za testiranje UVC [7].

Većina protokola specificira komunikaciju između dve komponente, mastera i slejva. Prema toj terminologiji UVC može biti konfigurisan ili kao slejv, ili kao master u zavisnosti koju stranu komunikacije on zamenjuje i oponaša. U okviru testnog okruženja postoje aktivne instance oba agenta, i mastera, i slejva. Glavna namena ovih agenata jeste proveravanje ispravnosti master i slejv drajvera. Čekeri monitora mogu biti isključeni u oba agenta zbog činjenice da su funkcije monitora čekirane u trećem agentu koji je konfigurisan kao pasivan. Ovakav pristup je primenjen kako bi se dokazalo da je monitor u potpunosti nezavisna komponenta, sposobna da izvršava sve svoje funkcije bez ikakvih informacija od bilo koje aktivne komponente, kao ni sekvenci.

- provera pobuđivanja i nadgledanja (*Driving and Monitoring check*)

Pobuđivanje(*driving*) i nadgledanje(*monitoring*) se proverava zajedno puštanjem istih testova. Testovi kreiraju sve moguće vrste validnih transakcija i šalju ih direktno drajveru sekvenci i u scoreboard. U slučaju da drajver ne radi u skladu sa interfejs protokolom, monitor će prijaviti grešku. Čak i da monitor ne detektuje grešku, *scoreboard* će prijaviti grešku prilikom poređenja loše poslate transakcije, sa transakcijom koju je monitor detektovao. U slučaju da drajver radi u skladu sa protokolom, a da monitor ne radi pravilno, u tom slučaju scoreboard će prijaviti neslaganje transakcija.

- Provera protokol čekera

Ova funkcija UVC se čekira namernim insertovanjem protokol grešaka (*error injection*). Veliki izazov kod ovakvog načina rada jeste napraviti okruženje koje će biti sposobno da prepozna grešku, ali isto tako i biti sposobno da izbegne prijavljivanje greške od strane čekera. Cilj je da u slučaju da čekeri ne prepoznaju grešku, *scoreboard* prijavi grešku usled neslaganja transakcija, jer će u tom slučaju scoreboard očekivati transakciju sa greškom koju bi čekeri trebalo da prepoznaju. U slučaju ovakvog načina rada neophodna je mala modifikacija drajvera, monitora, kao i čekera, što se radi njihovim ekstendovanjem.

- Čekiranje reseta (*reset check*)

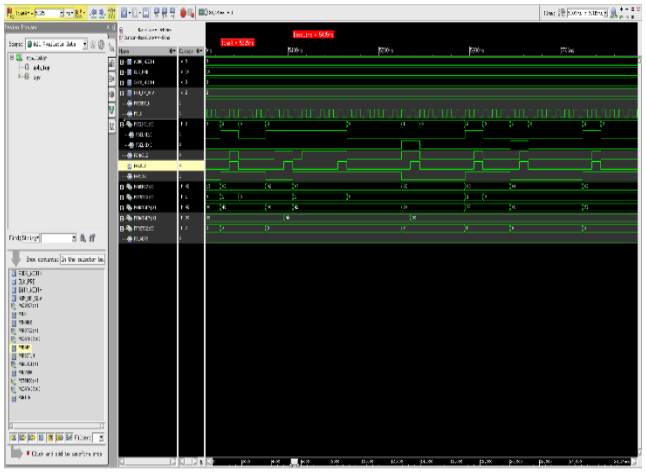
Sve komponente UVC potrebno je da reaguju pravilno na reset. To podrazumeva da one moraju da prekinu sve započete akcije i proračune, ponovo inicijalizuju sva svoja polja i počnu rad od početka nakon deaktiviranja reseta.

Pored toga što se reset primenjuje na početku svakog testa, neophodno je aktivirati reset i u toku transfera. Reset test aktivira reset u nasumičnom trenutku u toku nekog od transfera. Reset u okviru testnog okruženja podrazumeva resetovanje *scoreboard*-a, a to znači ponovnu inicijalizaciju svih njegovih polja, kao i brisanje svih elemenata lista sa očekivanim transakcijama. Ukoliko neka od komponenata nije u stanju da postupi pravilno po aktiviranju reseta, doći će do neslaganja unutar scoreboard-a što će rezultovati greškom.

V. REZULTATI TESTIRANJA

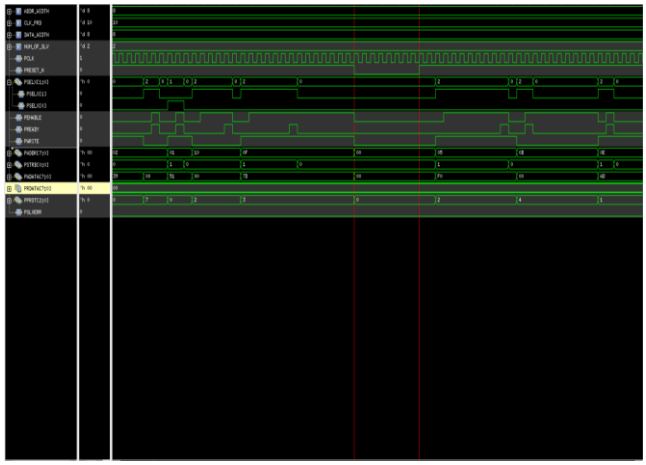
Testovi su delovi verifikacionog okruženja u kojima se kreiraju i startuju sekvence, odnosno prave različiti scenariji na osnovu kojih se testira ispravnost dizajna, u ovom slučaju ispravnost rada univerzalne verifikacione komponente. Test enkapsulira verifikaciono okruženje, što znači da se u okviru njega vrši importovanje top fajla verifikacionog okruženja (import apb_env/src/apb_env_top.e).

-apb_basic_test.e - ovaj test testira operacije čitanja i upisa korišćenjem APB univerzalne verifikacione komponente. Pored toga ovaj test je zapravo najvažniji test jer je test u potpunosti nasumičan (*random*), i može se reći da je to *full random* test. Test vrši nasumično upisivanje i čitanje, sa različitim kašnjenjem između komandi, kao i insertovanjem nasumičnih tipova grešaka.

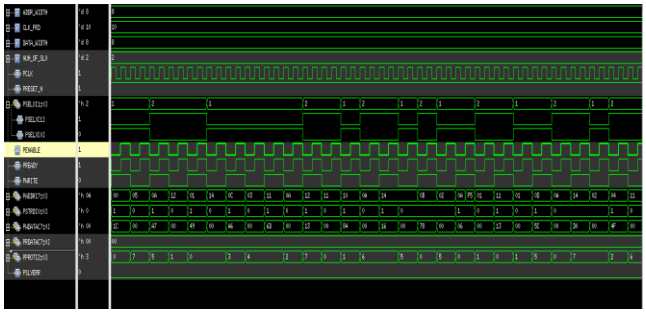


Slika 6. Rezultati apb_basic_test.e u vidu talasnih oblika signala.

-apb_reset_test.e - ovo je tzv. reset test. Ovaj test je od veoma velike važnosti prilikom testiranja bilo kog dizajna, jer se na ovaj način testira oporavljanje sistema nakon što mu je iznenadni reset primenjen. Ovo je jedna od stavki koja se obavezno mora testirati u procesu verifikacije, jer pravilan oporavak od reseta je od esencijalne važnosti za pravilan rad nekog digitalnog sistema.



Slika 7. Rezultati apb_reset_test.e u vidu talasnih oblika signala.



Slika 8. Rezultati apb_b2b_wrrd_test.e u vidu talasnih oblika signala.

-apb_b2b_wrrd_test.e- ovim testom se testira rad uređaja kada mu se šalju uzastopne transakcije, tzv. *back to back* transakcije. Uzastopne transakcije su takve da razmak između njih ne postoji, što znači da po završetku jedne transakcije, odmah počinje druga. Ovo je takođe neophodno testirati, jer se često dešavaju situacije da dizajn ne ume lepo da odgovori na transakcije ovakvog tipa, što ne bi smelo da se dešava, pa se prilikom izvršavanja ovih testova jako lako pronalaze potencijalni bagovi.

VI. Zaključak

U svetu industrije mikročipova vreme je zaista novac. S tim u vezi, svako ubrzanje procesa proizvodnje mikročipova doprinosi značajno razvoju kompletne industrije. Uzevši u obzir da verifikacija digitalnih sistema uzima najviše vremena u tom procesu, razvijanje fleksibilnih, univerzalnih i pouzdanih verifikacionih okruženja predstavlja značajan pomak na tom polju. UVM kao metodologija predstavlja prekretnicu u industriji čiji efekat možemo videti u ubrzanom razvoju digitalnih tehnologija uopšte. Razvijanje fleksibilnih UVC-eva predstavlja prioritet svakog verifikacionog inženjera. Programski jezik *e* zajedno sa UVM metodologijom zajedno čine moćan alat pomoću kojeg i najređi propusti bivaju otkriveni, a samim tim i čipovi bivaju sve pouzdaniji. Medjutim, ne sme se zaboraviti da i same verifikacione komponente moraju biti testirane u cilju izbegavanja lažnih bagova, uzrokovanih pogrešnim radom samih verifikacionih komponenata. Razvijanje metodologije za verifikaciju verifikacionih komponenata još više doprinosi cilju razvijanja pouzdanog hardvera za što kraće vreme.

[i] *Specman/e* – može se naći i pod ovim imenom

Zahvalnica

Autor rada se zahvaljuje profesorki Mioni Andrejević Stošović koja je bila mentor pri izradi ovog rada, kao i kompaniji Veriest koja je omogućila korišćenje svih potrebnih alata, kao i svima koji su doprineli izradi ovog rada na bilo koji način.

Literatura

[1] Srivatsa Vasudevan „Effective Functional Verification“, dostupno na: http://read.pudn.com/downloads100/ebook/409556/Springer.Effective.Functional.Verification.Principles.and.Processes.iNTERNAL.ebook.pdf
[2] Nepoznati Autor „Design and verication of digital systems“, dostupno na: http://web.eecs.umich.edu/~valeria/research/thesis/thesis2.pdf
[3] Cadence, „UVM e Reference“
[4] Arbel H., „Chapter 3.1 – Verification methodology fundamentals“
[5] AMBA, „AMBA APB protocol specification“, dostupno na: https://www.eecs.umich.edu/courses/eecs373/readings/IHI0024C_amba_apb_protocol_spec.pdf
[6] Sasan Iman and Sunita Joshi, „e (verification language)“, dostupno na: https://en.wikipedia.org/wiki/E_(verification_language)
[7] Milan Vlahovic, „Methodology for checking UVM VIPs“

Pametan novčanik

Nikola Timotić, Danilo Perović, Nikola Todorović

Sadržaj- U ovom radu je prikazan princip zaštite kreditnih kartica i novčanika od krađe ili gubljenja.

I. UVOD

U svetu postoji veliki problem krađe i falsifikovanja dokumenata, krađe sredstava iz novčanika kao i sa deviznih i kreditnih računa. Iz dana u dan naši lični i novčani podaci sve više postaju izloženi malverzacijama. Usled toga, kreirali smo projekat, pametni novčanik, u kome smo objedinili različite funkcije kao što su zaštita od krađe i kopiranja kreditnih i debitnih kartica, Anti-Lost zvučni sistem kao i geolokator koji prikazuje trenutnu lokaciju novčanika u slučaju krađe ili gubitka. Dodatna pogodnost je displej koji prikazuje količinu i vrednost kripto i drugih valuta. Cilj tima je da uređaj učini pristupačnim i korisnim svim korisnicima.

II. PRINCIP RADA I KORIŠĆENI HARDVER

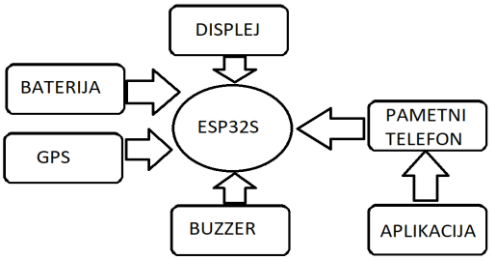
Novčanik je potrebno povezati sa pametnim telefonom preko WiFi ili Bluetooth konekcije. Uz pomoć baze podataka i internet konekcije, na displeju novčanika će se ispisivati trenutno stanje i vrednosti kripto i ostalih valuta. Na displeju će se nalaziti BitCoin adresa koja je predstavljena pomoću QR koda. Prilikom krađe ili zaboravljanja novčanika, kada novčanik napusti predodređeni opseg, počeće da emituje visokofrekventni zvuk, koji služi da upozori vlasnika da se udaljio od novčanika. Takođe u aplikaciji, na pametnom telefonu, možemo videti trenutnu i poslednju lokaciju novčanika kao i par dodatnih opcija.

Za potrebe izrade pametnog novčanika koristili smo sledeće komponente:

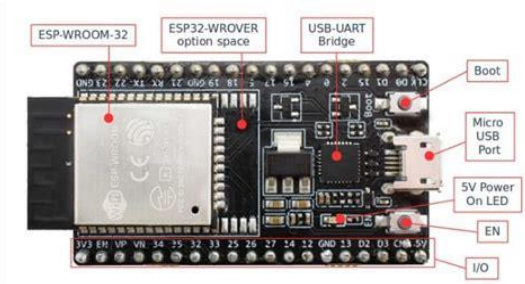
ESP32S mikrokontroler [2] - Opredelili smo se za ESP32S mikrokontroler zbog velike efikasnosti - male potrošnje mikrokontrolera, velike kompaktnosti i malih dimenzija, kao i već integrisanog WiFi i Bluetooth modula u samom kontroleru. ESP32S mikrokontroler poseduje Xtensa LX6 dvojezgrani procesor na radnom taktu 160-240 MHz koji obezbeđuje veliku brzinu rada i malu potrošnju.

OLED display [3] - Odabrali smo OLED displej zbog velike ekonomičnosti baterije i dobre vidljivosti na sunčevoj svetlosti. Na displeju se ispisuju trenutne vrednosti kripto i ostalih valuta, kao i BitCoin (BTC) adresa iscrtana pomoću QR koda na displeju novčanika.

Nikola Timotić, Danilo Perović i Nikola Todorović su učenici četvrte godine "Tehničke Škole Obrenovac", Kralja Petra I 12, 11500 Obranovac, Srbija, E-mail: timoticnikola.01@gmail.com, perke00l@yahoo.com, nikola07todorovic@gmail.com



Slika 1. Princip rada sistema.



Slika 2. "ESP32S" mikrokontroler.



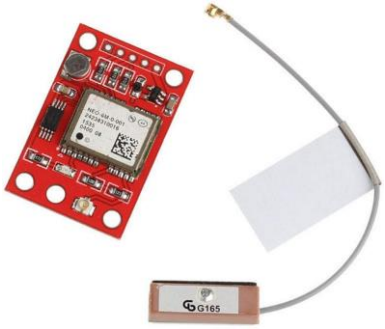
Slika 3. OLED displej.

Buzzer [4] - Odaje zvučni signal kad se uređaj udalji od pametnog telefona.



Slika 4. Bazer.

GPS modul [5] - Utvrđuje i šalje podatke o trenutnoj lokaciji novčanika mikrokontroleru.



Slika 5. GPS modul.

Arduino micro usb [6] - Služi za dopunjavanje kapaciteta baterije u pametnom novčaniku.

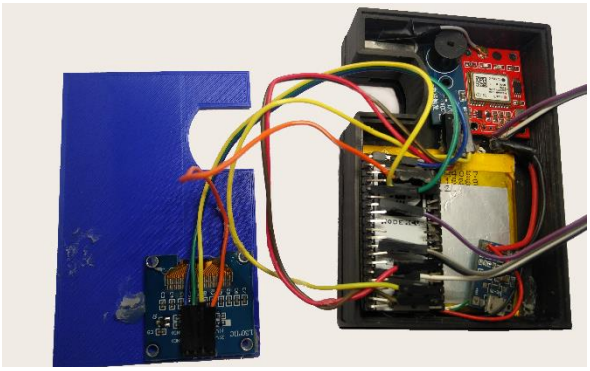


Slika 6. Micro usb.

Baterija [7] - Za napajanje pametnog novčanika opredelili smo se za punjivu Li-Po bateriju kapaciteta – 1800 mAh, napona 3.8 V.Sam uređaj netroši puno energije zbog male potrošnje displeja i čipa.



Slika 7. LiPo baterija.



Slika 8. Realizovan sistem.

III. SOFTVER I DIZAJN SISTEMA

Za izradu sistema korišćen je Visual Studio [7] i Arduino IDE [8] .

Kod za automatsko aktiviranje i prekidanje zvučnog oglašavanja bazera:

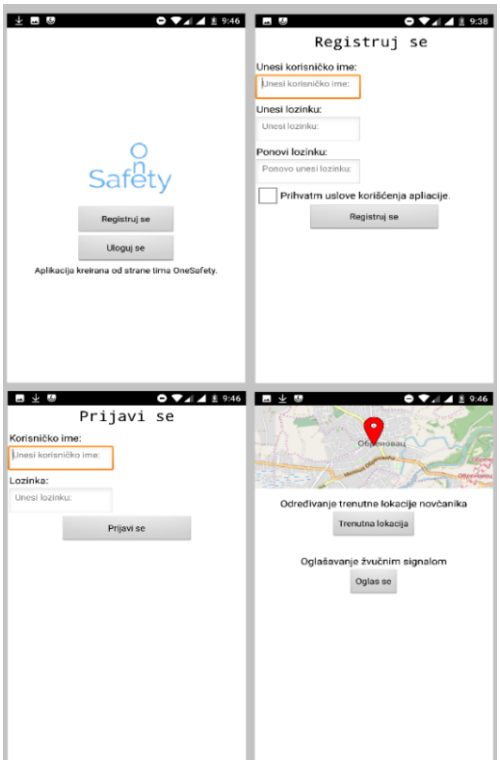
```
while (WiFiMulti.run() == WL_CONNECTED) {
        Serial.begin(115200);
        Serial.print("Connected to network");
        displayText += ".";
        display.clear();
        display.setTextAlignment(TEXT_ALIGN_CENT
ER);
        display.setFont(ArialMT_Plain_10);
        display.drawString(64, 0, displayText);
        delay(1000);
 }
 while (WiFiMulti.run() == WL_CONNECTION_LOST) {
        Serial.print("Connection lost");
        buzzerStart();
 }
 // Display IP if connection successful
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
}
```

Kod za ispisivanje “BitCoin” adrese u obliku “QR” koda:

```
void qrcodeStart() {
        Serial.begin(115200);
        Serial.println("");
        Serial.println("Generating...");
        display.init();
        display.clear();
        display.display();
        // Initialize QRcode display using library
        qrcode.init();
        // create qrcode
        qrcode.create("1F1tAaz5x1HUXrCNLbtMDqcw6
o5GNn4xqX");
}
```

Na slici 7. je prikazan aplikativni program kreiran za pametne telefone. Aplikativni program je rađen u visual studiu u jeziku c#. Pri ulasku u aplikaciju potrebno je napraviti nalog ili se ulogovati sa vec kreiranim nalogom. Svaki novčanik poseduje jedinstveni ID broj koji se vezuje sa nalogom. Preko aplikacije i naloga na telefonu šalje se zahtev serveru za informacije o trenutnoj lokaciji novčanika. Na telefonu bi bila prikazana mapa sa markerom koja predstavlja sam novčanik. Ovaj aplikativni program poseduje mogućnost ukljucivanja visokofrekventnog zvuka

radi lakšeg pronalaženja izgubljenog novčanika. U samoj aplikaciji postojale bi opcije gde bi se mogao podesiti izbor traženih valuta koje bi se prikazivale na novčaniku.



Slika 9. Dizajn aplikativnog programa.

IV. PREDNOSTI SISTEMA

Prednosti našeg uređaja su :

- Kompaktnost uređaja,
- Zaštita uređaja u slučaju gubljenja ili krađe,
- Zaštita kartica od kopiranja ili zloupotrebe i
- Mogućnost korišćenja raznih materijala za proizvodnju (koža, karbon, plastični filamenti…)

V. UNAPREĐIVANJE SISTEMA

Postoji mnogo opcija za nadogradnju sistema, poput upotrebe štampanih ploča za povećanje kompaktnosti i smanjenja dimenzija novčanika, upotrebe kvalitetnijih komponenti i materijala, kao i povećanja dometa i smanjenja zavisnosti novčanika od pametnih telefona.



Slika 10. Model novčanika.

ZAHVALNICA

Zahvaljujemo se Elektrotehničkoj školi Obrenovac, koja nam je obezbedila sredstva i prostor za rad. Takođe se zahvaljujemo našem mentoru profesoru Dušanu Grujiću.

LITERATURA

[1] ESP32S mikrokontroler https://www.mouser.com/images/marketingid/2018/microsites/0/ESP32-DevKitCBD.png
[2] Oled displej http://modtronix.com.au/product/oled-13-i2c-white-1vcc/
[3] Bazer https://www.ardumotive.com/how-to-use-a-passive-buzzer-module-en.html
[4] GPS modul https://www.amazon.com/GYNEO6MV2-NEO-6M-NEO6MV2-Antenna-Arduino/dp/B077Z21L4D
[5] Micro usb punjac https://www.evakw.com/en/adapters-chargers/590-5v-1a-li-ba
[6] Li-Po baterija https://www.amazon.com/1800mAh-battery-Rechargeable-Lithium-Connector/dp/B07BTS6FVM
[7] Visual Studio, link za preuzimanje https://visualstudio.microsoft.com/
[8] Android IDE, link za preuzimanje https://www.arduino.cc/en/main/software-

Bežična prezentacija merenja pomoću Android osciloskopa

Đorđe Kocić

Sadržaj - U ovom radu predstavljen je jedan inovativni metod električnih merenja sa ciljem edukacije ili prezentacije. Merenje se vrši pomoću pametnog Andorid uređaja a rezultatima merenja se može pristupiti bežično putem WiFi mreže, da bi ih pratili svi učesnici edukativnog ili prezentativnog skupa.

I. UVOD

Ubrzan razvoj modernih komunikacionih tehnologija u protekloj dekadi omogućava unapređenje i modernizaciju načina prezentovanja mnogih naučnih disciplina čiji predmeti istraživanja nisu u koraku sa vremenom (energetika, građevina, medicina, književnost...).

Ovde će biti objašnjen način rada jednog sistema za električna merenja koji omogućava da mereni podaci budu praćeni od veće grupe ljudi. Pametni uređaj na Android platformi se koristi za merenje električnog signala napona, preko audio ulaza. Isti uređaj zatim emituje merene podatke preko lokalne WiFi mreže. Merenje onda mogu pratiti svi koji su povezani na tu lokalnu mrežu. Moguće je takođe podatke slati na centralizivani displej (pametni TV) radi bolje vidljivosti. Na isti način se mogu pratiti merenja sa bilo kog drugog senzora koji poseduje pametni uređaj (jačina svetlosti, temperatura, ubrzanje...).

Merenje na ovaj način omogućava učesnicima lakši i kvalitetniji uvid u proces merenja električnih veličina. Idealan način za primenu u praktičnim, demonstrativnim prezentacijama ili laboratorijskim vežbama. Na ovaj način, predavač može publici proslediti razne druge informacije tokom prezentacije (beleške, audio-video materijal, animacije...).

II. MERENJE ANDROID UREĐAJEM

Princip merenja se svodi na merenje analognih veličina pomoću računara posredstvom akvizicione kartice. Analogni periodični električni signal se prikuplja pomoću naponskog transformatora, koji prilagođava naponski nivo merenog signala (kondicioniranje), ograničenjima uređaja kojim se meri [1]. Signal se zatim dovodi na analogni audio ulaz pametnog uređaja. Signal je odatle prosleđen sistemu za obradu zvuka, koji igra ulogu akvizicione kartice. U okviru tog sistema, analogni signal se filtrira i uzorkuje, vrši se analogno digitalna konverzija (A/D) i generiše se digitalna slika analognog signala. Nad digitalnim signalom pohranjenim u baferu audio sistema, sada je moguće vršiti razna ispitivanja. Mogu se meriti osnovni parametri signala, frekvencija, amplituda, oblik signala ili složeniji parametri kao npr. harmonijski spektar, harmonijska distrorzija itd. Podržano je merenje samo periodičnih signala bez jednosmerne komponente, jer ulazno kolo audio sistema filtrira jednosmerne komponente. Prikaz sistema dat je na Slici 1.



Slika 1. Merenje pomoću Android uređaja.

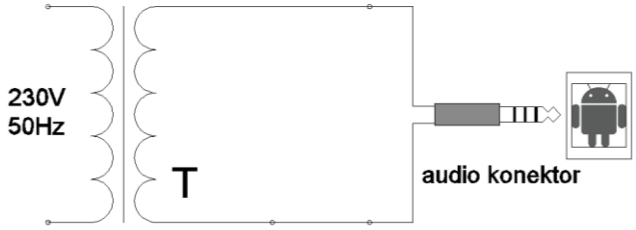
III. OBRADA MERENIH PODATAKA

A. Hardver

U ovom slučaju ulogu akvizicione kartice ima sistem za obradu analognog audio signala Android uređaja. Rezolucija analogno digitalne konverzije (ADC) je 24 bit, a stopa uzorkovanja je 48 kHz što pruža mogućnost za detekciju svih frekvencija ispod 24 kHz.

Meren je signal rezistivnog potrošača, mrežne frekvencije 50 Hz, amplitude 230 V. Mereni signal struje ili napona potrošača se prevodi, pomoću odgovarajućih mernih transformatora, na naponski nivo audio ulaza pametnog uređaja (50 mV do 1 V). Amplituda napona se može korigovati naponskim razdelnikom (ugrađuje se nakon trafoa). Principijelna šema sistema data je na slici 2.

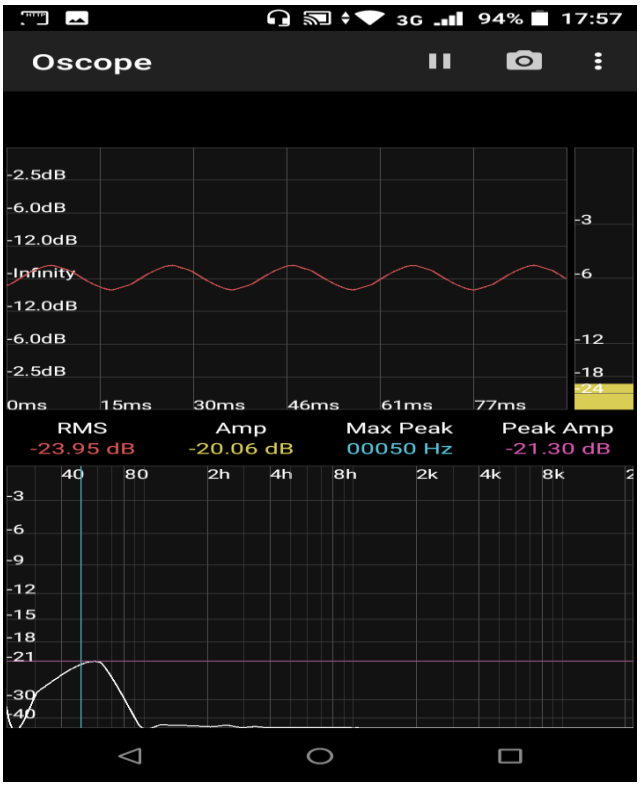
Đorđe Kocić je student treće godine doktorskih akademskih studija, modul Elektroenergetika, Elektronski fakultet, Univerzitet u Nišu Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: seriousdjoka@gmail.com



Slika 2. Principijelna šema hardverskih delova sistema.

B. Softver

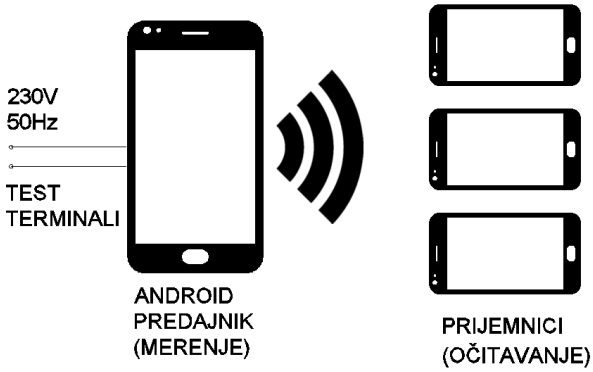
Za potrebe obrade digitalizovanog audio signala razvijen je veliki broj različitih aplikacija na različitim platformama (Windows [2], Android, Linux...). U ovom slučaju koristi se Andorid aplikacija Oscope [3]. Ona pruža funkcionalnosti osciloskopa (talasni oblik), voltmetra (efektivna vrednost), frekventni spektar signala (FFT - brza Furijeova transformacija, vrši analizu harmonijskog sadržaja signala). Detaljni opis programskih algoritama za uzorkovanje i procesiranje signala nije pokriven tematikom ovog rada. U gornjem delu displeja aplikacije vidi se talasni oblik signala sa amplitudom, a u donjem delu dominantna frekvencija kao i kompletan harmonijski spektar signala. Prikaz i funkcionalnosti aplikacije dati su na Slici 3.



Slika 3. Pregled korisničkog interfejsa Oscope aplikacije.

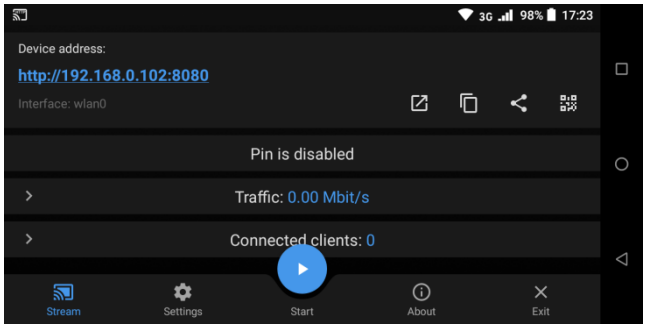
IV. PREZENTACIJA MERENJA

Emisija merenih podataka se vrši preko lokalne WiFi mreže. U ovom slučaju, uređaj kojim se vrše merenja aktivira svoju Hotspot pristupnu tačku (emituje svoju WiFi mrežu), pa nema potrebe za nikakvom dodatnom mrežnom infrastrukturom. Blok šema sistema data je na Slici 4.



Slika 4. Bežična prezentacija merenja.

Na pametnom uređaju kojim se meri, aktivira se aplikacija Oscope koja vrši obradu merenog signala i na displeju prikazuje merene rezultate (efektivna vrednost, frekvencija, harmonijski spektar). Nakon toga, aktivira se druga aplikacija koja služi za emitovanje podataka preko lokalne mreže. U tu svrhu koristi se aplikacija Screen Stream [4], koja omogućava da učesnici prate šta se dešava trenutno na displeju uređaja sa kog je pokrenuta, tako da se trenutni prikaz interfejsa aplikacije Oscope emituje preko lokalne mreže, pomoću aplikacije Screen Stream. Emisija podataka se vrši preko HTTP protokola. Što znači da se učesnici prezentacije povezuju na uređaj kojim se meri preko svog pretraživača (Chrome, Firefox...).



Slika 5. Pregled korisničkog interfejsa Screen Stream aplikacije.

Povezivanje se obavlja tako što učesnik ukucava IP adresu koju zadaje Screen Stream aplikacija (npr. http://192.168.0.102:8080), sa uređaja kojim se meri, u adresni bar svog pretraživača. Primer podešavanja aplikacije Screen Stream i priprema za slanje podataka dat je na Slici 5.

Pregled interfejsa za praćenje merenih podataka dat je na Slici 3. To je izgled aplikacije onako kako se vidi na uređaju koji koristi učesnik prezentacije, ili na centralizovanom displeju.

V. ZAKLJUČAK

U ovom radu je prikazan sistem za prezentaciju električnih merenja uz pomoć Android uređaja, preko WiFi mreže. Uređaj kojim se vrši merenje, emituje podatke preko mreže, tako da je veći broj učesnika može pratiti istovremeno.

Ovo je jedan zanimljiv način kojim se može unaprediti kvalitet predavanja i prezentovanja modernih električnih merenja. Na sličan način se u okviru prezentacije mogu deliti i drugi tipovi materijala (tekst, audio, video...) ili očitavati merenja sa drugih podržanih senzora.

ZAHVALNICA

Ovaj rad je deo realizacije projekata III44006 i delimično je finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije.

LITERATURA

[1] Đ. Kocić, N. Petrović, "Application of Android Based Devices in Analog Electric Signal Measurement", YuInfo 2019, Kopaonik, Serbia.

[2] Soundcard Oscilloscope (Online), C. Zeitniz, https://www.zeitnitz.eu, last accessed August 2019.

[3] Oscope (Online), https://sbaud.io/oscope/, last accessed August 2019.

[4] Screen Stream (Online), Dimitry Krivoruchko, https://github.com/dkrivoruchko/ScreenStream, last accessed August 2019.

Registrovanje prometa putnika u vozilima javnog prevoza

Nikola Tomašević, Marko Rosić, Viktor Savić, Nemanja Vidović, Igor Verkić, Dušan Maksimović, Aleksa Čereković, Miloš Šovljanski, Marko Ršte

Sadržaj - U ovom projektu je prikazano realizovano rešenje za registrovanje prometa putnika u raznim vozilima gradskog, prigradskog i međugradskog prevoza (autobusi, tramvaji, trolejbusi, vozovi, metro), korišćenjem "ARDUINO" mikrokontrolerske hardversko - softverske razvojne platforme.

I. UVOD

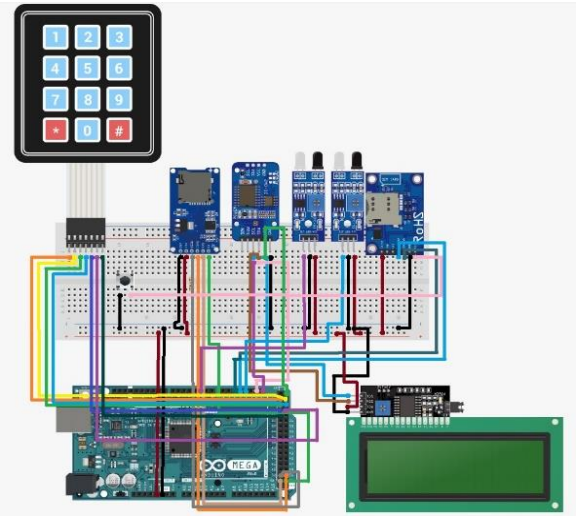
Analizom stanja na terenu u gradskoj i prigradskoj oblasti Novi Sad, ustanovili smo da nema tačne evidencije broja putnika u vozilima javnog saobraćaja, koja je ujedno u potpunosti ili delimično automatizovana. Tako, uzroci nedostataka potpune evidencije prometa putnika u vozilima javnog prevoza su: kupovina karata van vozila na ovlašćenim prodajnim mestima, mesečna ili godišnja pretplata (učenici, studenti, zaposleni, penzioneri itd.), kupljene pojedinačne karte u vozilu, besplatne povlastice bivših zaposlenih i sl. Da li je izražena jedna ili više opisanih pojava, naravno zavisi od vrste javnog prevoza i primenjenog rešenja.

Po informacijama sa kojima raspolažemo u Beogradu i Subotici registrovanje broja putnika je u sklopu kompleksnih *ticketing system* rešenja za prodaju elektronskih i papirnih karata i njihovu verifikaciju. Dakle, ideja je da nezavisno od takvog složenog sistema razvijemo rešenje gde se automatizovano registruje svaki ulaz/izlaz putnika u vozilima javnog prevoza i da ovlašćena lica (vozači, kontrolori i dr.), imaju uvid o prometu putnika u samom vozilu (na vožnji od početne do krajnje stanice, smenski, dnevno i po drugim kriterijumima zavisno od vrste vozila u javnom prevozu).

Nikola Tomašević (tomasevic444@gmail.com) i Marko Rosić (thespykid2002@gmail.com), učenici III razreda (smer: Elektrotehničar informacionih tehnologija); Viktor Savić (viktorsavicprviprojekat.16@gmail.com) učenik II razreda (smer: Elektrotehničar računara); Dušan Maksimović (dumax67@gmail.com), Nemanja Vidović (vidovicnemanja123@gmail.com) i Igor Verkić (igorverkic03@gmail.com), učenici II razreda (smer: Elektrotehničar procesnog upravljanja); Srednja elektrotehnička škola "Mihajlo Pupin" – Futoška 17; N. Sad. Miloš Šovljanski (milos.sovljanski01@gmail.com) student prve godine Fakultet tehničkih nauka u Novom Sadu smer Merenje i regulacija, Aleksa Čereković (acaleksacerekovic@gmail.com) student prve godine Prirodno-matematički fakultet u Novom Sadu smer Fizika, Marko Ršte (mareerste@gmail.com) (bivši učenici ETŠ „Mihajlo Pupin" -Novi Sad).

II. PRINCIP RADA SISTEMA REGISTROVANJA PUTNIKA U VOZILIMA JAVNOG PREVOZA

Projekat je planiran za realizaciju kroz dve faze. Za realizaciju prve faze projekta koja je okončana, kao centralna upravljačka jedinica korišćena je „ARDUINO" – Mega mikrokontrolerska hardversko - softverska platforma sa pratećom opremom i priborom, kao i odgovarajućim aplikativnim programskim kodom (Slika 1.).



Slika 1. Šema veze za sistem registrovanja prometa putnika.

- Centralni sklop opisanog sistema je hardverska platforma „ARDUINO" – Mega na kojoj je integrisan ATmega 2560 mikrokontroler.
- Generisanje datuma i vremena se vrši pomoću RTC *(Real Time Clock)* modula. Komunikacija između mikrokontrolera i RTC modula sprovodi se putem serijskog *interface-a* I^2C *(Inter-Integrated Circuit)*. Pinovi sa modula SDA i SCL preko kojih se sprovodi I^2C komunikacija se vezuju na pinove D20 i D211 na mikrokontroleru.
- Skladištenje podataka se obavlja na SD *(Secure Digital)* modulu sa odgovarajućom *micro* SD karticom potrebnog kapaciteta u zavisnosti od količine podataka i vremena njihovog čuvanja.

Komunikacija između SD modula i mikrokontrolera se zasniva na SPI *(Serial Peripheral Interface)* protokolu koji ima četiri veze između modula i mikrokontrolera (MOS, MISO, SCK, CS). Ove četiri veze odgovaraju pinovima D50-D53.

- LCD *(Liquid Crystal Display)* služi za prikaz podataka ovlašćenom licu. Komunikacija između mikrokontrolera i LCD-a se takođe sprovodi putem I^2C *interface-a*, a povezivanje je identično kao za RTC modul (SDA i SCL se vezuju na pinove D20 i D21 na mikrokontroleru).
- IR *(Infrared)* senzori za detekciju ulaza/izlaza putnika (broj senzora zavisi od broja i veličine vrata u vozilu). Za njihovu konekciju na mikrokontroler su specijalno odabrani pinovi D2 i D18, jer oni predstavljaju *interrupt* pinove na njemu. Ovakvom konfiguracijom postiže se veća preciznost pri brojanju putnika.
- GSM *(Global System for Mobile Communication)* modul za komunikaciju sa nadležnim osobama ili dispečerskim centrom. Komunikacija izmedju GSM modula i mikrokontrolera je serijska i naspram toga se koriste za njihovu vezu pinovi D16 (TX) i D17(RX) na mikrokontroleru. Takođe, važno je istaći da se kod ove serijske veze vrši ukrštanje, to jest TX od GSM modula se veže za RX mikrokontrolera (D17), RX od GSM modula se veže za TX od mikrokontrolera (D16).
- *keypad* tastatura 4X4/4x3 je primenjena radi lakšeg korišćenja sistema i upravljanja njim. Ona sa mikrokontrolerom nema posebno definisan vid komunikacije, već ima osam izvoda koji se vezuju na bilo kojih osam pinova u nizu na mikrokontroleru. Zbog takve konfiguracije izabran su od D23 do D29 na mikrokontroleru.
- Napon napajanja za mikrokontroler, pripadajuću opremu i pribor je pomoću odgovarajućeg adaptera će biti obezbeđen iz lokalnog izvora u vozilu koji se koristi i za druge svrhe (akumulator, priključak za elektrodistributivnu mrežu i dr. što zavisi od vrste vozila). Prva opcija je poseban (rezervni) podsistem za napajanje (punjenje baterijskog sklopa preko stabilizatora), a druga opcija uz prethodnu upotreba solarnih panela.

Automatska evidencija prometa (brojanje) putnika u mobilnoj varijanti po pravilu svakako podrazumeva grešku od 1% do 5 %. Zavisi od vrste upotrebljenih detektora (senzora) i njihove pozicije (rasporeda postavljanja), hardverskih komponenti i aplikativnog softvera koji je primenjen.

IR senzori su prihvaćeni kao odgovarajuće rešenje pošto termalne kamere nije moguće povezati u Arduino razvojno HW/SW okruženje. Ultrazvučni senzori nisu u ovom slučaju primenljivi zbog očekivano velikog broja pokretnih objekata u vrlo kratkom vremenskom intervalu.

Senzori na vratima vozila su povezani na digitalne ulaze mikrokontrolera. Test faza je obuhvatila razne rasporede različitog broja i vrsta senzora (merni, standardni) koji su simulirali realnu konfiguraciju u vozilima autobuskog javnog prevoza.

Ugradnja IR senzora planirana je sa gornje strane vrata vozila. Pošto je primarni problem bio da se na jednim vratima reši registrovanje i ulaza i izlaza, razdvojeni su ulaz i izlaz na posebna vrata, što je ujedno mogućnost realizacije na terenu koja se u praksi koristi npr. u autobuskom prevozu.

U planu je realizacija posebnog digitalnog sklopa, koji rešava problem istovremenog bidirekcionog prolaza putnika na jednim vratima i na više njih (česta pojava u javnom prevozu kod nas i u svetu, posebno u prometnim gužvama u toku dana) i njegovo testiranje na vozilima.

Prvom fazom projekta koja je okončana, omogućena je i komunikacija od vozila sa dispečerskim centrom, gde će biti omogućeno „*off-line*“ praćenje prometa putnika na rutama vozila javnogprevoza na svakoj od linija kontrola sprovođenja usvojenog reda vožnje. Maketa sistema je prikazana na slici 2.



Slika 2. Maketa sistema za registrovanje prometa putnika.

Ovim tehničkim rešenjem detektuje se svaki putnik prilikom ulaska i izlaska iz vozila na stanici što se evidentira putem razvijene aplikacije i instalirane opreme. Podaci se skladište lokalno sa tačnim vremenom dolaska vozila na stanicu i odlaska sa nje (vreme kada je prvi/poslednji putnik ušao/izašao) i time se evidentira broj putnika koji su ušli/izašli, tj. poznat je broj putnika koji se trenutno nalazi u vozilu.

Prikaz potrebnih podataka je omogućen lokalno na LCD-u za ovlašćena lica (vozač, kontrolor, i dr.). Tastatura oomogućava ovlašćenom licu da bira broj mobilnog telefona odgovornog lica ili dispečerskog centra radi slanja podataka bežičnim putem. Takođe, tastatura omogućava lakše upravljanje samim sistemom i izbor potrebnih podataka koji se u sklopu aplikativnog rešenja ispisuju na LCD-u i šalju odgovornom licu ili dispečerskom centru.

Druga faza, koja je planirana za ubuduće, podrazumeva da budu na raspolaganju odgovarajući

hardverski i softverski resursi. Hardverski resursi podrazumevaju servere i ostalu opremu u dispečerskom centru na kojoj će se čuvati i obrađivati podaci. Softverski resursi podrazumevaju instaliranu razvijenu aplikaciju i ostali softver na toj opremi, a komunikacija sa digitalnim sklopovima u vozilu (zaduženi su za brojanje putnika) se vrši putem GSM-a. Ovi resursi će svakako obezbediti centralizovano „*on-line*“ praćenje kretanje vozila na linijama vozila javnog prevoza i prometa putnika u njima.

Takođe će biti obezbeđen automatizovan prijem, obrada i arhiviranje podataka, a biće omogućeno brže i lakše izveštavanje po zadatim različitim kriterijumima, kratkoročno/dugoročno menjanje i planiranje reda vožnje, kao i nabavke novih vozila za proširenje voznog parka.

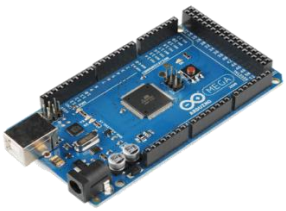
Testirana je na maketi, a po projektu je planirana, mogućnost korišćenja solarne energije za napajanje sistema za registrovanje prometa putnika. U okviru rešenja koje će biti implementirano planirana je primena specijalnih *flax* panela (namenjena za vozila i plovila). Vodiće se računa o što efikasnijem punjenju baterija jer je smer kretanja vozila često promenljiv zbog linija (trasa) kojima se vozilo kreće (koristiće se i vreme kada je vozilo u parking prostoru, tj. van upotrebe za prevoz putnika).

Primenjeni sistem u prvoj fazi koja je na maketi realizovana, omogućava:

- proveru broja putnika u vozilu (radi bezbednosti i komfora, određen je broj sedećih i ako je predviđeno stojećih mesta, u zavisnosti od tipa javnog prevoza)
- kontrolu vremena polaska i dolaska vozila na svakom od stajališta, a po prethodno utvrđenom redu vožnje
- (gotovo) tačnu evidenciju ulaska i izlaska putnika na svakoj stanici i njihov ukupan broj u vozilu na celoj trasi (liniji) u toku jedne vožnje, dnevno, nedeljno i u drugim vremenskim intervalima sve u zavisnosti od potreba. Moguće je preuzimanje podataka od strane dispečerskog centra, a za pregledni prikaz ove evidencije potrebna je dodatna obrada i postupak nije automatizovan.
- mogućnost planiranja novog i izmena postojećeg reda vožnje na osnovu prikupljenih podataka o prometu putnika (potrebna dodatna obrada, mada nije automaizovano).

A. Opis komponenti korišćenih u radu

1. ”Arduino“ MEGA mikrokontroler (na slici 3.) je upotrebljen zbog većeg broja digitalnih i analognih ulaza/izlaza u odnosu na UNO verziju.



Slika 3.”Arduino” – MEGA mikrokontroler.

2. RTC (na slici 4.) služi za generisanje datuma i vremena koji su neophodni u funkcionisanju celokupnog sistema, sa baterijom koja omogućava kontinuitet rada nakon inicijalizacije modula.



Slika 4. RTC modul.

3. SD modul (na slici 5.) služi za smeštaj podataka u domenu registrovanja ulaza/izlaza putnika, vremena dolaska vozila na stanicu i odlaska sa nje. Podaci se preuzimaju u sklopu aplikacije i šalju na predefinisane brojeve mobilnih telefona ili je moguće *off-line* preuzeti podatke i skladištiti na neki drugi medijum



Slika 5. SD modul.

4. GSM modul (na slici 6.) sa integrisanim kolom 800A/A6 (testirane su obe vrste) služi za komunikaciju sa dispečerskim centrom za prenos podataka po utvrđenim kriterijumima.



Slika 6.GSM modul.

5. LCD (na slici 7.) služi za vizuelnu kontrolu sistema, na displeju se ispisuju registrovani ulazi /izlazi putnika (za svaku pojedinačnu stanicu, vožnju, od prve do poslednje stanice, smenu vozača), poruke o datumu i vremenu.

Korišćena je varijanta LCD-a 4x20 karaktera i sa I^2C konekcijom prema mikrokontroleru, pošto koristi samo četiri kratkospojnika i time oslobađa pinove za vezu drugih komponenti. U redovnoj varijanti ima mnogo više veza - time se povećava zauzetost pinova, a ujedno je povećana mogućnost pojave problema sa kontaktima.



Slika 7. LCD.

6. IR - aktivni senzori (na slici 8.) služe za registrovanje prolaska putnika. Oni su razmešteni na vratima vozila.



Slika 8. IR sensor.

7. 4x3/4x4 keypad tastatura (na slikci 9.) kao ulazna jedinica za Arduino platformu. Verzija 4x3 služi za unos broja mobilnog telefona za komunikaciju (prenos podataka) sa dispečerskim centrom. U slučaju verzije 4x4 pored toga služi i za izbor različitih ispisa poruka na LCD-u pomoću tastera za predefinisane funkcije.



Slika 9. 4x3/4x4 keypad tastatura.

III. ZAKLJUČAK

Prvom fazom projekta predviđena je potpuna funkcionalnost čitavog sistema, jer je u tom slučaju moguće prosleđivanje podatka, tj. obaveštenja koja se odaberu, bilo da se odnose na vanredne događaje ili redovna obaveštenja o prometu putnika iako sve nije u potpunosti automatizovano.

Drugom fazom projekta planirana je dodatna funkcionalnost sistema koja se ogleda u pregledu situacije sa prometom putnika u realnom vremenu i na jednom mestu. Tada će biti omogućeno brže i lakše izveštavanje, planiranje i izmene reda vožnje, pošto će prijem, obrada i arhiviranje podataka biti automatzovani.

Opcioni sklopovi koji su navedeni daju dodatne funkcionalnosti i nezavisnost predočenog sistema u domenu napajanja električnom energijom. Prva opcija koja obuhvata punjenje baterijskog sklopa tokom vožnje daje nezavisnost od lokalnog/javnog elektrodistributivnog sistema. Druga opcija uz korišćenje prethodne, obuhvata korišćenje solarnih panela za punjenje baterijskog sklopa i daje naglasak na korišćenju obnovljivih izvora energije.

Tehničko rešenje koje je predstavljeno moguće je primeniti za registraciju prometa osoba u stacionarnim objektima (sportski i trgovinski centri, prosvetne ustanove (vrtići, škole i fakulteti), zdravstvene ustanove (domovi zdravlja, poliklinke, bolnice), javne institucije itd).

ZAHVALNICA

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

Takođe, autori se zahvaljuju Branislavu Ušanu, rukovodiocu sekcije “PiLab” u ETŠ „Mihajlo Pupin“- Novi Sad, na pruženoj tehničkoj podršci tokom realizacije ovog projekta.

LITERATURA

[1] „ARDUINO“ UNO - 45 projekata za početnike i stručnjake“, Bert van Dam, InfoElektronika, 2016
[2] https://store.arduino.cc/arduino-mega-2560-rev3
[3] https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/
[4] https://howtomechatronics.com/tutorials/arduino/arduino-ds3231-real-time-clock-tutorial/
[5] https://howtomechatronics.com/tutorials/arduino/arduino-sd-card-data-logging-excel-tutorial/
[6] http://www.ed.rs/ed/tekstovi/principi/opste.htm
[7] https://elementztechblog.wordpress.com/2016/06/28/interfacing-sim800-gsm-modem-with-arduino/
[8] https://gist.github.com/buildcircuit/2758308[9]http://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/

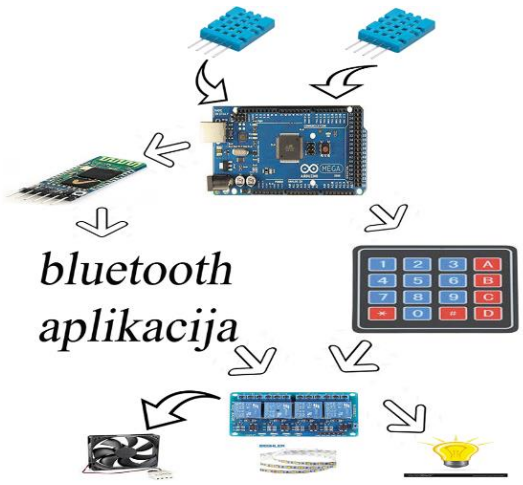
Teraino

Nikola Ristić, Lazar Čebuković, Petar Ljubinković

Sadržaj– Životinje koje se čuvaju u terarijumu zahtevaju specijalne uslove za život. Zbog toga je neophodno meriti parametre unutar samog terarijuma i na osnovu njih kontrolisati uslove u staništu. Ovaj rad ima za cilj da predstavi sistem za kontrolisanje životnih uslova u terarijumima uz pomoć Arduino Mega platforme.

I. UVOD

„Teraino“ je sklop električnih elemenata koji je projektovan i konstruisan sa ciljem da životinjama pruža savršeno okruženje za njihov život. „Teraino“ ima ulogu da meri temperaturu i vlažnost vazduha, ispisuje podatke na LCD ekranu i blutut aplikaciji i da automatski kontroliše vlažnost vazduha, temperaturu, dnevno-noćni ciklus i protok vazduha kroz terarijum. Kao osnova projekta upotrebljena je Arduino Mega [1] platforma. Kodovi su pisani u Arduino Software (IDE) [2] okruženju i MIT App Inventoru [3]. Osnovna šema sistema prikazana je na slici 1.



Slika 1. Osnovna šema sistema.

Krajnji rezultat rada sistema prikazan je na slici 2.

N. Ristić i L. Čebuković su učenici četvrtog razreda, smera: Elektrotehničar informacionih tehnologija – ogled, Tehnička škola Zaječar, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: risticza@gmail.com, cebukovicl@gmail.com.

P. Ljubinković je učenik trečeg razreda, smera: Elektrotehničar informacionih tehnologija – ogled Tehnička škola Zaječar, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: petarljubinkovic7@gmail.com.



Slika 2. Izgled „teraina”.

II. PRINCIP RADA

Sistem radi na principu prikupljanja informacija o temperature i vlažnosti vazduha sa dva DHT11 senzora. Arduino Mega obradjuje te podatke i rezultate ispisuje na lcd ekranu i na telefonu pomoću blutut aplikacije. Korisnik moze manuelno palitii, gasiti led traku, pumpu za vodu, grejalicu, ventilator ili uključiti automatski mod preko aplikacije ili tastature. Uključivanjem automatskog moda korisnik sam definiše željene parameter preko mobilne aplikacije.

A. Predstavljanje komponenata za izradu

U radu su korišćeni: Arduino Mega, HC-05 blutut modul, razvojna ploča sa četvorokanalnim relejom, digitalni senzor vlažnostii temperature (DHT11), LCD ekran dimenzija 16x2, Arduino tastatura dimenzija 4x4, ventilator dimenzija 12x12, led traka, grejalica i pumpa za vodu.

Arduino Mega 2560 (slika 3) je mikrokontrolerska ploča bazirana na ATmega2560 ima 54 digitalna ulaz/izlaz pina (od kojih se 14 mogu koristiti kao PWM izlazi), 16 analognih ulaza, 4 UART-a (hardverska seriska porta), 16 MHz kristal oscillator, poseduje 256 kB memorije i napaja se na 7-12 V.



Slika 3. Arduino Mega 2560.

Relej je komponenta koja se koristi kao prekidač u strujnom kolu, pomoću elektromagneta koji otvara ili zatvara strujne kontakte. U izradi projekta korišćen je četvorokanalni relej modula na slici 4.



Slika 4. Ploča sa četvorokanalnim modulom.

Digitalni senzor vlažnosti temperature DHT11 služi za dobijanje podataka o temperaturi koristi termistor (temperaturni otpornik), dok je za relativnu vlažnost vazduha zadužen kapacitivni senzor, kome se rad zasniva na merenju električnog kapaciteta, koji se menja u zavisnosti od delovanja vlažnosti vazduha na hemijski supstrat. Senzor temperature i vlažnosti vazduha prikazan je na slici 5.



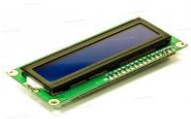
Slika 5. Digitalni senzor vlažnosti i temperature (DHT11).

HC-05 modul (slika 6) je jednostavan za korišćenje blutut SPP (Serial Port Protocol), dizajniran je za transparent serijsku vezu bežičnog povezivanja. Blutut modul HC-05 se može koristiti u konfiguraciji prijemnika ili predajnika, što ga čini odličnim rešenjem za bežičnu komunikaciju.



Slika 6. HC-05 Bluetooth modul.

LCD ekran (slika 7) predstavlja dvoredni ekran sa 16 karaktera i idealan je za povezivanje sa Arduino pločama. Modul je zasnovan na HD44780 kontroleru koji predstavlja industrijski standard. Napaja se na 5V i ima pozadinsko osvetljenje plave boje.



Slika 7. LCD ekran.

Membranska matrična tastatura (slika 8), matrica 4x4, pogodna je za Arduino projekte.



Slika 8. Membranska tastatura.

Real time clock za Arduino je modul koji se zasniva na čipu DS1307 visoke preciznosti. I2C interfejs omogućava čitanje vremena. Pogodan je za sistem zato što pamti vreme i kada sistem ostane bez napajanja. Real time clock prikazan je naslici 9.

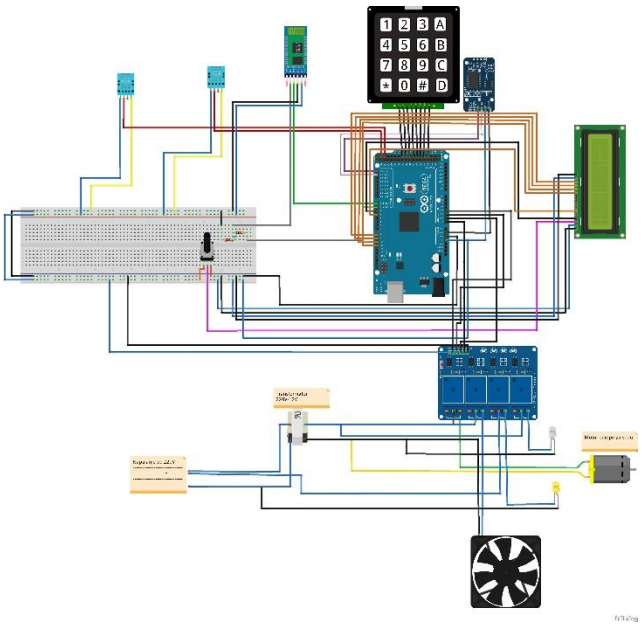


Slika 9. Real time clock.

B. Izrada sistema

Povezivanje DHT11 senzora, LCD displeja, četvoro kanalnog releja, HC-05 blutut modula, membranske tastature i real time clock-a sa Arduino Mega se vrši preko bread boarda, dok se led traka za osvetljenje, sijalica za grejanje i ventilator povezuju na relej. Šema povezivanja komponenti prikazana je naslici 10.

Softverski deo našeg projekta „Teraino" čine 2 dela. Prvi je osnovno programiranje i podešavanje Arduino Mega okruženja preko Arduino IDE programa, dok drugi deo čini programiranje blutut aplikacije preko MIT APP Inventora. Dva DHT11 senzora prikuplja podatke o temperature i vlažnosti vazduha (slika 11) nakon čega ih Arduino Mega obrađuje i ispisuje podatke na LCD ekranu i blutut aplikaciji (slika 12). Korisnik može pomoću tastature ili blutut aplikacije da pali i gasi led traku, grejalicu, ventilator i pumpu za vodu. U blutut aplikaciji postoji opcija uključivanja i isključivanja automatske kontrole. Automatska kontrola predstavlja regulaciju temperature i vlažnosti vazduha prema unosu korisnika.



Slika 10. Šema povezivanja.

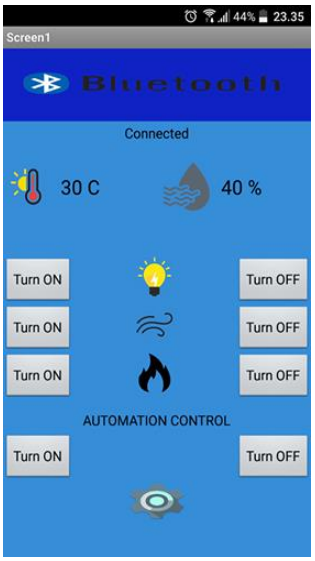
```
humidity = dht.readHumidity(); //prikuplja vlažnost vazduha sa prvog DHT11 senzora
temperature = dht.readTemperature(); //prikuplja temperaturu sa prvog DHT11 senzora

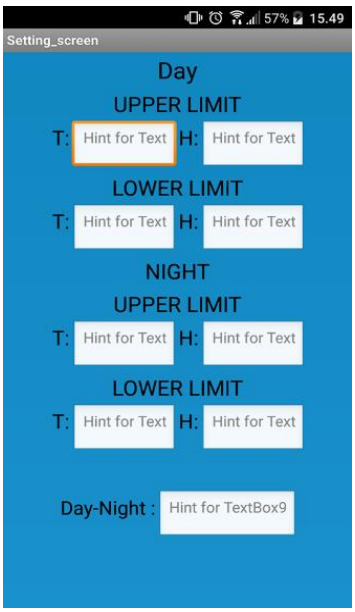
humidityl = dht1.readHumidity();//prikuplja vlažnost vazduha sa drugog DHT11 senzora
temperaturel = dht1.readTemperature();//prikuplja temperatur sa drugog DHT11 senzora


float avg_temperature = (temperature + temperaturel) / 2;//izračunava srednju vrednost temperature
float avg_humidity = (humidity + humidityl) /2;// izračunava srednju vrednost vlažnosti vazduha

lcd.setCursor(0,0);
lcd.print("Temperature:");
lcd.print(avg_temperature);  //ispis temperature na lcd ekranu
lcd.print("C");
lcd.setCursor(0,1);
lcd.print("Humidity : ");
lcd.print(avg_humidity);     //ispis vlažnosti vazduha na lcd ekranu
lcd.print(" %");
```

Slika 11. Kod za prikupljanje podataka.





Slika 12. Izgled blutut aplikacije.

III. ZAKLJUČAK

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje iz informatike, elektronike, elektrotehnike i praktičnog rada sa elektromaterijalima. Rezultat ovog rada je funkcionalni sistem za praćenje i upravljanje parametrima u terarijumu. Namenjen je životinjama i biljkama kojima su potrebni specijalni uslovi za život. Praćenje parametara omogućeno je manuelnim putem i mobilnim telefonom preko mobilne aplikacije. Prednost sistema je jednostavna i tačna regulacija parametara u terarijumima. Takođe dobijeni sistem je moguće je prilagoditi posebnim zahtevima korisnika. Nedostatak sistema je dizajn samog proizvoda i poteškoća u nabavci kvalitetnih materijala za izradu okvira terarijuma. Moguća su i dalja unapređenja projekta dodavanjem kamere i dodatnim priključenjem terarijuma direktno na internet.

ZAHVALNICA

Autori rada zahvaljuju se Tehničkoj školi u Zaječaru rad i pružanja tehničkih i materijalnih sredstava. Takođe velika zahvalnosti profesorima elektro-informatičkog aktiva škole na pruženoj podršci, a posebno mentoru dip.inž.elek. Dejanu Nikoliću.

LITERATURA

[1] Arduino mega dostupno na: *Arduino.cc/Arduino-mega-2560-rev3*
[2] Arduino Software IDE dostupno na: *Arduino.cc/en/Main/Software*
[3] MIT App Inventor dostupno na: Appinventor.mit.edu

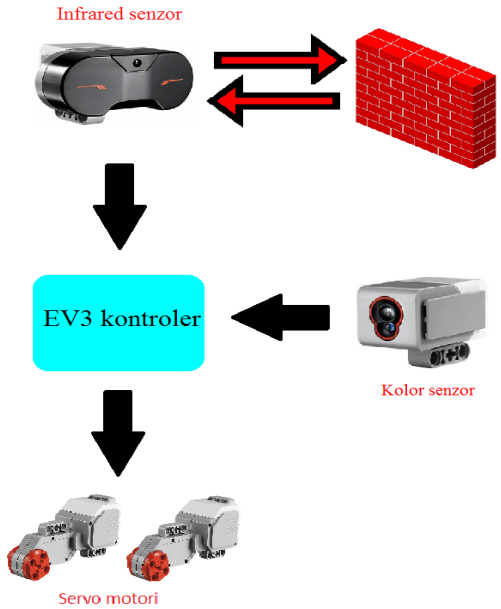
Rešavanje lavirinta pomoću infrared senzora na Lego platformi

Sofija Pavlović, David Nikolić, Dejan Nikolić

Sadržaj - Ovaj rad ima za cilj da predstavi jednu od mogućih primena infrared senzora na Lego platformi za rešavanje lavirinta. Osnovu sistema čine: Lego EV3 brick, infrared senzori, pomoćni kolor senzor i elektromotori za kretanje, iz Lego paketa [1]. Programiranje se vrši u softverskom paketu Lego mindstorms EV3.

I. UVOD

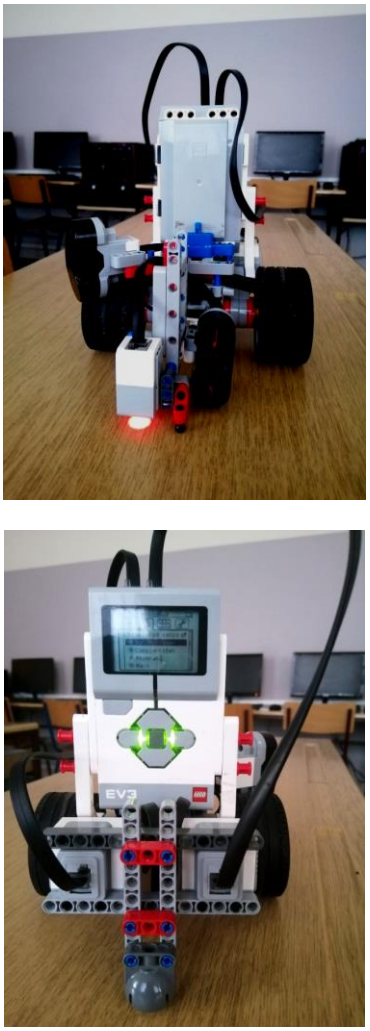
Sistem za upravljanje funkcioniše tako što infrared senzor pomoću infracrvene svetlosti detektuje udaljenost Lego platforme od zida lavirinta, a zatim te informacije prenosi u EV3 kontroler. EV3 kontroler obrađuje podatke na osnovu kojih odlučuje u kom smeru pokreće motore. Na osnovu izmerenih vrednosti infrared senzora, Lego platforma prati desni ili levi zid lavirinta, a pomoću kolor senzora detektuje izlaz iz lavirinta. Na slici 1. prikazana je osnovna šema Lego platforme.



Slika 1. Osnovna šema Lego platform.

Sofija Pavlović učenica drugog razreda smera elektrotehničar informacionih tehnologija, David Nikolić učenik druge godine smera elektrotehničar računara u „Tehničkoj školi Zaječar“, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: sofijaivana003@gmail.com, dakicar323@gmail.com

Na slici 2. Prikazana je gotova Lego platforma sa prednje i zadnje strane.



Slika 2. Lego konstrukcija sa prednje i zadnje strane.

II. PRINCIP RADA

Ovaj sistem funkcioniše tako što EV3 kontroler prikuplja informacije od infrared senzora i na osnovu obrade podataka kontroliše kretanje motora. Osnovna namena sistema je rešavanje lavirinta i samostalan prolaz kroz prostor sa preprekama.

A. Predstavljanje komponenata i priprema za izradu

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

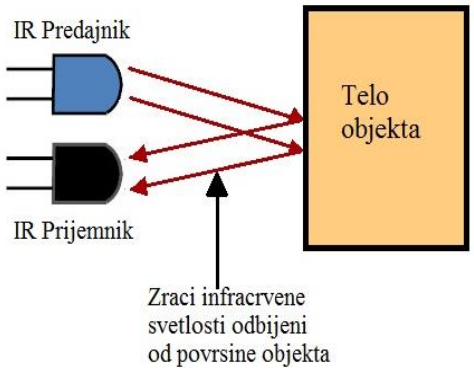
EV3 kontroler je moguce povezati na računar, preko koga se programira, putem USB Kabla, Bluetooth-a ili WiFi-a, a takodje je kompatibilan i sa Android i iOS operativnim sistemima.

Na slici 3 prikazan je EV3 brick koji je korišćen u razvoju ovog projekta.



Slika 3. EV3 brick.

Infrared senzor (IR senzor) [3] je uređaj koji emituje i detektuje IR zračenje kako bi izmerio udaljenosti tela (vezanog za njega) od određenog objekta, odnosno kako bi otkrio predmete/prepreke u njegovom dometu. U ovom slucaju je to digitalni IR sensor prikazan na slici 4. Senzor se sastoji od predajnika i prijemnika, koji je sačinjen od fotodiode i fototranzistora. Funkcioniše tako što predajnik emituje infracrvenu svetlost koja se odbija od objekta/prepreke u blizini. IR prijemnik detektuje odbijenu svetlost I na osnovu toga Lego platforma zna na kojoj se udaljenosti nalazi od prepreke.



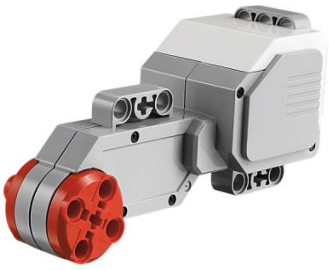
Slika 4. Princip rada IR senzora.

Na slici 5. prikazan je infrared sensor korišćen u projektu.



Slika 5. Infrared sensor.

Servo motor [4] (prikazan na slici 6) koristi talasne povratne informacije za preciznu kontrolu (precizno do jedan stepen) upravljanja motorom. Korišćenjem ugrađenih rotacionih senzora, servo motor se može usaglasiti i sa drugim motorima na robotu tako da može da se kreće ravnom linijom istom brzinom.



Slika 6. Servo motor.

Kolor senzor [5] detektuje intenzitet svetlosti koji do njega dopire i na osnovu toga Lego platforma odredjuje boju ka kojoj je usmeren senzor. Na slici 7. je prikazan kolor senzor korišćen u izradi sistema.



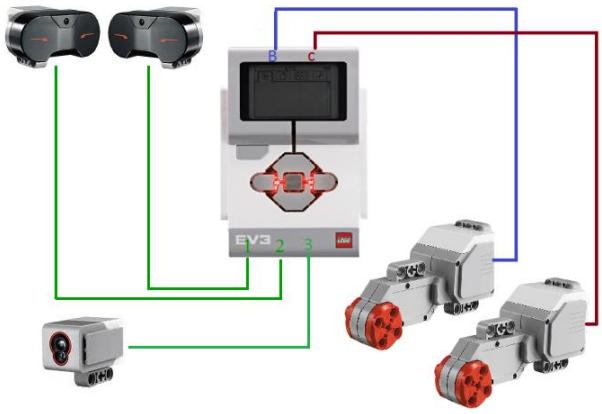
Slika 7. Kolor sensor.

B. Izrada sistema

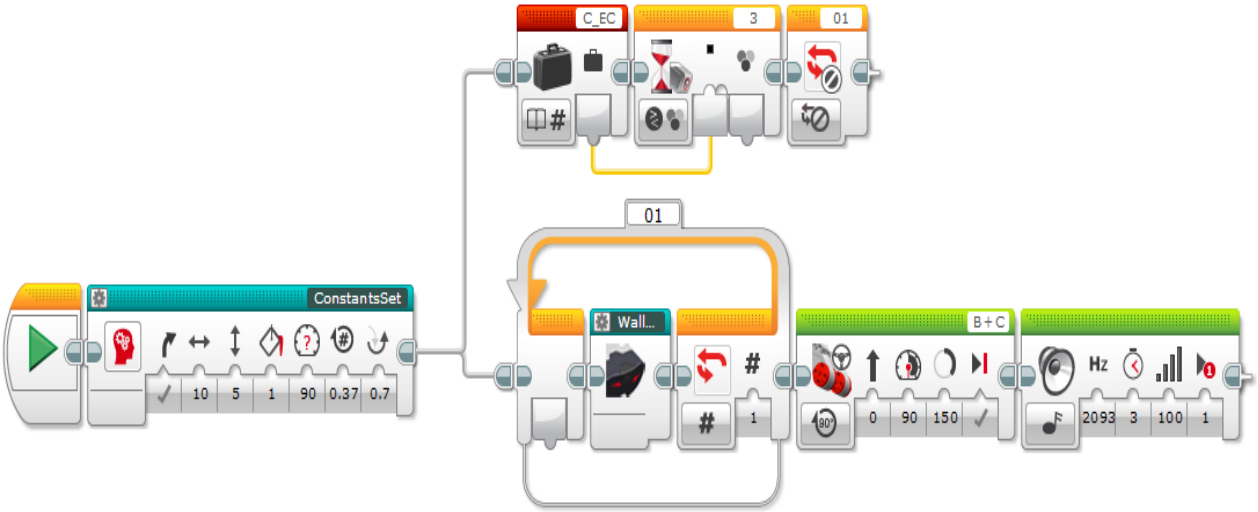
Povezivanje senzora i motora sa Lego EV3 se vrši pomoću kablova sa RJ12 konektorima. Infrared senzor se povezuje tako što se jedan kraj kabla poveže sa senzorom a drugi se konektuje u EV3 kontroler na ulaznom portu broj 1. Drugi infrared senzor se na isti način konektuje na

ulaznom portu broj 2, a pomoćni kolor senzor na portu broj 3. Motore povezujemo pomoću izlaznih portova i to tako da levi motor povežemo na port B,a desni na port C. Na slici 8. prikzana je šema za povezivanje komponenti na EV3 modul.

Nakon povezivanja, napisan je kod u softverskom paketu Lego Mindstorms EV3[6] koji se preko bluetooth-a (ili USB kabla) prebacuje u EV3 kontroler. Nakon pokretanja kontrolera infrared senzor se aktivira i počinje da meri udaljenost od zida lavirinta. Na osnovu očitanih vrdnosti infrared senzora aktiviraju se motori koji Lego platformu prilikom prolaska lavirinta održavaju na konstantnoj udaljedosti od zida. Izlaz lavirinta je označen crnom bojom, tako da robot prati zid sve dok kolor senzor ne detektuje crnu boju, tada prestaje praćenje zida, a robot zvukom signalizira izlazak iz lavirinta.



Slika 8. Šema za povezivanje.



Slika 9. Kod celokupnog sistema.

Na slici 9. Prikazan je ceo kod sistema. Prvi blok sistema je funkcija koja postavlja inicialne parametre. Petlja u donjoj grani sadrži fukciju za praćenje zida, koja se prekida u trenutku kada kolor sensor detektuje izlaz, što se izvršava u gornjoj grani koda. Nakon prestanka praćenja zida, odnosno izlaska iz lavirinta robot zvukom signalizira da je pronašao izlaz.

III. Zaključak

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje ne samo iz informatike već i iz elektronike, elektrotehnike, mehatronike i praktičnog rada sa elektro materijalima. Autori rada zadovoljni su krajnjim prozvodom.

Ovaj projekat pruža mogućnost za dalja istraživanja kao što je kretanje kroz ograničen prostor sa preprekama i kontrolisanje istog preko interneta ili mobilne aplikacije.

Zahvalnica

Autori rada zahvaljuju se Tehničkoj školi u Zaječaru radi pružanja tehničkih i materijalnih sredstava takođe velika zahvalnost i profesorima elektro-informatičkog aktiva škole na pruženoj podršci.

Literatura

[1] Lego Page (https://www.lego.com/en-us/mindstorms/about-ev3) Septembar, 2019
[2] EV3 kontroler (http://www.legama.si/en/pregled-produktov-za-ev3) Septembar, 2019
[3] Infrared sensor (http://www.legoengineering.com/ev3-sensors/) Septembar, 2019
[4] Servo motor (https://en.wikipedia.org/wiki/Servomotor) Septembar, 2019
[5] Kolor sensor (http://www.legoengineering.com/ev3-sensors/) Septembar, 2019
[6] Lego Mindstorms EV3 (https://www.lego.com/en-us/mindstorms/learn-to-program) Septembar, 2019

Projekat Centra za promociju nauke "Pretočimo znanje u praktičan rad" realizovan u Naučnim klubovima Niš i Leskovac

Miloš Marjanović, Danko Danković, Danijel Danković

Sadržaj – Cilj ovog rada je da predstavi značaj Centra za promociju nauke u realizaciji projekata promocije i popularizacije nauke. Opisane su misija i vizija Centra za promociju nauke, kao i tipovi projekata koje Centar finansira i realizuje sa svojim saradnicima. Detaljno je opisan tok realizacije i analiziran značaj projekta "Pretočimo znanje u praktičan rad" uz podršku Centra. Na osnovu rezultata anketiranja učesnika projekta zaključeno je da učenici osnovih i srednjih škola žele više praktičnog rada koji će pratiti predavanja u okviru redovne nastave.

I. Misija i vizija Centra za promociju nauke

Centar za promociju nauke (CPN) osnovan je 2010. godine od strane Ministarstva nauke Srbije sa zadatkom da promoviše nauku i tehnologiju. CPN svoje programe i aktivnosti sprovodi radeći sa istraživačkim i obrazovnim institucijama u Srbiji i širom sveta, a blisko sarađuje sa republičkim ministarstvima, medijima i pravnim sektorom (slika 1). Misija CPN-a je da premosti jaz između nauke i društva okupljajući sve relevantne aktere i širu javnost u procesu istraživanja i inovacija [1]. Krajnji cilj je integracija društva u istraživačke procese kako bi se stekao što bolji uvid u potrebe građana i što adekvatnije suočavalo sa društvenim izazovima. CPN organizuje veliki broj izložbi, predavanja, panel-diskusija, radionica i drugih događaja na različite teme. Prepoznatljiva nit svih formata je da su najčešće interaktivni, tako da posetioci, ukoliko to žele, imaju aktivnu ulogu u realizaciji programa. U okviru svoje izdavačke delatnosti, CPN izdaje četiri do pet novih naslova svake godine. Takođe, CPN objavljuje i naučnopopularni časopis "Elementi" četiri puta godišnje. CPN je poznat po svojoj međunarodnoj saradnji i u tom pogledu smatra se regionalnim liderom u jugoistočnoj Evropi. Trenutno je partner u osam H2020 projekata, na jednom projektu Kreativne Evrope i JRC-a i četiri Erasmus+ projekta, tako da je učestvovao na ukupno 32 projekta koje je finansirala Evropska Unija. Neke od programskih aktivnosti CPN-a su: Mejkers Spejs – laboratorija za sve kreativce, Festival nauke, Mobilni planetarijum, DNK – dečiji naučni kamp, Naučni kamion, Dan nauke – 10. jul…

Miloš Marjanović i Danijel Danković su sa Katedre za mikroelektroniku, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš. E-mail: milos.marjanovic@elfak.ni.ac.rs, danijel.dankovic@elfak.ni.ac.rs

Danko Danković je iz Tehničke škole "Rade Metalac" Leskovac, Dimitrija Tucovića bb, 16000 Leskovac. E-mail: ddanko@live.com

CPN ima mrežu od 15 naučnih klubova širom Srbije. Naučni klub je zamišljen kao mesto za učenje i sticanje znanja na neformalan način, kroz radionice, predavanja i druge aktivnosti [2]. Jedan od glavnih zadataka kluba je da poveže sve kojima je nauka bliska i zanimljiva, bez obzira na godine, nivo znanja i oblasti interesovanja. Članstvo u klubu je dobrovoljno, a za učenike, studente, nastavnike i naučne radnike je besplatno. Prvi Naučni klub otvoren je u Leskovcu, 5. aprila 2016. godine [3]. Do danas, naučni klubovi postoje i u Nišu, Čačku, Užicu, Knjaževcu, Kragujevcu, Smederevu, Kruševcu, Kikindi, Novom Pazaru, Kanjiži, Šabcu, Beogradu, Gornjem Milanovcu.



Slika 1. Logo Centra za promociju nauke.

II. Projekti Centra za promociju nauke

CPN obajvljuje jednom godišnje Javni poziv za finansijsku podršku projektima promocije i popularizacije nauke. Cilj javnog poziva je podizanje nivoa naučne pismenosti i širenja naučne kulture sprovodeći postupak decentralizacije promocije i popularizacije nauke na teritoriji Republike Srbije. Ukupna raspoloživa sredstva izdvojena za ovu namenu u 2019. godini su 9.400.000 dinara [4]. Projekti su podeljeni u 4 kategorije. Prvu kategoriju čine projekti promocije nauke u naučnim klubovima, druga kategorija obuhvata projekte promocije nauke, dok su treća kategorija idejna rešenja za interaktivne eksponate u parkovima nauke, a četvrta kategorija su projekti izrade eksponata u parkovima nauke.

Tim koga čine Danijel Danković, docent na Elektronskom fakultetu u Nišu, Miloš Marjanović, asistent na Elektronskom fakultetu u Nišu i Danko Danković, profesor u Tehničkoj školi "Rade Metalac" Leskovac realizovali su tokom 2018. godine projekat CPN-a iz prve kategorije pod nazivom "Pretočimo znanje u praktičan rad" [5]. Isti tim u toku 2019. godine realizuje dva projekta CPN-a, jedan iz prve kategorije pod nazivom "Praktičnim radom do znanja" i jedan iz druge kategorije pod nazivom "Idemo i mi na IEEESTEC konferenciju". Projekti iz prve

kategorije realizuju se u Naučnim klubovima Niš i Leskovac, uz podršku Regionalnog centra za profesionalni razvoj zaposlenih u obrazovanju Niš [6] i Centra za stručno usavršavanje u obrazovanju Leskovac [3] (slika 2).

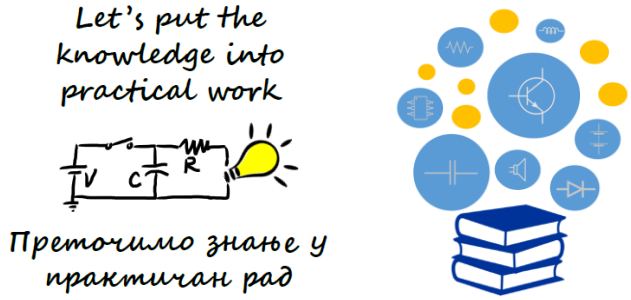


Slika 2. Logoi Regionalnih centara Niš i Leskovac.

III. PROJEKAT "PRETOČIMO ZNANJE U PRAKTIČAN RAD"

Praktičnim radom se formule, zakoni i proračun potvrđuju iz drugog, interesantnijeg ugla, pa je i interesovanje učenika za radionice veće. Oni eksperimentisanjem proveravaju svoje, ali i poznate teorije i zajedno sa vršnjacima realizuju vežbe. Kroz ove dve rečenice najkraće se opisuje cilj projekta "Pretočimo znanje u praktičan rad" (slika 3). Uvod u radionice bio je kroz prezentaciju "Elektricitet i električna energija". Kroz radionice je aktivno učestvovalo oko 600 učenika osnovih i srednjih škola. U tabeli I dat je pregled svih škola čiji su učenici bili deo radionica u okviru projekta "Pretočimo znanje u praktičan rad". Treba napomenuti da je u nekim školama organizovano i nekoliko radionica u nekoliko termina. Neke od radionica realizovane su u prostorijama Naučnih klubova Leskovac i Niš, a neke u školama. Učenici su uvek bili podeljeni u 4 grupe, a svaka grupa imala je poseban paket opreme za izvođenje eksperimenata. Promocija projekta izvršena je i u okviru obeležavanja Noći istraživača, IEEE dana 2018, na Forumu naprednih tehnologija i konferenciji studentskih projekata IEEESTEC [7].

U zavisnosti od uzrasta, praktična realizacija električnih kola za učenike osnovnih škola urađena je uz pomoć *ELENCO Snap Circuits* kompleta [8] (slika 4). Komplet se sastoji od komponenata sa kojima se izvode razni projekti iz oblasti elektronike, a glavna namena kompleta jeste da mladima približi ovu vrstu nauke iako nemaju prethodno stečeno znanje. Izvođenjem 750 dostupnih projekata, koji uključuju i oblast obnovljivih izvora energije, na zanimljiv način vrši se promocija nauke u cilju njene popularizacije. Tokom jedne radionice učenici savladaju desetak vežbi koje su prilagođene njihovom trenutnom nivou znanja. Koristeći bateriju, prekidač i/ili taster, sijalicu i provodnike, učenici kroz timski i samostalni rad obnavljaju i stiču nova znanja o pravilnom funkcionisanju i povezivanju elementarnog strujnog kola. Potencijalne greške koje su se javljale tokom povezivanja komponenata, a što je uobičajeno za svaki eksperimentalni rad, samostalno su tražili i otklanjali. Znali su unapred šta očekuju kao krajnji odziv, pa su na osnovu toga i zaključivali gde postoji greška. Deci je demonstrirano koji materijali provode struju, a koji ne. Treba istaći da je organizacija radionice u odeljenju petog razreda u OŠ "Radoje Domanović" Niš proglašena za "Delo meseca novembra", nagrada koja se dodeljuje stimulativno za kolektiv, roditelje, učenike i prijatelje škole [9].



Slika 3. Logoi projekta "Pretočimo znanje u praktičan rad".

TABELA I
SPISAK INSISTUCIJA KOJE SU BILE DEO PROJEKTA

Institucija	Broj radionica
ETŠ "Mija Stanimirović" Niš	2
OŠ "Radoje Domanović" Niš	2
OŠ "Branko Radičević" Brestovac	1
OŠ "Vuk Karadžić" Pečenjevac	1
TŠ "Rade Metalac" Leskovac	1
Gimnazija Leskovac	3
OŠ "Josif Kostić" Leskovac	1
Vrtić "Naše dete" Leskovac	1
OŠ "Vasa Pelagić" Leskovac	1
OŠ "Vuk Karadžić" Leskovac	2
Elektronski fakultet Niš	1
ETŠ "Nikola Tesla" Niš	2



Slika 4. Radionice u osnovnim školama u Leskovcu i Nišu.

Slično, za učenike srednjih škola u zavisnosti od razreda i usmerenja, izbor vežbi se menjao. Na primer, na radionici u ETŠ "Mija Stanimirović" Niš za učenike obrazovnog profila Elektrotehničar informacionih tehnologija posebna pažnja je posvećena logičkim kolima (NE, ILI, I, NILI, NI). Takođe, učenicima je pružena

prilika da korišćenjem unimera i primenom Omovog zakona izmere i izračunaju napone i struje u kolima. S druge strane za učenike istog smera u Tehničkoj školi "Rade Metalac" Leskovac organizovana je radionica sa ciljem realizacije prototipa semafora korišćenjem Arduino okruženja, otpornika i LED dioda (slika 5). Učenici su se prvo upoznali sa elektronskim komponentama, potom su realizovali električnu šemu na protoploči i konačno su napisali odgovarajući softver za mikrokontroler. Nakon što su uspešno realizovali zadatak, učenici su samoinicijativno vršili modifikacije softvera. Utisak posle ove radionice bio je da je učenicima teže bilo savladavanje hardverskog dela projekta jer su se prvi put susreli sa realnim komponenatama, dok im je savladavanje softverskih algoritama znatno lakše u skladu sa prethodnim iskustvom. Posle dvočasovnog rada srednjoškolci su pokazali interesovanje da samostalno nastave sa aktivnostima u ovoj oblasti.



Slika 5. Radionice u srednjim školama u Leskovcu i Nišu.

Projekat je realizovan i u školama koje se nalaze u seoskim sredinama (slika 6). Cilj je takođe bio da se deci približi praktičan rad imajući u vidu različite uslove života u tim sredinama i opremljenost škola sredstvima za savremenu nastavu. Realizatori projekta posetili su osnovne škole u selima Brestovac i Pečenjevac, oba na teritoriji grada Leskovca.

Najmlađi učesnici radionica bili su predškolci iz dečijeg vrtića "Naše dete" Leskovac (slika 7). Iako najmlađi, učesnici su pratili i bili strpljivi tokom rada, gde su kroz drugačiji pristup otkrili nova znanja o elektricitetu i struji. Kroz prezentaciju su imali priliku da razviju svest o dejstvu struje i njenih efekata na čoveka i okolinu. Predočeno im je koliko je struja korisna, ali su im pokazane i potencijalne opasnosti koje ona izaziva. Najstariji učesnici radionica bili su studenti treće godine Elektronskog fakulteta u Nišu, koji pored teorijskog imaju i ogromno iskustvo u praktičnom radu tokom realizacije laboratorijskih vežbi, ali i prilikom samostalnog rada na mnogim projektima. Zaključak sa ove radionice je da bi i oni uz minimlanu obuku mogli da budu edukatori srednjoškolaca.



Slika 6. Radionice u školama iz seoskih sredina.



Slika 7. Radionice u vrtićima.

Na kraju projekta pripremljena je anketa i pozvani su učesnici radionica da odgovore. Anketa se sastojala iz pitanja zatvorenog i otvorenog tipa. Prilikom ocenjivanja korišćena je sledeća gradacija: 1 – ne slažem se, 2 – delimično se ne slažem, 3 – nemam definisan stav, 4 – delimično se slažem i 5 – slažem se. U anketi, učesnici su odgovarali upisivanjem odgovora i na pitanja: šta im se najviše dopalo, šta im je smetalo, kakva je ukupna impresija, a mogli su i da dopišu šta smatraju da je važno, a nismo ih pitali. Anketu je popunilo 108 učesnika, rezultati ankete dati su u tabeli II.

Analizom rezultata ankete može se zaključiti da je uvodna prezentacija bila korisna za oko 88% anketiranih, 82% njih smatra da su vežbe korisne, dok taj stav u potpunosti ne deli oko 1.8%. Najveća podeljenost stava uočena je kod pitanja o angažovanju nastavnika, gde oko 65% anketiranih smatra da ovakav vid nastave iziskuje veliko angažovanje nastavnika, dok 4.7% smatra da nastavnici ne treba da se dodatno angažuju. Ukupno 97% anketiranih je zadovoljno realizacijom projekta.

Karakteristični odgovori na pitanje šta im se najviše dopalo su: kada smo pravili strujno kolo, dopalo mi se igranje, da upalimo sijalicu, to što smo radili zajedno i družili se, prespajanje žica, rad sa Arduino pločom, to što

smo proširili znanje o programiranju. Očigledno je iz odgovora da su neki od anketiranih učenici osnovnih, a neki srednjih škola. Mnogo manje odgovora bilo je na pitanje šta im je smetalo: neka deca nisu radila kao tim, neki su pričali dok su radili, bilo bi bolje da smo sami birali vežbe koje radimo. U ukupnoj impresiji, anketirani su istakli sledeće: "impresioniran sam", "mislim da bi trebalo da imamo još takvih časova", "neopisivo odlično", "ja ne volim ovakve stvari ali je sasvim dobro". Jedan srednjoškolac je sumirao "nova znanja, iskustva i pomoć za dalje opredeljenje", što pokazuje da su cilj i misija projekta ispunjeni. U napomeni je većina odgovora bila sa porukom "češće organizujte ovakve radionice, zašto ne dolazite svaki dan", što potvrđuje činjenicu da učenici mnogo lakše savladavaju gradivo kada vide da se ono i praktično primenjuje, te da žele više praktične nastave u redovnom školovanju.

IV. ZAKLJUČAK

Članovima tima su kroz višegodišnji rad sa učenicima osnovnih, srednjih škola i studentima uočili poznatu činjenicu da je primenljivo praktično znanje ono što im najviše nedostaje. Iz tog razloga je preko međunarodnog udruženja inženjera elektronike i elektrotehnike (IEEE), Elektronskog fakulteta i grada Niša pokrenuto nekoliko radionica sa najsavremenijom opremom (elektronske komponente, kola i merna oprema), kako bi se učenicima približio praktičan rad. Na taj način, učenici se podstiču da bolje uče. Skloni su eksperimentisanju, željni da provere poznatu ili svoju teoriju, neku stvarnost koja nas okružuje i koju uočavaju, samo im treba pružiti priliku. Potrebno im je omogućiti veći broj eksperimenata, simulacija, aplikacija i interaktivnih animacija. Realizacijom radionica sa osnovcima i srednjoškolcima kroz ovaj projekat dat je značajan doprinos praktičnoj primeni znanja, promociji i popularizaciji nauke i tehnike. Članovi tima izuzetno su zadovoljni realizacijom projekta i smatraju da se sa istim aktivnostima treba nastaviti i u budućnosti. S tim u vezi, u toku 2019. godine tim radi na projektu Centra iz prve kategorije "Praktičnim radom do znanja", kao i na projektu iz druge kategorije "Idemo i mi na IEEESTEC konferenciju".

TABELA II
REZULTATI ANKETE UČESNIKA PROJEKTA

Pitanje	1 (ne slažem se)	2 (delimično se ne slažem se)	3 (nemam definisan stav)	4 (delimično se slažem)	5 (slažem se)
Prezentacija "Elektricitet i električna energija" je korisna.	0	0	2	11	95
Bilo mi je zabavno prilikom realizacije vežbi.	2	0	3	9	94
Smatram da ove vežbe mogu da budu od velike koristi za učenike.	2	0	9	8	89
Za ovakve aktivnosti potrebno je i veliko angažovanje učitelja, nastavnika i profesora.	5	3	6	24	70
Zadovoljan sam kako je ovaj projekat realizovan.	0	0	0	3	105

ZAHVALNICA

Ovaj rad deo je aktivnosti na projektima Centra za promociju nauke: "Pretočimo znanje u praktičan rad", "Praktičnim radom do znanja" i "Idemo i mi na IEEESTEC konferenciju".

LITERATURA

[1] Centar za promociju nauke. [OnLine] Dostupno na: http://www.cpn.rs/o-centru/
[2] Naučni klubovi. [OnLine] Dostupno na: http://www.cpn.rs/programi/mreza-naucnih-klubova/
[3] Centar za stručno usavršavanje u obrazovanju Leskovac [OnLine] Dostupno na: http://www.csuleskovac.edu.rs/
[4] Javni poziv CPN-a za finansijsku podršku projektima promocije i pupolarizacije nauke u 2019. godini. [OnLine] Dostupno na: http://prijave.cpn.rs/
[5] Zvanični sajt projekta "Pretočimo znanje u praktičan rad". [OnLine] Dostupno na: https://pretocimoznanje.wordpress.com/
[6] Regionalni centar za profesionalni razvoj zaposlenih u obrazovanju u Nišu [OnLine] Dostupno na: http://www.rcnis.edu.rs/
[7] Zvanični sajt IEEESTEC konferencije. [OnLine] Dostupno na: ieee.elfak.ni.ac.rs
[8] ELENCO Snap Circuit kits u okviru EDS-ETC programa [OnLine] Dostupno na: https://eds.ieee.org/chapters/the-eds-etc-program
[9] Sajt OŠ "Radoje Domanović" Niš. [OnLine] Dostupno na: http://www.osrdomanovic.edu.rs/

Projektovanje i realizacija CNC mašine za iscrtavanje

Mikica Radenković, Sava Stojanović

Sadržaj – U ovom radu je prikazan proces projektovanja i realizacije CNC mašine za iscrtavanje vektorskih i rasterskih grafika baziranu na ATMEGA 328P mikrokontroleru.

I. UVOD

CNC (*Computer Numerical Control*) mašina je svaka mašina za mašinsku obradu koju kontroliše računar. Način funkcionisanja same mašine bazira se na realizaciji operacija prema unetom alfanumeričkom kodu preko računarske upravljačke jedinice koja direktno upravlja radom mašine. Može se realizovati u različitim konfiguracijama i samim tim ima i različitih primena. Glavne prednosti CNC mašina u odnosu na tradicionalne mašine su brži i precizniji rad, jeftiniji proizvod omogućuje rad sa opasnim materijama i uslovima nepogodnim za čoveka.

II. PRINCIP FUNKCIONISANJA

Princip funkcionisanja mašine najjednostavnije je objasniti podelom na softver i hardver uređaja.

Softver koji vrši čitanje grafika „*Benbox*“ je baziran na GRBL kodu [1]. GRBL kod predstavlja G-kod optimizovan mikrokontroleru Atmega 328. G-kod koji ima više varijanti, zajednički je naziv za najčešće korišćeni programski jezik numeričke kontrole. Koristi se uglavnom u računarskoj proizvodnji za kontrolu CNC mašina. G-kod je jezik kojim se zadaju mašinskim alatima komande. Komande G-koda daju mašinskom upravljaču (mikrokontroleru) komande kako motori moraju raditi, kojom brzinom i kojim smerom. Komande GRBL koda su prikazane na slici 1.

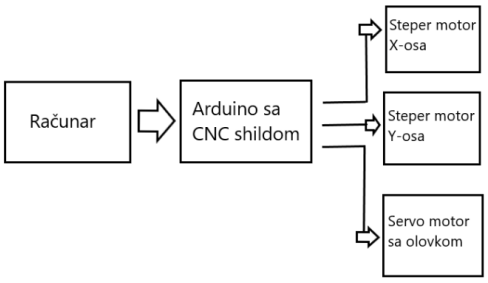
III. HARDVER UREĐAJA

Na slici 2 prikazana je blok šema hardverskog dela uređaja. Sastoji se iz mikrokontrolera koji povezuje računar sa motorima, step motore koji pomeraju olovku i servo motora koji kontroliše minicu olovke. U daljem tekstu rada biće opisan svaki od ovih delova mašine ponaosob.

Arduino Uno sa CNC shield V3 sa A4988 Stepper Driver-ima je mikrokontroler koji predstavlja digitalni elektronski uređaj u obliku integrisanog kola [2]. Namena mikrokontrolera je upravljanje uređajima i procesima, pa u sebi ima integrisan mikroprocesor, memoriju, digitalne i analogne ulaze i izlaze, digitalne satove, brojače, oscilatore, komunikacione sklopove i druge dodatke za koje je nekada bio potreban niz posebnih integrisanih kola. Mikrokontroler normalno radi u kontrolnoj petlji, dakle očitava ulaze i zatim podešava izlaze u skladu sa svojim programom. Petlja se stalno ponavlja dok traje kontrola procesa. Mikrokontroler vrši čitanje G-koda i na osnovu njega izdaje naredbe motorima.

```
$0 = 10 (step pulse, usec)
$1 = 25 (step idle delay, msec)
$2 = 0 (step port invert mask:00000000)
$3 = 0 (dir port invert mask:00000000)
$4 = 0 (step enable invert, bool)
$5 = 0 (limit pins invert, bool)
$6 = 0 (probe pin invert, bool)
$10 = 3 (status report mask:00000011)
$11 = 0.010 (junction deviation, mm)
$12 = 0.010 (arc tolerance, mm)
$13 = 0 (report inches, bool)
$20 = 0 (soft limits, bool)
$21 = 0 (hard limits, bool)
$22 = 0 (homing cycle, bool)
$23 = 0 (homing dir invert mask:00000000)
$24 = 25.000 (homing feed, mm/min)
$25 = 500.000 (homing seek, mm/min)
$26 = 250 (homing debounce, msec)
$27 = 1.000 (homing pull-off, mm)
$100 = 80.000 (x, step/mm)
$101 = 80.000 (y, step/mm)
$102 = 80.000 (z, step/mm)
$110 = 50000.000 (x max rate, mm/min)
$111 = 50000.000 (y max rate, mm/min)
$112 = 50000.000 (z max rate, mm/min)
$120 = 5000.000 (x accel, mm/sec^2)
$121 = 5000.000 (y accel, mm/sec^2)
$122 = 30.000 (z accel, mm/sec^2)
$130 = 310.000 (x max travel, mm)
$131 = 450.000 (y max travel, mm)
$132 = 200.000 (z max travel, mm)
```

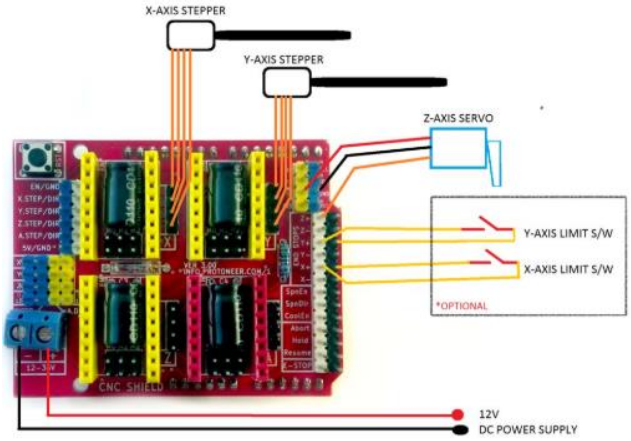
Slika 1. Komande GRBL koda.



Slika 2. Blok šema hardverskog dela sistema.

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

CNC shield V3 (slika 3) može se koristiti kao ploča za kontrolu mašine za graviranje, iscrtavanje, 3D štampač i druge uređaje [3]. Na ploču mogu se montirati 4 step kontrolera, graničnici za sve ose i moguće je dodavanje servo motora. Zajedno sa *Arduino* okruženjem odnosno mikrolontrolerom vrši čitanje GRBL koda.



Slika 3. Električna šema mikrokontrolera sa motorima.

Step motori predstavljaju elektromotor bez četkica koji pretvara digitalne pulseve struje u fiksne inkremente ugaonog pomeranja nazvane koraci (*steps*) [4]. Ova vrsta motora obezbeđuje precizno pozicioniranje tereta, a kontrola motora se vrši direktno računarom i miktrokontrolerom. Dva step motora vrše pomeranje olovke po XY osama i zahtevaju posebno napajanje.

Servo motor (slika 4) se koristi se u aplikacijama gde je potrebno upravljanje ugaonom pozicijom, brzinom i ubrzanjem [5]. Sastoji se od odgovarajućeg motora u kombinaciji sa senzorom koji daje informaciju o poziciji. Za kontrolu servo motora potreban je mikrokontroler. Servo motor poseduje tri priključka, jedan upravljački i dva naponska. Upravljački signal se koristi za upravljanje ugaonom pozicijom izlazne osovine i najčešće je to PWM (*Puls - Width Modulation*) signal. Trajanje impulsa PWM signala određuje poziciju osovine. Pri pojavi signala motor rotira osovinu u smeru u kom će se najpre ostvariti zadata pozicija i zadržava tu poziciji dokle god je signal prisutan. Maksimalni opseg uglova je 180°. Servo motor kontroliše visinu minice u odnosu na papir i na taj način se vrši iscrtavanje na papiru.

Za realizaciju projekta iskorišćene su sledeće komponente:

- Arduino UNO
- CNC shield V3
- A4988 stepper driver x2
- Servo motor SG90
- Čitači za CD x2
- Olovka
- Šper ploča
- Linoleum
- Šrafovi
- Žice
- Napajanje 12V

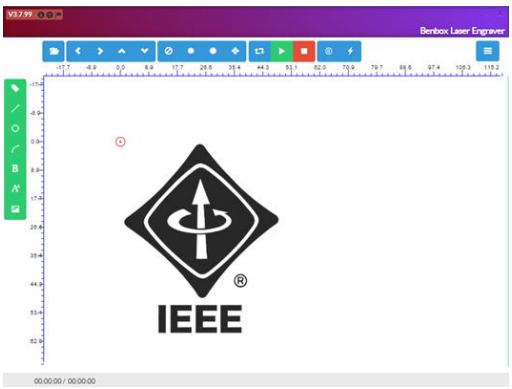


Slika 4. Servo motor.

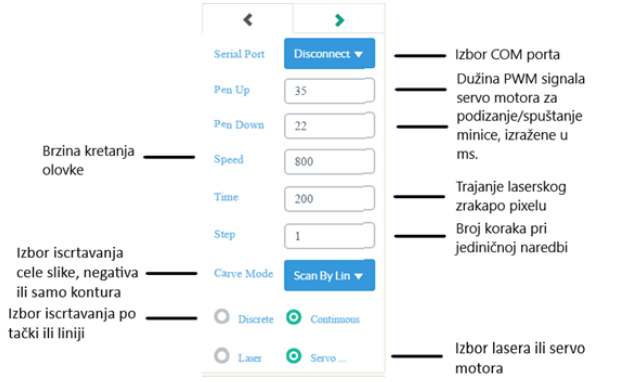
IV. SOFTVERSKI ALATI

Softverski program koji daje komande mikrokontroleru za iscrtavanje grafika je „*Benbox*“ [6]. Program nudi opcije unosa vektorskih i rasterskih grafika u 2D, njihovo skaliranje i pozicioniranje, mogućnost izbora iscrtavanja potpune slike, negativa ili samih kontura u jednoj boji prilikom iscrtavanja ili lasersko graviranje. Radna površina (Slika 5) „*Benbox*“-a sadrži i komadne za ručno pomeranje po XY osi, uključivanje-isključivanje lasera, podizanje i spuštanje minice, učitavanje grafičkih datoteka, startovanje izrade, pauziranje i stopiranje procesa, komandu za ubacivanje firmvera u mikrokontroler.

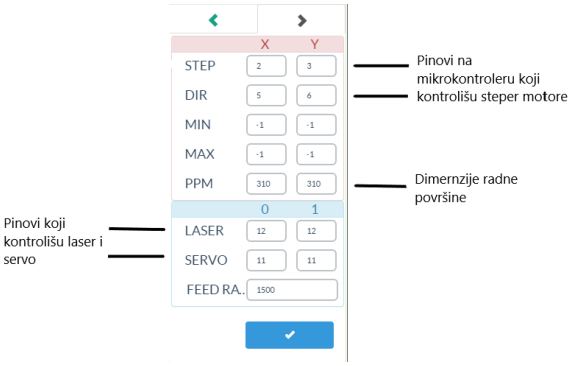
Na slikama 6 i 7 prikazana su podešavanja samog programa „*Benbox*“. Mogu se naći opcije za biranje komunikacionog porta računara sa mikrokontrolerom, podešavanja dužine PWM signala servo motora za podizanje i spuštanje minice, brzina kretanja olovke, trajanje laserskog zraka po pikselu, broj koraka pri jediničnoj naredbi, izbor iscrtavanja cele slike, kontura ili negativa, izbor korišćenja lasera ili servo motora, podešavanje pinova na mikrokontroleru koji kontrolišu step motore, laser ili servo, i podešavanje dimenzija radne površine.



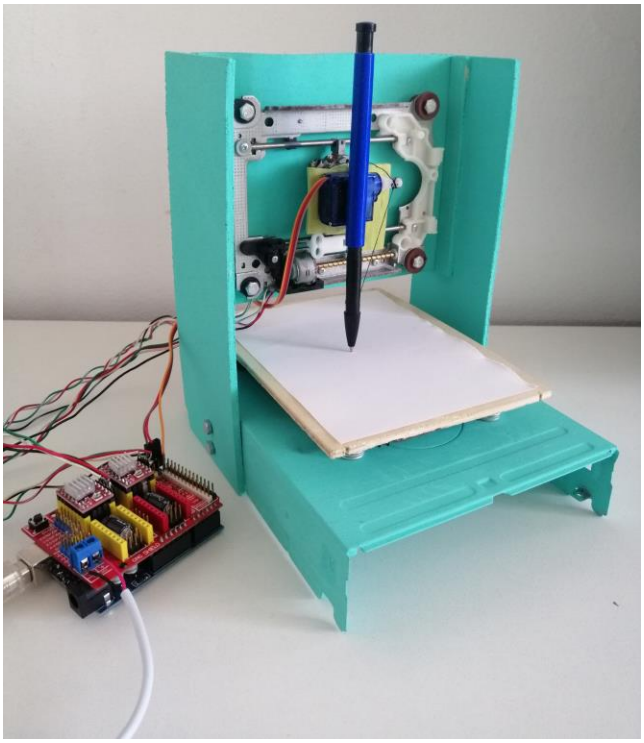
Slika 5. Radna površina „*Benbox*“ programa.



Slika 6. Prikaz prvog prozora za podešavanja.

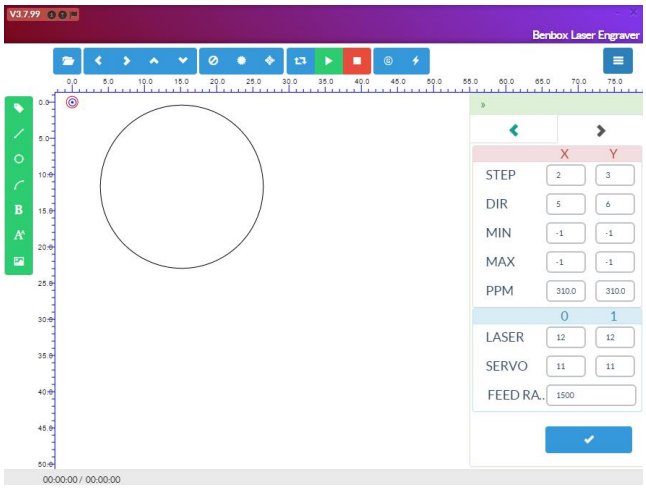


Slika 7. Prikaz drugog prozora za podešavanja.

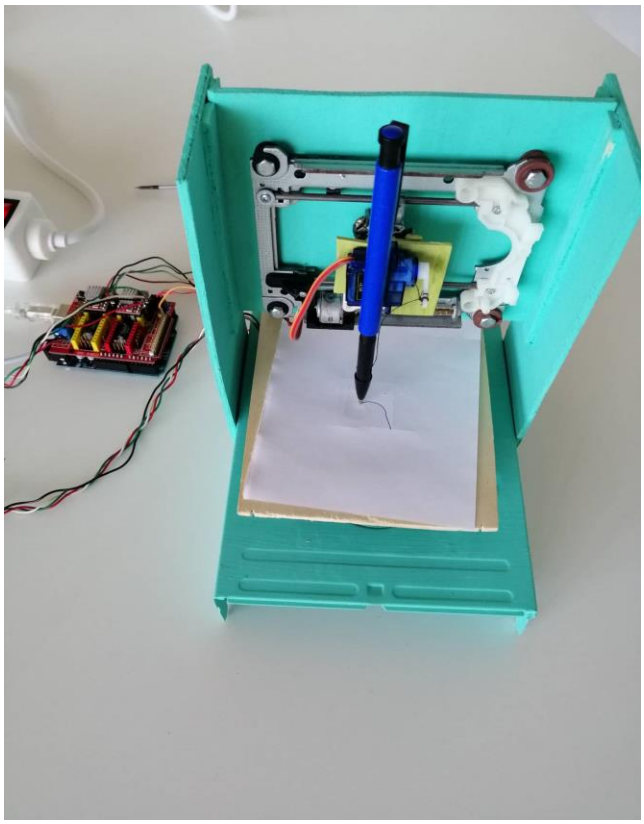


Slika 8. Kompletni izgled CNC mašine za iscrtavanje.

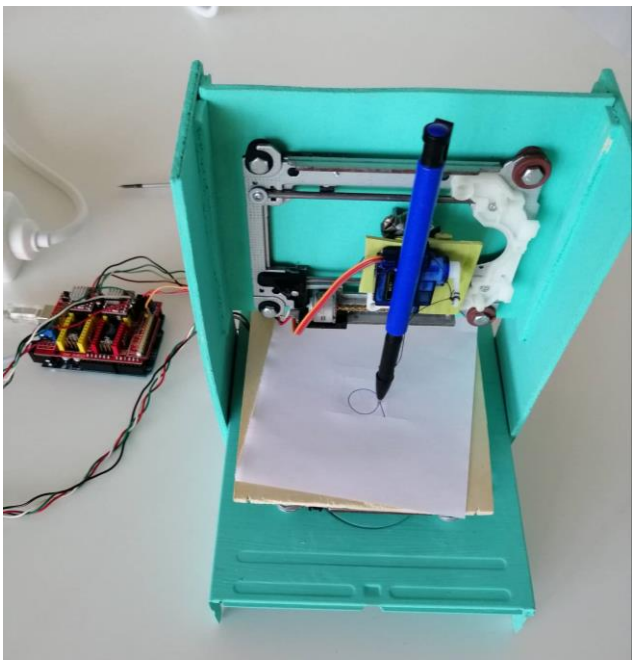
Na slici 8 je prikazan kompletni izgled CNC mašine za iscrtavanje koju smo realizovali. Na slici 9 program „*Benbox*“ za vreme iscrtavanja, na slici 10 mašina u toku iscrtvanja i na slici 11 posle iscrtavanja grafika.



Slika 9. Program „*Benbox*“ za vreme iscrtavanja.



Slika 10. Mašina u toku iscrtavanja.



Slika 11. Mašina posle iscrtavanja.

V. ZAKLJUČAK

U ovom radu prikazana je kompletno završena CNC mašina za grafičko iscrtavanje. Data su objašnjena o korišćenim komponentama i softverskom programu „*Benbox*“.

Moguće je olovku zameniti laserom, čime se postiže graviranje objekta umesto iscrtavanja. Postoji mogućnost dodavanja bušilice sa pomeranjem vrha po Z osi, čime se dobija 3D gravirač.

ZAHVALNICA

Autori se zahvaljuju nastavnicima i saradnicima Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu na pruženoj pomoći i saradnji tokom realizacije ovog projekta.

LITERATURA

[1] GRBL kod, opis, dostupno na: www.instructables.com/id/How-to-Control-a-Servo-Using-GRBL/
[2] Arduino UNO, opis, dostupno na: www.arduino.cc
[3] Arduino CNC Shield V3 i A4988, opis, dostupno na: osoyoo.com/2017/04/07/arduino-uno-cnc-shield-v3-0-a4988/
[4] Step motor, opis, dostupno na: https://learn.adafruit.com/all-about-stepper-motors
[5] Servo motor SG90, datasheet, dostupno na: www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_data sheet.pdf
[6] „*Benbox*“ and file Hex, program, dostupno na: www.mediafire.com/file/hb22rv2rcyn7vc7/file-CNC.rar

Primena OLED komponenti kao izvora svetlosti

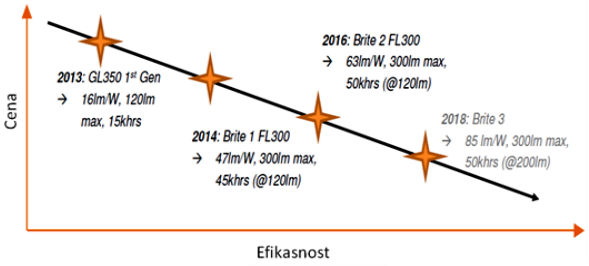
Sandra Veljković

Sadržaj - Stalni zahtevi za smanjenjem potrošnje energije, očuvanjem životne sredine, ali i poboljšanjem kvaliteta života dovele su OLED izvore svetlosti u žižu interesovanja, s obzirom da ne sadrže štetne materije i troše manje energije nego ostali izvori. Uz to, veoma bitna prednost ovih izvora svetlosti je što se svetlost emituje sa čitave površine izvora i što u manjoj meri sadrže svetlost kraćih talasnih dužina, koja može da bude štetna za ljudsko oko. Zato je dobro poznavanje principa rada, karakteristika, kao i mogućnosti prevazilaženja nedostataka OLED komponenata uopšte veoma bitno jer pruža potencijal za realizaciju inovativnih rešenja, ne samo u oblasti osvetljenja, već i u mnogim oblastima.

I. Uvod

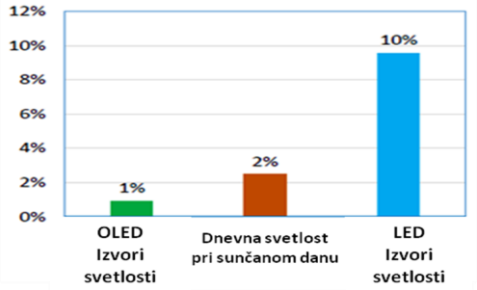
Razvoj različitih tehnologija, novih funkcionalnih materijala, elektronskih komponenata i drugih sistema doprinosi ubrzanom razvoju savremenog društva. Pri tome je imperativ korišćenje onih proizvoda koji doprinose očuvanju životne sredine, a pre svega uštedi energije. Tako nove tehnologije koje se koriste pri proizvodnji mobilnih telefona, tablet računara, drugih uređaja, kao i izvora svetlosti, a koje su bazirane na OLED (*Organic Light Emitting Diode*) tehnologiji postaju sve atraktivnije. Iako je ova tehnologija poznata više od pola veka, efikasnost ovih proizvoda je bila ograničavajući faktor za njihovu komercijalnu primenu.

Međutim, značajni napredak u proizvodnji i sve većoj efikasnosti OLED komponenata uopšte, pa i izvora svetlosti se konstantno ubrzava. Sa Sl. 1 se može videti da se kod komercijalnih izvora svetlosti u toku samo 3 godine efikasnost tri puta uvećala, kao i osvetljenost. Pored toga, može se uočiti da se radni vek tri puta uvećao, dok je cena tri puta smanjena [1].



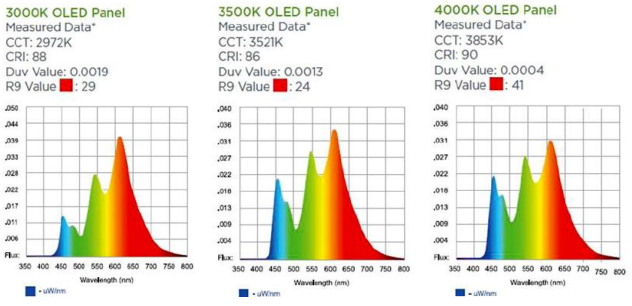
Slika 1. Tendencije u razvoju OLED izvora [1].

Efikasnost ovih izvora je veća i od 60 lm/W, a stepen reprodukcije boje (*color rendering index* - CRI) je veći od 90 [1]. Pored toga što imaju odličan stepen reprodukcije boje, sadrže u svom spektru mnogo manje „štetnog plavog svetla“ nego drugi izvori svetlosti (Sl. 2) [2].



Slika 2. Procenat udela "štetnog plavog svetla" u izvorima svetlosti [2].

Pod „štetnim plavim svetlom“ se podrazumeva svetlost kraćih talasnih dužina (λ), a time i veće energije $E = h\nu = hc/\lambda$, gde je h Plankova konstanta, ν frekvencija a c brzina svetlosti. Svetlost ovih talasnih dužina sadrži i Sunčeva svetlost, veštački izvori svetlosti, osvetljenje sa ekrana kompjutera, televizora, smartfona… Ona može dovesti do degradacije i oštećenja oka ukoliko je ono dugo izloženo ovom svetlu, ali i do poremećaja sna. Zato su OLED izvori koji sadrže mali udeo svetlosti manjih talasnih dužina i pogodni za korišćenje, kao što se vidi na osnovu njihovog spektra (Sl. 3). Na slici se vide spektri izvora svetlosti temperature boje od 3000, 3500 i 4000 K, a merenja su vršena pri temperaturi ambijenta od 25 °C [3].



(a) (b) (c)

Slika 3. Spektri OLED izvora svetlosti temperature boje od (a) 3000, (b) 3500 i (c) 4000 K [3].

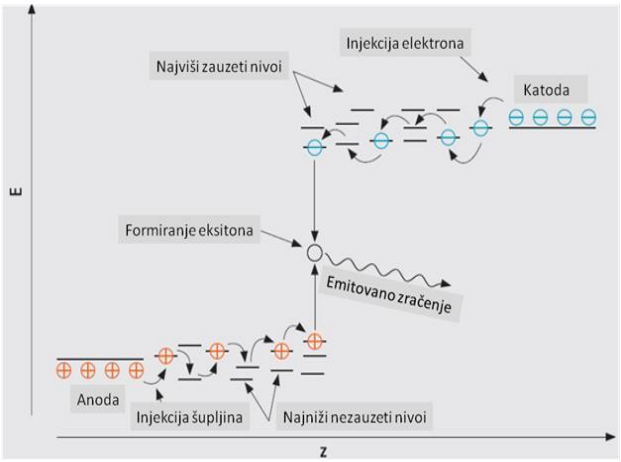
II. Princip rada OLED komponenata

U strukturi OLED-a se obično nalazi više slojeva organskog materijala čija je debljina u opsegu od 100 do 500 nm [4]-[6]. Svaki piksel OLED-a ima pet najbitnijih

Sandra Veljković je student master studija na modulu Elektronika i mikrosistemi na Katerdri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sandra.veljkovic@elfak.rs

elemenata, tj. slojeva. Prvi element predstavlja anodu koja je providna i njena uloga je da privlači elektrone prilikom proticanja struje kroz uređaj. Iznad anode se nalazi provodni sloj koji je izrađen od plastike organskog porekla, a ima ulogu da omogući kretanje šupljina od anode. Na ovaj sloj naleže emisioni koji je i najbitniji jer u njemu praktično nastaje svetlost. Ima ulogu da omogući kretanje elektrona od katode ka anodi, a formira se od plastike organskog porekla, mada drugačije od one koja se koristi za provodni sloj. Preko emisionog sloja se nalazi katoda čija je uloga da emituje elektrone prilikom proticanja struje kroz matricu. Supstrat na kome se svi ovi slojevi nalaze ima zaštitnu ulogu [7], [8].

Princip rada, kao i transport (*hopping transport*) elektrona i šupljina unutar OLED komponente je šematski prikazan na Sl. 4.

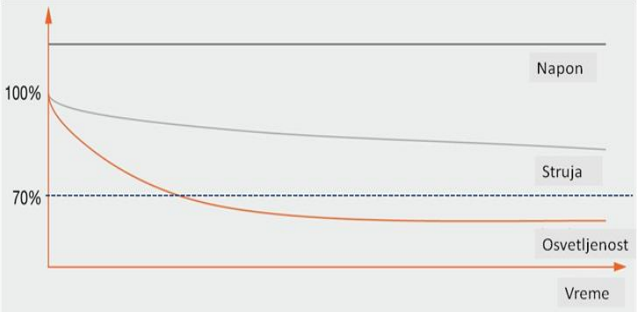


Slika 4. Šema principa rada i transport elektrona i šupljina unutar OLED komponente [7].

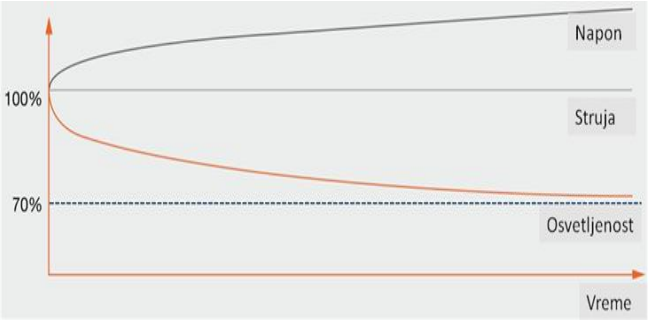
U zavisnosti od tipa matrice razlikuje se OLED sa pasivnom matricom (*passive-matrix*) i sa aktivnom matricom (*active-matrix*). Osim navedene podele, OLED se na osnovu namene i različite oblasti primene može podeliti i na providni OLED, *top-emitting* OLED, savitljivi - *flexible* OLED i beli OLED. Postoje i fosforescentni, nagomilani, modelirani OLED, polimerne svetlosno emitujuće diode (PLED), kao i svetlosno-emitujući polimeri (LEP) [8].

Rad OLED komponenata (koje se koriste za svetlosne izvore) sa elektronskim drajverima je po mnogo čemu sličan radu LED-a jer u mnogim primenama standardni drajveri za LED mogu da se koriste i za OLED [9]. Međutim, postoje veoma važne razlike i specifičnosti koje se moraju uzeti u obzir.

OLED treba da rade sa drajverima koji obezbeđuju konstantnu struju (*constant current drivers*). Naime, tokom rada OLED komponenata osvetljenost opada, a statička otpornost raste usled degradacionih procesa. Ukoliko rade sa konstantnim naponom javlja se porast statičke otpornosti i dolazi do opadanja struje tokom vremena (perioda eksploatacije). Usled toga dolazi do dodatnog opadanja osvetljenosti, koja pada ispod 70% početne vrednosti znatno ranije nego kada bi struja bila konstantna (Sl. 5). Zbog toga nije preporučljiv rad sa konstantnim naponom jer dovodi do smanjenja radnog veka. S druge strane, ukoliko OLED radi sa konstantnom strujom tokom celokupnog radnog veka dolaziće do porasta napona usled porasta statičke otpornosti (takozvano naponsko starenje - *voltage ageing*). Tada do opadanja osvetljenosti dolazi znatno kasnije nego pri radu sa konstantnim naponom, kao što se može videti na Sl. 6. Na taj način se radni vek znatno produžuje [9].



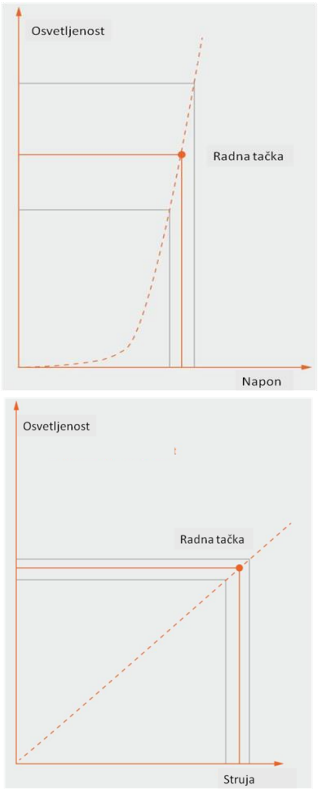
Slika 5. Opadanje struje i osvetljenosti pri radu sa konstantnim naponom [9].



Slika 6. Porast napona i sporije opadanje osvetljenosti pri radu sa konstantnom strujom [9].

Takođe je bitna karakteristika da je svetlosni fluks OLED komponenata proporcionalan struji, ali ne i radnom naponu (Sl. 7). Zavisnost osvetljenosti od napona je veoma strma u oblasti radne tačke, što znači da će i male promene radnog napona dovoditi do velikih promena osvetljenosti (Sl. 7a). S druge strane male promene radne struje ili male promene karakteristika (usled temperature ili tolerancije pri proizvodnji-izradi) dovode do malih promena osvetljenosti (Sl. 7b). Jasno je da je rad u strujno kontrolisanom modu znatno bolje rešenje za OLED komponente [9].

Sve ovo ukazuje da rad pri konstantnom naponu obezbeđuje maksimalni radni vek OLED komponenti i stabilnu osvetljenost u širokom opsegu promena temperature, tolerancije pri izradi, tolerancije drajvera kao i pri starenju.



Slika 7. Zavisnost osvetljenosti od (a) napona i (b) struje [9].

Da bi se obezbedio maksimalni radni (životni) vek veoma je važno i pri radu izabrati optimalnu radnu struju koja ne treba da ima veća odstupanja. Nominalna struja se bira iz tehničke dokumentacije i podešava se preko DC amplitude drajvera. Ukoliko se nominalna vrednost DC struje znatno prevazilazi može doći do značajnog smanjenja radnog veka, čak iako je srednja vrednost bliska nominalnoj. Da bi radni vek bio optimalan ne treba da odstupanja budu veća od 15% od srednje vrednosti DC struje.

Bitno povećanje radnog veka omogućilo je sve značajnije primene OLED komponenata u mnogim oblastima, kao što je proizvodna displeja gde je ova tehnologija najzastupljenija, ali i u mnogim drugim. Posebno postaje intenzivna primena i u automobilskoj industriji gde se ne koriste samo kao izvori svetlosti manjih ili većih površina već i za kontrolnu tablu, kao i za signalizaciju. Pored toga, naročito atraktivna rešenja u oblasti osvetljenja su postignuta upravo zahvaljujući primeni ovih komponenata koje sadrže organske materijale [10].

III. PRIMENA OLED IZVORA SVETLOSTI

Kada je reč o primeni u oblasti osvetljenja, posebno je značajna primena belog OLED-a, koji ima veliku energetsku efikasnost. Njegova masovnija primena za rasvetu bi značajno smanjila troškove za potrošnju električne energije. Pored toga što beli OLED ima mogućnost izrade na velikim panelima, svetlost koju emituje je znatno uniformnija, za razliku od LED izvora koji predstavljaju tačkaste izvore, tako da je mnogo prijatnija za oko i sličnija prirodnom svetlu a izbegava se i neprijatno blještanje. Navedene razlike se mogu uočiti pri upoređenju lampi u koje su ugrađeni LED i OLED izvori svetlosti, kao što je prikazano na Sl. 8.



(a) (b)

Slika 8. Stone lampe u koje su ugrađeni (a) LED i (b) OLED izvori svetlosti [11].

Na Sl. 8a je prikazana lampa napravljena od fleksibilne silikonske površine koja sadrži sedam ugrađenih polja sa LED diodama (6x9). Celo telo ove lampe može da se savija da bi se dobio najbolji položaj i da svetlost bude najbolje usmerena. Takođe u fleksibilnu silikonsku površinu može da se inkorporira beli OLED, kao što je prikazano na Sl. 8b. Ovakva silikonska osnova praktično može da se prilagodi bilo kom zahtevu i površini (postavi ili uvije oko nekog dela nameštaja) uz istovremeno obezbeđivanje prijatnog svetla za čitanje. Pri tome, inkorporirana svetleća površina može da se izvije i u smeru koji najviše odgovara za čitanje ili rad.

Fleksibilnost OLED izvora je omogućila formiranje lampi izuzetno atraktivnog dizajna. Lampa koja je inspirisana izgledom zmije kobre je prikazana je na Sl. 9a. Osnova je od aluminijuma i uspešno je formirana veoma istanjena forma koja ne zahteva upotrebu hladnjaka [12]. Takođe se mogu koristiti i materijali, kao što je drvo, koji do skora nisu mogli da budu u neposrednom kontaktu sa izvorima svetlosti. Na Sl. 9b je prikazan primer kreiranja skulpturalnog oblika od drveta u koji je inkorporiran OLED [13].



(a) (b)

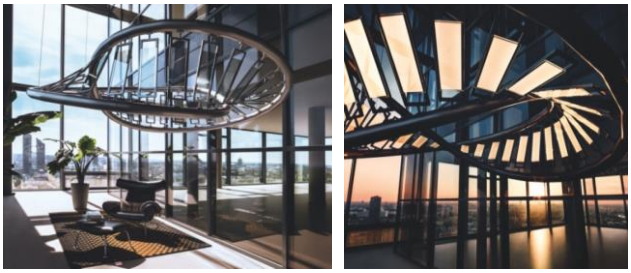
Slika 9. Dizajn lampe sa OLED izvorima (a) u obliku kobre [12] i (b) za kreiranja skulpturalnog oblika od drveta [13].

Veoma važna primena OLED-a je za osvetljenje ali i dekoraciju u enterijeru čime se prostoru daje drugačiji, atraktivniji izgled. Ovim izvorima može da se obezbedi prijatno, funkcionalno osvetljenje radnih prostora, ali i dinamičnost prostora opšte namene (Sl. 10).



Slika 10. Primena OLED-a za opšte osvetljenje u enterijeru [1], [14].

Transparentni OLED se može da koristi i za veoma atraktivne svetlosne instalacije čije površine ne smanjuju prodor dnevne svetlosti. Na Sl. 11 je prikazana takva "rolerkoster" instalacija sa 30 panela. Ovi paneli imaju površinu od 116 cm² i transparentnost od 57%. Svetlosna efikasnost je 20 lm/W, što je znatno manje od već postignutih 60 lm/W kod visoko efikasnih OLED komponenata. Razlog za to leži u činjenici da je kod transparentnih OLED neophodno da obe elektrode budu transparentne, tj. da katoda ne može da bude od metala. Zbog toga što je provodnost transparentnih elektroda manja od onih od metala i efikasnost je manja.



(a) (b)

Slika 11. Svetlosne instalacije formirane od transparentnih panela kada su (a) isključeni i (b) uključeni [12].

Primena transparentnog OLED-a je omogućila da praktično i prozori mogu da predstavljaju izvore i veštačke svetlosti. Pri tome transparentne površine prozora nisu od stakla, već od plastike. Kada postoji prirodna svetlost, ta površina je nesmetano propušta, a kasnije može da postane i sama izvor svetlosti.

Jasno je da je potencijal OLED komponenata izuzetno veliki, pa su istraživanja veoma intenzivna, pre svega zato što je uočeno da mogu imati primenu u različitim oblastima industrije, a naročito u veoma moćnoj, automobilskoj. Osim toga, pored primene za ekrane mobilnih telefona, ili digitalnih fotoaparata, veoma važna primena može biti za reklamne panoe, ali i za razne druge kućne i kancelarijske aparate. Zato je razvijanje ove tehnologije vrlo intenzivno i postoje izuzetni rezultati jer su mogućnosti primene gotovo neverovatne zahvaljujući kreativnosti inženjera i dizajnera.

IV. ZAKLJUČAK

Sve izraženiji su zahtevi da se primenom novih tehnologija kreiraju proizvodi koji će omogućiti smanjenje potrošnje energije i doprineti očuvanje životne sredine. Zato se sve više radi na razvoju uređaja koji za svoje funkcionisanje zahtevaju manje električne energije i ne sadrže štetne materije po okolinu. Ovom trendu može značajno da doprinese razvoj i dalje unapređenje OLED komponenata u različitim oblastima, a posebno u oblasti proizvodnje veštačkih izvora svetlosti. S obzirom da se svetlost OLED izvora emituje sa veće površine ova veštačka svetlost je sličnija dnevnoj i prijatnija je za ljudsko oko. Osim toga, stalno unapređenje karakteristika ovih izvora omogućava realizaciju do skora nezamislivih rešenja. Zbog toga je poznavanje prednosti OLED komponenata, ali i mogućnosti prevazilaženja nedostataka, veoma važno za dalju realizaciju kreativnih rešenja inženjera i dizajnera.

LITERATURA

[1] https://www.oledworks.com/wp-content/uploads/2017/05/OLED-Lighting-Berlin-Masterclass.pdf

[2] https://www.oledworks.com/wp-content/uploads/2017/10/OLED-World-Summit-John-Hamer-V24-2018-09-21-FINAL.pdf

[3] https://www.energy.gov/sites/prod/files/2016/06/f33/ssl_oled-products_2016.pdf

[4] D. Ban. S. Han, Z.H. Lu, T. Oogarah, A.J. SpringThorpe, and H.C. Liu, "Near-infrared to Visible Light Optical Upconversion by Direct Tandem Integration of Organic Light-emitting Diode and Inorganic Photodetector", *Appl. Phys. Lett*., 2007, vol. 90, pp. 093108-093110.

[5] S. Han, Y. Yuan, and Z.H. Lu, "Highly Efficient Organic Light-emitting Diodes with Metal/Fullerene Anode", *J. Appl. Phys.*, 2006, vol. 100, pp. 074504-074508.

[6] S. Han, X. Feng, Z.H. Lu, R. Wood, and D. Johnson, "Transparent-cathode for Top-emission Organic Light-emitting Diodes", Canadian Patent: CA2412379; US10/535893

[7] www.osram-oled.com 335716_Introduction to OLED technology.pdf

[8] Vesna Paunović, "OLED komponente", predavanja iz predmeta Tehnologije organskih poluprovodničkih materijala i komponenti, 2019.

[9] www.osram-oled.com 503914_OLED driving guide (GB).pdf

[10] S. Veljković, "OLED komponente i njihova primena", IEEESTEC 2016, pp. 273-276, 2016.

[11] http://www.systemdesignstudio.com

[12] https://www.osram-oled.com/application-general-lighting/index.jsp

[13] http://www.mojenterijer.rs/osvetljenje/igra-drveta-i-svetlosti-

[14] http://www.ofdesign.net/interior-design/kinetic-oled-lamp-design-floating-silently-in-space-2503

Alarmni sistem protiv krađe košnica

Ognjen Popović

Sadržaj – U radu je predstavljen alarmni sistem koji služi za sprečavanje krađe košnica ili drugih objekata koji su fizički udaljeni od vlasnika. Princip rada baziran je na detekciji pomeranja košnice, odnosno promeni ubrzanja senzora koji se fizički nalazi u košnici. Kada dođe do pomeranja košnice, alarmni sistem detektuje ubrzanje putem senzora akceleracije i preko GSM modula SMS porukom vlasnika obaveštava da je došlo do pomeranja košnice. Predstavljeni alarmni sistem spada u grupu tihih alarma.

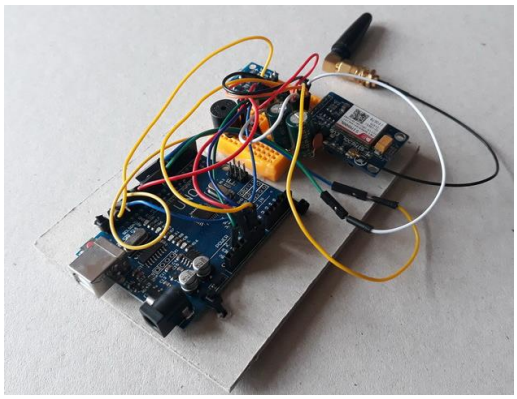
I. Uvod

Tokom trajanja vegetacije, pčelari sele košnice na mesta gde ima paše, odnosno gde se određene biljne vrste nalaze u fazi cvetanja. Korist od selidbe košnica imaju i vlasnici poljoprivrednih useva i pčela, odnosno pčelari. Međutim, mesta na kojima ima paše su često udaljena od vlasnika parcela i vlasnika košnica, odnosno od urbanih sredina. Samim tim košnice postaju primamljiv objekat za krađu.

Svrha ovde predstavljenog alarmnog sistema ima za cilj sprečavanje krađe košnica a zbog svojih osobina može poslužiti i za zaštitu drugih udaljenih objekata.

II. Princip rada i praktična realizacija

Princip rada alarmnog sistema (slika 1) baziran je na senzoru akceleracije GY-521 koji meri g silu i akceleraciju u tri ose. Mereći akceleraciju senzora, odnosno košnice mikrokontroler Arduino UNO prepoznaje da je došlo do pomeranja košnice i putem GSM modula SIM800L obaveštava korisnika da je detektovano pomeranje košnice.



Slika 1. Izgled uređaja.

Ognjen Popović, učenik Tehničke škole u Loznici, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, e-mail: ognjenpopovic1326@gmail.com

A. Hardver

Hardver uređaja čine mikrokontroler Arduino UNO, senzor akceleracije GY-521 (slika 2) i GSM modul SIM800L (slika 3). Za detekciju pomeranja košnice korišćen je senzor GY-521 koji putem I^2C interfejsa mikrokontroleru Arduino UNO šalje informacije o izmerenim vrednostima ubrzanja i g sile u sve tri prostorne ose. Komunikacija sa GSM modulom ostvarena je preko serijskog interfejsa.



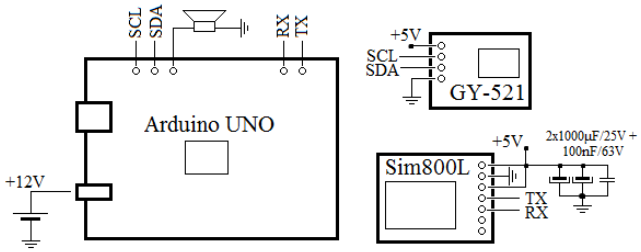
Slika 2. Senzor akceleracije GY-521.

Napajanje alarmnog sistema izvedeno je sa akumulatorom napona 12 V. GSM modul i senzor akceleracije napajaju se sa +5 V koji se dobijaju preko stabilizatora napona ugrađenog na ploči mikrokontrolera. S obzirom da GSM modul u trenucima komunikacije sa baznom stanicom i u uslovima slabog GSM signala može iz izvora da povuče kratkotrajne struje amplitude do 2 A, za stabilizaciju napona napajanja GSM modula u kolo dodata su dva paralelno vezana kondenzatora vrednosti 1000 µF/25 V i jedan keramički kondenzator od 100 nF/63 V (slika 4).

Za testiranje rada uređaja privremeno je korišćen aktivni piezo buzer koji je vezan na pin 13 gde su inače vezane i signalne diode mikrokontrolera Arduino Uno.



Slika 3. GSM modul SIM800L.



Slika 4. Šema povezivanja GY-521 I SIM 800l sa Arduino UNO.

B. Softver

Programiranje mikrokontrolera Arduino UNO izvedeno je u standardnom okruženju Arduino mikrokontrolera. Program ostvaruje stalnu komunikaciju sa senzorom akceleracije i proverava da li je izmerena vrednost akceleracije veća od određenog iznosa. Ukoliko je to slučaj mikrokontroler šalje GSM modulu unapred pripremljenu poruku koja će biti poslata vlasniku. Na ovaj način vlasnik se obaveštava o pomeranju košnice.

III. ZAKLJUČAK

Alarmni sistem predstavlja jednostavan i jeftin sistem za obaveštavanje vlasnika o pomeranju košnica. Uređaj se ne nalazi u samoj košnici, već u delu košnice, ispod poklopca, gde nema pčela tako da alarmni sistem nema uticaja na aktivnosti i funkcionisanje pčela. Primena izloženog alarmnog sistema moguća je na svim mestima gde postoji GSM signal. Mana ovog alarmnog sistema je ograničeno vremensko napajanje s obzirom na ograničen strujni kapacitet akumulatora. Upotreba solarnih izvora mogla bi produžiti autonomnost napajanja.

U toku testiranja uređaja izmerena je stand by struja od 40 mA. Budući da se na GSM modulu, senzoru akceleracije i mikrokontroleru nalaze led diode koje služe za signalizaciju napajanja odstranjivanjem led dioda mogla bi se dodatno smanjiti struja potrošnje uređaja, a samim time produžiti vreme autonomnosti rada.

Vrednost koja se u programu upoređuje sa izmerenom vrednosti ubrzanja utvrđena je empirijski.

ZAHVALNICA

Zahvalnost za realizaciju ovog istraživačkog projekta upućujem Tehničkoj školi Loznica, mentoru, profesoru Tomislavu Mijiću na nesebičnoj pomoći i datim savetima koji su mi pomogli u izradi ovog projekta. Takođe, upućujem zahvalnost profesorki Sanji Stanojević na strpljenju i razumevanju zbog mog odsustvovanja sa nastave radi izrade ovog uređaja.

LITERATURA

[1] GY-521 https://www.youtube.com/watch?v=wTfSfhjhAU0&t=40s
[2] Arduino UNO https://www.arduino.cc/en/Main/Tutorials
[3] SIM 800L V2 https://www.youtube.com/channel/UCMTerZA09j--fS1RYysYNdQ
[4] https://www.youtube.com/watch?v=-okAX7ZoGDk&t=1s

Automatizacija domaćinstva

Anđela Kocić

Sažetak - Život savremenog čoveka se mnogo razlikuje od života koji su nekada vodili njegovi preci. To je posledica društvenih i ekonomskih promena, ali u najvećoj meri tehnoloških. Modernom čoveku život je nezamisliv bez tehnoloških proizvoda. Cilj ovog rada je realizacija sistema koji će pozitivno uticati na čovekovu svakodnevnicu, olakšati mu je i omogućiti da repetitivne radnje prepusti tehnologiji.

I. Uvod

Automatizovana kuća se može opisati kao jedan sistem kod kojeg se podrazumeva integracija raznih električnih uređaja u domaćinstvu u jedinstveni sistem kojim se upravlja. To objekat nad kojim imamo apsolutnu mogućnost upravljanja, podešavanja, prilagođavanja i kontrole prema specifičnim željama i potrebama korisnika. Pomoću mikrokontrolera u domaćinstvu se može upravljati baš svime, npr. grejanjem, klimatizacijom, multimedijom itd. Upravljanje se može vršiti preko LCD-a, računara, mobilnih telefona, IR daljinskim upravljačima i slično [1].

II. Princip rada sistema automatizacije domaćinstva

Sistem se sastoji od dva podsistema koji upravljaju različitim oblastima domaćinstva:

- Zaštita od kiše i kontrola rasvete
- Klima, regulacija vazduha i zaštita od požara

Podsistem automatizovanog osvetljenja se odnosi na kontrolisanje rasvete u cilju obezbeđivanja prijatnije atmosfere u domaćinstvu, upozorenja o vremenskim uslovima i, naravno, uštede energije. Ovaj podsistem sadrži senzor kiše koji ima analogni izlaz te se može zaključiti koliko je obilna kiša, ukoliko pada. Čim se detektuje kiša, korisnik se obaveštava uključivanjem plavih dioda koje su postavljene po obodima prozora. Cilj ovoga je podsticanje korisnika da zaštiti stvari sa terase ili iz dvorišta koje nisu vodootporne i koje ne bi trebalo da budu izložene vodi (kao npr. veš koji se suši). Osim uključivanja plavih dioda, proverava se da li su prozori zatvoreni. Ukoliko nisu, zatvaraju se uz pomoć servo motora. Pored senzora kiše, tu je RTC (*Real Time Clock*) koji daje tačno vreme kako bi kontoler mogao da manipuliše rasvetom u zavisnosti od doba dana. Konačno, tu je i PIR (*Passive Inftrared*) senzor koji detektuje kretanje i uz pomoć koga ovaj posistem odlučuje da isključi svetlo u prostoriji ukoliko niko ne boravi tu više od 10 minuta. Ovo se odnosi na prostorije koje se ne posećuju često i u kojima se ukućani obično kratko zadržavaju, a pritom često zaborave da isključe svetlo: garaže, podrumi, kupatila...

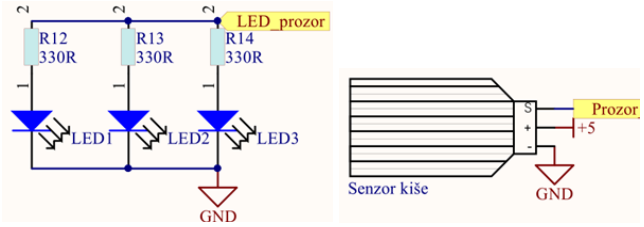
Podsistem klime i zaštite od požara se odnosi na kontrolisanje temperature, svežine i zagušljivosti vazduha, i upozorenja u sličaju prisustva plamena. Sastoji se od senzora ugljen-monoksida, senzora temperature i plamena. Mikrokontroler može znati kada bi trebalo uključiti ventilaciju, pa nikada neće biti zagušljivo. Jedan neizbežan senzor u domaćinstvu je svakako senzor plamena. Ukoliko se detektuje plamen, uključiće se alarm i sva svetla u kući da bi se probudili ukućani koji se mogu naći u opasnosti. Osim bezbednosti, sistem sam reguliše grejanje ukoliko temperatura padne ispod željene. Na kraju, omogućeno je isključivanje grejanja/klime kada su prozori otvoreni duže od 10 minuta da bi se sprečilo rasipanje toplog/hladnog vazduha.

Anđela Kocić je sa katedre za Mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: andejla.kocic@elfak.rs

III. Zaštita od kiše i kontrola rasvete

A. Senzor kiše i plave LED

Za detekciju padavina iskorišćen je analogni senzor nivoa vode, koji položen u horizontalni položaj, kao što je to u ovom slučaju, radi kao detektor kiše. To je analogni senzor koji radi po principu otpornosti paralelnih provodnih linija koje su izložene tečnosti. Dok linije nisu kratkospojene senzor ima otpornost od oko 1 MΩ. Kada senzor dođe u kontakt sa vodom, dolazi do kratkog spoja što je posledica provodnosti vode. U zavisnosti od jačine padavina, plave LE diode će svetleti određenim intenzitetom i time poslati informaciju korisniku da treba da proveri da li je potrebno zaštititi predmete koji su ostali na otvorenom.



Slika 1. Električna šema dela podsistema za zaštitu od kiše.

B. Detekcija položaja prozora i zatvaranje

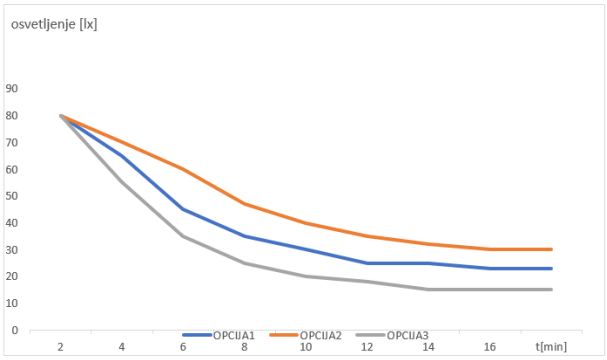
Položaj prozora je informacija od značaja kada su u pitanju padavine. Ukoliko korisnik nije kod kuće ili propusti obaveštenje plavih LE dioda o padavinama, ovaj sistem automatizacije ima način da bar zaštiti unutrašnjost time što će sam zatvoriti prozore bez ljudske asistencije. S obzirom da je veoma važno da budemo sigurni u kom su položaju

prozori ovde je tom zadatku pristupljeno iz dva ugla. Prvi je detekcija uz pomoć tastera a drugi uz pomoć Holovog senzora i magneta.

Servo motor služi za zatvaranje prozora. Poseduje tri priključka, jedan upravljački i dva preko kojih se motoru dovodi napajanje. Upravljački signal se koristi za upravljane ugaonom pozicijom izlazne osovine i najčešće je to PWM (*Pulse-Width Modulation*) signal. Trajanje impulsa PWM signala određuje poziciju osovine. Dok god je upravljački signal prisutan, servo motor zadržava poziciju svoje izlazne osovine. Kraći impulsi okreću osovinu u smeru kazaljke na satu, a duži suprotno. Koristeći Arduino biblioteku za upravljanje servo motorom, deluje se na prozor u smeru zatvaranja sve dok taster i Holov senzor ne potvrde da je prozor zaista zatvoren.

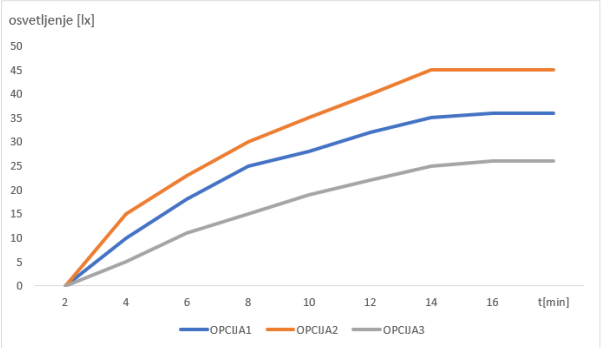
C. Prigušenje i postepeno uključivanje svetla

Svetlost se meša sa našim bioritmom, unutrašnjim časovnikom koji našem telu govori kada će spavati, kada da se budi, a kada da jede. Zbog toga, jedan deo upravljanja rasvetom automatskog domaćinstva posvećen je svetlu koje priprema telo za spavanje i postepenim uključivanjem svetala posle 23:00h da bi se izbeglo zaslepljivanje. Da li će se svetlo uključiti punim sjajem ili biti pigušeno zavisi od toga koliko je sati. Tu informaciju mikrokontroler uzima iz RTC modula. Koliko prigušiti svetlo prepušteno je izboru korisnika koji može odabrati jednu od tri opcije prigušenja koje se razlikuju po brzini i intenzitetu prigušenja.



Slika 2. Opcije brzine prigušivanja svetla posle 23:00h.

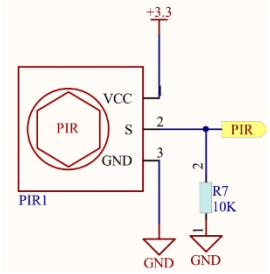
Kada se oči naviknu na mrak iznenadna promena jačine osvetljenje može biti neprijatna, pa čak i bolna. To se neretko dešava prilikom noćnih poseta toaletu/kupatilu. Da bi se ta neprijatnost izbegla, podsistem za osvetljenje reguliše brzinu uključivanja svetla kako bi korisniku dao vremena da se navikne. Na grafiku se mogu videti tri opcije koje predstavljaju različite brzine uključivanja svetla i krajnjeg intenziteta koje može odabrati korisnik. Dakle, nije dovoljno samo prigušiti svetlo u kupatilu posle 23:00h, već je veoma bitno detektovati kada se svetlo uključilo i od tog trenutka početi prigušenje. Svelo može biti uključeno mnogo posle 23:00h, ipak ne treba se naglo uključiti do prigušenog nivoa već postepeno, dozvoljavajući očima da se priviknu na promenu. Zbog toga je bilo neophodno korišćenje interrupta kojim se detektuje promena na prekidaču u kupatilu.



Slika 3. Opcije brzine uključivanje svetla posle 23:00h.

D. PIR senzor kretanja i automatsko gašenje svetla

U prostorijama u kojima se ne boravi često i ne zadržava dugo, zaboravljeno uključeno svetlo može biti nepotreban energetski izdatak (npr. garaža, podrum, ostava..). Automatizovano gašenje svetla kada nije detektovan pokret duže od 10 minuta je baš iz tog razloga implementirano u ovom sistemu. Kada nema kretanja na S pinu senzora će biti nizak naponski nivo tj. logička nula. Zbog toga je izlaz povezan *pull-down* otpornikom. Kada senzor detektuje kretanje, na izlazu se pojavljuje logička jedinica. Svaki put kada senzor detektuje kretanje, brojač počinje da meri vreme. Ukoliko u narednih 10 minuta ne postoji kretanje u prostoriji, mikrokontroler će isključiti svetlo. U *datasheetu* PIR senzora koji je ovde korišćen navedeno je da je potrebno napajanje od +3.3 V. Sve ostale komponente ovog podsistema zahtevaju napajanje od +5 V, te je bilo neophodno regulisati napon za napajanje PIR-a.



Slika 4. Električna šema PIR senzora.

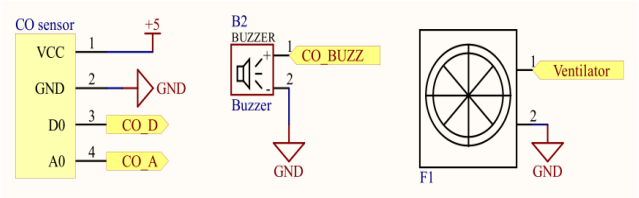
IV. KLIMA, REGULACIJA VAZDUHA I ZAŠTITA OD POŽARA

A. Regulacija zagušljivosti prostorije

Ugljen-monoksid je toksičan gas bez mirisa i boje. Zato što ga je nemoguće videti, okusiti ili pomirisati, može načiniti veliku štetu oranizmu pre nego što bude primećen. Efekti izlaganja ovom gasu variraju od osobe do osobe u

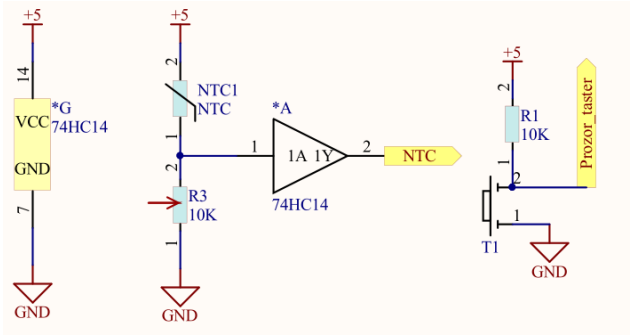
zavisnosti od godina starosti, zdravstvenog stanja, koncentracije i dužine izlaganja ugljen-monoksidu. Ovaj gas se, pored azbesta, nitrogen-dioksida, pesticida i dr. ubraja kao jedan od kriterijuma za kvalitet vazduha u zatvorenim prostorijama u kojima ljudi borave. Za detekciju ugljen-monoksida u ovom radu korišćen je modul za detekciju gasa koji je rasprostranjen za komercijalnu primenu. Može se pronaći u obliku modula ili kao samostalni senzor. Razlika između modula i senzora je što u modulu postoje komparator i digitalni izlazni pinovi, te je pogodniji u slučaju merenja samo prisustva gasa, a ne tačne koncentracije.

Električna šema podsistema za regulaciju nivoa CO u domaćinstvu je na sledećoj slici. Pored senzora CO, tu je i ventilator koji se uključuje kada koncentracija gasa premaši dozvoljen nivo i piezo-zujalica koja se uključuje kao upozorenje.



Slika 5. Električna šema podsistema za regulaciju nivoa CO u domaćinstvu.

B. Regulacija temperature

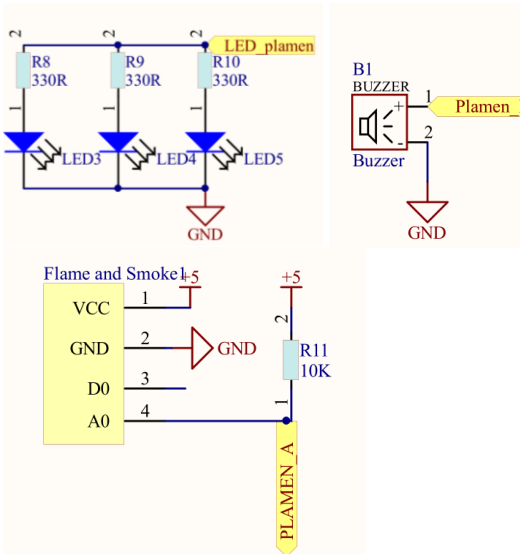


Slika 6. Električna šema podsistema za regulaciju temperature.

U prethodnom odeljku detaljno je objašnjeno zbog čega je bitno da temperatura prostorije bude konstantna i odgovarajuća. Da bi se to postiglo upotrebljen je NTC termistor i potenciometar u konfiguraciji naponskog razdelnika i 74HC14 [2] komponente koja u sebi ima operacione pojačavače u konfiguraciji Šmit-trigera. NTC termistor je tip otpornika čiji se električni otpor menja sa promenom temperature. Zbog te osobine, nalazi primenu kao senzor temperaturnih promena. Nažalost, zavisnost otpora od temperature je nelinearna, što otežava korišćenje termistora. Sa druge strane, osetljivost je velika i na male promene temperature. NTC znači da ima negativan temperaturni koeficijent i da sa porastom temperature, otpornost opada.

C. Zaštita od požara

Za detekciju plamena iskorišćen je *YS-17* senzor. Ovaj senzor je osetljiv na IR talase i detektuje prisustvo vatre ili drugog izvora infracrvene svetlosti talasne dužine od 0,76 μm do 1,1 μm. Prilikom postavljanja ovog senzora, mora se uzeti u obzir i njegova lokacija i razmišljati šta senzor zapravo „vidi“. To će u velikoj meri odrediti maksimalnu daljinu izvora IR svetlosti koju može detektovati. Usvojeno pravilo je da se senzor treba postaviti na visini dva puta većoj od najvišljeg objekta u prostoriji. Sa Slike 7. koja prikazuje električnu šemu ovog podsistema može se videti su, pored senzora, tu i LE diode i piezo-zujalica. Ukoliko se požar detektuje uključiće se sva svetla u domaćinstvu koja su predstavljena LE diodama i alarm koji se predstavljen zujalicom kako bi se ukućani probudili i zaštitili.

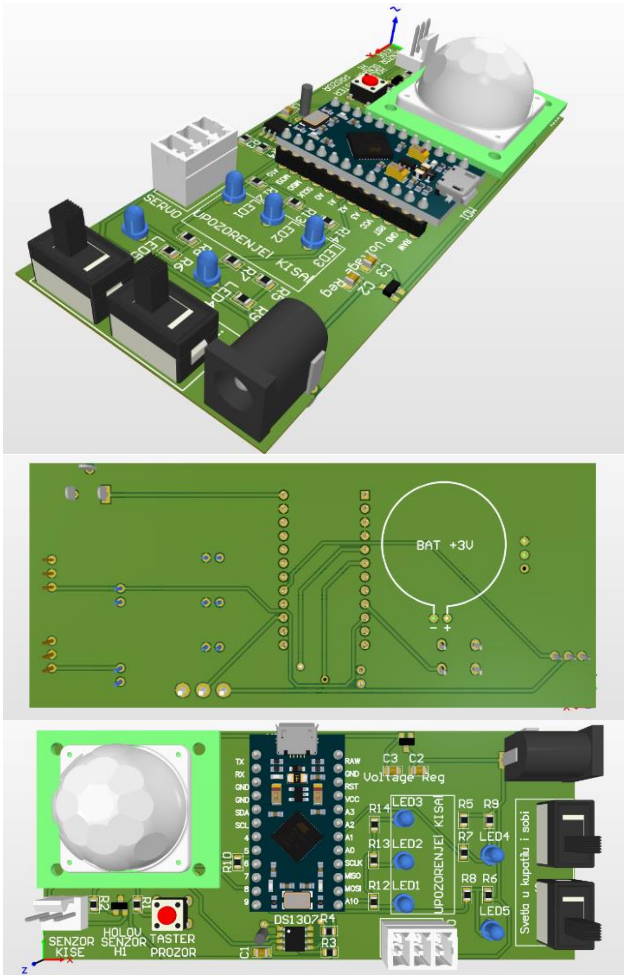


Slika 7. Električna šema podsistema za zaštitu od požara.

V. PROJEKTOVANJE PCB-A SISTEMA

Projektovanje je obavljeno u softverskom okruženju Altium Designer 18.1.1 [3]. Dva podsistema projektovana su kao dve zasebne štampane ploče. Većina komponenata koje su korišćene ne postoje u standardnim bibliotekama *Altium Designera*, pa je napravljena biblioteka sa novim komponentama. Neke komponente su pravljene potpuno nove (*symbol*, *footprint*, 3D model), a neke su korišćene iz standardnih biblioteka pa su im samo pridruženi *footprint* i 3D model tj. iskorišćen je već gotov simbol. U biblioteci se takođe nalaze informacije o svakoj iskorišćenoj komponenti, pa je po potrebi moguće izvući BOM (*Bill Of Materials*) fajl u kome je spisak svega potrebnog za fizičku izradu ovog PCB-a. Takođe je dostupan *Gerber* – standardni format koji se sastoji od 2D binarnih vektora a koristi se u industriji za tačno predstavljanje slojeva, maski i veza na PCB pločici. Pločice su projektovane u dva sloja. U gornjem i donjem

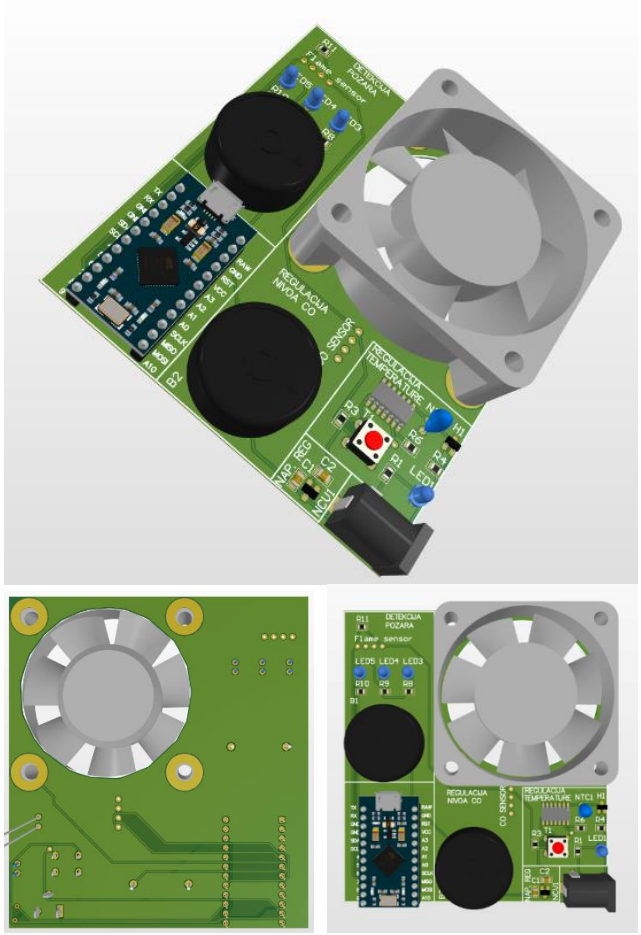
sloju razliven je poligon mase. Poligon podrazumeva veliku površinu metala koja je spojena sa napajanjem ili zajedničkom masom komponenata na PCB-u. Služi kao povratni put struje komponenata [4]. Poligon mase se obično pravi što je veći, zauzimajući sva mesta na pločici na kojima se ne nalaze veze ili komponente. U višeslojnim pločicama, često zauzima čitav jedan sloj tj. jednu celu površinu pločice. Razlog tome je da učini dizajnerov posao lakšim, da bi se bilo koja komponenta mogla povezati sa masom samo jednom vijom bez dodatnih žica. Na sledećim slikama može se videti izgled štampanih ploča ova dva podsistema.



Slika 8. 3D prikaz podsistema za zastitu od kiše i regulaciju rasvete.

VI. Zaključak

Poboljšanje kvaliteta života i efikasnosti je ono čemu čovek današnjice teži. Ušteda vremena je veoma važna, a bavljenje dosadnim poslovima koji zahtevaju ponavljanje iste radnje iznova i iznova sprečava pojedinca da to vreme iskoristi na kvalitetniji način. Ideja automatizacije je da rastereti i olakša život time što će na sebe preuzeti sve poslove koji čoveku mogu biti dosadni ili koji se lako i često zaborave. Računar će te poslove uraditi bolje i preciznije, a naravno i brže pritom ostavljajući korisniku više vremena za posao, slobodne aktivnosti ili odmor.



Slika 9. 3D prikaz PCB-a sistema za klimu, regulacija vazduha i zaštitu od požara iz ugla.

Zahvalnica

Autor se zahvaljuje prof. dr Danijelu Dankoviću na mentorstvu i podršci u toku pisanja ovog rada.

Literatura

[1] “A review of smart homes-past, present, and future” Alam, M.R, Alauddin, M.A. IEEE Trans. 2012
[2] Tehnička dokumentacija 74HC14, https:///74HC_HCT14.pdf
[3] Softversko okruženje Altium Designer, https://www.altium.com/altium-designer/
[4] „What is a PCB with a Ground plane?“, John Whitmore, sci.electronics, 199.

E - papir info panel sa malom potrošnjom energije i prikazom vremenske prognoze

Laza Kostić

Sažetak - U ovom radu su prikazani razni problemi i na koji način su oni prevaziđeni u realizaciji prototipa info panela sa malom potrošnjom energije. Potrošnja sistema je predstavljala najveći problem koji je rešen upotrebom modernih komponenata kao i E - papir ekrana koji su novi na potrošačkom tržištu, kao i raznim softverskim tehnikama.

Obezbeđen je radni vek info panela veći od 7 meseci na jednom punjenju baterije, što je veliki radni vek s obzirom na to da ekran bežično prihvata, i prikazuje nove informacije više puta dnevno. Svrha dugog radnog veka na jednom punjenju baterije je praktičnost korišćenja u realnom svetu kada bi se na osnovu ovog prototipa projektovao gotov proizvod za potrošačko tržište.

Prikaz vremenske prognoze je korišćen kao dokaz koncepta o info panelu koji ne mora konstantno da prihvata i prikazuje nove informacije, odnosno koji to radi periodično.

I. UVOD

U modernom svetu se često zaboravi da pametni uređaji bi trebali da koriste čoveku. To je delom zbog velike brige o tome da li uređaji imaju dovoljno napunjenu bateriju kako bi se mogli koristiti, pogotovo za pametne telefone koji se moraju puniti svakodnevno.

Da bi se napravio ovaj info panel neophodono je da se nekoliko celina uklope i rade zajedno. Koje celine, moduli i komponente su korišćene i na koji način su međusobno povezane će biti objašnjeno u sledećem poglavlju. Glavni cilj ovog info panela je da bude pregledan i čitak, kao i da troši izuzetno malo električne energije. Takođe trebalo bi da uređaj zahteva što manje održavanja od strane korisnika, odnosno da panel služi korisniku.

Zamisao za ovaj sistem je da se napravi modul sa E - papir ekranom i antenom, kako bi bežično primao informacije za prikaz, i takođe da izgleda poput rama za slike. Format poput rama za slike je pogodan, jer smo navikli na to da imamo uramljenu sliku na našem stolu ili zidu. Ne bi bio previše upadljiv, estetski bi bio privlačan, a opet bi imao i praktičnu vrednost. Osim praktičnih vrednosti rama za slike, modul bi trebalo da ispunjava i neke druge praktične vrednosti, a to su:

- Da na jednom punjenju baterije modul radi najmanje 6 meseci;
- Da je izuzetno jednostavan za korišćenje, odnosno da ima minimalne intervencije korisnika za pravilan rad panela.

Laza Kostić, student master akademskih studija na Elektronskom fakultetu, Univerzitet u Nišu, smer Elektronika i mikrosistemi, Aleksandra Medvedeva 14, 18000 Niš, E-mail: lazakostic@rocketmail.com

II. MODULI I KOMPONENTE

A. Ekran

Glavni deo modula je E - papir ekran. Ekran mora da bude čitiljiv sa veće daljine nego što je čitljiv recimo ekran prosečnog pametnog telefona. Znači najbolje bi bilo da ekran bude dimenzija nekog uobičajnog rama za slike.

Postoje svega dva prodavca E - papir ekrana koji prodaju ekrane na komad, oni su „*Good display*" i „*WaveShare*". Odlučeno je kupiti ekran od prodavca „*WaveShare*" jer pored samih panela ekrana, oni takođe prodaju i razvojna okruženja za ekrane, što je dosta pomoglo pri razvoju prototipa. Karakteristike ekrana[7] koji je kupljen za ovaj prototip su sledeće:

- Radni napon: 3,3 V
- Način komunikacije: SPI
- Dimenzije ekrana: 118,8 * 88,26 mm
- Debljina panela: 1 mm
- Rezolucija: 600 * 448 piksela
- Boje (trobojni ekran): crna, bela, žuta
- Brzina osvežavanja: 26 s
- Potrošnja pri osvežavanju: 26,4 mW (tipično)
- Ugao gledanja: > 170°.



Slika 1. E - papir panel sa razvojnim okruženjem.

Ispostavlja se da, razvojno okruženje za ekran (slika 1) nije ništa više nego štampana pločica sa konektorom za flet kabl ekrana i par pasivnih komponenata koje su neophodne za rad podizača napona, tako da je možda prikladniji naziv za ovu pločicu - adapter. Takođe poseduje i konektor pogodan za povezivanje sa mikrokontrolerom ili mikroprocesorom. Izvodi dostupni iz razvojnog okruženja (slika 1) su:

- VCC - Napajanje
- GND - Masa
- DIN - Ulazni podaci/komande za SPI komunikaciju
- CLK - Radni takt SPI komunikacije

- CS - Ulaz za naznačavanje ekranu da li se njemu obraćamo tokom komunikacije
- DC - Ulaz za naznačavanje toga da li šaljemo podatke ili komande preko SPI komunikacije
- RST - Ulaz za resetovanje ekrana
- BUSY - Izlaz za naznačavanje da li je ekran trenutno zauzet za komunikaciju.

Tehnička dokumentacija ekrana[2] je jako siromašna. Nigde se ne navodi ime drajvera ekrana, pa čak ni njegova unutrašnja blok šema. Tako da je najbliže pravoj tehničkoj dokumentaciji, tehnička dokumentacija drajvera E - papir ekrana drugog prodavca („*Good display*") za drugi ekran manjih dimenzija i rezolucije[3]. Ima dosta sličnosti i istih stvari u načinu komunikacije navedenom u toj tehničkoj dokumentaciji i demonstrativnom kodu za korišćeni ekran, koji je dostavio prodavac „*WaveShare*". Sličnosti se nastavljaju u načinu povezivanja flet kabla ekrana sa adapterom, tako da se može smatrati da je ova tehnička dokumentacija ekvivalentna u nekom opsegu sa drajverom na ovom ekranu.

Način rada drajvera, kako bi se postavio prikaz na ekranu je najlakše objasniti u koracima:

1. Uveze se prikaz (podatke) u drajver ekrana iz mikrokontrolera/mikroprocesora preko SPI komunikacije
2. Drajver skladišti te podatke o prikazu (boje svakog piksela) u memoriji
3. Drajver uporedi trenutnu boju svakog piksela (sve te podatke ima u memoriji) sa bojom u koju želimo da postavimo taj piksel (podaci koji su mu poslati u prethodnom koraku)
4. Na osnovu podataka iz tabele (eng. *LookUp table*) u memoriji, drajver zaključuje kojom kombinacijom signala na svakom pikselu ekrana može da dobije željenu boju koja mu je poslata (npr. ukoliko je bila crna boja piksela a sada želimo da bude bela postoji određena kombinacija impulsa koja se mora dovesti na taj piksel, a drugačija kombinacija ukoliko je bila žuta boja piksela prethodno; podaci o tim kombinacijama se čuvaju u tabeli u memoriji)
5. Drajver usklađuje rad okidnog kola (eng. T*imig controller*), kola podizača napona, drajvera za sors i gejt svakog MOSFET-a koji kontroliše pojedinačan piksel i memorije u kojoj se čuvaju podaci o boji svakog piksela i priprema se da postavi novi prikaz na ekranu
6. Drajver preko „BUSY" izvoda obaveštava da je trenutno zauzet i da ne može da prima nove informacije od mikrokontrolera/mikroprocesora
7. Drajver postavlja sledeći prikaz na ekranu
8. Nakon postavljenog prikaza stanje na izvodu „BUSY" se menja i drajver je ponovo spreman da prima nove podatke

Ukratko rečeno, drajver „zna" koji je trenutni prikaz na ekranu, Pa mu se „kaže" koji je sledeći željeni prikaz. On „odluči" koju kombinaciju signala da pošalje na koji piksel na osnovu informacija o trenutnom i budućem prikazu. Dalje, on uskladi rad podizača napona, rad drajvera za sors i gejt svakog tranzistora na ekranu, i rad bloka za kontrolu trajanja dužine svakog impulsa. Zajednički rad svih tih blokova je neophodan kako bi poslao tu kombinaciju signala koju je „naumio" za svaki piksel. Slanjem te kombinacije signala se menja boja svakog piksela na ekranu, i novi prikaz je postavljen.

B. Radio modul

Za bežičnu komunikaciju je korišćen već gotov modul prijemnika i predajnika za radio komunikaciju (slika 2). Oni rade na frekvenciji od 433MHz i na principu ASK (eng. *Amplitude Shift Keying*) modulacije, za slanje podataka.



Slika. 2 Moduli prijemnika (levo) i predajnika (desno) za radio komunikaciju.

Što se tiče izvoda modula prijemnika i predajnika, oni imaju po tri izvoda za povezivanje, a ti izvodi su:

- VCC - Izvod za ulazni napon u opsegu od: 3 - 12 V
- GND - Izvod za masu
- DATA - Izvod za podatke, u slučaju predajnika služi za slanje podataka, a u slučaju prijemnika služi za primanje podataka

ASK modulacija je podložna smetnjama i sam modul nema veliki domet. Uprkos ovim manama ovaj radio modul je iskorišćen zbog njegove cene, jednostavnosti implementacije i dovoljno dobrih performansi za prototip ovog projekta. Naravno, kada bi se ovaj info panel komercijalno proizvodio mnogo pouzdaniji i bolji radio modul bi se koristio. Pošto cilj ovog projekta nije radio komunikacija, nego samo njeno korišćenje za slanje informacija umesto fizičkog kabla, neće se zalaziti u način rada ovog modula, kao ni u kod neophodan za njegovo funkcionisanje. Modul za primanje informacija se nalazi na info panelu. Modul koji šalje informacije sa svojim mikrokontrolerom služi kao zamena za neki računar ili pametni uređaj koji šalje informacije za prikaz info panelu; kada bi info panel bio komercijalan proizvod.

C. Mikrokontroler

„Mozak" celog modula info panela je mikrokontroler. Njegova brzina, mala potrošnja energije i različite sposobnosti koje on poseduje su glavne karakteristike zašto se uklapa u ovaj prototip.



Slika 3. „Arduino Pro Mini" razvojno okruženje sa mikrokontrolerom „ATmega328P".

Korišćeno je „*Arduino Pro Mini*", razvojno okruženje sa mikrokontrolerom „*Atmel* ATmega328P"[6] (slika 3). Cena, dimenzije, i način programiranja razvojnog okruženja - preko USB-porta računara, su presudile u odluci odabira adekvatnog načina kontrolisanja prototipa ovog info panela. Neke od važnih karakteristika ovog mikrokontrolera su:

- Napon napajanja: 3,3 V
- Broj ulaznih i izlaznih digitalnih pinova: 14
- Namenski blok za SPI komunikaciju: Da
- Broj ulaznih analognih pinova: 6
- Maksimalna struja po digitalnom pinu: 40 mA
- Fleš memorija: 32 kB (30 kB dostupno korisniku)
- Radni takt: 8 MHz.

Digitalni pinovi su bitni za komunikaciju sa ekranom i radio modulom kao i za njihovo napajanje, a analogni pin se koristi za očitavanje napona na bateriji, o kojoj će biti više reči u sledećem delu.

D. Napajanje sistema; Sistem za upravljanje električnom energijom i baterija

Glavni kriterijum za ovaj info panel je dužina trajanja jednog punjenja baterije. Od pre nekoliko godina postoje komponente na tržištu koje to omogućavaju.

Potrošnja energije ekrana, kao i potrošnja radio modula je fiksna. Potrošnja energije mikrokontrolera se može malo korigovati, smanjenjem radnog takta ili isključivanjem unutrašnjeg analogno-digitalnog konvertora na primer. Ali suštinski postoji granica gde je potrošnja mikrokontrolera minimalna, a da mikrokontroler još uvek može da izvršava ono što je potrebno za ovaj projekat.

Jednostavnim proračunom dobija se koliki je kapacitet baterije koji je neophodan za to da ovaj sistem radi 6 meseci na jednom punjenju baterije. Podaci o potrošnji struje za napon napajanja od 3,3V koji su dobijeni empirijskim putem su:

- Potrošnja ekrana: 8 mA
- Potrošnja radio prijemnika: 3 mA
- Potrošnja mikrokontrolera: 10 mA.

Znači potrošnja celog sistema bi bila:

$$8\,\text{mA} + 3\,\text{mA} + 10\,\text{mA} = 21\,\text{mA}$$

Proračun potrošnje ovog sistema za 6 meseci rada je:

$$6 \text{ meseci} \cdot 30 \text{ dana} \cdot 24 \text{ časa} \cdot 21 \text{ mA} = 90\,720 \text{ mAh} \approx 90{,}7 \text{ Ah}$$

To bi značilo da je neophodna ogromna i teška baterija, nebitno o kojoj hemiji baterije se radi. Recimo, litijumska baterija tog kapaciteta (koja ima najmanju težinu za dati kapacitet u odnosu na ostale vrste baterija) bi težila preko 1,5 kg i zauzimala preko 300 ml zapremine!

Može se smanjiti potrošnja tako što će se samo u odabranim trenucima uključivati ekran i radio prijemnik pomoću digitalnih pinova mikrokontrolera, recimo svakog sata na po jedan minut. Znači novu informaciju bi panel mogao da primi samo u prvom minutu sata kada uključimo radio prijemnik, novi prikaz da se postavi u drugom minutu tog sata, i ostalih pedeset i osam minuta da samo radi mikrokontroler. Potrošnja bi bila sledeća:

$$6\,\text{meseci} \cdot 30\,\text{dana} \cdot 24 \cdot \left(\frac{1}{60}(3\,\text{mA} + 10\,\text{mA})\,\text{sati} + \frac{1}{60}(8\,\text{mA} + 10\,\text{mA})\,\text{sati} + \frac{58}{60}10\,\text{mA sati}\right) = 43\,991\,\text{mAh} \approx 44\,\text{Ah}$$

Potrošnja sistema je sada duplo manja od prethodnog slučaja, ali opet bi trebala glomazna baterija.

Glavni problem u ovoj situaciji je pedeset osam minuta kojih je uključen mikrokontroler. On za to vreme samo otkucava do sledećeg trenutka kada treba ponovo da primi informacije i prikaže ih na ekranu. Može se malo smanjiti njegova potrošnja raznim softverskim tehnikama, ali nikako ne bi bila za red ili dva reda veličine manja. Kada bi mogao da se napravi neki sistem koji troši izuzetno malo struje (< 0,1 mA), i da umesto mikrokontrolera otkucava do trenutka kada mora da se „radi" dobio bi se sistem koji troši vidno manje električne energije, pa bi trebala i manje glomazna baterija. Ukoliko se pretpostavi takav slučaj, potrošnja je:

$$6\,\text{meseci} \cdot 30\,\text{dana} \cdot 24 \cdot \left(\frac{1}{60}(3\,\text{mA} + 10\,\text{mA})\,\text{sati} + \frac{1}{60}(8\,\text{mA} + 10\,\text{mA})\,\text{sati} + \frac{58}{60}(\sim 0\,\text{mA})\,\text{sati}\right) = 2\,232\,\text{mAh} \approx 2{,}2\,\text{Ah}$$

Ovaj kapacitet baterije, za napon na kome radi sistem (3,3 V) je izvodljiv u praksi. Skoro u svim pametnim telefonima današnjice nalazi se litijumska baterija nominalnog napona 3,7 V i kapaciteta ~3 Ah. S obzirom na to da takav kapacitet baterije staje u telefon koji staje u naš džep, vrlo lako bi takva baterija stala u kućište dimenzija rama za slike

.Nakon prethodnog proračuna može se videti da je glavni cilj za ovaj prototip moguće postići, odnosno da sistem može da radi više od 6 meseci bez prestanka.

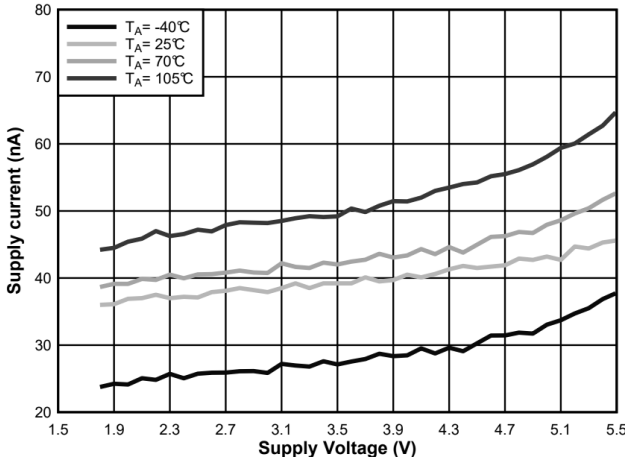
Ali moguće je samo ukoliko postoji sistem koji može da „probudi" mikrokontroler u jednakim vremenskim intervalima, a da pritom troši manje od 0,1 mA struje.

Ključ za ovakav sistem je komponenta pod imenom: „TPL5111 *Nano-Power System Timer for Power Gating*"[4] proizvođača „*Texas Instruments*" (slika 4).



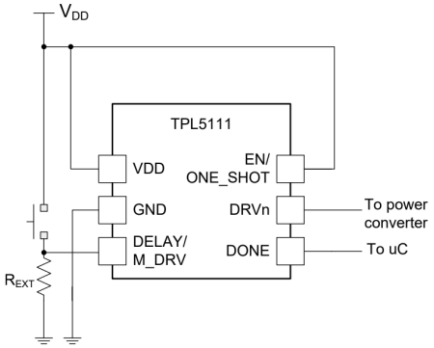
Slika 4. TPL5111 tajmer.

Ova komponenta ima izuzetno malu potrošnju električne energije koja se vidi na grafiku preuzetom iz tehničke dokumentacije ove komponente na slici 5.



Slika 5. Grafik potrošnje struje u zavisnosti od napona, TPL5111; Za temperaturu ambijenta 25°C potrošnja pri naponu od 3,3 V je ~38 nA.

Znači za konstantan rad pri naponu od 3,3 V potrošnja struje ove komponente je manja od 40 nA, to je 0,00004 mA. Kao što samo ime kaže TPL5111 je tajmer koji može da u redovnim vremenskim intervalima uključuje neki veći potrošač. Dužina vremenskog intervala se podešava jednim eksternim otpornikom. Kada istekne taj vremenski interval, TPL5111 na svom izlazu „propusti" napon svog napajanja i time signalizira većem potrošaču da se uključi. Kada je vreme da se ukine struja većem potrošaču, na ulaz TPL5111 se treba dovesti signal koji to nagoveštava, pa onda TPL5111 ukida svoj signal na izlazu, odnosno isključuje veći potrošač i ponovo započine odbrojavanje vremenskog intervala do sledećeg „buđenja". Šema opšte primene ove komponente je prikazana na slici 6 i preuzeta je iz tehničke dokumentacije.



Slika. 6 Primena TPL5111; REXT je otpornik kojim se određuje vremenski interval „buđenja", taster služi na manuelno „buđenje" tajmera pre isteka vremenskog intervala, DRVn izlaz služi za signalizaciju isteka vremenskog intervala i na njemu se javlja napon napajanja VDD, DONE ulaz služi za resetovanje tajmera.

Formula po kojoj se računa R_{EXT} je data u tehničkoj dokumentaciji (slika 7), ali su takođe date i vrednosti tog otpornika za česte vremenske intervale, tako da je za jedan sat vremenskog intervala data otpornost od 124,91 kΩ.

$$R_{EXT} = 100\left(\frac{-b+\sqrt{b^2-4a(c-100T)}}{2a}\right)$$

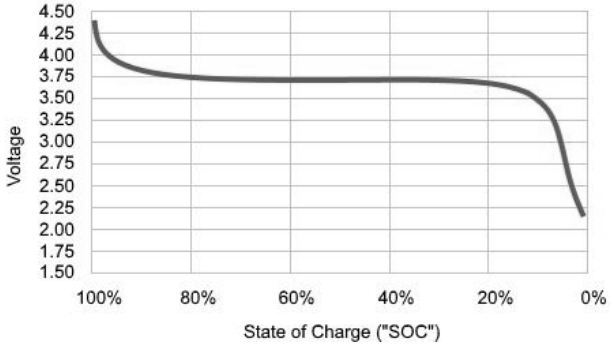
Za T [s]	a	b	c
1 < T ≤ 5	0.2253	-20.7654	570.5679
5 < T ≤ 10	-0.1284	46.9861	-2651.8889
10 < T ≤ 100	0.1972	-19.345	692.1201
100 < T ≤ 1000	0.2617	-56.2407	5957.7934
T > 1000	0.3177	-136.2571	34522.468

Slika 7. Formula za izračunavanje eksternog otpornika; T je vremenski interval [s], REXT je vrednost eksternog otpornika [Ω], a, b i c su koeficijenti iz tabele.

Za eksterni otpornik TPL5111-a za određivanje vremenskog intervala je korišćena redna veza dva otpornika jedan vrednosti 120 kΩ, a drugi 4,7 kΩ. Njihova ukupna vrednost je 124,7 kΩ. Kada se ova vrednost otpornika zameni u formuli datoj u tehničkoj dokumentaciji za izračunavanje vremenskog intervala, dobija se vremenski interval od 59 minuta i 46,36 sekundi. Navedeno je da ovaj vremenski interval treba biti 58 minuta. Radi olakšanja realizacije ovog prototipa u ovom slučaju se smatra da je korišćeni vremenski interval približan 58 minuta.

Veliki potrošač koji se uključuje je mikrokontroler. Ali mikrokontroler opet mora da troši neku energiju dok čeka da bude „probuđen". Trošio bi struju reda veličine bar desetinu µA, a možda i više, što je svakako mnogo bolje od 10 mA, ali i mnogo gore od TPL5111 koji troši 40 nA.

Napon litijumske baterije zavisi od njenog trenutnog kapaciteta. Taj napon je između 4,3 V i 2,2 V. TPL5111 radi na tim naponima, kao i ATmega328P. Radio prijemnik radi na naponima od 12 V do 3 V. Jedini deo ovog sistema koji ima fiksnu vrednost radnog napona je ekran. Ta vrednost napona je 3,3 V. Znači potreban je naponski regulator za 3,3 V kako bi se obezbedio rad ekrana. Mogao bi čak i da se stavi i mikrokontroler i radio prijemnik da rade na tom naponu, kako bi se obezbedila adekvatna komunikacija sa ekranom. Litijumska baterija se ne treba prazniti do manje od 20% njenog kapaciteta, a sa grafika na slici 8 vidi se zavisnost napona od kapaciteta baterije, i za 20% kapaciteta napon iznosi ~3,65 V.



Slika 8. Zavisnost napona litijumske baterije od kapaciteta.

Znači treba samo linearni ili prekidački naponski regulator koji će smanjiti napon sa 4,2-3,65 V na 3,3 V. Kada bi postojao naponski regulator na 3,3 V koji se može isključiti i uključiti TPL5111 tajmerom a da pritom u isključenom stanju troši manje od 1 μA, takav regulator bi se odlično uklopio u ovaj sistem kontrole potrošnje električne energije.

Naravno da takav naponski regulator postoji i zove se „TPS737xx *1-A Low-Dropout Regulator With Reverse Current Protection*"[5] (slika 9) od istog proizvođača kao i za TPL5111.



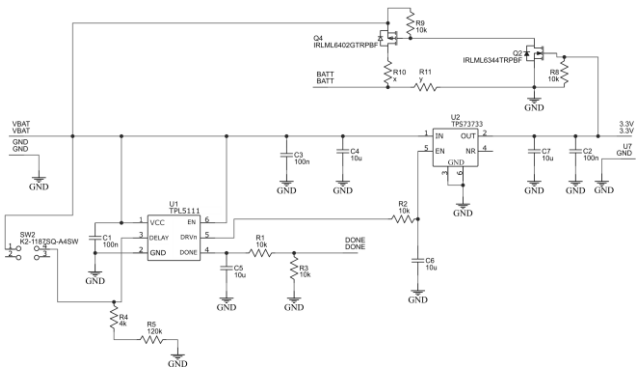
Slika 9. TPS737xx.

Podvrsta ovog regulatora je TPS73733 koji je fiksni naponski regulator na 3,3 V.

Njegove karakteriste od značaja za ovaj sistem su:

- Pad napona od ulaza do izlaza: < 130 mV (tipično za izlaznu struju od 1 A)
- Pin za uključivanje i isključivanje rada ovog regulatora
- Potrošnja regulatora kada je isključen i daje 0 V na izlazu: < 20 nA (tipično).

Šema ovog sistema za potrošnju električne energije sa TPL5111 tajmerom i TPS73733 regulatorom je prikazana na slici 10.



Slika 10. Šema sistema za kontrolu električne energije.

Na predodređeni vremenski interval će se mikrokontroler uključivati. Tajmer će uključiti regulator i samim tim i mikrokontroler. Kada mikrokontroler izvrši primanje radio signala i postavljanje prikaza na ekranu onda će dati signal tajmeru na njegovom „DONE" izvodu, tajmer će isključiti regulator, pa će se i ceo sistem isključiti. Tako da je potrošnja minimalna, skoro 0mA u onih 58 minuta.

Takođe bitna stvar u ovoj šemi su MOSFET-i Q2 i Q4. Pri uključivanju regulatora uključiće se i N-MOS Q2. Onda će Q2 uključiti P-MOS Q4. P-MOS će dovesti napon baterije na naponski razdelnik sačinjen od otpornika R10 i R11. Vrednosti razdenika nisu predodređene jer se kasnije mogu odrediti na osnovu dostupnih otpornika. Bitno je samo da naponski razdelnik skalira maksimalni napon litijumske baterije od 4,3 V na 3,3 V ili manje (< 0,76 puta).

Zašto da skalira na 3,3 V? Zato što izlaz naponskog razdelnika vezujemo na analogni ulaz mikrokontrolera koji radi na 3,3 V. A taj napon na bateriji je bitan kako bi mikrokontroler mogao da obavesti korisnika da je baterija „pri kraju" kada je napon na bateriji mali. Mora da obavesti korisnika promenom prikaza na ekranu, jer E - papir ekrani mogu mesecima da stoje sa nepromenjenim prikazom. Pa onda korisnik ne bi znao da li je baterija sistema prazna pa je isti prikaz, ili se slučajno pogodilo da se ništa nije promenilo od informacija koje ekran prikazuje; kao što je u slučaju vremenske prognoze, gde je ista temperatura nekoliko sati u toku dana.

Što se tiče kapaciteta baterije proračun glasi:

$$6\,\text{meseci}\cdot 30\,\text{dana}\cdot 24\cdot\left(\frac{1}{60}(3\,\text{mA}+10\,\text{mA})\text{sati}+\frac{1}{60}(8\,\text{mA}+10\,\text{mA})\text{sati}+\right.$$
$$\left.+\frac{58}{60}(40\,\text{nA}+20\,\text{nA})\text{sati}\right)=2\,232{,}25\,\text{mAh}\approx 2{,}2\,\text{Ah}$$

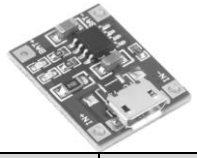
Znači kada se uračuna potrošnja TPL5111 i TPS73733 to je samo 0,25 mAh više na 6 meseci, nego kada je smatrano da je potrošnja u tih 58 minuta 0 mA. To je toliko mala potrošnja na tom vremenskom periodu da se može u potpunosti zanemariti za bilo koje dalje proračune.

Litijumske baterije imaju najmanji stepen samopražnjenja, 5% u prvih 24 časa od punjenja, pa onda 5% svakog meseca; uključujući i potrošnju kola za zaštitu baterije, koje se prodaje zajedno sa baterijom (zaštino kolo čuva bateriju od pregrevanja, prevelike struje punjenja i prevelikog i premalog napona na bateriji). Kapacitet baterije korišćen u ovom prototipu je 3500 mAh. Ukoliko se smatra da je realni kapacitet 3350 mAh, a treba prazniti bateriju do 20% njenog kapaciteta, onda je efektvni kapacitet baterije koji je dostupan 2680 mAh. Kada se uračuna potrošnja sistema i samopražnjenje baterije dobija se da ovaj sistem može da radi 5 meseci i 1 dan. To je jako blizu kriterijuma od 6 meseci. Jednostavno se može premašiti 6 meseci ukoliko se radio modul postavi da radi 30 sekundi umesto jednog minuta, što je opet i više nego dovoljno vremena da primi informaciju. Onda će sistem na jednom punjenju baterije raditi 194 dana, odnosno 6 meseci i 14 dana. Ukoliko se još smanji vreme rada prijemnika na 10 sekundi onda se dobija 225 dana rada, što je 7 meseci i 15 dana. Naravno da se može kapacitet baterije povećati dosta, pa bi onda i vreme rada sistema bilo još veće.

E. Punjač baterije

Takođe bitna stvar je način punjenja baterije. Korišćen je gotov modul za punjenje litijumske baterije u prototipu. Naziv tog modula je „TP4056A" (slika 11). Taj modul nije ništa više od jednog USB konektora, par pasivnih komponenti, jednog integrisanog kola i dva izvoda za bateriju. Integisano kolo se zove TP4056 i po njemu je

nazvan modul. To je integisano kolo koje je punjač za jednoćelijsku litijumsku bateriju i poseduje CC/CV metodu punjenja (eng. *Constant Current Constant Voltage*); to je najbolja metoda za punjenje litijumske baterije. Kojom maksimalnom strujom se može puniti baterija određuje se eksternim otpornikom vezanim na jedan pin TP4056. Preporučuje se da maksimalna struja punjenja litijumske baterije ne prevazilazi desetinu struje koju baterija može da preda u roku od jednog sata. Znači za ovu bateriju od 3350 mAh, maksimalna struja punjenja treba biti 335 mA ili manja. U tehničkoj dokumentaciji pišu različite vrednosti eksternog otpornika za različine vrednosti maksimalne struje (slika 11). Shodno tome odabran je otpornik od 3,9 kΩ kako bi maksimalna struja punjenja bila nešto veća od 300 mA.

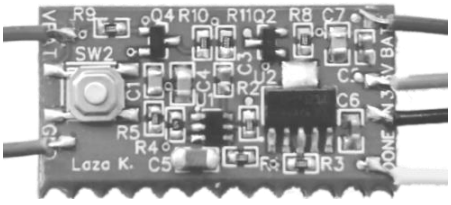


R_{prog} [kΩ]	I_{BATT} [mA]
10	130
5	250
4	300
3	400
2	580
1,66	690
1,5	780
1,33	900
1,2	1000

Slika 11. Modul TP4056A (gore); Tabela za odabir eksternog otpornika (dole).

Kada nema napona na ulazu u TP4056, postoji struja curenja iz baterije. Praktično TP4056 troši struju kada nema 5 V iz USB-a na njegovom ulazu. U tehničkoj dokumentaciji piše da je ta struja curenja od 2,5 μA do 6 μA. To bi za ovaj sistem značilo da bi jedno punjenje baterije trajalo 222 dana umesto 225 dana, što ne predstavlja veliki problem.

Kako su sve ove komponente sistema za kontrolu električne energije jako male i komplikovane da se koriste sa proto-pločom, odlučio sam da odradim PCB za taj sistem. Ova pločica je dosta olakšala proces sastavljanja ovog prototipa. Na slici 12 je prikazan realan gotov uređaj ovog sistema za kontrolu energije; PCB sa zalemljenim komponentama.



Slika 12. Prednja strana gotove štampane pločice sa zalemljenim komponentama (dužina pločice je 3,5 cm a širina 1,8 cm)

III. KRATKO OBJAŠNJENJE RADA PANELA

Već je rečeno okvirno kako bi info panel radio. Ali je neophodno napraviti algoritam, pa onda napisati kôd na osnovu tog algoritma kako bi mikrokontroler uskladio rad svih komponenata i modula.

Algoritam i deo kôda, na osnovu kog radi ceo sistem, će biti objašnjeni u ovom poglavlju.

Što se tiče informacija i prikaza za ovaj info panel, odlučeno je da to bude prikaz vremenske prognoze. Za prikaz vremenske prognoze će se koristiti 4 informacije:

- Da li je sunčano, oblačno, da li ima sunca i oblačno je, kao i da li je noć i da li je oblačno noću
- Da li ima padavina (kiša, sneg)
- Kolika je temperatura
- U koliko sati je pristigla nova prognoza.

Pošto se info panel napaja baterijom, a baterija se troši tokom rada, moraju se napraviti prioriteti; odnosno šta će se izvršiti pre nečeg drugog.

Logično bi bilo da prva stvar koju mikrokontroler treba uraditi pri uključivanju je da proveri napon na bateriji. Ukoliko je napon ispod granice od 3,7V, onda treba da postavi na ekranu prikaz da je baterija prazna, i nakon toga da se sistem ugasi.

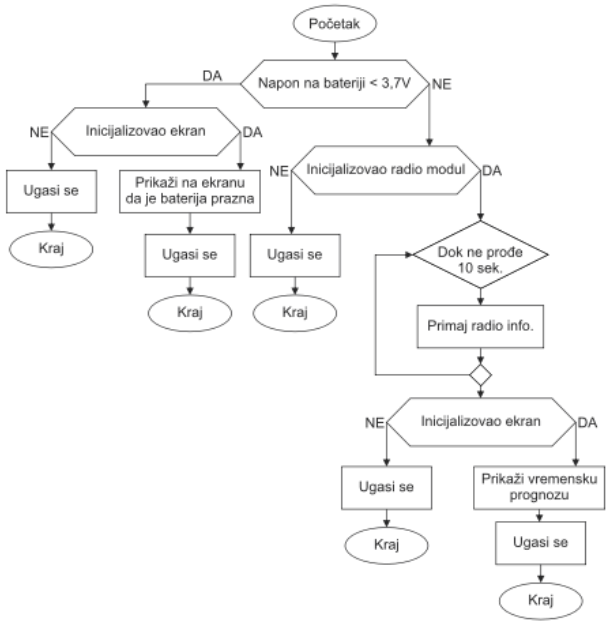
Sledeće što treba uraditi, ako je napon na bateriji iznad 3,7V, je da se otpočne primanje informacija za prikaz radio modulom. Vreme koje će radio modul primati informacije je 10 sekundi.

I na kraju, treba se na ekranu postaviti prikaz sa primljenim informacijama (što traje oko 1 minut).

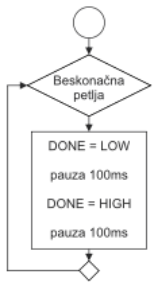
Ukoliko dođe do neke greške u radu sa nekom od komponenata ili modula, za ovaj prototip je odlučeno da se sistem ugasi. Ukoliko bi ovo bio komercijalni proizvod trebalo bi drugačije postupiti u slučaju greške. Greška u radu se smatra inicijalizacija ekrana ili radio modula.

Što se tiče inicijalizacije ekrana i radio modula, misli se na to da li je poslata odgovarajuća konfiguracija ekranu/radio modulu, i da li su oni odgovorili odgovarajućom porukom na tu konfiguraciju. Ukoliko jesu znači da sigurno rade i može se nastaviti sa daljom komunikacijom. Algoritam za rad ovog prototipa je dat na slici 13.

Pošto je jedna od najbitnih radnji da se sistem sam softverski ugasi, to će prvo biti opisano. Način na koji se sistem sam hardverski gasi je objašnjen kada je bio opisivan TPL5111 tajmer. Suštinski treba dovesti logičku jedinicu na „DONE" pin tajmera. Preciznije, tajmer će isključiti napajanje sistemu ukoliko na „DONE" izvodu prepozna prelaz sa logičke nule na logičku jedinicu. Možda tajmer neće registrovati uzlaznu ivicu signala kada se pošalje, zato bi bilo najbolje da se taj signal šalje na 200 ms sve dok tajmer ne odreaguje. S obzirom na to da kada tajmer odreaguje na signal on će ugasiti mikrokontroler, taj signal se može ponavljati beskonačno puta, i tako osigurati da će tajmer primetiti signal. Algoritam za blok samogašenja sistema je dat na slici 14.



Slika 13. Generalni algoritam rada sistema.



Slika 14. Algoritam samogašenja sistema.

Na početku rada je rečeno da je za ovaj prototip korišćen prikaz vremenske prognoze. Razlog za to je jednostavan: kako se panel periodično „budi" na sat vremena i trebalo bi da korisniku prikazuje neku korisnu informaciju, vremenska prognoza je dobar izvor informacija za tu primenu. Dodatno još, pošto je ekran trobojan (crna, bela žuta boja) bilo bi lepo da se ta karakteristika iskoristi, recimo za prikaz žutog sunca ili meseca. Slike koje trebaju ovaj prototip, prema prethodno navedenim uslovima su:

- Sunce
- Mesec
- Oblak
- Kap kiše
- Pahulja snega
- Cifre od broja nula do broja devet (i za temperaturu i za vreme)
- Znak za stepen i veliko slovo „C" (za stepen celzijusa)
- Dve tačke (za razdvajanje sata i minuta)
- Oznaka kako bi se naglasilo da nije trenutno vreme prikazano nego vreme kada je primljena trenutno prikazana prognoza (odlučio sam se za malo slovo „m" i znak „@" kao skračenica od eng. „measured at" („izmereno u"))
- Znak procenta za naznačavanje prazne baterije (%).

Dimenzije slika su određene vizuelno i na osnovu. rezolucije ekrana (600*448 piksela). Dimenzije slika u pikselima i njihova veličina u memoriji je:

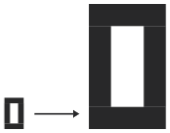
Naziv slike	Visina*Širina	Veličina[bajt]
Sunce	104*104	1352
Mesec	104*208	2704
Oblak	160*136	2720
Kap+Pahulja	24*24	144
Sve cifre	40*24	1200
„°C"	40*24	120
Dve tačke	40*24	120
„m"	40*24	120
„@"	40*24	120
„%"	40*24	120
	Σ =	8720 ≈ 8,5kB

Sa kriterijumima za veličinu i izgled slika se sada mogu i nacrtati te slike. Njihov izgled je prikazan na slici 15.



Slika 15. Izgled ikona/slika korišćenih u prototipu (nisu u srazmeri).

Razlog zašto su Sunce, Mesec i oblak „isečeni" je zato što bi inače bili prevelikih dimenzija da stanu u memoriju mikrokontrolera[1]. Cifre i znak za stepen celzijusa će biti softverski uveličane kako bi izgledale veće na ekranu (slika 16). A slike Sunca, Meseca i oblaka će biti softverski rotirane kako bi se prikazala celovita slika (slika 17).

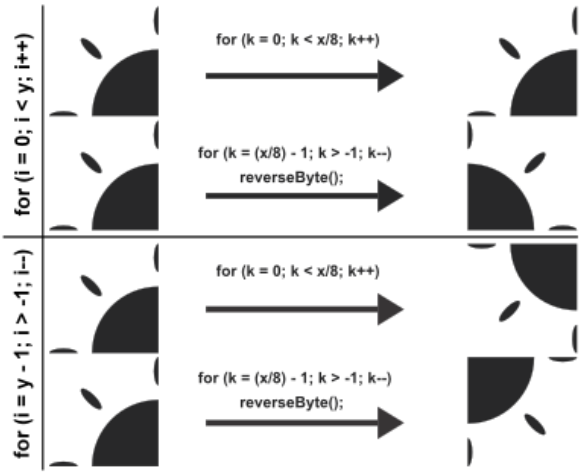


Slika 16. Primer uveličanja cifre nula (u srazmeri).

IV. REALIZACIJA PROJEKTA

Ekran je vrlo tanak (< 1 mm), i kako se ne bi polomio napravljen je za njega držač. Držač je napravljen od sunđera i tankog kartona, a karton koji ne dozvoljava da ekran ispadne iz držača je pričvršćen šrafovima za drvenu ploču na kome je ceo prototip smešten. Fotografisani su svi prikazi ekrana, svi delovi prototipa ponaosob, kao i ceo prototip. Ceo prototip je fotografisan po danu na sunčevoj svetlosti, kako bi se videla prednost E-papir ekrana u

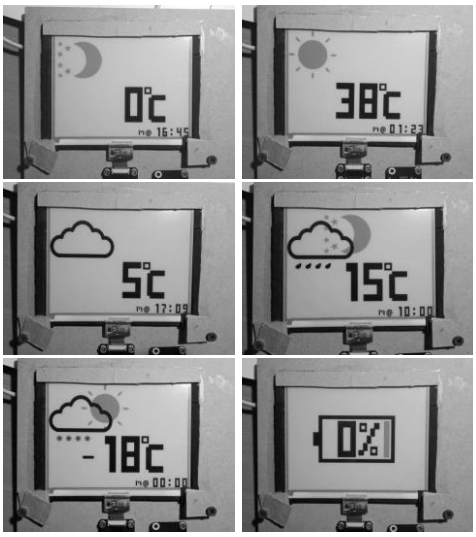
odnosu na klasičnu LED tehnologiju ekrana; njegova odlična preglednost pred intenzivnim izvorima svetlosti. Takođe je fotografisana realna potrošnja sistema za kontrolu energije, koja je manja u odnosu na zbir potrošnjia tajmera i regulatora iz njihovih tehničkih dokumentacija. Potrošnja sistema je izmerena preciznim izvorom napajanja „*KEYSIGHT B2901A*".

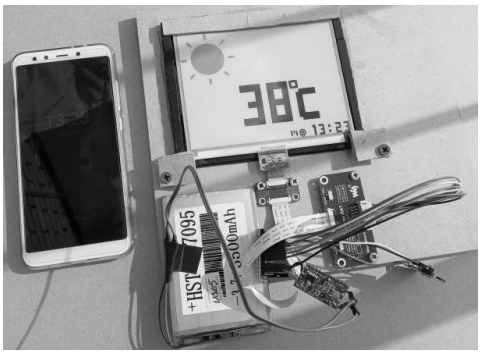


Slika 17. Kombinacije čitanja matrica za rotiranje i preslikavanje slika.



Slika 18. Realna potrošnja sistema za kontrolu energije, približno 46 nA, za napon napajanja od 4 V.





Slika 19. Sklopljen prototip napolju fotografisan, na sunčevoj svetlosti (poslednja slika); Različiti prikazi na ekranu, na osnovu primljenih informacija (ostale slike).

V. ZAKLJUČAK

Korišćenjem E - papir ekrana je prevaziđena velika prepreka u potrošnji info panela. To je zbog načina rada ekrana i njegove osobine da troši energiju jedino kada menja svoj prikaz. Uz ekran, odabirom modernih komponenata sa malom potrošnjom energije, je postignut veliki radni vek panela na jednom punjenju baterije. Prototip je realizovan u celosti i prikazan je njegov rad u realnom svetu, takođe su izvršena merenja kako bi se potvrdila teorijska predviđanja o potrošnji i ponašanju sistema. Dosta problema je bilo tokom izrade ovog prototipa koji nisu spomenuti u tekstu, ali kroz te probleme je autor mnogo toga naučio. Što je i svrha svakog studentskog projekta, da se steknu nova i prošire postojeća znanja, kako za autore projekta tako i za njihove kolege.

ZAHVALNICA

Autor se zahvaljuje Katedri za mikroelektroniku, Elektronskog fakulteta u Nišu na korišćenju laboratorijskog prostora i prenetom znanju tokom osnovnih akademskih studija.

LITERATURA

[1] Korišćenje fleš memorije mikrokontrolera ATmega328P https://www.avrfreaks.net
[2] Tehnička dokumentacija E-papir ekrana korišćenog u ovom radu https://www.waveshare.com/w/upload/7/70/5.83inch_e-Paper_Datasheet.pdf
[3] Tehničke dokumentacije više vrsta drajvera E-papir ekrana https://www.smart-prototyping.com/Other-Products/Modules/Displays/E-ink-E-paper-Display
[4] Tehnička dokumentacija komponente TPL5111 http://www.ti.com/lit/ds/symlink/tpl5111.pdf
[5] Tehnička dokumentacija komponente TPS73733 http://www.ti.com/lit/ds/symlink/tps737.pdf
[6] Tehnička dokumentacije mikrokontrolera ATmega328P http://ww1.microchip.com
[7] Specifikacije E-papir ekrana korišćenog u ovom radu https://www.waveshare.com

Testovi performansi u web aplikacijama

Risto Keković

Sadržaj – Cilj ovog rada je da predstavi testove performansi u web aplikacijama. U radu se čitalac najpre upoznaje sa konceptom testova performansi i potrebom da se oni primenjuju u produkcionim okruženjima. Pored toga, objašnjen je projektni kontekst koji se mora uzeti u obzir pri projektovanju i implementaciji testova. Dati su tipovi i vrste testova, i pojašnjenje svakog od njih. Zatim je predstavljen proces testiranja. Dati su svi koraci koje je potrebno preduzeti u ovom procesu i njihova veza sa projektnim kontekstom. Na kraju je dat kratak opis kako se testovi implementiraju.

I. UVOD

Teško je zamisliti današnji svet bez interneta. Web sajtovi donose nam vesti i zabavu, omogućavaju nam da se povežemo sa drugim ljudima, povećaju produktivnost, ali pružaju i mogućnost internet plaćanja - u bilo kom trenutku, iz bilo kog dela sveta, na bilo kom uređaju. Web aplikacije su softverske aplikacije nezavisne od platforme, koje se izvršavaju na web i/ili aplikacionom serveru, sa korisničkim interfejsom koji se kreira i prikazuje od strane klijentovog web pretraživača, a komunikacija se odvija preko računarske mreže.

Tokom ranih godina internet poslovanja, većina firmi nije ulagala vreme u bilo kakve pre-produkcione testove performansi. U najboljem slučaju, odlučivali su se za ručne testove, nadajući se da će njihova infrastruktura podržati opterećenje klijenata kada aplikacija bude puštena u produkciju. U takvim okolnostima, bilo je uobičajeno da sajt bude oboren nekoliko sati nakon što je pušten u produkciju, nakon ažuriranja, ili nakon što se kao posledica neke promotivne kampanje naglo poveća saobraćaj na tom sajtu [1].

U današnje vreme, velike firme ne mogu sebi dozvoliti čak ni kratkotrajne prekide, a nove generacije "nestrpljivih" korisnika očekuju pouzdan servis i brz odgovor u bilo kom trenutku. Ukoliko web sajt ne može da primi narudžbinu ili mu treba previše vremena da prikaže svoj sadržaj, korisnik neće oklevati da uslugu potraži kod nekog drugog. Velika konkurencija i nesigurna ekonomska klima zahtevaju da firme obraćaju mnogo veću pažnju na kvalitet, dostupnost i performanse njihovih aplikacija. Današnje web aplikacije su takođe mnogo kompleksnije nego ikada ranije. Skaliranje ovako kompleksne infrastrukture podrazumeva upravljanje performansama i kapacitetima pojedinačnih komponenti svakog sloja, kao i merenje ukupnog odziva celog sistema u toku transakcija.

R. Keković, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: risto.kekovic@gmail.com

Da bi se odgovorilo na ova pitanja, potrebno je predvideti kako će se data web aplikacija ponašati u produkcionom okruženju pre nego što se aplikacija pošalje tamo, i konstantno nakon toga meriti i posmatrati sistem tokom njegovog životnog ciklusa. Kao što je slučaj sa drugim tipovima softverskog testiranja, testiranje performansi se često zapostavlja ili se kompletno izostavlja u procesu projektovanja nekog softverskog rešenja. Ovo se delimično može povezati sa "mitovima" o testiranju performansi. Jedna od najvećih zabluda je da se testiranje performansi izvodi sa jedinom namerom da se obori sistem. Ovo ipak nije slučaj, iako se testovi performansi mogu izvršavati kako bi se identifikovala tačka zasićenja (eng. *saturation point*, maksimalno opterećenje koje sistem može da izdrži) [2].

Za bilo koju kompleksnu aplikaciju, bitno je da se testiranje počne obavljati u ranoj fazi projekta i da se testiranje obavlja često. Ovo ne samo da omogućava identifikaciju problema u ranoj fazi razvojnog ciklusa, dodatno pružajući mogućnost za ranu refaktorizaciju koda i druge korigujuće akcije, već pored toga omogućava i smanjenje troškova budućeg održavanja, koji često čine veliki deo ukupne cene aplikacije [1].

Testiranje performansi se takođe može iskoristiti za proučavanje skalabilnosti (mera koja pokazuje koliko dobro aplikacija reaguje na dodavanje ili smanjenje resursa koje ima na raspolaganju), pouzdanosti (mera koja pokazuje koliko je aplikacija robusna i otporna na greške) i iskorišćenosti resursa (procesiranje, zauzetost memorije i druge analize kroz profilisanje) aplikacije, kao i za upoređivanje različitih aplikativnih rešenja u pogledu performansi. Postoji nekoliko vrsta testova performansi koje se koriste u zavisnosti od konteksta i namene [1]. Bitno je naglasiti da se u obzir ne uzima metodologija razvoja softvera koji se testira, a koja može u mnogome uticati na veličinu i granularnost procesa testiranja.

Projektni kontekst

Kako bi testiranje performansi bilo uspešno, pristup testiranju i samo testiranje mora biti relevantno kontekstu projekta za koji se sprovodi. Bez potpunog razumevanja projektnog konteksta, testiranje se uglavnom fokusira samo na one aspekte aplikacije za koje individualni tester ili test tim smatra da su od značaja, umesto onih aspekata koji su zaista značajni za konačan uspeh projekta. Ovo često dovodi do izgubljenog vremena, frustracija i gubitka novca i resursa koje mogu dovesti i do celokupnog neuspeha projekta.

Projektni kontekst sastoji se od onih komponenata koje su ključne za uspeh projekta [3]. One mogu biti:

- Generalna vizija i namena projekta
- Ciljevi testiranja performansi
- Kriterijum uspešnih performansi
- Životni ciklus razvoja projekta
- Raspored projekta
- Budžet projekta
- Dostupni alati i okruženja

II. TIPOVI TESTOVA

Tehnike testiranja

Testiranje performansi aplikacije predstavlja način da se izmeri sposobnost te aplikacije da izvršava više transakcija simultano, u isto vreme zadržavajući prihvatljivo vreme odziva (eng. *response rate*). Pored toga, pomaže i u detekciji "uskog grla" (eng. *bottleneck*) u bilo kojoj komponenti mrežne infrastrukture u cilju identifikacije potencijalnih problema. Dva najčešća vida testova performansi su ručno (manuelno) i automatizovano [3].

Ručno testiranje, nažalost, suočava se sa nekoliko izazova:

- Poteškoće u emuliranju ponašanja nekoliko hiljada konkurentnih korisnika
- Koordinacija korisnika
- Precizno merenje odziva
- Poređenje rezultata između iteracija

Sa modernim alatima za automatizaciju testova performansi, moguće je istovremeno izvršavanje testova i merenje rezultata. Alati za automatizovano testiranje pomažu smanjenju faktora ljudske greške tokom testiranja. Iz tog razloga, ovaj tip testiranja postao je primaran izbor firmi svih tipova i veličina za poslovne aplikacije visokog stepena rizičnosti. Ovi alati najčešće koriste tri glavne komponente za izvršavanje testova:

- Kontrolna konzola koja organizuje, pokreće i upravlja opterećenjem i ponašanjem
- Virtualni korisnici (eng. *Virtual users* - Vusers) - procesi koji se koriste da simuliraju stvarne korisnike koji izvršavaju biznis transakcije nad klijentskom aplikacijom
- Serveri opterećenja (na njima se izvršavaju virtualni korisnici)

Korišćenjem ovih komponenti, alati za automatizovano testiranje mogu da:

- zamene stvarne test korisnike sa automatizovanim korisnicima
- simultano izvršavaju veliki broj virtualnih korisnika na istoj mašini za generisanje opterećenja na jednoj ili više lokacija širom sveta
- automatski mere vremena odziva transakcija i posmatraju infrastrukturne komponente
- lako ponavljaju scenarije opterećenja kako bi testirali bilo koju promenu napravljenu u cilju poboljšanja performansi

Četiri osnovna tipa (*LESS* pristup) [4] testova performansi su:

- Test opterećenja (eng *Load test*) – L
- Test izdržljivosti (eng. *Endurance test*) – E
- Stres test (eng. *Stress test*) – S
- Test impulsa (eng. *Spike test*) – S

Test opterećenja

Test opterećenja sprovodi se sa ciljem da se vidi kako se aplikacija ponaša pod različitim opterećenjima [4]. Ovo podrazumeva testiranje ponašanja sa različitim brojem simuliranih korisnika koji u isto vreme interaguju sa aplikacijom, različitim dužinama testiranja i koracima. Iako se u toku testiranja mogu proveravati i neka nerealna opterećenja sa kojima se sistem u realnom svetu neće susretati (ovo već predstavlja domen kojim se bave stres testovi), uglavnom se testiraju očekivana opterećenja u produkcionom okruženju. Test opterećenja predstavlja nefunkcionalni tip testa.

Alati koji se mogu koristiti za testiranje opterećenja su Dženkins (eng. *Jenkins*) i Taurus.

Test izdržljivosti

Testiranje izdržljivosti je usko povezano sa testom opterećenja, ali je ono orjentisano primarno na vremenski period. Ono podrazumeva testiranje sistema pod opterećenjem u dužem vremenskom intervalu i analiziranje njegovog ponašanje tokom tog perioda [4].

Ovakav tip testiranja se obavlja uglavnom pred kraj razvojnog ciklusa aplikacije. Osigurava da je aplikacija sposobna da podrži kontinuirano opterećenje bez bilo kakve degradacije u vremenu odziva, propusnosi i generalnom kvalitetu usluge koju pruža [3]. Jedna od najvećih mana ovog tipa testiranja je to što ono može trajati jako dugo. U zavisnosti od obima projekta, ono može potrajati i do godinu dana, za razliku od testa opterećenja koje traje samo nekoliko sati.

Stres test

Stres test je test dizajniran sa konkretnom namerom da pronađe gornje granice sistema tako što će ga testirati pod ekstremnim opterećenjima [4]. Test proverava kako se aplikacija ponaša pod intenzivnim opterećenjima, i na koji način se aplikacija oporavlja od ovakvih scenarija. Ovaj tip testova omogućava da se isplaniraju i isprojektuju tehnike za gašenje i oporavak sistema od otkazivanja. Stres test pokazuje u kojim situacijama sistem može da otkaže i padne i omogućava razvojnom timu da pokrije sve te granične slučajeve, kako u toku procesa korišćenja ne bi došlo do neprijatnih iznenađenja.

Test impulsa

Ovaj tip testa je zapravo specijalan slučaj stres testa, i koristi se za određivanje kako sistem reaguje na naglo povećanje opterećenja [4]. Ovakav scenario se često dešava u produkcionom okruženju, kada se usled povećanog broja korisničkih zahteva naglo poveća saobraćaj na sajtu i aplikacija to mora predvideti i na pravi način obraditi taj "impuls" u sistemu. Ime potiče iz oblasti elektronike, gde se često dešava da dođe do oštrog strujnog implusa (eng. *spike*). Ovaj tip testa testira upravo takve impluse u opterećenju i posmatra na koji način će sistem odgovoriti na njih.

III. PPROCES TESTIRANJA PERFORMANSI



Na najvišem nivou, testiranje performansi se skoro uvek sprovodi kako bi se pozabavio jednim ili više rizika vezanih za troškove, kontinuitet, sigurnost ili reputaciju firme [2]. Sam proces testiranja obuhvata sledeće korake:

- Identifikacija test okruženja. Identifikuju se fizičko test okruženje i produkciono okruženje, kao i alati i resursi koji su dostupni test timu. Fizičko okruženje uključuje hardver, softver i mrežne konfiguracije. Razumevanje celokupnog test okruženja omogućava efikasnije planiranje i projektovanje i pomaže u identifikaciji test izazova u ranoj fazi projekta. U nekim situacijama, ovaj proces mora biti periodično revidiran tokom projektnog životnog ciklusa.
- Identifikacija kriterijuma prihvatljivosti. U ovoj fazi bitno je identifikovati vreme odziva, propusnost, kao i ciljeve i ograničenja korišćenja resursa. Uopšteno govoreći, vreme odziva je korisnička briga, propusnost je poslovna (biznis) briga, a korišćenje resursa je sistemska briga. Pored ovih aktivnosti, potrebno je i identifikovati kriterijume uspeha projekta koji nisu obuhvaćeni već definisanim kriterijumima; na primer, korišćenje testova performansi kako bi se ocenilo koja podešavanja sistema će rezultovati najpoželjnijim performansama.
- Planiranje i projektovanje testova. Identifikuju se ključni scenariji, određuje se varijabilnost među reprezentativnim korisnicima i način kako ta varijabilnost može biti simulirana, definišu se test podaci i određuje se metrika koja će biti sakupljena. Ove informacije bi trebalo objediniti u jedan ili više modela korišćenja sistema koji će biti implementirani, izvršeni i analizirani.
- Podešavanje test okruženja. U ovoj fazi priprema se test okruženje, alati i resursi potrebni za izvršavanje svake strategije kako komponente postaju dostupne za testiranje. Potrebno je osigurati da je test okruženje projektovano za monitoring resursa po potrebi.
- Implementacija testova. Testovi performansi se implementiraju u skladu sa prethodno isprojektovanim test dizajnom.
- Izvršavanje testova. Testovi se pokreću i posmatraju. Potrebno je validirati testove, test podatke i kolekciju rezultata. Izvršavaju se validni testovi za analizu, a u isto vreme se posmatra testiranje i test okruženje.
- Analiza rezultata, izveštavanje i retestiranje. Objedinjuju se rezultati i ti podaci se analiziraju individualno i grupno. Ostali testovi se prioritetizuju i ponovo izvršavaju po potrebi. Ukoliko su sve metričke vrednosti u okvirima dozvoljenih granica, nijedan od postavljenih pragova nije prekoračen i sve željene informacije prikupljene, konkretan test scenario je prošao testiranje i može se ići dalje.

IV. IMPLEMENTACIJA TESTOVA

Testovi performansi se obično implementiraju kao skup *skripti* koje se izvršavaju sa ciljem da stvore *opterećenje* u jednoj ili više instanci odredišne aplikacije. Ovo se postiže korišćenjem nekoliko mašina za stvaranje opterećenja (eng. *load injector machines*), od kojih svaka pokreće posebnu instancu testnog skupa [3]. Najveći broj trenutno dostupnih alata za testiranje performansi dozvoljavaju koordinisano izvršavanje distribuiranih testova. Međutim, za scenarije manjih razmera, nije uvek

potrebno više instanci i često je dovoljna samo jedna mašina za generisanje opterećenja, jer većina alata za testiranje opterećenja može efektivno da simulira više konkurentnih korisnika, čak i na samo jednom injektoru.

Test skripte mogu biti ručno kreirane ili napravljene pomoću odgovarajućeg alata. Primer osnovne test skripte scenarija sa 100 konkurentnih mašina u trajanju od 5 minuta dat je u nastavku. Skripta je data u jeziku YAML.

```
execution:
- concurrency: 100
  ramp-up: 1m
  hold-for: 5m
  scenario: quick-test

scenarios:
  quick-test:
    requests:
      - http://blazedemo.com
```

V. ZAKLJUČAK

Web aplikacije postale su glavni vid komunikacije i poslovanja na internetu. Kao takve, privlače veliki broj posetilaca, što uzrokuje povećanim saobraćajem na tom sajtu. Tipična reakcija na porast saobraćaja na sajtu je kupovina dodatnog hardvera za svaku komponentu sistema. Međutim, takvi nasumični pokušaji da se performanse poboljšaju često dovode do kupovine prekomernih resursa i ne nužno do boljih performansi kojima se teži. U takvim slučajevima uglavnom je uzrok to što aplikacija nije inicijalno projektovana tako da bude elastična. Jedini način da se sa sigurnošću predvidi ponašanje sistema je projektovanje i izvršavanje testova performansi u sistemu i beleženje rezultata iz korisničke perspektive, pre odobravanja da sistem ode u produkciju.

Testovi performansi se sastoje od test skripti koje se pišu sa namerom da provere kapacitet opterećenja, izdržljivost, kao i gornje granice sistema. U zavisnosti od potrebe, koristi se jedan od četiri tipa testa: test opterećenja, test izdržljivosti, stres test ili test impulsa. Neki od ovih testova zahtevaju duži vremenski period, pa se stoga testiranje performansi mora planirati na samom početku projekta, kako ne bi uzrokovalo kašnjenje sa isporukom aplikacije.

Za web aplikacije, izvršavanje skripti se obično posmatra praćenjem HTTP saobraćaja između klijenta i odredišne aplikacije pomoću proksi (eng. *proxy*) servera. U zavisnosti od odabranog alata za testiranje, skripte mogu biti dopunjene tako da koriste uslovne testove, parametrizaciju (npr. korisničke podatke koji se dinamički generišu), petlje, provere, merače vremena i nasumične test elemente. Skripte mogu biti napisane i izvršene u velikom broju programskih jezika ili jezika za serijalizaciju, poput *XML*, *YAML*, *JSON*, *C*, *Java* ili *Python*.

Ključni deo testiranja performansi je izveštavanje. Iz tog razloga, alati za testiranje performansi moraju da ponude načine za analizu i isporuku rezultata testiranja. Broj i kvalitet izmerenih metrika zavise od tipa testa i odabranog alata za testiranje. Najčešći format rezultata za web aplikacije podrazumeva uključivanje vreme odziva i statusni kod HTTP odgovora. Rezultati su obično predstavljeni u tekstualnom formatu, korišćenjem nekih od najzastupljenijih formata za predstavljanje podataka kao što su XML (eng. *Extensible Markup Language*) ili CSV (eng. *Comma-Separated Values*), ili korišćenjem različitih vrsta grafikona.

LITERATURA

[1] J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea, "Performance Testing Guidance for Web Applications: Patterns & Practices", Microsoft, 2007

[2] Jukka Palomäki, "Web Application Performance Testing", University of Turku, Dept. of Information Technology, Master's thesis, 2009

[3] "Performance Testing for Web Applications", MicroFocus

[4] Subraya, "Integrated Approach to Web Performance Testing: a Practitioners Guide", B.M, Infosys Technologies Limited, Mysore, India, Januar, 2006

Maketa naponskog razdelnika sa potenciometrom

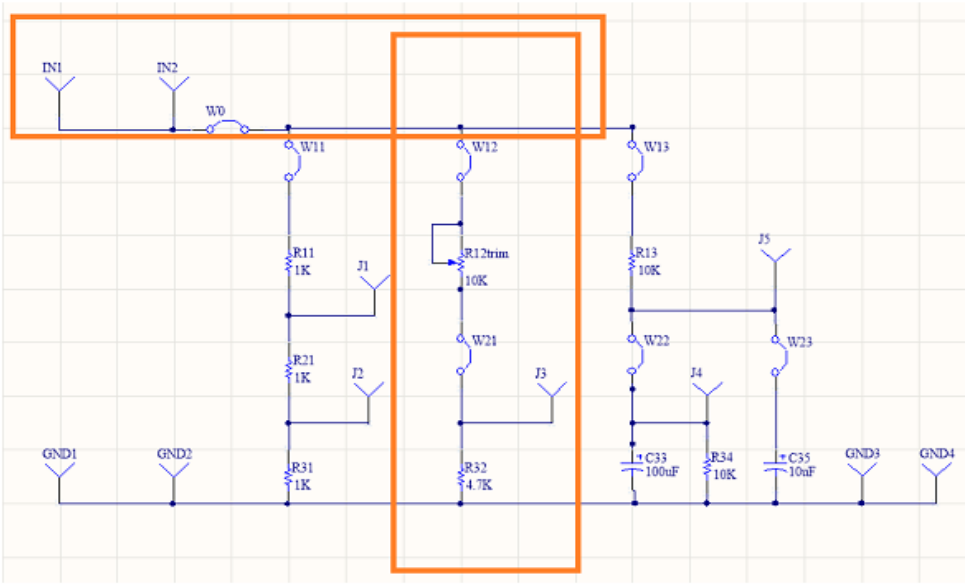
Dušan Golubović, Dimitrije Đermanović

Sadržaj- U ovom radu prikazan je princip rada makete naponskog razdelnika sa dva otpornika od kojih je jedan potenciometar.

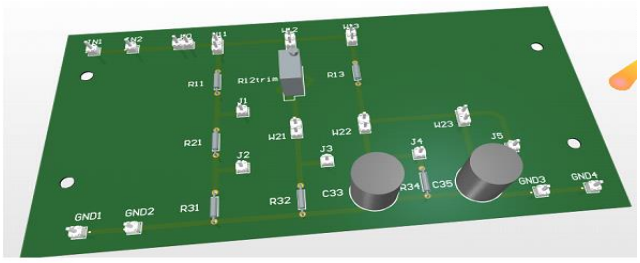
I. UVOD

Za realizaciju ove vežbe koristićemo maketu koja sadrži četiri električna kola. Za biranje odgovarajućeg električnog kola potrebno je da „džamperima“ spojimo kontakte: W0, W12 i W21 (slika 1). U tom slučaju aktivan je samo obeleženi deo sa slike 1.

Prikazan je i 3D model makete na kome se vide sve komponente i konektori neophodni za realizaciju ove vežbe (slika 2). 3D model makete realizovan je u programu Altium Designer [1].



Slika 1. Šema makete.



Slika 2. 3D model makete.

II. IZVOĐENJE LABORATORISKE VEŽBE

Za izvođenje laboratoriske vežbe pored makete potrebni su: osciloskop [2], generator funkcija [3], izvori napajanja i kratkospojnici.

Dušan Golubović, Dimitrije Đermanović, učenici tehničke škole „Rade Metalac“ Leskovac. E-mail: graffitibomber7@gmail.com

Na početku, podesićemo generator funkcija tako da daje sinusni naizmenični napon od 5 V frekvencije 100 Hz. Plus kraj generatora funkcija spojićemo sa konektorom IN2 a minus kraj sa nekim od GND konektora. Mernim instrumentom merimo napon na konektoru J3. Primetili smo da je napon na konektoru J3 dosta manji od napona na ulazu što je posledica postojanja razdelnika napona.

U tabeli 1 su prikazani naponi na izlazu u zavisnosti na otpornosti trimera.

TABELA I
VREDNOST NAPONA NA IZLAZU U ODNOSU NA RAZLIČITE OTPORNOSTI TRIMERA.

R12trim [kΩ]	10	5	3.7	0
VJ3 [V]	1.559	2.423	2.798	5

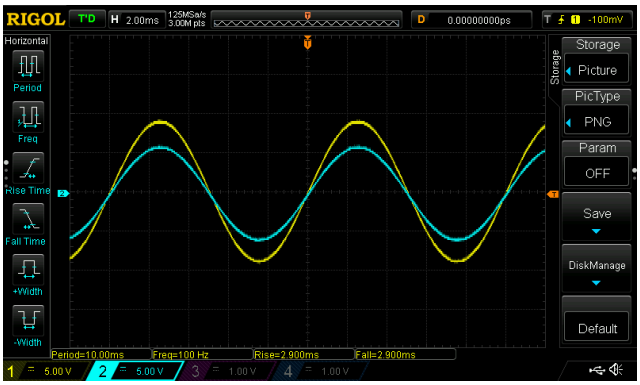
Korišćenjem Omovog zakona možemo izračunati koliki je napon na izlazu J3 primenom formule:

$$VJ3=(R32/(R32+R12trim))\cdot VIN \quad (1)$$

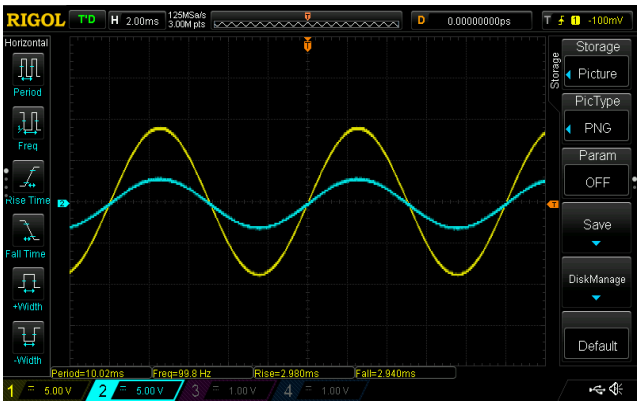
III. SNIMANJE SIGNALA OSCILOSKOPOM

Za povezivanje osciloskopa sa maketom potrebno je da pozitivni kraj sonde osciloskopa priključimo na konektor J3, a negativni kraj priključimo na neki od GND konektora.

Slike 3 i 4 prikazuju oblike ulaznog napona dovedenog iz generatora funkcija i usnimljenog napona na J3 konektoru u nekim proizvoljnim položajima potenciometra.



Slika 3. Oblik signala na izlazu u približno središnjem položaju potenciometra.

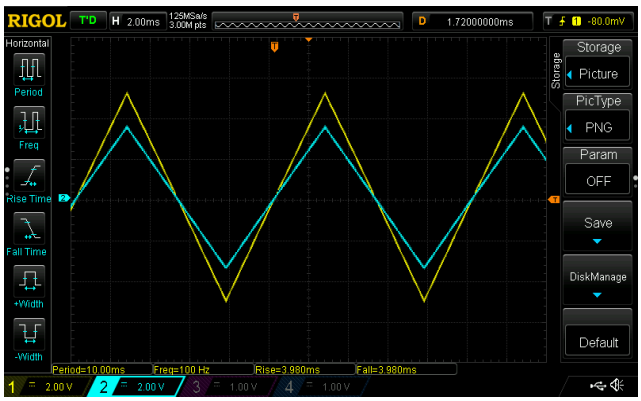


Slika 4. Oblik signala na izlazu kada je potenciometar okrenut tako da ima najveću otpornost.

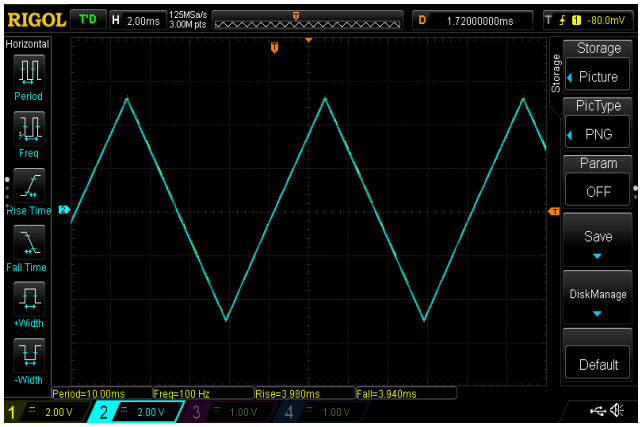
Menjanjem sinusnog napona u trouglasti (podešavanjem generatora funkcija) dobijamo slike gde se vidi jasna razlika u amplitudi izmedju ulaznog i izlaznog signala (slike 5 i 6), što je posledica postojanja naponskog razdelnika. Realizacija eksperimenta prikazana je na slici 7.

IV. ZAKLJUČAK

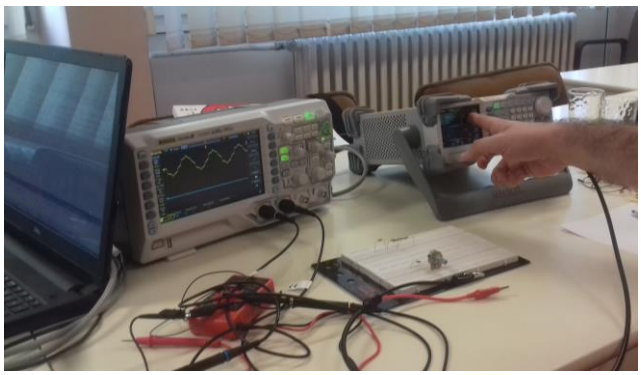
Zaključujemo da menjanjem otpornosti potenciometra dobijamo mogućnost regulacije napona na izlazu. Naponski razdelnik ima veliku primenu u elektronici i jedno je od najčešće upotrebljenih kola.



Slika 5. Trouglasti signal na izlazu sa približno srednjom otpornošću potenciometra.



Slika 6. Poklapanje ulaznog i izlaznog trouglastog signala kada potenciometar nema otpornost.



Slika 7. Realizacija eksperimenta korišćenjem protoploče.

LITERATURA

[1] Altium Designer. Dostupno na: https://www.altium.com/altium-designer/
[2] Osciloskop RIGOL DS 1054, https://www.rigol.eu/products/digital-oscilloscopes/
[3] Generator signala RIGOL DG1022, https://www.rigol.eu/products/waveform-generators/

Meteorološka stanica realizovana korišćenjem Internet of things (IoT) tehnologije

Denis Medarski

Sadržaj - Obrađena je ideja daljinskog pristupa ESP8266 modula ka internet portalu Ubidots koji ima svrhu praćenja temperature i vlažnosti vazduha, kao i automatskog kontrolisanja klima uređaja.

I. Internet of Things (IoT – Internet stvari)

IoT je jedan od najvećih trendova poslednjih godina. Laka dostupnost internetu i jeftini moćni mikrokontroleri sa ugrađenom WiFi tehnologijom omogućili su procvat internet stvari. Ideja je povezivanje svakodnevno korišćenih uređaja poput ručnih satova, frižidera, rasvete, fitnes pomagala na internet. To bi omogućilo razmenu informacija svakodnevnih stvari i njihovu dostupnost sa bilo kog mesta. Takav pristup omogućuje i direktnu međusobnu komunikaciju stvari što se naziva M2M (Machine to Machine) komunikacija.

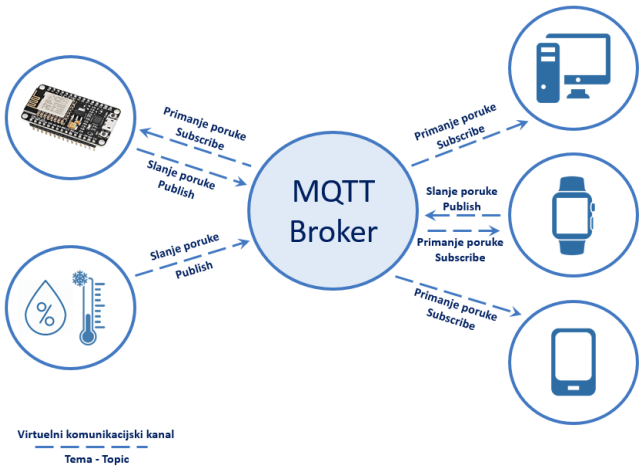
Do 2020. godine prema predviđanjima bi se ta brojka mogla popeti na 50 milijardi spojenih stvari na internet [1].

II. MQTT Protokol

MQTT (eng. *Message Queuing Telemetry Transport*) je M2M (eng. *Machine to Machine*) komunikacioni protokol koji služi za razmenu informacija između IoT članova [2]. Komunikacija može biti između jednog klijenta i drugog klijenta ili jednog klijenta prema više klijenata. Komunikacija je anonimna jer klijent koji šalje poruku ne zna kome je šalje, a kad primi poruku, ne zna od koga je primio poruku. Klijent je na MQTT sistem povezan samo sa posrednikom i za povezivanje na njega klijent bi trebalo da ima poznatu informaciju o IP adresi posrednika i porta putem koga se komunikacija obavlja. Najčešće je to port 1883 za MQTT TCP komunikaciju odnosno 8883 za MQTT TLS komunikaciju, odnosno 80/443 za MQTT webSocket komunikaciju.

MQTT posrednik: (eng. Broker) često se naziva i serverom i on je srce samog protokola (Slika 1.). Zavisno o implementaciji može se nositi s hiljadama klijenata istovremeno obrađujući i prosleđujući poruke istovremeno održavajući kontinuitet i pouzdanost komunikacije. Njegov je zadatak da primi poruku, filtrira po temama i da prosledi poruku klijentima koji zadovoljavaju kriterijum pretplate na temu.

MQTT klijent: (eng. *Client*) je pojam koji se koristi za bilo koji uređaj koji ima mogućnost spajanja na MQTT broker preko žičane ili bežične mrežne veze i ima mogućnost primanja odnosno slanja MQTT poruke. Klijent mora imati TCP/IP magacinsku memoriju koji je nezaobilazna za korišćenje samog MQTT protokola. Uređaj može biti mikrokontroler slabijih karakteristika, pa sve do moćnih računarskih servera.



Slika 1. MQTT posrednik: (eng. Broker).

MQTT ima tri nivoa kvaliteta usluge. Ukoliko je važno da poruka pouzdano stigne na odredište definiše se viši QoS [3] (eng. Quality of Service). Posledica toga je veće zauzimanje mrežnog prostora i latencija poruke.

- QoS 0 – poruke se dostavljaju na najbolje iskorišćenje mrežnog prostora ali može doći do gubitka poruke.
- QoS 1 – poruka pouzdano stiže na odredište, moguće duple poruke.
- QoS 2 – poruka pouzdano stiže na odredište uz nemogućnost pojavljivanja duplih poruka.

III. NodeMCU v3.0

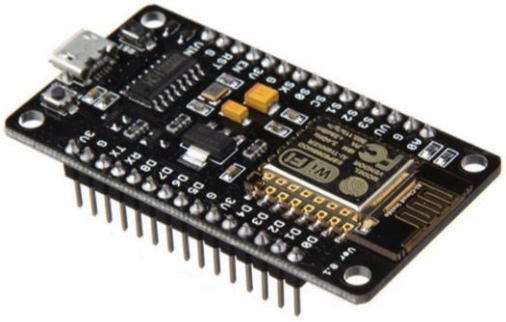
ESP8266 je SoC (System on a Chip) integrisani sklop niske cene dizajniran za prostorno i energetske ograničene mobilne platforme [4]. Ima ugrađenu sposobnost WiFi veze koja nudi mogućnost umrežavanja pametnih sistema ili pak deluje kao samostalna WiFi stanica (Slika 2.). Kada ESP deluje kao samostalna jedinica tada se njegov programski kod direktno pokreće sa spoljašnje memorije uz korišćenje priručne keš (eng. cache)

Denis Medarski, student Elektronskog fakulteta, modul: Elektronske komponente i mikrosistemi, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: denis.medarski@heineken.com

memorije za poboljšanje performansi u te svrhe. ESP može imati i namenu WiFi adaptera koji omogućava umrežavanje gotovo svih mikrokontrolera svojim jednostavnim načinom perifernog serijskog povezivanja (SPI/SDIO ili I2C/UART). Jeftiniji je od tradicionalnog Arduina sa WiFi adapterom uz gotovo iste mogućnosti. Zbog toga se puno koristi u sistemima kućne automatizacije [4].

Namena:

- Automatizacija stambenih zgrada
- Pametne utičnice i rasveta
- Umrežavanje Sistema
- Industrijska WiFi kontrola
- Baby monitor
- IP kamere
- Telemetrijski sistem senzora
- Pametni nosivi uređaji (narukvice, satovi, ..)
- Sigurnosne ID oznake
- WiFi sistemi pozicioniranja i drugo



Slika 2. NodeMCU razvojna ploča sa ESP8266 mikrokontorlerom.

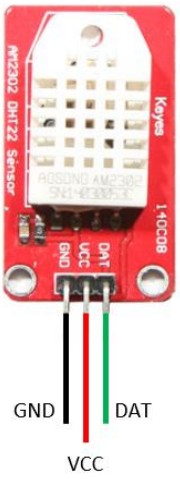
IV. PROGRAMIRANJE

Za programiranje samog mikrokontrolera potrebno nam je razvojno okruženje tj, program u kome ćemo programirati mikrokontroler. Za ovaj projekat koristićemo Arduino integrisano razvojno okruženje (eng. *Arduino Integrated Development Envoirment "IDE"*). Osim razvojnog okruženja, potreban nam je i modul kojim će se preko računara prebaciti serijskom komunikacijom kompajlirani program u mikrokontroler. U tu svrhu može se upotrebiti i Arduino (npr. Arduino Uno) ili pak samostalne niskobudžetne i lako dostupne module kompatibilne sa ESP8266 čipom poput „FTDI USB/UART programera".

Arduino IDE je besplatni razvojni alat koji služi za programiranje Arduino mikrokontrolera. Kompatibilan je sa ES8266 čipom zahvaljujući neprofitabilnoj društvenoj zajednici koja je osigurala podržavanje ESP8266 modula u Arduino razvojnom okruženju. Sam razvojni alat je kreiran u Java programskom kodu ali se za pisanje programa koji se koristi u mikrokontrolerima upotrebljava C/C++ prilagođen Arduino okruženju. Sadrži editor programskog koda koji znatno olakšava pisanje funkcionalnog programskog koda.

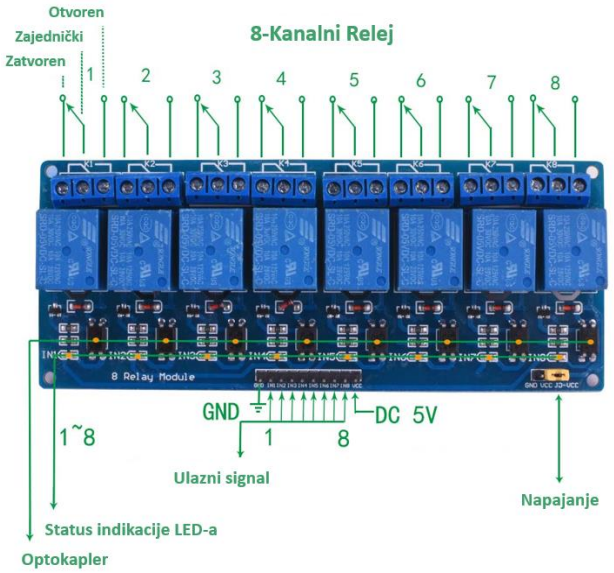
V. SENZORI I RELEJNI MODUL

DHT22 je senzor malih dimenzija namenjen je merenju temperature i vlažnosti vazduha u ne previše zahtevnim i agresivnim okruženjima obavlja svoj posao u automatizaciji i sličnim funkcijama (Slika 3.). Dužina komunikacije je do 20 metara. Sastoje se od A/D konvertera koji na izlaznom „DAT" pinu daje digitalnu vrednost temperature i vlažnosti. Jako je bitan podatak da sam senzor zahteva 1mA struje [5], zbog napajanja putem mikrokontrolera. Prvi pin sleva je GND pin koji se spaja na 0V. Drugi pin sleva VCC se spaja na 3-5V jednosmernog radnog napona. Sam modul senzora je spojen sa 10 kOhm pull-up otpornikom između DAT i VCC pina kao što je prikazano na slici ispod. Treći pin sleva DAT je digitalni izlazni pin sa kojeg se čitaju podaci i on se spaja na mikrokontroler. Takođe je na samom modulu dodat 100nF kondenzator između GND i VCC zbog filtriranja smetnji. Podatci se čitaju uz pomoć DHT biblioteke koja pomoću naredbi dht.readHumidity() i dht.readTemperature() omogućava čitanje podataka. Minimalna pauza između čitanja je 2s.



Slika 3. Pinout DHT22 senzora.

Relej je vrsta prekidača čije je stanje (uključeno ili isključeno) upravljano električnim naponom, tj. energijom elektromagneta ili nekog drugog mehanizma. Signalni pin pri određenoj vrednosti automatski zatvara ili otvara kontakte energetskog dela. Za 8-JQC3F-05VDC-C (Slika 4.) ta je vrednost 5VDC. Vreme uključenja kao i isključenja je 5ms [6]. Vreme uključenja je 10ms. Maksimalna struja koja može proteći kroz kontakte je 10A za 240VAC. Relej može raditi u temperaturnom opsegu okoline od -25°C do +70°C. Vreme života je 10^5 operacija. Slika ispod prikazuje oznake pinova i šemu spajanja energetskog dela 8-JQC3F-05VDC-C releja na potrošač [6].



Slika 4. Relej 8-JQC3F-05VDC-C.

VI. REALIZACIJA IOT OKRUŽENJA

U samoj realizaciji meteorološke stanice govoriće se o načinu povezivanja i komunikacije između komponenti navedenih u prethodnom poglavlju i samog interneta portala. Takođe ćemo prikazati korisničko okruženje (*eng. User Interface "UI"*).

U svrhu korišćenja MQTT protokola za razmenu poruka potreban je posrednik na kojeg se zatim spajaju klijenti. U ovom radu korišćena je verzija MQTT posrednika koja je već implementirana na portalu Ubidots. Za upravljanje i nadzor ESP8266 mikrokontrolera korišćena je web stranica things.ubidots.com [7]. Web portal Ubidots je asinhroni klijent koji se spaja na posrednika putem webSocketa i na taj način razmenjuje poruke.

Slanje podataka sa mikrokontrolera na Ubidots portal putem MQTT-a, vrši se sledećom linijom koda:

- client.ubidotsPublish("DEVICE_LABEL");

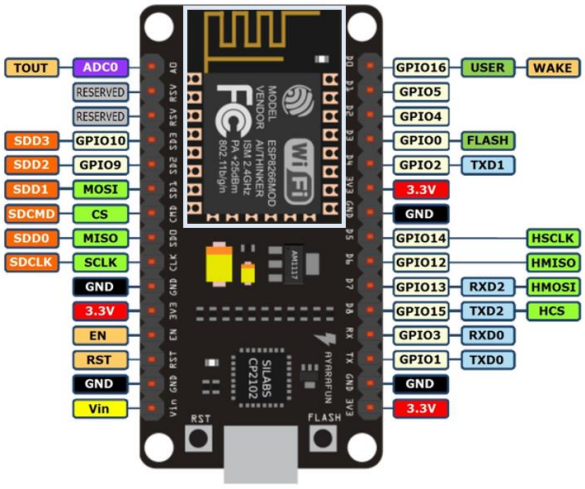
Primanje podataka na mikrokontroler sa Ubidots portala putem MQTT-a, vrši se sledećom linijom koda:

- client.ubidotsSubscribe(DEVICE_LABEL, VARIABLE_LABEL);

Sa MQTT protokolom, mikrokontroler može da „osluškuje" oblak (eng. Cloud) i dobija obaveštenje samo o promeni promenljive [8]. Na ovaj način, veza između uređaja i Ubidots portala ostaje otvorena, ali podaci putuju samo kada je to potrebno, čime se štedi propusnost mreže i poboljšava rad u realnom vremenu.

Postoje tri verzije NodeMCU razvojne pločice: v0.9, v1.0, i v3.0. U ovom radu korišćena je v3.0 razvojna pločica, koja na sebi sadrži ESP8266 12E SoC. ESP 12E verziju sa 32bit procesorom, 4MB memorije za čuvanje podataka, 128 KB radne memorije, 10 ulazno izlaznih pinova opšte namene a svaki ima PWM mogućnost. Uobičajena potrošnja električne struje je 80mA [9]. Struja koju pinovi daju na izlaz je 6mA, koja je više nego dovoljna za napajanje DHT22 senzora 1mA pri 3.3V [5].

Na razvojnoj pločici dimenzija 2.5400×4.8260cm [10] nalaze se: mikrokontroler, regulator napona, „flash" i „reset" dugme , ugrađena plava LED dioda, „USB to UART" komunikacija, spoljne pinove zatim ostale komponente koje omogućavaju razvoj i testiranje. Već ugrađena „USB to UART" komunikacija omogućava povezivanje razvojnog programa sa modulom. Na računaru je potrebno imati instaliran „USB to UART" upravljački program ili (eng. *Driver*). Razvojna pločica ima ugrađen regulator napona koji omogućava 5V napajanje preko microUSB kabla. Tim kablom ujedno i vršimo snimanje programa u modul ili se koristi za upravljačko nadzornu serijsku komunikaciju. Raspored pinova i naziva vidi na Slici 5. [10]



Slika 5. Raspored i oznake pinova [10].

UbidotsESPMQTT.h biblioteka olakšava povezivanje na MQTT posrednika na Ubidots portalu, pozivanjem gotovih naredbi. U početnom postavljanju mora se definisati klijent koji će se spojiti na posrednik naredbama WIFINAME, WIFIPASS, DEVICE_LABEL, VARIABLE_LABEL i TOKEN. Iz trajne memorije uzimaju se informacije o adresi MQTT posrednika i portu, zatim naredbom client.wifiConnection(WIFINAME, WIFIPASS); pridodaju se informacije klijentu. Nakon što su obuhvaćeni svi potrebni parametri, (programskim blokom 1.) u nastavku, povezuje se na MQTT posrednik, ulazi u petlju i šalje podatke portalu svakih 60s. Ukoliko posrednik nije povezan, modul poziva funkciju reconnect(), kojom se ponovo povezuje na Ubidots. Uspostavljanjem veze potrebno je poslati TOKEN kako bi se omogućila komunikacija. Nakon uspešno uspostavljene komunikacije klijentu (mikrokontroleru) je omogućeno slanje i prijem poruka. Ukoliko želimo da kontrolišemo izlazne pinove mikrokontrolera tj. da uključimo neke od relejnih modula, ESP8266 prima poruke i mora se posredniku dati do znanja zainteresovanost za određene teme, u konkretnom slučaju

su to DEVICE_LABEL i VARIABLE_LABEL. Osim toga mora se definisati funkcija koja će vršiti dohvat i kontrolu nad porukama. client.setCallback(callback); daje do znanja klijentu da je funkcija callback odgovorna za prijem poruka.

VII. Testiranje IoT okruženja

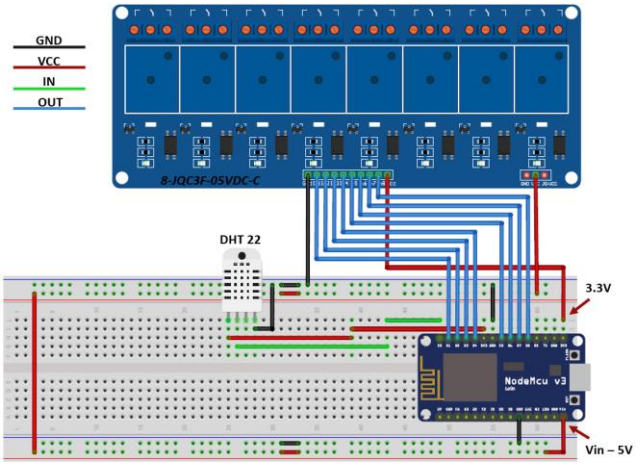
Nakon same provere kompatibilnosti svih gore navedenih komponenti: modula sa mikrokontrolerom, senzora, relejnog modula, programskog koda i konekcije ka Ubidots portalu, dolazimo do dela kada je potrebno sve spojiti i proveriti funkcionalnost sistema. Za kreiranje električne šeme i načina povezivanja koristili smo Fritzing program [11], koji je i više nego dovoljan za potrebe testiranja. (Slika 6.) Predstavlja ilustrovani prikaz povezivanja NodeMCU i relejnog modula u svrhu testiranja na proto ploči, i električnu šemu realizovane meteorološke stanice. Testiranje je takođe urađeno u realnom okruženju i nakon uspešne komunikacije između posrednika (Ubidots portala) i klijenta (NodeMCU-a sa senzorom) sistem je spreman za puštanje u rad (Slika 7.).

VIII. Zaključak

Pre Internet of Things (IoT) tehnologije, za potrebe automatizacije, koristili su se sistemi industrijske automatizacije koje čine skupa oprema i zahtevaju velika znanja iz tih oblasti. Razvojem tehnologije granica se polako pomera jer je hardver sve jeftiniji, a informacije su lako dostupne što je približilo sisteme automatizacije ne samo profesionalnom industrijskom okruženju već i amaterskom hobi osmišljavanju projekata.

Cilj ovog rada bio je postići upravljivo i vremenski štedljivo okruženje koje će nadzor i kontrolu približiti ljudima. Mikrokontroleri sa ugrađenom bežičnom komunikacijom omogućili su ugradnju sistema na postojeće instalacije uz iziskivanje minimalnih prilagođavanja. Celokupni sistem je smešten na lokalnom nivou. Lokalna bežična mreža je jedina spona koju sistem ima da bi se iskoristio njegov puni potencijal ali su ugrađeni algoritmi koji manipulišu procesima u slučaju prekida veze kako bi se sprečila nemogućnost izvršenja funkcionalnosti.

Dalji razvoj, nova saznanja zatim investicije u svim nabrojenim područijima tehnologije meteorološke stanice i uopšteno Internet of Things tehnologije učiniće pristupačnim i široko rasprostranjenim. Vrlo verovatno će ta tehnologija kroz neko vreme biti asistent u svakodnevnom životu.



Slika 6. Ilustrovani prikaz povezivanja NodeMCU i relejnog modula na proto ploči.



Slika 7. Prikaz temperature i vlažnosti vazduha na Ubidots portal.

Literatura

[1] Brad Schechter the-evolution-of-iot 07.2017.
[2] Brad Schechter the-evolution-of-iot 07.2017.
[3] MQTT_UserManual_en.pdf
[4] NODEMCU_DEVKIT_V1.0.PDF , 12.05.2015.
[5] Adafruit Industries – DHTxx datasheet.pdf
[6] SRD-05VDC-SL-C-Datasheet.pdf
[7] https://ubidots.com/education/
[8] https://mosquitto.org/
[9] ESP8266EX Datasheet, Espressif 2017.
[10] https://iotbytes.wordpress.com/nodemcu-pinout/ dostupan 27.5.2017.
[11] http://fritzing.org/home/

API Benchmark

Mladen Radivojević

Abstract - This paper gives an example and instructions for setting up and performing API Benchmark for single or multiple APIs. Project is publicly available on Github: https://github.com/mladcepes/apiBenchmark.git. Instructions are available in README.md file.

I. API

API (Application programming interface) is an interface or communication protocol between a client and a server [1]. It can be described as a contract between the client and the server: When client sends a request in a specific format, it will always get the response in the specific format from the server, or initiate specific actions.

II. API BENCHMARK

API Benchmark [2] is a powerful Node.js tool [3] that measures and compares performances of single or multiple APIs. Reports are generated and stored on user's local machine in .html format. API Benchmark tool is inspired by BenchmarkJs [6]. It can be installed on your local machine by using npm (node package manager) [7] by executing: *npm install api-benchmark* in terminal (Only requirement: User must have Node.js [3] installed on his local machine). Given API performance testing can be performed by measuring average response time for multiple API routes (endpoints). Response time is defined as the time it takes the server to respond to a client request (measured in seconds or milliseconds). For this paper purposes, the hosted REST-API public service called 'Reqres'[4] will be used. It consist out of following routes:

- List users: GET api/users (Returns list of users as response)
- List single user: GET api/users/id (Returns single user with the given id)
- List resources: GET api/unknown (Returns list of resources)
- List single resource: GET api/unknown/id (Returns single resource with the given id)
- Create new user: POST api /users (Creates new user with the given data).

Two types of API benchmark will be performed for these 5 routes and the results will be compared. First, benchmark will be performed with 50 samples (requests), with the maximum time of 60 seconds. These requests will be sent one by one. Options are defined in reqresApiBenchmark.js file: *minSamples*: 50 (minimum number of requests), *maxTime*: 60 (maximum benchmark time duration), *debug*: true.

Second, benchmark will be performed with 50 samples with the maximum time of 60 seconds, but the requests will be sent parallel. Options: *minSamples*: 50, *maxTime*: 60, *runMode*: parallel (default is serial, so this needs to be set to parallel), *maxConcurrentRequests*: 15 (maximum number of concurrent requests). Benchmark is executed with the following command: node test/ reqresApiBenchmark.js. Results of API benchmark are imported into the following tables:

TABLE I
BENCHMARK REPORT - RESULTS OF SERIAL API BENCHMARK

Route	Avg. response time
List Users	0.115 s
List Single user	0.112 s
List Resources	0.108 s
List Single resource	0.111 s
Create new user	0.170 s

TABLE II
BENCHMARK REPORT - RESULTS OF PARALLEL API BENCHMARK

Route	Avg. response time
List Users	0.168 s
List Single user	0.136 s
List Resources	0.128 s
List Single resource	0.184 s
Create new user	0.210 s

Route column represents the endpoints benchmark was performed on, and the Avg. response time column represents the average response time for the given routes measured in seconds. First table represents the results of benchmark with serial requests. Response time is quite impressive (note that the data is mocked and does not represent the real state). Regarding real server's response time, time under 0.3 s is considered as a good server response time. If server response time is greater than 1.3 s, it can be considered as very slow and underperforming.

III. RESULTS ANALYSIS

After performing serial and parallel benchmark, results are stored and compared. Results graph (generated with online chart tool [5]):

Mladen Radivojević is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: mladcepes@gmail.com

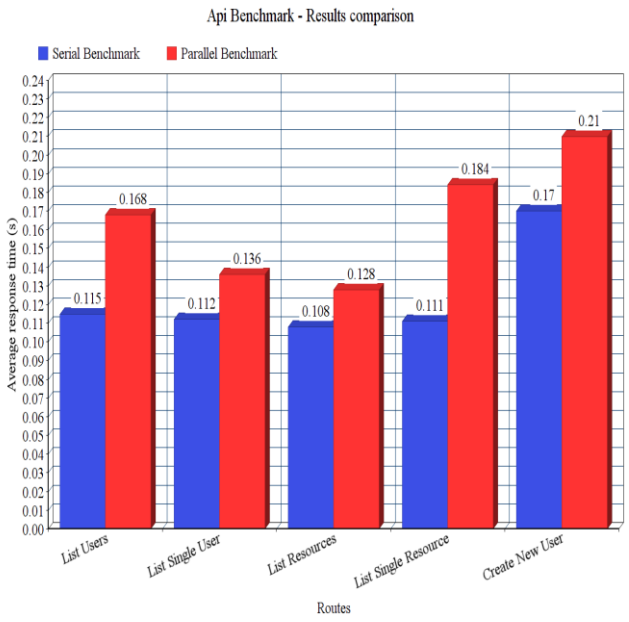


Figure 1. Api Benchmark results comparison.

Y axis represents average response time in milliseconds. Blue column represents serial benchmark routes, while the red column represents parallel benchmark routes. Reqres [2] public service can be considered as a server with a good response time. Average response time during both benchmarks never exceed 0.3 seconds (Refer to Table I and Table II results). Serial Benchmark conclusions:

- Route with the lowest response time : List resources (0.108 s).
- Route with the highest response time: Create new user (0.170 s).
- Average route response time: 0.1232 s

Parallel benchmark conclusions:

- Route with the lowest response time: List resources (0.128 s).
- Route with the highest response time: Create new user (0.210 s).
- Average route response time: 0.1652s.

The route with the lowest performance is: *Create new user* (refer to figure 1). Post methods have higher response time, due that the database insertion takes longer than just getting the list of data. Generally, GET methods should always have higher performances than other methods like POST, PUT, PATCH, etc. The route with the highest performance is: *List resources* (refer to figure 1).

The example of generated report for single route (*List users*) in .html format is given on the figure 2:



Figure 2. Example of generated report for single route.

Y axis represents response time in seconds, the X axis represents number of samples. It can be noted that there was the peak at the start of benchmark, the third sample had the response time of 0.489 seconds. Rest of the samples response time was around average (green line).

IV. Conclusion

By comparing the results, it can be determined that the server has slightly better performance while it is handling serial requests. When the server is handling parallel requests, response time is slightly increasing. This represents real state since the servers are handling parallel requests from multiple users. Server performance is important because it determines the feedback from users. API benchmark is mostly used by Quality assurance engineers for performing non-functional tests. Regarding Reqres[4] service, its performances can be noted as very high and it can be used as an example of desired API benchmark results. Due that the response time is increasing when server is handling parallel requests, its performance can be improved but the priority is not high since the average response time is less than 0.3 s even for the worst operating route.

References

[1] Braunstein, Mark L. (26 July 2018). Health Informatics on FHIR: How HL7's New API is Transforming Healthcare.
[2] Node package manager Api-Benchmark tool documentation: https://www.npmjs.com/package/api-benchmark.
[3] Node js documentation: https://nodejs.org/en/docs/.
[4] Reqres API public service: https://reqres.in/
[5] Online chart tool: https://www.onlinecharttool.com
[6] Benchmark JS: https://benchmarkjs.com/
[7] Node package manager: https://docs.npmjs.com/about-npm/

Laserski daljinomer sa opcionim merenjem nivoa nagiba i temperature

Stefan Lazarević

Sadržaj – U ovom radu opisan je laserski daljinomer sa opcionim merenjem nivoa nagiba i temperature. Nakon opisa uređaja i realizacije električne šeme, obrazložen je proces projektovanja štampane ploče i izrade 3D modela uređaja.

I. UVOD

Uređaj koji je tema ovog rada, odlikuje se ne samo sposobnošću merenja udaljenosti određenog objekta, već i određivanjem nivoa nagiba predmeta, kao i mogućnošću merenja ambijentalne temperature.

Jedan od primera, gde ovakva vrsta uređaja nalazi svoju primenu, a ujedno i najviše dolazi do izražaja, jeste kao „alat“ za izvođače raznoraznih građevinskih radova. Veliku prednost ovakvog uređaja, čini to što u malom, lako prenosivom pakovanju kombinuje tri veoma bitna i često korišćena uređaja u ovoj grani industrije, odnosno predstavlja zamenu za tradicionalni daljinomer, libelu i termometar (Slika 1).

Uređaji ovakvog tipa su veoma precizni i jednostavni za upotrebu. Poseduju karakterističnu izdržljivost, neophodnu za rad u nepredvidljivim građevinskim uslovima, kao i mogućnost pregleda istorije izvršenih merenja, koja je u svakom trenutku dostupna.

II. ELEKTRIČNA ŠEMA LASERSKOG DALJINOMERA

U ovom odeljku biće prikazana realizacija električne šeme (Slika 2), uz objašnjenje funkcionalnosti samog uređaja. Za izradu šeme sa Slike 2, upotrebljene su sledeće komponente:

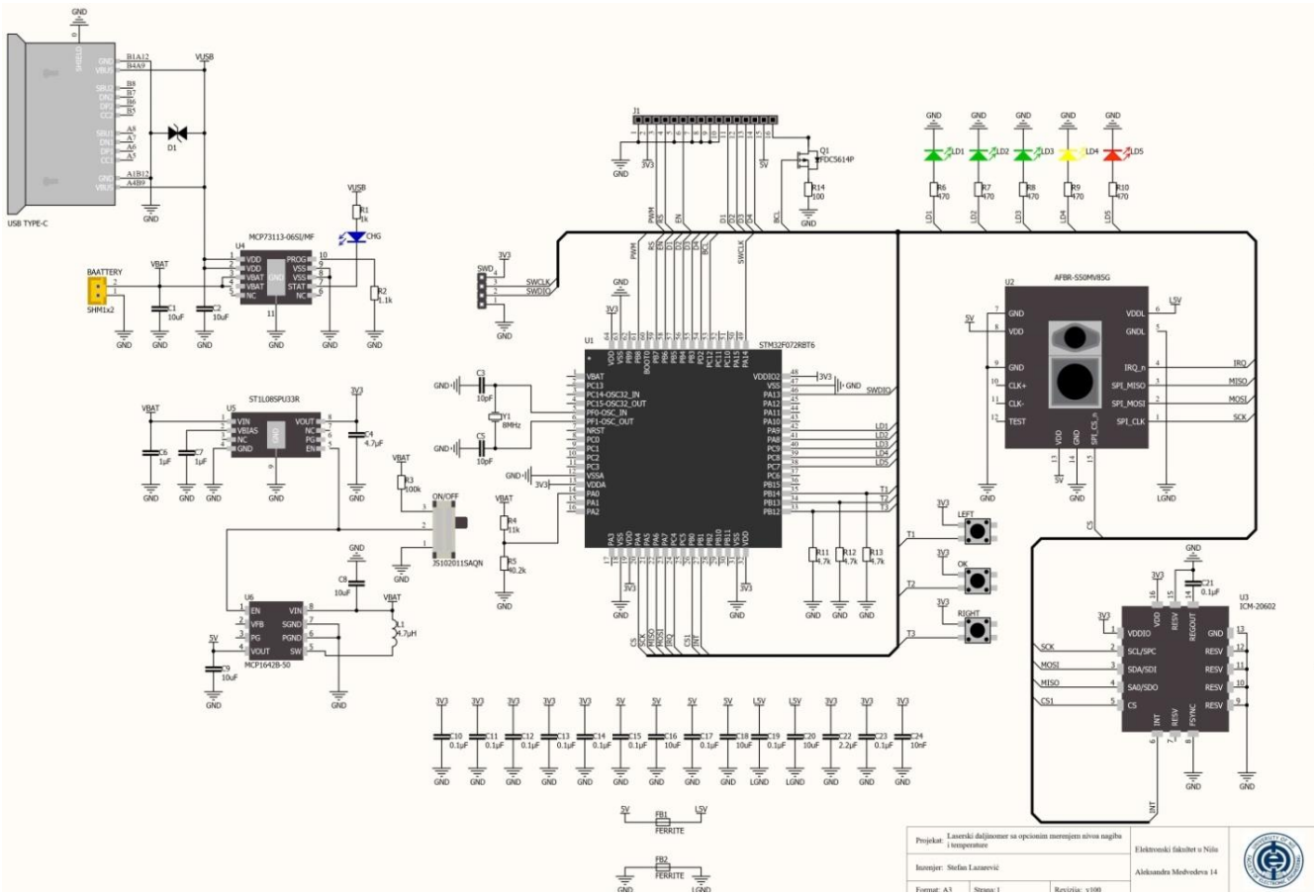
1. Mikrokontroler U1 (STM32F072RBT6) [1]
2. Laserski senzor za merenje daljine U2 (AFBR-S50MV85G) [2]
3. Žiroskop i temperaturni senzor U3 (ICM-20602) [3]
4. USB Type-C
5. Konektor za displej (J1)
6. Konektor za bateriju (SHM1x2)
7. Prekidač za uključivanje, odnosno isključivanje uređaja (JS102011SAQN)
8. Kontrolni tasteri (LEFT/OK/RIGHT)
9. Integrisano kolo za punjenje Li-ion baterije U4 (MCP73113-06SI/MF) [5]
10. Linearni regulator napona U5 (ST1L08SPU33R) [6]
11. Step up (Boost) konvertor U6 (MCP1642B-50) [7]
12. LED diode, kao indikatori nivoa napunjenosti baterije (LD1 - LD5)
13. SWD konektor za programiranje mikrokontrolera



Slika 1. Primer pakovanja laserskog daljinomera sa opcionim merenjem nivoa nagiba i temperature, kao ekvivalent tradicionalnim uređajima.

Uređaj je realizovan tako da se može napajati upotrebom Li-ion baterije, koja se povezuje preko SHM1x2 konektora. Status nivo napunjenosti baterije, korisniku je dostupan u vidu 5 dioda. Korisniku je uređaj na raspolaganju čak i tokom punjenja baterije. USB TYPE-C konektor, nalazi se na samom ulazu integrisanog kola za punjenje baterije U4 (MCP73113-06SI/MF) [5]. CHG dioda je pokazatelj da se baterija puni. Kondenzatori C1 na izlazu i C2 na ulazu ovog integrisanog kola, koriste se kako bi održavali stabilan napon. Baterija je napunjena kada njen napon dostigne 4,2 V, a prazna je ukoliko napon padne ispod 3,6 V.

Stefan Lazarević, Katedra za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: lazarevic.stefan@elfak.rs



Slika 2. Električna šema laserskog daljinomera sa opcionim merenjem nivoa nagiba i temperature.

Upotrebom linearnog regulatora napona U5 (ST1L08SPU33R) [6], koji ima širok opseg ulaznih napona, a čiji je izlaz 3,3 V i *Step up*, odnosno *Boost* konvertora, U6 (MCP1642B-50) [7], čiji je izlazni napon 5 V, napajaju se komponente uređaja sa neophodnim odgovarajućim naponom.

Prekidač za uključivanje, odnosno isključivanje uređaja (JS102011SAQN), realizovan je na takav način da se ne vrši fizički prekid električnog kola, već se digitalno omogućuje, odnosno onemogućuje rad istog.

Upotrebom kontrolnih tastera (LEFT/OK/RIGHT), zadaju se željene komande, koje mikrokontroler (STM32F072RBT6) [1] detektuje, a na osnovu kojih se izvršava set instrukcija, prethodno zadatog programskog koda. Izvršavanjem istog, aktivira se neki od dostupnih senzora, u ovom slučaju: laserski senzor za merenje daljine (AFBR-S50MV85G) [2], žiroskop ili senzor temperature, pri čemu su žiroskop i temperaturni senzor smešteni unutar ICM-20602 integrisanog kola [3]. Nakon izvršenog merenja, dobijene povratne informacije nekog od senzora, konvertuju se u pogodan oblik i prosleđuju displej modulu.

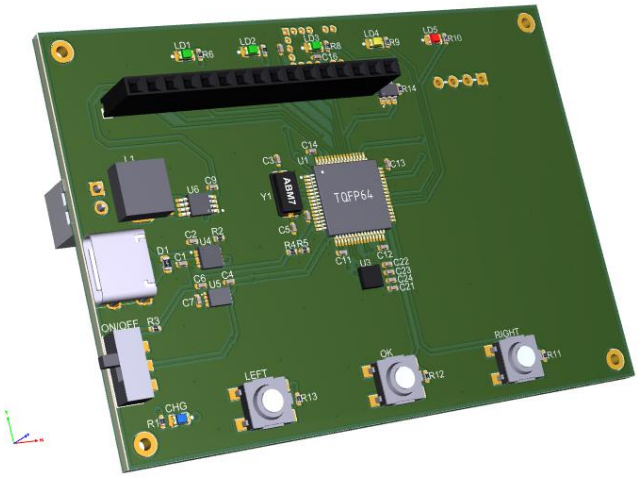
III. REALIZACIJA ŠTAMPANE PLOČE I 3D MODELA

Električna šema (Slika 2), štampana ploča (PCB), čiji su svi kreirani slojevi detaljno prikazani na Slici 5, kao i 3D model štampane ploče (Slika 3 i Slika 4), realizovani su u *Altium Designer*-u [8], programu za razvoj štampanih ploča. *Altium Designer* je jedan od najzastupljenijih alata ove vrste. Karakterišu ga brzina i efikasnost izrade složenih sklopova. Sadrži veliki broj biblioteka u sklopu kojih se nalaze 3D modeli, simboli i otisci različitih vrsta komponenata, a takođe i podržava mogućnost izrade istih.

Altium Designer pojednostavljuje proces proizvodnje, generisanjem *gerber* fajlova, koji sadrže sve neophodne informacije potrebne za izradu štampane ploče. *Gerber* fajlovi predstavljaju skup fajlova, od kojih je svaki pojedinačni fajl deo zasebnog procesa proizvodnje:

- Označavaju mesto gde će štampana ploča biti izrezana (*Keepout*)
- Fajlovi koji određuju gde je neophodno ukloniti bakar, kako bi se dobile željene električne veze na gornjem i donjem sloju štampane ploče

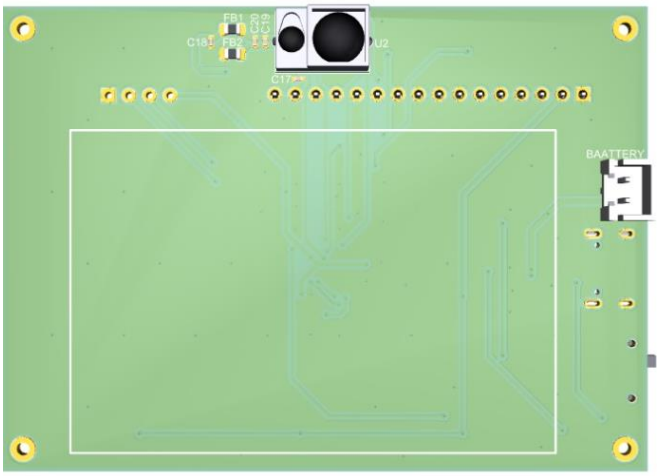
- Mesta sa kojih treba ukloniti *solder* masku na gornjem i donjem sloju štampane ploče, gde će kasnije biti zalemljeni *pad*-ovi komponenata
- Set instrukcija za CNC mašinu, koji definišu gde i koje veličine rupa treba izbušiti na štampanoj ploči
- Fajl za izradu *stencil*-a, odnosno pozicija na kojima kasnije treba naneti pastu za lemljenje
- Izrada *silk*-a, to jest oznaka komponenata, kako bi komponente ispravno bile postavljene na gornjem i donjem sloju štampane ploče



Slika 3. 3D model štampane ploče (prednja strana).

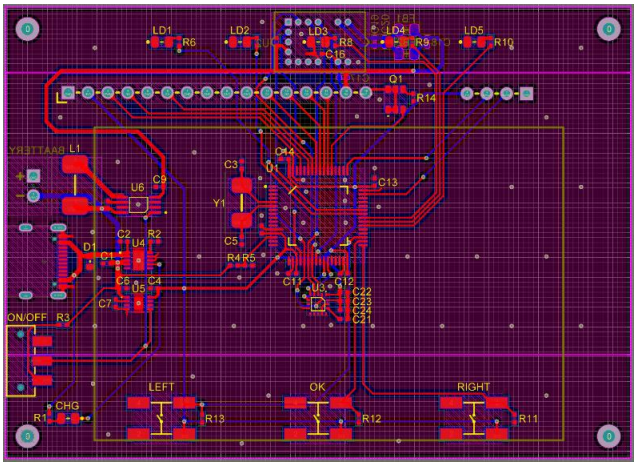
3D model svake komponente je napravljen kada i otisak komponente, nakon čega su i zajedno uvezeni u biblioteku sa komponentama koje su korišćene za izradu uređaja. Sam oblik štampane ploče, kao i njena debljina i debljina slojeva, kreirani su u samom projektu, dok su komponente naknadno postavljene.

Upotreba 3D modela omogućava jednostavan razmeštaj komponenata na štampanoj ploči, time sprečavajući njihovo preklapanje.



Slika 4. 3D model štampane ploče (zadnja strana).

Štampana ploča laserskog daljinomera sa opcionim merenjem nivoa nagiba i temperature je izrađena u 2 sloja, između kojih se nalazi jezgro napravljeno od kompozitnog FR4 materijala (*Fiberglas* u kombinaciji sa vezivnim materijalom od epoksidne smole, koja nije zapaljiva), koji se koristi u preko 95% slučajeva. Dimenzije štampane ploče su 8 x 6 cm, pri čemu je *keepout* 0,25 mm.



Slika 5. 2D prikaz svih slojeva.

IV. Zaključak

U sklopu ovog rada izneta je jedna od ideja, kako se upotrebom savremene tehnologije može izraditi multifunkcionalni uređaj, čijim bi se korišćenjem olakšao rad zaposlenima u građevinskoj industriji. Šematskim prikazom, uz kreiran model štampane ploče u *Altium Designer*-u, obrazložena je realizacija hardverskog dela laserskog daljinomera sa opcionim merenjem nivoa nagiba i temperature.

Zahvalnica

Ovaj rad predstavlja deo diplomskog rada autora, izrađen na Katedri za mikroelektroniku.

Literatura

[1] Mikrokontroler - STM32F072RBT6, dostupno na: https://www.st.com/resource/en/datasheet/stm32f072rb.pdf
[2] Laserski senzor za merenje daljine - AFBR-S50MV85G, dostupno na: https://docs.broadcom.com/docs/AFBR-S50MV85G-DS102
[3] Žiroskop i temperaturni senzor - ICM-20602, dostupno na: http://www.invensense.com/wp-content/uploads/2016/10/DS-000176-ICM-20602-v1.0.pdf
[4] Displej modul – LCD16x2, dostupno na: http://www.eigenstudium.de/fileadmin/user_upload/Download/Datenblaetter/LCD-1602A.pdf
[5] Integrisano kolo za punjenje Li-ion baterije - MCP73113-06SI/MF, dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/20002183E.pdf

[6] Linearni regulator napona - ST1L08SPU33R, dostupno na: https://www.st.com/resource/en/datasheet/st1l08.pdf
[7] Step up (Boost) konvertor - MCP1642B-50, dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/20005253A.pdf
[8] Altium Designer, dostupno na: https://www.altium.com

Take Care – Sistem pametnih uređaja za brigu o zdravlju starijih osoba

Marko Ilić, Vojislav Dimitrijević, Nenad Đorđević, Dušan Kovačević, Stefan Đokić, Milan Ivković

Sadržaj - U Srbiji u 2018. godini popijeno je više od 5 miliona kutija lekova. Veliki broj lekova koja prosečna osoba popije može da varira od 4 do 14 lekova dnevno, pogotovu kod starijih ljudi [1]. Ovim sistemom pametnih uređaja autori žele pomoći sugrađanima da se sa problemom zaboravljanja uzimanja terapije izbore i olakašaju im život uz pomoć pristupačnih najnovijih tehnologija.

I. UVOD

A. *O nastanku aplikacije*

Aplikacija Take Care je originalno delo autora, dok je ideja o njoj nastala marta 2019. godine, tokom takmičenja „EESTech Challenge“ na Elektronskom fakultetu u Nišu. Motivacija za izradu ove aplikacije potiče iz svakodnevnog života i priče da je bliska osoba jednog od autora doživela problem zaboravljanja uzimanja terapije a posledica toga bila je da je ovakav ishod mogao biti fatalan. Želja i misija autora je da Take Care sistem bude regularan deo svakodnevnog života porodica koje uzimaju neku vrstu terapije i da samim tim imaju poboljšan kvalitet života.

B. *Problem*

Kako su lekovi preuzeli važnu ulogu u svakodnevnom životu, proteklih je godina došlo do porasta broja slučajeva zanemarivanja medicinskih slučajeva vezanih za pogrešne lekove koji se daju pacijentima, kao što je slučaj medicinske sestre koja je pacijentu dala lek paralizator umesto antacida koji je propisao lekar, izazvavši smrt pacijenta. Posle toliko mnogo slučajeva, neophodno je da se pravi lek obezbedi pravoj osobi, u suprotnom, uzimanje pogrešne ili ne uzimanjem terapije, pacijent može biti izložen opasnim situacijama, od blagog poremećaja zdravlja do smrti [3].

Greške u konzumaciji lekova uzrokuju najmanje jednu smrt svaki dan i povrede oko 1,3 miliona ljudi godišnje samo u Sjedinjenim Američkim Državama. Globalno gledano, troškovi povezani sa greškama uzimanja pogrešnih lekova procenjuju se na 42 milijarde američkih dolara godišnje, što je skoro 1% ukupnih svetskih zdravstvenih troškova [2].

Mnoge medicinske greške nastaju zbog činjenice da se ljudi zaduženi za lekove pacijenta ili starijeg stanovništva moraju baviti sortiranjem ogromnih količina tableta svaki dan.

Bez obzira na uzrok, dokazano je da postoji značajan rizik da će ljudi popiti pogrešan lek ili dozu.

II. PRINCIP RADA

A. *Sistem pametnih uređaja*

Gore navedeni problem želimo rešiti implementiranjem sistema u vidu pametne kuće koji će voditi računa o starijim, senilim i osobama sa invaliditetom, i to uz pomoć uređaja najnovijih tehnologija. Sistem čine mnoštvo uređaja poput:

- Uređaj za detekciju ulaska i izlaska iz kuće;
- Uređaj za daljinsku kontrolu kućnih potrošača kao detekciju potencijalne opasnosti od istih;
- Pametna kutija za lekove koja na intuitivni način podseća osobu da popije lek;
- Pametan sat za praćenje i vitalnih znakova osobe i obaveštavanje korisnika;
- Mobilna aplikacija za korisnika koja prati stanje kako starije osobe tako i okruženja u kojem osoba živi.

B. *Pametna kutija za lekove*

Dispenser lekova „TakeCare“ fokusiran je na pacijente koji često uzimaju lekove ili vitaminske suplemente, ili na medicinske pomoćnike koji se bave starijim osobama ili bolesnicima. Naš dozer lekova može se programirati putem mobilne i web aplikacije koja omogućava negovateljima ili klijentima da odrede količinu tableta i vreme uzimanja tableta. Dispenser lekova može da proizvodi svetlosne i zvučne signale koji će podsetiti korisnika da uzme lek na vreme. Takođe, aplikacija obaveštava negovatelja da li je osoba popila lek. Ako lek nije uzet, šalje obaveštenje upozorenja kako bi negovatelj mogao da deluje na vreme.

Nenad Đorđević, Marko Ilić, Vojislav Dimitrijević, Dušan Kovačević, Stefan Đokić i Milan Ivković su studenti Elektronskog fakulteta u Nišu, članovi Studentskog inovacionog centra Elektronskog Fakulteta (SICEF) kao i osnivači startap-a TakeCare, Elektronski fakultet, Aleksandra Medvedeva 14, 18000 Niš, Srbija. Email: djordjevic.nenad@elfak.rs, ilic.marko@elfak.rs, vojislav.dimitrijevic@elfak.rs, dusan.kovacevic@elfak.rs, stefan.djokic@elfak.rs, milan.ivkovic@elfak.rs



Slika 1. Prototip - Pametna kutija za lekove.

C. Pametna narukvica

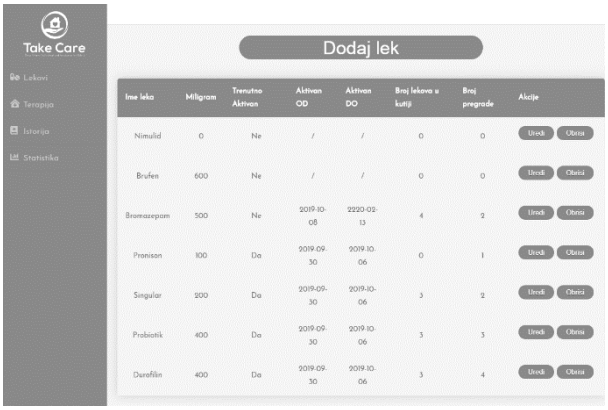
Pametna narukvica je kao poput pametnih satova koji trenutno postoje na tržištu. Ukoliko osoba koja treba da uzme lekove ne čuje ili ima oštećen vid, osetiće vibraciju koja će ukazivati da je vreme za lek. Takođe narukvica omogućuje GPS praćenje ukoliko je u pitanju dementna osoba i ukoliko izađe iz kuće ili stana i izgubi se. Tada se javlja negovatelju da je osoba van kuće ili ako je duži vremenski period van kuće čime javlja tačnu adresu osobe.

D. Pametna utičnica

Dosta ima slučajeva gde pored starijih osoba i mlađi naraštaji zaboravljau da isključe uređaje poput rerni, šporeta i pegli. Ovakvi uređaji se preko pametne utičnice lako mogu kontrolisati preko mobilne ili web aplikacije. Mogu se programirati kada da se uključe/isključe i da obaveštavaju kada se neki uređaj uključi ili isključi.

E. Mobilna i web aplikacija

Mobilna aplikacija koja je povezana sa kutijom preko IoT (Internet of Things) sistema obaveštava negovatelja da li je osoba popila lek u zakazano vreme. Ako lek nije uzet, šalje obaveštenje upozorenja kako bi negovatelj mogao da deluje na vreme. Aplikacija ima mogućnost dodavanja novog leka, brisanja, kao i modifikaciju postojećeg leka.

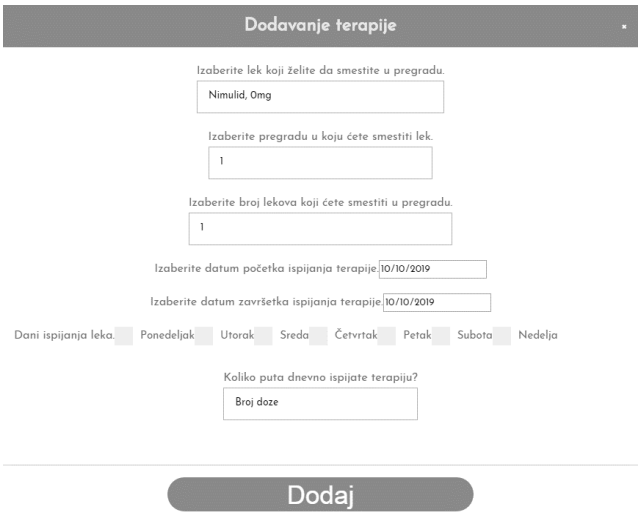


Slika 2. Ekran aplikacije – Lekovi

Još jedna mogućnost mobilne/web aplikacije jeste dodavanje terapije na nedeljnom nivou. Postoji mogućnost i uređivanja i brisanja postojeće terapije. Aplikacija prati celokupnu aktivnost korisnika kutije i celu istoriju smešta na cloud server odakle direktno korisnik može videti da li su, kada i koji lekovi popijeni. Na ovaj način se kreira statistika uzimanja lekova, tj. vrši se personalizacija kutije, što omogućava doktorima da imaju uvid u vođenju terapije. Da bi ovo bilo olakšano, moguće je da korisnik unese oboljenje od kojeg boluje i da na taj način pomogne i sistemu i doktoru u praćenju terapije.



Slika 3. Ekran aplikacije – Terapija.



Slika 4. Funkcionalnost aplikacije – Dodavanje terapije.

III. ZAKLJUČAK

Široka dostupnost, kao i brz razvoj tehnologije omogućava brigu o starijima i kada niste pored njih. Samim tim, stariji će dobiti veći osećaj samostalnosti, a staratelji će imati osećaj sigurnosti i rastećenja da će starija osoba biti u redu.

"Take Care" je namenjen za kućnu upotrebu, kao i upotrebu gerontoloških centara i bolnica. Cilj autora je da sistem bude pristupačan svim slojevima društva.

ZAHVALNICA

Autori projekta se ovom prilikom posebno zahvaljuju organizaciji Razlivalište koja je prepoznala potencijal ovog tima i same ideje i time im omogućila tromesečnu inkubaciju za razvoj ideje na takmičenju Social Impact Award Srbija. Ovaj tim je u finalu osvojio glavnu nagradu od 1500 € i time nastavio razvoj svoje ideje [4].

Takođe zahvalnost se upućuje i Startap centru na Elektronskom fakultetu u Nišu koji je omogućio besplatno korišćenje svojih prostorija u narednih 8 meseci.

Veliku zahvalnost autori duguju i mentoru Vladimiru Gojkoviću uz čiju pomoć je osvojena novčana nagrada na gorenavedenom takmičenju.

LITERATURA

[1] 021.rs, Dostupno na: https://www.021.rs/story/Info/Srbija/221205/Gradjani-Srbije-u-prosloj-godini-popili-55-miliona-kutija-lekova-za-smirenje.html

[2] ncbi.nlm.nih.gov, Dostupno na: https://www.ncbi.nlm.nih.gov/books/NBK519065/

[3] C. Crema, A. Depari, A. Flammini, M. Lavarini, E. Sisinni, Medical measurements and applications (MeMeA), IEEE International Symposium 2015 at Torino, Italy

[4] serbia.socialimpactaward.net, Dostupno na: https://serbia.socialimpactaward.net/sia-2019-ko-su-najbolji-mladi-drustveni-inovatori-srbije/

Modularne laboratorijske vežbe -ModuleClick-

Milan Ivković

Sadržaj – U ovom radu će biti predstavljen način izvođenja laboratorijskih vežbi korišćenjem personalnih računara. Ideja je da se na računaru prati laboratorijska vežba umesto iz praktikuma, gde se mogu meriti naponi, struje i generisati željeni talasni oblici signala. Dobijeni grafici, rezultati i komentari se konvertuju u PDF format i prosledjuju profesoru putem email-a.

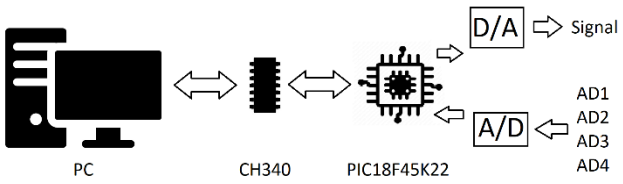
I. UVOD

Živimo u modernom dobu u kome se tehnologija naglo razvija. Takav nagli razvoj utiče kako na naše društvo tako i na okolinu koja nas okružuje. Danas smo u situaciji da svako pored mobilnog uređaja poseduje i personalni računar čime se otvara novi način skladištenja i razmene informacija. Da bi smo upotrebili tehnologiju na što bolji način treba gledati da dosadašnje navike zamenimo novim koje će pozitivno uticati na životnu sredinu oko nas. Jedan od načina je upotreba personalnih računara tokom laboratorijskih vežbi umesto praktikuma čime štedimo papir i čuvamo šume.

Laboratorijske vežbe se izvode tako što se prenosni računar povezuje sa glavnom pločicom ModuleClick koja na sebi sadrži mikrokontroler koji je zadužen za komunikaciju, merenje i generisanje željenog signala. Glavna pločica ima konektore putem koji se povezuje ModuleClick Board za datu vežbu. ModuleClick Board je maketa za datu vežbu samo što se ne koriste univerzalni merači napona i struje i generatori funkcija nego računar. U računaru je prilagođen softver studentima putem koga pristupaju PDF verziji za date laboratorijske vežbe, koja poseduje opcije za generisanje željenog signala sa sledećim parametrima, sinusuidalni signal, trougaoni signal, testerasti signal, impulsni signal, veličina amplitude, ofset kao i frekvencija signala. Zatim prikaz bitnih napona i struja sa makete. Dobijeni grafici, vrednosti napona/struje kao i komentari, tj. zapažanja studenta, se šalju u vidu PDF izveštaja profesoru na email čemu će profesor moći da oceni da li je student ovladao potrebnim znanjem za datu laboratorijsku vežbu. Ukoliko se radi druga laboratorijska vežba potrebno je zameniti ModuleClick Board odgovarajućom novom maketom kao i odabrati u softveru koja se vežba radi.

II. RAZVOJ

Glavna pločica ModuleClick koristi mikrokontroler PIC18F45K22 [1] koji preko kola CH340 [2] ostvaruje komunikaciju sa računarom putm USB porta. Za generisanje talasnog oblika koristi se D/A konvertor sa R2R [3] otpornom mrežom koji u 255 koraka reprezentuje željeni oblik signala tj. 8-bitni je. Kako se ne bi ugrozila otporna mreža na izlazu R2R mreže se koristi naponski bafer LM358 [4]. Tako dobijeni signal se dovodi na ulaz operacionog pojačavača sa dualnim napajanjem pojačanja A=1 kojim se obezbeđuje pozitivna i negativna polupriodu signala. Da bi se signal kontrolisao od strane računara posle operacionog pojačavača je postavljen još jedan operacioni pojačavač čije se pojačanje kotroliše preko digitalnog potenciometra MCP41010 [5]. Ofset signala se takođe koristi preko digitalnog potenciometra tako što se menja referentni napon na pozitivnom ulazu operacionog pojačavača. Dijagram modularnih lab vežbi je prikazan na slici 1.



Slika 1. Dijagram modulnih lab vežbi.

A. Generisanje signala

Za generisanje signala nije korišćena D/A konverzija samog mikrokontrolera zbog toga što poseduje 32 koraka koji u poređenju sa 255 koraka R2R D/A ne daje željenu preciznost. Sinusni signal se kreira upotrebom funkcije sin(rad) u programu MikroC [6]. Pretvaranjem stepena u radijane se dobija mogućnost korišćenja brojača koji se inkrementira od 0 do 360 za korak 1. Vrednost koju PORTB dobija je pokazana jednočinom (1):

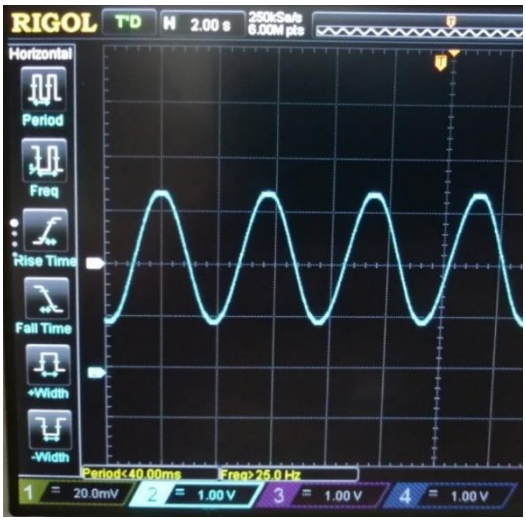
$$PORTB = 125 + 125 * \sin(\frac{i*\pi}{180}) \quad (1)$$

Registaru PORTB je postavljen ofset sa brojem 125 kako bi pravilno iscrtavao obe poluperiode, dok funkcija sin() daje vrednost od -1 do 1 u 360 koraka čime se dobija pravilan izgled sinosuide kao sa slike 2.

Frekvencija signala se može promeniti na dva načina. Prvi način je inkrementiranje vrednosti „i“ iz jednačine 1, dok je drugi način promena vremena za koju će se vrednost „i“ inkrementirati. Vreme inkrementiranja je definisano vremenom TMR0 [7] tajmerom koji svakim okidanjem odlazi u interapt i povećava vrednost ili resetuje vrednost kada dostigne 360. Veličina amplitude se menja putem

Milan Ivković student doktorskih studija, Nanotehnologije i mikrosistemi, Elektonski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milan.ivkovic@elfak.rs.

operacionog pojačavača čime se digitalnim potenciometrom određuje da li će biti pojačanje ili slabljenje ulaznog signala.



Slika 2. Izgled generisane sinusoide.

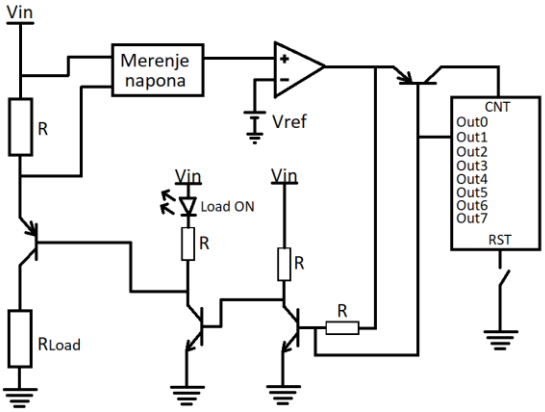
Generisanje testerastog signala se postiže običnom for petljom koja PORTB inkrementira od 0 do 255 i nakon postizanja maksimalne vrednosti se vraća na nulu. Testerasti signal je podeljen na tri dela. Prvi deo počinje od vrednosti 125 pa do 255, drugi deo je od 255 do 0 i zatim treći deo od 0 do 125. Impulsni signal se prosto generiše tako što je do pola periode tj. vrednosti brojača „i“ vrednost registra 0, a nakon toga 255.

B. Interrupt

Tokom razvoja došlo je do potrebe za inkrementiranjem određene varijable sve do 10000 kako bi se testirao signal zadate frekvencije. S obzirom da je tajmer TMR0 bio podešen da odlazi u interapt na svakih 50uS cilj je bio da se dobije zadrška od 0.5s. Testiranjem je utvrđeno da vreme nije dobro, i da se ne pravi zadrška od 0.5s. Dubljim razmatranjem je došlo do zaključka da na svaka 4-5 ulaska u interapt se pravi greška od 2.5us, što znači da će 4. ili 5. put ući u interapt za 52.5uS i tako dok se stigne do 10’000 imati grešku, grubo računato, od 5000us. Gledano na traženu vrednost to je 0.505s umesto 0.5s. Greška koja se stvara jeste mala ali nakon nekoliko iteracija željena zadrška više nije onolika kolika je tražena. Problem je u vremenu ulaska u interapt jer mikrokontroleru treba određeno vreme da sačuva stanje iz kojeg se prekida da bi izvršio kod u interaptu i zatim se vratio i nastavio izvršenje koda. Umesto korišćenja registra TMR0 koji je 8-bitni korišćen je 16-bitni tajmer TMR1[8] za ove svrhe čime se samo jednom ulazi u interapt i rešava problem kreiranja greške.

C. Elektronska zaštita kola

Pod elektronskom zaštitom kola se podrazumeva zaštita od prevelike struje koju zahteva maketa. S obzirom da se kolo napaja preko USB-a računara ne bi bilo poželjno da se izazove kratak spoj u kolu ili postavi preveliko opterećenje i time uništi USB port. Dijagram zaštite je dat na slici 3.



Slika 3. Dijagram zaštite kola od prevelike struje.

Zaštita funkcioniše tako što se instrumentacionim pojačavačem meri pad napona na otporniku od 1Ω. Kada kroz otpornik prolazi 500mA pad napona na otporniku će biti 500mV i to je maksimalna vrednost struje koja sme prolaziti kroz potrošač zbog samog limita USB-a. Izmereni napon se dovodi na drugi operacioni pojačavač čime se upoređuje vrednost sa zadatom vrednošću i ukoliko je vrednost veća uključuje se zaštita i isključuje potrošač. Odmah nakon isključenja mereni napon će biti ispod zadate vrednosti čime se opet stavaraju uslovi za isključenje zaštite i uključenje potrošača. Pošto je potrošač još uvek uključen mereni napon ponovo dostiže svoj limit i tako se stvari odvijaju u krug sve dok neka komponenta ne popusti. Da bi se sprečio ovakav slučaj treba dodati deo da se stanje aktiviranje zaštite zadržava sve dok korisnik ne pritisne dugme. Za takvu potrebu iskorišćen je brojač CD4017 [9] koji kada jednom biva aktiviran drži uključenu zaštitu sve dok se ne resetuje.

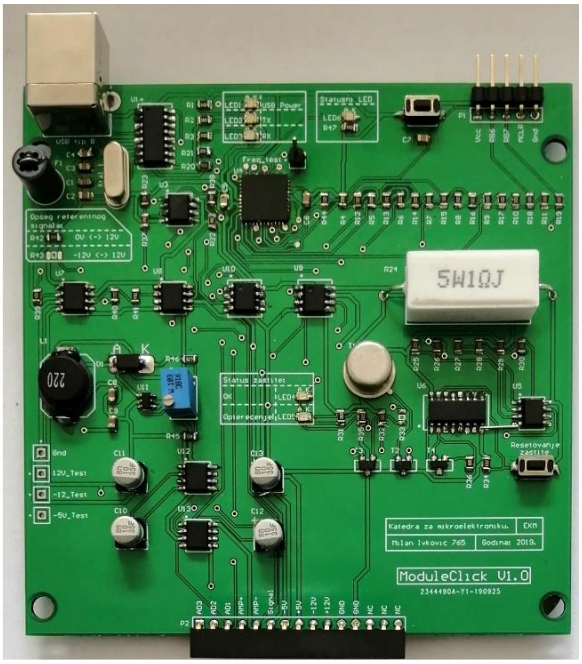
D. Izrada štampanih ploča

Dizajniranje štampanih ploča je rađeno u programu Altium Designer [10] kao i generisanje gerber fajlova za fabričku izradu. Izradu pločica je radila firma JLCPCB [11] u Kini. Na slici 4 i 5 se mogu videti slike glavne ploče i pločice za određenu vežbu.

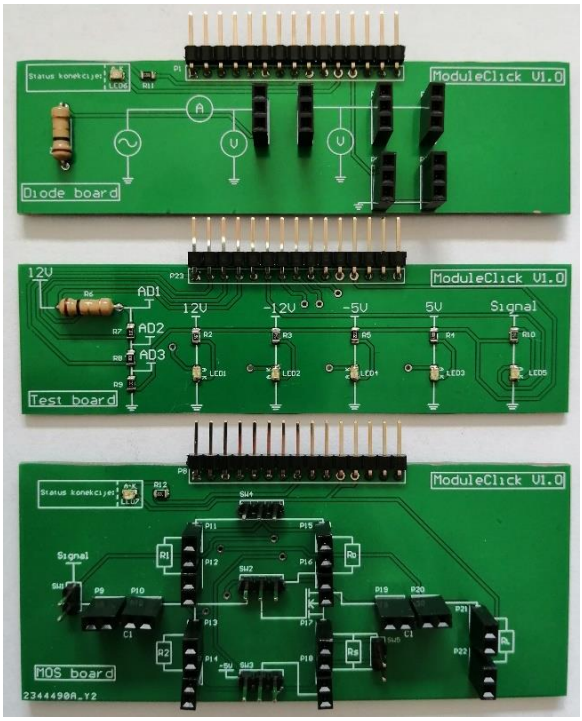
III. MODULECLICK PLOČICE

ModuleClick pločice su dizajnirane tako da prate datu laboratorijsku vežbu kao i standardne makete. Da bi određena maketa bila aktivna potrebno ju je prikačiti za glavnu pločicu (Slika 4) koja je povezana sa računarom i u programu na računaru odabrati koja se vežba radi. Pločica poseduje definisane merne tečke koje se mere od strane glavne pločice i prosleđuju programu na računaru. Na slici 5

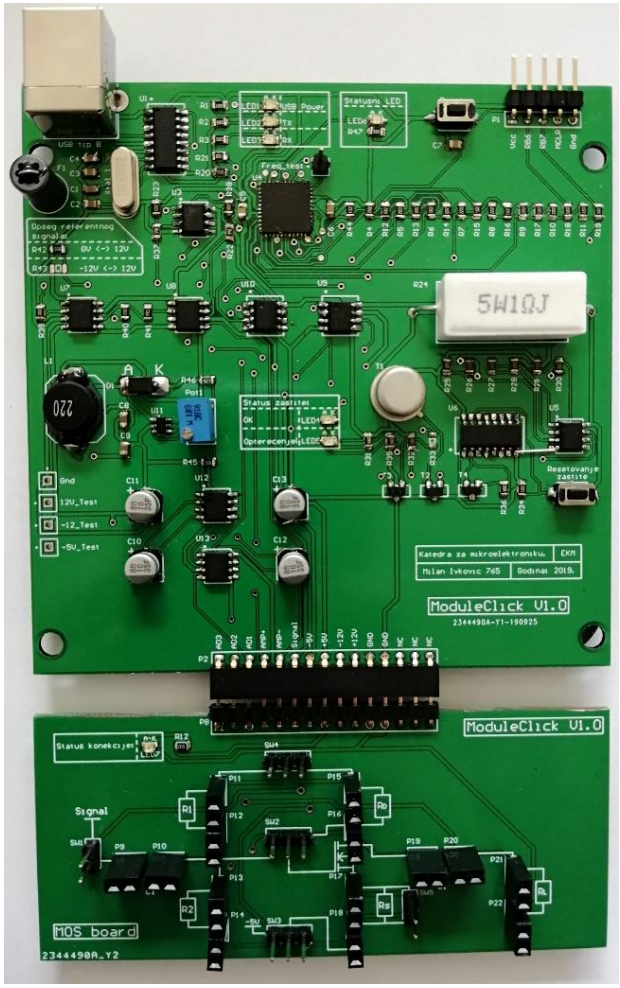
su prikazane tri pločice, jedna je test pločica (u sredini) dok su ostale dve za vežbu sa diodama (gornja pločica) i MOS tranzistorima (u donjem delu). Izgled sastavljene glavne ploče i modul ploče je dat na slici 6.



Slika 4. Izgled glavne ploče.



Slika 5. Izgled modul pločica.

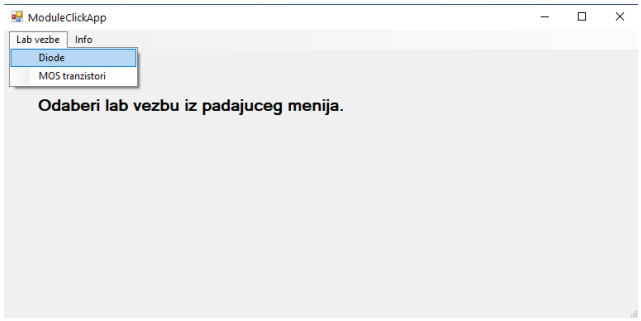


Slika 6. Izgled sastavljene glavne ploče i modul pločice.

IV. PROGRAM NA RAČUNARU

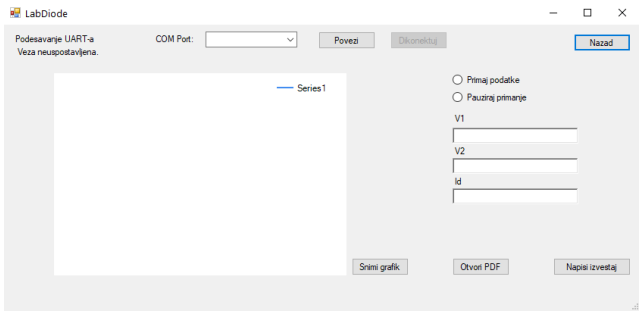
Program za laboratorijske vežbe je pisan u programu Visual Studio [12] i napravljen je tako da podaci koji se prikupljaju za određenu vežbu prikazuju na dijagramu ili u određenom polju za prikaz napona ili struje. Kada student odabere koju će laboratorijsku vežbu raditi pristupa pdf fajlu koji se nalazi na serveru i otvara se u pozadini gde student može pratiti korake. Svaki put pri odabiru laboratorijske vežbe proverava se verzija sa skinutim pdf fajlom i ako postoji nova verzija fajla zamenjuje se starom. Na kraju laboratorjske vežbe postoji polje gde student može upisati svoje inicijale, broj indeksa i kratko zapažanje gde profesor može ustanoviti u kojoj meri je student savladao vežbu. Nakon toga, opcijom „Pošalji“ šalje se izveštaj u pdf formatu profesoru na email sa svim prethodno navedenim parametrima uključujući slike i izmerene napone.

Student u padajućem meniju „Lab vežbe“ bira koju laboratorijsku vežbu radi, slika 7.



Slika 7. Prikaz početne stranice sa odabirom vežbi.

Otvaranjem lab vežbe "Diode" slika 8, dobijaju se podešavanja za konekciju sa ModuleClick pločom, grafički prikaz strujno naponske karakteristike kao i pojedinačni naponi i struje sa ploče.



Slika 8. Prikaz stranice za izradu vežbe sa diodama.

Dugme "Otvori PDF" otvara pdf fajl koji je prethodno automatski skinut sa servera. Nakon dobijenih grafičkih rezultata postoji dugme za snimanje dobijenog grafika koji će kasnije biti korišćen za izradu izveštaja. Dugme „Napiši izveštaj" otvara novu stranicu, slika 9, gde se piše zapažanje sa lab vežbi.



Slika 9. Prikaz stranice za kreiranje izveštaja.

Nakon završetka pisanja i odabira grafika, bira se mail profesora iz padajućeg menija i klikom na dugme "Pošalji" profesoru se šalje izveštaj na mail čime se lab vežba završava.

ZAKLJUČAK

U radu je prikazan novi način izrade laboratorijskih vežbi upotrebom trenutne tehnologije koja olakšava izvođenje laboratorijskih vežbi kako studentu tako i profesoru. Objašnjen je put kreiranja ModulClick pločica kao i problemi na koje je autor nailazio tokom projektovanja i razvoja. Na kraju je prikazana kako izgleda jedna laboratorijska vežba na računaru.

Mesta za unapređenje postoji, kako u hardveru tako i u softveru, ali značajne promene će se dogoditi kada se pločica i softver testiraju sa studentima gde će težiti ka lakšem i jednostavnijem rukovanju.

ZAHVALNICA

Autor rada želi da se zahvali prof. dr Danijelu Dankoviću sa katedre za Mikroelektroniku koji je prepoznao potencijal projekta i koji će ovakav projekat uvrstiti u svoje laboratorijske vežbe radi daljeg razvitka i unapređenja.

LITERATURA

[1] PIC18F45K22 Datasheet. Dostupno na: https://www.microchip.com/wwwproducts/en/PIC18F45K22
[2] CH340 Datasheet. Dostupno na: https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF
[3] You Li, Tao Zeng, Degang ChenA, *High resolution and high accuracy R-2R DAC based on ordered element matching.* Z. Mijanovic, R. Dragovic-Ivanovic, L. Stankovic, *R/2R/sup +/ digital-analog converter (DAC)*
[4] LM358 Datasheet. Dostupno na: https://www.onsemi.com/pub/Collateral/LM358-D.PDF
[5] MCP41010 Datasheet. Dostupno na: http://ww1.microchip.com/downloads/en/devicedoc/11195c.pdf
[6] MikroC. Info na stranici: https://www.mikroe.com/mikroc-pic
[7] TMR0, PIC18F45K22 Datasheet str[154-156]. Dostupno na: https://www.microchip.com/wwwproducts/en/PIC18F45K22
[8] TMR1, PIC18F45K22 Datasheet str[157-158]. Dostupno na: https://www.microchip.com/wwwproducts/en/PIC18F45K22
[9] CD4017 Datasheet. Dostupno na: https://www.ti.com/lit/ds/symlink/cd4017b.pdf
[10] Altium Designer. Info na stranici: https://www.altium.com/altium-designer/
[11] JLCPCB. Info na stranici: https://jlcpcb.com
[12] Visual Studio. Info na stranici: https://visualstudio.microsoft.com

Primena stabala odlučivanja korišćenjem Data Mining analize podataka o padavinama primenom C# aplikacije

Miljan G. Jeremić

Sadržaj – Uticaj klimatskih promena, a pre svega njenih efekata koji su nanoslili štetu društvu je problem koji nije lako odrediti. Da bi se uspešno borili protiv klimatskih promena potrebno je proceniti njen stvarni početak, kao i efekat trajanja i kraj, ali je i teško odrediti intenzitet i uticaje. Zbog velike štete koja se nanosi u dužem vremenskom period potrebno je naći pouzdane i standardizovane pokazatelje koji bi definisali pojave kao što su količina padavina, suše u prostoru i vremenu kao njen pokazatelj. U radu je predstavljena aplikacija u *C#* programskom jeziku koja koristeći podatke sa više mernih stanica u Srbiji koji su zapisani u bazi podataka i tehniku mašinskog učenja, kakva je *Data Mining*, prikazuje stablo odlučivanja za brojni indeks koji se odnosi na padavine u vremenskom period od 20 godina.

I. UVOD

Učenje je proces koji se kod ljudi odvija gotovo neprekidno. Kako se sposobnost učenja smatra osnovnom odlikom inteligentnih bića, to nije iznenađujuće što je mašinsko učenje jedna od centralnih oblasti istraživanja u veštačkoj inteligenciji. Veštački sistemi koji su sposobni da uče evidentno poboljšavaju svoje performanse, a biološki sistemi povećavaju verovatnoću svog opstanka i produžetka vrste.

Mašinsko učenje [1] je oblast koja proučava procese na kojima se zasniva učenje kod ljudi i kod veštačkih sistema. U cilju obuhvatanja svih relevantnih aspekata, oslanja se na veliki broj drugih disciplina, uključujući veštačku inteligenciju, verovatnoću i statistiku, teoriju informacija, psihologiju i neurobiologiju, teoriju upravljanja, filozofiju itd.

Od svih navedenih oblika učenja, od najvećeg značaja za tekuću praksu u domenu računarstva i veštačke inteligencije predstavlja induktivno učenje, koje se često proširuje u nazivu na induktivno empirijsko učenje. Suština ovog tipa učenja je učenje na osnovu raspoloživih primera, ili kako bi to rekli u svakodnevnom jeziku, učenje iz iskustva (drugih). S obzirom na objekat učenja najopštije je učenje funkcionalnih, odnosno ulazno-izlaznih preslikavanja.

Pitanje merenja kvaliteta obučenog sistema je jedno od ključnih pitanja mašinskog učenja. Lako se može zamisliti da je izabrana hipoteza g najbolja moguća u odnosu na dati obučavajući skup, ali da se pokazuje vrlo lošom na novim primerima koji nisu učestvovali u obučavanju. Stoga je opšteprihvaćena mera kvaliteta jednog sistema mašinskog učenja njegova sposobnost generalizacije, odnosno ponašanje sistema na instancama koje nisu vidjene u fazi obučavanja. Stoga tipično sintezu prate dva skupa, obučavajući trening i test skup. Test skup se koristi samo za procenu generalizacionih sposobnosti sintetisanog sistema i ne sme se koristiti u fazi obučavanja. Uobičajene numeričke mere performansi su ili srednjekvadratna greška izmedju izlaza sistema i ciljnih vrednosti ili ukupan broj grešaka.

U praktičnim primenama, susreću se različiti modeli mašinskog učenja u zavisnosti od toga koje informacije su prisutne u obučavajućim skupovima, na koji način se kontroliše proces obučavanja i da li sistem za obučavanje može da menja svet koji ga okružuje.

Miljan G. Jeremić, Knjaževačka gimnazija, Karađorđeva 16, 19350 Knjaževac, Serbia, E-mail: miljan.jeremic@gmail.com

II. ALGORITMI I TEHNIKE MAŠINSKOG UČENJA

U okviru nauke o podacima (Data Science) treba pomenuti koji su to najznačajniji algoritmi mašinskog učenja koji mogu biti primenjeni nad podacima koji su najčešće u obliku Excel CSV datotetke ili u JSON formatu. Svakako najznačajniji algoritmi koje treba pomenuti su: linearna regresija, logistička regresija, SVM (Support Vector Machine), Random Forest algoritam, K-means klasterizacija, neuronske mreže i konvolucione neuronske mreže.

Svakako najpoznatije tehnike mašinskog obučavanja su neuronske mreže i stable odlučivanja.

A. Sistemi odlučivanja

U cilju sticanja opštije slike o oblasti, navešćemo neke najvažije podklase sistema mašinskog učenja [1]. Razlikujemo 3 vrste sistema obučavanja.

Nadgledano obučavanje (supervised learning) – obučavajući skup je oblika {xi, f(xi)}, gde je f(xi) ciljna vrednost vektora obeležja xi. Zadatak obučavanja je nalaženje aproksimacije za f, a mera performanse je kvalitet aproksimaciije u tačkama koje ne pripadaju obučavajućem skupu.

Nenadgledano obučavanje, samoobučavanje (unsupervised learning) – u obučavajućem skupu su prisutne samo ulazne instance {xi}. Tipičan pristup je klasterovanje, odnosno grupisanje raspoloživih podataka u manji broj grupa unutar kojih su podaci sličniji u

poredjenju sa podacima iz ostalih grupa. Imenovanjem klastera se dolazi po- srednim putem do označenih uzoraka, pošto se podrazumeva da sve instance jednog klastera pripadaju istom konceptu.

Obučavanje sa podsticanjem (reinforcement learning) – je složeniji koncept od prethodna dva. Poreklo vodi iz teorije upravljanja u kome je dinamičko okruženje opisano trojkom (stanje, akcija, nagrada). U ovom tipu učenja potrebno je naučiti kako će se vršiti preslikavanja situacija u akcije, a da se pri tome maksimizira nagrada. Za razliku od obučavanja sa učiteljem algoritmu obučavanja nije rečeno koje akcije da preduzima u datoj situaciji. Dobar primer da se shvati ovaj scenario je učenje igranja šaha.

B. Stabla odlučivanja i induktivno učenje

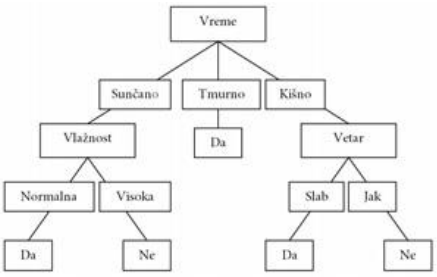
Ukoliko pretpostavimo da se prostor hipoteza sastoji od objakata strukture stabla dolazimo do koncepta induktivnog empirijskog učenja stabala odlučivanja [1], kao jedne od veoma važnih oblasti mašinskog učenja. Na slici 1. prikazani su elementi u induktivnom učenju.



Slika 1. Ključni elementi u induktivnom učenju [2].

Stabla odlučivanja se sastoje od čvorova i grana, koje spajaju čvorove roditelje sa čvorovima naslednika - dece. Čvor koji se nalazi na vrhu stabla (koren) nema roditelja, dok svi ostali tzv. unutrašnji čvorovi imaju samo jednog roditelja. Čvorovi koji nemaju naslednike se nazivaju listovi. Listovi predstavljaju sva moguća rešenja koja se mogu dobiti iz datog stabla. To su čvorovi odgovora, a ostali se nazivaju čvorovi odluke.

Stabla odluke klasifikuju primere u dve ili više klasa na osnovu vrednosti atributa kojima su primeri opisani, propuštajući ih niz stablo od korena do listova. Na početku se bira artribut koji čija vrednost najbolje deli raspoložive primere. Svaki čvor u stablu predstavlja test nekog atributa, a svaka grana koja polazi iz tog čvora odgovara jednoj od mogućih vrednosti tog atributa. Na taj način se primeri dele u podskupove, zavisno od vrednosti izabranog atributa. Ako svi atributi u podskupu pripadaju istoj klasi, stablu se dodaje list. Svaki put kroz stablo predstavlja jedno klasifkaciono pravilo, tj. konjunkciju testova atributa, a samo stablo je disjunkcija ovih konjunkcija. Stabla odluke su značajna za oblast mašinskog učenja, jer su zbog svoje sistematske strukture razumljiva za ljude i prestavljaju odličnu tehnologiju obrazlaganja u auto matizovanim dijagnostičkim sistemima. Obrazloženje zašto je doneta neka konkretna odluka se dobija izdvajanjem puta od pripadajućeg lista do korena, pri čemu se iz svakog čvora iščitava razlog parcijalnog izbora. Osim toga, svakom stablu odlučivanja se jednoznačno može pripisati skup pravila tipa: ako – onda (if – then), koja su osnovni gradivni blokovi baza znanja ekspertskih i drugih sistema zasnovanih na znanju, što je prikazano na slici 2.



Slika 2. Primer stabla odlučivanja za meteorološki koncept [2].

Postoje mnogi praktični problemi koji odgovaraju ovim karakteristikama. Takvi problemi, kod kojih je zadatak da se ulazni podaci klasifikuju u jednu od unapred zadatih disjunktnih klasa, nazivaju se klasifikacioni problemi.

Algoritam stabla odlučivanja ima za cilj da klasifikuje stavke tako što identifikuje pitanja o njihovim atributima koja će pomoći u odlučivanju u koju klasu da se stave. Svaki čvor u stablu je pitanje, sa granama koje dovode do dodatnih pitanja o stavkama, dok listovi predstavljaju poslednju klasifikaciju.

Primeri korišćenja stabla odlučivanja mogu da budu izgradnja platformi za upravljanje znanjem za usluge klijentima, predviđanja cena i planiranje proizvoda.

C. Osnovni algoritmi učenja stabala odluke

Većina algoritama koji su razvijeni za učenje stabala odluke su varijacije bazičnog algoritma koji koristi od-vrha-naniže (top-down), pohlepnu (greedy) pretragu kroz prostor mogućih stabala odluke [1]. Ovaj pristup je korišćen kod ID3 algoritma i njegovog naslednika C4.5.

Osnovni algoritam ID3 uči stabla odluke konstruišući ih od vrha naniže, počevši pitanjem: koji atribut treba testirati u korenu stabla? Da bi se odgovorilo na ovo pitanje, atribut svakog primera se procenjuje pomoću statističkog testa, da bi se odredilo koliko dobro klasifikuju obučavajuće uzorke. Odabere se najbolji atribut i koristi kao test u korenu stabla. Potom se generišu čvorovi naslednici za svaku moguću vrednost ovog atributa, a obučavajući primeri se propuštaju kroz grane i sortiraju na dostignutim čvorovima naslednicima. Proces odredjivanja najboljeg atributa za testiranje u svakom čvoru nasledniku se ponavlja, koristeći obučavajuće primere koji su dospeli do tog čvora. Na taj način se ustari vrši pohlepna potraga za odgovarajućim stablom odluke. Shodno svojstvu pohlepne pretrage, algoritam se nikada ne vraća u prethodno stanje da bi eventualno preispitao ranije izbore.

D. Traženje najboljeg atributa za klasifikaciju

Glavni izbor u ID3 algoritmu [1] je određivanje atributa koji se testirati u svakom čvoru stabla. Stoga nam je potrebna kvantitativna mera koja govori o vrednosti atributa. Definišemo statističku veličinu, informacioni dobitak, koja meri koliko dobro može dati atribut da razdvoji obučavajuće primere na osnovu njihove ciljne klasifikacije. ID3 koristi veličinu informacioni dobitak da među kandidatima odabere najbolji atribut u svakom koraku dok stablo raste.

Da bismo precizno definisali informacioni dobitak, podsetimo se smisla i definicije entropija [1] kao mere informacija i neodredjenosti jednog sistema. Za datu klasu S, koja sadrži pozitivne (p) i negativne (n) primere nekog ciljnog koncepta, entropija od S u odnosu na ovu binarnu klasifikaciju je:

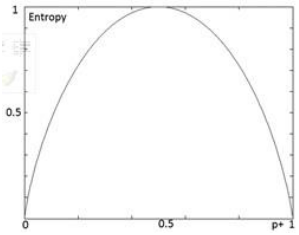
$$\text{Entropy(S)} = \textit{Entropy}([p,n]) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (1)$$

gde je,

$p_+ = p/p+n$, proporcija pozitivnih primera u S,

$p_- = p/p+n$, proporcija negativnih primera u S

U svim računanjima u vezi sa entropijom uzima se da je 0· log0=0.



Slika 3. Funkcija entropije u zavisnosti od p [2].

Na slici 3. je prikazana zavisnost funkcije entropije kada se p+ kreće u intervalu od 0 do 1.

Do sada se diskutovalo o entropiji u specijalnom slučaju kada je ciljna klasifikacija binarna. Generalno, ako ciljni atribut može uzeti jednu od c različitih vrednosti, onda entropiju definišemo kao:

$$\text{Entropy(S)} = \sum_{i=1}^{c} - \text{p}_\text{i} * \log_2 p_i \quad (2)$$

gde je pi proporcija primera u S koji pripadaju klasi i.

Pošto je entropija mera nehomogenosti u klasama obučavajućih primera, sada možemo da definišemo meru efektivnosti atributa pri klasifikaciji obučavajućih primera. Mera koju ćemo koristiti, informacioni dobitak (information gain), je očekivana redukcija entropije koju izaziva particija primera na osnovu ovog atributa. Preciznije, informacioni dobitak [1] za slučaj da se atribut A koristi za razvrstavanje primera iz S, tj. kao koren stabla, je:

$$\textit{Gain}(S,A) = \textit{Entropy}(S) - \sum_{v \in \textit{Values}(A)} \frac{|S_v|}{|S|} * \textit{Entropy}(S_v) \quad (3)$$

gde je Values (S) skup svih mogućih vrednosti atributa A, a Sv podskup od S za koji atribut A ima vrednost v, tj. Sv = {sϵS| A(s)=v}. Prvi izraz na desnoj strani jednačine je entropija originalne klase primera S, a drugi izraz je očekivana vrednost entropije nakon što je S podeljeno uz pomoć atributa A, tj. suma entropija svakog podskupa Sv pomnožena sa udelom Sv/S primera koji pripadaju Sv. Gain(S, A) je u stvari informacija o vrednosti ciljne funkcije za poznatu vrednost atributa A. Vrednost Gain(S,A) se može interpretirati i kao broj bita potrebnih za kodovanje ciljne vrednosti određenog elemeta iz S, kada znamo vrednost atributa A.

E. Induktivni pomeraj (bias)

Induktivna pomeraj (bias) je skup pretpostavki koje zajedno sa obučavajućim primerima opravdavaju klasifikacije budućih primera, koji nisu učestvovali u fazi obučavanja. Ili preciznije, induktivni pomeraj su restrikcije koje namećemo prostoru hipo- teza, smatrajući zadati algoritam obučavanja uspešnim ukoliko je pronašao najbolju hipotezu u prostoru hipoteza uz istovremeno zadovoljavanje nametnutih restrikcija.

Za skup obučavajućih primera postoji veliki broj stabala odluke koja su konzistentna sa ovim primerima. Opis induktivnog pomeraja algoritma ID3 se stoga sastoji u opisu pomeraja na osnovu koga on bira jednu od konzistentnih hipoteza među svim ostalima. ID3 bira prvo prihvatljivo stablo odluke na koje naiđe u svojoj pretrazi. Grubo govoreći, strategija pretraživanja metode ID3 (a) bira kraća stabla pre nego duža, i bira stabla kod kojih su atributi sa najvećim informacionim dobitkom postavljani najbliže korenu. Zbog malih, ali važnih interakcija između heuristike za selekciju atributa koju ID3 koristi i konkretnih obučavajućih primera sa kojima se susreće, teško je precizno definisati njegov induktivnu pomeraj. Pomeraj možemo približno definisati kao davanje prednosti kratkim stablima odluke nad kompleksnim stablima.

F. Data Mining

Data Mining koristimo kako bi uradili neku konkretnu analizu podataka. To je tehnika putem koje možemo da pronadjemo nekakvu skrivene zakonitosti u velikim skupovima podataka.

Nekada, klasičnim izveštajima nemožemo doći do podataka koji mogu biti jako interesantni za krajnjeg korisnika, recimo menadžere koji donose odluke.

Tada treba upotrebiti neke naprednije tehnike kako bi smo iz podataka mogli izvući neke konkretnije podatke, ili čak možda predvideti neko poslovanje.

Neki od primera Data Mining [2] analiza su, na primer, kako će se prodavati određeni proizvod na određenoj lokaciji, a sve u skladu sa navikama kupaca, ili, na primer putem Data Mining tehnika možemo da kupcima preporučimo kupovinu nekih proizvoda na osnovu istorije njegove kupovine. Takodje, vrlo često se Data Mining koristi i u bankarskom sektoru za analizu rizika u kreditnom plasmanu i slično. Banke to koriste kako bi mogle analizirati svoje klijente ko je od njih kreditno sposoban ili nije.

Ideja je da danas sve češće je posao onih koji odlučuju zamenjuju softveri. Oni treba da predvide šta bi moglo da se desi sa nekim klijentom koji želi da uzme kredit. Da bi mogle da se rade takve nalaize potrebno je poznavanje određenog seta algoritama, a za Data Mining je usko vezana statistika i matematika naravno, mnogo vise statisitka i statističke metode analize podataka.

Pre nego što se odluči da se radi sa Data Mining-om trebalo bi se upoznati sa algoritmima koje Data Mining pruža da bi se mogla uraditi neka analiza podataka.

Najčešće, algoritme Data Mining-a delimo u neke kategorije, a to su algoritmi klasifikacije, algoritmi regresije (linearna i logistička regresija za predviđanje jedne ili vise kontinuiranih promenljivih), algoritmi segmentacije, odnosno klasteringa (vrši se odvajanje objekata u neke skupove sa zajedničkim osbinama), algoritmi asocijacije (služe za pronalazak nekih korelacija u podacima, na primer, koji će se proizvodi zajedno kupovati) i algoritmi analize sekvenci (pronalaženje putanje korisnika kroz web, ili redosled ubacivanja artikala u korpu, itd.).

Najčešće se koriste algoritmi klasifikacije, kao i algoritmi regresije.

G. Poslovna inteligencija

U složenim poslovnim sistemima svest o korisnosti prihvatanja koncepta poslovne inteligencije raste svakim danom, a time raste i potreba za uvođenjem informacionih sistema koji su sposobni podržati takve trendove. Pomoću alata BI organizacije efikasnije povezuju ljude sa njihovim poslovanjem, sa kupcima, dobavljačima i partnerima, omogućavajući uvid u ogromnu količinu kompleksnih podataka. Ističu se sledeći tipovi alata:

- Alati za upite predstavljaju programske pakete koji omogućavaju korisnicima postavljanje upita o matricama ili detaljima na koje se podaci odnose.
- Alati za rudarenjepodataka, koji omogućavaju automatsko pretraživanje karakterističnih matrica ili korelacija među podacima.
- Softver za multidimenzionalne analize ili pozantiji kao OLAP (Online Analytical Processing) [3].

Današnje poslovanje karakteriše zagušenje poslovnih sistema enormnim količinama podataka i informacija internog i eksternog karaktera. Primena koncepta poslovne inteligencije omogućava organizacijama korištenje samo onih informacija koje su im u određenom vremenu potrebne za donošenje poslovnih odluka u obliku koji im odgovara. Dalje, primjena poslovne inteligencije smanjuje količinu podataka kojima su zaposleni izloženi uz istovremeno povećanje kvaliteta informacija. Iz ovoga proizilazi da je glavna namera koncepta poslovne inteligencije generisanje što kvalitetnijih informacija potrebnih u procesu donošenja odluka.

Karakteristika poslovne inteligencije da je proaktivna, potiče iz operativnih podataka i orijentisana na je na isporuku informacija korisnicima prema njihovim preferencijama [4].

Prema procenama, tipična organizacija danas analizira samo 10% prikupljenih podataka. Implemetacijom sistema poslovne inteligencije organizacija može koristiti i preostale podatke iz svih izvora transformišući ih u kvalitetne informacije.

Sa stanovišta informatičke infrastrukture, sistem poslovne inteligencije započinje sa izgradnjom skladišta podataka kao centralnom bazom organizacije u koju se slivaju svi podaci internog karaktera (nastali u organizaciji) ili podaci iz spoljnih izvora. Proces punjenja skladišta je omogućen uz korištenje ETL alata, a nakon toga se gleda mogućnost analize. Eksploatacija i vizuelizacija podataka se vrši uz pomoć multidimenzionalne analize ili korištenje OLAP (On-Line Analytical Processing) alata. Skraćenica OLAP podrazumeva kategoriju aplikacija i tehnologije namenjenu za skupljanje, upravljanje, obradu i prezentaciju multidimenzionalnih podataka namenjenih analizama za potrebe upravljanja. OLAP alati se međusobno mogu jako razlikovati, stoga je bitno da pri akviziciji alata u njihovom vrednovanju u cilju izbora najpovoljnijeg rešenja, učestvuju i korisnici i IT stručnjaci. Modeli OLAP su lagano razumljivi, pa korisnici ne moraju vladati posebnim analitičkim veštinama. Takođe, ih karakteriše velika brzina rada što ima efekta na efikasnost ukupnih napora kod donošenja odluka. Spektar mogućnosti koje obuhvataju OLAP alati je veoma širok, kreće se od jednostavnog pretraživanja, preko proračuna do visoko diferenciranih i sofisticiranih analiza kao što su vremenske serije i kompleksno modeliranje. Time oni obuhvataju kompletan sled koji počinje podacima, a završava poslovnom inteligencijom.

Prednost koju pružaju mehaniznmi OLAP alata je i provođenje ad-hoc analiza, pored rutinske i periodične analize i na osnovu njih produkovanih papirnih izveštaja. Ad-hoc analiza omogućava generisanje odgovora u stvarnom vremenu koje je izraženo veoma malim vremenskim jedinicama do nekoliko minuta i tekstualnom i grafičkom obliku koji je pogodan za upotrebu [5].

Na tržištu je danas moguće naći nekoliko varijanti OLAP alata, a jedan od njih je i relacioni OLAP alati, u njihovoj osnovi se nalaze relacione baze podataka i za razliku od multi dimenzionalnih verno odražava stvarni svet. Ipak, u slućaju promena koje proizvođač alata želi uraditi, moguća je otežana primena standardnih upitnih jezika (SQL) koji je blizak korisnicima, kao i neke druge interakcije sa bazom podataka.

OLAP alati su prisutni na tržištu više od deset godina, a kao najpoznatiji proizvođači navedenih alata su kompanije: Cognos, Microsoft, Oracle, SAS, IBM, Business Objects, Aplix, Hyperion Solution itd. [6].

III. PODACI I OBLAST PROUČAVANJA

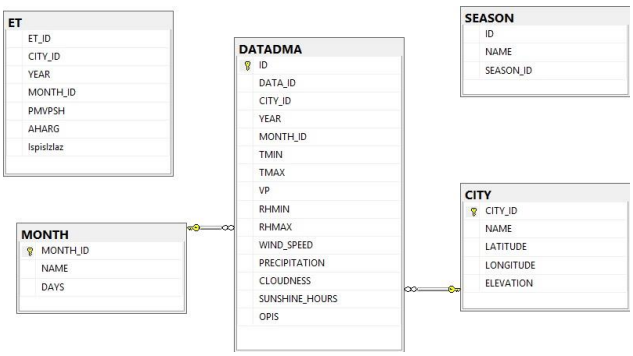
U ovom radu je odluka da se primeni neki od algoritama u praksi, pa je izabran prikaz jednog projekta premenom stabla odlučivanja i na koji način može to da se primeni kroz SQL Server Data Tools alat, konkretno kroz Visual Studio i programski jezik C# [7].

Prvo se mora definisati problem koji se želi rešiti i prikazati putem Data Mining analize.

U radu je prekazana analiza vrednosti parametara sa padavinama za grad Niš, gde se na osnovu 4 vrednosti donosi odlika o predvidjanju na izlazu za zavisnu promenljivu PMVPSH, i to prikazuje se prikazuje u formi stabla odlučivanja. Analiziraće se na osnovu diskretnih vrednosti za TSR, VP, WIND_SPEED, SUNSHINE_HOURS kakva će biti izlazna vrednost, odnosno kakvo je predviđanje, za zavisnu promenljivu PMVPSH, tj. da li je ona mala, srednja ili velika vrednost, i kako će aplikacija predstaviti stablo odlučivanja u grafičkom okruženju.

Na početku treba odrediti kakav će nam skup podataka biti na ulazu za Data Mining algoritam. Ovaj algoritam nije specifično vezan za određeni set podataka, mada on treba biti pripremljen za dalje analize i donošenje odluka. Ne može se povezati na skladište podataka, na relacionu bazu podataka ili Excel tabelu sa podacima i onda očekivati da će se iz gomile zapisanih podataka stvoriti neka izlazna informacija.

Podaci se za Data Mining analizu obično pripreme ranije, odredi se šta će biti ulazni podaci (promenljive), i šta će se očekivati, odnosno šta će se predviđati.



Slika 4. Relecioni model meteorološke baze podataka.

Ovde želimo da analiziramo vrednosti na izlazu za promenljivu PMVPSH, a na osnovu rekli bi rizičnih slučajeva nekih ulaznih parametara koji su zabeleženi i sačuvani u bazi podataka u SQL Server-u, gde je relacioni model podataka meteorološke baze podataka prikazan na slici 4.

Za potrebe ove aplikacije pripremljen je upit u bazi koja se pokreće u SQL Server-u, gde se kao novoformirana kolona prikazuje test sa vrednostima (mala, srednja i velika) na osnovu vrednosti PMVPSH. U bazi podataka se nalazi jedna tabela sa potrebnim vrednostima za 4 ulazne promenljive diskretnog tipa. Tabela se zove DM i u njoj će biti zapisane sve vrednosti potrebnih promenljivih koje se koriste na ulazu. Druga tabela, DM1, biće kreirana na osnovu upita koji se izvrši u SQL Server-u, gde će biti u koloni Predviđanje zapisane vrednosti na osnovu rezultata upita. Tabela će imati ukupno 372 sloga. Ovo će biti izvor podataka za dalju analizu pomoću Data Mining-a.

Nakon kreiranja skladišta podataka, sada se kreće sa izradom aplikacije u konkretnom okruženju, ovde je to Visual Studio, verzija 2015. Za projekat koji će da radi sa ovakvim skupom podataka mora se koristiti Data Tools alat.

IV. METODOLOGIJA

U ovom radu biće predstavljeni algoritmi klasifikacije, tačnije obradiće se stabla odlučivanja (Decision Tree) i neuronske mreže (Neural Network).

Algoritmi klasifikacije služe za predviđanje šta će se desiti sa nekim podacima, dakle, predviđanje jedne ili vise diskretnih promenljvih, ti. promenljivih koje imaju ograničen opseg vrednosti bazirano na nekim ulaznim atributima. Primenom stable odlučivanja može se predvideti da li će kupac kupiti određeni proizvod ili neće, i kako na izlazu imamo odgovor Da ili Ne, dok su na ulazu neke od navika kupaca. Na ulazu mogu da budu više diskretnih ili kontinuiranih promenljivih.

Kod algoritama Data Mining-a postoje ulazi, i na osnovu ulaza algoritam vrši procesiranje i daje neki izlaz.

Neki od primera algoritama klasifikacije su stabla odlučivanja, neuronske mreže, genetički algoritmi i slično.

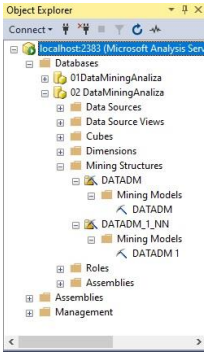
Za kreiranje novog projekta koristi se kartica Business Intelligence, izabere se Analisies Services Multidimensional and Data Mining anliza. Da bi se moglo raditi sa ovim alatom treba da bude instaliran Analysis Services u Multidimensional i Data Mining modu u okviru SQL Server-a.

Sada treba da se kreira mining struktura, gde treba da se kreira izvor podataka, a u ovom slučaju to je izvor podataka o meteorološkoj strukturi, koji je prikazan na slici 5. Nakon ovog koraka kreira se konekcija sa user-om i password-om.

Sledeći korak je kreiranje logičkog pogleda na podatke, odnosno DataSource View. Tu se koristi prethodno kreirana konekcija, zatim se izabere tabela iz baze podataka koja je nešto ranije kreirana. Struktura podatka je vrlo jednostavna sa ključem ID, i kolone koje želimo prikazivati kasnije u nekim analizama. Ovaj model podataka se povezuje direktno na bazu podataka.

U sledećem koraku treba kreirati Mining strukturu, gde se korak po korak vrši izbor algoritma za Mining strukturu, kao i izbor kolona za samu strukturu. Izabere se

algoritam sa Decision Trees, koji se koristi za predvidjanje jedne ili vise diskretnih promenljivih. Ovde se primenom ovog algoritma predviđa kakva će biti vrednost na izlazu za promenljivu PMVPSH, tačnije prikaz određenog broja rezultata. Prvo treba izabrati koja je kolona ključna u tabeli, koja je jedinstveno identifikuje kao u relacionim bazama podataka, mada se može koristiti i više tabela, a koje su relacijom povezane. Nakon ovoga, treba izabrati kolone koje će biti izabrane kao ulazne promenljive diskretnog ili kontinuiranog tipa, a zatim treba izabrati i kolonu za koju se vrši primena nekog algoritma, a to je u našem slučaju stablo odlučivanja. Ta kolona u ovom projektu se naziva Predviđanje.



Slika 5. Kreiranje Mining modela i neuronske mreže u Analysis Services.

Dalje treba podesiti koji će broj ili procenat uzeti za testiranje seta podataka, i to najčešće bude 20%, jer kad algoritam procesira podatke on odvoji deo podatka za testiranje.

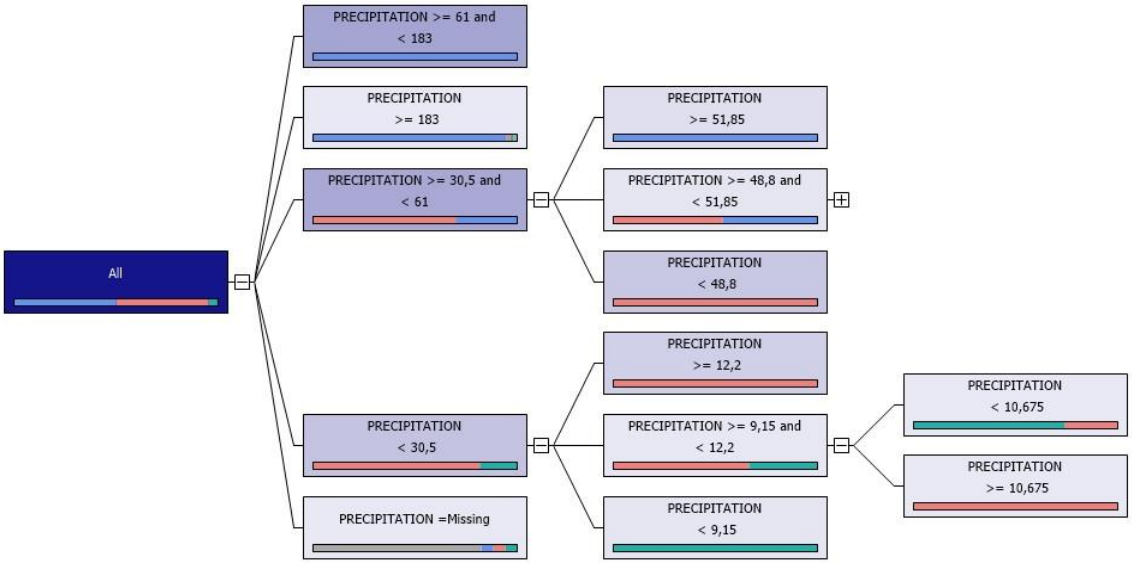
Na kartici Mining Model Viewer može da se vidi stablo odlučivanja nakon procesiranja podataka koji se nalaze u tabeli. Na kraju, nakon nekog vremena za obradu, dobija se grafički prikaz kako je algoritam rasporedio stablo odlučivanja. To stablo odlučivanja je raspoređeno do nivoa 4, a prikaz se može videti na slici 6.

Na osnovu nivoa u stablu odlučivanja, može se videti sa kojom verovatnoćom se posmatraju ulazni podaci iz tabele.

U narednom prozoru se može proveriti koliko pomenuti algoritam za analizu podataka radi efikasno, što je prikazano na slici 7.

S obzirom da su jako dobro složeni podaci, vidi se da je idealna plava linija, dok je crvena linija model iz same aplikacije. S obzirom na mali set podataka, može se videti da ovaj model predviđanja ima verovatnoću predviđanja 99.98% na osnovu stable odlučivnja koje je kreirano.

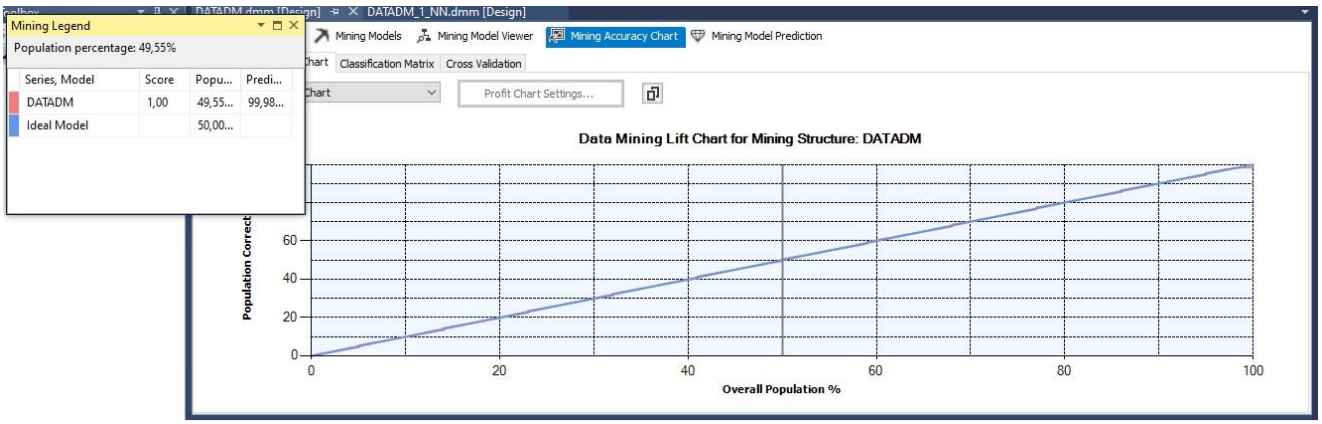
U poslednjem koraku u analizi može da se vrši i postavljanje upita nad modelom, a to se može videti na slici 8.



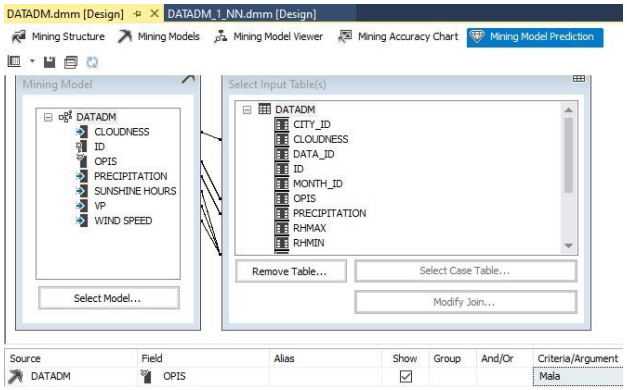
Slika 6. Stablo odlučivanja koje generiše C# program.

V. REZULTATI I DISKUSIJA

Ono što je ovde ključno je to da se formirani model podataka postavlja na Analisies Services u sklopu DM baze podataka. Treba napomenuti da se i u okviru SQL Server-a može u okviru baze podataka naći model u obliku stabla odlučivanja, kao isti onaj koji je prikazan u C# aplikaciji. Može se u ovom delu baze podataka kreirati i neki predikcijski upit. Na osnovu svega se mogu kreirati izveštaji koji izvlače podatke koji su filtrirani kroz neki model podataka. Upiti se generišu na osnovu on-click principa, bez neke velike potreba poznavanja pisanja upita. U samom upitu na kraju treba uzeti u obzir neku predikcijsku funkciju sa kojom se želi raditi, i da ona predviđa neku verovatnoću na osnovu jedne ili vise kolona za predviđanje. Nakon pokretanja upita, dobijamo verovatnoće za ispis izlazne vrenosti na osnovu prethodno kreiranog modela sa podacima.



Slika 7. Mining Accuracy chart.



Slika 8. Generisanje upita nad modelom.

VI. ZAKLJUČAK

Na osnovu analize meteoroloških podataka za period od 1990. do 2018. godine koji se nalaze u bazi podataka u SQL Server-u i uradjenog Data Mining modela nad tim podacima, dobili smo strukturu stable u četiri nivoa gde se kao parameter posmetranja uzima PRECIPITATION.

Na osnovu kreirane aplikacije i rezultata koji je prikazan u obliku stable odlučivanja može se zaključiti da je najmanji prosek padavina za grad Niš mora biti veći od 10.675 mm, i to u četvrtom nivou stable odlučivanja sa skoro potpunom verovatnoćom, a da bi idealna vrednost bila u opsegu većem od 61 do 183 mm.

Kreiranjem ovakve jedne aplikacije koja radi sa realnim podacima, kao i generisanjem stabala odlučivanja dobijaju se neophodne podloge za potrebe analiza i procene rizika od suša u cilju ublažavanja neželjenih posledica.

ZAHVALNOST

Na kraju ovoga rada autor se zahvaljuje Knjaževačkoj gimnaziji na razumevanju i pružanju potrebne podrške. Autor je zahvalan i profesorima Elektronskog fakulteta u Nišu koji su nas naučili da razmišljamo o problemima, i da ih rešavamo na jedan kreativni način, podarili nam dovoljno znanja o programiranju i primeni novih IT tehnologija.

LITERATURA

[1] M. Milosavljevic, *Veštačka inteligencija*, Univerzitet Singidunum, Beograd, 2015.

[2] Chiwara, Al-Ayyoub, Hossain, Gupta (2006). *Data Mining, Concepts and Techniques Association Rule Mining*, State University of New York, CSE 634, Chapter 8.

[3] Milenković, I., Pavković, M., *Primena OLAP poslovne analize na primeru baze podataka Megatrend univerziteta*, Zbornik radova SYM-OP-IS, Banja Koviljača, 2006.

[4] Panian Ž., Klepac G.; *Poslovna inteligencija*; Masmedia, Zagreb 2003.

[5] Alempije V. Veljović, *Menadžment informacioni sistemi*; Kompjuter biblioteka 2002.

[6] Višnja Ljubetić, *Upravljanje znanjem primjenom alata poslovne inteligencije*, Magistarski rad.

[7] Watson B (2011)., *C# 4.0 kako do rešenja*, SAMS, Mikro knjiga.

Cane for Visually Impaired People -The Stevie Wonder Model-

Ognjen Ćirić

Abstract - Approximately 12.000 visually impaired people live in Serbia. Few are trained to use the white cane technique. The design and programming of a cane that timely, at distances of 70 cm and 30 cm, indicates the existence of obstacles by vibration and acoustic warning was solved through a prototype based on Arduino components. The concept has been confirmed in practice, and improvements to the prototype are possible. Validity and reliability of information would contribute to raising the confidence of the target group.

I. The Principle of Work and Practical Implementation

Cane for visually impaired people, the Stevie Wonder model is based on an ultrasonic sensor [1] that sends an ultrasonic signal every 2 μs that bounces on objects or obstacles of another kind and returns to the sensor. The length of the distance between the sensor and the obstacle defines both the type of warning sound to be triggered and the type of vibration.

In order to create the system, the following components must be provided:

- Arduino uno
- Arduino piezo buzzer
- Arduino ultrasonic distance sensor
- Power switch
- 9V battery
- Battery housing
- Proto board cables
- Power connector
- Vibration motor

This prototype is programmed in Arduino 1.8.8. The code is divided into three sections. The first one declares the variables and sets the ultrasonic signal output from the sensor. The second part of the code checks the distance between the sensor and the obstacle: if the obstacle is less than 70 cm and greater than 30 cm, an acoustic signal [2] is activated. The last part of the code deals with distance checking and if the distance is 30 cm or less, a second acoustic signal and vibration motor are activated.

Ognjen Ćirić is a student of the Vocational School “Milenko Brzak Uča” Ruma, e-mail: ognjenci@gmail.com

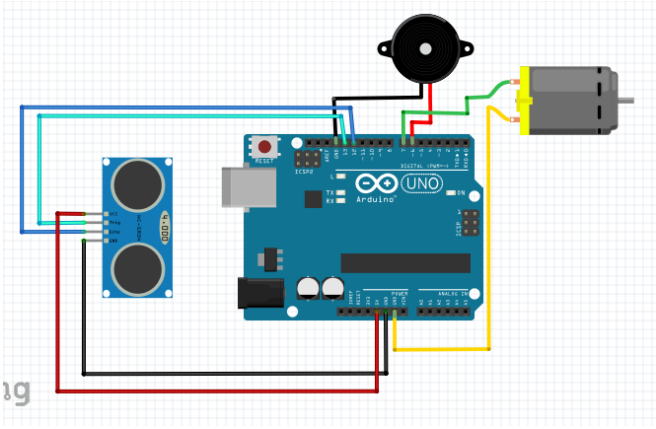


Figure 1. Linking scheme on the proto board.

```
void loop()
{
long duration, distance; //deklarisanje promenjivih
digitalWrite(trigPin, LOW); //ne šalje ultrasoničan signal
delayMicroseconds(2);//pravi pauzu
digitalWrite(trigPin, HIGH);//šalje ultrasoničan signal
delayMicroseconds(10);//pravi pauzu
digitalWrite(trigPin, LOW);//ne šalje ultrasoničan signal
duration = pulseIn(echoPin, HIGH);//računa vreme putovanja zvuka
distance = (duration/2) / 29.1;//računa razdaljinu prepreke u odnosu na senzor
if (distance < 70 && distance >30)// provereava razdaljinu prepreke u ondosu na senzor
{
 tone(buzzer, 1000);//uključuje pištalicu
} else{analogWrite(motor,0);//isključuje vibracioni motor
 noTone (buzzer);//isključuje pištalicu
 }
```

Figure 2. Part of the code from the Arduino IDE editor.

II. The Result

The cane can be used in two ways - as a typical walking cane or a cane for short distance obstacle detection. When the switch on the cane handle is on, its performance changes and it becomes a 21st century cane. A remote sensor, vibration when needed, and a whistle are activated.

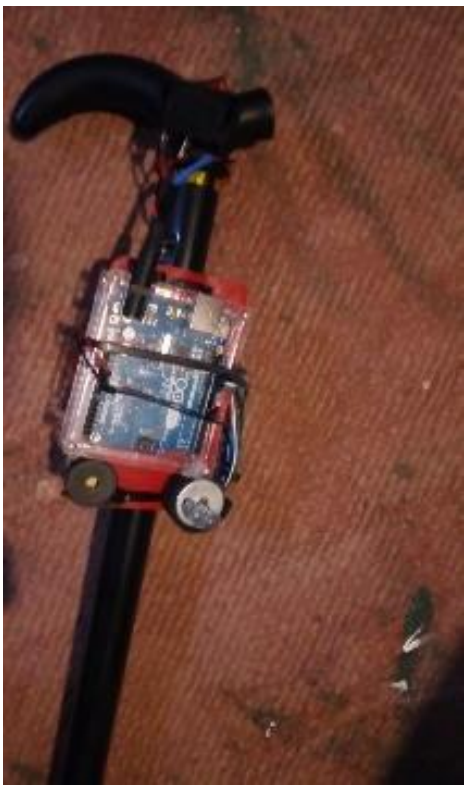


Figure 3. "Stevie Wonder" Prototype.

III. Conclusion

Cane for visually impaired people, the Stevie Wonder model has been tested in different environments: in the room, in the yard and on the street. In all places it reacted as it was programmed: at a distance of 70 cm from the obstacle, the cane beeps slightly and at 30 cm it continuously beeps and vibrates. The distances were chosen according to an average human step length of 75 cm.

The advantage of this prototype is that it can be adapted to the individual needs, the vibration option can be switched off, or the distances at which the sensors respond can be altered. The problem that has arisen is related to the use of a cane in the crowd - in which case it is possible that the sensors and vibration are triggered more often than necessary and may confuse the user.

With further development of the model, this problem can be solved. There is certainly room for improvement in design, which requires greater financial investment. It is also necessary to extend the test group and develop a questionnaire about user impressions/ expectations.

In the test phase, the model proved to be usable - successfully reporting obstacles, which gives hope that in the future, visually impaired citizens will receive another tool that will enable them to become more actively involved in community life and lead a more independent life.

Acknowledgement

I would like to thank the mentor, information technology teacher, Vesna Ilic, for the time and commitment to the project. I also thank the parents for their financial support for the purchasing of a part of the components, as well as Zoran Boskovic who helped me name the project.

Suggestions from the Geriatric Center of Ruma were of great importance for the development of the model.

References

[1] https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04"

[2] www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/

MK BOX

Nikola Crnobrnja

Abstract - One of the possibilities of protecting high value exhibits has been accomplished through RFID shield. The ultra-sound sensor beeps if one approaches the exhibit more than intended. Another layer of protection is the 4x4 keyboard, as a temperature, proto board and ARDUINOMEGA 2560 controller. The components are programmed individually and then placed in a Plexiglas case. The model was successfully tested.

I. The Principle of Work and Implementation

The temperature and humidity controllers activate the heater or fan, depending on whether the temperature should be increased or decreased compared to the benchmark temperature. Content is accessed using the 4x4 keyboard. If the code entered is incorrect, an audible alarm activates. If someone approaches the box before the code is entered, the speaker will sound.

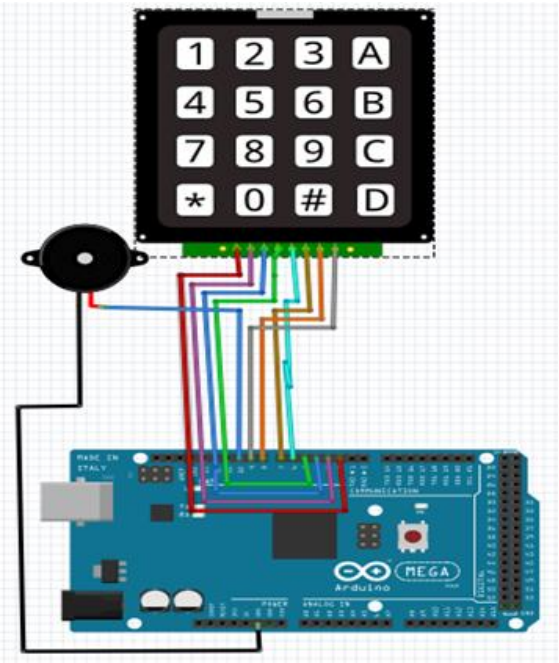


Figure 1. 4x4 keypad with buzzer connected to Arduino Mega 2560.

The first step was to connect each component per se and test the codes that govern them. At the end of the code test, a plexiglass box was made.

Nikola Crnobrnja is a student of the Vocational School "Milenko Brzak Uča" Ruma.

The box contains a double bottom since the regulator sensor, the heater and the fan are located between the first and the second part. They regulate the internal temperature of the box.

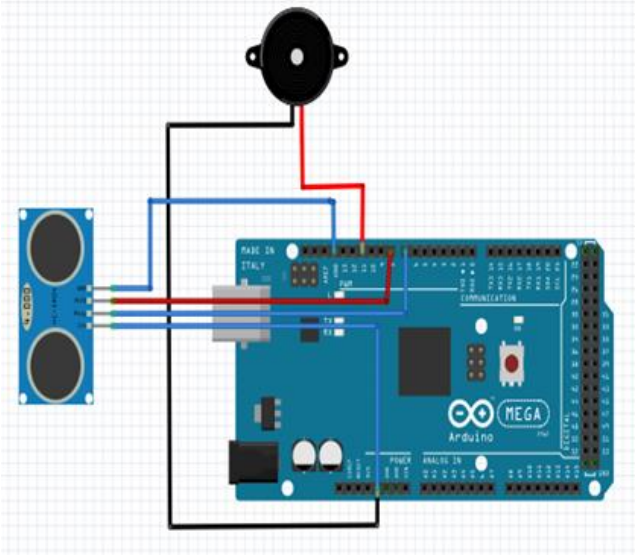


Figure 2. Schematic illustration of an ultra-sonic motion sensor with a buzzer connected to the Arduinomega 2560.

Some of the functions [1], [2], [3] used in the code:

```
void (keypadEvent(KeypadEventeKey){
switch (keypad.getState()){
  case PRESSED:
Serial.print("Enter: ");
Serial.println(eKey);
delay(10);
Serial.write(254);
  switch (eKey){
    case 'A': checkPassword(); delay(1); break;
    case 'B': password.reset(); delay(1); break;
     default: password.append(eKey); delay(1);
}
}
  }
void checkPassword(){
if (password.evaluate()){
Serial.println("Accepted");
Serial.write(254);delay(10);
    if(analogRead(7)<0){
tone(zujalica, 500, 1000);
      }
```

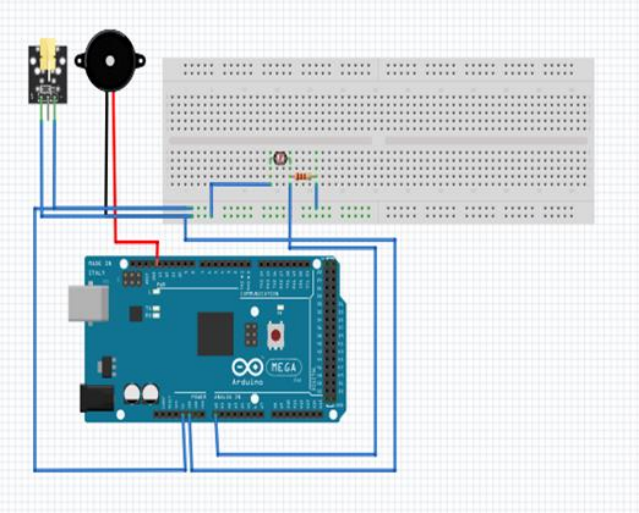


Figure 3. Scheme of laser sensor and buzzer connected to Arduinomega 2560.

II. The Result

The output, helped by the Arduino Mega 2560 board, temperature and humidity controller, an additional heater and a fan for increasing and decreasing the temperature is a MEGA BOX model.

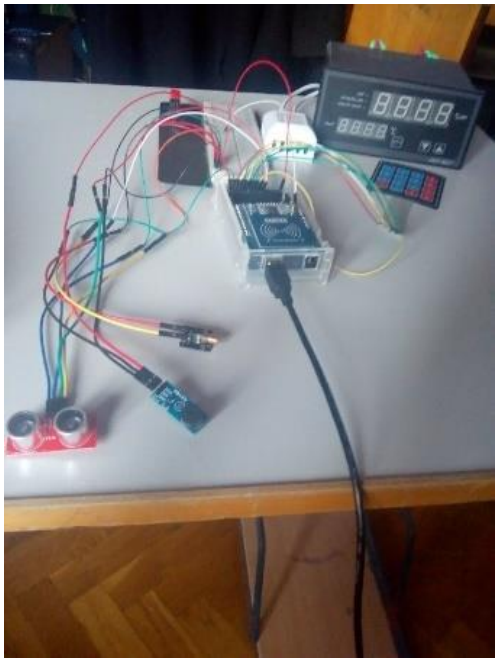


Figure 4. MK BOX components.

III. Conclusion

The model has not yet been made public. It will be used during school promotions among the students of the final grades of Ruma elementary schools, as well as at the fairs where our school will participate. This will send a message to young people that programming is not a "boo", but a great fun if you have a few electronic components, knowledge and a lot of effort and dedication.



Figure 5. MK BOX.

Acknowledgement

We would like to thank our mentor Vesna Ilic, for the time and effort dedicated to our project, as well as Pavle Kokanov and Uroš Milosavljevic, who initiated this project

References

[1] www.arduino.cc/reference/en/language/functions/analog-io/analogread/
[2] www.arduino.cc/reference/en/language/functions/time/delay/
[3] www.arduino.cc/en/Reference/ Void

Be Focused! Box

Jovica Jeremić, Aleksa Klještanović

Abstract - Project named BE FOCUSED! BOX is used to discourage students from using cell phones during a class. It enables more efficient classroom work with several diodes, resistors, connection conductors, microswitches, "RFID Arduino Shield" and the ARDUINO MEGA 2560 platform. Also, challenged students to change working habits and their mindset.

I. The Principle of Work and Implementation

Students are expected to leave their mobile phones in the BE FOCUSED!BOX at the beginning of the class. Using visual signals, a teacher checks (red - no, green - yes) the box whether all students have put away their cell phones. A pair of diodes is designed for each student.

The prototype is made of the following components:

- 4 red and 4 green LEDs (465 - 467,5 nm; 3-3,4 V)
- 4 resistors (220 Ω)
- 4 microswitches
- connection wires
- Arduino Mega 2560
- Arduino RFID receiver with transmitter

The project is based on 4 algorithms:

1. None of the students are logged in (all diodes are off)
2. Student checked in (red LED is on)
3. Student puts phone in the bearing (green LED is on)
4. The student picked up the phone from the bearing (red LED goes on again)

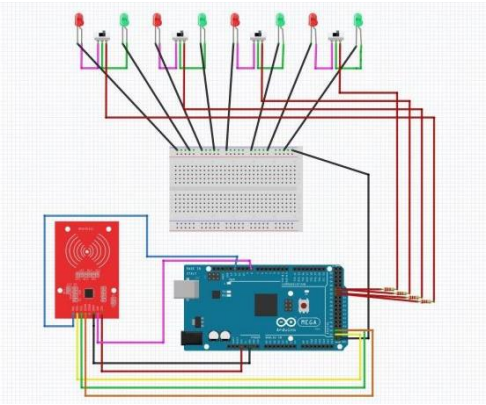
Some of the features used in the code are:

- int - this function was used to initialize variables (eg int activePin = 51;)
- long - function with extended number of variables
- pinMode - this function was used to set the output variable: The syntax of this function is: pinMode (pin, , mode); [1]

-pinMode (pin, mode);

- digitalWrite - this function was used to set / write values of variables; The syntax of this function is: digitalWrite (pin, value) pin is the number of the desired pin, while value is the pin state, which can have values HIGH and LOW (off) [2]
- else-if - The loop was used to activate the diodes recording the students' presence.

Jovica Jeremić and Aleksa Klještanović are students of the Vocational School “Milenko Brzak Uča” Ruma.



Picture 1. Visual connection scheme.

This prototype is programmed in Arduino Mega 2560 platform, e.i. The Arduino IDE editor. The first connection was made through proto boards that allow easier access and easier modifications. After the coding, a case was created to fit every mobile phone model. Part of the code describes the main idea:

```
void loop()
{
  if (!mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  Serial.print("UID of RFID TAGS:");
  for (byte i=0;i<mfrc522.uid.size;i++)
  {
    Serial.print(mfrc522.uid.uidByte[i],HEX);
    Serial.print(" ");
  }
  Serial.println();
  long code=0;
  for()byte i=0;i<mfrc522.uid.size;i++)
  {
    code=((code+mfrc522.uid.uidByte[i])*10);

  }
  Serial.print("Code is:");
  Serial.println(code);
  if (code == validCode)
  {
    Serial.println("Valid ID detected - activate lock!");
    Serial.println();
    digitalWrite (activePin,HIGH);
    delay(3000);
    digitalWrite (activePin, LOW);
  }
  }
```

Picture 2. Part of the code from the Arduino IDE editor.

II. The Results

The code implementation was done using the RFID Arduino shield. The keys are customized to respond to every mobile phone model manufactured in the last 10 years. In this way, students would perceive the obligation to leave their mobile phones in a box more as a fun than a punishment.



Picture 3. BE FOCUSED! BOX prototipe.

III. Conclusion

The final version of the BE FOCUSED! BOX does not have a proto board (it was omitted for the sake of cost savings), everything is connected by hand using conductors. The number of resistors is minimized so that each switch, not the diode, has one resistor.

Test enviroment was our classroom, and volunteers were our peers. Model worked as we predicted, but we have little resistance from participants in the test-phase to leave their mobile phones in the box.

If we want to continue with developing this idea, we need more financial support (for components), and to envolve psyhological support for participants.

Secondary idea was to encourage younger students not to shy away from programming and to make them aware that they can do whatever they imagine with the help of knowledge and electronic components.

Acknowledgement

We would like to thank our mentor Vesna Ilic, for her time and effort for the implementation of the project.

References

[1] https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/
[2] https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/

Ulazni deo optičkog prijemnika sa velikom impedansom

Tamara Nenadović, Ksenija Pešić

Sadržaj —U ovom radu prikazani su rezultati projektovanja i realizacije jednog optičkog prijemnika sa velikom impedansom i ekvilizacijom tako da se može koristi za bitske brzine do 50 Mb/s. Prototip je realizovan pomoću diskretnih tranzistora i njegovo funkcionisanje je potvrđeno eksperimentalno.

I. UVOD

Optički prijemnici mogu da se realizuju u jednoj od tri moguće konfiguracije: sa malom, velikom i prenosnom impedansom. U projektu je realizovan prijemnik sa velikom ulaznom impedansom i ekvilizacijom. Ovi prijemnici imaju veoma dobru osetljivost, zato što veoma mala fotostruja stvara relativno veliki napon na velikoj ulaznoj otpornosti. Propusni opseg ovakvih prijemnika je ograničen zbog velike ulazne impedanse koja u kombinaciji sa parazitnom kapacitivnošću dovodi do niske granične frekvencije na ulazu. Da bi se povećala granična frekvencija potrebno je koristiti ekvilajzer. Povećanje propusnog opsega je ograničeno i zbog toga se ne može očekivati veliko poboljšanje. Kako signal pri povećanju frekvencije slabi, potrebno je da ekvilajzer u tom istom opsegu poveća signal.

II. KOMPONENTE PROTOTIPA

Prototip sadrži sledeće komponente:

1. Na ulazu se nalazi fotodioda koja stvara malu fotostruju.
2. Kondenzator predstavljamodel ulazne parazitne kapacitivnosti od 18pF, koja dovodi do niske granične frekvencije prijemnika.
3. Nakon fotodiode i opterećenja predvidjen je bafer pojačavač sa jediničnim pojačanjem kao razdvojni stepen. Bafer je realiozovan uz pomoć diskretnog bipolarnog tranzistora u konfiguraciji sa zajedničkim kolektorom.
4. Bafer se povezuje sa bipolarnim tranzistorom sa zajedničkim kolektorom koji predstavlja još jedan razdvojni stepen.
5. Pojačavač i ekvilajzer se koriste u okviru jedne komponente. Pojačavač je realizovan kao bipolarni tranzistor u konfiguraciji sa zajedničkim emitorom. Dodavanjem kondenzatora u emitoru i kalema u kolektoru realizovan je i ekvilajzer.

Tamara Nenadović i Ksenija Pešić su studenti četvrte godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija (e-mail: tamara.nenadovic33@gmail.com , ksenija.pesic97@gmail.com).

6. Da bi kolo bilo prilagodjeno konektoru, čija je impedansa 50 Ω, korišćen je bipolarni tranzistor sa zajedničkim kolektorom, čije je pojačanje jedan.

III. IZBOR KOMPONENTI

U ovom projektu korišćene su sledeće komponente: dioda, otpornici, kondenzatori, kalem i bipolarni tranzistori. Pri odabiru komponenata vodilo se računa o nalaženju kompromisa izmedju cene i kvaliteta.

Parazitna kapacitivnost korišćene diode, ELPD333-3C-H0/L2, iznosi 18 pF. Na osnovu nje određene su vrednosti ostalih elemenata u kolu.

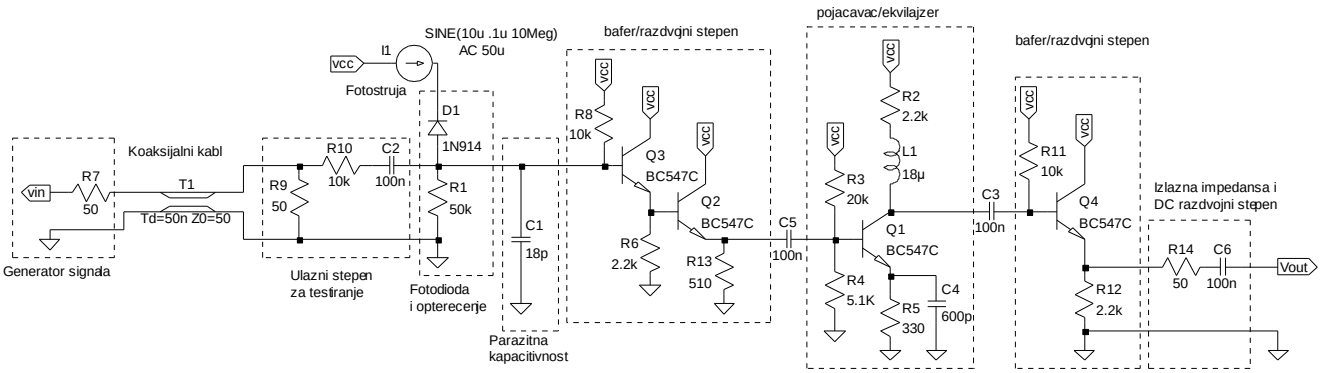
Idealni bipolarni tranzistori omogućavaju beskonačni propusni opseg, ali to nije slučaj i sa realnim. Vrednosti koje su korišćene za otpornosti, induktivnost i kapacitivnosti u kombinaciji sa idealnim tranzistorima nisu odgovarale vrednostima kada su korišćeni realni tranzistori. Tranzistori koji su omogućili postizanje zahtevanog propusnog opsega zajedno sa odgovarajućim vrednostima ostalih elementa su BC547C.

Nijedan otpornik nije imao kritičnu vrednost npr. reda MΩ koja bi u praksi sa velikom verovatnoćom onemogućila protok struje u toj grani, što bi značilo da je došlo do prekida u kolu.

IV. SIMULACIJA

Simulacija kola je urađena u grafičkom simulacionom okruženju LT Spice. Kompletna projektovana šema prikazana je na slici 1. Ulazni deo kola je strujni razdelnik koji služi za prilagođenje celog kola na konektor impedanse 50 Ω radi testiranja. Strujni izvor u šemi predstavlja fotostruju koja se generiše na fotodiodi. Da bi se simulirala parazitna kapacitivnost fotodiode, na ulaz je dodat kondenzator relativno velike kapacitivnosti 18 pF. Strujni izvor koji generiše malu fotostruju na impedansi od 50 kΩ stvara srazmerno veliki napon, pa se na taj način mogu detektovati veoma niski nivoi svetlosti. Na ulazu se nalazi fotodioda koja stvara malu fotostruju, koja na rednoj impedansi od 50 kΩ stvara srazmerno veliki napon pa se na taj način mogu detektovati veoma niski nivoi svetlosti. Sledeći je bafer, pojačavač sa jediničnim pojačanjem kao razdvojni stepen. Bafer je realizovan pomoću diskretnog bipolarnog tranzistora u konfiguraciji sa zajedničkim kolektorom. Osobina ove konfiguracije je da ima visoku ulaznu impedansu, koja je pogodna za realizaciju jedne od konfiguracija optičkog prijemnika, a malu izlaznu

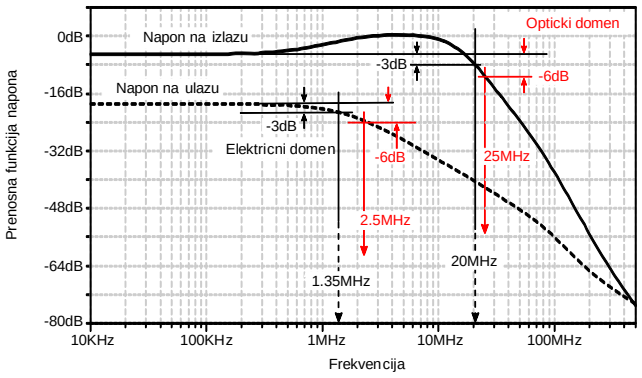
impedansu koja kao takva ne remeti ostatak kola. Bafer je povezan sa bipolarnim tranzistorom sa zajedničkim kolektorom koji predstavlja još jedan razdvojni stepen. Ovim je obezbeđena još veća izolacija dva kraja kola. Tako je u što većoj meri sprečena refleksija sa izlaza kola na ulaz.



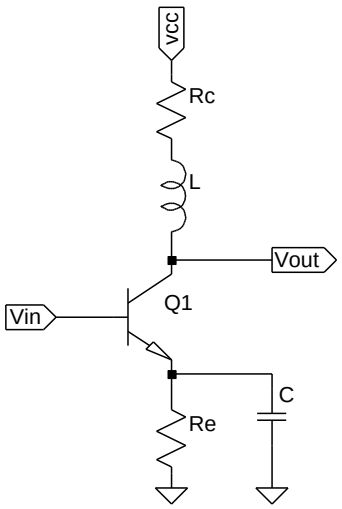
Slika 1. Kompletna šema prijemnika sa ekvilajzerom.

Pojačavač je realizovan kao bipolarni tranzistor u konfiguraciji sa zajedničkim emitorom. Dodajući kondenzator u emitoru i kalem u kolektoru dobija se bolja (poravnata) amplitudska karakteristika, što predstavlja ekvilajzer. Time je rešen problem malog propusnog opsega i povećana granična frekvencija prijemnika. Prenosna funkcija ulaznog dela kola ima prost pol. Ideja ekvilizacije je da se projektuje linearno kolo sa elementima čija prenosna karakteristika na identičnoj frekvenciji ima prostu nulu. U tom slučaju će se nula i pol međusobno poništiti i u idealnom slučaju neće biti ograničenja propusnog opsega. Sa druge strane, inženjerska praksa pokazuje da beskonačni propusni opseg ne postoji, pa zbog toga se očekuje da ostali parametri koje nisu razmatrani ograniče propusni opseg – ovog puta na znatno veću vrednost.

Korišćenjem ekvilajzera propusni opseg na izlazu je dostigao vrednost 22 MHz, što predstavlja znatno poboljšanje u odnosu na propusni opseg na ulazu ekvilajzera, koji iznosi oko 1.7 MHz.



Slika 2. Simulirane prenosne funkcije prijemnika u funkciji frekvencije na ulazu i izlazu prijemnika.



Slika 3. Pojačavač/ekvilajzer.

Izlazni napon pojačavača/ekvilajzer sadrži dve nule u svom izrazu

$$U_{\text{out}} = \frac{R_c}{R_e}\left(1+ j\omega\frac{L}{R_c}\right)\left(1+ j\omega R_e C\right)U_b ,$$

a u kolu postoje dva pola koja će se sa nulama međusobno poništiti.

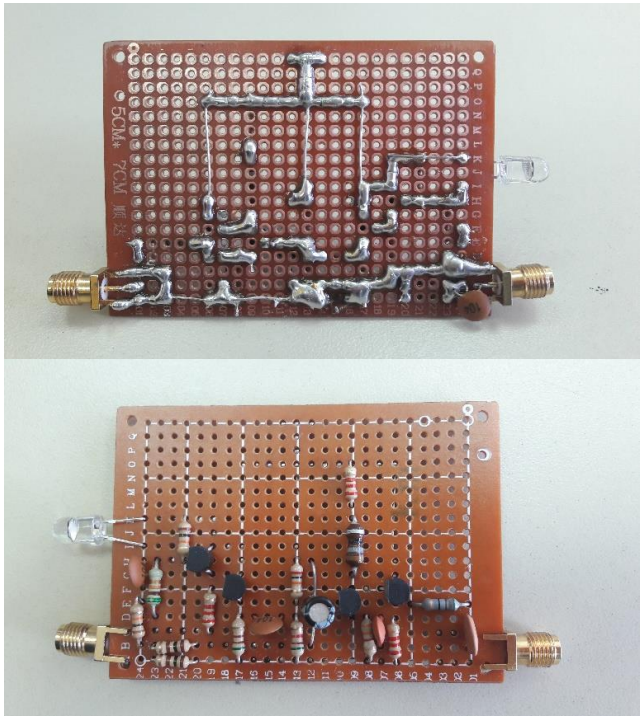
Nule se nalaze na frekvekcijama $f_1 = \frac{R_c}{2\pi L}$ i $f_2 = \frac{1}{2\pi R_e C}$.

Ako se zanemari naizmenični režim i posmatra jednosmerna komponenta pojačanje kola je $G = \frac{R_c}{R_e}$, gde je $R_c = 2.2k\Omega$, a $R_e = 330\Omega$.

Zamenom vrednosti iz simulacije dobija se vrednost pojačanja 6,67. Ovaj rezultat se poklapa sa vrednošću koja je dobijena pri očitavanju vrednosti sa grafika.

V. PRAKTIČNA REALIZACIJA

Vrednosti komponenata korišćenih u simulaciji omogućavaju postizanje propusnog opsega. Projekat je realizovan na raster-ploči, korišćenjem diskretnih komponenti.

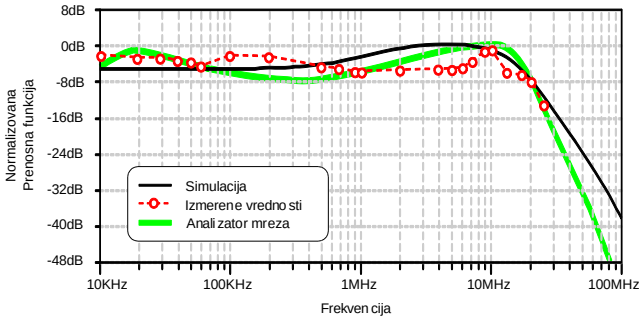


Slika 4. Izgled realizovanog prototipa na raster-ploči.

Nakon lemljenja glavnog dela kola (razdvojni stepeni i pojačavač sa ekvilajzerom) na raster-ploču stavljeni su i SMA konektori. Oni su omogućili lakše povezivanje sondi generatora i osciloskopa prilikom testiranja.

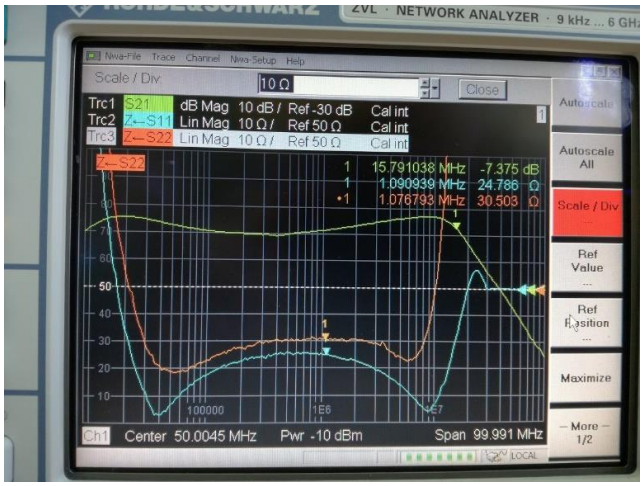
VI. REZULTATI MERENJA

Nakon realizacije izvršeno je testiranje prototipa. Od interesa je bilo potvrditi karakteristike dobijene simulacijom. Ploča je povezana sa osciloskopom GDS-2204, generatorom napajanja GPS-4303 i generatorom signala. Sonde generatora napajanja povezane su sa tačkom gde je napon Vcc i sa masom. Korišćeni napon napajanja je 5 V. Za frekvencije do 50 MHz korišćen je generator signala DDS-3X25, a za frekvencije do 5 MHz AFG-2105. Na ulazu kola pomoću kabla i SMA konektora povezan je generator signala, a na izlaz istim konektorom povezan je osciloskop. Na drugom kanalu osciloskopa pomoću sonde snimljen je signal na ulazu. Na taj način praćene su amplitude ulaznog i izlaznog signala istovremeno na različitim frekvencijama. Najpre su očitavane vrednosti ulaznog i izlaznog napona na različitim frekvencijama. Na osnovu njih nacrtan je grafik koji je upoređen sa grafikom dobijenim u LT Spice-u simulacijom. Sa slike 6. se može videti zadovoljavajuće slaganje između krivih. Prema tome,može se zaključiti da prototip zadovoljava unapred zadate kriterijume.



Slika 5. Normalizovana prenosna funkcija.

U cilju dodatne provere rezultata koje daje prototip korišćen je mrežni analizator Rohde&Schwarz. Prototip je za ovaj analizator povezan preko SMA konektora. Uporedjivanjem krivih dobijenih mrežnim analizatorom i simulacijom u LT Spice-u na slici 5. zapaža se da je prototip prilično dobro realizovan.



Slika 6. Rezultati merenja mrežnim analizatorom.

VII. ZAKLJUČAK

Zadatak projekta bio je da se realizuje optički prijemnik sa velikom impedansom i ekvilajzerom koji povećava ukupni propusni opseg, a time i bitsku brzinu na kojoj prijemnik može da funkcioniše ispravno. Izvršena je analiza zadatka i projektovan prototip čija je validnost ispitana korišćenjam simulatora. Prototip je zatim praktično realizovan korišćenjem raster ploče i diskretnih komponenata čije su vrednosti određene u simulaciji. Njegovo funkcionisanje je potvrđeno konkretnim merenjima na osciloskopu, a kasnije i mrežnim analizatoru.

Utvrdjeno je da se optički propusni opseg može povećati korišćenjem ekvilajzera. Propusni opseg koji je na ulazu bio 2.5 MHz, na izlazu je povećan 10 puta. S obzirom na ukupni propusni opseg od 25 MHz, zaključuje se da se

prijemnik može koristiti do bitskih brzina od 50 Mb/s, ako se koristi NRZ (Non Return to Zero) format signala u kojem je bitska brzina upravo dva puta veća od širine propusnog opsega.

ZAHVALNICA

Rad je deo studentskog projekta na ispitu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Ispit je u 6. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

LITERATURA

[1] Stephen B. Alexander, "Optical Communication Receiver Design", SPIE – The International Society for Optical Engineering, Bellingham, 1997.

[2] G. P. Agrawal, "Fiber-Optic Communication Systems", John Wiley & Sons, Inc., Hoboken, 2010.

[3] M&G Electronic, "BC547C", kataloški podaci, dostupno putem interneta na: http://www.farnell.com/datasheets/59764.pdf

[4] InterHIT elektronic, "SFH250", kataloški podaci, dostupno putem interneta na: https://www.mouser.com/ds/2/678/V02-3095EN-DS-SFH250_V-25Oct20110-909242.pdf

Planiranje kretanja modela Quadrotora u ROS okruženju

Fuad Hajdarević

Sadržaj - U ovom radu realizirani su modeliranje dinamike i razvoj simulatora za kvadrotorsku platformu, zasnovani na *Hector Gazebo* simulatoru. Kvadrotorska platforma je zasnovana na *Hector Quadrotor* modelu sa četiri propelera. Cilj rada je prilagođavanje *Hector Quadrotor* simulatora, te upravljanje i planiranje kretanja modela za potrebe izbjegavanja prepreka. Kompletno rješenje je realizirano kao *Robotic Operating System* (ROS) paket.

I. UVOD

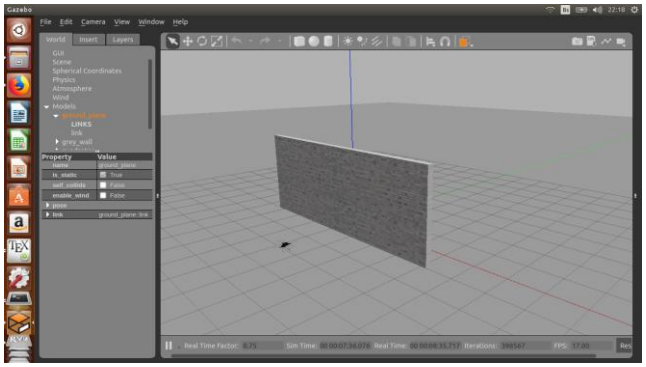
Današnji stepen razvoja tehnologije omogućio je primjenu bespilotnih letjelica (eng. *Aerial Vehicles* UAV) u različite istraživačke i komercijalne svrhe. *Unmanned* Bespilotne letjelice su,po svojoj definiciji, sve letjelice koje imaju mogućnost održavanja u zraku bez posade i, što je još važnije, imaju mogućnost da se njima upravlja. Bespilotne letjelice su već dugi niz godina u vojnoj upotrebi gdje predstavljaju koristan izvor informacija o položaju ili kretanju objekta ili subjekta koji se posmatra, te o terenu na kojem se on nalazi. Općeniti razvoj tehnologije u budućnosti, naročito razvoj digitalnih kamera srednjeg formata za fotogrametrijske potrebe, budući razvoj daljinskog prijenosa podataka, programske podrške, te samih letjelica omogućiti će još veću komercijalnu upotrebu popularnih tzv. dronova. Veća nosivost letjelica, duža autonomija leta, te potpuno samostalno upravljanje letom proširiti će mogućnosti upotrebe ovih letjelica na gotovo sva daljinska istraživanja. Na osnovu svega navedenog, može se zaključiti da je sve veća mogućnost upotrebe bespilotnih letjelica u civilne svrhe, što daje motivaciju za rad na ovakvom projektu. Nedostatak simulacijskih okruženja koja objedinjavaju stvarne uslove dinamike leta i jednostavnu integraciju sa postojećim softverima predstavljaju dodatnu motivaciju.[1]

II. DIZAJNIRANJE I MODELIRANJE OKRUŽENJA I MODELA

A. ROS Okruženje

ROS (eng. *Robot Operating System*) predstavlja skup više softverskih platformi, koji se koriste za razvoj razlicitih robotskih softvera. Sam ROS je razvijen 2007. godine na Stanford univerzitetu, te od tada predstavlja alat za simulaciju raznih vrsta robota, implementaciju raznih upravljačkih nivoa, na raznim nivoima apstrakcije, upravljanje i nadzor stvarnih robota, komunikaciju između različitih procesa, itd. ROS se zasniva na arhitekturi tipa grafova; procesiranje podataka se vrši koristeći čvorove, koji mogu vršiti prijem, odnosno slanje raznih poruka, koje uključuje podatke sa senzora, kao što su laseri, kamere, itd., podatke sa regulatora, podaci o samim stanjima robota, itd. Sama arhitektura je distribuiranog tipa, a treba istaći i postojanje ROS mastera, kao i to da sami čvorovi imaju osobinu multipleksiranja. [4]

Fuad Hajdarević je student master studija na katedri za Automatiku i elektroniku na Elektrotehničkom fakultetu u Sarajevu, Zmaja od Bosne, bb, 71000 Sarajevo, Bosna i Hercegovina. E-mail: fh15279@etf.unsa.ba.



Slika 1. Okruženje zajedno sa modelom *quadrotor*-a u *Gazebo* simulatoru.

B. Gazebo simulator

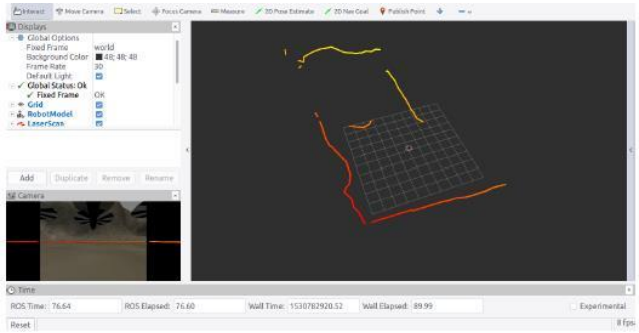
Gazebo simulator predstavlja alat u ROS-u koji na jednostavan način omogućuje 3D modeliranje i simuliranje robota u raznim okolinama. Sam *Gazebo* omogućava korištenje različitih algoritama upravljanja, dizajn robota, ispitivanje robota na različite uslove i okruženja, korištenje raznih senzora, itd.

C. Rviz simulator

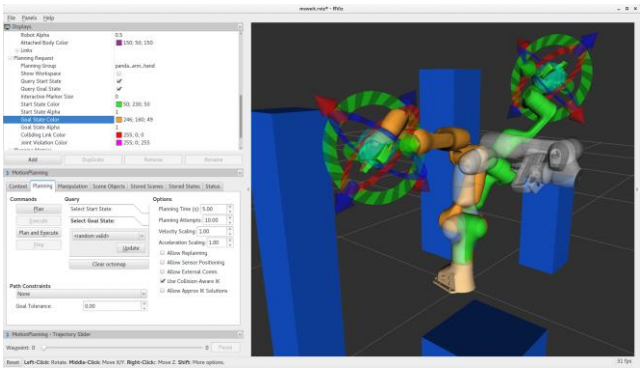
Da bismo bolje shvatili rad senzora, mapiranje, planiranje kretanja, ukratko ponašanje robota, koristimo Rviz simulator. Rviz (ROS vizualizacija) je 3D simulator za prikaz podataka senzora i informacija o stanju iz ROS-a. Pomoću rviz-a možete vizualizirati trenutnu konfiguraciju senzora na virtualnom modelu robota. Takođe možete prikazati reprezentativne prikaze vrijednosti senzora koje dolaze preko ROS tema, uključujući podatke kamere,

mjerenja infracrvenih lasera, sonarne podatke i još mnogo toga. [4]

Nakon što smo se upoznali sa principima korištenja Rviz simulatorom, prelazimo na dio koji se tiče konkretno mapiranja i planiranja kretanja. Da bismo mogli izvršiti mapiranje, koji je uslov na kraju i za planiranje, koristimo biblioteke OMPL (*Open Motion Planning Library)* i *Moveit*. Ove biblioteke služe kao *plugin* u Rvizu, koje koristimo da bismo mogli izvršiti mapiranje ili koristiti algoritme za planiranje. Na slici 3 je prikazan izgled *Motion Planning* interfejsa unutar Rviz, koji ćemo koristiti za planiranje i mapiranje. Nakon što smo instalirali potrebne biblioteke, možemo pristupiti rješavanju mapiranja i planiranja našeg modela.



Slika 2. Rviz simulator pokrenut za model *quadrotor*-a.



Slika 3. Prikaz Motion Planninga.

D. Quadrotor

U nastavku ćemo kratko opisati dinamički model *quadrotor*-a. Kretanje krutog tijela može se opisati zbirom svih sila F i momenata M koji djeluju na letjelicu. [3]

$$p^n = v^n \tag{1}$$
$$v^n = m^{-1} C_b^n F \tag{2}$$
$$\omega^b = J^{-1} M \tag{3}$$

Ovdje su p^n i v^n položaj i brzina težišta tijela u (inercijalni) navigacijskijom koordinatnom sistemu, ω^b je njegova ugaona brzina data u koordinatnom sistemu tijela i C_b^n je matrica rotacije koja transformira vektor tijela (indeks b) na navigacijske koordinate (indeks n). Masa m i inercija J *quadrotor*-a moraju se znati i biti estimirani vaganjem pojedinih komponenti i korištenjem geometrijskih modela. Vektor sile F sadrži potisak motora F_M, silu vuče F_d i vektor gravitacije F_g. Vektor zakretnog momenta M podijeljen je na zakretni moment M_M i moment povlačenja M_d. Sile i momenti povlačenja su dati:

$$F_d = -C_{d,F} C_n^b |v^n - v_w^n| (v - v_w) \tag{4}$$
$$M_d = -C_{d,M} |\omega^b| \omega^b \tag{5}$$

s dijagonalnim matricama koeficijenta povlačenja $C_{d,F}$ i $-C_{d,M}$ i vektorom vjetra v_w^n. Konačno, gravitaciona sila je data sa :

$$F_g = m C_n^b [00 q_e]^T \tag{6}$$

Pogonski sistem našeg kvadrotora sastoji se od četiri DC motora bez četkica. Pretpostavljajući vrlo nisku induktivnost motornih zavojnica, trenutačno vrijeme porasta može se zanemariti. Dinamično ponašanje motora se stoga pojednostavljuje U stabilnom stanju inducirani sidreni napon U_A ovisi o brzini vrtnje ω_M i struje sidra I_A:

$$U_A = m R_A I_A + \psi \omega_M \tag{7}$$

Elektromagnetski obrtni moment M_e za svaki motor dat je sa:

$$M_e = \psi I_A \tag{8}$$

S mehaničkim okretnim momentom M_m i inercijom motora J_M dolazi do promjene rotacije brzina te se može izračunati kroz:

$$\dot{\omega}_M = \frac{1}{J_M}(M_e - M_m) = \frac{1}{J_M}\left(\frac{\Psi}{R_A}(U_A - \Psi \omega_M) - M_M\right) \tag{9}$$

Hector Quadrotor je stvarni model *quadrotor*-a prema kome je modeliran i dimenzioniran virtuelni model za potrebe simulacije. *Hector Quadrotor* je dizajniran od strane dizajniran od strane *Team Hector Darmstadt* sa Tehničkog Univerziteta Darmstadt.Ono što je jako bitno da sam robot ima potpunu podršku unutar ROS paketa, te će se upravo neophodni fajlovi modelirati na bazi ovih fajlova. Ono što je jako bitno da sam robot ima potpunu podršku unutar ROS paketa, te će se upravo neophodni fajlovi modelirati na bazi ovih fajlova.



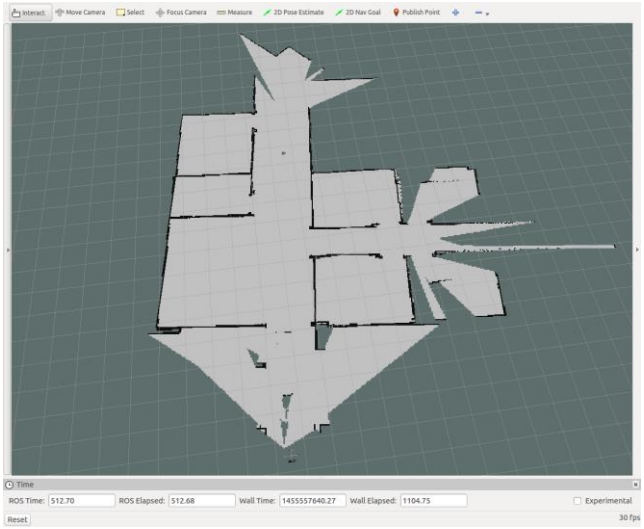
Slika 4. Izgled *Quadrotor*-a.

III. MAPIRANJE I PLANIRANJE

Mapiranje možemo izvršiti na dva načina, a to su 2D i 3D mapiranje. Kratko ćemo objasniti oba načina mapiranja, ali u realizaciji smo koristili 3D mapiranje.

A. 2D mapiranje

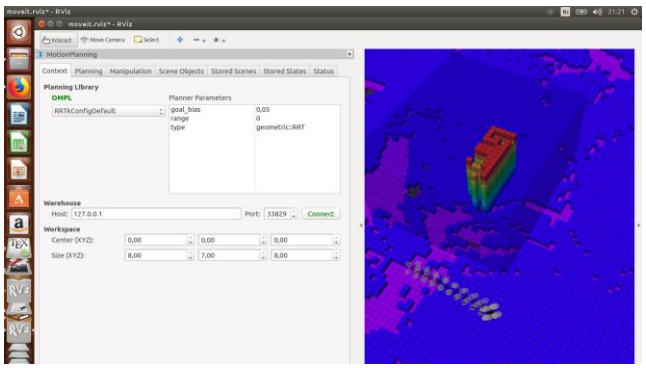
Da bi se izvršilo pravilno 2D mapiranje, koristimo *hector slam* paket. *Hector slam* koristi čvor *hector mapping* za učenje karte okoline i istovremeno procjenjuje 2D poziciju platforme pri brzini kadra laserskog senzora. Imena okvira i opcije za *hector mapping* moraju biti ispravno postavljeni. Prikaz 2D mapiranja je dat na slici 5.



Slika 5. Prikaz Rviza tokom 2D mapiranja okoline *Quadrotor*-a.

B. 3D mapiranje

U prethodnom poglavlju je kratko objašnjeno 2D mapiranje, gdje se za realizaciju koriste potrebni paketi iz porodice *Hector*. Postavlja se pitanje, kako izvršiti 3D mapiranje prostora. Da bismo ispravni izvršili 3D mapiranje, korištene su Octomape. Biblioteka OctoMap implementira 3D mapping pristup zauzetosti, pružajući strukture podataka i algoritme mapiranja u C++ posebno pogodnim za robotiku.



Slika 6. Izgled 3D mapiranja okoline koristeći Octomape.

C. Rapidly-Exploring Random Tree

Tokom istraživanja pokazalo se da od niza algoritama za planiranje, RRT se pokazao kao najefikasniji za planiranje kretanja *quadrotor*-a, pa ćemo u nastavku samo objasniti detaljno njega. Pokazalo se da je algoritam bazičnog RRT-a za planiranje puteva efikasan u velikim dimenzijama okruženja. Međutim, pronađene staze nisu nužno optimalne. Osnovni-RRT traži put od početnog čvora do ciljeva čvorova rastućem gomilom slučajnih stabala u a visoko-dimenziono okruženje sve dok ciljni čvor ne postane čvor slučajnog stabla. Drvo se gaji od početnog čvora do cilja čvora dodavanjem nove ivice i novog čvora na svakoj iteraciji. Algoritam bazičnog RRT-a funkcioniše sa tri koraka:

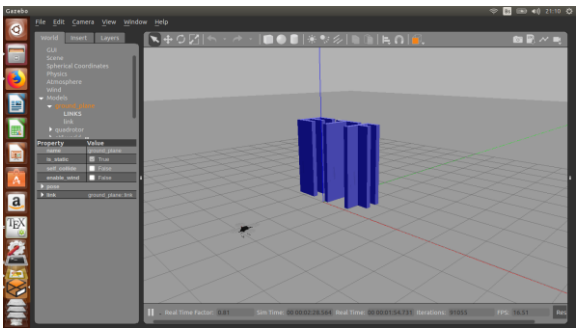
- Generisanje slučajnih čvorova: odabrati slučajne čvorove u 3D okruženju
- Ekspanzija: odabrati slučajni čvor i zatim izgraditi vezu između ovog čvora i najbližeg čvora u nasumičnom stablu.
- Uslov za prekidanje: pridružite se cilju čvoru na nasumično drvo.

IV. REZULTATI

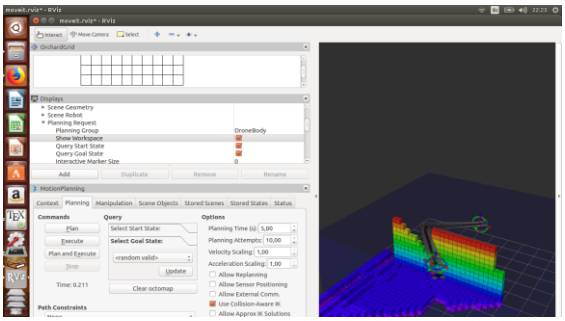
Nakon što smo detaljno objasnili platforme ROS, *Gazebo* simulator, Rviz simulator zajedno sa bibliotekama koje omogućavaju mapiranje i planiranje kretanje, možemo preći na realizaciju mapiranja i planiranja kretanja našeg modela, tj. *Quadrotor*-a. Radi ograničenih performansi računara koristili smo jednostavniju mapu radi bržeg izvršenja simulacije. Kreirali smo posebne fajlove za pokretanje simulacije i posebnu mapu, koja ima samo jedan zid kao objekt. Na slici 6 je prikazana ova situacija.

A. Kreiranje modela quadrotora u ROS-u

Sama instalacija *Hector Quadrotor*-a se može vršiti na više načina, da li je potrebno instalirati samo dio za simulaciju, ili i dio koji uključuje samu interakciju za hardverom. Za potrebe seminarskog rada koristit će se samo dio za simulaciju. Izgled pokrenutih fajlova je dat na slici 7.



Slika 7. Prikaz *Quadrotor*-a sa preprekom u *Gazebo* simulatoru.



Slika 8. Prikaz planiranja kretanja *Quadrotor*-a.

B. Kreiranje i pokretanje algoritma za planiranje kretanja optokoptera u ROS-u

Svi fajlovi kreirani su bazirani na *Hector Quadrotor* modelima. Sljedećim naredbama, pokrećemo simulaciju i launch fajl koji pokreće istraživanje mape. Da bi simulacija uspjela, potrebno je prvo istražiti okruženje, nakon toga možemo izvršiti planiranje kretanja u Rvizu, tj. preko Motion Planninga. Testiranjem mogućih algoritama za planiranje, zaključili smo da najbolje rezultate upravo daje RRT algoritam, dok ostale također daju zadovoljavajuće rezultate, ali su simulacije prespore i zahtjevaju jaču računarsku podršku. Na slici 8 možemo vidjeti realizaciju RRT algoritma za planiranje.

V. Zaključak

Kroz rad su objašnjene osnovne metode rada u ROS paketu, kao i osnove *Gazebo* simulatora, Rviz simulatora, te kako se vrši upravljanje *Quadrotor*-om. Opisane su osnovne procedure kojima se kreira okruženje za let *Quadrotor*-a, te principi korištenja alata i programa za mapiranje okruženja i planiranja kretanja u tom okruženju.

Literatura

[1] Sefer,Enela,"*Modeliranje dinamike i razvoj simulatora za oktokoptersku platformu zasnovanom na Hector gazebo simulatoru*", Bachelor rad, 2018

[2] Jasmin Velagic, Lectures in course "*Mobile Robotics*", Faculty of Electrical Engineering Sarajevo, Sarajevo, 2018.

[3] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf *Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo*, Technische Universität Darmstadt, Japan, 2012.

[4] *The Robot Operating System* (ROS) dostupno na: https://www.ros.org/

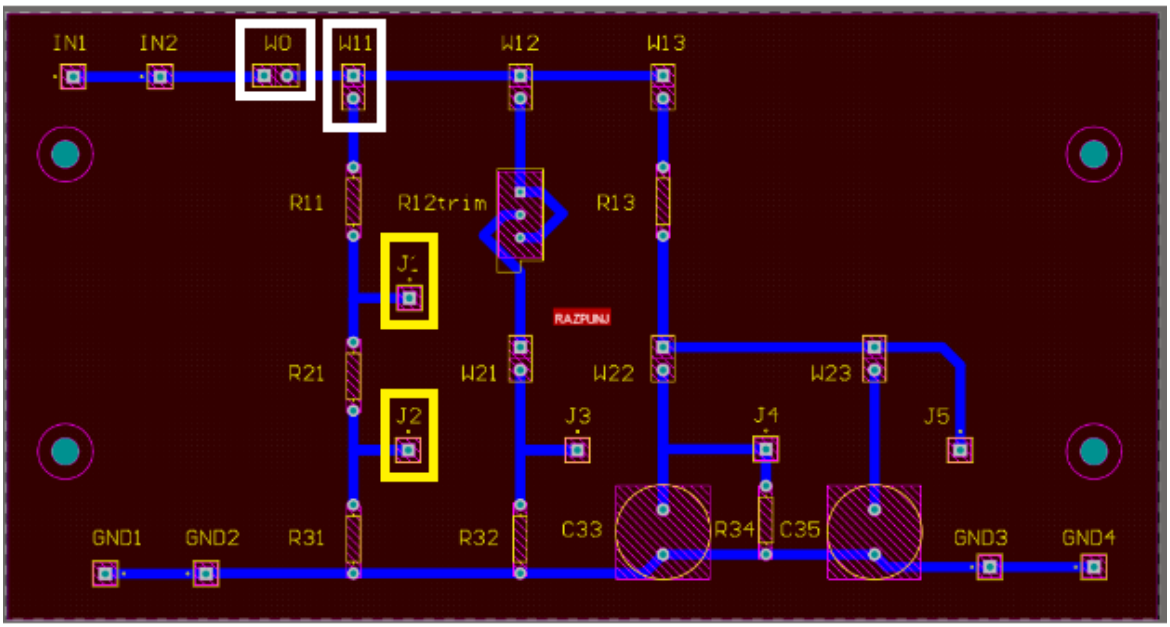
Maketa naponskog razdelnika

Milica Danković, Mihajlo Mihajlović, Voin Janjić

Sadržaj - U ovom radu opisan je princip rada naponskog razdelnika. Koristi se u svrhu izračunavanja pada napona na potrošačima. Pad napona na potrošačima očitavamo korišćenjem mernih instrumenata.

I. UVOD

Na slici 1 prikazana je univerzalna šema za realizaciju velikog broja projekata (naponski razdelnik sa tri otpornika, naponski razdelnik sa dva otpornika, punjenje kondenzatora preko naponskog razdelnika, punjenje kondenzatora preko otpornika, pražnjenje kondenzatora, …). Za projektovanje električne šeme i štampane ploče iskorišćen je softverski paket Altium Designer [1]. U ovom projektu realizovali smo samo jedan deo šeme kao što je ilustrovano na slici 2. Postavljanjem žičanih kratkospojnika (jumper-a) W0 i W11 (belom bojom označene pozicije na slici 1) posmatrali smo padove napona na otpornicima, odnosno snimali napone na izlazima J1 i J2 (žutom bojom zaokruženi izlazi na slici 1). Ostatak električne šeme nije korišćen, ali po ugledu na ovu vežbu može se jasno zaključiti kako bi bile realizovane i ostale laboratorijske vežbe.

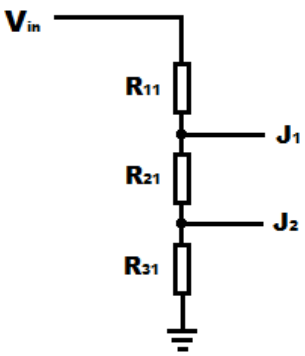


Slika 1. Univerzalna šema za realizaciju laboratorisjkih vežbi.

II. OPŠTI PRIMER

Primer naponskog razdelnika se sastoji od tri otpornika, odnosno potrošača, kao što je ilustrovano na slici 2.

Treba napomenuti, kao što je prikazano na slici 2 za tri otpornika, naponski razdelnik se dobija i spajanjem tri električne impedanse (može da se sastoji od kombinacije elemenata, kao što su otpornik, kalem i kondenzator) [2, 3]. Primenom Omovog zakona izračunavamo vezu između ulaznog napona (V_{in}) i napona na J_1 i J_2:



Slika 2. Opšti primer naponskog razdelnika sa tri različita otpornika.

Milica Danković, Mihajlo Mihajlović i Voin Janjić su učenici TŠ „Rade Metalac", Dimitrija Tucovića bb, Leskovac, E-mail: milicadkendi@gmail.com, mihajlobuss001@gmail.com, voin.janjic@gmail.com

$$I = \frac{V_{in}}{R_{11}+R_{21}+R_{31}} \quad (1)$$

$$V_{J2} = R_{31} * I \quad (2)$$

$$V_{J2} = \frac{V_{in}}{R_{11}+R_{21}+R_{31}} * R_{31} \quad (3)$$

$$V_{J2} = \frac{R_{31}}{R_{11}+R_{21}+R_{31}} * V_{in} \quad (4)$$

$$V_{J1} = R_{31} * I + R_{21} * I \quad (5)$$

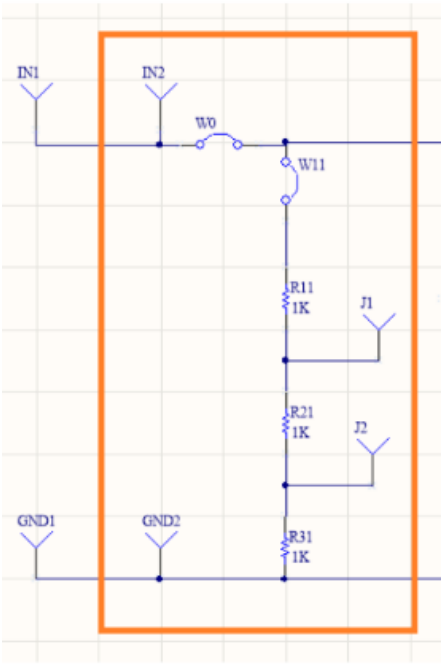
$$V_{J1} = (R_{31} + R_{21}) * I \quad (6)$$

$$V_{J1} = (R_{31} + R_{21}) * \frac{V_{in}}{R_{11}+R_{21}+R_{31}} \quad (7)$$

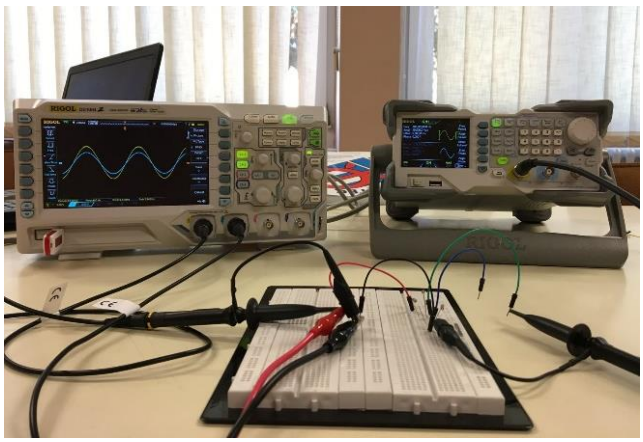
$$V_{J1} = \frac{R_{31}+R_{21}}{R_{11}+R_{21}+R_{31}} * V_{in} \quad (8)$$

III. NAPONSKI RAZDELNIK REALIZOVAN NA PROTO PLOČICI

Laboratorijska provera osmišljene vežbe (radi jednostavnosti i preglednosti korišćena električna šema prikazana je na slici 3) realizovana je korišćenjem protoploče, otpornika, kratkosponika, osciloskopa [2] i generatora signala [3] (fotografija napravljena prilikom eksperimentisanja prikazana je na slici 4).



Slika 3. Električna šema za realizaciju naponskog razdelnika sa tri različita otpornika.



Slika 4. Realizacija laboratorijske vežbe.

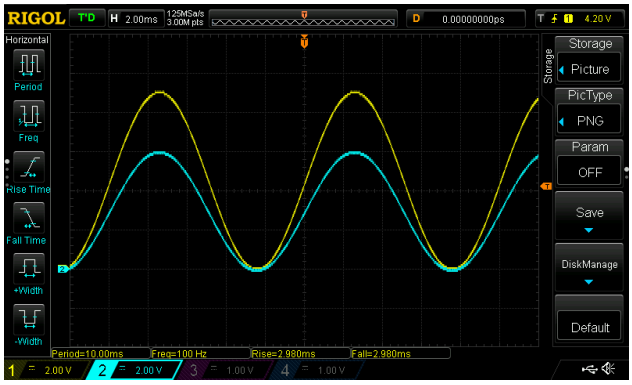
Imajući u vidu da je tokom realizacija vežbi korešćena najsavremenija oprema, bilo je moguće realizovati naponski razdelnik sa različitim oblicima napona koji se dovode na ulaz *Vin*. Kao ilustracija, u radu su prikazani rezultati za dva oblika napona koji su dovođeni na ulaz *Vin*: sinusnog oblika i pravougaonog oblika.

Tabelarni prikaz vrednosti elemenata u kolu naponskog razdelnika i paramatera signala koji je doveden na *Vin* date su u tabeli 1.

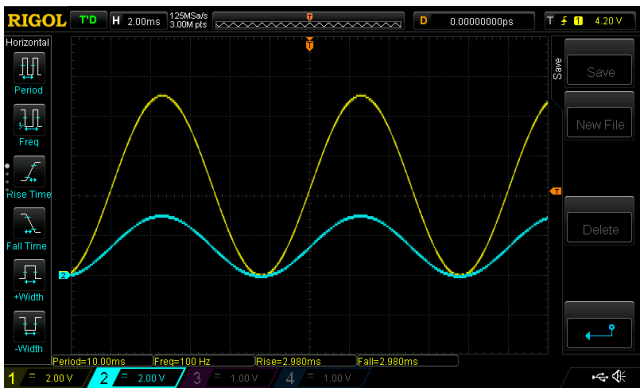
TABELA I
PRIKAZ VREDNOSTI ELEMENATA U KOLU I PARAMATERA SIGNALA

Oblik signala	V_{PP} (V)	f (Hz)	V_{offset} (V)	R_{11} (kΩ)	R_{21} (kΩ)	R_{31} (kΩ)
	9	100	4.5	1	1	1

Na slikama 5 i 6 prikazani su rezultati eksperimenta sa osciloskopa za formirani razdelnik napona. Ulazni signal prikazan je žutom bojom, vidi se da je napon sinusnog oblika, uvek pozitivan. Amplituda ulaznog napona je 9 Vpp. Signal plave boje na slici 5 je napon na izlazu J1. Amplituda ovog napona je 6 V, što je u saglasnosti sa proračunom na osnovu jednačine (8).

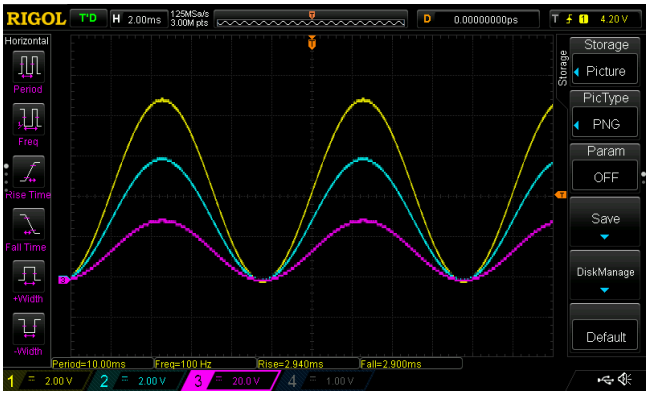


Slika 5. Ulazni (sinusni) i napon na izlazu J1 snimljeni pomoću osciloskopa.



Slika 6. Ulazni (sinusni) i napon na izlazu J2 snimljeni pomoću osciloskopa.

Na slici 6 signal plave boje je napon na izlazu J2, čija je amplituda 3 V, što je u saglasnosti sa proračunom na osnovu jednačine (4). Da bi uporedili sve signale u kolu razdelnika napona, priključena je još jedna sonda osciloskopa. Na slici 7 prikazani su sledeći signali: ulazni napon žutom bojom, amplitude 9 Vpp, napon na izlazu J1, plavom bojom, amplitude 6 Vpp, dok je ljubičastom bojom označen napon na izlazu J2, amplitude 3 Vpp. Na osnovu rezultata sa slike 7 može se zaključiti da izlazni napon uvek prati promenu ulaznog napona, dok vrednost zavisi od vrednosti otpornosti u kolu kao što je pokazano jednačinama (4) i (8).



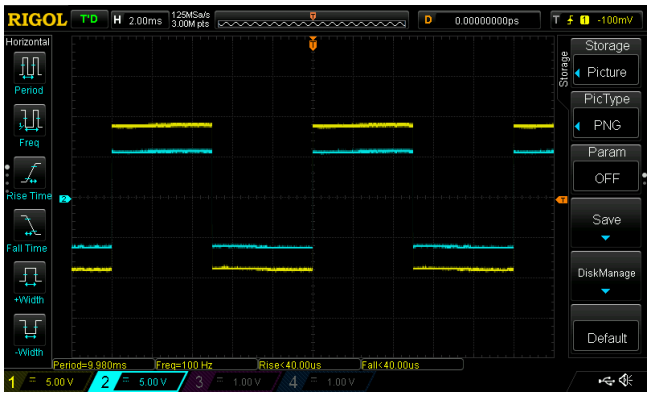
Slika 7. Ulazni (sinusni) napon i naponi na izlazima J1 i J2.

U tabeli 2 prikazane su vrednosti parametara ulaznog signala pravougaonog oblika i vrednosti otpornosti u kolu razdelnika napona. U ovom slučaju ulazni signal pravougaonog oblika u jednoj poluperiodi ima pozitivne, a u drugoj poluperiodi negativne vrednosti, tako da se signal menja sa -9 V na +9 V.

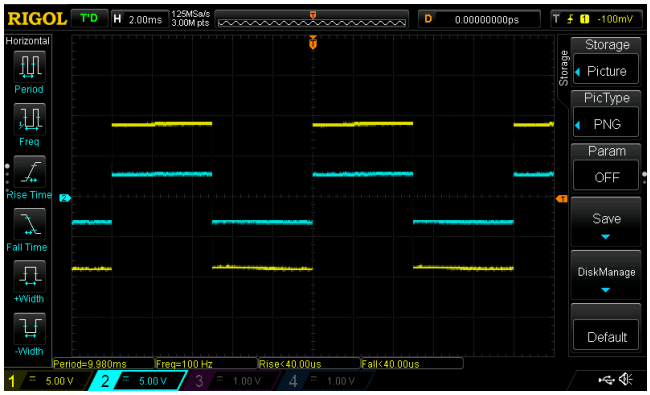
TABELA II
PRIKAZ VREDNOSTI ELEMENATA U KOLU I PARAMATERA SIGNALA

Oblik signala	V_{PP} (V)	f (Hz)	V_{offset} (V)	R_{11} (kΩ)	R_{21} (kΩ)	R_{31} (kΩ)
	18	100	0	1	1	1

Na slikama 8, 9 i 10 prikazani su eksperimentalni rezultati usnimljeni pomoću osciloskopa za slučaj pravougaonog ulaznog napona. Ulazni napon na svim slikama označen je žutom bojom, može se uočiti da ulazni napon ima pozitivne i negativne vrednosti, te da se signal menja amplitude napona na svakih 5 ms. Plavi signal na slici 8 je napon na izlazu J1, čija je vrednost 6 V ili -6 V, u zavisnosti od ulaznog napona. Slično, plavi signal na slici 9 je napon na izlazu J2, čija se vrednost menja između -3 V i +3 V. Ovi rezultati se potvrđuju jednačinama (4) i (8). Uporedni pregled svih signala: ulaznog i izlaznih J1 i J2 prikazan je na slici 10. Signal plave boje je izlazni napon J1, dok je ljubičastom bojom predstavljen napon na izlazu J2.



Slika 8. Ulazni (pravougaoni) i napon na izlazu J1 snimljeni pomoću osciloskopa.



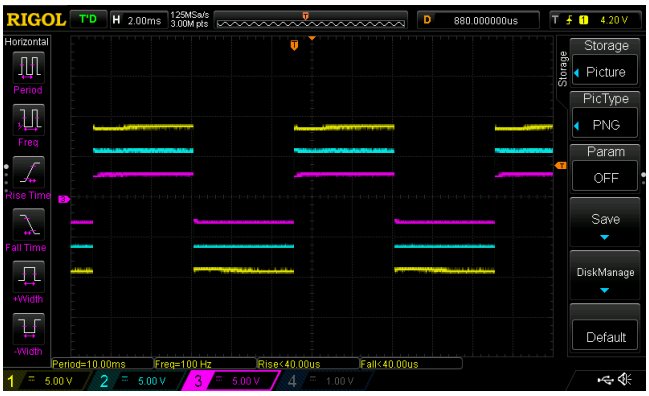
Slika 9. Ulazni (pravougaoni) i napon na izlazu J2 snimljeni pomoću osciloskopa.

Primećujemo da tip signala nije bitan te se raspodela napona na potrošačima uvek proporcionalno smanjuje (opada). S toga, uvek na izlazima J_1 i J_2 dobijamo napone manjih amplituda, istih talasnih oblika i frekvencija.

IV. ZAKLJUČAK

Realizovana je maketa naponskog razdelnika pogodna za realizaciju laboratorijskih vežbi. Kompletna teorija praktično je proverena u potvrđena u laboratorijskim uslovima. Deo rezultata prikazan je u radu. Treba

napomenuti da je vežba prilagođena srednjoškolskom uzrastu i nivou znanja koje oni poseduju.



Slika 10. Ulazni (pavougaoni) napon i napon na izlazima J1 i J2.

ZAHVALNICA

Ovaj rad deo je projekta “Idemo i mi na IEEESTEC konferenciju”, finansiranog od strane Centra za promociju nauke. Autori se zahvaljuju prof. dr Danijelu Dankoviću i asistentu Milošu Marjanoviću na pomoći u realizaciji rada.

LITERATURA

[1] Altium Designer. Dostupno na: https://www.altium.com/altium-designer/
[2] https://elektrotehnikamesbor.files.wordpress.com/2014/01/oet2tehnicari.pdf
[3] https://www.scribd.com/doc/135233335/Elektronika-1-Ratko-Opacic
[4] Generator signala RIGOL DG1022, https://www.rigol.eu/products/waveform-generators/
[5] Osciloskop RIGOL DS 1054, https://www.rigol.eu/products/digital-oscilloscopes/

Realizacija modela IoT sistema primenom Raspberry Pi

Aleksandra Dinić

Sadržaj - Mreža fizičkih objekata - uređaja, vozila, zgrada i drugih predmeta - sa ugrađenom elektronikom, softverom i senzorima, povezanih putem Interneta, što im omogućuje prikupljanje i razmenu podataka, predstavlja Internet stvari– Internet of Things (IoT). Uvođenjem IoT u različite sfere života moguće je olakšati život, živeti zdravije, povećati bezbednost. Cilj ovog rada je realizacija jednog ovakvog sistema koji omogućava korisnicima da sa bilo kog mesta manipulišu svojim uređajima u kući ili na poslu. Postignut je određeni stepen kontrole koji se ne može dobiti jednostavnim korišćenjem tradicionalnih uređaja. Model IoT sistema realizovan je korišćenjem Raspberry Pi platforme, Nginx servera sa dodatkom PHP-a i napravljena je dinamička web stranica pomoću koje se kontroliše rad diode.

I. UVOD

Razvoj Interneta je izmenio način na koji ljudi komuniciraju i koegzistiraju. Omogućava da informacije budu lako dostupne svakom korisniku. Kontinuirano integrisanje svih sfera života je nastavljeno, pa se tako poslovanje, ekonomija i industrija sve više oslanjaju na Internet usluge, u kome je čovek centralni faktor oko kojeg se razvijaju nove mogućnosti i usluge. Brz način života zahteva razvoj i projektovanje sistema kućne automatizacije. Takvi sistemi su uglavnom kreirani pomoću inteligentnih Internet of Things (IoT) uređaja. IoT je integrisani sistem komunikacionih uređaja u kojima je svaki uređaj sposoban da samostalno izvršava zadatke. U ovom radu realizovan je sistem daljinskog upravljanja mikroračunarom putem Interneta. Takav sistem omogućava korisnicima da sa bilo kog mesta manipulišu svojim uređajima u kući ili na poslu. U cilju jednostavnog praćenja rada sistema, omogućena je vizuelna prezentacija ostvarenih aktivnosti [1, 2, 5, 6].

Model IoT sistema koji je predstavljen u ovom radu ralizovan je pomoću Raspberry Pi 3 platforme, Nginx servera, LED dioda i web stranice. Faze kroz koje je potrebno proći prilikom realizacije ovog projekta su: instalacija Raspbian operativnog sistema na SD kartici, povezivanje U/I uređaja, povezivanje LED diode, instalacija web servera Nginx, instalacija dodatnog paketa za PHP, izrada dinamičke web stranice, kontrolisanje GPIO pinova korišćenjem PHP programskog jezika.

II. RASPBERRY PI

Raspberry Pi je mini računar koji ima za cilj dodatno obrazovanje studenata računarskih nauka. Od 2006. do 2008. godine dizajnirano je i napravljeno više verzija Raspberry Pi. U osnovi postoje četiri modela i njihove verzije (generacije), Model A, Model B, Compute Modul i Zero. Pri izradi projekta korišćen je model Raspberry Pi 3-B koji je zasnovan na Broadcom čipu (BCM2837) sa integrisanim ARM procesorom i VideoCore IV GPU (*Graphical Processing Unit*). BCM2837 SoC je zasnovan na procesoru koji se sastoji od četiri 64-bitna ARM Cortex-A53 jezgra radnog takta od 1.2 GHz sa 32 kB (Level 1) i 512 kB (Level 2) keš memorije, dok je količina radne memorije 1 GB LPDDR2. SD kartice se koriste za smeštanje operativnog sistema, i kao programske memorije. Na većini Raspberry Pi uređaja postoji od 1 do 4 USB porta, HDMI, kompozitni audio/video izlaz, 3.5 mm audio priključak. Na ploči je i 40-pinski konektor sa rasterom od 2.54 mm preko koga je omogućen pristup ka 27 GPIO (General-purpose input/output) pinova. Neki modeli, kao što je model Raspberry Pi 3-B koji je korišćen u radu (slika 1), imaju RJ45 Ethernet port, kao i inegrisan WiFi 802.11n i Bluetooth 4.1. Ovaj model je odabran zbog svoje dostupnosti i jednostavnog načina postavljanja bežičnih opcija [1, 3].



Slika 1. Raspberry Pi model 3-B.

III. OPERATIVNI SISTEM

Da bi imali funkcionalan Raspberry Pi, potrebno je na njemu instalirati operativni sistem (OS). U izradi projekta korišćen je Raspbian OS koji je zvanični operativni sistem za Raspberry Pi. Operativni sistem je u potpunosti optimizovan za rad na uređaju i veoma jednostavan. Ne predstavlja samo skup osnovnih paketa potrebnih za rad već postoji preko 35000 paketa pripremljenih za jednostavnu instalaciju na uređaju.

Aleksandra Dinić, Studijski program Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Srbija. E-mail: aleksandra.dinic@elfak.rs

A. Instalacija operativnog sistema

Korišćenje softvera NOOBS najlakši je način za instaliranje Raspbiana na SD karticu. Napredniji korisnici koji žele da instaliraju određeni operativni sistem treba da koriste image OS. Jednostavniji pristup podrzumeva korišćenje NOOBS softvera sa jednostavnim kopiranjem *.img* falja odabranog OS-a na SD karticu. Uloga NOOBS-a je upravo u upravljanju procesom instalacije OS-a bez upliva korisnika [3].

Napredniji pristup podrazumeva klasičnu instalaciju OS-a, koja je korišćena tokom rada a postupak je sledeći:

1. Preuzimanje falova BalenaEtcher i Raspbian OS sa *www.raspberrypi.org* sajta.
2. Potom se vrši pripremanje SD kartice. Poželjno je da bude veličine najmanje 8 GB i po mogućstvu klase 10 zbog brzine čitanja i upisa podataka.
3. Zatim se instalira balenaEtcher softver. *BalenaEtcher* je grafički alat za upisivanje na SD karticu koji radi na Mac OS, Linux i Windows OS.
4. Nakon prebacivanja .img falja Raspbian OS na SD karticu, istu je potrebno ubaciti u Raspberry Pi čime se stvaraju preduslovi za pokretanje operativnog sistema.
5. Potom se vrši povezivanje monitora putem HDMI kabla, kao i povezivanje tastature i miša koji su nam potrebni za korišćenje putem USB interfejsa.
6. Nakon povezivanja sledi pokretanje OS što podrazumeva dovođenje napajanja putem micro USB konektora sa računara.
7. Završni korak predstavlja *boot*-ovanje OS-a.

Nakon pokretanja operativnog sistema, pre bilo kakve upotrebe, poželjno je konfigurisati (podesiti) operativni sistem. To se najlakše može uraditi ukoliko se u Terminalu pokrene komanda *sudo raspi-config*. Dostupna podešavanja su: promena šifre, uključivanje i isključivanje interfejs parametara, izbor okruženja u kojem će se računar startovati, ažuriranje operativnog sistema i dostupnih paketa, itd.

IV. WEB SERVER

Server je uređaj koji pruža funkcionalnost drugim programima ili uređajima koji se nazivaju klijenti. Web serveri obrađuju datoteke napisane na različitim programskim jezicima kao što su PHP, Python, Java i drugi. Oni ih pretvaraju u statičke HTML strane i njih prikazuju korisniku. To je alat odgovoran za pravilnu komunikaciju između servera i klijenta.

Kako bismo mogli da pomoću web stranice upravljamo LED diodama priključenim na GPIO (eng. General-purpose Input/Output) pinove Raspberry Pi, neophodno je da imamo server na kome će se nalaziti naša web stranica. Osim ove opcije, moguće je podatke slati i na cloud, međutim u toku rada odlučeno je da se sve nalazi na lokalnom serveru.

A. Webserver Nginx

Nginx je webserver koji se takođe može koristiti kao *reverse proxy load balancer*, *mail proxy* i HTTP keš. Može se nazvati i *light* verzijom Apache servera. Nginx je besplatni i *open-source* softver, koji se izdaje pod uslovima BSD licence. Nginx je napisan sa eksplicitnim ciljem da nadmaši web server Apache, koristeći statičke datoteke. Nginx koristi značajno manje memorije nego Apache i može da obradi otprilike četiri puta više zahteva u sekundi. Međutim, ovo povećanje performansi dovodi do smanjenja fleksibilnosti.

Instalira se jednostavno i konfiguriše u par koraka. Za instalaciju dovoljno je uneti sledeću komandu u terminal: *sudo apt-get install nginx*.

Nakon završene instalacije server se pokreće komandom: *sudo /etc/init.d/nginx start*.

Podrazumevano Nginx kreira test .html fajl u web folder. Podrazumevana web stranica (slika 3.) se prikazuje kada se kao adresa ukuca http://localhost/ u *web browser-u* na Raspberry-ju ili kada se u browser-u drugog uređaja povezanog na naš AP ukuca adresa http://192.168.4.1 (IP adresa koju smo prethodno dodelil kao statičku adresu uređaja).



Slika 2. Podrazumevana web strana Nginx server.

B. Instalacija dodatnog paketa

Za prikaz dinamičke web stranice potrebno je korišćenje PHP (*PHP: Hypertext Preprocessor*) jezika. Potrebne aplikacije instaliramo sledećom komandom:

sudo apt-get install php-fpm.

Kako bismo omogućili PHP u Nginx treba izmeniti sadržaj sledećeg fajla

cd /etc/nginx
sudo nano sites-enabled/default

Potrebno je pronaći liniju u kojoj se nalazi „*index index.html index.htm;*“ i dodati *index.php* odmah iza *index*. Počevši od linije „*# pass the PHP scripts...*“ potrebno je obrisati karaktere „*#*“ na pojedinim mestima, čime se dobijaju linije koda, kao što je:

```
# pass the PHP scripts to FastCGI server listening on
127.0.0.1:9000
#
location ~ \.php$ {
include snippets/fastcgi-php.conf;
# With php5-cgi alone:
# fastcgi_pass 127.0.0.1:9000;
# With php5-fpm:
fastcgi_pass unix:/var/run/php5-fpm.sock;
}
```

Nakon sačuvanih izmena potrebno je ponovo pokrenuti konfiguracioni fajl komandom:

sudo /etc/init.d/nginx reload.

Omogućavanjem PHP-a naša stranica pored statičkog može imati i dinamički sadržaj.

U cilju realizacije modela sistema daljinskog upravljanja mikroračunarom Raspberry Pi putem Interneta, posle instalacije Raspbian OS i Nginx webservera, potrebno je kreirati web stranicu preko koje će se uključivati i isključivati LED dioda priključena na GPIO pin 21. Treba napomenuti da oznake GPIO pinova koje se koriste u softveru ne predstavljaju realne pozicije pinova na korisnički orijentisanom konektoru ploče. Pinovi su na ploči raspoređeni tako da je na poziciji 39 konektora GND pin tj. masa, dok je pin 21 procesora zapravo izveden kao poslednji pin ploče tj. 40. pin.

V. IZRADA WEBSTRANICE

A. Front-end

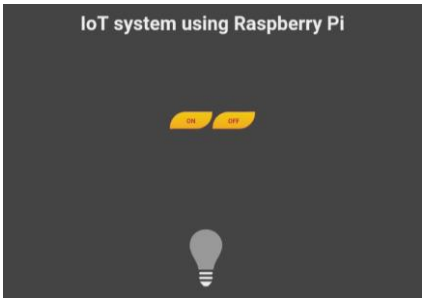
Jedna web stranica sadrži svoj *front-* i *back-end*. GUI (*Graphical User Interface*) je jako važan za ovakav sistem, kako bi korisnik već na prvi pogled mogao da shvati kako on zapravo i funkcioniše. Tokom kreiranja webstranice za ovaj projekat korišćeni su HTML5 i CSS3.

HTML (*HyperText Markup Language*) je prezentacijski jezik za izradu web stranica. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertext dokumenta.

CSS (*Cascading Style Sheets*) je stilski jezik koji služi za opis prezentacije dokumenta napisanog pomoću HTML jezika. CSS služi za kreiranje *layout*-a i daje ukrašavanje i lepotu stranici. CSS-om se mogu opisati bilo koje vrste XML dokumenata uključujući i sam XML, SVG ili XUL. Uz korišćenje HTML i Javascript jezika, CSS daje vizuelnu strukturu i opis web sajtova, web aplikacija kao i aplikacija za mobilne telefone.

Pomoću CSS-a kreiran je objekat sijalica, koja se uključuje ili isključuje onda kada se uključi/isključi dioda povezana na odgovarajući pin Raspberry Pi-a. Na ovaj način se dobija grafička prezentacija onoga što se realno dešava pritiskom na jedan od dva tastera (button) "ON" ili "OFF" na web strani (slika 4). Status diode se vizuelno prezentuje svim korisnicima koji su trenutno pristupili web stranici.

Uključivanje/isključivanje sijalice omogućeno je dodavanjem tj. ukidanjem klase „active“, a realizuje se pritiskom na odgovarajući taster na stranici.



Slika 3. Web stranica za uključivanje/isključivanje LED diode.

B. Back-end

Back-end predstavlja deo koji se izvršava na web serveru. Uloga serverskog dela sajta je ključna u svakom većem ili bitnijem sajtu, s obzirom na to da se serverski jezik smatra neuporedivo sigurnijim od klijentskih. Njegov zadatak je da prihvata zahteve korisnika, da ih obrađuje, da prihvata podatke od korisnika, analizira, upisuje u bazu podataka, preuzima iz baze podataka i vraća korisniku.

PHP (*PHP: Hypertext Preprocessor*) je specijalizovani skriptni programski jezik koji se koristi za izradu dinamičkih webstranica, tj. za dinamičko generisanje HTML koda. Na ovaj način PHP kod koji je generisao stranu se ne može videti i prikazuje se čisti HTML kod. PHP je jedan od najpopularnijih *server-side* jezika za izradu dinamičkih web sajtova. Neki od najpoznatijih web sajtova napravljeni baš pomoću PHP jezika su: Facebook, Wikipedia i Yahoo [4].

U radu se koristi Ajax (asynchronous JavaScript and XML) kao grupa međusobno povezanih tehnika za razvoj web-a korišćenih na klijentskoj strani za pravljenje asinhronih web aplikacija. Korišćenjam Ajax-a, web aplikacije mogu da šalju i primaju podatke sa servera asinhrono (u pozadini) bez menjanja trenutnog prikaza i ponašanja stranice, tj. bez osvežavanja stranice. Ajax se konektuje za određeni API (*application programming interface*) i povlači (očitava) podatke iz forme koja se nalazi na html stranici koju korisnik vidi i šalje podatke ka zatraženoj .php stranici.

Koraci koji se preduzimaju u procesu realnog uključivanja LED diode su sledeći:

1. Klikom na taster kreiramo XMLHttpRequest (Ajax kreira XMLHttpRequest objekat. U pozadini se upućuje HTTP zahtev prema serveru i pronalaze se odgovarajući podaci.)

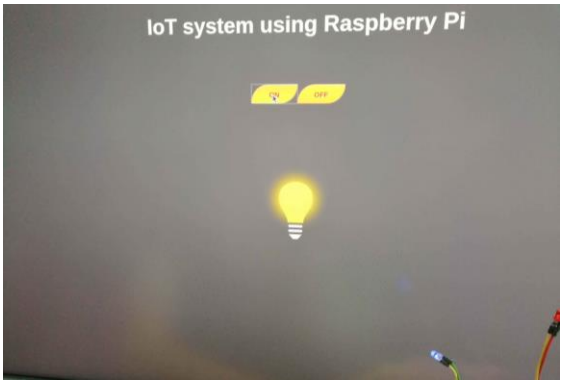
```
$('#aON').click(function(){
var a = new XMLHttpRequest();
a.open("GET","aON.php");
a.onreadystatechange=function(){
    if(a.readyState==4){
        if(a.status == 200){
       }
```

```
    }
}
a.send();
});
```

2. Pozivamo .php kod koji će izvršiti sistemsku funkciju na hardveru, postavljanjem odgovarajućeg pina u režim rada OUT, a zatim na visoki nivo ("1").

```
<?php
system("gpio -g mode 21 out");
system("gpio -g write 21 1");
?>
```

Izgled web strane sa statusom diode "uključena" i "isključena" prikazan je na slikama 4. i 6.



Slika 4. Uključivanje sijalica na web stranici.



Slika 5. Uključena LED diode.

Jedan od problema koji je trebalo rešiti predstavlja omogućavanje istovremenog pristupa web stranici sa više uređaja. Da bi se ostvarilo, bilo je potrebno kreirati još jednu .php stranicu koja će se pozivati kroz petlju kako bi se stalno ispitivalo stanje na izlazu. Ukoliko se stanje promenilo (pristigla informacija je jednaka 1) potrebno je dodati klasu „active“ kako bi sijalica na sajtu „počela da svetli“ iako korisnik nije preduzeo datu komandu.

```
<?php
$status = exec("gpio -g read 21");
echo $status;
?>
```

VI. Zaključak

Procenjuje se da ćemo do 2020. godine imati preko 50 milijardi međusobno povezanih uređaja [6]. To neće biti samo pametni telefoni već i naši tosteri, klima uređaji, automobili, vrata, kreveti - zapravo sve ono što sada smatramo običnim hardverom. Tehnologija koja stoji iza ove vizije nosi naziv *Internet of Things* i tu je da “opameti” hardver i promeni sve sfere našeg života.

U ovom radu realizovan je sistem daljinskog upravljanja mikroračunarom putem Interneta. Takav sistem omogućava korisnicima da sa bilo kog mesta manipulišu svojim uređajima u kući ili na poslu. U cilju jednostavnog praćenja rada sistema, omogućena je vizuelna prezentacija ostvarenih aktivnosti. Za realizaciju projekta korišćena je Raspberry Pi platforma sa Nginx serverom uz dodatak PHP podrške za kreiranje dinamičke webstrane.

Sistem koji je prikazan u radu pruža određeni stepen kontrole i interakcije, koje se jednostavno ne mogu dobiti od tradicionalnih uređaja. Na ovaj način mogu se rešiti problemi kao što su žurno istrčavanje iz kuće i ponovno vraćanje u istu radi provere stanja uređaja. Naravno, ovo predstavlja samo model koji je potrebno nadogratiti u cilju realne upotrebe.



Slika 6. Isključivanje sijalica na web stranici.

Zahvalnica

Autor rada se zahvaljuje profesorki Tatjani Nikolić koja je bila mentor pri izradi ovog rada, kao i profesoru Goranu Nikoliću za davanje smernica za korišćenje Raspberry Pi platforme.

Literatura

[1] Derek Molloy, "Exploring Raspberry Pi Interfacing to the Real World with Embedded Linux", Wiley, 2016.
[2] Brad Frost, "Atomic Design", ebook, 189 pages, Published 2016, http://atomicdesign.bradfrost.com
[3] The official Raspberry Pi projects book, www.raspberrypi.org
[4] Larry Ullman, "PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide", Peachpit Press, 2017, https://larryullman.com/books/
[5] Stan Schneider, "Understanding the Protocols Behind The Internet of Things", Electronic design, https://www.electronicdesign.com/iot/understanding-protocols-behind-internet-things
[6] Jimmy Daly, "What is the Internet of Things?", Statetech magazine

Driver for laser with low input impedance

Nikola Petrović, Marija Gajić, David Sanchez Illescas

Abstract - In this paper, a laser driver with a small input impedance is designed. Also, part of the system has been realized, by which we can control the voltage and current of the drivers in order to establish the appropriate laser operating conditions.

I. INTRODUCTION

Task of this project is a realization of digital driver for laser which has input impedance of 50 Ω and which can function at bit rates over 1 Mb / s. It is important to provide an output protection mechanism so it will not be damaged if the load is not connected.

The laser (Light Amplification by Stimulated Emission of Radiation or in the translation "Enhancing Light by Stimulated Radiation Emission") is a light source that emits a coherent photon beam as the source is stable in frequency, wavelength and power. In contrast to the light emitted by common sources, such as light bulbs, laser light is generally monochrome, i.e. only one wavelength (color) and is directed in a narrow beam. The beam is coherent, which means that electromagnetic waves interact at the same stage and spread in the same direction. It consists of a resonator cavity and an active environment that fills it. It was discovered in the US in 1960.

Laser air is produced by the phenomenon of stimulated emission. As the first condition for photon emission, Bohr's condition is: i.e. the laser medium must have energetic levels whose energy (energy difference) corresponds to the energy of the emitted photons. The second requirement is that most of the atoms (or molecules) are in an excited state. We must keep in mind that different processes of interaction between electromagnetic radiation and matter can occur in the laser medium: the absorption and spontaneous emission of radiation are most evident. Laser air can only be produced if the stimulated emission is more pronounced than the absorption and spontaneous emission of radiation. This is achieved by inversion of the population of atoms (or electrons) in the laser medium: the number of atoms in the excited state must be greater than the number of atoms in the ground state.

Nikola Petrović and Marija Gajić are students of third year of basic academic studies on Faculty of Electronic Engineering in Niš, A. Medvedeva 14, 18000 Niš, Serbia (e-mail: nikipet@gmail.com, marijagajic95@yahoo.com).

David Sanchez Illescas is an Erasmus exchange student at Faculty of Electronic Engineering in Niš from University of Jaén, Campus Las Lagunillas, s/n, 23071 Jaén, Spain (e-mail: dsi00001@red.ujaen.es).

Laser division is made: according to the type of material from which the source is made, according to the operating mode, according to the type of pump, according to the range of the spectrum in which the light emits. Laser drivers can be implemented in many ways, depending on the specific application needs. Some of the possible driver configurations can be of a regular nature, but most are intended for parallel laser connection. The project used a red laser (bought in a Chinese store). The radiation spectrum of a particular red laser is in the range of 620 to 750 nm. The current-voltage characteristic of the laser we use is measured in the laboratory and the results are shown in Table I and Figure 2.



Figure 1. Motherboard of disassembled laser used in this project.

TABLE I
MEASURED VALUES OF CURRENT THROUGH LASERS DIODE AND VOLTAGE ON ITS ENDS

Current[mA]	Voltage[V]	Current[mA]	Voltage[V]
0	0	5.17	2.26
0	0.55	5.42	2.3
0	1.70	6.53	2.38
0	1.77	7.66	2.47
0.59	1.84	9.06	2.58
0.85	1.87	9.15	2.59
1.32	1.92	9.96	2.65
2.41	2.03	10.27	2.67
2.77	2.06	11.64	2.78
3.84	2.16	11.76	2.79
4	2.17	12.4	2.84

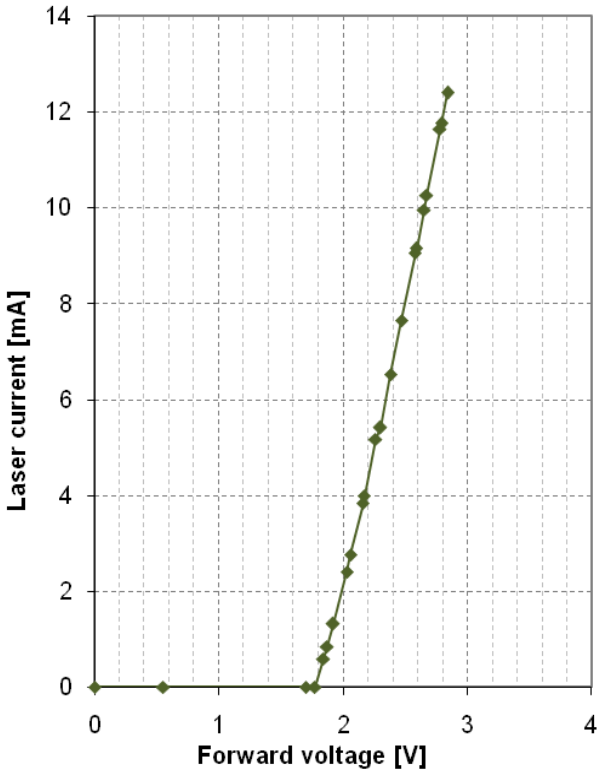


Figure 2. Recorded current-voltage characteristic of laser diode.

The voltage threshold of this laser diode is 1.9V which has current of 1mA. The value of current can be read from recorded current-voltage characteristic of laser diode. Waveform of laser can be calculated with this formula:

$$\lambda = h * c / (q * V_t) = 654\ nm \qquad (1)$$

where is h-Planck constant which has value of $6.625*10^{-34}$ m^2*kg/s, c-speed of light in vacuum which has value of $3*10^8$ m/s, q-quantity of charge which has value of $1.6*10^{-19}$ C and V_t-voltage threshold which has value of 1.9V. This value of waveform calculated with this value of voltage threshold is normal because it is in range of radiation spectrum.

II. The Working Principle

In order to accomplish the project task, it was first necessary to construct the scheme of the project shown in Figure 3. The scheme was developed in the LT Spice program and was realized using two operational amplifiers, four NPN transistors, a resistor of 7 KΩ, 36 KΩ, 5 KΩ, 100 Ω and two resistors of 50 Ω and 500 Ω, a 100 nF capacitor, two potentiometers of 10 KΩ and a red laser.

The linear driver circuit uses a 12 V power supply. The input signal is provided to the base of the transistor Q1 with resistance R1 of 50 ohms, the input resistance being very small and equal to zero. Since the input signal can range from 0 to 5 V, this is not a problem, as the output voltage range of the operational amplifier can cover the entire area of interest. The core of the drivers are the operational amplifiers U1 and U2, transistors Q4, Q3, Q2, potentiometers X1 and X2, and resistors R1, R8, R4, R5 and R6. According to the configuration we have realized, the current through the tip of R8 must be equal to the current through the collectors of the transistors Q2 and Q3. According to Ohm's law, this current is:

$$I = V_{cc} / R_8 \qquad (2)$$

Obviously, the current through the laser linearly depends directly on the input voltage Vin. The Q3 transistor is necessary to allow a relatively large output current, whereby the excitation current of the base can be adjusted by a potentiometer connected to the positive part of the input amplifier input. In addition, there is one resistor R6 that provides a part of the ignition current so that the output current of the operational amplifier is even smaller. The maximum current through the laser can be determined from here if the maximum voltage at the input is 10 V, divided by the resistance value of the resistor R8, which is 100 Ω.

$$I_{max} = V_{cc} / R_8 = 10\ mA \qquad (3)$$

We must not forget that laser diode can contain dynamic resistance. Dynamic resistance can be calculated with this formula:

$$r = dU / dI = 0.84V / 10.4mA = 80.8\Omega \qquad (4)$$

This is the total dynamic resistance in this circuit. Of this dynamic resistance the laser diode has dynamic resistance of 18.8 which is in series connection with some dynamic resistance of 62 Ω integrated in the motherboard.

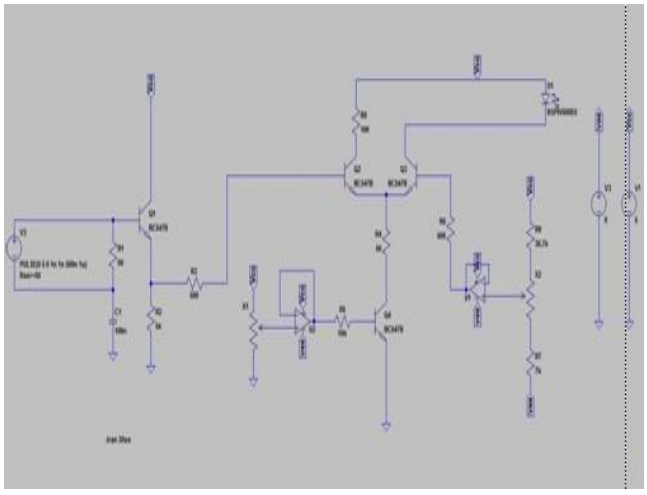


Figure 3. Configuration of laser with low input impedance.

In this configuration, the problem can appear if we do not receive(expect) the signal that we brought to the

input to be the same as on the output or similar and if the laser doesn't light up under the conditions we want though in the worst case the laser won't function. In order to avoid this problem, we first tested the resistance values corresponding to the required conditions through the simulation. Operational amplifiers is consisted of transistors Q3 and Q4 with a resistor 50 ohms, and thus the resistor is located on the emitter-coupled logic (shortened ECL) circuit transmitter. The input impedance of this laser must be 50 ohms in as wide a range as possible to avoid distortion of the impulse at the input. The value of 50 ohms was chosen because it corresponds to the impedance matching criteria for signal generators that have exactly such an output impedance. In addition to adjusting the impedance, it is necessary to ensure that the maximum and minimum laser current can be adjusted. Figure 5 shows the embodiment of a prototype laser driver with a small input impedance of 50 ohms with a mounted power supply on the transistor Q1 and a laser that lights at the output at an appropriate voltage and current value by adjusting with both potentiometers.



Figure 4. Students working on the realization of the prototype and soldering the components needed for this project.

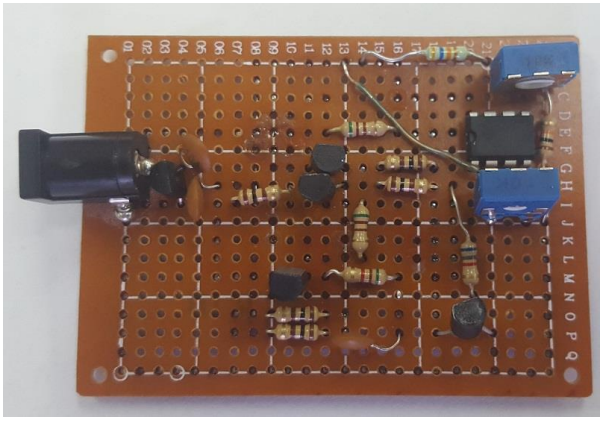


Figure 5. Prototype of laser with low input impedance.

III. TESTING PROTOTYPE

We tested the prototype with the appropriate laser pointer for which it was foreseen. For currents that are less than 2 mA, the laser does not light. In addition, the current threshold is 2 mA, as the current increases above this value, the laser starts to flash. The measured dependence of the input voltage and voltage across the diode frequency shown in Table II and Table III with graphs in Figure 6 and Figure 7 respectively.

TABLE II
MEASURED VALUES OF FREQUENCY AND INPUT VOLTAGE

Frequency[Hz]	V_{in}[V]	Frequency[Hz]	V_{in}[V]
10	4.2	80000	2.8
25	4.6	120000	2.8
50	4.4	160000	2.8
100	4.8	230000	2.6
200	4.6	300000	2.6
400	4.6	400000	2.02
800	4.6	500000	2.02
2000	4.6	600000	2
10000	4	700000	2
20000	3.4	800000	2
40000	3	1000000	2

The maximum voltage on the diode is at a minimum frequency and is about 30 mV. It decreases with increasing frequency and at a maximum frequency of 2 MHz it drops close to zero. The input voltage, unlike the voltage on the diode at high frequencies in the range of 600 kHz to 2 MHz, does not fall close to zero, but has a constant value of about 2 V.

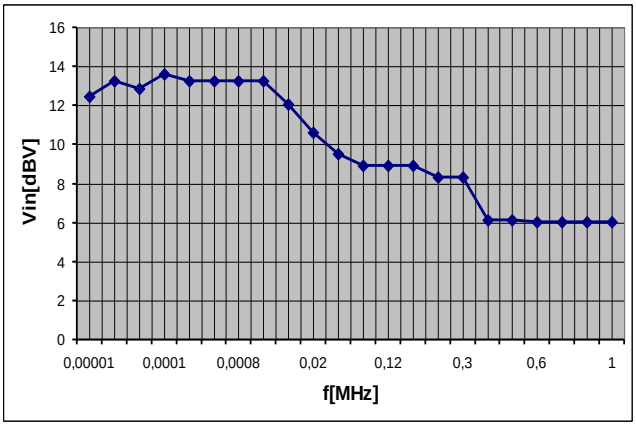


Figure 6. The dependence of the input voltage on the frequency.

In Figure 8, we fixed the laser and directed the laser beam on the photodiode, and at the 200 kHz frequency, we received the best laser illumination without the potentiometer setting.

TABLE III
MEASURED VALUES OF FREQUENCY AND DIODE VOLTAGE

Frequency[Hz]	V_d[mV]	Frequency[Hz]	V_d[mV]
10	29.6	80000	20
25	28.2	120000	18.4
50	27.4	160000	16.8
100	26.8	230000	16.6
200	25.4	300000	12
400	25.2	400000	9.6
800	25.2	500000	7.4
2000	25.2	600000	5.6
10000	25	700000	4.4
20000	24.8	800000	3.36
40000	23.4	1000000	3.04

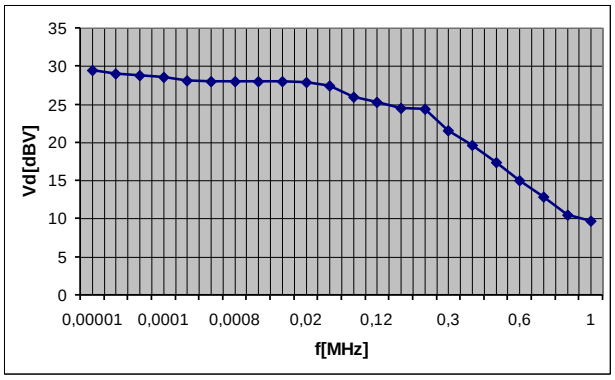


Figure 7. The dependence of the laser diode voltage on the frequency.

IV. CONCLUSION

The task of the project was to realize a linear driver with small input impedance for a laser pointer. An analysis of the task was performed and the design of the prototype was performed, the validity of which was tested using the simulator. The prototype was then practically realized using the proto plate and electronic components and its operation was confirmed by concrete measurements with the help of an oscilloscope and a multimeter. Even though we made some modifications in regards to the original schematic, we are satisfied that laser works finely in regards to what we expected.

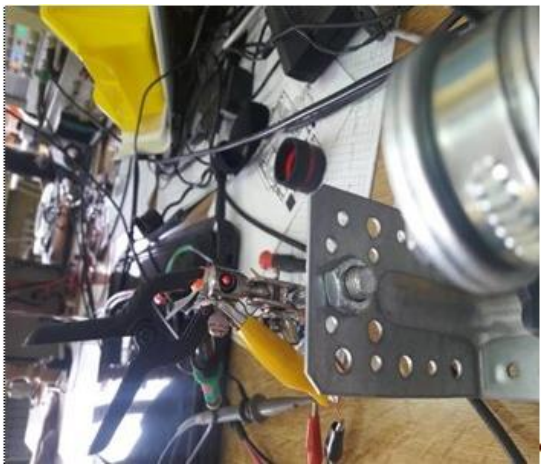


Figure 8. The laser fixing procedure and laser beam positioning on the photodiode.

ACKNOWLEDGEMENT

This paper is written as part of the student project at the Optical Communication Exam, in the direction of Telecommunications, Electronic Faculty of the University of Niš. The exam is on the 6th semester of basic academic studies. The hardware realization of the project was carried out at the laboratory for telecommunication systems, under the mentorship of prof. Dr. Dejan Milić.

REFERENCES

[1] Wikipedia, articles: „Laser and laser diodes“, available online on: https://en.wikipedia.org/wiki/Laser
https://en.wikipedia.org/wiki/Laser_diode
[2] Transistor,datasheet
https://www.mgelectronic.rs/ProductFilesDownload?Id=2593
[3] Opamp,datasheet
https://www.mgelectronic.rs/ProductFilesDownload?Id=6460
[4] Potentiometer,datasheet
https://www.mgelectronic.rs/ProductFilesDownload?Id=2582
[5] “John Senior, Optical Fiber Communications Principles and Practice, Third Edition, available online on: https://eceagmr.files.wordpress.com/2014/09/optical-fiber-communications-principles-and-pr.pdf
[6] Dr Dejan Milić, Optical Communications, Faculty of Electronic Engineering, Niš

Ispitivanje pouzdanosti *Littelfuse* gasnog odvodnika prenapona

Marija Živković, Sandra Veljković, Emilija Živanović

Sadržaj – Vreme odziva gasnih odvodnika prenapona proizvođača *Littelfuse* ispitivano je primenom metode vremena kašnjenja. To je kvalitativna metoda, kojom se može pratiti opadanje koncentracije pozitivnih jona i neutralnih aktivnih čestica tokom i nakon pražnjenja do vrlo niskih koncentracija, kao i proceniti njihova vremena rekombinacije i deekscitacije. Eksperimentalni podaci srednje vrednosti probojnog napona ovih komponenata dobijeni za brzine porasta napona od 1 do 10 V/s i analizirani su dinamičkom diskretizovanom metodom prikazani su takođe u ovom radu. Pored toga, uticaj prenapona na zavisnost vremena odziva u funkciji vremena relaksacije je takođe istraživan.

I. UVOD

Gasni odvodnici prenapona su najčešće korišćeni elementi za zaštitu od prenapona. Pod prenaponom se podrazumeva pojava da je potencijal jedne tačke neke komponente ili uređaja u odnosu na neku drugu tačku ili tačku nultog potencijala veći od dozvoljenog. Do pojave prenapona mogu dovesti atmosferska pražnjenja, elektrostatička pražnjenja, komutacioni prenaponi, radarski impulsi i elektromagnetni impulsi nuklearne eksplozije. Ove vrste pražnjenja značajno utiču i na telekomunikacione vodove preko kojih oštećuju komponente, odnosno uređaje. Od svih vrsta prenapona, atmosfersko pražnjenje je najopasnije jer je njegov nastanak nepredvidiv.

Elementi za zaštitu od prenapona se prema načinu primene napona na njihovim krajevima, pri povećanju struje kroz njih mogu podeliti na linearne i nelinearne. Linearni elementi za zaštitu od prenapona su električni filtri, čiji su najosetljiviji elementi kondenzatori. Nelinearni elementi za zaštitu od prenapona se mnogo češće koriste od linearnih i oni se prema tehnologiji izrade i principu rada mogu podeliti u tri grupe. To su prenaponske diode (*Transient suppresser diodes*, skraceno TSD), metaloksidni varistori (*Metaloxide varistors*, skraćeno MOV) i gasni odvodnici prenapona (*Gas filled surge arresters*, skraćeno GFSA). U cilju zaštite od prenapona nekada se koriste različite kombinovane (hibridne) šeme [1], [2].

Rad gasnih odvodnika prenapona je zasnovan na procesima koji dovode do električnog proboja i pražnjenja. Karakteristike svih nelinearnih elemenata za zaštitu od prenapona (prenaponske diode, mataloksidni varistori, gasni odvodnici prenapona) mogu se prikazati jednačinom $i = a \cdot U^b$, gde su a i b konstante.

Pokazano je da je element za zaštitu od prenapona bolji ukoliko je strujno-naponska karakteristika bliža pravoj liniji koja je paralelna naponskoj osi.

Gasni odvodnici, koji se nalaze u komercijalnoj upotrebi se najčešće izrađuju kao dvopinski (90 V-800 V) i tropinski (90 V-350 V), kao što je prikazano na slici 1.



(a) (b)

Slika 1. Gasni odvodnici (a) dvopinski i (b) tropinski.

Najvažnija karakteristika elemenata za zaštitu od prenapona je njihova brzina reagovanja koja zavisi od nekoliko parametara, i to:

- nagiba prednje ivice prenaponskog impulsa koji predstavlja brzinu porasta napona;
- statičkog nivoa reagovanja koji predstavlja prag reagovanja na spore promene impulsa;
- dinamičkog nivoa reagovanja koji predstavlja najveći napon koji se postiže na zaštitnom elementu kada je nagib ivice veliki,
- i vreme reagovanja koje predstavlja vremenski interval za koji prenaponski impuls promeni vrednost od statičkog do dinamičkog nivoa reagovanja bez dejstva zaštite pri određenoj brzini porasta napona.

Rad gasnih odvodnika prenapona se zasniva na principu proticanja električne struje kroz gas, tj. na električnom pražnjenju u gasovima. Spadaju u nelinearne elemente koji se isključivo koriste za zaštitu od prenapona.

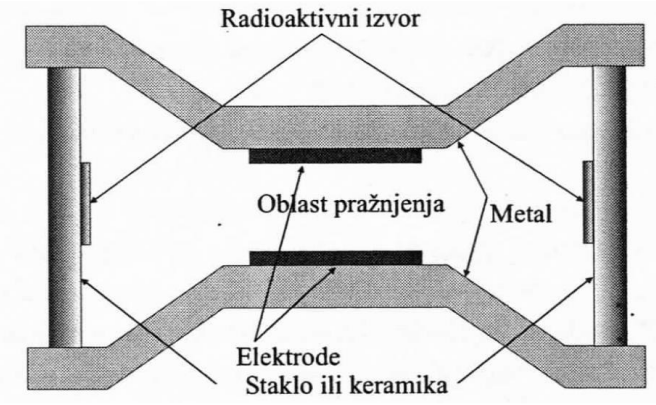
II. PRINCIP RADA GASNOG ODVODNIKA

Korišćeni uzorak u našim eksperimentima bio je gasni odvodnik proizvođača *Littelfuse*. Uopšeno gledano, gasni odvodnici se sastoje od dve ili tri elektrode koje su zatopljene u keramičko ili stakleno kućište (slika 2). Elektrode, koje su ravne i paralelne kako bi se u međuelektrodnom prostoru obezbedilo približno homogeno električno polje, se nalaze u kućištu koje se sastoji od keramike i metala. Rastojanje između elektroda

M. Živković i S. Veljković su studenti master studija modula Elektronika i mikrosistemi, E. Živanović je docent na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marijaz.elfak@gmail.com, sandra.veljkovic@elfak.rs, emilija.zivanovic@elfak.ni.ac.rs

je reda milimetra ili dela milimetra. Kao izolacioni materijal koriste se ili plemeniti gasovi (argon, neon, kripton ili ksenon) ili njihove smeše na pritiscima od 0.1 kPa do 70 kPa. Nekada su u upotrebi bile dve vrste gasnih odvodnika prenapona i to oni bez ugrađenog radioaktivnog izvora i sa ugrađenim radioaktivnim izvorom. Radioaktivni izvor obezbeđuje povećani broj jonskih parova u međuelektrodnom prostoru što dovodi do bržeg reagovanja odvodnika na pojavu prenapona. Danas se svi gasni odvodnici izrađuju bez radioaktivnog izvora. Gasni odvodnici prenapona sa tri elektrode predstavljaju dva gasna odvodnika prenapona u istom kućištu.

Tehničke karakteristike korišćenog *Littelfuse* gasnog odvodnika prenapona su: radni napon 120 V AC, probojni napon 230-340 V DC, izolatorska otpornost 10 GΩ na 100 V i kapacitivnost 1.5 pF. Detaljniji pregled tehničkih karakteristika ovog gasnog odvodnika dat je u Tabeli I [3].



Slika 2. Poprečni presek gasnog odvodnika prenapona.

TABELA I
SPECIFIKACIJA KORIŠĆENOG *LITTELFUSE* GASNOG ODVODNIKA PRENAPONA NA 25°C

Komponenta	Radni napon AC (V)	DC probojni napon (V)	DC proboj-tipično (V)	Izolatorska otpornost	Kapacitivnost (pF)	Napon luka (V)	Nominalna impulsna struja pražnjenja (x10@8/20μs) (kA)
LF	120	230 do 340	285	10 GΩ na 100 V	1.5	~ 25	5

III. PARAMETRI KOJI DEFINIŠU ELEKTRIČNI PROBOJ U GASU

Gasni odvodnik korišćen prilikom opisanih istraživanja je komercijalna komponenta, i iz tog razloga nije poznato koji se gas nalalzi unutar nje same. Kako su istraživanja, čiji su rezultati prikazani u ovom radu, bazirana na električnom proboju u gasu, biće ukratko dat opis osnovnih veličina koje ga karakterišu, a to su:

- t_d – vreme kašnjenja električnog proboja
- τ – vreme relaksacije
- t_g – vreme žarenja
- σ – standardna devijacija
- U_s – statički probojni napon.

Odmah nakon priključivanja napona na gasnu komponentu u gasovima ne dolazi do električnog proboja, čak i kada je on znatno veći od statičkog probojnog napona, već posle izvesnog vremena koje se naziva vreme kašnjenja električnog proboja t_d. Najčešće se pod vremenom kašnjenja električnog proboja podrazumeva vreme koje protekne od momenta priključenja radnog napona U_w na elektrode gasne komponente, koji je veći od statičkog probojnog napona U_s, do nastanka proboja, tj. dostizanja zadate vrednosti struje pražnjenja.

Vreme kašnjenja je veličina stohastičkog karaktera zbog fluktuacija njegovih vrednosti i ima određenu statističku raspodelu čiji oblik zavisi od vrednosti vremena relaksacije gasa. U cilju dobijanja što pouzdanijeg oblika memorijske krive, koja predstavlja zavisnost srednje vrednosti vremena kašnjenja električnog proboja od vremena relaksacije, potrebno je izvršiti veliki broj merenja vremena kašnjenja pri istim uslovima i uzeti srednju vrednost dobijenih podataka [4].

Period za vreme koga se napon na elektrodama gasne komponente isključuje naziva se vreme relaksacije τ, i ono se takođe unapred zada tokom eksperimenta. Kada istekne zadato vreme relaksacije proces počinje od početka.

Vreme žarenja (pražnjenja) t_g predstavlja vreme za koje posle uspostavljanja samostalnog pražnjenja kroz gas teče struja pražnjenja i_g.

Standardna devijacija σ je apsolutna mera disperzije u osnovnom skupu merenih veličina i govori nam koliko u proseku izmerene vrednosti odstupaju od srednje vrednosti vremena kašnjenja električnog proboja.

Statički probojni napon definisan jednačinom (1) predstavlja najveći radni napon U_w za koji je verovatnoća električnog proboja jednaka nuli, tj.

$$\lim_{P\to 0} U_w = U_s \,. \qquad (1)$$

Pojava memorijske krive predstavlja neželjeni efekat kod gasnih odvodnika, i taj efekat je kod ispitivane komponente dobijen. Merenja su izvršena automatski upravljanim eksperimentima posebno projektovanim za merenje probojnog napona i vremena kašnjenja električnog

proboja i softverski se vrši analiza dobijenih rezultata za svaku unapred zadatu vrednost vremena relaksacije i vremena žarenja [5], [6].

IV. PRIKAZ I ANALIZA DOBIJENIH REZULTATA

Na slici 3 je prikazan izgled okruženja koje je korišćeno za ovaj eksperiment.

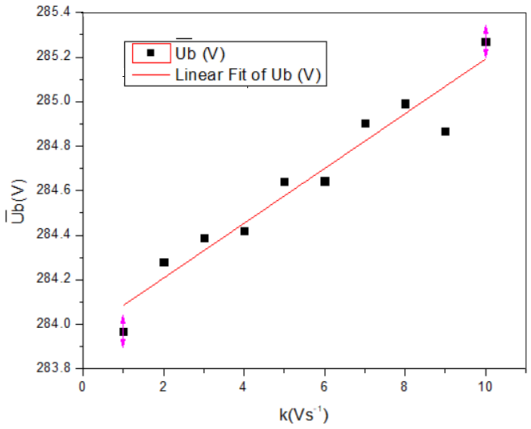


Slika 3. Merenje probojnog napona i vremena kašnjanja komercijalnog gasnog odvodnika.

Kolo koje se koristi za merenje vremena kašnjenja električnog proboja je priključeno na izvor visokog napona koji se primenjuje na gasni odvodnik. Primenom radnog napona na komponentu, naponski impuls stvara „start signal“. Od tog momenta kreće merenje vremena kašnjenja. Nastankom proboja, struja kroz cev na potenciometru stvara napon kojim se definiše kraj merenja vremena kašnjenja. Tada se računarom kontroliše vreme trajanja struje kroz gasni odvodnik i nakon, uslovima eksperimenta zadatog vremena prekida, dolazi do relaksacije gasa u njemu. Ciklus merenja se ponavlja unapred zadati broj puta.

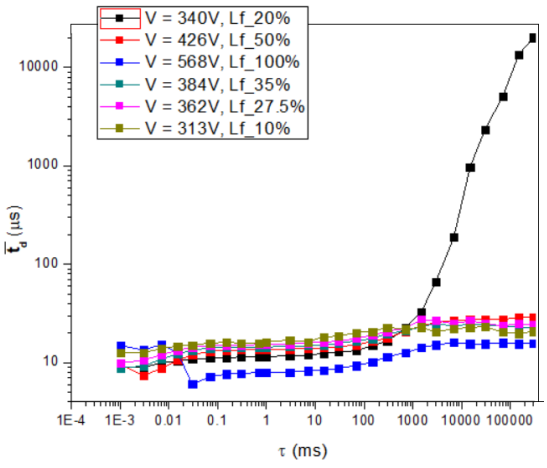
Merenje probojnog napona vršeno je uređajem koji sadrži programibilni izvor napona *Keithley* 2400, a akvizicija podataka vršena je računarom, tj. programom za podržavanje ovih merenja. Uređaj ima mogućnost da za zadato vreme povećava napon čija se vrednost programski zadaje. Brzine porasta napona u ovom eksperimentu menjane su od 1 do 10 Vs^{-1}.

Eksperimentalno dobijeni rezultati merenja su predstavljeni na slici 4, gde je simbolima na grafiku predstavljena zavisnost srednje vrednosti probojnog napona dobijene na osnovu 100 izmerenih vrednosti, od brzine porasta napona *k*. Vidi se da eksperimentalni rezultati, sprovedeni na komercijalnim gasnim odvodnicima dobro odražavaju linearnu zavisnost od *k*. Presekom fitovane krive sa $\overline{U}_b$ - osom (za $k = 0$) nalazi se vrednost statičkog probojnog napona U_s koja u ovom slučaju iznosi približno 248V.



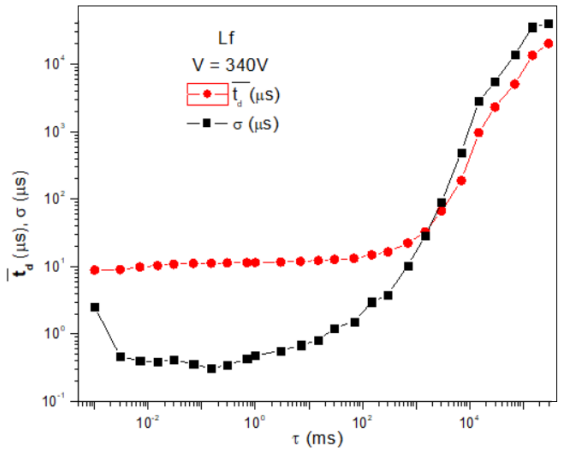
Slika 4. Srednja vrednost dinamičkog probojnog napona u funkciji brzine porasta napona *k* od 1 do 10 Vs^{-1}.

Na drugom grafiku (slika 5) predstavljena je zavisnost srednjeg vremena kašnjenja $\bar{t}_d$ izraženog u μs u zavisnosti od vremena relaksacije τ u ms za različite vrednosti prenapona koja je dobijena za komercijalni gasni odvodnik proizvođača *Littelfuse*. Vremena relaksacije su se u eksperimentu kretala od 1 μs pa do 300 s, što se može i videti na grafiku. Izvršeno je po 100 merenja vremena kašnjenja električnog proboja za svaku vrednost vremena relaksacije. Istraživanja su zasnovana na ispitivanju uticaja promene napona na ovu zavisnost, poput onih publikovanih u radu [7]. Iz dobijenih rezultata, primećuje se da srednje vrednosti vremena kašnjenja veoma malo rastu sa porastom vremena relaksacije i dobijena je vrednost reda veličine 10 μs, za sve vrednosti prenapona, osim za 340 V. Vidi se da u slučaju primenjenog prenapona od 340 V, pri vrednostima τ reda veličine od 1000 ms, dolazi do naglog porasta srednjeg vremena kašnjenja. Jasno je da, kao što se i vidi sa slike 5, jedino za tu vrednost prenapona dobijena memorijska kriva [5], [7].



Slika 5. Zavisnost srednjeg vremena kašnjenja od vremena relaksacije za različite vrednosti prenapona kod komercijalnog gasnog odvodnika.

Treći grafik prikazan na slici 6 pokazuje srednju vrednost vremena kašnjenja i standardnu devijaciju u funkciji vremena relaksacije za vrednost primenjenog napona od 340 V, s obzirom da se jedino za ovu vrednost prenapona javlja memorijska kriva.



Slika 6. Srednja vrednost vremena kašnjenja i standardna devijacija vremena kašnjenja za vrednost prenapona od 340V.

V. ZAKLJUČAK

Statički probojni napon, kao deterministička veličina, može se precizno proceniti pomoću diskretizovane metode, tj. fitovanjem zavisnosti srednje vrednosti probojnog napona od brzine porasta napona. Procenjena vrednost statičkog probojnog napona za korišćeni gasni odvodnik na ovaj način je u dobroj saglasnosti sa nominalnom vrednošću kao što se može videti iz Tabele I. Napon tolerancije za *Littelfuse* gasni odvodnik prenapona koji je korišćen u ovom radu je oko 1%.

Eksperimentalni podaci vremena kašnjenja električnog proboja su korišćeni za analizu vremenskog odziva gasnih odvodnika. Analiza čiji su rezultati prikazani na slici 5 pokazuje sledeće. Kao što se može videti, za vrednosti prenapona većih od 20%, ova vrsta gasnog odvodnika pouzdano radi. Međutim, za prenapone manje od 20% takođe radi pouzdano, ali za vremena relaksacije do 1 s. Pored toga, rezultati pokazuju da je odziv ove komponente oko 10 μs za sve vrednosti napona koje su primenjene na uzorak.

Dalji nastavak istraživanja ogleda se u mogućnosti upoređivanja dobijenih rezultata sa analizama uzoraka gasnih odvodnika prenapona proizvedenim od strane drugih proizvođača (*EPCOS*, *CITEL*,…). Pored toga, postoji mogućnost istraživanja mehanizama koji dovode do povećanja kašnjenja električnog proboja nakon određene vrednosti prenapona, utičući na pouzdanost samih komponenti.

ZAHVALNOST

Zahvaljujemo se ovom prilikom doc. dr Miliću Pejoviću koji nam je pomogao oko realizacije eksperimenata.

LITERATURA

[1] Momčilo M. Pejović, *Uvod u električna gasna pražnjenja. Gasne elektronske komponente*, Elektronski fakultet, Univerzite u Nišu, Niš, 2008.

[2] Momčilo M. Pejović, Milić M. Pejović, *Električni proboj gasova*, Elektronski fakultet, Univerzitet u Nišu, Niš, 2009.

[3] https://www.littelfuse.com/~/media/electronics/product_catalogs/littelfuse_gdt_catalog.pdf.pdf

[4] J. M. Meek and J. D. Craggs, *Electrical breakdown of gases*, New York, USA: Wiley, 1987.

[5] M. M. Pejović, G. S. Ristić and J. P. Karamarković, "Electrical breakdown in low pressure gases", *J. Phys. D: Appl. Phys.*, 2002, vol. 35, pp. R91-R103.

[6] M. M. Pejović, "Digital system for vacuum and gas-filled devices testing", *Rev. Sci. Instrum*., 2005, vol. 76, p. 015102.

[7] Emilija Živanović, Sandra Veljković, Marija Živković and Milić Pejović, "Reliability of Various Type of Gas-filled Surge Arresters Under DC Discharge", *Proc. 31th International Conference on Microelectornics*, Niš Serbia, 2019, pp. 113-116.

Realizacija brojača glasova pomoću Arduino Uno okruženja

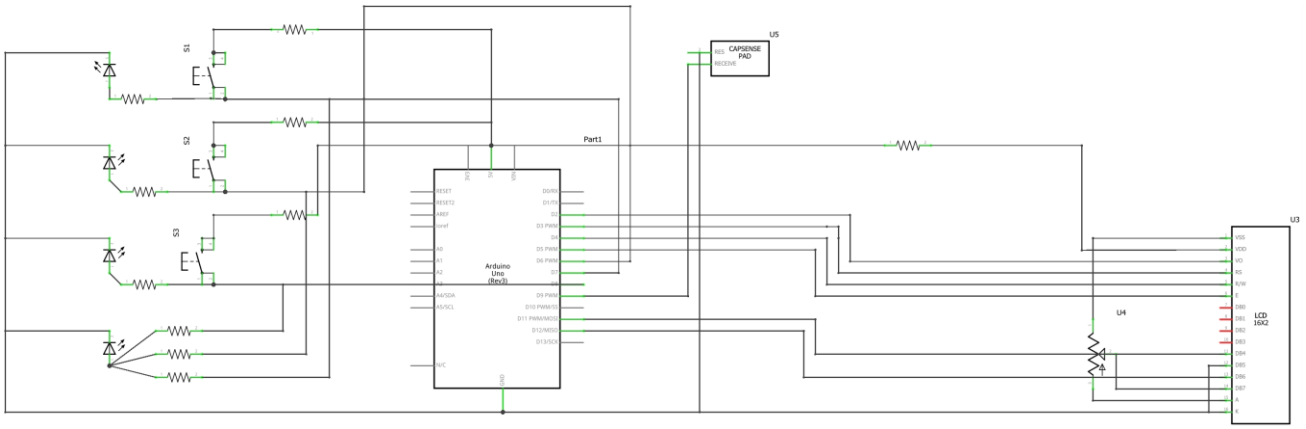
Ana Sentov

Sadržaj – U ovom radu prikazan je način realizacije brojača glasova, koji se može koristiti u situacijama gde su glasači za, protiv ili uzdržani za osobu ili predlog za koji se glasa. Ovo je dobar način kako ne bi došlo do greške konačnog brojnog stanja i kako za to ne bi krivili ljudski faktor.

I. UVOD

Kroz istoriju se nailazi na dosta zanimljivih načina glasanja. Počevši od Antičke Grčke i korišćenja delova polomljenih keramičkih sudova, preko Rima i korišćenja papirnih glasačkih listića, dok su se u Antičkoj Indiji koristili listovi palminog drveta za izbore u selu.

Svi ovi načini glasnja imaju dve mane: brojanje glasova predugo traje i velika mogućnost greške pri brojanju. Iz tih razloga, za realizaciju ovog rešenja je korišćeno okruženje *Arduino Uno* [1], LCD (*Liquid Crystal Display*) [2], senzor kapacitivnosti (*touch* senzor) [3], tri tastera, četiri LE diode, potenciometar, devet otpornika raznih vrednosti, provodnici (Slika 1).



Slika 1. Šematski prikaz brojača kola. [4]

Za povezivanje komponenata se mogu koristiti analogni i digitalni GPIO (*General Purpouse Input/Output*) pinovi Arduino okruženja, međutim u ovom radu su korišćeni samo digitalni pinovi. Preko šestog, sedmog i osmog pina na Arduinu su povezani tasteri preko kojih se omogućava glasanje. Kapacitivni senzor sadrži tri pina, pin za napajanje (V_{CC}), pin za uzemljenje (GND) i treći je izlazni pin (I_O) koji je povezan na deveti pin Arduina. LCD sadrži šesnaest pina za povezivanje ali samo dvanaest pina je iskorišćeno za povezivanje na Arduino pinove, preciznije koji je pin sa kojim povezan je prikazano u tabeli I.

TABELA I
MEĐUSOBNA POVEZANOST PINOVA NA LCD-U I ARDUINU

Ime pina na LCD-u	R_S	E	D_4	D_5	D_6	D_7
GPIO pin na Arudinu	12	11	5	4	3	2

Ana Sentov, student treće godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ana.sentov@elfak.rs

U tabeli I je naznačeno koji su pinovi međusobno povezani na LCD-u i na Aduinu.

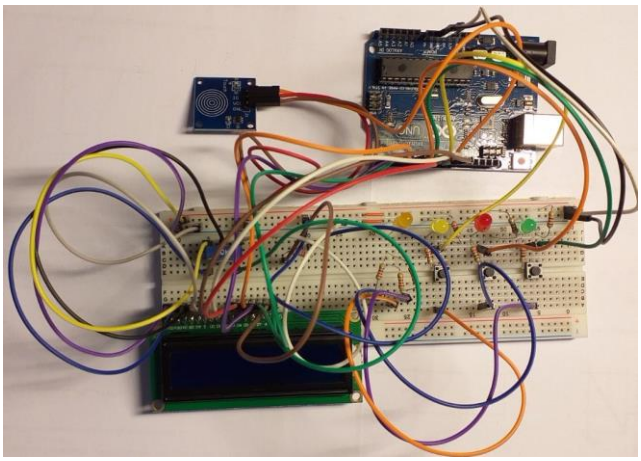
V_{SS} i R_W pin su povezani direktno na uzemljenje. Pinovi koji nisu napomenuti u tabeli I, nisu direktno povezani na Arduino pinove. Dva od tri pina potenciometra su povezana na V_{DD} i K pin LCD-a, dok su drugi krajevi tih čvorova vezani na napajanje i na uzemljenje, a V_0 pin je povezan na treći pin potenciomentra kojim se reguliše osvetljenje LCD-a. Pin A je povezan preko otpornika vrednosti 220Ω.

II. PRINCIP FUNKCIONISANJA

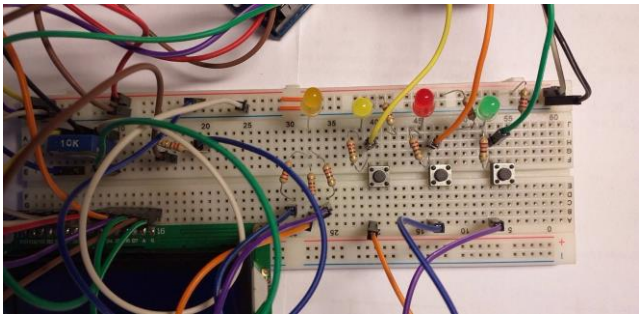
Kako je i sam proces glasanja jednostavan, tako i sam projekat ne sadrži nikakve kompleksne funkcije. Pomoću tastera je omogućeno registrovanje glasa kao i da se to zapamti u memoriju mikrokontrolera.

Kada se taster pritisne, LE dioda se uključi, i u zavisnosti od toga koja dioda se uključi prepoznaje se koji glas je bačen (zelena „za“, crvena „protiv“, žuta „uzdržan“), dok narandžasta LE dioda signalizira da se

glasanje dogodilo. Tokom pritiska tastera, na LCD-u se prikazuje „*loading bar*“ koji govori glasaču još koliko dugo taster treba biti pritisnut da bi glas bio važeći. Kako trenutno stanje glasova ne bi uticalo na buduće, nemaju svi pristup sumi glasova. Kao model senzora za prepoznavanje otiska prsta, korišćen je kapacitivni senzor. Ukoliko senzor detektuje neki signal duže od 5s, odnosno na senzoru za prepoznavanje otiska prsta bude odgovarajući otisak, na LCD-u će se prikazati trenutni rezulati glasanja, a ukoliko se uradi dupli klik na senzoru, rezultati će nestati sa LCD-a.



Slika 2. Celokupno povezano kolo.



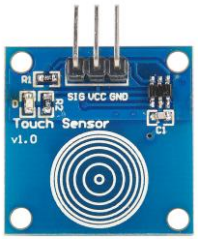
Slika 3. Deo kola sa tasterima, LE diodama i otpornicima.



Slika 4. Prikaz rezultata na LCD-u pre glasanja.

A. Senzor kapacitivnosti

Kapacitivni senzor funkcioniše na isti način kao i kondenzator. Sam senzor je provodna površina i počeće da se ponaša kao kondenzator u prisustvu bilo koje druge provodljive površine, na primer kože. Kada čovek dodirne senzor, imitira se kondenzator neke kapacitivnosti, što se očitava kao logička jedinica, dok se u suprotnom prikazuje kao logička nula. Kako je prethodno rečeno u radu, ako senzor očitava logičku jedinicu duže od 5s, prikazuju se rezultati glasanja na LCD-u, a ukoliko se izvrši dupli klik, nestanu rezultati sa LCD-a, a sve je to prethodno definisano softverskim putem preko *Arduino* kôda (Slika 6).



Slika 5. Kapacitivni senzor.

```
if (digitalRead(9) == HIGH && promena == false){
    pocetak = millis();
    promena = true;
}
if (digitalRead(9) == LOW && promena == true){
    kraj = millis();
    promena = false;
    trajanje = kraj - pocetak;
      lcd.setCursor(0, 0);
      lcd.print("Za:");
      lcd.print(ZA);
      lcd.setCursor(6, 0);
      lcd.print("Pr:");
      lcd.print(PR);
      lcd.setCursor(12,0);
      lcd.print("Uz:");
      lcd.print(UZ);
    delay(50);
}
if (trajanje < 300){
    lcd.clear();
}
```

Slika. 6. Isečak kôda prikazivanja rezultata na LCD-u [5].

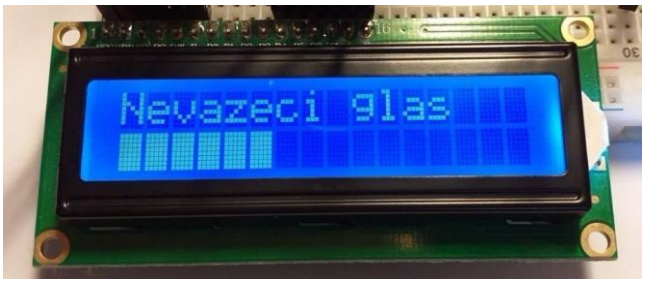
B. Loading Bar

Kako bi glas bio važeći, potrebno je da prilikom glasanja taster bude pritisnut određeni vremenski period. Taj vremenski period je korisniku ograničen „*Loading bar-om*“ koji se prikazuje na LCD-u. Pomoću njega omogućava korisniku da otprilike proceni koliko još treba držati taster pritisnutim. Da bi glas bio važeći potrebno je taster držati do kraja učitavanja *loading bara* (Slika 7).



Slika 7. Izgled LCD-a pri glasu koji je važeći.

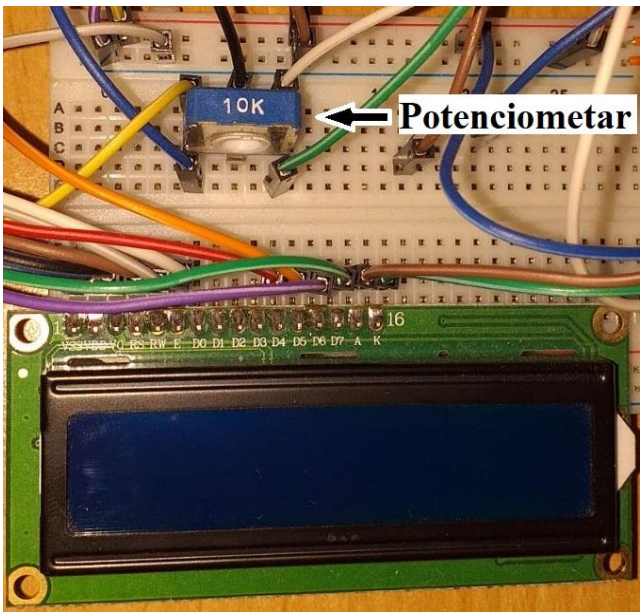
Na slici 8. je prikazano ukoliko dođe do prekida držanja tastera, tada učitavanje „*Loading Bar-a*“ prestaje i na ekranu se ispisuje „Nevazeci glas”.



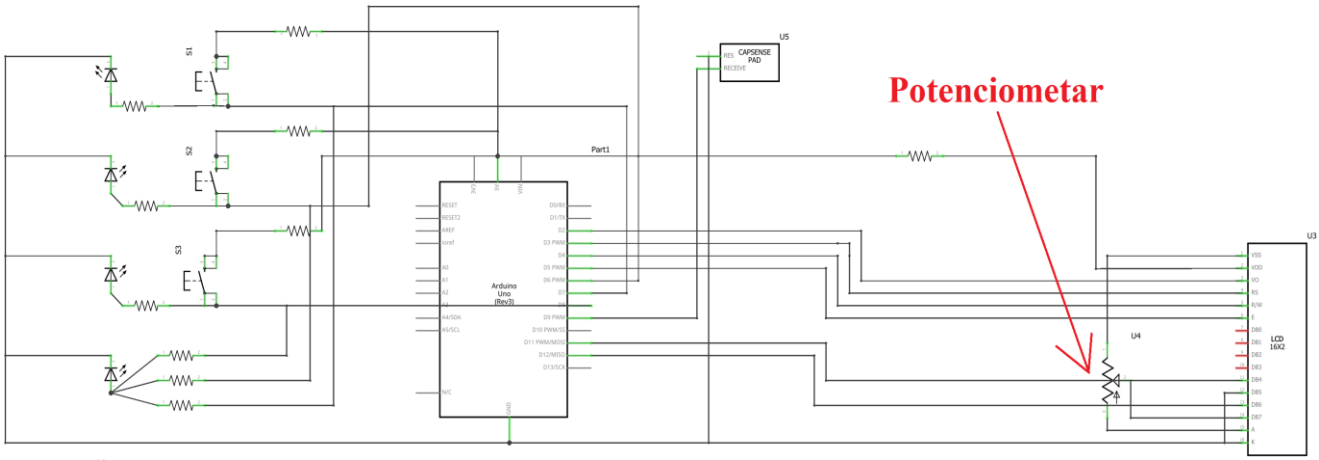
Slika 8. Izgled LCD-a pri glasu koji je nevažeći.

C. LCD displej

Korišćen LCD poznatiji kao „16x2 displej“ ispisuje karaktere u dva reda, a svaki red može prikazati 16 karaktera. Takođe, osvetljenje displeja tj. kontrast između karaktera i pozadine se može reguliasti potenciometrom vrednosti 10 kΩ povezanim na pin V_0 LCD-a (Slika 10).



Slika 9. Potenciometar korišćen za kontrolu osvetljenja LCD-a.



Slika 10. Šematski prikaz potenciometra [4].

Na slici 11. je prikazno uključivanje biblioteke „*LiquidCrystal lcd“* na početku kôda kako bi samo funkcionisanje displeja bilo omogućeno.

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Slika 11. Isečak kôda biblioteke „*LiquidCrystal lcd*“ [5].

Kombinovanjem *touch* senzora i funkcije za štampanje „*lcd.print(" ")*“, na displeju se prikazuju rezultati glasanja (Slika 12), gde „Za“ označava da je glasač saglasan, „Pr“ označava da je proziv i „Uz“ da je uzdržan. Korišćena su početna dva prva slova reči za, protiv i uzdržan jer nas broj karaktera na displeju ograničava na 16.



Slika 12. Displej nakon završenog glasanja.

III. ZAKLJUČAK

Prilikom izrade ovog rada, predviđeni su skoro svi problemi koji mogu nastati prilikom glasanja.

Postoji mogućnost optimizacije kôda, korišćenje kvalitetnijih komponenata samim tim i dodavanje raznih funkcionalnosti, što na kraju dovodi do boljih rešenja.

ZAHVALNOST

Svoju neizmernu zahvalnost dugujem doc. dr Emiliji Živanović koja me je motivisala da napišem rad za ovu konferenciju i prof. dr Danijelu Dankoviću na početnoj ideji. Zahvaljujem se i svom kolegi Nikoli Radisavljeviću koji mi je pomogao oko same praktične realizacije ovog projekta.

LITERATURA

[1] Tehnička dokumentacija okruženja Arduino Uno. Dostupno na: https://components101.com/microcontrollers/arduino-uno
[2] Tehnička dokumentacija *Liquid Crystal Display*. Dostupno na: https://www.makerguides.com/wp-content/uploads/2019/02/16x2-Character-LCD-Datasheet.pdf
[3] Tehnička dokumentacija Touch senzora. Dostupno na: https://www.uploadarchief.net/files/download/ttp223_module.pdf
[4] *Fritzing, Arduino Full Reference*. Dostupno na: https://fritzing.org/learning/full_reference
[5] *Arduino IDE, Arduino Full Reference*. Dostupno na: https://www.arduino.cc/reference/en/

Ispitivanje dinamičkih karakteristika optokaplera

Ana Sentov, Nina Dimitrijević

Sadržaj – U ovom radu dat je teorijski pregled principa rada optokaplera, kao simulacija njegovih dinamičkih karakteristika. Tabelarno i grafički su prikazane promene vrednosti vremena uključivanja i isključivanja optokapera u kolu, pri različitoj otpornosti i vrednosti parametra efikasnosti prenosa.

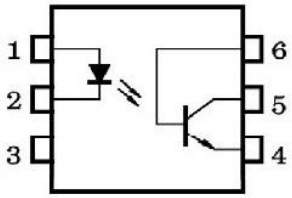
I. OPTOKAPLER

Optokapleri predstavljaju kombinaciju svetlosnog izvora (tzv. optički predajnik – LE ili GaAs dioda) i optičkog prijemnika (fototranzistor ili fotodioda) u jednom kucištu (slika 2). Optokapleri koriste svetlosni snop za prenos podataka (u analognom ili digitalnom obliku) kroz električnu barijeru, pa je time omogućena i dobra izolovanost [1].



Slika 1. Kućište optokaplera.

Optokapleri su obično zapakovani u DIP6 kućista kao što je prikazano na slici 1 (u ovom kućištu se koriste samo 5 izvoda, šesti izvod se ne koristi, nije fizički povezan i označen je kao NC od izraza *Not Connected*) [2].



1. Anoda
2. Katoda
3. Pin koji nije povezan
4. Emitor fototranzistora
5. Kolektor fototranzistora
6. Baza fototranzistora

Slika 2. Pin konfiguracija optokaplera 4N25.

Ana Sentov, student treće godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ana.sentov@elfak.rs

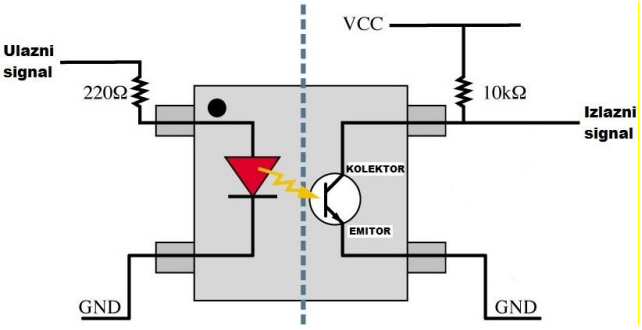
Nina Dimitrijević, student četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nina.dimitrijevic@elfak.rs

Postoje različiti tipovi optokaplera kao što su: 4N25, 4N26, 4N27, 4N28. Najčešće korišćeni tip optokaplera koji je takođe korišćen i u ovom radu je 4N25 [3].

II. PRINCIP RADA OPTOKAPLERA

A. Princip rada optokaplera

Optokapler je komponenta koja se sastoji od LE diode i fototranzistora integrisanih u jednom kućištu (Slika 3). Ova dva dela optokaplera su razdvojena transparentnom barijerom, koja blokira protok struje između njih, a dozvoljava prolazak svetlosti. Optokapler predstavlja komponentu sa svetlosnom spregom između ulaza i izlaza. Signal sa ulaza izaziva emisiju svetlosti LE diode. Ova svetlost predstavlja pobudni signal fototranzistora, tako da se na izlazu pojavljuje odgovarajući signal. Ulaz i izlaz su međusobno galvanski izolovani. Galvanska izolacija čini optokapler pogodnim za primenu u svim uređajima kod kojih korisnik može doći u dodir sa potencijalno opasnim nivoima signala (merni instrumenti, medicinski uređaji, telekomunikacioni uređaji, itd.). Optokapler se takođe može koristiti za prilagođenje naponskih nivoa između dva kola, kao i za smanjenje uticaja elektronskog šuma [1].



Slika 3. Šematski prikaz unutrašnjosti optokaplera.

B. Bitni parametri optokaplera

Karakterističan parametar optokaplera je efikasnost sprege η (*coupling efficiency*) [1]. Definiše se kao odnos struje kolektora fototranzistora I_C i struje fotodiode pri direktnoj polarizaciji I_F:

$$\eta = \frac{I_C}{I_F} \cdot 100\% . \quad (1)$$

Ovaj parametar se još naziva prenosnim odnosom struja ili efikasnost prenosa (CTR - *Current Transfer*

Ratio). Vrednost efikasnosti sprege zavisi i od temperature. Za optokapler je od značaja i maksimalni napon izolacije i on tipično iznosi nekoliko kV (za veće vrednosti može doći do električnog proboja između ulaza i izlaza kola) [1]. Drugi značajni parametri su: V_{CEmax} (maksimalni probojni napon kolektor-emitor izlaznog tranzistora kada je tranzistor u zakočenju) - predstavlja napon napajanja izlaznog kola, maksimalna ulazna struja direktno polarizovane LED diode $(I_{f(max)})$ - na osnovu nje izračunava se minimalna vrednost otpornosti koja se redno vezuje sa LE diodom, maksimalna vrednost inverznog napona na ulazu V_R – maksimalna vrednost probojnog inverznog napona na diode, propusni opseg optokaplera - određuje najvišu frekvenciju signala koji može da se prenese, probojni napon između ulaza i izlaza, t_{on} *i* t_{off} - predstavljaju vremena uključenja i isključenja optokaplera.

C. Tehnička dokumentacija optokaplera

Iz tehničke dokumentacije optokaplera izdvojene su vrednosti određenih parametara optokaplera koje se moraju poštovati kako ne bi došlo do prekida rada komponente ali i zbog preciznijeg i boljeg rada iste, kao i zavisnost vremena uključivanja i isključivanja optokaplera u odnosu na otpornost što se može videti na slici 6.

U tabeli I prikazane su vrednosti ulaznih parametara i njihove jedinice [3].

TABELA I
MAKSIMALNE VREDNOSTI ULAZNIH PARAMETARA

Naziv ulaznih paramentra, simbol i jedinica	Inverzni napon $V_R[V]$	Direktna struja $I_F[mA]$	Snaga disipacije $P_{diss}[mW]$
Vrednost	5	60	100

U tabeli II prikazane su vrednosti izlaznih parametara i njihove jedinice [3].

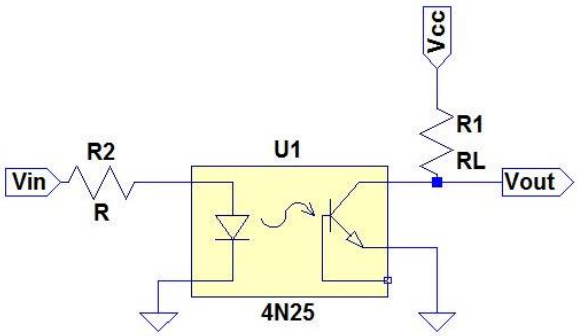
TABELA II
MAKSIMALNE VREDNOSTI IZLAZNIH PARAMETARA

Naziv izlaznih paramentra , simbol i jedinica	Probojni napon kolektor – emitor $V_{EBO}[V]$	Probojni napon emitor – baza $V_{EBO}[V]$	Kolektorska struja $I_C[mA]$	Snaga dispacije $P_{diss}[mW]$
Vrednost	70	7	50-100	150

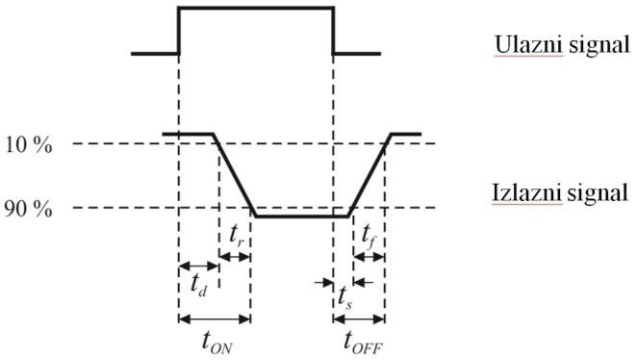
D. Karakteristična vremena

Na slici 4 prikazana je tipična konfiguracija za proučavanje dinamičkih karakteristika optokaplera. Sa nailaskom prednje ivice pozitivnog naponskog impulse, LE diode počinje da vodi, emituje svetlost koja pada na bazu fototranzistora koji počinje da vodi (najčešće je u oblasti zasićenja). Sa nailaskom zadnje ivice pozitivnog naponskoh impulsa LE dioda prestaje da vodi kao i fototranzistor koji je sada u oblasti zakočenja.

U konfiguraciji sa slike 5, izlazni signal odgovara ulaznom, samo je invertovan i kasni, zato što se ovaj prelaz ne može obaviti beskonačno brzo i uvek je potrebno vreme da se uspostavi stanje na izlazu. Kako bi vremena koja karakterišu dinamičke karakteristike optokaplera bila uporediva (kod različitih tipova optokaplera), uobičajeno je da ih merimo unapred pri definisanoj struji koja protiče kroz fotodiodu I_F, napona napajanja V_{CC}, otpornosti R_L, frekvenciji i faktoru ispune ulaznog signala (jer se na ulaz dovodi povorka periodičnih impulse, a ne samo jedan impuls) [2].

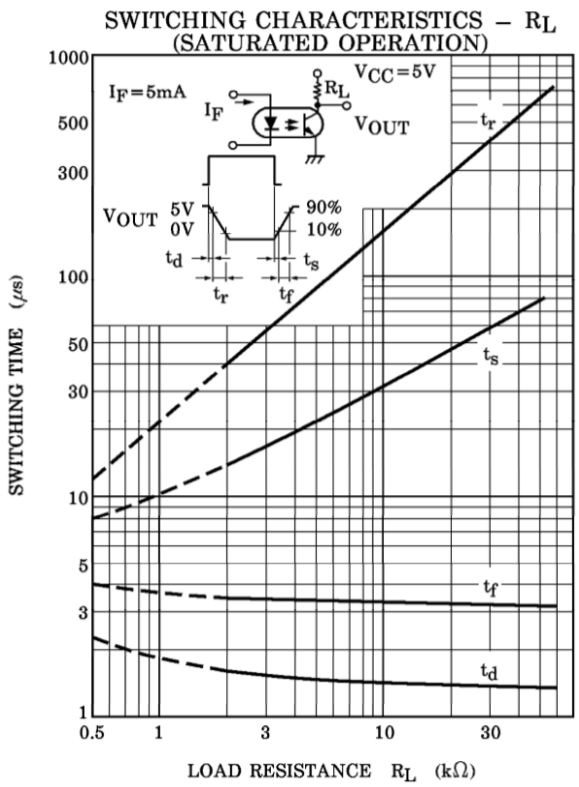


Slika 4. Tipična konfiguracija za proučavanje dinamičkih karakteristika optokaplera [2].



Slika 5. Grafički prikaz vremena isključivanja optokaplera u idealnom slučaju [2].

Vremena koja karakterišu dinamičke karakteristike optokaplera su:
t_d – vreme kašnjenja (od dolaska ulaznog signala do dostizanja 10% od vrednosti izlaznog signala)
t_F - vreme porasta (od 10% do 90% vrednosti izlaznog signala)
t_s - vreme skladištenja (od prestanka ulaznog signala do dostizanja 90% vrednosti izlaznog signala)
t_f - vreme opadanja signala (od 90% do 10% vrednosti izlaznog signala)
t_{on} - vreme uključenja optokaplera
t_{off} - vreme isključenja optokaplera.

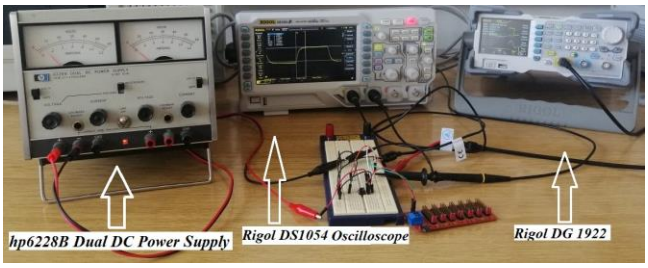


Slika 6. Zavisnost vremena uključivanja i isključivanja optokaplera u odnosu na otpornost [4].

III. OPIS EKSPERIMENTA

A. *Opis aparature*

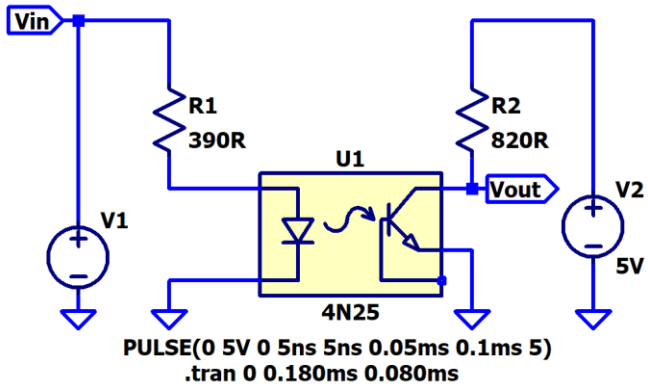
Za potrebe ovog eksperimenata, korišćeni su sledeći instrumenti prikazani na slici 4 (idući s leva na desno): „*hp6228B Dual DC Power Supply"* [5] – korišćen za napajanje, „*Rigol DS1054 Oscilloscope*" [6] – korišćen za posmatranje signala i „*Rigol DG 1922*" [7] – korišćen za generisanje signala u kolu.



Slika 7. Aparatura korišćena za potrebe eksperimenata.

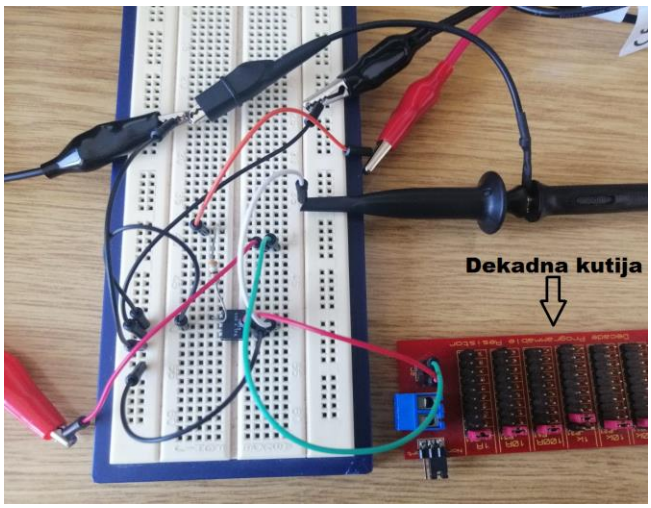
B. *Električna šema*

Kako bi se odredila vremena koja karakterišu dinamičke karakteristike optokaplera, prvo je ekspriment realizovan na protoploči gde su vremena merena na unapred pomenutim instrumentima, zatim je odrađena i simulacija u programu *LT Spice XVII* (Slika 8) [8].



Slika 8. Električna šema kola koje je realizovano u praktičnom delu rada.

Kao što je već pomenuto ekspriment je rađen i na protoploči. Uz pomoć dekadne kutije, radi lakše promene otpornosti, menjane su vrednosti otpornosti R2 (slika 4). Otpornosti su menjane u rasponu od 100 do 1000 Ω.



Slika 9. Realizovano kolo na protoploči.

U ovom eksperimentu je korišćen optokapler oznake 4N25 čija je tehnička dokumentacija prikazana u II delu rada [3]. Na slici se mogu uočiti i sonde instrumenata koji su korišćeni za potrebe ovog eksperimenta, koje su povezane u skladu sa šemom prikazanoj na slici 4.

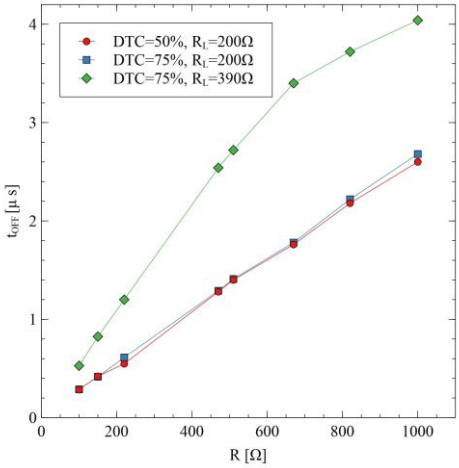
IV. REZULTATI ISTRAŽIVANJA

Nakon postavljanja instrumenata i formiranja kola na protoploči (slika 7 i slika 9), uz pomoć osciloskopa „*Rigol DS1054 Oscilloscope*", posmatrani su dobijeni signali (slika 12) i dobijeni su rezultati vremena uključivanja i isključivanja optokaplera prikazani u tabeli III.

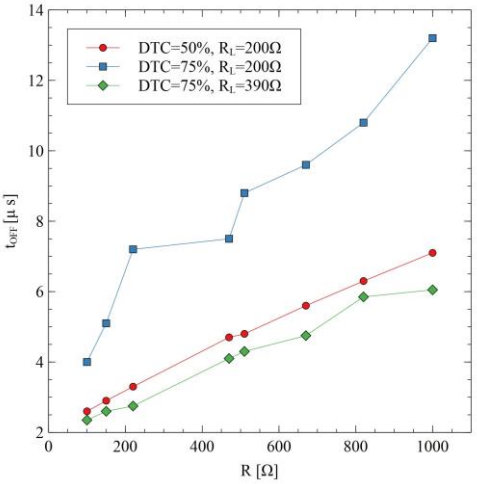
TABELA III
REZULTATI SNIMANJA VREMENA UKLJUČIVANJA I ISKLJUČIVANJA OPTOKAPLERA

Rx[Ω]	DTC=50% R_L=200 Ω		DTC=75% R_L=200 Ω		DTC=75% R_L=390 Ω	
	t_{on}[µs]	t_{off}[µs]	t_{on}[µs]	t_{off}[µs]	t_{on}[µs]	t_{off}[µs]
100	2.60	2.90	4.00	3.30	2.35	3.70
150	2.90	3.20	5.10	4.00	2.60	3.45
220	3.30	3.50	7.20	4.20	2.75	4.00
470	4.70	4.90	7.50	5.30	4.10	4.60
510	4.80	5.50	8.80	5.40	4.30	4.95
670	5.60	6.70	9.60	5.80	4.75	5.75
820	6.30	6.90	10.80	6.00	5.85	7.05
1000	7.10	8.20	13.20	6.20	6.05	7.35

Sa grafika (slika 10) se može primetiti da sa promenom parametra *DTC* i pri istoj otpornosti R_L, vreme uključivanja t_{on} je približno isto. Porast vremena t_{on} može se oučiti preko krive sa parametrima *DTC*=75% R_L=390Ω, odnosno pri porastu otpornosti.

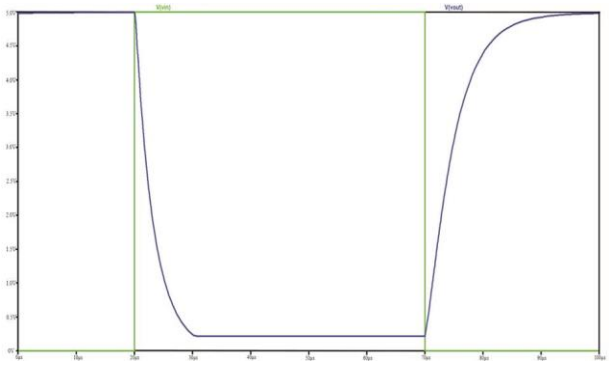


Slika 10. Grafički prikaz vremena uključivanja optokaplera.



Slika 11. Grafički prikaz vremena isključivanja optokaplera.

Sa grafika (slika 11) se može primetiti da sa promenom otpornosti R_L i pri istom parametru *DTC*, vreme uključivanja t_{off} je približno isto. Porast vremena t_{off} može se oučiti preko krive sa parametrima *DTC*=75% R_L=200Ω, odnosno pri smanjenju otpornosti.



Slika 12. Grafički prikaz vremena isključivanja optokaplera u realnom vremenu.

V. ZAKLJUČAK

U radu je detaljno obrađena tema dinamičkih karakteristika optokaplera koji se bazira na eksperimentalnom merenju pri različitim vrednostima otpornosti i CTR-a.

ZAHVALNOST

Veliku zahvlnost dugujemo prof. dr Danijelu Dankoviću, doc. dr Emiliji Živanović i studentu doktorskih studija Nikoli Mitroviću koji su nam omogućili svu potrebnu opremu i pomogli svojim savetima pri izradi rada.

LITERATURA

[1] Zoran Prijić, Aneta Prijić, *Uvod u poluprovodničke komponente i njihovu primenu*, Univerzitet u Nišu, Elektronski fakultet, 2014.
[2] Danijel Danković, *Laboratorijski praktikum sa zadacima za samostalan rad iz predmeta Komponente za telekomunikacije*, Univerzitet u Nišu, Elektronski fakultet, Niš, 2018.
[3] Optokapler 4N25 datasheet. Dostupno na: https://datasheetspdf.com/pdf/794957/Vishay/4N25/1
[4] Slika 6 je preuzeta sa sajta: https://www.tme.eu/Document/de774e42ff1f35ff360e0f3898b9126f/4N25-SHORT.F.pdf
[5] hp 6228B Dual DC Power Supply datasheet Dostupno na: https://accusrc.com/uploads/datasheets/3921_6228b.pdf
[6] Rigol DS1054 Oscilloscope datasheet Dostupno na: https://datasheet.octopart.com/DS1054Z-Rigol-datasheet-26580497.pdf
[7] Rigol DG1922 datasheet Dostupno na: https://www.sicamax.ch/downloads/datasheet_dg1022_en.pdf
[8] LTspice XVII, Linear Tehnology. Dostupno na: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

Sixth Sense – uređaj za slepe i slabovide osobe zasnovan na Doplerovom efektu

Milica Lazarević, Lazar Tolić, Nikola Mišić, Mihajlo Jovanović, Isidora Ilić, Danijel Jovanović, Lazar Karašićević, Petar Milanović

Sadržaj – U ovom radu je predstavljen uređaj za slepe i slabovide osobe koji radi na principu Doplerovog efekta. Ovaj uređaj obaveštava korisnika o udaljenosti prepreka ispred njega. Predstavljeni uređaj pomoću ultrazvučnog senzora, koji se nalazi u bedžu, određuje rastojanje prepreke od korisnika, a zatim preko bluetooth tehnologije šalje tu informaciju do korisnika koji je prima u obliku vibracija.

I. UVOD

Ultrazvuk je zvuk čija je frekvencija iznad gornje granice čujnosti za normalno ljudsko uvo, a koja iznosi 20 kHz [1]. Kada je frekvencija zvučnog talasa veća od 109 Hz, govori se o hiperzvuku. Talasne dužine ultrazvučnih talasa u vazduhu iznose do 0,5 μm, a u tečnostima i čvrstim telima veće su od 4 do 12 puta, zbog veće brzine širenja ultrazvuka. U prirodi se ultrazvuk pojavljuje uz zvučne talase, a veštački se može proizvesti ultrazvučnim generatorima, odnosno pretvaračima drugih oblika energije u ultrazvučnu energiju [2].

Neke životinje (npr. psi, delfini, slepi miš, miševi, ...) mogu čuti ultrazvuk jer imaju višu gornju graničnu frekvenciju od čoveka [3][4][5]. Mlađe osobe, a posebno deca, mogu čuti neke zvukove visokih frekvencija. Što je čovek stariji, gornja granica čujnosti mu pada, što znači da sve slabije čuje zvukove visokih frekvencija. Visoke frekvencije su sastavni deo spektra frekvencija koje proizvodi neki izvor zvuka, a spektar zvučnih frekvencija čini boju zvuka. Opadanjem čujnosti visokih frekvencija starenjem, starijim ljudima se menjaju i boje zvuka, što znači da simfonijski orkestar ili zvuk violine drugačije čuje dete od šest, odrastao čovek od 30 ili starija osoba od 80 godina.

Jedna od primena ultrazvuka u medicini je - ultrazvučna dijagnostika [6]. Princip rada ultrazvučnog primopredajnika je vrlo jednostavan: ultrazvučni talas se generiše, odbija od prepreke i prema vremenu potrebnom da se talas vrati određuje se udaljenost.

Doplerov efekat je pojava da usled kretanja prijemnika ili predajnika dolazi do menjanja frekvencije ultrazvučnih talasa, na strani prijemnika. Ako se prijemnik i predajnik kreću jedan ka drugom, frekvencija raste, a ako se prijemnik i predajnik udaljavaju jedan od drugog, frekvencija opada. Na primer, Doplerov efekat se može primetiti na auto-putu: buka koju motor automobila pravi dok se približava drugačija je od one koja se čuje dok se udaljava.

Uređaj za slepe i slabovide osobe, Sixth Sense, olakšava slepim i slabovidim osobama kretanje u prostoru i smanjuje potrebu za asistencijom. Zahvaljujući monokristalnom mini solarnom panelu omogućena je potpuna autonomnost osnovnog uređaja i pri vrlo slaboj svetlosti. Bluetooth tehnologija omogućava stalnu bežičnu povezanost uređaja za detekciju i uređaja za signalizaciju prepreka.

M. Lazarević, L. Tolić, N. Mišić, M. Jovanović, I. Ilić, D. Jovanović, L. Karašićević, P. Milanović, učenici na smeru Elektrotehničar Informacionih Tehnologija, Tehnička škola u Mladenovcu, Vuka Karadžića 75, 11400 Mladenovac, Srbija. E-mail: sixthsenseserbia@gmail.com

II. PRINCIP RADA UREĐAJA

Uređaj Sixth Sense je baziran na principu Doplerovog efekta, na principu koji koriste slepi miševi u letu. Sam uređaj se sastoji od sledećih komponenata:

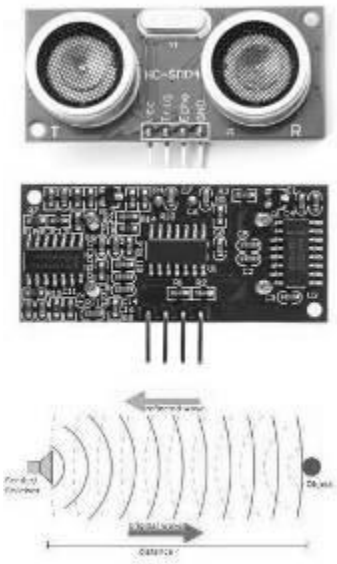
1. HC-SR04 ultrazvučnog senzora,
2. HC-05 Arduino bežičnog Bluetooth transmitera,
3. Multifunkcionalnog uređaja, koji služi za informisanje korisnika o razdaljini prepreke pomoću vibracija,
4. Programabilnog Arduino nano v3 uređaja, čijim se programiranjem uređaj može prilagođavati svakom korisniku,
5. Punjivih baterija za napajanje,
6. Monokristalnog mini solarnog panela,

HC-SR04 ultrazvučni senzor (slika 1.) predstavlja primopredajnik i odlično detektuje predmete u opsegu od 2 cm do 400 cm na frekvenciji od 40 kHz. Senzor pokriva horizontalni ugao do 15^o. Sam koncept ove tehnologije je jednostavan. Senzor emituje zvučne impulse visoke frekvencije, koji se, u slučaju da se ispred senzora nalazi prepreka, odbijaju od nje ka senzoru. Ako su impulsi detektovani nakon emitovanja, možemo pretpostaviti da se ispred senzora nalazi prepreka [7].

HC-05 Arduino bežični Bluetooth transmiter služi da izračunatu udaljenost prepreke od korisnika pošalje do multifunkcionalnog uređaja putem Bluetooth-a (slika 2.). Izradom posebnog softvera omogućeno je korišćenje multifunkcionalnih uređaja koji služe za primanje informacija o trenutnoj udaljenosti prepreke od korisnika i pomoću vibracija ga obaveštavaju o tome.

Solarni panel se odnosi na panel dizajniran da apsorbuje sunčeve zrake kao izvor energije za proizvodnju

električne energije ili grejanja. Monokristalni solarni panel omogućava potpunu autonomnost osnovnog uređaja i napajanje "zelenom energijom" U pitanju je panel dimenzija 80x55 [mm] koji obezbeđuje napon od 5V i pri vrlo slaboj svetlosti (slika 3.).



Slika 1. HC-SR04 ultrazvučni senzor [8].



Slika 2. HC-05 Arduino bežični Bluetooth transmitter [9].



Slika 3. Monokristalni mini solarni panel [10].

Kao drugi izvor napajanja uređaj koristi eksternu bateriju (za mobilne uređaje). Baterija je takvih dimenzija (95x63x7 [mm] i 54 gr.) da predstavlja minimalno opterećenje korisniku (slika 4.).



Slika 4. Punjiva baterija za napajanje.

Arduino nano (slika 5.) je dimenzija 43,18x18,54 mm što omogućava i korišćenje kutije manjih dimenzija za smeštanje samog elektronskog sklopa uređaja.

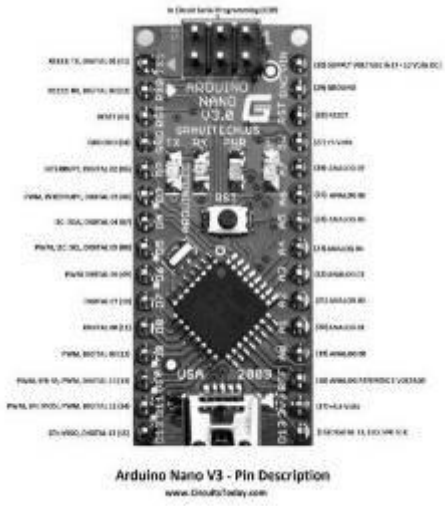
Programiranje Arduino nano uređaja se radi u programu scketch tsm3.ino. Fabričko podešavanje uređaja Sixth Sense je urađeno tako da uređaj radi na sledeći način:

- Ukoliko je rastojanje prepreke manje od 50 cm tada se tri puta naizmenično uključuje vibracija, pa se napravi dodatna pauza od 0,2 sekunde.
- Ukoliko je rastojanje prepreke manje od 100 cm tada se dva puta naizmenično uključuje vibracija, pa se napravi dodatna pauza od 0,4 sekunde.
- Ukoliko je rastojanje prepreke manje od 200 cm tada se jednom naizmenično uključuje vibracija, pa se napravi dodatna pauza od 0,6 sekundi.
- Ukoliko je rastojanje prepreke manje od 300 cm tada se jednom naizmenično uključuje vibracija, pa se napravi dodatna pauza od 1,4 sekunde.

Na ovakav način uređaj otkriva prepreke ispred senzora na rastojanju do 3 metra. Izmenom programskog koda moguće je promeniti ritam uključenja i isključenja motora koji stvara vibracije čime se omogućava prilagođavanje uređaja zahtevima svakog korisnika.

U samom planiranju izrade uređaja bila je uključena ideja korišćenja zvuka kao signala za obaveštavanje korisnika, ali konsultacijom sa slepim i slabovidim osobama, kompanija je došla do saznanja da zvučni signali više smetaju slepim i slabovidim osobama nego vibracije, iz razloga što je njihovo čulo sluha razvijenije i više se oslanjaju na njega nego ostali ljudi.

Sve elektronske komponente koje se koriste za izradu uređaja Sixth Sense su reciklabilne čime se izlazi u susret zahtevima svih konvencija o zaštiti životne sredine.



Slika 5. Programabilni Arduino nano v3 uređaj [11].

Ceo uređaj je kompletiran ubacivanjem komponenti u kutiju (slika 6.) koja služi pre svega kao zaštita elektronskim komponentama uređaja (slika 7.) i radi boljeg izgleda samog uređaja. Kutija je izrađena pomoću 3D štampača zbog brže i lakše proizvodnje.



Slika 6. Kućiste uređaja.



Slika 7. Unutrašnjost uređaja.

```
const int pingPin = 7;

void setup() {
pinMode(echoPin, INPUT); pinMode(motorPin,OUTPUT);
}

void loop() {

trajanje = pulseIn(echoPin, HIGH);

cm = trajanje / 29 / 2;

if(cm < 50)
{
digitalWrite(motorPin, HIGH);
}
else if(cm < 100)
{
digitalWrite(motorPin, HIGH);
```

Slika 8. Deo koda za podešavanje uređaja

A. Testiranje realizovanog sistema i diskusija

Testiranje rada realizovanog sistema izvršeno je u školi za slepe i slabovide osobe "Veljko Ramadanović" u Zemunu kako bi slepe i slabovide osobe mogle praktično da testiraju uređaj. Njihove primedbe i sugestije pomogle su u daljem razvoju proizvoda za koji su rekli da ima izuzetan praktični potencijal. Jedan od problema sa kojim se susreću slepe i slabovide osobe jesu rupe na putevima ili stepenice. Taj problem je moguće rešiti sa mikro kamerom koja će skenirati put ispred njih i davati im zvučnu informaciju o tipu prepreke na koju nailaze. Moguće je minimizovati veličinu proizvoda, za šta će se u buduće, umesto Arduino modula, koristiti Adafruit Flora komponente koje za svoj rad zahtevaju i niži napon napajanja, što dodatno vodi minimizaciji uređaja, pošto se mogu koristiti solarni paneli manjih dimenzija. Takođe je u planu i izrada aplikacije kojom će korisnik moći da podešava karakteristike senzora u realnom vremenu.

III. ZAKLJUČAK

Prototip uređaja, tj. verzija 1.0, je realizovan na PCB pločici koja je povezivala arduino modul, vibracione motore koji su stajali na ramenima i ultrazvučni senzor. Opremljen je sa Arduino nano v3, HC-SR04 ultrazvučnim senzorom i dva vibraciona motora.

Korišćenjem prve verzije uređaja ustanovljeno je da bi bilo najbolje uređaj napraviti kao bedž i smanjiti mu dimenzije. Takođe su merene i maksimalna i minimalna udaljenost koje senzor može da registruje i ispravljale su se greške koje su se pojavljivale.

Verzija 2.0 je u odnosu na verziju 1.0 unapredila sve njegove postojeće karakteristike i dodala još novih. U verziji 2.0 nije bilo potrebno praviti PCB pločicu već je sve direktno bilo spojeno na Arduino modul. Dodata je podrška za Bluetooth konekciju koja omogućava povezivanje uređaja sa bilo kojim pametnim telefonom, pametnim satom ili fitnes narukvicom koji bi vibrirali umesto

vibracionih motora koji su uklonjeni u ovoj verziji zbog dodatka za Bluetooth. Uređaj se nalazi u kutiji koja je napravljena pomoću 3D štampača i kači se na odeću kao bedž čije su trenutne dimenzije 110x75x45 [mm] i težina 194 grama. Nova, 3.0 verzija uređaja, sa Adafruit Flora komponentama, je projektovanih dimenzija 90x55x18 [mm] i težine 72 grama. Jedan od dodataka u verziji 2.0 je i solarni panel.

Sixth Sense je uređaj realizovan tako da korisnici kojima je namenjen mogu lako da ga primenjuju, i putem naše aplikacije prilagođavaju rad uređaja svojim potrebama. Prednost je, takođe, što uređaj pored napajanja punjivim baterijama ima i mini solarni panel preko koga se baterije pune sve vreme dok korisnik koristi uređaj.

ZAHVALNICA

Učenička kompanija „Sixth Sense“ se zahvaljuje kompaniji „Primus Home Inspection“ iz Vankuvera (Kanada) na nesebičnoj pomoći prilikom nabavke neophodne opreme za realizaciju projekta. Takođe kompanija se zahvaljuje i svim nastavnicima Tehničke škole iz Mladenovca na savetima i stručnoj pomoći u realizaciji uređaja.

LITERATURA

[1] Corso, J. F. (1963). „Bone-conduction thresholds for sonic and ultrasonic frequencies”. Journal of the Acoustical Society of America. 35 (11): 1738—1743.

[2] Ultrazvuk, "Hrvatska enciklopedija", Leksikografski zavod Miroslav Krleža, www.enciklopedija.hr, 2016.

[3] Novelline 1997, стр. 34–35.

[4] Kaplan, Matt (17. 7. 2009). „Moths Jam Bat Sonar, Throw the Predators Off Course”. National Geographic News.

[5] Some Moths Escape Bats By Jamming Sonar (video), https://www.npr.org/templates/story/story.php?storyId=106733884.

[6] Papadakis, Emmanuel P. (ed) (1999). Ultrasonic Instruments & Devices. Academic Press. стр. 752. ISBN 978-0-12-531951-5.

[7] Otpornik, (2013). HC-SR04 – Ultrazvučni senzor rastojanja [Online], dostupno na: http://www.otpornik.com/elektronika/komponente/hc-sr04-ultrazvucni-senzor-rastojanja.html, [pristupljeno 25.10.2019.]

[8] tutorialspoint, (2019). Hc-06 Bluetooth Module [Online], dostupno na: https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm, [pristupljeno 24.10.2019.]

[9] instructables circuits, (2019). STM32F103: Bluetooth Module HC05 / HC06 (using Mbed.h) [Online], dostupno na: https://www.instructables.com/id/STM32F103-Bluetooth-Module-using-Mbedh/ , [pristupljeno 24.10.2019.]

[10] wavetra, (2019). Solar Panel 5V 6V 12V Mini Solar System DIY For Battery Cell Phone Chargers Portable 0.15W 0.6W 1W 1.25W 1.5W Solar Cell [Online], dostupno na: https://www.wavetra.com/products/anbes-solar-panel-5v-6v-12v-mini-solar-system-diy-for-battery-cell-phone-charger-potrable-0-15w-0-6w-1w-1-25w-1-5w-solar-cell , [pristupljeno 30.10.2019.]

[11] CIRCUITS TODAY, (2018). Arduino Nano Tutorial – Pinout & Schematics [Online], dostupno na: http://www.circuitstoday.com/arduino-nano-tutorial-pinout-schematics, [pristupljeno 24.10.2019.]

Projektovanje okruženja za testiranje komponenata

Nikola Mitrović

Sadržaj - U ovom radu je prikazan proces projektovanja okruženja za testiranje komercijalnih komponenata i modula na štampanoj ploči. Centar sistema predstavlja mikrokontroler PIC16F877A. Okruženje je projektovano sa ciljem da može da testira jako veliki broj komponenata. Međutim, zbog fizičkih i praktičnih ograničenja, pokriveni su samo najzastupljeniji modeli i protokoli. Komponente će biti testirane tako što se za njihovo testiranje prvo napiše odgovarajući kod. Kod bi trebalo da bude napisan tako da se testira što je moguće više osobina komponenata ili modula. Napisani kod se programira u mikrokontroler, pa se proverava ispravnost rada komponente.

I. UVOD

Testiranje komponenata predstavlja bitan proces u njihovoj proizvodnji. Kako bi komponenta pronašla put od proizvođača do korisnika, proces testiranja je neophodan. Ipak, povećanjem kvantiteta proizvedenih modula, nema više realnih mogućnosti da se svaki proizvedeni modul testira. Posledica toga je ta da krajnji korisnik neretko dobije module koji su neispravni, ili nisu u potpunosti u skladu sa tehničkim specifikacijama. Sigurnost da određeno kolo ili modul radi u skladu sa očekivanjima je jako bitno prilikom debagiranja (*debugging*) sistema u razvoju. Zbog toga, jedno od mogućih rešenja bi bilo projektovati okruženje koje bi u svakom trenutku moglo da ispita karakteristike modula. Ovakvo okruženje bi bilo moguće realizovati na više načina, ali način koji je najprisutniji u radu se ogleda u tome da se „sumnjiv“ modul nezavisno poveže na proto ploči. Proto ploča se bira jer nudi najveću fleksibilnost, široko je dostupna i relativno jednostavna za povezivanje. Ipak, kod rada sa kompleksnijim sistemima, povezivanja na protoploči mogu dovesti do postojanja jako velikog broja žica (*jumpera*) na ploči, nesigurnosti da li je na ploči sve dobro povezano u šumi žica, pa i na kraju krajeva, da li je sama proto ploča ispravna. Zbog toga, rešenje koje je komplikovanije i skuplje za izradu, ali svakako preglednije i pouzdanije za dugoročno korišćenje je projektovanje testnog okruženja na štampanoj ploči. Pogodnosti ovog tipa realizacije se ogledaju i u lakoći testiranja i u uštedi vremena. Sa tendencijom da se realizuje okruženje koje će dugoročno kratiti vreme debagiranja sistema, okruženje je realizovano na štampanoj ploči.

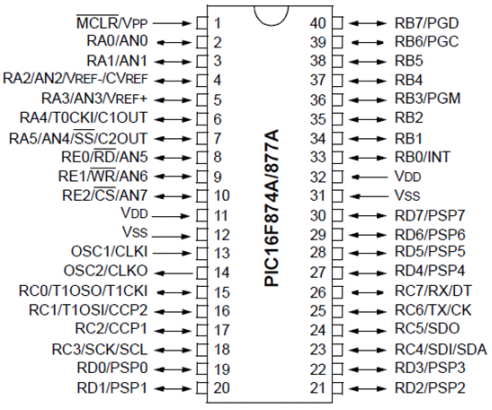
N. Mitrović je student doktorskih studija pri Katedri za Mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikola.mitrovic.nis@gmail.com

II. PROJEKTOVANJE ŠTAMPANE PLOČE

Projektovanje štampane ploče rađeno je pomoću softverskog alata *Altium Designer* [1]. Okruženje je podeljeno u nekoliko blokova, od kojih je svaki projektovan na zasebnom šematiku. Svi šematici su hijerarhijski organizovani unutar glavnog šematika u kom su prikazani svi blokovi.

A. Blok Mikrokontrolera i Periferala

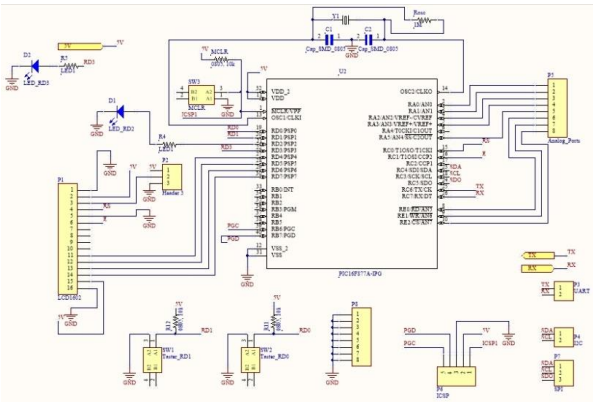
Centralnu upravljačku jedinicu ovog okruženja čini mikrokontroler (*Micro Controller Unit* – MCU), konkretno model proizvođača *Microchip*, PIC16F877A [2]. Karakteriše ga *flash* memorija, koja omogućava veliki broj upisa i ispisa, odnosno, omogućava veliki broj reprogramiranja. Ovaj MCU poseduje veliki broj GPIO pinova (*General Purpose Input Output pins*), čak 33 pina, grupisana u 5 portova, dok ostalo čine sistemski pinovi.



Slika 1. Raspored pinova kola PIC16F877A.

Odabrano kolo ne poseduje interni oscilator, pa je potrebno povezati eksterno oscilatorno kolo na odgovarajućim pinovima. Kako bi se obezbedila mogućnost resetovanja, MCLR pin je povezan preko tastera i *pull-up* otpornika sa naponom napajanja kola. Pinovi za napajanje (VDD) i masu (VSS) se nalaze sa obe strane, pa je prilikom projektovanja potrebno voditi računa o tome da odgovarajući pinovi sa obe strane budu povezani. Kako bi okruženje omogućilo i testiranje većeg broja istih MCU, kolo neće biti zalemljeno direktno na ploču, već će biti montirano na odgovarajuće podnožje (*socket*) za kućište DIP 40. Prilikom projektovanja bloka čiji je glavni deo PIC MCU, potrebno je ostaviti mesta i za ICSP port (*In-Circuit Serial Programming*). Preko ovog

porta, programatorom PIC *Kit* [3] moguće je programirati MCU. Pinove ICSP porta je potrebno povezati sa svim pinovima na MCU, kako bi programiranje bilo moguće izvršavati svakog puta (MCLR pin, VDD pinovi, VSS pinovi, PGD pin (*Programming Data*), PGC pin (*Programming Clock*)).

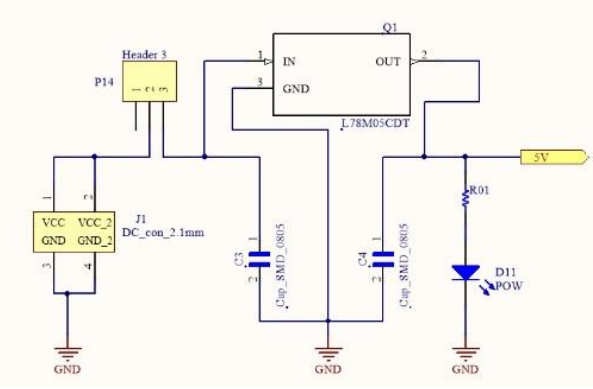


Slika 2. Sastavne jedinice bloka MCU i periferala.

Pod blokom periferala podrazumevaju se razne komponente koje nemaju svoju konkretnu namenu, već namena u potpunosti zavisi od zadatka koji MCU treba da izvršava. Sastoji se od dva tastera koji su povezani na pinove RD0 i RD1, kao i dve LE diode koje su povezane na pinove RDx i RDx. Takođe, u sastav ovog bloka ulaze i dva *headera* (zaglavlja) od 8 *socketa* (trnova). Svaki trn kod jednog od *headera* je povezan sa odgovarajućim pinom koji ima mogućnost da bude podešen kao pin za analogni ulaz (RA0 – RA7). Trnovi drugog *headera* su kratkospojeni i povezani sa VSS pinovima MCU (pinovi za masu).

B. Blok napajanja

Blok napajanja ima zadatak regulisanja napona napajanja za rad kola. Okruženje se može napajati preko standardnog *DC Power Jack*-a opsegom napona do 12 V. Centralna komponenta ovog bloka je regulator napona LM7805, konkretno SMD varijanta L78M05CDT[4].

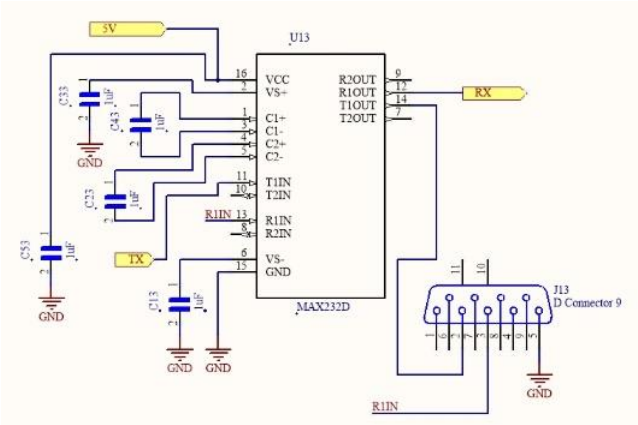


Slika 3. Regulator napajanja oko kola LM7805.

Opseg napona eksternog napajanja određen je konkretno ovom komponentom. Kondenzatori su odabrani i povezani prema tehničkoj dokumentaciji regulatora, kako bi se na izlazu dobio stabilisani napon od 5 V, a dodat je i svetlosni indikator u vidu LE diode.

C. Blok UART komunikacije

Ovaj tip komunikacije moguće je realizovati na dva načina. UART (*Universal Asynchronous Receive Transmit*) komunikacija se suštinski odvija između dva uređaja (modula, kola ili slično) tačnije, između *Trasmit* terminala jednog uređaja (u ovom slučaju TX pin, RC6) i *Receive* terminala drugog uređaja (kod ovog mikrokontrolera RX pin, RC7) [5]. Jednostavniji način realizacije ove komunikacije je baš korišćenjem ovih pinova. Ukoliko je cilj poslati podatke, onda se TX pin (RC6) mikrokontrolera povezuje na RX pin uređaja sa kojim treba uspostaviti komunikaciju, dok je za potrebe prijema obrnuto. Drugi tip komunikacije je zastupljeniji kada je potrebno povezati mikrokontroler sa računarom, a ne sa nekim drugim senzorskim modulom i označen je kao RS232 komunikacija.



Slika 4. Šema RS232 komunikacije.

Blok RS232 komunikacije je povezan oko integrisanog kola MAX232 [6], koje ima zadatak da izvrši prilagođavanje napona za komunikaciju sa 5 V (koliko koristi MCU) na 12 V koliko koristi kompjuterski serijski port. Na ploči je povezan i odgovarajući konektor za serijsku komunikaciju sa računarom, DB9. Na ovaj način, omogućena je komunikacija sa serijskim portom računara.

D. Blok I^2C i SPI protokola za komunikaciju

Dva najzastupljenija protokola za serijsku komunikaciju su I^2C (*Inter-Integrated Circuit*) i SPI (*Serial Peripheral Interface*). I^2C koristi dva pina za komunikaciju, po ugledu na TX i RX pin prilikom UART komunikacije, ali su ovi pinovi označeni kao SDA (*Serial Data*), pin RC4 na centralnom MCU i SCL (*Serial Clock*), pin RC3 na MCU [7]. Takođe, komunikacija se odvija

između dva uređaja, od kojih je jedan označen kao *Master*, a drugi kao *Slave*. *Master* uređaj određuje taktni signal (SCL), i upravlja naredbama o početku prijema i slanja. *Slave* prihvata izdate naredbe i započinje prijem ili slanje podataka preko linije za podatke SDA. Prilikom pisanja koda za testiranje uređaja sa kojim treba komunicirati preko I^2C, treba posebno obratiti pažnju na to da li će prilikom komunikacije MCU biti određen da bude *Master* ili *Slave*.

SPI protokol za komunikaciju koristi tri pina, SDO (*Serial Data Out*), RC5 pin na MCU, SDI (*Serial Data In*), RC4 pin i SCK (*Serial Clock*), pin RC3 [7]. U slučaju da je MCU određen kao *Slave*, potrebno je povezati i četvrti pin, SS (*Slave Select*), pin RA5. Ovaj protokol je jako sličan I^2C protokolu, i takođe predstavlja komunikaciju između *Master* i *Slave* uređaja. Međutim, način organizacije paketa podataka u protokolu se razlikuje. Za prenos podataka, koriste se dve linije, SDO i SDI, za razliku od I^2C protokola. Prilikom projektovanja okruženja, pretpostavljeno je da će MCU biti korišćen kao *Master* uređaj, pa su za potrebe testiranja komunikacije ostavljena samo tri pina.

U slučaju kada nijedan od protokola za komunikaciju nije deo pisanja koda za konkretnu primenu, pinovi RC3, RC4 i RC5 se mogu tretirati kao GPIO pinovi.

E. Blok za LCD

Jedan od uređaja koji se najčešće sreće u *embedded* sistemima je displej. Postoji velika raznovrsnost displeja, kako po obliku i dimenzijama, tako i po materijalima i načinu komunikacije, ali, jedan od najzastupljenijih modela displeja je QAPASS LCD1602 (*Liquid Crystal Display*) [8]. Može da ispisuje po 16 karaktera u dva reda. Sve karaktere može da ispisuje samo u jednoj boji i poseduje i pozadinsko osvetljenje. Iako poseduje ugrađene module koji ga čine jako jednostavnim za programiranje, ovaj tip displeja koristi veliki broj pinova za povezivanje (najmanje 9 pinova, a neretko i svih 16 pinova). Svi pinovi potrebni za komunikaciju sa displejem povezani su na port B odabranog MCU. Blok za LCD se, pored samog displeja, sastoji i od potenciometra koji ima ulogu podešavanja kontrasta displeja. Ovaj tip displeja projektuje se tako da sa gornje strane postoji 16 metaliziranih rupa koje bi trebalo povezati sa drugim uređajima. Kako bi korisnicima bilo olakšano rukovanje displejem, na metalizirane rupe potrebno je zalemiti letvice sa trnovima.

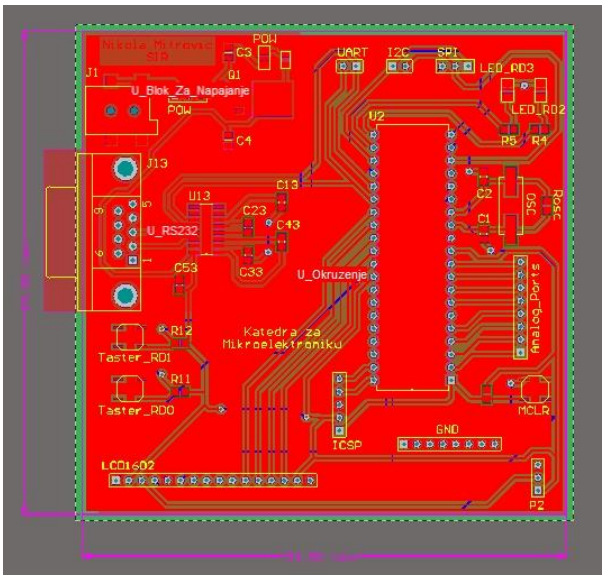
III. RASPOREĐIVANJE KOMPONENATA NA PLOČI

Prilikom projektovanja štampane ploče, bitno je na odgovarajući način rasporediti komponente na štampanoj ploči, odnosno projektovati optimalni *layout* [9]. Treba voditi računa o tome da se konektori za kablove stavljaju po ivicama ploče, kako ne bi prilikom povezivanja prekrivali druge komponente (naravno ukoliko za tim nema striktne potrebe). Takođe, poželjno je da naznake na ploči, oznake komponenata i slično ne prekrivaju otiske komponenata,

Prilikom rutiranja ploče postoje određena pravila kojih je potrebno pridržavati se, kako bi se izbegle greške. Povećanjem broja komponenata, a samim tim i otisaka na štampanoj ploči, povećava se i broj veza (ruta) koje je potrebno spojiti. Kako je cilj projektovati okruženje koje će zauzimati što je moguće manji prostor, a opet delovati kompaktno i pristupačno, potrebno je povezivati pinove komponenata koristeći obe strane ploče. Sa druge strane, kako se na štampanim pločama uglavnom montiraju SMD komponente, na pločama uglavnom postoji mali broj rupa.

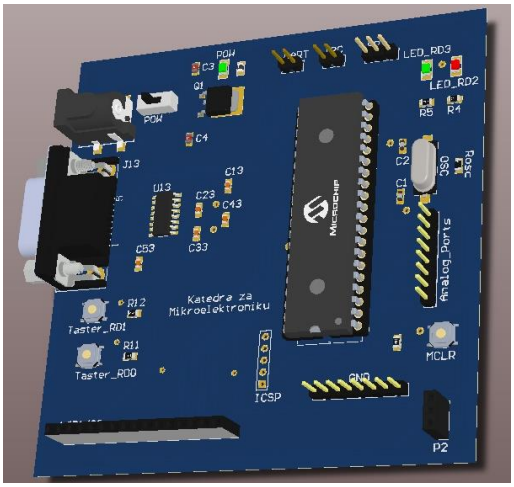
Korišćenjem opcije *Autoroute*, softverski alat *Altium Designer* nudi mogućnost da sam poveže sve projektovane veze. Prilikom ovog povezivanja, napraviće veliki broj vija koje mogu ili ne moraju biti linearno postavljene. Iako korišćenje ove komadne drastično smanjuje vreme projektovanja ploče, neretko je moguće povezati pinove sa manje vija i manje lomljenja linija ukoliko se rutiranje vrši ručno. Zbog limitarnih dimenzija ploče, a radi maksimalnog iskorišćenja površine ploče, okruženje je izrutirano ručnim povlačenjem ruta.

Kada je rutiranje završeno, proces projektovanja se završava nalivanjem mase preko ploče. Finalni izgled *layouta* štampane ploče prikazan je na slici 5.



Slika 5. *Layout* projektovanje štampane ploče.

U zavisnosti od namene ploče, neretko je potrebno dodati i 3D modele komponenata, kako bi korisnik bio siguran da ploča staje u odgovarajući prostor ili kutiju. Izgled ploče sa 3D modelima komponenata prikazan je slici 6. Proces projektovanja završava se izvozom *gerber* fajlova [9], i *NC drill* fajlova. Ovi podaci predstavljaju fajlove za fabričku izradu ploče.



Slika 6. 3D prikaz projektovane ploče.

IV. Testiranje komponente okruženjem

Primer testiranja komponente okruženjem biće prikazan na testiranju LCD1602. Kao što je ranije opisano, ovaj modul ima 16 pinova, i na okruženju je ostavljen poseban blok za pinove displeja. Prvo, potrebno je povezati displej na odgovarajuće mesto na ploči. Potom, pristupa se pisanju programskog koda koji ima zadatak da proveri da li se karakteri regularno ispisuju na svakom mestu na displeju. Za pisanje koda, moguće je koristiti bilo koji programski jezik, sve dok softverski alat ima mogućnost izvoza *hex* fajla (heksadecimalnog fajla). Ovaj fajl se programira u MCU preko ICSP porta. Radi preglednosti, testni kod neće biti prikazan i detaljnije analiziran, već će se prikazati samo rezultati testa.

Pokretanjem testa, karakter po karakter kreće da se ispisuje redom u prvom pa u drugom redu, na svako mesto, mesto po mesto. Sadržina testa se sastoji od provere da li će karakter na svakom mestu biti kompletno prikazan, da li se pozadinsko osvetljenje ponaša u skladu sa očekivanjima i da li ispisivanje traje onoliko koliko je predviđeno.



Slika 7. Displej istovremeno ispisuje karaktere, formira reči.

Ukoliko nešto od navedenog nije u skladu sa tehničkom dokumentacijom i pokrenutim testom, potrebno je zabeležiti ishode i pokušati ponovo. U drugom testu ispitati da li se, ukoliko je takva komanda softvera, neki karakteri mogu ispitivati istovremeno ili ne. Rezultati ovog testiranja prikazani su na slici 7. U zavisnosti od konkretne primene modula, moguće je ispitivati konkretnije osobine. Na taj način, debagiranje eventualnih grešaka prilikom projektovanja je značajno olakšano.

V. Zaključak

U radu je dat proces projektovanja okruženja za testiranje komponenata i modula na štampanoj ploči. Detaljnije su objašnjeni svi blokovi kojima je moguće testirati širok opseg modula i kola. Mogu se testirati svi moduli čiji rad zahteva korišćenje UART, SPI i I2C komunikacije, kao i senzori i druge komponente sa izlaznim analognim signalom. Prikazano je i realizovano okruženje kao i testiranje standardnog 1602 displeja. Prostor za napredak ogleda se u povećanju gabarita okruženja, i samim tim omogućavanju da se pokrije mogućnost testiranja još većeg broja protokola i modula. Uz to, na ploču je moguće dodati blok za programiranje mikrokontrolera, i samim tim omogućuti da se upisivanje koda u kolo ne vrši posredno (korišćenje PIC *Kit* uređaja), već neposredno, korišćenjem računara.

Zahvalnica

Autor se ovom prilikom zahvaljuje osoblju Katedre za mikroelektroniku sa Elektronskog fakulteta u Nišu za stimulisanje ideje i podršku pri radu. Takođe, autor se zahvaljuje kolegi Milanu Stojanoviću za pomoć pri izradi štampane ploče i podršku u radu. Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026.

Literatura

[1] Altium Designer 16, Altium. Dostupno na: https://www.altium.com/altium-designer/
[2] PIC16F877A datasheet. Dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/395.pdf
[3] PIC KIT 2 datasheet. Dostupno na: http://ww1.microchip.com/downloads/en/devicedoc/5155.pdf
[4] LM7085 datasheet. Dostupno na: https://eu.mouser.com/Datasheets/_/?Keyword=L78M05&FS
[5] Stanford University „*Circuit Basics: Basics of UART Communication*“ 2011. Dostupno na: https://web.stanford.edu/class/cs140e/notes/lec4/uart-basics
[6] MAX232 datasheet. Dostupno na: http://www.ti.com/lit/ds/symlink/max232.pdf
[7] Katedra za Mikroelektroniku, „*Praktikum za lasersku elektroniku*“ Elektronski fakultet, Univerzitet u Nišu.
[8] QAPASS1602 datasheet. Dostupno na: https://funino.de/1602
[9] D. Danković „*Projektovanje štampanih ploča – Laboratorijski praktikum*“ Elektronski fakultet, Univerzitet u Nišu.

Studije i život na Univerzitetu Balearskih ostrva, Palma de Majorka

Marija Živković

Sadržaj – U ovom radu biće ukratko opisani život i studije u Španiji na Majorci na Univerzitetu Balearskih ostrva u okviru programa Erazmus+. Cilj ovog rada je da studente upozna sa glavnim odlikama ovog područja, kao i da pokaže pozitivne aspekte studiranja u inostranstvu.

I. UVOD

Erazmus+ program predstavlja program Evropske unije i obezbeđuje finansiranje različitih programa za saradnju (obrazovanje, mladi i sport). Osnovni prioriteti ovog programa jesu da podrži i unapredi razvoj različitih nivoa obrazovanja, osnaži i ojača vezu između formalnog, neformalnog i informalnog učenja i poveže zemlje članice u definisanju obrazovnih politika. Srbija je do 2019. godine spadala u grupu partnerskih zemalja koje mogu da učestvuju u jednom delu programa, dok je od Opšteg poziva za 2019. godinu stekla status programske zemlje (kao i zemlje Evropske unije, Severna Makedonija, Island, Lihtenštajn, Norveška, Turska) [1].

Elektronski fakultet u Nišu sarađuje sa brojnim fakultetima sa kojima ostvaruje različite programe razmene za studente osnovnih, master i doktorskih studija, kao i nastavno osoblje. Jedan od njih je i Univerzitet Balearskih ostrva na Majorci u Španiji koji će biti opisan u nastavku.

II. UNIVERZITET

Univerzitet Balearskih ostrva se nalazi na Majorci, najvećem od četiri glavna ostrva (pored Ibice, Menorke i Formentere). Nalazi se oko 8km severno od grada Palme, glavnog grada ovog ostrva i do njega se može doći autobusom ili metroom.

Univerzitet se sastoji od 10 fakulteta, kao što se može videti na slici 1. Fakulteti i škole su podeljeni na autonomne katedre koje su podeljene u predmetna područja. Svaki fakultet vodi dekan, a svaki odsek direktor.

Na kampusu su takođe i Institut za nauku o obrazovanju, obuku nastavnika, Služba kulturnih aktivnosti koja organizuje Otvoreni univerzitet, studentski dom (Bartomeu Rossello-Porcel) i restoran [2].

Marija Živković je student na modulu Elektronika i mikrosistemi na Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marija.zivkovich@elfak.rs.

Erazmus+ program je tako osmišljen da student prilikom prijave sastavlja spisak predmeta koje bi pohađao na željenom fakultetu. Među tim predmetima poželjno je imati predmete koji se poklapaju sa onima koje bi student inače pohađao na svom fakultetu u tom ili nekom drugom semestru (zavisno od fakulteta, moguće je odabrati predmete koje je student već slušao na svom fakultetu ali nije položio, ili predmete koje bi slušao na nekoj višoj godini a koji bi mu bili priznati kasnije). Pri prijavi na sam Erazmus+ program potrebno je prvenstveno informisati se u vezi predmeta na nivou studija na kome je student trenutno.



Slika 1. Mapa Univerziteta Balearskih ostrva [3].

Katedre i profesori koji sarađuju sa Elektronskim fakultetom imaju sličan princip rada kao na Elektronskom fakultetu u Nišu, a redovni studenti i studenti na razmeni zajedno pohađaju predavanja na engleskom jeziku. Učionice i laboratorije su opremljene savremenom opremom, i dostupne su studentima pri izradi različitih projekata i laboratorijskih vežbi. Profesori su jako ljubazni i profesionalni, dostupni studentima za različite projekte i konsultacije.

Metro stanica se nalazi na južnoj strani glavnog kampusa Univerziteta na Balearskim ostrvima. Stanica je otvorena 25. aprila 2007. i metro predstavlja najbrži i najpraktičniji način prevoza. Pored metroa, dostupne su i dve autobuske linije, od kojih jedna vodi do centra glavnog grada (Slika 2) [4].



Slika 2. UIB metro stanica [4].

Na zvaničnom sajtu fakulteta u sekciji za studente postoje sve potrebne informacije u vezi studija i predmeta za svaki fakultet na univerzitetu. Najbitnije je da student obrati pažnju na to koji je jezik na kome se predavanja održavaju, jer u slučaju našeg fakulteta jedino može odabrati predmete na engleskom jeziku (postoje i predmeti i na španskom i katalonskom jeziku).

Ocene su u opsegu od 5-10 (gde je 5 najniža prolazna ocena, a 10 najviša ocena). Studenti vrlo često rade projekte, predavanja su negde obavezna, negde ne, i takođe iz nekih predmeta postoje i predispitne obaveze koje je potrebno ispuniti da bi student imao uslov za polaganje ispita. Profesori su relativno rigorozni što se tiče ispita, a studenti projekte uglavnom rade u manjim grupama ili samostalno. Studenti takođe imaju konstantan pristup laboratorijama, pa mogu i van laboratorijskih vežbi da koriste materijal i merne instrumente.

Univerzitet ima i studentski dom u kampusu. Ukoliko postoji prilika, povoljno je odabrati tu opciju stanovanja na ostrvu za vreme studija. To zavisi prvenstveno od toga za koji semestar se student prijavljuje (ukoliko je to jesenji semestar, ima mogućnost prijave za dom, dok je kod prolećnog semestra to malo teže – sve sobe su uglavnom već zauzete). Može se izabrati jednokrevetna ili zajednička soba. U svakom slučaju, postoji privatno kupatilo, telefon, centralno grejanje, internet veza i pristup „Campus Esport“-u, univerzitetskim sportskim objektima. Može se birati između tri različite vrste pansiona: noćenje sa doručkom, polupansion i puni pansion. Ukoliko pak student želi da živi u sobi u centru grada, postoji veliki asortiman soba i stanova, s obzirom na posećenost ostrva i veliki broj studenata koji deo svojih studija provedu tu.

III. MAJORKA

Majorka je jedno od četiri Balearska ostrva i nalazi se istočno od glavnog kopna Španije (Slika 3). Španija je po državnom uređenju monarhija, preciznije kraljevina. Kroz vekove je ostrvo bilo okupirano od strane različitih vladavina, i zato se i danas može osetiti uticaj Nemačke i Velike Britanije na različitim delovima ostrva. Glavni grad je Palma de Majorka, koja se nalazi na jugozapadnom delu ostrva, dok ukupno postoji 54 opština. Na slici 4 se može videti katedrala koja je potpuno izgrađena 1601. godine u gotskom stilu.



Slika 3. Balearska ostrva [5].



Slika 4. Katedrala na Majorci [6].

Na slici 5 se može videti i čuveni Belver zamak, koji se nalazi nedaleko od glavnog grada Palma de Majorka, sa koga se može videti najlepši pogled na grad. Još jedna znamenitost po kojoj je ostvro poznato je i Drak pećina, koja se nalazi na istočnom delu ostrva, do koje se može doći i gradskim prevozom (Slika 6).



Slika 5. Belver zamak [7].



Slika 6. Drak pećina [8].

Ostvro je takođe poznato po teniseru Rafaelu Nadalu, čija se sportska akademija, zajedno sa muzejem nalazi na istočnom delu ostrva (Slika 7). Pored ovih, postoji još mnoštvo atrakcija koje se mogu videti i posetiti.



Slika 7. Rafa Nadal sportski centar [9].

Pored glavnog grada, kao što je već spomenuto, postoje još 53 opštine. Svaka je drugačija, a u nekim delovima većinsko stanovništvo čak čine državljani Nemačke ili Velike Britanije. Veća i poznatija mesta su Alkudia, Kala Major, Magaluf, Manakor, Soljer, Valdemosa, El Arenal itd. (Slike 8 i 9).



Slika 8. El Arenal, Palma de Majorka [10].



Slika 9.Voz u Soljeru, Palma de Majorka [11].

IV. AKTIVNOSTI I STUDENTSKI ŽIVOT

Ostrvo Majorka je poznato kao turističko mesto sa velikim brojem aktivnosti u različitim oblastima. Tokom cele godine održavaju se muzička dešavanja, okupljanja, proslave praznika. Pored toga, postoji niz plaža, gradskih i divljih, brdovitih i nenaseljenih prirodnih obasti na ostrvu, kao što se može videti na slici 10.

Studentski život na Majorci je vrlo aktivan, postoji veliki broj događaja, radionica, proslava i žurki. U zavisnosti od interesovanja, student se može baviti i različitim sportovima i vannastavnim aktivnostima. Veliki broj aktivnosti se može naći na različitim društvenim mrežama (*facebook.com, couchsurfing.org, meetup.com…*)

Studenti takođe, kao privremeni stanovnici ostrva, imaju pravo na popuste za letove koji povezuju ostrvo sa ostatkom Španije, pa se po vrlo povoljnoj ceni mogu naći letovi do atraktivnih destinacija koje su vredne posete (npr. Madrid, Barselona, Tenerife…): Na ostrvu su zvanični jezici španski i katalonski, a Majorka pripada Kataloniji, autonomnoj pokrajini Španije [12].



Slika 10. Divlja plaža na ostrvu.

V. ZAKLJUČAK

Studentske razmene koje se odvijaju na Elektronskom fakultetu u Nišu, ali i na drugim fakultetima, imaju cilj da studente povežu sa Evropom i drugim kontinentima, razviju njihove sposobnosti i pruže im priliku da unaprede sebe u svakom smislu. Studenti pritom vežbaju engleski, ali vrlo često i neki drugi strani jezik, stiču nova poznanstva i razvijaju svoje meke veštine (*soft skills*).

Upravo zbog toga je ovaj vid studiranja visoko cenjen, kako u inostranstvu tako i kod nas, i studenti imaju šansu da kao državljani zemlje van Evropske Unije, upravo dobiju priliku da deo svojih studija provedu u njoj. Ovakvo iskustvo je vrlo pozitivno, i teško je objasniti ukoliko se student sam ne oproba u tome. Španija je pritom dobra opcija iz razloga što je mentalitet ljudi vrlo sličan srpskom, temperature su odlične u toku godine, hrana izvrsna, a studije vrlo kvalitetne.

LITERATURA

[1] https://erasmusplus.rs/
[2] https://en.wikipedia.org/wiki/University_of_the_Balearic_Islands
[3] http://www.uibcongres.org/jafaritribute/comollegar.en.html
[4] https://en.wikipedia.org/wiki/UIB_metro_station
[5] https://www.worldatlas.com/webimage/countrys/europe/balearic.html
[6] https://www.mallorca.com/en/mallorca/la-seu-cathedral-palma
[7] http://www.spainisculture.com/en/monumentos/mallorca/castillo_de_bellver.html
[8] https://www.tripadvisor.co.uk/Attraction_Review-g580311-d315498-Reviews-Cuevas_del_Drach-Porto_Cristo_Majorca_Balearic_Islands.html
[9] https://en.as.com/en/2018/10/10/other_sports/1539175695_278353.html
[10] https://www.abc-mallorca.com/el-arenal/
[11] https://www.abc-mallorca.com/soller-train-mallorca/
[12] https://sr.wikipedia.org/sr-ec/%D0%9C%D0%B0%D1%98%D0%BE%D1%80%D0%BA%D0%B0

Razvoj interaktivne aplikacije „blok igre“ u okviru PSoC razvojnog okruženja

Nikola Milenković

Sadržaj - Cilj ovog rada je da se u PSoC 4 okruženju realizuje takozvana „blok igra“, koja je bila jako popularna u prvim mobilnim telefonima koji su pružali funkciju igranja igara, kao i u konzolama za igranje igara. Kroz rad će detaljno biti opisan sam mikrokontroler, njegove funkcije i komponente koje se koriste u ovom projektu.

I. UVOD

Krajem 20. veka, posedovanje mobilnih telefona postalo je jako popularno u industrijalizovanom svetu usled uspostavljanja odgovarajućih industrijskih standarda i naglog pada cena mobilnih telefona. Kao rezultat ove eksplozije, tehnološki napredak proizvođača mobilnih uređaja postao je brz. Sa ovim tehnološkim napretkom, igre za mobilne telefone takođe su postale sve sofisticiranije, iskorištavajući prednosti eksponencijalnih poboljšanja prikaza, obrade, skladištenja, interfejsa, propusnosti mreže i funkcionalnosti operativnog sistema. Kako bi se podsetili starih vremena, u ovom radu će biti opisana realizacija jedne takve igre (blok igre) pomoću PSoC (*Programmable System on Chip*) mikrokontrolera i LCD displeja koji se povezuje na okruženje. PSoC koristi SPI (*Serial Peripheral Interface*) komunikaciju sa displejom. Cilj rada je da se prikaže deo mogućnosti implementacije PSoC čipa pri kreiranju interaktivnih aplikacija, a kroz primer “blok igre” (slika 1). Projekat daje uvid u komunikaciju između kapacitivnog senzora i TFT LCD displeja preko ugrađenih komponenti PSoC čipa.



Slika 1. Primer jedne blok igre.

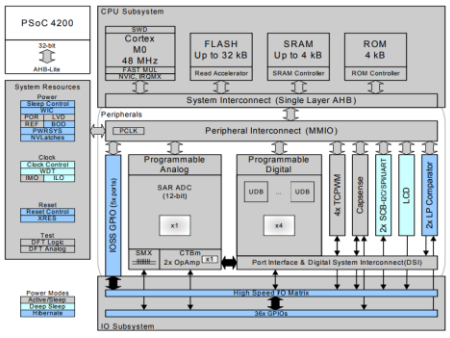
N. Milenković je student master studija na modulu Elektronika i miksrosistemi na Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikolakpd@elfak.rs

II. PSOC RAZVOJNO OKRUŽENJE

PSoC (programabilni sistem na čipu) je familija mikrokontrolera (MCU) kompanije *Cypress Semiconductor*. Ovi čipovi sadrže procesorsko (CPU) jezgro i nizove podesivih integrisanih analognih i digitalnih perifernih uređaja sa mešovitim signalima (*mixed-signals*). Podesivi blokovi u PSoC-u čine najveću razliku između tog okruženja i drugih mikrokontrolera [1].

PSoC 4 je skalabilna i rekonfigurabilna platforma ugrađenog (*embedded*) sistema sa mešovitim signalima za ARM® Cortex™ - M0 procesorom. Kombinuje programabilne i podesive analogne i digitalne blokove sa fleksibilnim automatskim usmeravanjem (*routing*).

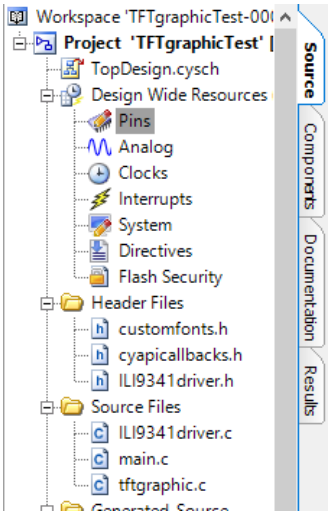
Cortex-M0 procesor u PSoC 4200 je deo 32-bitnog mikrokontrolerskog podsistema, koji je optimizovan za operacije male potrošnje. Uglavnom koristi 16-bitne instrukcije i izvršava skup instrukcija Thumb-2. Fleš memorija spada u grupu “*nonvolatile*” memorija (one čuvaju svoj sadržaj nakon nestanka napajanja). SRAM (*Static Random Access Memory*) sadrži podatke tokom izvršavanja programa. Njen sadržaj se može često menjati pri čemu se brisanje i upis izvršavaju veoma brzo. SROM (*Supervised Read Only Memory*) sadrži pokretačke i konfiguracijske rutine. U njemu se podaci samo jednom upisuju, i nakon toga koriste (čitaju). SROM čuva podatke u odsustvu napajanja. Sistem za napajanje daje sigurnost da su naponski nivoi tačni za svaki režim rada. PSoC 4200 sistem takta odgovoran je za obezbeđivanje takta za sve podsisteme kojima je potreban takt. PSoC čip takođe sadrži i konfigurabilne analogne i digitalne blokove. Njihovim podešavanjem moguće je dobite razne funkcije u okviru samog okruženja. Pored toga PSoC sadrži i fiksne digitalne funkcije: I2C, UART i SPI i 36 GPIO pinova.



Slika 2. Blok dijagram PSoC 4200 porodice čipova.

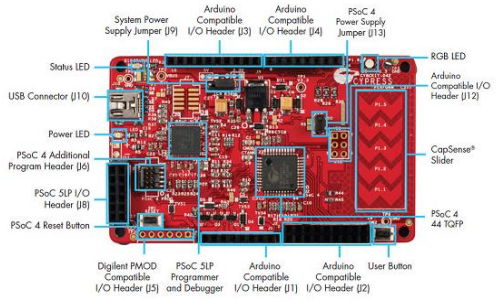
Softver koji dolazi uz PSoC razvojno okruženje naziva se PSoC Creator [2]. Trenutna verzija ovog softvera, u kojoj je rađen ovaj rad, je verzija 4.2. Softver se može naći na sajtu Cypress-a i predstavlja njihov zvanični proizvod. Ovaj razvojni alat omogućava konfigurisanje analognih i digitalnih komponenti sistema. Softver dolazi sa integrisanim Keil C51 kompajlerom. Pisanje programa može se realizovati u programskom jeziku C ili asembleru.

Najbitnija opcija koja se bira jeste verzija čipa koji se koristi. Verzija čipa i model PSoC okruženja može se pročitati na samoj pločici okruženja. U ovom radu korišćen je **PSOC 4200 CY8CAXI-483** čip sa Cortex-M0 procesorom [3]. Po kreiranju novog projekta, u levom delu ekrana se otvara prozor koji se naziva *workspace explorer*. U ovom prozoru se nalaze svi glavni fajlovi projekta, od onih koje program sam generiše do fajlova koje kreira ili učitava korisnik. Najbitniji fajlovi i folderi koji se nalaze u *workspace exploreru* (slika 3) su: TopDesign.cysch, Pins, Header files, Source files, Main.c.



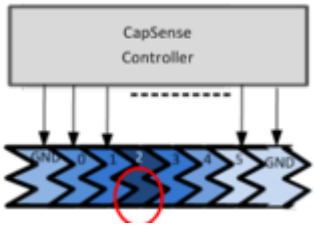
Slika 3. Izgled Workspace explorer-a.

Za realizaciju ovog projekta korišćen je **CY8CKIT-042 PSoC4** Pioneer Kit [3]. Razvojno okruženje PSoC4 Pioneer Kit je namenjeno za razvoj aplikacija na čipovima iz PSoC 4 familije. Izgled ovog okruženja sa njegovim delovima prikazan je na slici 4.



Slika 4. PSoC 4 razvojno okruženje.

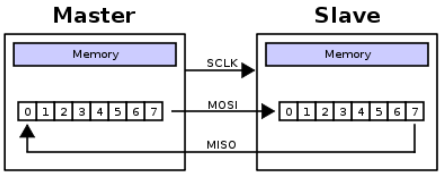
U elektrotehnici, kapacitivno očitavanje vrednosti (CapSense [4]) je tehnologija zasnovana na detektovanju promene kapacitivnosti. Kapacitivno senzorisanje na dodir promenilo je izgled industrijskog dizajna u proizvodima kao što su mobilni telefoni, računari, automobili i uređaji potrošačke elektronike i bele tehnike. Cypress-ovi CapSense kontroleri koriste promene kapaciteta kako bi otkrili prisustvo prsta na tasteru ili blizu njegove površine. Kapacitativni *pad*-ovi senzorske ploče formirani su na PCB-u. Softver je ključna komponenta CapSense sistema. Obrađuje podatke i donosi logične odluke iz njih. Postoji više CapSense *widget*-a: tasteri, klizači (slika 5), radijalni klizači, *Touchscreen* i senzori blizine. U ovom radu korišćen je kapacitivni linearni klizač (slika 3).



Slika 5. CapSense linearni klizač.

U ovom projektu kao izlazna komponenta korišćen je TFT LCD displej veličine 2.8 inča sa rezolucijom od 240x320 piksela koji je ugrađen na kontrolnu pločicu proizvođača ADAFRUIT [5]. Verzija čipa je jako važna za implementaciju displeja. Ovaj displej koristi SPI (*Serial Peripheral Interface*) komunikaciju sa okruženjem (mikrokontrolerom).

Serijski periferni interfejs (SPI) je serijski komunikacioni interfejs koja se koristi za komunikaciju na kratkoj udaljenosti, pre svega u ugrađenim (EMBEDED) sistemima. SPI uređaji komuniciraju koristeći *master-slave* arhitekturu sa jednim *master*-om. Glavni uređaj pokreće proces čitanja i pisanja. Tokom svakog ciklusa SPI takta dolazi do prenosa podataka u punom dupleksu (*full-duplex*) (slika 6). *Master* šalje bit na MOSI liniju i *slave* ga čita, dok *slave* šalje bit na MISO liniju i *master* ga čita. Taj niz se održava čak i kada je predviđen samo jednosmerni prenos podataka.



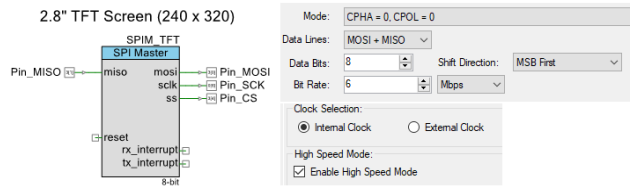
Slika 6. SPI Master-Slave komunikacija.

III. KONFIGURACIJA SISTEMA

U konfiguraciju sistema spada i hardversko i softversko podešavanje. Za okruženje PSoC-a se povezuje već pomenuti *shield* TFT displej, kome se pinovi

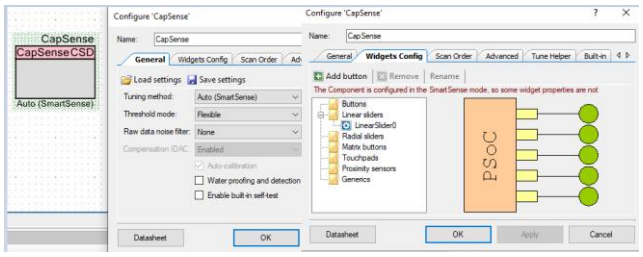
podudaraju u potpunosti sa ženskim pinovima na okruženju. Okruženje se zatim preko USB porta povezuje sa računarom i sav preostali rad se izvršava u računaru. Tu spada podešavanje PSoC okruženja sa svim njegovim komponentama koje se koriste, a zatim i kreiranje programa kako bi se pokrenuo povezani displej. Rad displeja sa okruženjem se uspostavlja preko četiri pina koji služe za SPI komunikaciju. Pored tih pinova koriste se i pinovi za napajanje, masu i kontrolu osvetljenja displeja.

Za ispravan rad displeja sa bilo kojim okruženjem potrebne su odgovarajuće biblioteke. U PSoC Creator-u postoje *header* i *seource* fajl koji sadrže potrebne instrukcije za pokretanje displeja. Pored tih osnovnih funkcija, nalaze se i funkcije koje služe za iscrtavanje raznih prostijih i složenijih geometrijskih elemenata po displeju. Još jedna bitna funkcija jeste i SPI komunikacija, preko koje se uspostavlja veza između displeja i okruženja. Podešavanje ove komunikacije vrši se podešavanjem komponente u šematiku softvera i podešavanjem pinova u delu programa vezanom za to.



Slika 7. Izgled prozora za podešavanje SPI komponente.

Na isti način vrši se i povezivanje i podešavanje CapSense komponente tj. klizača. Ovaj klizač se koristi za komunikaciju između korisnika i okruženja. Korisnik pomeranjem prsta po klizaču direktno kontroliše pomeranje bloka koji služi da odbije loptice u suprotnom smeru displeja. Opcija za inicijalizaciju početne vrednosti CapSense klizača (*tunning*) je postavljena na Auto (Smart Sense). Ovo je opcija koja automatski vrši određivanje i postavljanje svih početnih parametara, bez da korisnik ne mora da vodi računa o svim parametrima komponente.



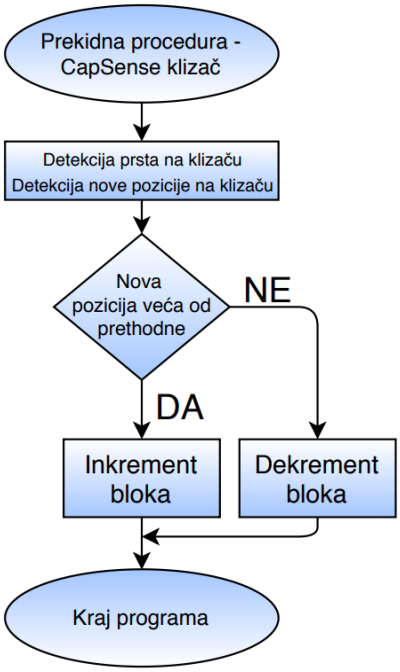
Slika 8. Izgled prozora za podešavanje CapSense komponente.

Po završetku podešavanja svih komponenti, pinova i taktova, sledi pisanje glavnog koda program u *main.c* fajlu. Ovaj fajl sadrži glavnu funkciju koja se po završetku pisanja fajla implementira u mikrokontroler.

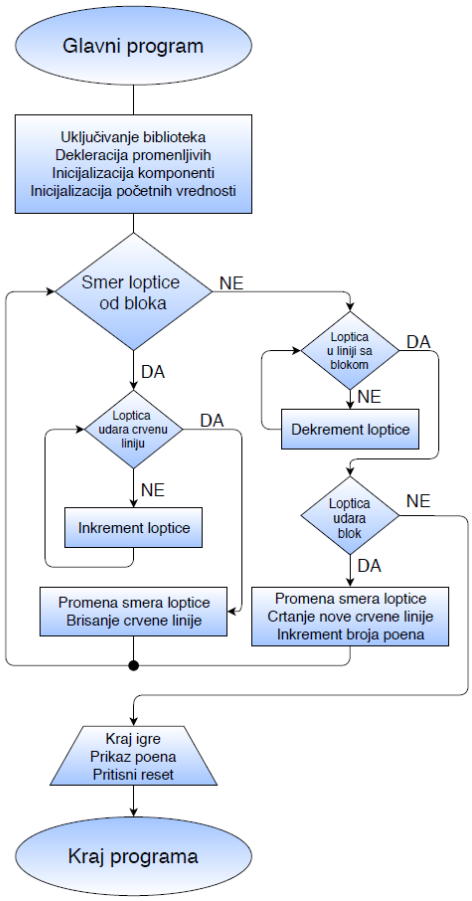
Na samom početku fajla se uključuju sve potrebne biblioteke. Ovde spadaju biblioteke koje predstavljaju osnovu programskog jezika za prepoznavanje funkcija korišćenih iz njih, kao i biblioteke koje su opisane u prethodnom poglavlju koje sadrže funkcije za pokretanje displeja i generisanje geometrijskih oblika. Nakon uključivanja svih biblioteka sledi definisanje i inicijalizacija svih promenljivih koje se koriste u daljem kodu. U ove promenljive spadaju dimenzije bloka, loptice, dimenzije ivica zidova od kojih se loptica odbija (rezolucija displeja), smer kretanja loptice i par drugih promenljivih koje služe kao pomoć pri izvršavanju raznih petlji programa. Ovim je završeno sa početnim delom koda. Sledeća stavka je glavna petlja programa int main(). U ovoj petlji se piše deo programa koji se kompajlira i kog mikrokontroler koristi za izvršavanje funkcija pisanih u njemu. Neke od funkcija koje su zadužene za početnu inicijalizaciju programa:

```
SPIM_TFT_Start();
TFT_Init();
TFT_setRotation(2);
TFT_clearScreen(TFT_BLACK);
CapSense_Start();
CapSense_InitializeAllBaselines();
CapSense_ScanEnabledWidgets();
TFT_fillRect(0,10,50, 15, TFT_WHITE);
```

Na slici 9 i 10 mogu se videti glavni algoritmi programa, koji objašnjavaju način funkcionisanja igre (tok igre). U programu postoje dva glavna dela. Jedan deo je zadužen za detekciju prsta na CapSense klizaču i vrši crtanje bloka u zavisnosti od pomeraja na klizaču, dok je drugi deo zadužen za iscrtavanje loptice i svih parametara i funkcija vezanih za njeno kretanje.



Slika 9. CapSense algoritam.



Slika 10. Algoritam glavnog programa.

Rezolucija displeja je 240x320 piksela, tako da svi elementi koji se kreiraju unutar igre moraju biti ograničeni nekim maksimalnim vrednostima celog broja. Pre samog izvršavanja programa vrši se iscrtavanje bloka u levom delu displeja. Položaj na displeju se posmatra kao tačka u koordinatnom x-y sistemu. X osa predstavlja vertikalni deo displeja (240 piksela) dok y osa predstavlja horizontalni deo veličine od 320 piksela. Blok se iscrtava u gornjem levom uglu sledećom komandom:

```
TFT_fillRect(190,10,240, 15, TFT_WHITE);
```

Nakon iscrtavanja bloka prelazi se na programiranje u glavnoj petlji for(;;). U ovom delu postoje dve glavne petlje koje se konstantno izvršavaju. Ove petlje su predstavljene pomoću dva zasebna algoritma u prethodnom odeljku. Prva petlja je zasnovana na stalnom čitanju CapSense senzora. Mikrokontroler konstantno vrši čitanje nakon početne inicijalizacije, i reaguje tek kada dođe do promene kapacitivnosti na klizaču. To znači da je detektovan prst korisnika. Ovo izvršavanje predstavljeno je sa dvema linijama koda:

```
sliderPosition=CapSense_GetCentroidPos(CapSense_LINEARSLIDER0_
_LS);
if(sliderPosition != NO_FINGER)
```

Kada dođe do detekcije prsta na klizaču dalje se ispituje da li je došlo do promene pozicije centroida (mesto najveće očitane vrednosti). U ovom radu pozicija centroida se izračunava matematičkom formulom i ima vrednost od 0 do 100. Ukoliko dođe do promene pozicije, ispituje se da li je ta nova pozicija veća ili manja od prethodne pozicije. U zavisnosti od ovog uslova, postoje dve mogućnosti: jedna je iscrtavanje pomeranja bloka ka koordinatnom početku, dok je druga pomeranje bloka u suprotnom smeru.

Druga petlja je zadužena za pomeranje loptice po displeju i ispitivanju uslova koji su potrebni za pravilan rad igre. I ovde se kao i u slučaju crtanja bloka pre početka postavljaju maksimalne vrednosti promenljivih koje predstavljaju granicu zida za odbijanje loptice. Te vrednosti su vrednosti same rezolucije displeja, tačnije vrednosti piksela.

Crtanje loptice vrši se funkcijom iz biblioteke vezane za čip displeja. Funkcija je crtanje kruga koji nije ispunjen iznutra, sa parametrima x, y za centar i r za prečnik kruga.

```
TFT_drawCircle ( ball_x, ball_y, r1, TFT_WHITE);
```

Ova funkcija se izvršava konstantno i postoje četiri verzije ove funkcije. Za svako kretanje loptice u sva 4 uslova, dodaje se još jedna funkcija koja je zadužena za brisanje prethodne pozicije loptice. Ovo se mora uraditi jer displej nakon iscrtavanja određenih piksela njih ne briše automatski iscrtavanjem novih. Zato se dodaje funkcija za crtanje crnog kruga koji je ispunjen iznutra takođe crnom bojom. Ovako se stiče utisak da se loptica kreće i da je svuda oko nje pozadina nepromenjena (crna boja).

```
TFT_fillCircle ( ball_x-1, ball_y-1, r2, TFT_BLACK);
```

U opisu parametara funkcije se vidi da se krug crta sa zakašnjenjem od jednog piksela, dok je prečnik r2 za jedan piksel takođe veći od prečnika loptice.

Kretanje u svakom smeru je određeno pomoću dve promenljive koje imaju vrednosti 0 ili 1. Te promenljive (ball_x_direction i ball_y_direction) se koriste u glavnim petljama za ispitivanje uslova igre i kretanja loptice.

Manje zahtevan uslov je kretanje duž x ose. Jedina stvar koja se ispituje jeste da li je loptica došla do maksimalnih granica za x osu. Ukoliko jeste vrši se promena promenljive za kretanje duž x ose i menja se smer kretanja loptice. Kretanje loptice duž y ose takođe se odvija u dva smera, ka bloku ili od bloka. Ukoliko se loptica kreće od bloka ka desnoj ivici displeja, konstantno se ispituje uslov da li je loptica došla do maksimalne vrednosti za y osu. Ukoliko nije, vrši se inkrement vrednosti y u funkciji za crtanje loptice. U ovom delu je sledeća petlja koja ispituje vrednosti promenljivih koje su zadužene za poziciju pomeranja bloka i vrednosti promenljivih koje predstavljaju centar kruga kretanja loptice. Ukoliko se centar kruga loptice, u trenutku kada ona dođe do bloka duž y ose, nalazi između dve promenljive koje predstavljaju donju i gornju ivicu bloka, tada se smatra da

je korisnik uspešno odbio lopticu u suprotnom smeru nazad ka desnoj strani i novoj crvenoj liniji. U tom trenutku se vrši promena smera kretanja loptice, i kroz posebnu promenljivu vrši povećanje vrednosti za 1. Ta promenljiva služi za pamćenje rezultata koji korisnik postigne tokom jedne igre, i prikazuje se na displeju kada se igra završi. Ukoliko se centar loptice nalazi izvan vrednosti granica bloka, smatra se da korisnik nije uspeo da uspešno odbije lopticu i igra se završava. U tom trenutku se vrši brisanje displeja a zatim i ispisivanje teksta na displeju da je došlo do kraja igre, i vrši se prikaz rezultata.

IV. Prikaz rezultata

Na slici 11 prikazana su 3 dela koda (igre) koja se pojavljuju na displeju nakon uključivanja mikrokontrolera i predstavljaju tok igre. Prva slika se zadržava 3 sekunde, zatim počinje igra i kada dođe do kraja igre pojavljuje se treća slika na kojoj je prikazan rezultat.



Slika 11. Prikaz rezultata igre na displeju.

V. Zaključak

U ovom radu prikazano je kako se može napraviti jedna igra samo pomoću PSoC okruženja i TFT LCD displeja koji je služio kao izlazna komponenta. Kompletan način realizacije projekta je opisan od početka, i jednom kada se uspostavi konekcija sa displejom, praktično je sve ostalo prepušteno mašti korisnika. Igra realizovana u ovom projektu je jednostavna. Njeno projektovanje uglavnom zahteva logično razmišljanje u koordinatnom sistemu, i poznavanje petlji tokom programiranja u programskom jeziku.

Zahvalnost

Zahvaljujem se prof. dr Aneti Prijić, prof. dr Danijelu Dankoviću i asistentu Milošu Marjanoviću, sa Elektronskog fakulteta u Nišu na pruženoj pomoći tokom izrade ovog rada.

Literatura

[1] Getting Started with PSoC 4, Authors: Ranjith M, Nidhin M S http://www.cypress.com/go/AN79953
[2] PSoC® 4: PSoC 4200 CY8C4245AXI-483 https://www.cypress.com/part/cy8c4245axi-483
[3] PSoC Creator https://www.cypress.com/products/psoc-creator-integrated-design-environment-ide
[4] CapSense® Controllers https://www.cypress.com/products/capsense-controllers
[5] a-Si TFT LCD Single Chip Driver 240RGBx320 Resolution and 262K color, ILI TECHNOLOGY CORP. Hsinchu Country 302 Taiwan R.O.C. https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf

Uređaj za demonstraciju Lencovog zakona

Iva Avramović

Sadržaj – Cilj projekta je demonstacija Lencovog zakona na principu merenja vremena prolaska magneta kroz provodnu bakarnu cev.

I. UVOD

Lencov zakon nazvan je po fizičaru Hajnrihu Friedrihu Emilu Lencu koji je primetio da je pravac struje koja se indukuje u provodniku pri promenljivom magnetnom polju takav da se protivi magnetnom polju koje ga je izazvalo. Efekat Lencovog zakona primenjuje se kod električnih brojila, indukcionih bezkontaktnih kočnica, a u ovom radu izložena je maketa koja se može koristiti u edukativne svrhe za potrebe demonstracije Lencovog zakona u nastavi iz fizike i osnova elektrotehnike.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Princip rada uređaja (slika 1) baziran je merenju vremena koje magnet provede putujući kroz izolacionu i bakarnu cev. Razlika u vremenu koje magnet provede u dve cevi posledica je indukcionog kočenja koje nastaje kao posledica Lencovog zakona.



Slika 1. Uređaj za dokazivanje Lencovog zakona.

Uređaj se sasatoji iz hardverskog dela koga čine magnet, izolaciona i bakarna cev, mikrokontroler Arduino Nano, dva reed relea i LCD displeja 2x16 karaktera.

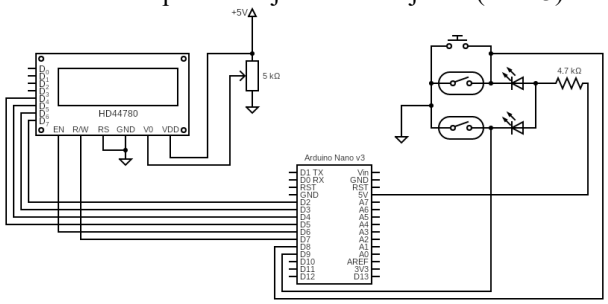
Iva Avramović, učenica četvrtog razreda Tehničke škole u Loznici. E-mail: ivaavramovic001@gmail.com

Merenje vremena koje magnet provede u cevi obavlja se pomoću dva reed relea. Reed rele (slika 2) je prekidač koji reaguje na magnetno polje. Kada se reed rele nađe u magnetnom polju tada dolazi do promene stanja njegovih kontakata. U ovom uređaju korišćeni su reed relei koji imaju radni kontakt, odnosno čiji se kontakti zatvaraju usled prisustva magnetnog polja.



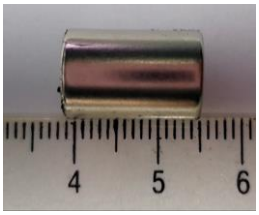
Slika 2. Reed rele.

Reed relei smešteni su na vrhu i na kraju cevi. Kada magnet prođe kraj reed relea tada dolazi do promene stanja njegovog kontakta čime se mikrokontroleru šalje signal da je magnet upravo prošao kraj reed relea, odnosno da je magnet ušao i izašao iz cevi. Mikrokontroler meri vreme između signala sa dva reed relea i to vreme ispisuje na LCD displeju (slika 4). Položaj reed relea na početku i kraju obe cevi omogućava da se sa istim hardverom meri vreme koje magnet provede i u jednoj i u drugoj cevi. Električna šema prikazana je na sledećoj slici (slika 3).



Slika 3. Električna šema.

Upotrebljene cevi su dužine 600 mm, unutrašnjeg prečnika 12 mm i spoljašnjeg prčnika 15 mm. Upotrebljeni magnet (slika 5.) je oblika valjka visine 15 mm i prečnika 10 mm.



Slika 5. Magnet.

Reed relei vezani su u kolo sa pull-up otpornicima. Signali sa reed relea dovode se na mikrokontroler Arduino NANO, ATMEGA 328. Kada se zbog prisustva magneta zatvore kontakti reed relea tada na ulazne pinove stigne stanje logičke nule. LCD displej komunicira sa mikrokontrolerom preko 4-bitnog paralelnog interfejsa.

Uređaj se napaja baterijski, sa 4 baterije po 1,5V.



Slika 4. Prikaz rezultata merenja.

Program koji obavlja merenje vremena između dva signala:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup()
{     lcd.begin(16, 2);
      lcd.clear();
      Serial.begin(9600);
      pinMode(8, INPUT);
      digitalWrite(8, HIGH);
      pinMode(9, INPUT);
      digitalWrite(9, HIGH);}
      double i = 0;
      double a = millis();
      double c ;
void loop()
{     lcd.clear();
      lcd.print("press start");
      delay(100);
      if(digitalRead(8) == LOW)
{     lcd.clear();
      a = millis();
      while(digitalRead(9) == HIGH)
      {    c = millis();
      i = (c - a) / 1000;
      lcd.print(i);
      lcd.setCursor(7,0);
      lcd.print("Sec's");
      lcd.setCursor(0,0);
      Serial.println(c);
      Serial.println(a);
      Serial.println(i);
      Serial.println(".....");
      delay(100);}
      if(digitalRead(9) == LOW)
{     while(digitalRead(8) == HIGH)
{     lcd.setCursor(0,0);
      lcd.print(i);
      lcd.setCursor(11,0);
      lcd.print("");
      lcd.setCursor(0,0);
          delay(100);}}}}
```

III. ZAKLJUČAK

Sa datim gabaritima cevi i jačinom magneta postignuti su sledeći rezultati:

- prosečno vreme koje je magnet u slobodnom padu proveo kroz neprovodnu cev iznosilo je 0,11 s,
- prosečno vreme koje je magnet proveo u bakarnoj cevi iznosilo je 4,64 s.

Na osnovu dobijenih rezultata Uređaj za demonstraciju Lencovog zakona predstavlja sistem za demonstraciju Lencovog zakona. Vreme koje se može izmeriti sa ovim uređajem direktno potvrđuje efekat indukcionog kočenja. odnosno postojenje/utrđivanje Lencovog zakona. U zavisnosti od jačine magneta ili provodnosti cevi, koja zavisi od materijala cevi, može se izmeriti uticaj Lencovog zakona, odnosno dobiti drugi rezultati merenja što ovaj uređaj omogućava.

Kako je pored teoretskog dela demonstrirana i eksperimentalno izvršeno merenje to ovaj uređaj čini primamljivim budućim učenicima u nastavi fizike i osnova elektrotehnike.

ZAHVALNICA

Zahvalnost na realizaciji ovog istraživačkog projekta uputila bih Tehničkoj školi u Loznici i mom mentoru, nastavniku Tomislavu Mijiću, dipl. inž. el. Posebnu zahvalnost uputila bih firmi Alfa Elektrik i mojoj porodici na finansijskoj i moralnoj podršci.

LITERATURA

[1] Reed rele, https://en.wikipedia.org/wiki/Reed_relay
[2] Arduino Software, https://www.arduino.cc
[3] Arduino timer https://www.how2electronics.com/stopwatch-using-arduino-lcd/
[3] Arduino timer https://www.how2electronics.com/stopwatch-using-arduino-lcd/
[4] Lencov zakon https://sh.wikipedia.org/wiki/Lencov_zakonS. Dimitrijev
[3] Arduino timer https://www.how2electronics.com/stopwatch-using-arduino-lcd/
[4] Lencov zakon https://sh.wikipedia.org/wiki/Lencov_zakonS. Dimitrijev
[5] Lencov zakon https://www.automatika.rs/baza-znanja/elektronika/sta-je-to-lencovo-pravilo.html

Realizacija gitarskog vakuumskog pretpojačala

Danilo Đorđević, Nikola Milenković

Sadržaj - U ovom radu dat je šematski prikaz gitarskog pretpojačala, koje ima ulogu da pojača gitarski zvuk i da ga daljim pojačanjem odvede u overdrive. Ovom pretpojačalu je dodata pasivna tonska kontrola i ono je realizovano na štampanoj ploči.

I. UVOD

U današnje vreme vakuumske cevi [1] su gotovo potpuno zamenjene tranzistorima. Njihovo malo pojačanje i prevelika potrošnja zbog grejača dovela je do njihovog potiskivanja iz upotrebe. Međutim, u nekim oblastima elektronike, posebno audio elektronike, vakuumske cevi su i danas nezamenjive. Zbog svog karakterističnog oblikovanja i pojačanja zvuka, najbolja današnja pojačala i pretpojačala se izrađuju u cevnoj tehnici. Početkom 1940-ih godina sa pojavom prvih električnih gitara pojavila su se i prva pojačala sa vakuumskim cevima (slika 1). Ova pojačala su bila prostih dizajna, obično su imala 2-3 cevi i nekoliko pasivnih komponenata vezanih "point to point" tehnikom. Kada su gitaristi shvatili da dolazi do distorzije tona pojačavanjem zvuka na ovim pojačalima (vakuumske cevi odlaze u overdrive), došlo je do novih dizajna pojačala koja sadrže nekoliko pretpojačavačkih (4-10), i nekoliko pojačavačkih cevi.



Slika 1. Mesa Boogie Dual Rectifier (levo) i Marshall Plexi 1967 (desno).

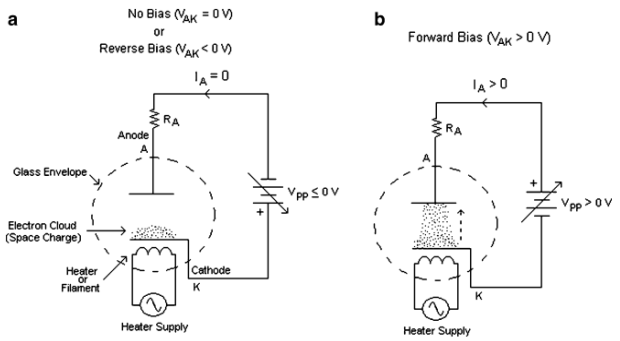
II. PRINCIP RADA VAKUUMSKE CEVI

U pretpojačalima se koriste vakuumske cevi koje se nazivaju triode. Triode imaju tri elektrode: katodu, anodu i rešetku.

D. Đorđević i N. Milenković su studenti master studija na smeru Elektronika i miksrosistemi na Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikolakpd@elfak.rs, danilo.djordjevic@elfak.rs

U jednom staklenom kućištu se obično nalaze po dve triode i neke od ovih trioda nose oznaku 12AU7, 12AX7, 12AT7, 6J5, 6SC7 itd. Ove cevi imaju 9 izvoda od kojih su 3 izvoda za grejač, 3 izvoda za prvu triodu i 3 izvoda za drugu triodu. Radi lakšeg razumevanja kako radi trioda potrebno je prvo razumeti kako radi vakuumska dioda.

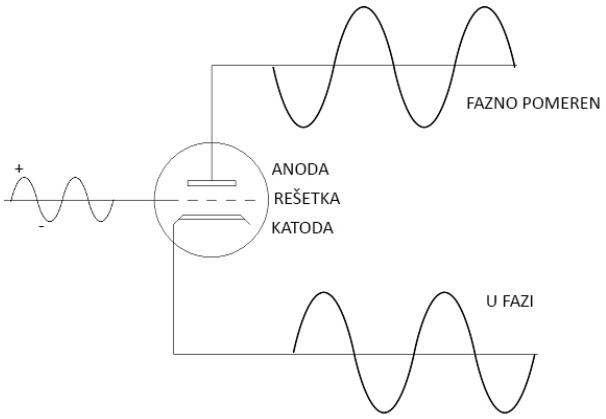
Vakuumske cevi su komponente koje rade na principu termoelektronske emisije. Njihov rad se zasniva na emisiji slobodnih elektrona sa zagrejanog metalnog emitora. Na slici 2, je prikazana struktura vakuumske diode. Katoda se greje direktno ili indirektno pomoću grejača koji se nalazi ispod nje. Grejač se napaja niskim naponom, ali je potrebno oko 500 mA do 3A struje u zavisnosti od veličine katode i maksimalne struje anode koju cev može da izdrži. Kada grejač zagreje katodu, koja je obložena materijalom koji poboljšava emisivnost elektrona sa površine, termalna energija će dovesti do stvaranja oblaka slobodnih elektrona koji okružuju katodu. Ovaj oblak elektrona se naziva oblast prostornog naelektrisanja. Kod vakuumske cevi sav vazduh je uklonjen iz staklenog kućišta. Ovaj vakuum sprečava grejač da pregori i sprečava oksidaciju anode i katode. Ukoliko postoji neki gas unutar cevi, elektroni koji putuju kroz cev jonizovaće molekule gasa i smanjiće se efikasnost diode. Zbog ovoga će vakuumska cev imati curenje, što će rezultovati emitovanjem blage plave svetlosti dok radi [2] .



Slika 2. Princip rada vakuumske diode. Inverzna polarizacija (a) Direktna polarizacija (b).

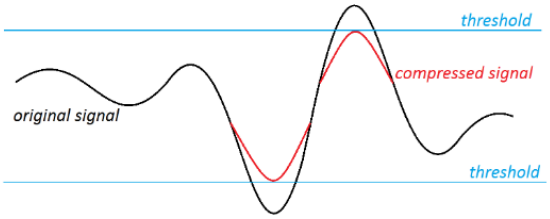
Kada se dovede negativan napon na anodu, a pozitivan na katodu ($V_{ak} < 0$ V), dioda je negativno polarisana. Anoda odbija elektrone u oblaku i struja ne teče. Međutim, kada se na anodi uspostavi veći napon u odnosu na katodu ($V_{ak} > 0$ V), dioda je direktno polarisana. Ova direktna polarizacija dovodi da elektroni koji okružuju katodu budu povučeni ka pozitivnoj anodi i doći će do proticanja struje. Primena vakuumske diode je kao ispravljač napona iz AC u DC.

Trioda se razlikuje od vakuumske diode po tome što ima jednu dodatnu elektrodu, u obliku rešetke. Rešetka se nalazi na putu elektrona između katode i anode i na nju se dovode naizmenični signali male amplitude koje treba pojačati. Kada je trioda direktno polarisana i kada je na rešetki pozitivan napon, rešetka potpomaže elektronima da pređu sa katode na anodu i na anodi se javlja uvećan napon koji je negativan zbog prisustva elektrona, dok se na katodi javlja isto toliko uvećan napon ali pozitivan. Kada se dovede negativan napon na rešetku, on će sprečiti prelazak elektrona sa katode na anodu pa anoda postaje pozitivna, a katoda negativna.



Slika 3. Šema vakuumske triode.

Zbog svog jednostavnog dizajna, trioda ima najmanje distorzije od svih ostalih cevi. Ona takođe ne proizvodi neželjene harmonike trećeg reda. Zbog male distorzije, nema potrebe za negativno povratnom spregom koja kompresuje signal i time ga zadržava u oblast ne-distorzije.



Slika 4. Kompresovanje signala.

Zbog unutrašnje kapacitivnosti koja se javlja između rešetke i anode, koja iznosi oko 3.8 pF, nije moguće pojačanje visokofrekventnih radio signala jer ova kapacitivnost predstavlja kratak spoj za takve signale. Ova kapacitivnost ne predstavlja problem za audio frekvencije jer su one oko 1000 puta manje od radio frekvencija. Takođe se ova kapacitivnost naziva Milerova kapacitivnost i postoje tehnike za njeno eliminisanje, ali su dosta teške za realizaciju i upravljanje [3] .

Kada se na rešetku dovedu signali male amplitude koje treba pojačati i ako ne postoji otpornik (zaštitnik) koji ograničava anodnu struju, na anodi će se javiti mnogo veći signal u odnosu na rešetku koji nije u fazi. Ovako veliki pojačani signal će negativno uticati na funkcionalnost rešetke koja se nalazi blizu anode.

Toplota koja se generiše na (unutar) cevi potiče od samog grejača i usled bombardovanja površine anode elektronima.

Postoje više načina na koje se cev hladi:

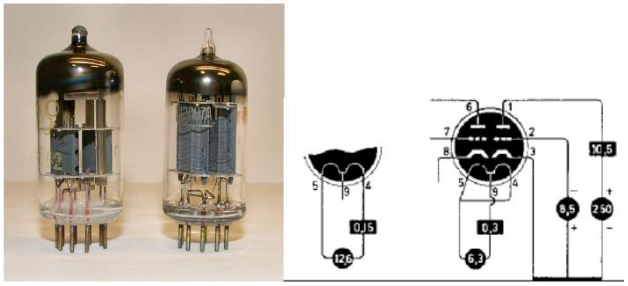
1. Konvekcijom vazduha preko staklenog omotača, gde je potrebno povećati njegovu površinu tj. veličinu, i zbog toga ove cevi mogu da budu velikih dimenzija.
2. Drugi način za hlađenje je oslobađanje toplote po principu radijacije crnog tela od anode kao IC radijacija.

Prvi način odnosno konvekcija nije uvek moguća jer je oko anode prisutan vakuum. Ovaj problem je kod nekih cevi rešen tako što je anoda izdvojena van cevi, gde je moguće hlađenje konvekcijom vazduha. Još jedna mogućnost ovakvog dizajna za hlađenje je putem vode. Kada je anoda potopljena u tečnost za hlađenje, napon je moguće dovesti direktno u tu tečnost. Potrebno je međutim, da ona bude dobar električni izolator kako bi se sprečilo curenje visokog napona kroz rashladnu vodu u sistem radijatora.

Da bi se poboljšalo emitovanje IC energije sa anode, anoda se presvlači slojem materijala koji će poboljšati njenu emisivnost energije.

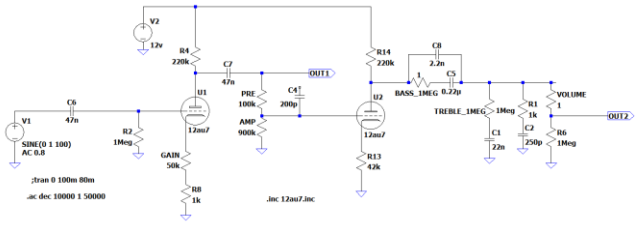
III. PRAKTIČNA REALIZACIJA

Gitarsko pretpojačalo realizovano je pomoću duple triode 12AU7 (slika 5). Pošto se u ovoj cevi nalaze dve triode, potrebna su i dva grejača da bi se omogućio pravilan rad ove cevi. Ukoliko se grejači vezuju serijski potrebno je na njih dovesti napon od 12 V, a ukoliko se vezuju paralelno svakom grejaču je potrebno po 6 V. Struja koja je potrebna da se grejač zagreje je 150 mA za serijsku vezu, odnosno 300 mA za paralelnu. Ova cev ima malo pojačanje između 10 i 20. Druge cevi iz ove serije imaju pojačanje 100, ali je za realizaciju izabrana 12AU7 baš zbog malog pojačanja jer ono neće previše distorzirati zvuk kada se cev odvede u overdrive. Maksimalan napon koji anoda ove cevi može da izdrži je 300 V jednosmernog napona, međutim ceo uređaj je projektovan da radi na 12 V jednosmernog napona.



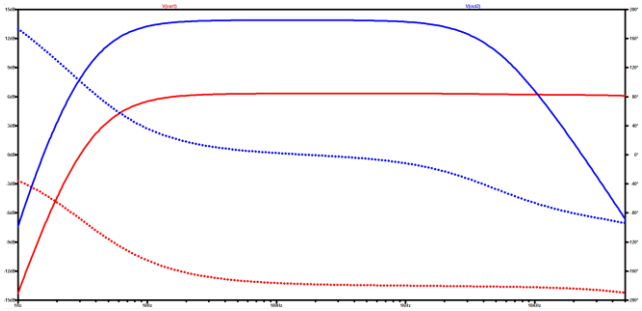
Slika 5. Izgled vakuumske cevi 12AU7 (levo) i načini vezivanja grejača (desno).

Realizacija ovog gitarskog pretpojačala započeta je simulacijom u programu LT spice (Slika 6). Pretpojačalo se sastoji iz dve pojačavačke faze. Za jednu faza iskorišćena je jedna trioda i za drugu druga trioda. Na rešetku prve triode dovodi se gitarski signal. Zbog sudaranja elektrona sa rešetkom dolazi do generisanja jednosmernog napona na njoj pa je taj jednosmerni napon potrebno ukloniti. On se uklanja pomoću otpornika R2 koji se nalazi između rešetke i mase. Takođe je potrebno zaštiti magnete gitare od oštećenja usled ovog jednosmernog napona. To se postiže kondenzatorom C6 koji predstavlja visokopropusni filtar. Ovaj kondenzator od 47 nF je dovoljno velik da ne slabi gitarske frekvencije. Za funkcionisanje ovog kola je potrebno odrediti otpornosti anode i katode. Ove otpornosti ograničavaju protok struje kroz vakuumsku cev i stvaraju napon na katodi oko 2 V. Ostatak napona se nalazi na anodi i on određuje kada će doći do distorzije signala. Što je manji ovaj napon pre dolazi do distorzije. Vrednosti ovih otpornosti utvrđene su pomoću simulacije u LT spice-u. Katodna otpornost se može menjati uz pomoć potenciometra u konfiguraciji reostata što dovodi do menjanja ukupnog pojačanja i tačke u kojoj dolazi do distorzije signala.

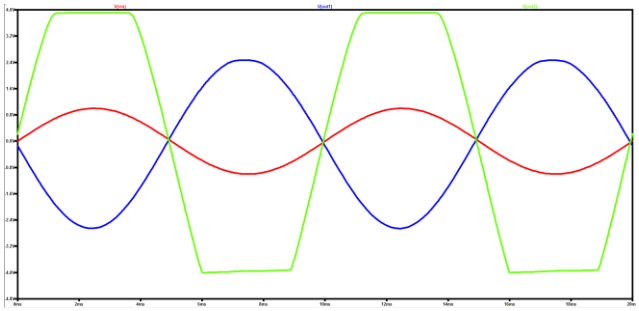


Slika 6. Šema pretpojačala.

Pojačani signal koji se uzima sa anode sadrži jednosmernu komponentu od napajanja koju treba ukloniti. Kondenzatorom C7 se uklanja ova jednosmerna komponenta i na OUT1 pinu se dobija uvećani naizmenični signal posle prve faze koji je fazno pomeren za 180°. Ovako uvećan signal se preko potenciometra, koji se nalazi u konfiguraciji naponskog razdelnika, vodi na rešetku druge triode. Uloga ovog potenciometra je da oslabi signal ukoliko se javlja distorzija na izlazu. Paralelno sa ovim potenciometrom je vezan kondenzator od 200 pF koji propušta visoke frekvencije. Druga trioda ima fiksne anodne i katodne otpornosti koje su određene putem simulacije u LT spice-u i omogućavaju maksimalno pojačanje. Posle druge faze pojačani signal se vraća u fazu sa ulaznim. Tako pojačan signal nailazi na jednostavnu tonsku kontrolu koja otklanja jednosmernu komponentu napona sa anode i oblikuje frekventni odziv pretpojačala. Kondenzatori C8 i C5 sa potenciometrom BASS čine filter koji propušta ili uklanja niske frekvencije. Kada potencimetar ima vrednost 0 signal ide linijom manjeg otpora odnosno kroz kondenzator C5 koji propušta sve frekvencije. Ako potenciometar ima otpornost 1 MΩ, ova otpornost predstavlja za male signale prekid u kolu i signali će ići preko kondenzatora C8 koji oslabljuje sve frekvencije ispod 200 Hz. Isti princip je iskorišćen za niskopropusni filtar koji je sačinjen od kondenzatora C1 i TREBLE potenciometra. Kada potenciometar ima maksimalnu otpornost od 1 MΩ signal ne prolazi kroz kondenzator C1. Kada se otpornost smanji ovaj potenciometar i kondenzator čine niskopropusni filtar koji propušta sve frekvencije ispod 1 kHz. Otpornik R1 i C2 imaju ulogu da dodatno uklone visoke neželjene visokofrekventne šumove i frekvencije. Na kraju signal nailazi na VOLUME potenciometar u konfiguraciji naponskog razdelnika koji kontroliše jačinu zvuka na izlazu pretpojačala.



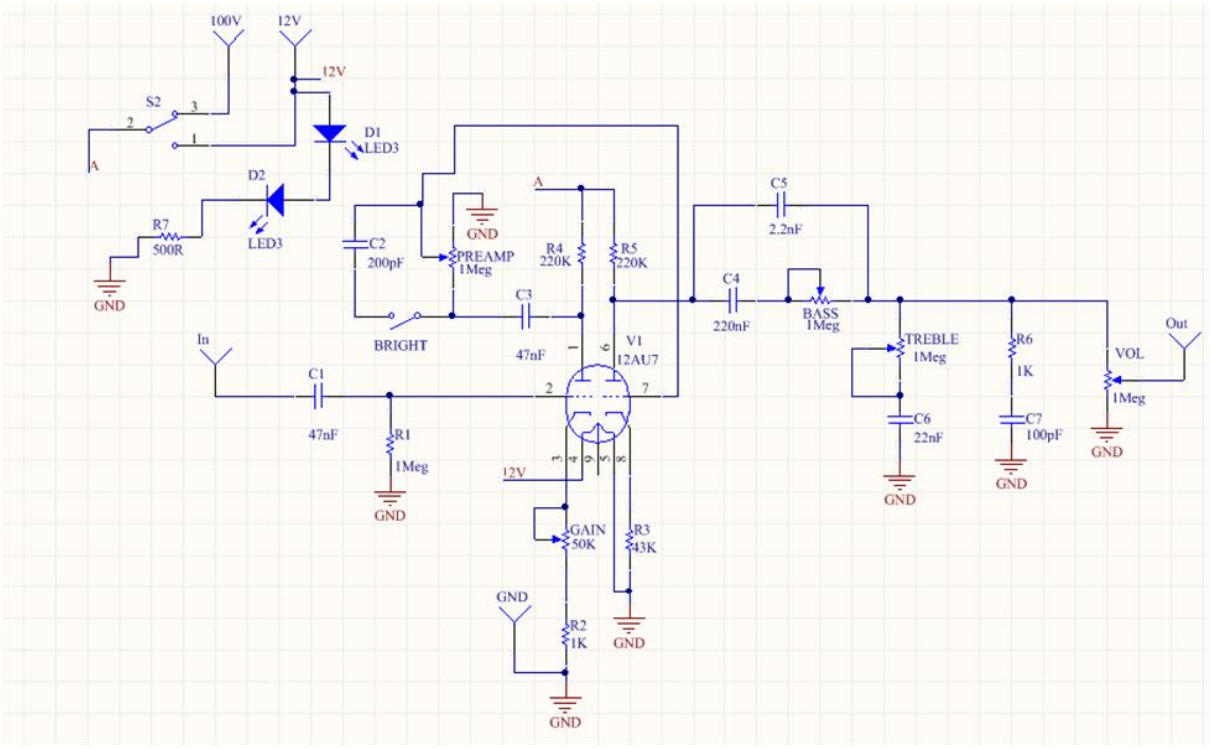
Slika 7. Grafički prikaz AC analize simulacije u LT spice-u.



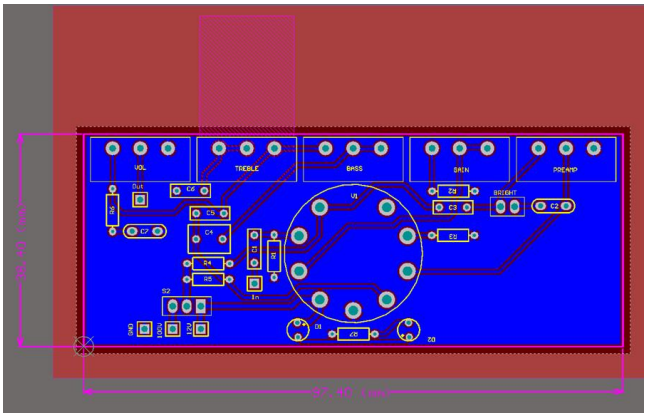
Slika 8. Grafički prikaz ulaznog signala (crven), posle prve faze (plavi) i na izlazu (zeleni).

IV. Realizacija u Altium Designer-u

Ovo gitarsko pretpojačalo je dizajnirano i realizovano na jednoslojnoj štampanoj ploči uz pomoć Altium Designer-a [4]. Na slici 9 prikazan je šematik. Pločica je realizovana tako da se može dovoditi napon veći od 12 V na anode trioda. Potenciometri su direktno povezani na ploču i dodate su dve LED diode kao indikacija da se na grejačima nalazi 12 V. Prilikom dizajniranja ploče potrebno je voditi računa o debljini linija da bi mogle da izdrže struju potrebnu da bi grejači mogli raditi. Masa je razlivena da bi se eliminisali svi mogući šumovi koji se mogu javiti u kolu.



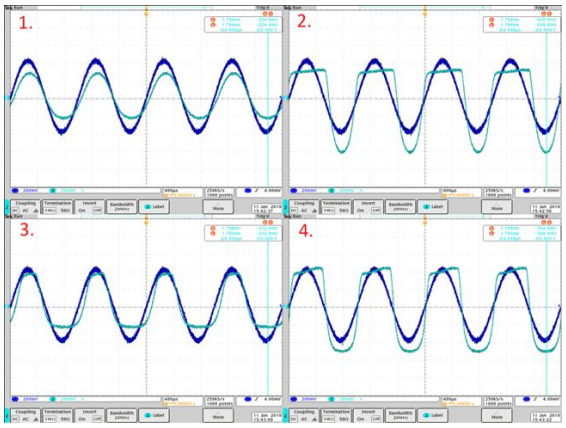
Slika 9. Šematski prikaz pretpojačala u Altium Designer-u.



Slika 10. 2D prikaz pločice i raspored komponenata.

V. ZAKLJUČAK

Nakon što je simulacija dala zadovoljavajuće rezultate, pretpojačalo je realizovano na štampanoj ploči i testirano. Međutim u praksi ono nije dalo zadovoljavajuće rezultate. Kada radi na 12 V izlazni signal je manje amplitude nego ulazni. Tek kada se na anodu dovede 30 V izlazni signal je iste amplitude kao ulazni. I pored ovog nedostatka, kada se cev odvede u overdrive, dobija se veoma prijatan i "prirodno" distorziran zvuk gitare.



Slika 11. Ulazni i izlazni signali vakuumskog pretpojačala.



Slika 12. Praktično realizovana pločica gitarskog pretpojačala.

ZAHVALNOST

Zahvaljujemo se prof. dr Emiliji Živanović i prof. dr Snežani Golubović na pruženoj pomoći tokom izrade ovog rada.

LITERATURA

[1] Reich, Herbert J. (13 April 2013). *Principles of Electron Tubes* (PDF). Literary Licensing, LLC. ISBN 978-1258664060. Archived (PDF) from the original on 2 April 2017.

[2] *Fundamental Amplifier Techniques with Electron Tubes: Theory and Practice with Design Methods for Self Construction*. Elektor Electronics. 1 January 2011. ISBN 978-0905705934. [

[3] John M. Miller, "Dependence of the input impedance of a three-electrode vacuum tube upon the load in the plate circuit," *Scientific Papers of the Bureau of Standards*, vol.15, no. 351, pages 367-385 (1920).

[4] https://www.altium.com/altium-designer/.

Projektovanje verifikacione komponente za HI-6130 čip korišćenjem UVM metodologije

Nikola Radisavljević

Sadržaj- Cilj ovog rada jeste da predstavi proces projektovanja verifikacione komponente koja omogućava verifikaciju HI-6130 čipa i interfejsa koji procesor koristi da pristupi čipu. Svi delovi verifikacione komponente će biti detaljno objašnjeni, pri čemu će se posebna pažnja pridodati ključnim delovima koda koji pokrivaju najbitnije funkcionalnosti verifikacione komponente. Takođe, detaljnije će biti objašnjeni delovi koda koji služe da provere vremenske zahteve koji se nalaze u tehničkoj dokumentaciji – trajanje signala, vreme koje je proteklo između dve promene, kada se desila promena signala, itd.

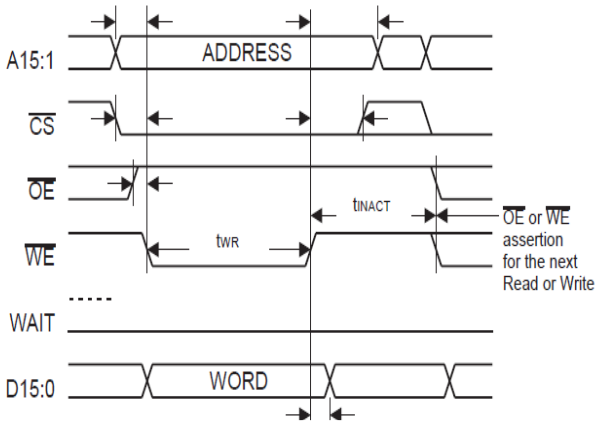
I. Uvod

Testiranje funkcionalnosti čipa pre same proizvodnje istog je možda i najbitniji korak u procesu projektovanja sistema. Ukoliko čip ili sistem dođu na tržište bez detaljne verifikacije, postoji realna šansa da uređaj ne radi po specifikaciji, pa čak i da ne radi uopšte. Ovakvi čipovi i sistemi, koji su često delovi većih sistema ili bitnih uređaja, moraju biti povučeni iz prodaje, čime su gubici za kompaniju koja proizvodi čip, kao i kompanije koje koriste proizvedeni čip, ogromni. Zbog toga je jako bitno da se pre proizvodnje samog čipa umanji šansa da dođe do lošeg funkcionisanja uređaja.

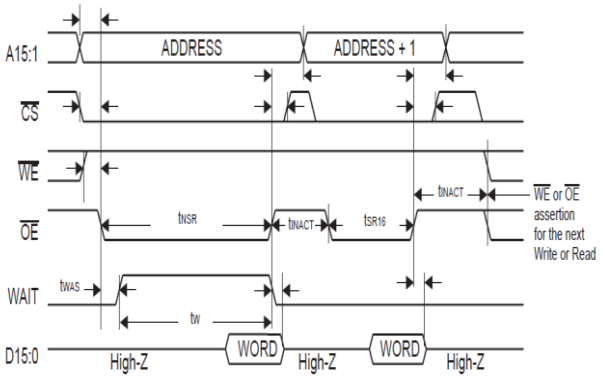
II. Karakteristike interfejsa koji se verifikuje

Da bi omogućili nekom procesoru (npr. procesoru sa FPGA čipa) da komunicira sa uređajima koji komuniciraju koristeći MIL-STD-1553 protokol, gde problem predstavljaju različiti naponski nivoi signala koji se koriste u pomenutom protokolu, potrebno je da između MIL-STD-1553 magistrale i host procesora, omogućimo neometanu komunikaciju. Ovo se postiže sa HI-6130 čipom, koji koristi 16-bitni paralelni interfejs da komunicira sa host procesorom sa jedne strane, a sa druge koristi naponske transformatore da prilagodi podatke iz memorije MIL-STD-1553 standardu. Pomenuti paralelni interfejs na HI-6130 čipu može da koristi dva stila predstavljanja podataka (Intel i Motorola stilovi), i podržava dve širine magistrale (8-bitna i 16-bitna). U ovom radu će biti detaljno opisano korišćenje 16-bitne magistrale, Intel stil predstavljanja podataka, međutim, sa veoma malo modifikacija je moguće lako iskoristiti druge kombinacije.

N. Radisavljević, verifikacioni inženjer u kompaniji Veriest, student na smeru Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikola.radisavljevic@elfak.rs

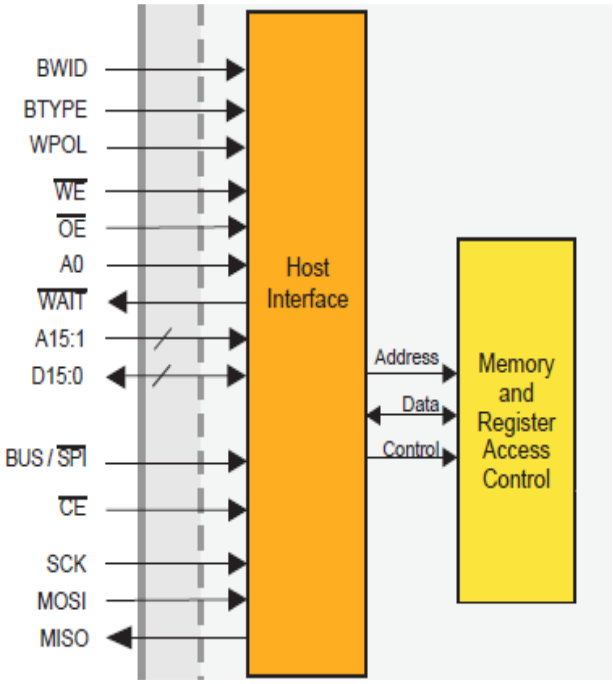


Slika 1. Izgled signala na interfejsu pri upisu u memoriju.



Slika 2. Izgled signala na interfejsu pri čitanju iz memorije.

HI-6130 interfejs omogućava komunikaciju između procesora i uređaja koji koristi MIL-STD-1553 protokol. Postoje dve opcije da procesor pristupi memoriji čipa: koristeći 16-bitni paralelni interfejs, ili koristeći SPI interfejs. U ovom radu, koristiće se isključivo prethodno opisani, 16-bitni paralelni interfejs, pri čemu svi vremenski uslovi protokola moraju biti ispoštovani. Blok dijagram interfejsa HI-6130 čipa se može videti na Slici 3.

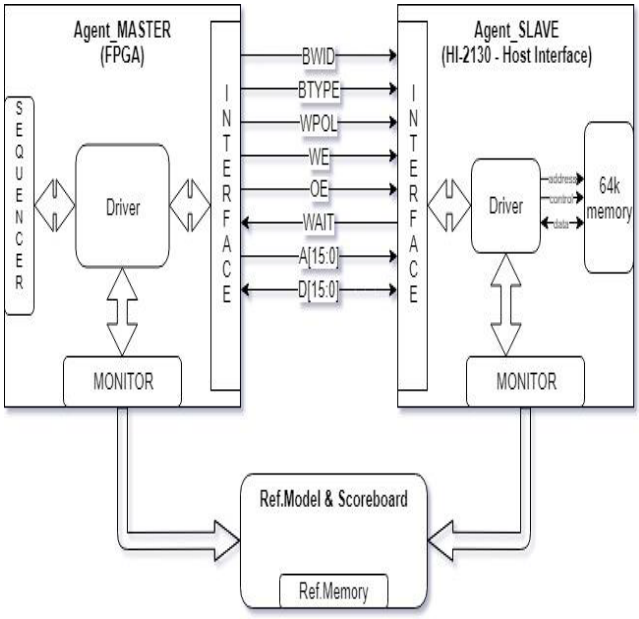


Slika 3. Blok dijagram paralelnog interfejsa HI-6130 čipa.

III. VERIFIKACIONA KOMPONENTA

Verifikaciona komponenta je projektovana u jeziku SystemVerilog i pisana je po UVM-u (Universal Verification Methodology). UVM predstavlja skup pravila i načina kodiranja koji teži da istovremeno olakša i proširi mogućnosti pri projektovanju, a istovremeno da ih standardizuje tako da je ponovno korišćenje kreiranih okruženja lako i smisleno. Ponovno korišćenje (reusability) je jedan od najbitnijih načela verifikacije, pa je zato UVM jako korisno sredstvo u samom procesu. Takođe, jedna od najvećih prednosti UVM-a je automatizacija unutar fabrike UVM-a. Fabrika služi da umesto korisnika izvrši razna podešavanja, povezivanja i uključivanja, čime se korisniku olakšava sam proces i smanjuje mogućnost greške. Takođe, sama fabrika u neku ruku omogućava nivo apstrakcije određenih postupaka – na primer, jednostavnim pozivanjem create() funkcije i definisanjem određenih parametara, korisniku se značajno olakšava postupak kreiranja objekta tačno kako on želi, dok fabrika u zavisnosti od unetih argumenata iza scene izvršava sve za korisnika komplikovane radnje.

Kada projektujemo verifikacionu komponentu, potrebno je prvo da osmislimo kako će izgledati okruženje te komponente. 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. Sve ove komponente će biti detaljno opisane u nastavku rada.



Slika 4. Arhitektura verifikacionog okruženja.

Pomenuti agent enkapsulira komponente tako što ih uključi u listu za kompajliranje komandom *include* (kao u drugim programskim jezicima), pa će komponente u nastavku biti opisivane redosledom po kome se uključuju u fajl na najvišem nivou.

A. Definicije

Prvi fajl koji se uključuje sadrži definicije različitih promenjivih koje se koriste u radu. Ovo se radi da bi sam kod bio jasniji i čitak, a kasnije, ukoliko se komponenta koristi u drugim uslovima, lakše je promeniti definiciju nego unositi izmenu u svim fajlovima koji se nalaze u okviru komponente. U fajlu za definicije korišćenom u ovom radu se nalaze definicije i vrednosti za eksterne fajlove za rad sa memorijom, kao i minimalna i maksimalna vremena za upis i čitanje, koja su definisana u tehničkoj dokumentaciji.

B. Memorija

S obzirom da memorija koju poseduje HI-6130 čip nije standardna, već joj je dodata mogućnost da određenim lokacijama u memoriji korisnik nema pristup, kao i mogućnost prefetch-a podataka sa naredne memorijske lokacije, potrebno je napraviti memoriju i uključiti je u verifikacionu komponentu.

C. Transakcija

Jedan od pristupa koji se koristi u verifikacije jeste TLM (Transaction Level Modeling).Cilj TLM-a jeste da omogući modelovanje sistema sa određenom dozom apstrakcije, budući da i sami inženjeri kada posmatraju

sisteme razmišljaju na nivou transakcije – paketi podataka između uređaja, skup instrukcija u komandi, itd. Kada kažemo transakcija, mislimo na jedan objekat koji sadrži razna polja – ona se mogu koristiti za konfiguraciju, da odrede kako će izgledati neka komanda ili operacija, da ukažu koji se podaci upisuju, na koju adresu, tj. sve stavke koje omogućavaju detaljno konfigurisanje verifikacione komponente radi kreiranja različitih scenarija. U UVM-u, transakcije imaju četiri glavne uloge: modeluju kako izgledaju transakcije podataka između različitih interfejsa DUT-a (uređaja koji testiramo, Device Under Test), definišu ograničenja okruženja, kontrolišu proces randomizacije i omogućavaju automatizaciju unutar UVM-a.

D. Drajver

Kao što samo ime kaže, driver služi da drajvuje signale interfejsa verifikacione komponente. U zavisnosti od transakcije i njenih polja, tj. njene konfiguracije, drajver kontroliše ponašanje signala na interfejsu, čime se zapravo izvršavaju protokoli komunikacije između komponenti. Bitno je naglasiti da, po pravilima metodologije, se nikakvo generisanje podataka ne dešava u drajveru. On isključivo "dekodira" podatke koji se nalaze u transakciji, i nakon toga po definisanom protokolu šalje podatke uređaju koji se verifikuje. S obzirom da drajver mora da zna trenutno stanje signala na magistrali kako se ne bi kršila pravila protokola, potrebno je omogućiti drajveru da pristupi signalima na interfejsu, bilo direktno, bilo dobijanjem informacije od strane monitora.

S obzirom da se jedna verifikaciona komponenta sastoji iz više različitih podkomponenata (drajver, monitor, itd.), jako je bitno da sve one budu sinhronizovane. To znači da je potrebno da istog trenutka kada drajver počinje sa promenom linija magistrale, monitor u isto vreme krene sa praćenjem tih linija. Za ovakvu sinhronizaciju se koriste tzv. faze. U UVM-u, postoje predefinisane faze koje imaju sve komponente nasleđene iz klasa UVM fabrike, pa se iz tog razloga sve metode i funkcije koje postoje u komponentama nalaze u okviru takvih faza. Na primer, sve komponente se generišu i povezuju u okviru build faze, glavna funkcionalnost se nalazi u run fazi itd. Takođe je bitno napomenuti, da kada se poziva build faza neke klase, neophodno je pozvati istu fazu i njenog roditelja klase.

E. Interfejs

Interfejs (signal mapa, port mapa) predstavlja, kao što ime kaže, interfejs verifikacione komponente ka spoljašnosti – koristimo ga da se povežemo na uređaj koji testiramo, sa ostatkom verifikacionog okruženja, sa drugim verifikacionim komponentama, itd. Iako je drajver zadužen za promenu signala, u interfejsu se nalaze deklaracije samih portova signala koje koristimo i njihovi opisi. U UVM-u, u interfejsu se takođe nalaze i tzv. assertion-i. Oni predstavljaju tvrdnje koje moraju biti tačne u svakom trenutku u vremenu, ili pri određenoj promeni. Assertioni imaju dosta prednosti u odnosu na checkere i dosta su fleksibilniji. U ovom radu, assertioni se koriste za provere protokola.

F. Sekvenca

Sekvence u verifikaciji predstavljaju određen redosled različitih akcija – čekanje određenih događaja ili uslova, generisanje transakcija na određeni način, prosleđivanje generisanih transakcija u određenom trenutku drajveru, itd. Ako transakciju zamislimo kao neku stvar koju možemo da menjamo na različite načine, sekvenca nam služi da zapravo iskoristimo transakciju da izvršimo radnje koje želimo.

G. Sekvencer

Sekvencer je komponenta koju UVM fabrika koristi kako bi automatizovala proces sekvenca-drajver. Ovo nam omogućava da koristimone samo jednu verifikacionu komponentu i njene sekvence i drajvere, već kada se radi na većem okruženju, gde se sa višeg nivoa kontroliše veći broj sekvenci, uspešno kontrolišemo i sinhronizujemo rad većeg broja drajvera različitih komponenti. Kod koji se koristi za sekvencer je često kratak i istovetan u različitim verifikacionim okruženjima.

H. Monitor

Monitor se može smatrati i najbitnijom komponentom u verifikacionoj komponenti opisanoj u ovom radu. Monitor zapravo imitira funkcionalnost pravog čipa i na taj način utvrđuje sa čekerima da li se uređaj ponaša po specifikaciji. Najveća funkcionalnost verifikacione komponente u ovom radu se zapravo i nalazi u samom monitoru. Budući da je kod za monitor dugačak i složen, u nastavku će biti opisani delovi koda kako bi se lakše razumeo način rada. Monitor ima zasebne delove u zavisnosti od toga da li se radi o operaciji čitanja ili pisanja, pa iako je teža, s obzirom da sadrži više različitih delova, radi objašnjenja principa rada monitora, razmatraćemo operaciju čitanja iz memorije čipa.

Da bi se detektovala komanda čitanja, potrebno je da određeni uslovi budu ispunjeni, u ovom slučaju, uslov je da je aktivan pin OE, neaktivan pin WE, da je zapravo selektovan čip o kome se radi (CS pin aktivan – CS pin se koristi kada na jednoj magistrali postoji veći broj komponenata, pri čemu se taj pin koristi da naznači sa kojim slave-om komunicira master. Na primer, ukoliko imamo 4 čipa HI-6130, master će postaviti samo CS pin onog slave-a sa kojim želi da komunicira, a iako ostali slave-ovi takođe dele ostale magistrale, CS pin odlučuje da li prihvataju ili ignorišu operaciju) i da je master neaktivan. Međutim, odmah nakon ulaska u petlju, potrebno je postaviti vrednost reset_triggered promenjive na nulu. Ova promenjiva se koristi za slučaj da dođe do reseta u toku

same operacije (na primer, operacija čitanja započne, pre njenog završetka dođe do reseta čipa).

Na početku svake operacije, treba proveriti koliko vremena je prošlo od poslednje operacije. Ovaj čeker će uvek proći kada se radi o prvoj operaciji, međutim, potrebno je ispitati svaku operaciju nakon prve. Vreme u odnosu na koje se ispituje koliko je prošlo se čuvau promenjivoj time_check, a na početku svake nove operacije, od trenutnog vremena se oduzima vreme iz promenjive i to vreme se upoređuje sa vremenskim zahtevom iz specifikacije. Postoje dve različite petlje – jedna ukoliko želimo da padanje ovog čekera izaziva grešku, i druga ukoliko želimo upozorenje.

Ukoliko je dovoljno vremena prošlo od prošle operacije, monitor ulazi u glavni deo petlje. Ukoliko nije došlo do greške u protokolu, monitor će poslati zahtev memoriji da dobije podatak sa adrese koja je stigla putem interfejsa. Memorija će zatim poslati odgovarajući odgovor koji može biti: čitanje iz adrese pri čemu podaci sa te adrese nisu prethodno učitani (prefetch), čitanje pri čemu su podaci prethodno učitani i čitanje iz zabranjene zone. U slučaju da je odgovor memorije (pozivanjem funkcije se dobija vrednost tipa integer koja predstavlja odgovor) nula, znači da se radi o prvoj situaciji. Monitor će u ovoj situaciji komunicirati sa drajverom i obavestiti ga da je potrebno da postavi podatke iz adrese nazad na magistralu. Zatim monitor čeka da drajver podigne WAIT signal i na njegovu promenu ispituje koliko vremena je prošlo, a u slučaju da je prošlo predugo vremena, javlja grešku ili upozorenje u zavisnosti od podešavanja, budući da je došlo do greške u protokolu. Da bismo ispitali koliko je vremena prošlo, koristeći funkciju $cast, skladišti se vreme u promenjivu u trenutku kad je pozvana funkcija.

U slučaju da je odgovor memorije 1, znači da se radi o čitanju iz lokacije koja je prefetch-ovana, pa ovoga puta drajver ne aktivira signal WAIT. Kao i u prethodnom primeru, drajveru monitor saopštava vrednosti koje on koristi da podesi linije na interfejsu na odgovarajuće vrednosti. Ovo se odvija koristeći posebne portove za komunikaciju između monitora i drajvera. Nakon deklaracije ovih portova i njihovih povezivanja, omogućena je direktna komunikacija između monitora i drajvera sa jednom velikom prednošću – svaki put kada se neki podatak ili transakcija prenese preko ovog posebnog porta, doći će do emitovanja eventa koji će automatski započeti postupak od strane drajvera. U prevodu, nema potrebe realizovati funkciju koja će beskonačno čekati na događaj (tzv. polling), već je dovoljno samo kreirati ovu vrstu porta i poslati podatak preko njega.

Poslednji mogući odgovor jeste da je došlo do čitanja iz zabranjene regije, pa će u zavisnosti od toga da li želimo da pristup zabranjenoj memorijskoj lokaciji bude registrovan kao greška ili upozorenje, biti ispisana odgovarajuća poruka.

Nakon što drajver postavi magistrale na odgovarajuće vrednosti, komandu treba privesti kraju tako što drajver deaktivira određene signale. Da bismo u potpunosti verifikovali protokol, potrebno je ispitati da li su vremena koja su protekla do deaktivacije signala u skladu sa tehničkim specifikacijama čipa. Budući da ima nekoliko različitih čekera, u nastavku će biti prikazan kod jednog od njih, dok će ostali biti zajedno sa ostatkom koda u okviru dodatka rada.

Na kraju, potrebno je da utvrditi da li je komanda korektna i da li je došlo do reseta tokom same komande. Da bismo registrovali reset koji se može desiti u bilo kom trenutku, potrebno je da napravimo poseban task (za razliku od obične funkcije, taskovi mogu trajati bilo koji vremenski period, ne samo u okviru jednog trenutka) koji će pratiti da li dolazi do reseta.

Ukoliko do reseta dođe, potrebno je to zabeležiti u posebnu promenjivu, koja će se ispitati na kraju komande i figurisati u tome da li je operacija validna.

U slučaju da nije došlo do reseta i da je operacija validna bez grešaka u protokolu, monitor će informaciju poslati drugoj komponenti koja se zove scoreboard i koja služi za proveru validnosti podataka, o kojoj će biti više reči u kasnijoj glavi rada. Pored slanja informacije, monitor beleži vreme kada se operacija završila, kako bi se na početku naredne operacije ispitalo koliko je vremena prošlo od završetka prošle, i takođe vraća polja koja služe da ispitaju da li je došlo do reseta ili greške u protokolu, budući da ove promenjive gledamo na nivou operacija.

I. Agent

Agent služi da enkapsulira sve ostale prethodno pomenute delove u jednu celinu. Agent je onaj koji se instacira u verifikaciono okruženje da bi se koristila verifikaciona komponenta i on u sebi često sadrži određena konfiguraciona polja, kako bi se naglasilo kakav je tip komponente. U ovom radu, projektovana verifikaciona komponenta ima dva tipa agenta: master i slave, i u zavisnosti od ovog podešavanja, biće instanciran različiti agent. Ovo nam omogućava da u jednom okruženju (o čemu će kasnije biti reči), instanciramo dva agenta, jednog kao master, drugog kao slave, povežemo ih međusobno i uspostavimo sistem koji nam omogućava da u potpunosti testiramo verifikacionu komponentu. U masteru se vrši deklaracija komponenti koje sadrži, kreiraju i registruju komponente u UVM fabriku, povežu interfejsi i virtuelni portovi, kao i podese konfiguraciona polja.

J. Paket

Budući da komponente u okviru jednog agenta komuniciraju između sebe i budući da se instanciraju u samom agentu, potrebno je da napravimo fajl koji uključuje sve prethodno napisane fajlove. U okviru jednog ovakvog paketa se nekad mogu naći i definicije nekih tipova (u ovom paketu se nalazi nova enumeracija koja služi za označavanje da li je agent MASTER ili SLAVE) i obavezno uključivanje posebnog fajla koji omogućuje korišćenje UVM makroa. Takođe, u paketu se nalazi

definisanje vremenske skale, u kojoj se definiše vremenska jedinica koja se koristi u okruženju.

IV. VERIFIKACIONO OKRUŽENJE

Da bismo koristili i instancirali verifikacione komponente, potrebno je da kreiramo verifikaciono okruženje i testbench. Ovo okruženje nam omogućava da povežemo veliki broj različitih ili istih komponenata i na taj način emuliramo neki veći, kompleksniji sistem. U okviru okruženja se takođe nalazi i scoreboard, koji služi da prima informacije od svih monitora svih agenata, upoređuje njihove podatke sa referentnim podacima i izbaci grešku ukoliko je neki rezultat različit od očekivanog.

A. Scoreboard

Često verifikaciona okruženja sadrže veliki broj komponenata koje simuliraju neki sistem. Pored toga što je potrebno da znamo da li pojedinačne komponente funkcionišu po specifikaciji, moramo da znamo da li te komponente međusobno pravilno komuniciraju. Na primer, ukoliko imamo čip gde podatak mora da prođe kroz nekoliko blokova da dođe do memorije, ne samo da moramo da verifikujemo da svi blokovi zasebno tačno prenose podatak, moramo da verifikujemo da od prve do poslednje tačke toka podatka kroz sistem, ne dolazi do neželjenih promena. Za sve navedene operacije, u UVM-u se koristi tzv. scoreboard. On je povezansa svim monitorima verifikacionih komponenti i od njih prima podatke o transakcijama. U ovom radu, svaki put kada monitor odredi da je transakcija uspešno odrađena, šalje podatke scoreboard-u da je došlo do transakcije, na kojoj adresi je došlo do koje operacije i sa kojim podacima. Na primer, ukoliko dođe do operacije upisa u čip, monitor će poslati informaciju scoreboard-u u koju adresu se upisuje i koji se podaci upisuju. Nakon toga, kada dođe do operacije čitanja iz čipa, kada monitor pošalje transakciju, scoreboard će iz svoje referentne memorije pročitati podatke koji su upisani i uporediti rezultate ove dve operacije. Kao što se može videti iz navedenog primera, pored prednosti koje su navedene, scoreboardpredstavlja još jedan dodatni mehanizam provere tačnosti podataka i operacija.

Tokom opisa monitora i u prethodnom pasusu, rečeno je da postoji posebna vrsta porta između monitora i scoreboard-a koji služi za njihovu neometanu komunikaciju i automatizovan rad. Ovi portovi se nazivaju analysis portovi. Komunikacija se odvija u jednom pravcu: deo porta koji šalje podatak se deklariše kao analysis port, dok se deo koji prima podatke deklariše kao implementacija tog porta. Zatim, definišemo funkcije vezane za taj port, koje označavaju da se pri događaju na tom portu dešavaju određene akcije. Pored komunikacije monitor-scoreboard, moguće je takođe povezati test-scoreboard, kako bi se prosledili dodatni podaci radi poređenja.

B. Okruženje

U okviru okruženja se instanciraju i kreiraju agenti i scoreboard, povezuju se portovi, podešavaju se konfiguraciona polja agenata. Konfiguraciona polja parametara se podešavaju preko posebne baze podataka koja postoji u UVM-u. Samoj bazi se može pristupiti iz svakog fajla u okruženju, pa je ona zgodna za podešavanje određenih parametara, zbog toga što je moguće podesiti ove parametre u okruženju ili testu, u zavisnosti od potrebe, što povećava ponovnu upotrebljivost komponenata.

C. Testovi

Kada se projektuju verifikacione komponente i izgradi okruženje, konačni korak je pisanje i pokretanje testova. Testovi služe da se okruženje simulira koristeći određene situacije. Najčešće se koristi mešavina nasumičnih testova (sva podešavanja i situacije su što opštije i gleda se da se što više ostavi nasumičnosti) i direktnih testova (testira se određeni scenario, gađa se neki granični slučaj i testiraju ekstremne vrednosti). Kada se radi o kompleksnim sistemima, moguće je pisanje testova ostaviti posebnom verifikacionom inženjeru, koji se naziva test inženjer, čiji je posao da zapravo piše testove za okruženje koje je projektovao neki drugi inženjer. Tokom ovog rada, koristi se veliki broj novih definisanih enumerisanih tipova promenjivih – cilj ovih promenjivih je da omogući test inženjeru (neko ko će u budućnosti za ovo okruženje pisati testove) lako razumevanje okruženja i lako ograničavanje promenjivih.

U test fajlu u ovom radu, koristi se bazni test kao template za sve ostale testove. S obzirom da okruženje poseduje veliki broj testova, bolji pristup je napraviti bazni test u kome se nalaze sve promenjive i polja koja se koriste u svim testovima, i onda pisati testove koji kao klase nasleđuju bazni test. Da se bazni test ne koristi, sva polja i promenjive bi morale da se definišu u svakom testu.

V. PROPRATNA DOKUMENTACIJA

Pri projektovanju verifikacione komponente koja se isporučuje klijentu, potrebno je obezbediti i potrebnu propratnu dokumentaciju. Pored verifikacionog plana, koji je opisan ranije u ovom radu, klijentu se takođe isporučuje i korisničko uputstvo (User Guide) u kome se nalaze sva uputstva potrebna za korišćenje komponente, kao i informacije o izdanju/verziji (release notes) u kome se nalaze spiskovi promena i funkcionalnosti verzije komponente, kao i sadržaj svake isporuke koja se šalje klijentu.

A. Korisničko uputstvo

Korisničko uputstvo komponente koja se koristi u ovom radu sadrži: uputstva za pokretanje simulacije i testova, integraciju verifikacione komponente u okruženje,

podešavanje opsega zabranjenih adresa, kontrolu čekera za vremena i kontrolu čekera za protokol.

Prvi deo korisničkog uputstva uključuje izgled foldera i pregled fajlova koji se koriste u okruženju. Korisniku se u koracima objašnjava u kom direktorijumu treba da se nalazi pre pokretanja komandi i gde treba da budu smešteni fajlovi (korisničko uputstvo podrazumeva da se sve komande pokreću iz Sim foldera, pa je zato bitno da korisnik bude u tom folderu pri pokretanju komande), koje su mu komande potrebne, kako izgledaju dodatne linije ukoliko želi da napuni memoriju podacima iz spoljašnjeg fajla, kao i uputstvo za podešavanje verboziteta poruka. Takođe, prikazani su talasni dijagrami signala na interfejsu pri operacijama čitanja i upisa, kako bi korisnik znao kako treba da izgledaju pravilni oblici signala pri datim operacijama. Ostatak uputstva pokriva podešavanja vezana za integraciju i podešavanje okruženja, koja su objašnjena u prethodnim glavama ovog rada.

B. Informacije o izdanju/verziji

Informacije o izdanju/verziji se isporučuju sa ciljem da isprate tok ispravki dodavanja novih funkcionalnosti kod verifikacione komponente. Na ovaj način se može pratiti razvoj komponente i faze projektovanja.

VI. ZAKLJUČAK

Proces verifikacije modernih čipova i uređaja je dug i kompleksan proces. Međutim, neretko su čipovi nove generacije veoma slični prošloj generaciji, tj. postoji veoma malo funkcionalnih razlika između dve generacije istog čipa. Iz tog razloga, projektovanje verifikacionih komponenata omogućava da, sa manjim modifikacijama, postoji već spremno rešenje za određeni blok ili sistem, koje omogućava da se olakša i skrati pomenuti proces.

U radu je akcenat stavljen na opis i objašnjenje komponenata okruženja, koje služe za verifikovanje HI-6130 čipa. Pored tehničkih informacija, uložen je trud da se pomenuti koncepti i komponente objasne što prostije, kako bi pristup rešavanju problema bio jasan čitaocu. Iako UVM kao metodologija teži da ceo proces bude standardizovan, verifikacioni inženjer u okviru metodologije ima veliku slobodu da primeni veliki broj različitih rešenja i pristupa. Istovremeno, iako ne popularna kao samo kodiranje i projektovanje okruženja, propratna dokumentacija ima svoje mesto u radu.

Konačno, budući da je verifikaciona komponenta projektovana od strane inženjera sa u tom trenutku jako malo iskustva, postoji dosta mesta za unapređenja i optimizaciju koda, kao i pristupa i rešenja iskorišćenih pri verifikaciji.

ZAHVALNICA

Autor se zahvaljuje prof. dr Mioni Andrejević Stošović na izdvojenom vremenu, sugestijama i pomoći pri pisanju i lektorisanju master rada, na osnovu koga se temelji ovaj rad. Takođe, autor se zahvaljuje kompaniji Veriest, u kojoj je projektovana pomenuta verifikaciona komponenta.

LITERATURA

[1] MIL-STD-1553: Complete online reference. Dostupno na: https://www.milstd1553.com/ (pristupljeno septembra 2019.)
[2] Web-site: http://www.holtic.com/products/3058-hi-6130.aspx (pristupljeno septembra 2019.)
[3] Web-site: http://www.holtic.com/products/3066-hi-2130.aspx (pristupljeno septembra 2019.)
[4] UVM User's Guide – Accellera. Dostupno na: https://www.accellera.org/images/downloads/standards/uvm/uvm_users_guide_1.1.pdf (pristupljeno septembra 2019.)
[5] IEEE Standard for SystemVerilog— Unified Hardware Design, Specification, and Verification Language -IEEE Computer Society and the IEEE Standards Association Corporate Advisory Group
[6] "SystemVerilog 3.1a Language Reference Manual" – Accellera. Dostupno na: http://www.ece.uah.edu/~gaede/cpe526/SystemVerilog_3.1a.pdf (pristupljeno septembra 2019.)
[7] UVM Cookbook – Mentor Graphics, 2018.
[8] Methodology for Checking UVM VIP - Milan Vlahović

Kvantni tranzistori sa efektom polja- HEMT (High Electron Mobility Transistor)

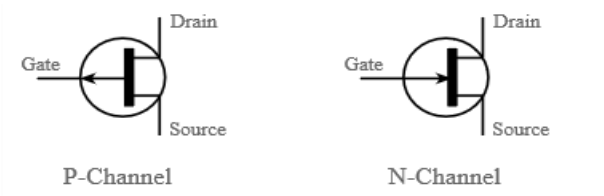
Jana Kotnik

Sadržaj–Tranzistori predstavljaju aktivnu poluprovodničku komponentu i njihova funkcija u kolu može biti različita. Često se u kolu koriste kao pojačavači, prekidači, koriste se u kolima za stabilizaciju napona, modulaciju signala i još mnogo toga. Njihova osnovna podela je na bipolarne tranzistore (BJT- *bipolar junction transistor*) kod kojih su nosioci naelektrisanja elektroni i šupljine, i na unipolarne (FET- *Field Effect Transistor*) kod kojih samo jedna vrsta nosilaca učestvuje u provođenju struje. Jedan od tranzistora koji se često primenjuje je upravo tranzistor koji će biti opisan u ovom radu, a to je HEMT.

I. FET(*FIELD EFFECT TRANSISTOR*)

HEM tranzistori pripadaju porodici FET tranzistora. FET koriste električno polje da kontrolišu struju koja protiče kroz njih. Takođe, imaju visoku ulaznu impedansu što može biti jako korisno u kolima. FET se sastoji od elektroda na oba kraja koje se zovu drejn i sors i prilikom dovođenja napona na gejt dolazi do daljeg nastanka poluprovodničkog kanala. Kontrolišuća elektroda je upravo gejt i ona se nalazi u neposrednoj blizini kanala tako da je napon na njemu u mogućnosti da utiče na kanal. Poluprovodnički kanal u kome dolazi do protoka struje može biti N ili P tipa. To govori da postoje dve vrste FET-ova, P-kanalni i N-kanalni. Tri izvoda FET-ova su drejn, gejt i sors. Najbitnije karakteristike ovih tranzistora su :

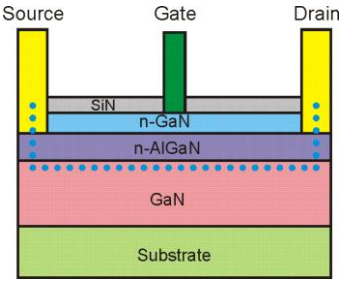
- Vgs- maksimalni napon između gejta i sorsa,
- Vdss- maksimalni napon između drejna i sorsa
- Igss- inverzna struja curenja,
- $Vgs_{(th)}$- napon praga- minimalni napon koji treba dovesti gejtu da bi došlo do formiranja kanala između drejna i sorsa,
- Idss- struja drejna kada je na sorsu napon 0V,
- $Vgs_{(off)}$- napon isključenja,
- $Rds_{(on)}$- drejn-sors otpornost- otpornost kanala između ova dva izvoda kada je FET uključen,
- Ptot- Maksimalna snaga koju FET može konstantno disipirati bez oštećenja tranzistora.



Slika 1. N-kanalni i P-kanalni FET.

Jana Kotnik je sa katedre za mikroelektroniku, Elektronskog fakulteta, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: jana.kotnik@elfak.rs

Pošto jedino električno polje kontroliše struju koja protiče kroz kanal, za ove uređaje se kaže da su naponski kontrolisani sa vrlo visokom ulaznom impedansom koja može biti i po nekoliko MΩ. Ovo je izrazito velika prednost u odnosu na bipolarne tranzistore koji su strujno kontrolisani i imaju znatno manju ulaznu impedansu. Glavna uloga ovih tranzistora je u integrisanim kolima. Oni su posebni zbog toga što im je potrošnja snage u kolima jako mala, pa to omogućava njihovu široku i jako funkcionalnu primenu. Koriste se u digitalnim kolima, kao uređaji za memoriju, analogni prekidači, kao *buffer*-i, multiplekseri, pojačavači, za ograničavanje struje itd. Pored toga što se koriste u integrisanim kolima, diskretne verzije ovih tranzistora se mogu naći i kao komponente za površinsku montažu (SMD) i kao komponente sa izvodima (TH).



Slika 2. Struktura HEM tranzistora baziranog na silicijum-nitridu i galijum-nitridu.

II. HEM TRANZISTORI

HEM (*High Mobility Transistor)* u bukvalnom prevodu može značiti tranzistor sa velikom pokretljivošću (nosilaca). On spada u grupu tranzistora sa efektom polja, koji poseduje neobično uzak kanal i time omogućava njegovo funkcionisanje na ekstremno visokim frekvencijama. Pored toga što je izvanredan na visokim frekvencijama (u mikrotalasnom i RF području), on se takođe odlikuje vrlo niskim nivoom šuma. Suštinski, to je komponenta koja koristi pripajanje dva materijala sa različitim energijama zabranjenih zona *(heterojunctions components)* kao kanal umesto dopirane regije koja se koristi kod standardnih MOSFET-ova.

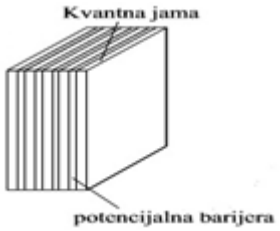
A.Razvoj HEMT-a

Iako HEMT ima poprilično slične karakteristike kao i drugi FET-ovi, za razliku od njih kod njega se javlja neobičan način kretanja elektrona. Ovakav način

prenošenja nosilaca je bio istraživan u 1969. godini, ali tek je tokom 1980. godine došlo do stvaranja prvog eksperimentalnog uređaja koji je bio dostupan za korišćenje [1]. Tokom osamdesetih su se krenuli masovnije koristiti, ali im je velika mana bila visoka cena i mali broj proizvoda. Sada, kada je njihova cena znatno opala, oni se koriste naširoko, čak nalazeći upotrebu u oblastima kao što su telekomunikacije, kao i varijacije na mikrotalasnom i radio-frekventnom području. Kada bi se rad HEMT-a gledao iz ugla kvantne fizike, moglo bi se reći da je za većinu njegovih performansi zadužena upravo struktura kvantne jame. Ideja da se struktura kvantne jame koristi kao tranzistor potekla je iz činjenice da se u ovim strukturama može postići pokretljivost nosilaca u ravni jame koja znatno prevazilazi pokretljivost u supstratu ("masivnom" poluprovodniku), mada samo na nižim temperaturama.

III. KVANTNA JAMA

Poluprovodničke kvantne jame i superrešetke nalaze se poslednjih godina na samom vrhu istraživanja u oblasti fizičke elektronike poluprovodnika. Razvoj specijalnih tehnoloških procesa omogućio je izradu tankih poluprovodničkih slojeva, čije su dimenzije manje od srednje dužine slobodnog puta nosilaca, i iznose do



Slika 3. Izgled kvantne jame.

nekoliko desetina nanometra. Zbog toga su u ovim slojevima izraženi kvantni efekti u smislu da je neophodan kvantnomehanički postupak za opis kretanja elektrona na nivou cele strukture, a ne samo u okviru elementarne ćelije kristala, što dovodi do bitno različitih fizičkih osobina ovih struktura u odnosu na klasične poluprovodnike. Ako su tanki slojevi okruženi balkom čije je dno provodne zone iznad dna provodne zone u sloju, tada većina elektrona u blizini medjupovršine prelazi u tanak sloj. Usled malih dimenzija i postojanja potencijalnih jama, ovi elektroni su konfinirani (ograničeni) i njihovo se ponašanje adekvatno može opisati samo zakonima kvantne mehanike. Najčešće analizirana poluprovodnička mikrostruktura (mikrostrukturu čine tanak sloj i balk) je kvantna jama. Debljina kvantne jame u z-pravcu je vrlo mala (reda nanometara), dok su dimenzije u x i y pravcu vrlo velike, uslovno rečeno beskonačne. Ako bi se u balku realizovala samo jedna jama, efekti bi bili zanemarljivi [2].
U zavisnosti od veličine udaljenja najbližih suseda razlikujemo dva slučaja:

1. Udaljenje je dovoljno veliko, tako da se svaka kvantna jama može analizirati izolovano, i ovo je slučaj višestruke kvantne jame.
2. Udaljenje je malo, tako se cela mikrostruktura može analizirati jedinstveno-superrešetka.

Elektronska struktura kvantne jame određuje se rešavanjem Šredingerove i Poasonove jednačine.

IV. POLUPROVODNIK I KVANTNE JAME

U poluprovodnicima se rasejanje elektrona ostvaruje kroz više mehanizama, tj. pokretljivost zavisi od rasejanja na fotonima, jonizovanim primesama, defektima rešetke, elektronima itd. U kvantnim jamama postoji mogućnost znatnog smanjenja primesnog rasejanja, putem prostornog razdvajanja dopiranih oblasti strukture i oblasti sa visokom koncentracijom elektrona. Ako se u strukturi kvantne jame dopiraju samo delovi barijere koji su nešto udaljeni od jamskog sloja, oslobodjeni elektroni se "slivaju" u vezana stanja jame, jer su ona energetski niža od provodne zone u barijeri. Sada je preklapanje talasnih funkcija elektrona sa centrima rasejavajućeg potencijala primesa zanemarljivo, pa će se i ovaj vid rasejanja jako smanjiti. Odgovarajuća pokretljivost (naravno u ravni jame) može znatno prevazići vrednost u balku, ali samo ako primesno rasejanje dominantno odredjuje pokretljivost, tj. na niskim temperaturama. Na fononsko rasejanje se ne može uticati na visokim temperaturama, pa je tada ono dominatno. Parametri FET-a kao što su npr. transkonduktansa ili brzina prekidanja rastu sa pokretljivošću elektrona (što je i bio razlog da se prilikom izrade FET-ova koristi GaAs umesto Si, pa je dalji razvoj prirodno išao u pravcu primene strukture kvantne jame) [4].

V. ZAKLJUČAK

Tranzistori sa efektom polja predstavljaju ključne poluprovodničke komponente u elektronskoj industriji. Od velike su važnosti kako u digitalnoj, tako i u analognoj elektronici i zbog toga su naširoko u primeni.

ZAHVALNICA

Autor se zahvaljuje prof. dr Goranu Ristiću na mentorstvu i podršci u toku pisanja rada.

LITERATURA

[1] Takashi Mimura: "The Early History of the High Electron Mobility Transistor (HEMT) ", 2002.
[2] Goran S.Ristić: "Osnovi kvantne i statističke fizike"
[3] Ray Dingle, Arthur Gossard and Horst Störmer: "High mobility multilayered heterojunction devices employing modulated doping"
[4] Daniel Delagebeaudeuf and Trong L. Nuyen: "Field effect transistor with high cut-off frequency and process for forming same"

Izrada sustava za otvaranje elektroničke brave putem RFID senzora

Antonio Boras

Sadržaj - U ovom radu opisuje se izrada funkcionalnog prototipa sustava za otvaranje elektroničke brave koristeći RFID tehnologiju i ESP32 mikrokontroler. Glavni cilj ovog rada bio istraživanje kompleksnosti izrade takvog sustava i sigurnost elektroničke brave upravljane RFID senzorima naspram običnih mehaničkih brava. U radu objašnjeni su koncepti RFID tehnologije i mikrokontrolera, kao i terminologija istih te komponente korištene za izradu sustava. Opisan je postupak izrade sustava te moguća unaprijeđena.

I. BRAVE KROZ POVIJEST

Još od najranijih dana ljudske civilizacije, ljudi su imali potrebu zaštiti sebe i druge. Najraniji primjeri zaštite može biti sam ulazak u špilje kako bi se čovjek sakrio od vremenskih nepogoda i tako osigurao suho i relativno toplo mjesto za odmor. Sljedeći važan korak u zaštiti sebe i drugih bio je način zaštite osobnog prostora u kojima su osobe obitavale te sprječavanje ulaska životinja u taj prostor. Rješenje je bilo očito - staviti prepreku na ulaz koju mogu koristiti inteligentnija bića, odnosno staviti vrata

A. Mehaničke brave

Napredak tehnologije donio je rješenja mnogih problema pa tako i problema osiguravanja osobnog prostora. Najranije brave potječu iz antičkog grada Nineveh, nekada najrazvijenijeg grada Asirije u današnjem sjevernom dijelu Mezopotamije. Takve brave kasnije su bile razvijene u egipatski tip drvene brave. Navedeni tip brave sastojao se od drvenog ključa, vijka i zubaca. Kada bi se drveni ključ stavio u bravu, ključem bi se zupci mogli dignuti što je omogućilo da se vijak izvuče van i vrata otvore. Brave koji se danas najlakše mogu prepoznati i okarakterizirati kao brave, nastale su drugom polovicom 9. stoljeća te se njihov izum prepisuje engleskim obrtnicima. Takve brave mogle su se otključati dugim metalnim ključem te su upravo one služile kao temelj za razvoj sigurnosti doma i općeniti napredak brava kakve dana poznajemo. Današnje brave, kakve se koriste skoro u svim vratima, nastale su 1805. godine u Engleskoj i djelo su američkog fizičara Abrahama O. Stansburyja. On je prvi patentirao bravu koja se otključavala kada je odgovarajući ključ podignuo zatike unutar cilindra i na taj način mogao okrenuti bravu u smjeru za otvaranje iste. Verzije kakve se danas koriste, nastale su 1848. godine kada je Amerikanac Linus Yale, Sr., napravio bravu s zatikama različite duljine te na taj način omogućio otvaranje brave jedino ključem koji je bio ispravan.

Antonio Boras je student prve godine specijalističkog studija Informatike na Tehničkom veleučilištu u Zagrebu,Vrbik 8, 10000 Zagreb. E-mail:antonio.boras@tvz.hr.

B. Elektroničke brave

Iako je kroz povijest bilo razvijeno mnogo različitih tipova brava, u ovom radu fokusirat ćemo se na brave koji se otvaraju uz pomoć struje. Elektroničku bravu okarakterizirat ćemo kao bravu koja radi kada struja prolazi kroz nju i daje snagu pokretaču koji zatim pomiče polugu koju blokira slobodno otvaranje brave te tako dopušta otvaranje. Tip brave koji se koristio u ovom radu najčešće se koristi kao brava za ulazak u zgrade te se takve brave otvaraju pritiskom na dugme portafona, drugim riječima, brava se otvara kada osoba klikne gumb kojim šalje struju do pokretača koji zatim oslobađanjem poluge otvara bravu. Brave koji se otvaraju prolaskom struje odlične su za otvaranje na manjoj udaljenosti zato jer ne zahtijevaju prisutnost osobe. Sigurnost brava razvijala se godinama kako bi se spriječila provala u zaštićeni prostor. To je samo jedan od razloga koji je doveo do sve većeg korištenja brava koje se otvaraju uz pomoć kartica i RFID tehnologije. Drugi razlog je ušteda vremena. U današnjem svijetu kada je sve umreženo i ubrzano, ljudi imaju sve manje vremena i strpljenja za otvaranje vrata ključem. Idealno rješenje je otvaranje vrata uz pomoć jednog od oblika autentifikacije. Autentifikacija je proces potvrđivanja identiteta osobe ili uređaja, dok je autorizacija proces utvrđivanja pristupa resursu. Kao što je ključ oblik autentifikacije u kojem osoba otvara vrata uz pomoć nečega što posjeduje, RFID tehnologija je, također, jednak oblik. Drugi načini otvaranje vrata uz autentifikacija može biti otvaranje vrata uz pomoć šifre. Taj način autentifikacije je „nešto što osoba zna“, a u ovom slučaju to je zaporka ili šifra. Posljednji način otvaranja vrata uz pomoć autentifikacije je onaj u kojem osoba otvara vrata zato što je potvrdila svoj osobni identitet. Drugim riječima, vrata se mogu otvoriti ako je otisak prsta osobe unesen u sustav kao validan za otvaranje vrata. U ovom radu, fokusirat ćemo se na oblik autentifikacije u kojem osoba može otvoriti vrata zbog nečega što posjeduje, a to je RFID token.

II. RFID TEHNOLOGIJA

Radio-frekvencijska identifikacija (eng. *Radio-frequency identification*) tehnologija je koja koristi elektromagnetsko polje za automatsku identifikaciju i praćenje oznaka (eng. *tag*) dodijeljenim objektu koji sprema informacije elektronski. Oznake su podijeljene na aktivne i pasivne, gdje pasivne oznake prikupljaju energiju prisutnog RFID čitača te tako ispituju radio valove koja čitač odašilje dok aktivne oznake imaju lokalni izvor energije, najčešće bateriju, te mogu obavljati svoju funkciju i na većoj udaljenosti od RFID čitača. Najveća prednost RFID oznaka je ta što ne moraju biti vidljive čitaču da bi ih isti mogao identificirati što pruža mogućnost skrivanja oznaka na neprimjetna mjesta radi veće sigurnosti. RFID tehnologija pronašla je funkcionalnost u mnogim industrijama kao što su: trgovinska industrija, farmaceutska industrija, veterina i automobilska industrija.

TABLICA I. PODJELA RFID TEHNOLOGIJE PREMA FREKVENCIJI

Frekvencija	**Duljina valova**	**Memorija**	**Udaljenost čitanja**	**Upotreba**
Niska	125 – 134.2 KHz	64 bita	0.2 – 2m	Veterina
Visoka	13.56 MHz	2048 bita	1m	Kontrola pristupa
Ultra visoka	850 – 960 MHz	96 bita	3m	Praćenje objekata
Aktivna	100 KHz – 2,45 GHz	32 bita	100m	Praćenje lokacije

U tablici 1. detaljno su opisane frekvencije koje se koriste u RFID tehnologije te za što se koja koristi.

A. Oznake

RFID tehnologija koristi oznake pričvršćene za objekt koji se treba identificirati. Oznake primaju signal od dvosmjernog radio prijemnika, koji se zovu ispitivači ili čitači, te ga zatim odašilju nazad i čitaju odgovor te oznake. Postoje tri vrste RFID oznaka – pasivne, aktivne i pasivne oznake potpomognute baterijom. Aktivne oznake u sebi sadrže bateriju koja periodički emitira svoj identifikacijski signal, dok pasivna oznaka potpomognuta baterijom unutar sebe sadrži bateriju koja se aktivira kada osjeti prisutnost RFID čitača. Najjeftinija verzija oznaka su pasivne oznake, koje su ujedno i najmanje zbog toga što u sebi ne sadrže bateriju. Takve oznake aktiviraju se kada osjete radio valove čitača, ali mogu biti operativne tek kada se emitira snaga tisuću puta jača od potrebne za slanje signala. Oznake se dijele na one koje služe samo za očitavanje, odnosno one koje već tvornički u sebi imaju pohranjen serijski broj koji se koristi kao ključ, ili mogu služiti za pohranu ili zapisivanje podataka unutar oznake. Takve oznake najčešće se koriste kao oznake na kojima se podaci mogu zapisati samo jednom, ali se iz njih može iščitavati više puta. Same oznake sastoje se od tri dijela: integrirani krug koji pohranjuje i procesira informacije, odnosno modulira i demodulira radio signale, dio koji prikuplja istosmjernu struju iz čitača signala i antenu za primanje i emitiranje signala. Iznimno je važno napomenuti kako se sve informacije pohranjuju u memoriji koja ima sposobnost zadržavanja spremljenih informacija čak i kada je struja potrebna za napajanje isključena. Sama oznaka u sebi još sadržava već ugrađenu programsku logiku ili mogućnost samog programiranja logike za procesiranje podataka i senzora. RFID čitač emitira signal da bi ispitao oznaku koja zatim odašilje signal sa svojim informacijama, koje mogu biti u obliku identifikacije, jedinstvenog serijskog broja oznake ili isprogramirani podaci poput informacija povezanih s proizvodnjom, brojem zaliha, datumom proizvodnje ili drugih detaljnijih informacija. Prednost RFID tehnologije je ta što svaka oznaka ima jedinstven serijski broj tako da čitač može raspoznati nekoliko oznaka u svom radijusu očitavanja.

B. Čitač

RFID sistemi mogu se podijeliti na dva osnovna sistema po funkciji čitača i oznake. Prvi, koji se sastoji od pasivnog čitača i aktivne oznake naziva se PRAT – eng. *Passive Reader Active Tag*. Takav sistem sastoji se od pasivnog čitača koji prima radio signale od aktivne oznake koja je potpomognuta baterijom. Radijus u kojem takav sistem djeluje je od 0 do 600 metara što omogućava relativnu fleksibilnost u pogledu zaštite i preglednosti. Drugi sistem naziva se ARPT – eng. *Active Reader Passive Tag*, te se on sastoji od aktivnog čitača i pasivne oznake. Čitač emitira signale te očitava signale, odnosno identifikaciju pasivne oznake, što znači da se oznaka aktivira tek kada je osjetila energiju čitača koju je zatim pretvorila da bi poslala identifikacijski signal. Postoji još i treći sistem RFID tehnologije, a to je ARAT – eng. *Active Reader Active Tag*. Čitač je aktivan isto kao i oznaka koja može biti potpomognuta baterijom te tako biti pasivna ili može imati integriranu bateriju u sebi te tako biti potpuno aktivna. RFID čitači su dvosmjerni radio-prijemnici, koji se mogu nazivati i ispitivačem. Prijemnik odašilje kodirani radijski signal na određenu oznaku te se tako ta oznaka „budi“ i izlazi iz stanja mirovanja. Zatim, ta oznaka odašilje signal nazad prema čitaču koji zatim očitava identifikacijski signal dobiven od strane oznake.

C. Komunikacija između čitača i oznake

Komunikacija između čitača i oznake moguća je na nekoliko različitih načina koji ovise o frekvencijskom pojasu u kojem sustav radi. Najširu i najdužu povijest primjene imaju sustavi koji koriste niskofrekventni opseg rada. Oni koriste metodu induktivnog povezivanja što znači

da čitač inducira struju u oznaci koja tako napaja samu oznaku koja prema čitaču šalje osnovne identifikacijske signale radi uspostavljanja komunikacije. Niskofrekventni RFID oznake najčešće su pasivne i ne zahtijevaju dodatni izvor napajanja ili bateriju. RFID sustavi koji rade u polju visoke frekvencije razmjenjuju podatke koristeći metodu elektromagnetske indukcije. To im daje prednost nad niskofrekventnim sustavima jer ne zahtijevaju velik broj namotaja na anteni što pridonosi izradi manje i kompaktnije antene. Frekvencija na kojoj takav sistem radi je 13,56 MHz, što je i dalje frekvencija kratkog vala, a samim time je i komunikacijska udaljenost kratka. To je prednost kod izrade sustava koja zahtjeva da se RFID tehnologija koristi na manjoj blizini poput onih u kojima se osobe i predmeti moraju potvrđivati u odnosu 1 na 1. NFC (eng. *Near Field Communication*) uređaj, koji se danas nalazi u bankovnim karticama i mobitelima također koristi visokofrekventni opseg rada. RFID sistemi koji za svoj rad koriste ultra visoku frekvenciju komuniciraju pomoću elektromagnetskih valova u opsegu od 850 MHZ do 960 MHz. Antena na čitaču usmjerava elektromagnetske valove na oznake koje se nalaze dovoljno blizu. Zatim se energija iz valova pretvara u struju unutar oznake koja se kreće prema središtu čipa tako napajajući integrirani krug. Krug se uključuje, modulira energiju s podacima iz memorije i usmjerava signal nazad prema anteni čitača. U ovakvom RFID sustavu moguće je napraviti ARAT sustav u kojem je oznaka potpomognuta baterijom te tako povećati udaljenost na kojoj je moguće očitati signal oznake. Bez obzira na valnu duljinu, svi RFID sistemi funkcioniraju da se struja stvara unutar čitača, prikuplja na anteni te se dalje usmjerava prema pasivnim oznakama. Nakon toga, energija elektromagnetskog vala dolazi do čitača gdje se skuplja sve dok je nema dovoljno da se provede do čipa unutar oznake koja zatim odašilje signal s identifikacijskim informacijama prikupljenim iz memorija unutar oznake. Signal se šalje prema anteni čitača koji zatim provjere identifikacijske informacije te postupa sukladno s programskom logikom čipa. Glavna razlike u komunikaciji su valne duljine te njihova sposobnost očitavanja na većoj, odnosno manjoj udaljenosti što zatim znači i različita primjena u stvarnom svijetu.

III. MIKROKONTROLERI

Mikrokontroleri jednostavna su računala ugrađena na pločicu s jednim integriranim krugom. Sam mikrokontroler u sebi sadrži minimalno jedan procesor, određenu veličinu memorije i dio koji se može isprogramirati prema željama i potrebama krajnjeg korisnika. Memorija za pohranu informacija na mikrokontrolerima najčešće je u formi NOR flash ili OTP ROM i uvijek dolazi sa malom količinom RAM memorije. Mikrokontroleri koriste se za „ugrađeno“ programiranje (eng. *Embedded programming*), što znači da se programiranje mikrokontrolera radi u svrhu izrade ugrađenih sustava, odnosno sustava koji imaju određenu funkciju i nisu namijenjeni niti jednom drugoj funkciji osim one za koju su isprogramirani.

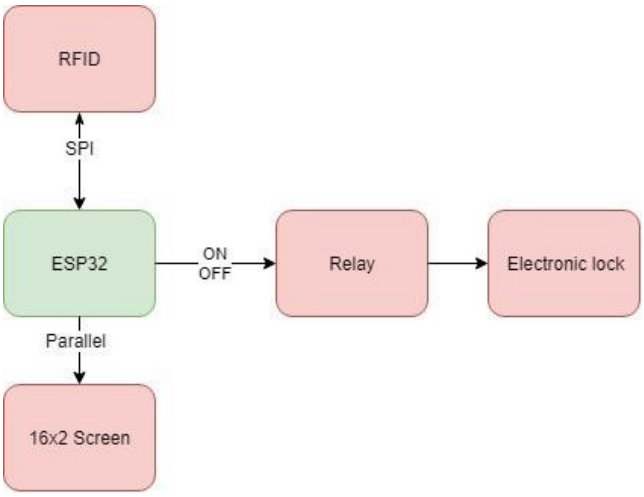
A. ESP32 mikrokontroler

ESP32 mikrokontroleri su jeftini, nisko-potrošni sistemi izgrađeni na jednom čipu te imaju integriranu WiFi i Bluetooth tehnologiju na sebi. Trenutno se ESP32 mikrokontroleri proizvode u Šangaju, Kini od strane kompanije Espressif Systems. Najveća prednost ESP32 mikrokontrolera je ta što su izrazito mali i imaju tvornički ugrađen WiFi na sebi što ih čini izrazito jednostavnima za spajanje na računala. Procesor na čipu je Xtensa dual core s 32-bitnom arhitekturom te se vrti na 160 ili 240 MHz. Taj tip procesora spada u izrazito niske potrošače energije, što je još jedna prednost samog mikrokontrolera. Radna memorija na čipu je veličine 520 KB i statična je. To znači da se koristi vrsta poluvodiča s nasumičnim pristupom (radna memorija) koja koristi bistabilni krug zatvarača za pohranu svakog bita. U suštinu, bitovi se zapisuju u memoriju sve dokle ima struje, nakon što je struja uskraćena bitovi u memoriji se gube. Zbog svoje kompatibilnosti i cijene, ESP32 mikrokontroler odabran je kao optimalan uređaj za izradu ovakvog sustava. Spajanje na sam kontroler i ubacivanje programskog koda, odvijat će se putem WiFi-ja dok će se sve druge komponente spajati na sam mikrokontroler da bi mogle zajedno tvoriti funkcionalan sustav. Drugi važan razlog odabira ovog mikrokontrolera je to što je cijena izrazito pristupačna – 70 kuna u Hrvatskoj ili 4 američkih dolara putem Internet stranica.

IV. IZRADA SUSTAVA

Mikrokontroler komunicira s RFID čitačem putem SPI komunikacije. SPI komunikacija je tip komunikacije između mikrokontrolera i perifernih uređaja putem koje se šalju informacije. Koriste se zasebne linije takta i linije podataka zajedno s odabranim linijama za odabir uređaja s kojim mikrokontroler treba komunicirati. Podatci se razmjenjuju sinkrono što znači da obje linije rade u isto vrijeme. Takt je oscilirajući signal koji govori prijemniku kada točno treba uzorkovati bitove na liniji za podatke. To može biti uzlazni ili silazni rub takta – tablica podataka određuju kakav će rub biti. Kada prijemnik prepozna rub, odmah se prebacuje na čitanje linije za podatke kako bi mogao pročitati nadolazeće podatke. Važno je naglasiti da se unutar SPI komunikacija događa vlasnik-rob (eng. *master-slave*) odnos mikrokontrolera i perifernih uređaja. Kod SPI komunikacije vlasnika uvijek ima manje nego robova, najčešće je to samo jedan mikrokontroler i više uređaja. Kada se podaci šalju s vlasnika onda je to događa na liniji za podatke zvanog MOSI (eng. *Master Out / Slave In*). Vlasnik šalje podatke dokle god ima predodređene taktova za to, a rob će povratne podatke slati putem linije za podatke zvane MISO (eng. *Master In / Slave Out*). Vrlo je važno za ovakav tip komunikacije da vlasnik zna kada i

koliko rob treba vratiti podatka nazad. Takav način rada može dovesti da mikrokontroler u isto vrijeme traži i čita podatke što znači da je SPI komunikacije „potpuni duplex“. Posljednja linija u SPI komunikaciji služi za odabiranje robova prema kojima će se komunikacija vršiti, odnosno kada se koji rob treba probuditi i primiti/poslati podatke. Ta linija koriste se kada je uključeno više robova u sustavu i može ih biti više tako da je svaka linija zasebna za svakog roba.



Slika 1. Blok šema sustava.

IV. ZAKLJUČAK

Izrada kompletnog sustava je veoma zanimljiva, kao i sam proces učenja. Još mnoga poboljšanja očekuju ovaj sustav da bi se mogao implementirati na vanjskim vratima i da bi se mogao koristiti svaki dan bez mogućnosti pada sustava. Trenutno će se ovaj sustav naći u osobnoj primjeni. Na kraju, hipoteza se može potvrditi – sustav je bilo umjereno kompleksan za izradu i pruža bolju zaštitu od osnovnih brava koje se otvaraju ključem. Isto tako, sustav pruža veću vremensku učinkovitost od brava otvorenih ključem.

LITERATURA

[1] Finkenzeller K. „RFID Handbook“, 2010.
[2] Schez P.; Monk S. „Practical Electronics for Inventors“, Fourth Edition, 2016.
[3] Culkin J.; Hagan E. Make: „Learn Electronics with Arduino“, 2017.
[4] Chubb Locks, History of Locks, Encylopaedia of Locks and Builders Hardware, 1958. URL: https://web.archive.org/web/20060923082741/http://www.chubblocks.co.uk/historyof locks.html (3. kolovoz 2019.)
[5] Cardullo M. Genesis of the Versatile RFID Tag, 21. travnja, 2003. URL: https://www.rfidjournal.com/articles/view?392 (3. kolovoz 2019.)

Uticaj metode izračunavanja korelacije na performanse algoritama mašinskog učenja

Anđela Antić

Sadržaj – U ovom radu razmatra se uticaj metode izračunavanja korelacije na performanse algoritama mašinskog učenja: taj uticaj je ispitan na različitim modelima, a u ovom radu prikazan je primer formiranja modela mašinskog učenja koji vrše predikciju kvara na sistemu za pritisak vazduha kamiona. Modeli su formirani pomoću programskog jezika Python.

I. UVOD

Mašinsko učenje predstavlja jedan od glavnih nosilaca četvrte industrijske revolucije [1]. U ovom radu, pomoću metoda mašinskog učenja, formirani su modeli za predikciju kvara na sistemu za pritisak vazduha kod kamiona. Kako bi formirani modeli vršili što tačniju predikciju, izvršena je analiza početnog skupa podataka i na osnovu koeficijenata korelacije između ciljne kolone i pojedinačnih parametara formiran je adekvatan trening i test skup. Korišćene su tri metode računanja korelacije i to Kendall-ova, Spearman-ova i Pearson-ova metoda.

Modeli su formirani na osnovu skupa podataka preuzetih sa UCI ML Repository [2]. Podaci su podeljeni u trening i test skupove. Posmatra se kolona *class* koja sadrži informaciju o tome da li je pokvaren neki deo sistema za pritisak vazduha (vrednost *„pos“*) ili ne (vrednost *„neg“*). Imena parametara, odnosno ostalih kolona na osnovu kojih je potrebno izvršiti predikciju kolone *class* su kodirana kako ne bi došlo do zloupotrebe podataka, tako da su u ovom slučaju nepoznata. Glavni zadatak ovog rada jeste predviđanje vrednosti u ciljnoj koloni na takav način da ukupni troškovi usled pogrešnih pretpostavki budu što je moguće manji. Ukupni troškovi računaju se po formuli:

$$\text{Cena} = \text{Cena}_1 \cdot \text{Pogrešno1} + \text{Cena}_2 \cdot \text{Pogrešno0} \qquad (1)$$

gde Cena_1 = 10,00$ predstavlja troškove usled nepotrebne provere kamiona bez kvara, dok Cena_2 = 500,00$ predstavlja troškove naknadnog servisiranja kamiona sa kvarom za koji je pogrešno pretpostavljeno da je ispravan.

II. PRELIMINARNA ANALIZA PODATAKA

Pre samog formiranja modela mašinskog učenja, potrebno je izvršiti analizu podataka, ispitati odnose među osobinama i na kraju izvršiti adekvatnu selekciju podataka koji će se koristiti pri formiranju modela. Trening skup sadrži 60.000 podataka u svakoj koloni, dok test skup sadrži 16.000 podataka. Tipovi podataka koji su prisutni u trening i test skupu su celobrojni, realni i stringovi. Za početak, potrebno je odrediti koje će se kolone, odnosno parametri, koristiti u predikciji. Odabir kolona vrši se na osnovu toga u kolikoj meri utiču na ciljnu kolonu koja sadrži informaciju o tome da li kamion poseduje kvar ili ne.

Pre konverzije podataka u odgovarajuće tipove, razmotreno je da li i u kojoj meri u kolonama postoje podaci koji nisu upisani. Broj neupisanih podataka je veliki. Iz tog razloga, kako bi se omogućila što tačnija analiza podataka, kroz *for* petlju izbačene su kolone koje sadrže više od deset hiljada neupisanih trening podataka iz trening i test skupa, a zatim su prazna mesta u preostalim kolonama popunjena srednjom vrednošću u njima upisanih podataka. Pored toga, ispitano je i da li neka od kolona ima varijansu 0, odnosno da li postoji kolona u kojoj sve vrste imaju istu upisanu vrednost. Utvrđeno je da postoji jedna kolona sa ovom osobinom i ona je izbačena iz trening i test skupa.

Nakon toga, pažnja je preusmerena na ciljnu kolonu *class*. Trening i test skup sadrže značajno manji broj vrsta sa vrednošću *„pos“* od broja vrsta sa vrednošću *„neg“*. Na ovu činjenicu je potrebno obratiti posebnu pažnju iz razloga što postoji mogućnost da modeli mašinskog učenja daju mnogo veću prednost klasi ispravnih kamiona pri predikciji i time pogrešno klasifikuju veliki broj neispravnih kamiona kao ispravne.

```
print(APS['class'].value_counts())
```

```
neg    59000
pos     1000
Name: class, dtype: int64
```

Kako su vrednosti u koloni *class* tipa string, potrebno je izvršiti njihovu konverziju. Vrednosti *„neg“* i *„pos“* mapirane su vrednostima 0 i 1, respektivno. Nakon mapiranja, omogućeno je računanje koeficijenata korelacije između ciljne i ostalih kolona. Pomoću ovih vrednosti može se utvrditi u kojoj meri neka kolona utiče na vrednosti u ciljnoj koloni. Ukoliko koeficijent korelacije dve promenljive ima vrednost blisku nuli, može se zaključiti da je zavisnost između ovih promenljivih jako mala. Imajući to na umu određen je vektor *korelacija* koji sadrži vrednosti korelacija svih kolona u odnosu na kolonu *class*. Najpre su koeficijenti korelacije izračunati Kendall-ovom metodom, a potom Pearson-ovom i Spearman-ovom metodom. Formirani vektor se koristi kako bi se kroz *for* petlju izbacile sve kolone sa vrednošću korelacije u odnosu na ciljnu kolonu koja je bliska nuli. Jako je važno naglasiti

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

da ovom koraku treba pristupiti oprezno jer ciljna kolona možda nema direktno veliku vrednost korelacije u odnosu na neku osobinu, ali postoji mogućnost da kombinacija nekoliko osobina utiče u velikoj meri na ciljnu iako pojedinačni rezultati to ne pokazuju. Ukoliko krajnji rezultati predikcije nisu zadovoljavajući, potrebno je pažljivije proučiti međusobni uticaj kolona.

Pomoću funkcije *heatmap(APS.corr())* proračunata je matrica korelacije na osnovu koje je uočeno da postoje parovi kolona čija je korelacija jednaka jedinici, što znači da su podaci u ovim kolonama potpuno identični ili proporcionalni. U ovakvim situacijama preporučljivo je da se po jedna kolona iz svakog ovakvog para izbaci kako bi se formirao što bolji model mašinskog učenja i iz tog razloga iz trening i test skupa izbačene su odgovarajuće kolone.

III. FORMIRANJE MODELA MAŠINSKOG UČENJA ZA PREDIKCIJU KVARA NA KAMIONIMA

Nakon što je proverena zavisnost između ciljne promenljive i ostalih parametara i nakon izbacivanja nepodesnih kolona, moguće je započeti formiranje modela mašinskog učenja. Za početak, na osnovu trening skupa formira se model, a nad test skupom proverava se tačnost modela. Pomoću funkcije *DecisionTreeClassifier()* i trening skupa formiran je model klasifikacionog stabla odlučivanja. Nakon toga izvršena je predikcija ciljne promenljive *„class“* na osnovu podataka iz test skupa. Kako bi se izvršila evaluacija rezultata, vrednosti dobijene predikcijom (*predvidjanje_stablo*) upoređene su sa stvarnim vrednostima iz test skupa (*y_test*) i formiran je klasifikacioni izveštaj, kao i matrica konfuzije.

```
#Stablo odlucivanja
modelstablo = DecisionTreeClassifier(random_state=41)
modelstablo.fit(X_train, y_train)
predvidjanje stablo=modelstablo.predict(X test)
print(classification report(y test, predvidjanje stablo))
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     15625
           1       0.81      0.69      0.75       375

    accuracy                           0.99     16000
   macro avg       0.90      0.84      0.87     16000
weighted avg       0.99      0.99      0.99     16000

accuracy_stablo = accuracy_score(y_test,
predvidjanje_stablo)
print(confusion matrix(y test, predvidjanje stablo))

[[15565    60]
 [  115   260]]

tn_stablo, fp_stablo, fn_stablo, tp_stablo =
confusion_matrix(y_test, predvidjanje_stablo).ravel()
cost_stablo = Cost_1*fp_stablo+Cost_2*fn_stablo
print(accuracy_stablo)

0.989062

print(cost_stablo)

58100
```

Na osnovu rezultata, može se primetiti da veliki problem koji se javlja kod ovog modela jeste prednost koju daje klasi „0“ odnosno klasi kamiona koji nemaju kvar. Iz tog razloga, za veliki broj kamiona pogrešno se pretpostavlja da su ispravni. Kako bi predikcija bila izvršena na što bolji način i uz što manje troškove razmotreno je kakve bi rezultate dao model slučajne šume kreiran pomoću funkcije *RandomForestClassifier()*. Broj klasifikacionih stabala koja se koriste pri proceni je 300.

```
#Slucajna suma
modelsuma = RandomForestClassifier(n estimators=300,
random state=41, n jobs=-1)
modelsuma.fit(X_train, y_train)
predvidjanje_suma = modelsuma.predict(X_test)
print(classification_report(y_test, predvidjanje_suma))
              precision    recall  f1-score   support

           0       0.99      1.00      1.00     15625
           1       0.93      0.73      0.82       375

    accuracy                           0.99     16000
   macro avg       0.96      0.86      0.91     16000
weighted avg       0.99      0.99      0.99     16000

accuracy_suma = accuracy_score(y_test, predvidjanje_suma)
print(confusion_matrix(y_test, predvidjanje_suma))

[[15606    19]
 [  103   272]]

tn_suma, fp_suma, fn_suma, tp_suma =
confusion_matrix(y_test, predvidjanje_suma).ravel()
cost_suma = Cost_1*fp_suma+Cost_2*fn_suma
print(accuracy_suma)

0.992375

print(cost_suma)

51690
```

Rezultati su pokazali da novoformirani model vrši klasifikaciju kamiona sa nešto većom tačnošću. Međutim, kao i stablo odlučivanja, i slučajna šuma je takođe veću prednost dala klasi „0“ i na taj način pogrešno je klasifikovala ogroman broj neispravnih kamiona kao ispravne. Ovaj problem dovodi do pojave velikih troškova usled pogrešnih pretpostavki zbog propuštenog servisiranja pogrešno klasifikovanih neispravnih kamiona. Kako bi se izvršila minimizacija ukupnih troškova, potrebno je povećati verovatnoću da se kamion klasifikuje kao kamion sa kvarom na sistemu za pritisak vazduha. Iz tog razloga formiran je model slučajne šume sa regresionim stablima, pri čemu je broj estimatora isti kao u prethodnom slučaju. Prag na osnovu koga se nekom rezultatu u opsegu [0,1] dodeljuje vrednost 0 ili 1, podešava se pomoću funkcije *dodeli_vrednost(predvidjanje,prag)*. Za vrednost praga potrebno je odabrati vrednost manju od 0.5 kako bi se smanjio broj kamiona za koje se pogrešno pretpostavlja da ne poseduju kvar na sistemu za pritisak vazduha.

```
def dodeli_vrednost(predvidjanje,prag):
    return [1 if y > prag else 0 for y in predvidjanje]

#Slucajna suma 1
modelsuma = RandomForestRegressor(n_estimators=300,
random_state=41, n_jobs=-1,oob_score=True)
modelsuma.fit(X train, y train)
```

Na ovaj način, vrednosti dobijene predikcijom prvo prolaze kroz funkciju *dodeli_vrednost(predvidjanje,prag)* koja je tako definisana da vraća vrednost 1 ukoliko je vrednost rezultata veća od 0.03 i vrednost 0 u suprotnom.

```
predvidjanje_suma1 = dodeli_vrednost
(modelsuma.predict(X_test),0.03)
print(classification_report(y_test, predvidjanje_suma1))
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     15625
           1       0.41      0.97      0.58       375

    accuracy                           0.97     16000
   macro avg       0.70      0.97      0.78     16000
weighted avg       0.99      0.97      0.97     16000

accuracy_suma1 = accuracy_score(y_test,
predvidjanje_suma1)
print(confusion_matrix(y_test, predvidjanje_suma1))

[[15104   521]
 [   13   362]]

tn_suma1, fp_suma1, fn_suma1, tp_suma1 =
confusion_matrix(y_test, predvidjanje_suma1).ravel()
cost suma1 = Cost 1*fp suma1+Cost 2*fn suma1
print(accuracy suma1)

0.966625

print(cost_suma1)

11710
```

Iako je korišćenje ove funkcije dovelo do smanjenja tačnosti modela za 2.575%, ukupni troškovi smanjeni su 4.414 puta. Dakle, funkcija *dodeli_vrednost()* sa definisanom vrednošću praga 0.03 dala je željene rezultate i omogućila je značajno smanjenje broja kamiona sa kvarom za koje se pogrešno pretpostavlja da ne poseduju isti. Kako bi se još više smanjili troškovi, formirani su modeli koji za parametar *min_samples_leaf* imaju definisane vrednosi 50, 70, 90 i 100. Najtačnijim od modifikovanih modela se pokazao model sa parametrom *min_samples_leaf* = 90.

```
#Slucajna suma 4
modelsuma = RandomForestRegressor(n_estimators=300,
random_state=41, n_jobs=-
1,oob_score=True,min_samples_leaf=90)
modelsuma.fit(X_train, y_train)
predvidjanje_suma4 = dodeli_vrednost
(modelsuma.predict(X test),0.03)
print(classification report(y test, predvidjanje suma4))
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     15625
           1       0.36      0.98      0.53       375

    accuracy                           0.96     16000
   macro avg       0.68      0.97      0.76     16000
weighted avg       0.98      0.96      0.97     16000

accuracy_suma4 = accuracy_score(y_test,
predvidjanje suma4)
print(confusion matrix(y test, predvidjanje suma4))

[[14986   639]
 [    8   367]]

tn_suma4, fp_suma4, fn_suma4, tp_suma4 =
confusion_matrix(y_test, predvidjanje_suma4).ravel()
cost_suma4 = Cost_1*fp_suma4+Cost_2*fn_suma4
print(accuracy_suma4)

0.959563

print(cost_suma1)

10390
```

Ovaj model je dao i najmanje troškove od 10.390,00 $

IV. UTICAJ METODE IZRAČUNAVANJA KORELACIJE NA PERFORMANSE FORMIRANIH MODELA

Već je rečeno da je u prethodno opisanom primeru vektor *korelacija*, koji sadrži vrednosti korelacija svih kolona u odnosu na ciljnu kolonu, formiran korišćenjem Kendall-ove metode za računanje korelacionih koeficijenata. Konačni trening i test skup sadrže parametre koji pokazuju dovoljno veliku korelaciju u odnosu na ciljnu kolonu. U cilju formiranja što boljeg i ekonomičnijeg modela, razmotreno je da li su ukupni troškovi usled pogrešnih pretpostavki manji ukoliko se vektor *korelacija* odredi na osnovu Spearman-ove ili Pearson-ove metode. Pearson-ova metoda najčešće se koristi kako bi se utvrdio stepen korelacije između dve numeričke promenljive sa normalnom raspodelom. Može se koristiti u slučaju kada je jedna od promenljivih dihotomna. Sa druge strane, Spearman-ova i Kendall-ova metoda najčešće se koriste kako bi se izračunao stepen korelacije između rangiranih promenljivih. Rezultati dobijeni primenom navedenih metoda prikazani su tabelarno. Za Kendall-ovu metodu formirana je tabela I u kojoj je prikazana tačnost svakog modela, kao i ukupni troškovi usled pogrešnih pretpostavki. Rezultati pri korišćenju Spearman-ove i Pearson-ove metode prikazani su u tabelama II i III respektivno.

TABELA I
TAČNOST MODELA I UKUPNI TROŠKOVI (KENDALL)

	Tačnost	Ukupni troškovi
Stablo odlučivanja	98.9062 %	58.100,00 $
Slučajna šuma	99.2375 %	51.690,00 $
Slučajna šuma sa izmenjenim pragom	96.6625 %	11.710,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=50	95.8812 %	10.510,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=70	95.9125 %	10.460,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=90	95.9563 %	10.390,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=100	95.9187 %	10.450,00 $

TABELA II
TAČNOST MODELA I UKUPNI TROŠKOVI (SPEARMAN)

	Tačnost	Ukupni troškovi
Stablo odlučivanja	98.8187 %	58.240,00 $
Slučajna šuma	99.2375 %	52.180,00 $
Slučajna šuma sa izmenjenim pragom	96.6813 %	11.190,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=50	95.89375 %	10.490,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=70	95.9 %	10.480,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=90	95.975 %	10.360,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=100	95.9125 %	10.460,00 $

TABELA III
TAČNOST MODELA I UKUPNI TROŠKOVI (PEARSON)

	Tačnost	Ukupni troškovi
Stablo odlučivanja	98.8687 %	50.810,00 $
Slučajna šuma	99.2375 %	53.650,00 $
Slučajna šuma sa izmenjenim pragom	96.8437 %	11.910,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=50	95.95 %	9.910,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=70	95.875 %	9.540,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=90	95.875 %	9.540,00 $
Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=100	95.9063 %	9.490,00 $

Na osnovu rezultata prikazanih u tabelama I i II može se zaključiti da je pri korišćenju Kendall-ove i Spearman-ove metode računanja korelacije razlika između krajnjih rezultata mala. Sa druge strane, rezultati prikazani u tabeli III pokazuju da je korišćenje Pearson-ove metode dodatno smanjilo ukupne troškove i dalo model koji je izvršio predikciju sa najmanjim ukupnim troškovima od 9.490,00$. Za svaku od metoda izdvojeni su modeli koji su dali najmanje troškove usled pogrešnih pretpostavki i za svaki od njih u tabeli 4. prikazan je broj kamiona pogrešno klasifikovanih kao kamioni sa kvarom, broj kamiona pogrešno klasifikovanih kao kamioni bez kvara, kao i ukupni troškovi usled pogrešnih pretpostavki.

TABELA IV
NAJBOLJI REZULTATI ZA SVAKU OD METODA IZRAČUNAVANJA KORELACIJE

	Pogrešne pretpostavke		Ukupni troškovi
	Nema kvara	Postoji kvar	
Pearson-ova metoda Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=100	649	6	9.490,00$
Spearman-ova metoda Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=90	636	8	10.360,00$
Kendall-ova metoda Slučajna šuma sa izmenjenim pragom i *min_samples_leaf*=90	639	8	10.390,00$

V. ZAKLJUČAK

U ovom radu obrađen je problem predikcije kvara na sistemu za pritisak vazduha kamiona uz što manje dodatne troškove. Pre formiranja modela mašinskog učenja izvršena je preliminarna analiza i pretprocesiranje podataka. Pomoću Kendall-ove, Spearman-ove i Pearson-ove metode izračunavanja korelacije određeno je koji će se parametri koristiti za formiranje i treniniranje modela i predikciju na osnovu test skupa. Korišćenje Kendall-ove i Spearman-ove metode omogućilo je kreiranje modela mašinskog učenja koji su predikciju izvršili uz ukupne troškove usled pogrešnih pretpostavki od 10.390,00$ i 10.360,00$ respektivno. Pearson-ova metoda omogućila je formiranje modela koji za iste vrednosti parametra *min_samples_leaf* = 90 daje ukupne troškove od 9.540,00$. Na osnovu dobijenih rezultata zaključuje se da metoda izračunavanja korelacije može imati veliki uticaj na performanse modela mašinskog učenja. Ukupni troškovi dodatno su smanjeni na 9.490,00$ postavljanjem parametra *min_samples_leaf* = 100. Iz tog razloga, može se zaključiti da je u ovom slučaju najpodesnije koristiti Pearson-ovu metodu izračunavanja korelacije i model slučajne šume sa izmenjenim pragom i vrednošću parametra *min_samples_leaf* =100.

ZAHVALNICA

Autor se zahvaljuje profesoru doktoru Marku Milojkoviću.

LITERATURA

[1] Li Ping Chu, “Data Science for Modern Manufacturing”, United States of America, O’Reilly Media, 2016.
[2] (2016. Sep.) UCI Machine Learning Repository: “APS Failure at Scania Trucks Data Set“, dostupno na: https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks

Fazna kontrola pomoću TRIAC-a

Aleksandar Jevtić, Stefan Ilić, Nikola Đikić, Goran Ristić

Sadržaj – U ovom radu je obrađena tema funkcionisanja i primena snažne elektronske komponente trijak, odnosno, triodnog poluprovodničkog prekidača naizmenične struje (engl. *TRIode Alternating Current semiconductor switch-TRIAC*) . Prikazana je praktična realizacija elektronskog kola kojim upravlja mikrokontroler ATmega328P-PU.

Napomena: Prikazana je upotreba napona koji su opasni po život. Ne pokušavati realizaciju datog kola bez prisustva stručnog lica.

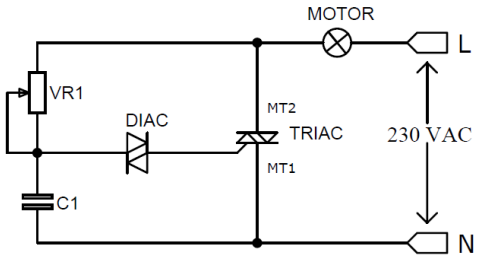
I. FAZNA KONTROLA

Fazna kontrola, takođe nazvana fazno odsecanje ili „regulacija faznog ugla“, je metoda za ograničavanje snage primenjena na naizmenične napone. Postiže se modulacijom snažnih prekidačkih komponenti kao što je trijak, koje su redno vezane sa potrošačem na izlazu i izvodi u unapred određenoj fazi primenjenog talasnog oblika. Često se koristi za kontrolu vrednosti napona, struje ili snage koju napajanje predaje potrošaču. Ovo se čini kako bi se razvila srednja vrednost snage na potrošaču. Ako napajanje ima jednosmerni izlaz, njegova vremenska baza nije bitna u odlučivanju kada uključiti ili isključiti napajanje, jer će vrednost koja će se uključiti i isključiti biti impulsirana. Fazni kontroleri uzimaju svoje ime po tome što pokreću impulsni izlaz u određenoj fazi od ulaznog signala. U osnovi, fazni kontroler je kontroler koji može sam da se sinhronizuje sa modulacijom prisutnom na ulazu.

II. FAZNA KONTROLA POMOĆU TRIJAKA I DIJAKA

Jedna od najčešćih upotreba trijaka u prekidačkim kolima jeste upotreba trijaka za kontrolu faze dovedenog naizmeničnog napona (Slika 1). U ovoj konfiguraciji trijak se koristi za kontrolu napona na potrošaču, samim tim i snage koja se razvija na potrošaču. U datom primeru, upotrebljen je motor za naizmenične struje. Za potrošače naizmenične struje potrebna je kontrola pozitivne i negativne poluperiode, koja se ostvaruje upotrebom dijaka. Naziv dijak dolazi od engleske skraćenice DIAC (eng. *Diode Alternating Current switch*) za diodni naizmenični strujni prekidač, a naziva se još i dvosmerni diodni prekidač - BDS (eng. *Bilateral Diode Switch*). Dijak počinje da propušta značajnu struju kada je razlika potencijala na njegovim krajevima veća od 30 V. Ovaj napon se podešava izborom otpornosti potenciometra VR1 i kondenzatora C1.

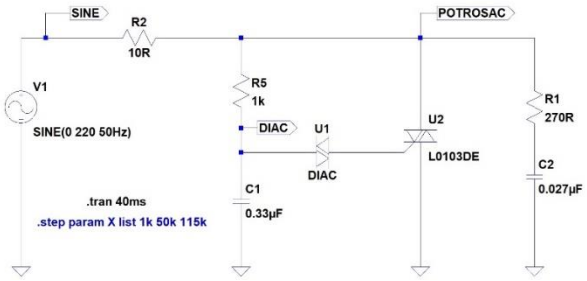
Na početku svake poluperiode, kondenzator C1 počinje da se puni preko potenciometra VR1. Ovaj proces punjenja kondenzatora se odvija sve dok se ne dostigne napon na elektrodi kondenzatora koji je dovoljno visok da okine dijak. Dijak tada prelazi u provodno stanje i uključuje trijak. Kondenzator se u tom trenutku prazni preko dijaka i trijaka, dok trijak ostaje uključen do završetka trajanja poluperiode. Trijak se potom automatski isključuje i započinje se ponovni ciklus punjenja i pražnjenja kondenzatora C1. Treba napomenuti, trijak nije idealno simetrična komponenta, samim tim vreme uključenja i isključenja nije iste dužine trajanja.



Slika 1. Fazna kotrola upotrebom dijaka i trijaka.

III. SIMULACIJA FAZNE KONTROLE U LTSPICE-U

Prikazano je prethodno objašnjeno kolo u programu LTSpice (Slika 2). Parametar koji se menja je otpornik R5 u koracima od 1 kΩ, 50 kΩ i 115 kΩ. Posmatramo napon na otporniku R2. U kolu se dodatno nalaze otpornik R1 i C2 koji imaju namenu prigušenja neželjenih oscilacija.



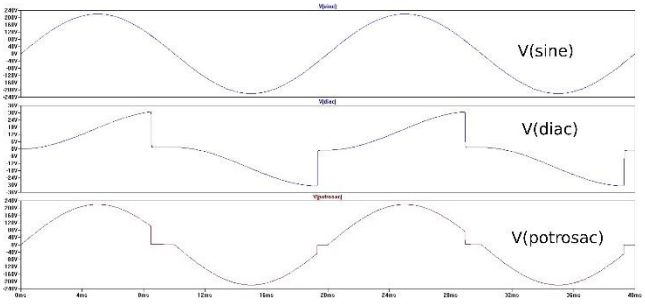
Slika 2. Simulirano kolo za faznu kontrolu.

Signal oznake V(sine) predstavlja napajanje iz mreže od 230 VAC. Signal oznake V(potrošač) predstavlja napon na potrošaču R2, a signal oznake V(diac) napon okidanja na

Aleksandar Jevtić, Stefan Ilić i Nikola Đikić su studenti osnovnih akademskih studija na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.s.jevtic@gmail.com, stefan.kocanac@gmail.com, nidjikic@gmail.com

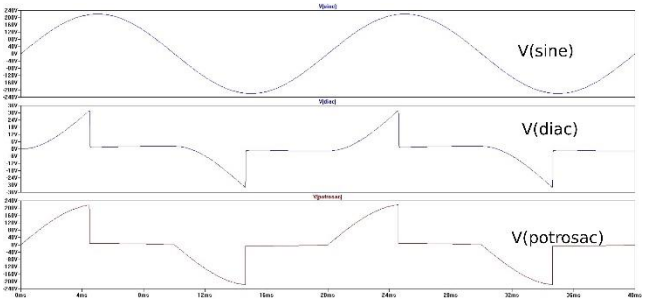
Goran Ristić je redovni profesor na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: goran.ristic@elfak.ni.ac.rs

trijaku. Za najveću vrednost otpornika R5 od 115 kΩ, kondenzatoru C1 je potrebno najviše vremena da se napuni i samim tim okine dijak, tako da trijak u ovom slučaju provodi najmanje vremena u provodnom režimu (Slika 3).



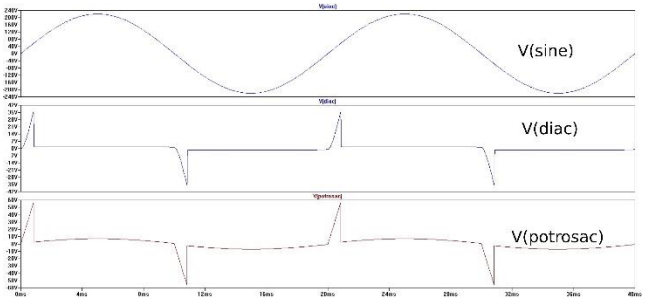
Slika 3. Signali u kolu za najveću vrednost otpornika R5 = 115 kΩ (100%).

Na narednoj slici je prikazan sličaj za R5 = 50 kΩ (Slika 4).



Slika 4. Signali u kolu za vrednost otpornika R5 = 50 kΩ (50%).

I zadnji slučaj kada je R5 = 1 kΩ (Slika 5).



Slika 5. Signali u kolu za vrednost otpornika R5 = 1 kΩ (≈ 0%).

Može se uočiti da što je vrednost otpornika R5 manja, kondenzator C1 se brže puni, samim tim dijak okida trijak ranije. Prema tome, snaga koja je proporcionalna površini ispod krive, koja opisuje napon na potrošaču, je manja.

IV. Električna šema sistema

Projektovani sistem se sastoji od četiri bloka prikazanih na (Slika 11):

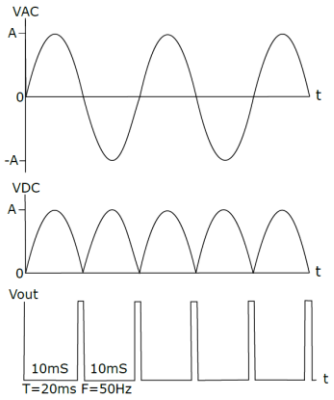
A. Blok za Raspored Faze;
B. Blok za Napajanje i Detekciju Nule;
C. Blok Mikrokontrolera;
D. Blok Trijaka.

A. Blok za Raspored Faze

Ovaj blok se sastoji od četiri LE diode koje imaju za namenu da pokažu raspored vodova faze, vodova uzemljenja i neutralnog voda prilikom uključivanja u utičnicu. Kada su uključene LE diode LED1 i LED3, imamo indikaciju da raspored datih vodova odgovara onom na šemi. U slučaju da LED3 nije uključena, potrebno je okrenuti utikač. LE dioda LED2 će biti uključena samo ukoliko nemamo uzemljenje u utičnici. LE Dioda LED4 će biti uključena u slučaju izgorelog osigurača F1.

B. Blok za Napajanje i Detekciju Nule (ZCP)

Blok za napajanje ima dve namene. Prvi, da obezbedi stabilan izvor napona od 5 VDC za mikrokontroler, a drugi da detektuje nulti prelaz faze [2]. Transformator T1 smanjuje napon na 12 VAC, koji se potom odvodi na Grecov spoj. Dobijamo jednosmerni pulsirajući napon koji uključuje LE diodu u optokapleru 4N25 i okida invertor čiji se signal (ZCP DETEKCIJA) šalje na na mikrokontroler. Dioda D1 odvaja Grecov most od kondenzatora C1 da se ne bi izgubila korisna informaciju u signalu. Ostatak kola je standardni L7805 naponski stabilizator na 5 VDC. Opisani signali su grafički prikazani (Slika 6).



Slika 6. Signali u kolu za detekciju nule.

C. Blok Mikrokontrolera

Blok mikrokontrolera vrši obradu primljenog signala o nultom prelazu i kontrolu samog trijaka. Upotrebljen je mikrokontroler ATmega328P-PU u kućištu DIP-28. Na pinu pod rednim brojem četiri koji nosi oznaku INT0, vrši se detekcija impulsa koji dolaze od optokaplera 4N25. Sam rad sistema se omogućava uključivanjem prekidača SW1, koji je povezan na unutrašnji pull-up otpornik. Indikaciju uključenog sistema vrši LE dioda LED5. Ulogu otpornika

koji određuje vreme okidanja trijaka vrši potenciometar POT1 koji preko unutrašnjeg ADC konvertora pretvara vrednost napona u vreme okidanja trijaka. Okidni impulsi se šalju na pin 11 koji nosi iznaku PD5. Dodatni materijal za ovaj blok i kod biće priložen u literaturi [3].

D. Blok Trijaka

Jedina razlika između ovog bloka i prethodno objašnjenih šema na početku rada (Slika 2) jeste u tome da se dijak ne kontroliše rednim RC kolim već optokaplerom koji kontroliše okidanje dijaka KR100. Impulsna pobuda se dovodi na LE diodu optokaplera koji okida ugrađeni dijak.

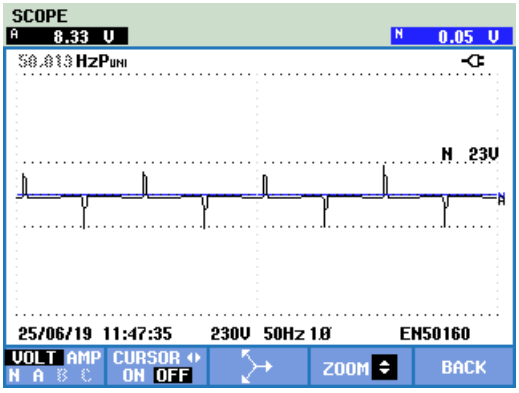
Primećujemo oznake MT2 i MT1 na simbolu za trijak BTA16-600B, kao i to da je gejt trijaka bliži izvodu MT1 (Slika 10). Ovakvo označavanje implicira da je tehnološkim postupcima gejt trijaka fizički unutar same strukture bliži izvodu MT1. Zarad lakšeg uključivanja trijaka je potrebno povezati izvod MT2 sa fazom a izvod MT1 sa neutralnim ulazom, u protivnom pod određenim uslovima može doći do oštećenja trijaka.

V. Rezultati merenja

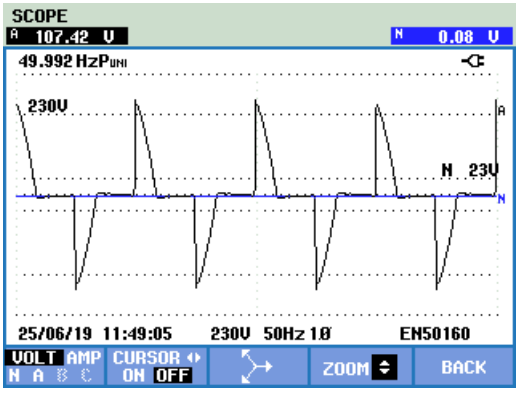
Kolo za kontrolu faze je praktično realizovano i izvršena su merenja pomoću analizatora snage marke FLUKE 435 koji je paralelno priključen na potrošač. Iz datih merenja na slikama 7,8 i 9 vide se dva glavna problema:

1. Nejednako vreme uključivanja i isključivanja dijaka i trijaka;
2. Distorzija napona na potrošaču usled nelinearnosti komponenata.

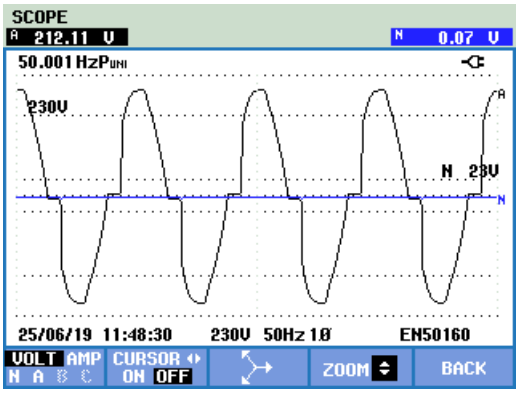
Predstavljen je napon na potrošaču za slučajeve gde se potenciometar (otpornik R5) nalazio kao u simulaciji na 100% svoje otpornosti, 50% svoje otpornosti i 0% svoje otpornosti.



Slika 7. Izmereni napon na potrošaču za 100% vrednosti potenciometra.

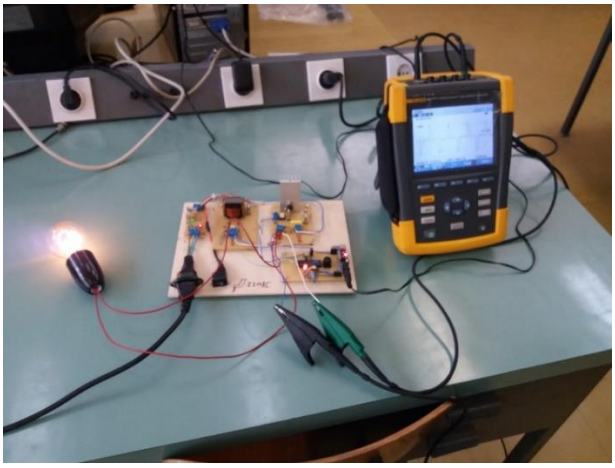


Slika 8. Izmereni napon na potrošaču za 50% vrednosti potenciometra.

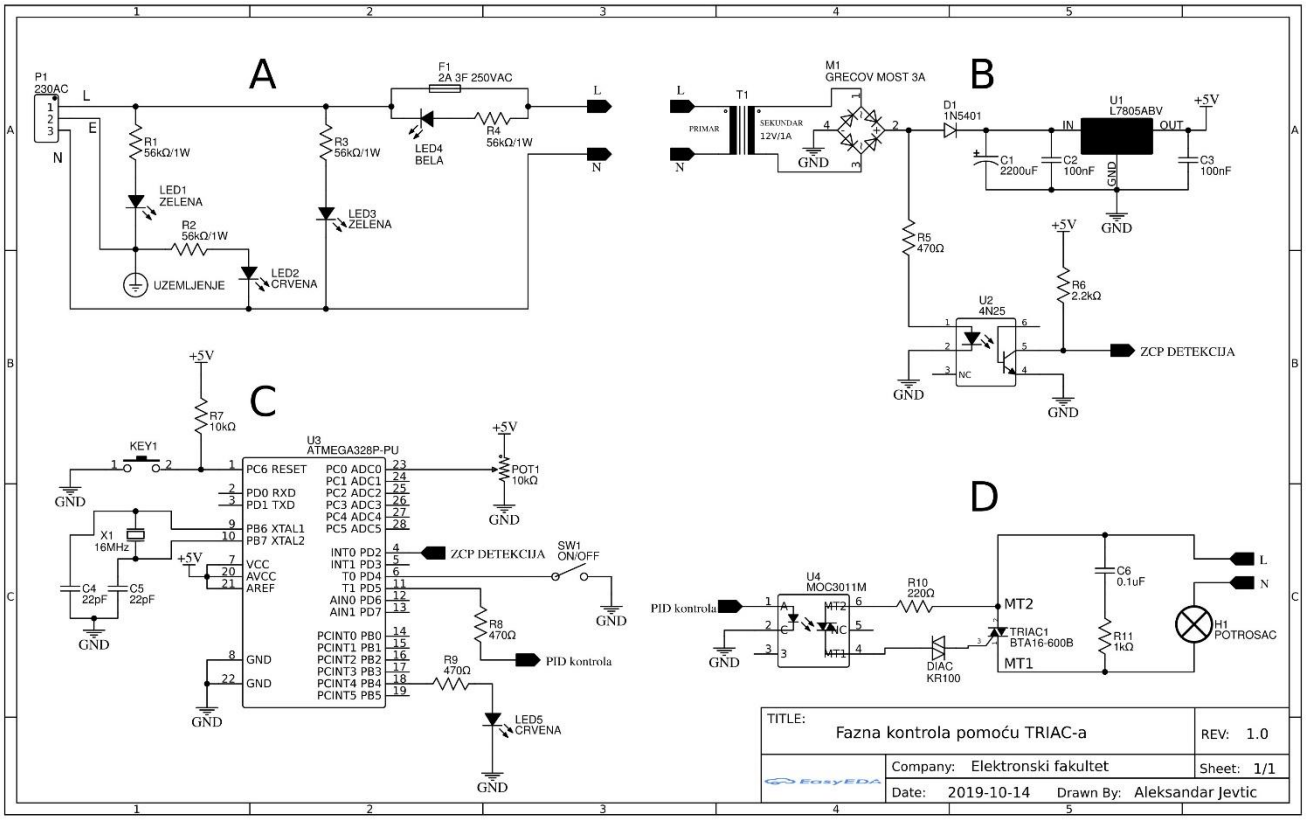


Slika 9. Izmereni napon na potrošaču za 0% vrednosti potenciometra.

Na narednoj fotografiji predstavljeno je eksperimentalno kolo priključeno na analizator snage (Slika 10).



Slika 10. Eksperimentalno kolo u Laboratoriji za merenja.



Slika 11. Kompletan šematski dijagram realizovanog kola.

VI. ZAKLJUČAK

Ovaj rad prikazuje pre svega postupak realizacije sistema koji se bazira na faznoj kontroli i princip razumevanja datog problema. Objašnjena je specifična primena snažne poluprovodniče komponente trijaka kroz konkretan primer i omogućava adaptaciju kola za specifičnu namenu. Prednost ovakve fazne kontrole se ogleda u jednostavnosti realizacije.

LITERATURA

[1] Z. Prijić i A. Prijić, „Uvod u poluprovodničke komponente i njihovu primenu“, Univerzitet u Nišu, Elektronski fakultet, 2014.

[2] Princip rada ZCP kola za detekciju: http://www.bristolwatch.com/ele2/zero_crossing.htm

[3] Dodatni materijal za kod mikrokontrolera: http://www.bristolwatch.com/arduino/arduino1.htm

[4] Upotreba trijaka: https://www.electronics-tutorials.ws/power/triac.html

[5] Upotreba dijaka: https://www.electronics-tutorials.ws/power/diac.html

Digitalna mapa kulturno-istorijskih znamenja grada Niša

Aleksandar Antić, Nina Stojiljković, Anđela Tasić, Anđela Milosavljević

Sadržaj – U ovom radu je prikazana digitalna mapa kulturno-istorijskih znamenja grada Niša.

I. UVOD

Zamisao našeg tima bila je da istoriju, kao humanističku nauku, digitalizujemo i uvedemo u svet savremenih tehnologija. U pitanju je multidisciplinarni projekat studenata istorije Filozofskog fakulteta u Nišu pod mentorstvom doc.dr Vladimira Aleksića. Saznanja o kulturnim spomenicama našeg grada, kao i njegove okoline pretočena su u ArcGIS (*geographic information system*) informacioni sistem. Reč je o programu koji omogućava rad sa mapama i geografskim informacijama, njihovo kreiranje, analiziranje, deljenje itd.

Izrada digitalnih mapa predstavlja budućnost metodologije pre svega istorijske geografije, kao pomoćne istorijske nauke. Na sednici Veća departmana za istoriju održanoj 14.11.2018. godine usvojen je predlog doc.dr Vladimira Aleksića da se internom departmanskom projektu priključi i program koji podrazumeva izradu Digitalne mape kulturno-istorijskih znamenja grada Niša.
Obuku studentima održao je doc.dr Vladimir Aleksić i objasnio kako da naprave svoju digitalnu mapu sa srednjevekovnim manastirima u okolini Niša i verskim objektima u gradskom jezgru, kao i kulturnim spomenicima iz različitih epoha.

II. PROCES IZRADE MAPE

A. Odabir objekata

Naša aktivnost usmerena je prvobitno na odabir objekata koji će biti ubačeni u mapu. Doneta je odluka da objekti budu grupisani ili po principu geografskog položaja ili istorijske epohe kojoj pripadaju. Za sada su birani spomenici na teritoriji grada Niša, ali u nastavku rada planiramo da mapiramo i značajne objekte iz ostalih krajeva Srbije. Svi objekti predstavljaju nepocrenjivo kulturno blago naše države.

Aleksandar Antić, Nina Stojiljković, Anđela Tasić, i Anđela Milosavljević su studenti Filozofskog fakulteta u Nišu na Departmanu za istoriju, ul. Ćirila i Metodija 2, 18000 Niš, Srbija, E-mail: srednjevekovnaistorijafb@gmail.com.

B. Opšte informacije o kulturnim spomenicima

Nakon što su objekti odabrani sledi priprema literarture koja će biti korišćena za izradu kratkog opisa objekta. Korišćena je isključivo zvanična literatura, opšteprihvaćena u istorijskoj nauci. Trudili smo se da podaci budu što konkretniji i da pruže korisniku kratak pregled istorijata objekta. Pored istorijskih podataka dodali smo i arhitektonske podatke o objektu, koji će pored priložene fotografije jasnije dočarati izgled kulturnog spomenika. S obzirom da je mapa namenjena ne samo stručnoj javnosti, već i laicima potrudili smo se da stil pisanja bude priemčiv svima. Izbegavana je upotreba stručnih termina.

C. Karta

Mape su rađene u vidu shortlist opcije koja omogućava ubacivanje geografskih koordinata objekta i njihovo tacno ubiciranje. Postojeće ArcGIS veb mape olakšale su ovaj postupak. Korisnici mogu da koriste 12 vrsta mapa :

- Snimci
- Imagery Hybrid
- Ulice
- Topografski
- Navigacija
- Ulice (noću)
- Teren sa oznakama
- Svetlosivo platno
- Dark Gray Canvas
- National Geographic Style mapa
- Okeani
- OpenStreetMap

Pomenute mape, pored toga što daju topografske i geografske podatke o objektu, olakšavaju turstima da lociraju ove spomenika kulture i što lakše i brže dođu do njih. Korisnik može da koristi mapu čak i bez mobilne ili mrežne veze, tako što je preuzme na svoj pametni telefon ili tablet pre samog polaska

D. Fotografija

Nezaobilazan detalj u izradi mape čini sama fotografija objekta. Korišćene su već postojeće fotografije, pritom su poštovana autorska prava, ali i fotografije koje smo lično napravili. Sve fotografije su u boji čime je

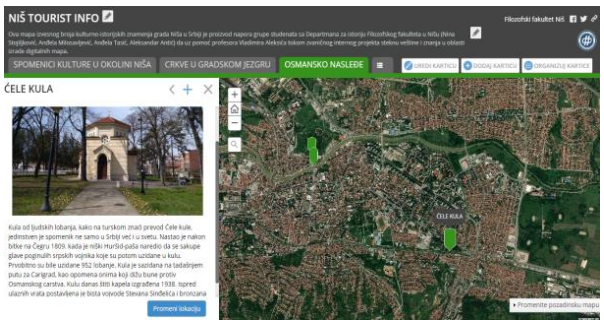
olakšano korisnicima da imaju uvid o kakvom objektu je reč.

III. SPOMENICI KULTURE

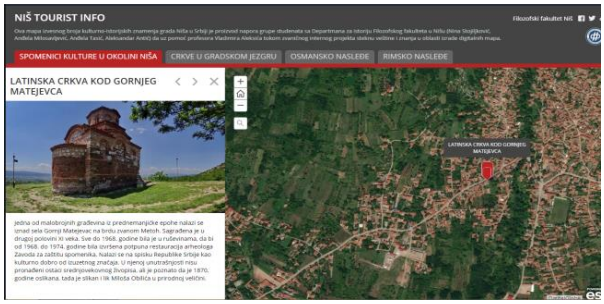
Rad je podeljen u 4 kartice, ali se one mogu podeliti u dve skupine. Prvu skupinu čine kulturni spomenici podeljeni u dve sekcije na osnovu njihove lokacije. Reč je o verskim objektima u gradskom jezgru (Saborna crkva, crkva Sv. Nikole, crkva Sv. Pantelejmona i bazilika sa martinijumom) i u okolini Niša (Latinska crkva kod Gornjeg Matejevca, manastir Sv. Jovana Krstitelja u Gornjem Matejevcu, manastir Sv. Đorđa u selu Kamenica, crkva Sv. Petke kod Donjeg Matejevca). Crkve na teritoriji grada često predstavljaju prvu stanicu turistima otud potreba da izdvojimo nekoliko većih i poznatijih, kako bi ih uputili u njihov istorijat, ali i tačnu lokaciju [1]. Za razliku od njih crkve i manastiri u Gornjem i Donjem Matejevcu, kao i u Kamenici uglavnom su poznati, isključujući stručnu javnost, lokalnom stanovništvu. Veliki broj građana Niša nije upoznat s tim kakvo kulurno blago se krije u neposrednoj blizini samog grada. Njihovo afirmisanje može da privuče i veliki broj turista koji će verovatno biti oduševljen i krajolikom ovih lokaliteta. S obzirom da je Niš od 4. veka, zahvaljujući Konstantinu Velikom, jedan od značajnijih verskih centara u Evropi, nije potrebno naglašavati značaj i potrebu za ovakvom mapom [2].

Drugu skupinu čine objekti podeljeni u dve kartice na osnovu epohe kojoj pripadaju. U pitanju su spomenici osmanske kulture u našem gradu (Niška tvrđava, Bali-Begova džamija, hamam i Ćele Kula) i rimskog nasleđa (Vila sa oktogonom, Medijana, ranohrišćanska grobnica u Niškoj Banji i lapidarijum u Niškoj tvrđavi). Ova kulturna dobra predstavljaju simbol grada Niša i uglavnom prvu asocijaciju na sam pomen grada. Nais, kao mesto rođenja cara Konstantina, ali i zbog samog geografskog polažaja, imao je naročit značaj za rimsko carstvo, otuda toliko objekata iz ove epohe na njegovoj teritoriji [3].

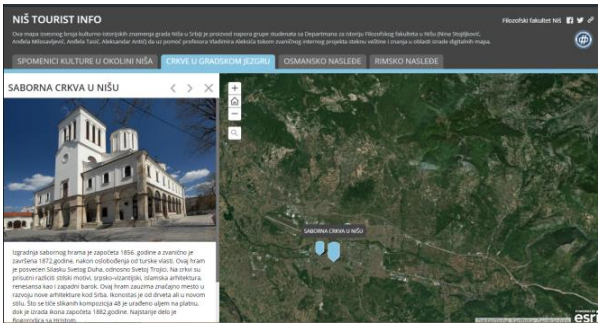
Niš je pao pod otomansku vlast konačno 1448. godine i ostao do 1737.godine kada je kontinuitet prekinula kratkotrajna austrijska okupacija grada. Neprekidnu tursku vlast od preko 250 godina pratilo je i podizanje kako objekata za odbranu grada tako i objekata islamske kulture. Izuzetak predstavlja Ćele Kula izgrađena 1809.godine, kao čin odmazde. Ovaj kulurni spomenik jedinstven je u svetu i predstavlja najveću građevinu napravljenu od ljudskih kostiju [4]. Ovi izdvojeni primeri iz pomenutih istorijskih perioda čine samo deo bogatog istorijskog nasleđa Niša.



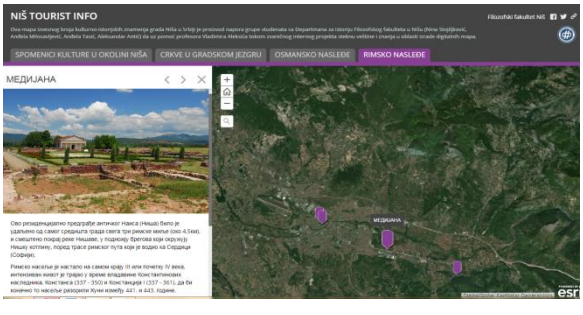
Slika 1. Mapa kulturnih spomenika osmanskog nasleđa; Ćele Kula.



Slika 2. Mapa verskih objekata u okolini Niša; Latinska crkva u Gornjem Matejevcu.



Slika 3. Mapa verskih objekata u gradskom jezgru; Saborna crkva u Nišu.



Slika 4. Mapa kulturnih spomenika rimskog nasleđa; Medijana.

IV. ZAKLJUČAK

U dosadašnjem radu na izradi mape uspeli smo da obradimo samo neke od kulturno-istorijskih gradjevina, koje su od enormnog značaja za naš grad.
Najveći problem predstavljao je upravo odabir objekata koji će se obrađivati. Pronalaženje odgovarajuće literature i sama izrada mape predstavljale su lakši deo s obzirom da smo imali pomoć i korisne savete od strane mentora tima, profesora Aleksića.

Mapa je naišla na pozitivan odjek u istorijskoj nauci jer predstavlja sasvim nov projekat koji će se vremenom nadograđivati i koristiti kako u naučnim krugovima, tako i u laičkim, s obzirom da će olakšati turističke obilaske spomenika i te kako značajnih, ali nepravedno zapostavljenih. Naš dosadašnji rad možete pogledati na sledećem linku: https://arcg.is/Pa4fG

ZAHVALNICA

Posebnu zahvalnost dugujemo doc. dr Vladimiru Aleksiću, bez čijih saveta i uputa ovaj projekat ne bi bio ostvaren.

LITERATURA

[1] Niški leksikon, Službeni glasnik, Niš 2011.
[2] M. Rakocija, Manastiri i crkve južne i istočne Srbije, Zavod za zaštitu spomenika kulture Niš, Niš 2013., 137-163.
[3] V.Petrović, Rimska naselja i komunikacije od Jadrana do Dunava (I-IV vek), Niš-Beograd 2015.
[4] Istorija srpskog naroda, III-1, Beograd 1993., 427-575.

Analogna distorzija V-2

Danilo Đorđević

Sadržaj – U ovom radu dat je šematski prikaz i PCB dizajn gitarske pedale V-2. Ova pedala ima ulogu da signal iz gitare, uz pomoć velikog pojačanja i mekog ili tvrdog klipovanja, dovede do izobličenja, odnosno do distorzije. Pedala se sastoji iz dva dela. Prvi deo koji je zadužen za distorziju i drugi deo je zadužen za aktivnu tonsku kontrolu.

I. UVOD

U današnje vreme gotovo sva elektronika se svodi na digitalnu, pa čak i audio elektronika. Međutim, da bi kvalitet zvuka bio što prirodniji i bolji, u kvalitetnim audio uređajima i dalje dominira analogna elektronika. Tako je i pedala V-2, koja je prikazana na slici 1, napravljena od nekoliko operacionih pojačavača okruženih pasivnim komponentama. Sama pedala nastala je isprobavanjem raznih kombinacija i vrednosti komponenata i doživela desetak revizija. Ima 6 potenciometra i 2 prekidača koji omogućavaju korisniku da kontroliše i oblikuje zvuk.

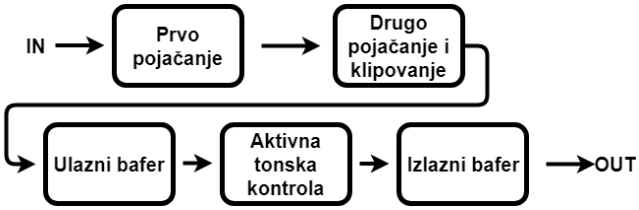


Slika 1. Izgled pedale V-2.

Danilo Đorđević je student master studija na modulu elektronika i miksrosistemi na katedri za mikroelektroniku, Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: danilo.djordjevic@elfak.rs

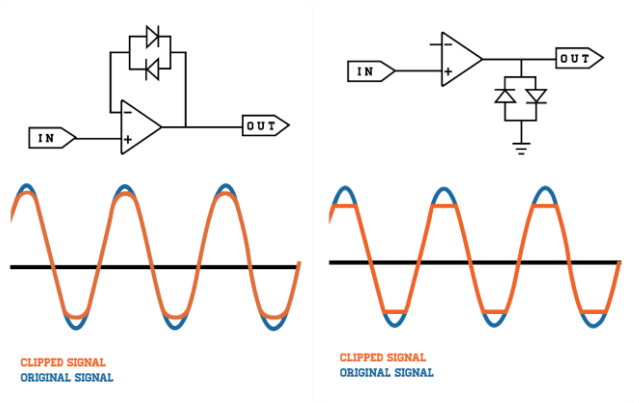
II. PROLAZAK SIGNALA KROZ PEDALU

Gitarski signal koji se obrađuje pedalom prolazi kroz nekoliko blokova koji su prikazani na slici 2. Signal prvo nailazi na blok za pojačanje koji takođe obezbeđuje veliku ulaznu impedansu. Nakon prve faze signal dolazi do drugog bloka gde se ponovo pojačava, ali se ovoga puta i klipuje.



Slika 2. Blok dijagram prolaska signala kroz pedalu.

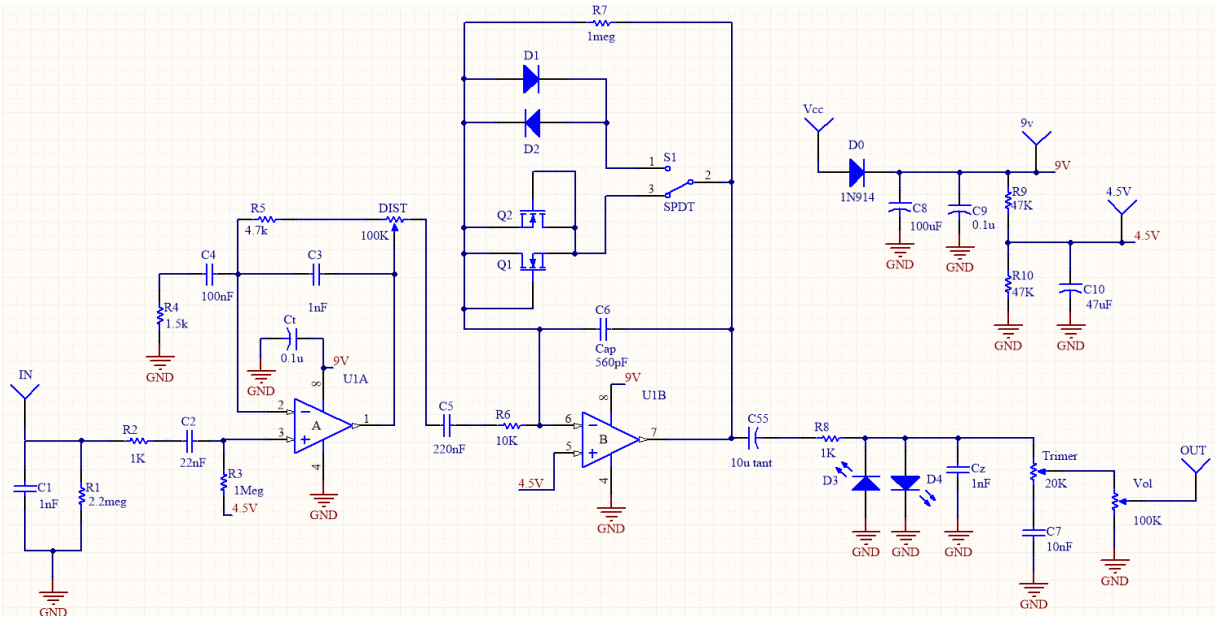
Klipovanje može biti tvrdo ili meko u zavisnosti kako je postavljen preklopnik. Obe vrste klipovanja su prikazane na slici 3. i imaju isti zadatak, da distorziraju i uobliče audio signal. Kod mekog klipovanja diode su postavljene u povratnoj sprezi operacionog pojačavača i blago odsecaju ivice signala čineći distorziju prirodnijom i blažom kao kod cevnih pojačala. Kod tvrdog klipovanja diode su postavljene posle operacinog pojačavača i jednim krajem su povezane na izlaz operacionog pojačavača dok su drugim krajem vezane za masu. Ovako postavljene diode grubo odsecaju signal čineći zvuk agresivnijim.



Slika 3. Konfiguracije i oblici napona mekog klipovanja (levo) i tvrdog klipovanja (desno).

Na samom ulasku u drugi deo pedale signal nailazi na ulazni bafer koji ima ulogu da spreči uticaj opterećenja

na ulazni signal. Aktivna tonska kontrola pojačava i smanjuje karakteristične gitarske frekvencije za 15 dB. Pošto pomoću četiri potenciometra nije moguće pokriti određene frekvencije, na pedali se nalazi preklopnik koji omogućava pomeranje propusnog opseg filtra koji propušta srednje frekvencije. I na kraju, signal izlazi iz pedale preko izlaznog bafera koji obezbeđuje nisku impedansu na izlazu.

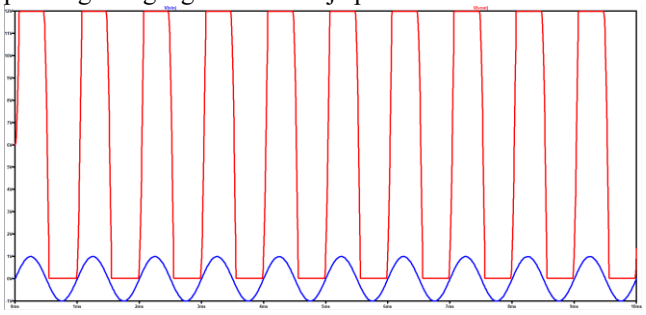


Slika 4. Električna šema prvog dela V-2 pedale.

III. ELEKTRIČNA ŠEMA PEDALE

Na slici 4. je prikazana električna šema prvog dela pedale. Šema je jednostavna i sastoji se od osnovnih konfiguracija operacionog pojačavača. Na ulazu signal nailazi na otpornik R1 koji ima ulogu da drži ulaz operacionog pojačavača na masi kada nema signala i da umanji pucanje prekidača kada se pedala uključuje ili isključuje. Kondenzator C1 uklanja sve visoko-frekventne šumove iz signala koji mogu dovesti do oscilacija. Zatim signal nailazi na otpornik R2 koji ograničava struju koja ulazi u operacioni pojačavač štiteći ga od elektrostatičkog pražnjenja i dekapling kondenzator C2 koji predstavlja visoko-propusni filtar i razdvaja jednosmernu komponentu od gitarskog signala. Otpornik R3 ima ulogu da podigne neinvertujući kraj operacionog pojačavača na 4.5 V. Naizmenični signal koji dolazi iz gitare će sada imati na sebi jednosmernu komponentu od 4.5 V i ovakav naizmenični signal neće biti odsečen od strane operacionog pojačavača. Ulazna impedansa operacionog pojačavača, koja je u paraleli sa otpornicima R1 i R3, je oko 680 kΩ i predstavlja dobru ulaznu impedansu. Iako se za gitarske pedale smatra da ulazna impedansa na bi trebala da bude ispod 1 MΩ, impedansa od 680 kΩ je dovoljno velika da spreči gubljenje gitarskih tonova.

Operacioni pojačavač U1A se nalazi u neinvertujućoj konfiguraciji i njegovo pojačanje se kreće od 4, kada je otpornost potenciometra 0, odnosno do 70 kada je otpornost potenciometra 100 kΩ (slika 3). Kako je amplituda ulaznog gitarskog signala oko 1 Vpp, ovo bi značilo da će signal posle ove faze pojačanja imati 70 V! Međutim, to u praksi nije moguće pošto je izlazni napon operacionog pojačavača ograničen njegovim napajanjem što je u ovom slučaju 9 ili 12 V, tako da će signal na izlazu imati maksimalno 12 Vpp i imaće oblik gotovo idealnog pravougaonog signala kao što je prikazano na slici 5.

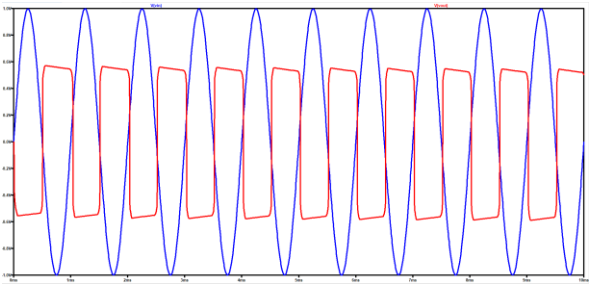


Slika 5. Izgled ulaznog signala (sinusni) i signala posle prve faze pojačanja (pravougaoni).

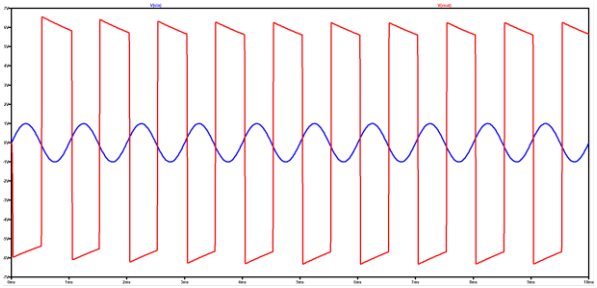
Kondenzatori C3 i C4 igraju jako bitnu ulogu u filtriranju i oblikovanju distorziranog signala. Kondenzator C4 zajedno sa otpornikom R4 čini visoko-propusni filtar, odnosno sve frekvencije ispod 1060 Hz će sprovesti ka masi i te niske frekvencije će se manje distorzirati. Sa druge strane, kondenzator C3 u paraleli sa otpornikom R5 i potenciometrom DIST, čini nisko-propusni filtar. U zavisnosti od položaja potenciometra, visoko-propusni

filtar imaće graničnu frekvenciju od 34 kHz kada je vrednost potenciometra 0 Ω, odnosno 1500 Hz kada je pojačanje maksimalno odnosno vrednost potenciometra je 100 kΩ. Ovakva sprega potenciometra i kondenzatora C3 je dobra zato što se sa porastom pojačanja uklanjaju visoke frekvencije koje kada se distorziju, nisu prijatne za čovekovo uvo.

Ovakav pravougaoni signal dolazi na dekapling kondenzator C5 i otpornik R6 koji definiše pojačanje operacionog pojačavača U1B koji se nalazi u invertujućoj konfiguraciji. Pojačanje ove konfiguracije koje je određeno otpornicima R6, R7 i potenciometrom DIST (Slika 4.). Kada je klizač potenciometra pomeren u levo, otpornost na invertujućem ulazu biće 110 kΩ, što znači da će pojačanje biti 9. Sa druge strane, kada je klizač pomeren i desnu stranu ulazna otpornost biće 10 kΩ, tako da u ovom slučaju će pojačanje biti 100. Međutim, i ako je pojačanje u ovoj fazi mnogo veliko, gitarski signal će u zavisnosti od položaja preklopnika S1 kretati od 1.4 Vpp do 4 Vpp, kao što se može videti na slici 6 i slici 7. Kada je prekidač S1 na položaju 1, kliperske diode D1 i D2 će odsecati naizmenični signal i ograničiti mu amplitudu na oko 1.4 Vpp. Na istu vrednost biće odsečen signal i kada je prekidač u položaju 2, ali tranzistori Q1 i Q2, koji su postavljeni u kliperskoj konfiguraciji, blaže odsecaju ivice signala čineći ton mekšim. Diode D1 i D2 kao i tranzistori Q1 i Q2 su postavljeni tako da meko klipuju signal čineći ivice signala blago odsečenim i sam zvuk je mekši. Pošto prekidač S1 ima 3 položaja, kada se nađe u srednjem, odnosno isključenom položaju, signal će biti odsečen od strane LE dioda D3 i D4. Ove diode se nalaze u konfiguraciji za tvrdo klipovanje i amplitudu signala će ograničiti na 4 Vpp, čineći zvuk glasnim i grubim. Uz LE diode se nalazi kondenzator Cz koji ima zadatak da sprovede visoke frekvencije na masu i time ublaži klipovanje istih. Na oblikovanje finalnog zvuka utiče i kondenzator C6 koji u paraleli sa otpornikom R7 čini nisko-propusni filtar čija je frekvencija odsecanja 320 Hz. Kao i u prvoj fazi pojačanja i ovo odsecanje visokih frekvencija se radi da bi u finalni zvuk ušlo što manje distorziranih visokih frekvencija koje smetaju čovekovom sluhu.



Slika 6. Izgled ulaznog signala (sinusni) i signala posle druge faze pojačanja (pravougaoni). Slika prikazuje klipovanje 1N4148 dioda.

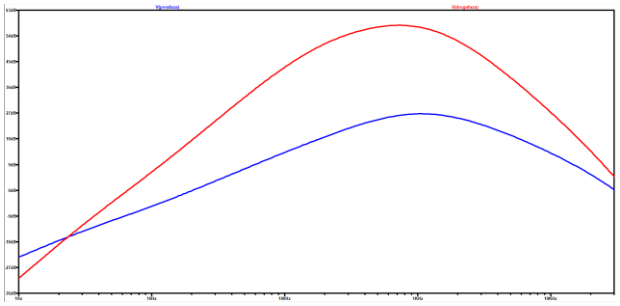


Slika 7. Izgled ulaznog signala (sinusni) i signala posle druge faze pojačanja (pravougaoni). Slika prikazuje klipovanje LED dioda.

Na izlazu iz ovog operacionog pojačavača signal prolazi kroz otpornik R8 koji ograničava izlaznu struju operacionog pojačavača i još jedan dekapling kondenzator C55 koji treba da ima što veću kapacitivnost kako bi što manje uticao na gitarski signal. Posle prolaska kroz ovaj kondenzator signal gubi jednosmernu komponentu od 4.5 V. Na samom izlazu iz kola signal nailazi na jednostavnu pasivnu tonsku kontrolu sačinjenu od Trimer potenciometra i kondenzatora C7, koji manipuliše visokim frekvencijama, sprovodeći ih na masu ili na izlaz pedale. Ova tonska kontrola je namenjena za fino podešavanje signala za sledeći deo pedale. Da bi se kontrolisala jačina zvuka, potrebno je postaviti logaritamski potenciometar kao naponski razdelnik. Upravo ovu ulogu ima potenciometar Vol.

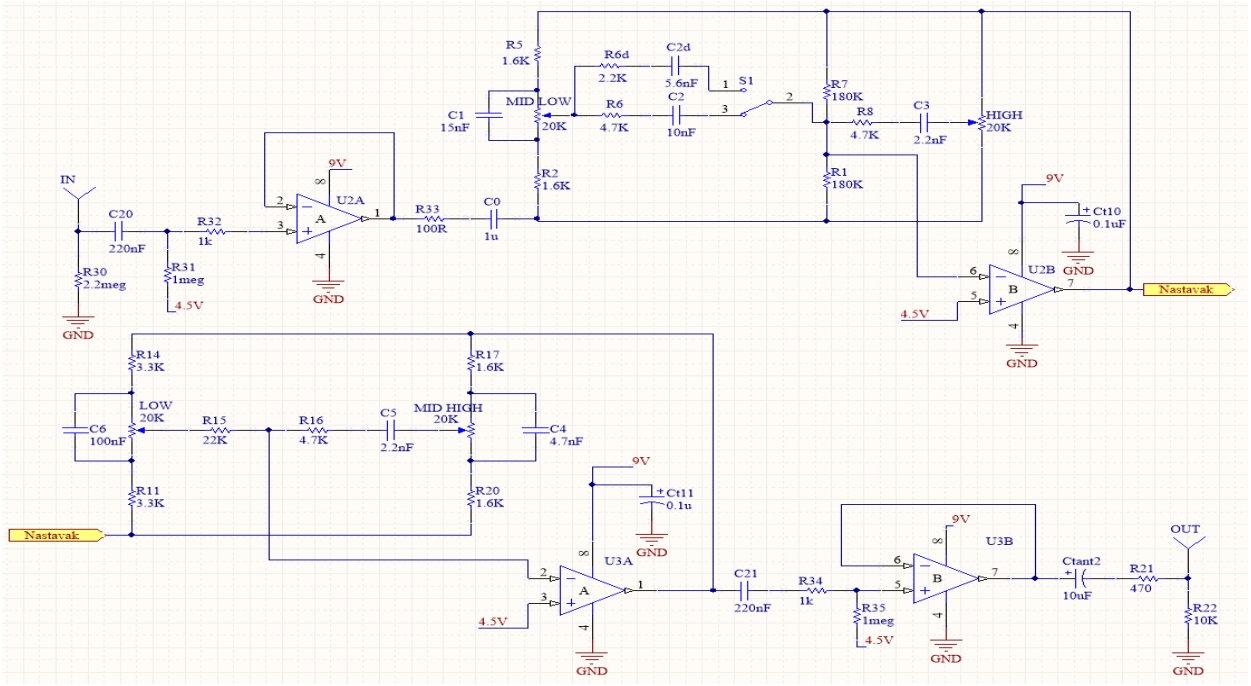
Naravno sva ova elektronika ne bi funkcionisala da nije obezbeđeno napajanje za njen rad. Glavno napajanje dolazi preko diode D0 koja štiti elektroniku od inverzne polarizacije. Kondenzatori C8 i C10 su low esr elektrolitski kondenzatori i služe za otklanjanje poskakivanja iz napajanja dok kondenzator C9 je kondenzator od 100 nF koji otklanja bilo kakve šumove i frekvencije iz jednosmernog napajanja.

Na slici 8. se mogu videti pojačanja posle prve i druge faze. U oba slučaja su naglašene srednje frekvencije. Posle prve faze najviše su distorzirane frekvencije oko 1000 Hz, dok su u drugoj fazi najviše distorzirane frekvencije oko 800 Hz. Upravo ovako isfiltrirane i obrađene frekvencije daju karakterističan zvuk ovoj distorziji.



Slika 8. AC analiza pojačanja signala u programu LTspice nakon prvog i drugog pojačanja.

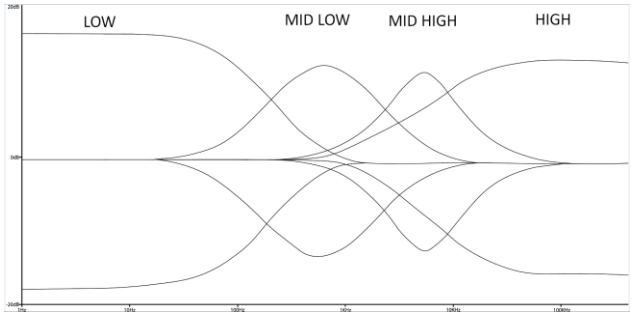
Drugi deo pedale je prikazan na slici 9. i se zasniva na *Baxandal* [1] aktivnoj tonskoj kontroli, nazvanoj po audio inženjeru Peter-u Baxandal-u. Ova tonska kontrola je izabrana zato što omogućava fino podešavanje tona tako što pojačava ili smanjuje određene frekvencije do 15 dB. Takođe, još jedna prednost ove tonske kontrole je što ne dolazi do preklapanja propusnih opsega prilikom pojačavanja, odnosno nema međusobnog uticaja između frekvencija koje se pojačavaju kao što je slučaj kod pasivnih tonskih kontrola.



Slika 9. Šematski prikaz drugog dela V-2 pedale.

Distorzirani signal iz prvog dela pedale dolazi na nekoliko pasivnih komponenti i bafer sa jediničnim pojačanjem. Kondenzator C20 kao i otpornici R30, R31 i R32 imaju istu ulogu kao i u prvom delu tako da ovde neće biti razmatrani. Jedinični bafer U2A sprečava gubljenje tonova i obezbeđuje veliku ulaznu impedansu u ovom delu kola i sprema signal za dalju obradu. Preko dekapling kondenzatora C0 i otpornika R33 signal dolazi na *Baxandal*-ovu tonsku kontrolu. Ona omogućava da se uz pomoć jednog operacionog pojačavača kontrolišu dva propusna opsega. Kaskadnim vezivanjem operacionih pojačavača dobija se tonska kontrola koja kontroliše četiri propusna opsega.

U programu LTspice je simulirana ova tonska kontrola i izvršena je AC analiza. Simulacija je rađena za svaki potenciometar posebno kada je klizač pomeren do kraja u jednu stranu, a zatim u drugu. Rezultati simulacije koji su prikazani na slici 10. pokazuju da potenciometar LOW manipuliše frekvencijama ispod 100 Hz, MID LOW je zadužen za frekvencije oko 800 Hz, MID HIGH upravlja frekvencijama oko 3.5 kHz i HIGH potenciometar pojačava ili smanjuje frekvencije iznad 5 kHz. Na izlazu iz Baxandala nalazi se operacioni pojačavač U3B u konfiguraciji jediničnog bafera koji ovog puta ima ulogu da obezbedi nisku izlaznu impedansu iz pedale. Izlazna impedansa određena je otpornicima R21 i R22 i iznosi 540 oma, što predstavlja odličnu izlaznu impedansu koja neće oslabiti izlazni signal.



Slika 10. AC analiza pojačanja signala u programu LTspice.

IV. Realizacija štampanih ploča u Altium Designer-u i praktična realizacija pedale

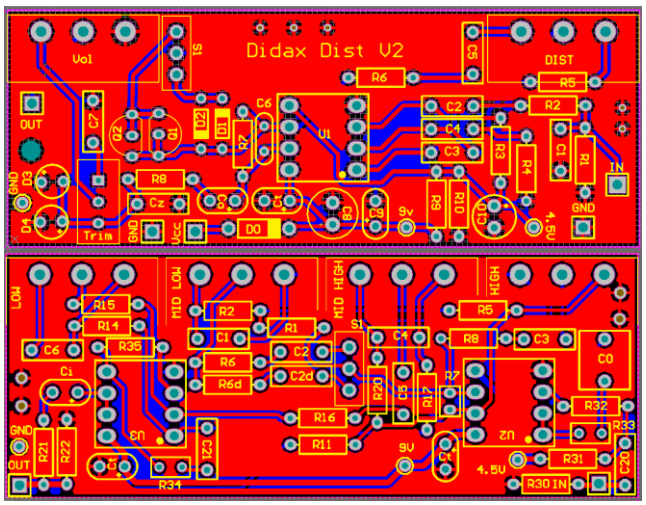
Pedala V-2 je realizovana pomoću dve štampane ploče. Ploče su dvoslojne i dizajnirane u Altium Designer-u. Same pločice na kojima su direktno zalemljeni potenciometri, nemaju rupe za montažu, nego su u metalnom kućištu pričvršćene pomoću tih potenciometra za kućište (Slika 11). Prilikom dizajniranja ploča, najviše se

vodilo računa o rasporedu komponenata što se može videti na slici 12. Dimenzije metalnog kućišta su 110 x 83 x 35 mm, tako da je veliki izazov bio smestiti veliki broj TH komponenata na malu površinu ploče. Ploče su imale preddefinisanu dimenziju od 75 x 28 mm da bi kablovi unutar kućišta mogli da se smeste između samih ploča i kućišta. Debljina veza u ovom slučaju ne predstavlja problem, jer se radi o veoma malim strujama tako da su veze za signale 0.4 mm, a veze za napajanje 0.6 mm. Naravno, radi eliminisanja šuma razlivena je masa na obe strane pločice.

Samo kućište je napravljeno od lima koje je savijano na presi. Kućište je napravljeno iz dva dela, gornjeg poklopca, na kome se montiraju sve komponente i ulazi, i donjeg dela kojim se zatvara kutija. Oba dela su plastificirana i nakon plastifikacije je odrađena UV štampa na gornjem belom poklopcu.



Slika 11. Unutrašnji izgled pedale V-2.



Slika 12. 2D prikaz pločice i raspored komponenata u Altium Designer-u.

V. Zaključak

Bilo je potrebno nekoliko meseci da bi pedala V-2 bila realizovana. I ako na samom po pedala nije dala zadovoljavajuće rezultate, bilo je potrebno desetak revizija i izmena vrednosti komponenta da bi proizvela odgovarajući zvuk. Nakon svih prepravki, pedala je spremna da proizvede karakterističan zvuk distorzije kojim krajni korisnik može pažljivo da manipuliše.

Zahvalnica

Autor rada se zahvaljuje Goranu Iliću na izdvojenom vremenu, sugestijama i značajnoj pomoći prilikom razvijana i testiranja pedale.

Literatura

[1] D. Self, Small signal audio design, The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK30 Corporate Drive, Suite 400, Burlington, MA 01803, USA First published 2010

Servosistemi za laboratorijske makete bazirani na Arduino Due mikrokontroleru

Snežana Đorđević, Marija Filipović

Sadržaj- U ovom radu opisan je servosistem baziran na Arduino Due miktokontroleru zajedno sa elementima sistema i njihovim mogućnostima i pomoću ovog sistema realizovana je laboratorijska maketa kojom se demonstrara rad servosistema sa i bez zatvorene povratne sprege, različiti zakoni upravljanja i uticaja njihovih parametara koji se mogu menjati uz pomoć makete i displeja, ili uz pomoć računara. Takođe, postoji mogućnost snimanja odskočnih odziva i direktnog upoređivanja odziva koji se dobijaju primenom različitih zakona upravljanja. Na taj način je prikazana primena najvažnijih principa samo uz pomoć makete, a povezivanjem sa računarom se dobija dodatna funkcionalnost.

I. UVOD

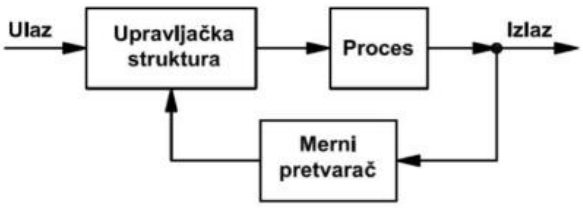
Servomehanizam ili servosistem je sistem upravljanja sa povratnom spregom (Slika 1) kod koga je upravljana promenljiva mehanička pozicija ili brzina. Osnovna funkcija servomehanizma je da njegov izlazni signal brzo i precizno prati promene ulazog signala. Nekada se naziva i prateći jer mu je samo ime izvedeno iz latinskih reči *servus* (onaj koji služi ili prati) i *mehanism* (mehanizam).

Način funkcionisanja servomehanizma se bazira na kontinualnom merenju izlazne veličine i njenog poređenja putem povratne petlje sa ulaznom veličinom. Osnovni cilj jeste smanjenje greške između ulazog i izlaznog signala uz pojačanje snage. To znači da izlazna veličina prati ulaznu uz određenu konstantu pojačanja [1]. Izlazni signal je mehanička veličina ugao, brzina ili ubrzanje i može se ostvariti poziciona, brzinska i strujna povratna sprega [2]. Servosistemi se koriste za upravljanje robotima kod automatizovanih procesa u industriji, u automobilskoj industriji, u mobilnoj robotici, na raznim uređajima koje koristimo svakodnevno kao što je npr. tempomat u automobilu.

A. Poziciona i brzinska povratna sprega

Poziciona povratna sprega, kao što i samo ime kaže, vraća povratnu informaciju o poziciji. Njena uloga je da obezbedi ugaono pozicioniranje motora. Poziciona petlja dobija informaciju o trenutnoj poziciji sa davača pozicije, a izlazna veličina ove petlje je komanda brzine. Kao i poziciona, i brzinska petlja dobija informaciju o trenutnoj brzini od davača pozicije, pri čemu je potrebno signal sa davača prvo obraditi u odgovarajućem dekoderu brzine.

Upravljački signali se od glavnog kontrolera prenose do svakog servo drajva u sistemu, koji na osnovu primljenog signala generiše na izlazu podatak koji se vodi ka servo motoru. Za realizaciju makete biće korišćena poziciona povratna sprega, jer omogućava lak prelaz na brzinsku, a to postižemo promenom regulatora [2].



Slika 1. Blok šema sistema automatskog upravljanja sa povratnom spregom.

Prednost sistema sa povratnom spregom je u tome što ovi sistemi upravljanja koriste povratnu spregu koja omogućava da odziv sistema bude relativno neosetljiv na eksterne smetnje i u izvesnoj meri, na varijacije parametara sistema. Zbog toga osim mernog pretvarača koji mora biti visoke tačnosti, ostali elementi sistema upravljanja mogu biti manje tačnosti i jeftiniji, da bi se dobio dovoljno tačan izlazni signal [3].

II. PRINCIP RADA I ELEMENTI SISTEMA

Zadatak ovog sistema kao laboratorijske vežbe je upoznavanje studenata sa izgledom i načinom funkcionisanja servosistema, pri čemu bi oni trebalo da uvide konkretnu razliku između stabilnog i nestabilnog sistema, probaju različite zakone upravljanja i analiziraju njihov uticaj na odziv sistema, kao i da se upoznaju sa različitim tipovima odziva.

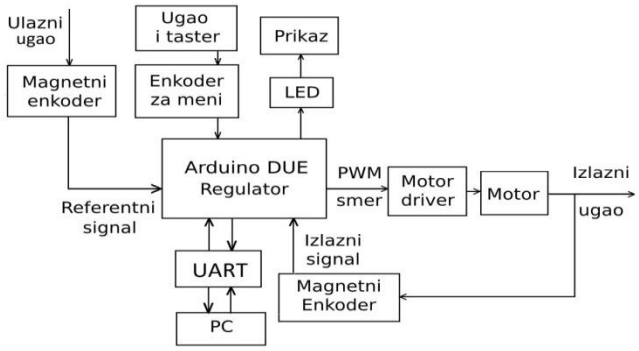
Moguće je vršiti prelaz sa jednog tipa regulatora na drugi, kako bi se videli njihovi uticaji. Trenutno su realizovani PI regulator, PID regulator i diferencijalni kompenzator.

Snežana Đorđević, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: snezana.elfak@gmail.com

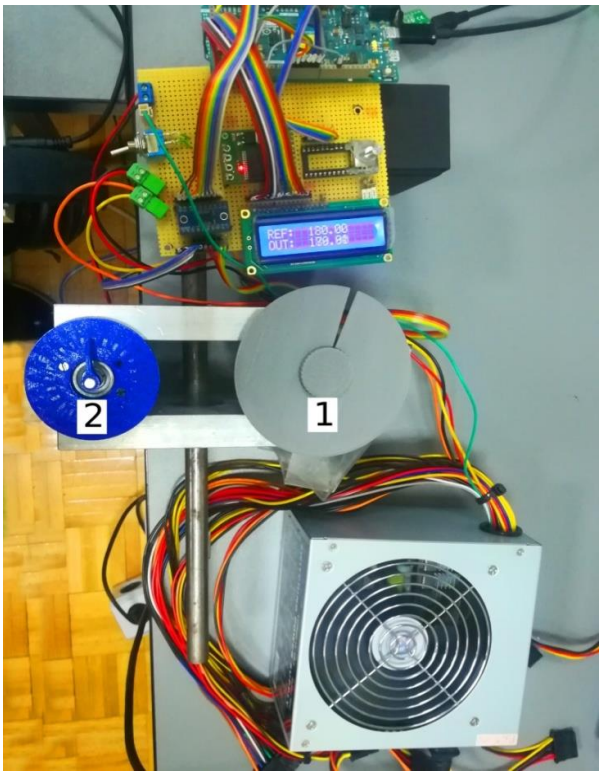
Marija Filipović, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: machefilipovic21@gmail.com

A. Demonstracija rada sistema

Na Slici 2 i 3 je prikazana realizovana laboratorijska maketa. U svojoj najosnovnijoj formi, sistem se sastoji od referenog i izlazog signala, pri čemu izlazni signal prati referentni. Referentni ugao se zadaje uz pomoć enkodera sa desne strane (1), dok je izlazna veličina pozicija motora koji se nalazi sa leve strane (2). Radi lakšeg praćenja pozicije motora za rotor je pričvršćena skazaljka. Pozicija motora se očitava enkoderom koji se nalazi sa zadnje strane motora. Prikazani LCD displej, veličine 16x2 karaktera, se koristi za prikazivanje podataka o trenutnoj i referentnoj poziciji. Putem displeja moguće je zadavati referentnu poziciju, kao i promena parametara.



Slika 2. Elementi makete.

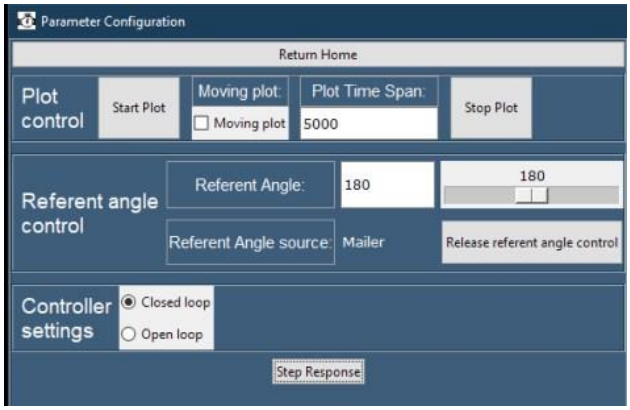


Slika 3. Realizovana laboratorijska maketa.

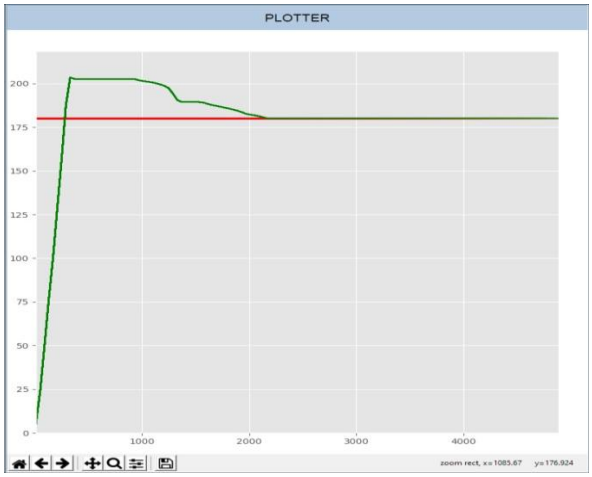
Zadavanje reference može se izvršiti i uz pomoć displeja. Dodatne opcije koje se mogu videti na displeju su tačne vrednosti referentnog i izlazog signala (preciznost na 2 decimale), kao i promena parametra kontrolera – promena pojačanja i promena svih konstanti u kontroleru, odnosno promena svih dejstva kontrolera, nakon čega se ponovo može pristupiti eksperimentu i videti razlika u ponašanju sistema. Cilj je da se sistem dovede do granice stabilnosti i pokaže sistem koji je blizu granice stabilnosti, kao i nestabilan sistem. Na ovaj način prikazujemo kako različiti parametri regulatora fizički utiču na ovaj sistem.

Drugi deo sistema je prikaz na računaru uz pomoć serijske komunikacije UART, gde studenti mogu videti grafički prikaz odziva ovakvog sistema. Promenom parametara mogu se snimiti različiti odskočni odzivi i pratiti kako određeni parametar utiče na odziv sistema.

Na ovaj način je obezbeđena potpuna kontrola sistema na maketi preko računara, ali i potpuna kontrola makete nezavisno od računara. Računari nam omogućavaju snimanje grafika koji se kasnije mogu koristiti u izveštaju za laboratorijsku vežbu i koji se mogu međusobno upoređivati. Kontrola putem računara kao i primer snimljenog odskočnog odziva može se videti na Slici 4 i Slici 5.



Slika 4. Kontrola putem računara.



Slika 5. Odziv sistema na odskočnu funkciju.

Elementi sistema su: mikrokontroler Arduino DUE [4], CROUZET motor 82850002 (24V, 52W) [5], dva identična apsolutna magnetna enkodera [6], drajver za motor koji se upravlja preko PWM-a, LCD displej [7], enkoder sa tasterom koji je namenjen za kretanje kroz displej meni i promenu parametara, napajanje (računarsko napajanje) i odgovarajuće propratne aktivne i pasivne komponente koje omogućavaju rad sistema. Pobrojane elemente i njihov način povezivanja možemo videti na Slici 3.

Za normalan rad sistema potrebna su tri naponska nivoa: 3.3V za arduino DUE, 5V za enkodere i drajvere i 12V za motore. Arduino DUE radi na naponskom nivou 3.3V i to predstavlja nekompatibilnost sa ostalim komponentama sistema jer se one napajaju naponom od 5V. Međutim, on je izabran zbog jačine procesora što omogućava kontinuirani razvoj i zadovoljavajuće preformanse.

Mikrokontroleri razmenjuju podatke sa ostalim kontrolerima ili perifernim jedinicama putem UART serijske komunikacije. Ovo je osnova svake serijske komunikacije i služi za pretvaranje paralelnog podatka (najčešće 8-bitnih odnosno 1 bajt) u serijski oblik sa svrhom da se smanji potreban broj linija za komunikaciju odnosno omogući prenos podataka kroz druge medije. U predajniku se paralelni podatak pretvara u serijski niz bitova, dok prijemnik taj niz bitova pretvara u paralelni podatak. Za komunikaciju se koriste dva pina, Tx-predajni i Rx-prijemni [8].

Enkoderi korišćeni u ovoj maketi koriste SSI konumikaciju. Za ovaj vid komunikacije za svaki enkoder je potrebno po 3 standardna digitalna pina. Jedan se koristi kao takt, koji mikrokontoler daje enkoderima, a druga dva su izlazi A i B sa enkodera.

Displej može koristiti osam ili četiri pina za prenos podataka. Konekcija je ostvarena pomoću četiri pina radi jednostavnijeg kabliranja. Pinovi LCD displeja koji su povezani serijskom komunikacijom su 2, 3, 4, 5, 11 i 12.

Enkoder sa tasterom koji je namenjen za kretanje kroz displej meni i promenu parametara ima 3 pina, dva se koriste za detekciju okretanja a jedan za selektovanje odnosno za potvrdu odabira.

Impulsno širinska modulacija (skraćeno PWM - eng. *Pulse Width Modulation*) je vrsta modulacije koja predstavlja način da se od digitalnog signala napravi signal analogne vrednosti. Odnos impuls/pauza se moduliše tako da odgovara specificiranom nivou analognog signala.

Kontrola zasnovana na PWM modulaciji signala je moćna tehnika za upravljanje analognim kolima pomoću digitalnih izlaznih signala. PWM je primenjena u širokom spektru aplikacija, od primene u merenjima i komunikacijama, sve do primene u kontroli snage i konverzijama električne energije u druge oblike (zvuk, mehaničku energiju) [9].

Drajver za motor je Pololu VNH5019 [10] i ima dva pina za određivanje smera ili kočenje, i jedan pin uz pomoć koga se određuje brzina obrtanja posredstvom PWM signala [11].

Enkoder za zadavanje referenog signala u slučaju pozicionog servosistema zadaje direktan referentni signal, dok u slučaju brzinskog servositema zadaje vrednost brzine u opsegu 0–100 %.

III. PROGRAMIRANJE MIKROKONTROLERA

A.Real Time (RT) komponenta sistema

Servosistemi su sistemi za rad u realnom vremenu i zbog toga je potrebno na Arduino DUE mikrokontroleru pored glavne petlje (*loop*), realizovati vremenski kritičnu petlju (RT petlja).

Prilikom realizacije potrebno je koristiti RT petlju sistema, koja nam omogućava da određeni delovi koda budu izvršeni u adekvatnom vremenskom trenutku i da se na taj način eliminiše greška u toku izvršavanja. Sistem mora raditi u RT režimu jer u suprotnom sistem realizovan na ovakav način ne bi funkcionisao. RT petlja se koristi za izvršavanje vremenski kritičnih aktivnosti u sistemu, odnosno nešto što uvek treba da se izvršava u određenom trenutku, tj. na određeni vremenski period.

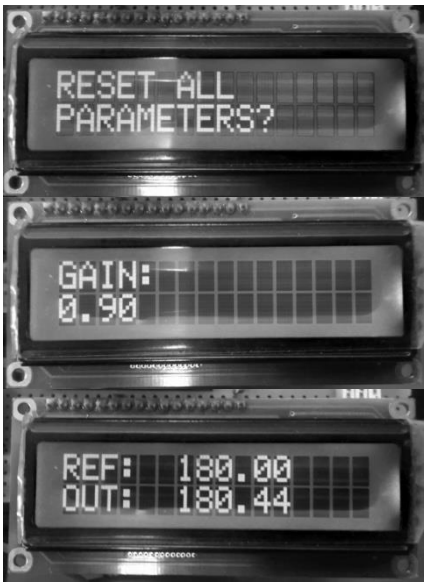
Sva izračunavanja u prekidačkoj funkciji moraju da se izvrše pre isteka periode diksretizacije. U suprotnom, prekidacka funkcija ce prekinuti samu sebe i mikrokontroler će prestati sa radom. Eksperimentalnom metodom smo došli do minimalne periode diskretizacije od 4ms. U toku datog vremenskog intervala je potrebno izvršiti: čitanje sa enkodera, primenu zakona upravljanja (zatvaranje povratne sprege po poziciji ili brzini) i čitanje enkodera sa tasterom koji je namenjen za kretanje kroz displej meni, dok osvežavanje displeja i komunikacija sa računarom nisu RT kritični.

B. Podešavanje parametara i prikaz na displeju

Pored zatvaranja povratne sprege i realizacije samog sistema veliki izazov je bio realizovati ostale komponente kao što su prikaz na displeju, promena parametara i komunikacija.

Već je navedeno da je čitanje enkodera za kretanje kroz meni kritična komponenta, tako da se u RT-petlji vrši njegovo čitanje, interpretacija komande i smeštanje komande u bafer. Komande koje su tom prilikom interpretirane su: okretanje u smeru kazaljke na satu, okretanje u suprotnom smeru i odabir, odnosno enter. U glavnoj petlji se vrši čitanje komandi iz bafera i prikaz odgovarajućeg sadržaja na displeju. Da bismo omogućili prikaz na displeju, čiji meni ima veliki broj stavki, kreirali smo dve switch-case petlje. Prva switch-case petlja vrši prelazak iz trenutnog stanja na odgovarajuće stanje na osnovu interpretirane komande, zatim se poziva druga switch-case petlja koja vrši prikaz odgovarajućih informacija u zavisnosti od trenutnog stanja. U slučaju da nema novih komandi duže od 80 ms, vrši se ponovo

pozivanje switch-case petlje za prikazivanje kako bi se sadržaj displeja stalno osvežavao. Ovo je jako bitno kako se osvežavanje displeja (koji npr. prikazuje vrednost neke promenljive) vrši na svakih 80 ms, a ne na promenu vrednosti te promenljive. Različita stanja na displeju su prikazana na Slici 6.



Slika 6. Različiti prikazi stanja na displeju.

C. Serijska komunikacija

Komunikacija između računara i Arduina je serijska master-slave, gde je računar master, a Arduino slave. To znači da Arduino ništa ne šalje dok mu računar prethodno ne zatraži. Podaci koji se ovom prilikom razmenjuju su: parametri odziva koji služe za njegovu vizuelizaciju na računaru, parametri i tipovi regulatora koji se mogu menjati, parametri mikrokontrolera koji se mogu menjati, pozivanje specifičnih funkcija kao što je funkcija za snimanje odskočnog odziva sistema. Komunikacija sa računarom pruža proširene funkcionalnosti makete, jer računar kao medijum ima bolje funkcionalnosti i pruža mnogo bolje mogućnosti za prikazivanje podataka i lakoću menjanja podataka. Takođe, proširena funkcionalnost se ogleda i u vidu snimanja grafika odskočnih odziva koji studenti kasnije mogu koristiti u izveštaju. Zbog toga što je modul za komunikaciju serijski, koji je standardizovan, za prikaz se osim računara može koristiti i neki drugi medijum. Zbog toga što je sve realizovano softverski, prelazak sa pozicionog na brzinski servosistem je vrlo jednostavan. Potrebno je samo izdvojiti podatak o brzini i primeniti adekvatan zakon upravljanja, dok sve ostale strukture ostaju iste tako da studenti mogu da vide uticaj sprege hardver-softver na funkcionalnost sistema.

IV. ZAKLJUČAK

Laboratorijska maketa nam daje uvid u osnovne komponente i rad servosistema i pomaže studentima da uoče i analiziraju transformaciju sistema i razliku između brzinskog i pozicionog sistema na istom hardveru sa adekvatnim softverskim elementima. Pored toga postoji opcija da se sve realizovano na računaru može implementirati i na tabletu ili serveru, dok veza može sa žičane da se prebaci na bežičnu komunikaciju dodavanjem WiFi modula. Takođe, zbog toga što su najbitniji delovi makete u softveru, jedna maketa se može koristiti kod studenata starijih godina i za projektovanje i implementaciju različitih zakona upravljanja kako bi studenti bili zaduženi za projektovanje dela softvera na mikrokontroleru.

LITERATURA

[1] dr Miroslav Bjekić, Stevanović Zoran, Servo *regulator Lexium LXM32M*, str.7 http://www.empr.ftn.kg.ac.rs/7%20REGULACIJE/Lexium%20LXM32M.pdf

[2] Sinhronizacija servo osa primenom PLC Open Motion Standarda, https://plc.rs/aplikacije/plc-open-motion-standard/

[3] Prof. dr Milovan Radulović, *Brodska automatika*, str. 17, https://www.ucg.ac.me/skladiste/blog_12929/objava_18361/fajlovi/tema1.pdf

[4] Arduino Due schematic, https://www.arduino.cc/en/uploads/Main/arduino-Due-schematic.pdf

[5] Crouset Motors, model Ø 42 mm 22 to 52 Watts 52 W Part number 82850002, https://soda.crouzet.com/pn/?i=82850002

[6] AEAT-6010/6012 Magnetic Encoder, https://docs.broadcom.com/docs/AV02-0188EN

[7] LCDWH1602B2-TMI-ET# Data Sheet, *lcd-2x16-datasheet*, http://www.winstar.com.tw

[8] Mladen Vuković, *Serijska komunikacija mikrokontrolera*, https://www.micromatic.hr/download/Serijska%20komunikacija%20mikrokontrolera.pdf

[9] Miličić Nikola, Branković Dragan, *Implementacija impulsno-širinske (PWM) i impulsno-amplitudske modulacije (PAM) korišćenjem mikrokontrolera PIC16F877*, pp. 2, Elektronski fakultet Univertitet u Nišu, http://es.elfak.ni.ac.rs/Papers/N.Milicic-D.Brankovic%20-%20Projekat%20(PWM%20&%20PAM).pdf

[10] Data sheet Pololu VNH5019 Motor Driver Carrier, www.https://www.pololu.com/product/1451

[11] Marija Filipović, Snežana Đorđević, *Realizacija digitalnog real-time servosistema na Arduino Due mikrokontroleru*, IEEESTEC 12th Student project conference

Nova metoda za automatizaciju ispitivanja kondenzatora upotrebom virtuelne instrumentacije

Miloš Đorđević, Branislav Jovičić

Sadržaj – U ovom radu je predstavljena nova metoda za automatizaciju ispitivanja kondenzatora. Metoda opisana u ovom radu je deo softverskog paketa realizovanog za ispitivanje elektronskih komponenata kao što su optoelektronske komponente, elektronske komponente kao senzori temperature i kondenzatori. U okviru aplikativnog softvera moguće je prikaz izmerenih vrednosti tokom merenja, kao i iscrtavanje grafika zavisnosti izmerenih vrednosti, kao i mogućnost poređenja više zavisnosti.

I. UVOD

Karakterizacija komponenata i materijala pruža uvid u svojstva i oblasti primene elektronskih komponenata. Na osnovu karakteristika određene komponente, korisnik ima mogućnost da na najbolji način odabere komponentu za mrsto primene. Da bi korisnik mogao da dođe do karakteristika iste, neophodno je izvršiti merenja i na osnovu izvršenih merenja i karakterizaciju ispitivane komponente. Na osnovu toga je moguće zaključiti kakve rezultate komponenta daje pri određenim uslovima rada.

U skladu sa tim, realizovana je metoda koja osim samog merenja, odnosno ispitivanja komponenata, pruža i mogućnost poređenja dobijenih rezultata, što u mnogome olakšava dalji rad, odnosno karakterizaciju. Kako bi se proces automatizovao korišćena je virtuelna instrumentacija za čije potrebe je realizovana aplikacija za računar.

II. PREGLED LITERATURE

Virtuelna instrumentacija poslednjih godina sve više preuzima vodeću ulogu u laboratorijama i industriji. Konkretno, postoji potreba za virtuelnom instrumentacijom u korišćenju instrumenata koji nemaju sopstveni operativni sistem. Iz tog razloga je nemoguće kontrolisati njihovo delovanje bez prisustva ljudskog faktora. Takođe, postoji potreba za virtuelnom instrumentacijom ukoliko je potrebno koristiti više instrumenata za realizaciju testa, odnosno praćenje parametara koji se mere tokom testa u isto vreme.

Miloš Đorđević, student doktorskih studija sa katedre za Mikroelektroniku, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: milos.djordjevic@elfak.ni.ac.rs

Branislav Jovičić, student na smera Računarstvo i informatika, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: badza.jo@gmail.com

U istraživanju [1], autori su koristili softverski paket LabVIEW [2] za realizaciju virtuelne instrumentacije. Kao komunikacioni protokol, autori su koristili USB (eng. *Universal Serial Bus*) protokol, preko koga računar komunicira sa akvizicionom karticom. Realizovana softverska aplikacija koristi se za određivanje otpornosti na laboratorijskim vežbama. Nedostatak ovako realizovanog sistema se ogleda u tome što ne postoji mogućnost skladištenja izmerenih podataka, već je neophodan ljudski faktor, za razliku od metode opisane u ovom radu.

Kod sistema virtuelne instrumentacije koriste se komunikacioni protokoli kao što su serijska komunikacija RS232, korišćenjem U(S)ART (*Universal Synchronous Asynchronous Receiver Transmitter*) serijske komunikacije [3]. Takođe se sreću sistemi zasnovani na standardnoj USB serijskoj komunikaciji, naravno tamo gde instrumenti i sama softverska aplikacija, odnosno softverski paket to omogućavaju. Jedan od najpoznatijih komunikacionih protokola u virtuelnoj instrumentaciji je GPIB (*General Purpose Interface Bus*) IEEE 488 protokol, koji je razvila firma HP (*Hewlett-Packard*), pa se u literaturi može naći i pod nazivom HPIB. Virtuelni instrumenti se mogu podeliti u dve osnovne kategorije:

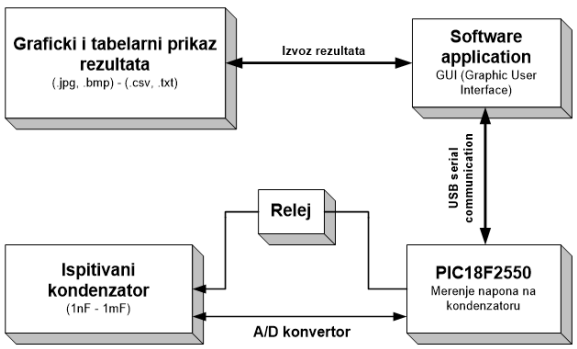
1. Softverska aplikacija za kontrolu modula ili instrumenata kao što su: upravljački modul je dodatna (eng. *Plug-in*) DAK ploča, kontrolisani instrument se zasniva na GPIB ploči, kontrolisani instrument je povezan preko serijskog porta (RS-232) ili USB standarda, kontrolisani instrument je VXI ploča (ili sistem).
2. Grafički korisnički interfejs (eng. *Graphic User Interface (GUI)*) bez fizičkih instrumenata povezan sa računarom. Umesto toga, računar dobija i analizira podatke iz datoteka ili sa drugih računara na mreži, ili može čak i matematički obraditi unete podatke i na taj način simulirati fizički proces, a ne da prikupljati realne podatke. Na računar u skladu sa tačkom (1) povezani su sledeći procesni merni uređaji: Senzori, GPIB ili USB serijski podržani instrumenti.

U ovom radu korišćen je standardna USB serijska komunikacija, kojom je sistem za merenje zasnovan na PIC mikrokontroleru komunicirao sa softverskom aplikacijom na računaru.

III. PRINCIP RADA

Metoda za automatizaciju ispitivanja kondenzatora upotrebom virtuelne instrumentacije, predstavljena u ovom

radu bazira se na PIC mikrokontroleru koji predstavlja srž mernog dela sistema i virtuelne instrumentacije zasnovane na softverskoj aplikacije. Blok šema realizovanog sistema je predstavljena na slici 1.

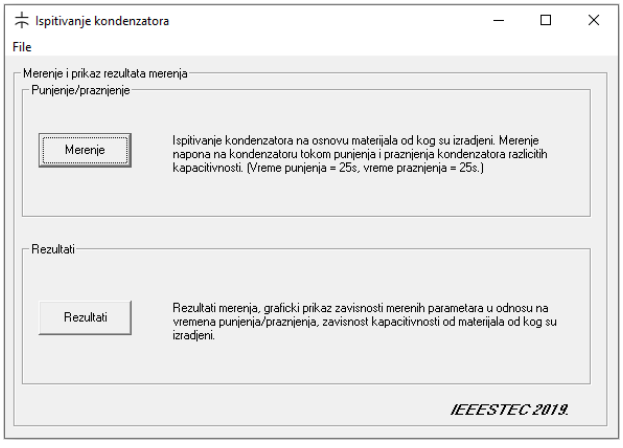


Slika 1. Blok šema metode za automatizaciju ispitivanja kondenzatora.

Realizovana metoda je koncipirana da se sastoji iz 5 segmenata, kao što su:

- Softverska aplikacija (GUI),
  - Kontrola mernog procesa,
  - Obrada i skladištenje podata.
- Mikrokontroler (merni blok):
  - Kontrola punjenja/pražnjenja,
  - Merenje i slanje podataka ka aplikaciji.
- Blok za ispitivanje (ispitivani kondenzator),
- Relejni modul (punjenje/pražnjenje kondenzatora),
- Grafički/tabelarni prikaz rezultata:
  - Grafički prikaz (poređenje) izmerenih rezultata,
  - Tabelarni prikaz izmerenih podataka.

Softverska aplikacija u sebi objedinjuje dva zasebna dela, tačnije sastoji se od dve zasebne softverske aplikacije (Slika 2.), deo namenjen za merenje napona na kondenzatoru tokom procesa punjenja, odnosno pražnjenja istog, respektivno i deo za prikaz rezultata.



Slika 2. Izgled (layout) početne strane softverske aplikacije za ispitivanje kondenzatora.

U okviru aplikacije zadužene za kontrolu mernog procesa, korisnik ima uvid u sve parametre bitne za test. U delu aplikacije koji je namenjen za prikaz rezultata, odnosno poređenje parametara za testirane kondenzatore u vidu grafika. Detaljnija analiza ispitivanja kondenzatora realizovanom metodom se može vršiti na osnovu objedinjenog jezika za modelovanje (UML (eng. *Unified Modeling Language*)), dato na slici 3.

Mikrokontroler korišćen kao centralni deo mernog bloka je PIC18F2550 radi na frekvenciji od 48 MHz [4]. Za postizanje ove frekvencije koristi se eksterni oscilator frekvencije 8 MHz i korišćena je mogućnost mikrokontrolera za povećanjem frekvencije 4 puta (*Primary Clock Frequency 4x multiplier* (PLLEN)) – na 48 MHz. Master reset (MCLR - *Master Clear Pin External Reset*) je softverski isključen, ali je u samoj realizaciji postavljen i pull-up otpornik vrednosti 10 kΩ, kako bi se videla mogućnost hardverskog isključenja MCLR pina na mikrokontroleru. Ovaj mikrokontroler je jedan od malog broja PIC mikrokontrolera koji poseduje modul za standardnu USB komunikaciju, što je razlog zašto je korišćen za realizaciju merne metode.

Pored toga, neophodno je bilo korišćenje analogno-digitalnog (A/D) konvertora, gde je u tu svrhu iskorišćen ugrađeni modul u samom mikrokontroleru, s obzirom da za potrebe realizovane metode nije bilo potrebe za velikom rezolucijom. Korišćeni A/D konvertor je rezolucije 10-bita, što daje mogućnost da uzorkovani signal bude u opsegu od 0 do 1023 naponskih nivoa (sa korakom od 4,89mV). Kao relejni modul preko kog je vršen proces punjenja, odnosno pražnjenja, korišćen je relej 10A/220V. Relej je kontrolisan mikrokontrolerom, koji zavisno od informacija dobijenih od strane softverske aplikacije priključuje kondenzator na napajanje u procesu punjenja, odnosno isključuje u procesu pražnjenja.

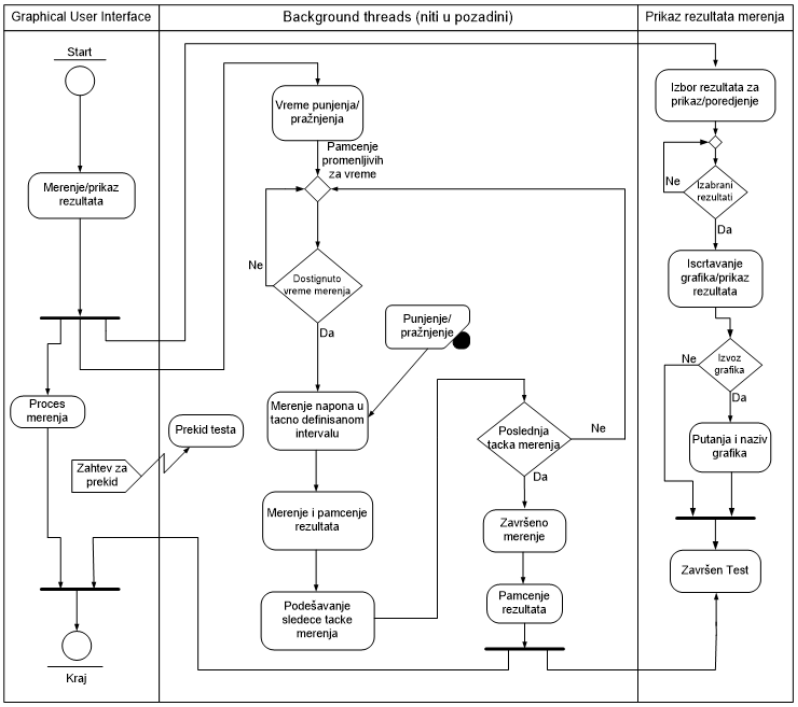
A. Software

Pre samog početka testa neophodno je kreirati fajl iz padajućeg menija - *File*, kao i izbor ekstenzije - *.cvs* ili *.txt*. Nakon toga softverska aplikacija proverava da li je merni blok povezan sa aplikacijom (vizuelna potvrda u donjem levom uglu). Pritiskom na taster „*Merenje*“ pokreće se proces punjenja, gde korisnik u posebno definisanom okviru „*Status*“ prati da li je u toku proces punjenja ili pražnjenja ispitivanog kondenzatora.

Na početku samog testa smatra se da je kondenzator potpuno ispražnjen. Kada je započet procesa punjenja, softverska aplikacija šalje informaciju mikrkontroleru putem USB komunikacije da je potrebno aktivirati relej kojim se dovodi napajanje na kondenzator i započinje njegovo punjenje.

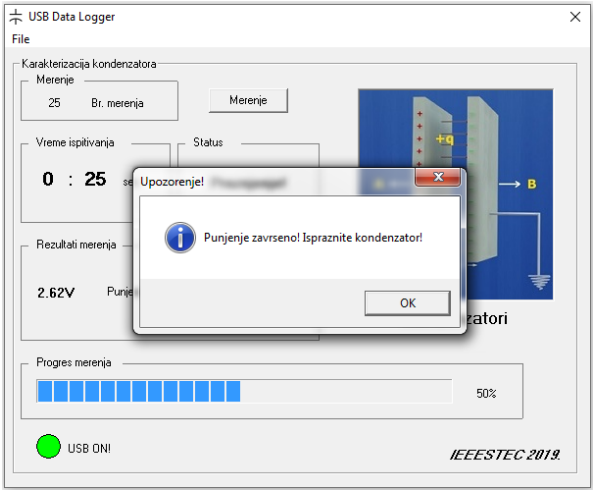
Od tog trenutka mikrokontroler meri napon na kondenzatoru i šalje ka softverskoj aplikaciji, gde se na dalje u tačno definisani intervalima primljeni podaci obrađuju i skladište u kreirani fajl. Korisnik ima uvid u to koliko je tačaka merenja završeno i grafičkim putem u vidu

progres bara (eng. *Progress bar*) i procentualno numeričkim podatkom.



Slika 3. UML dijagram aktivnosti softverske aplikacije za ispitivanje kondenzatora.

Izgled (eng. *Layout*) softverske aplikacije tokom testa kojim se kontroliše merni proces (upravljanje mernim blokom i skladištenje izmerenih podataka) je dat na slici 4.



Slika 4. Izgred (layout) aplikacije za kontrolu procesa merenja.

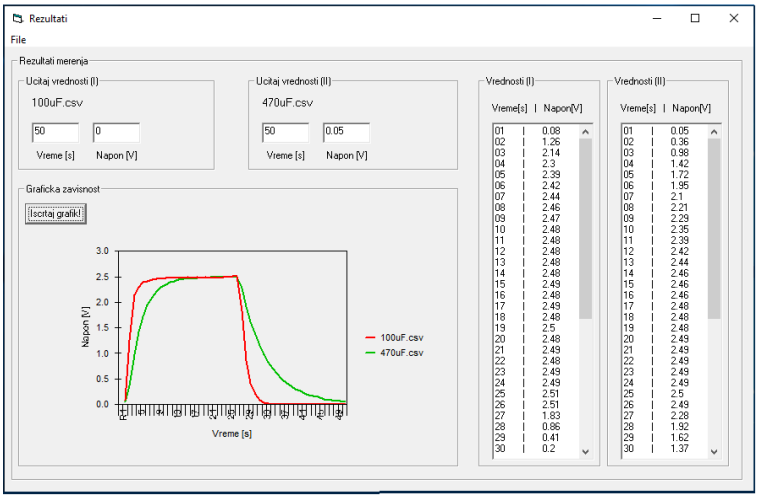
U definisanom okviru „*Rezultati merenja*“ može se pratiti trenutni napon na kondenzatoru. Kada istekne vreme predviđeno za proces punjenja kondenzatora (50% vremena testa), aplikacija u vidu poruke obaveštava korisnika o završenom procesu. Nakon toga je potrebno pokrenuti proces pražnjenja pritiskom na taster „*OK*“, koji traje isti vremenski period kao proces punjenja (u ovom slučaju 25s - 50 sekundi, trajanje kompletnog testa). Nakon završenog testa, korisnik dobija informaciju o završenom testu i softverska aplikacija se iz dela za test vraća na početnu stranu.

Drugi deo metode je vezan za prikaz izmerenih rezultata, odnosno njihovo poređenje. Iz padajućeg menija se biraju rezultati čije će se vrednosti poreditи grafički, propraćeno tabelarnim prikazom njihovih vrednosti. Izgled softverske aplikacije koja je namenjena prikazu rezultata je dat na slici 5. Kao što se može videti, grafički prikaz rezultata, dva ispitivana kondenzatora je propraćeno legendom, kako bi korisnik lakše moga izvršiti poređenje istih. Realizovana metoda za ispitivanje kondenzatora pruža korisniku mogućnost da dobijeni grafik „izveze“ i zapamti pod određenom ekstenzijom (*.jpg* ili *.bmp*). Na taj način korisnik nema potrebu za korišćenje dodatnih softverskih alata, kao što su alati za graičku prezentaciju merenih vrednosti, odnosno tabelarni prikaz istih. Grafička zavisnost dobijena realizovanom metodom u potpunosti odgovara grafičkoj reprezentaciji dobijenom korišćenjem komercijalnog alata kompanije „OriginLab“ [5].

B. Hardware

Kao što je navedeno, mikrokontroler PIC18F2550 je iskorišćen kao centralna komponenta mernog bloka. Za merenje napona na kondenzatoru korišćen je analogni ulaz

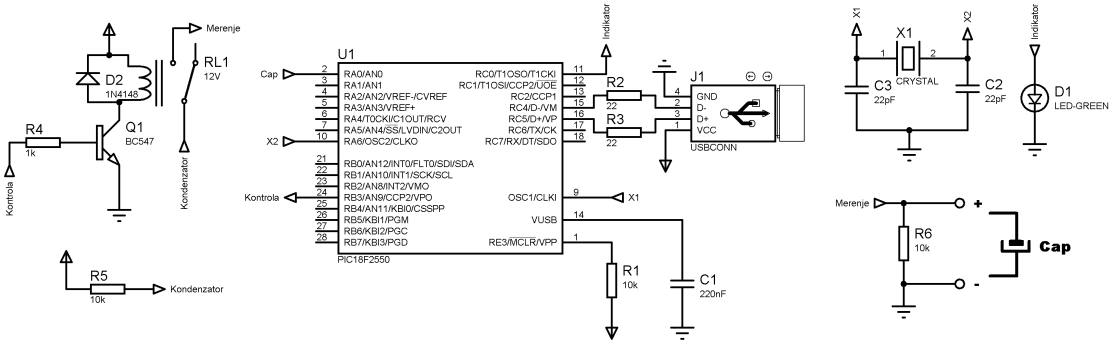
na pinu PORTA.0 (prvi kanal A/D konvertora). Relejni modul je na mikrokontroler povezan na pin PORTB.3.



Slika 5. Grafički i tabelarni prikaz izmerenih rezultata.

Komunikacija mikrokontrolera sa softverskom aplikacijom je ostvarena putem standardne USB komunikacije preko otpornika vrednosti 22Ω koji su vezani na pinove PORTC.4/D- i PORTC.5/D+, respektivno.

Električna šema mernog bloka realizovane metode je data na slici 6.



Slika 6. Električna šema realizovanog mernog sistema metode za ispitivanje kondenzatora.

V. ZAKLJUČAK

Virtualna instrumentacija u obliku softverske aplikacije predstavljena u ovom radu, pokazala se vrlo korisnom jer je nekoliko instrumenata povezano pomoću jednog mernog sistema. Budući da instrumenti koji se koriste nemaju operativni sistem, predložena metoda omogućila je automatizaciju upotrebe instrumenta i njihovu kontrolu. Što je još važnije, eksperimentalni rezultati dobijeni ovom metodom podudaraju se sa dobijenim rezultatima korišćenjem komrcijalnih softverskih alata.

ZAHVALNICA

Autori rada se zahvaljuju finansijskoj podršci Ministarstva prosvete, nauke i tehnološkog razvoja za projekat TR33035.

LITERATURA

[1] I. Garovic, B. Koprivica i A. Milovanović, "Primena Virtuelne instrumentacije u realizaciji laboratorijskih vežbi iz oblasti Električnih merenja", *Tehnologija, Informatika i Obrazovanje za društvo učenja i znanja, 6. Međunarodni simpozijum*, Čačak, Jun 2011.

[2] LabVIEW – softverski paket za realizaciju virtuelne instrumentacije, dosupno na: https://www.ni.com/en-rs/shop/labview.html.

[3] U(S)ART serijska komunikacija – odeljak, dostupno na: http://es.elfak.ni.ac.rs/pld/Materijal/UART.pdf.

[4] Mikrokontroler PIC18F2250 – Datasheet, dostupno na: https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf.

[5] Origin 8 Pro – Komercijalni softverski alat za grafičku i tabelarnu prezentaciju podataka, dostupno na: https://www.originlab.com.

Upravljanje asinhronim motorima u prehrambenoj industriji

Vladimir Spasojević, Uglješa Jovanović, Igor Jovanović

Sadržaj – U ovom radu prikazano je projektovanje i realizacija upravljačkog sklopa motornog pogona sistema za doziranje kakao krem mase. Korišćen je upravljački sistem koji je zasnovan na ON/OFF regulaciji motornog pogona. Isključivanje upravljanog motornog pogona direktno zavisi od zadatog broja detektovanih (ostvarenih) rotacija. Broj ostvarenih rotacija motora detektuje se induktivnim senzorom koji se ugrađuje na pokretni deo motora. Pokretanje motora se vrši preko ekrana osetljivog na dodir, pomoću kontaktora kojim upravlja komandni relej. Sama realizacija kontrole je implemetirana na razvojnom okruženju Raspberry Pi, kao brojanje prolaza induktivnog materijala u određenoj tački. Detekcija prolaza odgovara broju realizovanih punih krugova u sistemu za doziranje.

I. UVOD

Moderni tehnološki procesi zahtevaju složene načine pokretanja i upravljanja motornim pogonima s obzirom da najveći broj industrijskih potrošača čine trofazni asinhroni motori [1].

Upotreba motora u industriji zahteva precizno upravljanje njihovim radom. Pod upravljanjem motorima podrazumevaju se akcije kao što su: pokretanje i zaustavljanje radnog pogona, promena smera obrtanja rotora, kao i kontrola brzine i elektromehaničkog momenta [1]-[3].

U radu je prikazana realizacija adaptivnog sistema upravljanja radom motornih pogona na osnovu radne temperature i broja obrtaja. Sistemi za kontrolu broja obrtaja imaju veoma značajnu primenu u prehrambenoj industriji, kao i u industriji proizvodnje i obrade papira, gume, metala, tekstila i drugih materijala. Nezavisno od faze tehnološkog procesa, potrebna je velika preciznost brojanja i velika brzina obrađivanja podataka od strane upravljačkog sistema. U ovom radu razmatran je problem brojanja ostvarenih punih obrtaja asinhronih motora koji se koriste u sistemu za doziranje kakao krem mase.

II. REALIZACIJA SISTEMA ZA DOZIRANJE KAKAO KREM MASE

Doziranje kakao krem mase zadovoljavajućom preciznošću, u procesu njenog pakovanja u proizvodnu ambalažu, može se izvršiti uključivanjem i isključivanjem asinhronog motora koji pogoni pumpu za doziranje. Zbog velike viskoznosti i gustine kakao krem mase, za njeno doziranje najčešće se koristi zupčasta pumpa koja se pokreće asinhronim motorom sa kočnicom, a čija se brzina smanjuje reduktorom. Jednim punim obrtajem pumpe dozira se oko 46 cm^3 kako krem mase. Minimalna potrebna snaga asinhronog motora dobijena je eksperimentalnim putem i iznosi oko 750 W.

Ulaz pumpe za doziranje je cevovodom povezan sa sabirnim sudom kakao krem mase, a na njen izlaz montirana je lula za doziranje na način da se nalazi iznad radne površine, odnosno iznad mesta gde se postavlja proizvodna ambalaža za pakovanje gotovog proizvoda.

Količina dozirne mase meri se brojem punih obrtaja, a radi veće preciznosti sistema za doziranje, brojanje punih obrtaja se vrši pre redukcije brzine asinhronog motora. Detekcija punog obrtaja se izvodi induktivnim senzorom koji se montira pored pokretnog dela motora.

Prilikom samog procesa doziranja kakao krem mase temperatura iste ne sme preći 55°C pa je potrebno implementirati i hlađenje dozirane mase. Ukoliko temperatura dozirane kakao krem mase premaši 50°C neophodno je da sistem automatski aktivira vizuelni alarm kojim će upozoriti operatera na potencijalni problem. Za pokretanje pumpe za doziranje korišćen je trofazni asinhroni motor sa kratkospojenim rotorom [4].

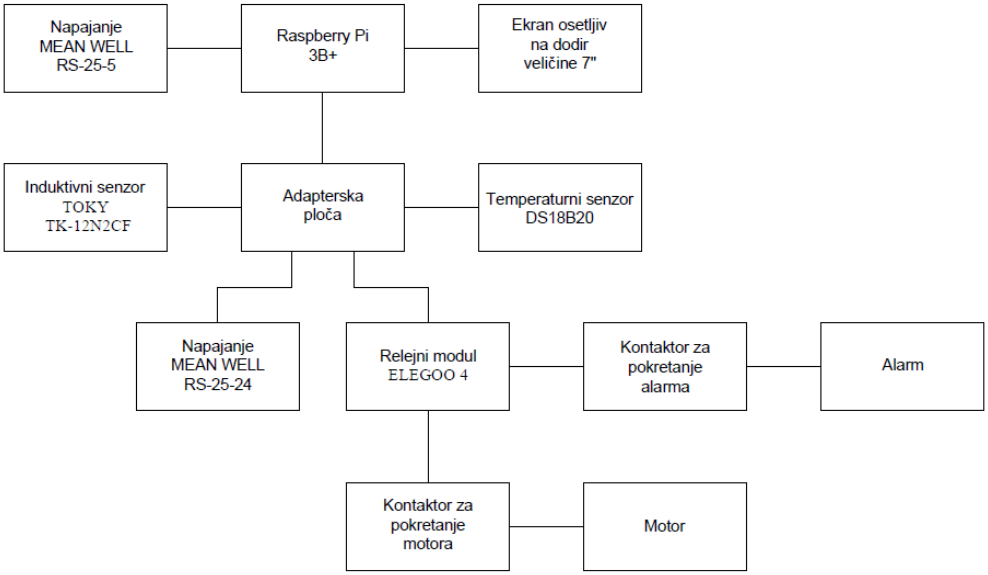
Najčešće se za industrijske primene koriste diskretni brojači sa relejnim komponentama [5]. Primena relejnih upravljačkih sistema podrazumeva korišćenje vremenskih releja, brojača signala, temperaturnih releja i drugih releja koji su zasnovani na relativno jednostavnim logičkim algoritmima. Prilikom rešavanja zadataka upravljanja ovom metodom, neophodno je podeliti sistem na više manjih i jednostavnijih sistema. Prednost ovakvog načina automatizacije je jednostavnost izrade, a nedostatak je velika cena upravljačkog sistema. Dodatni nedostatak predstavlja nemogućnost automatskog prilagođenja na promene radnih uslova pa je u tom slučaju neophodno ručno menjati i deo hardvera.

Ubrzani razvoj mikroprocesora osamdesetih godina doveo je do upotrebe istih u rešavanju složenih upravljačkih zadataka, akviziciji podataka i daljinskom upravljanju. Prednosti programabilno logičkih kontrolera (PLC) u odnosu na relejnu tehniku su: manja cena, veća fleksibilnost, mogućnost vizuelnog praćenja rada, veća brzina rada, jednostavna izmena programa i korekcija grešaka [6].

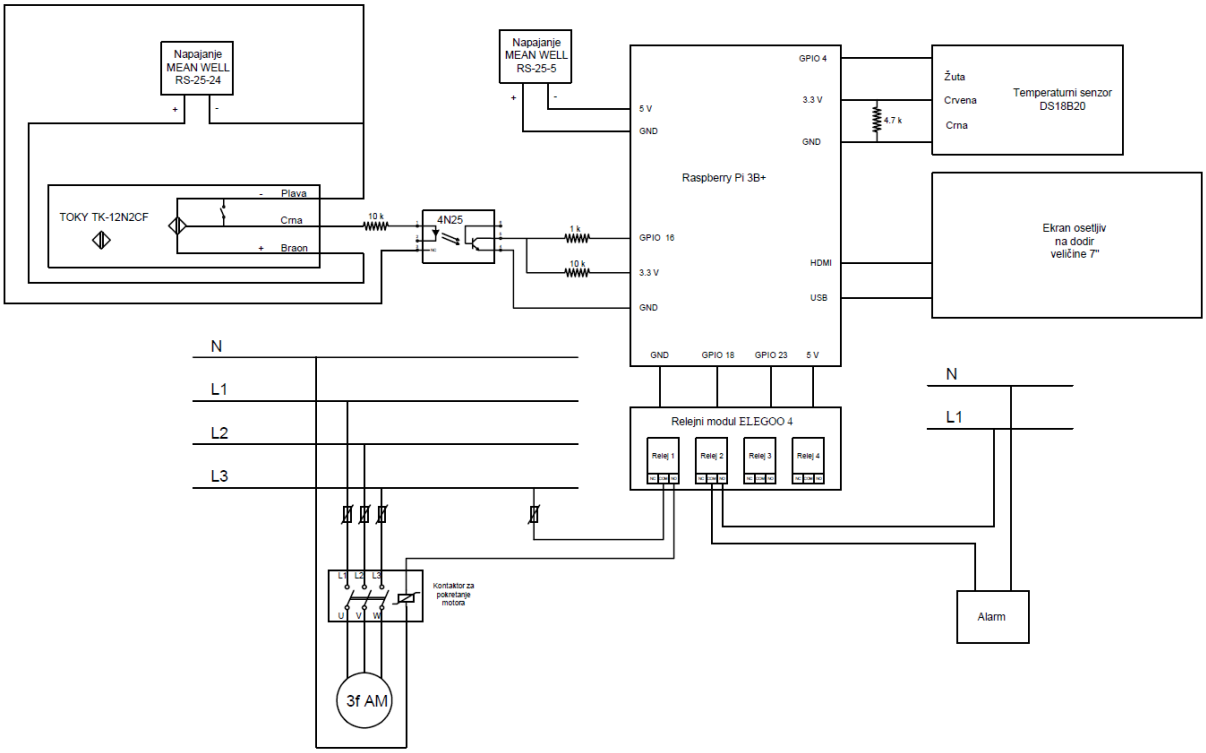
Vladimir Spasojević, Uglješa Jovanović i Igor Jovanović, Univerzitet u Nišu, Elektronski fakultet, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: vspasojevicv@gmail.com

Kako su PLC kontroleri dizajnirani za kontrolu u industriji, odlikuje ih velika pouzdanost i otpornost na prašinu, vibracije, visoku temperaturu i relativnu vlažnost vazduha. Dodatna, prednost korišćenja PLC kontrolera je ta što je kontrola centralizovana i eventualna izmena uslova se lako izvodi menjanjem uslova u leder dijagramu.

U ovom radu prikazana je realizacija sistema za doziranje kakao krem mase zasnovanog na Raspberry Pi razvojnom okruženju. U predloženom rešenju sve matematičke i logičke operacije kontrole upravljanja i nadzora procesa upravljanja obavlja 32-bitni procesor. Jedna od najvažnijih prednosti predloženog sistema u odnosu na sisteme zasnovane na PLC kontroleru je mogućnost direktne komunikacije sa temperaturnim senzorima. Blok šema sistema zasnovanog na Raspberry Pi razvojnom okruženju prikazana je na slici 1.



Slika 1. Blok šema sistema realizovanog pomoću Raspberry Pi razvojnog okruženja.



Slika 2. Šema realizovanog sistema zasnovanog na Raspberry Pi razvojnom okruženju.

Električna šema realizovanog sistema zasnovanog na Raspberry Pi razvojnom okruženju prikazana je na slici 2. Izgled prototipa realizovanog sistema prikazan je na slici 3.

Komponente korišćene za realizaciju sistema su: Raspberry Pi model 3B+, temperaturna sonda bazirana na senzoru DS18B20, induktivni senzor TK-12N2CF, relejni modul ELEGOO 4, adapterski modul za galvansku izolaciju (slika 2), DC napajanje od 5 V (MEAN WELL RS-25-5), DC napajanje od 24 V (MEAN WELL RS-25-24) i LCD ekran.

U zavisnosti od modela, na Raspberry Pi razvojnom okruženju se nalazi ARM procesor, sa jednim ili 4 jezgra, radne frekvencije između 700 MHz i 1.7 GHz, a operativni sistem je smešten na microSD kartici. Opremljen je RAM memorijom veličine od 256 MB do 4 GB i poseduje od jednog od do četiri USB priključaka. Svi modeli poseduju priključak za video izlaz čiji tip zavisi od konkretnog modela. Neki modeli poseduju i zvučnu karticu. Pored pomenutih priključaka, svi modeli su opremljeni i GPIO (General-purpose input/output pins) pinovima kojih ima između 26 i 40. GPIO pinovi podržavaju standardizovane protokole komunikacije kao što su I²C, SPI i UART



Slika 3. Izgled realizovanog sistema zasnovanog na Raspberry Pi razvojnom okruženju.

Relejni moduli u klasi modula ELEGOO 4 se koriste za kontrolu strujnih kola veće snage, a najčešću primenu nalaze u industriji i pametnim kućama. S obzirom na to da se naponski nivoi induktivnog senzora TK-12N2CF i procesora Raspberry Pi okruženja bitno razlikuju (senzor radi na 10 V, a procesor na 3.3 V) neophodno je izvršiti njihovo naponsko prilagođenje. Jedno od najefikasnijih rešenja predstavlja primena optokaplera (4N25 e) kojim se vrši galvanska izolacija.

III. IMPLEMENTACIJA REALIZOVANOG SISTEMA

Radni ciklus se pokreće pomoću kontrola dostupnih na grafičkom interfejsu prikazanom na LCD ekranu, a prilikom pokretanja ciklusa pokreće se asinhroni motor koji pogoni sistem za doziranje. Pokretanje asinhronog motora vrši se preko kontaktora kojim se upravlja pomoću releja 1 sa relejnog modula ELEGOO 4, a kojim upravlja GPIO pin 18 sa Raspberry Pi okruženja. Aktiviranjem kontaktora dovode se sve tri faze na stator asinhronog motora.

Usled rotacije rotora asinhronog motora, induktivni senzor detektuje prolazak induktivnog materijala što predstavlja jednu punu rotaciju rotora motora. Impulsi koje generiše induktivni senzor se preko optokaplera (smeštenog na adapterskoj ploči) prenose na GPIO pin 16 Raspberry Pi okruženja.

Svaka detekcija punog obrtaja asinhronog motora inkrementira vrednost promenljive “trenutnaVrednost” za jedan, a asinhroni motor se okreće sve dok važi uslov da je vrednost promenljive “trenutnaVrednost” manja od vrednosti promenljive “zadataVrednost”. U suprotnom, GPIO pin 18 deaktivira relej 1 usled čega se prekida rad asinhronog motora. S obzirom na to da je sistem doziranja kako krem mase inertan, rotor asinhronog motora će nastaviti da se okreće još nekoliko krugova čak i nakon isključenja motora sa napojne mreže. Radi poboljšanja preciznosti sistema za doziranje kakao krem mase, može se koristiti asinhroni motor sa mehaničkom kočnicom koja se aktivira u trenutku kada se stator motora odvoji sa napojne mreže.

Drugi deo kontrole odnosi se na merenje temperature i na aktiviranje alarma. Temperatura kakao krem mase se detektuje temperaturnom sondom baziranom na senzoru DS18B20. Raspberry Pi kontroler šalje upite o vrednosti temperature i ispisuje ih na LCD ekranu u delu “Radna temperatura u °C”. Kada sistem detektuje temperaturu višu od 50°C aktivira se relej 2 kojim se upravlja preko GPIO pina 23, a relej 2 upravlja radom kontaktora za pokretanje alarma. Alarm može biti zvučne i/ili svetlosne prirode.

Na LCD ekranu prikazan je grafički korisnički interfejs koji sadrži komande za pokretanje i zaustavljanje ciklusa kao i trenutno stanje sistema. Grafički korisnički interfejs kreiran je pomoću programskog jezika Python. Grafički interfejs se sastoji od komandnih tastera i indikatora stanja. Pre pokretanja ciklusa potrebno je podesiti zadatu vrednost obrtaja asinhronog motora koja se prikazuje u delu označenim labelom “Zadata vrednost“. Inkrementiranje ove vrednosti vrši se pomoću tastera sa simbolom “+“, a dekrementiranje pomoću tastera sa simbolom “-“, koji se nalaze pored zadate vrednosti obrtaja. Promena zadate vrednosti nije dozvoljena tokom samog ciklusa brojanja. U slučaju pokušaja promene zadate

vrednosti u “Polju Upozorenja“ biće prikazano da je menjanje zadate vrednosti zabranjeno tokom trajanja ciklusa.

Imajući u vidu da vrednost broja obrtaja asinhronog motora ne može biti negativna, minimalna zadata vrednost je limitirana na 0. Broj obrtaja asinhronog motora može se pratiti u polju označenom labelom “Trenutna vrednost“.

Pokretanje radnog ciklusa vrši se pritiskom na taster “START“ i zaustavlja se pritiskom na taster “STOP“. Nakon uspešnog pokretanja ciklusa, automatski se čekira indikator “Motor radi“ koji predstavlja indikaciju uključenosti motora. Pokretanje ciklusa podrazumeva automatsko resetovanje trenutne vrednosti broja obrtaja asinhronog motora. Trenutnu vrednost moguće je i ručno resetovati pritiskom na taster “Resetuj trenutnu vrednost“.

Prikazivanje trenutne radne temperature u sistemu za doziranje označeno je labelom “Trenutna temperatura u °C“, a podešavanje temperature za aktiviranje alarma vrši se pomoću tastera sa simbolima “+“ i “-“ koji se nalaze pored prikaza zadate temperature. Indikacija uključenosti alarma nalazi se pored prikaza trenutne temperature. U slučaju kada vrednost trenutne temperature nije u zadatim granicama alarm se aktivira i polje alarma se oboji crvenom bojom. Realizovani sistem uspešno je implementiran kao deo proizvodnog procesa za proizvodnju i doziranje kakao krem proizvoda i koristi se u firmi Gold Pack Mladenovac. Izgled dozirnog mesta prikazan je na slici 4. Zbog potreba procesa proizvodnje izvedena su još dva sistema za merenje temperature. Prvi se koristi za praćenje temperature pumpe i kontrolu njenog grejanja, a drugi služi za praćenje temperature tople vode. Pre doziranja potrebno je zagrejati pumpu za doziranje kako bi se zaostala krem masa u samoj pumpi otopila. Grejači se nalaze u kućištu pumpe i ona se greje temperaturom od 45°C. Topla voda se u duplikatorskom sudu i oko cevovoda krem mase koristi za otapanje krem mase pre procesa doziranja. Krem masa u duplikatorskom sudu i cevovodima se pre doziranja takođe otapa na temperaturi od 45°C.



Slika 4. Mesto za doziranje kakao krem mase.

IV. ZAKLJUČAK

U ovom radu prikazana je realizacija sistema za doziranje kakao krem mase čiji je zadatak upravljanje radom trofaznog asinhronog motora koji pokreće sistem za doziranje i merenje temperature kakao krem mase prilikom doziranja. Parametar na osnovu kog se vrši upravljanje je broj punih obrtaja pogonskog motora od koga direktno zavisi koliko će krem mase sistem dozirati u proizvodnu ambalažu.

Prikazani sistem je realizovan upotrebom razvojnog okruženja Raspberry Pi, što omogućava veliku prednost u vidu jednostavne implementacije korišćenih temperaturnih senzora koji za komunikaciju sa procesorom zahtevaju samo dva provodnika maksimalne dužine do 100 m. Zahvaljujući jedinstvenim adresama koje svaki senzor poseduje, moguće je vršiti adresiranje svakog senzora pojedinačno na isti par provodnika. Takođe, sistem zasnovan na Raspberry Pi razvojnom okruženju u odnosu na sistem zasnovan na PLC kontroleru ima višestruko nižu cenu.

Realizovani sistem doziranja moguće je primeniti na smese slične viskoznosti i gustine uz određene mehaničke promene samog sistema za doziranje (pumpe za doziranje).

ZAHVALNICA

Istraživanja prezentovana u ovom radu su finansirana od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije u okviru projekata TR33035.

LITERATURA

[1] E. Omorogiuwa, C. Ayor, “Energy Efficiency Optimization Of Three Phase Induction Motor Drives For Industrial Applications”, *International Journal of Engineering and Applied Sciences (IJEAS)*, vol. 5, no. 8, pp. 42-42, ISSN: 2394-3661, 2018.

[2] Z.P. Stajić, Đ.R. Vukić, M.M. Radić, *Asinhrone mašine*, Univerzitet u Nišu, Elektronski fakultet u Nišu, Niš, 2012, ISBN: 978-86-6125-066-8.

[3] R. Roy, S. Das, J. Kumar, S. Barat, B. Neogi, “Automatic Speed Control of Single Phase Induction Motor with the Variation of Ambient Temperature”, *International Journal of Scientific and Research Publications*, vol. 2, no. 11, 2012.

[4] S. A. Lingawar, S. M. Ingale, “Speed Monitoring and Control Of Three Phase Induction Motor for Batching Motion System of Textile Industry”, *International Journal of Scientific and Research Publications*, vol. 3, no. 6, pp. 2090-2095, 2013.

[5] R. Delpoux, L. Hetel, A. Kruszewski, “Parameter-Dependent Relay Control: Application to PMSM“, *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1628 - 1637, 2015.

[6] M. Bandyopadhyay, N. Mandal, S. Chatterjee, “PLC based flow control system using a motor operated valve“, *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018, doi: 10.1109/ICCAR.2018.8384

Primena programskog paketa PSCAD za analizu rada odvodnika prenapona

Tamara Mitić

Sadržaj – U radu je izvršena analiza uticaja direktnog atmosferskog pražnjenja u fazni provodnik nadzemnog voda na rad energetskog transformatora. Razmatran je slučaj kada u okolini energetskog transformatora ne postoji odvodnik prenapona, ali i slučajevi kada se odvodnik prenapona ugrađuje na različitim mestima u odnosu na energetski transformator. Analiza se bazira na simulaciji u mreži naponskog nivoa 220 kV korišćenjem programskog paketa PSCAD (Power Systems Computer Aided Design).

I. UVOD

Atmosfersko pražnjenje je pražnjenje statičkog elektriciteta koje se dešava u atmosferi. Može se odvijati unutar grmljavinskih oblaka, između oblaka, od oblaka prema vazduhu u atmosferi, u samom vazduhu kao i između oblaka i zemlje i obratno. Usled atmosferskog pražnjenja u deo elektroenergetskog sistema javljaju se veliki prenaponi koji mogu uništiti opremu u okviru sistema. Za zaštitu opreme od atmoferskih prenapona koriste se odvodnici prenapona.

Odvodnici prenapona [1] predstavljaju elektroenergetske komponente koje služe za zaštitu opreme i sistema od atmosferskih i sklopnih prenapona. Oni funkcionišu kao elementi promenljive otpornosti. Pri pojavi prenapona dolazi do znatnog smanjenja otpornosti odvodnika, pri čemu se omogućava odvođenje energije. Nakon nestanka prenapona otpornost odvodnika se vraća na početnu vrednost. Pravilno izabran odvodnik prenapona, izložen nominalnom naponskom nivou, se ponaša kao velika otpornost sa strujom curenja reda veličine mikroampera i ne utiče na pravilan rad ostale opreme u sistemu.

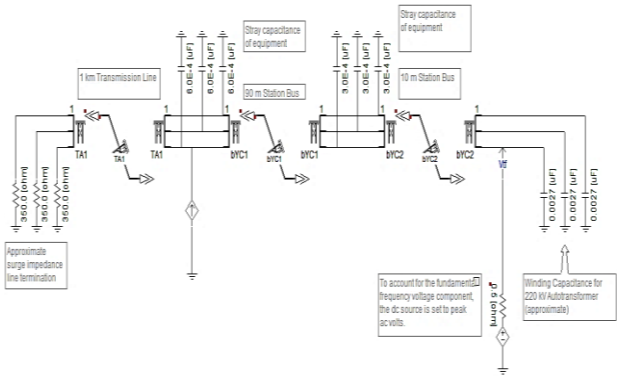
Glavna tema rada je analiza uticaja direktnog amosferskog pražnjenja u fazni provodnik nadzemnog voda kada u sistemu nema odnosno ima odvodnika prenapona pomoću programskog paketa PSCAD (Power Systems Computer Aided Design). Razmatrano je i kako mesto postavljanja odvodnika prenapona utiče na efikasnost zaštite odvodnika prenapona.

PSCAD je moćan i fleksibilan programski alat koji pruža mogućnost formiranja električnih kola, pokretanja simulacije, analize rezultata simulacije, kao i mogućnost promene parametara unutar grafičkog okruženja. Ovaj programski alat omogućava i crtanje grafika, kontrolu procesa i merenje. Dozvoljava korisniku da u toku simulacije menja parametre u sistemu i na taj način uoči kako sistem reaguje na promenu parametara.

Tamara Mitić je student master studija na modulu za Elektroenergetiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: tamara.mitic@elfak.rs

II. PSCAD SIMULACIJA SISTEMA BEZ ODVODNIKA PRENAPONA

Simulacija, koja se analizira u nastavku rada, bazira se na posmatranju dejstva direktnog atmosferskog pražnjenja u jednu fazu nadzemnog voda na energetski transformator u sistemu naponskog nivoa 220 kV. Atmosfersko pražnjenje odvija se direktnim udarom u fazni provodnik na ulazu u transformatorsku stanicu. Na slici 1. prikazana je šema koja se koristi prilikom simulacije. Vreme trajanja svih simulacija je 0.4 ms.

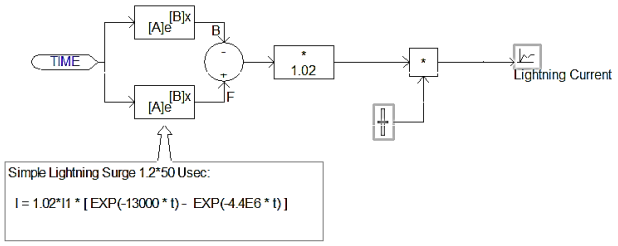


Slika 1. Šema sistema koja se koristi za simulaciju.

U šemi na slici 1. za simulaciju je uzeta vrednost približne prenaponske impedanse prenosnog voda od 350 Ω. Lutajuća kapacitivnost opreme između prenosnog voda i sabirnice je 0,0006 μF, dok je između sabirnica i 0,0003 μF. Za dužinu trase prenosnog voda uzeta je vrednost od 1 km, dok je dužina prvog sabirničkog portal 90 m, a drugog 10 m. Kapacitivnosti namotaja 220 kV energetskog transformatora aproksimirane su vrednostima od 0,0027 μF. U šemi je sa TA1 označen prenosni nadzemni vod. Sabirnice ispred transformatora modelovane su stubovima bYC1 i bYC2 na kojima se nalaze fazni provodnici [2].

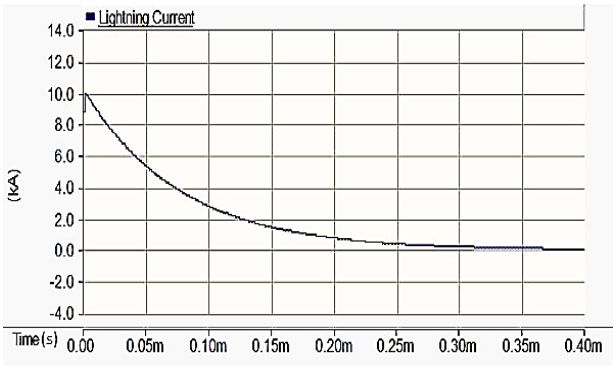
Atmosfersko pražnjenje se u simulaciji modeluje strujnim izvorom koji injektira struju u fazni provodnik na mestu udara groma [2]. Struja atmoferskog pražnjenja se predstavlja funkcijom sa eksponencijalno rastućim čelom i eksponencijalno opadajućim začeljem i standardnim

impulsom 8/20 μs/μs. Ona se u simulaciji modeluje korišćenjem blok dijagrama prikazanog na slici 2.



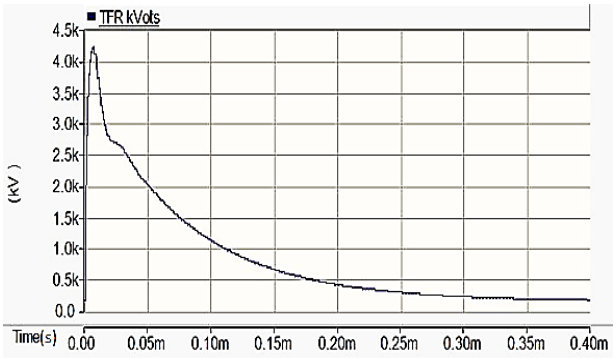
Slika 2. Blok dijagram za modelovanje struje atmosferskog pražnjenja.

U simulaciji za maksimalnu vrednost struje pražnjenja uzima se vrednost Imax = 10 kA. Zavisnost struje atmosferskog pražnjenja od vremena prikazana je na slici 3.



Slika 3. Oblik struje atmosferskog pražnjenja koji se koristi u simulaciji.

Kada u sistemu nema odvodnika prenapona pri direktnom atmosferskom pražnjenju u jednu fazu voda, fazni napon transformatora ima oblik prikazan na slici 4.

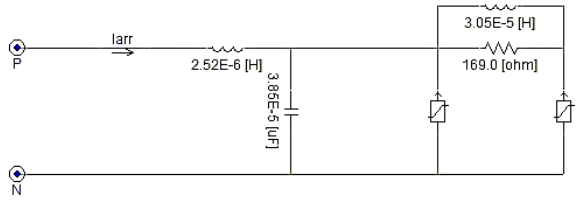


Slika 4. Oblik faznog napona energetskog tranformatora pri atmosferskom pražnjenju bez odvodnika prenapona.

Na slici 4. vidi se da maksimalna vrednost napona atmosferskog pražnjenja prelazi 4000 kV. Visoki naponi mogu dovesti do oštećenja opreme, kao i energetskog transformatora. U ovom slučaju problem predstavlja velika strmina struje atmosferskog pražnjenja zbog čega se indukuje visoki naponi koji prouzrokuje kvarove. Da do toga ne bi došlo, potrebno je postaviti odvodnik prenapona [1].

III. PSCAD SIMULACIJA SISTEMA SA ODVODNIKOM PRENAPONA

Simulacija se vrši kada se koristi odvodnik prenapona proizvođača ABB. Kako se radi o simulaciji zaštite transformatora potrebno je izabrati odvodnik prenapona oznake EXLIM-Q proizvođača odvodnika prenapona koji specificiran za zaštitu transformatora [3]. Naznačeni napon odvodnika prenapona iznosi Ur = 192 kV, dok je maksimalna ostatna vrednost napona odvodnika prenapona pri struji atmosferskog pražnjenja oblika 8/20 μs/μs i amplitude 10 kA iznosi 461 kV po piku. Šema kojom se odvodnik prenapona modeluje u simulaciji prikazana je na slici 5.



Slika 5. Šema za modelovanje odvodnika prenapona.

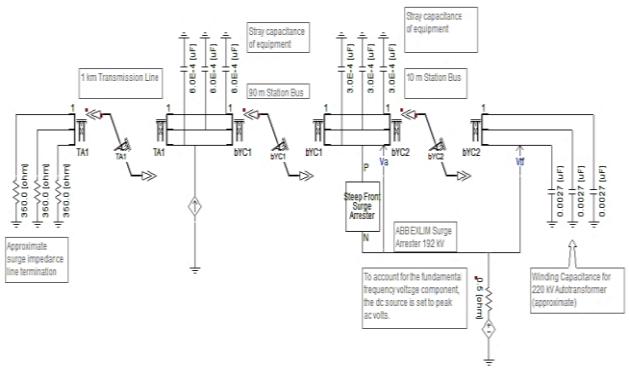
U šemi na slici 5. koriste se dva odvodnika prenapona vezana paralelno kako bi se ostvario bolji efekat odstranjivanja prenapona u sistemu pri brzim početnim delovima prelaznih pojava [2]. Induktivnost u šemi je 2,52 μH i sastoji se od induktivnosti veze koja iznosi 2 μH i induktivnosti odvodnika prenapona od 0,52 μH. Kapacitivnost odvodnika prenapona je 3,85 · 10-5 μF. Odloženi efekat prelaznog uticaja metaloksidnog materijala modeluje se kolom koje čine električni otpornik, čija električna otpornost iznosi 169 Ω, i induktivnosti, koja iznosi 30,5 μH. Sa Iarr označena je struja koja protiče kroz odvodnik prenapona.

Da bi istražili uticaj mesta postavljanja provodnika na efikasnost zaštite razmatrana su dva slučaja:

- odvodnik prenapona postavljen na priključne sabirnice transformatora i
- odvodnik prenapona postavljen na priključke transformatora

A. Odvodnik Prenapona Postavljen na Priključne Sabirnice Transformatora

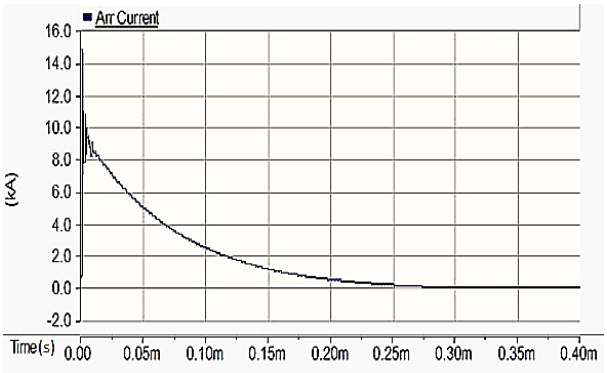
Kada je odvodnik prenapona postavljen na priključne sabirnice transformatora odnosno na 10 m udaljenosti od energetskog transformatora, za simulaciju je korišćena šema prikazana na slici 6.



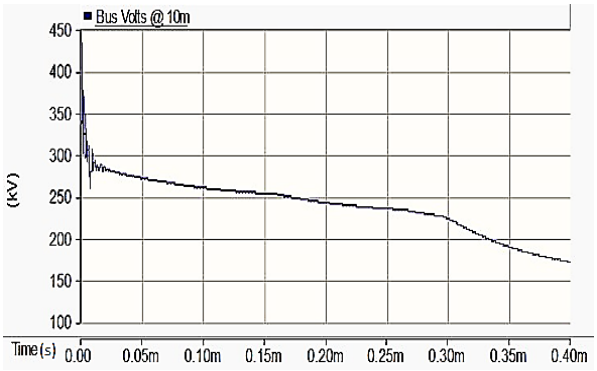
Slika 6. Šema za simulaciju sa odvodnikom prenapona postavljenim na priključne sabirnice transformatora.

Dijagrami struje i napona na odvodniku prenapona prikazani su na slici 7. i slici 8, respektivno. Na slici 9. prikazan je dijagram faznog napona energetskog transformatora.

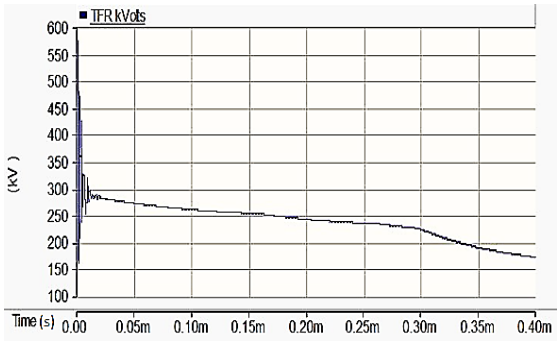
Poređenjem dijagrama koji su prikazani na slici 4. i slici 9. može se zaključiti da je ugradnja odvodnika prenapona neposredno ispred transformatora značajno uticala na smanjenje fazne vrednosti napona transformatora prilikom atmosferskog pražnjenja. Na taj način ostvaren je cilj zaštite transformatora pri atmosferskom pražnjenju. Na slici 7. i slici 8. može se videti da struja i napon odvodnika prenapona odgovaraju performansama odvodnika prenapona koje su definisane u katalogu proizvođača.



Slika 7. Dijagram struje odvodnika prenapona za Imax = 10 kA.



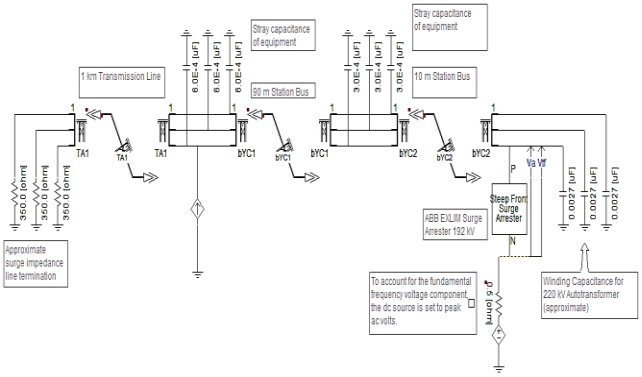
Slika 8. Dijagram napona odvodnika prenapona za Imax = 10 kA.



Slika 9. Dijagram faznog napona energetskog transformatora sa odvodnikom prenapona za Imax = 10 kA.

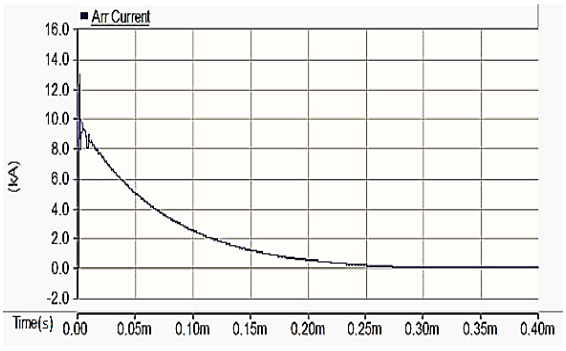
B. Odvodnik Prenapona Postavljen na Priključne Krajeve Transformatora

Kada je odvodnik prenapona postavljen na priključne krajeve transformatora za simulaciju se koristi šema prikazana na slici 10.



Slika 10. Šema sistema kada se odvodnik prenapona montira na priključne krajeve energetskog transformatora.

U ovom slučaju napon odvodnika prenapona i fazni napon energetskog transformatora imaju iste vrednosti. Na slici 11. i slici 12. prikazani su dijagrami zavisnosti struje odvodnika prenapona i napona odvodnika prenapona od vremena, respektivno.

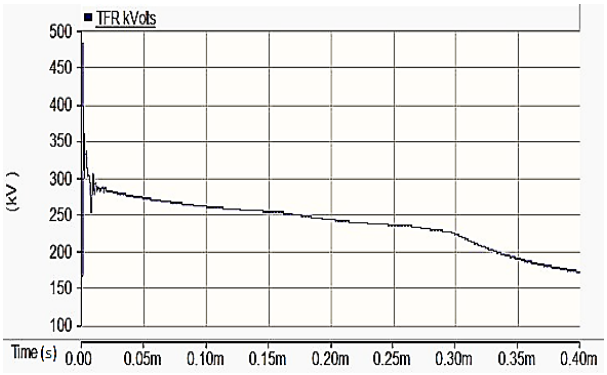


Slika 11. Dijagram struje odvodnika prenapona sa odvodnikom prenapona na priključcima energetskog transformatora za Imax = 10 kA.

Kada se posmatra struja odvođenja odvodnika prenapona, poređenjem Slike 11. sa Slikom 7. vidi se da pri istim uslovima atmosferskog pražnjenja struja odvođenja odvodnika prenapona ima manju vrednost u slučaju kada se odvodnik prenapona nalazi na priključnim krajevima energetskog transformatora. Ukoliko se uporedi dijagram sa slike 12. sa dijagramom sa slikom 8, može se primetiti da je, u slučaju kada se odvodnik prenapona montira na priključne krajeve energetskog transformatora, napon odvodnika prenapona viši nego u slučaju kada se on nalazi na priključnim sabirnicama, ali se nalazi unutar dozvoljenih vrednosti definisanih katalogom proizvođača za posmatrane uslove. Međutim, fazni napon energetskog transformatora ima manju maksimalnu vrednost kada se odvodnik prenapona montira na priključne krajeve energetskog transformatora u odnosu na slučaj kada se nalazi na priključnim sabirnicama što se može videti poređenjem dijagrama sa slike 12. sa dijagramom sa slike 9.

IV. ZAKLJUČAK

Korišćenjem programskog paketa PSCAD pokazalo se da je transformator, kada nema odvodnika prenapona u blizini, ugrožen ukoliko dođe do direktnog atmosferskog pražnjenja u fazni provodnik nadzemnog voda. Kako bi se proverilo da li transformator može da se zaštiti od atmosferskog pražnjenja korišćen je odvodnik prenapona proizvođača ABB. Razmatrano je i kako mesto postavljanja odvodnika prenapona utiče na zaštitu transformatora. Na osnovu simulacije zaključuje se da mesto postavljanja odvodnika prenapona utiče na zaštitu transformatora i to tako da što je odvodnik prenapona bliže postavljen transformatoru to je bolja zaštita transformatora.



Slika 12. Dijagram napona odvodnika prenapona sa odvodnikom prenapona na priključcima energetskog transformatora za Imax = 10 kA.

LITERATURA

[1] M. Savić, Z. Stojković „Tehnika visokog napona – atmosferski prenaponi", Monografija, Elektrotehnički fakultet, Beograd, 2001.

[2] https://hvdc.ca/knowledge-base/read,article/174/mov-surge-arrester-test-lightning-strike/v:

[3] https://new.abb.com/high-voltage/surge-arresters/high-voltage-arresters/protection-of-high-voltage-equipment/substation-applications-porcelain-housed-surge-arrester-exlim

Online aplikacija za prikaz aktivacionog sloga i asemblerskog kôda C programa

Maša Radenković, Nenad Petrović

Sadržaj – Upravljanje memorijom i njena organizacija u toku izvršenja programa jedna je od bitnih tema koja se obrađuje u okviru kurseva koji se tiču programskih prevodilaca. U ovom radu, predstavljena je implementacija online aplikacije zasnovane na Java tehnologiji i Xtext radnom okviru za vizuelizaciju aktivacionog sloga i prikaz asemblerskog kôda C programa, sa ciljem da studentima olakša razumevanje zadataka iz ove nastavne teme, kao podrška tokom učenja.

I. UVOD

Programski prevodioci predstavljaju jednu od fundamentalnih oblasti koja se obrađuje u okviru studija računarstva, informatike i srodnih disciplina. Ovaj kurs se uglavnom izučava na završnoj godini osnovnih akademskih studija (kao što je slučaj na Elektronskom fakultetu Univerziteta u Nišu [1]) ili na prvoj godini master studija.

Spektar tema obrađenih u ovom kursu obuhvata osnovne pojmove i procese vezane za prevođenje programa: formalne jezike, leksičku, sintaksnu i semantičku analizu, generisanje i optimizaciju međukôda i kôda [2, 3]. Međutim, bitan aspekat vezan za proces generisanja kôda jeste i organizacija i upravljanje memorije u toku izvršenja programa.

Upravljanje memorijom obuhvata različite aspekte: 1) određivanje gde će se u memoriji smestiti kôd programa 2) memorisanje podataka različitih tipova 3) gde će se u memoriji smestiti podaci različitih klasa memorije 4) pristup podacima smeštenim u različitim zonama memorije 5) promena sadržaja memorije u trenutku poziva potprograma i povratka na pozivajući modul. U toku izvršenja programa u memoriji se čuva [2]: 1) kôd programa 2) podaci koji se u programu obrađuju 3) podaci potrebni za poziv, izvršavanje i uspešno vraćanje u pozivajući modul – aktivacioni slogovi.

U ovom radu, prikazana je implementacija online Java aplikacije za vizuelizaciju aktivacionog sloga i generisanje asemblerskog kôda za dati C izvorni kôd (sa pojedinim pojednostavljenjima). Cilj je da se studentima olakša razumevanje zadataka iz ove nastavne teme i pruži alat kao podrška samostalnom učenju. Jedino slično rešenje je desktop apliakcija prikazana u [4], koja omogućava prikaz aktivacionih slogova C/C++ programa za Pep/8 virtuelnu mašinu za edukaciju. Međutim, naše rešenje cilja 8086 arhitekturu, a njegova glavna prednost je veća fleksibilnost i online dostupnost, što eliminiše potrebu za instalacijom.

N. Petrović je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: penenadpi@gmail.com

M. Radenković je student četvrte godine osnovnih akademskih studija, Elektronski fakultet, Univerzitet u Nišu Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: masa.radenkovic@elfak.rs

II. TEORETSKA OSNOVA

A. Organizacija memorije kod polustatičkih jezika

U odnosu na promenu sadržaja i organizacije dodeljene memorije programu u toku njegovog izvršenja, viši programski jezici se dele na statičke, dinamičke i polustatičke. U slučaju statičkih jezika, pre početka izvršenja programa, kompletan kôd programa upisuje se u predviđeni deo memorije i rezerviše se prostor za sve podatke koji se koriste u programu. Struktura memorije ostaje nepromenjena do kraja izvršenja programa. Ovi jezici ne dozvoljavaju rekurzivne pozive potprograma.Za razliku od statičkih jezika, dinamički jezici dozvoljavaju dodavanje novih delova kôda, rekurzivne pozive potprograma, kao i rezervaciju prostora za nove podatke i u toku izvršenja programa. Takođe, podaci koji više nisu potrebni daljem izvršenju uklanjaju se iz memorije.

Polustatički jezici predstavljaju mešavinu elemenata statičkih i dinamičkih jezika, te memoriju delimo u dve zone: statičku i dinamičku. Kod ovih jezika, u programu postoje statički podaci koji se nalaze u memoriji sve vreme dok se program izvršava. Takođe, kôd programa se ne menja tokom njegovog izvršenja. Zbog toga su veličina generisanog kôda i memorijski prostor potreban za smeštanje statičkih podataka poznati još tokom prevođenja. Ove informacije kompilator može iskoristiti da za kôd i podatke odredi lokacije u koje će biti smešteni (apsolutne ili relativne adrese). Iz ovog je jasno da se u statičku zonu smeštaju kôd programa i statički podaci. U programu mogu da postoje rekurzivni pozivi i dinamički podaci za koje se rezerviše i oslobađa memorijski prostor u toku izvršenja. Komponenta kompilatora odgovorna za upravljanje memorijom ne zna dubinu rekurzije u toku prevođenja, ni broj aktivacionih slogova unapred. Ovo implicira da se aktivacioni slogovi koji odgovaraju pozivima potprograma i dinamički podaci smeštaju u dinamičku zonu. Dalje, dinamička zona memorije se deli na dva segmenta: 1) deo koji se organizuje u vidu steka i služi za smeštanje aktivacionih slogova potprograma 2) *heap* za smeštanje dinamičkih podataka. Za razliku od statičke zone, veličina steka i dinamičkog dela memorije mogu se menjati u toku

izvršenja programa. Zbog toga se memorija organizuje tako da stek raste u jednom smeru, a deo za dinamičke podatke u drugom, tako da neiskorišćena memorija ostaje između ova dva dela. Nama je od interesa deo memorije (stek) za smeštanje aktivacionih slogova potprograma, čija je vizuelizacija glavna tema ovog rada. Programski jezik C je odabran kao pogodan pedagoški primer, zato što se razmatranjem polustatičkih programskih jezika obuhvataju i elementi isključivo statičkih i isključivo dinamičkih.

Bitno je napomenuti da kôd programskog jezika C postoji mogućnost eksplicitnog definisanja klase memorije u koju će promenljiva biti smeštena:

- `static`: promenljiva se smešta u statičku zonu memorije, njena vrednost definisana u jednom pozivu potprograma može se koristiti i tokom nekog od narednih poziva
- `auto`: podrazumevano, promenljiva se smešta u stek
- `register`: ako ima slobodnih registara, promenljiva se smešta u neki registar procesora, u suprotnom se ponaša isto kao auto.

B. Aktivacioni slog

Svaki put kada se potprogram pozove, kompilator generiše aktivacioni slog koji sadrži sve informacije neophodne tom pozivu potprograma. Aktivacioni slog u asembleru za 8086 obično ima strukturu kao što je prikazano na slici 1 [2], a može da varira u zavisnosti od jezika, ali i konkretnog kompilatora za taj jezik. Kao što se može videti, na početku sloga se smešta vrednost koju potprogram vraća glavnom programu. Ovu vrednost moguće je vratiti i korišćenjem nekog od procesorskih registara. Zatim, ide region namenjen prenosu vrednosti parametara od glavnog programa koje odgovaraju konkretnom pozivu potprograma, u obrnutom redosledu od navođenja.

Rezultat
Stvarni parametri
Adresa povratka
Kontrolna veza (staro BP)
Stanje procesora u trenutku poziva
Lokalni podaci
Privremeni podaci

Slika 1. Struktura aktivacionog sloga u asemblerskom jeziku 8086.

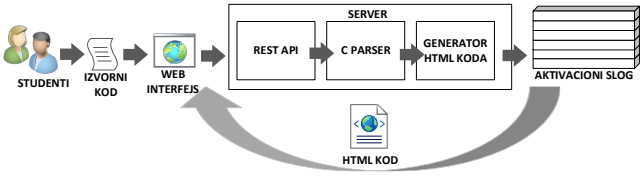
Polje sa upravljačkim linkovima je opciono i koristi se sa ciljem da obezbedi vezu sa aktivacionim slogom modula iz koga je pozvan potprogram. Nakon toga, nalazi se deo aktivacionog sloga namenjen čuvanju statusa procesora (vrednosti registara opšte namene, brojač naredbi). Posle statusa procesora, smeštaju se podaci koji samo pripadaju potprogramu (lokalni podaci). I konačno, sledi polje za memorisanje podataka koje generiše kompilator kao pomoćne lokacije za međurezultate.

III. IMPLEMENTACIJA

A. Arhitektura i princip rada

Aplikaciji se pristupa preko web pretraživača. Klijentski deo napisan je korišćenjem HTML, CSS i JavaScript tehnologije, dok je serverski deo implementiran u Javi. Serverska aplikacija se poziva preko REST API.

Kliejntska aplikacija se sastoji od korisničkog interfejsa u kome studenti mogu da nalepe izvorni C kôd za koji žele da dobiju grafički prikaz aktivacionog sloga. Zatim se, preko REST API poziva serverska aplikacija, koja preuzima C kôd, parsira ga, a za vreme obilaska stabla parsiranja se generiše HTML kôd za prikazivanje aktivacionog sloga i asemblerskog kôda C programa. Generisani HTML kôd se vraća klijentskoj aplikaciji i prikazuje u web pretraživaču. Princip rada aplikacije ilustrovan je na slici 2.



Slika 2. Ilustracija principa rada i arhitekture aplikacije.

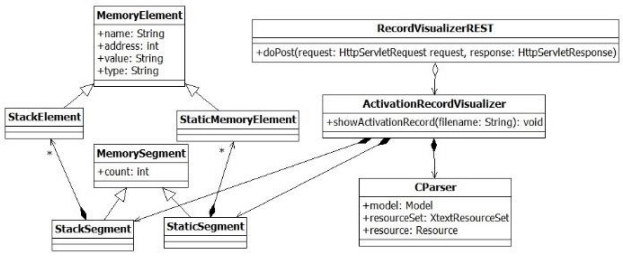
B. UML dijagram klasa serverske aplikacije

Srž funkcionalnosti ove aplikacije jeste njen serverski deo, čiji je dijagram klasa dat na slici 3. Osnovna klasa za predstavljanje memorijskih elemenata jeste MemoryElement. U zavisnosti od toga da li se element čuva u statičkoj zoni memorije ili na steku, imamo dve podklase: StaticMemoryElement i StackElement. Za memorijske elemente se čuvaju adresa, naziv, tip i vrednost. ActivationRecordVisualizer je klasa koja je zadužena za generisanje HTML kôda aktivacionog sloga na osnovu izvornog C kôda. Ona sadrži dve liste elemenata: jednu za one iz statičke zone, a drugu za one koji treba da idu na stek. Ova klasa se za funkcionalnosti parsiranja oslanja na CParser, koji je generisan upotrebom Xtext radnog okvira, a funkcija showActivationRecord ove klase zapravo izvršava algoritme date u Listingu 2 i 3. RecordVisualizerREST je klasa koja pruža REST interfejs ka ActivationRecordVisualizer klasi.

C. Xtext specifikacija gramatike

Za implementaciju C parsera za potrebe generisanja aktivacionog sloga, koristi se Xtext [5, 6]. Ovaj radni okvir omogućava da se na osnovu specifikacije pravila gramatike, terminala i neterminala, automatski generišu sve

klase neophodne za obilazak sintaksnog stabla programa u Javi. Xtext pravila korišćene pojednostavljene C gramatike su data u Listingu 1.



Slika 3. Dijagram klasa serverske aplikacije.

```
grammar org.xtext.example.mydsl.MyC with org.eclipse.xtext.common.Terminals
generate myC "http://www.xtext.org/example/mydsl/MyC"
Model:
	funcdeclarations+=FunctionDeclaration
	'{'
	(declarations+=Declaration)*
	(statements+=Statement)*
	rets+=Ret
    '}'
	;
Ret:
'return' retvalue=ID';'
;
FunctionDeclaration:
returntype=Returntype funcname=ID '('(argdecls+=Argdecl)*')'
;
Returntype:
	'void'|'int'|'double'|'float'
;
Assignment:
left=ID'='INT';'|left=ID'='right=ID';'
;
Binary:
res=ID'='op1=ID Operator op2=ID';'
;
Functioncall:
	functionname=ID '('(args+=Arg)*');'
;
Statement:
 Assignment|Binary|Functioncall
;
Operator:
	'+'|'-'|'*'|'/';
Type:
	'int'|'char'|'double'|'float'
;
Category:
	'static'|'auto'|'register'
;
Arg:
	argname=ID|argname=ID','
;
Argdecl:
	argtype=Type argname=ID','|argtype=Type argname=ID
;
Declaration:
	varcategory=Category vartype=Type varname=ID';'
;
```

Listing. 1. Xtext specifikacija pojednostavljene gramatike programskog jezika C za potrebe generisanja aktivacionog sloga.

Kao što se može videti, nije implementirana kompletna sintaksa C programskog jezika, već samo jedan manji deo, koji je dovoljan za ilustrovanje aktivacionog sloga i njegovog smeštanja na stek, a to je: parsiranje argumenata funkcije, parsiranje deklaracije promenljivih i rezultata koji funkcija vraća. Takođe, konstruisani parser ima mogućnost da rukuje i naredbama dodele, a i pojedinim aritmetičkim operacijama.

D. Algoritmi generisanja kôda

Krajnji rezultat ove pozivanja serverske aplikacije jeste HTML stranica koja sadrži prikaz aktivacionog sloga smeštenog na stek korišćenjem <table> HTML elementa i elementi asemblerskog kôda datog programa. Elementi aktivacionog sloga se generišu postepeno, polazeći od rezervacije prostora za rezultat do lokalnih promenljivih (kao na slici 1). Na kraju se svi dobijeni rezultati konkateniraju, a generisana HTML stranica prikazuje u web pretraživaču.

Algoritam generisanja kôda za prikaz aktivacionog sloga je dat u Listingu 2. Kao što se može videti, tokom obilaska sintaksnog stabla C programa se izdvajaju i prepoznaju odgovarajući elementi – povratna vrednost funkcije, njeni argumenti i deklarisane promenljive. Deklarisane promenljive u zavisnosti od tipa, idu ili na stek (u slučaju auto) ili u statičku zonu memorije (kada su static). Odgovarajući razmaci memorijskih lokacija se računaju po formulama datim u okviru Listinga 1, uzimajući u obzir i broj memorijskih lokacija koje zauzimaju promenljive različitih tipova.

Ulaz: izvorni kôd programa
Izlaz: HTML kôd za vizuelizaciju aktivacionog sloga
1. Kreirati zaglavlje HTML stranice sa tabelom
2. Parsirati izvorni kôd zadatog programa
3. Pronaći sve argumente funkcije iz dobijenog modela programa i dodati u listu
4. Izračunati razmak rezultata funkcije po formuli:
   broj lokacija koje zauzimaju argumenti+broj lokacija rezultata+2(za povratnu adresu)
5. Generisanje vrste u HTML tabeli za rezultat
6. brojač=0;
7. Za svaki od argumenata funkcije polazeći od zadnjeg
   offset=(broj argumenata-brojač)*veličina argumenta+2;
   Generisati vrstu HTML tabele koristeći offset i ime argumenta
   brojač=brojač+1
8. Za svaku deklaraciju promenljive u programu
   If(declaration.varcategory==static) then dodati u StaticSegment.list
   If(declaration.varcategory==auto) then dodati u StackSegment.list
9. Generisati vrstu u tabeli za povratnu adresu
10. Generisati vrstu u tabeli za staru vrednost BP
11. Za svaku promenljivu iz StackSegment.list
    offset=(i+1)*(-veličina promenljive u bajtovima)
    Generisati vrstu HTML tabeli koristeći offset i ime promenljive
12. Generisati HTML kôd za zatvaranje tabele i stranice
13. Nadovezati rezultate HTML kôda iz koraka: 1, 5, 7, 9, 10, 11, 12
14. Vratiti rezultujući string koji sadrži HTML kôd aktivacionog sloga
15. Kraj

Listing. 2. Algoritam za generisanje vizuelizacije aktivacionog sloga u HTML-u na osnovu C kôda.

Nakon generisanja aktivacionog sloga, memorijske lokacije promenljivih su poznate i dodate odgovarajućim listama, pa se može generisati asemblerski kôd. Na početku kôda se dodaje sekvenca instrukcija:

```
PUSH BP                    (1)
MOV BP, SP                 (2)
```

Zatim, nakon toga se alocira prostor za lokalne promenljive po formuli:

$$SUB\ SP, \sum^{broj_p} veličina_promenljive_i \quad (3)$$

U sledećem prolasku kroz sintaksno stablo programa se generišu asemblerske instrukcije, pri čemu se za svaku od promenljivih proverava kojoj listi je prethodno dodeljena. Osim toga, što se poziva funkcija tiče, za svaku

od njih se alocira jedna memorijska lokacija za rezultat na steku. Nakon toga, argumenti funkcije se stavljaju na stek, u obrnutom redosledu od navođenja Nakon poziva funkcije, radi se *pop* za prihvatanje rezultata ako se vraća preko steka i oslobađa prostor alociran za argumente, po formuli:

$$ADD\ SP,\sum^{broj_arg} veličina_argumenta_i \quad (4)$$

Nakon generisanja celog asemblerskog kôda, dodaje se i sekvenca instrukcija za rekonstrukciju sadržaja BP i povratak u pozivajući modul.

```
MOV SP, BP        (5)
POP BP            (6)
RET               (7)
```

Algoritam za generisanje 8086 asemblerskog kôda potprograma napisanog u C-u je dat u Listingu 3.

Ulaz: izvorni kôd programa, memorijska reprezentacija promenljivih
Izlaz: asemblerski kôd C programa

1. PUSH BP
2. MOV BP, SP
3. SUB SP,$\sum^{broj_p} veličina_promenljive_i$
4. Za svaku naredbu u sintasknom stablu
   if(naredba je poziv funkcije)
      Alocirati lokaciju za rezultat
      Staviti argumente na stek u obrnutom redosledu
      CALL naredba.ime_funkcije
      POP rezultat
      ADD SP,$\sum^{broj_arg} veličina_argumenta_i$
   else
      odrediti memorijsku lokaciju operanda
      generisati asemblerski kôd naredbe
5. MOV SP, BP
6. POP BP
7. RET
8. kraj

Listing. 3. Algoritam za generisanje 8086 asemblerskog kôda C programa.

IV. Evaluacija

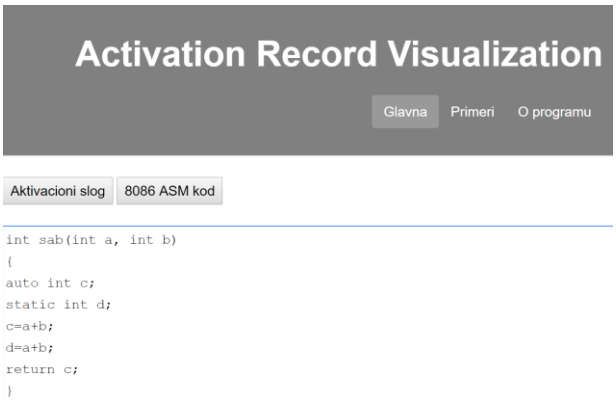
U Tabeli I su dati rezultati evaluacije. Prva kolona prikazuje dužinu izvornog C kôda koji je korišćen kao ulaz, izraženu kao broj linija. Druga kolona je vreme neophodno za generisanje prikaza aktivacionog sloga, dok treća prikazuje vreme za genersianje asemblerskog kôda. Sva vremena su data kao prosek 10 uzastopnih izvršenja programa u sekundama. Konačno, četvrta kolona je ukupno vreme neophodno za generisanje kôda, nastalo kao zbir prethodna dva.

Na slici 4 je prikazan korisnički interfejs aplikacije sa primerom kôda iz prvog eksperimenta (8 linija C kôda). Na slici 5 se vidi prikaz aktivacionog sloga za taj primer.

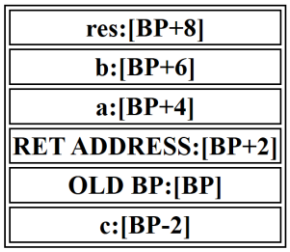
Serverski deo aplikacije izvršavan je na laptopu sledećih karakteristika: Intel i7 7700-HQ quad core 2.80GHz CPU i 16GB DDR4 RAM. Kao što se može videti, vreme neophodno za generisanje kôda se povećava sa brojem linija izvornog C kôda.

TABELA I
VREME IZVRŠENJA U ZAVISNOSTI OD DUŽINE C KÔDA.

Dužina C kôda [broj linija]	Gen. aktivacionog sloga [s]	Gen. asm kôda [s]	Ukupno vreme gen. [s]
8	0.47	0.49	0.96
16	0.51	0.52	1.03
51	0.93	1.04	1.97



Slika 4. Prikaz korisničkog interfejsa sa primerom kôda.



Slika 5. Prikaz generisanog HTML kôda aktivacionog sloga.

V. Zaključak

Glavni cilj prikazanog rešenja jeste njegova primena u okviru kursa Programski prevodioci na četvrtoj godini osnovnih akademskih studija na Elektronskom fakultetu u Nišu, kao pomoć pri razumevanju ispitnih zadataka. Po ostvarenim rezultatima, ova implementacija pokazuje potencijal, s tim da bi za aktivnu online upotrebu od strane većeg broja studenata u realnim uslovima (oko 150 studenata na četvrtoj godini) bio neophodan server viših performansi od laptopa koji je korišćen za evaluaciju u ovom slučaju, s obzirom da je Xtext radni okvir prilično zahtevan, zbog velikog broja Java biblioteka na koje se oslanja. Osim toga, planirano je dodavanje podrške i za druge programske jezike (poput C++, Java, C#) i arhitekture.

LITERATURA

[1] Knjiga predmeta za osnovne akademske studije, modul Računarstvo i informatika: http://old.elfak.ni.ac.rs/downloads/akreditacija-2013/oas/oas-rii.pdf . Poslednji pristup 23/09/2019.
[2] M. Stanković, S. Stojković, Ž. Tošić, "Programski prevodioci", Elektronski fakultet u Nišu, 2018.
[3] A. Aho, M. Lam, R. Sethi, J. Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education, 1986.
[3] J. S. Warford, C. Dimpfl, "The pep/8 memory tracer: Visualizing Activation Records on the Run-Time Stack", Proceedings of the 41st ACM Technical Symposium on Computer Science Education - SIGCSE '10. str. 371-375, 2010. https://doi.org/10.1145/1734263.1734389,
[4] Xtext [Online], https://www.eclipse.org/Xtext/, poslednji pristup: 08/02/2019.
[5] L. Bettini, "Implementing Domain-Specific Languages with Xtext and Xtend", Packt Publishing, 2016.

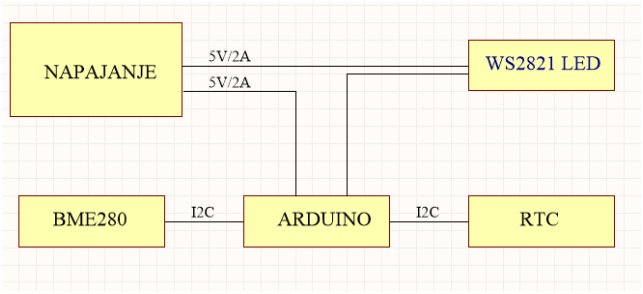
Projektovanje i realizacija digitalnog sata WORD CLOCK

Jelica Janić

Sadržaj - Poslednjih godina dolazi do sve većeg interesovanja kupaca za digitalne časovnike. Razvojem elektronike došlo je i do razvoja ovih uređaja. U ovom radu opisan je jedan takav uređaj, koji je primamljiv kupcima, kako zbog svoje moderne unutrašnjosti, tako i spoljašnjosti.

I. OSNOVNI PRINCIP RADA UREĐAJA

Centralni deo uređaja je mikrokontroler sa Arduino Uno platforme. Ovaj blok se napaja pomoću adaptera koji daje 5 V DC i 2 A. Mikrokontroler komunicira sa RTC modulom putem I^{2}C komunikacije i neprekidno proverava koliko je sati i minuta. Na osnovu očitanog vremena, on šalje informacije LED traci WS2812b na osnovu koje se uključuju određene diode. Na svakih pet minuta se aktivira određena RGB dioda koja je namenjena za određene minute, a na svakih sat vremena se aktivira ona koja je predviđena za dati sat. Dve diode su konstantno aktivne i one služe za osvetljenje reči sat i min. Zatim, senzor BME280 je senzor koji služi za merenje temperature, vlažnosti i pritiska u vazduhu. Mikrokontroler konstantno proverava vrednosti koje senzor daje, te na osnovu algoritma "predviđa" kakve su vremenske prilike. U zavisnosti od toga kakve su vremenske prilike (da li je sunčano, oblačno ili olujno), RGB diode će svetleti određenom bojom. Na slici 1 prikazan je blok dijagram projektovanog uređaja.



Slika 1. Blok dijagram projektovanog uređaja.

II. FUNKCIONALNE CELINE UREĐAJA

Funkcionalne celine predstavljaju pet blokova uređaja. U ovom poglavlju će biti opisana svaka celina ponaosob, kao i načini komunikacija između njih.

Jelica Janić, student Master akademskih studija na Elektronskom fakultetu, Univerzitet u Nišu, smer Elektronika i mikrosistemi, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: jelicajanic96@gmail.com.

Neophodni deo svakog električnog uređaja je njegovo napajanje. Pri projektovanju uređaja opisanog u ovom radu, korišćen je prekidački izvor napajanja. Karakteristike ovog izvora napajanja su sledeće:

- Napon na ulazu: 230 VAC;
- Napon na izlazu: 5 VDC;
- Struja na izlazu: 2 A;
- Snaga: 10 W.

RTC (*Real Time Clock*) je modul koji je namenjen određivanju preciznog vremena. U ovom radu biće korišćen DS3231 modul koji je prikazan na slici 2:



Slika 2. RTC modul [1].

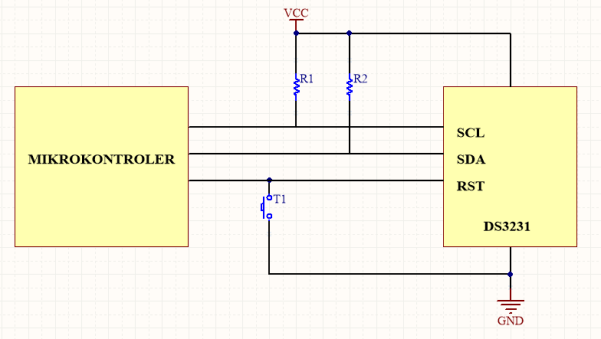
DS3231 pruža informacije o tačnom vremenu (sati, minuti, sekunde), kao i o datumu (dan u mesecu, mesec i godina). Kada se jednom podesi vreme, moguće je pratiti kalendar sve do 2100. godine. Sat radi u 24 i 12- satnom formatu, takođe sadrži i dva programabilna alarma. U tabeli 1 prikazane su električne karakteristike ovog modula.

TABELA 1
ELEKTRIČNE KARAKTERISTIKE RTC MODULA

Parametar	Vrednost
Radni napon	-0.3V – 6V
Radna temperatura	-40°C - 85°C
Napon napajanja	2.3 – 5.5V
Napon baterije	Vbat<Vcc
Struja baterije u aktivnom režimu (Vbat=5.5V)	150µA
Struja baterije (5.5V) u neaktivnom režimu	3.5µA

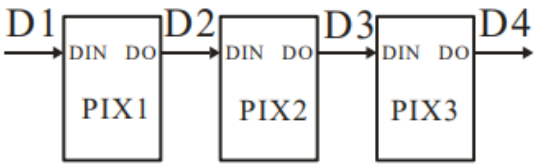
RTC DS3231 ima 16 izvoda, od kojih su tri direktno povezana na mikrokontroler. Izvodi SDA i SCL služe za

komunikaciju sa mikrokontrolerom. Na slici 3 prikazan je način na koji RTC ostvaruje vezu sa mikrokontrolerom:

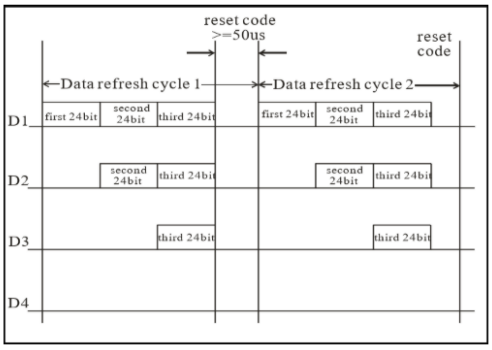


Slika 3. Povezivanje DS3231 sa mikrokontrolerom.

WS2812b je inteligentni LED izvor svetlosti. Ova LED traka, WS2812b se od ostalih razlikuje po načinu na koji je možemo kontrolisati. Ona se sastoji iz niza RGB dioda, koje možemo pojedinačno kontrolisati. WS2812b je naziv integrisanog kola koje upravlja RGB diodama. Ono što se vidi na traci su zapravo kućišta 5050 (5.0x5.0 mm), u kojima se nalaze pomenuto integrisano kolo i RGB diode. Traka komunicira sa mikrokontrolerom tako što on šalje niz nula i jedinica na data liniju. Integrisano kolo WS2812b skladišti prvih 24 bita, odnosno 3 bajta (po bajt za svaku boju) u latch , a ostatak šalje dalje narednim integrisanim kolima, i tako redom do poslednjeg piksela. Kada stigne do kraja, izvrši se reset i sve kreće ponovo. Serijski kaskadni interfejs, prijem podataka i dekodiranje zavise samo od jedne signalne linije. Na slici 4 prikazan je način ostvarivanja serijskog kaskadnog interfejsa, dok je na slici 5 prikazan način slanja informacija LED traci



Slika 4. Ostvarivanje kaskadnog serijskog interfejsa.



Slika 5. Grafički prikaz načina slanja informacije LED traci WS2812b [2].

U tabeli 2 dat je prikaz električnih karakteristika LED trake WS2812b.

TABELA 2
ELEKTRIČNE KARAKTERISTIKE LED TRAKE WS2812B

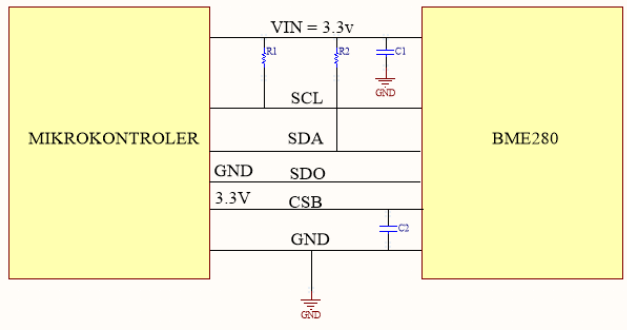
Boja	Crvena, zelena i plava
Radni napon	5V DC
Snaga	16W/m za belu boju
Struja	3.2A/m za belu boju
Vrsta diode	WS2812b SMD LED
Zaštita	IP20
Broj dioda	60 dioda po metru
Dimenzije	10x2mm

Senzor Bosch Sensortec BME280 predstavlja digitalni senzor za merenje vazdušnog pritiska, relativne vlažţnosti i temperature vazduha. Elektronika senzora smeštena je u kućište 2.5 mm x 2.5 mm. Senzor nudi SPI i C interfejse. U radu je primenjen C interfejs. Senzor ima mogućnost rada u tri režima: sleep, normalni i prisiljeni (forced). Za aplikaciju opisanu u ovom radu, programiran je da radi u normalnom režimu rada. Senzor koji je iskorišćen u ovom radu se nalazi u okviru modula prikazanog na slici 6.



Slika 6. Izgled modula sa senzorom BME280, sa prikazom prednje i zadnje strane [3].

Napon napajanja modula je 3.3 V što je prema električnim karakteristikama u okviru granica minimalnog i maksimalnog napona napajnja. Iskorišćen je ceo modul, kao štampana ploča sa projektovanim senzorom na njoj, električno povezivanje sa mikrokontrolerom se vrši pomoću šest žica, kao što je i prikazano na slici 7.



Slika 7. Električna šema povezivanja mikrokontrolera i senzora BME280.

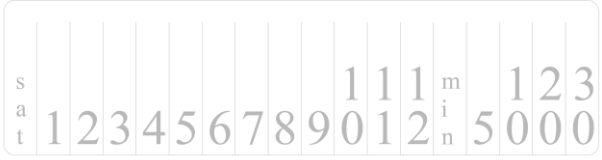
S obzirom da je potrebno predvideti vremenske prilike na osnovu rezultata koje nam da senzor BME280, kako bismo znali kojom će bojom svetleti diode časovnika, neophodno je pri pisnju programskog kôda koristiti neki softverski (digitalni) filtar. Softverski filtri su zapravo algoritmi koji koriste ulazni signal i modifikovani izlazni signal. Funkcija koja se obavlja zavisi od vrste filtra. Propusni opseg je takođe bitan parametar koji utiče na rad filtra. Sa stanovišta upravljačkog softvera, funkcija filtra je definisana u samoj strukturi kôda i operacijama koje se izvode. Širinu propusnog opsega ovih filtra obično definišu konstante, nad kojima se vrše određene matematičke operacije. Zahvaljujući ovoj vrsti filtra moguće je manipulisati širinom propusnog opsega, kao i vremenom odziva, a pored toga oni služe za redukovanje smetnji, tj. šumova. Ukoliko je propusni opseg filtra manji, manje će biti i šumova, ali biće potrebno više vremena da bi se izvršila željena radnja. Međutim ukoliko je propusni opseg veći, biće više šumova, ali će i manje vremena biti potrebno da se izvrši zadata radnja. Kombinovanje više različitih filtra omogućava dizajneru da iskoristi dobre strane svih primenjenih filtra, kao i da zaobiđe njihove mane.

III. Izgled spoljašnosti uređaja

Spoljašnjosti projektovanog uređaja treba se sastojati od izrezbarenog pleksiglasa, koji predstavlja displej projektovanog časovnika, i postolja koje može biti od proizvoljnog materijala. Optimalno rešenje što se tiče postolja je drvo, zbog niske cene i lake obrade.

Klirit ili pleksiglas je prirodni providni materijal, koji predstavlja termo-plastičnu masu. Može biti obojen tako da se postiže gotovo beskonačan spektar boja i nivoa svetlosne propustljivosti. Ovo je jedini plastični materijal koji ima prirodnu UV zaštitu. Ovaj materijal nalazi široku primenu zbog svojih osobina i mogućnosti termo oblikovanja. Prvenstveno se koristi za zastakljivanje i prekrivanje širokih površina poput industrijskih postrojenja, tj. gde god postoji izvestan rizik od loma. Pošto je daleko lakši od stakla, naročito kada je u pitanju sama obrada, svoju primenu nalazi u auto i avio industriji, zatim u industriji reklama, nameštaja, obeležavanja, akvaristici itd. Daleko izraženija postojanost i izdržljivost kada se uporedi sa staklom. Izloženost udarcima koji premašuju njegovu otpornost, ne dovodi u rizik opasnost od povrede - naprotiv, razbija se u velike komade, bez oštrih ivica.

Na slici 8 je prikazan izgled displeja časovnika projektovanog u ovom radu:



Slika 8. Izgled displeja od pleksiglasa projektovanog uređaja.

IV. Programske celine uređaja

Kada se uređaj pokrene, počinje izvršavanje programskog kôda koji je upisan u mikrokontroler ATMega328P. Tok izvršenja programa predstavljen je algoritmom, koji je prikazan na slici 9. Na samom početku programskog kôda potrebno je uključiti biblioteke neophodne za rad, pomoću naredbe *#include*. Neophodne su biblioteke za senzor BME280, kao i za LED traku WS2812b i RTC modul. Takođe se uključuju i biblioteke Wire.h, koja služi za omogućavanje I^2C komunikacije, i biblioteka math.h, koja služi za omogućavanje matematičkih operacija u programskom kôdu. Matematičke operacije su neophodne zbog definisanja softverskog filtra za određivanje vremenskih prilika.

Nakon toga definiše se pin na kome je povezana LED traka WS3231b, kao i broj RGB dioda u kaskadnom nizu dioda. Potom sledi definisanje promenljivih koje su neophodne za dalji rad, kao i definisanje nizova za displej, R, G i B parametre dioda, niz nula, niz jedinica i niz za boje. Kada su svi nizovi i sve promenljive definisane, potrebno je kreirati funkaciju za podešavanje vremena na RTC modulu (void setDS3231time). Nakon toga se kreira funkacija koja služi za očitavanje podataka iz RTC modula (void readDS3231time).

Kada su definisane funkcije za rad sa RTC modulom, potrebno je kreirati softverski filtar za određivanje vremenskih prilika na osnovu očitavanja senzora BME280. Definisanje softverskog filtra za određivanje vremenskih prilika se određuje na osnovu sledeće formule:

$$P_0 = P\left(1 - \frac{0.0065h}{T + 0.0065h + 273.15}\right)^{-5.257} \quad (1)$$

P_0 predstavlja pritisak vazduha na 0 metara nadmorske visine (nivo mora), dok je P pritisak koji izmeri BME280 našeg uređaja. T je izmerena temperatura u datom trenutku, dok je h nadmorska visina grada Niša (moguće je odabrati i nadmorsku visinu koju izmeri senzor, ali je zbog veće preciznosti iskorišćena zvanična nadmorska visina grada Niša).

V. Praktična realizacija

Na slici 10 prikazana je praktična realizacija uređaja. Diode svetle plavom bojom, što znači da je bilo oblačno u trenutku fotografisanja. Na slici 11 prikazan je printscreen ekrana telefona kao dokaz o vremenskim prilikama tokom fotografisanja za sliku 10.

VI. Zaključak

Rad opisuje projektovanje i realizaciju prototipa savremenog digitalnog časovnika, koji pruža informaciju o vremenu, kao i o trenutnim vremenskim prilikama.

Projektovani prototip se može u potpunosti realizovati i proizvoditi za tržište.

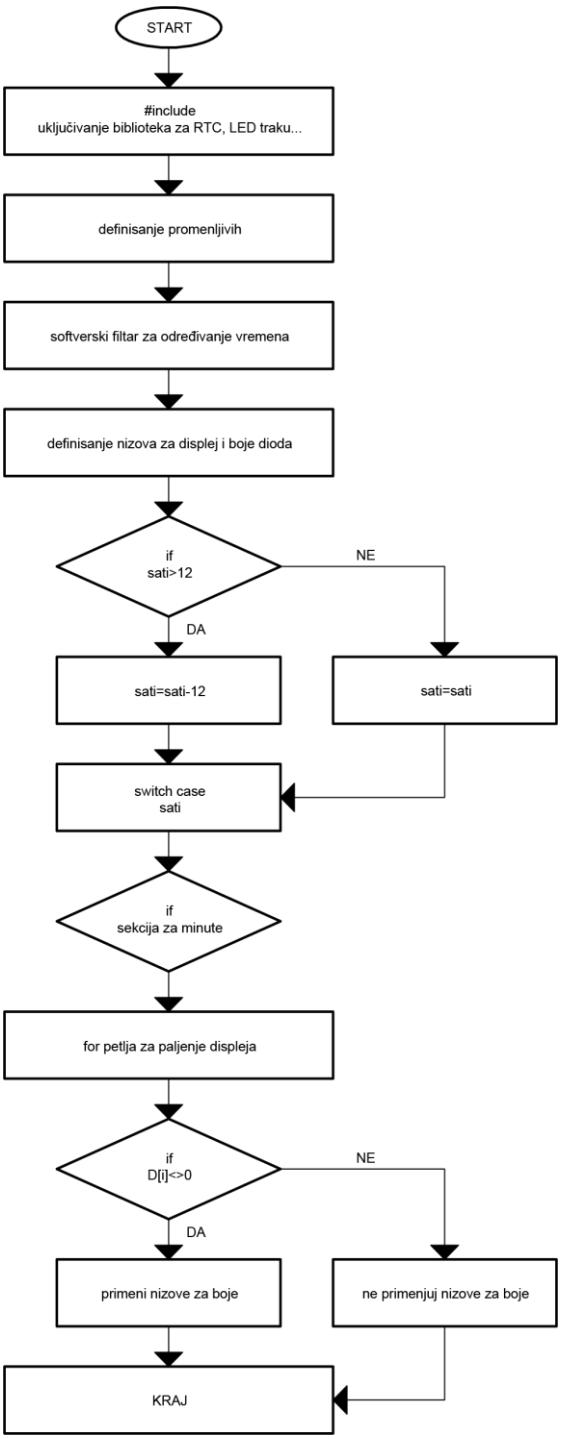
Ovaj rad pokazuje primenu elektronike za proizvodnju uređaja namenjenih širokom spektru korisnika. Uređaj možda nije naučno primenljiv, ali je prikaz proizvoda koji je uz minimalnu dogradnju spreman za širu proizvodnju i slanje na tržište. Takođe je jako primamljiv kupcima zboog svoje interesantne spoljašnjosti i neobičnosti. Ostavljen je prostor za unapređenje projekta. Osnovni princip projektovanja predočen je čitaocu, te je on u mogućnosti da iz ovog rada dobije gotov proizvod.



Slika 10. Prikaz praktične realizacije sa dokazom o vremenu u trenutku fotografisanja.



Slika 11. Dokaz o vremenskim prilikama u trenutku fotografisanja.



Slika 9. Algoritam programskog koda projektovanog uređaja

LITERATURA

[1] Dokumentacija za RTC https://potentiallabs.com/cart/buy-ds3231-rtc-module-hyderabad-online-india
[2] https://e-radionica.com/hr/blog/2018/02/28/ws2812b-rgb-led-traka/
[3] Dokumentacija za senzor BME: https://ae01.alicdn.com/kf/HTB1tIyyEr9YBuNjy0Fgq6AxcX

Xaq/BME280-Digital-Sensor-Temperature-Humidity-Barometric-Pressure-Sensor-Breakout-Module-Board-GY-BME280-I2C-IIC-SPI.jpg_640x640.jpg
[4] Dokumentacija za senzor BME280: https://datasheet.octopart.com/BME280-Bosch-Tools-datasheet-119732019.pdf
[5] https://earthscience.stackexchange.com/questions/16366/weather-forecast-based-on-pressure-temperature-and-humidity-only-for-implement
[6] Dokumentacija za LED traku: https://www.kitronik.co.uk/pdf/WS2812B-LED-datasheet.pdf
[7] https://potentiallabs.com/cart/buy-ds3231-rtc-module-hyderabad-online-india
[8] http://ba.ledlightfeel.org/smd-5050-led-strip-lights/rgb-smd-5050-led-strip-lights/rgbw-rgbcw-white-sk6812-similar-ws2812b-3-3ft.html
[9] https://ae01.alicdn.com/kf/HTB1tIyyEr9YBuNjy0Fgq6AxcXXaq/BME280-Digital-Sensor-Temperature-Humidity-Barometric-Pressure-Sensor-Breakout-Module-Board-GY-BME280-I2C-IIC-SPI.jpg_640x640.jpg.

Razvoj aplikacije za praćenje kretanja simbola na displeju u okviru PSoC razvojnog okruženja

Stefan Vasiljević

Sadržaj - U ovom radu opisuje se realizacija interakcije između LCD modula i ugrađenih komponenti na PSoC razvojnom okruženju sa ciljem prikaza funkcionalnosti ovakvog sistema, a kroz vid zabave korisnika. Predstavljene su karakteristike korišćenog modula i razvojnog okruženja neophodnih za izradu projekta. Rad je realizovan na razvojnom okruženju CY8CKIT-050, proizvođača Cypress Semiconductor. Za konfiguraciju i programiranje PSoC čipa korišćeno je softversko razvojno okruženje PSoC Creator 3.0. Opisan je princip rada i način realizacije sistema.

I. UVOD

U savremenoj elektronici postoji sve veća potreba za sistemima visokih performansi. Realizacija celokupnog sistema na jednom čipu i velika fleksibilnost pri razvoju se nameću kao imperativ. PSoC je programabilni ugrađeni (embedded) sistem na jedinstvenom čipu koji obuhvata podesive analogne i digitalne periferije, memoriju i mikrokontroler. Omogućava izradu analogno-digitalnog sistema sa velikom fleksibilnošću rutiranja signala, osobinom rekonfigurabilnosti i imunošću na šum [1]. Sama konfiguracija se vrši na vrlo jednostavan i intuitivan način pomoću namenskog softverskog paketa koji se bazira na grafičkom korisničkom okruženju.

Ovakvim pristupom ostvaruje se velika produktivnost u radu, s obzirom da projektant umesto da ponaosob razvija svaku potrebnu komponentu, vrši njihovu integraciju na jednom čipu ostvarujući kompleksne funkcije u relativno kratkom vremenu.

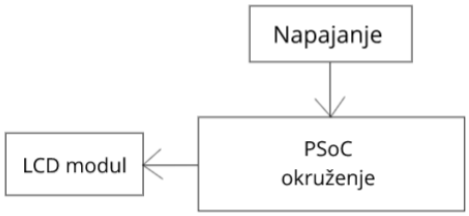
Osnovni elementi igre su figura korisnika i podloga sa algoritmom za nasumično kretanje. Igrač svoj položaj kontroliše okretanjem potenciometra koji se nalazi na razvojnom okruženju. Cilj je da što duže ostane na podlozi koja se konstantno kreće po ekranu. U zavisnosti od odabranog režima igre razlikuje se i dozvoljeno vreme čekanja na potez korisnika. Na kraju igre prikazuje se korisnikov rezultat.

II. FUNKCIONALNE CELINE UREĐAJA

Funkcionalne celine predstavljaju blokove uređaja, a u ovom delu je dat detaljan opis svih blokova.

Stefan Vasiljević je student na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, Niš, Srbija. E-mail: stefanvasiljevic1995@hotmail.com

Blok šema realizovanog projekta se može videti na slici 1.

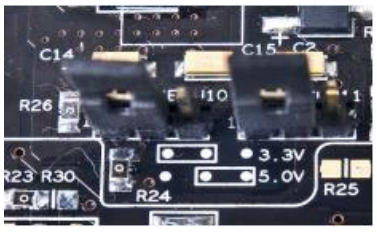


Slika 1. Blok šema projekta.

A. Napajanje

Sastavni deo svakog električnog uređaja, bez obzira na njegovu funkciju, jeste izvor napajanja. Zadatak izvora napajanja je da naizmenični napon iz mreže pretvori u jednosmerni, svodeći pri tome njegovu amplitudu na željeni nivo [2], kao i da takav jednosmerni napon učini dovoljno stabilnim u smislu imunosti na varijacije mrežnog napona i šumove.

PSoC razvojna ploča može da se napaja na nekoliko načina: korišćenjem USB porta računara ili nekog spoljašnjeg izvora. Napon koji se dovodi na ploču može biti u opsegu od 3.3 V do 5 V. U zavisnosti od odabrane vrednosti napona mora se izvršiti pravilno postavljanje kratkospojnika na ploči (Slika 2.).



Slika 2. Postavljanje kratkospojnika na ploči.

B. CY8CKIT-050 razvojno okruženje

Razvojno okruženje CY8CKIT-050 je programabilna hardverska platforma koja služi za dizajniranje sistema za rad sa mešovitim signalima. Obezbeđuje pristupačan i fleksibilan način korišćenja novih koncepata i izradu prototipa koristeći različite kombinacije performansi, potrošnje i karakteristika.

Komunikacija između podsistema se obavlja uz pomoć sistemske magistrale. Analogni podsistem PSoC-a se

povezuje sa okolinom preko odgovarajućeg interfejsa. Informacije sa većine senzora su uglavnom analogne prirode i neophodno je putem A/D konverzije pretvoriti ih u digitalni signal koji se dalje obrađuje Digitalni podsistem prenosi digitalne signale ka svojim periferijama koristeći DSI interfejs. Takođe, obezbeđuje niz funkcionalnih fleksibilnosti uz pomoć malih, brzih univerzalnih digitalnih blokova (UDB) izuzetno niske potrošnje. Procesorski podsistem je izgrađen oko 32-bitnog Arm Cortex-M3 procesora koji može da radi na frekvencijama do 80 MHz. U sam procesor je integrisan i kontroler signala prekida - NVIC. Pored tog kontrolera procesor sadrži i DMA kontroler, fleš i RAM memoriju, kao i module za otklanjanje grešaka. NVIC omogućava bržu obradu signala prekida. Samo jezgro procesora poseduje dva nivoa rada i dva moda rada, što dovodi do toga da neke instrukcije, specijalni registri, konfiguracioni registri i komponente za otklanjanje grešaka mogu biti blokirane u zavisnosti od trenutno aktivnog nivoa (slika 3.).

Uslov	Privilegovani	Korisnički
Izvršavanje izuzetka	Hendler mod	Ne koristi se
Izvršavanje glavnog programa	Nitni mod	Nitni mod

Slika 3. Nivoi rada procesora.

U okviru samog sistema postoje 3 tipa memorije. To su: statički RAM, fleš memorija i EEPROM. Fleš memorija je memorija u koju se smeštaju firmver i korisnički konfiguracioni podaci. Njen kapacitet je do 256 kB memorijskog prostora. PSoC 5LP obezbeđuje i eksterni memorijski interfejs. On omogućava povezivanje sa spoljašnjim memorijskim uređajima. Ulazno/izlazni pinovi PSoC-a su veoma fleksibilni. Svaki pin ima mogućnost analogne i digitalne konfiguracije. Dele se u dve osnovne grupe. To su pinovi opšte i pinovi specijalne namene. Oba tipa obezbeđuju sličnu digitalnu funkcionalnost, a glavna razlika je u njihovim analognim mogućnostima [3].

C. LCD modul

LCD modul korišćen u projektu je standardni 2x16 alfanumerički displej. Funkcioniše pri naponima od 3.3 V do 5.3 V. Može da radi u dva moda: 4-bitnom i 8-bitnom. Pre samog korišćenja displeja mora se obezbediti njegovo stabilno napajanje. LCD modul zahteva serijsko slanje podataka, a PSoC Creator podržava samo 4-bitni interfejs.

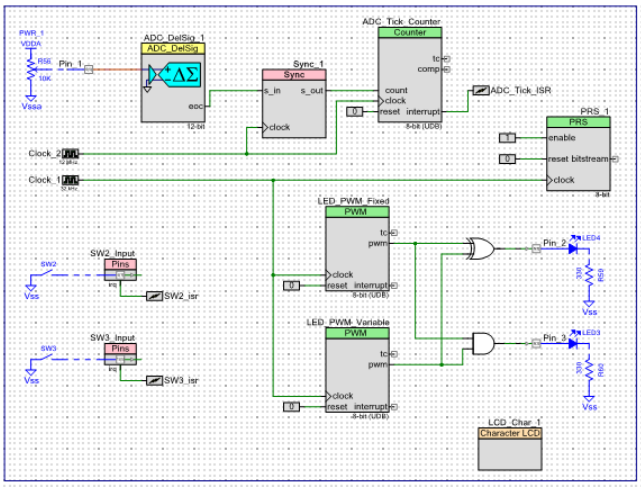
Ova komponenta je sastavni deo ugrađenih biblioteka PSoC Creatora. Kao takva sadrži sve funkcije koje omogućavaju njeno korišćenje na pravi način u pisanju softvera.

III. SOFTVERSKO I HARDVERSKO OKRUŽENJE

Za pisanje programskog koda koristi se softverski alat PSoC Creator verzije 3.0. PSoC Creator je integrisano dizajnersko okruženje koje omogućava projektovanje hardverskog povezivanja komponenata kao i pisanje softvera projekta [4]. Pravilo je da se prvo razvija hardverski deo dizajna, a nakon kompajliranja i generisanja potrebnih API funkcija, prelazi se na razvijanje softvera. Hardverski deo se programira šematskim pristupom pri čemu se može koristiti preko 150 bibliotečkih komponenata.

A. Hardverski razvoj

Kreiranje željenog dizajna se vrši prevlačenjem i postavljanjem potrebnih komponenata koje se koriste za izgradnju hardverskog dela projekta u glavni radni prostor, tj. šemu. U šemu PSoC projekta moguće je ubaciti i komponente koje će eksterno biti dodate na proto ploči (slika 4.).



Slika 4. Šema projekta.

Sa prethodne slike se mogu uočiti sve komponente koje su korišćene u ovom projektu. To su:

- 1 ADC delta-sigma konvertor
- 1 LCD modul
- 2 prekidača (uključujući i prekidne rutine)
- 2 LED diode
- 1 brojač
- 2 PWM modula
- 1 PRS modul
- 1 potenciometar
- 1 kolo za sinhronizaciju.

Svi signali iz ljudske okoline su analogne prirode, dok računari, mikrokontroleri i drugi elektronski uređaji rade sa digitalnim signalima. Zbog toga je za obradu analognih podataka od strane nekog elektronskog sklopa, neophodno naći njihovu digitalnu vrednost. U ovom projektu je korišćen jedan delta-sigma konvertor. Njih karakteriše visoka rezolucija i tačnost konverzije. Poseduje mogućnost biranja od 8 do 20 bitova rezolucije. Podržava jednostrani i diferencijalni režim rada sa podesivim opsegom ulaznog napona. ADC delta-sigma konvertor čita vrednost napona koja je određena naponom napajanja i pozicijom klizača potenciometra i pretvara je u digitalnu vrednost sa 12-to

bitnom rezolucijom pri 1000 uzoraka po sekundi. Komponenta za sinhronizaciju se koristi kako bi se otklonile sve razlike između analognog i digitalnog domena [5]. Izlazni signal konvertora se vodi na ulazni pin komponente za sinhronizaciju. Na drugi ulazni pin se dovodi taktni signal frekvencije 12 MHz i to onaj signal po kome se vrši sinhronizacija.

Izlazni signal iz komponente za sinhronizaciju se deli sa 10 uz pomoć brojača kako bi se u firmveru generisale prekidne procedure na svakih 10 ms. Kako bi ta funkcionalnost bila moguća u konfiguracionom dijalogu se za vrstu implementacije moraju odabrati UDB blokovi.

Komponenta PWM služi za generisanje povorke impulsa iste ili različite dužine trajanja. Perioda signala koji se generiše zavisi od izabrane frekvencije taktnog signala. Komponenta radi na 32 kHz i služi da pobuđuje dve led diode. Za vrednost Period promenljive uzimamo 255, dok se Compare parametru zadaje vrednost od 127.

Komponenta PRS (Pseudo Random Sequence), koja je konfigurisana da radi u kontinualom modu, obezbeđuje nasumične osmobitne vrednosti koje se kasnije koriste u softveru kako bi se odredio naredni položaj određenih struktura igre. Pritiskom na tastere generišu se odgovarajući prekidni signali.

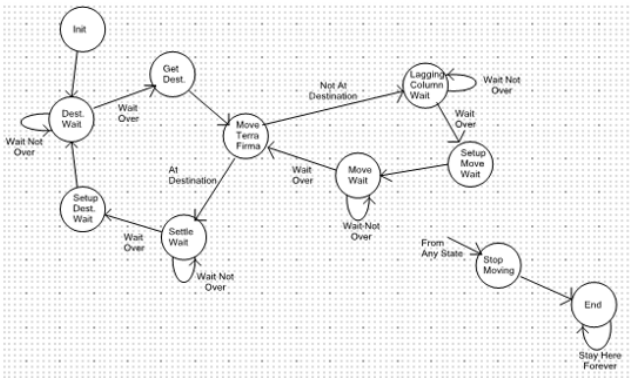
B. Softverski razvoj

Nakon uspešnog konfigurisanja komponenata vrši se pisanje koda. Kako bi se kod što bolje struktuirao, podeljen je na više celina. Svaka celina predstavlja jednu funkcionalnost, a sve one se spajaju u main.c fajlu kao glavnom fajlu firmvera. Za uspešnu relizaciju projekta potrebno je napisati nekoliko korisničkih fajlova. Sama logika je takva da se sve odvija preko konačnih automata, pa se tako i u kodu definiše nekoliko takvih tipova. To su GamePlayMachine, GameOverMachine, GameStartMachin e kao i dodatan fajl koji određuje poziciju igrača na LCD displeju - PlayerInput.

GamePlayMachine simulira sam tok igre. Sama igra je takva da korisnik ima mogućnost da bira kojom brzinom će igrati. U sporijem režimu minimalno i maksimalno vreme čekanja na potez se nalaze u opsegu između 70 i 300 ms. Dakle ukoliko korisnik svoj potez odigra u trenutku koji nije deo ovog intervala potez će se smatrati nevalidnim. Ista pravila važe i za brzi režim, pri čemu je taj vremenski opseg manji i granične vrednosti su 50 i 250 ms.

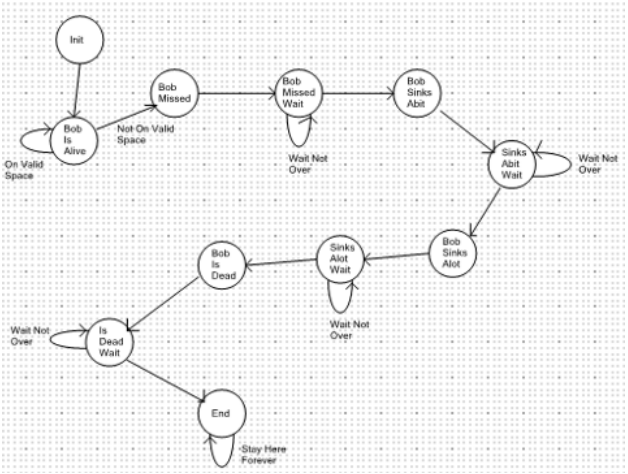
```
// constants below filter results from pseudo-random generator
#define SLOW_MOVE_DELAY_MIN_VALUE       70      // ms
#define SLOW_SETTLE_TIME_MIN_VALUE      300     // ms
#define FAST_MOVE_DELAY_MIN_VALUE       50      // ms
#define FAST_SETTLE_TIME_MIN_VALUE      250     // ms
```

Potrebno definisati promenljive kojima se opisuju sva moguća stanja u kojima može da se nalazi podloga koju korisnik mora da prati svo vreme. Iz inicijalnog stanja se prelazi u stanje čekanja destinacije – naredne pozicije na koju će se podloga pomeriti. Kada se čekanje završi prelazi se u naredno stanje. To je stanje u kome se podloga kreće na zadatu lokaciju na LCD displeju. Kretanje se izvršava prelaskom kroz tri stanja pri čemu se mora voditi računa da se prikaže efekat pomeranja podloge. Kada se podloga nađe na tačnoj lokaciji sistem ponovo odlazi u stanje čekanja. Iz svakog stanja je moguće preći u stanje prestanka kretanja (slika 5.)



Slika 5. Dijagram stanja podloge koju korisnik prati.

Sličan princip je iskorišćen za definisanje ponašanja korisnika (slika 6.).



Slika 6. Dijagram stanja korisnika.

Iz inicijalnog stanja se odmah vrši provera da li je korisnik i dalje živ, to jest da li se nalazi na podlozi. Pravila su takva da kada se podloga ne pomera susedni položaji se smatraju nevalidnim pozicijama. Međutim kada je podloga u pokretu dva položaja iza podloge se takođe smatraju validnim. Kada podloga stigne na unapred zadatu poziciju susedni položaji su validne pozicije još određeno vreme. To vreme je definisano u kodu. Zbog toga se prilikom provere da li je korisnik i dalje u igri mora sačekati da istekne vreme dozvoljeno za boravak na pozicijama različitim od trenutne pozicije podloge.

Inicijalizacija i sve provere trenutnih položaja korisnika i podloge se obavljaju pozivanjem odgovarajućih

funkcija. Najpre se očitava trenutna vrednost napona podešenog preko potenciometra. U zavisnosti od te vrednosti postavlja se podloga. Ukoliko je vrednost napona minimalna, podloga će biti smeštena u krajnji levi ugao displeja. Ukoliko je vrednost maksimalna podloga će biti uz desnu ivicu. Ukoliko ima vrednost između graničnih, izvršiće se skaliranje i podloga će se postaviti negde na displeju.

U okviru funkcije koja određuje novu poziciju podloge najpre se korisniku dodeljuju poeni za svo trenutno kretanje. Zatim se određuje nova pozicija čitanjem vrednosti sa komponente koja generiše nasumične vrednosti.

```
//povecavanje rezultata korisnika
playerScore += tfMoveData.moveValue;

//odredjivanje nove nasumicne pozicije
do
{

    randomNumber = PRS_1_Read();
    destCol = randomNumber/LCD_DISPLAY_COLS;
} while(destCol == terraFirmaCurrentCol);
tfMoveData.destCol = destCol;
```

Zatim je potrebno odrediti nove poene koje će korisnik dobiti ako na vreme isprati kretanje podloge.

U fajlu PlayerInput nalazi se logika koja implementira željeni rotacioni ugao potenciometra, odnosno definiše osetljivost promene napona sa okretanjem osovine potenciometra. Naime, kako bi korisnik imao što bolji i realniji doživljaj same igre potrebno je redukovati veličinu rotacionog ugla potenciometra.

Nakon čitanja vrednosti sa ADC potrebno je izvršiti traženo skaliranje i dobijenu vrednost proslediti u dalji tok programa. Na početku igre korisniku je omogućeno da bira režim brzine kojom će igrati pritiskom na odgovarajući prekidač.

```
CY_ISR(SW2_isr_Interrupt)
{

    // slow play speed selected
    playSpeed = PLAY_SPEED_SLOW;

    // we just want one interrupt - disable interrupts for this input
    SW2_isr_Disable();

}
```

Ulaskom u ovu servisnu rutinu postavlja se željena brzina igre, a zatim se vrši onesposobljavanje generisanja novih prekida kako se ne bi dozvolilo korisniku da u toku igre modifikuje početne konfiguracije i na taj način dovede do greške u kodu. Fajl GameStartMachine sadrži funkcije koje su zadužene da prikažu odgovarajuće poruke i uvodne animacije na displeju prilikom pokretanja igre i biranja režima.

Fajl GameOverMachine sadrži funkcije koje ispisuju odgovarajuće poruke prilikom završetka igre. U glavnom fajlu se vrši inicijalizacija funkcija i implementira početno stanje igre. To početno stanje je u stvari niz prethodno opisanih modela i u zavisnosti od razvoja događaja poziva se odgovarajuće stanje.

IV. Realizacija

Za realizaciju igre potrebno je poznavanje programskog jezika C i poznavanje osnova hardvera PSoC razvojnog okruženja. Pri samom pokretanju aplikacije aktivira se već opisano početno stanje aplikacije i na displeju se prikazuje početna poruka. Pošto je poruka predugačka za LCD modul koristi se tehnika skrolovanja teksta (slika 7.).



Slika 7. Inicijalna poruka.

Korisnik nakon toga bira željenu brzinu igre. Nakon odabira na ekranu se izvršava mala animacija koja ujedno označava i početak igre. Algoritam određuje naredni položaj podloge, pomera je i u trenutku kada dođe do konačne destinacije kreće odbrojavanje. Korisnik mora da u zadatom vremenu dođe do podloge inače je igra završena (slika 8.).



Slika 8. Tok igre.

Po završetku igre na ekranu se ispisuje konačan broj poena.Oni ostaju ispisani sve dok korisnik ne pritisne reset dugme (slika 9.).



Slika 9. Kraj igre.

ZAKLJUČAK

Opisana aplikacija ima za cilj da prikaže jednu od mogućih funkcionalnosti PSoC mikrokontrolera u sprezi sa dodatnim hardverskim komponentama. Opisani uređaj je realizovan na nivou prototipa. Prava praktična realizacija uključuje projektovanje i realizaciju uređaja na štampanoj ploči.

LITERATURA

[1] Web–site: https://www.etran.rs/common/pages/proceedings/ETRAN2017/ EL/IcETRAN2017_paper_EL1_1.pdf
[2] Web-site: http://mikro.elfak.ni.ac.rs/wpcontent/uploads/2017/10/Analogna/Analogna_lin_pred.pdf
[3] Web-site: http://www.farnell.com/datasheets/2795447.pdf
[4] Web-site: https://www.cypress.com/products/psoc-creator-integrated-design-environment-ide
[5] Web-site: https://www.cypress.com/documentation/component-datasheets/sync

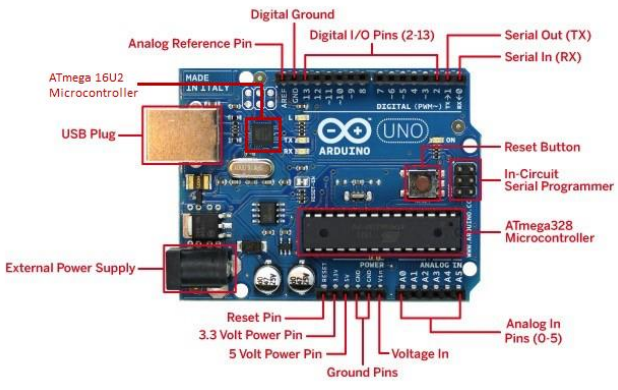
Sistem za praćenje temperature preko mobilne mreže baziran na Arduino platformi

Milan Ilić

Sadržaj– U ovom radu je Arduino Uno pločica koja sadrži mikrokontroler povezana preko serijske veze sa SIM900 modulom koji omogućava razmenu podataka sa mobilnom mrežom, koristeći GSM i GPRS standarde. Vršeno je merenje temperature pomoću temperaturnog senzora LM335 i podaci o izmerenoj temperaturi su poslati u vidu SMS poruke na mobilni telefon ili su poslati preko interneta na „*cloud*“. Slanje temperature preko SMS poruke se vrši u zadanim vremenskim intervalima kao i na zahtev koji se šalje na Arduino preko SMS poruke od strane korisnika.

I. UVOD

Arduino je *open-source* platforma koja se u savremenim komunikacijama koristi za povezivanje raznih uređaja i ostvarivanje komunikacije između njih prema određenim standardima i protokolima, [1]. Arduino prikazan na Slici 1 [2], se sastoji od programabilnog mikrokontrolera koji se programira pomoću *software* IDE (*Integrated Development Environment*) koji se nalazi na kompjuteru i koji se koristi za pisanje kodova koji se *upload*-uju na mikrokontroler. Arduino ne koristi dodatni *hardware*, pod nazivom external programmer kako bi implementirao novi kod, već koristi *bootloader*. *Bootloader* je mali deo *software*-a koji se nalazi na mikrokontroleru i koji čeka da kod bude prebačen na mikrokontroler preko USB porta putem kabla sa kojim je spojen na računar i nakon toga izvršava kod.

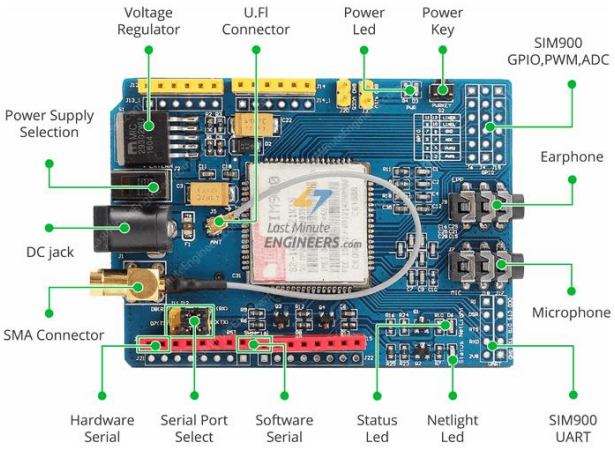


Slika 1. Arduino Uno.

Atmega 328P mikrokontroler je glavna komponenta Arduina a sve ostale elektronske komponente i sklopovi koji se nalazi na ploči omogućavaju eksperimentisanje i rad sa ovim čipom. Programi koji se pišu za Arduino nazivaju se *sketch*-evima i čuvaju se na internoj memoriji mikrokontrolera koji i izvršava programe. Dodatni sklopovi i uređaji se povezuju na mikrokonrtoler koji se nalazi na Arduinu, dok Arduino okruženje samo olakšava povezivanje sa čipom. Mikrokontroler ima iste pinove kao i Arduino.

Atmega 16U2 je zadužen za konverziju signala sa USB-a na serijski signal koji se nakon konverzije šalje do Atmega 328P čipa i na ovaj način se omogućava komunikacija između Arduino ploče i *host* računara.

GSM SIM900 modul, [3], [4], podržava GSM i GPRS standarde. GSM je skraćenica za *Global System for Mobile Communication* i svetski je standard za mobilnu komunikaciju, dok je GPRS skraćenica za *General Packet Radio Service*, i to je standard pomoću koga se omogućava paketska komutacija u mobilnoj telefoniji. Modul je prikazan na Slici 2 kao i portovi i ostale komponente koje se nalaze na modulu.



Slika 2. SIM900 modul.

SIM900 modul omogućava veliki broj funkcija kao što su:

- Podržava četiri propusna opsega: GSM850, EGSM900, DCS1800 i PCS1900,
- Povezuje se na globalnu mreže preko 2G SIM kartice,
- Obezbeđuje mogućnost vršenja poziva ako se povežu mikrofon i slušalice na portove koji to omogućavaju i prikazani su na Slici 2,

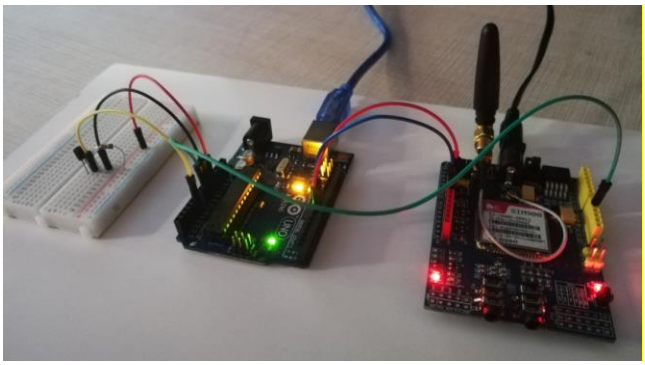
Milan Ilić je student Master studija na studijskom programu Telekomunikacije, Elektronskog fakulteta, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ilicmilannis@gmail.com

- Šalje i prima SMS poruke,
- Šalje i prima GPRS podatke (TCP/IP, HTTP, itd),
- Podržava FM radio,
- Ima predajnu snagu od: Klasa4 (2W) za GSM850 i klasa1 (1W) za DCS1800,
- Omogućava serijsku komunikaciju preko AT komandi,
- Ima UFL i SMA konektore za antenu,
- Podržava SIM karticu u punoj veličini.

Na ploči postoje 3 LED diode koje pokazuju da li je ploča povezana na mrežu i da li ima odgovarajuće napajanje. Posmatrajući diode može se zaključiti šta se dešava sa pločom. SIM900 GSM/GPRS modul korsiti UART protokol kako bi komunicirao sa Arduinom. Čip podržava brzine prenosa od 1200 bps do 115200 bps. SIM900 modul sadrži i antenu kako bi bila omogućena bilo koja vrsta saobraćaja. Antena se može povezati na modul preko dva konektora jedan je UFL, dok je drugi SMA, pri čemu je SMA konektor spojen sa UFL konektorom preko kabla.

II. ARDUINO I SIM900 MODUL

U cilju merenja temperature i prikaza vrednosti potrebno je prvo povezati Arduino sa SIM900 modulom, [5]. Pin D7(Tx) i D8(Rx) na SIM900 modulu se povezuju na digitalne pinove 7 i 8 na Arduinu. U ovom radu koristi se *Software Serial* komunikacija i povezivanje Arduina i SIM900 modula kao što je prikazano na Slici 3. Crvena žica na Slici 3 prima signal sa SIM900 modula, plava šalje signal na modul a zelena je GND. Da bi SIM900 mogao uspešno da se poveže na mrežu koristi se eksterno napajanje od 5 V i 3 A.

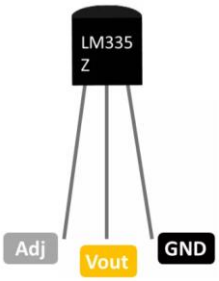


Slika 3. Povezivanje senzora, Arduina i GSM modula.

SIM900 modul sadrži mesto za SIM karticu koja omogućava registraciju modula na mobilnu mrežu.

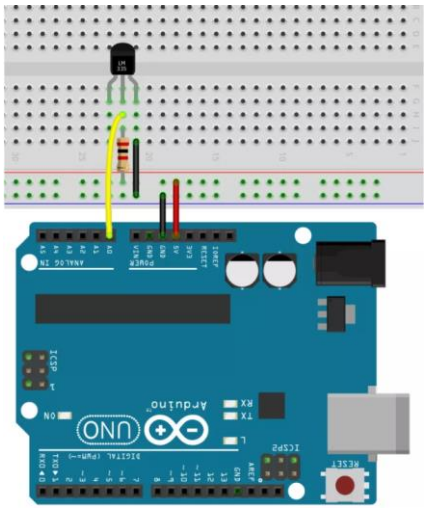
A. Merenje temperature

Informacija o temperaturi koja treba da se pošalje se najpre izmeri pomoću temperaturnog senzora. Za očitavanje temperature koristi se analogni senzor LM335 prikazan na Slici 4, [6], koji je zasnovan na Zener diodi, te je izlazni napon na diodi - senzoru srazmeran sa temperaturom (10 mV/K) [2].



Slika 4. LM335 temperaturni senzor.

Senzor temperature je priključen na prototipsku ploču i sa Arduinom je povezan preko kratkospojnika. Pin Vout je spojen sa napajanjem od 5 V preko otpornika od 1 kΩ i još je povezan na analogni ulaz Arduina, u ovom slučaju je to A0, dok je GND senzora spojen na GND Arduina, kao što je prikazano na Slici 5. Mogu se, takođe, umesto priključaka na Arduinu koristi priključci na SIM900 modulu.



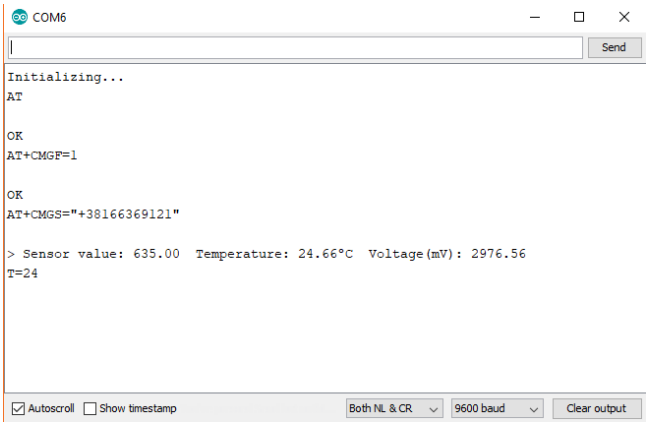
Slika 5. Povezivanje temperaturnomg senzora i Arduina.

Izlazni napon sa senzora se dobija u mV tako što se vrednost pročitana sa senzora pomnoži sa najvećim nivoom ulaznog napona, koji se meri između napajanja od 5 V i GND (u ovom projektu je izmerena vrednost od 4.8 V). Ova dobijena vrednost se podeli sa brojem kvantizacionih nivoa, što iznosi 1024, jer ADC (*Analog to Digital Convertor*) na Arduinu ima 10-bitnu rezoluciju, a dobijena vrednost je dodeljena promenljivoj *VoltageOut*. Potrebno je izmeriti ovaj napon iz razloga što je kao izvor napajanja iskorišćen USB priključak preko koga je Arduino povezan na PC, u drugom slučaju se uzima vrednost od 5 V. Sa senzora LM335 se dobija vrednost temperature u

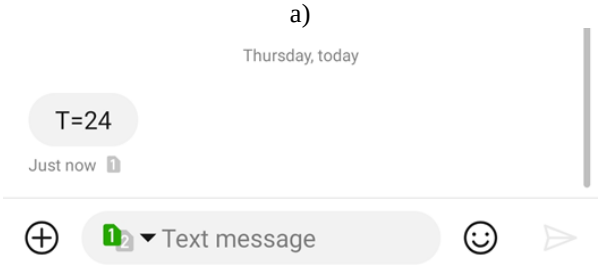
Kelvinima tako da se *VoltageOut* deli sa 10 mV, gde 10 mV odgovara 1 K. Da bi se dobila temperatura u °C od vrednosti temperature u kelvinima se oduzima 273.

B. Slanje temperature preko SMS poruke

Kako bi se informacije o temperaturi poslale na mobilni telefon u vidu SMS poruke treba povezati SIM900 modul sa Arduinom kao što je predhodno objašnjeno i prikazano na Slici 3. Merenje temperature je napisano u vidu funkcije koja se poziva da bi se dobile vrednosti temeprature sa senzora, [5]. Dobijena vrednost je tipa ***float***, pa se ta vrednost pretvara u ***integer*** kako bi se funkcijom **itoa** pretvorila u ***char*** (tipa karakter) jer je potrebno da bude tipa karakter kako bi se upisala u drugi ***char*** pod nazivom *Msg*. Promenljiva *Msg* posle ubacivanja dobijene vrednosti za temperaturu treba da bude oblika“T=20“, gde vrednost 20 predstavlja temperaturu koja se meri. Ovakav niz karaktera se šalje preko SIM900 modula na mobilni telefon.



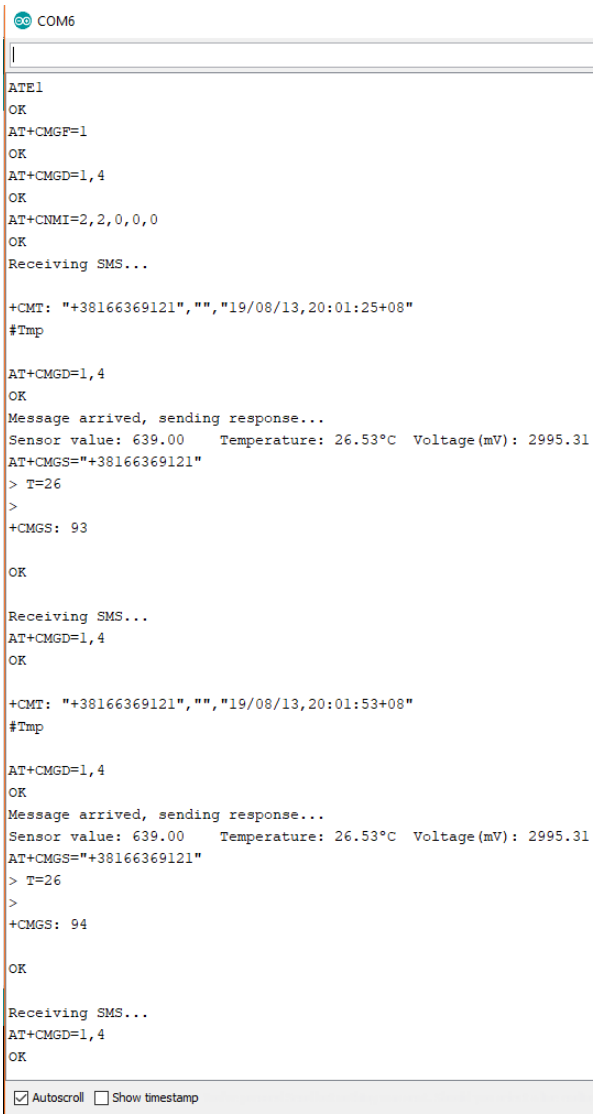
a)



b)

Slika 6. a) Serijski monitor poslate temperature; b) primljena poruka na mobilnom telefonu.

Slanje informacije preko SIM900 modula se obavlja koristeći AT komande kojima se komunicira sa SIM 900 modulom. Posle AT komande, **AT+CMGS=broj mobilnog telefona**, prikazane na Slici 6, treba da se pošalje niz karaktera tj. promenljiva *Msg*. Ovako napisan kod šalje informacije o temperaturi samo jedanput a ako je potrebno da se temperatura šalje više puta treba da se restartuje Arduino, ili je potrebno da se **ReadTemp()** funkcija prebaci u loop (petlju) deo sketch-a, kao i odgovarajuće AT komande, i da se postavi **delay** kako bi se obezbedio vremenski razmak između uzastopnih merenja i slanja. Na Slici 6 je dat prikaz poslate informacije o temperaturi na serijskom monitoru i na ekranu mobilnog telefona.

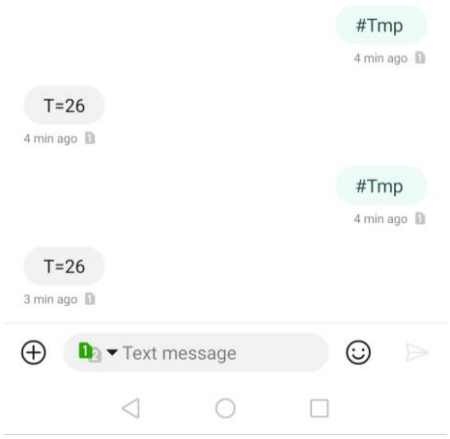


Slika 7. Prikaz serijskog monitora o izmerenoj temperaturi na zahtev.

Pored periodičnog slanja temperature informacija o temperaturi se može slati preko SMS poruke u onom trenutku kada se ta informacija bude zatražila. U odnosu na prethodni slučaj razlikuje se funckija **CheckResponse()** koja proverava da li je primljena SMS poruka, njen sadržaj kopira u ***char*** niz *incomingbyte* i upoređuje sadržaj niza sa predefinisanim tekstom- u ovom slučaju #Tmp. Ako se dva niza poklapaju, parametar SMSArrived se postavlja na *true* što omogućava slanje poruke sa informacijom o temperaturi preko funckije **Send_SMS().** Nakon slanja,

program čeka na sledeću SMS poruku sa sadržajem #Tmp kako bi poslao informacije.

Prikaz o izmerenoj temperaturi na zahtev na serijskom monitoru je dat na Slici 7, dok se na ekranu mobilnog telefona nalazi na Slici 8.



Slika 8. Odgovor na zahtev za stanje temperature prikazan na mobilnom telefonu.

C. Slanje temperature na Cloud

U ovom delu rada, koristeći GPRS vezu i HTTP protokol, podaci o izmerenoj temperaturi se šalju na web stranicu pod nazivom *ThingSpeak*. Kako bi se podaci poslali na „*cloud*“ prvo se mora omogućiti GPRS usluga koristeći AT komande kao i povezivanje na odgovarajuću pristupnu tačku, APN (ove AT komande nisu prikazane). AT komande korišćene za slanje podataka preko GPRS veze su:

AT+HTTPTERM- Prekida HTTP zahtev, u slučaju da prethodni nije zatvoren

AT+HTTPINIT- Omogućava HTTP režim

AT+HTTPPARA="CID",1- Postavlja parametre za HTTP vezu

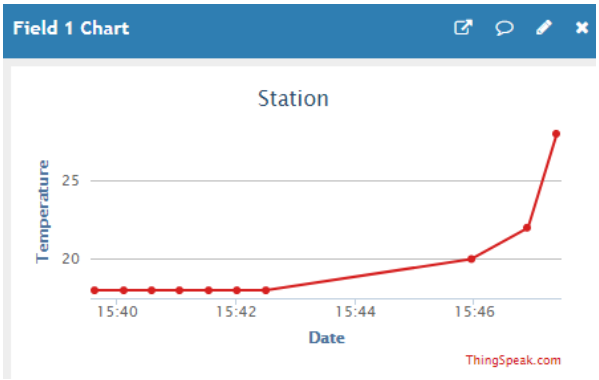
AT+HTTPPARA="URL","api.thingspeak.com/update "- Podešava na koju stranicu se *upload*-uju podaci

AT+HTTPDATA=35,10000- Podešava veličinu podataka koji se *upload*-uju, gde prva cifra označava veličinu niza karaktera koji se šalje a druga vreme koje je dodeljeno čitanju podataka

AT+HTTPACTION=1- Pokreće sesiju za *upload*-ovanje podataka, za čitanje sadržaja se korsiti 0 umesto 1.

Da bi se podaci *upload*-ovali potrebno je napraviti kanal na web stranici *ThingSpeak.com*. Nakon toga, treba uzeti *Write API key* koji omogućava upis podataka, koji je dostupan na sajtu i koji je drugačiji za svakog korisnika. API ključ se nalazi u sledećem nizu karaktera: "api_key=Q0O83E72DR3C045N&field1=". Na ovaj niz karaktera se nadovezuje vrednost koja je izmerena na temperaturnom senzoru, tako što se izmerena vrednost pretvara u niz karaktera i nadovezuje na niz kako bi se informacija poslala na server. Ovaj niz karaktera se šalje na SIM900 modul nakon AT komandi: **AT+HTTPPARA="URL","api.thingspeak.com/update** i **AT+HTTPDATA=35,10000.**

Rezultati i izgled grafika na *ThingSpeak* stranici su prikazani na Slici 9. Temperatura se šalje svakih 15 s usled ograničenja web stranice koja se odnosi na dozvoljenu brzinu slanja podataka.



Slika 9. Izgled *ThingSpeak* stranice.

III. ZAKLJUČAK

U ovom radu je analiziran rad Arduino mikrokontrolera i SIM900 GSM modula koji su povezani serijskom konekcijom i omogućavaju dobru i jednostavnu realizaciju sistema za prenos podataka o temperaturi koja se meri senzorom. Pored toga, u radu je pokazano kako se informacija o temperaturi može poslati periodično kao i na zahtev, koristeći SMS poruku. Razmena informacija je vršena pomoću GSM konekcije koristeći SMS, kao i preko interneta kada se uspostavi GPRS konekcija, pri čemu su rezultati slati na ’*cloud*’. Prikaz rezultata je realizovan na monitoru računara preko serijskog monitora i prikazane su SMS poruke sa telefona. Pored slanja informacije o temperaturi, uspostavljen sistem, koji omogućava kontrolu slanja podataka preko GSM ili GPRS konekcije i dobijanja povratnih informacija, može da se koristi i u mnoge druge kompleksnije svrhe.

ZAHVALNICA

Rad je urađen u okviru Studentskog istraživačkog rada u Laboratoriji za Mikrotalasnu tehniku i bežične komunikacije, Elektronskog fakulteta u Nišu.

LITERATURA

[1] Opis Arduino Uno ploče https://www.circuito.io/blog/arduino-uno-pinout/

[2] Arduino Uno *overview* https://www.farnell.com/datasheets/1682209.pdf

[3] GPRS /GSM SIM900A MODEM WITH ARDUINO COMPATIBLE, https://researchdesignlab.com/projects/GPRSGSM%20SIM900A%20MODEM%20with%20aurdino%20compatible.pdf
[4] SIM 900 Hardware design, https://simcom.ee/documents/SIM900/SIM900_Hardware%20Design_V2.05.pdf
[5] Dogan Ibrahim i Ahmet Ibrahim, "*GSM/GPRS projekti*", Agencija Eho,
[6] LM335 temperaturni senzor, http://www.ti.com/lit/ds/snis160e/snis160e.pdf

Projektovanje i realizacija elektronske brave za otključavanje vrata pomoću tastature

Andreja Cvetković

Sadržaj – Cilj projekta je realizacija elektronske brave pomoću *Arduino Uno* razvojnog okruženja i programiranja u *Arduino* IDE softverskom programu koji omogućuje interakciju između računara i *Arduino* okruženja.

I. UVOD

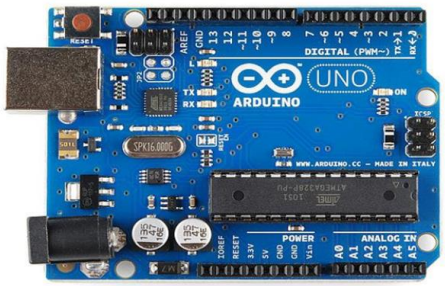
Elektronska brava je sistem zaštite objekata, privatnih kuća, sefova, stanova... Nekada je bio primenjivan klasičan način zaštite pomoću mehaničkih uređaja, dok se danas princip rada zasniva na programskom kodu koji je integrisan u mikrokontroler. Šifra i ostali oblici identifikacije koje korisnik unosi se čuva u EEPROM (*Electrically Erasable Programmable Read-Only Memory*) memoriji da bi podaci i prilikom nestanka struje ostali sačuvani. Otključavanje je moguće izvršiti nakon što je identifikacija potvrđena, pri čemu izvršni element pokreće mehanički sklop za otključavanje. Postoje i dodatne sigurnosne mere, kao što je trajno zaključavanje, zbog mogućnosti izvršenja pogrešne identifikacije.

II. STRUKTURA ELEKTRONSKE BRAVE

Funkcionalne celine elektronske brave su: *Arduino* Uno razvojno okruženje, *Arduino* IDE, LCD, servo motor SG90, *Keypad*, LED. Arduino Uno je platforma otvorenog koda, koja je namenjena za razvoj aplikacija i sa kojom je moguće na jednostavan način povezati hardverske i softverske komponente aplikacije [1]. *Arduino* Uno platforma poseduje 6 analognih pinova. Ovi pinovi su isključivo ulazni i uglavnom se na njih povezuju senzori. Pored analognih pinova, sadrži i 14 digitalnih pinova, koji mogu biti ulazni i izlazni. Od navedenih 14 digitalnih pinova, dva se koriste za UART komunikaciju, i to su 0-Rx i 1-Tx (slika 1). Sama struktura uređaja je potrebna da bude manjih dimenzija kako bi sama elektronska brava bila funkcionalnija i lakša za upotrebu.

Pisanje programskog koda se vrši u *Arduino* IDE softveru. Kod se sastoji iz dve dve funkcije, *void setup () i void loop (). D*eo koda koji je napisan u *void setup ()* funkciji se izvršava samo jednom, obično se u ovoj funkciji definiše koji su ulazni i izlazni pinovi, kao i vrednost promenljivih. Međutim, u *void loop () f*unkciji je deo koda čije se naredbe izvršavaju u beskonačnoj petlji [2].

Andreja Cvetkovic je student master studija sa Katedre za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: andreja40.ac@gmail.com



Slika 1. *Arduino* Uno razvojno okruženje [1].

III. PRINCIP FUNKCIONISANJA ELEKTRONSKE BRAVE

Postoji mnogo tipova elektronskih brava, ali na najosnovnijem nivou, su elektronske brave koje se otključavaju pomoću šifre. Koristimo tastaturu za unos šifre i otključavanje vrata.

Način funkcionisanja brave je da korisnik unese odgovarajuću šifru pomoću koje otvara vrata. Ako je uneta odgovarajuća šifra vrata se otvaraju i uključuje se zelena LED koja signalizira otvorena vrata i unetu tačnu šifru u suprotnom ako je uneta pogrešna šifra vrata ostaju zatvorena i uključuje se crvena LED koja signalizira da je uneta pogrešna šifra i da vrata ostaju zatvorena.

IV. KOMPONENTE ZA REALIZACIJU

Komponente su odabrane na osnovu svojih individualnih karakteristika, ali i na osnovu mogućnosti povezivanja sa ostalim delovima sistema.

A. Keypad 4x4

Unos šifre ili navođenje kroz meni elektronske brave vrši se pomoću tastature (*Keypad*) 4x4 (slika 2). Tastatura sadrži 16 tastera i ima 8 izlaznih pinova kod kojih prva 4 služe za redove, dok preostala 4 služe za kolone. Pozivanje biblioteke za rad sa tastaturom vrši se naredbom *#include <Keypad.h>*.

B. Servo motor SG90

Servo motor SG90 [4] može da vrši rotaciju kraka koja odgovara uglu od 180˚, po 90˚ u oba smera. Ima pin za napajanje (VCC), za masu (GND) i za signal (PWM).



Slika 2. *Izgled Keypada 4x4* [3].



Slika 3. Servo motor SG90 [4].

Ugao rotiranja motora određuje se širinom impulsnog signala koji se šalje do motora. Ukoliko je trajanje signala duže, veći je i ugao rotiranja servo motora.

C. LCD 16x02

LCD sa I^{2}C koristi se ako bi se korisniku olakšao rad sa elektronskom bravom i kako bi znao u svakom trenutku u kom stanju je sam *Arduino* i kada može uneti šifru. LCD koji se koristi u ovom projektu je 16x2, što znači 16 znakova u svakom redu sa maksimalno 2 reda (Slika 4). Radni napon LCD-a je 5 V DC. LCD sadrži dva registra: *Command* i *Data*. *Command* registar prima isključivo naredbe koje su mu date, kao npr. brisanje trenutnog sadržaja na ekranu, postavljanje kursora, kontrola ekrana... Sa druge strane se nalazi *Data* registar koji prima podatke koji se moraju prikazati na LCD-u. Podatak je u ASCII kodu što znači da se podaci šalju u 8 bitnoj kombinaciji te je potrebno 8 digitalnih pinova.

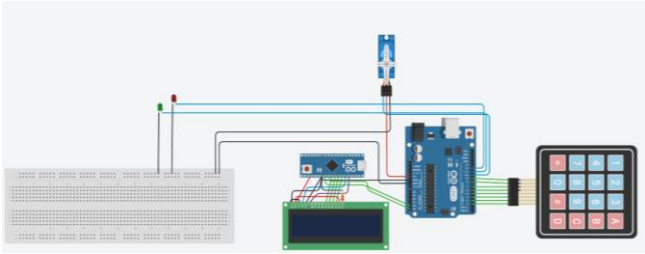


Slika 4. Izgled Keypada 4x4 [5].

D. I^{2}C komunikacija

Ovaj vid komunikacije se koristi na malim rastojanjima, uglavnom za povezivanje komponenata na štampanoj ploči. Za ovaj komunikacioni protokol se koriste dve fizičke linije: SCL (*Serial Clock*) koja se koristi za prenos podataka taktnog signala i SDA (*Serial Data*) za prenos podataka. Brzina prenosa podataka je veća od 100 kB/s i moguće je povezati 128 uređaja u sistemu. U ovom projektu nam smanjuje broj pinova koji se koriste za povezivanje LCD.

E. Šema projekta



Slika 5. Električna šema realizovana u *TinkerCad [6]*.

Pre povezivanja i same realizacije projekta, treba se realizovati šema projekta i odraditi sam prototip. Realizacija šeme se vrši u softverskom program *TinkerCad*. Prvo je potrebno pronaći komponente koje koristimo u projektu, a zatim smo ih povezali, komponente koje se koriste su *Keypad, Servo motor, Arduino Uno,LCD 1602 I LED*. Povezivanje šeme je prikazano na slici 5.

V. PRAKTIČNA REALIZACIJA UREĐAJA

Pri prvom pokretanju vrši se inicijalizacija i postavljanje početnih stanja vrednosti određene varijable. Na LCD-u se ispisuje poruka i nakon toga se prikazuje glavni meni (Slika 6).



Slika 6. Početno ispisivanje na ekranu.

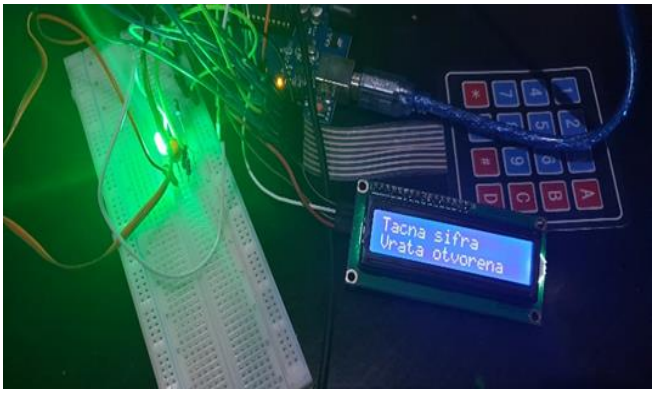
Pritiskom tastera za unos šifre na ekranu se ispisuje sledeća poruka „Unesite sifru”, dok se u drugom redu unosi šifra, kao što je prikazano na slici 7.



Slika 7. Izgled ekrana prilikom unosa šifre.

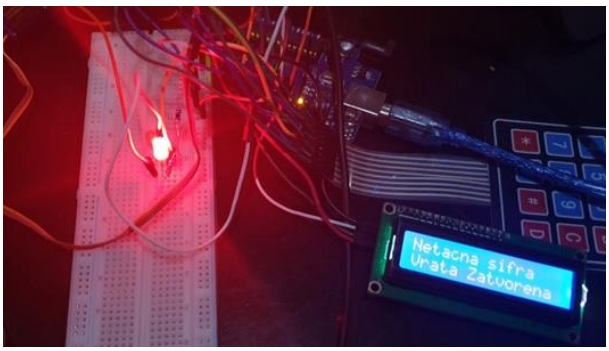
Kao što se može videti na slici 7, prilikom unosa šifre, prethodni broj se zamenjuje zvezdicom.

Korisnik unosi šifru, nakon toga se vrši provera i ako je šifra tačna na ekranu se ispisuje poruka prikazana na slici 8.



Slika 8. Poruka na ekranu prilikom unosa šifre.

U suprotnom na ekranu biće ispisana poruka prikazana na slici 9.

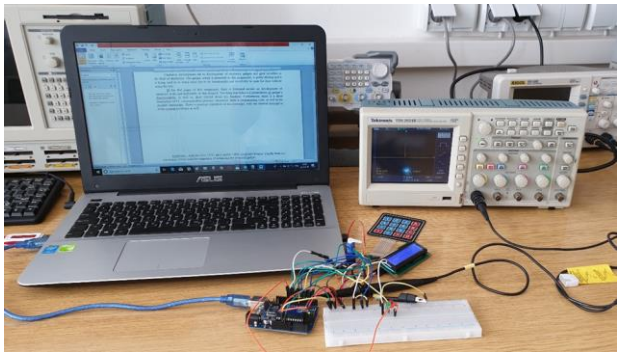


Slika 9. Poruka na ekranu prilikom unosa netačne šifre.

VI. RAD U LABORATORIJI

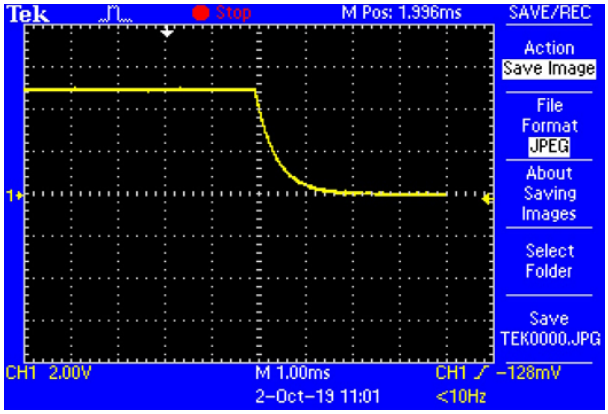
Prilikom rada u laboratoriji, potrebno je izvršiti merenja kako bi se uočile neke anomalije koje poseduje elektronska brava i sami elektronski uređaji. Pritiskom tastera na tastaturi, se očekuje da se na ekranu zabeleži jedan kontakt, odnosno ispiše poruka. Međutim, postoji početni kontakt, blagi odskok, zatim drugi kontakt kada se odskok završi, još jedan odskok nazad, ove sve promene se ne mogu videti, jer se sve to dešava u milisekundama, pa se pomoću osciloskopa vrše merenja kako bi se uočile promene. Ova anomalija kod elektronskih uređaja se naziva poskakivanje (*Bouncing*). Uobičajeno rešenje je softver za odvajanje koji osigurava da se u toku određenog vremena (obično milisekunde) može registrovati samo jedan digitalni signal, odnosno potrebno je napraviti određenu pauzu između pritiska dva taster.

Izvršena su osnovna podešavanja osciloskopa (slika10). Povezana je sonda osciloskopa sa testnom tačkom na Arduinu, gde je priključen jedan od pinova *Keypad*a (slika 10). Preko USB porta dovodi se odgovarajuće napajanje za Arduino i pritiskom odgovarajućeg tastera na *Keypad*-u dobijamo određene karakteristike na osciloskopu.



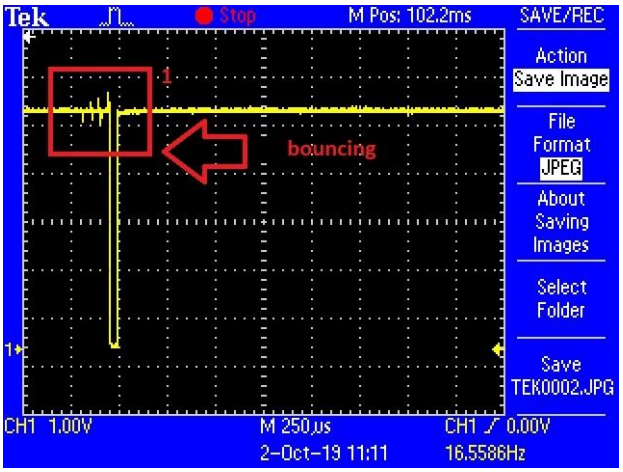
Slika 10. Povezivanje uređaja sa osciloskopom.

Tokom posmatranja signala na osciloskopu snimljena je prednja i zadnja ivica signala i nakon toga je zaustavljena akvizicija. Snimljeno je mesto prelaza sa naponskog nivoa logičke jedinice na naponski nivo logičke nule i analizirano je. Rezultati sa osciloskopa za *Keypad* su prikazani na slici 11.



Slika 11. Prikaz prelaza sa naponskog nivoa logičke jedinice na naponski nivo logičke nule pri otpuštanja tastera na *Keypad*u.

Na slici 12 se može videti poskakivanje signala, odnosno *Bouncing*.



Slika 12. Prikaz sa osciloskopa na kome se vidi poskakivanje signala (*Bouncing*).

Kako bi se otklonila anomalija u ovom uređaju, koristi se linija koda tj. gotova funkcija *keypad.setDebounceTime(100)*. Ova funkcija određuje vreme koje potrebno sačekati do narednog pritiska tastera, kako bi se izbegao, ali samo za *Keypad* i na ovaj način se rešavaju problemi sa poskakivanjem.

VII. ZAKLJUČAK

U radu su dati osnovni principi realizacije elektronske brave. Opisana je namena i princip rada ove tehnologije. Prikazano je projektovanje jednog sistema, gde su posebno objašnjene sve komponente. Realizovana je i maketa koja potvrđuje ispravan rad sistema.

Prilikom prvog pokretanja sistema vrši se inicijalizacija, nakon toga dolazi početni prikaz koji je vidljiv na ekranu LCD-a. Preko tastature biraju se određene opcije, unosom odgovarajuće šifre, dolazi do otključavanja brave.

Neka od unapređenja koja se mogu izvršiti su: da prilikom odabira tastera se emituje zvučni signal, da se uz pomoć tastature vrši promena šifre, da je moguće otključavati pomoću mobilnog telefona, da sistem radi na otisku prsta ili RFID kartice.

ZAHVALNICA

Rad je realizovan kao diplomski rad na modulu Elektronske komponente i mikrosistemi. Autor se zahvaljuje mentoru prof. dr Danijelu Dankoviću i doktorandu Nikoli Mitroviću na pruženom znanju i pomoći.

LITERATURA

[1] Softverski alat, *Arduino Uno* dostupan na linku: https://learn.sparkfun.com/tutorials/what-is-an-arduino
[2] Interfejs softverskog alata, *Arduino* IDE dostupno na linku: https://www.arduino.cc
[3] *Keypad* 4x4, *datasheet* dostupno na linku: https://components101.com/misc/4x4-*Keypad*-module-pinout-configuration-features-datasheet
[4] Servo motor SG90, *datasheet* dostupno na linku: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
[5] LCD 1602, *datasheet* dostupno na linku: https://www.diyelectronics.co.za/store/displays/229-lcd-display-16x2-white-on-blue-i2c-backpack-controlled.html
[6] Softverski alat, *Tinkercad* dostupan na linku: https://www.tinkercad.com/things/5rVQBsCylKQ-copy-of-i2c-lcd-liquidtwi2-emulator/editel

Govorni časovnik

Petar Virijević, Nikola Stojanović, Aleksa Ćirić, Ilija Veličković, Aleksandar Denić, Nemanja Ćirić, Nemanja Stojanović, Uroš Zdravković, Radmila Mitić, Predrag Šubarević

Sadržaj- U ovom radu je prikazana potreba za izradom govornog časovnika za slepe i slabovide osobe, kao i način izrade ovog časovnika i njegov princip rada.

I. UVOD

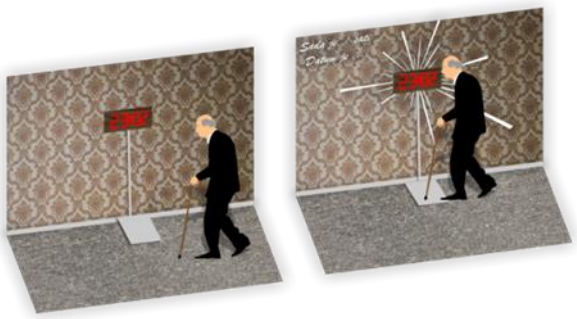
Razvoj ljudskog društva prati stepen povećanja kvaliteta života. To između ostalog znači povećanje kvaliteta zdravstvenog života svakog pojedinca. Osobe sa oštećenim vidom svakodnevno se susreću sa ograničenjima u obavljanju svakodnevnih delatnosti, utiču na njihov stepen samostalnosti i ograničenosti u kretanju [1]. Profesor Rupert Bourne procenjuje da će do 2050 godine broj slepih osoba porasti sa sadašnjih 36 miliona na čak 115 miliona, što je posledica sve veće starosti stanovništva. Zato je potrebno da društvo omogući svim ljudima veći kvalitet života [2].

Govorni časovnik je školski projekat koji je namenjen za prikazivanje i kazivanje tačnog vremena na zahtev korisnika. Časovnik je namenjen za korišćenje na javnim mestima i pre svega je namenjen slabovidim i slepim osobama. Korisnik aktiviranjem senzora koji se nalazi na obeleženoj stazi za slepe i slabovide osobe, aktivira senzor tako što stane na označeni deo na taj način dobija zvučni signal o tačnom vremenu. Na uređaju se takođe nalazi i sedmosegmentni displej LED traka, koji obezbeđuje svetlosni prikaz tačnog vremena.

II. PRINCIP RADA

Govorni časovnik je električni uređaj koji je projektovan i konstruisan sa ciljem da daje informacije, vizuelno i zvučno, o tačnom vremenu, kako na javnim mestima tako i u privatnim domaćinstvima. Uređaj se sastoji iz tastera, električnog sklopa i displeja. Taster, kada je pritisnut, šalje signal do glavnog kola. U njemu mikrokontroler obrađuje signal i na osnovu posebno odabranog modula na displeju prikazuje i na zvučniku daje informaciju o tačnom vremenu. Tačnost i pouzdanost u radu daje korišćeni softver, mikrokontroler i odabrane komponente. Mogućnost unapređenja ovog uređaja, kao što je informacija o trenutnoj temperaturi, UV zračenju i pritisku, ako i mogućnost korišćenja kao podsetnika o događajima za ličnu upotrebu, daje ovom uređaju veliki potencijal u ostvarenju velike komercijalne upotrebe.

Uloga govornog časovnika je da oglašava i prikazuje tačno vreme. Arduino kao logička jedinica, čita podatke sa modula u kućistu i emituje podatke o vremenu na displeju i zvučniku i na taj način svi korisnici mogu da vide i čuju tačno vreme. Vreme koje je potrebno za izvršavanje ove funkcije je veoma kratko, reda milisekundi, tako da je preciznost rada uređaja na veoma visokom nivou. Govorni časovnik funkcioniše tako što se potisna ploča, koja se nalazi na podu, aktivira tako što korisnik stane na nju. Pošto je pre svega, ovaj uređaj namenjen za slepe i slabovide osobe, deo do ploče i oko ploče je obeležen stazom za slepe. Površina nagazne ploče je 400x400 mm. Senzor koji se nalazi ispod ploče šalje signal do Arduina. Arduino obrađuje signal, uzima informaciju sa modula o tačnom vremenu i zvučni signal iz baze podataka emituje na zvučniku, takođe emituje na displeju i svetlosni signal. Displej je postavljen u drvenom kućištu, dmenzije su 400x250x100 mm. Kućište je pričvršćeno metalnim stubom sa postoljem za tlo a pored postolja se nalazi potisna ploča. Model Govornog časovnika prikazan je na slici 1.



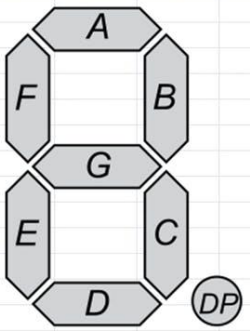
Slika 1. Prikaz rada Govornog časovnika.

III. PRAKTIČNA REALIZACIJA

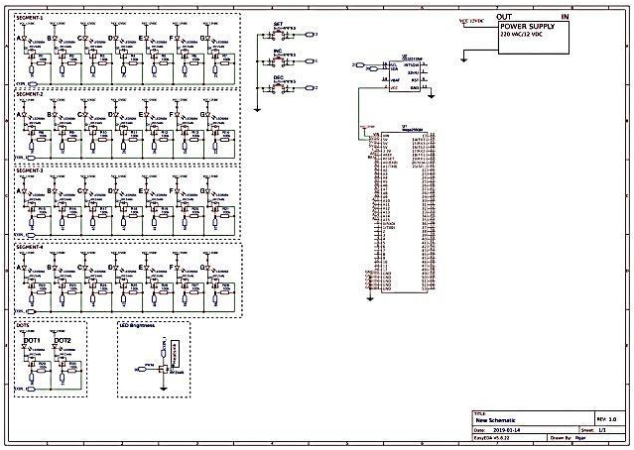
Govorni časovnik je uređaj koji predstavlja celinu hardverskog i softverskog dela. Šematski prikaz sistema dat je na slici 2.

Aleksa Ćirić, Ilija Veličković, Nikola Stojanović, Petar Virijević su učenici smera Tehnicar Mehatronike trećeg razreda. Aleksandar Denić, Nemanja Ćirić, Nemanja Stojanović i Uroš Zdravković su učenici drugog razreda, Radmila Mitić i Predrag Šubarević nastavnici stručnih predmeta Elektrotehničke škole "Nikola Tesla" u Nišu, Aleksandra Medvedeva 18, 18000 Niš, Srbija, e-mail: p.subarevic92@gmail.com, racamitic@gmail.com

	A	B	C	D	E	F	G	H	I	J	K	L
1	num	A	B	C	D	E	F	G	DEC			
2	0	1	1	1	1	1	1	0	126			
3	1	0	1	1	0	0	0	0	48			
4	2	1	1	0	1	1	0	1	109			
5	3	1	1	1	1	0	0	1	121			
6	4	0	1	1	0	0	1	1	51			
7	5	1	0	1	1	0	1	1	91			
8	6	1	0	1	1	1	1	1	95			
9	7	1	1	1	0	0	0	0	112			
10	8	1	1	1	1	1	1	1	127			
11	9	1	1	1	1	0	1	1	123			
12		b6	b5	b4	b3	b2	b1	b0				
13												
14	DO	35	33	31	29	27	25	23	SEG1			
15		34	32	30	28	26	24	22	SEG2			
16		49	47	45	43	41	39	37	SEG3			
17		48	46	44	42	40	38	36	SEG4			



Slika 2. Tabelarni prikaz rada sedmosegmentnih brojeva.



Slika 3. Prikaz povezivanja LED traka u 4 sedmosegmentnih brojeva.

Softverski deo sistema je program koji je programiran u mikrokontroleru, Arduinu. Mikrokontroler nudi velike mogućnosti za upotrebu, Nudi velike mogućnosti za upotrebu i ima širok spektar primena. U ovom projektu programiran je Arduino da upravlja sledece navedenim hardverskim komponentama. Glavnu komponentu sistema predstavlja Arduino Mega 2560. Arduino [8] predstavlja razvojni sistem otvorenog koda zasnovan na 8-bitnom Atmel AVR ili 32-bitnom AtmelAMR mikroprocesoru. Ploča Arduino razvojnog okruženja u osnovi sadrži: oscilator odnosno kristal koji reguliše vremenske uslove procesora (clock), naponske regulatore vrednosti 5 V i 3.3 V i USB-to-serial konvertor.

TABELA I

HARDVERSKI DEO SISTEMA ARDUINA PREDSTAVLJAJU SLEDEĆE KOMPONENTE:

Mikrokontroler	ATmega2560
Radni napon	5V
Napon napajanja	7-12V
Digitalni I/O pinovi	54 (14 PWM izlaz)
Analogni ulazni pinovi	16
DC struja po I/O pinu	40mA
DC struja za pin (3.3V)	50mA
Flash memorija	256KB
SRAM	8KB
EEPROM	4KB
Takt procesora	16MHz

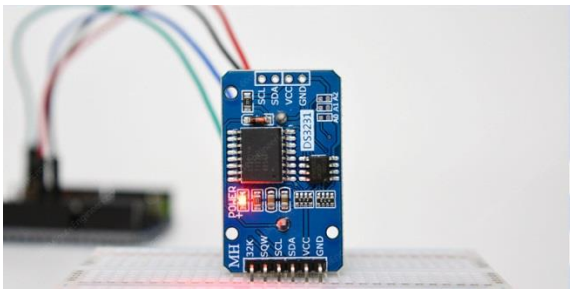
Pored navedenih komponenti Govorni časovnik sadrži i sledeće komponente:

1. RTC (DS3231) modul omogućava precizno čitanje vremena, i tačno iskazuje vreme glavnoj ploči. Ovaj modul zahteva sopstveno napajanje jednom litijumskom baterijom oblika novčića, koja zahteva zamenu na par godina. Određenim kodom može da pročita lokalno vreme i da bez veštačkog ubacivanja vremena, automatski meri lokalno vreme [3].
2. MP3 modul omogućava da preko *Arduina* emituje zvučni signal. Opseg zvučnog signala je širok i prati opseg govornog signala od 4 Hz do 44,8 kHz. Sve što je potrebno za zvuk je ovaj modul i jedna memorijska TF kartica, na koju su upisani zvučni zapisi brojeva koje Arduino čita i kazuje preko zvučnika. Na toj kartici je moguće izdvojiti 100 foldera sa po 255 zvučnih zapisa [4].
3. Zvučnik je izlazna komponenta, preko koje se emituje zvučni signal tačnog vremena koji je potreban korisniku. Arduino informaciju o tačnom vremenu dobija preko RTC modula, na osnovu dobijene informacije iz memorijske kartice odabira potrebne podatke, usnimljene brojeve i pravi kombinaciju za vreme. Nakon toga daje informaciju o tačnom vremenu preko zvučnika, hardverskog dela. *Arduino* se napaja preko breadboard-a i preko njega se napajaju i ostale komponente.
4. LED trake se koriste za formiranje 4 sedmosegmentno broja LED trake su tipa 5054, voltaže do 3V i snage 3W, i to može da je prikazano na slici 3. Ove LED trake mozemo podesiti određeni gradijent boje, i tim možemo olakšati čitanje sa časovnika u bilo kom delu dana[5].
5. N-kanalni tranzistor IRFZ44N služi za preusmeravanje signala na 4 sedmosegmentnog broja, tj LED traka. Njime se olakšava preusmeravanje signala i lako upravljanje LED trakama podeljenih na sedam segmenta [6].
6. Otpornik od 220 Ω se koristi za taster i njegova uloga je u stabilizaciji napona i u regulisanju rada sistema za aktiviranje.

IV. OPIS RADA

Časovnik za slepe koji ostvaruje svoj rad kontrolisanom komunikacijom hardverskog dela. Arduino je glavna komponenta ovog sistema, ona prima i daje informacije ulaznim i izlaznim komponentama. RTC modul (Real Time Clock modul) je modul koji meri lokalno vreme uz pomoć litijumske baterije oblika

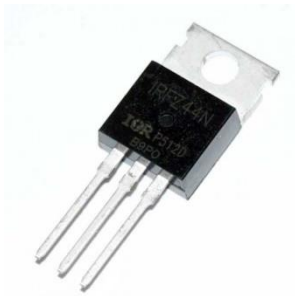
novčića. Ova baterija ima veliki radni vek, zahteva zamenu na par godina. MP3 modul sluzi da čita snimljeni zvuk raspoređen po datotekama sa memorijske kartice, i preko Arduina sklapa reč koja opisuje traženo vreme i datum. Signal koji aktivira Arduino je signal koji se dobija sa nagazne ploče. Takođe, za dati uređaj može da se koristi i taster. Zvučnik je izlazna komponenta koja nam omogućava iskazivanje satnice. Preko protoploče sve komponente povezuju se za Arduino.ss



Slika 4. RTC DS3231 [3].



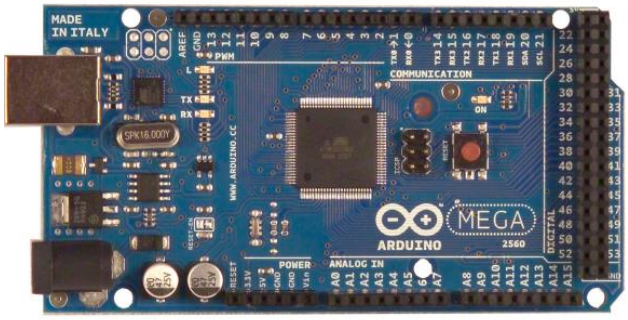
Slika 5. MP3 modul TF 16P [4].



Slika 6. N kanalni MOSFET IRFZ44N.

V. ZAKLJUČAK

Uređaj koji je realizovan pod nazivom „Govorni časovnik“ je zadovoljio potrebne karakteristike. Pod tim se podrazumeva da je jednostavan za upotrebu, da sa velikom tačnošću može da da podatak o tačnom vremenu, zvučno i brojčano. Niska proizvodna cena omogućava veliku konkurentnost na tržištu, jer se cena drugih sličnih uređaja za pomaganje slepim osobama kreće od 100 $ pa naviše. Takođe, ostvarena je saradnja sa Društvom slepih osoba u Nišu, koji su zainteresovani za ovaj rad. Daljom saradnjom sa ovim centrom, nadamo se unapređenju rada i njegovoj praktičnoj širokoj upotrebi kako kod slepih i slabovidih osoba, tako i širokoj upotrebi na javnim mestima .



Slika 7. Arduino Mega 2560 [7].

ZAHVALNICA

Ovaj red je finansiran od strane elektrotehničke škole „Nikola Tesla“ iz Niša. Autori se zahvaljuju nastavnicima i rukovodstvu škole na pomoći pri realizaciji rada.

LITERATURA

[1] Kvalitet života odraslih osoba sa oštećenjem vida, Branka JABLAN, Vesna VUČINIĆ, Marija VRANIĆ, Univerzitet u Beogradu.
[2] Detaljno istraživanje o procentu ljudi sa otežanim vidom ili oštećenim vidom, dostupno na linku: https://aru.ac.uk/people/rupert-bourne
[3] RTC datasheet je dostupan na linku: https://www.google.com/amp/s/lastminuteengineers.com/ds3231-rtc-arduino-tutorial/amp/
[4] MP3 datasheet je dostupan na linku: https://www.google.com/url?sa=t&source=web&rct=j&url=http://www.electronicoscaldas.com/datasheet/DFR0299-DFPlayer-Mini-Manual.pdf&ved=2ahUKEwi7xOmZ3L_lAhUlxKYKHSD5CUMQFjAAegQIARAB&usg=AOvVaw0hNeL6tX7cbJNEzJ0n4ekh
[5] Ideja i skica za povezivanje LED traka sa Arduinom se može naći na linku: https://create.arduino.cc/projecthub/Deep_Sky/digital-wall-clock-b8e8bf.
[6] Datasheet o N kanalnom MOSFETU možete naći na linku: https://alltransistors.com/mosfet/transistor.php?transistor=3398
[7] Datasheet o Arduinu 2560 se može naći na linku: https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf&ved=2ahUKEwjImYiN3r_lAhWpyqYKHer1DGYQFjADegQIARAB&usg=AOvVaw0IjOpadV2fyExiYk32boJX&cshid=157229253971

Softverska emulacija UART komunikacije

Dejan Anđelković

Sadržaj- U ovom radu je prikazan princip realizacije softverskog UART port-a na mikrokontroleru

I. Uvod

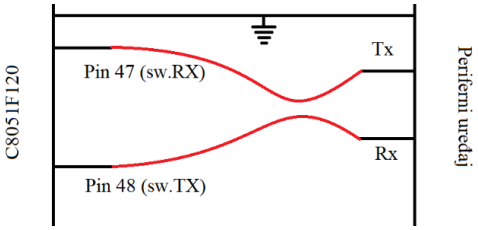
U modernoj tehnologiji postoje hiljade različitih mikrokontrolera raznih primena. Postoje vrste mikrokontrolera izuzetno naprednih mogućnosti koji u nekim slučajevima menjaju i računare, ali postoje i vrste malih, prostih mikrokontrolera sa izuzetno niskom potrošnjom energije. U zavisnosti od projekta koji se realizuje mikrokontrolerom, inženjer je u obavezi da nađe savršen mikrokontroler koji ispunjava sve uslove za realizaciju zadatka, dok pritom troši što manje energije kao i da je što niže cene. U nekim slučajevima, postoje mikrokontroleri koje sve ove uslove ispunjavaju sem jednog, nedostatak neophodnog interfejsa. Ovaj rad pokriva realizaciju i razvoj softverske emulacije jednog od tih interfejsa. U pitanju je interfejs asinhrone komunikacije *UART*.

II. Princip rada

A. Osnove UART komunikacije

UART predstavlja akronim za *Universal Asynchronous Receiver/Transmitter*. Koristi se za transfer serijskih podataka između dva uređaja, i predstavlja jedan od najstarijih vidova komunikacije koji je još uvek u čestoj upotrebi.

Za razliku od I^{2}C i SPI komunikacije, UART nije samo protokol, već se hardverski realizuje kao zasebno kolo unutar mikrokontrolera (slika 1) ili čak i kao eksterno integrisano kolo. Na ovoj slici nije korišćen protokol CTS/RTS, pa je dovoljno povezati dva pina za slanje i primanje podataka, ali i masu kao treće povezivanje.

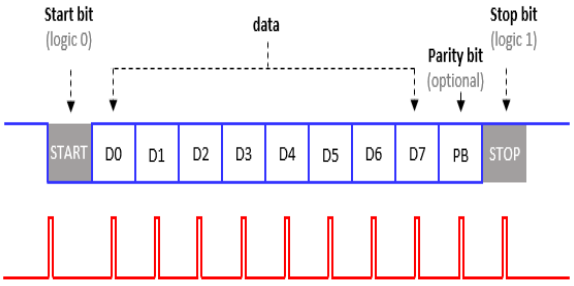


Slika 1. Povezivanje UART-om mikrokontrolera i perifernog uređaja.

Dejan Anđelković je student prve godine doktorskih akademskih studija na Elektronskom Fakultetu u Nišu. E-mail adresa: dejan.andjelkovic@elfak.rs

B. Princip rada UART komunikacije

Princip rada UART komunikacije je, kako i samo ime nalaže, asinhrona razmena podataka bez usaglašenog clock-a. Umesto clock-a, UART podatke prenosi preko TX pina dodavanjem početnog bita (*starting bit*) i završnog bita (*stop bit*). Ovi bitovi ukazuju na početak i kraj paketa podataka koji se prenosi. Kada uređaj, koji je u ulozi primaoca u ovom primeru, detektuje početni bit na svom RX pinu – počinje sa čitanjem bitova na specifičnoj frekvenciji (*baud rate*). Detekcija završnog bita je znak uređaju da je transfer paketa podataka završen. Vremenom se UART komunikacija razvijala u zavisnosti od primene i potrebe, pa sada postoji i nekoliko varijacija prethodno opisanom načinu rada. Moderni UART uređaji sada mogu funkcionisati i sa tzv. bitovima parnosti (*parity bits*) ali i sa više završnih bitova. Takođe postoji i vrsta handshake potvrde između modernih UART uređaja koja se vrši preko dva odvojena pina. Ti pinovi su označeni sa CTS (*clear-to-send*) i RTS (*request-to-send*).



Slika 2. Prikaz UART paketa podataka u obliku signala.

C. Opis mikrokontrolera i pratećeg hardvera

Hardver korišćen za realizaciju ovog projekta je mikrokontroler proizvođača Silabs, oznake C8051F120 [1] (prikazan na slici 3) kao i prateći razvojni modul. Osnovne karakteristike ovog 8-bitnog mikrokontrolera su:

- Jezgro mikrokontrolera: 8051
- Maksimalna clock frekvencija: 100 MHz
- Memorija programskog koda: 128 kB
- Dostupni interfejsi: I^{2}C, UART (x2), SPI, SMBus
- 5 tajmera, 8 ADC kanala



Slika 3. Mikrokontroler C8051F120.

Mikrokontroler je već bio primenjen kao sastavni deo industrijskog proizvoda, a daljim razvojem istog ponestali su slobodni vidovi komunikacije sa perifernim uređajima. Tada je postalo neophodno razviti softversku emulaciju trećeg UART-a.

III. Softver i dizajn sistema

A. Cilj softvera i razvojno okruženje

Realizacija softverske emulacije UART komunikacija je kombinacija koda pisanog u asembleru i koda pisanog u programskom jeziku C. Okruženje korišćeno za realizaciju ovog projekta je *Silabs IDE* [2], bazirano na kompajleru KeilV5. Interfejs za programiranje mikrokontrolera jeste *debug adapter* istog proizvođača. Deo koda pisan u asembleru ima zadatak da mikrokontroler na najnižem nivou softvera uputi koje namene određenim pinovima da dodeli, ali i da odredi koji baud rate će biti očekivan. U ovom projektu, emulira se najprostija varijanta UART komunikacije, pa se deklarišu samo dva pina (RX i TX). Deo koda pisan u programskom jeziku C je zadužen za realizaciju funkcija višeg nivoa, povezivanje UART komunikacije sa jednim od tajmera, kao i za određivanje *interrupt* procedure.

B. Deo asembler koda

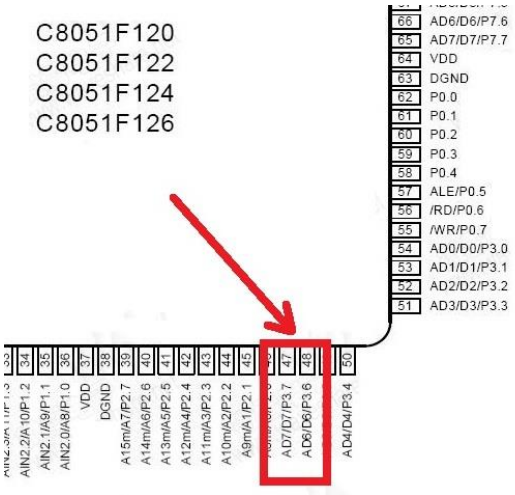
```
1.   ?SU?PUTC SEGMENT CODE
2.   ?SU?GETC SEGMENT CODE
3.
4.   PUBLIC _putc
5.   PUBLIC getc
6.
7.   txd_pin EQU P3.1
8.   rxd_pin EQU P3.0
9.   BITTIM EQU 45
10.  RSEG ?SU?PUTC
11.  _putc:
12.  push ACC
13.  Push PSW
14.  mov a,r7
15.  CLR txd_pin
16.  MOV R0,#BITTIM
17.  DJNZ R0,$
18.  MOV R1,#8
19.  putc1:
20.  RRC A
21.  MOV txd_pin,C
22.  MOV R0,#BITTIM
23.  DJNZ R0,$
24.  DJNZ R1,putc1
25.  SETB txd_pin
26.  RRC A
27.  MOV R0,#BITTIM
28.  DJNZ R0,$
29.  POP PSW
30.  pop ACC
31.  RET
32.
33.  RSEG ?SU?GETC
34.  getc:
35.  Push ACC
36.  Push PSW
37.  JB rxd_pin,$
38.  MOV R0,#BITTIM/2
39.  DJNZ R0,$
40.  JB rxd_pin,getc
41.  MOV R1,#8
42.  getc1:
43.  MOV R0,#BITTIM
44.  DJNZ R0,$
45.  MOV C,rxd_pin
46.  RRC A
47.  DJNZ R1,getc1
48.  mov r7,a
49.  POP PSW
50.  pop ACC
```

Na samom početku asembler koda deklarišu se segmenti *PUTC* (preko koga će se slati podaci) i *GETC* (preko koga će se primati podaci). Za oba segmenta se deklariše po jedan pin. Pin P3.1 je deklarisan kao *Txd* pin,

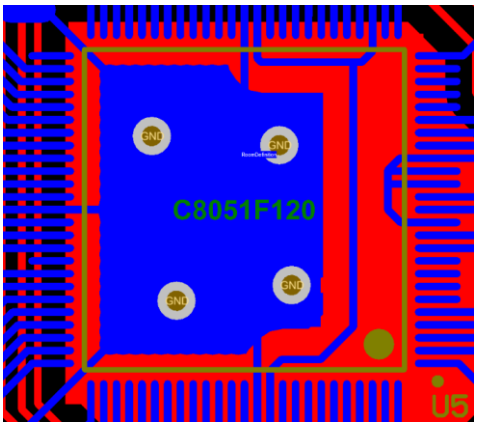
namenjen segmentu *PUTC*, a pin P3.0 je deklarisan kao *Rxd* pin, namenjen segmetnu *GETC*. Takođe je deklarisan i *BITTIM*, vrednost kojom će se deliti clock procesora i na taj način definisati *baud rate* UART komunikacije. Oba segmenta koda se sastoje od detaljne analize podataka koji se šalju ili primaju. Oba procesa počinju čitanjem ili slanjem startnog bita, nakon čega dolazi prenos podata u 8 bita, a na samom kraju se šalje ili čita završni bit.

IV. Simulacija i projektovanje

Projektovanje rada, kao i sama analiza mikrokontrolera je rađena izradom PCB fajlova kao i snimanjem signala osciloskopom. Realizacija PCB fajlova je znatno ubrzala planiranje neophodnih pinova, formiranje fizičke veze između mikrokontrolera i uređaja sa kojim komunicira, kao i pristup osciloskopom za dalju analizu dobijene komunikacije. Osciloskopom je rađena provera dobijene UART komunikacije, ostvareni baud rate između uređaja, kao i validnost poslatih i primljenih paketa podataka sa strane softverski emuliranog UART uređaja.

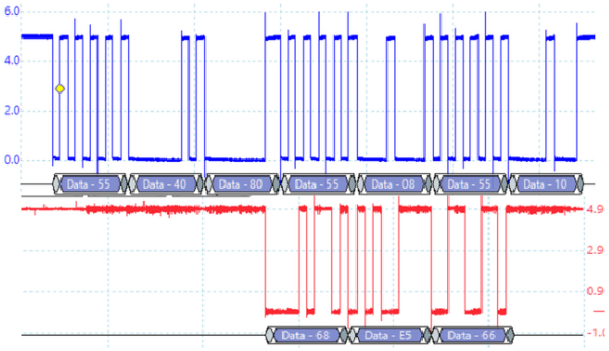


Slika 4. Pinout mikrokontrolera i odabrani pinovi za Rx i Tx.



Slika 5. Prikaz mikrokontrolera u *Altium Designer*-u.

Za projektovanje PCB fajlova korišćen je softverski paket *Altium Designer* [3], a rezultat se može videti na slici 5. Osciloskop korišćen za analizu signala razmenjenih podataka je proizvođača *PicoTechnology*, oznake *PicoScope 3204D* [4]. Signal je snimljen i prikazan na slici 6. Svaki signal UART komunikacije je odvojen segmentom *data*, pa se jasno mogu videti svih 8 bitova jednog paketa podataka. Pored svakog od *data* segmenata stoje po jedan bit na početku i na kraju, koji naravno predstavljaju početni i završni bit.



Slika 6. Osciloskopom prikazan signal.

V. Zaključak

Ovako realizovana softverska emulacija UART komunikacije predstavlja memorijski najlakšu verziju, ali i najosnovnijih performansi. Dodeljivanjem dodatnih pinova, i uvođenjem logike za RTS i CTS znatno se šire mogućnosti emulacije, ali se povećava i memorija dodatog koda. Razvoj softvera za dinamičku promenu baud rate-a je izvodljiv, ali zbog prevelike razlike u potrebnoj memoriji se često izostavlja. UART uređaji uglavnom komuniciraju sa unapred poznatim uređajima, kao i po unapred uspostavljenim uslovima. Fiksiran baud rate zbog toga ne predstavlja problem.

Zahvalnica

Autor se zahvaljuje prof. dr Danijelu Dankoviću na mentorstvu i podršci, kao i kolegi Ivanu Lončaru na teorijskoj i praktičnoj asistenciji.

Literatura

[1] Silabs C8051F120 datasheet, https://www.silabs.com/documents/public/data-sheets/C8051F12x-13x.pdf
[2] Silabs IDE, https://www.silabs.com/products/development-tools/software/8-bit-8051-microcontroller-software
[3] Altium Designer, https://www.altium.com/altium-designer/
[4] PicoScope 3204MD, https://www.picotech.com/oscilloscope/3000/picoscope-3000-oscilloscope-specifications

RFID čitač kartica za evidenciju rada zaposlenih

Petar Kotnik

Sadržaj - Evidencija rada zaposlenih oduvek je oblast u kojoj su nove tehnologije i izumi dobrodošli. Svaki vlasnik firme, šef ili direktor ovu temu smatra za prioritet i nešto što će njega kao poslodavca zaštititi od mogućih malverzacija zaposlenih. Cilj ovog rada je dati teorijski pregled RFID čitača kartica za evidenciju rada zaposlenih.

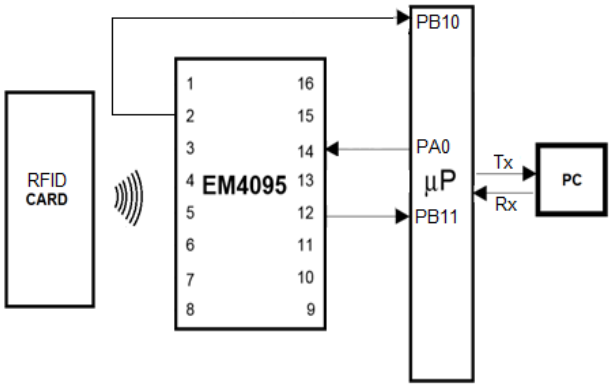
I. RFID TEHNOLOGIJA

RFID je sistem daljinskog slanja i prijema podataka pomoću RFID kartica [1]. RFID kartica je veoma mali objekat koji se može zalepiti ili ugraditi u željeni proizvod i uređaj. RFID kartice sadrže u sebi antenu koja im omogućava prijem i slanje radio-talasa od RFID primopredajnika. Neku veću primenu, kao i primenu u aplikacijama za praćenje, dobio je osamdesetih godina i brzo zadobio veliku pažnju zbog svoje sposobnosti da prati pokretne objekte. Kao prefinjena tehnologija, sa neslućenim mogućnostima primene, on se stalno razvija i spektar mogućih upotreba ove tehnologije se stalno širi. Pretpostavlja se da je prvi istraživački rad koji je objavljen delo Hari Stokmana koji je taj rad objavio 1948. godine pod naslovom ”Komunikacija kao odraz moći” i bilo je potrebno skoro 40 godina da bi RFID kartice zaživele u praksi [2].

II. PRINCIP RADA

STM32 mikrokontroler predstavlja mozak ovog sistema. Kada uz antenu koja je povezana na EM4095 dekoder prinesemo RFID karticu, dolazi do napajanja mikroprocesora unutar kartice koji frekvencijom od 125 kHz komunicira sa EM4095. Ova komponenta frekventno modulisani signal sa kartice pretvara u Mančester kod i u tom obliku šalje informaciju mikrokontroleru u vidu 64-bitne reči, uz naravno taktni signal. Mikrokontroler ovaj Mančester kod dekodira i na osnovu njegovog sadržaja prepoznaje koji je radnik u pitanju Prepoznavanje radnika može se obaviti na dva načina, mikrokontrolerom ili računarom. Mikrokontroler sa računarom komunicira UART komunikacijom i šalje mu ili ime i prezime korisnika, ili 64-bitnu reč. Ukoliko se prepoznavanje zaposlenog vrši u računaru, taj računar mora imati bazu podataka radnika i kodova njihovih kartica. Tehnika koja se dosta češće koristi je prepoznavanje radnika putem računara zbog njegove memorije.

P. Kotnik, Elektronski fakultet u Nišu, modul Elektronske komponente i mikrosistemi, Univerzitet u Nišu, Aleksandra Medvedeva 14, Niš, Srbija, E-mail: kotnik.petar@gmail.com



Slika 1. Blok dijagram sistema.

III. STM32F072B DISCOVERY

STM32F072 DISCOVERY predstavlja jeftino razvojno okruženje u oblasti programiranja mikrokontrolera [3]. Pored njegove cene odlikuje ga lako korišćenje i brzo prilagođavanje korisnika na njega. Ovo razvojno okruženje dozvoljava projektantu da otkrije sve napredne opcije ove familije mikrokontrolera. Ono što ovo okruženje čini lakšim za korišćenje od samog mikrokontrolera su periferije koje sadrži, kao i mnoštvo primera programa koji dolaze u paketu sa njim.

IV. EM4095

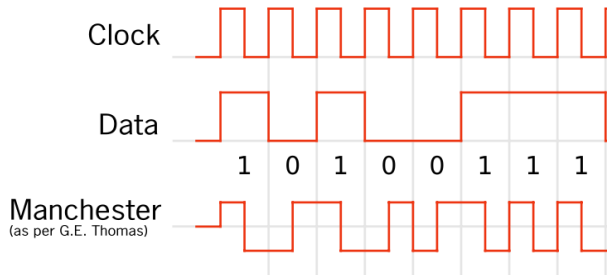
EM4095 (nekada nazvan P4095) čip predstavlja CMOS kompatibilno *transciever* kolo koje se koristi u RFID stanicama kako bi vršilo određene funkcije Uz pomoć samo nekoliko dodatnih komponenata EM4095 generiše elektromagnetno polje frekvencije 125 kHz potrebno za čitanje i pisanje RFID tagova (kartica). EM4905 je sa mikrokontrolerom povezan pomoću četiri signalne linije. One su RDY/CLK, MOD, SHD i DEMOD_OUT. SHD pin definiše *sleep* način rada. Kada se na njemu postavi signal logičke jedinice, EM4095 radi u *sleep* modu i modul se smatra isključenim. Signal MOD određuje način rada (čitanje ili pisanje). Za čitanje ovaj signal postavlja se na nizak naponski nivo. RDY/CLK linija služi za sinhronizaciju, na način da generiše signal takta od 125 kHz koji mikrokontroler koristi za dalju demodulaciju. DEMOD_OUT pin sadrži demodulisan izlazni signal. Dobijeni signal na mikrokontroleru se sastoji od dva dela: prvi deo je signal takta frekvencije 125 kHz, dok je drugi signal podatka (čija je osnovna frekvencija jednaka signalu takta deljenog s faktorom 16, 32 ili 64).

V. MANČESTER KOD

U telekomunikacijama i računarstvu, Mančester kodiranje ili fazno kodiranje je način predstavljanja logičkog simbola fizičkim signalom, na način da se nivo signala uvek menja kod svakog prenesenog bita. Za jednu vrednost signal počinje na visokom nivou, pa se na polovini intervala menja na nizak, dok je za drugu vrednost to obrnuto. Zavisno od redosleda binarnih vrednosti, signal se može i ne mora menjati na prelazima iz vrednosti u vrednost. Ti prelazi na granicama ne nose informaciju, informaciju nose samo prelazi na sredinama intervala.

Postoje dve konvencije. Prvu je objavio G. E. Thomas godine 1949. godine i prema njoj se logička nula predstavlja promenom s niskog na visoki nivo, a logička jedinica prelazom sa visokog na nizak nivo. Druga je obrnuta, koristi se u IEEE 802.4 (token bus), i u IEEE 802.3 (žični Ethernet) pri nižim brzinama. Obe konvencije imaju brojne sledbenike.

Za razliku od non-return-to-zero metode koda, Mančester kodiranje ima prelaz nakon svakog intervala signala pa se time rešava problem sinhronizacije, tj. oporavka signala vremenskog vođenja, ali se dvostruko povećava potrebni frekventni opseg. Široko se primenjuje, ali samo u situacijama gde fizički raspoloživi frekventni opseg potpuno premašuje potrebe uz zadatu brzinu prenošenja podataka.



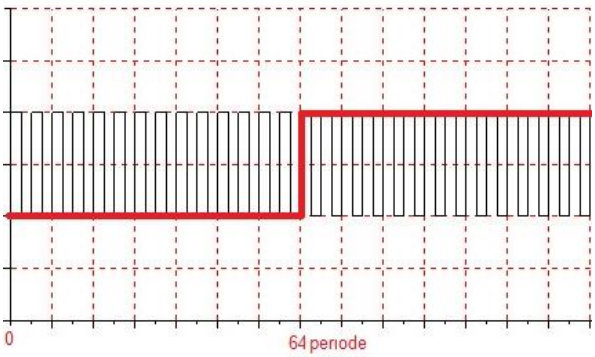
Slika 2. Prikaz mančester koda.

VI. ALGORITAM I DETALJAN OPIS RADA UREĐAJA

Inicijalizacija pinova i konfiguracija periferija mikrokontrolera urađena je u programskom alatu STM32CubeMX. CubeMX je softver koji je grafičkog tipa što dosta olakšava podešavanje svih početnih parametara. Kada se inicijalno podešavanje završi CubeMX učitava programski kod u Keil alat u kome se programiranje nastavlja. Svaki digitalni signal pa i Mančester mora imati taktni signal pomoću kojeg bi mogao biti dekodiran u prijemniku. Taktni signal u komunikaciji EM4095 sa mikrokontrolerom funkcioniše upravo ovako, razlika je samo u tome što jedan bit Mančester signala ne traje jednu periodu takta već 64 periode. Ova pojava znatno olakšava dekodiranje.

Prvi uslov na koji treba obratiti pažnju pri dekodiranju jesu startni bitovi. RFID kartica sadrži 64-bitnu informaciju u kojoj devet početnih bitova ima vrednost jedan. Ovo znači da mikrokontroler treba da uzme u obzir 64-bitnu reč samo ukoliko počinje sa devet jedinica. Mikrokontroler prvo pita da li je pristigli bit prvi, i ako jeste da li je jedinica. Ukoliko jeste jedinica ona se upisuje na prvu poziciju u nizu i čeka se naredni bit. Ukoliko nije jedinica sve promenljive se vraćaju na nulu (u ovom slučaju ostaju na nuli) i čeka se prvi validan bit. Ukoliko je prvi bit jedinica određene promenljive dobijaju vrednosti. Promenljiva kartica, predstavlja član niza koji predstavlja sadržaj sa kartice i ona dobija vrednost jedan. Start_bitovi predstavlja brojač koji beleži broj jedinica koje su na ulazu i kada njih bude 9 algoritam ulazi u drugu petlju. U ovom delu koda promenljiva *clock* dobija vrednost 32 zbog toga što se promena Mančester koda uvek dešava na polovini ukupnog ciklusa taktnog signala. Prvi_bit predstavlja *flag* promenljivu koja je indikator da je prvi bit primljen i da ni jedan naredni nije prvi.

Posle prvog primljenog bita koji ima vrednost jedan, jedino o čemu mikrokontroler vodi računa je da li i ostalih osam bitova imaju vrednost jedan. U ovom delu algoritma takođe se upisuju vrednosti bitova u elemente niza kartice ali se inkrementira i promenljiva start_bitovi koja prati koliko startnih bitova je pristiglo uspešno, kako bi posle njih devet promenila režim rada. U ovom delu koda prvi put se pojavljuje provera promenljive *clock*. Promenljiva *clock* je pri prvoj promeni vrednosti u Mančester magistrali postavljena na 32 i skoro uvek će se oko te vrednosti *clock*-a i u narednim bitovima desiti promena. Ova promenljiva služi da omogući mikrokontroleru detektovanje praznog bita. Pod praznim bitom smatra se situacija kada nema promene nivoa napona, a Mančester kod je takav da je uvek mora imati ukoliko se informacija prenosi. Zbog ove mogućnosti mikrokontroler koristi *flag* pod nazivom dozvola. Na 20. periodi *clock* signala *flag* dozvola biće podignut. U rutini koju aktivira promena Mančester signala nalazi se naredba za spuštanje ovog *flag*-a. Na 45. periodi *clock* signala sledi ispitivanje da li je flag podignut ili spušten. Ukoliko je spušten, sve je u redu jer to znači da se rutina promene u Mančesteru dogodila i spustila ga. Ukoliko se *flag* ne spusti nastaje problem jer to znači da je preskočen bit u detekciji ili ga nije ni bilo. Očitavanje u ovom slučaju kreće iz početka.



Slika 3. Približna reprezentacija izgleda signala.

Pre uslova za prvi bit, mikrokontroler ispituje sa kog pina je došla vest o *interrupt*-u. Ukoliko je sa magistrale Mančester koda ulazi u rutine koje su objašnjene iznad. Ukoliko je rutina pokrenuta od strane taktnog pina, program ulazi u potpuno drugu granu. U ovoj grani proverava se nekoliko uslova a prvi je promenljiva kreni_clk koja ukoliko nije jedan, program neće nastaviti sa ispitivanjem. Ukoliko kreni_clk nije jedan to znači da još uvek ni jedan bit Mančester koda nije stigao, pa bi u tom slučaju mikrokontroler brojao taktne signale u prazno. Ovo bi opterećivalo čip bez ikakve potrebe što nije poželjno.

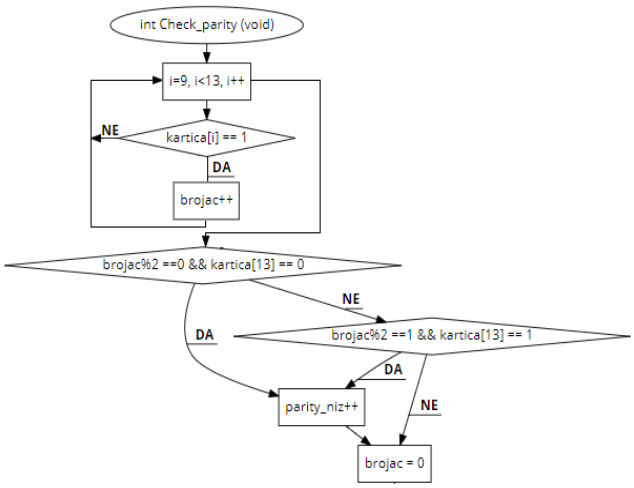
Sledeći uslov predstavlja ispitivanje promenljive *clock*, ukoliko je ona dostigla vrednost 64, treba biti vraćena na jedan kako bi ciklus brojanja krenuo iz početka. Kada *clock* iznosi 16, promenljiva dozvola postaje jedan i čeka Mančester promenu da je spusti. Na 48 periodi *clock*-a ispituje se dozvola i odlučuje da li se *reset*-uje očitavanje. Ukoliko promenljiva dozvola ne bude spuštena na logičku nulu, to će značiti da je došlo do praznog bita i automatski do netačnog Mančester koda.

Iako taktni signal i signal informacije budu skoro uvek perfektno isklađeni, ipak može doći do greške poslate informacije zbog mnogo faktora kako hardverskih tako i softverskih. Upravo iz ovog razloga pored 64-bitne informacije koju nosi RFID kartica, ona nosi unutar ovih bitova mogućnost provere validnosti poslatih podataka. Sredstvo za proveru validnosti poslatih bitova predstavljaju bitovi parnosti. Postoje dve vrste *parity* bitova konkretno u ovom nizu koje prima mikrokontroler. Prva vrsta su redni *parity* bitovi. Oni predstavljaju deset grupa od po četiri bitova unutar kojih se proverava parnost. Drugu grupu predstavljaju *parity* bitovi kolone. Na slici 4 predstavljeni su *parity* bitovi obe vrste.

1	1	1	1	1	1	1	1	1	9 header bitova
8 identifikacionih bitova korisnika				D00	D01	D02	D03	P0	
				D10	D11	D12	D13	P1	
32 bita informacije				D20	D21	D22	D23	P2	
				D30	D31	D32	D33	P3	
				D40	D41	D42	D43	P4	
				D50	D51	D52	D53	P5	
				D60	D61	D62	D63	P6	
				D70	D71	D72	D73	P7	
				D80	D81	D82	D83	P8	
				D90	D91	D92	D93	P9	10 linijskih parity bitova
				PC0	PC1	PC2	PC3	S0	

Slika 4. Parity bitovi reda i kolone.

Prvu grupu parnih bitova reda predstavlja niz D00, D01, D02, D03 i P0. Provera validnosti ovog niza vrši se brojanjem jedinica u njegovim „D" članovima. Broj jedinica u ovom nizu podeljen sa 2 treba imati ostatak jednak „P" članu niza. U ovom slučaju ukoliko je broj jedinica 0, 2 ili 4, poslednji član niza P0 mora biti 0, dok ukoliko je broj jedinica jedan ili tri, poslednji član P0, mora biti 1. Ovo važi za svih 10 redova bitova parnosti. Pored ove provere, proveravaju se i parity bitovi kolone. Za razliku od redova, postoji samo jedna kolona bitova parnosti. Ova kolona predstavlja niz od P0 do P9.



Slika 5. Algoritam provere bitova parnosti.

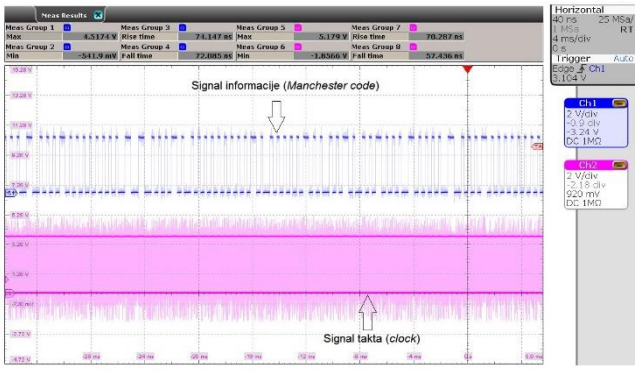
Posle svih gore navedenih uslova koje ispita mikrokontroler, na kraju uvek ispita da li je čitava 64-bitna reč uzorkovana. Cela reč biće uspešno primljena samo ukoliko apsolutno sve bude u redu sa svakim bitom koji je primljen. Ovo ne ostavlja prostora za grešku i očitavanje pogrešne vrednosti sa kartice. Ukoliko je cela reč primljena sa RFID kartice, sledi njeno slanje na računar.

U ovom delu koda proverava se prvo brojač „i". Ovaj brojač doći će do vrednosti 63 samo ukoliko je cela reč prihvaćena od strane mikrokontrolera, u suprotnom će biti vraćen na početnu vrednost 0 pre toga. Prva promenljiva koja se vraća na 0 je parity_niz.

VI. EKSPERIMENTALNO TESTIRANJE RADA ČITAČA

Nakon izbora komponenata za projektovanje uređaja, potrebno je neke od karakteristika komponenata i eksperimentalno potvrditi. Kako je glavna komponenta uređaja čitač za tagove, akcenat će biti na testiranju osobina ovog dela. Glavni merni instrumenti korišćeni za merenje karakteristika su osciloskop Rohde&Schwarz RTE 1034, lenjir i štoperica [4].

Na slici 6 prikazani su taktni signal i signal informacije snimljeni na osciloskopu.

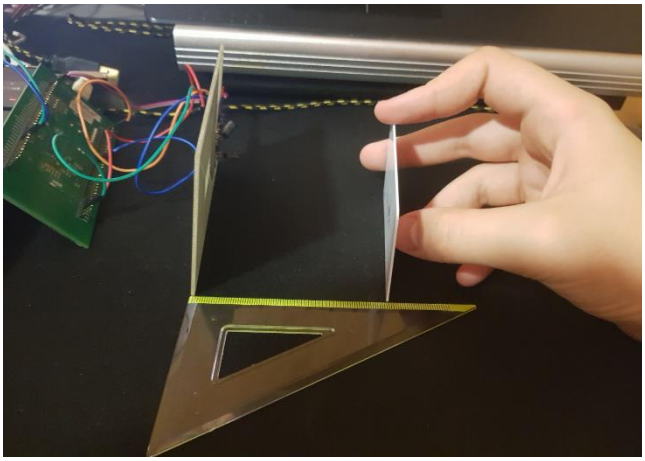


Slika 6. Taktni signal RDY/CLK i signal informacije

Eksperimentalnim merenjem ustanovljeno je da je taktni signal u skoro savršenom skladu sa signalom informacije. Promena u Mančester kodu uvek se dešava oko 32. *clock* signala, i ovo može biti poremećeno samo očitavanjem kartice sa maksimalne udaljenosti gde se faze taktnog i signala informacije malo pomere. Ovaj problem očitavanja kartice na određenu daljinu rešen je podešavanjem *flag* promenljivih u samom kodu čime je postignuta maksimalna fizički moguća udaljenost za očitavanje kartice. Ova udaljenost izmerena je školskim lenjirom.

Merenje je izvršeno po principu srednje vrednosti udaljenosti na kojoj se kartica prvi put očita. Merenja su varirala 1 mm pa možemo zaključiti da čitač radi stabilno i uvek na istoj razdaljini pročita karticu. Ova razdaljina iznosi 8,4 cm i veoma je zadovoljavajuća.

Brzina čitanja RFID kartice nije uspešno izmerena. Pokušaj je predstavljao merenje vremena od kada kartica uđe u polje dometa čitača do čitanja. Neuspeh je posledica ogromne brzine čitača koji očitava karticu za manje od 0,1 sekunde od prislanjanja kartice. Na slici 7 prikazan je način merenja maksimalne udaljenosti.



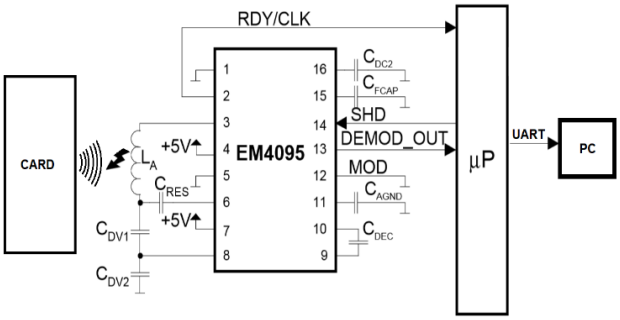
Slika 7. Merenje maksimalne udaljenosti kartice za očitavanja.

Pored merenja brzine i maksimalnog rastojanja kartice izmeren je i uticaj metala na komunikaciju RFID predajnika i modula. Parče metala debljine 1,4 mm stavljeno je ispred RFID kartice/predajnika, tako da bude na putu signala do modula. Pomoću merenja donet je zaključak da kartica ne može biti pročitana u ovom slučaju. Metal ne propušta ovu vrstu signala i modul ne dobija nikakvu informaciju na svojoj anteni.

VII. Praktična realizacija

Električna šema celog uređaja prikazana je na slici 8. Sve prateće komponente uz EM4095 su povezane linijama na samoj ploči. Na istoj ploči nalazi se i antena. Rastojanje između mikrokontrolera i EM4095 je malo.

Ukoliko bi bilo veće moglo bi biti problema u sinhronizaciji između taktnog signala i Mančester koda kao i pojava šumova. Ovaj problem doveo bi do pogrešnog očitavanja informacije sa kartice i samim tim loše realizovanog uređaja.



Slika 8. Električna šema uređaja.

VIII. Prototip uređaja

Prototip uređaja realizovan je povezivanjem komponenata na protoploči. Komponenta EM4095 iskorišćena je unutar ploče sa antenom i dodatnim potrebnim komponentama za njen rad. U prototipu korišćeno je razvojno okruženje STM32F072B koje je velikih dimenzija i u finalnom proizvodu ne bi bilo pogodno..



Slika 9. Prototip uređaja.

IX. Zaključak

Evidencija rada zaposlenih jedan je od najbitnijih zadataka u svakoj kompaniji pa čak i manjoj firmi. Dobra evidencija odstranjuje potencijialne malverzacije radnim vremenom i drži sve pod kontrolom. Pored radnog vremena RFID čitačem može se voditi evidencija o pauzama, broju slobodnih dana itd.

Literatura

[1] Fatah Chetouane ,,An Overview on RFID Technology Instruction and Application“ Faculty of Engineering, Université de Moncton, Moncton, E1A 3E9, New Brunswick Canada, 2015

[2] Communication by means of reflected power, Harry Stockman https://ieeexplore.ieee.org/document/1697527

[3] STM32F072B datasheet, https://www.st.com/resource/en/datasheet/stm32f072c8.pdf

[4] https://www.rohde-schwarz.com/vn/product/rte-products

Elektronski klavir realizovan pomoću integrisanog kola NE555

Stefan Krstić, Miloš Krstić, Miljana Milić

Sadržaj – U radu je opisan jedan od načina realizacije prenosnog elektronskog klavira korišćenjem tajmera NE555, njegov princip rada, kao i kriterijume koje treba ispuniti sa ciljem poboljšanja preciznosti. Elektronski klavir pokriva jednu oktavu i ima mogućnost štimovanja svih tonova. Izlaz tajmera se dovodi na audio utikač od 3,5 mm što omogućava njegovo priključivanje na različite uređaje.

I. Uvod

Elektronski klavir predstavlja instrument sa klavijaturama odnosno tasterima koji simulira klavir korišćenjem odgovarajućih analognih i digitalnih elektronskih komponenata. On generiše svoje tonove uz pomoć pogodno podešenih oscilatornih kola. Prvi elektronski piano se pojavio 1970-ih, i bio je napravljen u Italiji. Slični modeli su se kasnije pojavili i u Japanu [1].

Kao glavni nedostatak prvih modela ovakvog instrumenta bilo je to što su tonovi koji su generisani na ovaj način bili potpuno iste jačine, odnosno, jačina pritiskanja dirke nije uticala na to koliko se glasno čuje ton. Ova osobina elektronskog klavira, kod muzičara koji su koristili klavir ili orgulje, zadavala je velike probleme.

U narednom periodu pojavili su se i odmah postali popularni elektronske klavijature realizovane u digitalnoj tehnici kao i polifoni sintisajzeri zvuka, čime se stvorilo ogromno tržište ovakvih muzičkih instrumenata.

U ovom radu predstavljen je postupak projektovanja i realizacije jednog jednostavnog kola za sintezu zvukova određenih frekvencija realizovanog korišćenjem tajmera NE555. Ovo kolo predstavlja rezultat rada studenata na projektu iz predmeta „Projektovanje elektronskih kola“, na Elektronskom fakultetu u Nišu – u školskoj 2018/19. godini.

II. Princip rada tajmera i ideja

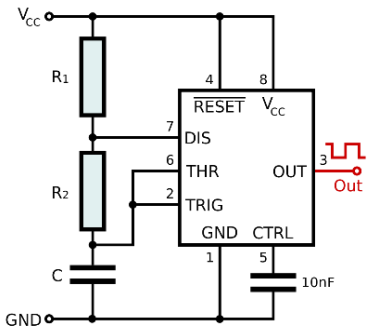
Kolo elektronskog klavira kao najvažniju komponentu korisati tajmer NE555. NE555 je bio jedan od prvih masovno pravljenih komercijalnih integrisanih tajmera koji je pušten u prodaju 1971. [2]. Ovaj tajmer važi za jedan od najkorisnijih i najkorišćenijih integrisanih kola ikada napravljenih.

Stefan Krstić, Miloš Krstić i Miljana Milić su sa Katedre za elektroniku, Elektronski fakultet ,Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: krstic.stefan@elfak.rs, krstic.milos@elfak.rs, miljana.milic@elfak.ni.ac.rs.

NE555 je bio i ostao “hit” ne samo zbog svoje niske cene već i zbog stabilnosti rada, maksimalne frekvencije od par stotina kiloherca već i po tome što zahteva malo dodatnih komponenti da bi se dobio funkcionalan uređaj. Sem primene na kojoj se bazira projekat, tajmer može koristiti kao kontoler impulsno širinske modulacije (PWM), brojač frekvencije, Šmitov komparator, generator kašnjenja, DC-DC konvertor, itd. Potreban napon napajanja je od 4.5 V do 15 V, a struje se kreću u opsegu od 3 mA do 6 mA. Rise/Fall vreme prilkom generisanja impulsa iznosi 100ns, temperaturna stabilnost je 0.005 % / °C. Ovo su bili argumenti za izbor ovog integrisanog kola pri realizaciji elektronskog klavira.

A. Astabilni režim

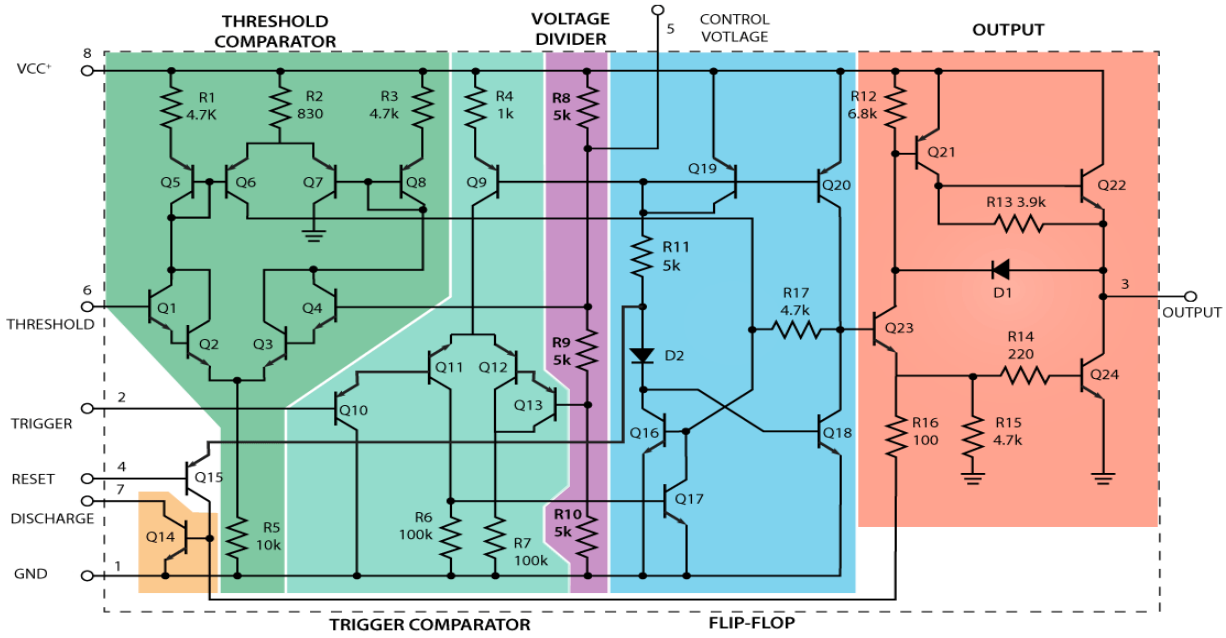
Kolo elektronskog klavira funkcioniše na sledeći način. U astabilnom režimu, prikazanom na slici 1, kolo vrši funkciju oscilatora gde je na izlaznom pinu tajmera pravougaoni impuls čija frekvencija i faktor ispune (*duty cycle*) zavise od vrednosti otpornika R_1 i R_2, kao i kondenzatora C.



Slika 1. Astabilni režim tajmera NE555.

Za bolje razumevanje astabilnog režima, na slici 2 je prikazana interna blok šema tajmera. Tajmer se sastoji iz pet celina: dva komparatora, razdelnik napona koji daje referentne vrednosti napona na njihove ulaze, flip-flop, tranzistor za pražnjenje i izlazni stepen.

Po povezivanju izvora napajanja kondenzator ne sadrži naelektrisanje, pa je napon na pinu 2 niži od 1/3 napona napajanja koji je na drugom ulazu komparatora što dovodi da se tranzistori Q_{10} i Q_{11} uključe. Sa padom napona na otporniku R_6 i baznom strujom, pali se Q_{17}, Q_{16} se isključuje, a Q_{18} prelazi u zasisećenje. Takođe, sa uključenim Q_{18}, baza



Slika 2. Interna šema tajmera NE555.

tranzistora Q_{23} je efektivno na masi pa je i tranzistor Q_{24} isključen. Izlaz je na visokom nivou. Treba zapaziti da je Q_{14} takođe isključen pa se kondenzator C puni preko otpornika R_1 i R_2.

U trenutku kada na pinu 6 napon iznosi 2/3 napona napajanja Q_1 i Q_2 kreću da vode, a sa njima i Q_5 i Q_6. Zbog asimetrije strujnog ogledala koja čine ova dva tranzistora, Q_6 će provesti veću struju, dovoljnu za uključivanje tranzistora Q_{16}. Prethodni isključuje tranzistor Q_{18} što za posledicu ima uključivanje Q_{23}, odnosno Q_{24}, šaljući izlaz tajmera u stanje niskog napona. Ovde treba zapaziti da je sa uključenim Q_{23} uključen i Q_{14}, što pin 7 vezuje za masu. Ovim kondenzator kreće da se prazni preko otpornika R_2 sve dok napon na njemu ne padne ispod 1/3 napona napajanja, kada ciklus kreće da se ponavlja.

S obzirom da su RC konstanta i napon na pinu 6 direktno proporcionalni naponu napajanja, na frekvenciju izlaznog signala ne utiče vrednost jednosmerne komponente napona napajanja. Sa druge strane, u slučaju superponirane naizmenične komponente, prag napona na kondenzatoru potreban za promenu stanja na izlazu će varirati i time unositi grešku. Korišćenjem baterije kao izvora napajanja se izbegava potreba za stepenom za filtriranje ali i poboljšava prenosivost uređaja.

B. Model elektronskog klavira

Kako je frekvencija impulsa data formulom:

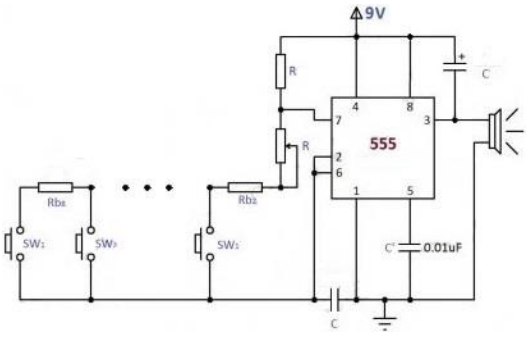
$$f = \frac{1}{\ln(2) \times C \times (R_1 + 2R_2)} \qquad (1)$$

može se doći do zaključka da se sa promenom vrednosti pasivnih elemenata menja i frekvencija izlaznog signala odnosno tona. Zbog zahteva da frekvencija pojedinačnih nota bude podesiva, visoke cene varijabilnih kondenzatora, kao i većeg faktora promene od R_1, za promenljivu je odabran otpornik R_2. Što je vrednost otpornika R_2 manja, to frekvencija signala na izlazu tajmera veća, i obratno.

Saglasno sa zahtevom da se pokrije jedna puna tonska oktava, uključujući osnovne (C, D, E, F, G, A, H, C) i povišene tonove (Cis, Dis, Fis, Gis, B), bilo je potrebno da se ova izračunavanja ponove za svaki od 13 tonova.

Ovim se dolazi do osnove šeme prikazane na slici 3. Tasterima se zatvaraju odgovarajuća oscilatorna kola i generiše zvuk odgovarajuće frekvencije.

Generisani zvuk moguće je vezati za zvučnik, slušalice ili proslediti do računara.



Slika 3. Osnovno kolo elektronskog klavira.

III. ELEKTRIČNA ŠEMA I IZRADA

Kao što je već napomenuto, za pokrivanje jedne oktave potrebno je trinaest dirki (od toga pet povisilica),

odnosno trinaest tastera kojima se realizuje njihova funkcija. Vrednosti R_2 za date frekvencije su prikazane u tabeli 1 [3], a za odabrane vrednosti R_1 i C od 1 kΩ i 10 nF, respektivno. Treba imati na umu da nije dobro izabrati kondenzator visoke vrednosti kapacitivnosti – prvenstveno zbog visoke struje curenja i većeg opsega tolerancije i osetljivosti. Ukoliko je neophodno, tantalovi kondenzatori su daleko bolja opcija od elektrolitskih. Takođe, ukoliko je tajmer rađen u bipolarnoj tehnologiji, za aktiviranje Darlingtonovog diferencijalnog para (komparatora) neophodna je minimalna bazna struja. Inherentna dizajnu tajmera, ona iznosi 0.25 μA. Otpornost između napona napajanja i šestog pina mora da propusti struju veću od ove. Pošto je u astabilnom režimu šesti pin kratkospojen sa drugim, sledi da je maksimalna vrednost napona na njemu 2/3, odnosno 1/3 je na otpornicima. Praktično, za bateriju od 9 V, maksimalna vrednost otpornika je 12 MΩ [4].

TABELA I

Nota	f [Hz]	R_2_eq [Ω]	Povećanje R_2 u odnosu na prethodni ton [Ω]
C'	512	13589	/
B	480.00	14528	626/
A♯	460.80	15154	1252
A	426.67	16406	705
G♯	409.60	17111	1174
G	384.00	18285	1027
G♭	364.09	19312	225
F♯	360.00	19537	216
F	341.33	20633	1096
E	320.00	22042	1409
D♯	307.20	22981	939
D	284.44	24860	1879
C♯	273.07	25916	1056
C	256.00	27678	1762

Prilikom realizacije kola, nametnulo se jedno ograničenje. Naime, kolo elektronskog klavira trebalo je izraditi u SMD tehnici štampanih kola. Ova tehnika je izabrana sa ciljem da se studenti upoznaju sa radom dva uređaja za fabrikovanje malih serija SMD štampanih ploča i to: uređajem za montiranje ovakvih komponenti – Charmhigh CHM-T48VA Pick-and-Place Machine [5], i uređajem za lemljenje raspoređenih SMD komponenti na štampanoj pločici – Puhui T962A Infrared IC heater [6]. Za rad prve CNC mašine potrebno je sve SMD komponente složiti na odgovarajuće koture. Za to su izabrane dve vrednosti otpornosti (100 Ω i 1 kΩ), dok su sve ostale otpornosti dobijane kombinacijom ovih dveju vrednosti.

Zbog ograničenja od dve raspoložive vrednosti otpornika, broj komponenti u kolu bi se povećao na oko osamdeset. Za dobijanje svih potrebnih frekvencija, na pločicu su dodati TH potenciometri čija je namena fino štimovanje tonova.



a)

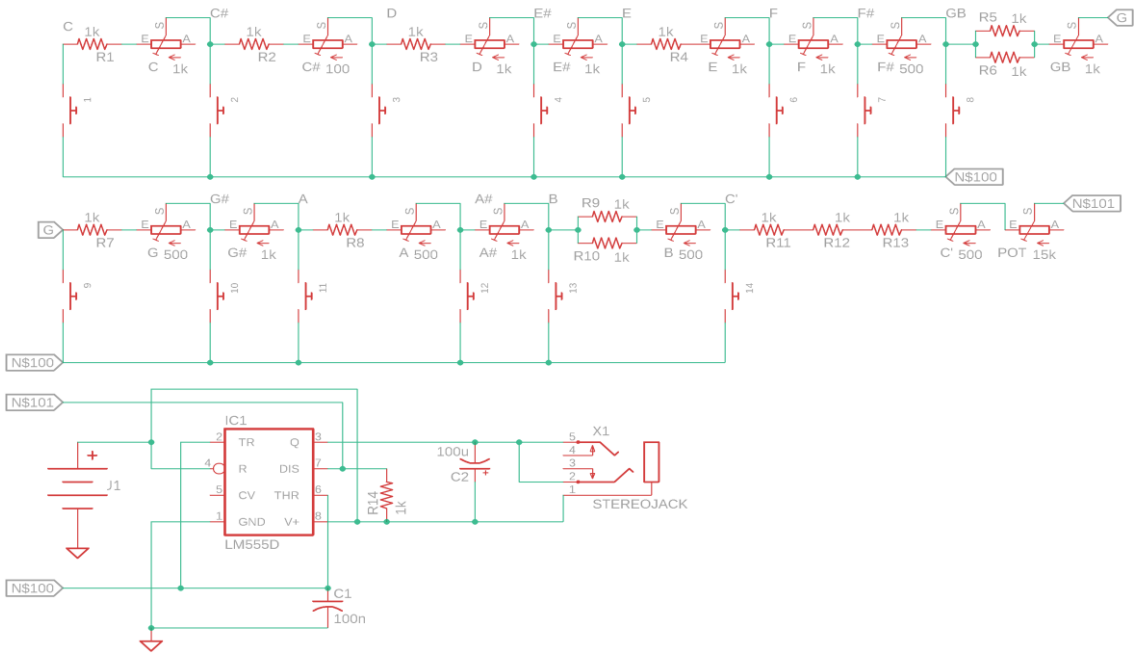


b)

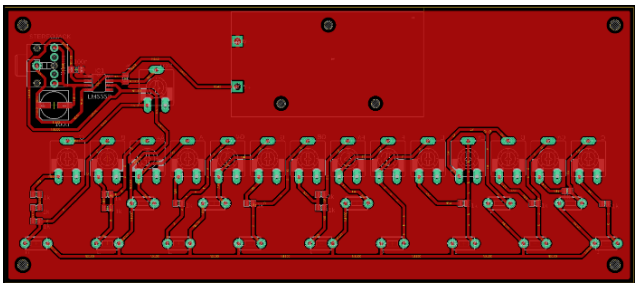
Slika 4. CNC mašine za a) razmeštaj i b) lemljenje SMD komponenata na štampanoj ploči.

Za projektovanje štampane ploče je korišćen softverski alat Eagle [7]. Električna šema urađena u Eagle okruženju je prikazana na slici 5, a dizajnirana štampane ploče na slici 6. Odabrano je da debljina signalnih linija bude 32 mils-a sa minimalnim rastojanjem između njih od 15 mils-a. Pri razmeštaju komponenata, vodilo se računa o tome da razmak između tastera bude srazmeran debljini prstiju, kao i da se pozicioniranje pokretnog kraja potenciometra u krajnji levi položaj dobija maksimalna frekvencija izlaznog signala/tona. Takođe, predviđena je i mogućnost ugrađivanja odstojnika sa ciljem da se sve komponente sem tastera mogu postaviti sa donje strane ploče.

Na linku u referenci [8] se može pogledati kako funkcioniše realizovao kolo.

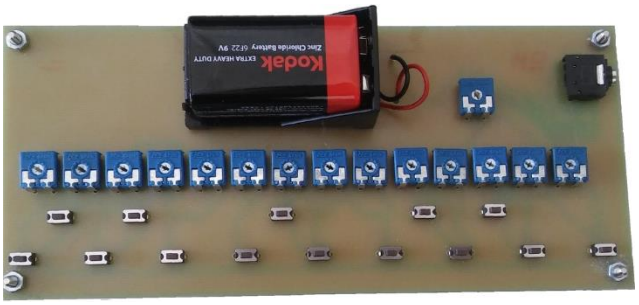


Slika 5. Električna šema elektronskog klavira.



Slika 6. Dizajnirana štampana ploča iz Eagle programskog paketa.

Realizovano kolo elektronskog klavira nakon izrade štampane pločice i montiranja svih komponenata prikazano je na slici 7.



Slika 7. Finalni izgled elektronskog klavira.

IV. Zaključak

Realizacijom elektronskog klavira kao studentskog projekta uspešno je ostvareno nekoliko ciljeva. Studenti su se upoznali sa radom i svojstvima integrisanog kola NE555. Proučili su mogućnosti korišćenja programa za simulaciju i projektovanje štampanih elektronskih ploča iz programskog paketa Eagle. Ovladali su korišćenjem dva uređaja i to: CNC mašinom za punjenje SMD komponenti na štampanoj pločici, i CNC mašinom za lemljenje SMD komponenti na štampanoj pločici.

Imajući sve to u vidu, može se zaključiti da je time kompletno teoretski i praktično pokriven postupak projektovanja i proizvodnje elektronskih štampanih kola na predmetu Projektovanje elektronskih kola sa modula za Elektroniku, Elektronskog fakulteta, Univerziteta u Nišu.

Zahvalnica

Rezultati prikazani u ovom radu ostvareni su u okviru projekta „Elektronski fakultet – Projektovanje, verifikacija i testiranje elektronskih kola EF-PVT EK, Broj ugovora: 451-02002644/2018-06. čiju realizaciju finansira Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije.

Literatura

[1] https://www.automatika.rs/baza-znanja/obrada-signala/ne-555-timer.html
[2] https://en.wikipedia.org/wiki/555_timer_IC
[3] https://en.wikipedia.org/wiki/Piano_key_frequencies
[4] http://www.555-timer-circuits.com/datasheets/555an.pdf
[5] http://www.charmhigh-tech.com
[6] http://www.puhuit.com/main/page_products_t962a_ir_ovenic_heater.html
[7] https://www.autodesk.com/products/eagle/overview
[8] https://youtu.be/S_OaDetIc_Q

Realizacija kalkulatora pomoću Arduina

Aleksandar Denić, Miloš Ristić, Uroš Zdravković, Stefan Stojčić, Stefan Tošić, Filip Nikolić, Nikola Ćirić, Miloš Marjanović

Sadržaj – U ovom radu opisana je realizacija prototipa kalkulatora pomoću Arduino razvojnog okruženja. Kalkulator se sastoji iz tastature realizovane pomoću diskretnih tastera, LCD displeja i Arduino mikrokontrolera. Uređaj može da radi sa celim brojevima i nad njima obavlja osnovne aritmetičke operacije: sabiranje, oduzimanje, množenje i deljenje. U radu su opisani hardverski i softverski deo projekta i dati predlozi za unapređnje prototipa.

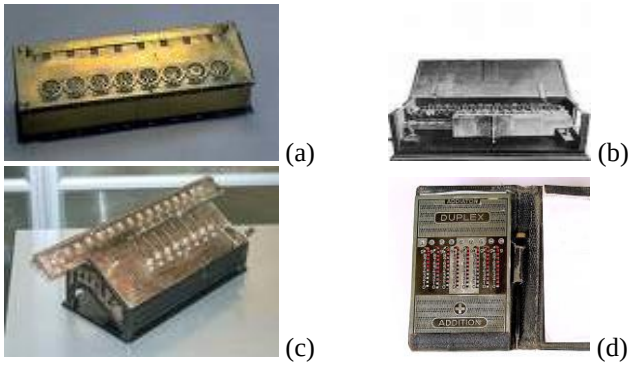
I. Uvod

Sve do 17. veka računanje se obavljalo ručno, dok Nemac Wilhelm Schickard nije konstruisao prvi mehanički kalkulator koji je mogao da sabira i oduzima šestocifrene brojeve [1]. Blez Paskal je 1642. godine otpočeo rad na mehaničkom kalkulatoru "*Pascaline*" (slika 1a), završio ga je tri godine kasnije, da bi 1652. godine bilo napravljeno 50 prototipova, od kojih je 12 prodato. Rad ovog kalkulatora zasnivao se na zupčanicima i mogao je da računa sa decimalnim vrednostima, ali rad sa negativnim brojevima nije bio moguć. Nekoliko godina kasnije, 1671. godine, Lajbnic je izumeo računaljku sa bubnjem (slika 1b) koja je mogla da obavlja sve četiri osnovne aritmetičke operacije, kao i da izračunava kvadratni koren, a mana je bila da operacija prenosa prilikom računanja nije bila automatizovana. Ovaj izum uveo je ovog naučnika u Londonsko kraljevsko društvo. Prvi uspešni, komercijalni kalkulator kreirao je Šarl Savijer Tomas de Kolamr 1820. godine pod nazivom "aritmometar" (slika 1c), koji je mogao da sabira, oduzima i množi, a delio je uz intervenciju korisnika. Ove sprave zauzimale su ceo sto i bile su u širokoj upotrebi i početkom 20. veka. Kalkulatori kakve danas poznajemo pojavili su se 70-tih godina prošlog veka (slika 1d), pri čemu su prvi proizvođači bili *HP*, *Motorola*, *Texas Instruments*. Digitron je u srpskom jeziku odomaćena reč za kalkulator, po istoimenoj firmi iz Istre u Hrvatskoj, koja ga je prva sklapala u SFRJ. Današnji kalkulatori uglavnom su zasnovani na štampanoj ploči sa integrisanim kolima koja obavljaju izračuvanja, tastaturi i displeju, pri čemu je napajanje najčešće baterijsko.

U ovom radu biće predstavljen prototip kalkulatora realizovan korišćenjem Arduino razvojnog okruženja. Kalkulator može da obavlja osnovne aritmetičke operacije nad celim brojevima i ima predviđen restart. Postoji veliki broj realizacija kalkulatora pomoću Arduina [2-4], ali svi koriste komercijalni modul tastature. U ovom projektu tastatura kalkulatora realizovana je korišćenjem diskretnih tastera i otpornika u kolu razdelnika napona. Biće opisani hardverski deo projekta, kao i softverski algoritmi. Na kraju rada prikazan je rezultat testiranja prototipa i dati su predlozi za unapređenje funkcionalnosti uređaja.

Aleksandar Denić, Miloš Ristić, Uroš Zdravković, Stefan Stojčić, Stefan Tošić, Filip Nikolić, Nikola Ćirić su učenici ETŠ "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Srbija, E-mail: adenic079@gmail.com

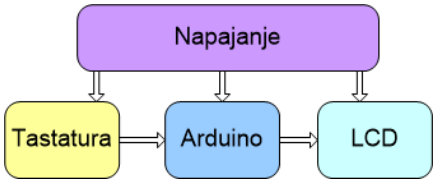
Miloš Marjanović je asistent na Katedri za mikroelektroniku, Elektronski fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milos.marjanovic@elfak.ni.ac.rs



Slika 1. (a) Mehanički kalkulator "*Pascaline*", (b) Računaljka s bubnjem – "*Stepped Reckoner*", (c) Aritmometar, (d) "*Addiator Duplex*" kalkulator, model koji pripada zbirci Muzeja nauke i tehnike Beograd.

II. Realizacija uređaja

Blok šema uređaja prikazana je na slici 2. Napajanje prototipa uređaja: mikrokontrolera, bloka tastature i LCD (*Liquid Crystal Display*) displeja izvedeno je sa Arduino razvojnog okruženja. Podaci se unose preko tastature realizovane od diskretnih tastera. Arduino mikrokontroler preuzima podatke sa tastature, vrši izračuvanja i kontrolu LCD displeja na kome se prikazuju rezultati.



Slika 2. Blok šema kalkulatora.

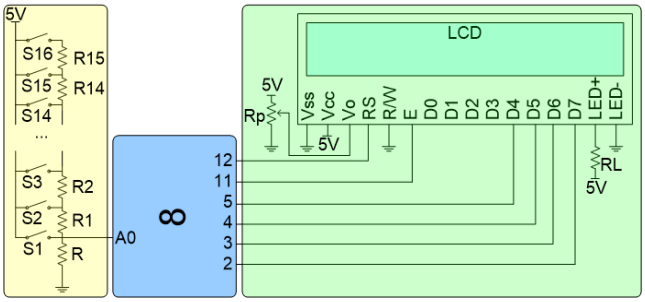
Korisnik unosi prvi željeni ceo broj, operaciju i drugi željeni ceo broj. Klikom na taster "=" mikrokontroler izračunava zadatu aritmetičku operaciju nad unetim ciframa i prikazuje rezultat na LCD displeju. Unete cifre i

operacija prikazuju se na displeju u realnom vremenu. Nakon prikazivanja rezultata izračunavanja, korisnik resetuje kalkulator klikom na taster "C" i ponovo može izvršiti novo izračunavanje.

A. Hardverska realizacija kalkulatora

Kalkulator se napaja sa 5 V DC. Napajanje Arduino razvojnog okruženja izvedeno je preko USB kabla sa laptopa. Sa ovog razvojnog mikrokontrolerskog okruženja, napajanje je iskorišćeno za ostale delove uređaja. Arduino okruženje, a samim tim i prototip uređaja se može napajati eksterno sa 12 V DC koji se dovode preko DC konektora, dok na štampanoj ploči okruženja postoji regulator napona.

Električna šema uređaja prikazana je na slici 3. Blok tastature i LCD displeja realizovan je na protoploči, dok su blokovi napajanja i mikrokontrolera na Arduino razvojnom okruženju.



Slika 3. Električna šema kalkulatora.

Tastaturu čini 16 tastera povezanih jednim krajem na napon napajanja 5 V i drugim na otpornu mrežu (*resistor ladder*) [5]. Upotrebljeni su otpornici iste vrednosti otpornosti $R=R1=\ldots=R15=10$ kΩ. Otpornik R ima ulogu *pull-down* otpornika. Deset tastera koristi se za unos cifara od 0 do 9, četiri tastera koriste se za unos operacije (+, -, *, /), i po jedan taster za "=" i reset kalkulatora. Koji taster je pritisnut očitava se preko samo jednog analognog pina mikrokontrolera (A0). Naime, prilikom pritiskanja različitih tastera, različite vrednosti napona biće očitane na pinu A0. Ako ni jedan od tastera nije pritisnut, vrednost na pinu A0 biće jednak 0 V. Ako je pritisnut taster S1, na pinu A0 pojaviće se napon napajanja. Pritiskom tastera S2, formira se razdelnik napona koga čine otpornici R i R1, dok su ostali otpornici u mreži otkačeni, pa je napon na A0:

$$V_{A0(S2)} = \frac{R}{R+R1} 5\,V = 2.5\ \mathrm{V}\,. \qquad (1)$$

Slično, ako je pritisnut taster S3, napon na pinu A0 određen je naponskim razdelnikom koga čine otpornik R i redna veza otpornika R1 i R2. Konačno, ako je pritisnut taster S16, napon je određen razdelnikom napona koga čine R i redna veza svih otpornika od R1 do R15. U opštem slučaju, napon koji se dobija sa tastature može se izraziti kao:

$$V_{A0(Sn)} = \frac{R}{R+\sum_{i=1}^{n-1} R_i} 5\,. \qquad (2)$$

gde je n – redni broj tastera u tastaturi koji je pritisnut. U tabeli I date su vrednosti napona koje se očitavaju za svaki od tastera u tastaturi kalkulatora.

TABELA I
RASPORED TASTERA I OČITANIH VREDNOSTI TASTATURE

Taster	Značenje	Očitani napon (V)	Rezultat AD konverzije
S1	0	5.0	1023
S2	1	2.5	508
S3	2	1.67	339
S4	3	1.25	253
S5	4	1.0	202
S6	5	0.83	168
S7	6	0.71	144
S8	7	0.625	126
S9	8	0.56	111
S10	9	0.5	100
S11	+	0.454	91
S12	-	0.417	83
S13	*	0.385	76
S14	/	0.357	71
S15	=	0.33	66
S16	C	0.3125	62

Arduino Uno razvojno okruženje korišćeno u ovom radu ima ATMega328P mikrokontroler kao glavnu komponentu [6]. Ostale komponente (oscilatorno kolo, regulacija napona, komunikacija) omogućavaju rad ovog mikrokontrolera. Ovaj mikrokontroler ima 6 analognih pinova, od kojih je jedan iskorišćen kao ulazni pin (A0) preko koga se očitava tastatura, 14 digitalnih pinova od kojih je 6 iskorišćeno za povezivanje sa LCD-om, kao i set pinova za napajanje i programiranje (ICSP – *In Circuit Serial Programming*).

LCD displej se koristi za prikaz alfanumeričkih karaktera, iskorišćen je displej koji može da prikaže 32 karaktera, po 16 karaktera u svakom od dva reda [5]. Pinovi VCC i VSS se koriste za napajanje LCD-a, dok se kontrast displeja podešava pomoću potenciometra, čiji je klizač povezan na pin Vo. Iskorišćen je trimer potenciometar ukupne otpornosti R_P=10 kΩ. Dozvola rada LCD-a kontroliše se preko pina E (*Enable*), koji je povezan na digitalni pin 11 mikrokontrolera. RS (*Register Select*) pin kontroliše gde će se karakteri pojaviti na ekranu i povezan je na digitalni pin 12 mikrokontrolera. R/W (*Read/Write*) pin služi za podešavanje moda ekrana, da li će se čitati vrednost sa ekrana ili će biti korišćen za

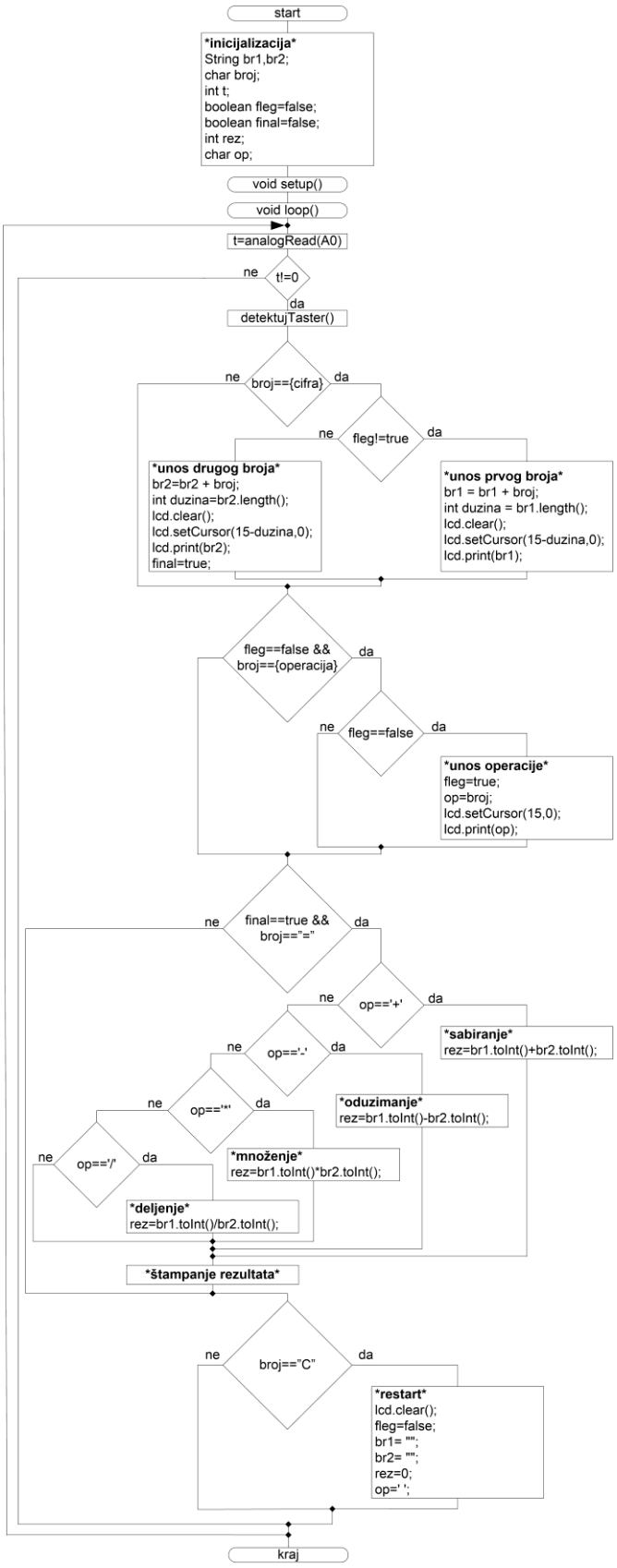
ispisivanje karaktera. Korišćeni LCD je u *Write* modu, te je R/W pin povezan na masu. Pinovi D0-D7 koriste se za slanje podataka koji će biti ispisani na ekranu. U ovom slučaju se koristi 4-bitni prenos podataka na pinove D4-D7 sa pinova 5 do 2 mikrokontrolera, respektivno. Pinovi LED+ i LED- koriste se za podešavanje pozadinskog LED osvetljenja LCD ekrana. Iskorišćen je otpornik otpornosti R_L=220 Ω, sa ciljem da ograniči struju kroz LED-ove.

B. Softverska realizacija kalkulatora

Softver mikrokontrolera pisan je u besplatnom okruženju Arduino IDE [6]. U program je uključena biblioteka za rad sa LCD-om `<LiquidCrystal.h>` i kreiran je objekat pod nazivom "`lcd`", te definisano na koje je pinove mikrokontrolera displej povezan. U okviru funkcije `void setup()`, koja se izvršava samo jednom, nakon priključenja mikrokontrolera na napajanje, definisan je tip displeja (16x2) i iskorišćena je funkcija `lcd.clear();` tako da je nakon pokretanja kalkulatora ekran prazan. Algoritam toka programa mikrokontrolera sa funkcijom `void loop()` koja se izvršava sve dok mikrokontroler ima napajanje, tj. ne bude hardverski resetovan, prikazan je na slici 4.

Očitava se vrednost napona na pinu A0. Vrednost koja se upisuje u promenljivu posle analogno digitalne konverzije je broj između 0 i 1023, srazmerno sa naponom na A0, koji je između 0 V i 5 V. Ako je neki od tastera pritisnut, poziva se funkcija koja detektuje koji je taster pritisnut. Kao rezultat izvršenja te funkcije u odgovarajuću promenljivu tipa char upisuje se alfanumerički karakter u zavisnosti od toga koji je taster pritisnut. Algoritam toka dela funkcije za detektovanje tastera prikazan je na slici 5.

Nakon izvršenja ove funkcije u glavnom programu ispituje se da li pritisnut taster za neku od cifara, operacije, znak "=" ili reset kalkulatora. Ako je pritisnut taster sa nekom cifrom u zavisnosti da li se unosi prvi ili drugi broj nad kojim se izvršava neka od operacija, menja se sadržaj promenljive tipa string (`br1` ili `br2`) u skladu sa unetom cifrom, određuje se trenutna dužina stringa i na ekranu se štampa uneti broj, prikazivanjem vrednosti stringa. Pri tome, broj se ispisuje u prvom redu displeja, od 15. pozicije ka prvoj u zavisnosti od dužine stringa, dok je 16. pozicija rezervisana za prikaz unete operacije. Ako je unet i drugi broj postavlja se vrednost logičke promenljive `final` (tipa *boolean*) na vrednost `true`, da se signalizira izvršenje određenih operacija nad unetim brojevima. S druge strane, ako je detektovano da je uneta neka od operacija i da prethodno nije uneta operacija (što se signalizira kroz logičku promenljivu `fleg`), u promenljivu `op` tipa char čuva se koja je operacija izabrana, štampa se oznaka operacije na ekranu u prvom redu na 16. poziciji i setuje se vrednost pomoćne promenljive `fleg` na `true`, koja signalizira da je korisnik izabrao operaciju koju želi da izvrši nad unetim brojevima. Inače ako je detektovano da je pomoćna promenljiva `final` setovana na `true` vrednost i

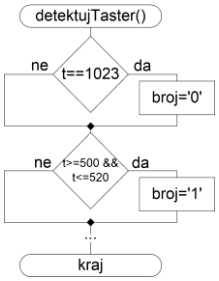


Slika 4. Algoritam toka programa mikrokontrolera.

da je korisnik uneo znak "=", tada se ispituje koja je operacija izabrana. Ako je izabrana operacija sabiranja, na primer, tada se u promenljivu `rez` upisuje zbir unetih brojeva, pri čemu se promenljive tipa string konvertuju u promenljive tipa integer, korišćenjem funkcije:

```
rez=br1.toInt()+br2.toInt();
```

Po istom principu izračunavaju se operacije oduzimanja, množenja i deljenja. Nakon izračunavanja rezultata, isti se štampa na ekranu pri čemu se koristi funkcija `lcd.autoscroll();` za pravilan ispis rezultata na ekranu. Na kraju, inače ako je korisnik pritisnuo dugme za reset, vrši se brisanje ekrana, brisanje promenljivih tipa string koje čuvaju vrednosti unetih brojeva i operacije, resetuju se vrednost promenljive koja čuva rezultat na nulu i vrednosti odgovarajućih pomoćnih logičkih promenljivih na `false`. Nakon svih ovih provera, program se vraća na početak – očitavanje tastera i tako dok je Arduino mikrokontroler pod napajanjem.



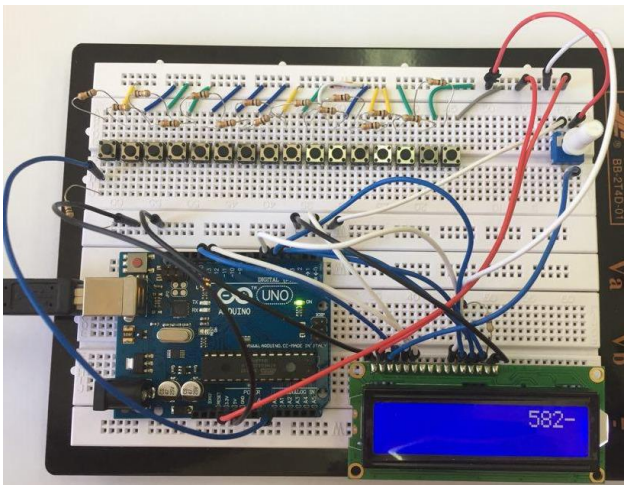
Slika 5. Algoritam dela funkcije za detektovanje tastera.

III. TESTIRANJE UREĐAJA

Projektovani hardver realizovan je na protoploči (tasteri sa otpornom mrežom i LCD sa pratećim komponentama) uz korišćenje Arduino Uno razvojnog okruženja. Softver je kompajliran i uploadovan na mikrokontroler korišćenjem Arduino IDE. Za napajanje razvojnog okruženja korišćen je USB kabl povezan sa laptopom. Tokom realizacije uređaja, korišćena je serijska komunikacija (*Serial Monitor*) prilikom debagovanja softvera. Na slici 6 prikazan je prototip projektovanog kalkulatora. Jedan od nedostataka realizovanog prototipa je što obavlja operacije samo sa celim brojevima, što je moguće unaprediti u narednim verzijama. Drugi uočeni nedostatak je što prilikom deljenja sa 0, umesto greške na ekranu se ispisuje "-1", te će i ovaj nedostatak biti otklonjen u budućem radu.

IV. ZAKLJUČAK

U ovom radu opisan je hardver i softver prototipa jednostavnog kalkulatora, koga čine tastatura realizovana od diskretnih tastera, Arduino mikrokontroler i LCD. Uređaj može da izvodi osnovne aritmetičke operacije nad celim brojevima. Za ovaj prototip moguće je projektovati štampanu ploču, pri čemu se ne bi koristilo razvojno okruženje već bi sekcija napajanja i mikrokontrolera bile realizovane na PCB-u. Hardver je moguće unaprediti dodavanjem većeg broja tastera u tastaturi sa ciljem povećanja funkcionalnosti kalkulatora. Pored uočenih i opisanih nedostataka, softver je moguće unaprediti dodavanjem mogućnosti za brisanjem samo poslednjeg koraka ili dodatnih operacija nad više realnih brojeva.



Slika 6. Realizacija kalkulatora na protoploči.

ZAHVALNICA

Ovaj rad deo je projekta "Idemo i mi na IEEESTEC konferenciju" koji se finansira sredstvima Centra za promociju nauke (CPN). Posebnu zahvalnost autori duguju prof. dr Danijelu Dankoviću, vanrednom profesoru na Katedri za mikroelektroniku, Elektronskog fakulteta u Nišu na nesebičnoj pomoći i zalaganju da se ovaj projekat realizuje. Rad je podržan i od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026.

LITERATURA

[1] Automatizacija računanja, istorijat. [On Line] Dostupno na: http://poincare.matf.bg.ac.rs/~cvetana/Istorijat/racunaljke/racunaljke.html
[2] Arduino Calculator using 4x4 Keypad. [On Line] Dostupno na: https://circuitdigest.com/microcontroller-projects/arduino-calculator-using-4x4-keypad
[3] Learn how you can create your own calculator using an Ardunio UNO, a 4x4 keypad, and an LCD display. [On Line] Dostupno na: https://maker.pro/arduino/projects/how-to-create-an-arduino-controlled-calculator
[4] Simple UNO calculator. [On Line] Dostupno na: https://create.arduino.cc/projecthub/Joprp05/simple-uno-calculator-01e6fe
[5] Scott Fitzgerald, Michael Shiloh, Tom Igoe, The Arduino Projects Book, 2012, pp. 79, 115-121.
[6] Arduino. [On Line] Dostupno na: https://www.arduino.cc

Projektovanje i implementacija SMTP protokola sa dokazom uloženog rada klijenta

Nađa Gavrilović

Sadržaj - Kao najzastupljeniji način komunikacije korisnika, elektronska pošta podrazumeva svakodnevno slanje velikog broja imejl poruka. Veliki deo ukupnog saobraćaja čini spam, koji predstavlja zloupotrebu elektronskih sistema u svrhu masovnog slanja neželjenih poruka. Cilj ovog rada je projektovanje i implementacija anti-spam rešenja koje koristi Proof of work sistem ulaganja određene količine rada uređaja pre prosleđivanja imejl poruke. U radu će detaljno biti prikazana implementacija SMTP protokola sa proširenjem koje omogućava ispitivanje kredibiliteta klijenta Proof of work algoritmom, u cilju redukcije spam saobraćaja na mreži. Biće prikazana evaluacija uloženog rada klijenta za različite parametre, kao i rezultati testiranja uticaja uloženog rada na redukciju broja spam poruka na mreži.

I. UVOD

Elektronska pošta je mrežni servis koji omogućava razmenu poruka različitog sadržaja preko elektronskih uređaja. Osnovni protokoli koji definišu pravila razmene elektronske pošte su SMTP (eng. *Simple Mail Transfer Protocol*), POP (eng. *Post Office Protocol*) i IMAP (eng. *Internet Mail Access Protocol*). SMTP protokol omogućava slanje i prenos imejl poruke kroz mrežu i definiše pravila komunikacije uređaja na mreži koji omogućavaju samo rutiranje poruke, dok se za preuzimanje pošte na strani klijenta koriste protokoli POP i IMAP [1].

Istraživanja iz 2018. godine pokazuju da je broj poslatih/primljenih imejl poruka za jedan dan čak 280 milijardi. Približno 50% ukupnog saobraćaja čini spam [2]. Spam predstavlja zloupotrebu elektronskih sistema u svrhu slanja masovnih neželjenih poruka. Najčešće su to poruke reklamnog sadržaja, koje pak mogu sadržati viruse, različite napade na računar korisnika, pokušaje krađe podataka i sl. [3]. Osnovna karakteristika spama koja ga čini lako upotrebljivim i veoma rasprostranjenim je da pošiljaocu slanje jedne imejl poruke ne predstavlja značajan gubitak, ni vremenski, ni finansijski. Zato je slanje ogromnog broja poruka te vrste skoro potpuno besplatno.

POW sistemi - sistemi sa dokazom uloženog rada, uvode koncept ulaganja određene količine rada uređaja pre pristupa deljenom resursu, u cilju sprečavanja njegove zloupotrebe. U pogledu spama to podrazumeva ulaganje procesorskog vremena pre slanja svake poruke. Utrošak vremena klijentskog procesora se osigurava izdavanjem zadatka koji SMTP klijent mora tačno da reši, da bi njegova poruka bila prosleđena [4].

Nađa Gavrilović, student master studija na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva14, 18000 Niš, Srbija, e-mail: nadja.gavrilovic@elfak.rs

Važna karakteristika POW sistema je da slanje uobičajene količine poruka za pošiljaoca ne predstavlja značajan utrošak vremena, ali ako se radi o velikom broju poruka, što ukazuje na spam, njihovo slanje biva otežano.

Cilj ovog rada je projektovanje i implementacija proširenja SMTP protokola, koje omogućava ispitivanje kredibiliteta SMTP klijenta POW algoritmom. SMTP protokol će biti implementiran sa određenim izmenama u komunikaciji klijenta i servera, koje njegove postojeće funkcionalnosti nadograđuju mogućnošću zahtevanja dokaza uloženog rada klijenta, u cilju onemogućavanja neželjenog imejl saobraćaja na mreži.

Inicijalnu ideju ulaganja rada prilikom pristupa željenom resursu, prvi put su predstavili autori C. Dward i M. Naor 1992. godine [4]. Rad *Proof-of-work proves not to work* [5] iz 2004. godine, izlaže teoriju da Proof of work (u daljem tekstu POW) sistem, sam po sebi ne može biti od koristi u borbi protiv spama, zbog uticaja na validne korisnike elektronske pošte. Kao odgovor na zapažanja istaknuta u prethodno pomenutom radu, 2006. godine objavljen je rad *Proof of work can work* [6], koji izlaže ideju o upotrebi reputacionih sistema unutar POW sistema. Nakon razmatranja ideje i koncepta POW sistema, objavljeni su različiti radovi koji proučavaju implementaciju ovog algoritma u raznovrsnim sistemima. Zajednička osobina analiziranih sistema je zaštita pristupa deljivim resursima. POW koncept je značajno dobio na popularnosti kada je 2008. godine predstavljen koncept kriptovalute Bitcoin, u čijoj se pozadini koristi POW sistem. Jedan od radova koji je predložio upotrebu POW koncepta u anti-spam rešenjima je *Design and Analysis of an Improved BitMessage Anti-spam Mechanism* iz 2015. godine [7], koji proučava upotrebu POW sistema u borbi protiv spama na P2P mreži i vrši njegovu evaluaciju. Sistem opisan u radu [7] se razlikuje od onog koji je implementiran i predstavljen u nastavku ovog rada po načinu određivanja količine rada koju klijent treba da uloži, u radu [7] rad zavisi od same poruke, dok u sistemu koji će biti predstavljen u nastavku zavisi od odluke SMTP servera. Za razliku od P2P sistema, anti-spam rešenje projektovano i implementirano kao deo ovog rada se zasniva na klijent-server komunikaciji.

Nakon uvoda, u drugom poglavlju rada biće objašnjen princip rada mejl servisa i SMTP protokola. Treće poglavlje biće posvećeno POW sistemu. Četvrto poglavlje predstaviće implementaciju SMTP protokola sa dokazom uloženog rada klijenta. U petom poglavlju biće dati rezultati testiranja implementiranog anti-spam sistema

i evaluacija uticaja uloženog rada klijenta na redukciju broja spam poruka na mreži. Zaključak će biti dat u šestom poglavlju.

II. SIMPLE MAIL TRANSFER PROTOCOL

SMTP protokol definiše pravila na osnovu kojih se obavljaju slanje i pouzdani prenos imejl poruke kroz mrežu. Uređaji koji učestvuju u samom prenosu poruke nazivaju se agentima i njihova komunikacija je definisana SMTP protokolom.

Transfer imejl poruke otpočinje slanjem poruke od strane korisničkog mejl klijenta - MUA (eng. *mail user agent*). Poruka se pomoću SMTP protokola šalje do predajnog mejl agenta - MSA (eng. *mail submission agent*), koji šalje imejl do mejl transfer agenta - MTA (eng. *mail transfer agent*). Zavisno od lokacije pošiljaoca i primaoca na mreži, prenos poruke se može odvijati kroz jedan ili više MTA uređaja. Poslednji MTA u nizu uređaja na mreži koji prosleđuju poruku, predaje poruku MDA (eng. *mail delivery agent*) uređaju. MDA dostavlja poruku u sanduče primaoca, koja u njemu biva sačuvana [3].

Mejl transakcija se sastoji iz tri osnovna koraka. Započinje komandom MAIL. Ovom komandom klijent definiše imejl adresu pošiljaoca. Drugi korak je definisanje imejl adrese primaoca komandom RCPT. Treći korak transakcije započinje DATA porukom od strane klijenta, koja označava da se svaka naredna poruka klijenta tumači kao deo poruke za slanje, sve do prijema tačke u zasebnoj liniji, što označava kraj poruke. Komandom QUIT klijent zahteva prekid konekcije. Odgovori servera osiguravaju sinhronizaciju akcija u procesu SMTP komunikacije klijenta i servera. Na početku sadrže trocifrenu numeričku oznaku, koja klijentu pruža informaciju o trenutnom stanju servera i akciji koju klijent treba da preduzme u nastavku [1]. Osnovna komunikacija SMTP klijenta i servera prilikom uspešnog slanja poruke, prikazana je u nastavku [1]:

S: 220 smtpServer.example.com
C: HELO smtpClient.example.com
S: 250 smtpClient.example.com, pleased to meet you
C: MAIL FROM:<from@example.com>
S: 250 Sender OK
C: RCPT TO:<recipient@example.com>
S: 250 Recipient OK
C: DATA
S: 354 Enter mail, end with \".\" on a line by itself
C: From: "From Example" <from@example.com>
C:To: Recipient Example <recipient@example.com>
C: Date: Tue, 15 November 2018 16:02:43
C: Subject: Hello
C: Hello world!
C: .
S: 250 Queued mail for delivery
C: QUIT
S: 221 Service closing transmission channel

III. PROOF OF WORK SISTEM

Proof of work sistemi su nastali u cilju prevencije i sprečavanja zloupotrebe procesorske moći računara, npr. prilikom DOS (*Denial of Service*) napada, slanja masovnih spam poruka i sl. Osnovna ideja ovog koncepta je ulaganje relativno male količine procesorskog vremena uređaja, svaki put kada on zahteva pristup zaštićenom resursu na mreži. Na taj način se onemogućava delovanje osnovne karakteristike ranije pomenutih napada - velikog broja pristupa resursu u kratkom vremenskom periodu [4].

A. Princip rada POW algoritma

Princip rada POW sistema zasniva se na tipičnom kriptografskom scenariju, u kom jedna strana koja učestvuje u komunikaciji, pokušava da svoju validnost dokaže drugoj. Konkretno, klijent koji traži od servera uslugu ili resurs, prvenstveno mora dokazati svoju pouzdanost. Način ostvarivanja takvog dokaza je ulaganje određene količine procesorskog vremena, u cilju ispunjenja kriterijuma zadatog od strane servera.

Najčešće, izvršavanje matematičke ili kriptografske funkcije predstavlja način da se uloži dovoljno procesorskog vremena kao dokaz kredibiliteta. One nisu preterano zahtevne, ali jesu dovoljno kompleksne da obezbede da za slučaj da je neophodno njihovo višestruko izvršavanje, klijent mora uložiti značajno vreme procesora. Zadatak klijenta je izvršavanje vremenski zahtevne računice nad određenim podacima, više puta uzastopno, sve dok dobijeno rešenje ne zadovoljava kriterijum postavljen od strane servera. Zatim, klijent takvo rešenje šalje serveru. Zadatak servera je provera validnosti dobijenog rešenja i identifikacija klijenta kao pouzdanog ili nepouzdanog, na osnovu prethodne analize.

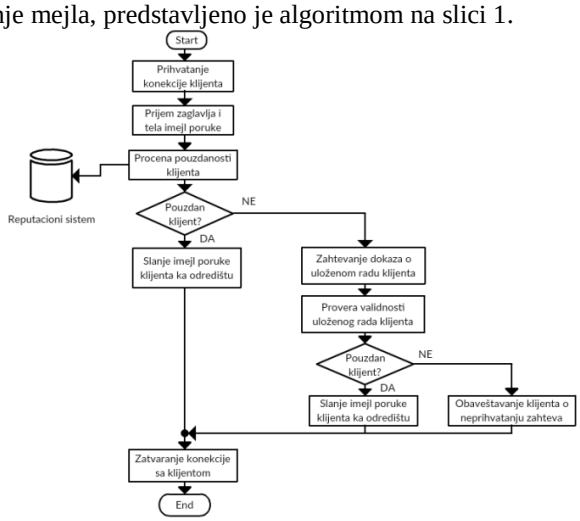
IV. IMPLEMENTACIJA ANTI-SPAM SISTEMA

Osnovna prednost implementacije POW sistema na mejl serveru je mogućnost da od potencijalno nevalidnog klijenta zatraži ulaganje određene količine procesorskog vremena. Uloženo vreme predstavlja dokaz o njegovom kredibilitetu. Takvom klijentu se postavlja zahtev za ulaganjem određenog rada za svaku imejl poruku koju pokušava da pošalje. Na ovaj način, server može zaključiti da li je klijent sa kojim komunicira validan korisnik elektronske pošte ili ima nameru zloupotrebe mejl servera. Implementacija ovakvog sistema podrazumeva određene izmene u implementaciji SMTP protokola, tj. u klijent-server komunikaciji prilikom slanja mejla. U nastavku ovog poglavlja biće opisan precizan tok SMTP komunikacije, sa naglaskom na izmenama predloženim u ovom radu.

Klijent serveru dostavlja informaciju o pošiljaocu, primaocu i celokupnoj poruci koja se šalje, onako kako je definisano osnovnim SMTP protokolom, bez ikakvih izmena. Server, nakon primljenih svih podataka koji čine

mejl, ima mogućnost prosleđivanja imejl poruke do njenog odredišta, na isti način kako to čini u protokolom definisanoj komunikaciji. Ali, ako server proceni da bi klijent mogao biti nepouzdan, projektovano proširenje SMTP protokola predloženo u ovom radu, mu omogućava da pre prosleđivanja poruke od klijenta zatraži dokaz o kredibilitetu. Sa tim ciljem, načinjene su izmene u osnovnoj razmeni SMTP poruka u tom trenutku komunikacije. Izmene omogućavaju serveru slanje celobrojne vrednosti, koja predstavlja *težinu*. Njena vrednost određuje kriterijum koji klijent mora da ispuni. Izvršavanjem kriptografske heš funkcije nad imejl porukom, klijent mora da generiše niz koji ima onoliko nula na početku, koliko je to definisano prethodno primljenim parametrom težine. Definisanjem vrednosti parametra težine, server može različitim klijentima zatražiti različitu količinu rada procesora. Primena iste heš funkcije nad istim nizom podataka uvek daje isti izlaz. Zato se podacima nad kojima se ona primenjuje pripaja vrednost *nonce*, a zatim se računa sažetak novodobijenog niza podataka. Jedini način za pronalazak *nonce* vrednosti koja ispunjava uslov servera, je primenom grube sile (eng. *brute force*). U cilju pronalaska *nonce* vrednosti koja će zadovoljiti kriterijum servera, heš funkcija se uzastupno vrši više puta, a sa svakom primenom funkcije vrednost *nonce* se menja. Uzimajući u obzir osobinu heš funkcija da za male promene na ulazu daju drastične promene na izlazu, sa svakom primenom funkcije nad podacima dobijaju se potpuno različite vrednosti. Heš funkcija se primenjuje uz različite vrednosti parametra *nonce* sve dok generisani izlaz ne ispunjava kriterijum servera. Jedno izvršavanje heš algoritma nad podacima ne zahteva znatan rad procesora, ali dobijanje zadovoljavajućeg izlaznog niza je dovoljno retko da omogući da ukupan proces oduzme klijentu značajno procesorsko vreme. Kada je pronađena vrednost *nonce* čijim korišćenjem je zadovoljen zadati uslov, klijent je prosleđuje serveru.

Funkcionisanje servera prilikom obrade zahteva za slanje mejla, predstavljeno je algoritmom na slici 1.



Slika 1. Algoritam rada PoWSMTP servera po prijemu klijentskog zahteva

Kao što je prikazano na slici, u slučaju komunikacije mejl servera i validnog korisnika elektronske pošte, server ima mogućnost da, na osnovu procene pošiljaoca reputacionim sistemom, ne zatraži ulaganje procesorskog vremena. Procedura slanja imejla, u tom slučaju, ostaje ista. Ako ipak odluči da zatraži od klijenta ulaganje rada, nakon prijema *nonce* vrednosti klijenta, vrši se provera da li dobijena vrednost ispunjava zadati kriterijum tj. da li je zatraženi rad uložen. Provera se postiže jednom primenom heš funkcije na prethodno dobijeni mejl u kombinaciji sa upravo pristiglom *nonce* vrednošću. Na ovaj način je ispunjen važan kriterijum za implementaciju POW sistema, da je potrebno uložiti značajno više procesorskog vremena za rešavanje zadatog problema u odnosu na količinu vremena potrebnog za verifikaciju njegovog rešenja. Ako server utvrdi da primljena *nonce* vrednost ispunjava prethodno zadati kriterijum, klijent biva okarakterisan kao validan. Njegov mejl se uspešno prosleđuje.

Validni korisnik, za razliku od klijenta koji vrši spam, veoma retko serveru postavlja zahtev za slanjem velikog broj imejl poruka u kratkom vremenskom periodu. Vreme koje procesor legitimnog korisnika mora uložiti za pronalazak adekvatnog sažetka nad malim brojem imejlova neće korisniku elektronske pošte otežati korišćenje mejl servisa. Na taj način je ispunjen kriterijum implementacije POW protokola, da korišćenje servera od strane validnih korisnika nema osetnih promena.

Kompletna klijent-server komunikacija, sa istaknutim izmenama predloženim u ovom radu, koje predstavljaju primer validacije klijenta, data je u nastavku:

S: 220 smtpServer.example.com
C: HELO smtpClient.example.com
S: 250 smtpClient.example.com, pleased to meet you
C: MAIL FROM:<from@example.com>
S: 250 Sender OK
C: RCPT TO:<recipient@example.com>
S: 250 Recipient OK
C: DATA
S: 354 Enter mail, end with \".\" on a line by itself
C: From: "From Example" <from@example.com>
C:To: Recipient Example <recipient@example.com>
C: Date: Tue, 15 November 2018 16:02:43
C: Subject: Hello
C: Hello world!
C: .

S: 250 Challenge number 03
C: Nonce : 8167

S: 250 Queued mail for delivery
C: QUIT
S: 221 Server closing connection

Uokvireni deo komunikacije prikazuje uspešnu validaciju korisnika. Server klijentu, kao potencijalno opasnom, traži ulaganje rada. Vrednost parametra težine (u primeru *Challenge number*) 3, označava da klijent mora generisati izlazni niz koji na svom početku ima upravo 3

nule. Na način koji je prethodno opisan, klijent višestrukim izvršavanjem heš funkcije nad imejl porukom koju šalje, pokušava da generiše izlazni niz koji će zadovoljavati zadati kriterijum. U datom primeru, vrednost parametra nonce je 8167. Ona označava da je vrednost 8167, nadovezana na podatke koji čine imejl poruku, sačinila podatak koji je ispunio kriterijum servera.

V. REZULTATI TESTIRANJA

Sa povećanjem vrednosti parametra težine, količina procesorskog vremena koje se mora uložiti raste. Tačno procesorsko vreme koje predstavlja uloženi rad varira, u zavisnosti od brzine procesora klijenta, njegove zauzetosti i funkcije verovatnoće, koja procenjuje u kom koraku će tražena vrednost biti dobijena. SMTP server je implementiran kao .NET aplikacija, korišćenjem *C#* programskog jezika, po algoritmu sa slike 1. Zahtevi SMTP klijenta simulirani su upotrebom Seagull softverskog alata za generisanje mrežnog saobraćaja. Testiranje servera vršeno je na AMD A10-5745M modelu procesora, sa radnim taktom 2.10 GHz. Višestrukim testiranjem za različite vrednosti težine prosleđene od strane servera, dobijeni su rezultati dati u tabeli I.

TABELA I
EVALUACIJA RADA KLIJENTA ZA RAZLIČITE VREDNOSTI TEŽINE

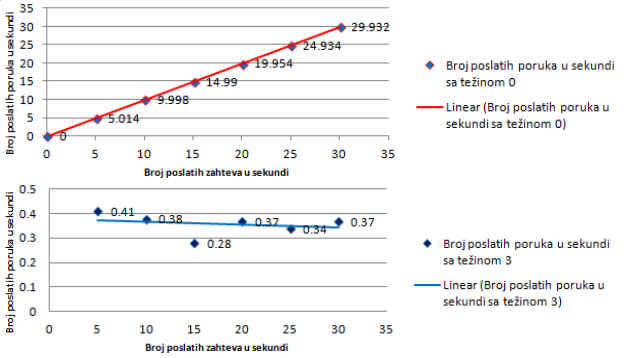
Težina	Prosečno vreme izvršenja f-je cene	Standardna devijacija vremena izvršenja	Prosečan broj izvršenja heš f-je	Broj izvršenih testiranja
1	1.14ms	0.41ms	41	20
2	28.7ms	25.09ms	1728	20
3	4.35s	2.88s	223620	20
4	2.71min	1.81min	10996000	20

Vrednost standardne devijacije je značajna za svaku vrednost težine, a razlog su velika odstupanja u ulaganju rada klijenta, upravo zbog prethodno diskutovane funkcije verovatnoće. Broj mogućih poslatih imejl poruka klijenta u sekundi zavisi od prosečnog vremena ulaganja rada klijenta (tabela I) i parametra *HP* (*hash power)*, koji predstavlja prosečan broj mogućih izvršenja heš funkcije u sekundi na klijentu. Broj poslatih imejl poruka ne zavisi od broja poslatih zahteva za slanje poruke od strane klijenta.

$$BR \approx \frac{HP}{N(T)} \qquad (1)$$

BR - Broj poslatih poruka klijenta u sekundi, *HP* - Hash power klijenta, *T* - Parametar težine, *N(T)* - Prosečan broj potrebnih izvršenja heš algoritma za težinu *T*.

Za vrednost težine 3, prosečan broj izvršenja heš funkcije dat u tabeli I i *hash power* parametar pomenutog procesora, po datoj formuli (1) broj poslatih poruka klijenta u sekundi približno je jednak 0.345. Testiranjem rada implementiranog sistema za različit broj poslatih zahteva za slanje mejla klijenta u sekundi dobijeni su rezultati prikazani na slici 2.



Slika 2. Broj poslatih poruka u sekundi u zavisnosti od broja poslatih zahteva klijenta, bez i sa ulaganjem rada.

Iz predstavljenih rezultata može se zaključiti da je implementacijom opisanog anti-spam rešenja klijent koji ima nameru zloupotrebe mejl servera ograničen na konstantan broj poslatih imejl poruka u sekundi, nezavisno od broja zahteva za slanjem koje prosledi mejl serveru. U slučaju da se od klijenta ne zahteva ulaganje rada, tj. vrednost težine je 0, klijent šalje približno onoliko mejlova koliko je zahteva uputio serveru, a slanje imejl poruke se obavlja gotovo trenutno.

VI. ZAKLJUČAK

Iz rezultata testiranja sistema se može zaključiti da implementirano anti-spam rešenje utiče na najizraženiju osobinu spama - slanje velikog broja imejl poruka tokom kratkog vremenskog perioda. Uticaj na brzinu slanja spam poruka značajno redukuje količinu poruka tog tipa na mreži, što rezultira smanjenjem opterećenja mejl servera i mreže i unapređenjem sigurnosti korisnika.

LITERATURA

[1] J. Klensin, "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008.

[2] The Radicati Group Inc, Email Statistics Report, *2018-2022.*

[3] A. Nakulas, L. Ekonomou, S. Kourtesi, G. Fotis, "A Review of Techniques to Counter Spam and Spit", 10.1007/978-0-387-8414-3_50, *Springer Science Business Media*, 2009.

[4] C. Dwork, M. Naor, "Pricing via Processing or Combating Junk Mail", *In Proc.12th Annual International Cryptology Conference on Advances in Cryptology*, London, UK, 1992.

[5] B. Laurie, R. Clayton, "Proof-of-work proves not to work", *Workshop on Economics and Information Security*, volume 2004, 2004.

[6] D. Liu, "Proof of Work can Work", *Fifth Workshop on the Economics of Information Security*, Indiana University, 2006.

[7] A. Schaub, D. Rossi, "Design and analysis of an improved bitmessage anti-spam mechanism," *2015 IEEE International Conference on Peer-to-Peer Computing (P2P)*, Boston, MA, 2015, pp. 1-5, doi: 10.1109/P2P.2015.7328523

Realizacija digitalnog Real-Time servosistema na Arduino Due mikrokontroleru

Marija Filipović, Snežana Đorđević

Sadržaj - U ovom radu opisan je servosistem baziran na Arduino Due mikrokontroleru. Opisan je princip rada *Real Time* sistema, metoda očitavanja magnetnog inkrementalnog enkodera (SSI protokol), snimanje odskočnog odziva sistema kao i identifikacija sistema u kontinualnom i pretvaranje u digitalni domen, projektovanje digitalnog regulatora metodom podešavanja polova, implementacija regulatora na mikrokontroleru i zadavanje komande motoru preko širinsko impulsne modulacije (PWM). Ovaj servosistem je korišćen kao deo laboratorijske makete.

I. UVOD

Servomehanizam ili servosistem je sistem upravljanja sa povratnom spregom kod koga je upravljana promenljiva mehanička pozicija ili brzina. Osnovna funkcija servomehanizma je da njegov izlaz brzo i precizno prati promene ulaza. Nekada se naziva i prateći jer mu je samo ime izvedeno iz latinskih reči *servus* (onaj koji služi ili prati) i *mehanism* (mehanizam).

Servomehanizam funkcioniše na taj način što se izlazna veličina stalno meri i posredstvom povratne sprege upoređuje sa ulaznom veličinom, dejstvujući pri tom tako da grešku koja nastaje kao rezultat poređenja izlazne i ulazne veličine svodi na nulu, uz pojačanje snage [1].

Servosistem može da obezbedi tri povratne sprege: pozicionu, brzinsku, strujnu. Servosistemi se koriste za upravljanje robotima kod automatizovanih procesa u industriji, u automobilskoj industriji, u mobilnoj robotici, na raznim uređajima koji se mogu susresti svakoga dana kao što je npr. tempomat u automobilu.

II. REAL TIME SISTEMI (RTS)

RTS su sistemi za rad u realnom vremenu. Tu spada veliki opseg sistema. RTS su sistemi koji kontrolišu aplikacije specijalne namene, kao što su: specijalni grafički sistemi, sistemi za medicinsku grafiku, sistemi za industrijsku kontrolu, kontrolisanje naučnih eksperimenata itd.

Sistem za rad u realnom vremenu koristimo jer projektovanje digitalnog regulatora zahteva fiksnu periodu diskretizacije.

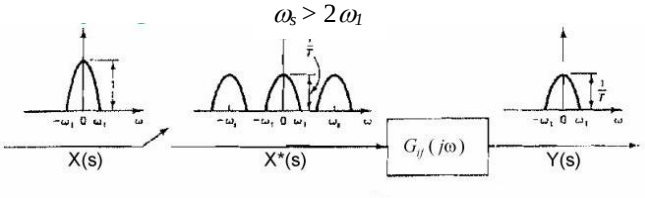
Marija Filipović, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: machefilipovic21@gmail.com

Snežana Đorđević, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: snezana.elfak@gmail.com

A. Šenonova teorema odabiranja

Šenonova teorema odabiranja (Slika 1) glasi: Vremenski kontinualni signal *x(t)*, koji ne sadrži frekvencijske komponente veće od ω_1, se može kompletno rekonstruisati na osnovu uzrokovanih vrednosti *x*(t)* ukoliko je kružna učestanost uzrokovanja ω_s veća od $2\omega_1$ [2].

Eksperimentalnom metodom smo došli do minimalne periode diskretizacije od 4ms.

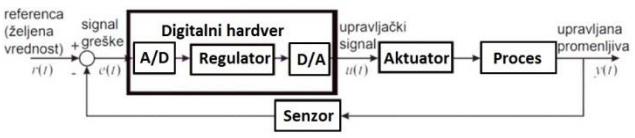


Slika 1. Rekonstrukcija signala.

B. Digitalni sistemi automatskog upravljanja (DSAU)

Digitalni sistem automatskog upravljanja (DSAU) koristi digitalne signale i digitalni računar u cilju upravljanja procesom i njegovom regulacijom (Slika 2).

Kod digitalnih sistema automatskog upravljanja, regulator je realizovan korišćenjem digitalnog hardvera. Izlaz digitalnog regulatora je formiran na osnovu programiranog upravljačkog algoritma (upravljačkog zakona) uz pomoć numeričkih kalkulacija. Analogno-digitalni (A/D) i digitalno-analogni (D/A) konvertori su neophodni kao spona između digitalnog domena i kontinualnog sveta. Raznovrsan broj realizovanih DSAU struktura i regulatora se mogu sresti u praksi.



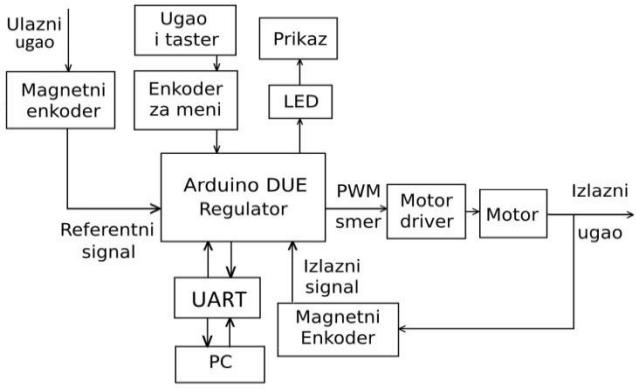
Slika 2. Opšta struktura digitalnog sistema automatskog upavljanja.

Sistem funkcioniše tako što se referenca oduzme od stvarne pozicije, formira se signal greške i onda se on dovodi na ulaz regulatora koji daje upravljanje. Signal greške može biti pozitivan ili negativan i govori koliko se

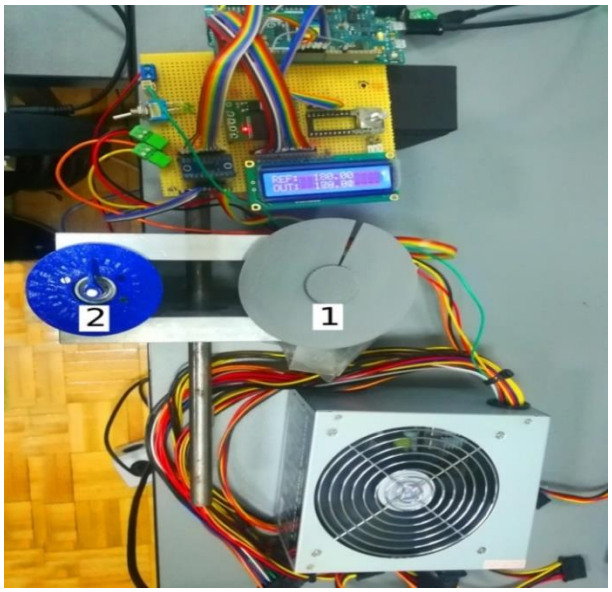
trenutna ostvarena vrednost (pozicija ili brzina) razlikuje od željene, odnosno referentne. Na aktuator dovodimo upravljanje koje je dovedeno od regulatora. Aktuator je u našem slučaju motor koji vrši kompenzaciju greške (ima ulogu da je smanji). Cilj sistema automatskog upravljanja je da grešku dovede na nulu. Prelazak iz kontinualnog u digitalni domen i primena zakona upravljanja su uslovljeni fiksnom periodom diskretizacije. Zbog toga je realizovana *Real Time* prekidačka funkcija koja se osvežava na svakih 4ms.

III. REALIZACIJA SISTEMA

Elementi sistema su: mikrokontroler Arduino DUE, CROUZET motor 82850002 (24V, 52W), dva identična apsolutna magnetna enkodera, drajver za motor koji se upravlja preko PWM-a, napajanje (računarsko napajanje) i odgovarajuće propratne aktivne i pasivne komponente koje omogućavaju rad sistema (Slika 3).



Slika 3. Blok šema povezivanja komponenata.



Slika 4. Realizovana laboratorijska maketa.

Softver je realizovan u C++ programskom jeziku koji Arduino platforma podržava što nam omogućava lakše razdvajanje celina korišćenjem principa objektno-orjentisanog programiranja. To nam je omogućilo da kreiramo klasu za čitanje podataka sa jednog enkodera izvršimo instanciranje te klase kako bismo očitali podatke sa dva enkodera (referenca i ostvarena pozicija).

A. Real Time petlja

Servosistemi su sistemi za rad u realnom vremenu i zbog toga je potrebno na Arduino DUE mikrokontroleru pored glavne petlje (*loop*), realizovati vremenski kritičnu petlju (RT petlja).

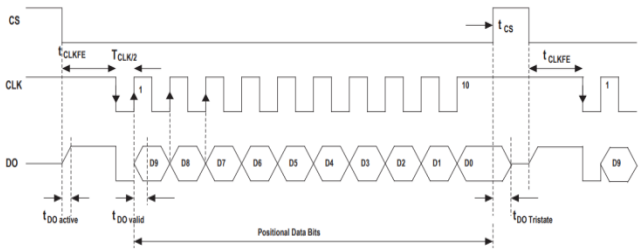
Da bi sistem funkcionisao sva izračunavanja moraju da se izvršavaju dovoljno brzo, odnosno pre isteka vremena koje predstavlja period diskretizacije sistema. Perioda diskretizacije treba da bude duplo veća od gornje granične frekvencije signala po Šenonovoj teoremi odabiranja koju smo naveli ranije. Eksperimentalnom metodom smo došli do minimalne periode diskretizacije od 4ms. Za to vreme potrebno je izvršiti: čitanje sa enkodera, primenu zakona upravljanja (zatvaranje povratne sprege po poziciji ili brzini).

B. Očitavanje podataka sa enkodera i SSI serijski protokol

Enkoder [9] koristi SSI protokol koji koristi tri pina - *Chip Select*, *Clock* i *Data*.

Sinhroni serijski interfejsi (SSI) su široko korišćeni standardni serijski interfejsi za industrijske aplikacije između glavnog *master* (npr. kontrolera) i podređenog - *slave* (npr. senzora). SSI je zasnovan na RS-422 standardima i ima visoku efikasnost protokola na raznim hardverskim platformama. Veoma je pogodan za aplikacije koje zahtevaju pouzdanost i robusnost u merenjima u različitim industrijskim okruženjima [4].

Chip Select se spusti na nulu, na *Clock pin* se dovodi pravougaoni periodični signal periode 2µs, širine 1µs i pri svakoj rastućoj ivici (*Rising Edge*) očitava se vrednost sa *Data* pina (Slika 4) [5].



Slika 4. Sinhroni serijski interfejs.

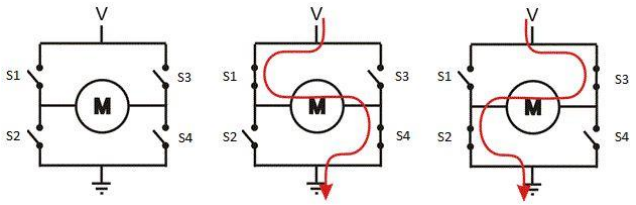
"Angle reader" je klasa koja služi za očitavanje uglova. Ona čita informaciju sa enkodera i formiraju se dva objekta klase, jedan je "motor angle" a drugi objekat je za enkodere. Oni primaju podatke sa pinova 42, 44, 46. Da bi

se izvršila korekcija podataka očitanih sa makete potrebno je da se realizuje *offset* od 276 inkrementa u odnosu na nulu.

Informacija sa enkodera se pretvara u ugao, a ugao u brzinu. Postoje apsolutni i relativni ugao. Apsolutni ugao je ugao koji se dobija kada se uključi sistem i očita ugao na trenutnoj poziciji. Relativni ugao je ugao koji ima vrednost nula pri samom uključivanju sistema. Povratna sprega se zatvara po apsolutnom uglu, jer kada bi se zatvorila po relativnom uglu postojao bi offset prilikom startovanja sistema. Najveći problem kod očitavanja je to što senzor daje apsolutni ugao, odnosno ugao u krugu (od 0 do 360 stepeni). Potrebno je vrednost pretvoriti u relativnu pri čemu mora da se vodi računa o prolasku kroz pun krug, odnosno koda očitana vrednost pređe sa 360^0 na $0\ ^0$ i obrnuto. Mogućnost za grešku u očitavanju je u ovom slučaju najveća kao i sama greška.

C. Motor driver, H-most i PWM

Električno kolo za upravljanje motorom funkcioniše na principu H-mosta. H-most je strujni krug koji se sastoji od 4 prekidača spojenih s motorom na način prikazan na Slici 5. Ako S2 i S3 prekidače zatvorimo, a S1 i S4 prekidače otvorimo, struja će teći u određenom smeru. Ako zamenimo stanja prekidača tako da su S2 i S3 otvoreni a S1 i S4 zatvoreni, struja će kroz motor proteći u suprotnom smeru. To dovodi do rotacije osovine motora u suprotnom smeru od prvobitnog. Uz to, ne smemo zatvoriti sva četiri prekidača odjednom ili oba prekidača na jednoj strani H-mosta (npr. S1 i S2) jer time izazivamo kratak spoj [6].



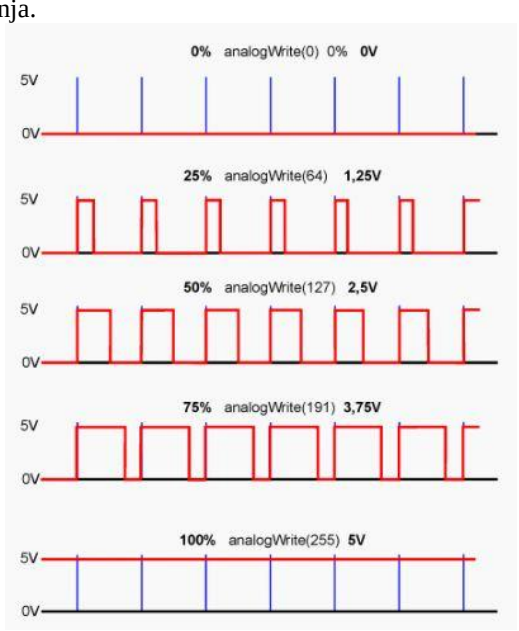
Slika 5. H-most.

D. Zadavanje komande motoru

Motor Driver ima tri pina, INA, INB koji služe za podešavanje smera motora i PWM, pin koji služi za zadavanje brzine obrtanja motora.

Impulsno širinska modulacija (Slika 6) ima širok spektar primena. Od primena u komunikacijama i raznim merenjima pa sve do primene u upravljanju snagom i do konverzije energije u neke druge oblike energije. Promena smera motora se vrši preko promene pinova.

Brzina obrtanja motora je proporcionalna širini impulsa koji se dovodi na PWM pin drajvera pri čemu treba naglasiti da širina impulsa od 0% ne garantuje brzinu 0 0/s ako motor nije zakočen kako se njegova osovina može pomerati usled inercije ili spoljašnjih uticaja. Kočenje motora se osigurava pinovima za promenu smera obrtanja.



Slika 6. Impulsno širinska modulacija.

Smer obrtanja motora je definisan određenom kombinacijom pinova INA i INB. Kombinacija 0 i 1, određuje jedan smer obrtanja, kombinacija 1 i 0 suprotan smer, dok kombinacija 0 i 0 vrši kočenje motora [3].

U zavisnosti od toga da li je greška koja treba biti kompenzovana pozitivna ili negativna, izlaz regulatora može biti pozitivan ili negativan. Znak ove vrednosti određuje smer okretanja motora.

Kako bismo promenu smera vršili na najbolji mogući način potrebno je prvo dovesti brzinu na 0 i izvršiti kočenje motora i potom promeniti smer obrtanja motora.

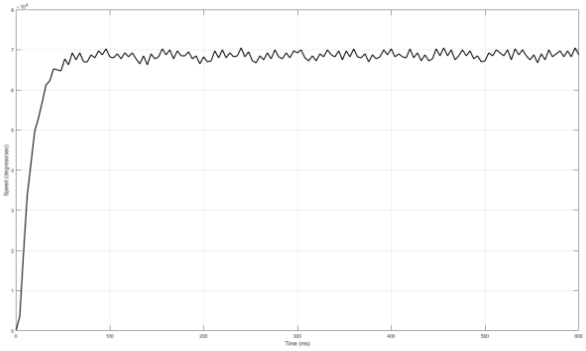
IV. PROJEKTOVANJE DIGITALNOG PI REGULATORA

A. Snimanje odskočnog odziva i identifikacija sistema

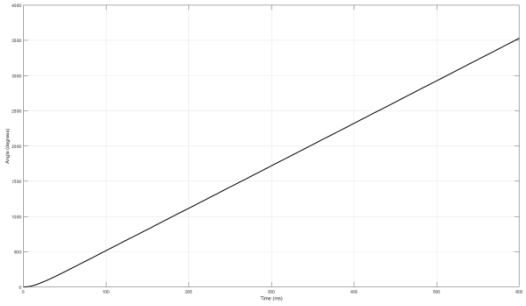
Poseban Arduino kod je pisan samo za određivanje odskočnog odziva sistema. Uz pomoć serijske komunikacije se kontroleru zadaje zadatak snimanja odskočnog odziva tako što se motoru dovede 50% širine PWM signala, resetuje sistem za merenje vremena i pokrene slanje podataka na računar. Na računaru smo izvršili prijem tih podataka koji se sastoje iz trenutnog vremena u milisekundama trenutne očitane pozicije (Slika 8) i trenutne očitane brzine (Slika 7). Te podatke smo prikazali grafički i pristupili identifikaciji sistema (Slika 9).

Šum koji možemo videti na odzivu brzine javlja se zbog zaokruživanja prilikom izračunavanja brzine na mikrokontroleru. Brzina se računa kao razlika očitanog ugla u jedinici vremena. Kako je jedinica vremena 4 ms, a rezolucija enkodera 0,088 ^{0} šum čitanja koji se vidi na grafiku iznosi 21,97 ^{0}/s i to je minimalna promena koju

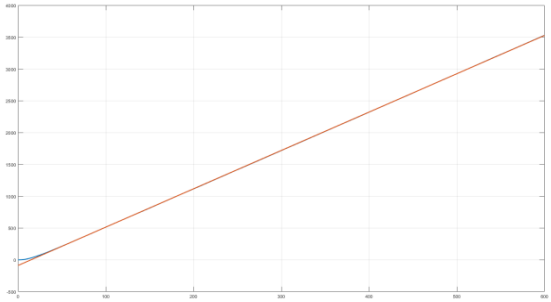
sistem može da detektuje. Šum se javlja zbog toga što sistem napravi razliku u očitavanju između dve periode diskretizacije u iznosu od 1 inkrementa.



Slika 7. Odziv brzine.



Slika 8. Odziv pozicije.



Slika 9. Identifikacija sistema.

Funkcija prenosa za dati odziv je oblika:

$$G(s) = \frac{\theta(s)}{U(s)} = \frac{k_m}{s(1+T_m)} = \frac{b}{s(s+a)}$$

Vidimo da imamo astatičan sistem sa negativno prigušenim odzivom:

$t = 0{,}6s$
$\theta(\infty) = \infty$

Na liniju odziva povlačimo tangentu i cilj nam je da odredimo u kojim tačkama tatangenta seče t i θ ose. Za koordinate uzimamo proizvoljne tačke sa linije odziva ($x_1 < x_2$ i $y_1 < y_2$).

$x_1 = 428 \quad y_1 = 2491$
$x_2 = 504 \quad y_2 = 2949$

Sređivanjem jednačine prave kroz dve tačke opštem obliku (y=kx+n) dobijamo da su:

$$k = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{i} \quad n = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}$$

Izračunavanjem dobijamo vrednosti:

$$k = 6.026 \quad \text{i} \quad n = -0.088263$$

$$Za \quad x = 0 \quad => \quad y = -0.088263$$
$$Za \quad y = 0 \quad => \quad x = 0.0146 = T_m$$

Sada možemo da izračunamo k_m, a i b:

$$K_m = \frac{K}{200} = \frac{6{,}026}{200} = 0{,}03013$$

$$a = \frac{1}{T_m} = \frac{1}{0{,}0146} = 68{,}49315$$

$$b = K_m \cdot a = 2{,}0637$$

Sada ćemo preći u Z-domen primenom Z-transformacije:

$$G(Z) = Z \cdot L^{-1}\{G_{h_0}(S) \cdot G(S)\}$$

gde je $G_{h0} = \frac{1-e^{-sT}}{s}$ kolo zadrške nultog reda.

$$G(z) = L^{-1}Z\{G_{h0}(s) \cdot G(s)\} = L^{-1}Z\left\{\frac{1-e^{-sT}}{s} \cdot \frac{b}{s(s+a)}\right\}$$
$$= (1-z^{-1})L^{-1}Z\left\{\frac{b}{s^2(s+a)}\right\}$$
$$= \frac{b}{a}(1-z^{-1})L^{-1}Z\left\{\frac{a}{s^2(s+a)}\right\}$$
$$= \frac{b}{a}(1-z^{-1})\left[\frac{T \cdot z}{(z-1)^2} - \frac{z(1-e^{-aT})}{a(z-1)(z-e^{-aT})}\right]$$
$$= \frac{b}{a}\frac{(z-1)}{z}\frac{z}{(z-1)}\left[\frac{T}{(z-1)} - \frac{(1-e^{-aT})}{a(z-e^{-aT})}\right]$$
$$= \frac{b}{a^2}\left[\frac{z(T \cdot a + e^{-aT} - 1) - T \cdot a \cdot e^{-aT} - e^{-aT} + 1}{z^2 - z(e^{-aT}+1) + e^{-aT}}\right]$$

$$G(z) = \frac{0{,}000015004(z + 0{,}9178885630498)}{(z-1)(z-0{,}7607)}$$

Iz formule sledi:

$a_1 = -1{,}7607$
$a_2 = 0{,}7607$
$b_0 = 0{,}000015004$
$b_1 = 0{,}0092795006$

B. Projektovanje digitalnog PI regulatora

Nakon identifikacije projektuje se digitalni PI regulator pojačanja Kp i Ki metodom podešavanja polova, tako da odskočni odziv sistema karakterišu sledeće vrednosti relativnog faktora prigušenja (ξ) i neprigušene prirodne učestanosti (ω_n)

$$\xi = 0.9$$
$$\omega_n = 12\ rad/s$$

Za date vrednosti možemo da odredimo polove s_1 i s_2. Polove određujemo na osnovu jednačine:

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

$$s_1 = -10.8 \pm j5.23$$

Zatim menjamo u jednačinu:

$$z_{1,2} = e^{-\zeta\omega_n T \pm j\omega_n T\sqrt{1-\zeta^2}}$$

$$z_{1,2} = 0.95751 \pm j0.020034$$

Opšti oblik željene jednačine regulatora je:

$$z^3+z^2*(-(z_1+z_2)-\sigma_z)+z*(z_1*z_2+\sigma_z*(z_1+z_2))- z_1*z_2*\sigma_z=0 \quad (1)$$

Izračunavanjem sledeće formule dobijamo oblik željene jednačine za dati regulator:

$$W_p(z) = \left(K_p + K_i \frac{z}{z-1} \right) * W(z)$$

Karakteristična jednačina sistema ima oblik:

$$1+Wp(z)=0$$

Tako dobijamo oblik željene jednačine PI-regulatora:

$$z^3 + z^2 * (a_1 - 1 + b_0*K_p + b_0*K_i) +$$

$$+z*(a_2 - a_1 - b_0*K_p + b_1*K_i) - a_2 - b_1* K_p=0 \quad (2)$$

Na osnovu dobijenih vrednosti $z_{1,2}$ imamo:

$$z_1*z_2 = 0.917226761$$
$$z_1+z_2=1.91502$$

Poređenjem jedna

$$a_1 - 1 + b_0*K_p + b_0*K_i + z_1 + z_2 + \sigma_z = 0$$
$$a_2 - a_1 - b_0*K_p + b_1*K_i - z_1*z_2 - \sigma_z*(z_1+z_2) = 0$$
$$- a_2 - b_1* K_p + z_1*z_2* \sigma_z = 0$$

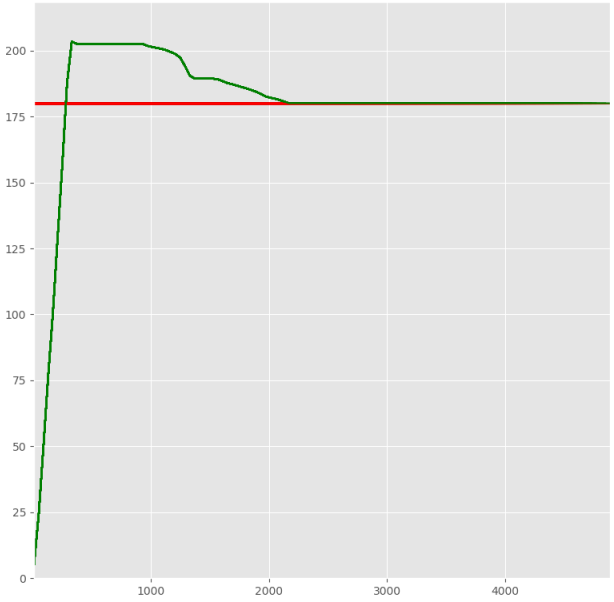
Nakon izračunavanja ove tri jednačine dobijamo sledeće vrednosti parametara regulatora:

$$K_p = 0.536426177157006$$
$$K_i = 0.012432547378634$$
$$\sigma_z = 0.837727002982122$$

Na osnovu dobijenih parametara, prethodnog upravljanja i trenutne greške formira se novo upravljanje. Formula na izlazu daje jačinu PWM-a, dok nam odskočni odziv pokazuje ispravnost rada regulatora.

$$U(kT) = U((k-1)T) + K_i e(kT) + Kp(e(k)-e(k-1))$$

Nakon implementiranog regulatora na mikrokontroleru dobija se sledeći odskočni odziv sistema:



Slika 10. Odskočni odziv sistema.

V. ZAKLJUČAK

Projektovanje ovog sistema nam je pokazalo koliko su zapravo *Real Time* sistemi osetljivi i koliko se mora voditi računa o izboru periode diskretizacije. Mnoge stvari u sistemu su kritične tako da se mora voditi računa da se i pri najosnovnijim stvarima kao što je promena smera okretanja motora ne napravi greška prilikom primene zakona upravljanja. Ovaj sistem je samo deo mnogo većeg sistema [7] i koncipiran je tako da povezuje projektovanje sistema u digitalnom domenu i realizaciju istog na mikrokontroleru.

LITERATURA

[1] Dr Miroslav Bjekić, Stevanović Zoran, Servo regulator Lexium LXM32M, str.7
[2] Dr Boban Veselić, Metode digitalnog upravljanja i estimacije
[3] Ceca Radlovacki, PWM, https://cecaradlovacki.wordpress.com/2015/04/10/sta-je-pwm/

[4] SSI (Serial Synchronous Interface), https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019MgLSAU&l=en-RS
[5] AEAT-6010/6012 Magnetic Encoder, https://docs.broadcom.com/docs/AV02-0188EN
[6] KKM: DC motor driver dual H-bridge, https://e-radionica.com/hr/blog/2017/12/10/kkm-dc-motor-driver-dual-h-bridge/
[7] Snežana Đorđević i Marija Filipović, Servosistemi za laboratorijske makete bazirane na Arduino Due mikrokontroleru, IEEESTEC 12th Student project conference

Sistem za relejnu zaštitu zasnovan na Arduino Uno u okviru podacima-vođene arhitekture pametne mreže

Nenad Petrović, Đorđe Kocić

Sadržaj – U slučaju otkaza, u električnoj mreži je od ključne važnosti da se oprema što pre dovede u stanje kojim ne ugrožava bezbednost i okolinu. U ovom radu, prikazana je upotreba Arduino Uno u okviru sistema za relejnu zaštitu unutar podacima-vođene arhitekture pametnih mreža i opisana implementacija.

I. Uvod

Oprema i potrošači u elektroenergetskim sistemima zahtevaju zaštitne uređaje, koji imaju zadatak da ih štite prilikom nastalih anomalija, smetnji, grešaka i otkaza [1]. Glavna funkcija zaštitnih uređaja je ne da spreče da dođe do otkaza, već da izvrše određene akcije nakon detekcije neispravnosti. Zaštitini uređaji treba da detektuju, lokalizuju i uklanjaju problematičnu opremu iz električne mreže u što kraćem roku [1]. Osim toga, na ovaj način se sprečava da problemi nastali u električnoj mreži ugroze život u okolini.

Poslednjih godina, dolazi do evolucije postojećih sistema elektrodistribucije pomoću informaciono-komunikacionih tehnologija u takozvane *pametne mreže*. Pametne mreže su elektroenergetske mreže nove generacije implementirane kao dvosmerni sajber-fizički sistemi, sa ugrađenom računarskom inteligencijom, koja iskorišćava prikupljene podatke za upravljanje sa ciljem da se ostvari čista, bezbedna, pouzdana i efikasna isporuka električne energije krajnjim korisnicima [2, 3]. Osim toga, jedna od glavnih karakteristika pametne mreže jeste mogućnost detekcije nastalih promena i primena odgovarajuće strategije kao odgovor na novonastale uslove rada u približno realnom vremenu [2].

Relejna zaštita je ključna za bezbedno funkcionisanje elektroenergetskih sistema [4], među kojima su i pametne električne mreže, kod kojih ugrađena računarska inteligencija omogućava primenu adaptivnih mehanizama za efikasniju zaštitu [5, 6]. U ovom radu, prikazana je implementacija sistema za relejnu zaštitu u okviru podacima-vođene pametne mreže [3] zasnovana na Arduino Uno hardveru. Potencijal Arduino Uno za primenu u okviru sistema relejne zaštine pokazan je u [7].

Nenad Petrović je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: penenadpi@gmail.com

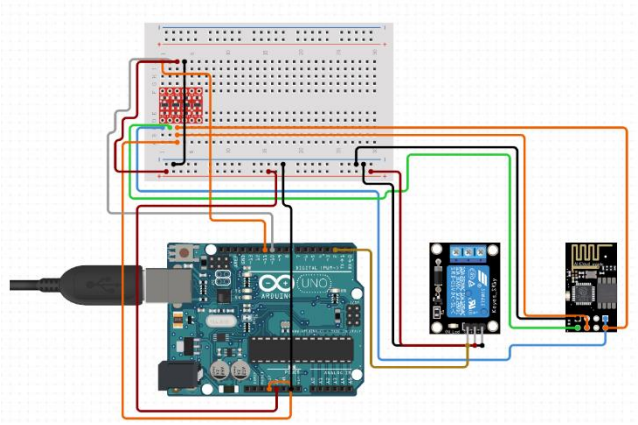
Đorđe Kocić je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: seriousdjoka@gmail.com

II. Pregled implementacije

A. Hardverska platforma

Srž hardverske konfiguracije ovog sistema jeste Arduino Uno [8], mikrokontroler pristupačne cene i malih dimenzija. Što se njegovih karakteristika tiče, izdvaja se to da koristi 8-bit mikroprocesor, napaja sa 5V, poseduje 16 digitalnih pinova koji mogu biti ulazni ili izlazni, 6 analognih ulaza, 16MHz kvarcni kristal 32KB FLASH, 2KB SRAM, 1KB EEPROM i USB konektor.

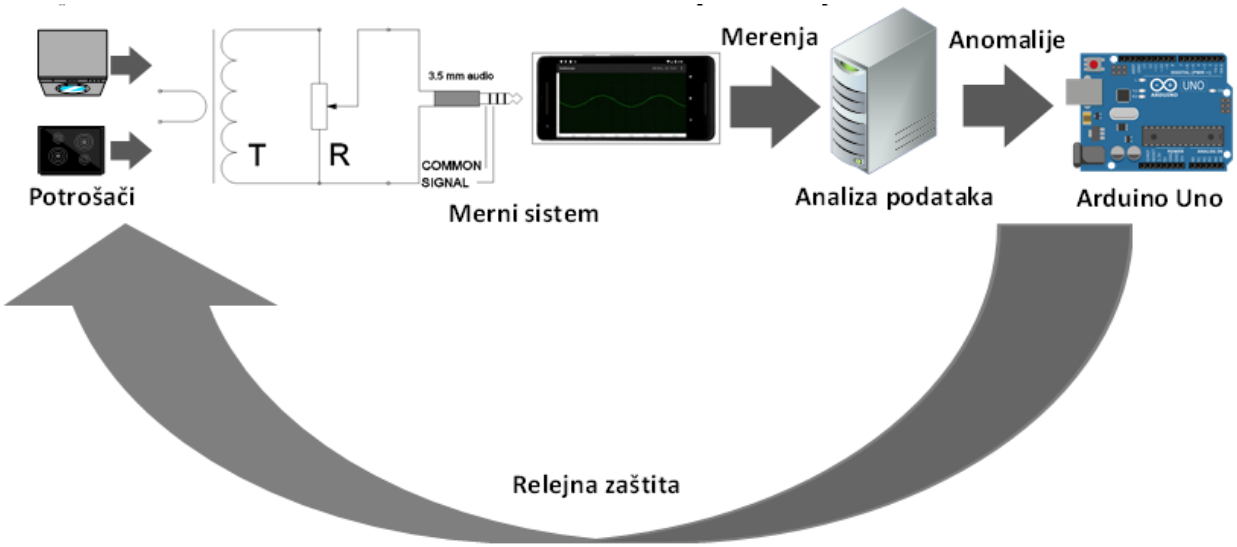
Za relejnu zaštitu, koristi se povoljni jednokanalni relej TONGLING JQC-3FF-S-Z [9], a za komunikaciju putem Wi-Fi, sistem se oslanja na ESP8266 [10] mikročip koji koji podržava kompletan TCP/IP stek. Na slici 1 je prikazana šema povezivanja Arduino Uno sa prethodno navedenim dodatnim hardverom koji realizuje sistem za relejnu zaštitu.



Slika 1. Šema korišćenog hardvera sistema za relejnu zaštitu zasnovanog na Arduino Uno.

B. Komunikacioni protokol

Za komunikaciju između mašina (serveri i IoT uređaji) u našoj podacima-vođenoj arhitekturi za pametne mreže [3], koristimo MQTT (Messsage Queuing Telemetry Transport) [11]. To je lagan, *publish-subscribe*, ISO standardizovan protokol za razmenu poruka koji radi povrh TCP/IP protokola. MQTT je dizajniran za sluačjeve korišćenja u kojima je poželjno što manje zauzeće memorije ili je protok mreže ograničen, a to se uklapa u naš scenario primene Arduino uređaja prilično ograničenih hardverskih performansi u okviru pametne mreže.



Slika 2. Princip rada, tok razmene poruka i izvršenje akcija u sistemu za relejnu zaštitu u okviru podacima-vođene pametne mreže zasnovan na Arduino Uno.

Za implementaciju protokola se u našem slučaju koristi PubSubClient [12] biblioteka koja omogućava jednostavnu razmenu poruka preko MQTT protokola na Arduino uređajima. Međutim, *publish-subscribe* mehanizam MQTT protokola zahteva brokera za posredovanje prilikom razmene poruka. Za ovu svrhu, koristimo HiveMQ [13]. Po ovom mehanizmu, primaoci i pošiljaoci se prijavljuju na određene teme (*topic*) za razmenu poruka. Prilikom realizacije u ovom radu, korišćen je onlajn servis [14].

Poruke se šalju kao JSON-kodirani stringovi. U ovom formatu, poruka koja sadrži izmerene vrednosti napona i frekvencije električnog signala sa frižidera za temu pod nazivom *fridge* bi bila: `{"voltage":"201.2", "frequency":"47.3"}`. Merenje se obavlja sistemom za bežični nadzor potrošača uz pomoć Andriod uređaja [15, 16]. Ovako prikupljena merenja se zatim analiziraju primenom našeg radnog okvira za detekciju anomalija električnih signala [3] koji se izvršava na *edge* serveru. U slučaju da je došlo do anomalije, šalje se poruka ka Arduino Uno za istu temu: `{"anomaly":"1"}`. Konačno, Arduino Uno reaguje na ovu poruku tako što gasi frižider sa anomalijom koji je povezan na njegov relej. Na slici 2, prikazana je ilustracija toka razmene poruka i akcija u ovom sistemu.

C. Softver za upravljanje potrošačima uz pomoć releja

Za razvoj softvera korišćen je Arduino IDE [17] razvojno okruženje. Kod koji izvršava Arduino Uno u sistemu za relejnu zaštitu je u većem delu zasnovan na primeru za razmenu MQTT poruka preko WiFi [18], uz nekoliko modifikacija.

Prvo, u *setup* funkciji je pin kontrolnog signala za relej definisan da bude u izlaznom režimu. U našem slučaju, korišćen je PIN2, kao što se vidi na šemi sa slike 1. Nakon toga, poziva se *setup_wifi* funkcija koja inicijalizuje ESP8266 Wi-Fi modul, pri čemu su odgovarajući *ssid* i *password* mreže prethodno definisani u programu. Zatim, vrši se poveziavnje na odgovarajući MQTT server brokera preko porta 1883. Konačno, definisana je *callback* funkcija koja se poziva nakon prijema MQTT poruke. Osim toga, neophodno je prijaviti se na teme od interesa u kodu (recimo *fridge* ako konitrolišemo frižider). U listingu 1, prikazan je prethodno opisani kod za inicijalizaciju sistema i povezanih modula.

LISTING 1. KOD ZA INICIJALIZACIJU SISTEMA.

```
void setup() {
  pinMode(relay_pin, OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

Osim toga, redefinisana je *callback* funkcija iz primera [9], kao što je prikazano u listingu 2. Nakon prijema poruke o anomaliji koja ima *payload* = 1 na kontolni pin releja se postavi 1 da bi se ugasio potrošač usled anomalije. Ukoliko dođe poruka koja ima vrednost *payload* = 0, znači da treba ponovo uključiti potrošača ako je anomalija otklonjena.

LISTING 2. KOD LOGIKE ZA KONTROLU POTROŠAČA NA OSNOVU PRIMLJENE MQTT PORUKE.

```
void callback(char* topic, byte* payload,
unsigned int length) {
  //Ugasiti potrošača usled anomalije
  if ((char)payload[0] == '1') {
    digitalWrite(relay_pin, HIGH);
  }
  //Uključiti potrošača
  if ((char)payload[0] == '0') {
    digitalWrite(relay_pin, LOW);
  }
}
```

D. Mehanizam za detekciju anomalija

Za implementaciju mehanizma detekcije anomalija prilikom rada potrošača, oslanjamo se na TensorFlow [19], biblioteku otvorenog koda za mašinsko učenje u programskom jeziku Python. Ova biblioteka pruža mogućnost izvršenja na GPU, što je pogodno kada je kritično vreme obrade u realnom vremenu.

U našem slučaju, merenja frekvencije i napona treba klasifikovati u jednu od dve kategorije: "0" – normalan rad potrošačkog uređaja; "1" – anomalija pri radu. Frekvencija i napon su u ovom slučaju nezavisne promenljive, dok je prisustvo anomalije zavisna promenljiva.

Merenja su prikupljena uz pomoć [15, 16]. Obuka neuronske mreže je obavljena uz pomoć trening skupa podataka kome su dodate oznake da li je anomalija prisutna ili ne. Na osnovu trening skupa (1000 merenja), neuronska mreža ima zadatak da za svako novo merenje detektuje anomalije. U našem slučaju, ostvarena tačnost klasifikacije je nešto iznad 92%.

III. EVALUACIJA

U Tabeli I su dati rezultati evaluacije. Detekcija anomalija izvršavana je na laptopu sledećih karakteristika: Intel i7 7700-HQ quad-core 2.80GHz CPU, 16GB DDR4 RAM i NVIDIA GTX1050 GPU 2GB DDR5 VRAM.

Prva kolona prikazuje broj nadziranih potrošačkih uređaja. Druga kolona je vreme neophodno za otklanjanje neispravnih uređaja nakon detekcije anomalije uz pomoć sistema za relejnu zaštitu (odziv), dat u sekundama kao prosek 10 uzastopnih izvršenja.

Detekcija anomalija vršena je svake sekunde, pri čemu za svaki uređaj stiže 100 merenja u sekundi. Potrošački uređaji korišćeni za evaluaciju su grejalica, frižider i mašina za pranje veša. Kao što se može videti, odziv sistema opada sa povećanjem broja potrošača, što je očekivano, jer i broj merenja za analizu postaje veći.

TABELA I
ODZIV SISTEMA ZA RELEJNU ZAŠTITU U ZAVISNOSTI OD BROJA POTROŠAČA.

Broj potrošača	Odziv [s]
1	1.03
2	1.41
3	2.32

IV. ZAKLJUČAK I BUDUĆI RAZVOJ

Sistem ima dosta potencijala za praktičnu primenu u okviru pametnih mreža, s obzirom na pristupačnu cenu, nisku potrošnju i male dimenzije. Međutim, glavno ograničenje prikazane implementacije jeste to što jednokanalni releji zahtevaju zaseban Arduino Uno po potrošačkom uređaju. Osim toga, odziv celokupnog sistema u sprezi sa arhitekturom za pametne mreže predstavljene u [3] još uvek nije dovoljno brz za kritične primene u realnom vremenu.

U budućnosti, planiramo da primenimo i višekanalne releje, što bi omogućilo povezivanje većeg broja potrošača na jedan Arduino Uno mikrokontroler, ali i unapređenje performansi postojećeg podacima-vođenog radnog okvira za pametne mreže [3], a unutar njega sistem predstavljen u ovom radu ima ulogu aktuatora.

LITERATURA

[1] C. R. Bayliss, B. J. Hardy, "Relay Protection. Transmission and Distribution Electrical Engineering", pp. 287–359, 2012. https://doi.org/10.1016/b978-0-08-096912-1.00010-1

[2] E. Kabalci, Y. Kabalci, "Introduction to Smart Grid Architecture", Smart Grids and Their Communication Systems, pp. 3–45, 2018. https://doi.org/10.1007/978-981-13-1768-2_1

[3] N. Petrović, Đ. Kocić, "Podacima-vođena arhitektura za prilagodljive energetske mreže zasnovana na IoT uređajima", ETRAN 2019, Silver Lake, Serbia, pp. 880-885, 2019.

[4] B. Zhang, Z. Hao, Z. Bo, "New development in relay protection for smart grid", Protection and Control of Modern Power Systems, 1(1), pp. 1-7, 2016. https://doi.org/10.1186/s41601-016-0025-x

[5] K. Kauhaniemi, S. Voima, "Adaptive Relay Protection Concept for Smart Grids", Renewable Efficient Energy II Conference, pp. 1-10, 2012.

[6] F. Kawano et al., "Intelligent protection relay system for smart grid", 10th IET International Conference on Developments in Power System Protection (DPSP 2010), pp. 1-5, 2010. https://doi.org/10.1049/cp.2010.0211

[7] M. F. Kotb, M. El-Saadawi, E. H. El-Desouky, "Overcurrent protection relay-based Arduino Uno for FREEDM system", International Transactions on Electrical Energy Systems, e12017, pp. 1-13, 2019. https://doi.org/10.1002/2050-7038.12017

[8] Arduino Uno Rev 3 [Online]. Dostupno na: https://store.arduino.cc/arduino-uno-rev3, poslednji pristup 02/11/2019.

[9] TONGLING JQC-3FF-S-Z [Online]. Dostupno na: https://www.aliexpress.com/item/32480128984.html, poslednji pristup 02/11/2019.

[10] ESP8266 [Online]. Dostupno na: https://www.espressif.com/en/products/hardware/esp8266ex/overview, poslednji pristup 02/11/2019.

[11] MQTT [Online]. Dostupno na: http://mqtt.org/, poslednji pristup 02/11/2019.

[12] PubSubClient [Online]. Dostupno na: https://github.com/knolleary/pubsubclient, poslednji pristup 02/11/2019.

[13] HiveMQ [Online]. Dostupno na: https://www.hivemq.com/, poslednji pristup 02/11/2019.

[14] HiveMQ Websocket Client [Online]. Dostupno na: http://www.hivemq.com/demos/websocket-client/, poslednji pristup 02/11/2019.

[15] Đ. Kocić, N. Petrović, “Application of Android Based Devices in Analog Electric Signal Measurement”, YuInfo 2019, Kopaonik, Serbia, pp. 1-5, 2019.
[16] Đ. Kocić, “Wireless Android power consumer monitoring system”, eNergetics 2019, pp. 1-5, 2019.
[17] Arduino IDE [Online]. Dostupno na: https://www.arduino.cc/en/main/software, poslednji pristup 02/11/2019.
[18] Basic ESP8266 MQTT example [Online]. Dostupno na: https://github.com/knolleary/pubsubclient/blob/master/examples/mqtt_esp8266/mqtt_esp8266.ino, poslednji pristup 30/10/2019.
[19] TensorFlow [Online]. Dostupno na: https://www.tensorflow.org, poslednji pristup 30/10/2019.

Primena programa MATLAB za simulaciju rada višenivovskih pretvarača za obnovljive izvore energije

Dario Javor

Sadržaj - U ovom radu su prikazani izlazni naponi i struje dvonivovskih i tronivovskih pretvarača koji se koriste u obnovljivim izvorima energije, ali i za mnoge druge primene. Simulacija njihovog rada je izvršena u programskom paketu MATLAB/Simulink za jednofazne i trofazne invertore, u slučaju korišćenja univerzalnog bloka i za primenu tehnike modulacije SVPWM (Space Vector Pulse Width Modulation).

I. Uvod

Na srednjenaponsku i niskonaponsku mrežu se priključuje veliki broj distribuiranih izvora za proizvodnju električne energije kao što su vetrogeneratori, fotonaponski sistemi i sistemi sa gorivnim ćelijama. Da bi se obnovljivi izvori energije povezali na mrežu i obavilo podešavanje napona i frekvencije, potrebni su različiti tipovi pretvarača. Pretvarači se koriste i za aplikacije kao što su aktivni filtri, statički VAR kompenzatori, elektromotorni pogoni, itd. Prednosti višenivovskih pretvarača u odnosu na dvonivovske su: manja harmonijska distorzija, manja elektromagnetna interferencija, veća efikasnost i rad na više naponskih nivoa. Njihovi nedostaci su potreba za izolovanim izvorima napona napajanja na pojedinim nivoima, složeno upravljanje i veća cena nego za dvonivovske pretvarače. Višenivovska konverzija koristi se u različitim tipovima pretvarača AC/DC, DC/AC, DC/DC i AC/DC/AC, a njihove topologije su prilagođene tipu konverzije. Postoji više topologija invertora među kojima su najčešći višenivovski invertori sa sprežnim diodama, višenivovski invertori sa plivajućim kondenzatorima, kaskadni H-mostni invertori sa odvojenim DC izvorima za napajanje i hibridne topologije [1-4].

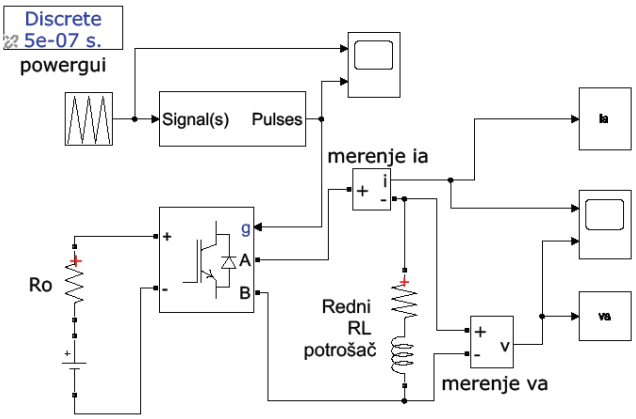
U višenivovskim invertorima koriste se različite modulacione tehnike, kao što su sinusna impulsno-širinska modulacija (Sinusoidal Pulse Width Modulation - SPWM) koja koristi tehniku faznog pomeraja nosećeg signala, tehnika modulacije prostornim vektorom (Space Vector Modulation - SVM) koja radi na višim prekidačkim frekvencijama, tehnike koje rade na nižim frekvencijama kao što su upravljanje prostornim vektorom (Space Vector Control - SVC), selektivna harmonijska eliminacija (Selective Harmonic Elimination - SHE), itd.

U radu je izvršena simulacija rada dvonivovskih i višenivovskih pretvarača u programu MATLAB/Simulink [5]. Tehnike upravljanja se stalno usavršavaju [6-8], a u radu su prikazani rezultati simulacija u slučaju primene SVPWM tehnike (Space Vector Pulse Width Modulation).

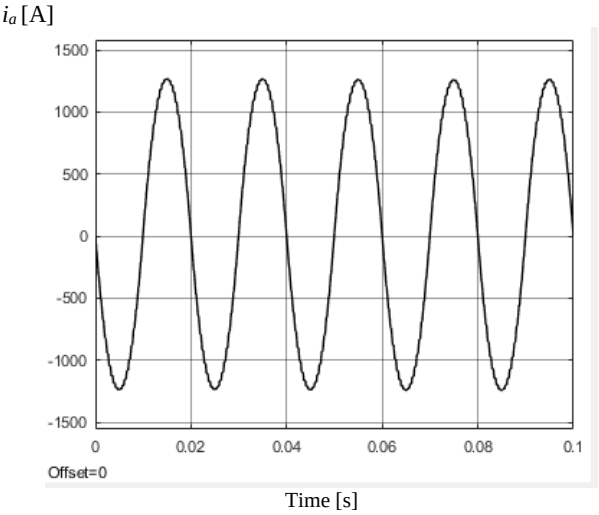
D. Javor, Ph.D. student, studijski program doktorskih studija Elektrotehnika i računarstvo, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dariojavor@mts.rs

II. Simulacija Rada Dvonivovskih Pretvarača

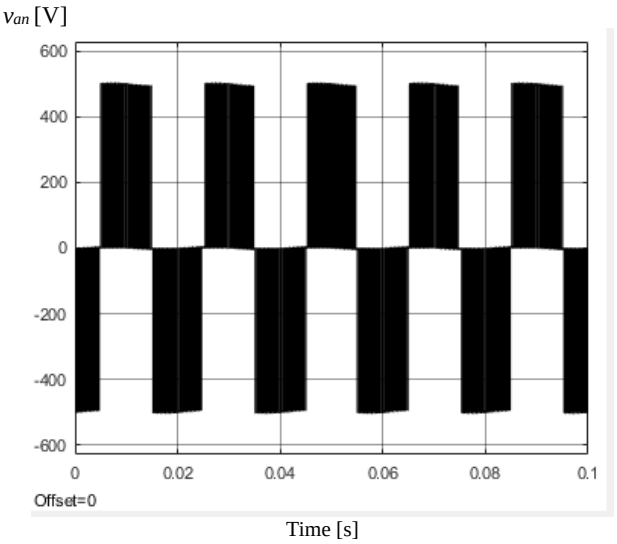
Kolo za simulaciju jednofaznog dvonivovskog pretvarača u MATLAB/Simulink programu je prikazano na Slici 1, izlazna struja na Slici 2.



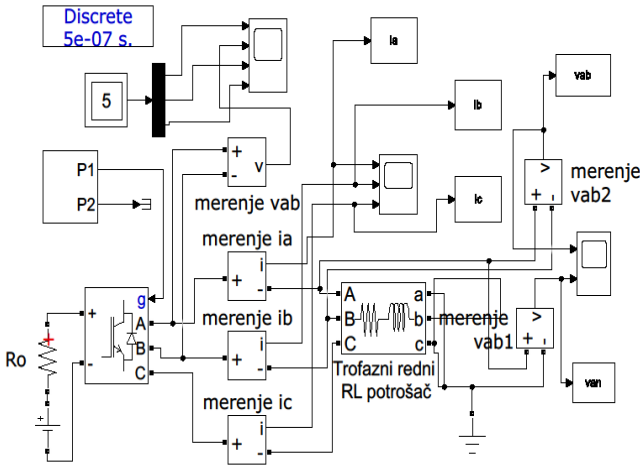
Slika 1. Kolo za simulaciju jednofaznog dvonivovskog pretvarača u MATLAB/Simulink programu s korišćenjem univerzalnog mosta.



Slika 2. Izlazna struja i_a [A] jednofaznog dvonivovskog pretvarača.



Slika 3. Izlazni napon v_{an} [V] jednofaznog dvonivovskog pretvarača.

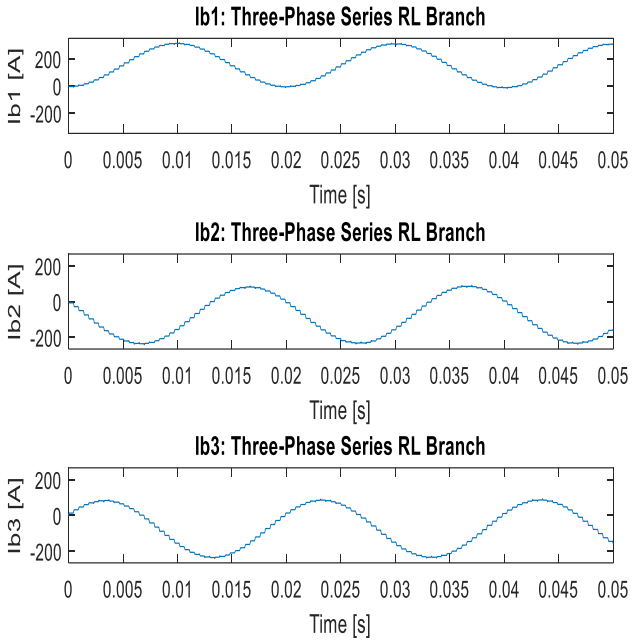


Slika 4. Kolo za simulaciju trofaznog dvonivovskog pretvarača sa korišćenjem univerzalnog bloka u MATLAB/Simulinku.

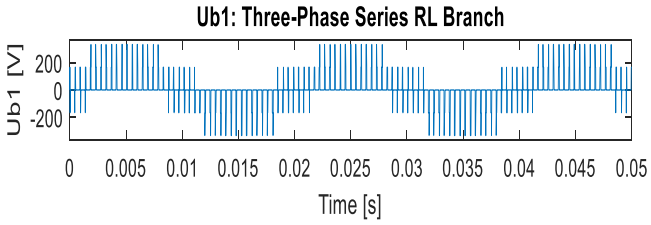
Na blok univerzalnog mosta sa 2 grane dovode se impulsi iz PWM generatorskog bloka (Slika 1).

Izlazni napon ovog jednofaznog dvonivovskog pretvarača sa rednim *RL* potrošačem (R=1mΩ i L=1mH), prikazan je na Slici 3. Parametri kola su V_{dc}=500V, f_b=50Hz i fc=1080Hz.

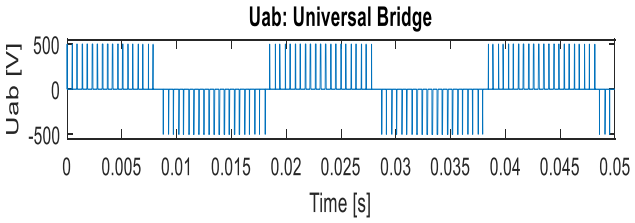
Kolo za simulaciju rada trofaznog dvonivovskog invertora sa korišćenjem univerzalnog bloka u programu MATLAB/Simulink prikazano je na Slici 4. Fazne struje su date na Slici 5, fazni napon na Slici 6, linijski na Slici 7. Parametri kola za simulaciju rada trofaznog dvonivovskog invertora su: V_{dc}=500V, f_b=50Hz, fc=1080Hz, indeks modulacije m_f = 0.2, a za trofazni *RL* potrošač R=1mΩ i L=1mH.



Slika 5. Izlazne fazne struje i_{b1}[A], i_{b2}[A], i_{b3}[A] trofaznog dvonivovskog invertora sa korišćenjem univerzalnog bloka.



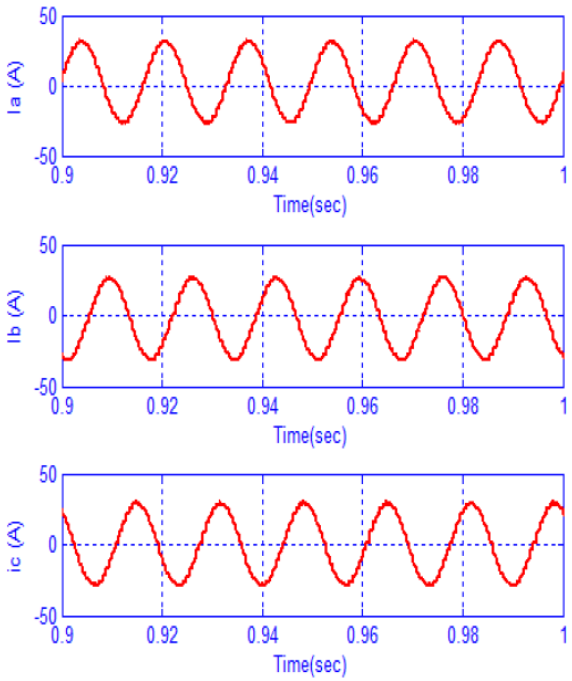
Slika 6. Izlazni fazni napon v_{b1}[V] trofaznog dvonivovskog invertora sa korišćenjem univerzalnog mostnog bloka.



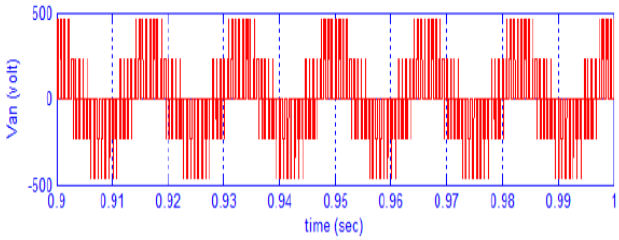
Slika 7. Izlazni linijski napon v_{ab}[V] trofaznog dvonivovskog invertora sa korišćenjem univerzalnog mostnog bloka.

U slučaju korišćenja SVPWM algoritma simulacijom rada trofaznog dvonivovskog invertora [7], za f=60Hz, u programu MATLAB/Simulink dobijaju se izlazne fazne struje kao na Slici 8.

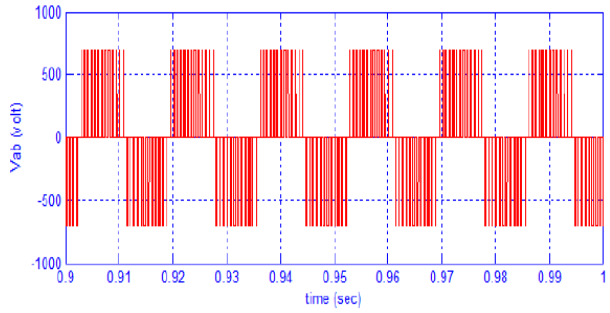
Izlazni fazni napon v_{an} prikazan je na Slici 9, a izlazni linijski napon v_{ab} na Slici 10.



Slika 8. Izlazne fazne struje i_a[A], i_b[A], i_c[A] za trofazni dvonivovski invertor sa korišćenjem SVPWM algoritma.



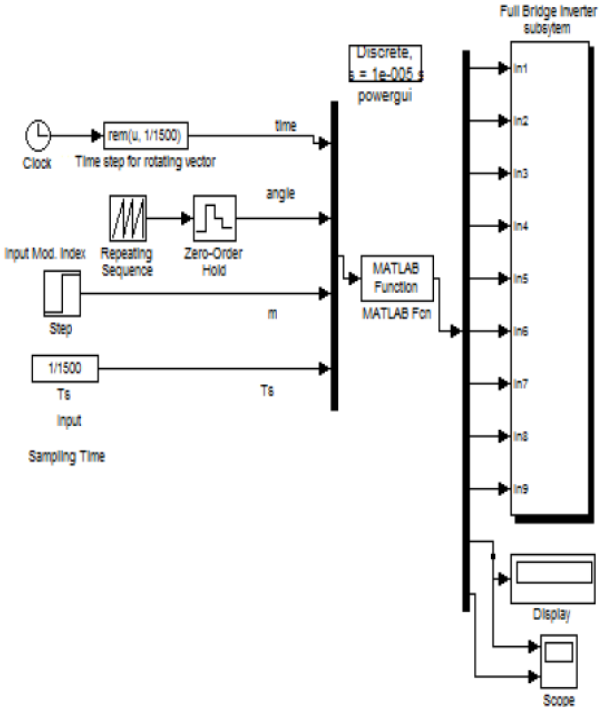
Slika 9. Izlazni fazni napon v_{an} [V] za trofazni dvonivovski invertor sa korišćenjem SVPWM algoritma.



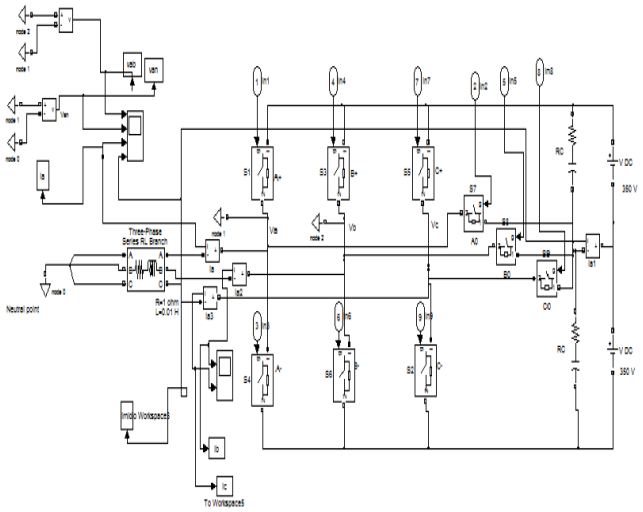
Slika 10. Izlazni linijski napon v_{ab} [V] za trofazni dvonivovski invertor sa korišćenjem SVPWM algoritma.

III. Simulacija Rada Višenivovskih Pretvarača

Kolo za simulaciju rada trofaznog tronivovskog invertora sa korišćenjem SVPWM tehnike u programu MATLAB/Simulink prikazano je na Slici 11, a podsistem kola dat je na Slici 12.



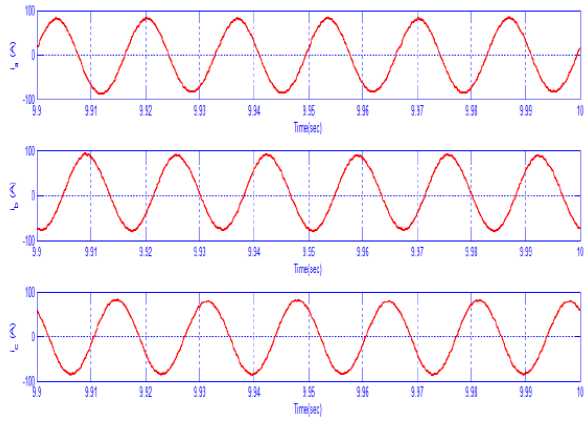
Slika 11. Kolo za simulaciju trofaznog tronivovskog pretvarača sa korišćenjem SVPWM tehnike u MATLAB/Simulinku.



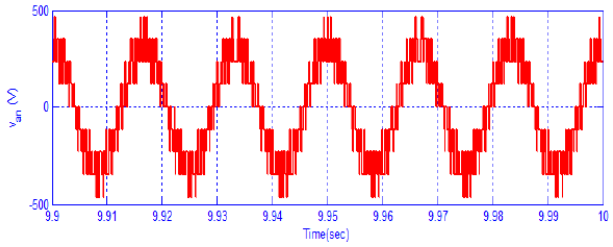
Slika 12. Podsistem univerzalnog mostnog bloka (datog na Slici 11) u kolu za simulaciju trofaznog tronivovskog pretvarača u programu MATLAB/Simulink sa korišćenjem SVPWM tehnike.

Blok MATLAB funkcije na Slici 11 obavlja SVPWM upravljanje, a M-file je dat u [7]. Izlazne fazne struje date su na Slici 13. Izlazni fazni napon v_{an} trofaznog tronivovskog pretvarača sa korišćenjem SVPWM tehnike, prikazan je na Slici 14. Izlazni linijski napon v_{ab} prikazan je na Slici 15 [7]. Na osnovu ovih talasnih oblika može se zaključiti da višenivovski pretvarači daju izlazne napone

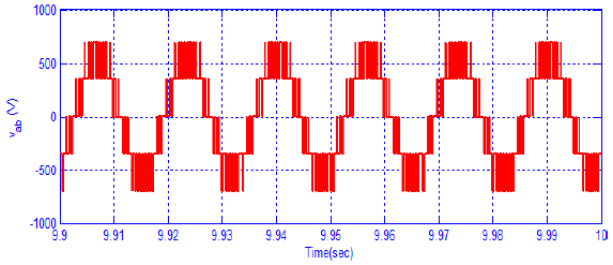
koji imaju manju harmonijsku distorziju nego u slučaju dvonivovskih pretvarača [1].



Slika 13. Izlazne fazne struje za trofazni tronivovski invertor sa korišćenjem SVPWM algoritma.



Slika 14. Izlazni fazni napon za trofazni tronivovski invertor sa korišćenjem SVPWM algoritma.



Slika 15. Izlazni linijski napon za trofazni tronivovski invertor sa korišćenjem SVPWM algoritma.

IV. ZAKLJUČAK

U ovom radu su prikazani rezultati simulacije rada dvonivovskih i tronivovskih pretvarača u programu MATLAB/Simulink. Dati su rezultati za struje i napone jednofaznih i trofaznih pretvarača uz korišćenje univerzalnog bloka kao i za slučaj primene modulacione tehnike upravljanja prostornim vektorom.

Kod višenivovskih pretvarača, za upotrebu SVPWM tehnike, dobijaju se manja harmonijska izobličenja izlaznih napona, veća efikasnost, a manji su i problemi elektromagnetne kompatibilnosti nego kod dvonivovskih pretvarača. Sve šira primena ovih pretvarača zahteva i bolju analizu parametara kola i tehnika modulacije i njihovog uticaja na rad pretvarača.

LITERATURA

[1] Dokić B. L., Blanuša B., Ivanović Ž. "Energetski pretvarači u obnovljivim izvorima", ETF Banja Luka, Akademska misao, Beograd, 2013.

[2] Blaabjerg F., Chen Z., Teodorescu R., Iov F., "Power electronics in wind turbine systems", IPEMC 2006, CES/IEEEE 5th Int. Conference, Vol.1, Aug. 2006, pp. 1-11.

[3] Blaabjerg F., Teodorescu R., Chen Z., Liserre M., "Power converters and control of renewable energy systems", 1st Int. Conference of Energy Inovation, Barcelona, Spain, 2005.

[4] Hansen L. H., Helle L., Blaabjerg F., Ritchie E., Munk-Nielsen S., Bindner H., Sørensen P., Bak-Jensen B., "Conceptual survey of Generators and Power Electronics for wind turbines", Risø National Laboratory, Roskilde, Denmark, Dec. 2001.

[5] https://www.mathworks.com/products/simulink.html

[6] Nordvall A., "Multilevel inverter topology survey", M.Sc. Thesis, Chalmers University of Technology, Göteborg, Sweden, 2011.

[7] Gebreel G. M. A. A., "Simulation and implementation of two-level and three-level inverters by MATLAB and RT-LAB", M.Sc. Thesis, Ohio State University, 2011.

[8] Medina M. L., "Design and control of single-phase modular multilevel converter", M.Sc. Thesis, Universidad de Oviedo, July 2013.

Ispitivanje karakteristika optičkog senzora TCRT5000

Nikola Mitrović

Sadržaj - U ovom radu prikazane su karakteristike optičkog senzora TCRT5000. Detaljnije je prikazan princip rada senzora i dat je opis mogućnosti rada sa senzorom. Neke od osobina senzora su eksperimentalno proverene i prikazane tabelarno i grafički.

I. UVOD

Polje elektronike koje povezuje osnove optike (svetlosnog i drugog zračenja) i osnove elektronike obrazuje optoelektroniku. Jedna od glavnih oblasti proučavanja optoelektronike su optoelektronske komponente. Ove komponente se mogu podeliti u nekoliko grupa (*Light Emission diode* (LED), *Organic Light Emmsion diode* (OLED), solarne ćelije, komponente za lasersku elektroniku, displeji i slično), međutim te podele nisu striktne. Često dolazi do kombinacije nekih osobina zarad dobijanja traženog efekta. Razvojem komponenata, razvio se i spektar primena istih. Opto komponente se najčešće koriste kao indikatori (LED), mada se često sreću i primene kao senzorskog elementa (fotootpornik, fototranzistor i slično). Komponenta koja će biti predstavljena u ovom radu spada u kategoriju senzora i opisuje se kao reflektivni optički senzor sa tranzistorskim izlazom. Konkretan model komponente je TCRT5000 [1].

II. PRINCIP RADA KOMPONENTE

Optički senzor TCRT5000 je komponenta koje je nastala kao spoj dve komponente, infracrvene diode i fototranzistora (slika 1).



Slika 1. Optički senzor TCRT5000.

Infracrvene diode (*Infra Red diode*) spadaju u grupu LE dioda. Za razliku od standardnih LE dioda koje prilikom vodenja emituju svetlost iz opsega vidljivog zračenja, IR diode takode emituju svetlost, ali svetlost talasne dužine koje ljudsko oko ne može da detektuje. Standarni napon vodenja IR diode je oko 1,3 V. Princip rada ovog tipa diode, kao i polarizacija, identični su principu rada bilo koje druge LE diode.



Slika 2. Simbol IR diode (isti je kao simbol za LED).

Fototranzistor je tip bipolarnog tranzistora koji pripada grupi optoelektronskih komponenata [2]. Za razliku od standardnog bipolarnog tranzistora, fototranzistor se realizuje tako da mu je oblast baze izložena dejstvu upadne svetlosti. Takode, ova oblast se projektuje tako da ima veću površinu nego kod standarnih bipolarnih tranzistora. Tranzistor se polariše tako da kolektor bude na višem pozitivnom potencijalu u odnosu na emitor. Prilikom dejstva upadne svetlosti na bazu tranzistora, dolazi do procesa fotogeneracije nosilaca, odnosno do procesa stvaranja parova elektron-šupljina. Zbog opisane polarizacije spojeva kolektor-baza i baza-emitor, generisani elektroni će driftovati ka kolektoru. Na ovaj način, ostvaruje se usmereno kretanje naelektrisanih čestica, odnosno, tok struje. Smer kretanja struje kroz kolo je obrnut smeru kretanja elektrona, pa struja teče od kolektora ka bazi. Generalno, princip rada bipolarnog tranzistora ukazuje na pojačanje struje baze. U slučaju fototranzistora, efektivno, struja baze ne postoji. Postoji samo fotostruja, pa se samo ona i pojačava. Praktično, fototranzistor pojačava struju koja je nastala kao posledica generacije nosilaca u oblasti baze usled dejstva upadne svetlosti.



Slika 3. Simboli fototranzistora u kolima.

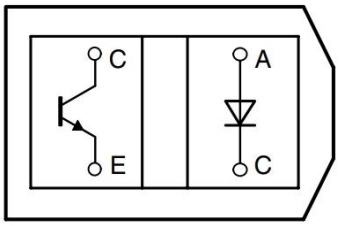
Fototranzistori se često projektuju tako da oblast baze bude osetljiva samo na zračenje odredenih talasnih dužina. Konkretan primer je baš u realizaciji kola kao što je TCRT5000. Fototranzistor koji čini sastavni deo ovog kola je projektovan tako da mu oblast baze bude osetljiva

N. Mitrović je student doktorskih studija pri katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikola.mitrovic@elfak.ni.ac.rs

na infracrveno zračenje. Na ovaj način, neće doći do pobudivanja tranzistora prilikom interakcije sa drugim tipovima zračenja, već samo sa onim koje emituje IR dioda. Sličan princip iskorišćen je i kod projektovanja komponente optokapler (*optocoupler*) [3].

III. OSNOVNO KOLO KOMPONENTE

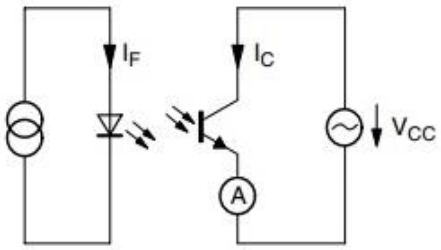
Kao i svaka druga komponenta, i ova poseduje svoj simbol za šeme električnih kola. Simbol senzora je prikazan na slici 4.



Slika 4. Simbol komponente TCRT5000.

Tačke na simbolu predstavljaju izvode komponente, pa se može zaključiti da ova komponenta ima 4 izvoda, odnosno 4 pina. Simbol je proistekao iz simbola za IR diodu i fototranzistor, koji su prikazani ranije u radu. Takode, karakteristika ovog tipa pakovanja je zasečenost pravougaonika na simbolu sa jedne strane. Ta strana predstavlja stranu IR diode. Slična nepravilnost u obliku biće primenjena i prilikom tehnološke realizacije pakovanja komponente. Sa koje strane pakovanja se nalaze izvodi koje od podkomponenata najlakše se utvrduje baš po toj nepravilnosti.

Iz simbola komponente moguće je naslutiti formu osnovnog kola. Šema osnovnog kola komponente TCRT5000 je prikazana na slici 5.

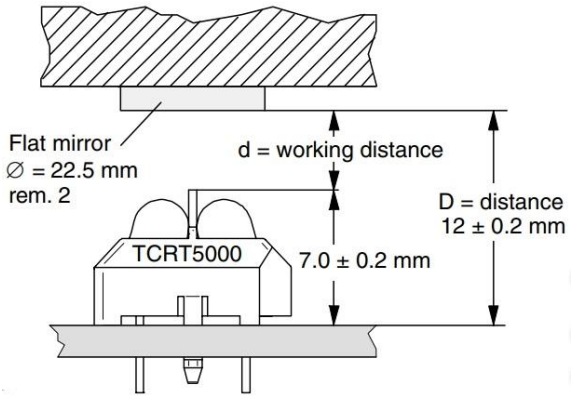


Slika 5. Šema osnovnog kola TCRT5000.

Kao što je navedeno, kolo se sastoji iz IR diode i fototranzistora koji je projektovan tako da bude osetljiv na infracrveno zračenje. Kada IR dioda vodi, emtiuje zračenje koje pobuđuje tranzistor.

Prilikom ovakvog prikaza kola moguće je medutim doći do zaključka da će bilo kakvo vođenje diode prouzrokovati vođenje tranzistora. Ukoliko to zaista jeste slučaj, ne bi bilo potrebe povezivati i tranzistor, jer on zapravo samo prepisuje stanje diode. Ovakav zaključak načelno nije pogrešan, ali prilikom analize TCRT5000, potrebno je obratiti pažnju na još jedan element.

Kao što se može videti sa slike 6, enkapsulacija IR diode i enkapsulacija foto tranzistora radvojeni su crnom neprobojnom preprekom.



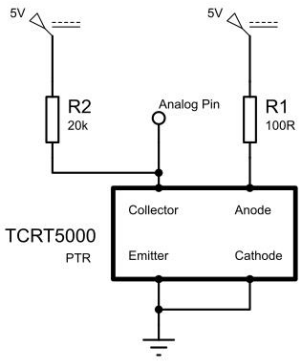
Slika 6. Princip rada kola TCRT5000.

Na ovaj način, ukoliko je cilj ostvariti pobudivanje fototranzistora preko pobuđivanja IR diode, potrebno je da se ispred komponente nade telo koje može da reflektuje IR zračenje. Na taj način, fototranzistor se pobuđuje ne zračenjem koje dolazi direktno sa IR diode, već reflektovanim zračenjem, u interakciji sa preprekom.

IV. POVEZIVANJE SENZORA U ELEKTRONSKIM KOLIMA

Prilikom opisa povezivanja ovakvih senzorskih elemenata, obično bude uvedena pretpostavka da se misli na povezivanje senzora sa nekim kompleksnijim intergisanim kolima, najčešće sa nekim mikrokontrolerom.

Na diodnoj strani kola, potrebno je povezati otpornik redno sa diodim. Ovaj otpornik ograničiće struju kroz diodu, i na taj načon sprečiće se oštećenje i otkaz komponente. Dioda je polarisana direktno.



Slika 7. Osnovno povezivanje senzora TCRT5000.

Deo kola sa fototranzistorom sastoji se od fototranzistora u konfiguraciji sa zajedničkim emitorom.

Tranzistor u ovom slučaju je konfigurisan da radi kao prekidač. Zadatak tranzistora je da za detektovanje IR zračenja (prijem signala), promeni stanje koje se čita na izlazu. Kada tranzistor ne vodi, napon logičke jedinice se čita sa izlaza (*Analog Pin*). Kada tranzistor provede, kolo se zatvori prema masi, pa nema napona logičke jedinice na izlazu, pa se smatra da je na izlazu logička nula. Naravno, ni logička jedinica ni logička nula nisu konkretno 5 V odnosno 0 V, ali je razlika izmedu ova dva signala dovoljno velika da jedan od njih bude smatran za logičku jedinicu, a drugi za logičku nulu.

V. ISPITIVANJE KARAKTERISTIKA SENZORA

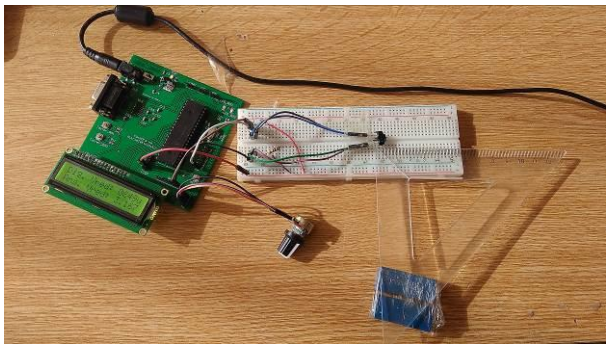
Za ispitivanje karakteristika komponente, prvo je potrebno oformiti odgovarajuće radno okruženje. Treba ispitati odziv tranzistora pri različitim udaljenostima prepreke i prilikom različitih otpornosti povezanih sa kolom.

Način provere odziva tranzistora biće preko merenja napona logičke jedinice odnosno logičke nule prilikom upadnog IR zračenja. Kako je ranije napomenuto, ovi naponi ne predstavljaju striktno napone od 5 V odnosno 0 V. Ti naponi predstavljaju dva različita naponska nivoa, medu kojima postoji dovoljno velika naponska razlika da bi se jedan smatrao jedinicom, a drugi nulom.

A. Povezivanje Radnog Okruženja

Za potrebe merenja iskorišćeno je okruženje oko mikrokontrolera PIC16F877A [4]. Ovo okruženje imaće funkciju voltmetra. Merenje napona pomoću ovog mikrokontrolera biće vršeno korišćenjem postupka analogno-digitalne konverzije (ADC) [5]. Na displeju će se ispisivati digitalni nivoi merenih vrednosti, kao i vrednosti napona. Udaljenost senzora od prepreke biće merena lenjirom.

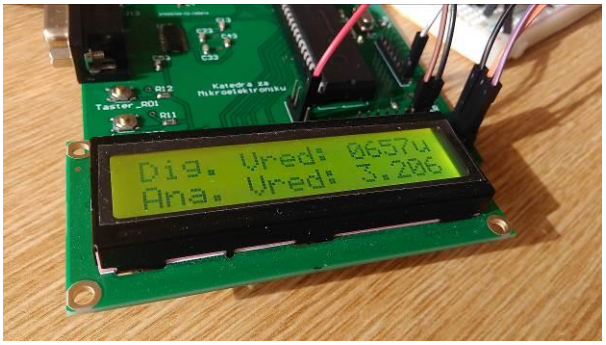
Prva grupa merenja biće izvršena za vrednosti elemenata u šemi koje odgovaraju uputstvima iz tehničke dokumentacije. Ova šema prikazana je na slici 7 (otpornost u grani kolektora R_C = 4.7 kΩ, otpornost u kolu diode R_D = 100 Ω.



Slika 8. Povezano okruženje za rad.

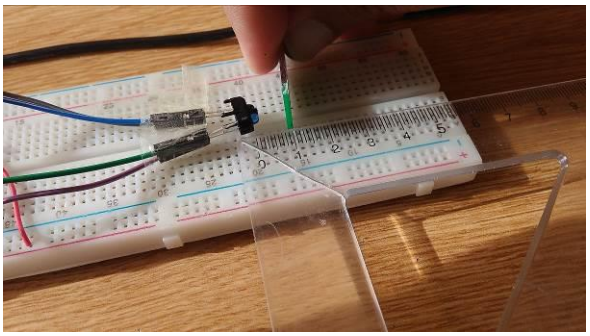
B. Rezultati merenja

Prvo je potrebno izmeriti napon na izlazu TCRT5000 (napon sa kolektora fototranzistora). Ovaj napon predstavlja napon logičke nule. Izmerena je digitalna vrednost od 669, što odgovara analognoj vrednosti od 3.3 V.



Slika 9. Očitavanje vrednosti sa displeja.

Nakon toga, počinje merenje sa preprekom. Prepreka je prvo postavljena na udaljenost od 5 cm od površine senzora. Očitani napon je 2.93 V. Udaljenost prepreke od površine senzora se onda smanjuje za po jedan centrimetar, sve dok prepreka ne dodirne površinu.



Slika 10. Merenje sa preprekom ispred TCRT5000.

Rezultati merenja dati su u tabeli I, a onda i naknadno objašnjeni.

TABELA I
ZAVISNOST PROMENE NAPONA OD UDALJENOSTI PREPREKE (R_C = 4.7 kΩ)

	Dig. vr:	An. vr: [V]
d = 0.5 cm	38	0.180
d = 1 cm	145	0.702
d = 2 cm	299	1.480
d = 3 cm	441	2.171
d = 4 cm	502	2.449

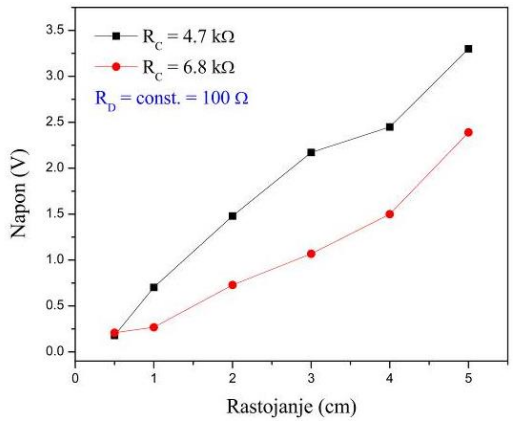
Izmereni rezultati pokazuju da sa smanjenjem rastojanja od prepreke dolazi do smanjenja izlaznog napona

kola. Izmedu dobijenih napona za rastojanje od 1 cm i rastojanje od 3 cm postoji velika razlika kojom je moguće detektovati udaljenost od prepreke. Merenja su ponovljena za drugu vrednost otpornosti R_C, koja prilikom drugog merenja iznosi 6.8 kΩ. Rezultati merenja prikazani su u tabeli II.

TABELA II
ZAVISNOST PROMENE NAPONA OD UDALJENOSTI PREPREKE (R_C = 6.8 KΩ)

	Dig. vr:	An. vr: [V]
d = 0.5 cm	43	0.209
d = 1 cm	55	0.268
d = 2 cm	152	0.729
d = 3 cm	218	1.063
d = 4 cm	315	1.537

Prilikom povećanja otpornosti kolektora fototranzistora dolazi do slabijih prekidačkih funkcija komponente u pogledu opsega. Pri rastojanju od 4 cm, vrednost izlaznog napona prilikom prvog merenja je gotovo duplo veća od vrednosti prilikom drugog merenja. Ova činjenica ne mora direktno da upućuje na to da se TCRT5000 ne može koristiti sa otpornošću R_C = 6.8 kΩ, već da takva primena nudi finija podešavanja na manjem opsegu. Rezultati su prikazani i grafički.



Slika 11. Zavisnost izlaznog napona senzora od otpornosti i rastojanja senzora od prepreke.

Grafik funkcije napona pokazuje visok stepen sličnosti sa bilo kojom drugom prekidačkom funkcijom. To je posledica konfiguracije tranzistora u načinu rada prekidača. Na slici 11 su prikazane karakteristike kola TCRT5000 koje su merene eksperimentalno za odreden opseg otpornosti R_C.

VI. ZAKLJUČAK

Opisani eksperiment pokazuje devijaciju izlaznog napona senzora u zavisnosti od povezane otpornosti. Mogućnost primene ove vrste optičkih senzora je jako velika (*proximity sensor*, odnosno senzor blizine, zatim brojač rotacije rotora oko osovine i slično). Svaka od praktičnih realizacija koje uključuje povezivanje ovog senzora zahteva teorijsko poznavanje principa rada komponente i karakteristika komponente koje su opisane u radu.

ZAHVALNICA

Autor se ovom prilikom zahvaljuje mentor prof. dr Danijelu Dankoviću i prof. dr Vesni Paunović sa Katedre za mikroelektroniku za podsticanje ideje i podršku prilikom realizacije rada. Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026.

LITERATURA

[1] TCRT5000 datasheet, Vishay. Dostupno na: https://www.vishay.com/docs/83760/tcrt5000.pdf.
[2] Z. Prijić i A. Prijic, „*Uvod u poluprovodnicke komponente*“, Elektronski fakultet, Univerzitet u Nišu, februar 2014.
[3] D. Danković, „*Laboratorijski praktikum sa zadacima za samostalan rad iz predmeta Komponente za telekomunikacije*“, Elektronski fakultet, Univerzitet u Nišu, 2018.
[4] PIC16F877A datasheet. Dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/395.pdf
[5] Z. Prijić, „Analogna Mikroelektronika“ slajdovi, Elektronski Fakultet, Univerzitet u Nišu, 2019.

Primena Arduina i ultrazvučnog senzora u izradi radarskog sistema

Milivoje Mihajlović

Sadržaj – Radar je sistem za otkrivanje objekata koji koristi radio talase za određivanje udaljenosti, nadmorske visine, smera ili brzine objekata. Radarska antena emituje impulse radio talasa ili mikrotalasa koji se odbijaju od predmeta na njihovoj putanji. Ovaj projekat ima za cilj ptaktičnu realizaciju radara koji će verno prikazati princip rada profesionalnih radara.

I. UVOD

Radar je sistem za otkrivanje objekata koji koristi radio talase za određivanje udaljenosti, nadmorske visine, smera ili brzine objekata. Radarski sistemi se isporučuju u različitim dimenzijama i imaju različite performanse. Neki radarski sistemi koriste se za kontrolu vazdušnog saobraćaja na aerodromima, a drugi se koriste za nadzor raketa dugog dometa i sisteme ranog upozoravanja. Radarski sistem je srce sistema navođenja raketa. Na raspolaganju su mali prenosni radarski sistemi koje jedna osoba može da održava i upravlja njima, kao i sistemi koji zauzimaju nekoliko velikih prostorija.

Radar je tajno razvijan od strane nekoliko nacija pre i tokom Drugog svetskog rata. Sam termin RADAR nastao je 1940. godine od strane mornarice Sjedinjenih Američkih Država kao akronim za radio detekciju i domet (Radio Detection and Ranging).

Savremena upotreba radara veoma je raznolika, koristi se u kontroli vazdušnog saobraćaja, astronomiji, sistemima protivvazdušne odbrane itd.. Radarski sistemi visokih tehnologija povezani su sa digitalnom obradom signala i sposobni su da obrade korisne informacije i pored visokog nivoa šuma.

II. KOMPONENTE SISTEMA

A. Arduino Uno

Srž hardverske konfiguracije ovog sistema jeste Arduino Uno, mikrokontroler pristupačne cene i malih dimenzija. Što se njegovih karakteristika tiče, izdvaja se to da koristi 8-bit mikroprocesor, napaja sa 5 V, poseduje 16 digitalnih pinova koji mogu biti ulazni ili izlazni, 6 analognih ulaza, 16MHz kvarcni kristal 32KB FLASH 2KB SRAM, 1KB EEPROM i USB konektor.

M. Mihajlović je student treće godine doktorskih akademskih studija, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milivoje.mihajlovic@yahoo.com



Slika 1. Arduino Uno [1].

TABELA I
ARDUINO UNO SPECIFIKACIJA [2]

Microcontroller		*ATmega328*
1	Operating Voltage	5 V
2	Input Voltage (recommended)	7-12 V
3	Input Voltage (limits)	6-20 V
4	Digital I/O Pins	14 (of which 6 provide PWM output)
5	Analog Input Pins	6
6	DC Current per I/O Pin	40 mA
7	DC Current for 3.3 V Pin	50 mA
8	Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
9	SRAM	2 KB (ATmega328)
10	EEPROM	1 KB (ATmega328)
11	Clock Speed	16 MHz

B. Servo motor

Servomotor je rotacijski aktuator koji omogućava preciznu kontrolu ugaonog položaja, brzine i ubrzanja. Sastoji se od odgovarajućeg motora spojenog sa senzorom za povratnu informaciju o položaju. Takođe zahteva relativno sofisticiran regulator, često namenski modul dizajniran posebno za upotrebu sa servomotorima.

Servomotori imaju primenu u disciplinama kao što su robotika, izrada CNC mašina ili automatizovana proizvodnja.

C. Ultrazvučni senzor

Ultrazvučni senzori rade na principu sličnom radaru ili sonaru koji procenjuju atribute cilja interpretirajući odjeke iz radio ili zvučnih talasa. Ultrazvučni senzori generišu

visokofrekventne zvučne talase i procenjuju odjek koji senzor prima. Senzori izračunavaju vremenski interval između slanja signala i primanja odjeka da bi odredili udaljenost do objekta.

Ova tehnologija se može koristiti za merenje brzine i pravca vetra (anemometar), nivoa rezervoara ili kanala i brzine kroz vazduh ili vodu. Za merenje brzine ili pravca uređaj koristi više detektora i izračunava brzinu od relativnog rastojanja do čestica u vazduhu ili vodi.

Da bi izmerio nivo rezervoara ili kanala, senzor meri rastojanje do površine tečnosti. Ostale aplikacije uključuju: ovlaživače, sonar, medicinsku ultra sonografiju, protivprovalne alarme.

Sistemi obično koriste pretvarač koji stvara zvučne talase u ultrazvučnom opsegu, iznad 18.000 Hz, pretvaranjem signala u zvuk, a potom primanjem eha pretvori zvučne talase u električni signal koji se može meriti i prikazati.



Slika 2. Servo motor [3].



Slika 3. Ultrazvučni senzor HC-SR04 [4].

III. PRAKTIČNA REALIZACIJA

A. Povezivanje servo motora

Servomotor je rotacijski aktuator koji omogućava preciznu kontrolu ugaonog položaja, brzine i ubrzanja.

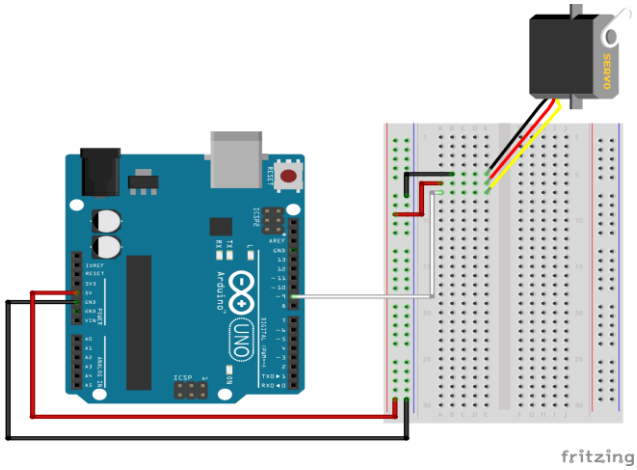
Servo motor najčešće ima tri pina:

1.VCC

2. GND

3. PULSE

Servo motor radi na naponu od 4.8 V do 6 V. Ukupno vreme pulsa servo motora obično je 20 ms. Da bi smo ga pomerili ka jednom kraju, recimo do 0 stepenog ugla, koristimo impuls od 1ms, a za njegovo pomeranje na drugi kraj, tj. 180 stepeni, primenjujemo impuls u trajanju od 2ms. Impulsni pin motora je vezan za PWM pin Arduino-a, preko kog možemo precizno kontrolisati dužinu trajanja impulsa. Dakle, zadavanjem određenog intervala impulsa servo motor se može lako kontrolisati.

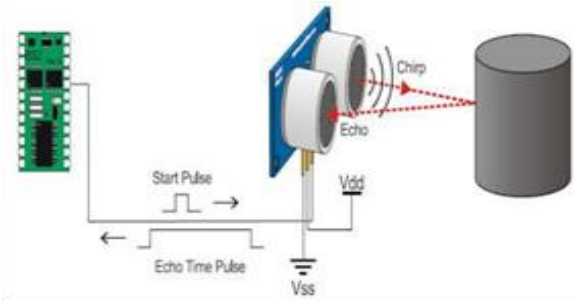


Slika 4. Povezivanje servo motora [5].

B. Povezivanje ultrazvučnog senzora

Ultrazvučni senzor ima tri izlaza. Jedan je za VCC, drugi za GND i treći za pulsni signal. Ultrazvučni senzor je montiran na servo motor i oba su dodatno povezani na Arduino ploču. Ultrazvučni senzor koristi princip refleksije za svoj rad. Kada je povezan sa Arduinom, Arduino daje impulsni signal ultrazvučnom senzoru koji potom šalje ultrazvučni talas u smeru napred. Dakle, kad god je bilo koja prepreka otkrivena ili prisutna ispred, od nje se odbijaju talasi koje prima ultrazvučni senzor.

Ako je otkriven, signal se šalje Arduino-u, a potom PC-u/laptopu na kom se nalazi softver za obradu koji pokazuje prisustvo prepreke na rotirajućem RADAR ekranu sa rastojanjem i uglom pod kojim je detektovana prepreka.

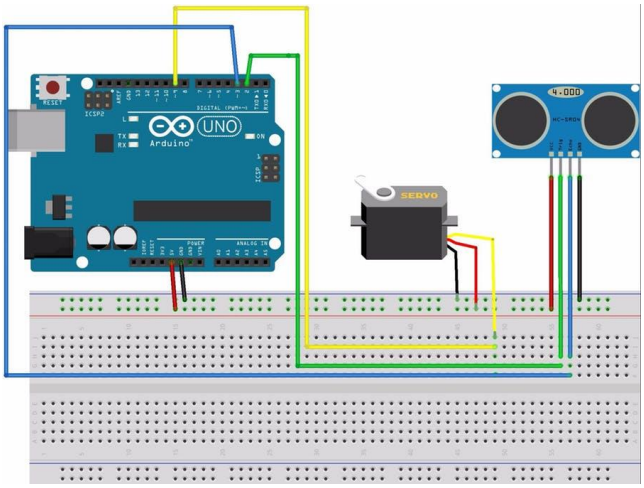


Slika 5. Princip rada ultrazvučnog senzora [6].

C. Električna šema projekta u Fritzing software-u

U duhu razvoja Arduino mikrokontrolera, univerzitet u Potsdamu razvio je vrlo zanimljivu platformu Fritzing [7]. Fritzing je open-source EDA software namenjen studentima i ljudima koji se elektronikom bave iz hobija.

Savršen je alat za dizajnere, inovatore i edukatore za kreiranje prototipa ili čak izrade PCB pločica.



Slika 6. Električna šema projekta kreirana u Fritzing-u.

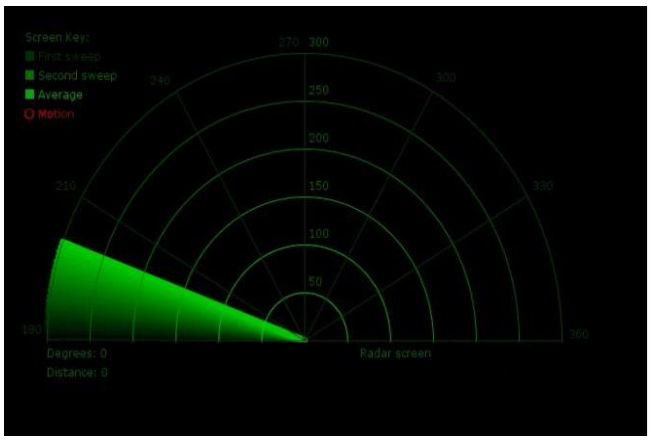
IV. PROCESSING SOFTWARE

Processing je open-source programski jezik i integrisano razvojno okruženje (IDE) koji se koristi pri izradi grafičkog interfejsa. Projekat su pokrenuli 2001. godine Casey Reas i Benjamin Fry, obojica iz odseka za estetiku i programiranje u laboratoriji MIT-a (Massachusetts Institute of Technology). Jezik se temelji na Java programskom jeziku, ali koristi pojednostavljene modele sintakse i grafičkog programiranja.



Slika 7. Processing software [8].

Nakon završenog programiranja grafičkog interfejsa i povezivanja sa Arduinom korisćenjem serijske komunikacije, dobija se radarska slika prikazana na slici 8.



Slika 8. Izgled radarskog ekrana koriscenjem Processing software-a.

V. ZAKLJUČAK

Ovaj projekat ima za cilj korišćenje ultrazvučnog senzora povezanog na Arduino UNO razvojnu ploču koja signal od senzora aproksimira na ekran laptopa za merenje prisustva bilo koje prepreke ispred senzora, kao i odredjivanje razdaljine i ugla pod kojim senzor otkriva prepreku.

Moguće je dodati LED diode različitih boja koje bi se uključivale i isključivale u zavisnosti od razdaljine detektovanog predmeta, kao i buzzer koji bi adekvatnom jačinom zujanja pratio diode koje su uključene.

ZAHVALNICA

Ovaj rad je podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije kroz projekat TR-32026.

LITERATURA

[1] http://www.arduino.cc/
[2] http://www.atmel.com/atmega328/
[3] https://www.arduino.cc/en/reference/servo
[4] https://components101.com/ultrasonic-sensor-working-pinout-datasheet
[5] https://www.allaboutcircuits.com/projects/servo-motor-control-with-an-arduino/
[6] https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/
[7] https://fritzing.org/home/
[8] https://processing.org/