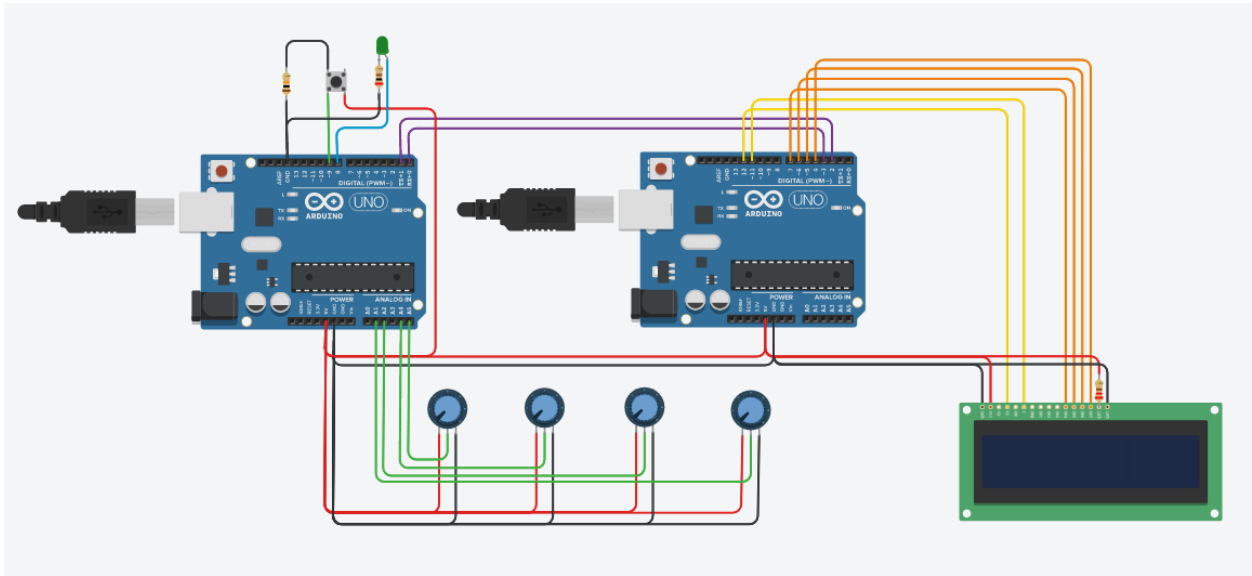


MIDI kontroler



Slika 1 Shema vezja

Komponente in ostala oprema

Uporabljene so naslednje komponente:

- 2x kopija Arduino Uno R3 ploščica,
- 2x 10k ohm potenciometer,
- 1x joystick,
- 1x zelena LED
- 1x gumb
- 2x 220 ohm upor
- 1x 10k ohm upor
- 1x LCD
- 1x USB kabel

Za programiranje, razhroščevanje in testiranje so bile uporabljena naslednja programska orodja:

- Ableton 10 Live
- Arduino IDE
- FLIP 3.4.7

Ostalo:

- vreča dobre volje
- copati

Začetek projekta

Razvoja sem se lotil z raziskavo terena in s tem odgovoril na osnovna vprašanja: ali imam vso potrebno opremo, ali obstajajo že napisane knjižnice za pošiljanje MIDI podatkov, ali bom potreboval dodatne programske vmesnike, ki bodo pretvarjale MIDI signale... Hitro sem našel mnogo knjižnic, ki omogočajo delo z MIDI podatki. Izbral MIDI knjižnico avtorjev FortySevenEffects, ki je zelo popularna, je enostavna za uporabo, zanjo pa obstaja veliko gradiva in objavljenih končnih izdelkov, zato sem bil prepričan, da za vse težave, na katere bom naletel, že obstajajo rešitve.

Potek razvoja

Na Arduino sem priklopil gumb, ki bo sprožal enostaven MIDI signal, npr. vklop in izklop določenega tona. Priklopil sem tudi LED, ki bo služila kot indikator, ali je gumb pritisnjen ali ne.

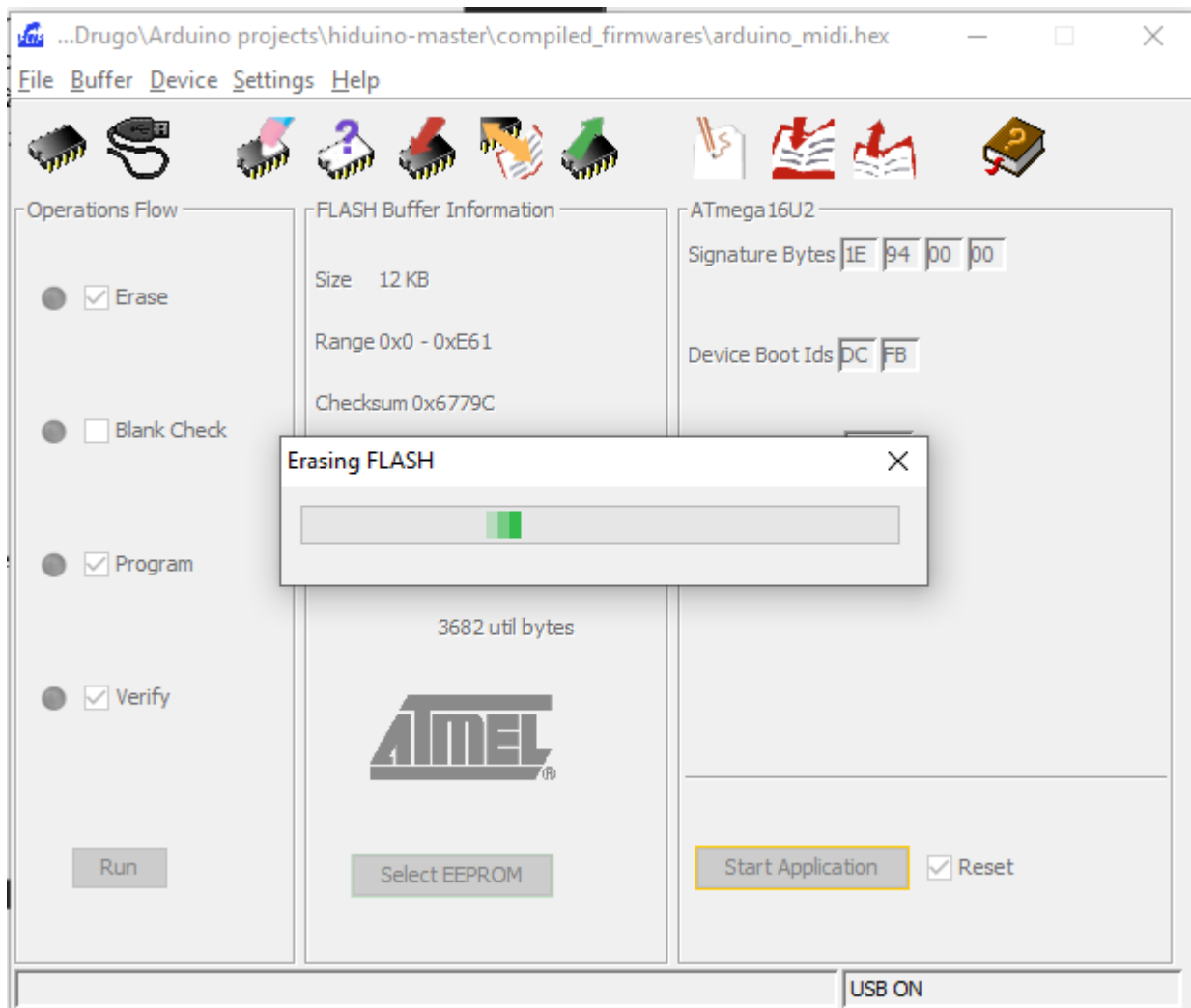
Nato sem zagnal Arduino IDE, v katerem sem programiral kodo. Napisal sem enostaven program, ki ob pritisku na gumb, pošlje MIDI signal za začetek tona C4 s polno glasnostjo.

Ko sem zagnal kodo na ploščici, sem ugotovil da se MIDI signal ne pošilja, oz. ga računalnik ne zazna, zato sem se posvetil temu problemu.

Po brskanju po različnih člankih in raznih forumih sem naletel na programsko opremo Hiduino. Hiduino omogoča pošiljanje MIDI signalov iz Arduina v računalnik brez dodatnih programskih vmesnikov.

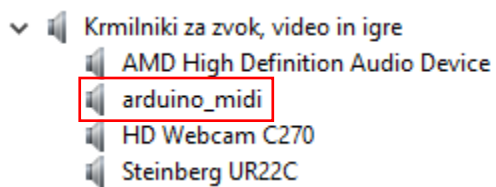
Namestiti ga je potrebno na USB krmilnik (v mojem primeru Atmega16u2) in ne direktno na procesno enoto (v mojem primeru Atmega328p-PU). To je seveda predstavljalo nov izziv, ker tega do tistega trenutka nisem nikoli prakticiral.

Po branju nešteti formih in zastarelih Arduino dokumentacij sem naletel na program FLIP, ki je namenjen posodabljanju gonilnikov. Izgubil sem precej časa, da sem ugotovil, kako preklopiti Arduino v DFU način. Na javnem git repositorju sem našel .hex datoteko, ki sem jo nato naložil na krmilnik Atmega16u2 s programom FLIP. Nato sem iztaknil USB kabel, ki povezuje ploščico z računalnikom in ga nazaj vstavil.



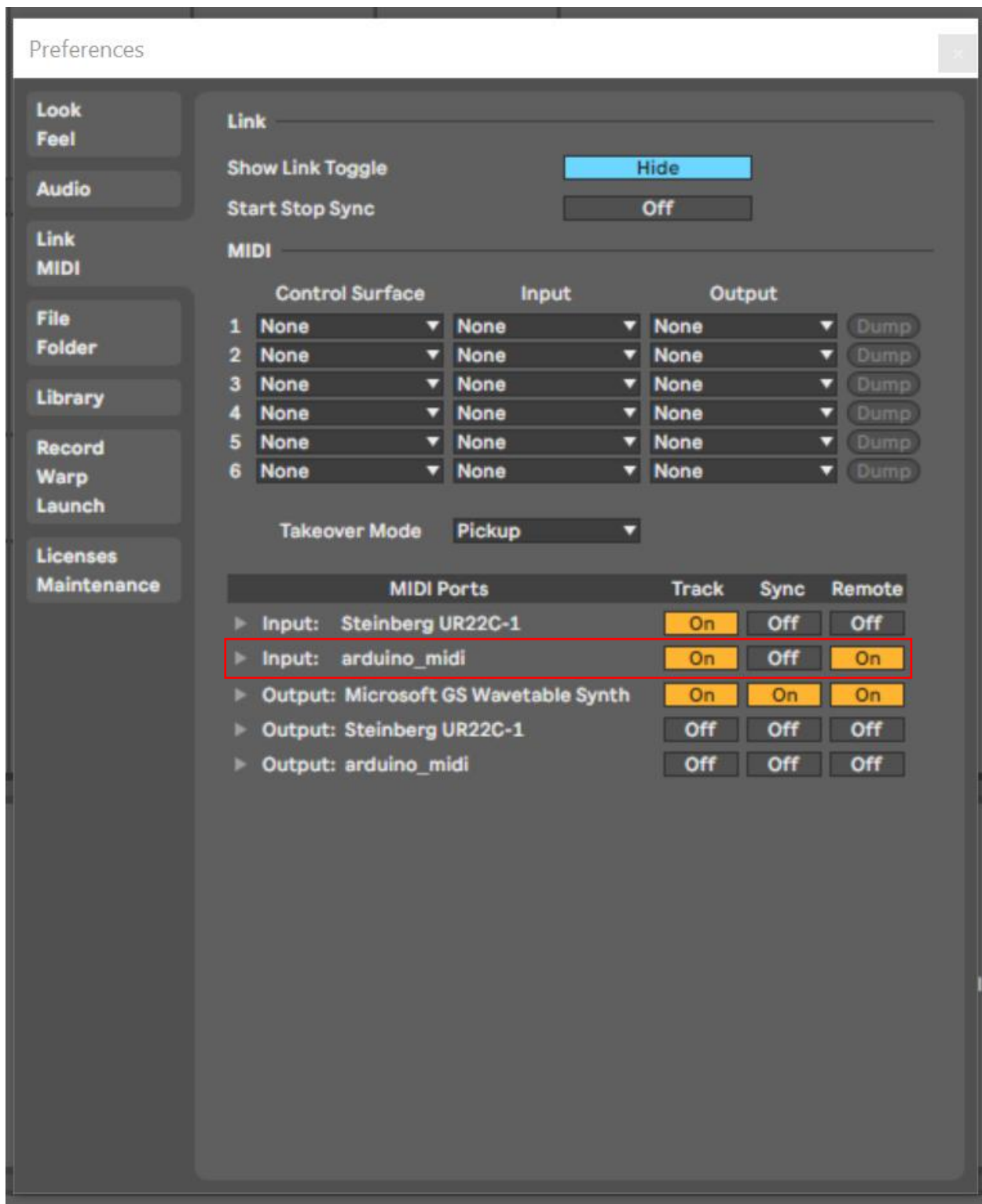
Slika 2 Posodabljanje USB gonilnika

V upravitelju naprav sistema Windows 10 se je pojavila nova naprava, ki je bil zelo dober znak.



Slika 3 Detekcija nove naprave

Odprl sem Ableton 10 Live, v katerem sem testiral izdelek. Na začetku je bila potrebna konfiguracija nove MIDI naprave.



Slika 4 Konfiguracija

V pogledu se je sem dodal nov instrument in mu kot vhod določil novo dodano MIDI napravo – arduino_midi. Ob pritisku na gumb se je uspešno sprožil ton C4, tako kot sem določil v kodi.



Slika 5 Sprejem pravilnega signala

MIDI kontroler s samo eno tipko je seveda precej neuporabna zadeva, zato sem priklopil še 2 potenciometra in joystick, ker sem imel te komponente na voljo in so najbolj uporabne za to vrsto naprav.

Dopolnil sem tudi kodo za mikroprocesor. Seveda sem najprej naložil originalno kodo za USB krmilnik ploščice, saj brez tega ni bilo mogoče naložiti nove kodo na mikroprocesor.

Kmalu sem odpravil vse napake programa, ki sem ga napisal in produkt je začel delovati tako kot sem si zamislil.

Lotil sem se naslednje naloge – serijska komunikacija med 2 mikrokrmilnika in prikazovanje podatkov na LCD zaslonu.

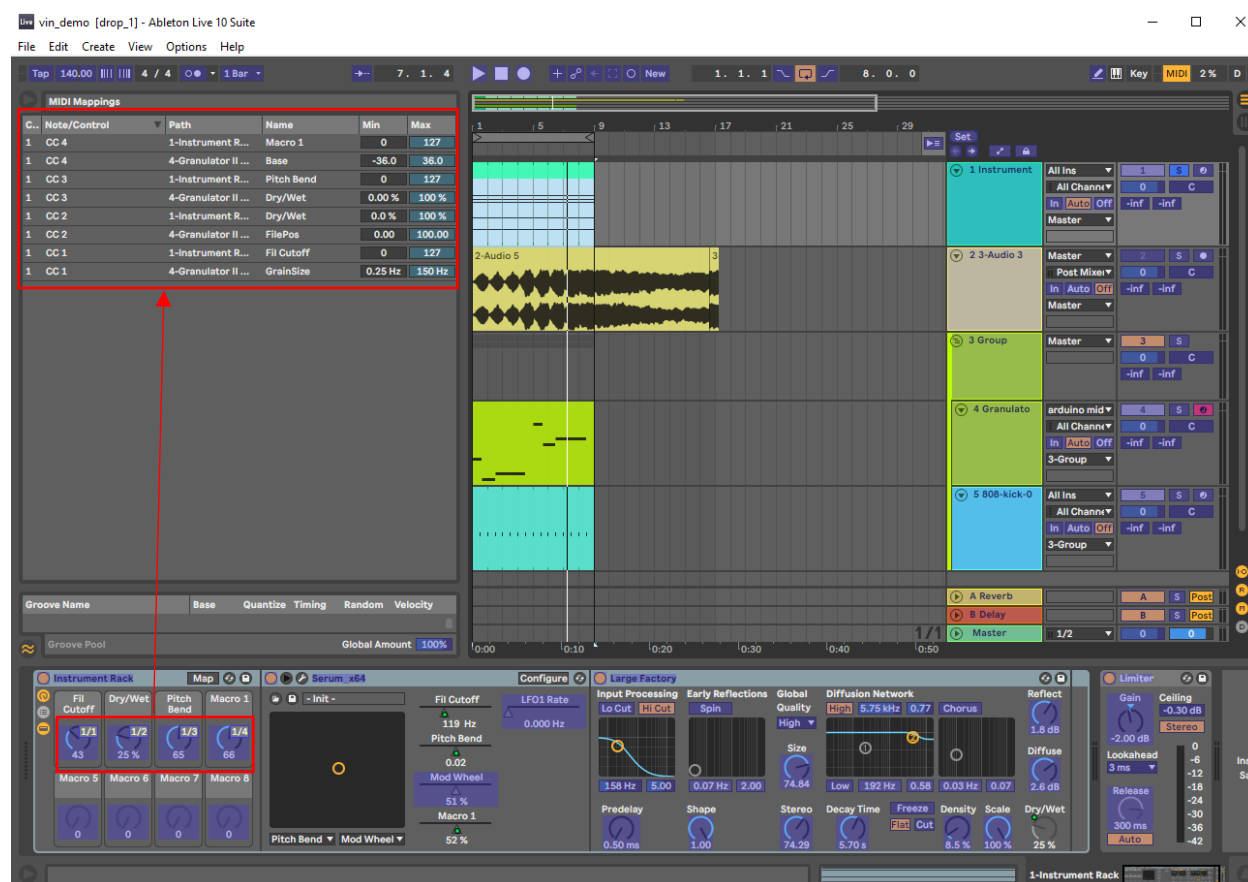
Ker sem uporabljal serijsko komunikacijo za pošiljanje MIDI signalov v računalnik in strojne specifikacije ploščic, ki sem ju uporabljal, ne omogočajo 2 serijskih komunikacij hkrati, sem se zatekel k programskim rešitvam. Uporabil sem knjižnico SoftwareSerial, ki ponuja enake funkcionalnosti kot knjižnica Serial.

Arduinota sem povezal na primeren način, in napisal enostavno kodo, ki sprejema podatke po UART protokolu, in jo naložil na dodan Arduino. Na Arduino, na katerem se procesirajo MIDI signali, sem naložil spremenjeno kodo, ki konstantno pošilja podatke o pozicijah potenciometra in joysticka.

Ugotovil sem, da je UART komunikacija zelo nestabilna, sploh s povezavami preko breadboarda, kjer so stiki zelo rahli. Za reševanje te težave mi je zmanjkalo časa. Načrtujem eksperimentiranje z drugimi protokoli, npr. I2C in SPI. Alternativa pa vedno ostane spajkanje kvalitetnejših kablov direktno na Arduino ploščici, ali pa večja dopustnost napak in počasnejše posodabljanje podatkov.

Rezultat

Potenciometra in joystick je mogoče dodeliti vsakemu spremenljivemu elementu v Abletonu.



Slika 6 Prikaz dodeljevanja funkcij vhodov

Zaključek

Projekt je bil zelo naporen za izdelavo, ampak zabaven za uporabo – to mi je dalo motivacijo za reševanje težav, na katere sem naletel med razvojem. S tem sem tudi poglobil znanje o prenosu podatkov, o delovanju MIDI naprav, o delovanju izbranega Arduinota...

Čeprav je koda relativno preprosta, saj uporabljene knjižnice že vsebujejo vse potrebne metode, je bilo razhroščevanje precej duhomorno, saj vsakič, ko sem hotel testirati popravke, sem moral te korake ponavljati: vklop DFU načina, posodabljanje gonilnikov, vklop in izklop ploščice, nalaganje popravljenih kode, ponoven vklop DFU načina, posodabljanje gonilnikov, vklop in izklop ploščice. To je seveda zahtevalo precej časa in volje, pa vendar mi je glavni cilj uspelo doseči.

Projekt bom v prihodnosti nadgradil z izboljšano komunikacijo med mikrokontroleroma in delujočim LCD zaslonom.