

Mladi za napredek Maribora 2020
37. Srečanje

Nadgradnja MIDI DMX krmilnika

Raziskovalno področje: **ELEKTROTEHNIKA, ELEKTRONIKA**

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor: KEVIN BARON LAH, MARK VEZJAK, MATIC ŠULC

Mentor: IVANKA LESJAK

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Število točk: 156/ 170

Maribor, februar 2020

Mladi za napredek Maribora 2020
37. srečanje

Nadgradnja MIDI DMX krmilnika

Raziskovalno področje: **ELEKTROTEHNIKA, ELEKTRONIKA**

Raziskovalna naloga

PROSTOR ZA NALEPKO

Maribor, februar 2020

KAZALO VSEBINE

1. ZAHVALA	5
2. POVZETEK	6
3. HIPOTEZE	6
4. VSEBINSKI DEL	7
4.1 MIDI Krmilniki	7
4.2 Princip delovanja	7
4.3 Arduino MEGA	8
4.4 Vhodne enote	9
4.4.1 Cherry MX Blue stikala	9
4.4.2 Linearni potenciometer	10
4.4.3 Rotacijski kodirnik	11
4.4.4 IC 74HC4051	13
4.4.5 IC 74HCT595	13
4.5 Uporaba multipleksiranja za dodajanje večjega števila vhodov	14
4.6 Nalaganje strojne programske opreme na mikrokrmilnik	15
4.7 Kontroliranje RGB LED diod	16
4.8 MIDI	17
4.9 Arduino kot MIDI vmesnik	18
5. ZAKLJUČEK	19
6. DRUŽBENA ODGOVORNOST	20
7. VIRI	21

KAZALO SLIK

Slika 1: Arduino Mega (vir: Arduino Store)	8
Slika 2: Mehansko stikalo CHERRY MX Blue (vir: Avtor naloge)	9
Slika 3: Linearni potenciometer (vir: Avtor naloge)	10
Slika 4: Signali rotacijskega kodirnika (vir: Avtor naloge)	11
Slika 5: Sestava rotacijskega kodirnika (vir: circuitdigest.com)	11
Slika 6: Rotacijski kodirnik (vir: Avtor naloge)	11
Slika 7: Primer popačenega signala (vir: allaboutcircuits.com)	12
Slika 8: Odprava popačenih signalov na rotacijskem kodirniku (vir: hackday.com)	12
Slika 9: IC 74HC4051 (vir: avtor naloge)	13
Slika 10: IC 74HCT595 (vir: https://www.diodes.com/)	13
Slika 11: Multipleksiranje gumbov	14
Slika 12: Povezave na multiplexer (vir: Avtor naloge)	14
Slika 13: DFU resetiranje (vir: Arduino.cc)	15
Slika 14: Atmel FLIP (vir: Avtor naloge)	15
Slika 15: RGB LED dioda (vir: https://learn.sparkfun.com/)	16
Slika 16: matrica LED diod (vir: https://learn.sparkfun.com/)	16
Slika 17: Eliminacija "ghosting" efekta (vir: http://www.learningaboutelectronics.com/)	16
Slika 18: MIDI signal (vir: frank-buss.de)	17
Slika 19: Sestava MIDI sporočila (vir: Ramp me up, Scotty!)	17
Slika 20: Arduino kot MIDI naprava	18
Slika 21: Pošiljanje MIDI signalov	18
Slika 22: Zaznan MIDI signal	18

KAZALO GRAFOV

Graf 1: Delovanje linearnega potenciometra(vir: Avtor naloge)	10
---	----

1. ZAHVALA

Zahvaljujemo se vsem, ki so pripomogli pri izdelavi te raziskovalne naloge. Sprva mentorici, za vso pomoč in podporo pri pisanju naloge. Posebna zahvala gre tudi podjetju Lasertehnik, za pomoč in izdelavo ohišja za krmilnik. Predvsem pa gre zahvala tudi šoli, za vso finančno podporo, brez katere bi bila realizacija naloge verjetno neuspešna.

2. POVZETEK

Cilj naše raziskovalne naloge je izdelava DMX (digitalno multipleksiranje) krmilnika, ki bo na lahko komuniciral z računalniškimi programi preko MIDI (Musical Instrument Digital Interface) signalov. Krmilnik bomo izdelali s pomočjo Arduino mikrokontrolerjev, ki bodo poganjali jedro sistema. Za uporabniški vnos pa bomo uporabljali stikala (MX Cherry), linearne potenciometre in encoderje. Ohišje krmilnika bo izdelano po meri, pri podjetju Lasertehnik.

Za to raziskovalno nalogo smo se odločili, saj so naše izkušnje z večjimi proizvajalci bolj kot ne negativne. Večji problem krmilnikov je njihova prenosljivost in cena, saj nizkocenovni proizvodi (npr. Showtech ipd.) po navadi ne ponujajo takšne fleksibilnosti glede dela, saj nas omejujejo glede funkcionalnosti. Na drugi strani pa se pojavljajo dražji ponudniki (npr. MA Lighting, Avolites, ETC Lighting), ki po navadi potrebujejo napravo, ki poganja zunanjo programsko opremo ali pa niso prenosni zaradi svoje teže.

Po mnenju uporabnikov je večji problem tudi cena, saj se krmilniki takšne velikosti prodajajo od 300€ in vse do 90.000€ pri večjih ponudnikih, zato si manjše korporacije ali posamezniki tega ne morejo privoščiti.

Največja prednost našega krmilnika je, da je kompatibilen z vsemi programi ki za svoje delovanje uporabljajo MIDI signale. Sprva smo se mislili omejiti le na osvetlitev, a smo med pisanjem algoritmov ugotovili, da se ga lahko uporablja tudi kot samostojna MIDI naprava, npr. glasbeni instrument.

3. HIPOTEZE

Hipoteza 1: Celoten izdelek bo cenovno ugoden za cilje uporabnike

Hipoteza 2: Program bo napisan v jeziku C++

Hipoteza 3: Uporabili bomo zunanjo bazo podatkov za prijavo v sistem

4. VSEBINSKI DEL

V vsebinskem delu naše raziskovalne naloge bomo opisali in predstavili sestavne dele našega izdelka ter prikazali bomo programsko kodo, ki je odgovorna za delovanje le tega krmilnika.

4.1 MIDI Krmilniki

MIDI krmilniki so naprave, ki posameznikom omogočajo pošiljanje digitalnih impulzov, katere programski vmesnik v računalniku lahko pretvori v glasbo ali druge signale po naših željah.

Ohišje našega izdelka je specifično prilagojeno za uporabo pri osvetlitveni tehniki, a z nekaj modifikacijami ga lahko uporabnik spremeni v praktično karkoli.

Ker se tudi sami ukvarjamo s osvetlitveno tehniko se zavedamo problemov, ki nastajajo pri obstoječih krmilnikih: cena in teža. Redki proizvajalci so uspeli na trg dostaviti prenosljivost, a so večinoma za to postavili ceno, ki si jo posameznik redko lahko sam privošči.

Naš MIDI krmilnik omogoča uporabniku, da se prosto sprehaja po prostoru, kjer dela. S tem pridobi veliko prednosti, npr. pogled kot občinstvo, premikanje luči kjerkoli se nahaja s katerokoli mobilno napravo...

4.2 Princip delovanja

Po tehtnem premisleku smo se odločili, da postavimo jedro našega sistema v razvojne ploščice Arduino Mega. To nam omogoča sprotno programiranje samega sistema, ter olajšano pisanje samih algoritmov, saj nam Arduino ponuja mnogo pripravljenih knjižnic.

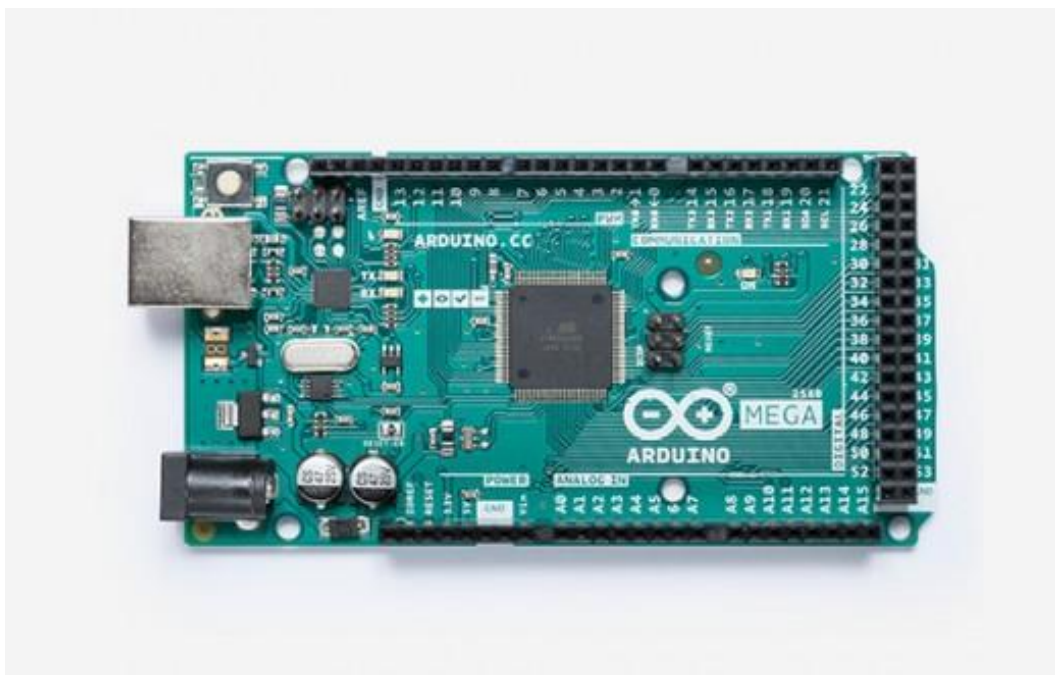
Vsak Arduino ima na svojem ATMega čipu naloženo modificirano programsko opremo, ki ga spremeni v t.i »MIDI« vmesnik. Ob kliku uporabnika na gumb, bo le ta poslal zahtevan MIDI signal na napravo, ki je priključena na USB. Po želji pa lahko dodamo tudi LED efekte, ki se bodo sprožili ob klikih.

4.3 Arduino MEGA

Arduino, ali drugače imenovan tudi Genuino, je razvojna ploščica, s katero lahko upravljamo različno vrsto naprav. Arduino ponuja več vrst razvojnih ploščic, kot npr. Uno za začetnike ali Mega za napredno uporabo. V našem primeru, bo Arduino Mega prišel prav, saj moramo priključiti več stikal in priključkov.

Arduino Mega je razvojna plošča z mikrokrmilnikom, ki temelji na ATmega2560. Ima na voljo do zdaj največje število digitalnih vhodov in izhodov – 54, 16 analognih vhodov, 4 strojne serijske priključke, 16Mhz oscilator, USB in napajalni priključek.

Za naš izdelek smo uporabili 4 razvojne ploščice, saj jih potrebujemo za povezavo vseh stikal in priključkov.



Slika 1: Arduino Mega (vir: Arduino Store)

4.4 Vhodne enote

Za vhodne enote našega izdelka smo izbrali linearne potenciometre, stikala MX Cherry blue in rotacijske kodirnike, saj slednje uporablja tudi večina orodij za delo v tej stroki.

4.4.1 Cherry MX Blue stikala

Na trgu obstaja mnogo različnih vrst stikal, ki jih proizvajalci uporabljajo pri izdelavi tipkovnic. Najbolj pogosti sta mehanično in membransko. Membranska stikala, najdena v večini cenejših tipkovnicah imajo namesto mehanskega stikala le prevodno črnilo, ki ob stiku z vezjem sklene potrebne kontakte. Mehanska stikala pa so zgrajena iz vzmeti, plastike in dveh pozlačenih metalnih kontaktov, ki se skleneta ob kliku gumba. Delimo jih na dve večji skupini, linearna in taktilna. Pri taktilnih občutimo, kdaj je tipkovnica klik registrirala, pri linearnih pa ne. Mehanske tipkovnice se največkrat uporabljajo za igranje iger ali pisanje daljših besedil, saj omogočajo večjo učinkovitost.

Največje podjetje, ki izdeluje mehanična stikala je MX Cherry. Vsa njihova stikala imajo 4mm hoda in se aktivirajo pri 2mm.

Najbolj razširjena so:

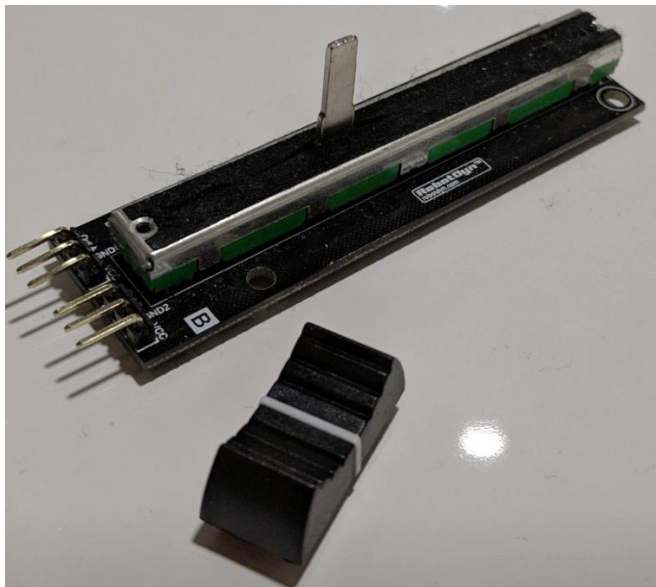
- Črna (linearna, uporabljena med »gamerji«)
- Rjava (taktilna, tišja, kombinacija med stikali za gaming in pisanje)
- Modra (taktilna, najglasnejša, najprimernejša za pisanje)
- Prozorna (podoben občutek membranskim, redko uporabljena)
- Rdeča (linearna, najpogostejše uporabljena v »gaming« tipkovnicah)



*Slika 2: Mehansko stikalo CHERRY MX Blue
(vir: Avtor naloge)*

4.4.2 Linearni potenciometer

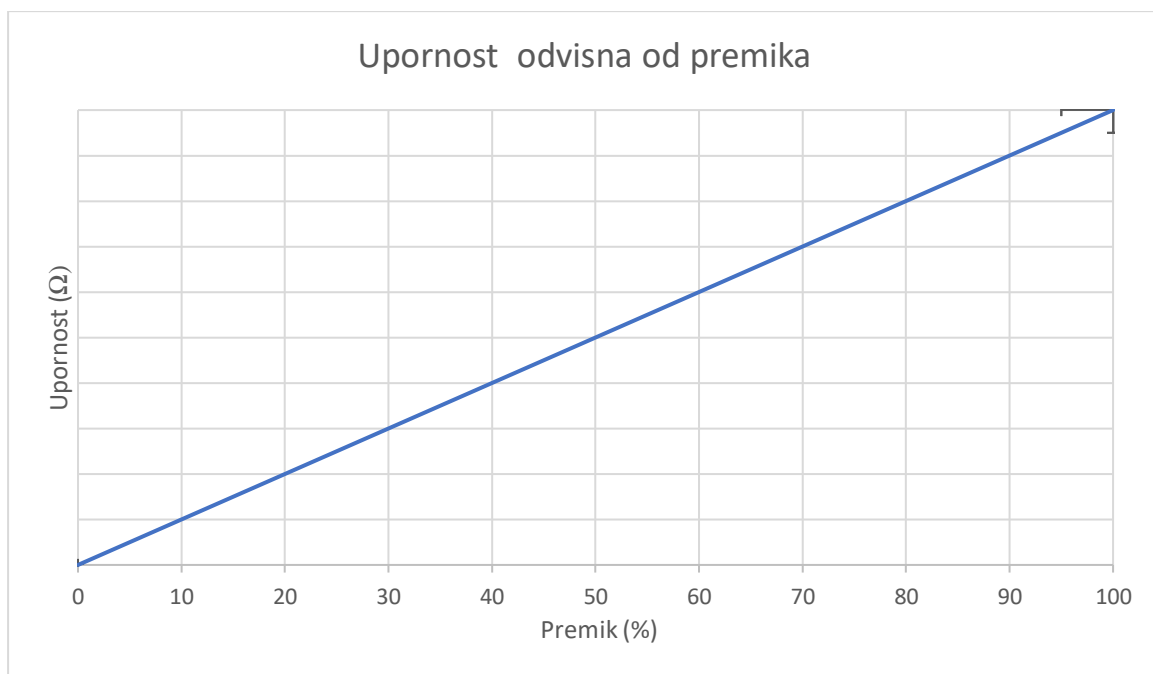
Potenciometer je delilnik napetosti s tremi terminali, ki je lahko ali drseči ali pa rotacijski. Ker bo potrebno pošiljati MIDI signale, smo v našem primeru uporabili **linearne potenciometre**.



Slika 3: Linearni potenciometer (vir: Avtor naloge)

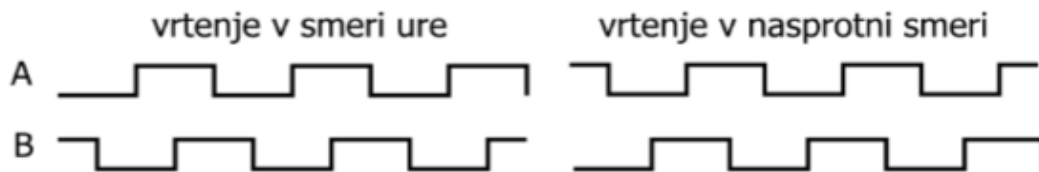
Linearni potenciometer se uporablja za detekcijo linearnih premikov. Uporablja se tam, kjer je zaželena napetost na izhodu, ki je linearno odvisna od premika.

Graf 1: Delovanje linearnega potenciometra (vir: Avtor naloge)

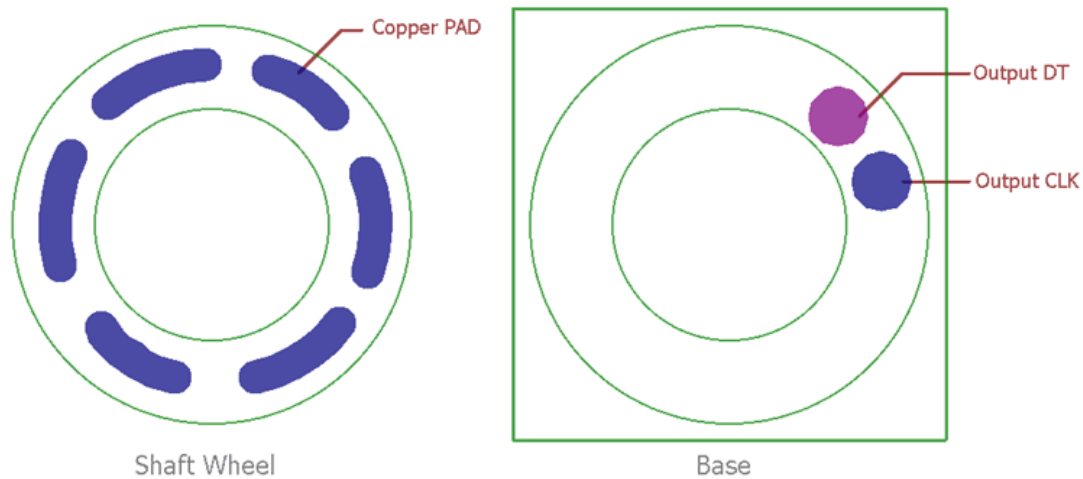


4.4.3 Rotacijski kodirnik

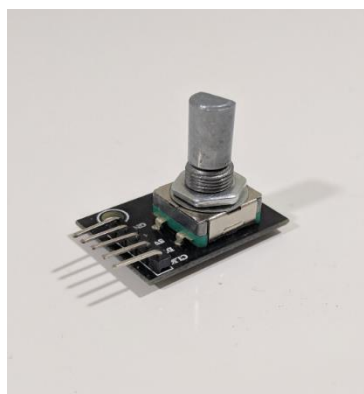
Rotacijski kodirnik je elektro-mehanska naprava, ki pretvarja fizično kotno pozicijo elementa v analogne oz. digitalne signale. Na kolesčku so nameščene bakrene podlage, katere imajo stik z bazo(base), ki se nahaja pod njim. Na bazi se nahajata dva priključka po katerih kolesček oz. bakrene podlage drsajo kadar ga obračamo.



Slika 4: Signali rotacijskega kodirnika (vir: Avtor naloge)

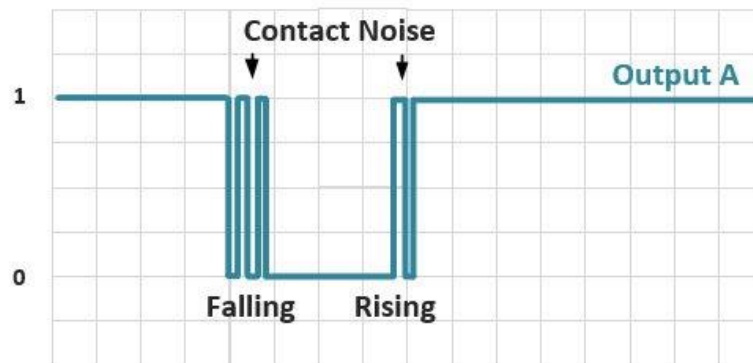


Slika 5: Sestava rotacijskega kodirnika (vir: circuitdigest.com)



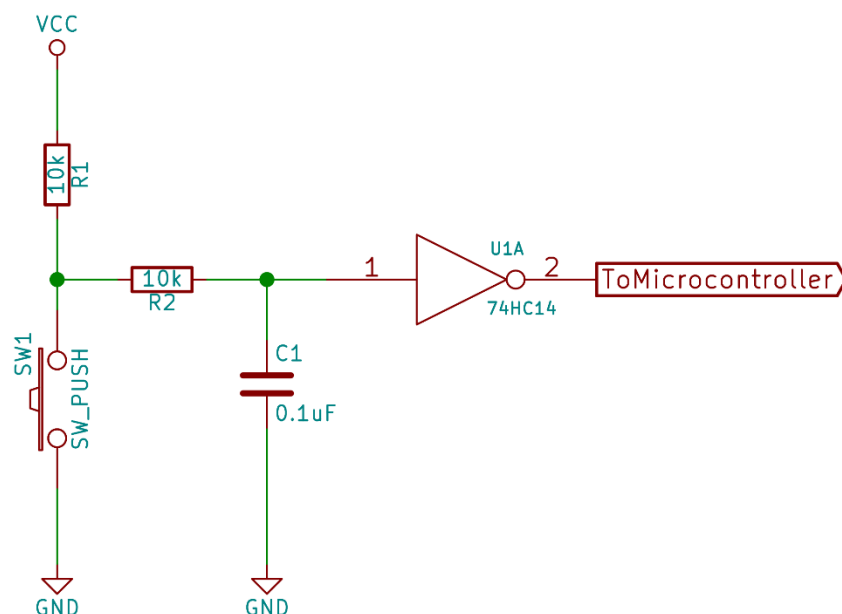
Slika 6: Rotacijski kodirnik (vir: Avtor naloge)

Ko smo prvič preizkusili nas rotacijski kodirnik smo ugotovili, da izhodni signal ni bil enakomeren. Kmalu smo ugotovili, da ima to težavo večina rotacijskih kodirnikov. Pri vrtenju gumba pride do popačenega signala oz. tako imenovanega "Contact Noise". To lahko preprečimo s tako imenovanim "Signal Debouncing". Ko zavrtimo gumb, prejmemo signal, ki izgleda popačeno.



Slika 7: Primer popačenega signala (vir: allaboutcircuits.com)

To težavo je možno odpraviti sistemsko, vendar ker smo uporabili Firmato kot naš protokol za komuniciranje smo si to pot onemogočili. Zato smo se odločili to težavo odpraviti strojno z "hardware debouncing circuit". Uporabili smo tako imenovani "hex inverting" Schmitt trigger (74HC14), kondenzator in dva upora.



Slika 8: Odprava popačenih signalov na rotacijskem kodirniku (vir: hackday.com)

4.4.4 IC 74HC4051

IC 74HC4051, tudi imenovan 74HCT4051, je enopolno analogno stikalo, ki je zanesljivo za uporabo skupno z multiplekserji in demultiplekserji. Stikalo vključuje tri poljubne digitalne vhode, osem samostojnih vhodov in izhodov, skupen vhod in izhod in "digital_enable" vhod (E). Za delovanje multiplekserja, moramo pin E kratko skleniti s pinom GND (zemlja). Potencial je tako 0V.

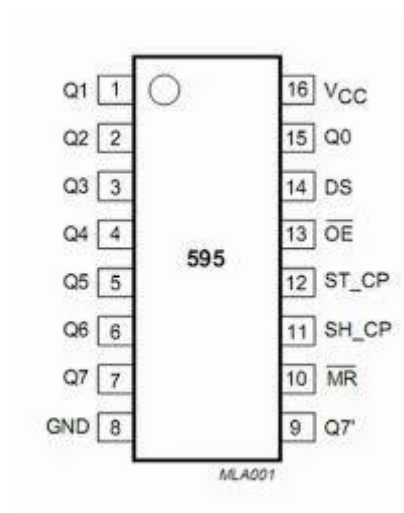


Slika 9: IC 74HC4051 (vir: avtor naloge)

4.4.5 IC 74HCT595

IC 74HCT595 je visoko hitrostni CMOS čip. 8 bitni pomikalni register sprejema podatke preko serijskega vhoda (DS) na vsak pozitivni interval, oziroma kadar se ura shift registra (STCP) spremeni.

Podatki iz vhodnega serijskega shift registra se postavi v izhodni register z naraščajočim pulzom na uporni uri (SHCP). Kadar sta obe uri povezani skupaj, bo vhodni shift register vedno en časovni cikel pred izhodnim registrom.



Slika 10: IC 74HCT595 (vir: <https://www.diodes.com/>)

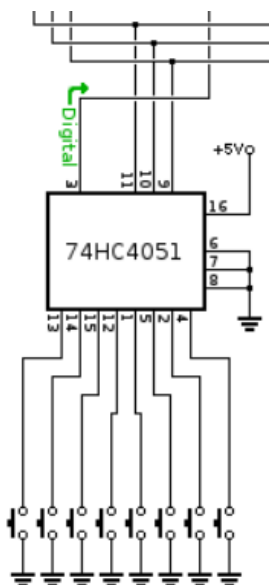
4.5 Uporaba multipleksiranja za dodajanje večjega števila vhodov

Zato, ker ima naš produkt več kot 100 digitalnih vhodov, kar Arduino ne podpira, smo morali uporabiti tehniko imenovano multipleksiranje. Odločili smo se za analogni multiplekser / demultiplekser 74HC4051.

Zapisali smo algoritem, ki bo uporabniku omogočal branje digitalnih ali analognih vrednosti iz naprav, ki so povezane na multiplekser. Vsak čip ima lahko priključenih 8 dodatnih naprav, za kar potrebuje 4 vhodno/izhodne pine na mikrokrmilniku. Sami smo se posluževali tudi tehnike »daisy chaining«, kar pomeni, da smo uporabili 3 skupne pin-e za čipe, ter vsakemu posebej dodelili analogni pin, po katerem bo pošiljal podatke nazaj.

```
void loop () {  
  int btnCnt = 0; //stevec gumbov  
  for(int i=0;i<number_of_mux;i++){ //skozi multiplekserje  
    for (int buttonCount = 0; buttonCount < 8; buttonCount++) { //skozi vse vhode  
      b0 = bitRead(buttonCount,0); //preberemo prvi bit iz števila na katerem je zanka  
      b1 = bitRead(buttonCount,1); //enako za drugi in tretji bit  
      b2 = bitRead(buttonCount,2);  
  
      digitalWrite(commonpin_1,b0); //posiljanje signala v multiplekser  
      digitalWrite(commonpin_2,b1);  
      digitalWrite(commonpin_3,b2);  
  
      buttonValue[btnCnt] = digitalRead(mux_pins[i]); //nastavitev vrednosti gumba v programu  
      btnCnt++;  
    }  
  }
```

Slika 11: Multipleksiranje gumbov (vir: Avtor naloge)



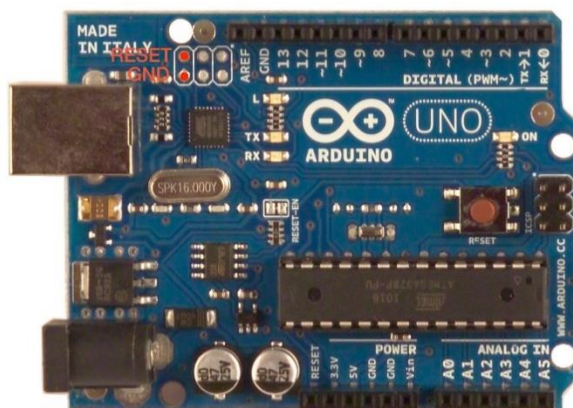
Na sliki so prikazane povezave na multiplekserju. Vsakemu digitalnemu vhodu smo dodali še »pulldown« upor, ki preprečuje popačene signale. Na napajanje vsakega čipa smo dodali tudi 0.1uF keramični kondenzator, ki skrbi za stabilnost napetosti, da ne bi prišlo do popačenih vrednosti.

Slika 12: Povezave na multiplexer (vir: Avtor naloge)

4.6 Nalaganje strojne programske opreme na mikrokrmilnik

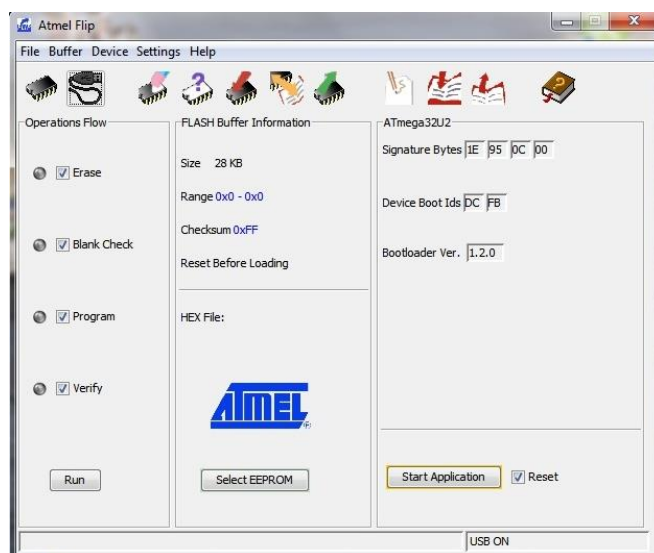
Ob prvem zagonu naprave smo opazili, da je strojna programska oprema na mikrokrmilniku namenjena serijski komunikaciji, zato bi potrebovali dodatno strojno opremo za pretvorbo signalov brez motenj. Zato smo se odločili, da bomo na naš Arduino namestili modificirano strojno programsko opremo z imenom Forty Seven Effects.

Sprva smo morali naložiti kodo, saj je za to operacijo potrebna serijska komunikacija. Ko je bila koda v naložena, smo morali Arduino resetirati v DFU način. Z vodnikom smo kratko sklenili RESET in GND pin označen na sliki, da ga je računalnik prepoznal kot DFU napravo.



Slika 13: DFU resetiranje (vir: Arduino.cc)

Ko smo imeli Arduino v DFU načinu, smo uporabili programsko orodje FLIP proizvajalca Atmel, da smo nanj naložili svoj firmware.

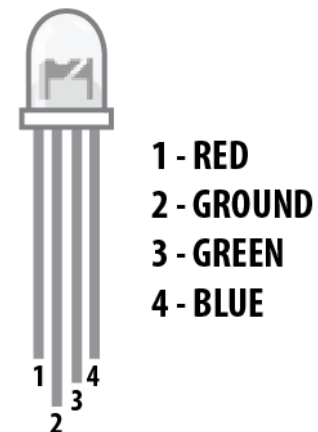
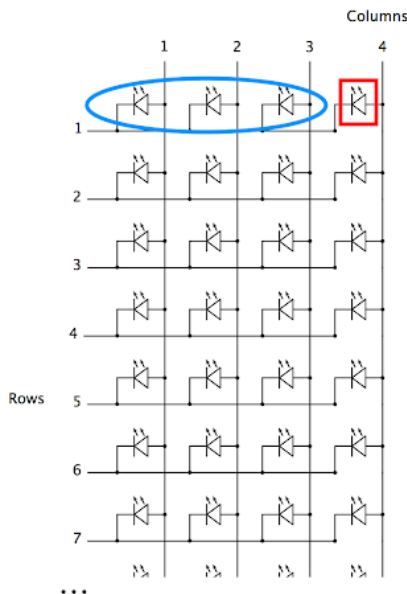


Slika 14: Atmel FLIP (vir: Avtor naloge)

Ko je bilo vse uspešno naloženo, smo Arduino resetirali in računalnik ga je prepoznal kot »MIDI« napravo.

4.7 Kontroliranje RGB LED diod

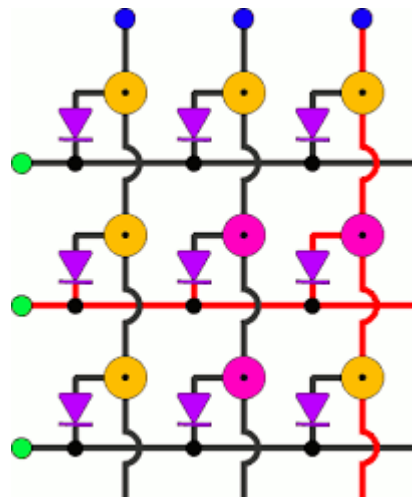
Za kontroliranje LED diod na krmilniku smo uporabili shift register 74HC595, ki lahko procesira 8 analognih vhodov/izhodov. Za vsako barvo (rdeča, zelena, modra) smo uporabili svoj shift register. Vse izhode smo povezali v t.i matrično shemo, ki omogoča kontrolo izhodov s toliko pini, kolikor je vrstic in stolpcev.



Slika 15: RGB LED dioda (vir: <https://learn.sparkfun.com/>)

Slika 16: matrica LED diod (vir: <https://learn.sparkfun.com/>)

Ko smo sestavili vezje smo ugotovili, da prihaja do problema imenovanega "ghosting". Do tega pride, ko aktiviramo tri diode v 2x2 kombinaciji, kar bo sprožilo tudi četrto. Da smo ta problem odpravili, smo uporabili dodatne diode, ki so omejile da tok v vezju teče le od diode, ne pa tudi proti njej.

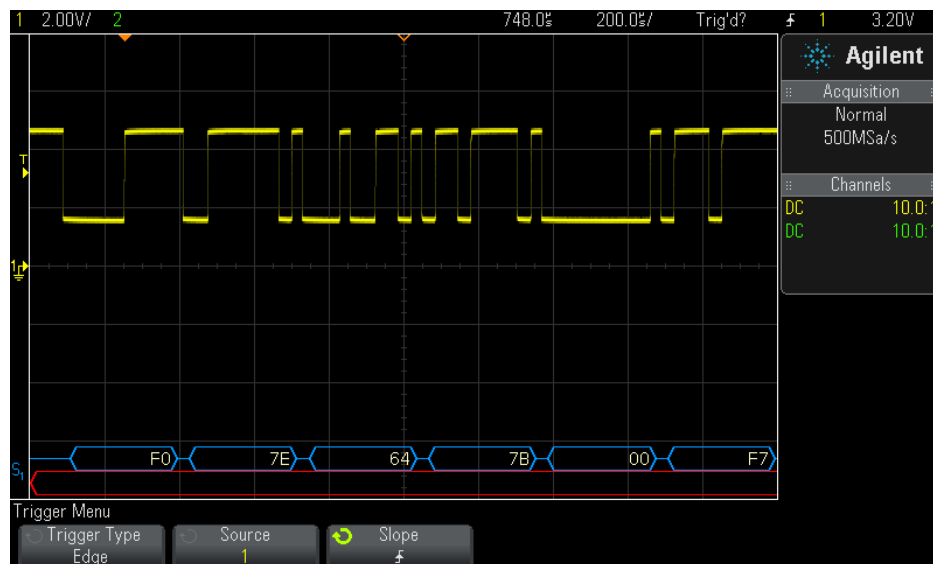


Slika 17: Eliminacija "ghosting" efekta (vir: <http://www.learningaboutelectronics.com/>)

4.8 MIDI

»MIDI, oziroma **M**usical **I**nstrument **D**igital **I**nterface je elektronski standardni protokol, s pomočjo katerega komunicirajo različne elektronske glasbene naprave, kot so npr. mešalna miza, sintetizator zvoka in računalnik.«(Wikipedija, 2019)

Sam protokol MIDI ne vsebuje nobenega zvočnega zapisa, temveč sprejema skupke podatkov ki jih pretvarja v točno določene signale ali tone.



Slika 18: MIDI signal (vir: frank-buss.de)

Pri pošiljanju MIDI sporočil poznamo točno določene parametre, med katerimi lahko izbiramo. To so: Note ON(pritisk note), Note OFF(spust note), Note Number(katera tipka je v uporabi), Velocity(moč pritiska na tipko), Program Number(spomin – memorija), Pitch Bend(barva in globina zvoka).

Posamezno sporočilo je sestavljeno iz treh bajtov. Prvi bajt(imenovan tudi statusni bajt) je sestavljen iz tipa sporočila(parameter) in kanala(channel) ki ga bomo uporabili. Drugi bajt vsebuje noto(note), ki jo bomo poslali(med 1 in 127), tretji bajt pa hitrost(velocity).

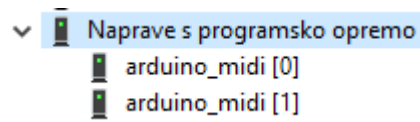
Maksimalna hitrost in nota lahko imata vrednost 127, saj ima ta bajt le 7 uporabnik bitov, posledično lahko izračunamo, da je $2^7 = 127$.

Status Byte	Data Byte 1	Data Byte 2
1001 0000	00111100	01111111
<div> <div>1001</div> <div>0000</div> </div>		
Note On	Note Number	Note Velocity
0	60 (C4)	127

Slika 19: Sestava MIDI sporočila (vir: Ramp me up, Scotty!)

4.9 Arduino kot MIDI vmesnik

Z uporabo MIDI strojne programske opreme smo zagotovili delovanje mikrokrmilnika kot MIDI naprave. Ko smo ga priključili na računalnik, smo ga lahko po namestitvi gonilnikov prepoznali v upravitelju naprav.



Slika 20: Arduino kot MIDI naprava (vir: Avtor naloge)

Nato smo se odločili preskusiti delovanje kode, ki smo jo uporabili za pošiljanje MIDI signalov. Program ob pritisku tipke pošlje 1 ali 0 na »note«, ki je določen s številko gumba.

```
if(digitalRead(mux_pins[i]) == 1){  
    MIDI.sendNoteOn(btnCnt, 127, 1);  
}  
if(digitalRead(mux_pins[i]) == 0){  
    MIDI.sendNoteOff(btnCnt, 0, 1);  
}
```

Slika 21: Pošiljanje MIDI signalov (vir: Avtor naloge)

Nato smo za namene testiranja uporabili še MIDI-OX, da smo prebrali signal.

```
000076D4 1 -- 90 00 7F 1 C -1 Note On  
000076D5 1 -- 80 01 00 1 C#-1 Note Off
```

Slika 22: Zaznan MIDI signal (vir: Avtor naloge)

5. ZAKLJUČEK

Pri izdelavi raziskovalne naloge smo se naučili mnogo novega o programiranju v okolju Arduino. Izpopolnili smo tudi znanje programiranje v drugih jezikih, ter pridobili dodatne kompetence pri sodelovanju z drugimi podjetji.

Ob začetku raziskovanja smo si postavili tudi nekaj hipotez, ki smo jih potrdili ali ovrgli med samim raziskovanjem.

Hipoteza 1: Celoten izdelek bo cenovno ugoden za ciljne uporabnike.

To hipotezo lahko deloma potrdimo, saj je končni produkt v skladu s svojo uporabnostjo res cenejši od svoje konkurence.

Hipoteza 2: Program bo napisan v jeziku C++

To hipotezo smo potrdili, saj je vsa koda napisana z uporabo Arduino vmesnika, v jeziku C++.

Hipoteza 3: Uporabili bomo zunanjo bazo podatkov za prijavo v sistem.

To hipotezo smo ovrgli, saj smo se odločili, da ne bomo uporabili spletne aplikacije.

6. DRUŽBENA ODGOVORNOST

Naš izdelek, MIDI DMX krmilnik, smo zgradili z namenom, da bi delo lučarskih tehnikov hitreje in bolj učinkovito. Za primer lahko vzamemo program Titan One, ki ga je razvilo podjetje Avolites. Je eden najbolj naprednih programov za delo z osvetlitveno tehniko, a glede na število funkcij ki jih ima, je bilo podjetje primorano skriti določene funkcije v menije.

Z uporabo našega izdelka bo lahko uporabnik dostopal do teh funkcij z uporabo posebnih gumbov, brez potrebnega vložka v izdelek, ki bi deloval samo z eno programsko opremo.

7. VIRI

Bervar, G.(2008). C++ NA KOLENIH. 2. posodobljena izdaja. Ljubljana: Študentska založba (Zbirka Scripta)

Fitzgerald, S. in Shiloh, M.(2015). ARDUINO PROJECTS BOOK. 1.izdaja. Torino, Italija (Arduino).

Krkoč, P. (2014). Programirajmo Arduino z lahkoto. 1. izdaja. Ljubljana: Ax Elektronika d.o.o

Autodesk eagle. Pridobljeno iz <https://www.autodesk.com/products/eagle/overview>, dne 15.1.2020.

Arduino. Pridobljeno iz <https://www.arduino.cc/>, dne 25.1.2020

Avolites (20.1.2020), pridobljeno iz: <https://www.avolites.com/>

MIDI (1.2.2020), pridobljeno iz: <https://en.wikipedia.org/wiki/MIDI>

MX Cherry (10.1.2020), pridobljeno iz: [https://en.wikipedia.org/wiki/Cherry_\(keyboards\)](https://en.wikipedia.org/wiki/Cherry_(keyboards))

Rotary encoder (28.1.2020), pridobljeno iz: https://en.wikipedia.org/wiki/Rotary_encoder