

»Mladi za napredek Maribora 2019«

36. srečanje

MIDI MEŠALNA MIZA

Raziskovalno področje **ELEKTROTEHNIKA, ELEKTRONIKA**

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor: ALJAŽ ROŽIČ

Mentor: BOJAN DEŽMAN

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Število točk: 163

Mesto: 1

Priznanje: zlato

Maribor, 2019

»Mladi za napredek Maribora 2019«

36. srečanje

MIDI MEŠALNA MIZA

Raziskovalno področje **ELEKTROTEHNIKA, ELEKTRONIKA**

Raziskovalna naloga

PROSTOR ZA NALEPKO

Maribor, 2019

1. Kazalo

1. Kazalo	3
2. Kazalo slik	5
3. Zahvala.....	1
4. Povzetek.....	2
5. Uvod.....	3
6. Vsebinski del	4
6.1 Teensy.....	4
6.2 De-multiplekser 74hc4067 in 74HC4051	5
6.3 Raspberry pi 3B+.....	6
6.4 Zaslona na dotik	7
6.5 Dmx usb pretvornik.....	8
6.6 Polnilnik litijevskih in lipo baterij	8
6.6.1 Integrirano vezje za polnjenje litijevskih baterij TP4056	8
6.6.2 Integrirano vezje za zaščito baterij FS312F-G	9
6.6.3 Step up integrirano vezje MT3608	10
6.7 Tipka.....	11
6.8 USB razdelilec.....	12
6.9 QLC +.....	12
6.9.1 Zavihek Fixtures.....	13
6.9.2 Zavihek Functions.....	14
6.9.3 Zavihek Shows.....	15
6.9.4 Zavihek Virtual Console	16
6.9.5 Zavihek Simple Deslk	17
6.9.6 Zavihek Input/Output.....	18
6.10 MIDI.....	19

6.10.1	Sporočila kanalu (channel messages).....	19
6.10.2	Sistemska sporočila.....	20
6.11	Dmx.....	20
6.11.1	Delovanje DMX sistema	20
6.11.2	Sestava DMX paketa.....	21
7.	Programska koda	22
7.1	Midi mešalna miza.....	22
7.2	Namestitev programa QLC+.....	25
8.	Izdelava	26
8.1	Eagle	26
8.1.1	Midi kontroler	26
8.1.2	Izdelava polnilnega vezja	31
8.1.3	Dmx kontroler.....	31
8.2	3D načrtovnje	33
8.2.1	Izdelovanje vezji v 3d obliki.....	33
8.2.2	Risanje ohišja v 3d.....	33
8.3	Maketa.....	36
9.	Zaključek	37
10.	Družbena odgovornost	37
11.	Viri	38

2. Kazalo slik

Slika 1 Teensy 3.5 (vir: www.sparkfun.com).....	4
Slika 2 multiplekser (viri: assets.nexperia.com).....	5
Slika 3 Raspberry 3B+ (vir: https://c1.neweggimages.com).....	6
Slika 4 Zaslon na dotik (vir: www.raspberrypi.org).....	7
Slika 5 Integrirano vezje za polnjenje lit ionskih baterij TP4056 ((vir: lcsc.com)).	8
Slika 6 Integrirano vezje za zaščito baterij FS312F-G (vir: lcsc.com).....	9
Slika 7 Step up integrirano vezje MT3608 (vir: lcsc.com).....	10
Slika 8 Logična shema vezja (vir: https://www.olimex.com)	10
Slika 9 Prerez tipke (vir:Avtor naloge).	11
Slika 10 Karieristka vklopa in izklopa (vir:Avtor naloge).	12
Slika 11 QLC+ (vir: www.qlcplus.org).....	12
Slika 12 Zavihek Fixtures (vir:Avtor naloge).	13
Slika 13 Zavihek Functions (vir:Avtor naloge).....	14
Slika 14 Zavihek Shows (vir:Avtor naloge).	15
Slika 15 Zavihek Virtual Console (vir:Avtor naloge).....	16
Slika 16 Zavihek Simple Deslk (vir:Avtor naloge).	17
Slika 17 Zavihek Input/Output (vir:Avtor naloge).	18
Slika 18 Primer uporabe MIDI standarda (vir: https://cdn.sparkfun.com)	19
Slika 19 Sestava dmx-okna (vir: playground.arduino.cc).....	21
Slika 20 Shema 1 modula (vir:Avtor naloge).....	27
Slika 21 Tiskanina 1 modula (vir:Avtor naloge).....	28
Slika 22 Shema modula 2 (vir:Avtor naloge).....	29
Slika 23 Tiskanina modul 2 (vir:Avtor naloge).....	30
Slika 24 Shema polnilnega vezja (vir: Avtor naloge).....	31
Slika 25 Shema usb DMX pretvornik (vir:Avtor naloge).	32
Slika 26 Tiskanina USB DMX pretvornik (vir:Avtor naloge).	32
Slika 27 Vezja v 3d obliki (vir:Avtor naloge).....	33
Slika 28 Razporeditev vezij (vir:Avtor naloge).	34
Slika 29 2d model ohišja (vir:Avtor naloge).....	34
Slika 30 3D model ohišja (vir:Avtor naloge).	35
Slika 31 Model z vsemi komponentami (vir:Avtor naloge).	35

Slika 32 DMX sprejemnik (viri: www.banggood.com).	36
Slika 33 Načrt makete (vir: Avtor naloge).	36
Slika 34 Model makete (vir: Avtor naloge).	37

3. Zahvala

Zahvaljujem se podjetju Lasertehnik za pomoč pri izdelavi ohišja za to raziskovalno nalogo. Zahvaljujem se metroju in staršem, ki so mi nudili pomoč pri izdelavi raziskovalne naloge. Zahvaljujem pa se tudi šoli, ki mi je nudila prostor in potreben material za izdelavo raziskovalne naloge.

4. Povzetek

V tej raziskovalni nalogi bom predstavil delovanje MIDI komunikacijskega protokola. S pomočjo mikrokontrolerja teensy bom izdelal mešalno mizo namenjeno za opravljanje odrskih luči in glasbe v gledališčih. Mešalna miza bo imela vse funkcionalnosti profesionalnih mešalnih miz za majhno ceno. Imela bo možnost izbire načina delovanja in prikaz vrednosti posameznih potenciometrov in gumbov na zaslonu na dotik. Imela pa bo tudi možnost kontroliranja odrskih luči brez računalnika preko vgrajenega dmx vmestnika. Na sami mešalni mizi pa bo tipkovnica, ki bo služila za pomoč pri programiranju nastavitev luči v programu luči.

5. Uvod

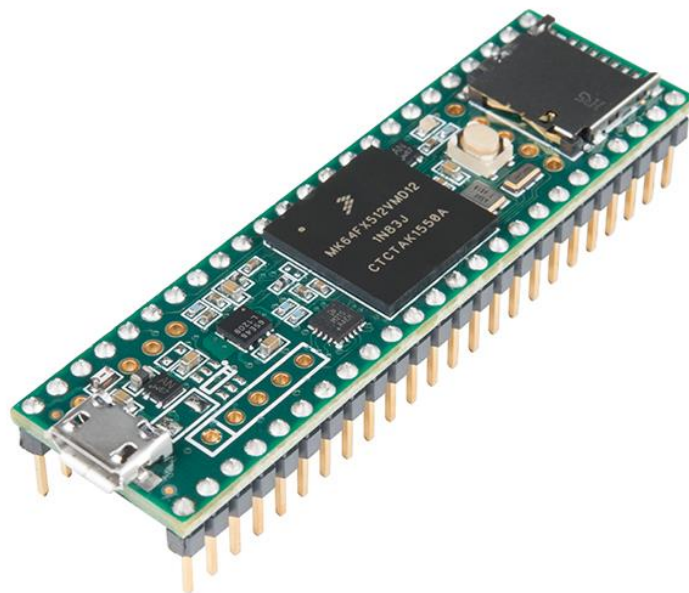
Za raziskovalno nalogo sem se odločil, ker smo v domačem lutkovnem gledališču začeli prenovo razsvetljave. Profesionalne mešalne mize s takšnimi funkcijami bi stale več tisoč eurov in bi imeli omejeno delovanje na samo določen program.

Ker je bila stara mešalna miza le za osnovno uporabo prižiganja luči, bom v tej nalogi izdelal mizo, ki bo lahko opravljala zahtevne animacije a bo kljub temu enostavna za uporabo. Imela bo vgrajeno MIDI tipkovnico, kar bo omogočalo fleksibilnost pri uporabi programov in različnih računalnikov in vgrajen USB DMX pretvornik.

Imela pa bo vgrajen računalnik, s katerim bo lahko delovala samostojno. V samem ohišju pa bo tudi baterija z zaščito , ki bo omogočala samostojno delovanje mize brez potrebe po napajanju za 5 ur.

6. Vsebinski del

6.1 Teensy



Slika 1 Teensy 3.5 (vir: www.sparkfun.com).

Za pretvorbo oziroma spreminjanje signalov potenciometrov v MIDI podatek sem uporabil razvojno ploščico teensy 3.5. Teensy 3.5 se ponaša s 32 bitnim 120 MHz procesorjem , kar omogoča hitro delovanje pri več zahtevnih logičnih operacijah. Ima 58 digitalnih vhodno/izhodnih pinov, med katerimi je kar 20 takšnih, ki jih lahko uporabljamo za PWM izhode in 27 analognih vhodno/izhodnih pinov.

Primeren je za zahtevnejše projekte, oziroma tiste projekte, ki potrebujejo hitro delovanje in veliko vhodno izhodnih enot. kot je moja raziskovalna naloga.

Razvojna ploščica se programira s brezplačnim razvojnim orodjem Arduino IDE.

Z računalnikom komunicira preko USB priključka. Ploščica pa ima tudi funkcijo, da lahko deluje kot Midi naprava, tipkovnica ali igralna ročica. To možnost bom izkoristil v nadaljevanju .

Dodatne specifikacije :

- Delovna napetost 3.3V tolerira tudi 5 V .
- Napajalna napetost (priporočena) 3.6-6 V.
- 120 MHz ARM Cortex-M4 procesor
- 512K Flash pomnilnika , 256K RAM pomnilnika , 4K EEPROM
- 1 CAN Bus Port
- 16 General Purpose DMA Channels

6.2 De-multiplekser 74hc4067 in 74HC4051



Slika 2 multiplekser (vir: assets.nexperia.com).

Zaradi velikega števila potrebnih vhodov bom uporabil digitalni de-multiplekser, ki bo služil za povečanje števila vhodov na mikrokrmilnik. V raziskovalni nalogi bom uporabil dva različna de-multiplekserja, ki se razlikujeta le po številu pinov oziroma različnih kombinacij. Prednost je tudi ta, da bom lahko povezal celotni modul s 48 elementi s 10 žicami kar bo omogočala modularnost modulov ter njihovo dodajanje po potrebi. V našem primeru bom uporabil multiplekser za branje analognih in digitalnih vrednosti. Pri tem moram biti pozoren, saj lahko beremo samo analogne ali samo digitalne vrednosti na enem multiplekserju.

Multiplekser deluje tako:

Multiplekser ima v tem primeru 4 ali 3 kontrolne pine. Te pini mi omogočajo do $2^4 = 16$ ali $2^3 = 8$ različnih vrednosti oziroma vhodov. Ob stanju določenega kontrolnega pina je na vhodu napetost določenega vhodnega pina multipleksrja. Kar pomeni, če imamo

vrednost HIGH na s0 in s1 bomo brali vrednost iz pina 3. Vse kombinacije pa so podane v podatkovnem listu obeh komponent.

6.3 Raspberry pi 3B+



Slika 3 Raspberry 3B+ (vir: <https://c1.neweggimages.com>).

Raspberry bom uporabil kot nadzorni računalnik za nadzor luči in glasbe ob odsotnosti glavnega računalnika. Raspberry pi je mikroračunalnik, ki ga odlikuje majhna velikost (velikost mobilnega telefona) in zelo prijazno programsko okolje oziroma operacijski sistem Linux. Raspberry pi je razvila fundacija z istim imenom kot sama ploščica z namenom učenja programiranja in osnov računalnika v šolah kasneje pa je postal računalnik priljubljen v različnih drugih panogah (robotika, hobiji itd.). Raspberry pi 3b+ vsebuje vse kar potrebuje računalnik. Ima procesor, grafično enoto, ram, usb prikllope, HDMI, Wifi, Bluetooth, Ethernet. Ne vsebuje pa fizičnega pomnilnika, namesto njega pa ima rezo za SD kartico. Na SD kartico predhodno naložimo operacijski sistem. Izbiramo lahko med različnimi operacijskimi sistemi kot so Windows 10 IoT Core, Ubuntu Mate, Openelec, OSMC, PiNet itd.. Priporočljiv je operacijski sistem Raspbian, saj je narejen posebej za ta računalnik in ga bom uporabljal v tej raziskovalni nalogi. Računalnik ima serijski vhod za kamero in LCD zaslon, kar bom uporabil kasneje. Ima pa tudi 17 digitalnih pinov, ki se lahko uporabijo v raznih projektih.

6.4 Zaslón na dotik



Slika 4 Zaslón na dotik (vir: www.raspberrypi.org).

Za lažje programiranje in izbiro luči bom uporabil zaslón na dotik. Zaslón deluje preko serijskega vmesnika, ki ga ima raspberry pi že vgrajen na sami ploščici. Zaslón je razvila enaka fundacija kot raspberry pi. Tako ima ta zaslón najboljšo podporo za delovanje v Rasberian operacijskem sistemu. Sam zaslón ima resolucijo 800 x 480 pikslov. Ima možnost zaznave do 10 dotikov na enkrat, kar zadošča za normalno uporabo.

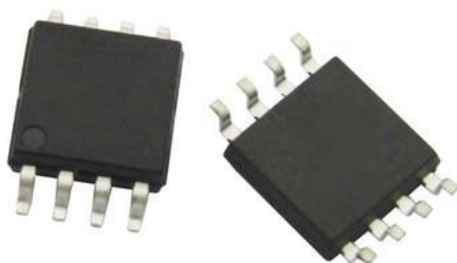
6.5 Dmx usb pretvornik

Za povezavo računalnika in samih luči bom uporabil usb na dmx pretvornik. Seveda že na trgu obstajajo takšni pretvorniki, a se njihova cena giblje okoli 100 Eur. Izdelava samega pretvornika je lahko dokaj preprosta, če upoštevamo, da dmx protokol deluje po standardu RS-485. Obstajajo pa tudi odprtokodne rešitve. Ena teh rešitev je uporaba FIDI integriranega vezja, za serijsko komunikacijo, ki s pomočjo gonilnika deluje kot prevajalnik Tako lahko strošek znižamo na samo 15 eurov na pretvornik.

6.6 Polnilec litionskih in lipo baterij

Midi mešalna miza bo imela tudi lit ionsko baterijo, da bo lahko delovala brez potrebe po omrežnem napajanju, kar bo omogoča samostojno delovanje in fleksibilnost pri prenosu same naprave v primeru gostovanja v drugem gledališču ali odru. Polnilec bo moral imeti polnilni tok 1A, stalno napetost 4.2 V in zaščito , vseboval bo tudi napetostni ojačevalec angl. step up , kar bo omogočalo napajanje računalnika raspberry pi s 5v.

6.6.1 Integrirano vezje za polnjenje lit ionskih baterij TP4056



Slika 5 Integrirano vezje za polnjenje lit ionskih baterij TP4056 ((vir: lcsc.com).

Za polnjenje baterije sem izbral integrirano vezje TP4056. TP4056 vsebuje vse kar potrebujemo za polnjenje limonskih/lipo baterij. Uporablja se v telefonih ,GPS navigacijah, fotoaparatih oziroma v vseh prenosnih napravah z lionsko baterijo Vsebuje generator stalnega toka in generator stalne napetosti, kar omogoča pravilno

in varno polnjenje lipo in litionskih baterij. Odlikuje ga tudi majhna velikost in preprosto vezje. Vsebuje termalno zaščito, ki v primeru, da se čip segreje avtomatsko zmanjša polnilni tok. Polnilna napetost je stalnih 4.2 V , pomnilni tok lahko nastavimo samo z enim uporom, kar zelo olajša samo projektiranje vezja.

Za najbolj uporabne pomnilne tokove, pa že imamo podane vrednosti upora. Integrirano vezje ima vgrajena tudi izhoda led diod, s katerima lahko prikažemo polnjenje in konec polnjenja. Vezje ima vgrajeno tudi zaščito za baterijo, ki v primeru manjše napetosti kot 2,9 V izključi izhod, ob 1/10 željenega pomnilnega toka pa konča polnjenje.

6.6.2 Integrirano vezje za zaščito baterij FS312F-G



Slika 6 Integrirano vezje za zaščito baterij FS312F-G (vir: lcsc.com).

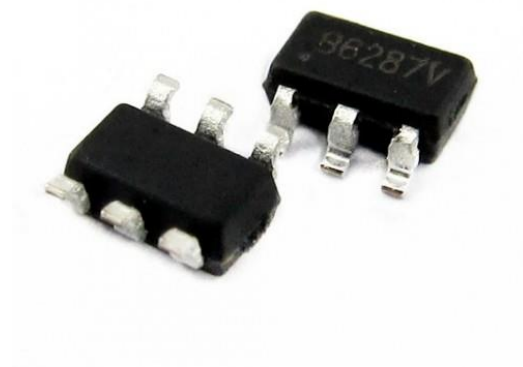
Zaradi same varnosti in podaljšanja življenjske dobe baterije bom uporabi integrirano vezje FS312F-G. Integrirano vezje ima naslednje funkcije.

- Zaščita pred kratkim stikom
- Prenapetostna zaščita polnjenja
- Pod napetostna zaščita praznjenja

Izvedba samega vezja je dokaj enostavna , saj je za tipično uporabo že podano vezje v specifikacijah samega vezja. Vse kar je potrebno dodati, sta dva mosfet tranzistorja.

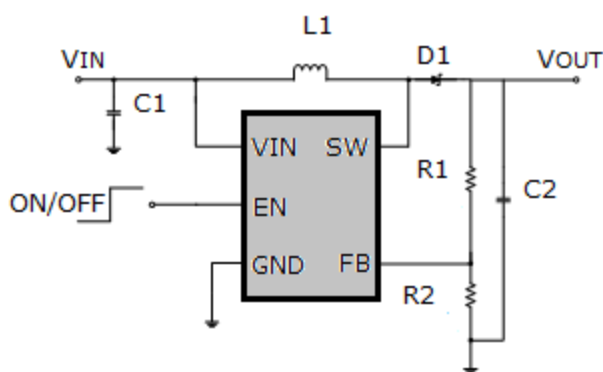
Najboljša možnost je bila uporaba dvojnega mosfet integriranega vezja FS8205 saj sta v enem ohišju dva mosfet tranzistorja.

6.6.3 Step up integrirano vezje MT3608



Slika 7 Step up integrirano vezje MT3608 (vir: lcsc.com).

Ker imam baterija napetost 3.7V , raspberry pi pa potrebuje 5v moramo napetost baterije ojačati na 5v. To bom storil z integriranim vezjem MT3608. To vezje ima vhodno napetost med 2 do 24V in izhodno napetost do 28V.



Slika 8 Logična shema vezja (vir: <https://www.olimex.com>) .

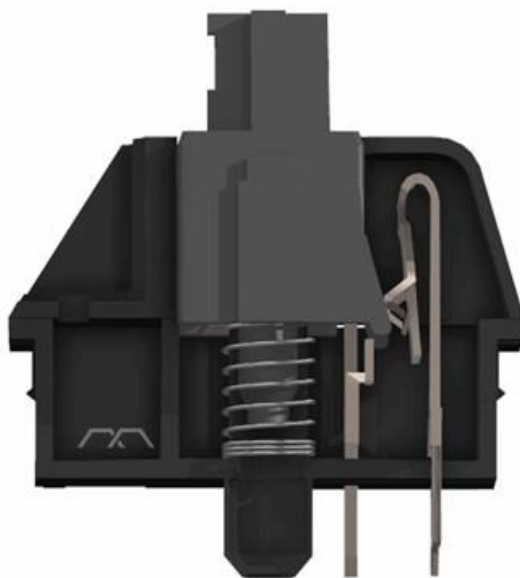
Tako kot ostala integrirana vezja ima tudi ta tipično vezje s podanimi elementi. Izhodno napetost lahko nastavljamo z uporom R1 in R2 po enačbi $V_{OUT} = V_{REF} \times (1 + \frac{R_1}{R_2})$ vgrajena referenčna napetost je 0.6V (V_{REF}) . Nato sledi izbira tuljave in kondenzatorja

, izbral sem največjo priporočeno vrednost tuljave in kondenzatorja, ki je 22μH in 22μF kar pripomore k boljšemu izkoristku vezja in boljšemu delovanju. Za diodo, ki je predpisana v vezju sem izbral sockijevo diodo, ker ima boljšo karakteristiko pri delovni frekvenci samega integriranega vezja. Tok diode se izračuna po naslednji enačbi $I_D \approx$

$$\sqrt{I_{OUT} \times I_{PEAK}}$$

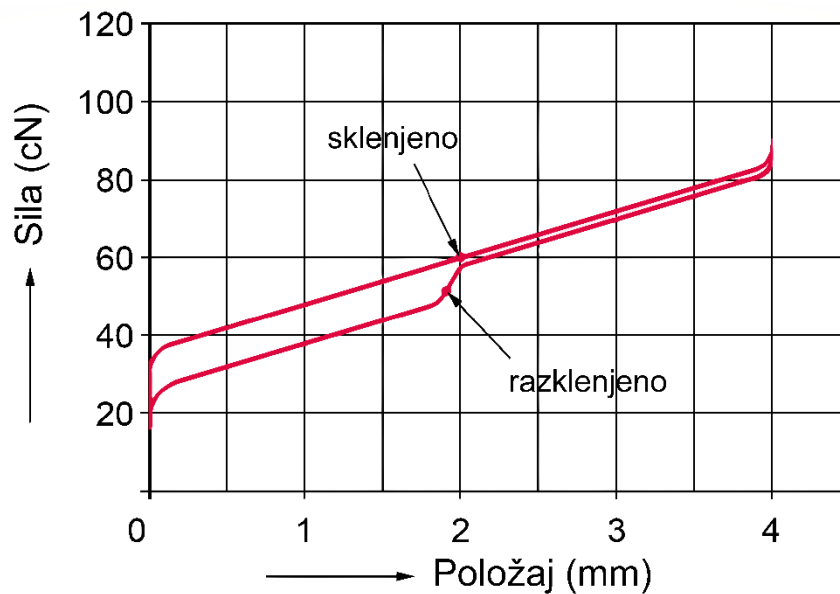
Za najboljšo možno delovanje sem moral vsa tri vezja povezati v celoto.

6.7 Tipka



Slika 9 Prerez tipke (vir: Avtor naloge).

Za to raziskovalno nalogo sem uporabil posebno vrsto tipk. Te tipke se uporabljajo za mehanične tipkovnice, ki se od navadnih membranskih tipkovnic razlikujejo v tem, da jo sestavljajo vzmet in mehanska kontakta, ki skleneta ali razkleneta povezavo. Tipka pa ima tudi prednost saj ima točno določen položaj v katerem se sklene in razklene. Prednost pa je tudi v tem, da lahko tipko preprosto vstavimo v vezje in ne potrebujemo posebnih pedov v primerjavi z membranskimi tipkovnicami oziroma tipkami.



Slika 10 Karieristka vklopa in izklopa (vir: Avtor naloge).

6.8 USB razdelilec

Midi mešalna miza bo imela tudi možnost preklopa na zunanji računalnik. Ker vsebuje več različnih usb naprav bom izdelal razdelilec za USB naprave. Tako bom lahko z računalnikom komuniciral preko enega usb kabla.

6.9 QLC +



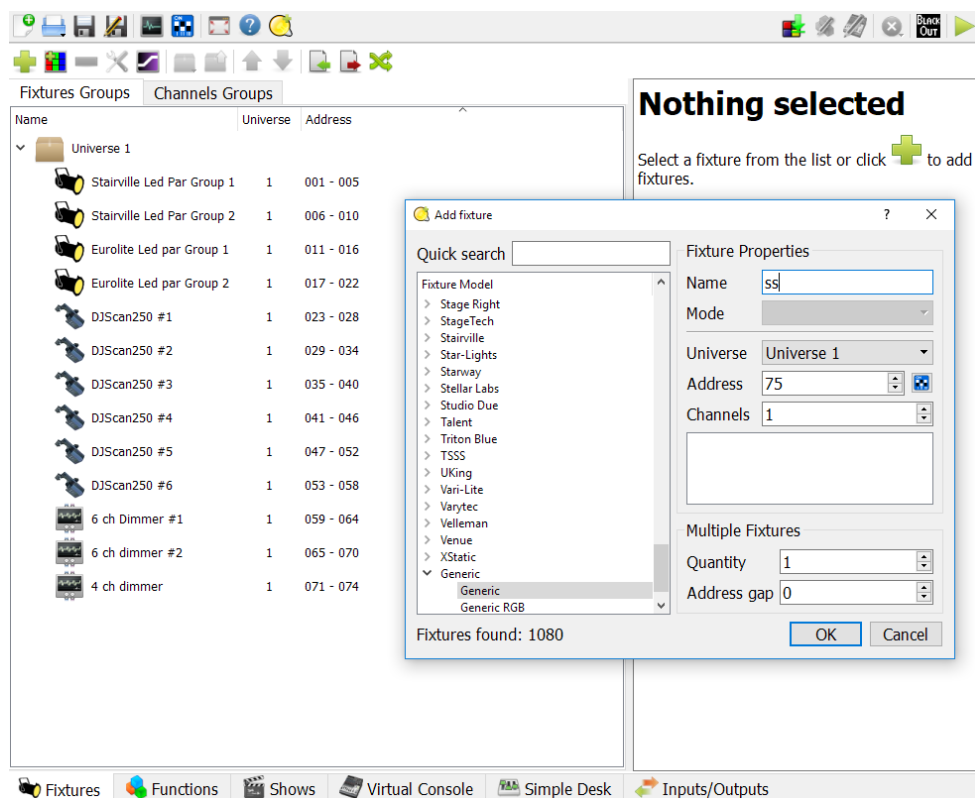
Slika 11 QLC+ (vir: www.qlcplus.org).

QLC + je odprtokodni program za nadzor odrskih luči na različnih prireditvah, kot so koncerti v živo, gledališča itd. Nudi preprost in prilagodljiv grafični vmesnik. Omogoča

uporabo zaslona na dotik in ima vgrajeno spletno stran na kateri lahko nadzorujemo luči preko spletne strani. To mogoča, da luči opravljamo na daljavo s pomočjo mobilnega telefona ali tablice. Deluje na vseh večjih operacijskih sistemih, tako deluje tudi na raspberry pi ju kamor ga bom namestil. QLC+ se deli na 6 različnih zavihkov :

- Fixtures
- Functions
- Shows
- Virtual Console
- Simple Desk
- Input/Output

6.9.1 Zavihek Fixtures

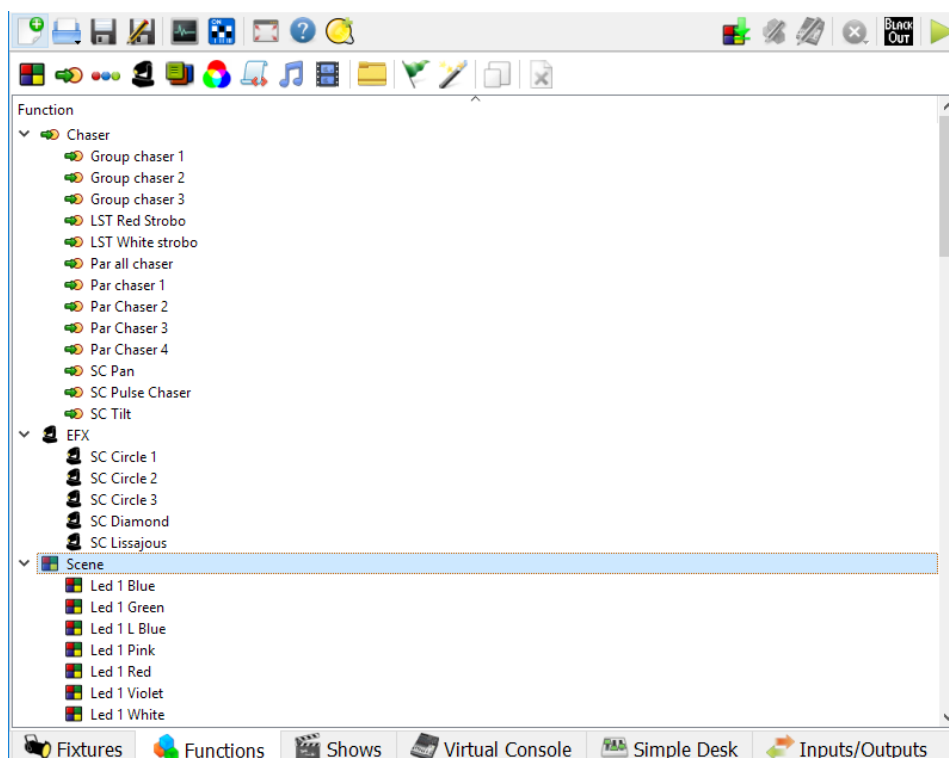


Slika 12 Zavihek Fixtures (vir: Avtor naloge).

V tem zavihku vnašamo luči in ustvarjamo skupine luči s katerimi lahko iz enega fizičnega elementa (potenciometra) nadzorujemo več luči . Izbiramo lahko med kar 1080 že v naprej ustvarjenimi luči glede na proizvajalca. Za izbiro določene luč kliknemo na »+« in se nam odpre okno z imeni različnih proizvajalcev. Med njimi si

izberemo tistega, ki ga imamo sami. Nato nam nudi izbiro začetka kanala oziroma od kje se kanali za določeno luč začnejo . Za novejši led luči po navadi potrebujemo več kanalov saj nadzorujemo več različnih parametrov (barva, svetlost, filter, hitrost. utripanja, itd) .

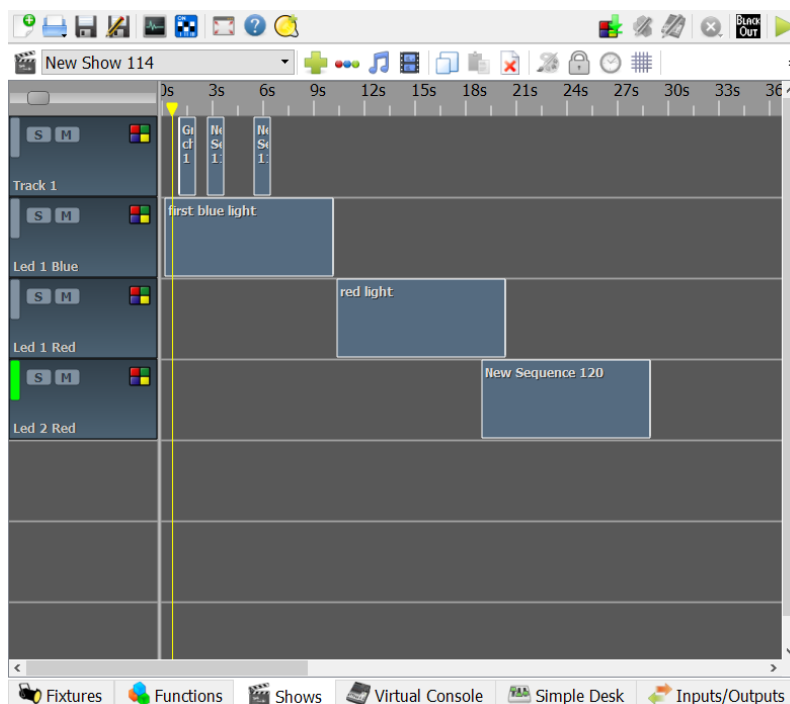
6.9.2 Zavihek Functions



Slika 13 Zavihek Functions (vir: Avtor naloge).

V zavihku Functions lahko ustvarimo scene. Scene nam omogočajo, da lahko vklopimo več luči v isti barvi istočasno kar pride zelo prav pri velikem številu luči. S to možnostjo bomo lahko kasneje upravljali kot da je en sam parameter. Sceno ustvarimo tako, da kliknemo na okno scene in v njej izberemo luči ali skupine, ki smo jih predhodno vnesli v program in nastavimo kateri parameter želimo nadzorovati pri določeni luči. Naslednja možnost je izdelava »Chaser« kar omogoča povezovanje scen v razne efekte, ki se izvajajo v določenem vrstnem redu. Na primer združimo vžig modrih luči in rdečih luči in tako izkoristimo chaser. Sledi možnost kontroliranja premikajočih luči (moving head). Z to funkcijo lahko nastavljamo kot luči smer luči, vključimo lahko vrtenje v točno določenem času za več luči hkrati.

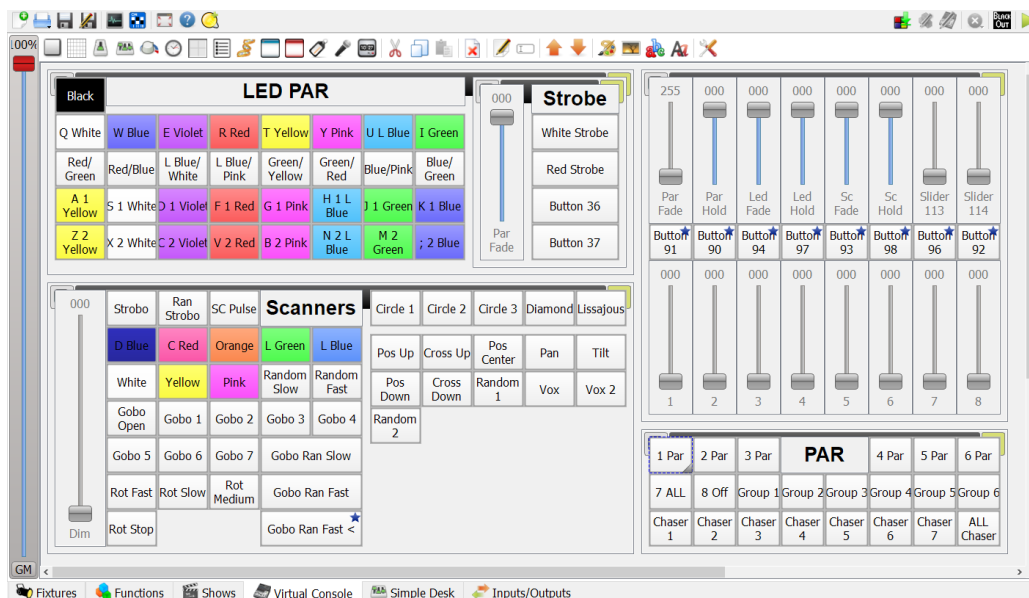
6.9.3 Zavihek Shows



Slika 14 Zavihek Shows (vir: Avtor naloge).

V zavihku shows izdelujemo časovnico določenega dogodka. Izbiramo lahko med vsemi efekti, ki smo ji predhodno ustvarili v zavihku functions. Tako lahko zelo natančno časovno vklapljammo in izklapljammo efekte. Dodamo lahko tudi glasbeno podlago, ki se predavaj isto kot se prižigajo luči. Program nudi tudi možnost priključitve projektorja, kar omogoča predvajanje videja, ki ga lahko enako kot ostale efekte vključimo v časovnico. V enem projektu lahko ustvarimo več različnih časovnic, ki jih lahko izbiramo z pomočjo za to namenjenega okna. Časovnice lahko tudi shranimo in jih lahko uporabimo na drugem računalniku.

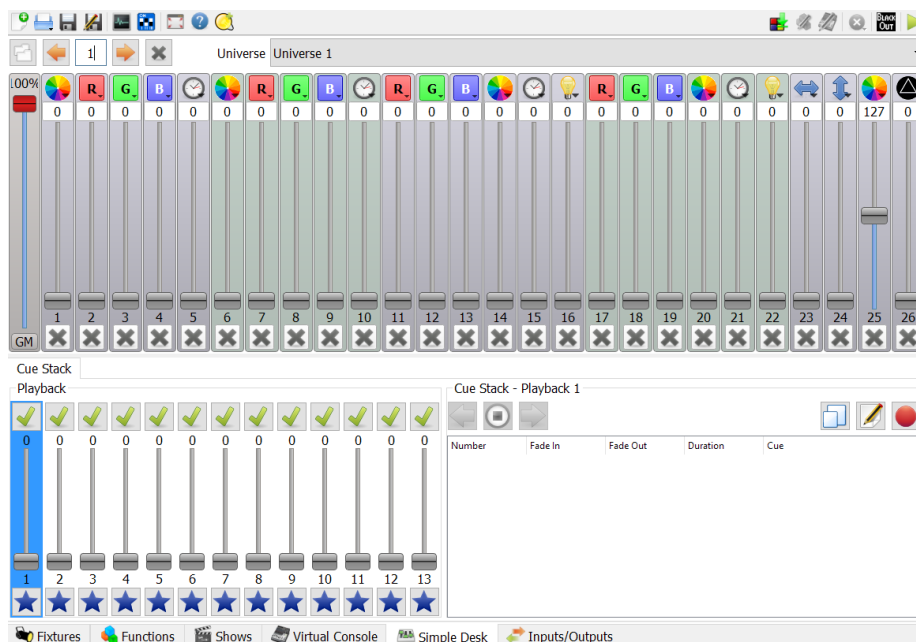
6.9.4 Zavihek Virtual Console



Slika 15 Zavihek Virtual Console (vir:Avtor naloge).

V zavihku ustvarjamo virtualno konzolo, ki povezuje razne efekte in programe v celoto. Ta zavihek omogoča tudi povezavo določene tipke z midi napravo oziroma z fizičnim elementom. V tem zavihku lahko združimo efekte in jih razdelimo na razna podokna. Uporabimo lahko navidezne potenciometre in gumbe. Te gumbe pa lahko povežemo z MIDI napravo. Vse kar naredimo na tem okno je vidno tudi na spletni strani, ki jo program avtomatsko ustvari. Na tej strani lahko prilagajamo barve gumbov za lažjo prepoznavo. Imamo tudi možnost izbire ozadja in ostalih parametrov povezanih z izgledom programa.

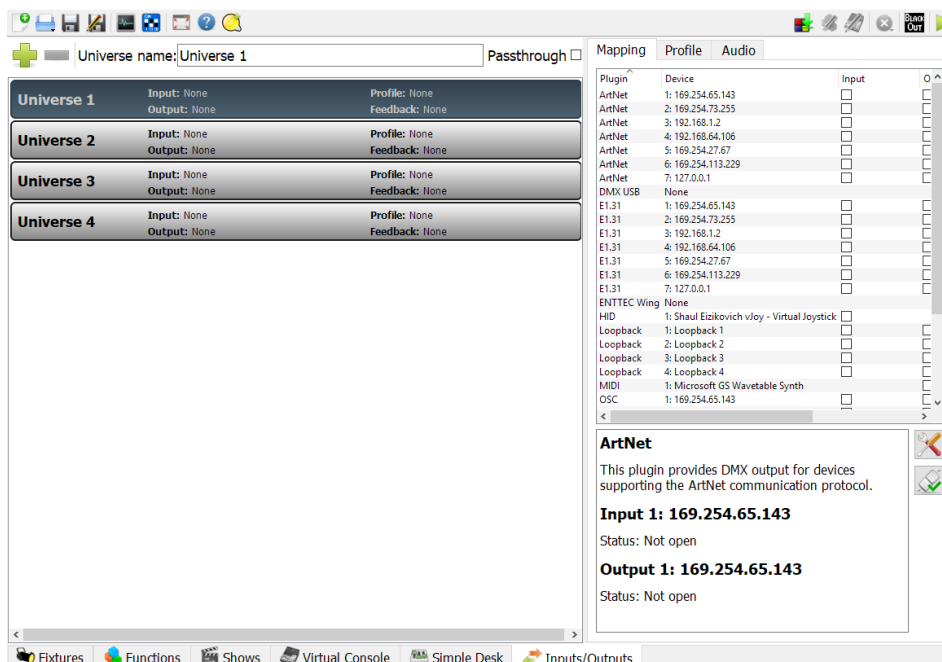
6.9.5 Zavihek Simple Desk



Slika 16 Zavihek Simple Desk (vir: Avtor naloge).

V zavihku simple desk lahko kontroliramo vsak dmx kanal posebej. Program nam glede na izbrane luči sam izbere na katerem kanalu imamo določen parameter. Tako lahko preizkušamo določene parametre in jih nato programiramo v tako imenovane playbacke. S playbacki lahko kontroliramo te parametre z enim potenciometrom. Ta playback pa lahko uporabimo še v drugih zavihkih.

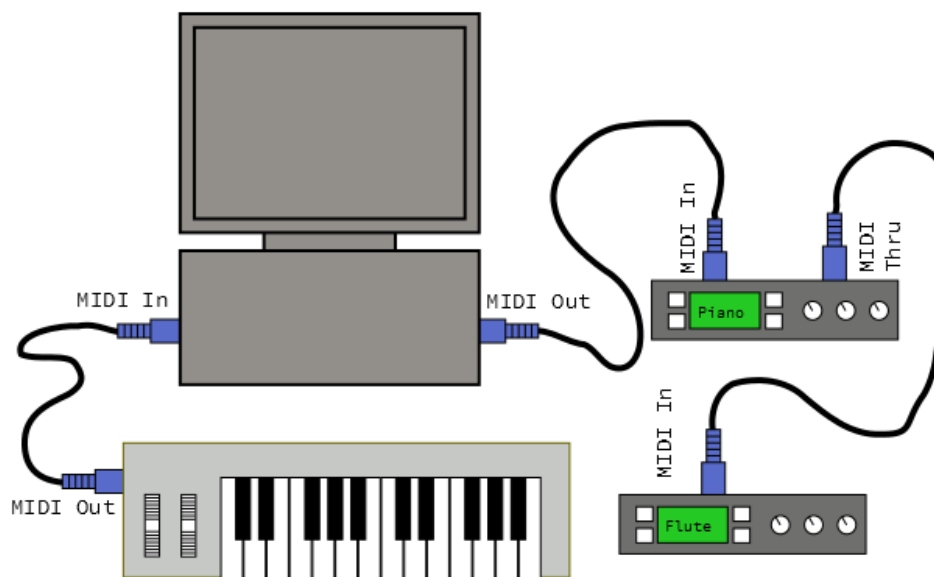
6.9.6 Zavihek Input/Output



Slika 17 Zavihek Input/Output (vir: Avtor naloge).

Zadnji med zavihki je zavihek zavihek Input/Output: Ta zavihek nam omogoča dodajanje MIDI, DMX, OSC, HID naprav. Ko izberemo določeno napravo lahko ustvarimo profil. Sam profil se razlikuje glede na vrsto naprave. Najbolj je pomemben je profil za MIDI napravo, ki omogoča dodajanje posameznih tipk in potenciometrov in encorderjev ročno. Lahko pa zaženemo orodje, ki ob premiku določenega gumba samodejno doda in obenem izpiše vse parametre. Proces izbire določene naprave je dokaj enostaven, saj izbiramo med seznamom na desni strani tega zavihka in enostavno obkljukamo določeno možnost (vhoda ali izhod), ki jo želimo koristiti za to napravo. Ustvarimo lahko več enakih naprav, ki imajo svoj naslov.

6.10 MIDI



Slika 18 Primer uporabe MIDI standarda (vir: <https://cdn.sparkfun.com>) .

MIDI (Musical Instrument Digital Interface) je protokol, ki se uporablja predvsem v glasbeni industriji za povezavo inštrumentov, ki niso neposredno povezane z glasbo. Razvil se je zaradi potrebe po digitalizaciji glasbene opreme. Tako je leta 1985 prišel v uporabo prvi MIDI standard čigar namen je bil standardizirati komunikacijo med različnimi elektronskimi napravami. Kljub temu, da kratica vsebuje »Musical Instrument« pa se standard uporablja tudi na drugih področjih. Nudi nam način komunikacije z napravami, kot so sekvenčni krmilniki, krmilniki za odrsko osvetljavo klaviature, itd. .

Sama MIDI sporočila se delijo na dva tipa .

6.10.1 Sporočila kanalu (channel messages)

Kot že samo ime pove, se ta sporočila nanašajo samo na kanal. Pri MIDI krmilnikih uporabljamo kanale zato, da lahko ločimo različne fizične elemente naprave in da ne pride do kontroliranja istega parametra pri več elementih. To možnost pa lahko izkoristimo v obratni smeri, tako da z enim fizičnim elementom kontroliramo več parametrov v programski opre. Ta način bom uporabil tudi v moji raziskovalni nalogi, saj lahko kontroliramo več luči hkrati.

6.10.2 Sistemska sporočila

Ta sporočila nimajo podatka na kateri kanal se nanašajo, saj se nanašajo na vse kanale. Ta sporočila se delijo na :

- splošna
- realno-časovna
- ekskluzivna

6.11 Dmx

DMX je komunikacijski protokol, ki se uporablja pri upravljanju luči in ostalih odrskih sistemov na daljavo. Kratica DMX izhaja iz angl. Digital Multiplex , kar je postopek pošiljanja podatkov v zaporedju preko ene povezave. Zasnovan je za komunikacijo med več odrskimi napravami ne glede na proizvajalca. To je bil prelomni trenutek za odrsko razsvetljavo , saj protokol omogoča kontroliranje vsega iz samo enega vira, kar omogoča lažjo in hitrejšo nastavitvev luči in ostalih odrskih elementov.

6.11.1 Delovanje DMX sistema

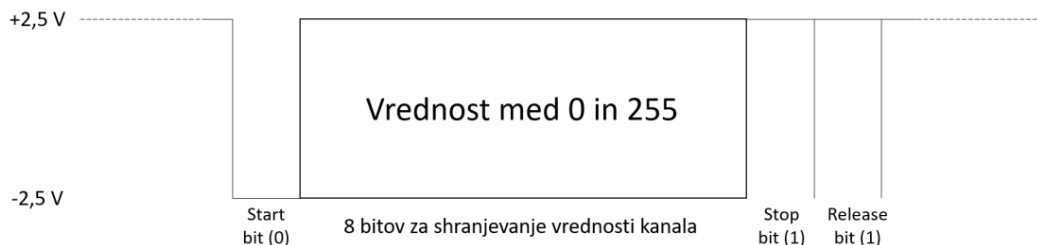
DMX je protokol ,ki se v osnovi uporablja za upravljanje jakosti odrskih luči , krmiljenje premikajočih se luči (angl. moving head) , dimnih naprav, stroboskopov ipd. Protokol je sestavljen iz enega 8-bitnega signala za vsako luči. Za vsako dmx napravo lahko nastavimo jakost svetlobe (0–100 %) oziroma nastavimo eno od 256 vrednosti.

Protokol v DMX-vodilu je podoben navadni serijski komunikaciji, kakršno najdemo pri standardu RS-232, ki deluje z 8-podatkovnimi biti in 1 stop bitom. DMX je serijski prenos podatkov preko standarda RS-485 s hitrostjo 250 kbit/. Podatkovno okno je sestavljeno iz 8 podatkovnih bitov, 1 start bita in 2 stop bitov.

Prenos DMX-podatkov vsebuje tudi "reset" stanje, kar označuje začetek novega paketa, okno z začetkom kode in do 512 podatkovnih oken. Začetek paketa se označi z bitom Break, ki mu sledi oznaka MAB (angl. Mark After Break). Sledi prvo okno paketa, imenovano "start code" okno, ki pove, kakšen tip paketa se pošilja, in nato do 512 podatkovnih oken.

6.11.2 Sestava DMX paketa

Paket protokola DMX sestavljajo manjše enote, ki se imenujejo DMX-okno. DMX-okno je iz 11 bitov.



Slika 19 Sestava dmx-okna (vir: playground.arduino.cc).

DMX-okno sestavljajo naslednji biti:

- Start bit, ki ima logično vrednost 0 in na oznani začetek okna.
- Zaporedje osmih podatkovnih bitov, ki vsebujejo vrednost med 0 in 255 (256 vrednosti) kontroliranje naprav, svetlosti itd.
- Stop bit in Release bit, ki imata logično vrednost 1 konča dmx-okno .

DMX-signal je sestavljen iz zaporedja, imenovanega paket (angl. packet). DMX-paket podatke nenehno posodablja, kar omogoča zanesljivost sistema. Od pošiljatelja podatkov je odvisno, koliko od 512 vrednosti se pošlje. Če je paket manjši oziroma vsebuje manj podatkov je cikel pošiljanja hitrejši.

DMX-paket je sestavljen iz več različnih delov:

- Break, ki je oznaka za začetek paketa, ima logično vrednost 0 in je velika minimalno 22 bitov.
- MAB (angl. Mark After Break) označuje konec oznake Break in začetek podatkovnih paketov. Ima logično vrednost 1 in velikost 2 bita.
- MTBP (angl. Mark Time Between Packets) označuje čas med posameznimi paketi in ima logično vrednost 1.
- MTBF (angl. Mark Time Between Frames), označuje čas med posameznimi okni in ima logično vrednost 1. Njegova časovna dolžina je med 0 in 1 s, manjša ko je časovna dolžina, bolje je.

- "Start code" oz. okno z začetno kodo, ki se pošlje v prvem oknu paketa, pove sprejemniku, kakšen tip paketa se bo poslal. Uporablja se za spreminjanje pomena podatkovnih oken, ki mu sledijo. Privzeta vrednost tega okna je 0, preostale vrednosti se redko uporabljajo

7. Programska koda

7.1 Midi mešalna miza

Pred začetkom pisanja programa moramo najprej uvoziti knjižnice, ki jih potrebujemo za komuniciranje z MIDI protokolom, branja tipk in za glajenje analogne vrednosti, na vhodu.

```
#include <ResponsiveAnalogRead.h> // Knjižnica za glajenje analognega vhoda
#include <Bounce.h> //Knjižnica za uporabo stikal
#include <usbMIDI.h> Knjižnica za MIDI komunikacijo
```

Nato moramo definirati kontrolne pine, za multiplexer in število analognih in digitalnih vhodov. V program sem vključil tudi uporabo zbirke(angl. *Array*), ki omogoča lažjo dodajanje več elementov z istim imenom in različno številko oziroma indeksom.

```
// Kontrolni pini s0,s1,s2,s3
int SELECTOR_PINS[] = {2,3,4,5};

const int AN = 4; // število analognih vhodov
const int DN = 4; // število digitalnih vhodov
const int ANALOG_PINS[AN] = {A0,A1,A2,A3}; // analogni vhodi (mux)
const int DIGITAL_PINS[DN] = {6,7,8,9}; // digitalni vhodi mux

const int MAX_PINS = 16; // maksimalno število pinov na multiplekserju
const int ANALOG_MUX_PINS[AN] = {16,16,16,16}; // št. Pinov na multiplekser
const int DIGITAL_MUX_PINS[DN] = {16,16,16,16}; // št. Pinov na
multiplekseru
```

Sledi nastavljanje potrebnih parametrov iz knjižnice za glajenje analognih vhodov in knjižnice za uporabo stikal.

```
//Nastavitev branja analognih pinov
ResponsiveAnalogRead *analog[AN][MAX_PINS];
byte analog_data[AN][MAX_PINS];
byte analog_data_prev[AN][MAX_PINS];
// Nastavljanje določenih pinov multiplexerja za določeno funkcijo
```

```
Bounce *digital[DN][MAX_PINS];
const int BOUNCE_TIME = 5; // zakasnitveni čas 5 ms
const int ON_VELOCITY = 127; // Vrednost midi sporočila »note«
```

Po končanem definiranju spremenljivk sledi uporaba funkcije *void setup()*.

Funkcija *void setup()* se izvede samo enkrat ter se ne ponavlja. Če mikrokrmilnik izključimo iz napetostnega napajanja in ga ponovno vključimo, se bo zanka *void setup()* ponovno izvedla. Enako se zgodi, če pritisnemo na tipko reset, nameščeno na teensy mikrokrmilniku. V tej funkciji program pokliče dve ločeni funkciji za nastavitve analognih vhodov in nastavitve digitalnih vhodov.

```
void setup() {
    Serial.begin(9600);
    setupAnalog(); // Definiranje vhodnih in izhodnih analognih pinov
    setupDigital(); // Definiranje vhodnih in izhodnih digitalnih pinov
}

void setupAnalog() {
    for (int i = 0; i < AN; i++) {
        for (int j = 0; j < ANALOG_MUX_PINS[i]; j++) {
            analog[i][j] = new ResponsiveAnalogRead(ANALOG_PINS[i], true);
        }
    }
}

void setupDigital() {
    for (int i = 0; i < DN; i++) {
        for (int j = 0; j < DIGITAL_MUX_PINS[i]; j++) {
            digital[i][j] = new Bounce(DIGITAL_PINS[i], true); // initialize
        }
    }
}
```

Če želimo, da se bo neka funkcija izvajala v neskončnost uporabimo glavno funkcijo *void loop()*. Funkcija *void loop()* nam omogoča vnašanje že ustvarjenih funkcij, ki jih lahko pokličemo, ko jih potrebujemo in s tem povečamo preglednost programa. Tako sem storil tudi jaz. Ker imam več multiplekserjev sem moral narediti zanko, ki omogoča branje vsega multiplekserja. Nato pa sem v program vstavil še funkcije, ki skrbijo za branje analognih in digitalnih vhodov.

```
void loop() {
    for (int j = 0; j < MAX_PINS; j++) { // loop over multiplexer
```

```

    setSelectorPins(j);
    readAnalog(j);
    readDigitals(j);
}

```

Če želimo brati pine multiplekserja moramo imeti 4 kontrolne pine. Zato sem za te pine ustavril funkcijo, ki jih vklaplja tako, da prebere vseh 16 možnosti .

```

void setSelectorPins(int j) {
    byte s0 = (j & B0001) ? HIGH : LOW;
    byte s1 = (j & B0010) ? HIGH : LOW;
    byte s2 = (j & B0100) ? HIGH : LOW;
    byte s3 = (j & B1000) ? HIGH : LOW;
    digitalWrite(SELECTOR_PINS[0], s0);
    digitalWrite(SELECTOR_PINS[1], s1);
    digitalWrite(SELECTOR_PINS[2], s2);
    digitalWrite(SELECTOR_PINS[3], s3);

    // »zamik« 100us, da se stabilizira napetost na vhodu
    delayMicroseconds(100);
}

```

Najbolj pomemben del celotnega programa je bilo pravilno branje analogni vhodov. To sem storil z to z to zanko.

```

void readAnalog(int j) {
    for (int i = 0; i < AN; i++) { // zanka za analogne vhode
        if (j < ANALOG_MUX_PINS[i]) { // branje omogočenih pinov
            analog[i][j]->update();
            if (analog[i][j]->hasChanged()) {
                analog_data[i][j] = analog[i][j]->getValue() >> 3;
                if (analog_data[i][j] != analog_data_prev[i][j]) {
                    analog_data_prev[i][j] = analog_data[i][j];
                    usbMIDI.sendControlChange(16 + j, analog_data[i][j], i + 1);
                }
            }
        }
    }
}

```

7.2 Namestitev programa QLC+

Pred začetkom nalaganja programa moramo najprej iz spleta prenesti najnovejšo verzijo operacijskega sistema rasbian in ga namestiti na SD kartico. Ko smo to opravili na računalnik priključimo napajanje in žično internetno povezavo. Nato pogledamo v nastavitvah ruterja ali program za odkrivanje IP naslovov naprav (npr. Andvance IP scanner). katri IP naslov ima raspberi pi . Po pridobljenem IP naslovu se lahko povežemo s raspberripijem preko SSH povezave. Jaz sem za to uporabil program Putty. Nato vpišemo uporabniško ime in geslo in lahko začnemo s nalaganjem programa z naslednjimi ukazi .

```
Sudo apt-get update
```

```
sudo apt-get install g++ make git build-essential qt5-default qtbase5-dev  
qtscriptdevqtmultimedia5-dev libqt5multimedia5-plugins qttools5-dev-tools fakeroot  
debhelper devscripts pkg-config libxml2-utils
```

```
sudo apt-get install libasound2-dev libusb-dev libftdi-dev libudev-dev libmad0-dev  
libsndfile1-dev libfftw3-dev
```

Z temi ukazi smo na naložili program Qt5 , ki ga potrebujemo za delovanje programa qlc+. Sedaj pa lahko iz spleta naložimo datoteke programa QLC+. To lahko storimo na dva načina, eden je da sami ročno naložimo potrebne datoteke lahko pa v terminal izveršimo ta ukaz.

```
git clone git://github.com/mcallegari/qlcplus.git
```

Po opravljenemu nalaganju moramo najprej prevesti. To naredimo s temi ukazi.

```
cd qlcplus  
qmake  
make
```

Sedaj lahko začnemo z namestitvijo samega programa. Po končani namestitvi zaženemo program.

```
sudo make install  
qlcplus  
qlcplus-fixtureeditor
```

8. Izdelava

8.1 Eagle

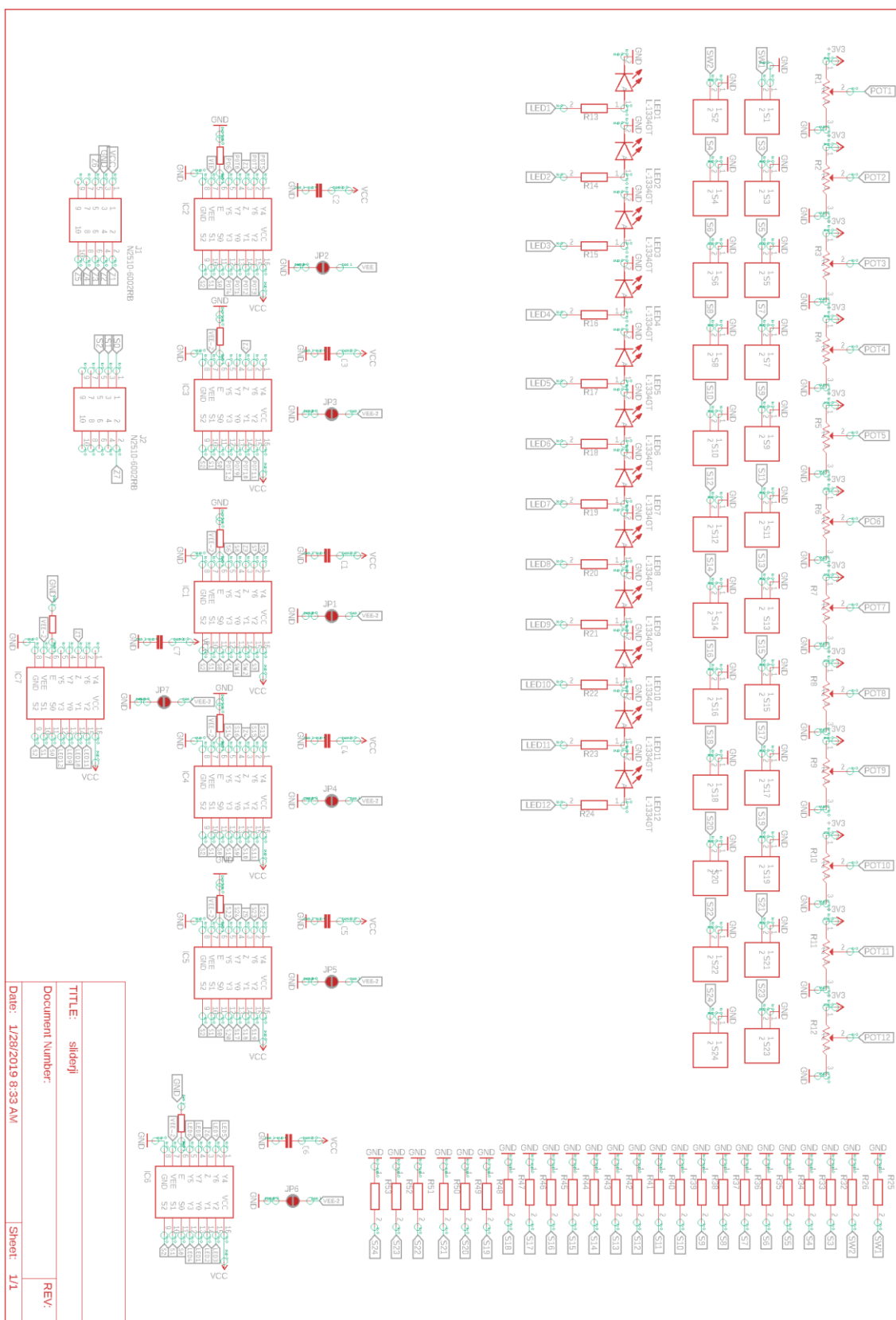
Autodesk Eagle (Easily Applicable Graphical Layout Editor) je učinkovit in zmogljiv CAD/CAM računalniški program za načrtovanje ploščic tiskanih vezij in risanje električnih načrtov. Program je namenjen za profesionalno in hobi uporabo. Ima možnost izvoza vezja v 3D obliki v program Autodesk Fusion 360, kar zelo koristi pri izdelavi ohišja za določeno napravo. V programu izdelujemo v dveh oknih. Najprej moramo narisati električno shemo vezja. To moramo narediti v oknu za izdelavo shem. Sam program ima že veliko knjižnic komponent, ki imajo svoj simbol in položaj na tiskanini, če pa med knjižnicami ni komponent, ki jih potrebujemo ustvarimo svojo knjižnico. Tako v program vnesemo komponente in jih smiselno povežemo. Ko smo opravili z risanjem sheme se lahko preselimo v okno za izdelavo tiskanine. Program nam samodejno pripravi komponente z zračnimi povezavami, mi pa jih moramo razporediti na površino ploščice in nastaviti vrednosti oziroma debeline povezav. Sedaj lahko začnemo z povezovanjem. Če želimo lahko naredimo to ročno, lahko pa uporabimo modul *autorouter*.

8.1.1 Midi kontroler

Midi kontroler vsebuje štiri module oziroma tri različna vezja. Najprej sem se lotil izdelave modula z potenciometri. Za branje vrednosti potenciometrov sem uporabil metodo delilnika napetosti, ki nam na izhodu poda določeno napetost oziroma sredno napetost med uporoma R1 in R2 v potenciometru. Z pomočjo napetosti lahko v programu izračunamo vrednosti od 1 do 255. Pri izračunu vrednosti moramo v programu upoštevati to, da mikrokrmilnik pretvarja analogno vrednost z 8 bitnim A/D pretvornikom. Če želimo imeti območje med 0 in 255 moram izmerjeno vrednost deliti z 4.

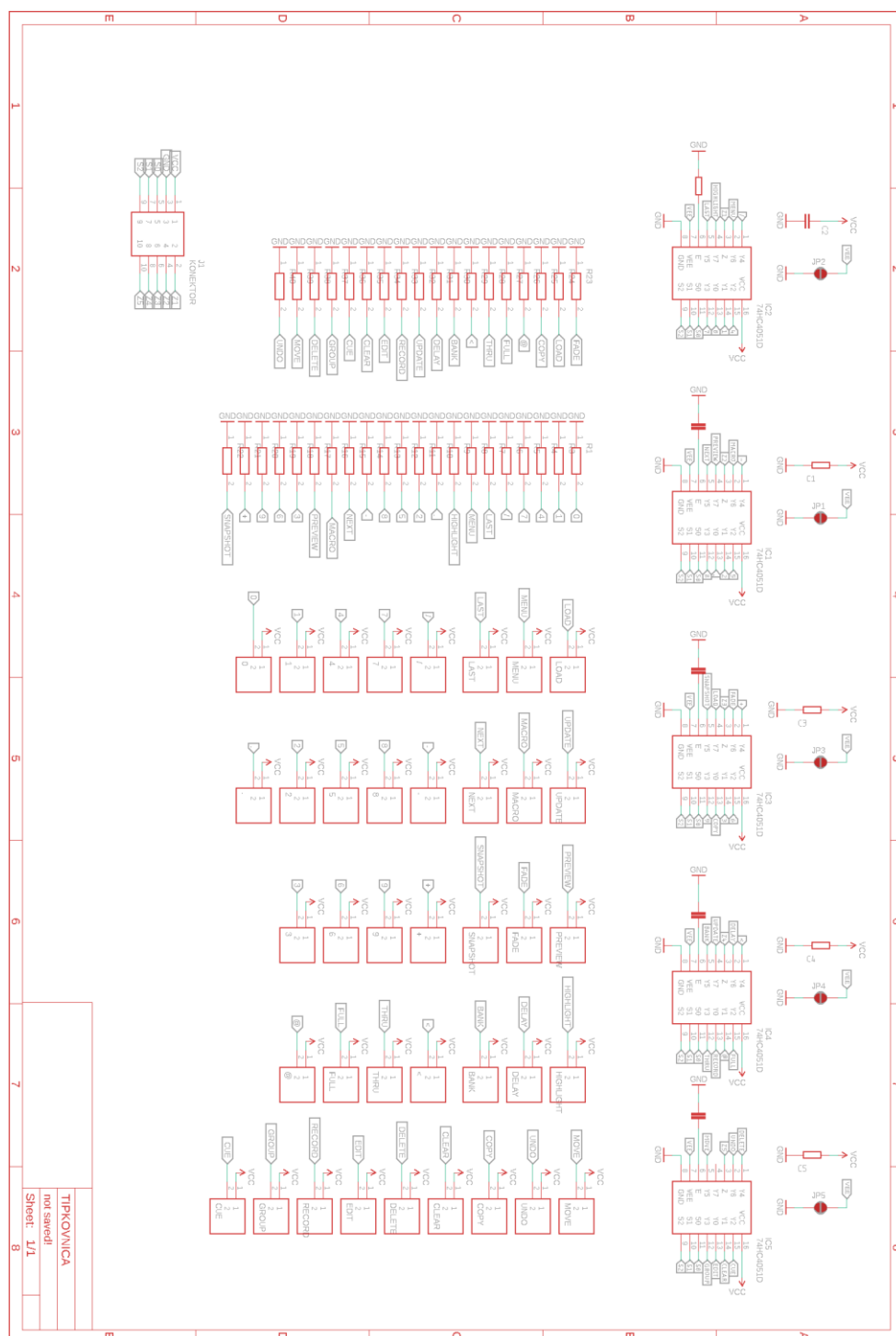
$$položaj(0 - 255) = \frac{Izmerjen\ položaj}{4}$$

Zaradi potrebe po tipkah sem na ta modul dodal še stikala. Vsako stikalo je vezano z pull up uporom. Nato sem dodal še multiplexerje, ki so število potrebnih povezav zelo zmanjšali.

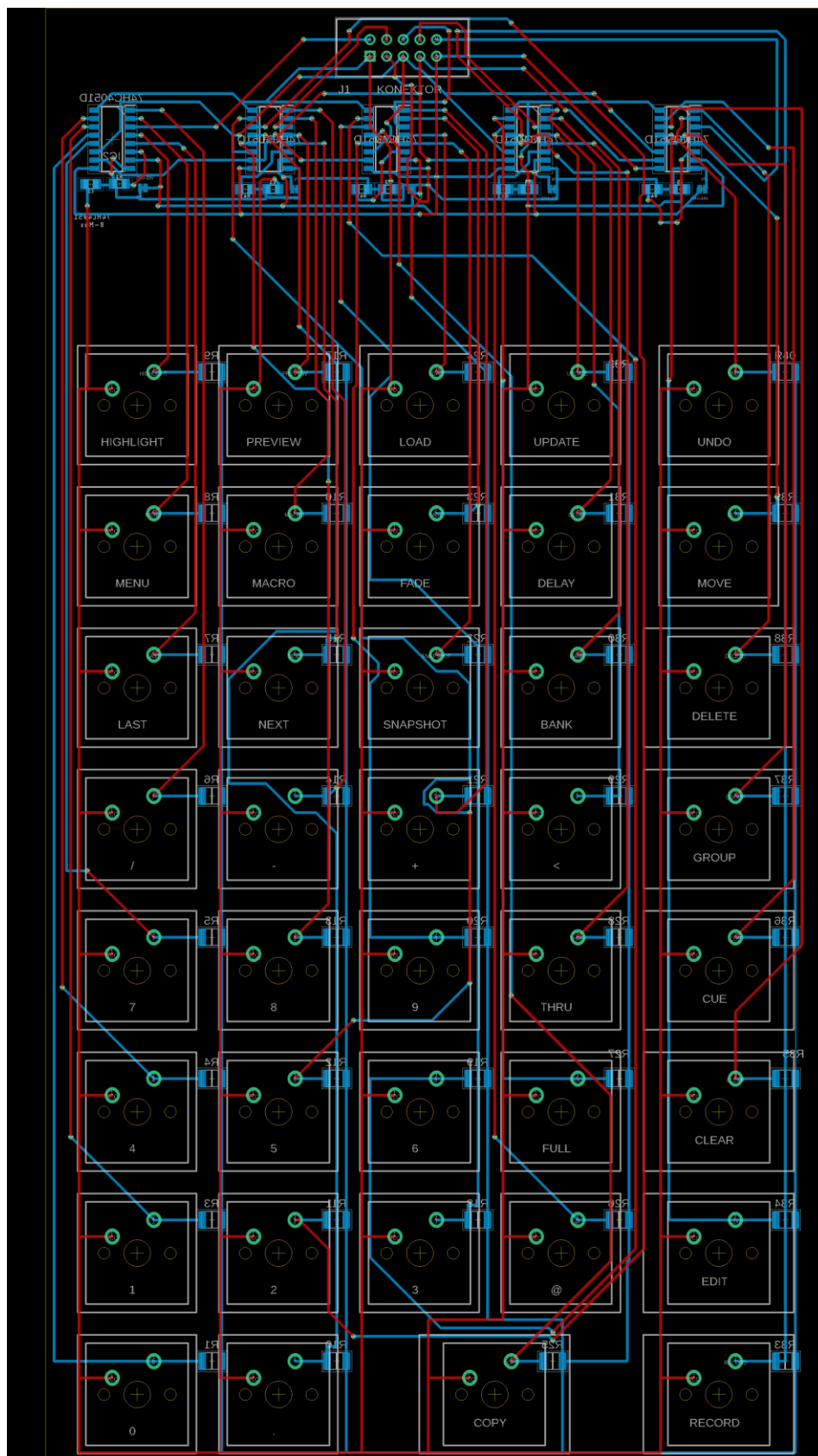


Slika 20 Shema 1 modula (vir:Avtor naloge).

Sledila je izdelava modula tipkovnice za programiranje. Tipkovnica vsebuje numerično tipkovnico in funkcijske tipke (edit, delete, group, cue). Tipke so tako kot pri prejšnji plošči povezane v z pull up uporom in povezane na multiplekser. Vsak multiplekser pa ima še upor in kondenzator, ki skrbita za izločanje moten.



Slika 22 Shema modula 2 (vir: Avtor naloge).

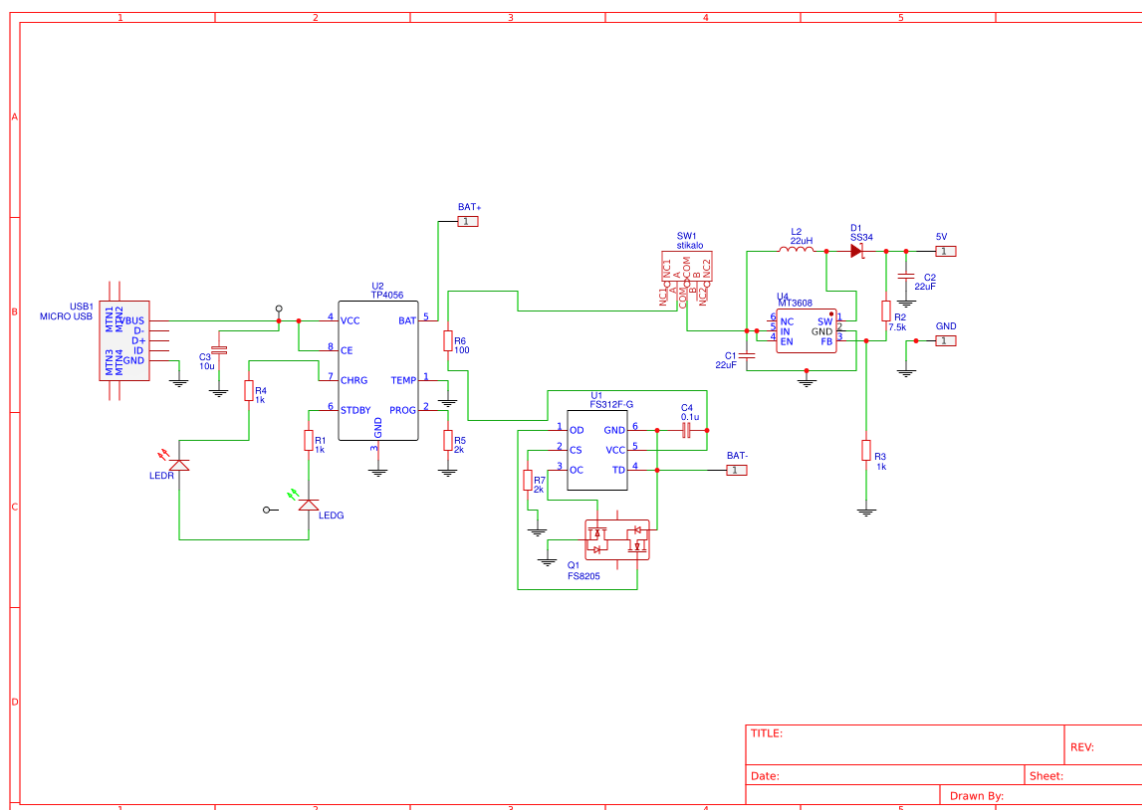


Slika 23 Tiskanina modul 2 (vir: Avtor naloge).

Da bi vse skupaj povezal sem ustvaril ploščico, ki poveže vse module vsebuje sam mikrokrmilnik teensy.

Predhodno pa sem vsa vezja preizkusil na preizkusni ploščici v manjši obliki.

8.1.2 Izdelava polnilnega vezja



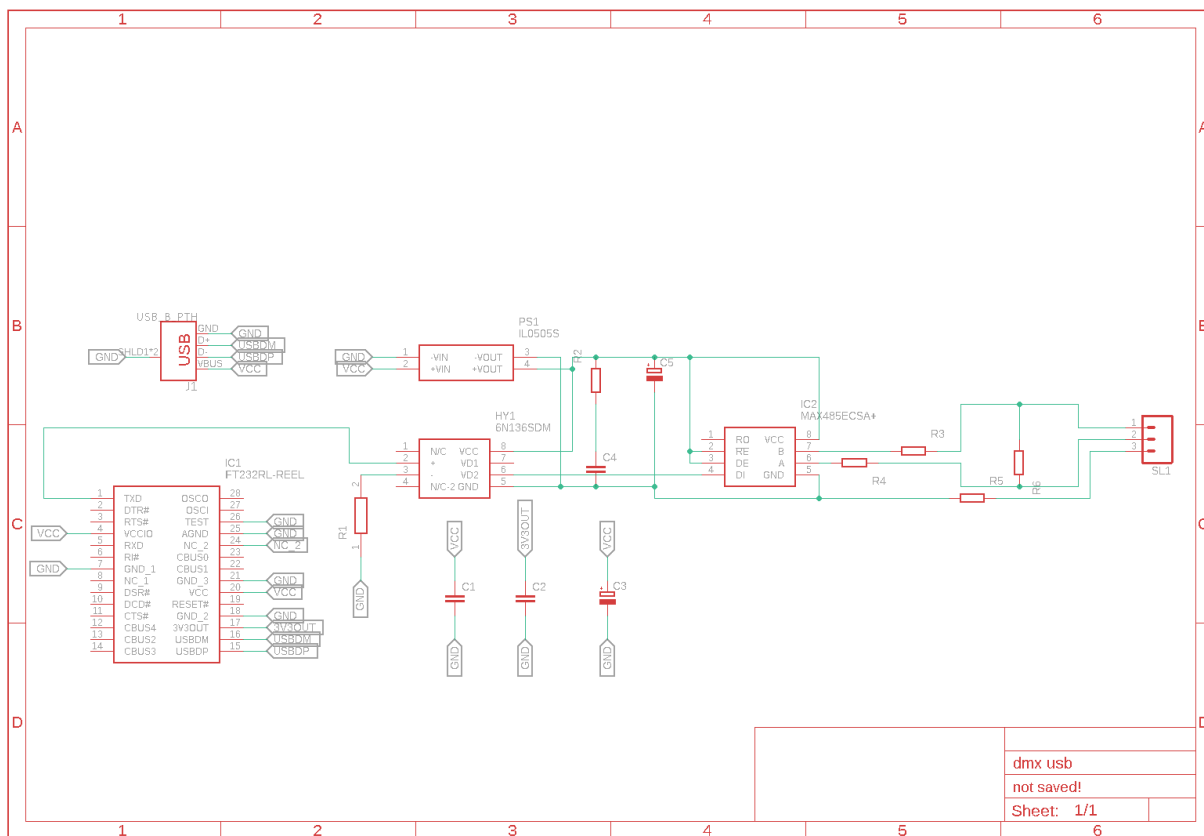
Slika 24 Shema polnilnega vezja (vir: Avtor naloge).

Sestavlil sem vezje, ki sem ga že predhodno opisal.

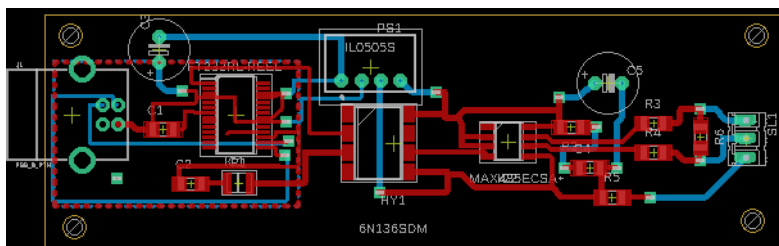
8.1.3 Dmx kontroler

Za kontroliranje luči sem ustvaril dva USB dmx pretvornika. Oba pretvornika imata enak serijski čip FIDI FT232RL, ki služi kot serijski vmesnik med računalnikom DMX napravami. Vendar sam čip ne more komunicirati z Dmx napravami saj je moč na izhodu majhna. Za to potrebujemo ojačevalec signala. Za sam ojačevalec signala sem uporabil MAX485, ki omogoča prenos podatkov na daljše razdalje (dolžina kabla do 1200m). Prva plačica vsebuje do sedaj našteje komponente in ostale potrebne elemente za delovanje le teh. Druga ploščica se razlikuje po tem, da sta povezava

med računalnikom in dmx linijo galvansko ločena. Vezje tako vsebuje dva dodatna elementa. Za ločeno napajalno napetost skrbi DC/DC pretvornik, ki na izhodu onemogoča povratno napetost na napajalno usb linijo. Za povezavo FIDI serijskega pretvornika in ojačevalnega čipa pa skrbi optosklopnik 6N136.



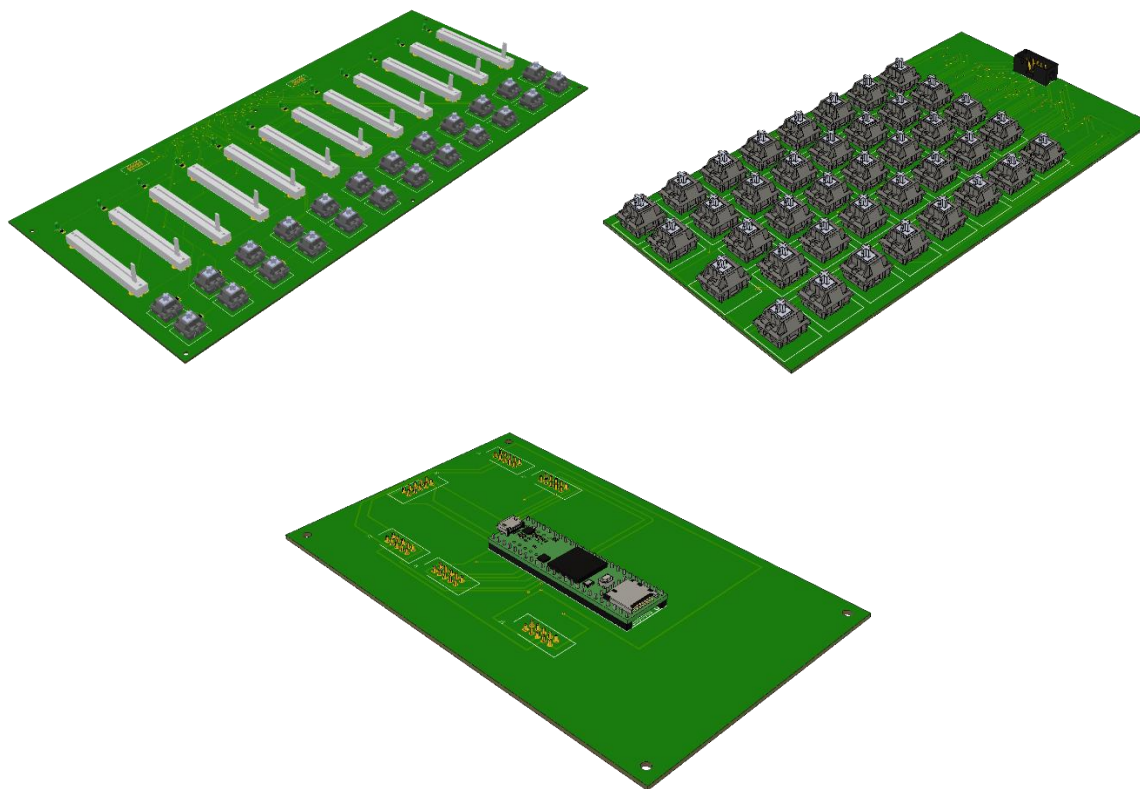
Slika 25 Shema usb DMX pretvornik (vir:Avtor naloge).



Slika 26 Tiskanina USB DMX pretvornik (vir:Avtor naloge).

8.2 3D načrtovanje

8.2.1 Izdelovanje vezji v 3d obliki



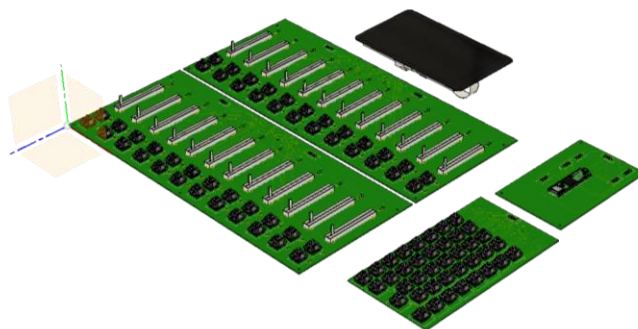
Slika 27 Vezja v 3d obliki (vir: Avtor naloge).

Vsa vezja, ki sem jih narisal sem lahko izvozil v 3d obliki. Predhodno pa sem moral v k knjižnicam komponent dodati 3d modele za vsako komponento. Večino 3d modelov sem našel na spletu če pa kakšnega modela ni bilo pa sem ga narisal s podatki iz podatkovnega lista. Tako sem lahko po končanem risanju vezji enostavno izvozil 3d model vsakega vezja v program fusion 360.

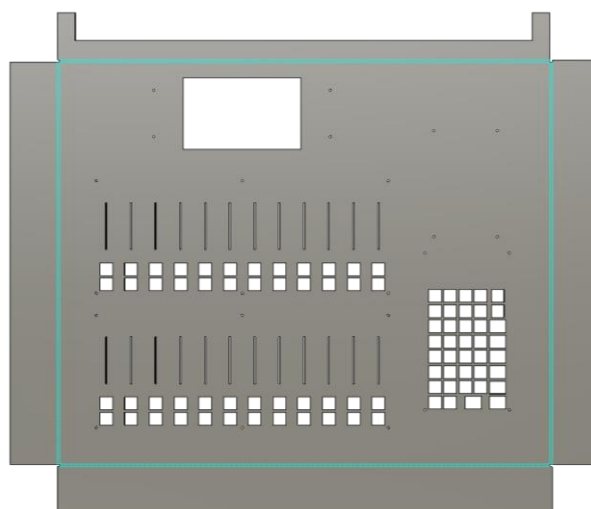
8.2.2 Risanje ohišja v 3d

Ker bom ohišje z kovine in ker ga bom dal v izdelavo podjetju, sem moral celotno ohišje narisati v 3D obliki. Za risanje ohišja sem uporabil program fusion 360. Pri tem sem si pomagal že s prej narisanimi tiskanimi vezji v 3D obliki. Tako sem lahko dobil zelo natančno predstavbo kakšna bo prostorska razporeditev vezij in zelo natančne mere za

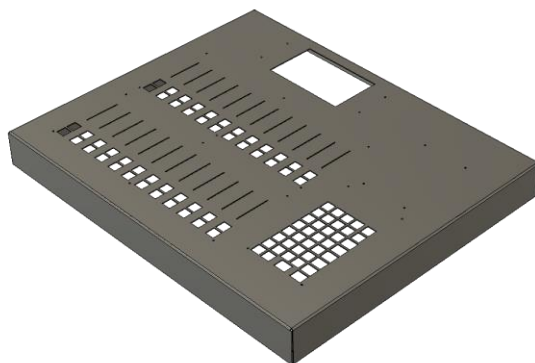
samo izdelavo. Ker sem imel vsa vezja že narisana sem jih lahko razporedil in po njih narisal ohišje, ki se povsem prilagaja.



Slika 28 Razporeditev vezij (vir: Avtor naloge).

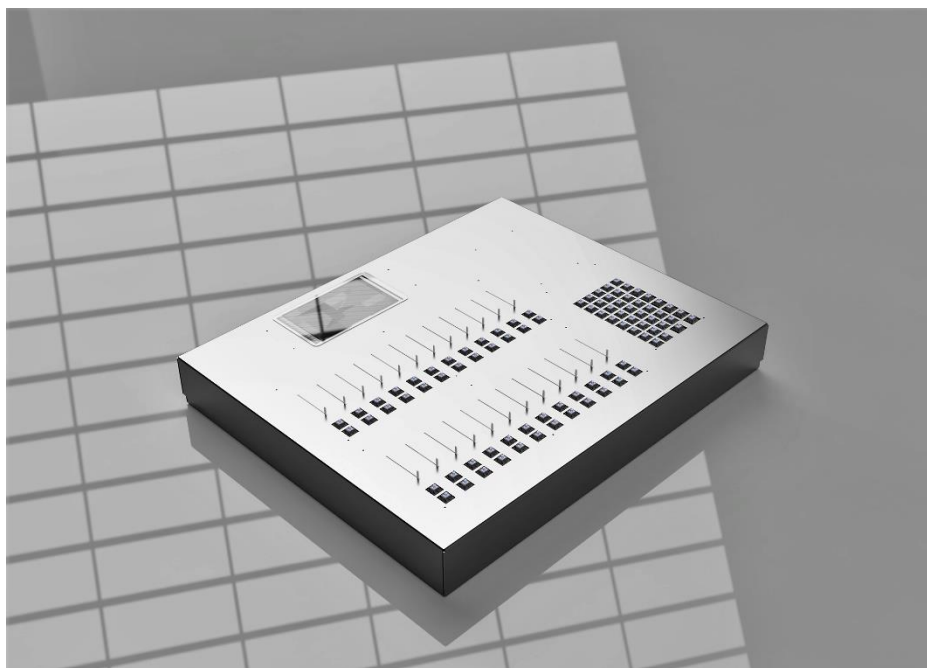


Slika 29 2d model ohišja (vir: Avtor naloge).



Slika 30 3D model ohišja (vir:Avtor naloge).

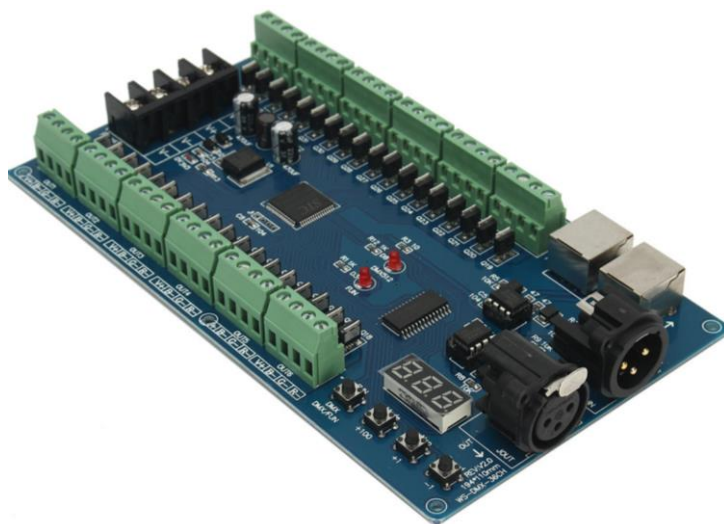
Ohišje sem risal z pomočjo orodja za izdelavo kovinskih plošč, ki omogoča risanje v 3d in 2d obliki. Po končani izdelavi sem moral še izdelati postavitev tipk za tipkovnico, ki sem jo dal v tisk.



Slika 31 Model z vsemi komponentami (vir:Avtor naloge).

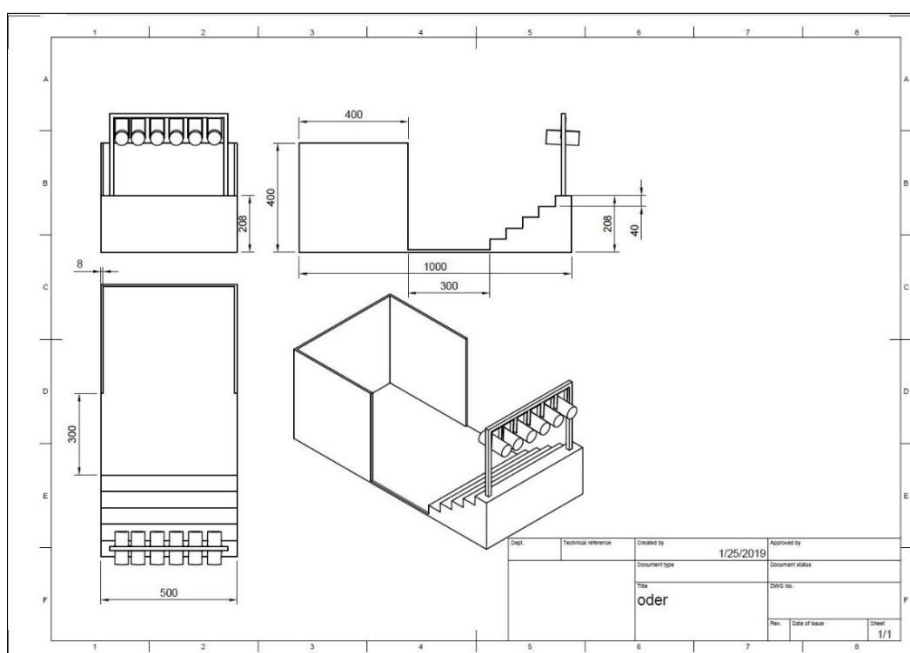
8.3 Maketa

Za prikaz samega delovanja mešalne mize pa sem se odločil za izdelavo makete. Maketa je iz kartona in ima 6 RGB led luči, ki se kontrolirajo z pomočjo posebnega 12 kanalnega DMX sprejemnika .

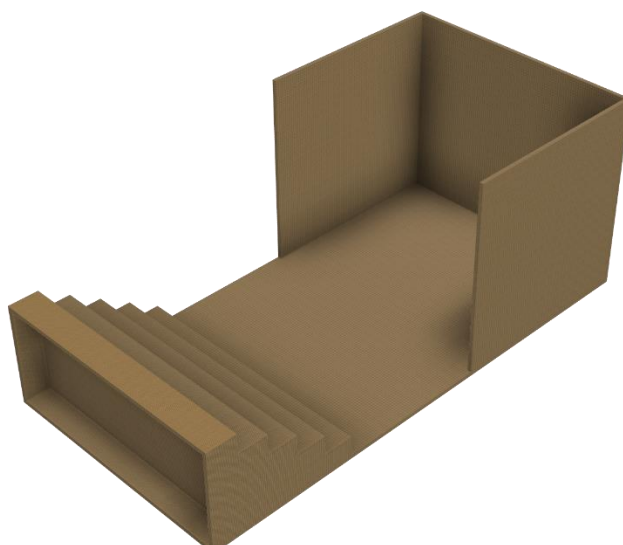


Slika 32 DMX sprejemnik (vir: www.banggood.com).

V domačem gledališču sem dobil tudi miniaturne reflektorje, ki so bili ravno pravšni za demonstracijo na maketi.



Slika 33 Načrt makete (vir: Avtor naloge).



Slika 34 Model makete (vir: Avtor naloge).

9. Zaključek

Za izdelavo moje raziskovalne naloge sem porabil zelo veliko časa. Poleg pisanja programskih kod, sem povezoval posamezne elemente v celoto, preizkušal njihovo delovanje z meritvami in testiranjem ter po potrebi spreminjal določene elemente. Ni mi žal časa, ki sem ga porabil, saj mi bo pridobljeno znanje koristilo pri mojem bodočem delu. V raziskovalni nalogi sem spoznal delovanje različnih sistemov. Uresničil sem vse cilje, ki sem si jih zadal. Izboljšal sem si znanje na različnih področjih in v uporabi programa Autodesk eagle in Fusion 360.

10. Družbena odgovornost

Z to raziskovalno nalogo sem dosegel da jo bomo lahko uporabljali v domačem lutkovnem gledališču. Z nalogo sem ustvaril cenovno zelo ugodno MIDI mešalno mizo, saj celoten strošek izdelave stane okoli 1000 eurov. To omogoča da bodo lahko mojo mešalno mizo uporabljali ljudje, ki nimajo toliko denarja ali pa ne potrebujejo tako zahtevne mize in jo bodo mogoče potrebovali v prihodnosti. Vsa vezja in programska koda pa bodo dostopna na spletu kot odprtokoden projekt.

11. Viri

Autodesk eagle . (8. januar 2019). Pridobljeno iz www.autodesk.com:
<https://www.autodesk.com/products/eagle/free-download>

Data-sheet 74HC_HCT4067. (20. januar 2019). Pridobljeno iz
<https://assets.nexperia.com/>: https://assets.nexperia.com/documents/data-sheet/74HC_HCT4067.pdf

FS312F Data-shet. (januar. 18 2019). Pridobljeno iz www.ic-fortune.com:
https://www.ic-fortune.com/upload/Download/FS312F-G-DS-10_EN.pdf

MIDI Tutorial. (10. januar 2019). Pridobljeno iz learn.sparkfun.com:
<https://learn.sparkfun.com/tutorials/midi-tutorial/all>

MT3608 Data-shet. (10. januar 2019). Pridobljeno iz <https://www.olimex.com>:
<https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf>

TP4056 Data-shet. (6. januar 2019). Pridobljeno iz dlnmh9ip6v2uc.cloudfront.net/:
<https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>

Using USB MIDI. (4. januar 2019). Pridobljeno iz www.pjrc.com:
https://www.pjrc.com/teensy/td_midi.html

PERC, Anton, 2013, Modularni MIDI krmilnik : diplomsko delo [na spletu]. Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. [Dostopano 1 februar 2019].
Pridobljeno od: http://eprints.fri.uni-lj.si/2025/1/Perc_A%2D1.pdf