

Usmjerivač signala kroz gitarske efekte

Benja, Borna

Undergraduate thesis / Završni 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://um.nsk.hr/um:nbn:hr:200:073247>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-30**

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

USMJERIVAČ SIGNALA KROZ GITARSKE EFEKTE

Završni rad

Borna Benja

Osijek, 2020.

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

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.06.2020.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na
preddiplomskom sveučilišnom studiju**

Ime i prezime studenta:	Borna Benja
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4035, 18.09.2019.
OIB studenta:	15324730086
Mentor:	Izv. prof. dr. sc. Davor Vinko
Sumentor:	Luka Filipović
Sumentor iz tvrtke:	
Naslov završnog rada:	Usmjerivač signala kroz gitarske efekte
Znanstvena grana rada:	Elektronika (zn. polje elektrotehnika)
Predložena ocjena završnog 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: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	15.06.2020.
Datum potvrde ocjene Odbora:	15.07.2020.
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, 16.07.2020.

Ime i prezime studenta:

Borna Benja

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4035, 18.09.2019.

Turnitin podudaranje [%]:

6%

Ovom izjavom izjavljujem da je rad pod nazivom: **Usmjerivač signala kroz gitarske efekte**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Davor Vinko

i sumentora Luka Filipović

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:

Sadržaj

1. UVOD	1
1.1. Zadatak završnog rada	1
2. GITARSKÉ PEDALE I <i>PEDALBOARD</i>	2
3. UREĐAJI ZA USMJERAVANJE SIGNALA	3
3.1. Podjela uređaja za usmjeravanje signala	3
3.2. Trenutno dostupni uređaji za usmjeravanje signala	4
4. KORIŠTENE TEHNOLOGIJE I ALATI.....	5
4.1. EAGLE	5
4.2. Mikroupravljači	5
4.2.1. Arduino.....	5
4.3. Releji.....	6
4.4. Optoizolatori.....	7
4.5. Tranzistori.....	7
4.5.1. Bipolarni tranzistori.....	7
4.6. Naponski regulatori	8
4.7. Ostale komponente	8
5. IZRADA UREĐAJA ZA USMJERAVANJE SIGNALA	9
5.1. Mogućnosti uređaja	9
5.2. Izrada sheme i popis dijelova	10
5.3. Sastavljanje uređaja	14
5.3.1. Lemljenje komponenti na tiskanu pločicu	14
5.3.2. Kućište.....	17
5.4. Izrada i objašnjenje koda	18
5.5. Poboľšanja predviđena za sljedeću verziju uređaja	24
6. TESTIRANJE UREĐAJA	25
7. ZAKLJUČAK	27
LITERATURA.....	28
SAŽETAK.....	30
ABSTRACT	30
ŽIVOTOPIS	31
PRILOZI.....	32

1. UVOD

Glazbenici današnjice često koriste efekte za oblikovanje zvuka svojih instrumenata kako bi se bolje uklopili u zvuk cjelokupnog glazbenog sastava, kako bi novim zvukom pronašli inspiraciju ili jednostavno pretvorili zvuk svog instrumenta u nešto posve novo. U ovom radu fokusirat ćemo se na gitariste i gitarske efekte.

Zbog velike količine različitih gitarskih efekata, svaki gitarist najčešće ima nekoliko efekt pedala. Kombinacije tih efekata omogućavaju mu da zvuk svog instrumenta prilagodi potrebama glazbenog djela koje izvodi. Tako se ostvaruje velika fleksibilnost zvuka u studiju, prilikom snimanja djela, ali problem nastaje pri uključivanju kombinacija efekata u situaciji sviranja pred publikom (engl. *live*).

Za lakše kontroliranje više efekata u istom trenutku koriste se usmjerivači signala (engl. *signal router, loop switcher*). U ovom radu opisana je teorija i postupak izrade usmjerivača signala za 8 gitarskih efekata. Izrađeni uređaj imat će mogućnost pohrane kombinacija korištenih izlaza i jednostavno pozivanje istih. Za usmjeravanje signala koristit će se releji kojima će upravljati mikroupravljač.

1.1. Zadatak završnog rada

Zadatak je završnog rada razviti i izraditi usmjerivač signala kroz više gitarskih efekata. Cilj rada je uz minimalne smetnje u signalu pojednostaviti usmjeravanje signala kroz više gitarskih efekata. Usmjeravanje je potrebno ostvariti pomoću mikroupravljača i modula s relejima.

2. GITARSKE PEDALE I *PEDALBOARD*

Efekti u glazbi pojavili su se četrdesetih godina 20. stoljeća kada su glazbenici i producenti počeli eksperimentirati s pozicioniranjem mikrofona i pomicanjem magnetske vrpce za vrijeme snimanja. Tako su dobivali efekte jeke ili treperenja zvuka. Prve popularne samostalne jedinice namijenjene mijenjaju zvuka gitare nastale su šezdesetih godina, npr. *Maestro Fuzz* pedala koju je proslavio sastav *Rolling Stones* i *Uni-Vibe* pedala koju je koristio Jimi Hendrix. [1] Tada je počela komercijalizacija efekata u formatu pedala (engl. *stompbox*), malih metalnih ili plastičnih kutija u kojima se nalaze elektronički sklopovi za promjenu zvuka instrumenta.

Danas postoje tisuće različitih gitarskih efekt pedala, ali većina ih se može svrstati u neku od osnovnih kategorija: efekti pojačanja signala, efekti modulacije signala, efekti jeke, efekti za promjenu visine tona, filteri te ostali efekti kao sintesajzeri (engl. *synthesizer*), djelitelji (engl. *bitcrush*) itd. [2]

Format pedala opstao je jer omogućuje glazbeniku da nastavi svirati dok stopalom može uključiti ili isključiti željeni efekt. Uz to, pedale su jednostavne za prijenos i upotrebu jer su potrebni samo kabeli za ulaz i izlaz signala te kabel za napajanje. Kako bi zadovoljili potrebe za različitim zvukovima gitaristi skupljaju sve veći broj različitih pedala, a u tome im pomaže *pedalboard*.

Pedalboard je platforma najčešće napravljena od aluminijske namijenjena za prijenos više gitarskih efekata u formatu pedala. Naime, na *pedalboard* se poslože pedale, dovede im se odgovarajuće napajanje te ih se poveže serijski tako da se izlaz jedne pedale spoji u ulaz sljedeće. [3] Zbog takvog spajanja treba voditi brigu o redosljedu efekata kako bi se dobili željeni rezultati. Npr. stavljanje efekta jeke prije efekta distorzije značilo bi da će zvuk svakog odjeka biti deformiran dok bi spajanje obrnutim redosljedom značilo da će ulazni signal gitare biti deformiran te kao takav proći kroz efekt jeke.

Kada su svi efekti spojeni željenim redosljedom nameće se problem kako uključiti samo određene efekte u istom trenutku.

3. UREĐAJI ZA USMJERAVANJE SIGNALA

3.1. Podjela uređaja za usmjeravanje signala

Uređaji za usmjeravanje signala (signal routers, loop switchers) koriste se kako bi se riješio problem uključivanja željenih kombinacija efekata. Takvi uređaji imaju parove ulaznih i izlaznih portova (*loops*) na koje se spajaju gitarske pedale, način informiranja korisnika koji *loop* je uključen i nožne prekidače kojima se signal usmjerava kroz određene grupe efekata. [4] Postoje dvije glavne vrste takvih usmjerivača:

- a) *True bypass* usmjerivači sastavljeni su samo od *3PDT* (engl. *triple-pole, double-throw*) prekidača i konektora za signal. Kao takvi imaju sposobnost usmjeravanja signala samo kroz susjedne efekte (u istom *loop-u*). Npr. ne može se istovremeno koristiti prva i peta pedala u nizu bez da se s njima ne koriste i druga, treća i četvrta. Zaključujemo da *true bypass* usmjerivačima nije moguće imati jednu pedalu u više različitih kombinacija.
- b) Usmjerivači s relejima obično koriste mikroupravljač za upravljanje relejima i usmjeravanje signala. Upotrebom mikroupravljača otvara se mogućnost korištenja programabilnih memorija za čuvanje spremljenih kombinacija i korištenje sedam segmentnih pokaznika ili *LCD display-a* za informiranje korisnika o ulazima i izlazima koji se trenutno koriste. Tako se ostvaruje jednostavnije i preciznije usmjeravanje signala tj. moguće je napraviti različite kombinacije nesusjednih efekata, pohraniti ih u memoriju i kasnije pozvati. Uz to, releji su dugotrajniji jer mogu napraviti preko 100 000 preklapanja, dok mehanički prekidači mogu oko 30 000. [5, 6]

U ovom radu opisan je razvoj i izrada usmjerivača s relejima koji omogućava pohranu kombinacija izlaza u memoriju. Tako dobivamo dugotrajnost i veću fleksibilnost uz jednostavnost korištenja uređaja.

3.2. Trenutno dostupni uređaji za usmjeravanje signala

Većina trenutno dostupnih uređaja za usmjeravanje signala koristi mikroupravljače za pohranu kombinacija i upravljanje funkcijama uređaja te releje za usmjeravanje signala. Neki od uređaja koji imaju karakteristike slične karakteristikama uređaja koji je opisan u ovome radu prikazani su u tablici 3.1. Mogućnosti takvih uređaja razlikuju se u posjedovanju ulaza za paralelno spajanje efekata, posjedovanju ulaza za mijenjanje kanala gitarskog pojačala, ali i broju petlji za gitarske efekte i broju memorijskih lokacija za pohranu kombinacija. [7, 8, 9]

Tablica 3.1. Usporedba uređaja za usmjeravanje gitarskog signala

Naziv uređaja	Harley Benton StompControl-4 ISO	Moen GEC8 JR	ElectroHarmonix Super Switcher	Uređaj predstavljen u ovom radu
Broj petlji za efekte	4	8	8	8
Broj memorijskih lokacija	5 banki s 4 memorijske lokacije	6 banki s 4 memorijske lokacije	16 banki s 8 memorijskih lokacija	2 banke s 8 memorijskih lokacija
Dimenzije	335·75·45 mm ³	480·60·52 mm ³	483·153·64 mm ³	253·53·52.2 mm ³
Ostalo	Ugrađeno napajanje za priključene efekte	Izlaz za mijenjanje kanala pojačala	<i>Mono</i> i <i>Stereo</i> izlazi, mogućnost određivanja tempa gitarskih efekata, slanje MIDI signala	Spajanje efekata između pretpojačala i pojačala
Cijena ¹	669 HRK	1222 HRK	2870 HRK	1000 HRK

S obzirom na predstavljene mogućnosti, cijenu i površinu koju uređaj zauzima, zaključujemo kako je uređaj predstavljen u ovome radu u prednosti nad trenutno dostupnim uređajima ako je potrebno usmjeravati signal kroz veći broj efekata na maloj površini *pedalboard*-a. Mogućnosti uređaja detaljnije su objašnjene u petom poglavlju.

¹ Cijene uređaja su okvirne i podložne su promjenama, predstavljaju vrijednosti u trenutku pisanja ovog rada

4. KORIŠTENE TEHNOLOGIJE I ALATI

U ovom poglavlju predstavljani su alati i teorija iza korištene tehnologije i elektroničkih komponenti potrebnih za realizaciju ovog rada.

4.1. EAGLE

EAGLE (engl. *Easily Applicable Graphical Layout Editor*) je aplikacija za crtanje elektroničkih shema i dizajn tiskanih pločica (engl. *Printed Circuit Board - PCB*). Dizajn u EAGLE-u temelji se na korištenju biblioteka (engl. *libraries*) u kojima su definirani elektronički simboli te veličine pojedinih pakiranja elektroničkih komponenti koje se koriste za dizajn tiskanih pločica i 3-D modeliranje. EAGLE omogućava jednostavno provjeravanje dosljednosti i podudaranja pločice i sheme (engl. *back-annotation, Electrical Rule Check – ERC*). Uz to, mogu se definirati i provjeriti pravila za dizajn tiskanih pločica (engl. *Design Rule Check – DRC*) kao što su debljina i udaljenost bakrenih vodova. Dizajn tiskane pločice može se spremirati u obliku Gerber datoteka koje se prosljeđuju tvornicama tiskanih pločica. [10, 11]

4.2. Mikroupravljači

Mikroupravljač je integrirani krug koji se ponaša poput računala. Mikroupravljač posjeduje procesorsku jedinicu, radnu memoriju, *ROM*, ulazno-izlazne *port-ove* i *flash* memoriju na koju se sprema programski kod. Dizajnirani su za nadziranje, mjerenje i upravljanje događajima. [12] Za to se koriste različiti ulazni uređaji kao prekidači, senzori, tipkala te izlazni uređaji kao motori, displeji i dr.

4.2.1. Arduino

Arduino je *open-source* platforma koja olakšava korištenje mikroupravljača tako što integrirani krug mikroupravljača okružuje svim potrebnim komponentama koje su potrebne za njegovu uporabu. Na Arduino tiskanoj pločici, osim mikroupravljača, nalaze se potrebni otpornici i kondenzatori, svjetleće (engl. *LE*) diode, tipka za *reset*, integrirani krug za serijsku komunikaciju, *USB port*, regulator napona i ulazno-izlazni *pin-ovi*. [13, 14]

Za ovaj rad korištena je platforma bazirana na Arduino Nano platformi zbog svog malog formata, ali velikog broja ulaza i izlaza. Bazirana je na mikroupravljaču ATmega328. Napon napajanja je od 7 V do 12 V koji se na samoj platformi regulira na 5 V. Kako bi se platforma zaštitila od zagrijavanja u ovom radu korišten je vanjski regulator te se zaobilazi integrirani regulator. Bitno je za naglasiti da je maksimalna vrijednost struje koju izlazi mikroupravljača mogu podnijeti 40 mA, a preporučeno je da vrijednost struje bude manja od 20mA. U ovom radu korištena je i integrirana programabilna memorija (*EEPROM*) veličine 1 KB za pohranu kombinacija uključenih petlji. [15, 16]

U Arduino ekosustav pripada i Arduino razvojno okruženje (engl. *Integrated Development Environment - IDE*). Razvojno okruženje sadrži uređivač teksta za pisanje programskog koda, omogućuje upravljanje bibliotekama i *bootloader* programima za svaku inačicu Arduino razvojne platforme. *Bootloader* programi uspostavljaju serijsku komunikaciju između razvojnog okruženja i razvojne platforme. Time je pojednostavljeno programiranje mikroupravljača jer ne zahtijeva dodatni *hardware* ili *software*. Programski kodovi napisani u Arduino *IDE* sadržavaju funkcije *setup()* koja se izvodi jednom pri pokretanju programa i *loop()* koja se ponavlja dok se programirani mikroupravljač ne isključi. [17]

4.3. Releji

Releji su strujno kontrolirani prekidači. Kada dovoljna struja prolazi zavojnicom releja tada se prekidački kontakti releja preklape. Releji omogućuju galvansku izolaciju upravljačkog signala od signala kojim želimo upravljati te imaju zanemarivo mali serijski otpor. [18, str. 59., 19, str. 397.] Kao takvi predstavljaju dobro rješenje za nesmetano preklapanje audio signala bez utjecaja na kvalitetu zvuka.

U ovom uređaju koriste se trenutni (engl. *momentary*) *DPDT* releji s jednom zavojnicom. *DPDT* (engl. *double pole double throw*) znači da relej ima dvije strane, a svaka od strana ima po dva kontakta za uklapanje (uobičajeno otvoreni i uobičajeno zatvoreni od engl. *normally open*, *normally closed*) i aktuator tj. zajednički kontakt (engl. *common*). Obje strane preklapaju se istovremeno pod utjecajem jedne zavojnice tako da se aktuator preklopi s uobičajeno zatvorenog kontakta na uobičajeno otvoreni i zatvori drugi strujni krug. *Trenutni* označava da je za održavanje uklopljenog stanja potrebna neprekidna struja.

4.4. Optoizolatori

Optoizolatori (optospojnici ili fotospojnici) kombiniraju *LE* diodu i fotodetektor kako bi se ostvario prijenos signala uz galvansku izolaciju dijelova uređaja. Fotodetektor može biti fotootpornik ili fototranzistor. [18, str. 843., 20, str. 331.] U ovom se slučaju optoizolatori koriste kako bismo dodatno odvojili digitalne upravljačke signale od analognih audio signala. Tako istovremeno možemo zaštititi mikroupravljač od potencijalnih kvarova na relejima i izolirati audio signal od šumova digitalnih dijelova uređaja.

U radu su korišteni optoizolatori koji implementiraju fototranzistor kao element osjetljiv na svjetlost. Korištenjem fototranzistora možemo dizajnirati strujni krug koji bi doveo struju na zavojnicu releja i ostvario uklapanje.

4.5. Tranzistori

Tranzistor je aktivna elektronička komponenta koja ima sposobnost pojačanja snage. Može se koristiti kao pojačalo ili kao prekidač. [18, str 71.]

4.5.1. Bipolarni tranzistori

Bipolarni tranzistori (engl. *bipolar junction transistor - BJT*) građeni su od 3 naizmjenično P-dopirana i N-dopirana sloja silicija ili germanija. Tako postoje PNP i NPN tranzistori čiji se izvodi nazivaju baza, kolektor i emiter. U radu su korišteni NPN tranzistori pa će se daljnja objašnjenja temeljiti na toj pretpostavci.

Bipolarni znači da obje vrste nositelja naboja (elektroni i šupljine) imaju ulogu u protoku struje. Kada se između baze i emitera pojavi napon U_{BE} veći od napona potrebnog za provođenje kroz jedan PN spoj (napon praga, oko 0.6 V) elektroni teku iz N-dopiranog emitera prema šupljinama P-dopirane baze i tako stvaraju struju baze I_B . U bazi se emitirani elektroni rekombiniraju sa šupljinama. Pošto je sloj baze tanji od emitera i kolektora, većina elektrona ne nađe slobodne šupljine u bazi i nastavi teći prema kolektoru. Elektroni koji teku prema kolektoru i šupljine koje teku iz kolektora prema bazi stvaraju struju kolektora I_C . U normalnom aktivnom području rada postoji omjer struje kolektora i struje baze što se naziva faktor strujnog pojačanja β , negdje naveden i kao h_{FE} . [18, str. 71., 21, str. 122.]

Uz normalno aktivno područje rada postoje još i zasićenje i zapiranje pomoću kojih se tranzistor može koristiti kao prekidač.

4.6. Naponski regulatori

Naponski regulatori su integrirani strujni krugovi napravljeni za održavanje stabilnog i preciznog izlaznog napona uz što manje šumove i što manju valovitost izlaznog napona. Regulatori imaju integrirane (engl. *on-chip*) komponente za postavljanje izlaznog napona i ograničavanje struje. Također, naponski regulatori imaju ugrađene mogućnosti isključivanja u slučaju prevelike struje ili pregrijavanja (engl. *overcurrent protection*). Naponski regulatori mogu biti fiksni (izlazni napon zadan im je prilikom izrade integriranog kruga) ili prilagodljivi čiji se izlazni napon može jednostavno definirati naponskim djelilom pomoću dva otpornika. [18, str. 600., 19, str. 527., 20, str. 324., 21, str. 189.] U ovom radu korišteni su fiksni naponski regulatori 7805 kojima je izlazni regulirani napon 5 V što je dovoljno za napajanje korištenog mikroupravljača i releja.

4.7. Ostale komponente

Ostale komponente korištene za izradu ovog uređaja su otpornici, kondenzatori, *LE* diode, posmačni registri, 6.35 mm konektori za audio signal i trenutni (engl. *momentary*) prekidači. Popis svih korištenih komponenti nalazi se u prilogu 5.2.

5. IZRADA UREĐAJA ZA USMJERAVANJE SIGNALA

U ovom poglavlju opisane su mogućnosti uređaja i teorija te proračuni koji objašnjavaju izbor komponenti i osobine dizajna sheme i pločice. Na kraju poglavlja opisan je programski kod koji je prenesen na mikroupravljač.

5.1. Mogućnosti uređaja

Završeni uređaj može usmjeravati signal kroz osam odvojenih petlji (engl. *loop*) u kojima se nalaze gitarski efekti u formatu pedala. Svaka pedala spojena je na zasebni ulaz i izlaz pomoću 6.35 mm konektora (engl. *jack*). Nakon prvih pet petlji korištena je tehnika prekidanja (engl. *breaking*) koja se sastoji od prekidanja vodiča, dovođenja signala na izlazni konektor i omogućavanja povratka signala u uređaj na susjedni ulazni konektor (engl. *send and return*) kako bi se mogao usmjeravati kroz preostale petlje. (Prilog 5.1) Korištenjem tehnike prekidanja moguće je koristiti prvih pet petlji prije gitarskog pretpojačala, a posljednje tri petlje nakon gitarskog pretpojačala tzv. *four cable method*, ali ostavlja se i mogućnost korištenja svih osam petlji ispred pretpojačala ako se *send* i *return* konektori povežu prikladnim kabelom.

Uređaj ima dva načina rada. U prvom načinu rada korisnik može uključivati svaku petlju pojedinačno i spremati željene kombinacije uključenih petlji u memoriju mikroupravljača. Pojedinačne petlje uključuju se pritiskom prekidača, a kombinacije se spremaju držanjem prekidača. U drugom načinu rada korisnik može pozvati spremljene kombinacije uključenih petlji pritiskom na prekidače. Može se spremati ukupno 16 kombinacija raspoređenih u dvije memorijske lokacije tzv. banke (engl. *bank*).

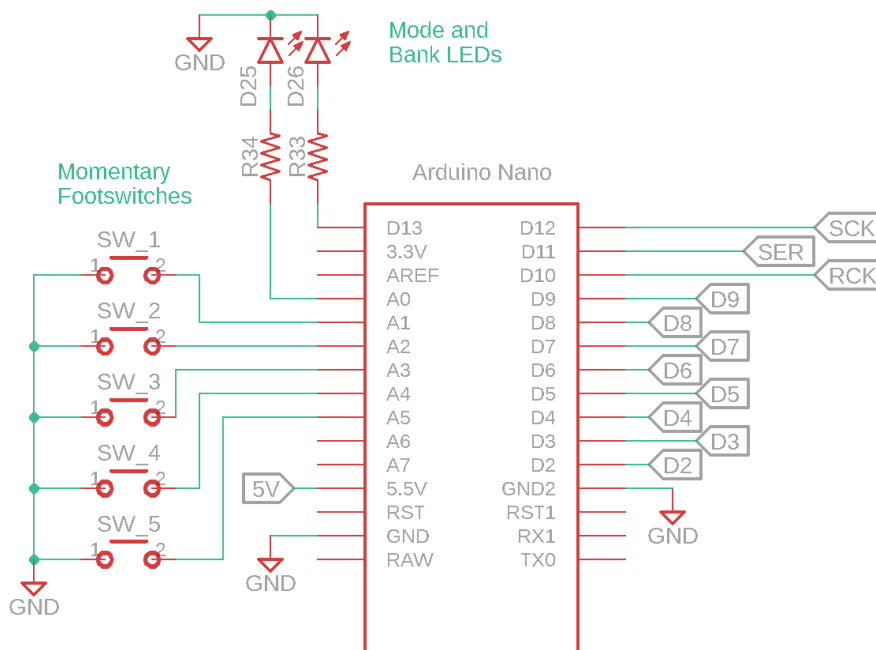
Kako bi se smanjila površina koju uređaj zauzima (engl. *footprint*) korišteno je samo pet prekidača za upravljanje svim mogućnostima uređaja. Zbog toga uključivanje nekih petlji zahtijeva pritisak dva prekidača istovremeno. Prekidači se nalaze na dovoljno maloj udaljenosti kako bi se mogla pritisnuti dva prekidača odjednom, ali su dovoljno udaljeni kako bi se mogao pritisnuti i samo jedan.

Uređaj se može napajati pomoću vanjskog istosmjernog pretvarača s izlaznim naponom između 7 V i 12 V kako bi se izbjeglo pregrijavanje vanjskih 7805 regulatora.

5.2. Izrada sheme i popis dijelova

Za izradu sheme i dizajn tiskane pločice korišten je program EAGLE. Shema će biti prikazana u obliku funkcionalnih cjelina zbog preglednosti rada.

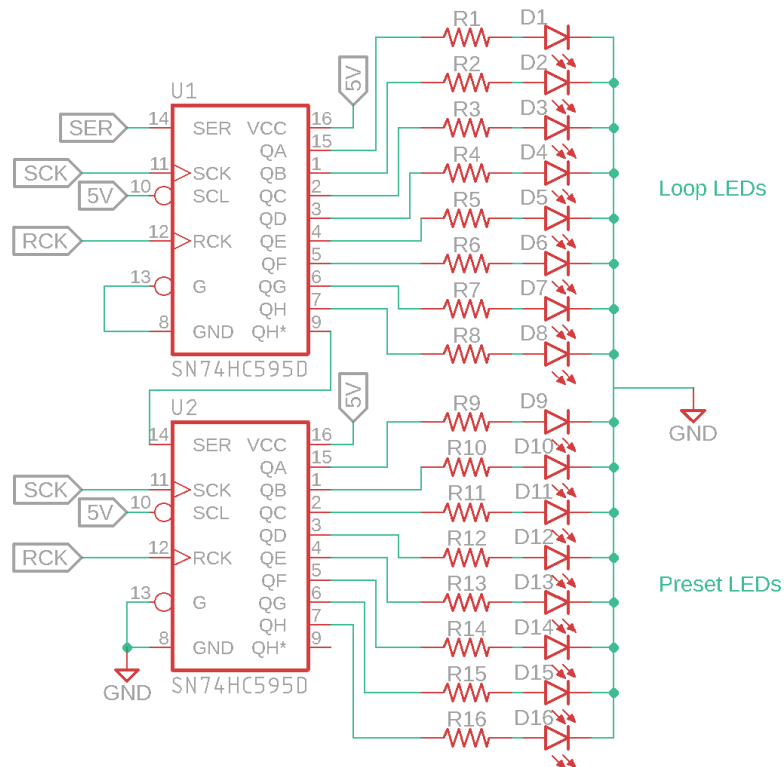
Prva cjelina je mikroupravljač i komponente povezane na mikroupravljač. Prema slici 5.1 na lijevoj strani sheme nalaze se prekidači označeni *SW_1* do *SW_5* koji služe za upravljanje uređajem te dvije *LE* diode s pripadajućim otpornicima za prikaz trenutnog načina rada i trenutne memorijske lokacije za pohranu kombinacija (engl. *bank*). Uz prekidače nisu stavljeni vanjski *pull-up* ili *pull-down* otpornici zato što su oni već integrirani u platformu mikroupravljača. S desne strane mikroupravljača nalaze se izvodi *D12*, *D11* i *D10* koji su spojeni na *shift* registre. Tako se na jednostavan način postiže upravljanje većim brojem *LE* dioda potrebnih za prikaz uključenih petlji. Izvodi *D2* do *D9* spojeni su na optoizolatore. Tako se postiže posredno upravljanje relejima koje štiti mikroupravljač od mogućeg kvara na relejima. (Slika 5.1.)



Slika 5.1. Shema spajanja mikroupravljača

Slika 5.2. prikazuje povezivanje posmačnih registara s mikroupravljačem i *LE* diodama. Posmačni registri označeni su kao *U1* i *U2*. Korišteni registri 74HC595 koriste se kao *serial-in*, *parallel-out* registri što znači da će serijskom komunikacijom od mikroupravljača primiti jedan

bajt bit po bit, a na izlaze prosljediti sve primljene bitove istovremeno. Ako registre spojimo serijski (Sl. 5.2.) i registrima prosljedimo dva bajta, možemo u istom trenutku upravljati sa 16 izlaza. Na izlaze posmačnih registara spojene su *LE* diode (*D1* do *D16*) s pripadajućim otpornicima (*R1* do *R16*). Svjetleće diode koriste se za prikaz trenutno uključenih petlji (oznaka *Loop LEDs*) i trenutno korištene kombinacije pohranjene u memoriji (oznaka *Preset LEDs*).



Slika 5.2. Shema spajanja posmačnih registara s *LE* diodama i mikroupravljačem

Struju koja teče kroz svjetleće diode na slikama 5.1. i 5.2. potrebno je ograničiti kako ne bi došlo do oštećenja dioda. Najjednostavniji način je spajanje otpornika u seriju s diodom. Najmanju potrebnu vrijednost otpora možemo izračunati ako znamo vrijednost struje koju dioda može podnijeti I_F (engl. *forward current*) i vrijednost napona potrebnog da dioda provede tj. zasvijetli U_F (engl. *forward voltage*). Te podatke možemo pronaći u tehničkim specifikacijama komponente (engl. *datasheet*). Prema [22] saznajemo da je vrijednost struje za korištene crvene diode 20 mA, a očekivana vrijednost napona 2 V. Za računanje otpora možemo koristiti sljedeću formulu izvedenu pomoću Kirchhoffovog i Ohmovog zakona:

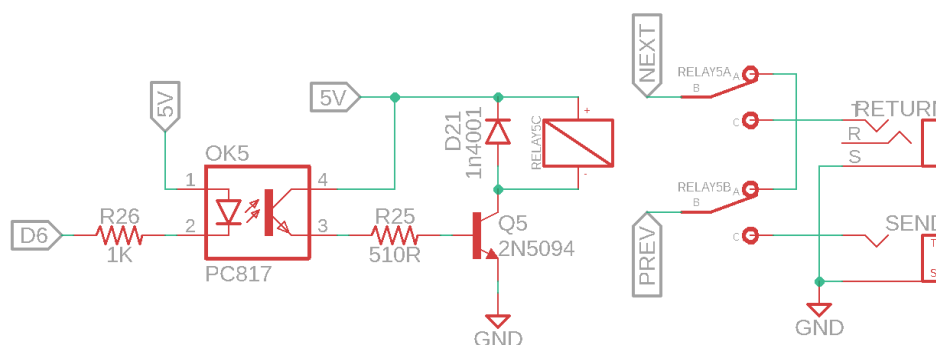
$$R = \frac{U_{CC} - U_F}{I_F} \quad (5-1)$$

gdje je U_{CC} napon napajanja tj. napon koji dolazi iz mikroupravljača za diode $D25$ i $D26$ ili napon koji dolazi iz posmačnog registra, U_F napon između izvoda diode, I_F najveća struja koju dioda može podnijeti, a R vrijednost otpora.

Kada u (5-1) uvrstimo napon veličine 5 V, što predstavlja logičku jedinicu, koji dolazi iz mikroupravljača ili registra, struju diode 20 mA i napon na diodi 2 V dobivamo minimalnu vrijednost otpora 150Ω ^[2]. U radu su korištene vrijednosti otpora između 10 K Ω i 47 K Ω jer moderne svjetleće diode zahtijevaju vrlo male struje kako bi svijetlile dovoljnim intenzitetom da se mogu koristiti kao indikatori. Uz to, ranije je napomenuto kako struje pojedinih izvoda mikroupravljača ne bi trebale biti veće od 20 mA kako bi se izbjeglo pregrijavanje i oštećenja.

Drugu funkcionalnu cjelinu uređaja predstavlja spoj optoizolatora i releja kojima se vrši usmjeravanje signala. Optoizolatori su spojeni na izvode mikroupravljača $D2$ do $D9$ pomoću otpornika koji služi za ograničavanje struje jer se u korištenim optoizolatorima nalazi svjetleća dioda. Pokazan je samo jedan od osam identičnih spojeva kako bi rad bio pregledan. (Slika 5.1. i Slika 5.3.) Na shemi sa slike 5.3. korištena je manja vrijednost otpora nego kod svjetlećih dioda koje koristimo za indikatore jer želimo zadržati veliku brzinu odziva optoizolatora koja ovisi o intenzitetu svjetla integrirane LE diode. Iz tehničkih specifikacija optoizolatora [23] saznajemo da struja integrirane diode ne smije prijeći 50 mA, a očekivani napon diode iznosi oko 1.2 V. Uz pretpostavku da upravljački signal označen $D6$ može biti 0 V ili 5 V (logička nula ili logička jedinica), možemo izračunati najmanju vrijednost potrebnog serijskog otpora koristeći (5-1). Tako dobivamo vrijednost 76Ω . Korišteni će otpornik od 1 K Ω ograničiti vrijednost struje na 3.8 mA što je dovoljno da se integrirana dioda ponaša očekivano, a ne zahtijeva prolazak struja većih od 20 mA kroz izvode mikroupravljača. Dioda $D21$ služi kao *flyback* dioda tj. pomaže pri pražnjenju zavojnice releja kada se relej ugasi.

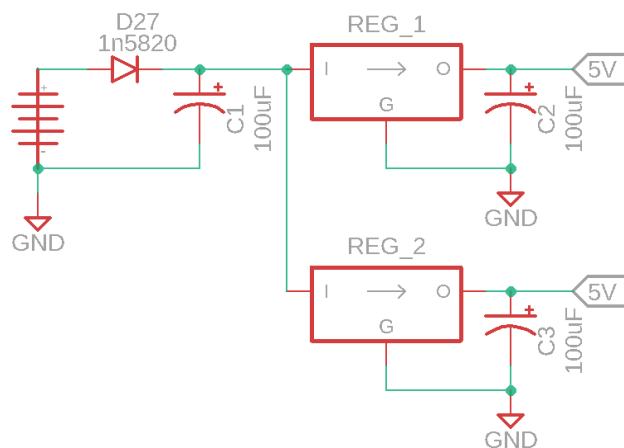
² Ohm (Ω) – jedinica otpora izvedena kao omjer napona i struje.



Slika 5.3. Shema spajanja releja i optoizolatora

Iz [23] saznajemo da je moguća vrijednost struje kolektora integriranog fototranzistora I_c između 2.5 mA i 30 mA pri naponu napajanja 5 V i struji diode oko 5 mA. Ako primijenimo Kirchhoffov zakon i od napona napajanja oduzmemo napon zasićenja između kolektora i emitera fototranzistora $U_{CE(sat)}$ (oko 0.2 V) [23] i napon između baze i emitera bipolarnog tranzistora označenog $Q5$ U_{BE} (oko 0.8 V) [24] znamo da će napon na otporniku $R25$ biti oko 4 V. Primjenom Ohmovog zakona dobivamo vrijednost struje od 7.8 mA što je unutar predviđenih granica optoizolatora, a dovoljno da bi tranzistor $Q5$ provodio. Kada tranzistor $Q5$ provede, napon između kolektora i emitera $U_{CE(sat)}$ je oko 0.2 V [24]. Ako to oduzmemo od napona napajanja (Kirchhoffov zakon) znamo da će napon na zavojnici releja iznositi 4.8 V što je dovoljno za preklapanje kontakata releja. [5]

Posljednju funkcionalnu cjelinu čini regulacija ulaznog napona. U radu su korišteni regulatori 7805 koji ulazni napon između 7 V i 35 V reguliraju na 5 V. (Slika 5.4) Razlog za korištenje dva regulatora je raspodjela struje. Time je smanjeno zagrijavanje regulatora. Kondenzatori označeni $C1$, $C2$ i $C3$ služe za stabilizaciju ulaznog i izlaznog napona tj. smanjuju valovitost (engl. *ripple*). Dioda $D27$ služi kao zaštita od ulaznog napona suprotnog polariteta. Bolje metode zaštite uređaja i reguliranja napona objašnjene su u poglavlju 5.5.



Slika 5.4. Shema reguliranja ulaznog napona

Prilog P.5.2. sadrži popis komponenti (engl. *bill of material* – *BOM*) korištenih za realizaciju predstavljenog uređaja.

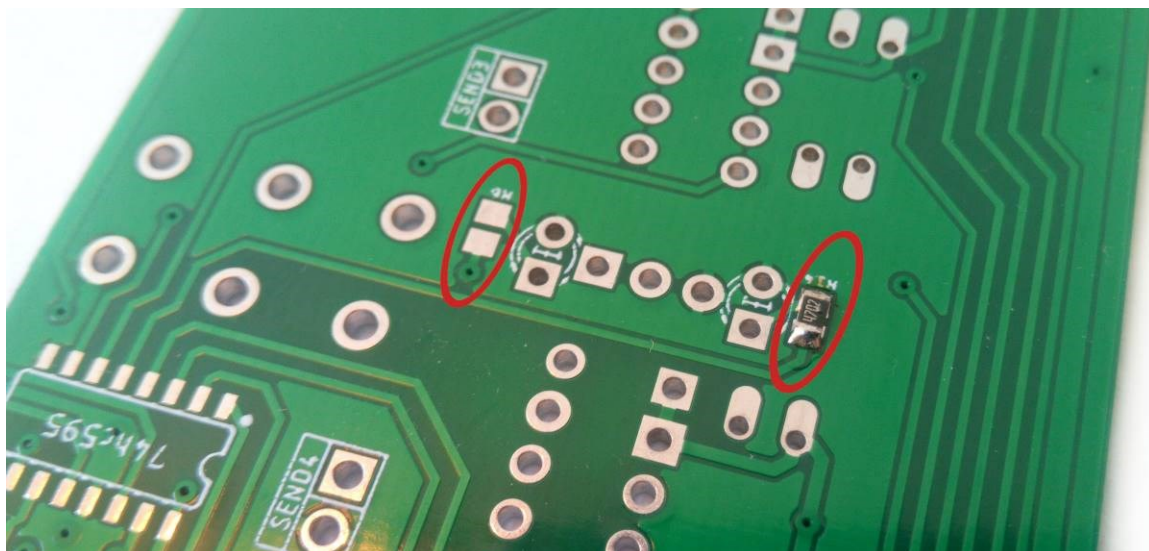
5.3. Sastavljanje uređaja

5.3.1. Lemljenje komponenti na tiskanu pločicu

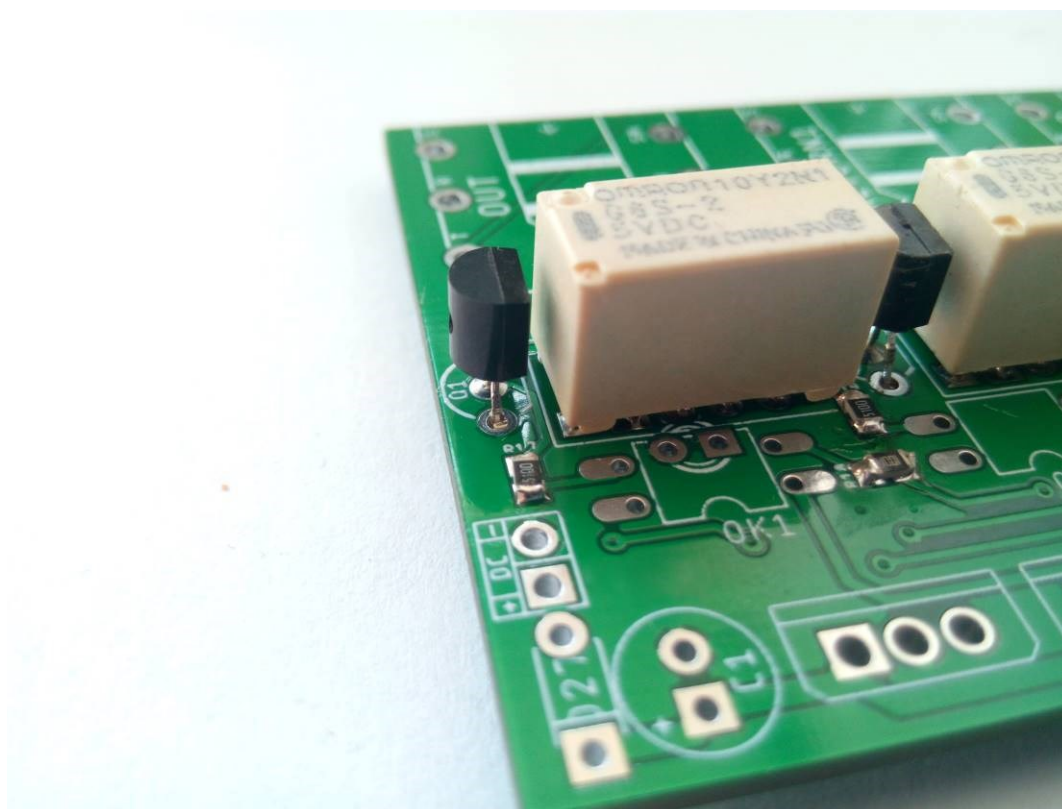
Tiskana pločica dizajnirana je u već spomenutoj aplikaciji EAGLE i poslana na izradu prema nacrtu prikazanom u prilogu 5.3. Vodovi na gornjoj strani pločice označeni su crvenom bojom, a na donjoj plavom. Za napajanje je korištena debljina vodova 40 mil ^[3], a za vodove kojima prolaze digitalni ili audio signali debljina je postavljena između 16 mil i 24 mil.

Sastavljanje uređaja započinje lemljenjem elektroničkih komponenti na tiskanu pločicu. *SMD* (engl. *surface mount device*) komponente kao što su otpornici 0805 pakiranja zaleme se na odgovarajuća polja (engl. *pad*) dok se izvodi *THT* (engl. *through-hole technology*) komponenti (npr. tranzistori u TO-92 pakiranju) umetnu u odgovarajuće utore i zaleme, a ostatak izvoda se odsiječe. (Slika 5.5. i Slika 5.6.) Na pločicu je poželjno prvo zalemiti fizički manje komponente kako bi se očuvao neometan pristup poljima i izvodima komponenti.

³ Mil – izvedena jedinica uobičajeno korištena za debljine vodova, jedna tisućina inča.

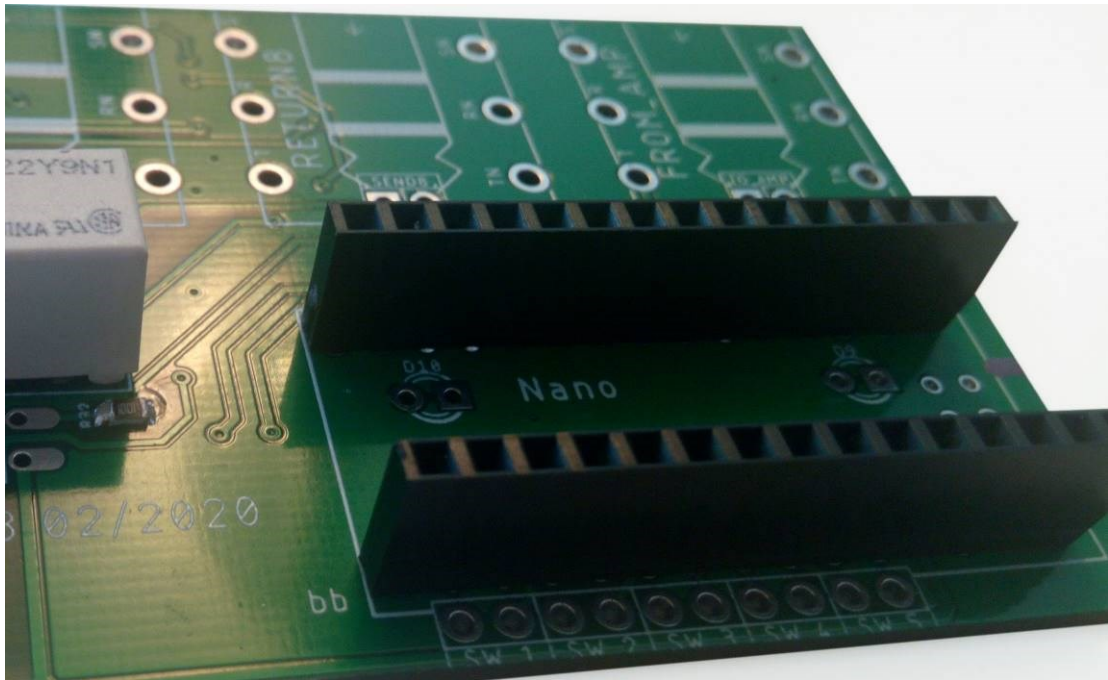


Slika 5.5. SMD polje i zalemljeni SMD otpornik (označeni crvenom bojom)



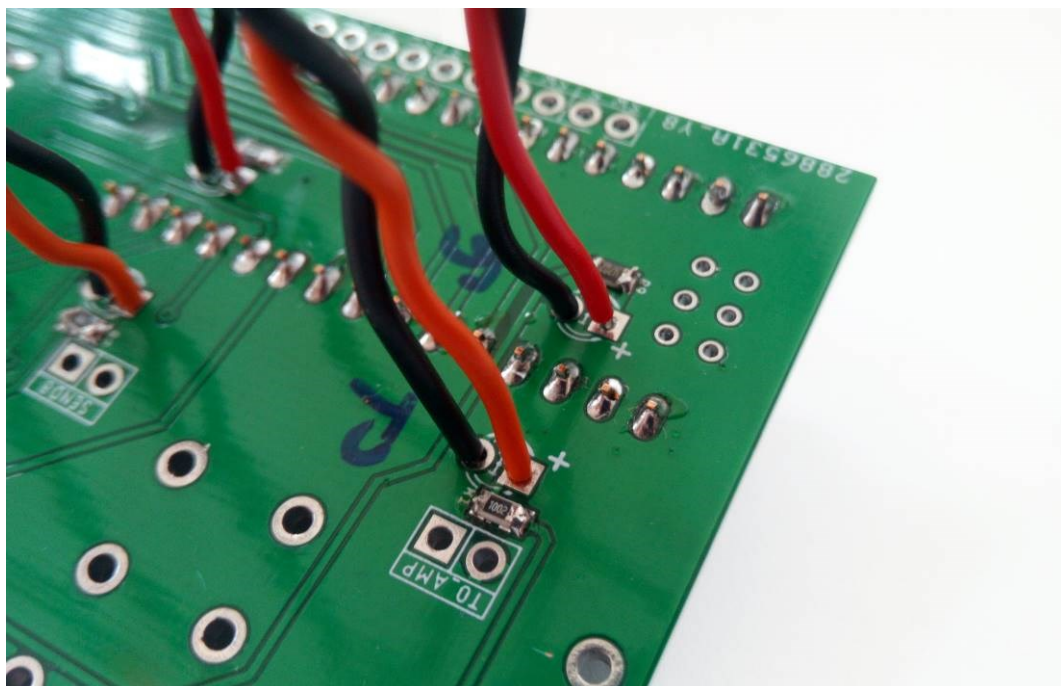
Slika 5.6. Pločica sa zalemljenim tranzistorima, SMD otpornicima i relejima

Nakon što su otpornici, tranzistori, flyback diode, releji i posmačni registri zalemljeni na pripadajuća mjesta, možemo zalemiti utore za mikroupravljačku platformu. (Slika 5.7.) Kada se koriste utori mikroupravljačka platforma može se izvaditi radi lakšeg programiranja ili zamjene u slučaju kvara.



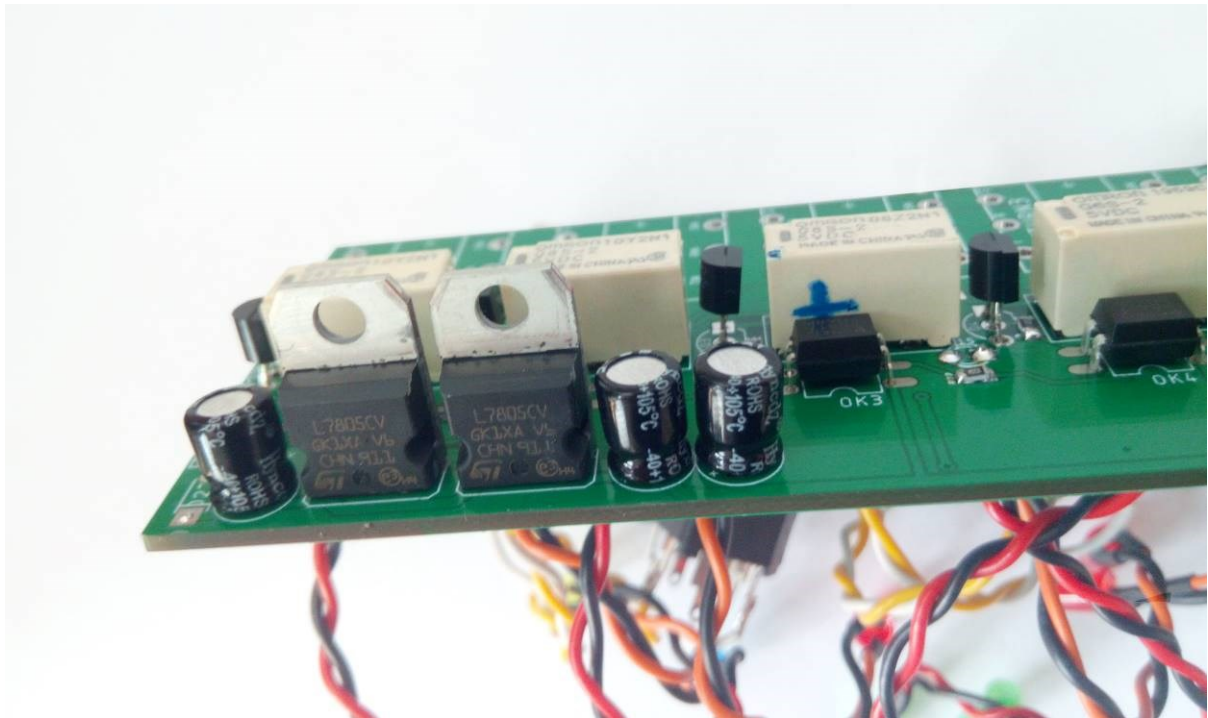
Slika 5.7. Zalemljeni utori za mikroupravljačku platformu

Nakon toga potrebno je zalemiti *LE* diode, prekidače i konektore jer se utorima za neke od komponenti neće moći pristupiti nakon što se zaleme optoizolatori. Poželjno je koristiti različito obojene žice radi lakšeg raspoznavanja gdje su pojedine komponente zalemljene. (Slika 5.8.)



Slika 5.8. Dioda i konektori žicama povezani s tiskanom pločicom

Naposljetku, mogu se zalemiti optoizolatori, a zatim naponski regulatori i kondenzatori kao najviše komponente na tiskanoj pločici. (Slika 5.9.)



Slika 5.9. Zalemljeni regulatori i kondenzatori

5.3.2. Kućište

Kućište za uređaj je aluminijska kutija veličine 253mm·53mm·52.2mm. Potrebno je označiti i probušiti utore za konektore, prekidače i svjetleće diode. Zatim kućište treba izbrusiti kako bi se uklonile nepravilnosti materijala i očistio površinski sloj. Nakon toga kućište je pripremljeno, obojeno i lakirano koristeći odgovarajući *primer* za aluminij, crnu akrilnu boju i lak u spreju.

Kada su kućište i tiskana pločica pripremljeni potrebno je prekidače, konektore i svjetleće diode umetnuti u predviđena mjesta i zategnuti matice na konektorima. (Slika 5.10.) Zadnji korak u izradi ovog uređaja je programiranje mikroupravljača.



Slika 5.10. Završeni uređaj za usmjeravanje signala kroz gitarske efekte

5.4. Izrada i objašnjenje koda

Za izradu koda i programiranje mikroupravljača korišten je Arduino IDE. Glavni zadatak programskog koda je detekcija pritiska prekidača i uključivanje odgovarajuće kombinacije releja. Uz to potrebno je korisniku prikazati koja kombinacija je uključena tj. koje petlje su uključene. Kako bi uređaj bio autonoman omogućeno je jednostavno spremanje kombinacija u dvije banke i mijenjanje načina rada *on the fly* tj. bez potrebe za računalom.

Korištene su dvije biblioteke: EasyButton.h koja služi za detekciju pritiska i držanja prekidača [25] te EEPROM.h koja nam pojednostavljuje zapisivanje i čitanje podataka s *EEPROM-a* mikroupravljača. Na početku su definirane konstante koje predstavljaju izvode mikroupravljačke platforme korištene za upravljanje relejima i posmačnim registrima. Nakon toga definirane su varijable za mjerenje vremena, čuvanje kombinacije trenutno uključenih petlji, označavanje zadnjeg pritisnutog prekidača i označavanje trenutnog načina rada i korištene

banke. Ispod konstanti i varijabli instancirano je pet objekata klase EasyButton povezanih s odgovarajućim izvodima mikroupravljača na koje su spojeni prekidači. (Slika 5.11.)

```
#include<EasyButton.h>
#include<EEPROM.h>

#define latchPin 10
#define clockPin 12
#define dataPin 11
#define modeLED A0
#define bankLED 13
int relayPin[8] = {2, 3, 4, 5, 6, 7, 8, 9};

byte preset = B00000000;
byte loops = B00000000;

unsigned long currentMillis = 0;
unsigned long previousMillis = 0;

int i = 0;
int buttonFlag = -1;
unsigned int mode = 0;
unsigned int bank = 0;

EasyButton BTN0(A1);
EasyButton BTN1(A2);
EasyButton BTN2(A3);
EasyButton BTN3(A4);
EasyButton BTN4(A5);
```

Slika 5.11. Korištene biblioteke, definirane konstante, varijable i instancirani objekti

Za usmjeravanje signala i informiranje korisnika potrebno je upravljati relejima i posmačnim registrima za što su napisane funkcije na slici 5.12. Funkcija *shiftBits()* prosljeđuje slijed bitova spremljenih u varijable *preset* i *loops* posmačnim registrima koji će ovisno o vrijednosti primljenih bitova uključiti odgovarajuće svjetleće diode. Funkciji *writeLoops(int)* prosljeđuje se redni broj petlje koju se želi uključiti, funkcija promijeni odgovarajući bit u varijabli *loops*, uključi relej i pozove funkciju *shiftBits()* kako bi se dobio odgovarajući ispis.


```

void shiftBits() {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, preset);
    shiftOut(dataPin, clockPin, LSBFIRST, loops);
    digitalWrite(latchPin, HIGH);
    delay(100);
}

void writeLoops(int num) {
    bitWrite(loops, num, !bitRead(loops, num));
    digitalWrite(relayPin[num], !bitRead(loops, num));
    shiftBits();
}

```

Slika 5.12. Funkcije za upravljanje relejima i posmačnim registrima

Spremanje i pozivanje kombinacija petlji implementirano je u funkcijama na slici 5.13. Funkcija *memorizePreset(int, int)* čita logičko stanje svakog izvoda za upravljanje relejima i to stanje zapisuje na prosljeđenu memorijsku adresu. Nakon toga pozivom pomoćne funkcije *flashPresetLED(int)* zatreperit će LE dioda koja predstavlja tu memorijsku lokaciju. Na kraju funkcije poziva se pomoćna funkcija *pullAllLow()* koja isključuje sve svjetleće diode i releje.

```

void memorizePreset(int address, int led) {
    for (i = 0; i < 8; i++)
        EEPROM.write((address) + i, bitRead(loops, i));
    flashPresetLED(led);
    pullAllLow();
}

void readPreset(int address, int led) {
    for (i = 0; i < 8; i++) {
        digitalWrite(relayPin[i], !EEPROM.read((address) + i));
        bitWrite(loops, i, EEPROM.read((address) + i));
    }
    preset = B00000000;
    bitWrite(preset, led, 1);
    shiftBits();
}

```

Slika 5.13. Funkcije za spremanje i pozivanje spremljenih kombinacija petlji

Pomoćne funkcije napravljene su kako bi se osigurala preglednost programskog koda, a prikazane su na slici 5.14. To su funkcije *calculateTime()* koja koristi ugrađenu funkciju *millis()* za računanje proteklog vremena, *pullAllLow()* koja isključuje sve svjetleće diode i releje,

flashPresetLED(int) za treperenje svjetlećih dioda te *changeBank()* i *changeMode()* koje kada se pozovu povećavaju vrijednost varijabli *bank* i *mode* i tako mijenjaju način rada i korištenu banku spremljenih kombinacija.

```

void changeBank() {
    bank++;
    digitalWrite(bankLED, !digitalRead(bankLED));
    for (i = 0; i < 8; i++) {
        if (bitRead(preset, i) == 1) {
            if (bank % 2 == 0)
                readPreset(11 + 10 * i, i);
            else if (bank % 2 == 1)
                readPreset(111 + 10 * i, i);
        }
    }
}

void changeMode() {
    mode++;
    digitalWrite(modeLED, !digitalRead(modeLED));
    pullAllLow();
}

bool calculateTime() {
    previousMillis = currentMillis;
    currentMillis = millis();
    if (currentMillis - previousMillis >= 50)
        return true;
    return false;
}

void pullAllLow() {
    loops = B00000000;
    preset = B00000000;
    shiftBits();
    for (i = 0; i < 8; i++)
        digitalWrite(relayPin[i], HIGH);
}

void flashPresetLED(int led) {
    bitWrite(preset, led, 1);
    shiftBits();
    bitWrite(preset, led, 0);
    shiftBits();
    bitWrite(preset, led, 1);
    shiftBits();
    bitWrite(preset, led, 0);
    shiftBits();
}

```

Slika 5.14. Korištene pomoćne funkcije

EasyButton biblioteka omogućuje dodjelu *callback* funkcije svakom *EasyButton* objektu. To se postiže funkcijama *onPressed()* koja poziva dodijeljenu *callback* funkciju kada je prekidač pritisnut i *onPressedFor()* koja poziva *callback* funkciju kada korisnik drži prekidač pritisnutim neko vrijeme. U ovom slučaju vrijeme je 1000 ms. Navedene funkcije pozivaju se u *setup()* funkciji gdje određujemo koji su izvodi mikroupravljača ulazi, a koji izlazi. U *loop()* funkciji pozivaju se *read()* funkcije za svaki prekidač. Time smo izbjegli korištenje *interrupta* i smanjili uporabu tzv. *blocking code*-a pa se program izvodi bez vidljivih zastoja. (Slika 5.15.)

```

void setup() {
    pinMode(latchPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(modeLED, OUTPUT);
    pinMode(bankLED, OUTPUT);
    for (i = 0; i < 8; i++)
        pinMode(relayPin[i], OUTPUT);

    BTN0.onPressed(BTN0_Pressed);
    BTN0.onPressedFor(1000, BTN0_Held);
    BTN1.onPressed(BTN1_Pressed);
    BTN1.onPressedFor(1000, BTN1_Held);
    BTN2.onPressed(BTN2_Pressed);
    BTN2.onPressedFor(1000, BTN2_Held);
    BTN3.onPressed(BTN3_Pressed);
    BTN3.onPressedFor(1000, BTN3_Held);
    BTN4.onPressed(BTN4_Pressed);
    BTN4.onPressedFor(1000, BTN4_Held);
    BTN0.begin();
    BTN1.begin();
    BTN2.begin();
    BTN3.begin();
    BTN4.begin();
    pullAllLow();
}

void loop() {
    BTN0.read();
    BTN1.read();
    BTN2.read();
    BTN3.read();
    BTN4.read();
}

```

Slika 5.15. Funkcije *setup()* i *loop()*

Glavni dio programa je prepoznavanje pritisnutog prekidača i pozivanje funkcija za uključivanje petlji, spremanje ili pozivanje kombinacija u skladu s trenutnim načinom rada i odabranom bankom. Za svaki prekidač napravljene su funkcije oblika *BTNX_Pressed()* i *BTNX_Held()* (gdje *X* predstavlja redni broj prekidača) koje dodjeljuju odgovarajući broj varijabli *buttonFlag* ovisno koji je prekidač pritisnut, a nakon toga pozivaju funkcije *Pressed()* ili *Held()* koje određuju koja se funkcija sljedeća treba pozvati. Funkcije *BTNX_Pressed()*, *BTNX_Held()* i *findPressedButton()* korištene su kako bi se izbjegla dvostruka pozivanja i slične logičke pogreške u slučaju kada bi svi *EasyButton* objekti pozivali istu *callback* funkciju. (Slika 5.16.)

Funkcije *Pressed()* i *Held()* određuju u kojem se načinu rada uređaj nalazi i koja je odabrana banka, a tada pozivaju odgovarajuće funkcije. Funkciju *writeLoops(int)* za uključivanje određene petlje, *changeBank()* i *changeMode()* za promjenu banke i načina rada, *memorizePreset(int, int)* za spremanje kombinacije na *EEPROM* i *readPreset(int, int)* za dohvaćanje željene kombinacije iz memorije. (Slika 5.17.) Na početku funkcije *Pressed()* računa se vrijeme proteklo od posljednjeg pritiska kako bi se izbjegla dvostruka pozivanja i *bouncing* prekidača. Kod funkcije *Held()* to ne predstavlja problem jer se prekidač treba držati pritisnutim te je ugrađeni *debouncing EasyButton* objekata dovoljan.

```
void BTN0_Held() {
    buttonFlag = 0;
    Held();
}

void BTN0_Pressed() {
    buttonFlag = 0;
    Pressed();
}
```

Slika 5.16. Funkcije *BTN0_Held()* i *BTN0_Pressed()*

Funkcije za ostale prekidače napisane su na isti način

```
void Pressed() {
    if (calculateTime()) {
        int number = findPressedButton();
        if (mode % 2 == 0) { //LOOP MODE
            preset = B00000000;
            if (number != -1)
                writeLoops(number);
            else
                changeBank();
        }
        else { // PRESET MODE
            if (number != -1) {
                if (bank % 2 == 0)
                    readPreset(10 * number + 1, number);
                else if (bank % 2 == 1)
                    readPreset(10 * number + 101, number);
            }
            else
                changeBank();
        }
    }
    currentMillis = millis();
    buttonFlag = -1;
}

void Held() {
    int number = findPressedButton();
    if (number != -1 && mode % 2 == 0) {
        if (bank % 2 == 0)
            memorizePreset(10 * number + 1, number);
        else if (bank % 2 == 1)
            memorizePreset(10 * number + 101, number);
    }
    else
        changeMode();
    currentMillis = millis();
}

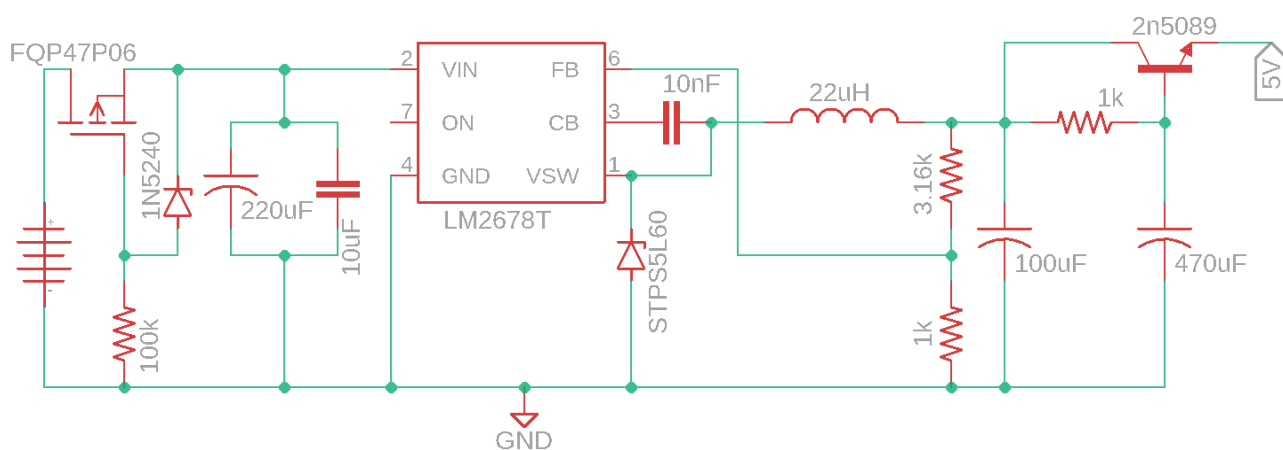
int findPressedButton() {
    if (buttonFlag == 4) {
        if (BTN4.isReleased() && BTN3.isPressed())
            return 6;
        else if (BTN4.isReleased())
            return 7;
    }
    if (buttonFlag == 3) {
        if (BTN3.isReleased() && BTN2.isPressed())
            return 4;
        else if (BTN3.isReleased() && BTN4.isPressed())
            return 6;
        else if (BTN3.isReleased())
            return 5;
    }
}
```

Slika 5.17. Korištene funkcije *Pressed()*, *Held()* i *findPressedButton()*

Zadnji korak u izradi uređaja jest prijenos programskog koda na mikroupravljač. Mikroupravljač se programira preko *USB port*-a tako što se izabere odgovarajući model mikroupravljača i *port* na koji je priključen. Nakon što je programiranje završeno, mikroupravljačka platforma može se umetnuti u utore na tiskanoj pločici. Time je završena izrada uređaja.

5.5. Poboljšanja predviđena za sljedeću verziju uređaja

Prilikom analize rada uređaja zamijećene su karakteristike uređaja koje bi se mogle poboljšati kako bi se dodatno smanjila potrošnja energije i zaštitio uređaj. To su zaštita uređaja od napona suprotnog polariteta i naponska regulacija. Umjesto naponskih regulatora moguće je koristiti istosmjerni pretvarač napona kako bismo smanjili vrijednost ulaznog napona s 8 do 30 V na 5 V. Time bi se izbjeglo pretvaranje viška energije u toplinu što je slučaj kod naponskih regulatora. Uz to moguće je koristiti P-kanalni *MOSFET* umjesto diode kao zaštitu od napona suprotnog polariteta. Energija pretvorena u toplinu na diodi jednaka je umnošku struje i napona potrebnog da dioda provede dok je na *MOSFET*-u to umnožak otpora između odvoda i uvoda $R_{DS(on)}$ i kvadrata struje. Zbog veće efikasnosti tj. manjeg zagrijavanja *MOSFET* je bolji odabir nego trenutno korištena dioda. (Slika 5.18.)



Slika 5.18. Poboljšana zaštita od napona suprotnog polariteta i naponska regulacija

6. TESTIRANJE UREĐAJA

Na uređaju su obavljena tri mjerenja: mjerenje izobličenja signala glazbenog instrumenta, mjerenje potrošnje uređaja i frekvencijski odziv uređaja (engl. *frequency response*). Frekvencijski odziv i izobličenja signala izmjereni su u laboratoriju na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek pod nadzorom mentora, a mjerenje potrošnje uređaja proveo sam samostalno.

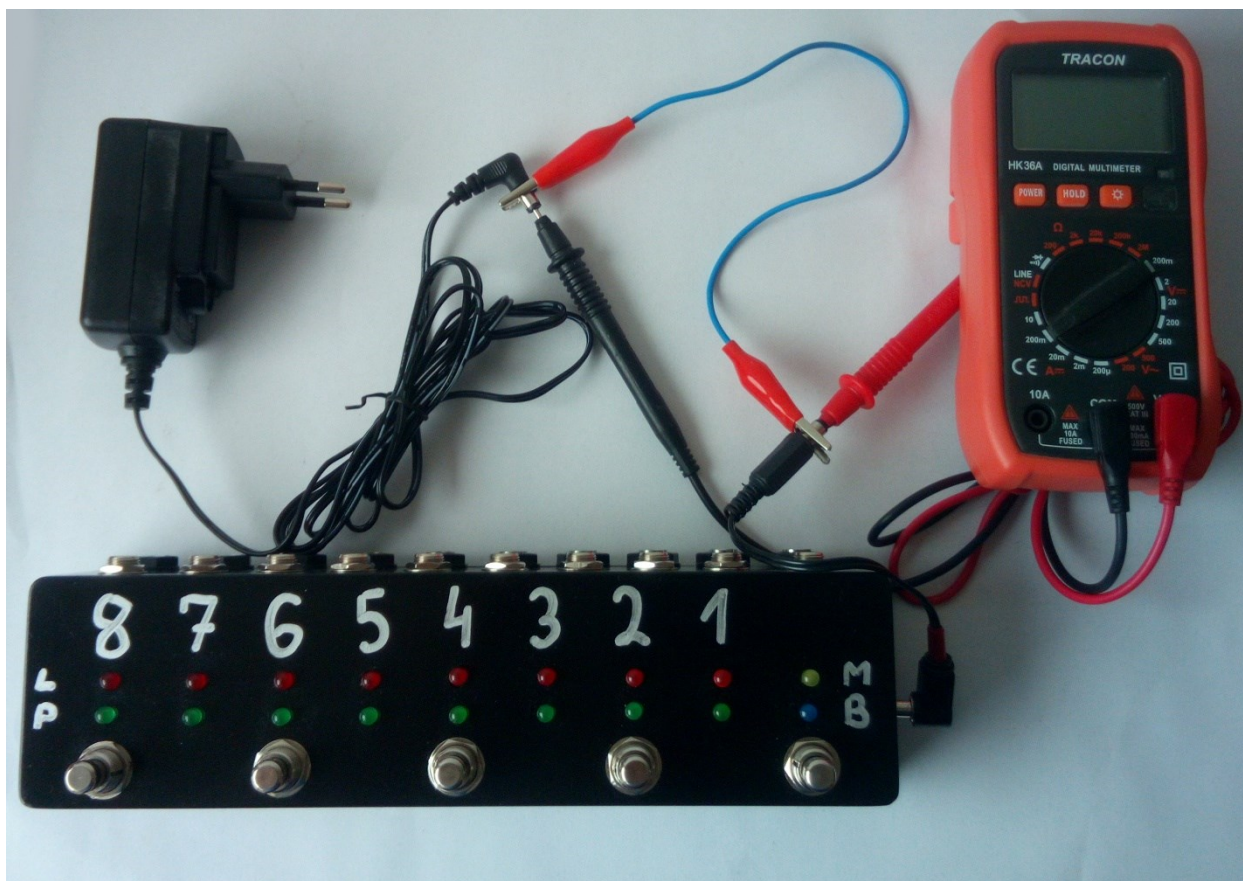
Izobličenja signala mjere se tako da se na ulaz uređaja dovede signal funkcijskog generatora te se taj signal uspoređuje sa signalom na izlazu uređaja pomoću osciloskopa. Pri mjerenju je zaključeno da ne dolazi do izobličenja signala jer signal na izlazu uređaja ne sadrži dodatne šumove ili valovitosti te je jednak ulaznom signalu.

Frekvencijski odziv uređaja mjerio se pomoću osciloskopa tako da se uspoređuju ulazni i izlazni signali. Funkcijskim generatorom stvaraju se signali različitih frekvencija te se na zaslonu osciloskopa iscrtava Bodeov dijagram. Bodeov dijagram prikazuje odnos amplitude signala i frekvencije signala. Kako uređaj ne filtrira ulazni signal, dijagram je pokazao jednaku amplitudu izlaznog signala na svim frekvencijama u rasponu od 20 Hz do 20 KHz.

Potrošnja uređaja mjerena je multimetrom s mogućnošću mjerenja struje tj. ampermetrom. Prvo se namjesti odgovarajuće mjerno područje, a zatim se ampermetar serijski spoji s napajanjem uređaja. (Slika 6.1.) Tablica 6.1. prikazuje izmjerenu struju i izračunatu snagu uređaja pri poznatom nominalnom naponu napajanja što u ovom slučaju iznosi 9 V. Formula za snagu uređaja:

$$P = U \cdot I \quad (6-1)$$

Gdje je P snaga, U nominalni napon napajanja, a I izmjerena struja uređaja. Snaga uređaja povezana je s energijom koja je potrebna za njegovo funkcioniranje, a mjeri se u vatsatima (Wh). To je energija koju uređaj potroši za vrijeme od jednog sata.



Slika 6.1. Prikaz mjerenja potrošnje uređaja

Tablica 6.1. Izmjerena struja uređaja pri različitom broju uključenih petlji

Broj uključenih petlji uređaja	Izmjerena struja uređaja	Snaga uređaja
0	26.7 mA	240.3 mW
2	105.0 mA	945 mW
4	184.8 mA	1.6632 W
8	330 mA	2.970 W

Zbog dobrog očuvanja ulaznog signala, jednakog frekvencijskog odziva na cijelom *audio* frekvencijskom spektru i maloj potrošnji energije, uz prethodno navedene mogućnosti uređaja, ovaj uređaj predstavlja održivu opciju za usmjeravanje gitarskog signala.

7. ZAKLJUČAK

U ovom je radu opisan postupak izrade i testiranja usmjerivača signala kroz gitarske efekte. Cilj je bio napraviti usmjerivač pomoću mikroupravljača i modula s relejima uz minimalna izobličenja ulaznog signala. Kako bi se zadržala mala površina koju uređaj zauzima na *pedalboard*-u, modul s relejima zamijenjen je tiskanom pločicom.

Sastavljanje sheme i dizajniranje tiskane pločice pokazali su se kao iterativni procesi gdje pronalazak ili nepronalazak elektroničke komponente sa željenim karakteristikama ili primjećivanje grešaka u dizajnu znači da se proračuni i dizajn moraju provesti ponovno. Neka od mogućih poboljšanja koja bi zahtijevala redizajn tiskane pločice navedena su u petom poglavlju.

Programiranje mikroupravljača također se može gledati kao iterativni proces gdje se piše novi programski kod (u cjelini ili jednim dijelom) kada se spoznaju nova i jednostavnija rješenja. Arduino IDE i mikroupravljačka platforma pokazali su se kao jednostavni i intuitivni alati za rad s mikroupravljačima ako korisnik posjeduje osnove programiranja u C i C++ ili sličnim programskim jezicima.

Nakon testiranja i usporedbe s trenutno dostupnim usmjerivačima gitarskog signala zaključeno je da je uređaj ispunio sve zahtjeve postavljene u zadatku završnog rada te predstavlja adekvatno rješenje za usmjeravanje gitarskog signala.

LITERATURA

- [1] Effects Unit - https://en.wikipedia.org/wiki/Effects_unit [22.4.2020.]
- [2] C. Corfield, Guitar Pedals Explained - <https://www.dawsons.co.uk/blog/guitar-pedals-explained> [22.4.2020.]
- [3] Sweetwater, Pedalboard - <https://www.sweetwater.com/insync/pedalboard-2> [22.4.2020.]
- [4] B. O'Neal, State of the Stomp: Do I Need a Loop Switcher on My Pedalboard - <https://www.premierguitar.com/articles/29518-state-of-the-stomp-do-i-need-a-loop-switcher-on-my-pedalboard> [22.4.2020.]
- [5] G6S-2-5VDC Relay Datasheet - https://omronfs.omron.com/en_US/ecb/products/pdf/en-g6s.pdf [30.4.2020.]
- [6] 3PDT Pushbutton Datasheet - <https://hr.mouser.com/datasheet/2/140/FS5700-1147045.pdf> [30.4.2020.]
- [7] StompControl-4 - https://www.thomann.de/gb/harley_benton_stompcontrol_4_iso.htm
- [8] GEC8 JR - https://www.thomannmusic.com/moen_gec8_jr_looper_switcher.htm
- [9] Super Switcher - https://www.thomann.de/gb/electro_harmonix_super_switcher.htm
- [10] J. Lindblom, Using Eagle: Schematic - <https://learn.sparkfun.com/tutorials/using-eagle-schematic> [7.5.2020.]
- [11] J. Lindblom, Using Eagle: Board Layout - <https://learn.sparkfun.com/tutorials/using-eagle-board-layout> [7.5.2020.]
- [12] C.J. Abate, What Is a Microcontroller? - <https://www.elektormagazine.com/news/what-is-a-microcontroller> [31.4.2020.]
- [13] Sparkfun, What is an Arduino? - <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all> [31.4.2020.]
- [14] Arduino, Introduction - <https://www.arduino.cc/en/Guide/Introduction> [6.5.2020.]
- [15] Arduino Store, Nano - <https://store.arduino.cc/arduino-nano>
- [16] T. Joseph, Arduino Nano V3.0 Specification - <https://www.tomsonelectronics.com/blogs/news/arduino-nano-v3-0-specification> [31.4.2020.]
- [17] Arduino, Environment - <https://www.arduino.cc/en/Guide/Environment> [6.5.2020.]
- [18] P. Horowitz, W.Hill, The Art of Electronics Third Edition, Cambridge University Press, Ujedinjeno Kraljevstvo, 2015.

- [19] D. Self, Small Signal Audio Design, Elsevier Ltd., Ujedinjeno Kraljevstvo, 2010.
- [20] K. Robinson, Practical Audio Electronics, Routledge, Sjedinjene Američke Države, 2020.
- [21] I. Sinclair, J. Dunton, Practical Electronics Handbook, Newnes, Sjedinjene Američke Države, 2007.
- [22] 5mm Diffused LED Datasheet - <https://www.vishay.com/docs/83171/tlur640.pdf>
[10.5.2020]
- [23] PC817 Optocoupler Datasheet - <https://www.farnell.com/datasheets/73758.pdf>
[10.5.2020]
- [24] 2N3904 NPN BJT Datasheet - <https://www.onsemi.com/pub/Collateral/2N3903-D.PDF> [10.5.2020]
- [25] EasyButton Arduino Library - <https://easybtn.earias.me> [11.5.2020.]

SAŽETAK

U ovom su radu opisane teorijska i praktična strana izrade uređaja za usmjeravanje signala kroz gitarske efekte korištenjem mikroupravljača i releja. U prvom je dijelu rada obrazložena potreba za takvim uređajem, a nakon toga opisan je način rada korištenih alata i tehnologija.

U drugom dijelu rada analizirane su karakteristike dizajniranog uređaja uz naznačene dijelove sheme na koje se analiza odnosi. Nakon toga prikazan je postupak sastavljanja uređaja na dizajniranoj tiskanoj pločici. Na kraju je objašnjen programski kod mikroupravljača koji omogućava željeni rad uređaja.

Ključne riječi: mikroupravljač, Arduino, relej, optoizolator, gitarski signal, usmjeravanje, tiskana pločica

ABSTRACT

This paper describes theoretical and practical sides of designing a guitar effects router by means of microcontroller and relays. In the first part of the paper the need for such a device is reasoned, after which the operation of used tools and technologies is described.

In the second part of the paper, the characteristics of the designed device are analyzed with indicated parts of the schematic to which the analysis refers. Then, the procedure of assembling the device on designed printed circuit board is shown. Lastly, the program code of the microcontroller which ensures the desired operation of the device is explained.

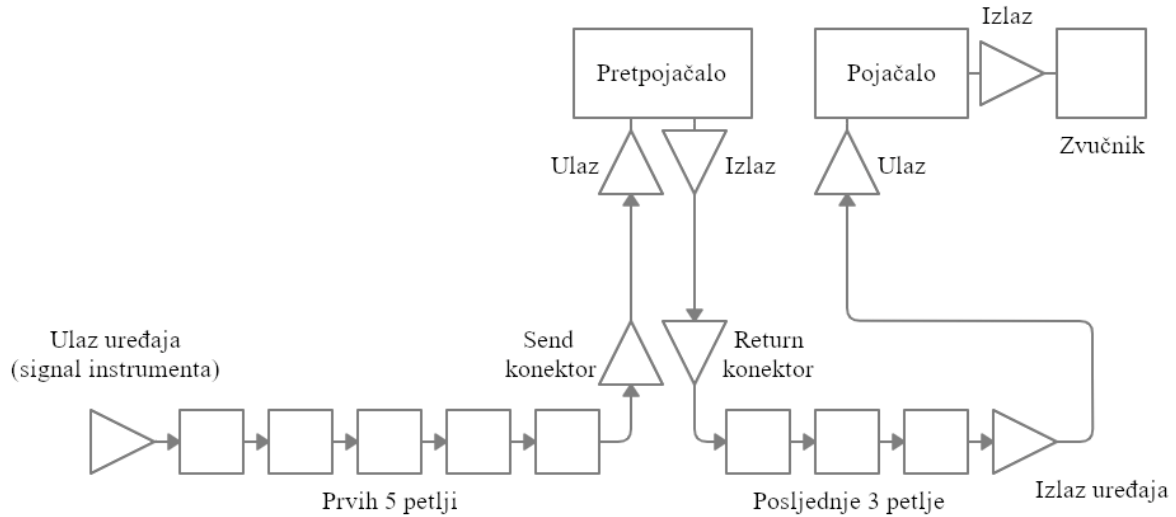
Key words: microcontroller, Arduino, relay, optocoupler, guitar signal, routing, printed circuit board

ŽIVOTOPIS

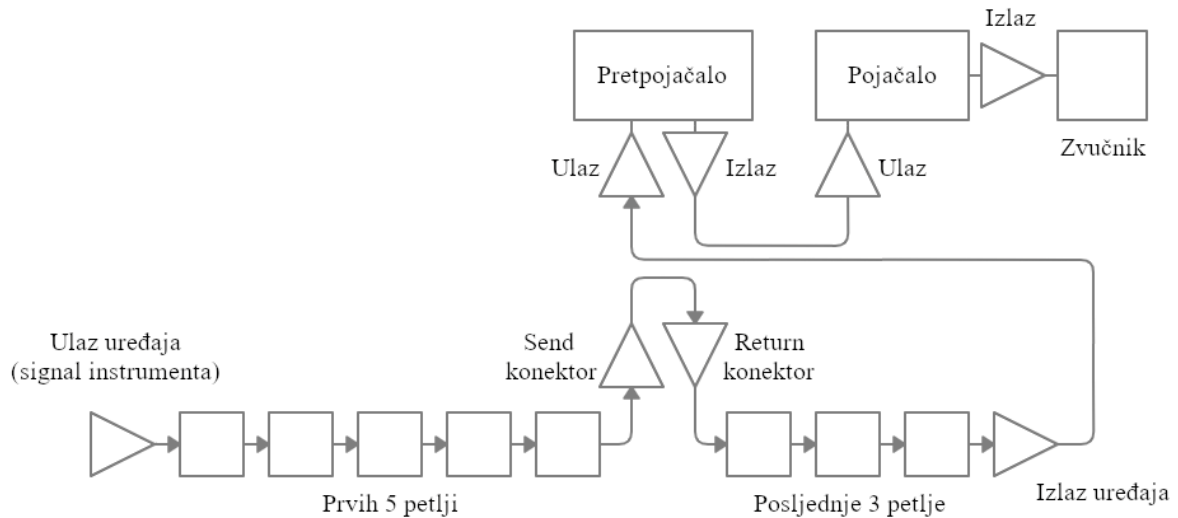
Borna Benja rođen je 28.6.1998. u Osijeku. Poslije završetka osnovne škole u Iloku upisao se u III. Gimnaziju u Osijeku. Godine 2017. završava gimnazijski program te se upisuje na sveučilišni preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI

Prilog 5.1. Blok dijagrami spajanja uređaja s pojačalom i pretpojačalom



Dijagram 1. Spajanje uređaja tako da su prvih 5 petlji ispred pretpojačala, a posljednje 3 petlje iza pretpojačala (tzv. four cable method)



Dijagram 2. Spajanje uređaja tako da su svih 8 petlji ispred pretpojačala

Prilog 5.2. Popis korištenih komponenti

Otpornici	Otpor [Ω]	Pakiranje
<i>R1, R2, R3, R4, R5, R6, R7, R8</i>	22 K	SMD 0805
<i>R9 do R16</i>	47 K	
<i>R17, R19, R21, R23, R25, R27, R29, R31</i>	510	
<i>R18, R20, R22, R24, R26, R28, R30, R32</i>	1 K	
<i>R33, R34</i>	10 K	
Kondenzatori	Kapacitet [F]	Pakiranje
<i>C1, C2, C3</i>	100 μ	Elektrolitski kondenzator promjera 6.3mm
Diode	Naziv	Pakiranje
<i>D1 do D16, D25, D26</i>	LE diode	Promjer 5mm
<i>D17 do D24</i>	1N4001	DO-41
<i>D27</i>	1N5820	
Tranzistori	Naziv	Pakiranje
<i>Q1 do Q8</i>	2N3904	TO-92
Optoizolatori	Naziv	Pakiranje
<i>OK1 do OK8</i>	PC817	DIP4
Posmačni registri	Naziv	Pakiranje
<i>U1, U2</i>	74HC595	SO16
Regulatori	Naziv	Pakiranje
<i>REG_1, REG_2</i>	LM7805	TO-220
Releji	Naziv	Pakiranje
<i>RELAY1 do RELAY8</i>	G6S-2-5VDC	DIP10
Konektori	Naziv	Pakiranje
<i>J1 do J16</i>	Konektor	6.35mm
<i>DC1</i>	Konektor	2.1mm
Prekidači	Naziv	Pakiranje
<i>SW_1 do SW_5</i>	SPST prekidači	Nožni prekidači, 12mm

Mikroupravljačka platforma	Naziv mikroupravljača	Pakiranje
U3	ATmega328	DIP30 – nano

Prilog 5.3. Nacrt tiskane pločice

