



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Joakim Mattson

GEMINI MDJ-1000 MIDI
-OHJELMOINTI MIXXX DJ-
OHJELMALLE
Mixxx versio 2.0

Tekniikka
2017

TIIVISTELMÄ

Tekijä	Joakim Mattson
Opinnäytetyön nimi	Gemini MDJ-1000 MIDI -ohjelmointi Mixxx DJ-ohjelmalle
Vuosi	2017
Kieli	suomi
Sivumäärä	42 + 3 liitettä
Ohjaaja	Timo Kankaanpää

Tämän opinnäytetyön tavoite on suunnitella ja toteuttaa MIDI-kartoitusta ja ohjausohjelmaa DJ-ohjelmalle, mitä voidaan ohjata ulkoisella MIDI-laitteella.

Valmiin tuotteen on tarkoitus tulla tekijän omaan käyttöön DJ-keikoilla vaihtoehtoisena ratkaisuna. Valmis tuote tulee tämän projektin jälkeen kaikille saatavaksi osoitteessa www.joakim-mattson.fi.

Projekti toteutetaan käyttäen XML- ja JavaScript-ohjelmointikieliä. Tutkimukseen kuuluu myös MIDI-viestien rakenne ja katsaus DJ-ohjelman perustoimintoihin.

Työn tavoitteet saavutettiin odotettua paremmin. Kaikkia ulkoisen laitteen toimintoja ei voitu mallintaa DJ-ohjelmalle, koska ne olivat tarpeettomia tai niille ei ole vastaavaa toimintoa DJ-ohjelmassa. Ulkoisen laitteen toimintoihin, joita ei ohjelmoitu, voidaan tulevaisuudessa ohjelmoida jotain muuta tarpeen tullen.

ABSTRACT

Author	Joakim Mattson
Title	Gemini MDJ-1000 midi programming for Mixxx DJ software
Year	2017
Language	Finnish
Pages	42 + 3 Appendices
Name of Supervisor	Timo Kankaanpää

The purpose of this thesis was to design and implement a midi mapping and control program for a DJ program, which can be controlled by an external MIDI device.

The finished product is intended to be used in the author's own use as an alternative to DJ gigs. The finished product will be available to everyone after this project at www.joakim-mattson.fi.

The project is implemented by using XML and Javascript programming languages. The study also includes the structure of MIDI messages and a review of the basic functions of the DJ program.

Work objectives were achieved better than expected. All the functions of an external device could not be modeled for a DJ program because some of them were unnecessary or did not have the corresponding function in the DJ program. In the future, the external device functions that are not programmed can be programmed in the future when something else is needed.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENTEET JA TERMIT.....	6
1 JOHDANTO.....	10
2 KÄYTETYT TEKNOLOGIAT.....	11
2.1 MIDI.....	11
2.1.1 Laitteisto.....	11
2.1.2 Viestit.....	12
2.2 XML.....	15
2.3 JavaScript.....	16
3 VALMIIN TUOTTEEN MÄÄRITTELY.....	17
4 PROJEKTIN SUUNNITTELU.....	19
4.1 MIDI-viestit.....	20
4.2 Viestikohtainen ongelmaratkaisu.....	21
4.3 Vaihtoehtoiset ohjelmat ja laitteet.....	21
4.3.1 DJ-ohjelmat.....	22
4.3.2 DJ-ohjaimet.....	22
5 TOTEUTUS.....	23
5.1 MIDI -viesteistä Mixxxx -toimintoihin.....	24

5.2	Monitilatoiminnot.....	27
5.3	Uudelleen määritettäviä toimintoja.....	29
6	TOIMINNAN TOTEAMINEN.....	30
6.1	Testilaitteisto.....	30
6.2	Ohjelmoinnin yhteydessä tapahtuva testaus.....	36
6.3	Live-testaus.....	38
7	YHTEENVETO.....	39
	LÄHTEET.....	40
	LIITTEET	

LYHENTEET JA TERMIT

Baud rate	Symboli lähetysnopeus, bittien siirtonopeus
CSS	Cascading Style Sheet, ryöpyttävä tyyliarkki
Gemini -laite	Gemini MDJ-1000 ammattikäyttöinen multimediasoitin
HID	A human interface device, ihmisen liitántälaite
HTML	HyperText Markup Language, hyperteksti merkintäkieli
JavaScript	Ohjelmointikieli, lyhennettynä js. Tunnetaan myös nimellä ECMAScript
MIDI	Musical Instrument Digital Interface, musiikkisoittimien digitaalinen rajapinta
MIDI-OX	Midiviestien tarkasteluun suunniteltu ilmainen ohjelma
Mixxxx	Ammattikäyttöinen DJ-ohjelma
SGML	Standard Generalized Markup Language, standardoitu yleistetty merkintäkieli
UART	Universal asynchronous receiver/transmitter, yleiskäsitteinen asynkroninen vastaanottaja/lähetäjä
USB	Universal Serial Bus, yleiskäsitteinen sarjaväylä
XML	Extensible Markup Language, laajennettava merkkikieli

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. MIDI-kytkentä.....	12
Kuvio 2. MIDI-viestien tavutyypit.....	13
Kuvio 3. MIDI-viestien tyypit.....	13
Kuvio 4. MIDI-OX -ohjelman miditietovirta ja sen tulkinta.....	14
Kuvio 5. XML-esimerkki.....	15
Kuvio 6. JavaScript-esimerkki HTML:n joukossa.....	16
Kuvio 7. Ympyrämalli.....	19
Kuvio 8. Gemini MDJ-1000:n ulkoasu viittauksineen liitteessä 1.....	20
Kuvio 9. Play -ja kiekkokosketuksen xml.....	24
Kuvio 10. Init -funktion osa.....	26
Kuvio 11. Yksinkertainen ledohjaus.....	27
Kuvio 12. Mixxx-ääniasetukset.....	31
Kuvio 13. Mixxx-ohjainasetukset.....	31
Kuvio 14. Laitteiston yleiskuva.....	32
Kuvio 15. Mixxx-käyttöliittymä.....	32
Kuvio 16. Mixxx-kanava-asetukset.....	33
Kuvio 17. Mixxx-toisto- ja liikutustoiminnot.....	34
Kuvio 18. Mixxx-tempoliukuvipu.....	34
Kuvio 19. Mixxx-silmukkatoiminnot.....	35
Kuvio 20. Mixxx-hyppypisteet.....	36

Taulukko 1. Vaatimuksien tärkeysjärjestys.....17

Taulukko 2. Uudelleen määritettävät toiminnot.....29

LIITELUETTELO**LIITE 1.** MIDI-viesti ja Mixxx-toimintotaulukko**LIITE 2.** JavaScript-kokoelma

1 JOHDANTO

Tässä projektissa toteutetaan Mixxx -nimisen DJ -ohjelman ja Geminin valmistaman MDJ-1000 -mallin mediaohjaimen kommunikaatio-ohjelma. Valmistuksessa tutustutaan MIDI -protokollaan, JavaScriptin rakenteeseen ja XML -tiedon tallentamiseen. Projektin kannalta tuotos on ainutlaatuinen, koska kukaan ei ole sitä tehnyt aiemmin, ainakaan julkisesti. Valitun DJ -ohjelman ja Gemini -laitteen välille ei olla koskaan ennen tätä projektia julkaistu rajapintaa. Gemini MDJ-1000:n ollessa nimenomaan multimedioasointin, voitaisiin sillä toistaa musiikkia joko USB -massamuistista tai CD -levyltä. Laitteen MIDI -toiminto on siinä mielessä hieno, että sillä voidaan ohjata jotain toista laitetta niin halutessa, kuten esimerkiksi tässä projektissa ohjataan DJ -ohjelmaa nimeltään Mixxx.

Valmista ohjelmaa tullaan testaamaan live-ympäristössä, missä siitä saadaan kaikki mahdolliset virheet karsittua pois 6.4.2017 Warantti ry :n järjestämänä Wykypäivänä Tropiclandiassa. Tuotteen on siis määrä olla käyttövalmis tuohon päivään mennessä siten, jotta sillä ehditään harjoitella ennen päivää.

Gemini MDJ-1000 on valittu projektiin ohjaimeksi, koska se saattaa olla ensimmäinen Pioneer-valmistajan veroinen kilpailija. Toki Pioneerin valmistamat laitteet ovat olleet DJ -laitteiden huipputuotteita jo pitkään. Ensimmäistä kertaa tälle lähes monopoliasemalle on tullut vastustaja, koska Gemini MDJ-1000 on noin kuusi kertaa edullisempi kuin markkinoiden johtava vastaava laite. DJ -ohjelmien puolella Mixxx ei ole suinkaan eniten käytössä oleva, mutta sen sijaan se on täysin ilmainen ohjelma ja on alunperin suunniteltu Linux -ympäristöön. Mixxx on myös ainoa Linux -ympäristössä toimiva ammattitasoinen DJ -ohjelma, mikä oli koko tämän projektin alkuidea. Tästä syystä valittiin Mixxx, koska haluttiin käyttää Linux -ympäristöä.

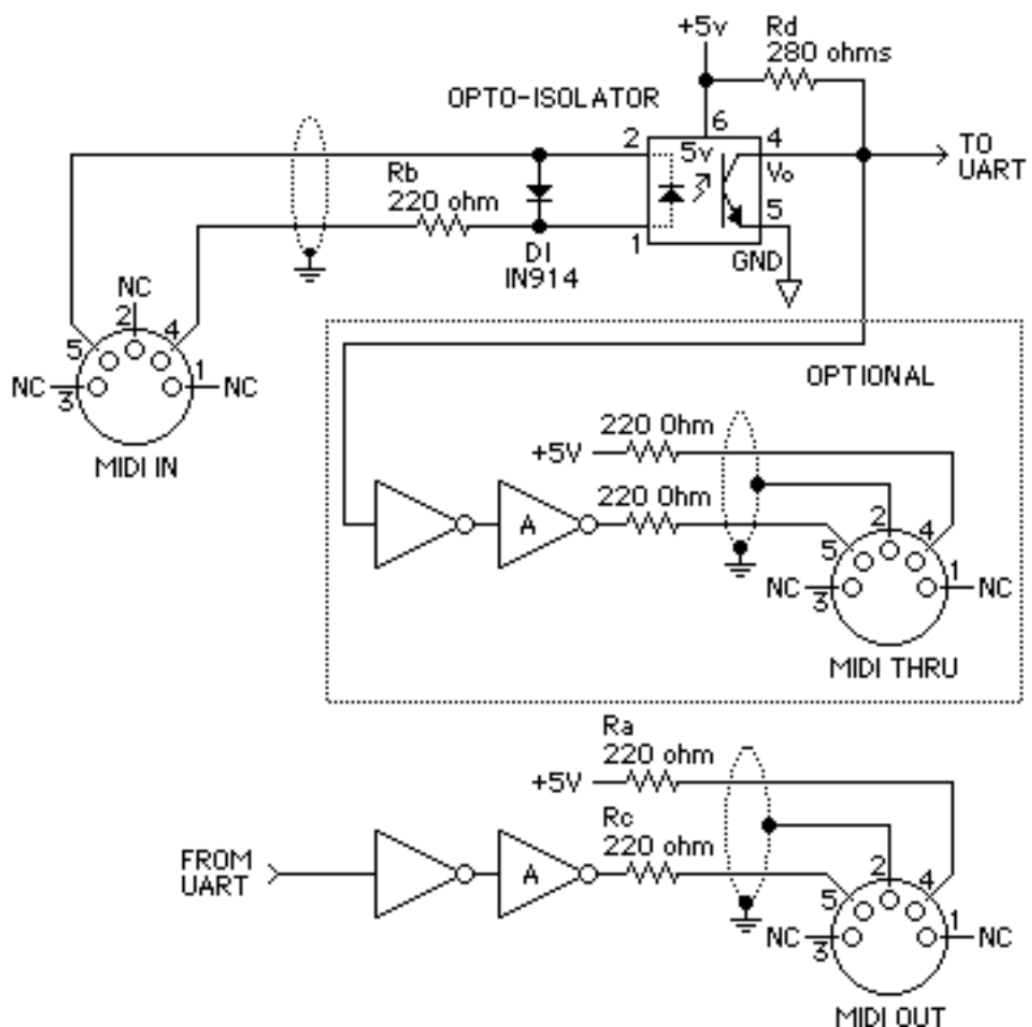
2 KÄYTETYT TEKNOLOGIAT

2.1 MIDI

Musiikkisoitinten digitaalista rajapintaa kutsutaan tuttavammin MIDIksi, joka tulee sanoista Musical Instrument Digital Interface. Se kehitettiin 1980 -luvulla lähinnä synteettisaattoreiden välisen viestinnän toteuttamiseksi. Täten voitiin ohjata monta laitetta yhdellä master -laitteella tai toisinpäin. Vuosien saatossa MIDIn määritelmää on muutettu eri tavoin, mutta lopulta se on löytänyt tiensä miltein jokaisen musiikkilaitteen valmistajan tuotteeseen ja toimii eräänlaisena standardina eri laitteiden välillä vielä tänä päivänäkin. MIDI-määritelmään viitattaessa puhutaan yleensä MIDI 1.0 versiosta, vaikka kyseiseen versioon on tuotu paljon lisää sen ikääntyessä. Periaate onkin säilynyt samana vuosikymmenet. /1/

2.1.1 Laitteisto

Midi on kommunikointiprotokolla, jossa data kulkee 31,25 (+/-1%) Kbaudin nopeudella asynkronisesti. Lähetetty viesti sisältää yhden alkubitin, kahdeksan databittiä ja loppuu yhdellä lopetusbitillä. Alkubitti on looginen nolla virran ollessa ylhäällä ja lopetusbitti looginen yksi virran ollessa alhaalla. Bitit lähetetään vähiten merkittävä bitti edellä. Kytkenältään MIDI -laitteisto toimii virtasilmukalla. Lähtevä virta kiertää laitteen A ulostuloportista laitteen B sisääntuloportin ja ulostuloportin välillä. MIDI-kytkennästä (**Kuvio 1.**) huomataan kuinka kahden MIDI-laitteen välinen kytkentä ei ole välttämätön viestien lähettämistä varten. Sen sijaan lähetetty viesti menee niin sanotusti harakoille, koska mikään toiminto ei valvo viestien lähetystä. Yhdessä MIDI-portissa tulee kulkea viestit vain ja ainoastaan yhteen suuntaan. Tarvitaan siis vähintään kaksi eri suuntaan toimivaa MIDI -porttia onnistuneen viestin välitykseen sisääntuloportin ja ulostuloportin välillä. Kytkentäkaaviosta (**Kuvio 1.**) nähdään kuinka ohjelmoidun laitteen UART tulee kytkeä, jotta se olisi standardinmukainen. /1/

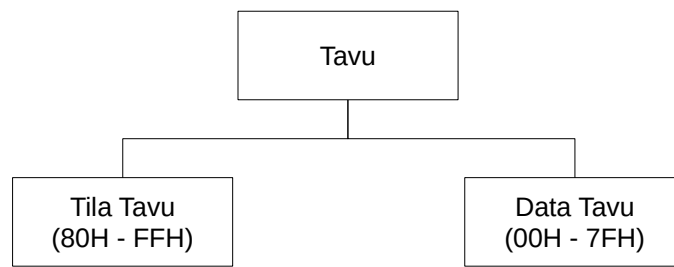


Kuvio 1. MIDI-kytkentä

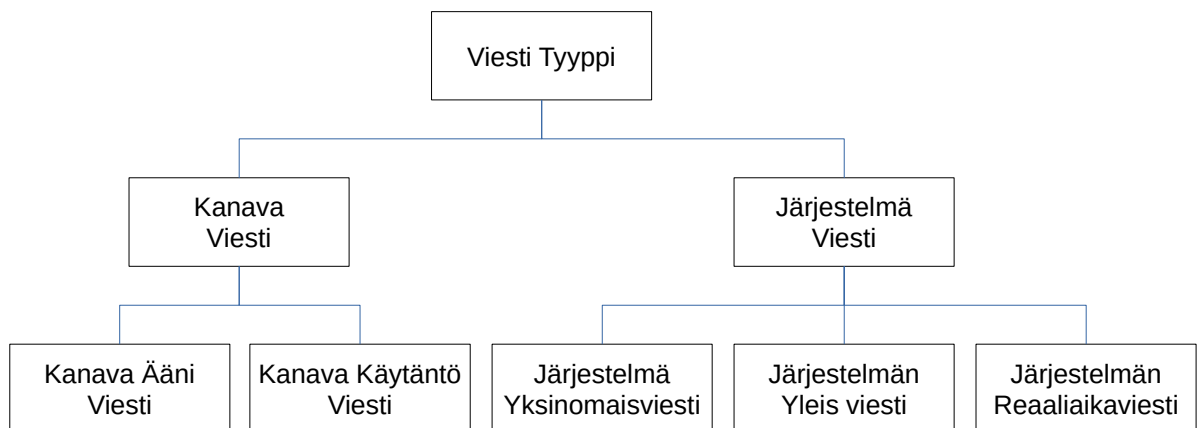
Tässä projektissa käytetyt laitteet eivät käytä 5 pinnisiä MIDI-portteja, vaan viestit siirtyvät USB-standardin yli yhdistäen tulevan ja lähtevän viestivirran saman protokollan alle. Periaatteeltaan ja toiminnaltaan lopputulos on kuitenkin sama, koska MIDI-laitteelle löytyy Windows 7 -käyttöjärjestelmästä USB-ajurit. Windows tunnistaa näin ollen MIDI-laitteen MIDI-laitteena ilman erillistä ulkopuolista panosta.

2.1.2 Viestit

Vaikka tässä projektissa käytetään MIDI-porttien sijaan USB-standardia viestien lähettämiseen, ovat MIDI-viestit siitakin huolimatta samat. MIDI-viestin tavut jakautuvat kahteen eri tyyppiin (**Kuvio 2**). /1/



Kuvio 2. MIDI-viestien tavutyypit



Kuvio 3. MIDI-viestien tyypit

Yksi MIDI -viesti sisältää useimmiten kolme tavua, jotka jakautuvat kolmeen osaan:

- Status: Useimmin status -tavulla tarkoitetaan tietyssä kanavassa tapahtuvaa muutosta, kuten koskettimistolla tapahtuvaa nuotti päälle/pois. Voi olla myös joitain käskyjä, kuten aloita, lopeta, jatka tai ajoituskelloa.
- Data1: Ensimmäisellä data-tavulla merkitään usein nuotin numero mitä nuottia käsky koskee. Toimintoja on valittavissa 0-127 desimaalin (00-7F H) väliltä. Statuksen koskiessa esimerkiksi sävelkorkeutta, voidaan tällä määritellä vähiten merkitsevä bitti.
- Data2: Toisella datatavulla merkitään yllä olevan nuotin voimakkuus tai jälkikosketus. Osalla toimintoja on valittavissa 0-127 desimaalin (00-7F H) väliltä. Statuksen koskiessa esimerkiksi sävelkorkeutta, voidaan tällä määritellä eniten merkitsevä bitti.

TIMESTAMP	IN	PORT	STATUS	DATA1	DATA2	CHAN	NOTE	EVENT
000000F0	2	--	91	31	1D	2	C# 3	Note On
000002BC	2	--	81	31	01	2	C# 3	Note Off
00001429	2	--	91	34	21	2	E 3	Note On
000015B3	2	--	81	34	0C	2	E 3	Note Off

Binary Hex	Chan	Event
10010001=91	2	Note On
10000001=81	2	Note Off
10010001=91	2	Note On
10000001=81	2	Note Off
10010000=90	1	Note On
10000001=81	2	Note Off
10010010=92	3	Note On
...		
10011111=9F	16	Note On

Kuvio 4. MIDI-OX -ohjelman miditietovirta ja sen tulkinta.

MIDIvirran kaappaukseen käytetään MIDI-OX nimistä ohjelmaa, joka on valmistettu juuri MIDI-datan kaappaamista ja analysointia varten (**Kuvio 4**). Tilanteessa on painettu kahta koskettimiston näppäintä yksitellen peräkkäin. Kuvan sarake CHAN kertoo käyttäjälle mihin kanavaan lähetetty viesti on osoitettu. Sama voidaan todeta tarkastelemalla Status -tavun sisältöä neljästä viimeisestä bitistä. $2^4=16$. MIDI viestejä voidaan lähettää yhteensä 16 kanavaan. Status -tavu jakautuu puolestaan edelleen kahteen osaan, missä ensimmäisessä puoliskossa kerrotaan mitä halutaan tehdä ja jälkimmäisessä kanava mitä käsky koskeen. Status -tavut välillä 80-EF ovat kanavakohtaisia, minkä jälkeen Fn -tason status -viestit eivät ole kanavakohtaisia, vaan muutos koskee järjestelmää (**Kuviot 3-4**).

2.2 XML

XML on itsenäinen merkkäuskieli tiedon tallentamiseen ja lähettämiseen. Se on jatkokehitetty alunperin SGML -kielestä ja on hyvin samannäköinen kuin HTML. XML -kieli kehiteltiin kuvaamaan sen sisältämää dataa hyvin yksinkertaisesti, jotta tieto voitaisiin siirtää järjestelmästä toiseen viestin säilyessä täysin samana. Sen kehittäjä, World Wide Web Consortium, aloitti XML -kielen kehittämisen vuonna 1996. XML ei siis itsessään tee yhtään mitään. Se on vain tiedon tallennustapa, joka vaatii erillisen ohjelman tallennetun tiedon esittämiseen. Monessa tapauksessa XML:ää käytetään tiedon varastointiin ja HTML -kieltä tiedon formatointiin ja esille tuontiin. Niinpä XML -kielessä ei ole ennalta määrättyjä tunnisteita, vaan kirjoittaja voi ne määritellä itse. Koska XML -muotoon tallennettu tieto ei sisällä minkäänlaista tietoa siitä kuinka tieto tulee esittää, voidaan samaa sisältöä käyttää helposti monessa eri esityksessä. Yksinkertaisessa XML -esimerkissä (**Kuvio 5.**) viestejä voi olla monta peräkkäin. Viestiin voidaan lisätä tunnisteita tai poistaa ilman sen vaikuttamista viestin esitystapaan, koska viestin esitysohjelma on oma itsenäinen ohjelma tai osana isompaa kokonaisuutta. /2-3/

```
<kirja>
  <otsikko kategoria="web">HTML:n Perusteet</otsikko>
  <kirjoittaja>Maija Meikäläinen</kirjoittaja>
  <vuosi>2009</vuosi>
  <hintaa>25.00</hintaa>
</kirja>
```

Kuvio 5. XML-esimerkki.

Tunnisteita kutsutaan elementeiksi, joilla voi olla jokin attribuutti. Attribuutti liittyy aina tiettyyn elementtiin (tässä tapauksessa otsikko-elementtiin) ja kerrotaan elementin aloitus -tagissa. Vuosi, hinta, kirjoittaja ja otsikko ovat periytymisen suhteen siskoksia keskenään. Tiedon tallentamisen kannalta ei kuitenkaan ole järkevää tallentaa yhtä kirjaa yhteen tiedostoon. Järkevän rakenteen kannalta kirja -elementin tulisi periytyä vielä ylemmältä tasolta, kuten esimerkiksi <kirjakauppa>. Näin voidaan tallentaa monta kirjaa alekkain. Mitä tarkempi rakenne on, sitä paremmin voidaan hallita tallennettua tietoa.

2.3 JavaScript

JavaScript on Brendan Eichin kehittelemä ohjelmointikieli lähinnä web-sovellusten kehittämistä varten. Sitä on kehitelty ja paranneltu nykyiseen muotoonsa sen luomisvuodesta 1995 lähtien. Alkujaan JavaScriptiä kutsuttiin Mochaksi , kunnes se muutettiin LiveScriptiksi ja myöhemmin JavaScriptiksi. Ensimmäisen kerran JavaScript otettiin käyttöön Netscape Navigator 2.0 -selaimessa vuonna 1996. ECMA otti silloisen Mocha -kielen hellään hoivaansa, mistä julkaistiin ensimmäinen standardiversio nykyisestä JavaScriptistä. Toisella nimellä JavaScriptiä kutsutaan ECMAScript, minkä viimeisintä versiota kutsutaan yksinkertaisella nimellä ECMAScript 2016. Voidaan siis jopa sanoa, että JavaScript on yhtä vanha kuin web itsessään. /4/

Dynaamisena ja oliopohjaisena ohjelmointikielenä JavaScript muistuttaa enemmän C-kieltä kuin Javaa. On myös hyvä muistaa, että JavaScript ei ole Java-kieli, vaikka molemmat ovatkin olio-pohjaisia.

JavaScript on yksi web -ohjelmoijan kielistä, mitä käytetään nykyisin melkein jokaisen sivuston yhteydessä. Siinä missä HTML:n avulla määritellään sivuston sisältö voidaan Css:llä määritellä sen ulkoasu. JavaScriptin avulla voidaan ohjelmoida sivusto käyttäytymään tietyllä tavalla (**Kuvio 6**). JavaScriptiä käytetään myös pelien tai, kuten tässäkin projektissa, työpöytäsovellusten luomiseen (vaikkakin osana toista ohjelmaa).

```
<html>
<body>
  <p id="demo">JavaScript voi muuttaa HTML sisältöä.</p>
  <button type="button"
    onclick="document.getElementById('demo').innerHTML =
    'Hei vain!'">Paina tästä!</button>
</body>
</html>
```

Kuvio 6. JavaScript-esimerkki HTML:n joukossa

Ulkoisella JavaScriptillä tarkoitetaan erillistä tiedostoa, mikä sisältää niin sanotun js -koodin. Tällöin ulkoinen tiedosto tulee lisätä pääohjelmaan omalla lisäyskomennolla.

3 VALMIIN TUOTTEEN MÄÄRITTELY

Valmis tuote ei ole millään tavalla nähtävissä muulla tavalla kuin toimivana kokonaisuutena Mixxx -ohjelman ja Gemini -laitteen välillä. Mixxx -ohjelman ruudulla näkyviä toimintoja ja ns. piilotettuja toimintoja voidaan muuttaa ilman, että käyttäjä joutuu käyttämään hyväksi tietokoneen hiirtä tai näppäimistöä. Syynä on, että nopeat ohjaustoiminnot tuovat käyttäjälle enemmän aikaa keskittyä olennaiseen ja saada aikaan merkittävämpiä tuloksia nopeammin. Gemini -laitteella tulee voida ohjata kaikkea, mitä laitteesta itsestään löytyy merkintöjä kyseiselle toiminolle. Kappalehaun yhteydessä tulee käyttää näppäimistöä ja hiirtä, mutta tämä toiminto ei kuulu tähän projektiin. Gemini-laitteessa olemassa olevien nappien omat merkit tulee käyttää hyväksi niin pitkälle kuin mahdollista.

Taulukosta 1 nähdään projektille asetettuja vaatimuksia tärkeysjärjestyksessä. Numero yksi tarkoittaa, että vaatimus on pakko täyttyä, kolmonen, että olisi kiva ominaisuus.

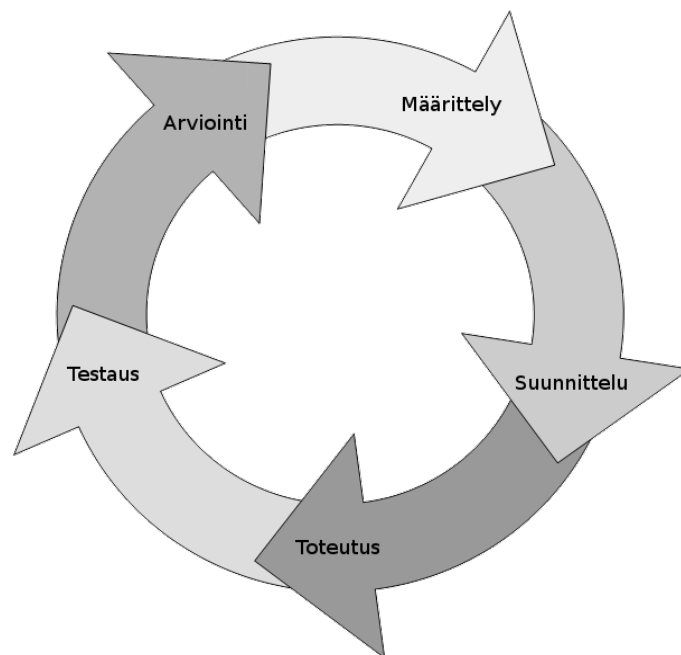
Taulukko 1. Vaatimuksien tärkeysjärjestys.

Viite	Vaatimuksen kuvaus	Tärkeys
T1	Gemini -laitteeseen merkittyjä nappien merkintöjä tulee matkia.	1
T2	Gemini -laitteen LCD -näyttöön ei kajota	1
T3	Gemini -laitteen toimintoihin, jotka eivät lähetä tai vastaanota MIDI-viestejä, ei kajota.	1
T4	Ohjelman tulee toimia niin, että sitä voidaan käyttää neljään Mixxx:n tarjoamaan kantaan (deck) neljällä eri ohjaimella.	1
T5	Kirjoitetun koodin tulee toimia Mixxx -ohjelmanympäristössä riippumatta käytössä olevan tietokoneen laitteistosta. (Win/Linux/Mac)	2
T6	Gemini -laitteen MIDI-ohjattavia ledejä tulee käyttää hyväksi	2

T7	Gemini -laitteen levylautasen nopeuden mallintaminen virtuaaliseksi	3
----	---	---

4 PROJEKTIN SUUNNITTELU

Projektin suunnittelussa käytettiin protoilun ja vesiputousmallin yhdistelmää, spiraalimallia tai ympyrämallia (**Kuvio 7.**), missä edellisen toteutuksen pohjalta määritellään seuraava vaihe. Kierros lähtee liikkeelle määrittelystä ja päättyy arviointiin. Arvioinnissa päätetään, testauksen pohjalta saadun tiedon perusteella, mitä tehdään seuraavaksi. Tätä ympyrää tehdään niin pitkään kunnes tuote on halutun kaltainen.



Kuvio 7. Ympyrämalli.

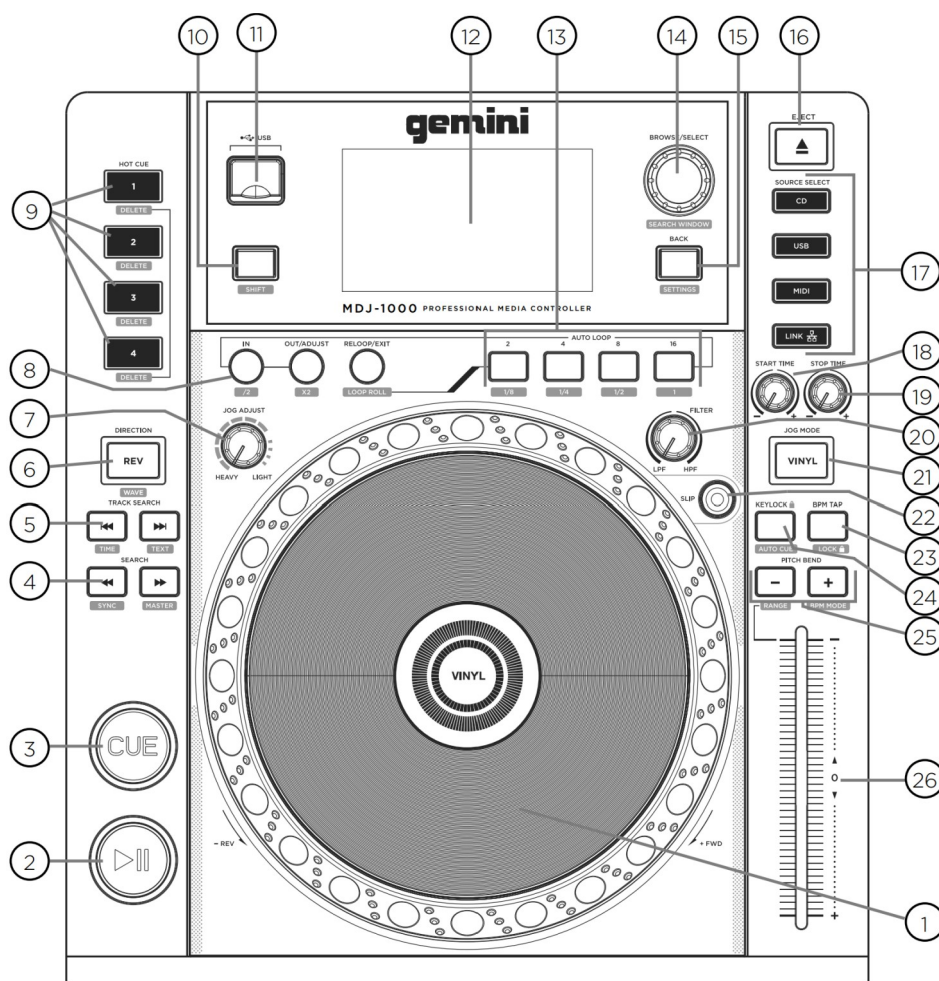
Määrittelyosiossa määritellään haluttu muutos. Suunnitteluosiossa pohditaan kuinka haluttu muutos voidaan saavuttaa. Toteutusosiossa suunniteltu muutos toteutetaan. Testausosiossa toteutettu muutos testataan heti sen lisäyksen jälkeen. Testauksen jälkeen saadaan arviointi siitä, oliko tehty muutos kannattava vai ei. Arvioinnin pohjalta tehdään päätös mitä asialle tehdään.

Projekti koostuu useasta pienestä osasta, joihin kaikkiin käytetään samaa ympyrämallia, kuten esimerkiksi Play toiminto Mixxx-ohjelmassa sekä Gemini -laitteessa. Samaa ympyrämallia käytetään koko projektin läpi eri versioiden kohdalla. Mielekkään version valmistuttua voidaan todeta tuote valmiiksi.

Projektin ensimmäisen toimivan version valmistuttua, sen todellinen toimivuus testataan todellisessa tilanteessa.

4.1 MIDI-viestit

Suunnittelun kannalta on järkevää katsoa mitä Gemini -laite lähettää ja tehdä MIDI -viesteistä taulukko helpottamaan viestien hallintaa ohjelmoitaessa. Gemini -laitteen jokainen viesti luetteloidaan. Laitteen käyttöliittymäkartasta (**Kuvio 8.**) nähdään missä ohjelmoitavat napit sijaitsevat Gemini -laitteessa. Viestit kaapataan käyttäen MIDI-OX -nimistä MIDI-viestien kaappaukseen tarkoitettua ilmaista sovellusta. Samaan taulukkoon sijoitettiin myös Mixxx -ohjelman toiminnot, mihin Gemini -laitteen lähettämät viestit liitetään (LIITE 1).



Kuvio 8. Gemini MDJ-1000:n ulkoasu viittauksineen liitteessä 1.

4.2 Viestikohtainen ongelmaratkaisu

Suunnitelmana on toteuttaa valmis ohjelma iteroiden ongelma kerrallaan. Tätä yksityiskohtaista nappi kerrallaan tapahtuvaa ongelman ratkaisua ei kannata selittää, ellei siihen liity joitain merkittäviä alitoimintoja, kuten looproll- toiminto ja sliptila, mitkä ovat Mixxx -ohjelman toimintoja. Iteroinin aikana aikaisemmin tehtyyn taulukkoon (LIITE 1) lisätään Mixxx -ohjelman sisäisiin toimintoihin liitetyt MIDI -viestit. Ongelmalliset tapaukset kerätään talteen ja käsitellään toteutusosiossa.

4.3 Vaihtoehtoiset ohjelmat ja laitteet

Vaihtoehtoisia ohjelmia ja laitteita tämän projektin tiimoilta löytyy kymmeniä ellei jopa satoja. Todellinen syy käyttää Mixxx -ohjelmaa johtaa juurensa kauempaa, jolloin Mixxx oli hyvin varhainen versio. Tuolloin neljälle kanavalle ei ollut tukea, eikä käyttöliittymä ollut mukautuva. Mixxx kuitenkin on tänä päivänä ainoa Linux -ympäristöön kehitetty ammattitasoinen DJ -ohjelma. Koska käytössä olevassa kannettavassa sattui olemaan Linux, haluttiin myös löytää DJ -ohjelma Linuxille. Aikaa kului ja Mixxx kehittyi versioksi 2.0. Tässä versiossa tuli lopulta käyttöön nelikanavainen toistomahdollisuus muiden hienouksien kanssa. Gemini MDJ-1000 valittiin projektiin sen takia, koska se on kenties ensimmäinen laite, mikä kykenee kilpailemaan Pioneer -valmistajan kanssa markkinoiden kärkipaikasta edullisella hinnallaan. Osasyynä myös se, että MDJ-1000 -laitteen kirjastohallinta ei ole (vielä) yhtä hyvä kuin PC:lle tarkoitettussa ohjelmassa. Tempolaskuri ei myöskään ole MDJ-1000:ssä (vielä) niin tarkka kuin tietokoneohjelman laskuri. Tarkemmin sanottuna, siksi haluttiin yhdistää kaksi edullista vaihtoehtoa keskenään tukemaan toinen toisiaan.

Tarkastellaan seuraavaksi hiukan mitä muita vaihtoehtoja voi myös käyttää. On otettava kuitenkin huomioon, että jokaisella ihmisellä on omat mieltymykset ja on mahdotonta sanoa mikä on hyvä tai huono, koska kaikki eivät käytä työkaluja samalla tavalla. On myös huomioitava sekin seikka, ettei kaikilla ohjelmilla ole kaikkien ohjainten MIDI-kartoistusta.

4.3.1 DJ-ohjelmat

Markkinoiden kenties ylivoimaisesti suosituin dj -ohjelma on Native Instrumentin julkaisema Traktor pro 2. Voisi helposti sanoa, että siinä on kaikki mitä tarvitsee. Native Instruments on myös johtava DJ -ohjauslaitteiden valmistaja Pioneerin kanssa. Pioneerin julkaisema Recordbox on tekemässä myös hyvää nousua maailmalla. Se toimiikin täydellisesti jo olemassa olevien Pioneer -laitteiden kanssa uusimman päivityksen jälkeen. Listan kärkipäässä on myös Serato, joka kamppailee kovasti Native Instrumentin kanssa ykkössijasta. On hyvä mainita sekvenssimasterointityökalut, kuten Ableton. Ableton ei oikeastaan ole DJ -ohjelma vaan sekvenssi -nauhoitus -masterointityökalu. Sitä kuitenkin voidään käyttää live-esiintymisessä sen live-esiintymisominaisuuksien tarjoaman ominaisuuksian kautta. Samaa toimintaa tekee myös Image Linen kehittämä FL Studio. FL Studiossa on myös olemassa niin sanottu performer mode, joka sallii Abletonin kaltaisen samplerin ominaisuudet.

4.3.2 DJ-ohjaimet

Ylivoimaisesti käytetyin DJ -laite lienee Pioneerin valmistama CDJ-2000. Tämä tuote löytyy lähes jokaisesta yökerhosta nykyisin. Tuota tuotetta voidaan toki käyttää myös ohjaimena, vaikka sen ei tarvitsekaan sitä olla. Sama pätee myös Geminin MDJ-1000:n. Tarkastellaan kuitenkin pelkästään ohjaimia eikä CD-soittimia. Behringerin CMD Studio 2a lienee aloittavalle tähdelle se edullisin ohjain. Laite toimii samalla ohjaimena ja äänikorttina. Tarvittaessa voi valita 4a-version, mikä on suunniteltu neljän kanavan ohjaamiseen. Numark tekee loistavaa työtä kehittäessään omaa Mixtrack pro -asemaansa. Geminilta löytyy myös pelkkiä ohjaimia, esimerkiksi G4V, joka sisältää myös DJ-mikserin dekkien välissä. Pioneerilta löytyy myös XDJ-1000 MK2, joka on periaatteessa sama kuin CDJ-1000, mutta ilman CD-soitinta.

5 TOTEUTUS

Tätä projektia varten käytettiin Mixxx:n stabiilia versiota 2.0.0, mikä on viimeisin julkaistu versio Mixxx:stä. Mixxx DJ-ohjelman wiki -sivustolla on luetteloitu kaikki Mixxx:n sisältämät toiminnot, joita voidaan muuttaa. Mixxx:stä on tulossa uusi 2.1.0 -versio, minkä mukana osa vanhemmista toiminnoista todennäköisesti tullaan ottamaan pois ja korvataan vastaavalla käskyllä. Tämä tarkoittaa toteutuksen osalta sitä, että on käytettävä enintään 2.0.0. version toimintoja kunnes 2.1.0 -versio on valmis. On siis hyvin mahdollista, että tässä projektissa tuotettua tuotetta on muutettava, jotta se vastaa 2.1.0 -version vaatimuksia. Valmis ohjainkoodi jakautuu kahteen osaan, joista ensimmäinen on xml -tiedosto ja toinen js -tiedosto. Xml -tiedosto sisältää kaikki käytössä olevat MIDI -viestit ja niihin liitetyt Mixxx -toiminnot. Toimintoja on laajennettu js -tiedostoon, missä tapahtuu monitila -ja led -ohjaus.

Xml -ja js -tiedostot on myös tallennettava oikeaan hakemistoon. Tuo hakemisto sijaitsee valitussa asennuskohteen alihakemistossa nimeltään 'controllers'. Tästä kohteesta saadaan käyttöön MIDI-ohjaimelle kirjoitettu ohjainasetukset Mixxx:n MIDI -asetuksista. Tämä hakemisto ei kuitenkaan ole otollinen paikka kirjoittaa keskeneräistä ohjelmaa, koska silloin se jouduttaisiin jokainen kerta ottamaan käyttöön erikseen. Mixxx onkin suunniteltu ja rakennettu siten, että se ei käytä asennuspolussa sijaitsevan 'controllers' -hakemiston tietoja suoraan, vaan kopioi tarpeellisen datan alla olevaan sijaintiin ja käyttää tietoa sieltä käsin./8/

Windows: C:\Users\\AppData\Local\Mixxx\controllers

GNU/Linux: home/<username>/mixxx/controllers

OSX: Users/<username>/Library/Application Support/Mixxx/controllers

Tiedostoja tallentaessa on huomattava myös tallennusnimessä tapahtuva lisäys tiedostopäätteessä, mikä vaikuttaa sen tulkintatapaan. MIDI -ohjainkoodi on tallennettava muotoon 'tiedostonimi.midi.xml' tai tiedostonimi.hid.xml, mikäli kyseessä on jokin HID -laite Tässä projektissa on kyse MIDI -ohjaimesta, joten valinta on midi.xml.

5.1 MIDI -viesteistä Mixxx -toimintoihin

Yksi Gemini -laitteista on kytketty tietokoneeseen ja määritetty MIDI -kanavalle yksi. Tämän jälkeen on painettu play-nappia. Laitteen lähettämät MIDI -viestit näyttävät siis seuraavilta.

0x91, 0x02, 0x7F ja 0x91, 0x02, 0x00

Kaksi viestiä, koska nappi painetaan alas ja päästetään vapaaksi. Tätä toimintoa ja Gemini -laitteen kiekon kiertoa varten luotu midi.xml -tiedosto (**Kuvio 9.**) on jaettu eri osiin.

```

<?xml version='1.0' encoding='utf-8'?>
<MixxxControllerPreset mixxxVersion="2.0.0+" schemaVersion="1">
  <info>
    <name>Gemini MDJ-1000</name>
    <author>Joakim Mattson</author>
    <description>Our mighty Controller code</description>
    <forums>none</forums>
  </info>
  <controller id="gemini">
    <scriptfiles>
      <file functionprefix="GeminiMDJ1000" filename="Gemini-MDJ-1000-script.js"/>
    </scriptfiles>
    <controls>
      <control>
        <group>[Channel11]</group>
        <key>play</key>
        <status>0x91</status>
        <midino>0x02</midino>
        <options>
          <normal/>
        </options>
      </control>
      <control>
        <group>[Channel14]</group>
        <key>GeminiMDJ1000.WheelTurn</key>
        <status>0xB3</status>
        <midino>0x05</midino>
        <options>
          <script-binding/>
        </options>
      </control>
    </controls>
  </controller>
</MixxxControllerPreset>

```

Kuvio 9. Play -ja kiekkokosketuksen xml.

Ohjauskoodi alkaa niin sanotulla otsikoinnilla (Kuvion 9 osa A), minkä alussa määritetään mihin Mixxx:n versioon ohjelma on suunniteltu. Otsikointiin lisätään myös vähintään ohjelman nimi, tekijä ja kuvaus. Jos samassa tiedostossa olisi useamman ohjaimen MIDI -viestitiedot, tulisi controller -osion id mainita vain kerran kutakin laitetta kohden. B -osiossa tulee määrittellä, missä JavaScript-tiedosto sijaitsee ja kertoa myös funktion esimääritelmä, mikä sisältää init -ja shutdown -metodien kutsut, kun ohjain avataan tai suljetaan Mixxx:n toimesta. /6/

Controls -osion sisälle sijoitetaan ne control -toiminnot mitä halutaan Gemini -laitteen tekevän. Control -osiossa määritellään mitä tehdään kyseisen MIDI -viestin saavuttua. Controls -osiota kutsutaan myös nimellä Input (sisääntulo).

Group -tagi kertoo mitä Mixxx:n osiota halutaan muuttaa. Näitä toimintoja on useita, kuten [Master], [Playlist], [Channel1] ja [Channel2]. Key -tagi kertoo mitä Mixxx:n toimintoa halutaan muuttaa. Näitä toimintoja on useita, kuten play, cue_default ja back. Tarkempi taulukko Mixxx:n toiminnoista löytyy Mixxx:n wiki -sivustolta. Status -tagiin määritellään tulevan MIDI-viestin kanava. Midino -tagiin määritellään tulevan MIDI -viestin ensimmäinen datatavu, mikä kertoo mitä nappia on painettu. Mixxx on suunniteltu sillä tavalla, ettei xml -tiedostoon tarvitse määrittellä MIDI -viestien toista dataosaa, koska xml -tiedostossa olevilla controlilla ainoastaan vaihdetaan halutun Mixxx -toiminnon tilaa. Miksi sitten tilaa ei vaihdeta kahdesti, jos yksi napin painallus luo kaksi MIDI -viestiä? Siksi, koska control -osioon kuuluu vielä options -tagi. Tässä projektissa options -tagin sisällöksi tulee 'normal', ellei kyseinen 'control' ole liitettyä JavaScriptin metodiin. Siinä tapauksessa optionsin tilaksi tulee 'Scrip-Binding'. Normal -tilassa ei tehdä muutoksia MIDI:n toisen datatavun arvon ollessa nolla. Tämä vastaa samaa kuin nappia nostettaisiin. Muussa tapauksessa arvo tulkitaan napin painallukseksi alas. Script-Binding -tila liittää MIDI -viestin sisällön key -tagissa olevaan ohjelmakoodiin halutun funktionimen avulla, mikä löytyy JavaScript -tiedostosta. Tällöin siirrytään suorittamaan JavaScript -ohjelman tiettyä metodia. Controls -osion jälkeen on myös mahdollista luoda output -osio, mutta tässä projektissa suoritetaan uloslähtevät MIDI-viestit JavaScriptillä. /6/

Jotta JavaScript olisi toimiva, tulee se aloittaa olion esittelyllä. Se näyttää seuraavalta: `function GeminiMDJ1000() {}`. Huomattavaa on, että funktion nimi tulee olla sama kuin xml -tiedostossa esimääritetyn 'functionprefix' -kohdan. `//`

Tällä oliolla tulisi olla funktiot nimeltään `init` ja `shutdown`. Ne voivat olla tyhjä, mutta ovat hyödyllisiä asettamaan Gemini -laite tiettyyn tilaan, ohjaten esimerkiksi ledit tiettyyn tilaan toimintoja aloitettaessa tai ohjelmasta poistuttaessa. `//`

Tämän projektin `init`-funktiossa lähetetään neljään ensimmäiseen MIDI-kanavan kaikkiin mahdollisiin data 1 -tavuihin tila `7FH`. Tämän jälkeen käydään läpi ennalta määritetyt `Mixxx` -toiminnot, mitkä yhdistetään yksinkertaiseen ledi-funktioon. Funktiolla `engine.connectControl()` -yhdistetään `Mixxx:n` toiminnot yksi kerrallaan `for` -silmukan avulla (**Kuvio 10**). Esimerkkinä `connectControl("[Channel1]", "[play_indicator]", GeminiMDJ1000.Simple)`. Joka kerta, kun ensimmäisen kanavan 'play_indicator' vaihtaa arvoaan, kutsutaan myös samalla funktiota `GeminiMDJ1000.SimpleLedControl()`. `Shutdown` -funktio toimii käänteisesti. Lähetetään kaikkiin neljään ensimmäiseen MIDI -kanavaan käsky sammuttaa ledit. `Enige.triggerilä` päivitetään `Mixxx-` toiminnon tila siihen, mikä se jo on. Täten saadaan varmistettua, että yhdistetty toiminto on päivitetty ajan tasalle.

```
for(var n = MixxxChannels.length-1; n >=0; n-- ) {
  for(var i = MixxxControls.length; i >= 0; i--) {
    engine.connectControl(MixxxChannels[n], MixxxControls[i], "GeminiMDJ1000.SimpleLedControl");
    engine.trigger(MixxxChannels[n], MixxxControls[i])
  }
}
```

Kuvio 10. `Init` -funktion osa

`MixxxChannels` -taulukko sisältää kaikki `Mixxx:n` kanavat, mitkä ovat `[Channel1]`, `[Channel2]`, `[Channel 3]` ja `[Channel4]`. `MixxxControls` -taulukko puolestaan on hiukan isompi listattavaksi tähän. Se sisältää kaikki ne `Mixxx:n` toiminnot, mitä käytetään yksinkertaiseen lediohjaukseen, esimerkkinä `'loop_in'`, `loop_out'` ja `play_indicator`.

SimpleLedControl -funktio on nimensä mukainen, sillä se lähettää vain uloslähtevän MIDI -viestin (**Kuvio 11**). Viestin sisältö on täysin riippuvainen siitä, mitä arvoja sille syötetään value, control ja group -muuttujiin. On otettava myös huomioon oliot 'GeminiMDJ1000.ChannelMessages' ja 'GeminiMDJ1000.LedControl'. Group -muuttujalla haetaan ChannelMessages -oliosta Mixxx -kanavaa vastaava MIDI -kanava. Control -muuttujalla haetaan LedControl -oliosta Mixxx -toimintoa vastaava MIDI -toiminto. Valuen arvolla määritetään missä tilassa ledin tulee olla (päällä tai pois). Muuttujien arvot tulevat suoraan Mixxx:stä funktiota kutsuttaessa.

```
GeminiMDJ1000.SimpleLedControl = function(value, group, control){
    midi.sendShortMsg(GeminiMDJ1000.ChannelMessages[group], GeminiMDJ1000.LedControl[control], value * GeminiMDJ1000.on);
}
```

Kuvio 11. Yksinkertainen ledohjaus

Mainittakoon vielä, että Mixxx:n rate -toiminnon valinta tulee määrittää käänteiseksi, koska muutoin Gemini -laitteen liukua säädettäessä ylös, meneekin Mixxx:n tempokuvaaja alas.

5.2 Monitilatoiminnot

Ohjelmassa on kolme mutkikkaampaa toimintoa, mitkä määräävät kuinka joku toinen toiminto käyttäytyy. Yksi niistä on SlipEnabled -funktio, millä muutetaan vain ohjelman sisäistä julkista muuttujaa nimeltään GeminiMDJ1000.SlipEnabled. Toinen toiminto on LoopRollEnabled -funktio, missä muutetaan julkista GeminiMDJ1000.LoopRollEnabled -muuttujan tilaa. Kolmas on VinylModeEnabled -funktio, missä muutetaan julkista GeminiMDJ1000.VinylModeEnabled -muuttujan tilaa. Kaikkien kolmen muuttujan tila tulee olla joko tosi tai epätosi.

Helpoin tapa ymmärtää sliptilan käyttö on yhdistää se ReverseRoll -toiminnon kanssa. Kappale jatkaa soittoa taustalla äänettömänä ja yhtä aikaa toistetaan kappaletta takaperin siitä kohtaa, kun nappia on painettu. Kun sliptila otetaan pois päältä, jatkuu kappaleen toisto siitä missä se oikeasti olisi./9/ Tämä tapahtuu siis silloin, kun käyttäjä painaa Gemini -laitteesta reverse -nappia riippuen siitä, missä

tilassa slip on. Sliptilan ollessa päällä, käytetään ReverseRoll -toimintoa. Sliptilan ollessa pois päältä, käytetään normaalia Reverse -toimintoa. Sliptilaa tarkastellaan myös BeatLoopControl- funktiossa, missä toistetaan kappaleen osia silmukassa niin pitkään kuin käyttäjä tahtoo. Kun suoritetaan scratch -toimintoa, sliptila ei vaikuta sen suoritukseen.

Vinyylitilalla on vain yksi merkitys tässä projektissa. Sillä määritetään, onko kyseessä scratch -toiminto vai jog-toiminto. Jog -toiminnolla vaikutetaan väliaikaisesti kappaleen soitt nopeuteen ja suuntaan. Vinyylitilan ollessa tosi, käytetään scratch2 -toimintoa, mikä ottaa kappaleen toiston täydellisen hallinnan käyttäytyen kuin vinyylilevy. /9/ Toki tähän scratch -tilaan päästään ainoastaan koskemalla itse levyä, mikä asettaa Mixxx:n valitun kanavan scratch2_enable -toiminnon tilan päälle tai pois, riippuen kosketaanko siihen tai ei.

LoopRollEnabled -funktiolla sytytetään myös Gemini -laitteen silmukkanappien valot tai sammutetaan ne. Mikäli siirrytään LoopRolltilaan, sytytetään kaikki silmukkaedit. Tämä kertoo käyttäjälle missä tilassa ohjelma on. LoopRoll -toiminto tarkoittaa tavallista silmukointia yhdistettynä sliptilaan. Sliptilaa ei kuitenkaan jätetä päälle, vaan sitä käytetään vain sen ajan kun looproll on päällä.

Ohjelmakokoelmasta (LIITE 2) löytyy vielä funktio tempoliu'un määrän muuttamiseksi. Jo init -funktion aikana määritellään tempoliu'un prosentuaalinen vaikutusala, mikä alkaa 4%:sta seuraavassa järjestyksessä. 4 %, 8 %, 16 %, 24 %, 50 %, 100 %.

Kerrottakoon vielä, että JavaScriptin funktiot ottavat vastaan XML -tiedoston puolella tallennetut tiedot aina samassa järjestyksessä riippumatta miten uudet muuttujat nimetään. Jos haluttaisiin käyttää vain group- muuttujan sisältämää merkkijonoa, mikä on siis käytännössä "[Channel1]", tai "[Channel2]" ja niin edelleen, on merkittävä sitä edeltävät muuttujat myös, eli channel, control, value ja status. /7/

5.3 Uudelleen määritettäviä toimintoja

Projektin aikana todettiin joidenkin toimintojen olevan tarpeettomia tai niitä ei voida mallintaa Mixxx:lle, koska niille ei ole vastaavaa toimintoa Mixxx:ssä. Taulukossa 2 on uudelleen määritelty nämä toiminnot.

Taulukko 2. Uudelleen määritettävät toiminnot..

Sijainti kuviossa 8	Nimitys laitteessa	MDI-viesti	Toiminnon uusi määritys
5	Track Search [Time]	9x, 0FH [2EH]	Jää tyhjäksi.
5	Track Search [Text]	9x, 10H [2EH]	Jää tyhjäksi.
6	[Wave]	9x, [33H]	Jää tyhjäksi.
14	[Select]	9x, [31H]	Jää tyhjäksi.
15	Back (Settings)	9x, 0AH	Jää tyhjäksi.
18	Jog start time	Bx, 02H	Jää tyhjäksi.
19	Jog stop time	Bx, 03H	Jää tyhjäksi.
21	[Vinyl]	9x, [32H]	Jää tyhjäksi.
22	[Slip mode]	9x, [32H]	Jää tyhjäksi.
23	BPM Tap [Lock]	9x, 16H [17H]	beat_translate_cur pos [quantize]
24	[Auto Cue]	9x, [1CH]	Jää tyhjäksi.
25	[BPM Mode]	9x, [15H]	Jää tyhjäksi.

6 TOIMINNAN TOTEAMINEN

Ensimmäinen livetestausta suoritettiin 8.3.2017 Vaasassa sijaitsevassa Tropiclandia-kylpylässä, missä järjestettiin Snöörit nimisen Youtube -sarjan viimeisen jakson kuvauksiin liittyvät allasjuhlat. Toinen livetesti järjestettiin samassa paikassa 6.4.2017. Tällöin kyseessä oli Warrantti ry:n järjestämä Wykypäivä -allasbileet. Ensimmäisen testauksen yhteydessä ei ehditty havainnoimaan valmistuksessa tapahtuneita virheitä. Niinpä aikaa kulutettiin dokumentointiin odotellessa seuraavaa esiintymistä. Wykypäivän aikana huomattiin muutama esteettinen ongelma.

6.1 Testilaitteisto

Testien aikana käytettiin sellaista laitteistoa, mitä voidaan käyttää missä tahansa Gemini -laitteen ja Mixxx:n välisen kommunikoinnin toteuttamisessa. Tämän projektin aikana käytettiin kuitenkin seuraavan tapaista laitteistoa:

- kannettava PC (Ubuntu Studio 14.04.3 LTS).
- 2x Gemini MDJ-1000
- ulkoinen äänikortti
- ulkoinen klubimikseri.

Tietokoneeseen yhdistetään Gemini -laitteet USB -kaapelilla. Kytetään virta laitteisiin, mutta ennen siirtymistä MIDI -tilaan. Ohjainlaitteelle tulee kertoa missä MIDI -kanavassa sen tulee toimia. Gemini-laitteen asetuksista valitaan ”MIDI Channel”. Ensimmäisen pöydän ohjaimen tulee olla MIDI -kanavassa nolla, toisen pöydän MIDI -kanavassa yksi. Tämä siksi, koska nolla on myös numero. Lopulta kyseessä on vain yhden tavun mittaisesta binäärijonosta, millä voidaan osoittaa 16 eri tilaa. Valmis koodi voidaan myös lopulta sijoittaa järjestelmän karttaohjainhakemistoon. Hakemisto sijaitsee eri käyttöjärjestelmissä eri paikassa.

/6/

GNU /Linux: /usr/share/mixxx/controllers tai /usr/local/share/mixxx/controllers

OS X: /Applications/Mixxx.app/Contents/Resources/controllers/

Windows: C:\Program Files\Mixxx\controllers

Mixxx -ohjelman avauduttua, tulee asetuksista määrittää käytettäväksi äänilaitteeksi ulkoinen äänikortti. Tässä projektissa käytetään kahta Gemini-laitetta, joilla ohjataan kahta eri dekkiä. On siis määritettävä Deck 1 ja Deck 2:n ulostulot äänikortin neljään valittuun kanavaan. Mixxx:n ääniasetuksissa on esitetty asetukset käyttäen Esi Gigaport HD + -äänikorttia (**Kuvio 12.**).

Output	Input
Master	None
Headphones	None
Left Bus	None
Center Bus	None
Right Bus	None
Deck 1	Speakers (2- GIGAPort HD+), Channels 1 - 2
Deck 2	Speakers (2- GIGAPort HD+), Channels 3 - 4
Deck 3	None
Deck 4	None

Kuvio 12. Mixxx-ääniasetukset.

Controller Setup | Input Mappings | **Output Mappings** | Scripts

gemini MDJ-1000 audio MIDI Controller

Enabled

Learning Wizard (MIDI Only)

Load Preset: ...

- Denon DN SC2000
- Denon MC3000 v0.995
- Denon MC6000MK2
- Electrix Tweaker
- Evolution X-Session
- FaderFox DJ2
- Gemini CDMP-7000 L
- Gemini CDMP-7000 R
- Gemini FirstMix
- Gemini MDJ-1000**

Preset Info

Name:

Author:

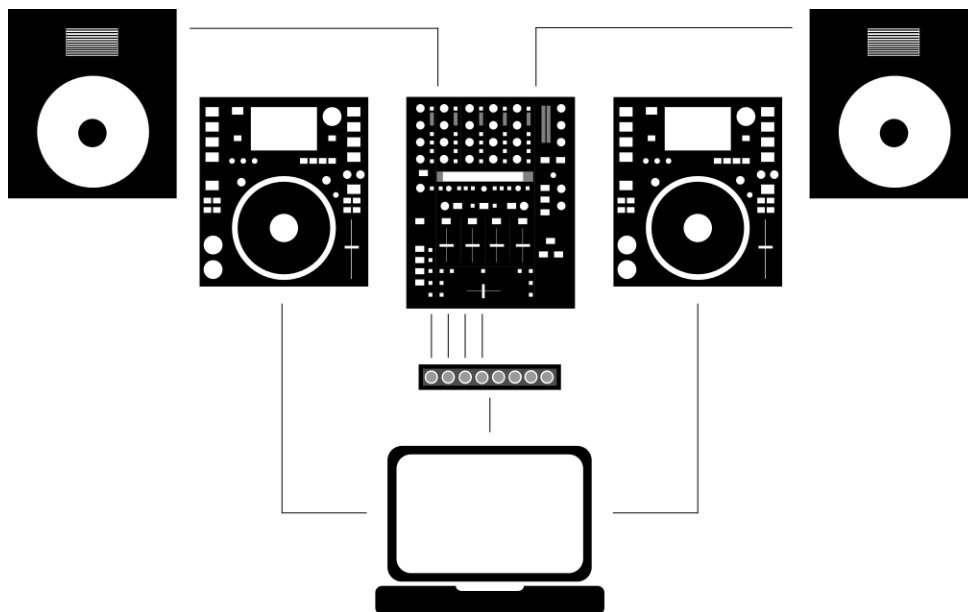
Description:

Support:

Kuvio 13. Mixxx-ohjainasetukset.

Asetuksissa tulee määrittää vielä kullekin Gemini-ohjaimelle oma MIDI-kartoitus, mitä käyttää. Mixxx:n ohjain asetuksissa on esitetty Windowsin tunnistama Gemini-laite, jolle ollaan asettamassa sille ohjelmoitua MIDI-karttaa (**Kuvio 13.**).

Laitteiston yleiskuvassa on esitetty karkeasti millaisesta kytkennästä on kysymys (Kuvio 14.). Kannettavaan on yhdistetty kaksi Gemini -laitetta sekä ulkoinen äänikortti. Äänikortista ulostuleva äänisignaali kulkeutuu klubimikserille signaalille valittuun kanavaan. Mikserillä voidaan lopulta sekoittaa soitettavaa ääntä mielivaltaisesti. Mikseristä uloslähtevä pääsignaali jaetaan lopulta kaiutinjärjestelmään.



Kuvio 14. Laitteiston yleiskuva.



Kuvio 15. Mixxx-käyttöliittymä

Mixxx:n käyttöliittymästä (Kuvio 15) on riisuttu tarpeettomat lisäikkunat pois. Vektoripohjaisen ohjelmoinnin ansiosta näkymä venyy oikeassa suhteessa kuvaruudun kanssa ja saadaan käytettyä kaikki tila hyväksi. Loppujen lopuksi Mixxx toistaa vain kappaleita. Vasemmalla puolella on pöytä 1 (tai kanava 1) ja oikealla puolella pöytä 2 (tai kanava 2). Nämä pöydät on nyt ohjattu ulostuloksi äänikortin neljään ensimmäiseen kanavaan. Gemini -laitteen napit ovat siis lopulta kytköksissä tämän käyttöliittymän toimintoihin. Tarkastellaan hieman tarkemmin mitä Mixxx:n toimintoja käytetään.



Kuvio 16. Mixxx-kanava-asetukset.

Mixxx:n kanava asetuksissa (**Kuvio 16.**) on kuusi erilaista toimintoa joista kuitenkin vain neljä on otettu käyttöön, aloittaen vasemmasta ylänurkasta kulkien oikealle toiminnot ovat seuraavat:

1. Näytä/piilota vinyylikiekkko. Ei ohjelmoida
2. Toista kappale uudestaan sen loputtua. Ei ohjelmoida.
3. Poista kappale kyseiseltä pöydältä. Ohjelmoitu Gemini -laitteen eject -nappiin.
4. Aseta rytmiviiva sen hetkiseen vallitsevaan soittosijaintiin. Ohjelmoitu Gemini -laitteen BPM tap -nappiin.
5. Magnetisoi kaikki hyppymerkit, toistosilmukat alkavat rytmiviivalta. Ohjelmoitu Gemini -laitteen Lock -nappiin.
6. Lukitsee sävellajin kappaleen alkuperäiseen sävellajiin. Ohjelmoitu Gemini - laitteen Keylock -nappiin.



Kuvio 17. Mixxx-toisto- ja liikutustoiminnot.

Mixxx:n toisto- ja liikutustoiminnot (**Kuvio 17.**) luetellaan alkaen vasemmalta yltäältä myötäpäivään:

1. Pikakelaus taakse. Ohjelmoitu Gemini -laitteen vasemmanpuoleiseen Search -nappiin.
2. Pikakelaus eteen. Ohjelmoitu Gemini -laitteen oikeanpuoleiseen Search -nappiin.
3. Aloitusmerkintä. Ohjelmoitu Gemini -laitteen Cue -nappiin.
4. Takaperin toisto. Monikäyttöohjelmoitu Gemini -laitteen Rev -nappiin.
5. Toisto/Tauko. Ohjelmoitu Gemini -laitteen Play -nappiin.



Kuvio 18. Mixxx-tempoliukuvipu.

Sävellajin säätöä (**Kuvio 18.**) ei ohjelmoitu, koska sitä käytetään hyvin harvoin sen takia, että se vaikuttaa äänenlaatuun ei-halutulla tavalla. Iso liukuvipu esittää kappaleen soittonopeuden muuttujaan. Nopeusliuku on ohjelmoitu Gemini -laitteen vastaavaan liukusäätimeen. Seuraavat toiminnot alkavat minus -merkistä.

1. Vähennä nopeutta. Ohjelmoitu Gemini -laitteen Minus -nappiin.
2. Lisää nopeutta. Ohjelmoitu Gemini -laitteen Plus -nappiin.
3. Väliaikaisesti vähentää nopeutta. Ei ohjelmoida.
4. Väliaikaisesti lisää nopeutta. Ei ohjelmoida.
5. Synkronisointi. Synkronisoidaan tempo viereisen kanavan mukaiseksi. Ohjelmoitu Gemini -laitteen Sync- nappiin.



Kuvio 19. Mixxx-silmukkatoiminnot.

Silmukkatoiminnot (**Kuvio 19.**) vasemmalta oikealle toiminnot ovat seuraavat:

1. Miinus -merkki. Puolittaa silmukan pituuden. Ohjelmoitu Gemini -laitteen (/2) -nappiin.
2. 1/8. Aktivoi 1/8 tahdin pituisen silmukan. Ohjelmoitu Geminin 1/8 -nappiin.
3. 1/4. Aktivoi 1/4 tahdin pituisen silmukan. Ohjelmoitu Geminin 1/4 -nappiin.
4. 1/2. Aktivoi 1/2 tahdin pituisen silmukan. Ohjelmoitu Geminin 1/2 -nappiin.
5. 1. Aktivoi 1 tahdin pituisen silmukan. Ohjelmoitu Geminin 1 -nappiin.
6. 2. Aktivoi 2 tahdin pituisen silmukan. Ohjelmoitu Geminin 2 -nappiin.
7. 4. Aktivoi 4 tahdin pituisen silmukan. Ohjelmoitu Geminin 4 -nappiin.
8. 8. Aktivoi 8 tahdin pituisen silmukan. Ohjelmoitu Geminin 8 -nappiin.

9. 16. Aktivoi 16 tahdin pituisen silmukan. Ohjelmoitu Geminin 16 -nappiin.
10. Plus -merkki kaksinkertaistaa silmukan pituuden. Ohjelmoitu Gemini -laitteen (x2) -nappiin.
11. Silmukka sisään. Asetetaan silmukka alkamaan mielivaltaisesta pisteestä. Ohjelmoitu Gemini -laitteen In -nappiin.
12. Silmukka ulos. Asetetaan silmukka loppumaan mielivaltaisesta pisteestä. Ohjelmoitu Gemini -laitteen Out -nappiin.
13. Loop. Aktivoidaan viimeisin silmukka tai vapautetaan silmukka.



Kuvio 20. Mixxx-hyppypisteet.

Hyppypisteet on ohjelmoitu Gemini-laitteen vastaaviin hyppypiste -nappeihin. Hyppypisteiden avulla voidaan siirtyä soitettavassa kappaleessa ennaltamääritettyyn kohtaan.

6.2 Ohjelmoinnin yhteydessä tapahtuva testaus

Ohjelmoinnin yhteydessä tapahtuva testaus ja ohjelmointi aloitettiin yhdistämällä kaikki Gemini -laitteessa olevat ledit niitä vastaavaan Mixxx -toimintoon. Ledejä sytyteltiin suoraan hiirellä klikkailemalla niitä Mixxx:n toimintoja, mihin ne liitettiin. Seuraavaksi kirjoitettiin XML -tiedostoon kaikkien neljän kanavan MIDI -viestit. Tämä tiedon tallentaminen vei paljon aikaa, koska itse Mixxx jouduttiin aina käynnistämään uudelleen, mikäli XML -tiedostoon tehtiin muutoksia, jotta muutokset tulisivat voimaan. Jokaisen Gemini -laitteen napin MIDI -viestit ohjelmoitiin ainoastaan käyttäen yhtä MIDI -kanavaa, koska lopputulos on aina sama kaikilla neljällä kanavalla. Viesti taulukkoon (Liite 1) kerätyssä aineistossa mainitaan kaikki MIDI -viestit joita Gemini -laitteesta saadaan. Kun Mixxx:n ensimmäisen kanavan ohjaus saatiin toimimaan, kopioitiin se koodin osa, mikä juuri kirjoitettiin ja liitettiin takaisin lähdekoodiin kolmesti.

Näistä kolmesta uudesta rivistä vaihdettiin group -ja midinoarvot seuraavia kanavia osoittaviksi. Näin saatiin ohjelmoitua kaikki neljä kanavaa nopeasti. Tämä todettiin vasta paljon myöhemmin, että toimiiko ohjaus todella kaikilla neljällä kanavalla.

Seuraavana ohjelmointitestaukseen otettiin Reverseroll. Reverseroll kuuluu sliptilan alaiseen toimintaan. Siispä reverse -nappia painettaessa ajetaankin suoraan JavaScript Funktio, mikä tarkastelee sliptilaa. Myös itse slip -nappi suorittaa JavaScriptin avulla oman funktion, millä sliptilan lippu saadaan aktivoitua. Sliptilaan siirtyminen testattiin siten, että kun sliptila on aktivoitu, syttyy slip -napin led. Itse Mixxx-ohjelman toiminnoissa ei tapahdu mitään. Käyttäjälle kerrotaan vain, että olet vaihtanut slip -aktivoitumuuttujan arvoksi tosi. Kun sliptila on aktivoitu, voidaan samaa MIDI -viestiä käyttää kahteen tarkoitukseen, reverse tai reverserolliin. Samaa menetelmää käytettiin automaattisilmukoinnin ohjaukseen. Kaikki kahdeksan MIDI -viestiä yhdistettiin JavaScript -funktiioon BeatloopControl, missä tarkastellaan kahden lipun tilaa. Ensimmäinen ja tärkeämpi näistä on LoopRollEnabled ja toinen SlipEnabled. LoopRoll on kuin ReverseRoll, mutta kyseessä onkin silmukatoistovieritys. Ollessa LoopRoll -tilassa, SlipEnabled -muuttujaa ei tarvitse tarkastella, koska vieritys on sama asia kuin sliptilassa oleminen. Erona näillä on se, että vieritys tapahtuu LoopRoll -tilassa sen aikana, kun Gemini -laitteen nappia pidetään painettuna alas. Sliptilassa aktivoitessa silmukka, siirrytään sliptilaan vasta, kun silmukka alkaa ja poistutaan, kun silmukka loppuu. Tämä tapahtui oletetulla tavalla.

Tempo -vivulle ohjelmoitiin myös funktio, millä saatiin muutettua sen vaikutus- aluetta. Tässä funktiossa siirrytään yksiulotteisessa taulukossa sen seuraavaan arvoon, kunnes se alkaa jälleen uudestaan. Tätä arvoa päivitetään samassa funktiossa valitun kanavan tempoalueeseen.

Siirryttäessä vinyyli -tilaan käytetään samaa metodia kuin sliptilaan siirryttäessä. Eli ainoastaan muutetaan VinylModeEnabled -muuttujan arvoa ja vinyl -napin ledin tilaa. Gemini -laitteen kiekon kosketus -funktio on ainoa funktio JavaScriptissä, mikä tarkastelee VinylModeEnabled -muuttujan tilaa. Mixxx:n

skrätssäys on luotu hyvin yksinkertaiseksi. Skrätssäys -tila tulee asettaa ensin päälle, minkä jälkeen voidaan ohjata toisella toiminnolla siirtymistä kappaleen aikajanalla suntaan tai toiseen. VinylModeEnabled -muuttujan arvon ollessa tosi, sallitaan skrätssäys -tilaan siirtyminen.

6.3 Live-testaus

Alkuperäinen suunnitelma oli testata tuotetta viimeistään Wykypäivänä, mutta sitä ennen tarjoutui tilaisuus Snöörien allasjuhliin. Ohjelma ehdittiin valmistaa hyvissä ajoin ja harjoittelullekin jäi rutkasti aikaa. Kuitenkaan Snöörien juhlassa ei havaittu muita puutteita kuin tempo liu'un suuruuden muutos. Tätä muutosta ei pystytty seuraamaan missään muulla tavalla kuin asettamalla liukuvipu ääriasentoon. Tempomuuttujan suuruutta voidaan kuvata ainoastaan jollain toisella laitteella. Koska Gemini MDJ-1000:tta ei ole suunniteltu alkuunkaan Mixxx:lle, on siis tehtävä kompromisseja. Suuruuden muutostoiminto saa jäädä. Vastuu siirtyy napin käytön jälkeen käyttäjälle.

Wykypäivänä havaittiin toinen puute, mikä koskee rytmiviivan siirtämistä. Aiemmin rytmiviivaa sirrettiin jollain toisella ulkoisella enkooderilla. Gemini -laitteessa kyseiselle toiminnolle ei ole varattu omaa nappia sitä varten. Gemini-valmistajan ratkaisu rytmiviivan siirtämiselle omassa ohjelmistossaan on siirtää soittoviiva siihen ajanhetkeen mihin viiva halutaan ja painamalla 'shift' + 'Cue'. Mutta cue -napille ei ole shift -toimintoa MIDI -tilassa lainkaan, joten päätettiin varata rytmiviivan määrittelyyn oma nappi. Koska magnetisointi ja rytmiviivan siirto ovat aiheellisesti lähellä toisiaan, päätettiin asettaa rytmiviivan napiksi Gemini -laitteen BPM tab -nappi.

Koodi toimi muutoin täydellisesti juuri niin kuin sen pitikin. Ylimääräistä dataa ei kulkenut laitteiden välillä. Virheitä ei tapahtunut koko 5,5 tunnin aikana. Ilta saatiin päätökseen ilman ongelmia laitteiden välillä.

7 YHTEENVETO

Projektin tuloksena toteutetun tuotteen suunnitelma alkoi jo kauan ennen aiheen valintaa, mikä helpotti suunnittelua. Projekti oltaisiin toteutettu joka tapauksessa. Projektin kannalta tuotos on ainutlaatuinen, koska kukaan ei ole sitä tehnyt aiemmin, ainakaan julkisesti. Valitun DJ -ohjelman ja Gemini -laitteen välille ei olla koskaan ennen tätä projektia julkaistu rajapintaa. Tämän rajapinnan olemattomuus johtunee kenties siitä, että se on tavallaan tarpeeton. Gemini MDJ-1000:n ollessa nimenomaan multimediasoitin voitaisiin sillä toistaa musiikkia joko USB -massamuistista tai CD -levyltä. Laitteen MIDI -toiminto on siinä mielessä hieno, että sillä voidaan ohjata jotain toista laitetta niin halutessa, kuten tässä projektissa ohjataan DJ -ohjelmaa nimeltään Mixxx.

LÄHTEET

- /1/ Heckroth J. 2014. The complete midi 1.0 detailed specification. MIDI Manufacturers Association. Los Angeles, California. Viitattu 20.8.2017.
<https://www.midi.org/specifications/item/the-midi-1-0-specification>
- /2/ Quin L. 2016. Extensible Markup Language. World Wide Web Consortium. Viitattu 20.8.2017
<https://www.w3.org/XML/>
- /3/ Nykänen O. 2003. XML 10 kohdan tiivistelmänä. W3C Suomen toimisto. Viitattu 20.8.2017
<http://www.w3c.tut.fi/translations/xml/xmlin10pts/>
- /4/ Wilton-Jones M. 2003. JavaScript history. Viitattu 20.8.2017
<http://www.howtocreate.co.uk/jshistory.html>
- /5/ 2005. Gemini MDJ-1000 User Manual. Innovative Concepts and Design LLC. Viitattu 20.8.2017
http://joakim-mattson.fi/manuals/gemini_mdj-1000_usermanual.pdf
- /6/ Dashuer. 2016. MIDI Controller Mapping File Format. Viitattu 20.8.2017
http://mixxx.org/wiki/doku.php/midi_controller_mapping_file_format
- /7/ Ronso. 2017. Controller Scripting. Viitattu 20.8.2017
http://mixxx.org/wiki/doku.php/midi_scripting
- /8/ be.ing. 2016. Controller Mapping File Locations. Viitattu 20.8.2017
http://mixxx.org/wiki/doku.php/controller_mapping_file_locations
- /9/ be.ing. 2017. Mixxx Controls. Viitattu 20.8.2017
<http://mixxx.org/wiki/doku.php/mixxxcontrols>
- /10/ sblaisot. 2016. Updating Controller Configurations for 2.0. Viitattu 20.8.2017
https://www.mixxx.org/wiki/doku.php/updates/controllers_for_2.0
- /11/ The Mixxx Development Team. 2016. An Overview of the Mixxx Interface. Viitattu 20.8.2017
https://mixxx.org/manual/latest/chapters/user_interface.html

LIITE 1

REF NO.	I/O	FUNCTION	STATUS (channel)	DATA1 (function code)	DATA2 (action)	Mixxx Group	Mixxx Control(key)	
		[When shift is enabled]						
		[When shift is enabled]					GeminiMDJ1000.x = javascript function	
1	O	Touch	9x, X=0-3	03H	7FH: ON / 00H:OFF	[ChannelN], N = 1-4	GeminiMDJ1000.WheelTouch	
1	O	Jogwheel	BX	05H	7FH: TURN Clockwise FAST ~ 41H: TURN Clockwise SLOW / 3F:TURN CounterClockwise SLOW ~ 01H: TURN CounterClockwise FAST	[ChannelN]	GeminiMDJ1000.WheelTurn	
2	I/O	Play / Pause	9X	02H	7FH: ON / 00H:OFF	[ChannelN]	play	
3	I/O	Cue	9X	01H	7FH: ON / 00H:OFF	[ChannelN]	cue_default	
4	O	Search [Sync]	9X	0DH [34H]	7FH: ON / 00H:OFF	[ChannelN]	back [sync_enabled]	
4	O	Search [Master]	9X	0EH [35H]	7FH: ON / 00H:OFF	[ChannelN]	fwd [sync_master]	
5	O	Track search [Time]	9X	0FH [2EH]	7FH: ON / 00H:OFF			Empty
5	O	Track search [Text]	9X	10H [1BH]	7FH: ON / 00H:OFF			Empty
6	I/O	Reverse [Wave]	9X	14H [33H]	7FH: ON / 00H:OFF	[ChannelN]	reverse/reverseroll (when slip is enabled)	Half used
7	-	Jog adjust	-	-	-			
8	I/O	In [2]	9X	04H [07H]	7FH: ON / 00H:OFF	[ChannelN]	loop_in [loop_half]	
8	I/O	Out /Adjust [x2]	9X	05H [08H]	7FH: ON / 00H:OFF	[ChannelN]	loop_out [loop_double]	
8	O	Reloop / Exit [Loop Roll]	9X	06H [2FH]	7FH: ON / 00H:OFF	[ChannelN]	reloop_exit [GeminiMDJ1000.LoopRollEnabled]	
9	I/O	Hot cue 1 [Delete]	9X	26H [2AH]	7FH: ON / 00H:OFF	[ChannelN]	hotcue_1_activate [hotcue_1_clear]	
9	I/O	Hot cue 2 [Delete]	9X	27H [2BH]	7FH: ON / 00H:OFF	[ChannelN]	Hotcue_2_activate [hotcue_2_clear]	
9	I/O	Hot cue 3 [Delete]	9X	28H [2CH]	7FH: ON / 00H:OFF	[ChannelN]	Hotcue_3_activate [hotcue_3_clear]	
9	I/O	Hot cue 4 [Delete]	9X	29H [2DH]	7FH: ON / 00H:OFF	[ChannelN]	Hotcue_4_activate [hotcue_4_clear]	
10	-	Shift	-	-	-			
11	-	USB	-	-	-			
12	-	4.3" LCD	-	-	-			
13	I/O	Auto Loop 2 [1/8]	9X	22H [1EH]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.BeatloopControl [GeminiMDJ1000.BeatloopControl]	
13	I/O	Auto Loop 4 [¼]	9X	23H [1FH]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.BeatloopControl [GeminiMDJ1000.BeatloopControl]	
13	I/O	Auto Loop 8 [½]	9X	24H [20H]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.BeatloopControl [GeminiMDJ1000.BeatloopControl]	
13	I/O	Auto Loop 16 [1]	9X	25H [21H]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.BeatloopControl [GeminiMDJ1000.BeatloopControl]	
14	O	Browse Clockwise / Counterclockwise (Search Window)	9x	0CH / 0BH	7FH Following 00H	[Playlist]	SelectNextTrack / SelectPrevTrack	
14	O	Select [Select]	9X	09H [31H]	7FH: ON / 00H:OFF	[Playlist]	LoadSelectedIntoFirstStopped [none]	Half used
15	O	Back [Settings]	9x	0AH	7FH: ON / 00H:OFF			Empty
16	O	Eject	9x	1DH	7FH: ON / 00H:OFF	[ChannelN]	eject	
17	-	Source select	-	-	-			
18	O	Jos start time	BX	02H	7FH: + / 00H: -			Empty
19	O	Jog stop time	BX	03H	7FH: + / 00H: -			Empty
20	O	Filter	BX	04H	7FH: HPF / 00H: LPF	QuickEffectRack1_[ChannelN]	super1	
21	O	Vinyl [Vinyl]	9X	19H [30H]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.VinylModeEnabled [none]	Half used
22	O	Slip mode	9X	1AH [32H]	7FH: ON / 00H:OFF	[ChannelN]	GeminiMDJ1000.SlipEnabled [none]	Half used
23	O	BPM Tap [Lock]	9X	16H [17H]	7FH: ON / 00H:OFF	[ChannelN]	beats_translate_curpos [quantize]	
24	I/O	Keylock [Autocue]	9X	13H [1CH]	7FH: ON / 00H:OFF	[ChannelN]	Keylock [none]	Half used
25	O	Pitch bend (-) [Range]	9X	12H [18H]	7FH: ON / 00H:OFF	[ChannelN]	rate_perm_down_small [GeminiMDJ1000.BPM_Range]	
25	O	Pitch bend (+) [BPM Mode]	9X	11H [15H]	7FH: ON / 00H:OFF	[ChannelN]	rate_perm_up_small [none]	Half used
26	O	Tempo Slider	BX	01H	7FH: + / 00H: -	[ChannelN]	rate (inverted)	

LIITE 2

```

/** Control for slip/roll enabled**/
GeminiMDJ1000.SlipEnabled = function(channel, control, value, status, group){
    if(value == GeminiMDJ1000.on){
        if(!GeminiMDJ1000.SlipEnabled){
            GeminiMDJ1000.SlipEnabled = true;
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["slip_enabled"], GeminiMDJ1000.off);
        }
        else{
            GeminiMDJ1000.SlipEnabled = false;
            engine.setValue(group, "slip_enabled", 0);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["slip_enabled"], GeminiMDJ1000.on);
        }
    }
}

/** Control for Reverse/Reverseroll**/
GeminiMDJ1000.Reverse = function(channel, control, value, status, group){
    if(!GeminiMDJ1000.SlipEnabled){
        engine.setValue(group, "reverseroll", value * 1);
    }
    else if(GeminiMDJ1000.SlipEnabled){
        engine.setValue(group, "reverse", value * 1);
    }
}

/** Control for Vinyl mode enabled (on/off) **/
GeminiMDJ1000.VinylModeEnabled = function(channel, control, value, status, group) {
    if(value == GeminiMDJ1000.on){
        if(!GeminiMDJ1000.VinylModeEnabled){
            GeminiMDJ1000.VinylModeEnabled = true;
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["vinyl_mode_enabled"], GeminiMDJ1000.off);
        }
        else{
            GeminiMDJ1000.VinylModeEnabled = false;
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["vinyl_mode_enabled"], GeminiMDJ1000.on);
        }
    }
}

/** Control for turning wheel **/
GeminiMDJ1000.WheelTurn = function (channel, control, value, status, group) {
    var newvalue = value-64;
    if(engine.getValue(group, "scratch2_enable")){
        engine.setValue(group, "scratch2", newvalue/21);
    }
    else{
        engine.setValue(group, "jog", newvalue/80);
    }
}

/** Control For Wheel Touch **/
GeminiMDJ1000.WheelTouch = function (channel, control, value, status, group) {
    if(!GeminiMDJ1000.VinylModeEnabled){
        if(value==GeminiMDJ1000.on){ //on
            engine.setValue(group,"scratch2_enable", 1);
        }
        if(value == GeminiMDJ1000.off){ //off
            engine.setValue(group,"scratch2_enable", 0);
        }
    }
}

/** Control for beatlooproll on/off **/
GeminiMDJ1000.LoopRollEnabled = function(channel, control, value, status, group) {
    if(value == GeminiMDJ1000.on){
        if(!GeminiMDJ1000.LoopRollEnabled){
            GeminiMDJ1000.LoopRollEnabled = true;
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_2_enabled"], GeminiMDJ1000.off);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_4_enabled"], GeminiMDJ1000.off);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_8_enabled"], GeminiMDJ1000.off);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_16_enabled"], GeminiMDJ1000.off);
        }
        else{
            GeminiMDJ1000.LoopRollEnabled = false;
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_2_enabled"], GeminiMDJ1000.on);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_4_enabled"], GeminiMDJ1000.on);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_8_enabled"], GeminiMDJ1000.on);
            midi.sendShortMsg(GeminiMDJ1000.Channelmessages[group], GeminiMDJ1000.LedControl["beatloop_16_enabled"], GeminiMDJ1000.on);
        }
    }
}

```