Upravljanje jednom granom trofaznog izmjenjivača sinusno pulsnom-širinskom moduluacijom pomoću razvojnog sustava Arduino ili Rasrberry Pi

Galić, Petar

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:261267

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Upravljanje jednom granom trofaznog izmjenjivača sinusno pulsnom-širinskom modulacijom pomoću razvojnog sustava Arduino ili Raspberry Pi

Diplomski rad

Petar Galić

Osijek, 2020.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 16.09.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Petar Galić
Studij, smjer:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. studenta, godina upisa:	D-18ARK, 23.09.2019.
OIB studenta:	97000818355
Mentor:	Prof. dr. sc. Denis Pelin
Sumentor:	Andrej Brandis
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Davor Vinko
Član Povjerenstva 1:	Prof. dr. sc. Denis Pelin
Član Povjerenstva 2:	Toni Varga
Naslov diplomskog rada:	Upravljanje jednom granom trofaznog izmjenjivača sinusno pulsnom-širinskom moduluacijom pomoću razvojnog sustava Arduino ili Rasrberry Pi
Znanstvena grana rada:	**Elektrostrojarstvo (zn. polje elektrotehnika)**
Zadatak diplomskog rada:	Pomoću razvojnog sustava Arduino osmisliti i adaptirati sustav za upravljanje jednom granom trofaznog izmjenjivača (jednofazni izmjenjivač u polumosnom spoju). Upravljanje izvesti sinusnom pulsno-širinskom modulacijom (SPWM). Provesti analizu rada za dani izmjenjivač upravljan pomoću SPWM. Također, dizajnirati korisničko sučelje (Arduino monitor i tipkovnica) za unos i prikaz parametara SPWM. Testirati nadograđeni izmjenjivač sa radno-induktivnim trošilom u nekoliko karakterističnih radnih točaka. Sumentor: Andrej Brandis
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	16.09.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 23.09.2020.

Ime i prezime studenta:	Petar Galić
Studij:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. studenta, godina upisa:	D-18ARK, 23.09.2019.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Upravljanje jednom granom trofaznog izmjenjivača sinusno pulsnom-širinskom moduluacijom pomoću razvojnog sustava Arduino ili Rasrberry Pi**

izrađen pod vodstvom mentora Prof. dr. sc. Denis Pelin

i sumentora Andrej Brandis

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Želim zahvaliti roditeljima, prijateljima i obitelji Jukic na pruženoj podršci kroz moje akademsko obrazovanje.

Želim zahvaliti mentoru, prof. dr. sc. D. Pelinu te asistentima A. Brandisu i I. Biondicu na pomoci pri izradi diplomskog rada i makete.

Every journey begins with a single step. – Viktor Reznov

Sadržaj

1. Uvod ... 1
2. Pregled područja rada ... 2
3. Izmjenjivač ... 3
3.1. Jednofazni izmjenjivači ... 3
3.2. Trofazni izmjenjivači ... 6
3.3. Polumosni spoj ... 8
3.4. Pulsno-širinska modulacija ... 9
4. Implementacija ... 13
4.1. Logika rješenja ... 13
4.2. Komponente ... 14
4.3. Shema sklopovlja ... 16
4.4. Objašnjenje koda ... 17
4.5. Typhoon HIL ... 20
4.6. UniTrain i harmonijska analiza ... 27
5. TESTIRANJE ... 28
5.1. Pokretanje programa ... 28
5.2. Harmonijska analiza izlaznog napona ... 30
5.3. Testiranje ... 34
5.4. Analiza dobivenih rezultata ... 37
5.5. Nedostatci predloženog upravljanja ... 45
5.6. Prijedlog rješenja nedostatka pri sljedećoj doradi makete ... 47
6. ZAKLJUČAK ... 49
LITERATURA ... 50
SAŽETAK ... 51
ABSTRACT ... 51
ŽIVOTOPIS ... 53

PRILOZI 54
Prilog 1 : Arduino kod 54
Prilog 2 : Slike makete i radnog okruženja 61

1. UVOD

Povezivanje istosmjernih izvora i izmjeničnih trošila oduvijek je bio jedan od važnih zahtjeva energetske elektronike. Razvojem iste razvili su se izmjenjivači, koji su značajno promijenili način povezivanja ovih sustava. Danas se uređaji energetske elektronike koriste u svakodnevnom životu, uključujući sustave za proizvodnju, pretvorbu i prijenos električne energije, kao i napajanje u telekomunikacijama, kućanstvu i slično. Povećanjem upotrebe energetske elektronike, došlo je do povećanog razvoja, što za rezultat ima povećanu korisnost, smanjene dimenzije i cijene proizvodnje elektroničkih energetskih pretvarača.

U ovom radu je opisano upravljanje jednom granom trofaznog, autonomnog, naponskog izmjenjivača koji je upravljan pulsno širinskom modulacijom. Za izvršenje upravljanja, upotrijebljen je Arduino mikrokontroler. Prikazan je spoj mikrokontrolera unutar sheme spoja i izvršeno je testiranje unutar didaktičko-mjernog okruženja TYPHOON HIL i Lucas Nülle.

2. PREGLED PODRUČJA RADA

Arduino je „open-source“ platforma koja se temelji na jednostavnosti razvoja kako programske podrške, tako i sklopovlja. Ova platforma se može koristiti za rješavanje jednostavnih problema poput uključenja i isključenja LED dioda sve do kompleksnih problema poput izrade robotske ruke, električnih bicikala i slično. Arduino je popularno rješenje zbog obilja biblioteka za rad ss periferijom, do aktivne zajednice koja stalno unaprjeđuje i dodaje nova rješenja, biblioteke, sučelje i slično [1].

Matija Pavin [2] u diplomskom radu opisuje upravljanje autonomnog trofaznog naponskog izmjenjivača pulsno širinskom modulacijom. Koristi integrirani krug SA828, s kojim dobiva zadovoljavajuće rezultate, te predlaže poboljšanje korištenjem Arduino mikrokontrolera, na čemu je temeljen ovaj diplomski rad.

Veljko Špica u diplomskom radu [3] upravlja granom autonomnog izmjenjivača putem integriranog kruga SA8281. Zaključuje da se problem značajnijeg udjela trećeg harmonika može izbjeći korištenjem SA8282 integriranog kruga.

A.E. TOUBAL MAAMAR [4] je upotrebio Arduino ATmega328 mikrokontroler za upravljanje elektroničkim vratima jednofaznog rezonantnog izmjenjivača. Napravio je eksperimentalnu ploču kojom upravlja Arduino, te ona mijenja stanje unutar RLC rezonantnog kruga. Stanje u krugu je pratio osciloskopom. Predlaže daljnje istraživanje drugih topologija rezonantnih krugova ili upotrebu drugih mikrokontrolera.

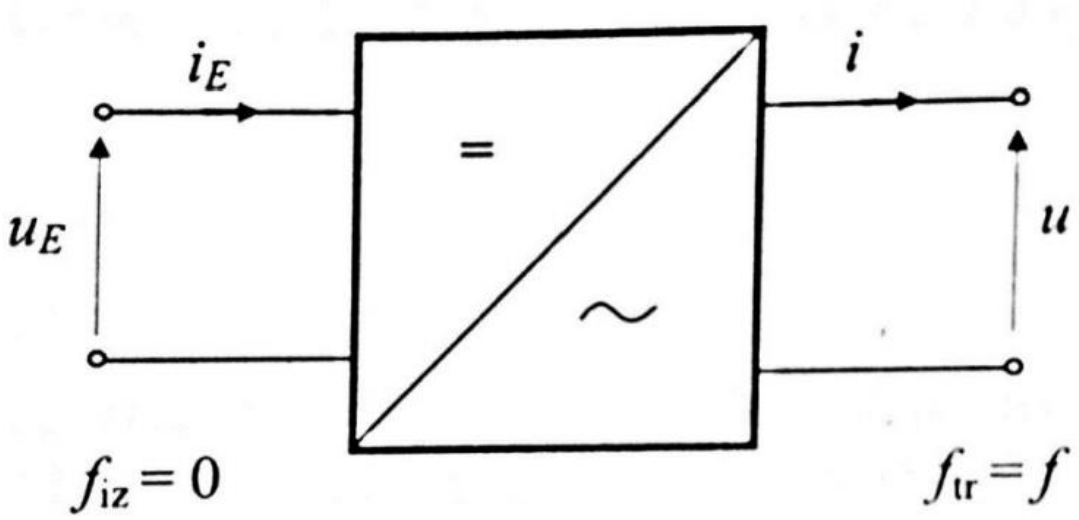
S.A.Zulkifli je u svom radu [5] testirao Arduino kao jeftini mikrokontroler za upravljanje trofaznim izmjenjivačem. Koristeći Arduino, moguće je upravljati kompleksnijim kontrolerom, kao što je „PI-Repetitive“ kontroler, te je zaključio da je MATLAB dobar način za implementiranje PŠM modulacije zbog zadovoljavajuće razine programskog jezika.

Shubham Pal u radu [6] opisuje solarnu mikro-mrežu unutar Typhoon HIL programskog paketa. Korišteno sklopovlje je Typhoon HIL 402. Nakon modeliranja unutar sučelja Typhoon HIL-a, te testiranja istog, Pal zaključuje da je programski paket sasvim zadovoljavajući za projektiranje solarnih fotonaponskih sustava.

3. IZMJENJIVAČ

3.1. Jednofazni izmjenjivači

Izmjenjivači su sklopovi koji pretvaraju istosmjerni ulazni napon u izmjenični izlazni napon. Autonomni izmjenjivač je vrsta izmjenjivača koja ne ovisi o frekvenciji priključene mreže, nego stvara vlastitu, autonomnu mrežu na izlazu. Koristi se u trošilima kojima je potrebna promjena frekvencije [7].



Slika 3.1. Simbol jednofaznog izmjenjivača

Na slici 3.1. je prikazan najjednostavniji prikaz izmjenjivača. Na ulazu je istosmjerni napon u_E s odgovarajućom strujom i_E. Frekvencija ulaza je 0. Na izlazu iz sklopa se dobiva izlazni napon u sa odgovarajućom strujom i, te frekvencijom f_{tr} koja je definirana sklopnom frekvencijom izmjenjivača.

Topologija izmjenjivača može biti mosni i polumosni spoj. Mosni spoj se sastoji od 4 pretvaračke komponente, za razliku od polumosnog koji ima dvije pretvaračke komponente, koje se sastoje od IGBT-a i njemu antiparalelno spojene diode. Time se osigurava punoupravljivost, jer struja u jednom smjeru teče kroz IGBT, dok u drugom kroz njemu antiparalelno spojenu diodu. Mosni spoj i njegovi karakteristični valni oblici pri analizi modela pretvaračkog sklopa su prikazani na slici 3.2. i 3.3 gdje su:

x_{vn} – upravljački impulsi pretvaračkih komponenata

u_{izl} – izlazni napon

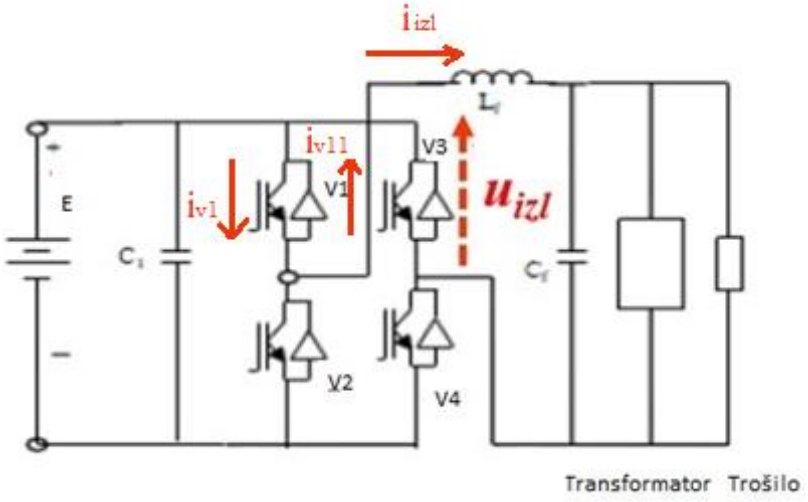
i_{izl} – izlazna struja

E – napon izvora

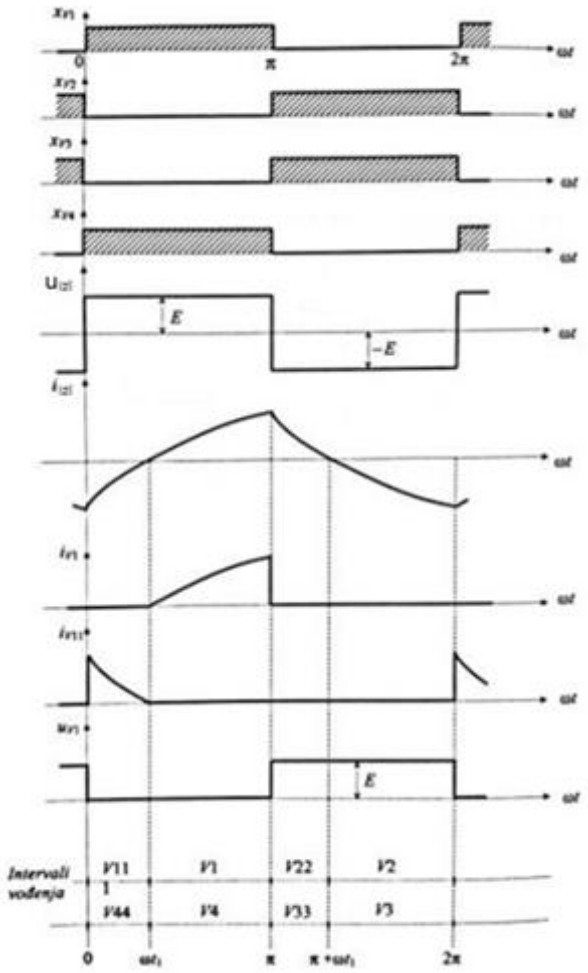
i_{vn} – struja ventila

i_{vnn} – struja antiparalelne diode ventila

u_{vn} – napon ventila

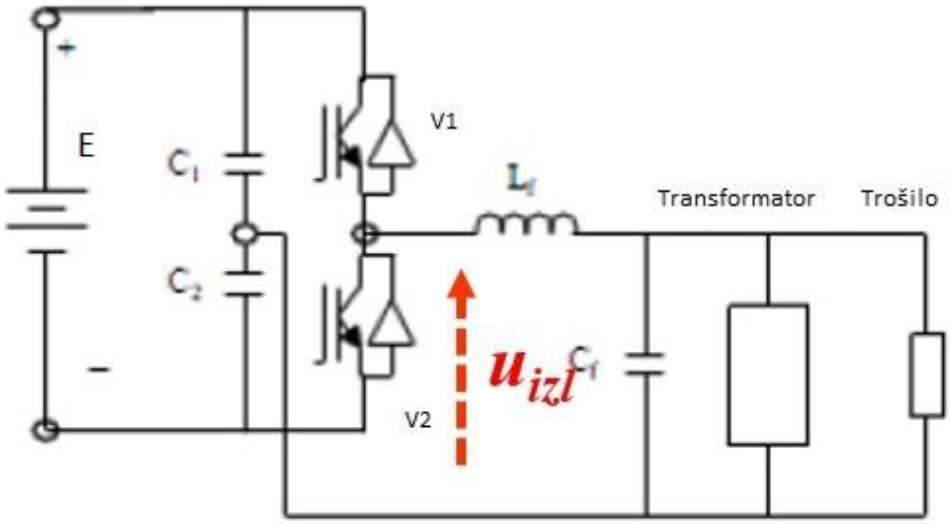


Slika 3.2. Shema mosnog spoja autonomnog naponskog izmjenjivača



Slika 3.3. Karakteristični valni oblici mosnog spoja

Polumosni spoj se sastoji od dvije pretvaračke komponente, te srednje točke na koju se priključuje trošilo. Srednja točka se postiže priključivanjem dva kondenzatora spojenih u seriju tako da čine naponsko djelilo. Nakon toga, trošilo se priključuje između kondenzatora. Prednosti polumosnog spoja očituju se u manje pretvaračkih komponenata i lakšem upravljanju, dok mosni spoj posjeduje mogućnost višerazinskog izlaznog napona i struje, te time i manje harmonijsko izobličenje (THDU I THDI). Valni oblici struje i napona su isti kao na slici 3.3, uz drugačije upravljanje pretvaračkih komponenti [7]. Na slici 3.4. prikazana je shema polumosnog spoja.



Slika 3.4. Shema polumosnog spoja autonomnog izmjenjivača

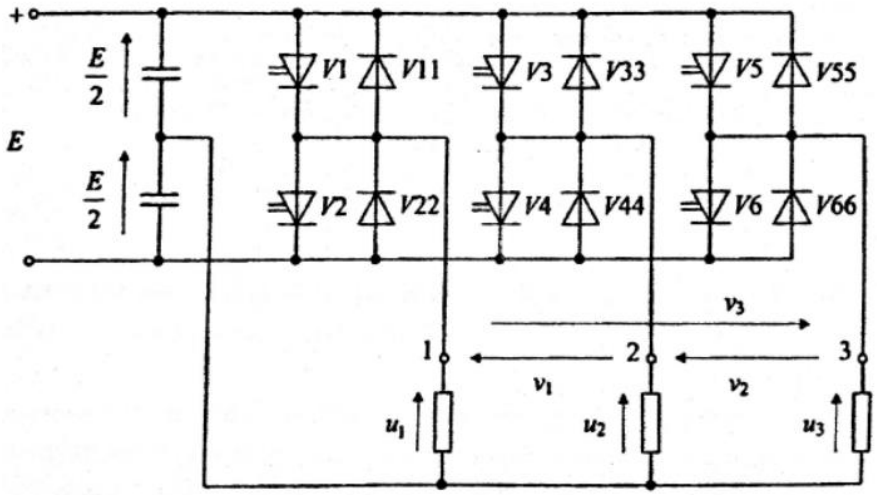
3.2. Trofazni izmjenjivači

Trofazni izmjenjivači su se pojavili za potrebe trofaznih trošila. Najčešće se koriste pri napajanju motornih trošila. Postoji mogućnost korištenja tri jednofazna ispravljača čiji su kutevi izlaznog napona pomaknuti za 120 stupnjeva, no kompleksnost i trošak čine takvo rješenje neoptimalnim. Zbog toga se često koristi topologija 3 polumosna spoja. Trošilo se u takvoj izvedbi može priključiti u trokut, ili u zvjezdište koje se nalazi između dva kondenzatora koji služe kao naponsko djelilo. Na slici 3.5. i 3.6. su prikazane sheme trofaznog polumosnog spoja i karakteristični valni oblici gdje su:

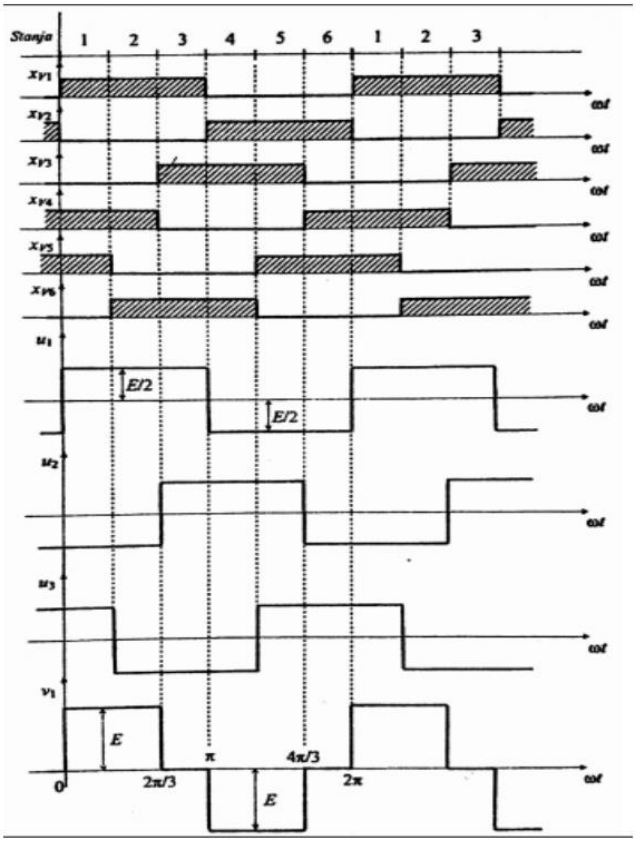
x_{v1} – upravljački impulsi pretvaračkih komponenata

u_n – fazni napon

v_n – međufazni napon



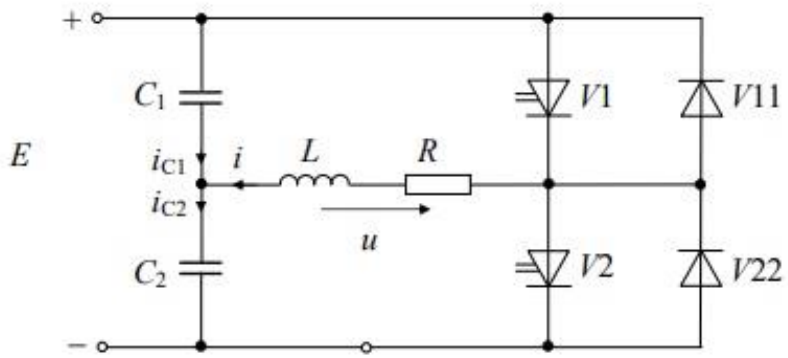
Slika 3.5. Shema polumosnog trofaznog izmjenjivača



Slika 3.6. Karakteristični valni oblici polumosnog trofaznog izmjenjivača

3.3. Polumosni spoj

Zbog upotrebe samo jedne grane trofaznog spoja izmjenjivača, moguće je gledati na njega kao na jednofazni polumosni spoj. Prednost polumosnog spoja nad mosnim spojem je smanjivanje broja komponenti potrebnih za izradu izmjenjivača. Time se dobivaju značajne uštede na cijeni izrade, ali i smanjenje padova napona unutar jednog intervala u analizi rada, zbog čega polumosni spojevi imaju veći stupanj djelovanja.



Slika 3.7. Shema polumosnog spoja

Iz sheme na slici 3.7. uočava se :

1) Struja trošila u bilo kojem trenutku prolazi kroz samo jedan ventil, čime se postižu manji gubici vođenja.

2) Srednja točka se postiže serijskim spojem dvaju kondenzatora.

Struja je u svakom trenutku jednaka.

$$i = i_{C2} - i_{C1} \tag{2-1}$$

Zbog toga što je srednja vrijednost struje kapaciteta jednaka nuli, tako je i srednja vrijednost struje trošila jednaka nuli.

$$I_{C1}(0) = I_{C2}(0) = 0 \tag{2-2}$$

Time se izbjegava opasnost od predmagnetiziranja, kao i eventualno zasićenje transformatora kada se zahtjeva galvanska odvojivost od trošila

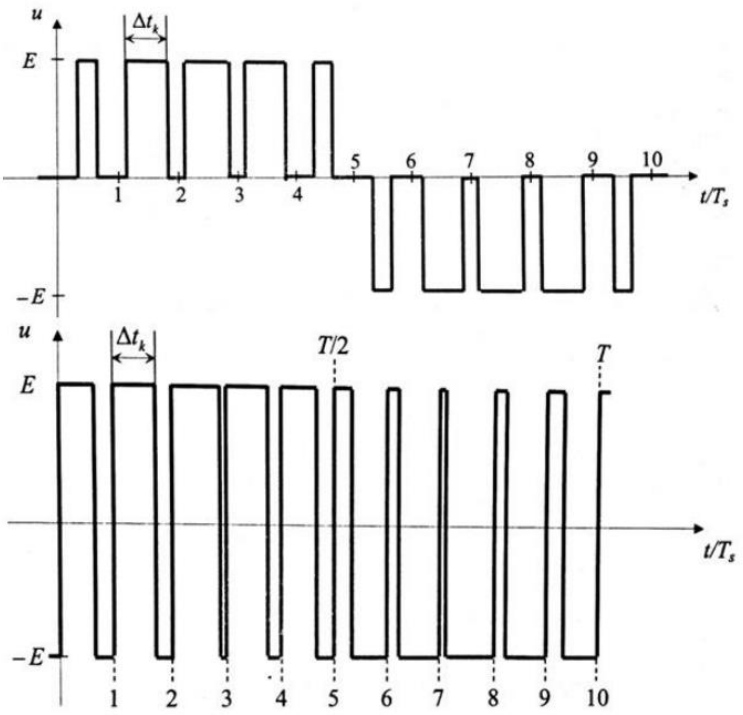
3) Napon trošila je dvorazinski.

Napon trošila uvijek ima jedan od dva iznosa, *+E/2* ili *-E/2*. Zbog nemogućnosti postizanja stanja u kojem napon trošila iznosi 0 V, ovaj sklop je moguće upravljati samo bipolarnom pulsno-širinskom modulacijom [7].

3.4. Pulsno-širinska modulacija

Pulsno-širinska modulacija (PŠM) rješava problem nepovoljne ukupne harmonijske distorzije struje motornih trošila. Umjesto dodavanja filtera u izlazni krug trošila kojim bi se smanjili izlazni harmonici, koristi se PŠM, jer je jednostavnije i jeftinije rješenje [7]. Uz to, može se koristiti i za upravljanje frekvencija rada.

Načelo rada PŠM-a se sastoji od generiranja pravokutnih signala, čije zbrojene površine imaju jednaku vrijednost kao površina sinusnog signala. Drugim riječima, u jednoj periodi moduliranog signala površina volt-sekundi mora biti jednaka kao površina volt-sekundi odgovarajućeg sinusnog signala. Postoje dvije glavne vrste PŠM – unipolarna i bipolarna. Unipolarna ima trorazinski oblik napona izlaza, dok bipolarna ima dvorazinski. Usporedba modulacija se nalazi na slici 3.8.



Slika 3.8. Usporedba PŠM modulacija (unipolarni – gore, bipolarni – dolje)

Sinusna pulsno-širinska modulacija se radi injektiranjem trokutastog signala više (prijenosne) frekvencije u sinusni signal niže (modulacijske) frekvencije. Prijenosna frekvencija je sklopna frekvencija izmjenjivača, dok je modulacijska frekvencija željena frekvencija napona na trošilu.

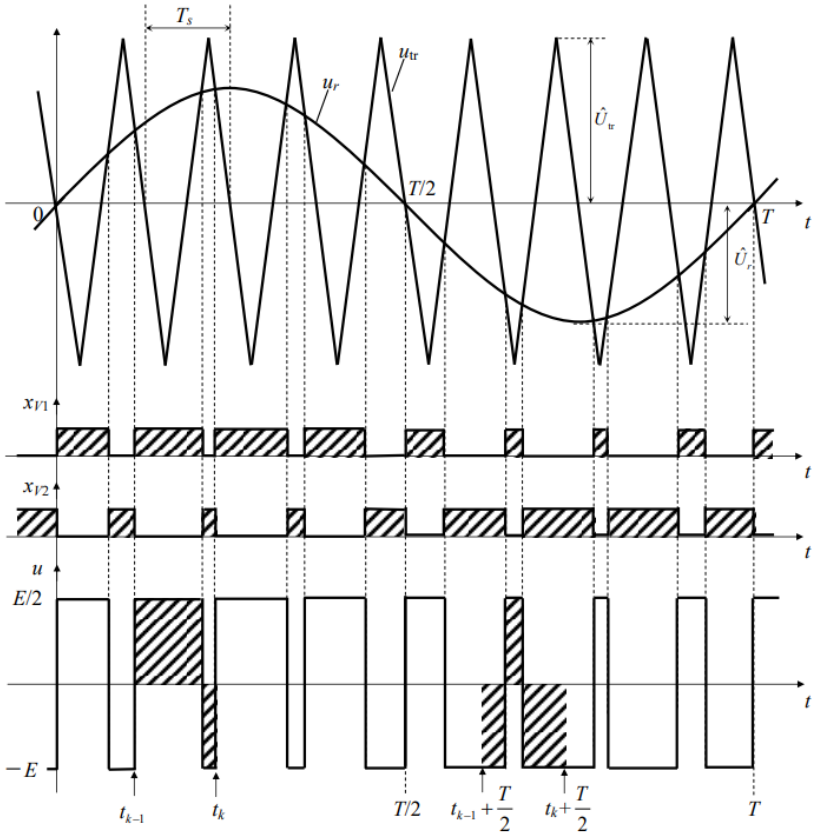
Dvije glavne veličine PŠM se nazivaju modulacijski faktor (m_f) i modulacijski indeks (m_a). Modulacijski faktor označava omjer prijenosne i modulacijske frekvencije, te se računa na sljedeći način.

$$\boldsymbol{m_f = \frac{f_0}{f}} \qquad \text{gdje je : } \boldsymbol{f_0} \text{ – frekvencija prijenosnog signala} \qquad \boldsymbol{f} \text{ - frekvencija sinusnog signala} \qquad (2\text{-}3)$$

Modulacijski indeks (dubina modulacije) je veličina koja se izračunava kao omjer vršne vrijednosti sinusnog i trokutastog napona.

$$\boldsymbol{m_a = \frac{\widehat{U}_r}{\widehat{U}_{tr}}} \qquad \boldsymbol{0 \leq m_a \leq 1} \qquad (2\text{-}4)$$

PŠM se izvodi tako što se uspoređuju prijenosni i referentni sinusni signal. U trenutku kada je sinusni signal većeg iznosa u odnosu na prijenosni, otvara se odgovarajući ventil. U trenutku kada iznos prijenosnog signala bude veći od sinusnog, zatvara se isti ventil. Time se dobivaju pravokutni signali koji su površinom u jednoj periodi odgovaraju sinusnom signalu. Na slici 3.9. je primjer bipolarne modulacije za m_a = 0.75 i m_f = 8.



Slika 3.9. Primjer bipolarne modulacije za m_a = 0.75 i m_f = 8

Osnovni harmonik je harmonik koji se nalazi na modulacijskoj frekvenciji, te je njegov iznos direktno proporcionalan modulacijskom indeksu

$$\boldsymbol{u(1,t) = \hat{U}(1)sin(\omega_1 t) = m_a \frac{E}{2} \sin(\omega_1 t)\,, m_a \leq 1} \tag{2-5}$$

Spektar izlaznog napona se osim osnovnog harmonika sastoji od drugih harmonijskih članova, od kojih se prvi značajni član pojavljuje na sklopnoj frekvenciji, te se ostali značajni članovi niže amplitude pojavljuju na umnošcima sklopne frekvencije. Ova pojava je neovisna o dubini modulacije. Frekvencija harmonijskog člana data je izrazom :

$$\boldsymbol{f_i = (jm \pm k) f_1} \tag{2-6}$$

Gdje je $\boldsymbol{f_i}$ harmonijski član, $\boldsymbol{j}$ višekratnik frekvencijskog indeksa i k bočni pojas tražene frekvencije.

Problem pulsno-širinske modulacije je veliki udjel harmonijskih članova u naponu trošila. Prednost pulsno-širinske modulacije je to što se povećanjem frekvencijskog indeksa ti se isti harmonijski članovi guraju u područje viših frekvencija, gdje zbog prigušnih svojstava induktiviteta oni imaju manji efekt na izlaznu struju. Amplitude harmonika za određeni modulacijski i frekvencijski indeks su prikazani u tablici 3.1.

Tablica 3.1. Udio harmonika ovisno o dubini i faktoru modulacije

Redni broj harmonika	Modulacijski indeks m_a				
	0.2	0.4	0.6	0.8	1.0
1	0.2	0.4	0.6	0.8	1.0
m_f	1.242	1.150	1.006	0.818	0.601
$m_f \pm 2$	0.016	0.061	0.131	0.220	0.318
$m_f \pm 4$					0.018
$2m_f \pm 1$	0.190	0.326	0.370	0.314	0.181
$2m_f \pm 3$		0.024	0.071	0.139	0.212
$2m_f \pm 5$				0.013	0.033
$3m_f$	0.335	0.123	0.083	0.171	0.113
$3m_f \pm 2$	0.044	0.139	0.203	0.176	0.062
$3m_f \pm 4$		0.012	0.047	0.104	0.157
$3m_f \pm 6$				0.016	0.044
$4m_f \pm 1$	0.163	0.157	0.008	0.105	0.068
$4m_f \pm 3$	0.012	0.070	0.132	0.115	0.090
$4m_f \pm 5$			0.034	0.084	0.119
$4m_f \pm 7$				0.017	0.050

Za bipolarnu modulaciju vrijedi sljedeći izlaz za ukupnu harmonijsku distorziju napona:

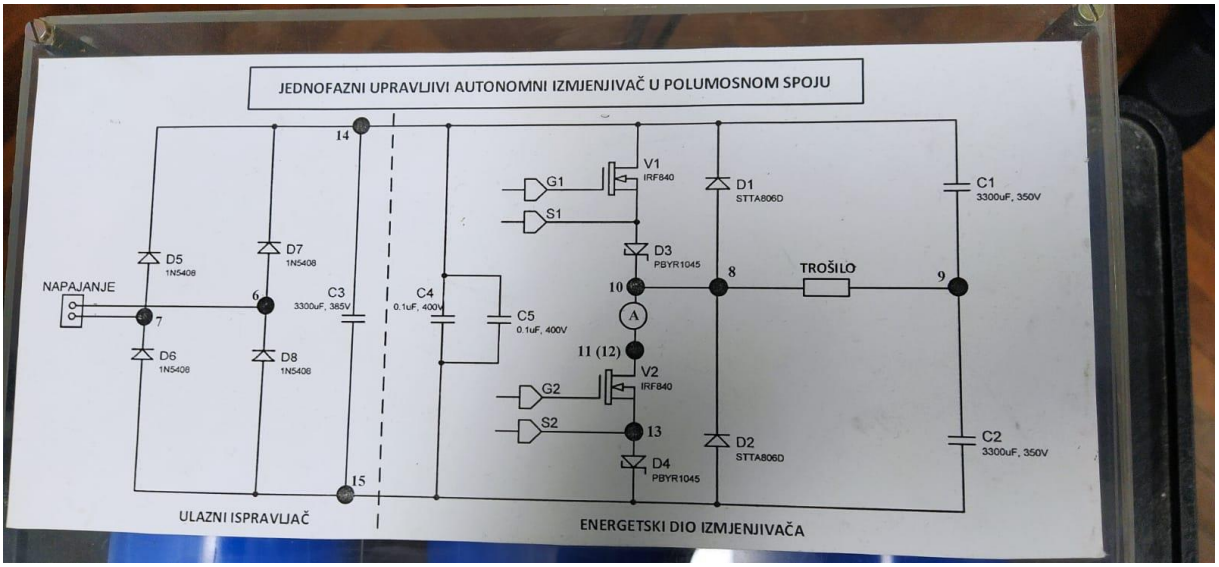
$$\boldsymbol{THDu = \sqrt{\frac{\Sigma_{n>1} U^2(n)}{U^2(1)}} = \sqrt{\frac{U^2}{U^2(1)} - 1} = \sqrt{\frac{2}{m_a^2} - 1}} \qquad (2\text{-}7)$$

Kada je m_a = 1, tada se dobije minimalna distorzija napona, THDU = 1.

4. IMPLEMENTACIJA

Zadatak ovog diplomskog rada je dograditi već postojeću maketu. Na postojećoj maketi postoji integrirani krug koji je izvršavao pulsno-širinsku modulaciju prema datim parametrima s vanjske konzole. Cilj je izbaciti integrirani krug iz makete, te pomoću Arduino mikrokontrolera, odgovarajuće tipkovnice i monitora napraviti samostalnu maketu na kojoj će se moći izvršavati pokusi za upravljanje jedne grane trofaznog izmjenjivača. Problem postojeće makete je korištenje stolnog računala koje koristi zastarjelo sučelje MS-DOS preko paralelne veze. Takvo sučelje više nije standard te stvara probleme pri pokretanju makete [2].

Na slici 4.1 se nalazi shema postojeće makete. Potrebno je pomoću Arduino sklopa napraviti izlaze koji bi na MOSFET-e ventila V1 i V2 davali potrebne upravljačke signale logičke „1“ vrijednosti 5V kako bi se sklapali.



Slika 4.1. Shema postojećeg sklopa

4.1. Logika rješenja

Ideja rješenja je, kao i u stvarnosti, uspoređivati vrijednosti sinusnog signala i trokutastog. To je riješeno zapisivanjem dviju tablica u memoriju Arduino mikrokontrolera. Tablice se sastoje od 1000 vrijednosti, koje rangiraju od 0 do 1000. Korisnik unosi koji modulacijski faktor i modulacijski indeks želi. Nakon toga se vrši usporedba. Svaki element sinusne tablice se uspoređuje sa m_f-tom vrijednošću tablice trokuta. Time se efektivno dobiva trokutasti signal ponovljen m_f puta. Ukoliko je vrijednost iz sinusne tablice veća, u polje

rezultata se upisuje vrijednost 1, a u suprotnom slučaju upisuje se 0. Polje se tada u beskonačnoj petlji prosljeđuje na dva izlazna pina koji predstavljaju upravljačke signale za MOSFET-e na maketi.

4.2. Komponente

Za rješenje upravljačkog dijela makete korištene su sljedeće komponente:

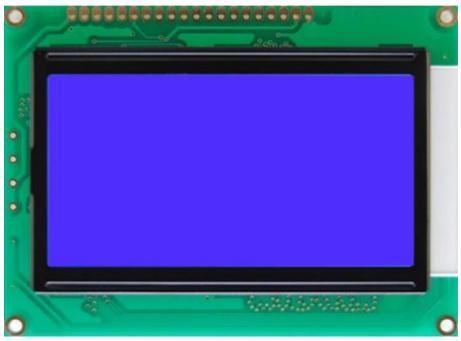
Tablica 4.1. Korištene komponente

Komponenta	Broj
Arduino Mega 2560 Rev3	1
LCD ST7920 ekran	1
Matrix keypad 4x4	1
Konektori	19

Arduino Mega 2560 je mikrokontroler zasnovan na ATmega2560 procesoru. Sastoji se od 54 digitalna pina, 16 analognih pinova, 4 UART porta. Na njemu se nalazi 16MHz procesor, 8KB radne memorije, 256KB flash memorije, port za serijsku USB konekciju, priključak za napajanje, ICSP header i gumb za reset. U razvoju programskog rješenja su korišteni druge Arduino razvojne ploče, kao što su Arduino Uno i Arduino Leonardo, no potreba za više od 2KB radne memorije je rezultirala odabiranjem Arduino Mega 2560 kao mikrokontrolera za rješenje upravljanja MOSFET-ima. Arduino Mega 2560 se nalazi na slici 4.2.

Slika 4.2. Arduino Mega 2560

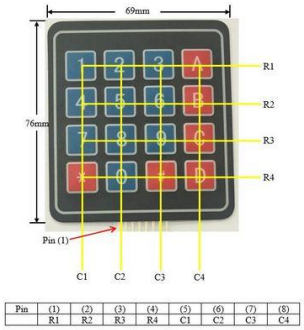
ST7920 je ekran rezolucije 128x64 piksela. Na njemu se nalazi 20 pinova, no u ovom radu je iskorišteno samo 8 (GND, VCC, RS, R/W, E, PSB, RST, BLA i BLK). Veličina komponente je 93x70x12.5 mm, dok je sam ekran veličine 71.8x40mm. ST7920 je prikazan na slici 4.3.



Slika 4.3. ST7920

Za vanjsko upravljanje program koristi se tipkovnica Keypad 4x4 matrix. Tipkovnica ima 8 izlaza koji predstavljaju sve moguće kombinacije tipki (4 reda i 4 stupca). Na slikama 4.4. i 4.5. su prikazana tipkovnica i shema rada tipkovnice.

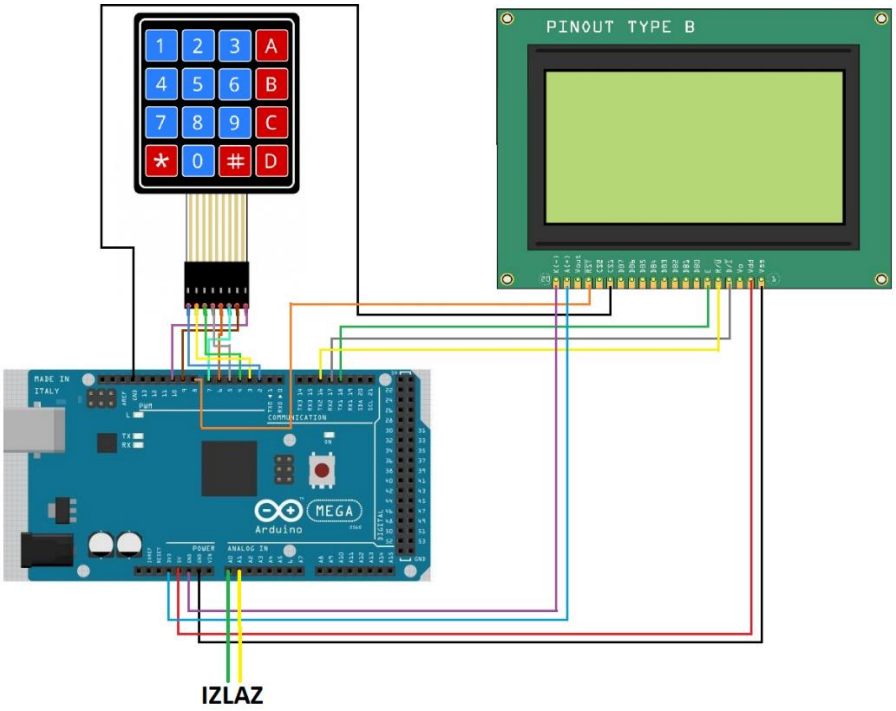
Slika 4.4. Tipkovnica Keypad 4x4



Slika 4.5. Shema rada tipkovnice

4.3. Shema sklopovlja

Na slici 4.6. se nalazi shema spajanja datih komponenata s Arduino pločom, dok se na slici 4.7. nalazi slika izrađene makete.



Slika 4.6. Shema spajanja

Slika 4.7. Slika makete

Zbog jednostavnosti spajanja, u tablici 4.2. se nalaze izlazi svih komponenti koji su spojeni s Arduino pločom.

Tablica 4.2. Arduino spajanje

Komponente		Arduino pinovi
Tipkovnica	1	D10
	2	D9
	3	D7
	4	D6
	5	D5
	6	D4
	7	D3
	8	D2
Ekran	GND	GND
	VCC	5V
	RS	D17
	R/W	D16
	E	D18
	GND	GND
	RST	D8
	BLA	3.3V
	BLK	GND
Izlaz	Izlaz 1	A0
	Izlaz 2	A1

4.4. Objašnjenje koda

Program započinje naredbama potrebnim za inicijalizaciju potrebne periferije. Potrebno je na početku dodati potrebne biblioteke za rad ekrana i tipkovnice, kao što su U8g2lib.h i Keypad.h. Za ekran se pronalazi odgovarajuća naredba za inicijalizaciju. Za ovaj diplomski rad je korišten ekran tipa ST7920 sa 128x64 rezolucijom, što znači da je odabrana naredba U8G2_ST7920_128x64_f_SW_SPI. Argumenti ove funkcije daju broj pinova na koji se priključuju RS, R/W i E pinovi ekrana. Tipkovnica se inicijalizira naredbom Keypad. Potrebno

je prvo deklarirati varijable koji će biti argumenti funkcije. Postavlja se broj redova (ROWS) i stupaca (COLS) tipkovnice koji se nalaze na tipkovnici. Potom je potrebno deklarirati matricu keys koja predstavlja tipke na tipkovnici, te na posljetku definirati pinove na koje će se tipkovnica priključiti na Arduino mikrokontroler, varijablama rowPins i colPins. Pin broj 8 je preskočen iz razloga što je potreban za rad ekrana.

Unutar setup funkcije se aktiviraju i inicijaliziraju funkcije potrebne za rad. Na početku se aktivira serial port Arduino mikrokontrolera, potom započinje rad ekrana. Pinovi A0 i A1 koji će služiti kao izlaz programa se postavljaju u mod OUTPUT. Registar koji obuhvaća pinove A0-A7 se naziva PORTF, te postavljanjem njega na 0 automatski se oba pina postavljaju na 0.

Funkcija unosParametara vrši funkciju unosa parametara od strane korisnika, te zapisivanja istih u memoriju programa kako bi se s njima izvršio rad samog programa. Funkcija započinje ispisom poruke dobrodošlice korisniku. Nakon toga, na ekranu se ispisuje tekst koji obavještava korisnika o mogućim odabirima amplitudnog indeksa m_a, te zahtjeva od njega odabir jednog od ponuđenih indeksa putem tipkovnice. Nakon toga, ovisno o odabiru, program upisuje zadanu vrijednost u varijablu m_a, te ispisuje je na serial monitoru. Po upisu zadanog amplitudnog indeksa, ponavlja se proces, no ovaj put za upis frekvencijskog indeksa m_f. Na ekranu se ispisuje tekst koji pokazuje moguće odabire indeksa, te traži od korisnika odabir pomoću tipkovnice. Nakon korisnikovog odabira, na ekran se ispisuje obavijest da program radi, te su ponuđene opcije za pauzu pritiskom gumba A na tipkovnici, ili ponovno pokretanje programa pritiskom gumba B.

Kako je prethodno navedeno, program izvršava usporedbu dvije tablice u kojima su zapisane vrijednosti koje predstavljaju amplitude jedne periode sinusnog i trokutastog valnog oblika podijeljenjih u 1000 jednakih dijelova. To izvršava funkcija izracunVrijednosti. Predaju joj se zadani parametri m_a i m_f. Unutar funkcije postoji dvije FOR petlje, jedna koja se izvršava m_f puta, a druga koja se izvršava 1000 puta. Druga petlja se aktivira samo na višekratnike broja m_f. Kada se pojavi višekratnik broja m_f unutar petlje, u varijablu *trokut* se sprema vrijednost koja se u tablici TRI_TABLE nalazi na m_f -tom mjestu, dok se u varijablu *sinus* sprema vrijednost u tablici SIN_TABLE koja se nalazi na *br*-tom mjestu. Varijabla *br* služi za zapis koliko se puta usporedba izvršila, te se koristi kako bi se odabrala točna vrijednost *sinusa*. Tada se vrijednost *sinusa* množi s faktorom m_a, i sprema novu varijablu *masinus*. Tada se vrši

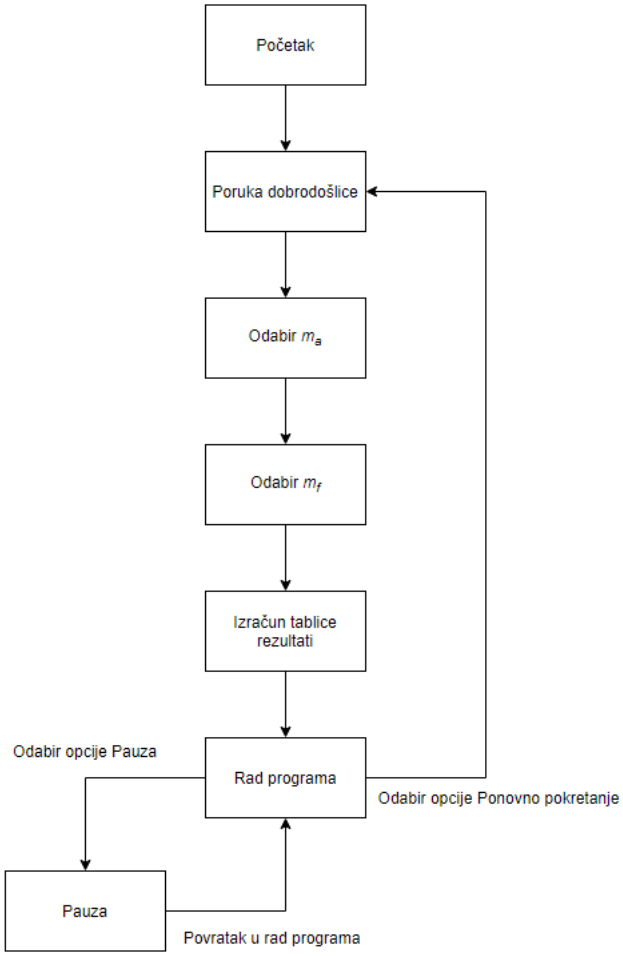
usporedba vrijednosti *masinus* i *trokut*, te ako je *masinus* veća ili jednaka, u polje *rezultat[br]* se upisuje 1, a u suprotnom 0. Na primjer, odabere se frekvencijski indeks m_f = 24 i amplitudni indeks m_a = 1. Početkom prolaska kroz drugu FOR petlju, brojač petlje k = 0. Broj 0 je cjelobrojno djeljiv s 24, te se IF petlja izvršava. Varijabla *masinus* je jednaka kao varijabla *sinus* u ovom slučaju jer se dobije množenjem varijable *sinus* s 1. Uspoređuju se nulte vrijednosti SIN_TABLE I TRI_TABLE. Pogledom u tablicu obje su vrijednosti 500, te se u polje rezultati upisuje 1. Brojač *br* se uveća za 1, te sad ima vrijednost 1. Idući put kad se zadovolji IF uvjet je kad je brojač petlje k = 24. Tada se uzima prva vrijednost iz tablice SIN_TABLE, što je 503, no uzima se 24. vrijednost iz tablice TRI_TABLE, koja iznosi 546. Usporedbom se zaključuje da je vrijednost iz TRI_TABLE veća, te se u tablicu rezultati upisuje 0. Brojač *br* se uvećava za 1, te sada iznosi 2. Proces se ponavlja. Time se efektivno kroz tablicu TRI_TABLE prođe 24 puta brže, te se zbog toga prolaženje kroz nju ponavlja 24 puta, i time se dobije 24 periode trokutastog signala unutar jedne periode sinusnog signala. Na kraju postoji uvjet redundancije, ako brojač *br* prođe veću vrijednost od 999, da se izađe iz petlje. Time se uštedilo na vremenu kako se ne bi moralo provjeravati daju li svaka kombinacija parametara m_a i m_f točno 1000 rezultata.

Funkcija pauza služi za privremeno zaustavljanje programa. Pri pritisku tipke A na tipkovnici dok se program izvršava, izlazi se iz beskonačne petlje LOOP, te se ulazi u funkciju pauza. Unutar funkcije se konstantno provjerava je li pritisnuta bilo koja tipka na tipkovnici. Ukoliko je pritisnuta tipka A, buffer *gkey* se očisti, ponovno se ispiše tekst izvršavanja programa na ekranu, te se vrati nazad u beskonačnu petlju LOOP gdje se nastavi izvršavanje programa.

Unutar beskonačne petlje LOOP vrši se prosljeđivanje tablice rezultata na izlazne pinove. To se obavlja FOR petljom koja se izvršava 1000 puta. Prolaskom kroz petlju, provjerava se stanje polja rezultati[]. Ukoliko je rezultat[i] = 1, registar PORTF se postavlja u 0b00000001, što na izlazu daje vrijednost 1 na pinu A0, a 0 na pinu A1. U suprotnom, ako je rezultat[i] = 0, registar PORTF se postavlja u 0b00000010, što na izlazu daje vrijednost 0 na pinu A0, a 1 na pinu A1. Time je postignuta trenutna zamjena vrijednosti na pinovima izlaza, što ne krši mrtva vremena IGBT komponenti, te automatski vrši vremensku sinkronizaciju valnih oblika na izlazu A0 i A1. Ukoliko bi se koristile dvije naredbe za prosljeđivanje rezultata na izlazne pinove, stvarao bi se problem kod mrtvih vremena IGBT-ova i valni oblici ne bi bili sinkronizirani. Uz prosljeđivanje rezultata na izlazne pinove, provjerava se i postoji li unos s tipkovnice. Ukoliko postoji, te ako je pritisnuta tipka A, ulazi se u funkciju pauza koja

privremeno zaustavlja rad programa. Ukoliko je pritisnuta tipka B, ponovno se ulazi u funkcije unosParametara i izracunVrijednosti, koje za rezultat imaju drukčiji ispis na izlazu, u ovisnosti od odabranih parametara m_a i m_f.

Na slici 4.8. je prikazan blok dijagram rada programa.



Slika 4.8. Blok dijagram rada programa

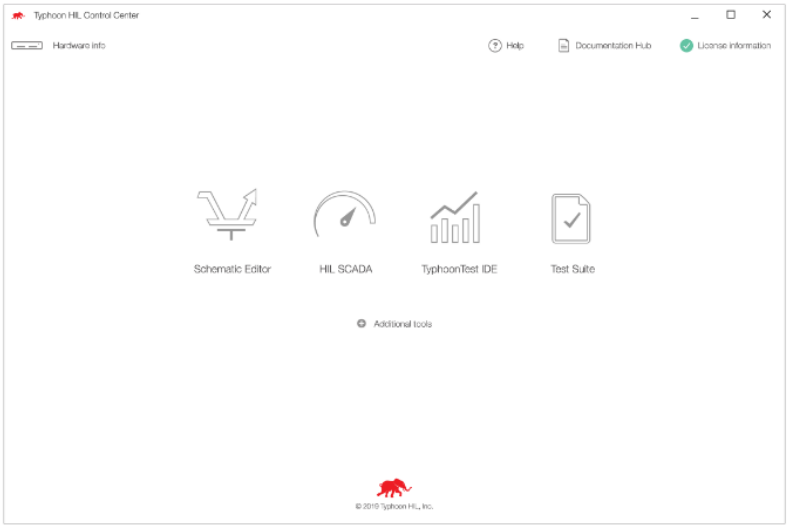
4.5. Typhoon HIL

HIL (Hardware-in-loop) je hardversko sučelje koje se koristi za testiranje ugradbenih računalnih sustava unutar računalne situacije. Može se opisati kao okruženje za testiranje programske podrške koja se ugrađuje u sofisticirane sustave kao što su elektronički energetski pretvarači, kao i mikro mreže. Sustav emulira stvarno vanjsko sklopovlje koje se koristi, te prima ulazne i izlazne signale koje daje programska podrška, te ih obrađuje na jednak način kao što bi se obrađivali na stvarnom vanjskom sklopovlju. Typhoon HIL paket se sastoji od sklopovlja (Typhoon HIL 402) i programske podrške (Typhoon HIL Control Center).

HIL 402 je uređaj koji predstavlja fizičko sklopovlje Typhoon HIL paketa. Sastoji se od četverojezgrenog procesora, po 16 analognih ulaza i izlaza, te po 32 digitalna ulaza i izlaza. Naponski rang iznosi ± 10 V. Samo sklopovlje se sastoji od dva dijela, modula za procesiranje i kartice s ulazima i izlazima (DSP sučelja). Na slici 4.9. je prikazan HIL 402 [8].

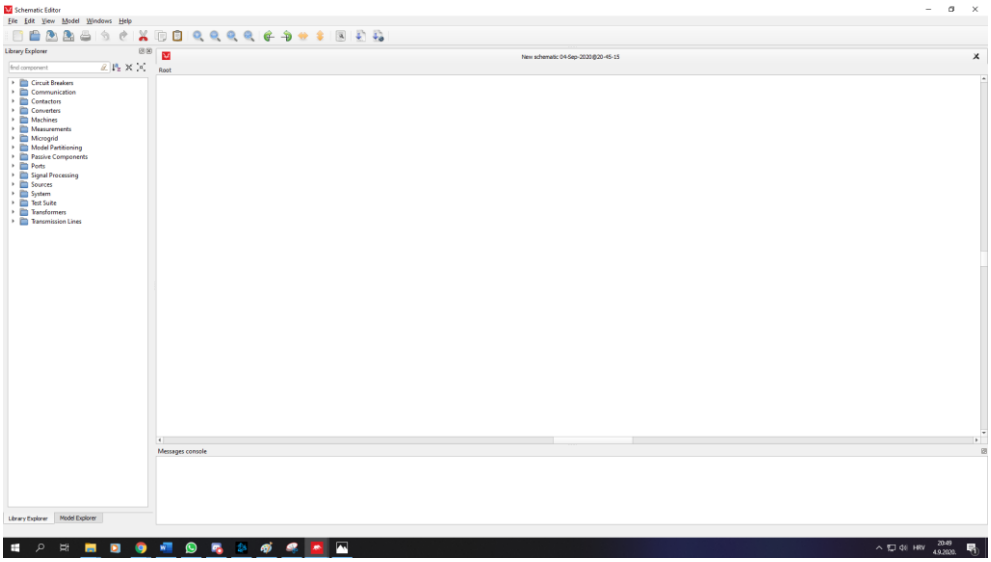
Slika 4.9. HIL 402

Typhoon HIL Control Center je programski dio Typhoon HIL paketa. Unutar navedenog programa su izvršeni softverski dijelovi simulacije, kao što su crtanje sheme i njeno testiranje. Sastoji se od Schematic Editor-a, HIL SCADA-e, Typhoontest IDE i Test Suite-a. Za potrebe diplomskog rada korišteni su Schematic Editor i HIL SCADA. Na slici 4.10. je prikazano ulazno sučelje Typhoon HIL Control Center-a.



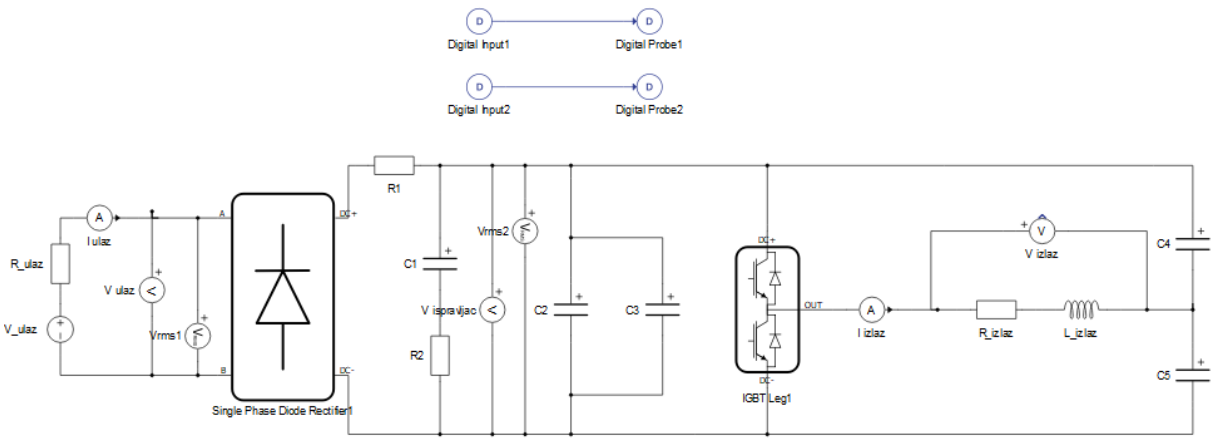
Slika 4.10. Ulazno sučelje Typhoon HIL Control Center-a

Schematic Editor je dio razvojnog okruženja Typhoon HIL-a zadužen za stvaranje shema nad kojim se u daljnjim koracima vrši simulacija. S lijeve strane prozora se nalazi Library Explorer, gdje se mogu naći sve postojeće komponente unutar Typhoon HIL-a. Programski paket također podržava dodavanje novih komponenti putem C/Python skripti. Na slici 4.11. je prikazano sučelje Schematic Editor-a.



Slika 4.11. Schematic editor

Unutar Schematic Editor-a je napravljena shema po uzoru na već postojeću maketu, prikazanu na slici 4.1. Zbog prirode rada programa, neke komponente su spojene u već gotov sklop, kao što su npr. diodni ispravljač. Na slici 4.12. je prikazana shema, te u tablici 4.2. parametri komponenata.

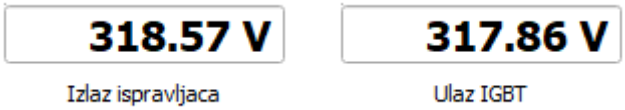


Slika 4.12. Shema unutar Typhon HIL-a

Tablica 4.2. Parametri sheme

Komponenta sheme	Vrijednost
V_ulaz	230 [V], 50 [Hz] sinus
R_ulaz	0.1 [Ω]
R1	10 [Ω]
R2	0.01 [Ω]
R_izlaz	800 [Ω]
C1	3300 [μF]
C2	0.1 [μF]
C3	0.1 [μF]
C4	3300 [μF]
C5	3300 [μF]
L_izlaz	330 [mH]

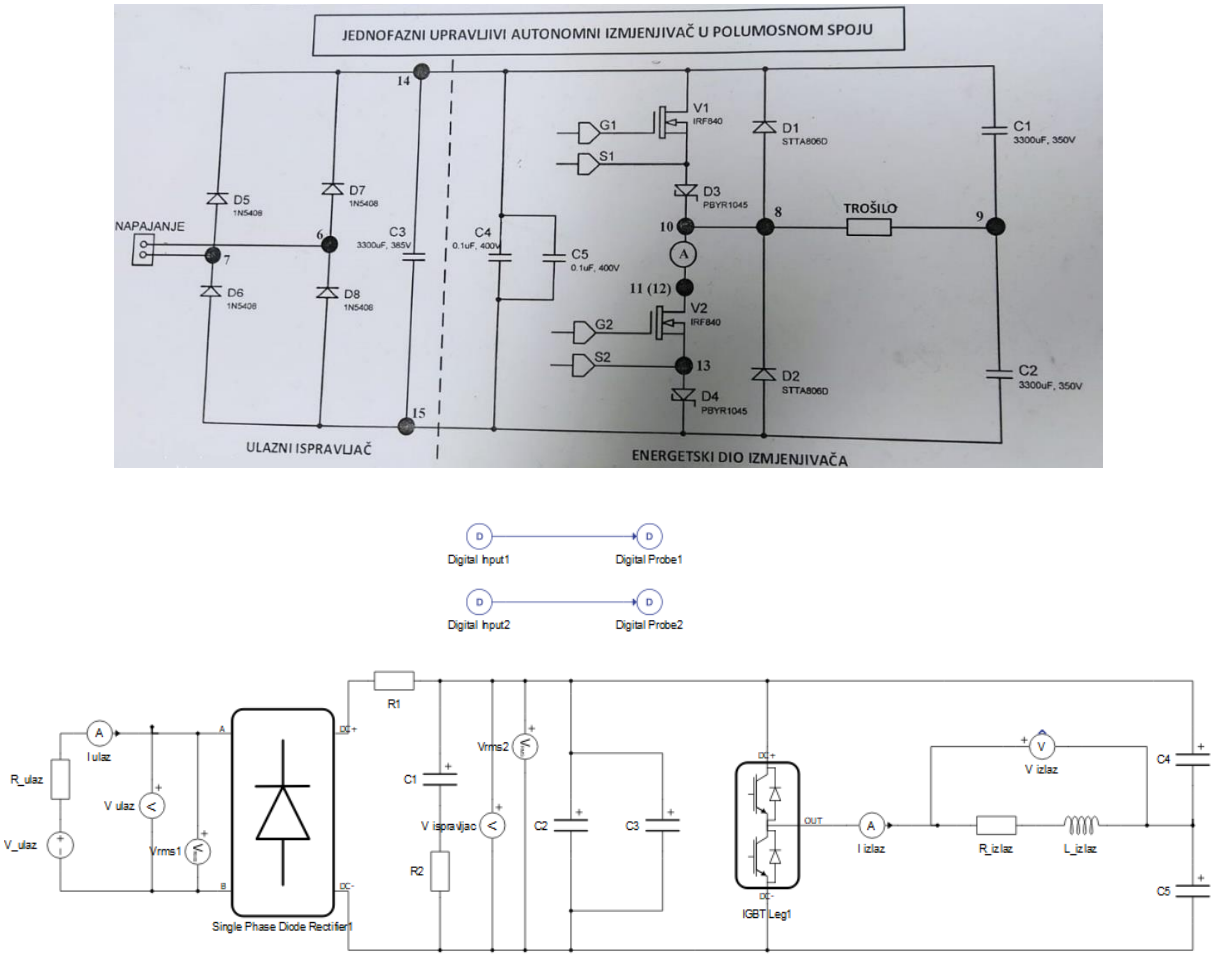
Na ulaz ispravljača doeveden je sinusni napon 230V, 50 Hz koji predstavlja izvor naponske mreže. R_ulaz ograničava struju na ulazu. Ispravljač na izlazu daje efektivnu vrijednost napona od 318.5 V. R1 otpornik predstavlja ograničenje struje na izlazu iz ispravljača. Na IGBT sklop dolazi ispravljeni napon 317.83 V. Naponi su prikazani na slici 4.13.



Slika 4.13. Naponi ispravljača i ulaza grane izmjenjivača

Na upravljačka vrata IGBT dolaze upravljački signali s Arduino ploče. Ukoliko u jednom trenutku tablica rezultata iznosi 1, to znači da se jedinica šalje na upravljačka vrata prvog IGBT-a, a na vrata drugog IGBT-a daje komplementiranu vrijednost, odnosno 0. To znači da se prvi IGBT otvara, te kroz njega protječe struja, dajući na trošilu pozitivni napon u iznosu *+E/2*. Kada tablica rezultata iznosi 0, na upravljačka vrata prvog IGBT-a dolazi 0, a na vrata drugog IGBT-a 1. Zbog toga struja protječe kroz drugi IGBT, dajući na izlaz negativni napon, iznosa *-E/2*. Izlazno trošilo je induktivnog karaktera. Kondenzatori C4 i C5 tvore središnju točku

izmjenjivača, osiguravajući nulište za spajanje trošila. Na slici 4.14. se nalazi usporedba sheme s makete i sheme u Typhoon HIL CC-u.



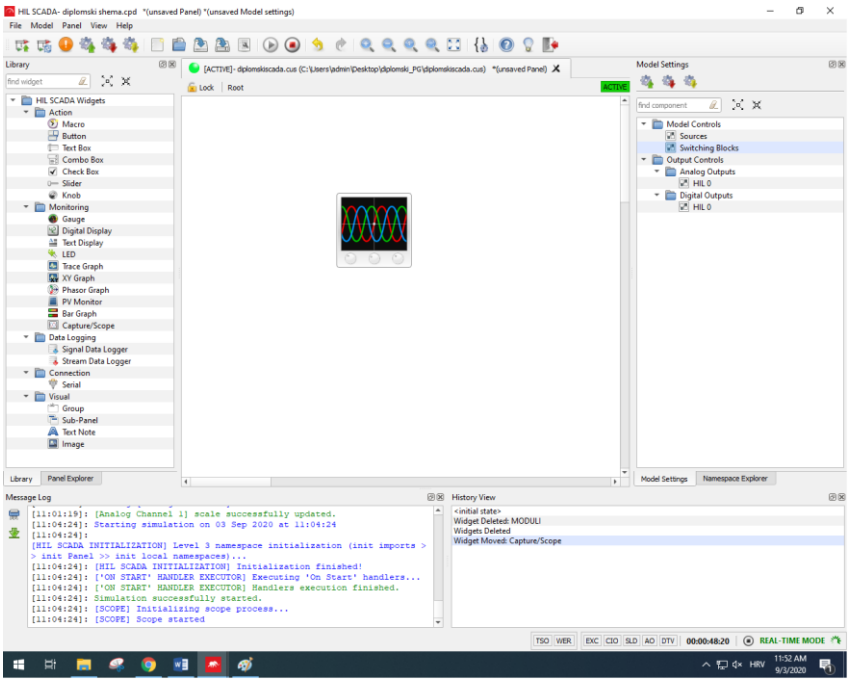
Slika 4.14. Usporedba shema s makete i sheme u Typhoon HIL CC

Nakon uspješnog kompajliranja sheme, otvara se drugi dio Typhoon HIL CC, HIL SCADA. HIL SCADA je dio programa koji služi za praćenje stanja sheme, mijenjanje parametara emulacije i očitavanje ključnih dijelova sheme. Unutar prozora s lijeve strane se nalazi Library meni, iz kojeg se na ploču SCADA-e dodaju ponuđeni alati. Oni su podijeljeni u pet grupa :

- Action (akcijski dio) – sadrži različite upravljačke gumbe, kojima je moguće dodijeliti određenu funkcionalnost unutar SCADA-e
- Monitoring (nadzorni dio) – sadrži različite ekrane, crtače grafova, LED i Text prikaze, kojima je moguće pratiti određeno stanje unutar SCADA-e

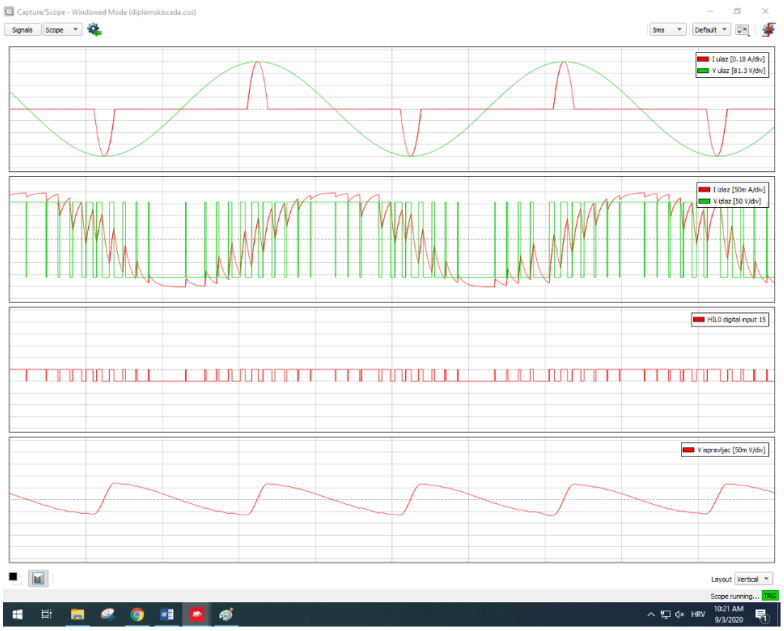
- Data logging (spremanje podataka) – sadrži dva modula kojim se može određena veličina unutar SCADA-e u određenom periodu zabilježavati u .xslx ili .csv formatu
- Connection (povezivanje) – služi za dovod serijskog priključka unutar SCADA-e
- Visual (vizualni dio) – sadrži različite module za crtanje, pisanje teksta i dodavanje slika unutar SCADA-e

S desne strane nalaze se postavke modela, u kojim se mogu mijenjati promjenjivi dijelovi sheme, kao što su iznosi naponskih izvora i sklopke. Također, unutar postavki modela se mogu odrediti i izlazi modela. Na slici 4.15. je prikazano sučelje HIL SCADA-e.



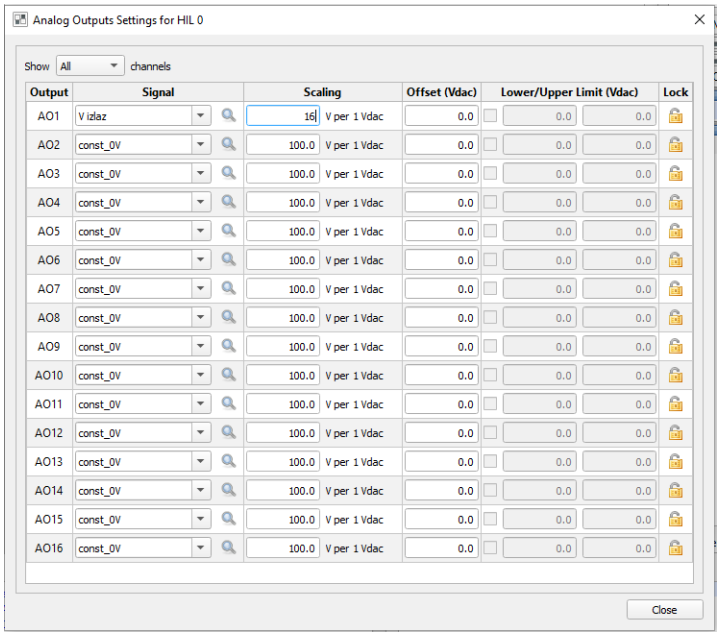
Slika 4.15. HIL SCADA

Za potrebe diplomskog rada, korišten je samo modul Digital Display. Unutar njega su postavljeni četiri različita Scope-a (pregleda). Na svaki pregled dovodi se različit signal, ili signali koji se žele promatrati. U ovom slučaju, na prvom pregledu se nalaze valni oblici ulaznog napona i struje, na drugom valni oblici izlaznog napona i struje, na trećem upravljački signali vrata IGBT-a i na četvrtom promjene napona ispravljača. Za pokretanje rada SCADA-e, potrebno je u vrhu programa pritisnuti gumb Start Simulation. Na slici 4.16. se nalazi primjer izgleda Digital Display-a.



Slika 4.16. Primjer Digital Display-a

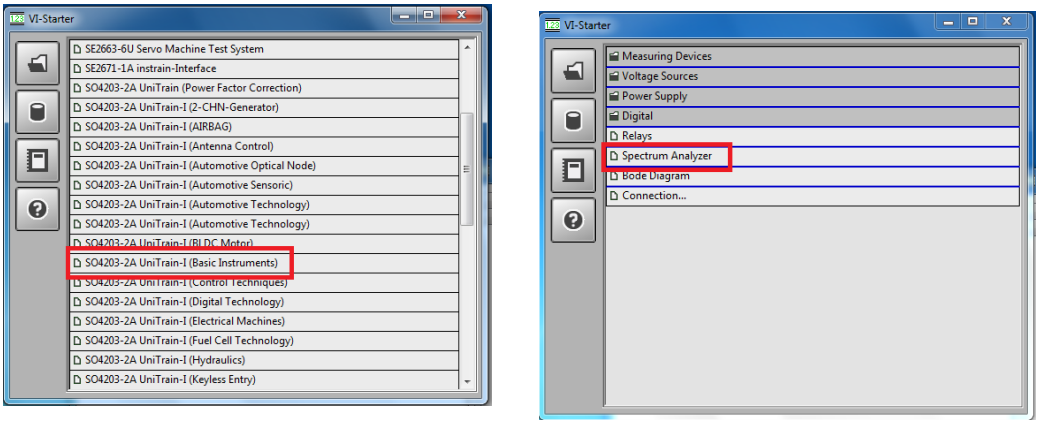
Zbog nemogućnosti izvršavanja harmonijske analize unutar Typhoon HIL paketa, potrebno je izlazni napon iz sheme postaviti na neki od izlaznih pinova HIL402 sklopovlja. Postavljen je na izlazni pin A0, s omjerom napona 16:1. Odabran je taj omjer jer se amplituda naponskog signala kreće oko 160 V, što na ulazu harmonijskog analizatora daje napon od 10 V, čineći ga jednostavnim za pratiti na grafovima harmonijske analize. Na slici 4.17 je prikazano podešenje izlaznih parametara za slanje naponskog signala na harmonijsku analizu.



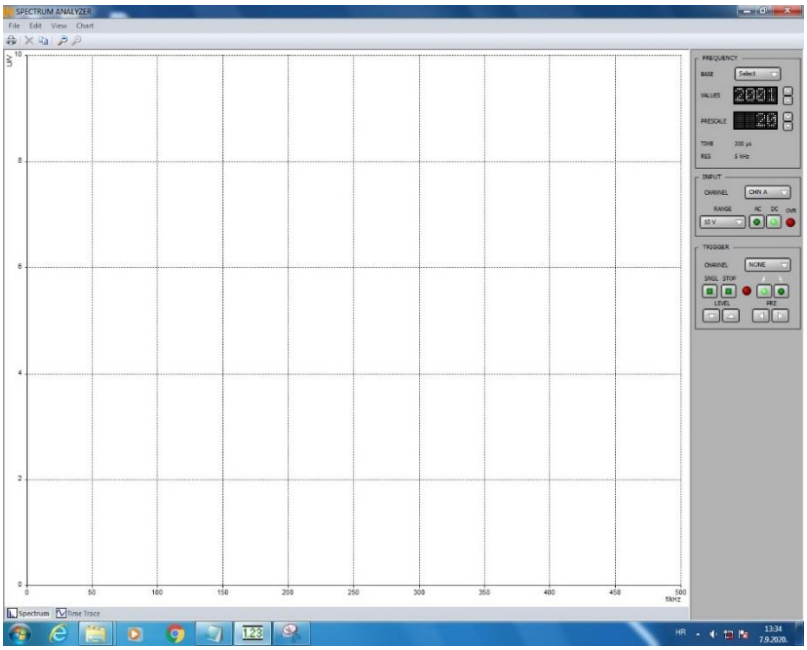
Slika 4.17. Analog output SCADA-e

4.6. UniTrain i harmonijska analiza

Uređaj koji je korišten za harmonijsku analizu se zove UniTrain S04203-2A. Izlaz iz HIL402 je priključen na ulaz UniTrain-a, te je sam uređaj priključen USB sučeljem na računalo. Pokretanjem Lucas Nulle Starter VI programskog sučelja, otvara se meni koji se sastoji od mnogih ponuđenih instrumenata. Za potrebe harmonijske analize, odabire se Basic Instruments opcija. Nakon toga, potrebno je pritisnuti na Spectrum Analyzer opciju, čime se otvara prozor za harmonijsku analizu. Pokretanjem rada Arduina, pojavljuju se vrijednosti harmonika izlaznog napona na pojedinim frekvencijama, čime je odrađena harmonijska analiza. Na slikama 4.18. i 4.19. se nalaze sučelje Starter VI programa.



Slika 4.18. Odabir harmonijskog analizatora u sučelju Starter VI programa



Slika 4.19. Sučelje harmonijskog analizatora

5. TESTIRANJE

5.1. Pokretanje programa

Pokretanje programa započinje dovođenjem napajanja na serijsku USB konekciju na Arduino mikrokontroleru. Pri dobivanju napajanja, na ekranu se pojavljuje poruka dobrodošlice, prikazana na slici 5.1.

Slika 5.1. Poruka dobrodošlice

Nakon što je prošlo 5 sekundi od korisnika se traži unos parametra amplitudnog indeksa modulacije. Zbog načina izvedbe programskog koda, na tipkovnici nije moguće unijeti bilo koji željeni indeks, već je moguće izabrati jednu od 5 predefiniranih opcija koje iznose : 0.2, 0.4, 0.6, 0.8 i 1.0. Korisnik pritiskom na gumbe od 1 do 5 bira željeni iznos amplitudne modulacije. Prikaz odabira amplitudnog indeksa prikazan je na slici 5.2.

Slika 5.2. Odabir amplitudnog indeksa m_a

Poslije odabiranja amplitudnog indeksa, od korisnika se traži odabir frekvencijskog indeksa modulacije. Slično prethodnom odabiru, na ekran se ispisuju moguće vrijednosti m_f, te

korisnik upotrebom tipkovnice odabire željeni m_f. Dostupne vrijednosti m_f su : 12, 24, 48, 96. Prikaz odabira frekvencijskog indeksa prikazan je na slici 5.3.

Slika 5.3. Odabir frekvencijskog indeksa m_f

Kada korisnik unese željeni frekvencijski indeks, program izračuna potrebnu tablicu rezultata, te je pođe prosljeđivati na izlazu. U međuvremenu, na ekran se ispiše poruka da program radi, te ukoliko korisnik želi pauzirati program, mora pritisnuti 'A' na tipkovnici ili ako želi ponovno pokrenuti program, treba pritisnuti 'B'. Na slici 5.4. je prikazan ekran s tekstom.

Slika 5.4. Tekst dok se program izvršava.

U modu rada „Pauza“, program se privremeno zaustavlja, te je za nastavak rada s jednakim parametrima dovoljno ponovno pritisnuti tipku 'A'. Ispis na ekranu je prikazan na slici 5.5.

Slika 5.5. Pauza

Ukoliko korisnik pritisne tipku 'B' tijekom rada programa, program počinje iznova s radom, te se na ekranu ispisuje tekst dobrodošlice, prikazan na slici 5.1.

5.2. Harmonijska analiza izlaznog napona

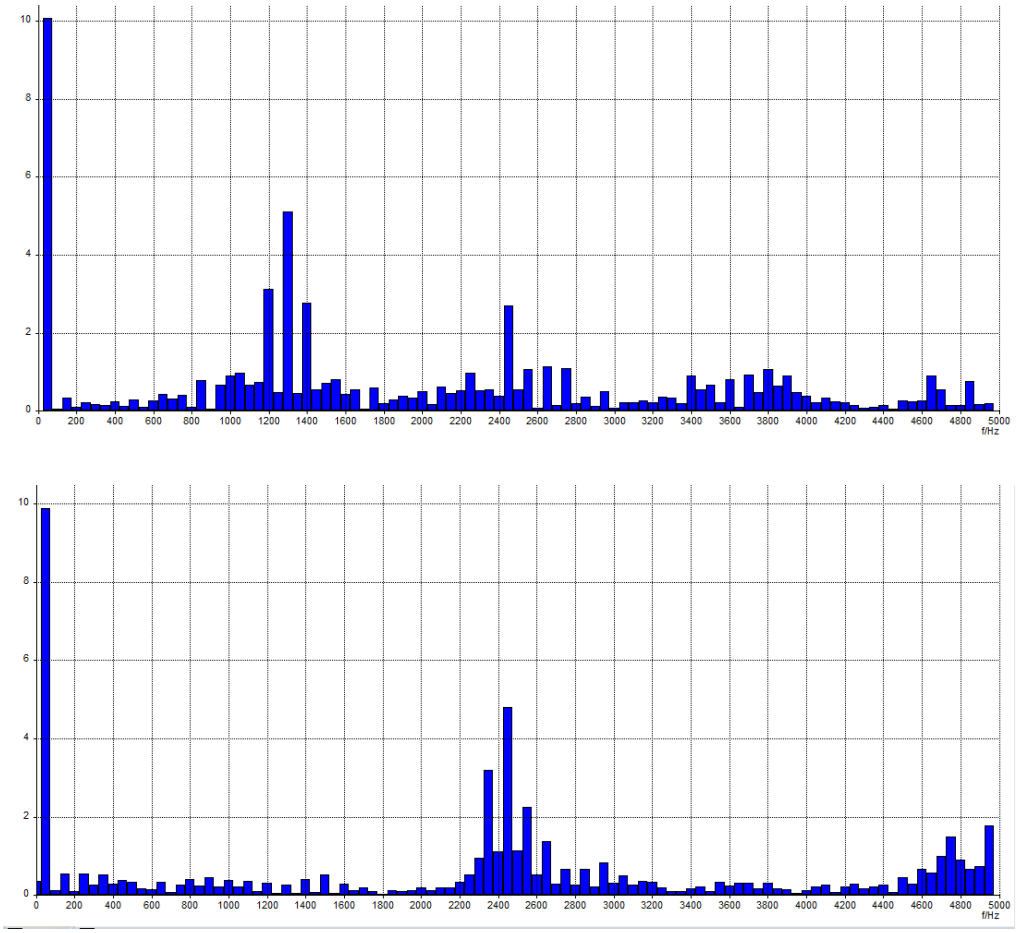
Cilj je potvrditi dvije pretpostavke :

a) Povećanjem m_f se značajni harmonici izlaznog napona guraju u više frekvencijsko područje

b) Harmonici izlaznog napona zadovoljavaju parametre iz tablice 3.1.

a) Povećanjem m_f se značajni harmonici izlaznog napona guraju u više frekvencijsko područje

Na slici 5.6. prikazani su dva stupčasta prikaza harmonika. Na gornjem je odabrana vrijednost $m_f = 24$, dok je na donjem $m_f = 48$. U oba slučaja amplitudni modulacijski indeks je iznosio $m_a = 1$.

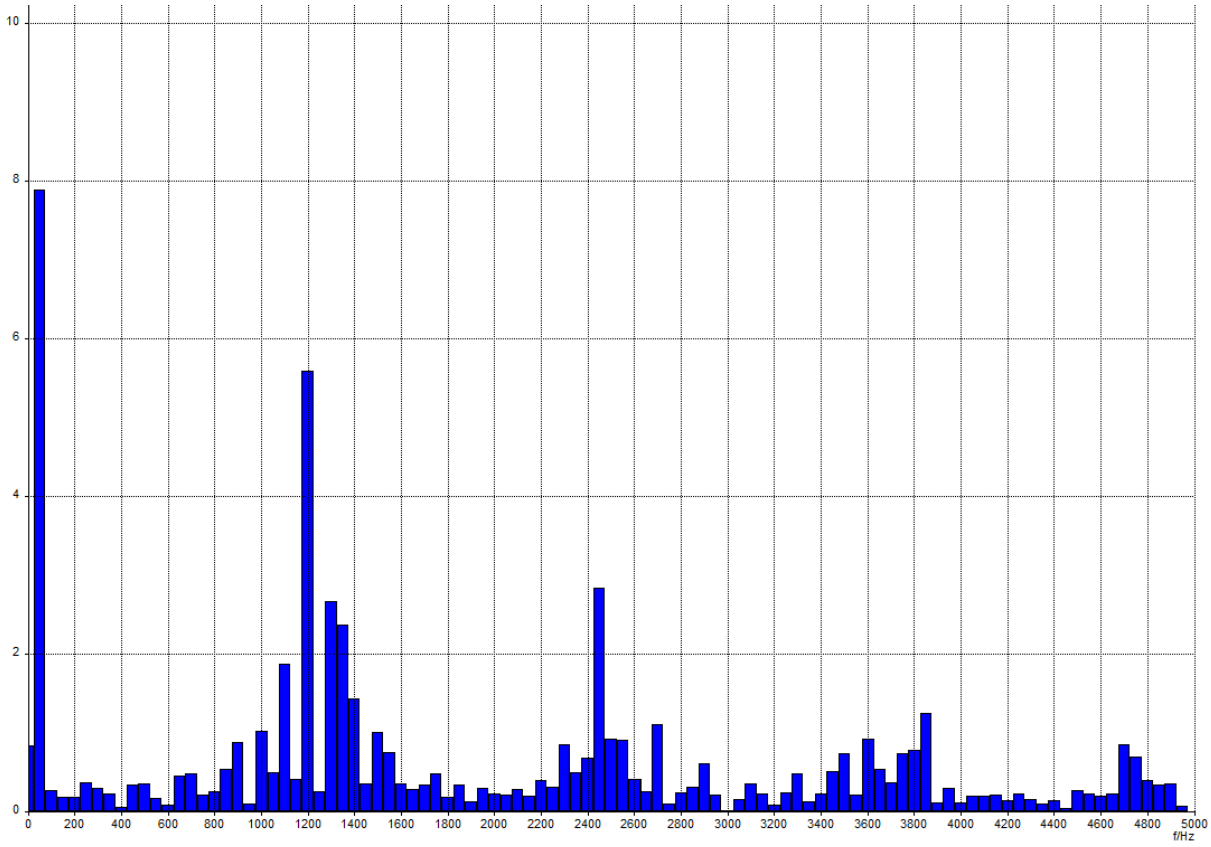


Slika 5.6. Usporedba dvaju harmonijskih prikaza (gore m_f = 24, dolje m_f = 48)

Iz slike 5.6. se može zaključiti kako je osnovni harmonik napona 10 V na zadanoj frekvenciji od 50 Hz, što odgovara m_a = 1.0, ako se uzme u obzir konverziju 16:1 s izlaza HIL402. Također, na gornjem dijelu slike prvi značajni harmonik se pojavljuje u rangu od 1200 Hz, što je ispravno za m_f = 24. Na donjem dijelu slike je m_f = 48, te se prvi značajni harmonik nalazi u rangu od 2400 Hz. Time je potvrđeno da se prvi značajni harmonik pojavljuje na m_f - tom višekratniku frekvencije osnovnog harmonika, te ako se m_f udvostruči, udvostručit će se i frekvencija na kojoj se pojavi prvi harmonik.

b) Harmonici izlaznog napona zadovoljavaju parametre iz tablice 3.1.

Za proučavanje točnosti tablice, korišten je signal m_f = 24, m_a = 0.8. Dobiveni stupčasti prikaz harmonika je prikazan na slici 5.7.



Slika 5.7. Harmonijski prikaz za m_a = 0.8, m_f = 24.

Očitavajući pojedine harmonike sa slike, te koristeći izraz 5-1, moguće je izračunati m_a za pojedinačne harmonike unutar grafa.

$$m_a = \frac{2\hat{U}(\boldsymbol{mf})}{E} = \frac{2U * \sqrt{2}}{E} \tag{5-1}$$

Svaki plavi stupac na grafu prikazuje pojas od 50 Hz. Primjer izračuna iz jednadžbe je sljedeći:

$$m_a = \frac{2\hat{U}(\boldsymbol{mf})}{E} = \frac{2U * \sqrt{2}}{E} = \frac{2 * 5{,}65 * \sqrt{2}}{20} = 0{,}797;\ za\ m_f \tag{5-2}$$

U tablici 5.1. su prikazani teorijski rezultati m_a zajedno sa izračunatim.

Tablica 5.1. Izračun m_a pojedinih harmonika

Harmonik	m_a (tablica)	Amplituda [V]	m_a (izračun)
1	0.8	7,9	0.79
m_f	0.818	5.65	0.797
$m_f \pm 2$	0.220	1.5	0.366
$2m_f \pm 1$	0.314	2.78	0.391
$2m_f \pm 3$	0.139	0.95	0.134
$2m_f \pm 5$	0.013	/	/
$3m_f$	0.171	1.28	0.180
$3m_f \pm 2$	0,176	0.77	0.108
$3m_f \pm 4$	0,104	0.55	0.077
$3m_f \pm 6$	0,016	/	/
$4m_f \pm 1$	0.105	0.8	0.1128
$4m_f \pm 3$	0.115	0.42	0.059
$4m_f \pm 5$	0.084	/	/
$4m_f \pm 7$	0.017	/	/

Gledajući u pojedine izračune tablice, može se primijetiti kako u većini slučajeva izračun iz grafa harmonika daje približne rezultate kao i teorijska tablica izračuna. Neizračunati dijelovi tablice predstavljaju harmonike čija je amplituda preniska, te su rezultati izračuna nepouzdani, ili nemogući za napraviti. Veća odstupanja od teorijskog m_a potječu zbog nepravilnosti instrumenta i očitanja, te zbog određenih „šiljaka“ i „propada“ izlaznog naponskog signala, što je primijećeno kao nedostatak rješenja, a što će biti detaljno objašnjeno u sljedećem poglavlju.

Izračun THDu se obavlja pomoću izraza 5-3.

$$THDu = \sqrt{\frac{U^2}{U^2(1)} - 1} \tag{5-3}$$

Koristeći izraz 5-3, dobiva se da je THDu za dobivsa teni graf:

THDu = 1.484 = 148.4%, što je zadovoljavajući rezultat, ako se usporedi s teorijskom izračunatom vrijednošću THDu za m_a = 0.8, koja iznosi 145, 77% [2].

5.3. Testiranje

Testiranje programa se vršilo na dva načina :

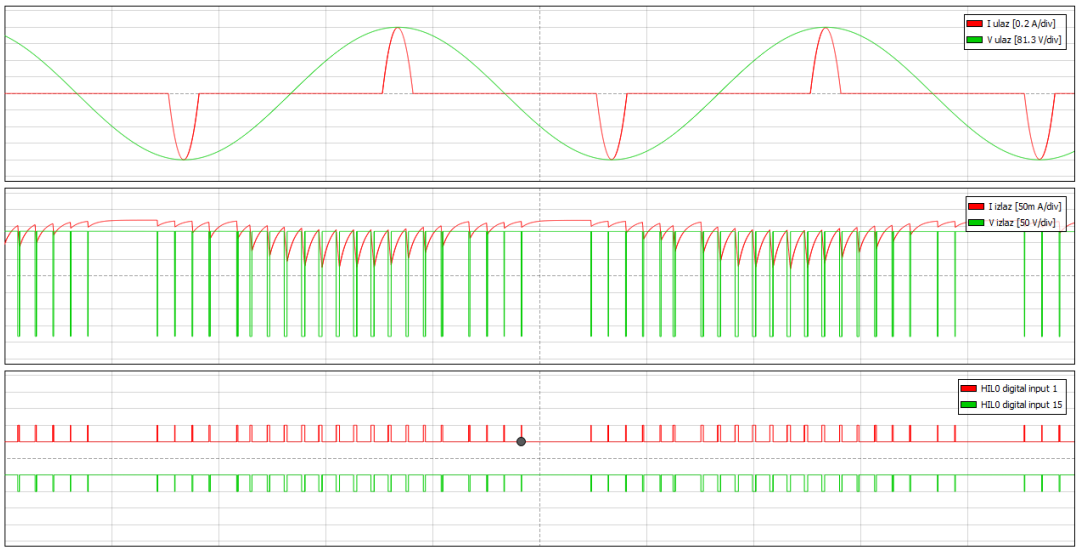
1) promjenom m_a dok je m_f konstantan

2) promjenom m_f dok je m_a konstantan

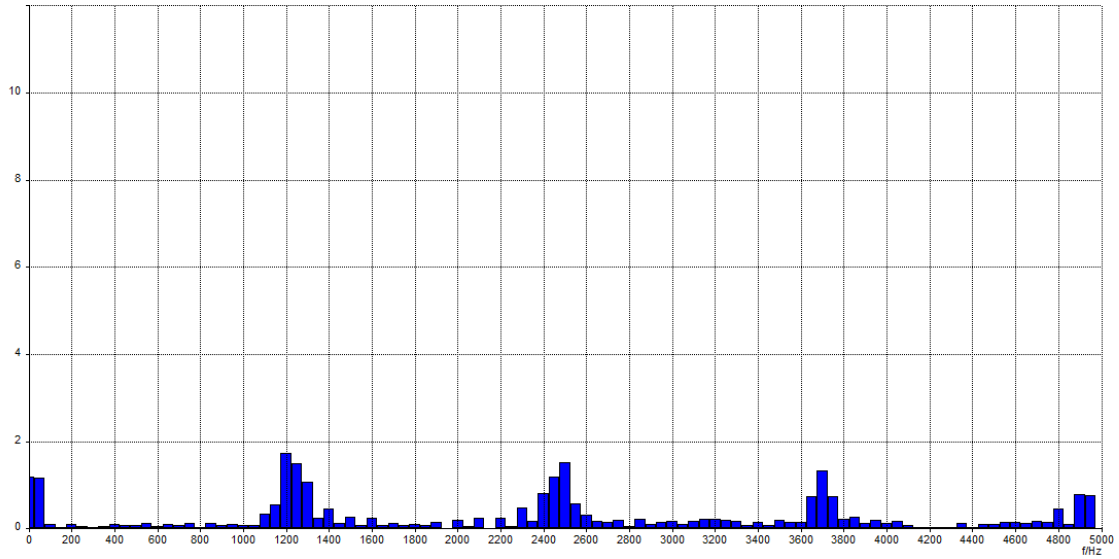
Mijenjanje parametara vršilo se isključivo odabirom na tipkovnici, što znači da se parametri sheme ne mijenjaju, te ostaju :

$$\boldsymbol{E = 230\ [V]} \qquad \boldsymbol{f = 50\ [Hz]} \qquad \boldsymbol{R = 800\ [\Omega]} \qquad \boldsymbol{L = 330\ [mH]}$$

Na slici 5.8. se nalaze karakteristični valni oblici odabrani za analizu. Odabrani su parametri m_a = 0.2 i m_f = 24.Na prvom sinkogramu se nalaze valni oblici ulaznog napona i struje, na drugom se nalazi izlazni napon i struja, a na trećem upravljački signali. Valni oblik izlazne struje uopće ne sliči sinusnom obliku, te se ovakva kombinacija amplitudnog i frekvencijskog indeksa modulacije ne preporučuje za korištenje u praksi. Harmonijska analiza na slici 5.9. pokazuje prvi značajni harmonik na 1200 Hz, te mu je amplituda vrlo visoka, više od 100% osnovnog harmonika (skoro 2V, imajući na umu 16:1 omjer konverzije s izlaza HIL 402).



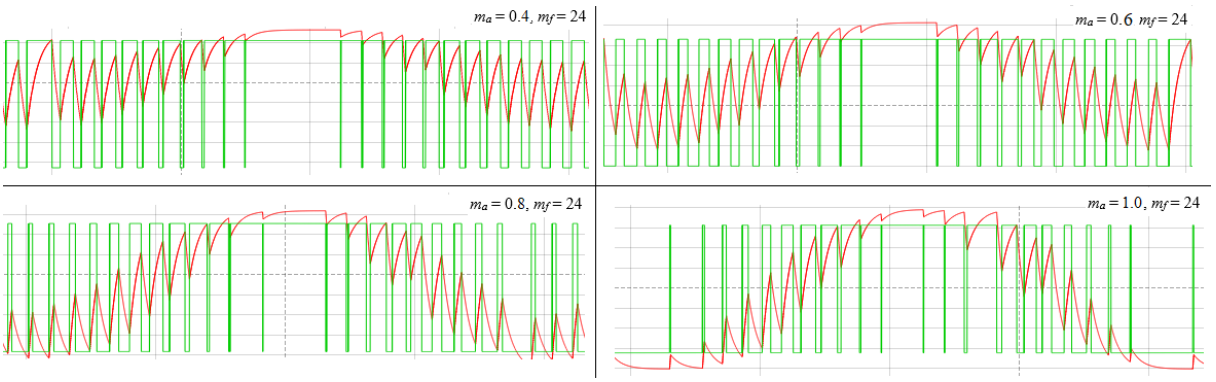
Slika 5.8. Odabrani karakteristični valni oblici pripremljeni za analizu, m_a = 0.2, m_f = 24



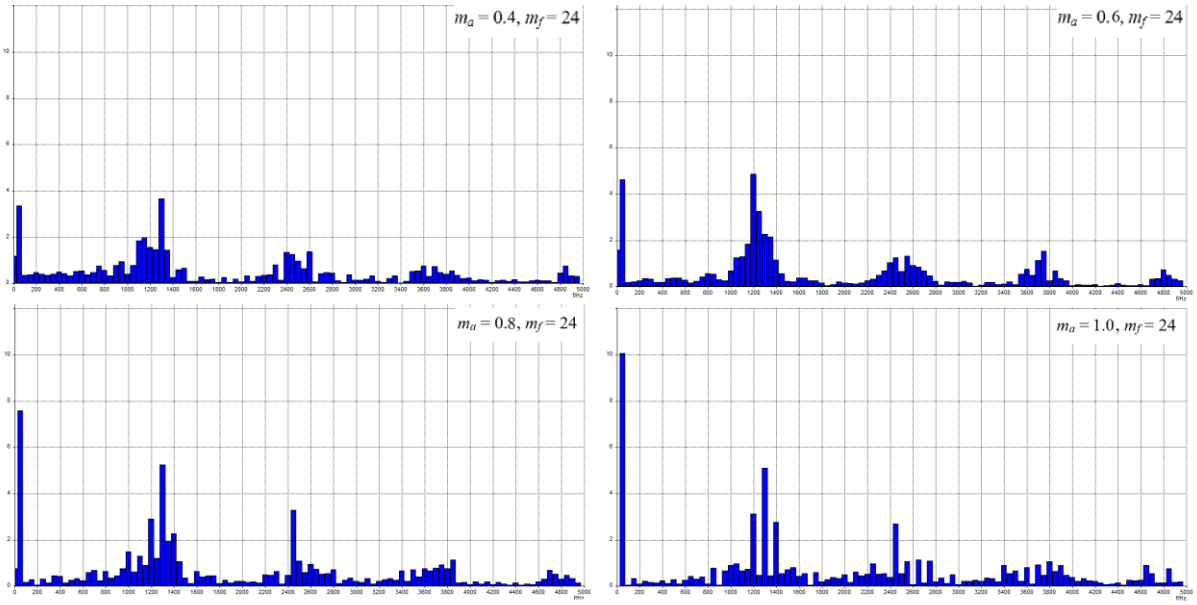
Slika 5.9. Harmonijski prikaz za m_a = 0.2, m_f = 24

1) Promjena m_a uz konstantni m_f = 24.

Na slici 5.10. je prikazana po jedna perioda valnog oblika izlaznog napona i struje. Iz date slike može primijetiti kako povećanjem modulacijskog indeksa m_a valni oblik izlazne struje počinje više sličiti sinusnom obliku, što znači da se popravlja THDI. Iz stupčastog prikaza harmonika se može zaključiti kako povećanjem m_a raste amplituda osnovnog i dominantnog harmonika. Dominantni harmonik se nalazi na frekvenciji od 1200 Hz, što je uvjetovano frekvencijskim indexom *mf* = 24. Stupčasti prikaz harmonika prikazan je na slici 5.11.



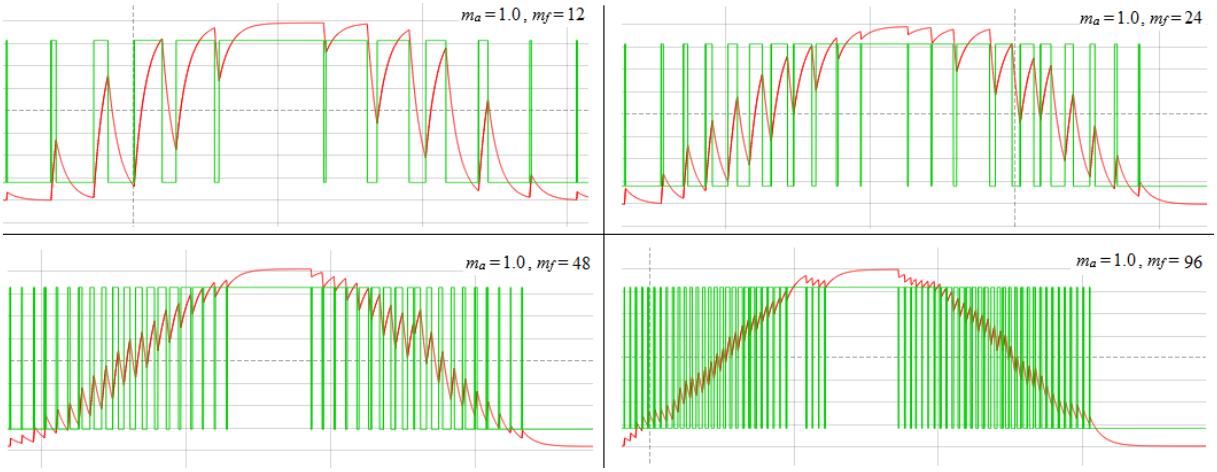
Slika 5.10. Valni oblici pri mijenjanju m_a uz konstantan parametar m_f = 24



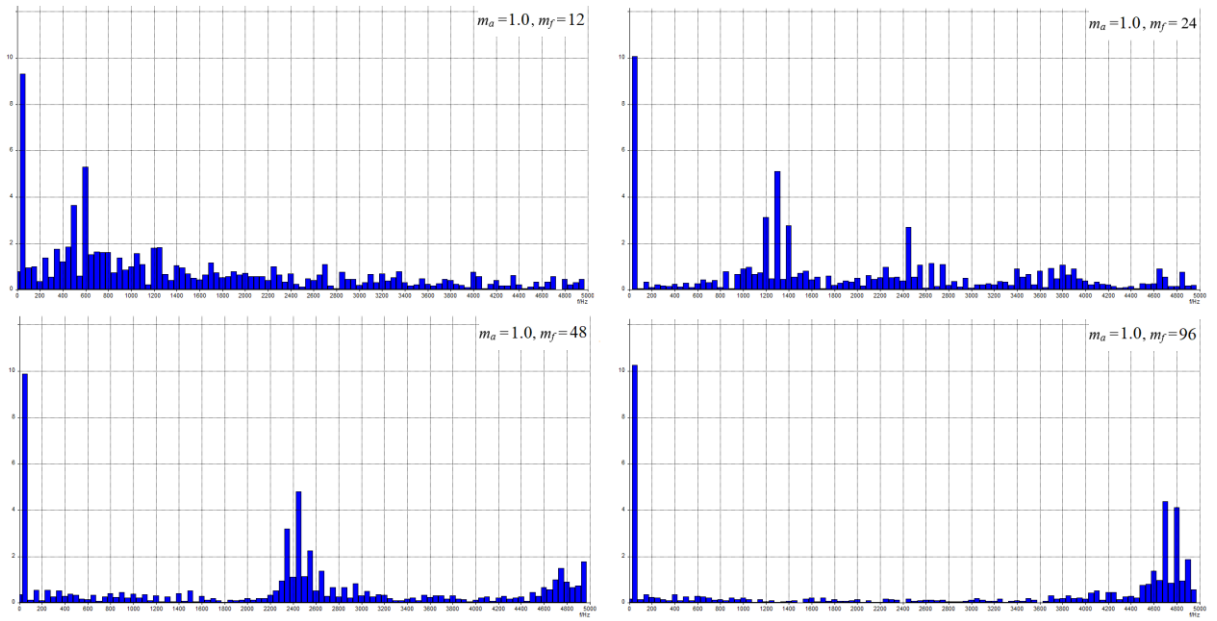
5.11. Harmonijski prikaz izlaznog napona pri mijenjanju m_a uz konstantan parametar $m_f = 24$

2) Promjena m_f uz konstantni $m_a = 1.0$

Iz slike 5.12. se može zaključiti kako povećanjem frekvencijskog indeksa upravljački signali povećavaju svoju gustoću izmjene i to rezultira poboljšanjem THDI za veće frekvencijske indekse. Iz stupčastog prikaza harmonika je očigledno da se povećanjem m_f dominantni harmonik gura u više frekvencijsko područje, dajući bolji THDI. Amplituda harmonika ostaje približno ista, neovisno o frekvencijskom indeksu. Stupčasti prikaz harmonika je prikazan na slici 5.13.



Slika 5.12. Valni oblici pri mijenjanju m_f uz konstantan parametar m_a = 1.0



5.13. Harmonijski prikaz izlaznog napona pri mijenjanju m_f uz konstantan parametar m_a = 1.0

5.4. Analiza dobivenih rezultata

Analiza dobivenih rezultata se slično kao i u prošlom poglavlju vrši za dva slučaja :

1) Analiza promjene m_a uz konstantni m_f = 24.

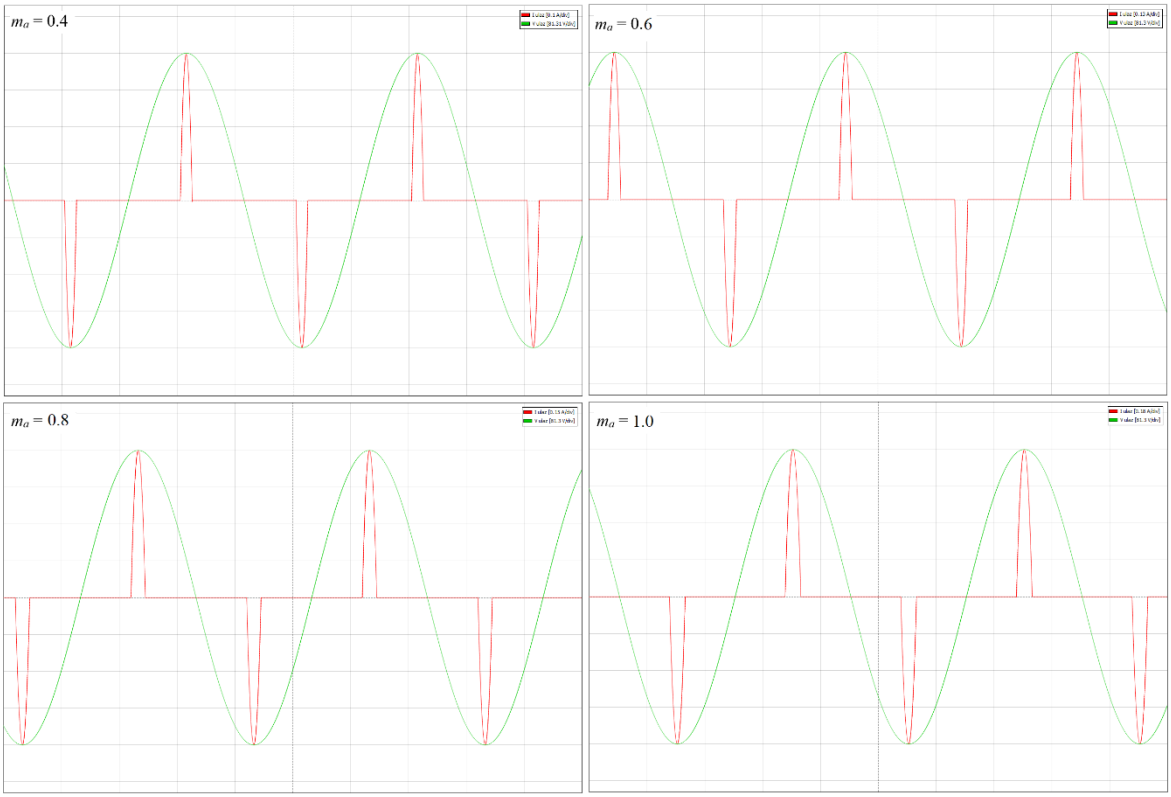
2) Analiza promjene m_f uz konstantni m_a = 1.0.

Ostali parametri ostaju nepromijenjeni, te iznose :

E = 230 [V] **f = 50 [Hz]** **R = 800 [Ω]** **L = 330 [mH]**

1) Analiza promjene m_a uz konstantni m_f = 24.

a) Ulazni napon i struja



Slika 5.14. Usporedba valnih oblika ulaznog napona i struje pri različitim m_a, m_f = 24.

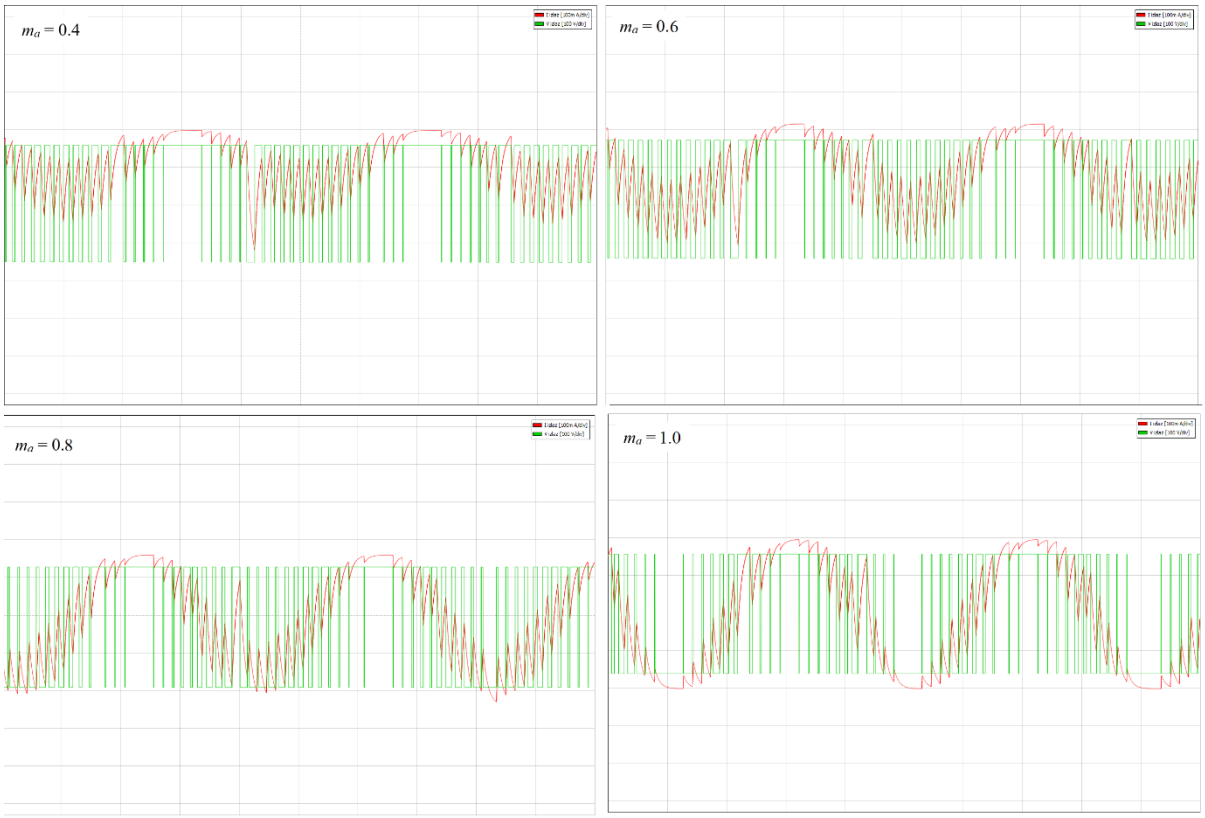
Na slici 5.14. su prikazani valni oblici ulaznog napona i struje za različite modulacijske indekse. Kako je i očekivano, zbog toga što se u testiranju nije mijenjao nijedan parametar unutar sheme, pa tako ni ulazni napon, pri testiranju se dobiju kvalitativno slični grafovi za ulazni napon i struju.

b) Upravljački signali

Slika 5.15. Usporedba upravljačkih signala pri različitim m_a, m_f = 24.

Na slici 5.15. se nalaze upravljački signali pri različitim modulacijskim indeksima. Pri niskoj vrijednosti m_a može se primijetiti kako na vrata prvog IGBT-a dolaze upravljački signali kojima su vrlo kratki impulsi jedinica, a mnogo duži impulsi nula, dok na vrata drugog IGBT-a dolazi komplementirani signal. Povećanjem m_a vremensko trajanje jedinica na prvom IGBT-u se povećava, dok se na vratima drugog IGBT-a smanjuje.

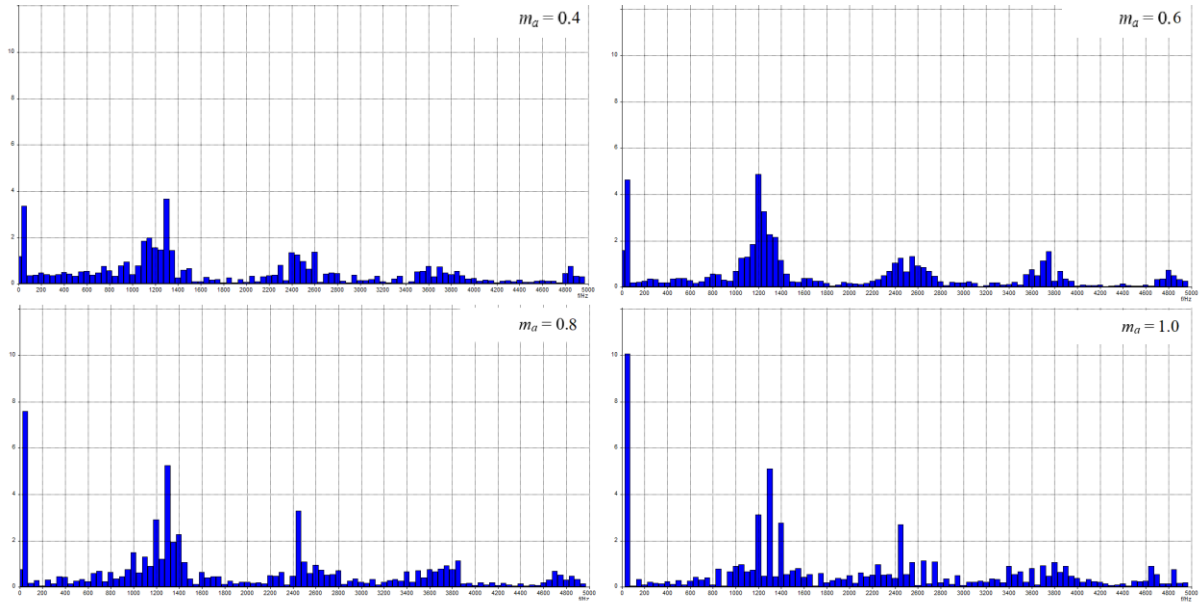
c) Izlazni napon i struja



Slika 5.16. Usporedba valnih oblika izlaznog napona i struje pri različitim m_a, $m_f = 24$.

Iz slike 5.16. se može primijetiti kako povećanjem m_a valni oblik izlazne struje poprima sve bliži oblik sinusnom valnom obliku. Pri niskim razinama m_a izlazna struja ima valoviti sinusni oblik, no s izraženom istosmjernom komponentom. Povećanjem m_a očito je po valnom obliku da se poboljšava THDI, a struja postaje čisto izmjenična.

d) Harmonijska analiza

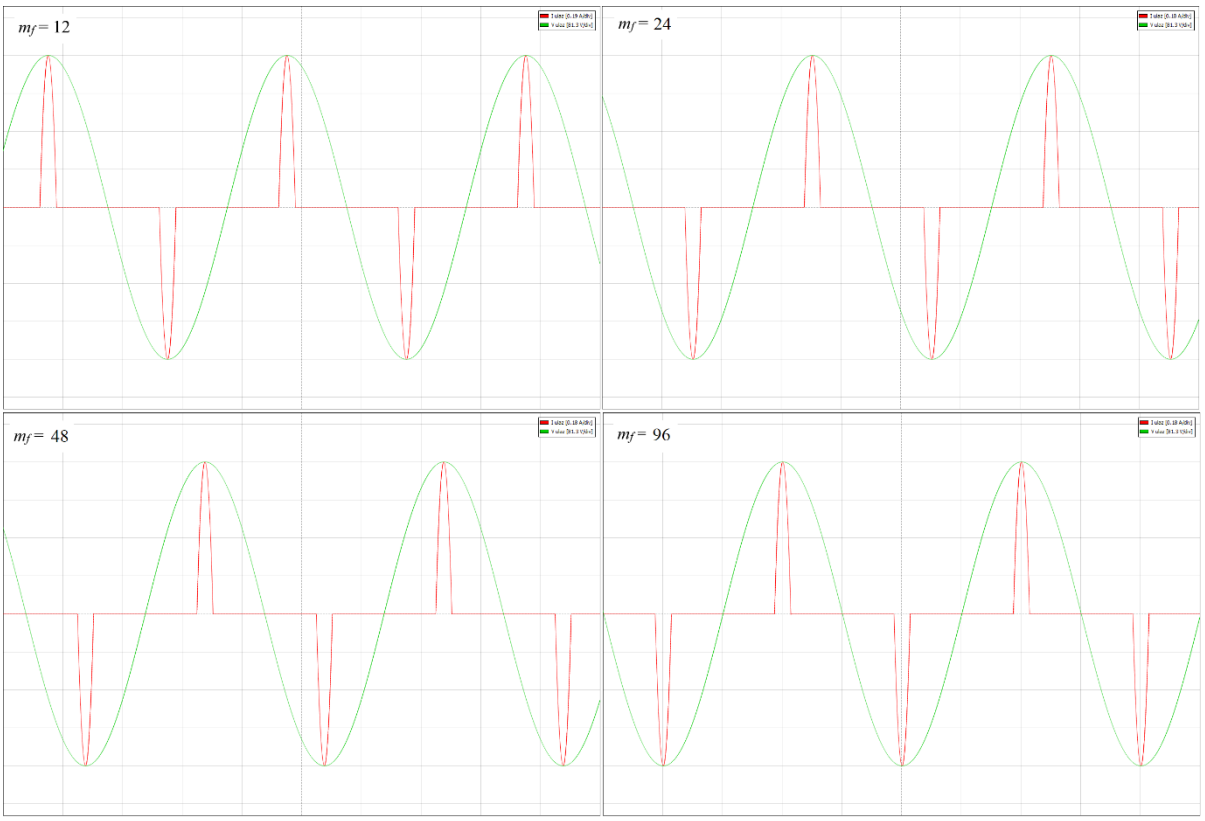


Slika 5.17. Harmonijska analiza izlaznog napona pri različitim m_a, m_f = 24.

Iz harmonijske analize prikazane na slici 5.17. se uočava kako povećanjem m_a raste amplituda osnovnog harmonika. Zbog omjera 16:1 na izlazu HIL 402 sklopovlja, maksimalni napon prikazan harmonijskom analizom je 10 V za m_a = 1.0. Za niže m_a, amplituda osnovnog harmonika proporcionalno opada. Dominantni harmonik se uvijek nalazi na 1200 Hz, zbog iznosa m_f =24. Amplituda dominantnog harmonika nije uvijek proporcionalna amplitudi osnovnog harmonika, te je amplituda dominantnog harmonika veća u odnosu na osnovni harmonik za m_a = 0.4 i 0.6, dok je za m_a = 0.8 i 1.0 amplituda dominantnog harmonika manja od amplitude osnovnog harmonika.

2) Analiza promjene m_f

a) Ulazni napon i struja



Slika 5.18. Usporedba valnih oblika ulaznog napona i struje pri različitim m_f, m_a = 1.0.

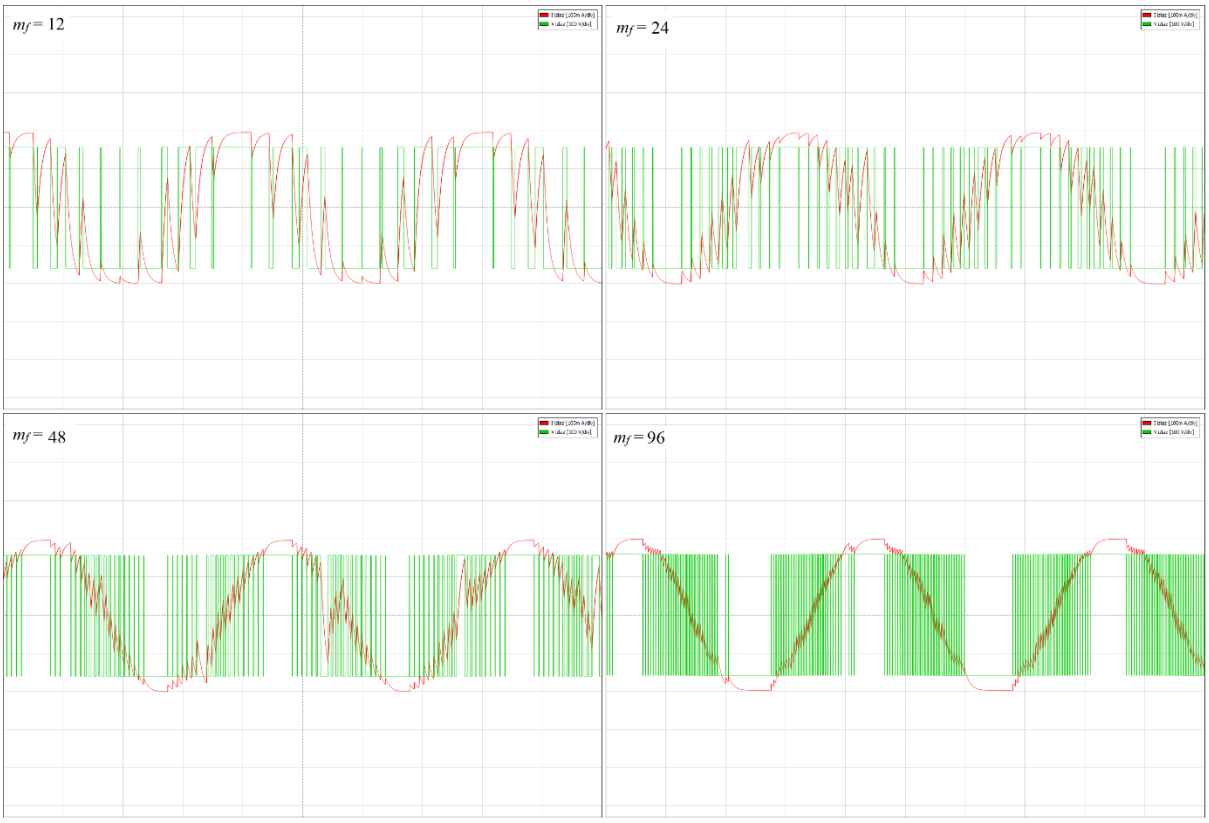
Na slici 5.18 su prikazani valni oblici ulaznog napona i struje za različite m_f. Kao što je slučaj u promjeni amplitudnog indeksa, pri promjeni frekvencijskog indeksa valni oblici ulaznih napona i struje su slični pri sva četiri frekvencijska indeksa. Razlog za to su konstantni parametri trošila, kao i izvora.

b) Upravljački signali

Slika 5.19. Usporedba upravljačkih signala pri različitim m_f, m_a = 1.0.

Na slici 5.19. primijećeno je da porastom frekvencijskog indeksa upravljački signali mijenjaju svoju gustoću izmjene. Pri m_f = 96, broj izmjena je višestruko veći u odnosu na m_f = 12, što rezultira kraćim vremenima vođenja IGBT-ova, odnosno bržim sklapanjima.

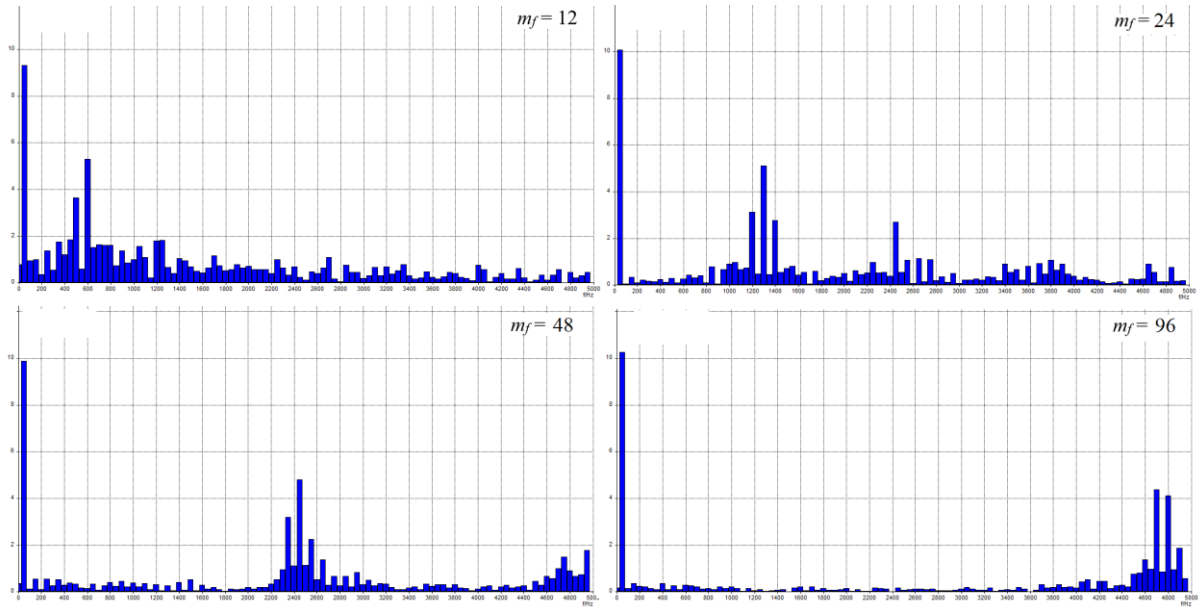
c) Izlazni napon i struja



Slika 5.20. Usporedba valnih oblika izlaznog napona i struje pri različitim m_f, m_a = 1.0.

Iz slike 5.20. očigledno je da povećanjem frekvencijskog indeksa valni oblik izlazne struje počinje ima bolji THDI. To se događa zbog toga što povećanje m_f pomjera ostale značajne harmonike u više frekvencije, gdje je gušenje prouzrokovano trošilom (RL sklop) puno veće. U slučaju niskog m_f naponski impulsi traju puno duže u odnosu na visoki m_f. To u praksi znači da struja ima više vremena povećati ili smanjiti svoj iznos.

d) Harmonijska analiza

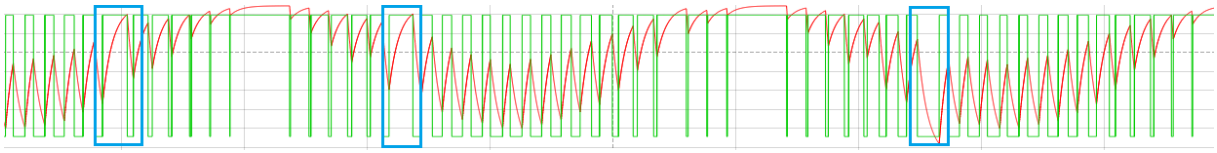


Slika 5.21. Harmonijska analiza izlaznog napona pri različitim m_f ,m_a = 1.0.

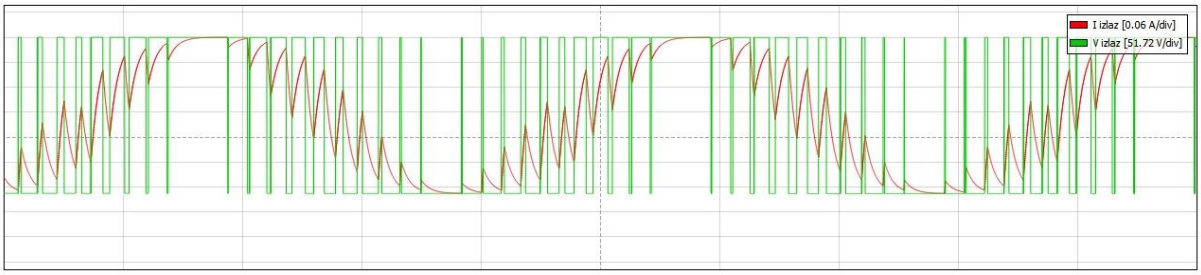
Iz slike 5.21. uočeno je kako porastom frekvencijskog indeksa prvi harmonik se gura u više frekvencije. Za m_f = 12 prvi harmonik se nalazi na frekvenciji od 600 Hz, a za m_f = 96 prvi harmonik se nalazi na frekvenciji od 4800 Hz. Zbog guranja prvih harmonika u više frekvencije, njihovo prigušivanje postaje veće zbog induktivnog trošila, ostavljajući osnovni harmonik znatno većim u odnosu na dominantni harmonik, samim time dobiva se valni oblik izlazne struje koji mnogo više sliči monoharmonijskom u odnosu na signale koji proizilaze iz nižeg m_f.

5.5. Nedostatci predloženog upravljanja

Nepravilnosti koje se uočavaju na valnim oblicima izlaznog napona, pa tako i izlazne struje, posljedice su provjere rada tipkovnice u LOOP dijelu Arduino koda. Zbog toga što program konstantno nakon svakog prosljeđivanja pojedinog izlaza provjerava je li na tipkovnici pritisnuta bilo koja tipka, te u slučaju da jest odradi dio koda i zauzme određeni dio vremena rada procesora, uslijed čega se produžava trajanje maksimalnog ili minimalnog iznosa napona. Time izlazna struja dobije dodatno vrijeme za povećati ili smanjiti svoj iznos, zbog čega se javljaju određeni „šiljci“ i „propadi“ struje, a zbog čega se produžuje/smanjuje vrijeme vođenja IGBT-a, što u konačnici narušava THDI. Ispravak ovog nedostatka otežava naizgled nasumično pojavljivanje nepravilnosti. Na slici 5.22. se mogu primijetiti spomenute nepravilnosti, dok je na slici 5.23. prikazan valni oblik prije implementiranja tipkovnice.

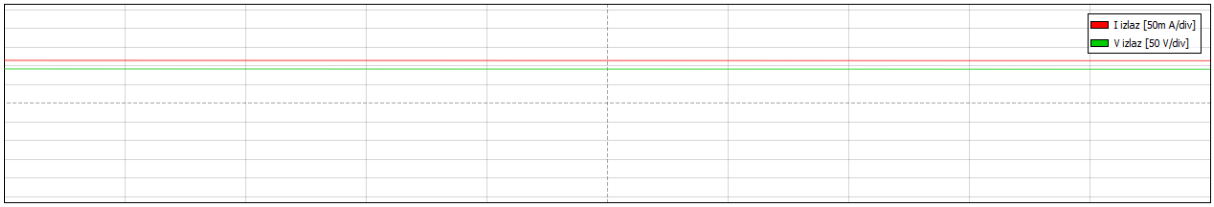


Slika 5.22. Primjeri nepravilnosti valnog oblika izlazne struje



Slika 5.23. Valni oblik izlaznog napona i struje prije ugradnje tipkovnice

Još jedan od potencijalnih nedostataka koji se pojavljuje je naglo zaustavljanje rada sustava. To se događa u dva slučaja, pri aktiviranju Pauza načina rada, te ponovnim pokretanjem sustava. Pri naglom isključivanju upravljačkih signala, na izlazu se pojavi istosmjerni napon iznosa 160V, koji može biti opasan za stvarnu maketu, jer tada izlazna impedancija gubi svoj reaktivni dio, ostavljajući relativno nizak otpor, čime se na izlazu pojavljuju velike struje, koje mogu uništiti komponente makete. Iako se izlazni napon i struja s vremenom smanjuju putem zatvaranja izlaznog RL kruga spoja, postoji rizik od uništenja. Izlazni napon i struja nakon ponovnog pokretanja su prikazani na slici 5.24.



Slika 5.24. Valni oblik izlaznog napona i struje neposredno nakon ponovnog pokretanja

5.6. Prijedlog rješenja nedostatka pri sljedećoj doradi makete

Glavni problem ovakve izvedbe je korištenje funkcije delayMicroseconds. Ova funkcija, kao i svaki druga delay funkcija u Arduinu, zauzima procesorsko vrijeme. Zbog nemogućnosti jednoznačnog određivanja gdje se nepravilnosti struje se točno pojavljuju, te koliko učestalo, za doradu se programski kod treba doraditi. To bi se postiglo izbacivanjem funkcije delayMicroseconds, te ubacivanjem prekidnih (interrupt) funkcija. Prekidne funkcije ne zauzimaju procesorsko vrijeme, nego izvrše određenu naredbu kada se prekid pozove. Ukoliko procesor vrši neke druge radnje, a prekidna funkcija se pozove, procesor prestaje s izvršavanjem trenutne radnje, izvrši naredbu određenu prekidom, te nakon njenog izvršenja nastavlja s radom koji je bio prekinut.

Drugi problem koji se može pojaviti prije ugradbe sustava u fizičku maketu je problem pauze. Pauziranje u ovom sustavu daje na izlazu istosmjerni napon visoke vrijednosti što može uzrokovati uništenje makete. To se može popraviti implementacijom neke vrste tzv. Metoda mekog pokretanja unutar koda. Prilikom pokretanja, moguće je postupno povećavati m_a dok se ne dođe do odabranog m_a na tipkovnici. Također, jednako vrijedi za zaustavljanje rada, gdje bi se m_a postupno smanjivao. Testiranjem bi se utvrdilo koliko perioda određenog m_a je potrebno da izlazni signal napravi zadovoljavajuću tranziciju prema krajnjem m_a.

Dodatno rješavanje problema ponovnog pokretanja se može riješiti paralelnim računanjem nove tablice rezultata dok se program još izvršava. Program bi tada računao, te mijenjao nove rezultate u postojeću tablicu rezultata, dok bi prekidna funkcija ispisivala stare rezultate dok se novi zapisuju u tablicu rezultata, te kada se zamjena u potpunosti izvršena, program bi na izlazu ispisivao nove rezultate. Drugo moguće rješenje je upotreba „hard-coded“ tablica rezultata koji su prethodno izračunati, te spremljeni u memoriju (radnu ili flash). Problem ove izvedbe je relativno veliko zauzeće memorije, gdje bi svaka tablica uzela minimalno 1000 bajtova memorije (za tablicu rezultata se koriste samo 0 i 1 zbog čega je dovoljno koristiti uint8_t tip podataka, čija jedna vrijednost iznosi 1 bajt). U ovom slučaju je potrebno 20 tablica (kombinacija 5 m_a i 4 m_f), te bi to zauzelo 20 kB SRAM memorije, dok Arduino Mega posjeduje 8 kB SRAM memorije. Postoji mogućnost zapisa u FLASH memoriju programa, no nije testirano koliko čitanje iz programske memorije uzima vremena, te je li to kompatibilno s trenutnim razvojem programa.

Prethodni problem je rješiv korištenjem moćnijeg mikrokontrolera ili upotrebom malog jednopločnog računala, kao što je Raspberry Pi. Raspberry Pi posjeduje mnogo moćniji procesor, te višestruko više RAM memorije. Standardni model, Raspberry Pi 3, posjeduje 1GB RAM-a i procesor od 1.2 GHz, u odnosu na Arduino Mega 2560, koji posjeduje 8 kB RAM-a i procesor od 16 MHz. Korištenjem jednog ovakvog jednopločnog računala koji posjeduje moćniji procesor bi se mogla povećati rezolucija ispisa na izlazu, jer bi procesor mogao davati više izlaza u istom vremenskom periodu u odnosu na Arduino.

6. ZAKLJUČAK

Arduino se pokazao kao dobrom platformom za upravljanje jednom granom trofaznog izmjenjivača pomoću pulsno-širinske modulacije. Broj priključnica koji su potrebni za priključivanje na Arduino je dovoljan i na manje procesorski moćnim tipovima, no za predloženo rješenje je uzet Arduino Mega 2560 zbog svoje veće SRAM memorije (8 kB).

Typhoon HIL programski paket se pokazao dobrim za emulaciju jednofaznog mosnog spoja. Unutar Typhoon HIL-a je izrađena shema, te su postavljeni instrumenti za očitavanje karakterističnih veličina kao i grafovi. Nedostatak Typhoon HIL sučelja je nemogućnost prikaza harmonijskog sadržaja izlaznog napona, pa je zbog tog korišten dodatni uređaj, UniTrain S04203-2A. Priključen na izlaz iz HIL 402 sklopovlja, UniTrain-om se pomoću Starter VI programa može izvršiti harmonijska analiza.

Testiranjem programa dobivaju se zadovoljavajući valni izlazi, uz nekoliko nasumičnih odstupanja. To se događa zbog produžavanja trajanja izlaznog signala pri provjeravanju tipkovnice u LOOP funkciji Arduino koda. Provjera tipkovnice zauzima određeno procesorsko vrijeme, što rezultira produžavanjem trajanja izlaznog signala. Harmonijska analiza pokazuje zadovoljavajuće rezultate kod promjene m_f i promjene m_a, uz neka nasumična odstupanja kod amplituda pojedinih harmonika.

Rješenje problema odstupanja u valnom obliku izlaznog napona se može riješiti upotrebom prekidnih (interrupt) rutina umjesto funkcije delayMicroseconds. Tako bi se provjera tipkovnice izvršavala u pozadini, dok bi se prekidna rutina pozivala i ispisivala na izlazu neovisno o ostatku rada programa. Za ugradnju u stvarnu maketu izmjenjivača će vjerojatno biti potrebna ugradnja procedure mekog pokretanja. To se može riješiti postupnim povećavanjem ili smanjivanjem amplitudnog indeksa. Poboljšanje rezolucije upravljačkih, a tako i naponskih signala se može postići upotrebljavanjem mikrokontrolera/jednopločnog računala koji ima veću procesorsku moć, npr. Raspberry Pi 3.

LITERATURA

[1] https://www.arduino.cc/en/guide/introduction

[2] M. Pavin, Pulsno-širinska modulacija autonomnog, trofaznog, naponskog izmjenjivača, Osijek, 2018.

[3] V. Špica, Tehničke upute za rad s autonomnim izmjenjivačem, Osijek, 2000.

[4] A. E. TOUBAL MAAMAR, Design and Control of a Single-Phase Series Resonance Inverter using an Arduino Microcontroller, Chlef, Algeria, 2019.

[5] S. A. Zulkifli, MATLAB-Arduino as a Low Cost Microcontroller for 3 phase inverter, Batu Pahat, 2014.

[6] S. Pal, Modeling of Solar Energy Grid Integration System Using Typhoon HIL, Krabi, Thailand, 2018.

[7] I.Flegar, Elektonički energetski pretvarači, Kigen d.o.o., Zagreb, 2010.

[8] https://www.typhoon-hil.com/

SAŽETAK

Opisano je upravljanje jednom granom trofaznog izmjenjivača pomoću Arduina. Napravljena je teorijska podloga za analizu rada. Cilj rada je bio napraviti zadovoljavajuću pulsno-širinsku modulaciju na Arduino mikrokontroleru koja će davati IGBT-ovima upravljačke signale. Nakon toga je napravljen pregled sklopa. Dat je pregled komponenata, objašnjenje shemi, objašnjenje koda, te pregled korištenih alata, Typhoon HIL programskog paketa i Lucas Nulle Starter VI. Unutar Typhoon HIL-a su implementirani shema i pregled grafova, dok je u Lucas Nulle-u izvršena harmonijska analiza. Testirana su dva slučaja, sa konstantnim frekvencijskim indeksom i promjenjivim amplitudnim indeksom, te obrnuti, konstantan amplitudni indeks sa promjenjivim frekvencijskim indeksom. Napravljena je analiza osjetljivosti upravljanja s obzirom na modulacijski indeks i dubinu modulacije. Definirani su nedostatci predloženog upravljanja i ponuđena su rješenja uočenih nedostataka pri ponovnoj doradi makete.

Ključne riječi : izmjenjivač, pulsno-širinska modulacija, Arduino, harmonijska analiza, Raspberry Pi

ABSTRACT

Control with switching components of the one leg three phase SPWM inverter by Arduino or Raspberry Pi platform

This thesis describes the control of a single phase of the three-phase inverter using an Arduino microcontroller. Theoretical basis for the problem has been made in the beginning of the thesis. The aim of this work was to make a sufficient pulse-width modulation on the Arduino board which will in turn give IGBT-s control signals. After that, an overview of the board has been made, followed up by an overview of components used, explanation of the circuit, explanation of the code and finally the overview of used programs, Typhoon HIL and Lucas Nulle. Circuit and the graph overview has been made within Typhoon HIL, while the spectral analysis has been made by the Lucas Nulle instrument and software. Testing has been made in two categories, one with a constant frequency index and a variable modulation index, while the other category the process has been reversed, having a constant modulation index with a variable frequency index. Sensitivity analysis of the controller was made while changing the modulation indeks and the depth of the modulation. Drawbacks of this solution have been defined and solutions have been offered to improve future models.

Key words : inverter, pulse-width modulation, Arduino, spectral analysis, Raspberry Pi

ŽIVOTOPIS

Petar Galić je rođen 17. srpnja 1996. godine u Žepču. Pohađao je Osnovnu školu „Žepče“ u Žepču. Srednju školu je upisao u Katoličkom Školskom Centru „don Bosco“ u Žepču, smjer Tehničar za mehatroniku. Radio je praksu 2 godine u JP „Elektroprivreda HZHB“. Sudjelovao je na Državnom natjecanju u Osnovama elektrotehnike u Republici Hrvatskoj 2013. godine, te na Federalnom natjecanju iz matematike u BiH 2015. godine. Oslobođen je školske mature zbog odličnog uspjeha sve 4 godine. 2015. upisuje preddiplomski studij elektrotehnike na FERIT-u. 2018. sudjeluje na STEM Gamesima, u kategoriji Engineering. 2018. završava preddiplomski studij te stječe titulu Prvostupnika elektrotehnike i informacijske tehnologije. Nakon toga upisuje diplomski studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, smjer Automobilsko računarstvo i komunikacije. Član je IEEE Student Branch Osijek.

PRILOZI

Prilog 1 : Arduino kod

```
1.  #include <U8g2lib.h> //include za ekran
2.  #include <U8x8lib.h>
3.  #include <Arduino.h>
4.
5.
6.  #ifdef U8X8_HAVE_HW_SPI
7.  #include <SPI.h>
8.  #endif
9.  #ifdef U8X8_HAVE_HW_I2C
10. #include <Wire.h>
11. #endif
12.
13. #include <Keypad.h> //include za tipkovnicu
14.
15. //inicijalizacija ekrana
16. U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 18 /* A4 */ , /* data=*/ 16 /*
     A2 */, /* CS=*/ 17 /* A3 */, /* reset=*/ U8X8_PIN_NONE);
17.
18. //inicijalizacija tipkovnice
19. const byte ROWS = 4;
20. const byte COLS = 4;
21. char keys[ROWS][COLS] = {
22.   {'1', '2', '3', 'A'},
23.   {'4', '5', '6', 'B'},
24.   {'7', '8', '9', 'C'},
25.   {'*', '0', '#', 'D'}
26. };
27. byte rowPins[ROWS] = {10, 9, 7, 6}; //connect to the row pinouts of the keypad
28. byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
29. Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
30.
31. //globalne varijable
32. float ma;
33. uint8_t mf;
34. char key, gkey, dkey;
35. uint8_t rezultat[1000];
36.
37. //deklaracije funkcija
38. void unosParametara();
39. void izracunVrijednosti(float ma, uint8_t mf);
40. void pauza();
41.
42. //tablice za usporedbu
43. const PROGMEM uint16_t SINE_TABLE[] = {500, 503, 506, 509, 513, 516, 519, 522, 525,
    528, 531, 535, 538, 541, 544, 547, 550, 553, 556, 560, 563, 566, 569, 572, 575, 578,
     581, 584, 588, 591, 594, 597, 600, 603, 606, 609, 612, 615, 618, 621, 624, 627, 630
    , 633, 636, 639, 643, 646, 649, 652, 655, 657, 660, 663, 666, 669, 672, 675, 678, 68
    1, 684, 687, 690, 693, 696, 699, 701, 704, 707, 710, 713, 716, 719, 721, 724, 727, 7
    30, 733, 735, 738, 741, 744, 746, 749, 752, 755, 757, 760, 763, 765, 768, 771, 773,
    776, 778, 781, 784, 786, 789, 791, 794, 796, 799, 801, 804, 806, 809, 811, 814, 816,
     819, 821, 824, 826, 828, 831, 833, 835, 838, 840, 842, 845, 847, 849, 851, 854, 856
    , 858, 860, 862, 864, 867, 869, 871, 873, 875, 877, 879, 881, 883, 885, 887, 889, 89
    1, 893, 895, 897, 899, 901, 903, 905, 906, 908, 910, 912, 914, 915, 917, 919, 920, 9
    22, 924, 925, 927, 929, 930, 932, 934, 935, 937, 938, 940, 941, 943, 944, 946, 947,
    948, 950, 951, 952, 954, 955, 956, 958, 959, 960, 961, 963, 964, 965, 966, 967, 968,
     969, 970, 971, 973, 974, 975, 976, 976, 977, 978, 979, 980, 981, 982, 983, 984, 984
    , 985, 986, 987, 987, 988, 989, 989, 990, 991, 991, 992, 992, 993, 993, 994, 994, 99
    5, 995, 996, 996, 996, 997, 997, 997, 998, 998, 998, 999, 999, 999, 999, 999, 1000,
```

```
    1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000,
    999, 999, 999, 999, 999, 998, 998, 998, 997, 997, 997, 996, 996, 996, 995, 995, 994,
     994, 993, 993, 992, 992, 991, 991, 990, 989, 989, 988, 987, 987, 986, 985, 984, 984
    , 983, 982, 981, 980, 979, 978, 977, 976, 976, 975, 974, 973, 971, 970, 969, 968, 96
    7, 966, 965, 964, 963, 961, 960, 959, 958, 956, 955, 954, 952, 951, 950, 948, 947, 9
    46, 944, 943, 941, 940, 938, 937, 935, 934, 932, 930, 929, 927, 925, 924, 922, 920,
    919, 917, 915, 914, 912, 910, 908, 906, 905, 903, 901, 899, 897, 895, 893, 891, 889,
     887, 885, 883, 881, 879, 877, 875, 873, 871, 869, 867, 864, 862, 860, 858, 856, 854
    , 851, 849, 847, 845, 842, 840, 838, 835, 833, 831, 828, 826, 824, 821, 819, 816, 81
    4, 811, 809, 806, 804, 801, 799, 796, 794, 791, 789, 786, 784, 781, 778, 776, 773, 7
    71, 768, 765, 763, 760, 757, 755, 752, 749, 746, 744, 741, 738, 735, 733, 730, 727,
    724, 721, 719, 716, 713, 710, 707, 704, 701, 699, 696, 693, 690, 687, 684, 681, 678,
     675, 672, 669, 666, 663, 660, 657, 655, 652, 649, 646, 643, 639, 636, 633, 630, 627
    , 624, 621, 618, 615, 612, 609, 606, 603, 600, 597, 594, 591, 588, 584, 581, 578, 57
    5, 572, 569, 566, 563, 560, 556, 553, 550, 547, 544, 541, 538, 535, 531, 528, 525, 5
    22, 519, 516, 513, 509, 506, 503, 500, 497, 494, 491, 487, 484, 481, 478, 475, 472,
    469, 465, 462, 459, 456, 453, 450, 447, 444, 440, 437, 434, 431, 428, 425, 422, 419,
     416, 412, 409, 406, 403, 400, 397, 394, 391, 388, 385, 382, 379, 376, 373, 370, 367
    , 364, 361, 357, 354, 351, 348, 345, 343, 340, 337, 334, 331, 328, 325, 322, 319, 31
    6, 313, 310, 307, 304, 301, 299, 296, 293, 290, 287, 284, 281, 279, 276, 273, 270, 2
    67, 265, 262, 259, 256, 254, 251, 248, 245, 243, 240, 237, 235, 232, 229, 227, 224,
    222, 219, 216, 214, 211, 209, 206, 204, 201, 199, 196, 194, 191, 189, 186, 184, 181,
     179, 176, 174, 172, 169, 167, 165, 162, 160, 158, 155, 153, 151, 149, 146, 144, 142
    , 140, 138, 136, 133, 131, 129, 127, 125, 123, 121, 119, 117, 115, 113, 111, 109, 10
    7, 105, 103, 101, 99, 97, 95, 94, 92, 90, 88, 86, 85, 83, 81, 80, 78, 76, 75, 73, 71
    , 70, 68, 66, 65, 63, 62, 60, 59, 57, 56, 54, 53, 52, 50, 49, 48, 46, 45, 44, 42, 41
    , 40, 39, 37, 36, 35, 34, 33, 32, 31, 30, 29, 27, 26, 25, 24, 24, 23, 22, 21, 20, 19
    , 18, 17, 16, 16, 15, 14, 13, 13, 12, 11, 11, 10, 9, 9, 8, 8, 7, 7, 6, 6, 5, 5, 4, 4
    , 4, 3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1
    , 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 11, 11, 1
    2, 13, 13, 14, 15, 16, 16, 17, 18, 19, 20, 21, 22, 23, 24, 24, 25, 26, 27, 29, 30, 3
    1, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 44, 45, 46, 48, 49, 50, 52, 53, 54, 56, 5
    7, 59, 60, 62, 63, 65, 66, 68, 70, 71, 73, 75, 76, 78, 80, 81, 83, 85, 86, 88, 90, 9
    2, 94, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125,
    127, 129, 131, 133, 136, 138, 140, 142, 144, 146, 149, 151, 153, 155, 158, 160, 162,
     165, 167, 169, 172, 174, 176, 179, 181, 184, 186, 189, 191, 194, 196, 199, 201, 204
    , 206, 209, 211, 214, 216, 219, 222, 224, 227, 229, 232, 235, 237, 240, 243, 245, 24
    8, 251, 254, 256, 259, 262, 265, 267, 270, 273, 276, 279, 281, 284, 287, 290, 293, 2
    96, 299, 301, 304, 307, 310, 313, 316, 319, 322, 325, 328, 331, 334, 337, 340, 343,
    345, 348, 351, 354, 357, 361, 364, 367, 370, 373, 376, 379, 382, 385, 388, 391, 394,
     397, 400, 403, 406, 409, 412, 416, 419, 422, 425, 428, 431, 434, 437, 440, 444, 447
    , 450, 453, 456, 459, 462, 465, 469, 472, 475, 478, 481, 484, 487, 491, 494, 497};
44. const PROGMEM uint16_t TRI_TABLE[] = {500, 502, 504, 506, 508, 510, 512, 514, 516, 5
    18, 520, 522, 524, 526, 528, 530, 532, 534, 536, 538, 540, 542, 544, 546, 548, 550,
    552, 554, 556, 558, 560, 562, 564, 566, 568, 570, 572, 574, 576, 578, 580, 582, 584,
     586, 588, 590, 592, 594, 596, 598, 600, 602, 604, 606, 608, 610, 612, 614, 616, 618
    , 620, 622, 624, 626, 628, 630, 632, 634, 636, 638, 640, 642, 644, 646, 648, 650, 65
    2, 654, 656, 658, 660, 662, 664, 666, 668, 670, 672, 674, 676, 678, 680, 682, 684, 6
    86, 688, 690, 692, 694, 696, 698, 700, 702, 704, 706, 708, 710, 712, 714, 716, 718,
    720, 722, 724, 726, 728, 730, 732, 734, 736, 738, 740, 742, 744, 746, 748, 750, 752,
     754, 756, 758, 760, 762, 764, 766, 768, 770, 772, 774, 776, 778, 780, 782, 784, 786
    , 788, 790, 792, 794, 796, 798, 800, 802, 804, 806, 808, 810, 812, 814, 816, 818, 82
    0, 822, 824, 826, 828, 830, 832, 834, 836, 838, 840, 842, 844, 846, 848, 850, 852, 8
    54, 856, 858, 860, 862, 864, 866, 868, 870, 872, 874, 876, 878, 880, 882, 884, 886,
    888, 890, 892, 894, 896, 898, 900, 902, 904, 906, 908, 910, 912, 914, 916, 918, 920,
     922, 924, 926, 928, 930, 932, 934, 936, 938, 940, 942, 944, 946, 948, 950, 952, 954
    , 956, 958, 960, 962, 964, 966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986, 98
    8, 990, 992, 994, 996, 998, 1000, 998, 996, 994, 992, 990, 988, 986, 984, 982, 980,
    978, 976, 974, 972, 970, 968, 966, 964, 962, 960, 958, 956, 954, 952, 950, 948, 946,
     944, 942, 940, 938, 936, 934, 932, 930, 928, 926, 924, 922, 920, 918, 916, 914, 912
    , 910, 908, 906, 904, 902, 900, 898, 896, 894, 892, 890, 888, 886, 884, 882, 880, 87
    8, 876, 874, 872, 870, 868, 866, 864, 862, 860, 858, 856, 854, 852, 850, 848, 846, 8
    44, 842, 840, 838, 836, 834, 832, 830, 828, 826, 824, 822, 820, 818, 816, 814, 812,
    810, 808, 806, 804, 802, 800, 798, 796, 794, 792, 790, 788, 786, 784, 782, 780, 778,
```

```
 776, 774, 772, 770, 768, 766, 764, 762, 760, 758, 756, 754, 752, 750, 748, 746, 744
, 742, 740, 738, 736, 734, 732, 730, 728, 726, 724, 722, 720, 718, 716, 714, 712, 71
0, 708, 706, 704, 702, 700, 698, 696, 694, 692, 690, 688, 686, 684, 682, 680, 678, 6
76, 674, 672, 670, 668, 666, 664, 662, 660, 658, 656, 654, 652, 650, 648, 646, 644,
642, 640, 638, 636, 634, 632, 630, 628, 626, 624, 622, 620, 618, 616, 614, 612, 610,
 608, 606, 604, 602, 600, 598, 596, 594, 592, 590, 588, 586, 584, 582, 580, 578, 576
, 574, 572, 570, 568, 566, 564, 562, 560, 558, 556, 554, 552, 550, 548, 546, 544, 54
2, 540, 538, 536, 534, 532, 530, 528, 526, 524, 522, 520, 518, 516, 514, 512, 510, 5
08, 506, 504, 502, 500, 498, 496, 494, 492, 490, 488, 486, 484, 482, 480, 478, 476,
474, 472, 470, 468, 466, 464, 462, 460, 458, 456, 454, 452, 450, 448, 446, 444, 442,
 440, 438, 436, 434, 432, 430, 428, 426, 424, 422, 420, 418, 416, 414, 412, 410, 408
, 406, 404, 402, 400, 398, 396, 394, 392, 390, 388, 386, 384, 382, 380, 378, 376, 37
4, 372, 370, 368, 366, 364, 362, 360, 358, 356, 354, 352, 350, 348, 346, 344, 342, 3
40, 338, 336, 334, 332, 330, 328, 326, 324, 322, 320, 318, 316, 314, 312, 310, 308,
306, 304, 302, 300, 298, 296, 294, 292, 290, 288, 286, 284, 282, 280, 278, 276, 274,
 272, 270, 268, 266, 264, 262, 260, 258, 256, 254, 252, 250, 248, 246, 244, 242, 240
, 238, 236, 234, 232, 230, 228, 226, 224, 222, 220, 218, 216, 214, 212, 210, 208, 20
6, 204, 202, 200, 198, 196, 194, 192, 190, 188, 186, 184, 182, 180, 178, 176, 174, 1
72, 170, 168, 166, 164, 162, 160, 158, 156, 154, 152, 150, 148, 146, 144, 142, 140,
138, 136, 134, 132, 130, 128, 126, 124, 122, 120, 118, 116, 114, 112, 110, 108, 106,
 104, 102, 100, 98, 96, 94, 92, 90, 88, 86, 84, 82, 80, 78, 76, 74, 72, 70, 68, 66,
64, 62, 60, 58, 56, 54, 52, 50, 48, 46, 44, 42, 40, 38, 36, 34, 32, 30, 28, 26, 24,
22, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 2
4, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 6
6, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 10
6, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 1
40, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172,
174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206,
 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240
, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 27
4, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 3
08, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340,
342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374,
 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408
, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 44
2, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 4
76, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498};

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB
  }
  delay(100);
  u8g2.begin();
  Serial.println("Pocetak");

  //postavljanje pinova za output i stavljanje u početno stanje 0
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  PORTF = 0;

  //izračun parametara
  unosParametara();
  izracunVrijednosti(ma, mf);
}

void loop() {

  //   put your main code here, to run repeatedly:

  //prolaz kroz tablicu rezultata, ispis na pinove, provjera jesu li pritisnuti A i
B na tipkovnici
```

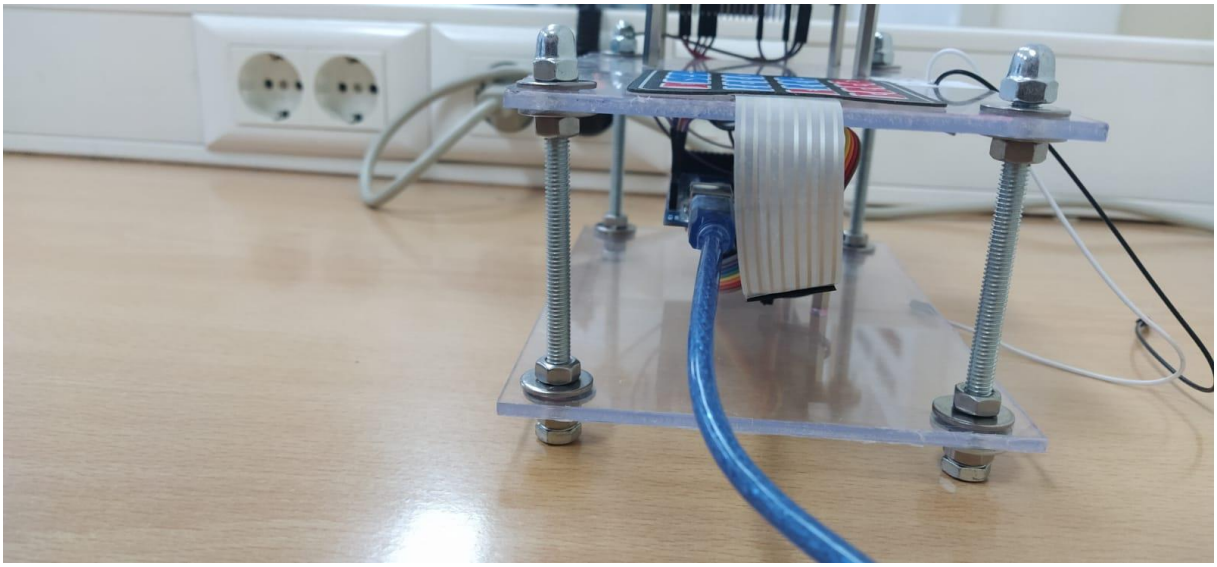
```
71.   for (uint16_t i = 0; i < 999; i++) {
72.     if (rezultat[i] == 1) {
73.       PORTF = 0b0000001; //ispis na pinovima
74.       dkey = keypad.getKey();
75.       if (dkey != NO_KEY) {
76.         gkey = dkey;
77.       }
78.       if (gkey == 'A') {
79.         pauza();
80.       }
81.       else if (gkey == 'B') {
82.         gkey = NO_KEY; //ciscenje buffera
83.         PORTF = 0;
84.         unosParametara();
85.         izracunVrijednosti(ma, mf);
86.       }
87.       delayMicroseconds(12); //delay potreban za zeljenu frekvenciju
88.
89.     } else {
90.       PORTF = 0b00000010; //ispis na pinovima
91.       dkey = keypad.getKey();
92.       if (dkey != NO_KEY) {
93.         gkey = dkey;
94.       }
95.       if (gkey == 'A') {
96.         pauza();
97.       }
98.       else if (gkey == 'B') {
99.         gkey = NO_KEY; //ciscenje buffera
100.              PORTF = 0;
101.              unosParametara();
102.              izracunVrijednosti(ma, mf);
103.            }
104.           delayMicroseconds(12); //delay potreban za zeljenu frekvenciju
105.         }
106.       }
107.     }
108.
109.      void unosParametara()
110.      {
111.        //pocetni ispis na ekranu
112.        u8g2.clearBuffer();
113.        u8g2.setFont(u8g2_font_chikita_tf);
114.        u8g2.drawStr( 15, 10, "UPRAVLJANJE JEDNOM");
115.        u8g2.drawStr( 20, 20, "GRANOM TROFAZNOG");
116.        u8g2.drawStr( 7, 30, "IZMJENJVACA KORISTECI");
117.        u8g2.drawStr( 2, 40, "ARDUINO MIKROKONTROLER");
118.        u8g2.drawStr( 35, 50, "FERIT 2020");
119.        u8g2.drawStr( 15, 60, "galic.petar96@gmail.com");
120.        u8g2.sendBuffer();
121.        delay(5000);
122.
123.            //odabir amplitudnog indexa ma
124.
125.        //ispis na ekranu
126.        u8g2.clearBuffer();
127.        u8g2.drawStr( 18, 10, "POMOCU TIPKOVNICE");
128.        u8g2.drawStr( 11, 20, "ODABERITE AMPLITUDNI");
129.        u8g2.drawStr( 14, 30, "INDEX MODULACIJE ma");
130.        u8g2.drawStr( 5, 40, "->|_1_|_2_|_3_|_4_|_5_|");
131.        u8g2.drawStr( 5, 50, "ma| 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |");
132.        u8g2.sendBuffer();
133.
134.        //ispis na serial monitoru
```

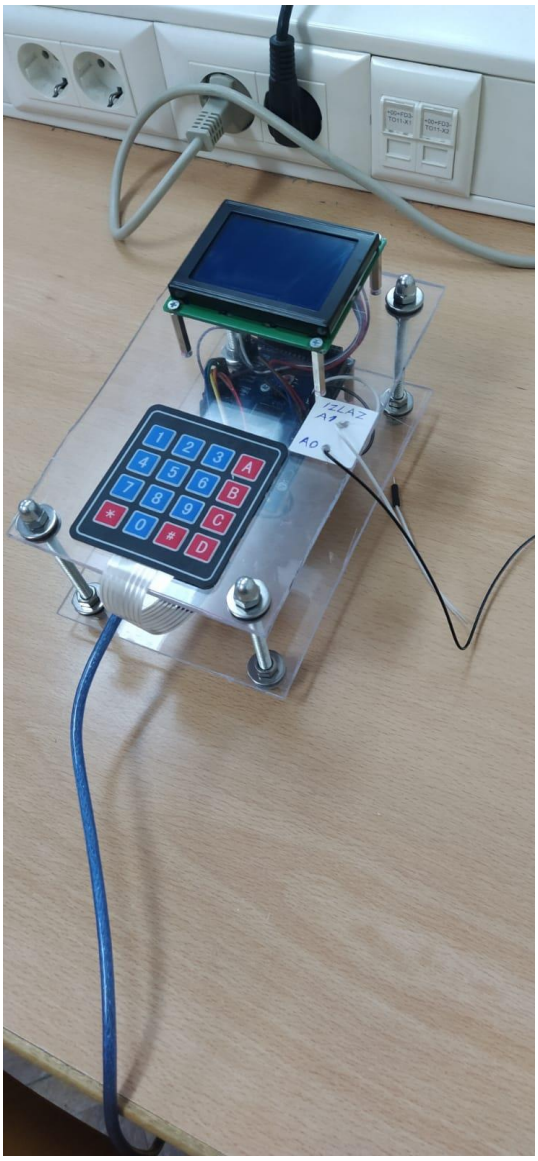
```
135.          Serial.println("ma|[0.2]|[0.4]|[0.6]|[0.8]|[1.0]|");
136.          Serial.println("  | [1] | [2] | [3] | [4] | [5] |");
137.          Serial.println("Odaberite amplitudni indeks modulacije ma");
138.          delay(100);
139.
140.          //petlja za odabir ma
141.          while (1) {
142.            key = keypad.getKey();
143.            if (key) {
144.              Serial.println(key);
145.            }
146.            if (key == '1') {
147.              ma = 0.2;
148.              Serial.println(ma);
149.              break;
150.            }
151.            if (key == '2') {
152.              ma = 0.4;
153.              Serial.println(ma);
154.              break;
155.            }
156.            if (key == '3') {
157.              ma = 0.6;
158.              Serial.println(ma);
159.              break;
160.            }
161.            if (key == '4') {
162.              ma = 0.8;
163.              Serial.println(ma);
164.              break;
165.            }
166.            if (key == '5') {
167.              ma = 1;
168.              Serial.println(ma);
169.              break;
170.            }
171.          }
172.          delay(100);
173.
174.               //odabir  frekvencijskog indexa mf
175.
176.          //ispis na ekranu
177.          u8g2.clearBuffer();
178.          u8g2.setFont(u8g2_font_chikita_tf);
179.          u8g2.drawStr( 15, 10, "POMOCU TIPKOVNICE");
180.          u8g2.drawStr( 0, 20, "ODABERITE FREKVENCIJSKI");
181.          u8g2.drawStr( 12, 30, "INDEX MODULACIJE mf");
182.          u8g2.drawStr( 15, 40, "->|_1_|_2_|_3_|_4_|");
183.          u8g2.drawStr( 15, 50, "mf| 12 |  24  |  48  |  96  | ");
184.          u8g2.sendBuffer();
185.
186.          //ispis na serial monitoru
187.          Serial.println("mf| [12] | [24] | [48] | [96]|");
188.          Serial.println("  | [1]  | [2]  | [3]  | [4] |");
189.          Serial.println("Odaberite frekvencijski indeks modulacije mf");
190.
191.          //petlja za odabir mf
192.          while (1) {
193.            key = keypad.getKey();
194.            if (key) {
195.              Serial.println(key);
196.            }
197.            if (key == '1') {
198.              mf = 12;
```

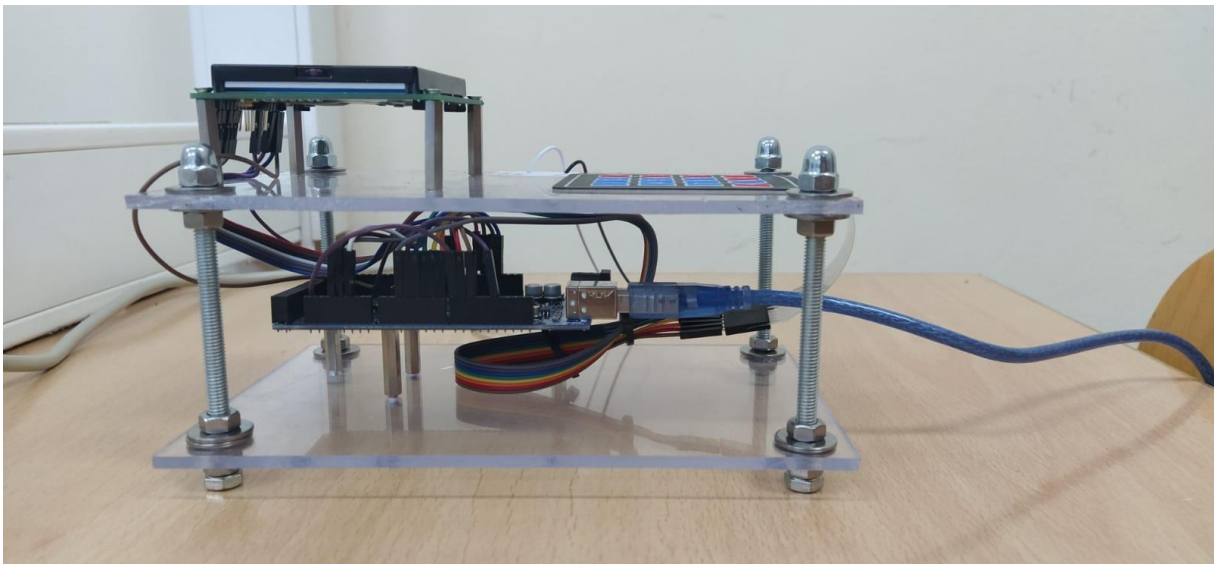
```
199.              Serial.println(mf);
200.              break;
201.            }
202.            if (key == '2') {
203.              mf = 24;
204.              Serial.println(mf);
205.              break;
206.            }
207.            if (key == '3') {
208.              mf = 48;
209.              Serial.println(mf);
210.              break;
211.            }
212.            if (key == '4') {
213.              mf = 96;
214.              Serial.println(mf);
215.              break;
216.            }
217.          }
218.          delay(100);
219.
220.          //ispis na ekranu da se program izvršava
221.          u8g2.clearBuffer();
222.          u8g2.setFont(u8g2_font_chikita_tf);
223.          u8g2.drawStr( 15, 10, "PROGRAM SE IZVRSAVA");
224.          u8g2.drawStr( 13, 20, "ZA PAUZU PRITISNITE A");
225.          u8g2.drawStr( 8, 30, "ZA PONOVNO POKRETANJE");
226.          u8g2.drawStr( 35, 40, "PRITISNITE B");
227.          u8g2.sendBuffer();
228.        }
229.
230.
231.        /*============================================================================
    ===*/
232.        //funkcija za izracun vrijednosti
233.        void izracunVrijednosti(float ma, uint8_t mf) {
234.          uint16_t br = 0;
235.          uint16_t sinus;
236.          float trokut;
237.          float masinus;
238.
239.          //izracun polja rezultata
240.          for (uint8_t j = 0; j < mf; j++) {
241.            for (uint16_t k = 0; k < 999; k++) {
242.              if (k % mf == 0) {
243.                sinus = pgm_read_word_near(SINE_TABLE + br);
244.                trokut = pgm_read_word_near(TRI_TABLE + k);
245.                masinus = sinus * ma;
246.                if (masinus >= trokut) {
247.                  rezultat[br] = 1;
248.                } else {
249.                  rezultat[br] = 0;
250.                }
251.                br++;
252.                if (br > 999) {
253.                  break;
254.                }
255.              }
256.            }
257.          }
258.        }
259.
260.        //funkcija za pauzu
261.        void pauza() {
```

```
262.         PORTF = 0; //postavljanje pinova na nulu
263.
264.         //ispis na ekranu
265.         u8g2.clearBuffer();
266.         u8g2.setFont(u8g2_font_chikita_tf );
267.         u8g2.drawStr( 50, 10, "PAUZA");
268.         u8g2.drawStr( 35, 20, "ZA NASTAVAK");
269.         u8g2.drawStr( 35, 30, "PRITISNITE A");
270.         u8g2.sendBuffer();
271.         Serial.println("Pauza");
272.
273.         //vracanje u rad programa
274.         while (1) {
275.           char key = keypad.getKey();
276.           if (key == 'A') {
277.             gkey = NO_KEY; //ciscenje buffera
278.             delay(100);
279.
280.             //ispis na ekranu
281.             u8g2.clearBuffer();
282.             u8g2.setFont(u8g2_font_chikita_tf);
283.             u8g2.drawStr( 15, 10, "PROGRAM SE IZVRSAVA");
284.             u8g2.drawStr( 10, 20, "ZA PAUZU PRITISNITE A");
285.             u8g2.drawStr( 8, 30, "ZA PONOVNO POKRETANJE");
286.             u8g2.drawStr( 35, 40, "PRITISNITE B");
287.             u8g2.sendBuffer();
288.             delay(100);
289.             break;
290.           }
291.         }
292.       }
```

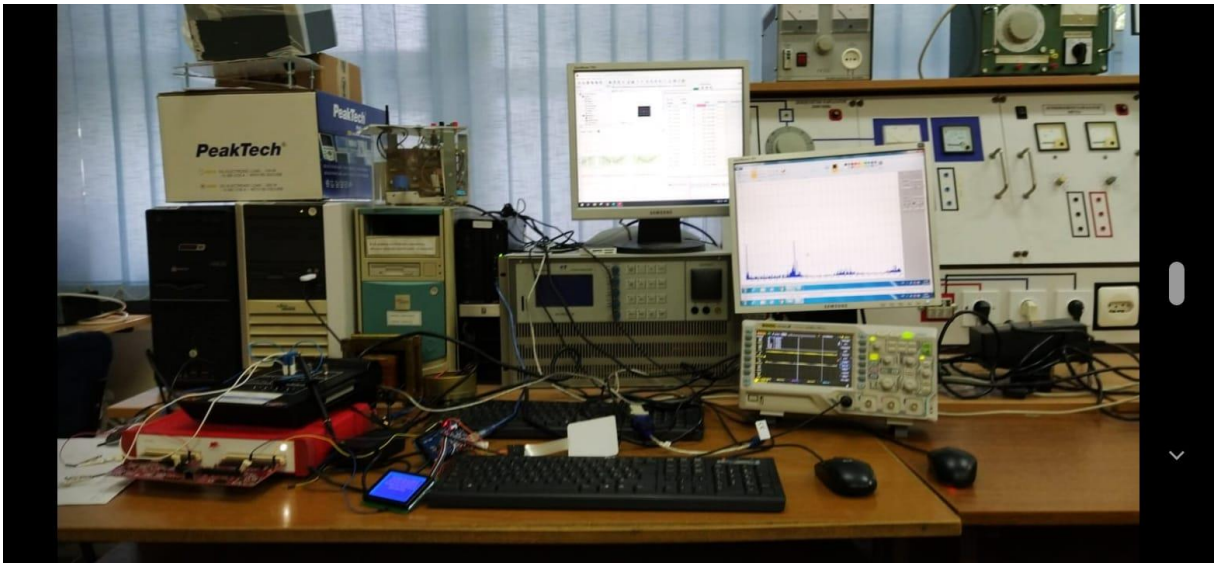
Prilog 2 : Slike makete i radnog okruženja




IZLAZ
A1
A0
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D



IZLAZ
A1
A0
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D


PeakTech