

Ville Varanka

**MIDI-STANDARDIN KÄYTTÖ JA TOIMINTAPERIAATE SEKÄ  
TUOTANNON SÄHKÖISEN TESTAUSLAITTEISTON SUUNNIT-  
TELU JA TOTEUTUS**

**MIDI-STANDARDIN KÄYTTÖ JA TOIMINTAPERIAATE SEKÄ  
TUOTANNON SÄHKÖISEN TESTAUSLAITTEISTON SUUNNIT-  
TELU JA TOTEUTUS**

Ville Varanka  
Opinnäytetyö  
Syksy 2017  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelun suuntautumisvaihtoehto

---

Tekijä: Ville Varanka  
Opinnäytetyön nimi: MIDI-standardin käyttö ja toimintaperiaate sekä tuotannon sähköisen testauslaitteiston suunnittelu ja toteutus  
Työn ohjaaja(t): Kari Jyrkkä, Veijo Väisänen  
Työn valmistumislukukausi ja -vuosi: Syksy 2017  
Sivumäärä: 59 + 1 liite

---

Tämä opinnäytetyö on suoritettu kahdessa osassa, joista ensimmäinen on 5 ja toinen 10 opintopisteen kokonaisuus. Ensimmäinen osa valmistui keväällä 2016 ja toinen syksyllä 2017.

Opinnäytetyön ensimmäisessä osassa käsitellään Musical Instrument Digital Interface eli MIDI-standardia, jonka avulla elektroniset musiikki-instrumentit ja muut liittyvät laitteet voivat kommunikoida keskenään. Työn tavoitteina oli selvittää, mikä MIDI on, miten se toimii, mihin sitä voidaan käyttää sekä mikä sen historia sekä tulevaisuus ovat. Tavoitteista tullaan keskittymään eniten MIDI:n toimintaan pureutuen järjestelmän teknisiin osuuksiin, joiden avulla voidaan selvittää toimintaperiaate.

Toinen osa on tehty toimeksiannosta MoveSole OY:lle, joka on lääkinnällisten laitteiden tuotekehitykseen erikoistunut yritys. Tavoitteena oli suunnitella ja toteuttaa prototyyppitestilaitte tuotannon sähköistä testaamista varten. Prototyypin toiminnallisuus todistettiin lähettämällä siniaalto firman tuotteen piirilevyn läpi, jolla myös todennettiin piirilevyn toiminnallisuus. Siniaalto oli sisääntulossa ja ulostulossa sama, joten prototyyppi toimi halutulla tavalla.

---

Asiasanat: koosteopinnäyte, Musical Instrument Digital Interface, testauslaitteet, sulautetut järjestelmät, lääkinnälliset laitteet

# SISÄLLYS

1 JOHDANTO	5
2 OPINNÄYTETYÖN ENSIMMÄISEN OSAN ESITTELY	6
3 OPINNÄYTETYÖN TOISEN OSAN ESITTELY	7
4 YHTEENVETO	8
LIITTEET	
Liite1. MIDI-standardin käyttö ja toimintaperiaate	
Liite2. Prototyping of an Electrical Test Equipment for Production	

# 1 JOHDANTO

Tämä opinnäytetyö on tehty koosteopinnäytetyönä, joka tarkoittaa opinnäytetyön tekemistä pienemmissä osissa. Työn ensimmäinen osa valmistui keväällä 2016 ja toinen 2017. Osat eivät liity toisiinsa, vaan ensimmäinen osa on selvitys musiikkiin vahvasti liittyvästä teknologiasta ja toinen osa on yritykselle tehty työ, jossa on tuotekehitykselle ominaista suunnittelua, toteutusta ja testaamista.

Työn ensimmäinen osa käsittelee MIDI-standardia, jossa selvitetään standardin historia, toimintaperiaate sekä tulevaisuudennäkymä. Työssä on keskitytty eniten standardin toimintaperiaatteeseen. Ensimmäinen osa opinnäytetyöstä on 5 opintopisteen laajuinen.

Työn toinen osa on tehty toimeksiannosta yritykselle, jossa työskentelen. Työn tarkoituksena on valmistaa sähköisen testauslaitteen prototyyppi tuotannon sähköistä testaamista varten. Laitteella testataan yrityksen älypohjallistuotteen piirilevyn toimintaa. Toinen osa opinnäytetyöstä on 10 opintopisteen laajuinen.

## **2 OPINNÄYTETYÖN ENSIMMÄISEN OSAN ESITTELY**

Opinnäytetyön ensimmäinen osa (liite 1) on tehty keväällä 2016 muiden opintojen lomassa. Työn ohjaajana toimi Kari Jyrkkä. Työn aihe on valittu kiinnostavuuden perusteella, sillä olen ensimmäiseltä ammatiltani muusikko. Työssä tutkitaan MIDI-standardia, joka on ollut läsnä lähes jokaisella musiikin osa-alueella 80-luvulta asti. Standardia käytetään mm. elektronisten musiikkilaitteistoiden kommunikaatiossa, jotka tukevat MIDIä.

Sain työstä paljon kokemusta liittyen MIDI-standardiin sekä opinnäytetöiden tekemiseen. Eniten kehittyi tiedonhaku, koska työ vaati paljon tutkimista ja erillisten tiedonjyvästen yhdistämistä suuremmaksi kokonaisuudeksi.

### 3 OPINNÄYTETYÖN TOISEN OSAN ESITTELY

Työn toinen osa (liite 2) on tehty toimeksiannosta MoveSole OY:lle, joka on lääkinnällisten laitteiden tuotekehitykseen erikoistunut yritys. Aloitin työskentelyn yrityksessä tammikuussa 2017. Nykyinen työnkuvani on tuotetestauksen insinööri, jonka lisäksi toimin sulautettujen järjestelmien kehittäjänä ja toteuttajana.

Työn aiheena oli valmistaa sähkötestauslaitteiston prototyyppi, jolla voidaan testata yrityksen tuotteen piirilevyn toiminta. Yrityksen tuote on älypohjallinen, jota käytetään lääkinnällisissä tarkoituksissa.

Prototyypin tuli olla toiminnaltaan ohjelmoitava signaaligeneraattori, jonka avulla voidaan syöttää haluttua signaaliaaltoa piirilevylle. Jos syötetty aalto oli ulostulosta mitattuna sama kuin sisäänmenossa, oli piirilevyn ja prototyypin toiminta todistettu.

Työn toisessa osassa jouduin ottamaan paljon käyttöön koulun projektien aikana opittuja työskentelymenetelmiä. Työ oli juuri sitä mitä olen oppinut koulutusohjelmassani, eli tuotteen tai laitteiston suunnittelua, toteutusta ja testaamista. Työn aikana jouduin käyttämään monipuolisesti osaamistani, sillä työ vaati mm. ohjelmointia, piirilevysuunnittelua ja komponenttien juottamista mikrooskooppien alla. Osaamiseni kehittyi huomattavasti tämän työn aikana, joka vaati onnistuakseen uusien asioiden opiskelua ja poistumista omalta mukavuusalueelta.

## 4 YHTEENVETO

Tämä kahdessa osassa tehty opinnäytetyö vastaa 15 opintopisteen kokonaisuutta. Ensimmäinen osan aihe on lähellä ensimmäistä ammattiani, joka on muusikko, mutta tietoteknisestä näkökulmasta. Toinen osa liittyy puhtaasti tietotekniikkaan ja minun suuntautumisvaihtoehtooni, joka on laite- ja tuotesuunnittelu. Työn toinen osa on kirjoitettu englanniksi, koska yrityksessä, missä työskentelen, kaikki dokumentaatiot ovat englanniksi.

Loppujen lopuksi molemmat osakokonaisuudet olivat itselle mielenkiintoisia aiheiltaan. Ensimmäisen osan aiheen sain valita itse ja valinta perustuu omaan nuoruuteen, kun vielä opiskelin elektronisia musiikkilaitteita. En vielä silloin ymmärtänyt MIDI-laitteiden toimintaperiaatetta tarkemmin, joten koin, että haluan yhdistää musiikki- ja tietoteknisen ymmärrykseni ja parantaa omaa tietämystä liittyen MIDI-standardiin. Toisen osan aihe tuli suoraan yritykseltäni ja tämänkään osakokonaisuuden aihe ei ollut ollenkaan huono. Mielenkiintoni pysyi yllä ja sain prototyypin valmistettua nopeampaa kuin kuvittelin.

Molemmat osakokonaisuudet olivat oppimisen ja ammattitaidon kehityksen kannalta hyviä. MIDI-standardin tietoa voi joutua käyttämään tulevaisuudessa, jos siirryn valmistamaan laitteita, jotka toimivat kyseisen standardin mukaisesti. Näkisin myös, että sähköisten testilaitteiden tietämystä tulee tarvitsemaan lähes jokaisen laitteen tai tuotteen kohdalla, jonka sisällä on piirilevy.





Ville Varanka

## **MIDI-STANDARDIN KÄYTTÖ JA TOIMINTAPERIAATE**

## **MIDI-STANDARDIN KÄYTTÖ JA TOIMINTAPERIAATE**

Ville Varanka  
Opinnäytetyö  
Kevät 2016  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## SISÄLLYS

1 JOHDANTO	5
2 MIDI	6
2.1 Historia	8
2.2 Erilaiset MIDI-ohjaimet (MIDI Controller)	10
2.2.1 MIDI-instrumentit	10
2.2.2 MIDI-sekvensserit	12
2.2.3 MMC-laitteet ja ohjelmat (MIDI Machine Control)	14
2.2.4 MSC-laitteet ja ohjelmat (MIDI Show Control)	14
3 TOIMINTAPERIAATE MIDI-SOITTIMISSA	16
3.1 Kanavaääniviestit (Channel Voice Messages)	17
3.1.1 Sävelen On/Off tila ja voimakkuus (Note On/Note Off/Velocity)	17
3.1.2 Koskettimeen kohdistuva paine (Aftertouch)	17
3.1.3 Sävelkorkeuden taivutus (Pitch Bend)	18
3.1.4 Ohjelmanvaihto (Program Change)	18
3.1.5 Ohjauksenvaihto (Control Change)	19
3.1.6 Kentän vaihto (Bank Select)	19
3.1.7 Rekisteröidyt ja ei-rekisteröidyt parametrit (RPN / NRPN)	19
3.2 Kanavatilaviestit (Channel Mode Messages)	20
3.2.1 Ohjauksien nollaus (nro.121)	20
3.2.2 Paikallinen hallinta (Local Control) Päälle/Pois (nro.122)	20
3.2.3 Nuottien mykistys (All Notes Off) (nro.123)	21
3.2.4 Omni-tila Päällä/Pois (nro.124 ja 125)	21
3.2.5 Mono- ja Poly-tilat (nro 126 ja 127)	21
3.3 Järjestelmäviestit (System Messages)	22
3.3.1 Yleiset järjestelmäviestit (System Common Messages)	22
3.3.1.1 MIDI-aikakoodin neljännes kehys (MTC Quarter Frame)	22
3.3.1.2 Kappaleen valinta (Song Select)	22
3.3.1.3 Kappaleen kohdan osoitus (Song Position Pointer)	22

3.3.1.4 Yksinoikeutetun viestin lopetus (End Of Exclusive)	22
3.3.2 Reaaliaikaiset järjestelmäviestit (System Real Time Messages)	23
3.3.2.1 Ajastuskello (Timing Clock)	23
3.3.2.2 Aloitus, jatkaminen ja lopetus (Start,Continue &Stop)	23
3.3.2.3 Aktiivinen tunnustelu (Active Sensing)	23
3.3.2.4 Järjestelmän nollaus (System Reset)	23
3.3.3 Yksinoikeutetut järjestelmäviestit (System Exclusive Messages)	23
4 MIDIN TULEVAISUUS	24
5 YHTEENVETO	26
6 LÄHTEET	27

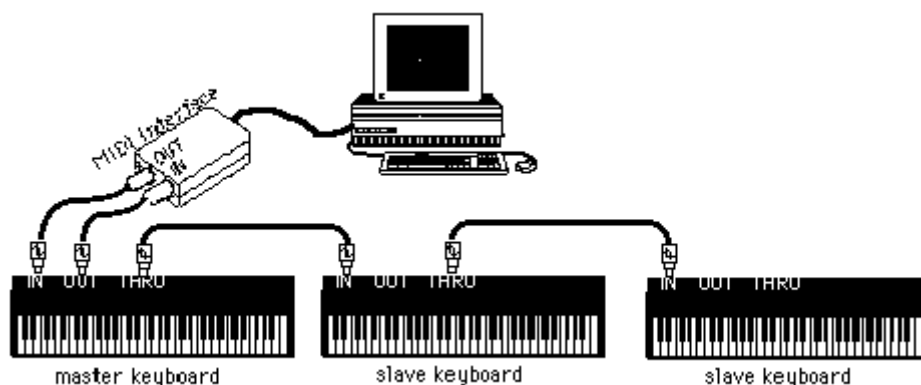
## 1 JOHDANTO

Tässä kolmiosaisen opinnäytetyön ensimmäisessä osassa käsitellään Musical Instrument Digital Interfacea eli MIDI-standardia, jonka avulla elektroniset musiikki-instrumentit ja muut liittyvät laitteet voivat kommunikoida keskenään. Työn tavoitteina on selvittää mikä MIDI on, miten se toimii, mihin sitä voidaan käyttää sekä mikä sen historia sekä tulevaisuus ovat. Tavoitteista tullaan keskittymään eniten MIDI:n toimintaan pureutuen järjestelmän teknisiin osuuksiin, joiden avulla voidaan selvittää toimintaperiaate. Esimerkkeinä toiminnasta tullaan käyttämään elektronisia musiikkilaitteita, jotka tukevat MIDIä.

Valitsin tämän aiheen siksi, koska olen ammatiltani muusikko ja minua on vanhemmiten alkanut kiinnostamaan, miten MIDI-syntetisaattorit oikeasti toimivat. Lapsena en ollut kiinnostunut teoriasta MIDI-soittimien ja ohjelmien takana juuri ollenkaan, vaan erehdyksen ja oivalluksen kautta opin käyttämään niitä. Nyt nuorena aikuisena koen itselleni tarpeellisenä tutkia MIDIä enemmän, koska joudun käyttämään kyseisiä laitteita töissä silloin tällöin ja nyt voin täydentää tietovarastoani ja kokea lapsenomaista kiinnostusta uudelleen MIDIä kohtaan teknisestä ja teoreettisesta näkökulmasta.

## 2 MIDI

MIDI eli Musical Instrument Digital Interface on tekninen standardi, joka määrittää käyttöprotokollan niiden laitteiden välille, jotka tukevat MIDIä. MIDI-laitteiden välisessä viestinnässä isäntälaitte antaa käskyjä MIDI-ulostulokanavan (MIDI-OUT) kautta ns. orjalaitteille, jotka ottavat käskyt vastaan MIDI-sisääntulokanavaan (MIDI-IN) (Kuva 1). Tällä tavalla isäntälaitte voi ohjata kohdelaitteiden toimintaa. Esimerkiksi musiikkilaitteiden välillä MIDI-viestit sisältävät tiedot parametreista, joilla isäntälaitte (esim. MIDI-sekvensseri) pystyy määrittämään kohdelaitteen (esim. syntetisaattori) luoman äänen notaation, korkeuden ja voimakkuuden, panoroinnin, pituuden ja tempon (1;2;3). Lisäksi syntetisaattoreita ja muita MIDI-soittimia varten on tehty General MIDI -luokitus (GM), joka lisää MIDI-standardin jatkeeksi 128 soitinäänänen valikoiman, jota voi käyttää vapaasti 16 MIDI-kanavalla (6) (Kuva 2). Tämä tarkoittaa käytännössä sitä, että esim. syntetisaattorin MIDI-kanavalle 1 voidaan valita kitaraääni ja kanavalle 5 viulu.



*KUVA 1. Kuva MIDI-kytkennästä, jossa isäntälaitte (master keyboard tai tietokone) ohjaa siihen linkitettyjä muita laitteita*

Piano	Chromatic Percussion	Organ	Guitar
0 Acoustic Grand Piano	8 Celesta	16 Hammond Organ	24 Acoustic Guitar (nylon)
1 Bright Acoustic Piano	9 Glockenspiel	17 Percussive Organ	25 Acoustic Guitar (steel)
2 Electric Grand Piano	10 Music box	18 Rock Organ	26 Electric Guitar (jazz)
3 Honky-tonk Piano	11 Vibraphone	19 Church Organ	27 Electric Guitar (clean)
4 Rhodes Piano	12 Marimba	20 Reed Organ	28 Electric Guitar (muted)
5 Chorused Piano	13 Xylophone	21 Accordion	29 Overdriven Guitar
6 Harpsichord	14 Tubular Bells	22 Harmonica	30 Distortion Guitar
7 Clavinet	15 Dulcimer	23 Tango Accordion	31 Guitar Harmonics
Bass	Strings	Ensemble	Brass
32 Acoustic Bass	40 Violin	48 String Ensemble 1	56 Trumpet
33 Electric Bass (finger)	41 Viola	49 String Ensemble 2	57 Trombone
34 Electric Bass (pick)	42 Cello	50 SynthStrings 1	58 Tuba
35 Fretless Bass	43 Contrabass	51 SynthStrings 2	59 Muted Trumpet
36 Slap Bass 1	44 Tremolo Strings	52 Choir Aahs	60 French Horn
37 Slap Bass 2	45 Pizzicato Strings	53 Voice Oohs	61 Brass Section
38 Synth Bass 1	46 Orchestral Harp	54 Synth Voice	62 Synth Brass 1
39 Synth Bass 2	47 Timpani	55 Orchestra Hit	63 Synth Brass 2
Reed	Pipe	Synth Lead	Synth Pad
64 Soprano Sax	72 Piccolo	80 Lead 1 (square)	88 Pad 1 (new age)
65 Alto Sax	73 Flute	81 Lead 2 (sawtooth)	89 Pad 2 (warm)
66 Tenor Sax	74 Recorder	82 Lead 3 (caliope lead)	90 Pad 3 (polysynth)
67 Baritone Sax	75 Pan Flute	83 Lead 4 (chiff lead)	91 Pad 4 (choir)
68 Oboe	76 Bottle Blow	84 Lead 5 (charang)	92 Pad 5 (bowed)
69 English Horn	77 Shakuhachi	85 Lead 6 (voice)	93 Pad 6 (metallic)
70 Bassoon	78 Whistle	86 Lead 7 (fifths)	94 Pad 7 (halo)
71 Clarinet	79 Ocarina	87 Lead 8 (brass + lead)	95 Pad 8 (sweep)
Synth Effects	Ethnic	Percussive	Sound Effects
96 FX 1 (rain)	104 Sitar	112 Tinkle Bell	120 Guitar Fret Noise
97 FX 2 (soundtrack)	105 Banjo	113 Agogo	121 Breath Noise
98 FX 3 (crystal)	106 Shamisen	114 Steel Drums	122 Seashore
99 FX 4 (atmosphere)	107 Koto	115 Woodblock	123 Bird Tweet
100 FX 5 (brightness)	108 Kalimba	116 Taiko Drum	124 Telephone Ring
101 FX 6 (goblins)	109 Bagpipe	117 Melodic Tom	125 Helicopter
102 FX 7 (echoes)	110 Fiddle	118 Synth Drum	126 Applause
103 FX 8 (sci-fi)	111 Shanai	119 Reverse Cymbal	127 Gunshot

KUVA 2. Lista General MIDIn suomista äänivaihtoehdoista

## 2.1 Historia

1980-luvun alkupuolella syntetisaattoreiden valmistajien (Ronald, Sequential Circuit) kesken oltiin huolissaan eri laitteiden yhteensopivuudesta, eli valmistajien välisiä laitteita ei voitu välttämättä yhdistää toisiinsa. Tästä esimerkkinä jonkun valmistajan jalkapedaalit eivät välttämättä olleet yhteensopivia toisen valmistajan pianoiden kanssa. Myös laitteiden välisessä kommunikaatiossa ja ohjauksessa oli ongelmia. Tavoitteena oli myös, että yhdellä syntetisaattorilla pystyttäisiin ohjaamaan kahta tai useampaa muuta syntetisaattoria, mikä mahdollistaisi eriltä kuulostavien äänien yhtäaikaissoittamisen. Ohjaamiseen vaadittiin yhteinen kieli, joka kertoisi muille syntetisaattoreille parametrit siitä, mitä isäntälaitte tekee (mikä sävel, nuotin pituus jne.) Yhteistä kieltä alettiin miettiä vuonna 1981 ja insinöörit Dave Smith ja Chet Wood Sequential Circuits -firmasta esittivät samana vuonna konseptin nimeltään Universal Synthesizer Interface, joka toimi 19,2 kilobitin nopeudella ja käytti ¼ tuuman audiotulppavastaketta. Tammikuussa 1982 järjestettiin kokous johtavien amerikkalaisten ja japanilaisten syntetisaattorivalmistajien kesken. Kokouksessa tehtiin parannuksia konseptiin: nopeus nostettiin 31,25 kilobittiin ja konseptiin lisättiin optoeristetty kytkentä. Myös nimi muutettiin nykyiseen muotoon (16).

Joulukuussa 1982 ilmestyi ensimmäinen uutta MIDI-protokollaa käyttävä instrumentti: Sequential Prophet-600 (Kuva 3). Kuukausi sen jälkeen tammikuussa 1983 Ronald valmisti toisen MIDI-instrumentin: Ronald Jupiter-6 (Kuva 4). Samassa kuussa kyseiset instrumentit saatiin kommunikoidaan keskenään onnistuneesti MIDI-kaapeleiden avulla. Vuonna 1983 MIDI-protokolla oli vain 8 sivun mittainen ja se sisälsi ainoastaan vaatimattomimmat toiminnot syntetisaattoreiden välillä, kuten sävelien muodostaminen koskettimistolla ja äänenvoimakkuuden säädön. Protokolla päivitettiin nykyiseen muotoonsa hyvin pian, jotta MIDI-laitteita pystyttiin ohjaamaan ja hallinnoimaan paremmin. Tulevina vuosina protokolla tuli kattamaan uusia ominaisuuksia, kuten General MIDIn ja USB-, FireWire- ja Wi-Fi-yhteydet. MIDillä tehty musiikki tuli myös tutuksi etenkin 1980–90-luvuilla, kun sitä alettiin käyttää tietokone- ja videopeleissä (16).





*KUVA 3. Sequential Prophet-600*



*KUVA 4. Ronald Jupiter-6*

## 2.2 Erilaiset MIDI-ohjaimet (MIDI Controller)

MIDI-ohjaimella (MIDI Controller) tarkoitetaan laitetta tai ohjelmistoa, joka lähettää MIDI-dataa vastaanottavaan MIDI-laitteeseen halliten tämän toimintaa. Alla on muutama esimerkki erilaisista MIDI-ohjaimista.

### 2.2.1 MIDI-instrumentit

Tunnetuin MIDI-instrumentti on MIDI-koskettimisto (Kuvat 3 ja 4), jolle koko järjestelmä alun perin suunniteltiin. Ajan myötä instrumenttiosasto alkoi laajentumaan ns. vaihtoehtoisin ohjaimiin. Näihin kuuluvat muun muassa:

- MIDI-puhallinsoittimet (Wind Controllers). Nämä soittimet toimivat kuten normaalit puhallinsoittimet, mutta lähettävät MIDI-dataa hallittavalle MIDI-laitteelle, kuten tietoa sävelkorkeudesta, ilmanpaineesta, puremapaineesta ja soittimen nappeihin kohdistuvasta paineesta (Kuva 5) (18).
- MIDI-rumpusoittimet (Struck Controllers). Näihin kuuluvat erilaiset lyömäsoittimet, kuten kokonaiset sähkörumpusetit (Kuva 6) tai rumpupadit (Kuva 7), joita ohjataan perkussiivisesti kapulalla tai muulla kappaleella lyömällä (19).
- MIDI-kielisoittimet (String Controllers). Näihin kuuluvat erilaiset kielisoittimet, joihin kiinnitetään MIDI-mikrofoni. Mikrofoni muuttaa kielen värinän digitaaliseen MIDI-muotoon, jolla pystyy hallitsemaan MIDI-syntetisaattorin ääniä (20). Tämä mahdollistaa esimerkiksi huiluäänen soittamisen kitaralla. MIDI-mikrofonin voi kiinnittää mihin tahansa kielisoittimeen (Kuva 8), mutta on myös olemassa MIDI-kielisoittimia, joihin mikrofoni on sisäänrakennettu (Kuva 9).



*KUVA 5. MIDI-puhallinsoitin*



*KUVA 6. MIDI-rumpusetti*



*KUVA 7. MIDI-rumpupadi*



*KUVA 8. Kitara, johon on kiinnitetty MIDI-mikrofoni*



*KUVA 9. MIDI-kitara sisäänrakennetulla MIDI-mikrofonilla*

### **2.2.2 MIDI-sekvensserit**

MIDI-sekvensserit ovat MIDI-laitteita tai ohjelmia, jotka säilyttävät, muokkaavat ja palauttavat MIDI-dataa ja lähettävät datan ohjattaviin MIDI-laitteisiin (Kuva 10). Sekvensserin toimintoihin kuuluvat:

- Reaaliaikainen äänitys. MIDI-soittimella voidaan äänittää reaaliaikaisesti sekvensserin muistiin dataa.
- Askellettu äänitys. Äänitys voidaan suorittaa kohta/nuotti kerrallaan.
- Editointi. Tallennettua dataa voi editoida.
- Sävelen kvantisointi. Mahdollisuus kvantisoida eli siirtää halutut nuotit oikeaan kohtaan. Toisin sanoen, jos soitetaan 16-osanuotti hieman pieleen, voidaan kvantisoida se tahdin 16-osan kohtaan. Silloin kyseinen nuotti menee juuri oikeassa rytmissä.
- Sävelten siirtäminen. Sävelten korkeutta ja myös koko nauhoitteen sävel-lajia voidaan vaihtaa.
- Kopiointi/Leikkaus/Liittäminen. Mahdollisuus kopioida ja leikata osia nauhoitteesta ja liittää ne haluttuun kohtaan.
- Mykistys. Halutut kohdat tai kaikki raidat voidaan mykistää.
- ”Looppaus” eli kierrätys. Mahdollisuus kierrättää tiettyä kohtaa nauhoitteessa.
- Peruutus. Edellinen toiminto voidaan peruuttaa (21).



KUVA 10. MIDI-sekvensseri

### 2.2.3 MMC-laitteet ja ohjelmat (MIDI Machine Control)

MMC-laitteet ja ohjelmat on tehty äänitystä ja toistoa varten. Näillä pystyy esimerkiksi ohjaamaan sekvenssereiden dataa. MMC-laitteiden toimintoihin kuuluvat datan toisto (Play), pikatoisto (Fast Forward), pikakelaus (Rewind), pysäytys (Stop), tauko (Pause) ja äänitys (Record) (22,17). Kyseiset toiminnot kuuluvat yksinoikeutettuihin järjestelmäviesteihin (System Exclusive Messages), joten niitä löytyy kaikista MIDI-laitteista, jotka tukevat MMC-protokollaa. Jotkut MIDI-laitteet toimivat pelkästään MMC-protokollalla.



KUVA 11. MMC-laite

### 2.2.4 MSC-laitteet ja ohjelmat (MIDI Show Control)

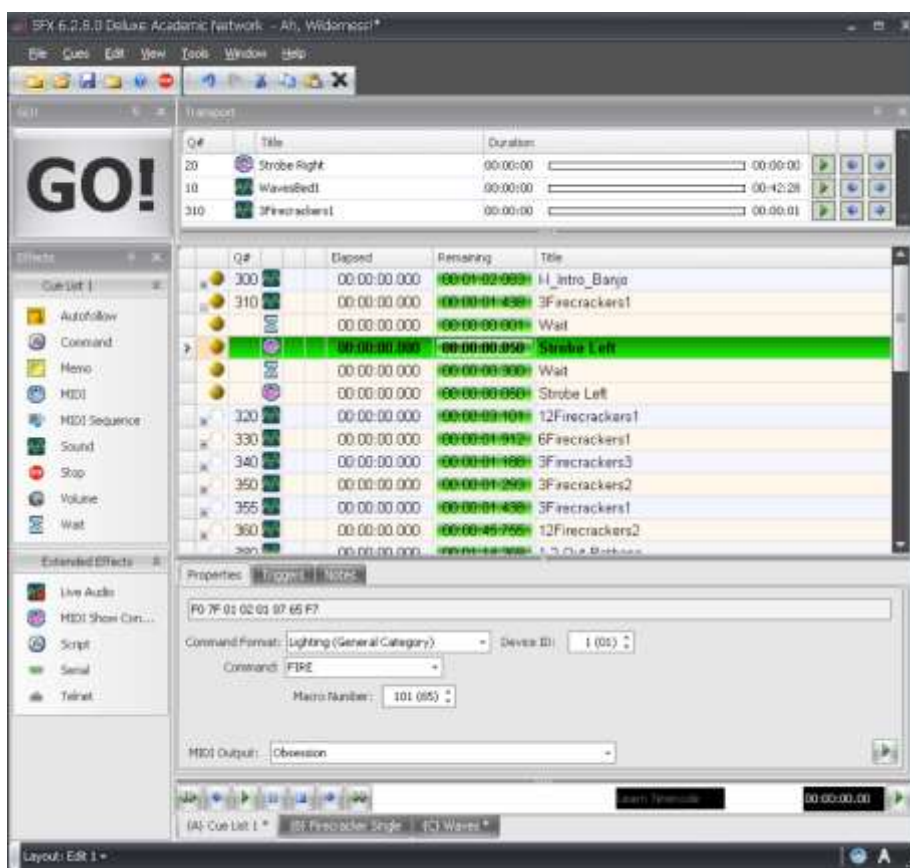
MSC-laitteita ja ohjelmia käytetään ohjaamaan ääntä, valoja, projektoreita ja muita automaattisia toimintoja huvipuistoissa, konserteissa, teattereissa ja teemapuistoissa. MSC-viestit ovat osajoukko yksityisten järjestelmäviestien status-tavua. Esimerkki MSC-viestistä on:

```
F0 7F <device_ID> 02 <command_format> <command> <data> F7
```

jossa

- F0 on statustavu, joka kertoo viestin olevan yksityinen järjestelmäviesti.
- 7F tavu kertoo valmistajan ID:n eli henkilöllisyyden.

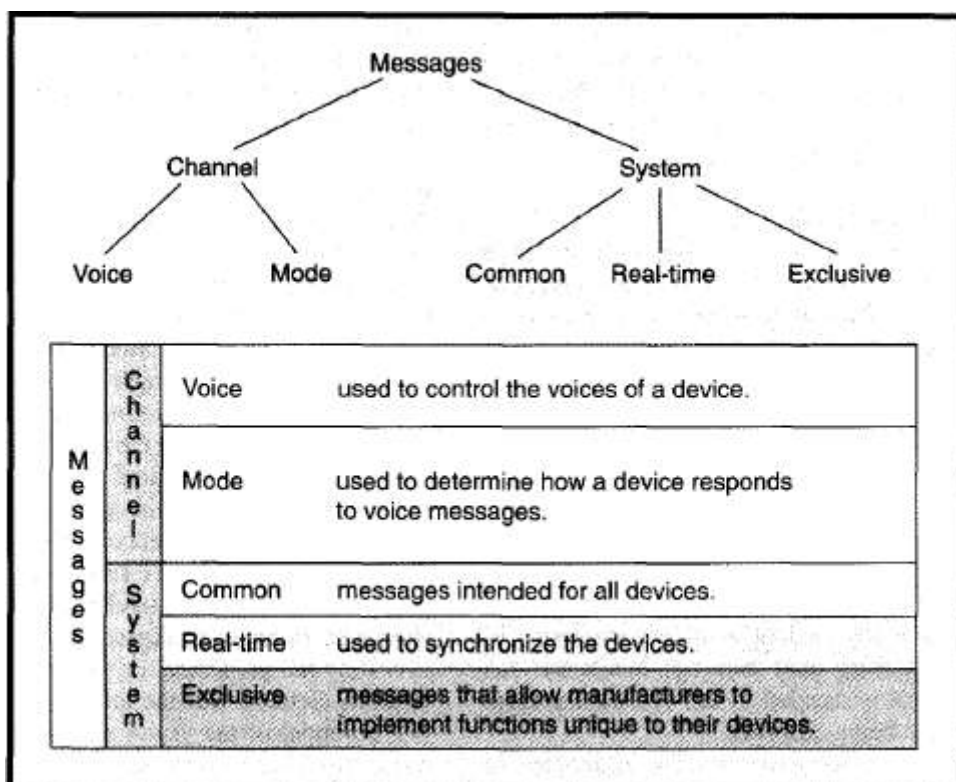
- <device\_ID> voi olla mikä tahansa numero 0x00 – 0x7F välillä ja kertoo ohjattavan laitteen ID:n.
- 02 kertoo, että seuraavat tavut ovat MSC-käskyjä
- <command\_format> kertoo ohjattavan laitteen tyyppin. Esimerkiksi numero 0x01 kertoo kyseessä olevan valolaite ja 0x60 pyrotekninen laite.
- <command> on käsky, mikä annetaan ohjattavalle laitteelle. Esimerkiksi numero 0x01 on Toisto-komento ja 0x02 on Pysäytys-komento.
- <data> kertoo vaihtelevan datatavujen numeron, jota tarvitaan jokaiselle command-käskylle. Datatavujen numeroilla voidaan määrittellä komentojen tehtävät (cue). Jos esimerkiksi halutaan, että valot menevät ensiksi päälle ja sitten sammuvat, niin tämä toiminta pitää määrittellä tehtävälis-tään.
- F7 kertoo, että viesti loppuu (17).



KUVA 12. MSC-ohjelma, jossa näkyy lista komennoista.

### 3 TOIMINTAPERIAATE MIDI-SOITTIMISSA

MIDI-soittimien toimintaperiaate MIDI-protokollassa on MIDI-viestit (Kuva 13). MIDI-viestit ovat binäärilukuja, joista ensimmäinen tavu kertoo viestin tilanteen (statustavu) (9) ja sitä seuraavat yksi tai kaksi tavua kertovat, mitä dataa siirretään (datatavut) (8). Datatavut ja statustavut erotetaan ensimmäisellä bitillä (1 = statustavu, 0 = datatavu) (11). Viestien lähettämisenopeus on 31 250 bittiä sekunnissa. Vaikka erityyppisiä viestejä on lukuisia, voidaan MIDI-viestit jakaa kahteen pääkategoriaan: kanava- ja järjestelmäviesteihin (Channel Messages & System Messages). Kanavaviestit toimivat vain halutuilla MIDI-kanavilla (1–16 kanavaa) ja käytettävän kanavan numero on ilmoitettu kanavaviestien statustavussa. Järjestelmäviestit eivät taas ole kanavakohtaisia ja niiden statustavuissa ei ilmoiteta käytettävää kanavaa. Nämä kaksi pääkategoriaa jaetaan seuraaviin alla oleviin alakategorioihin. Ensimmäinen pääkategoria (kanavaviestit) alkaa alaotsikosta 3.1 ja toinen (järjestelmäviestit) 3.3.



KUVA 13. MIDI-viestien rakenne



### **3.1 Kanavaääniviestit (Channel Voice Messages)**

Kanavaääniviestit lähettävät informaatiota musikaalisesta performanssista sille kanavalle, jota käytetään. Kanavaääniviestit jaetaan alla oleviin alakategorioihin.

#### **3.1.1 Sävelen On/Off tila ja voimakkuus (Note On/Note Off/Velocity)**

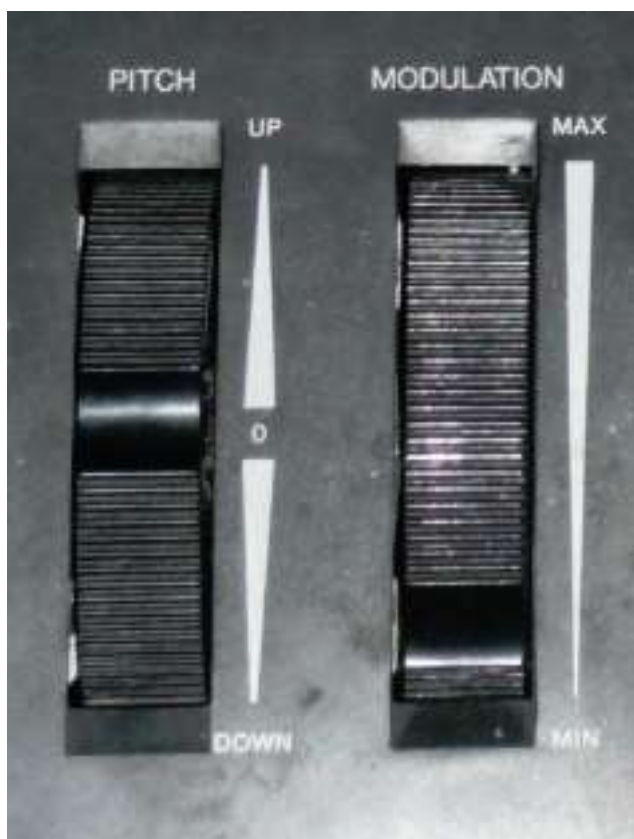
Kun MIDI-koskettimistolta painetaan jotain kosketinta, koskettimisto lähettää Note On -viestin MIDI OUT -kanavaan. Koska koskettimisto lähettää informaatiota jollekin halutulle MIDI-kanavalle (kanavat 1–16), sisältää Note On -viestin statustavu käytettävän MIDI-kanavan numeron. Statustavun jälkeen tulee kaksi datatavua, joista ensimmäinen kertoo painetun sävelen (C, A, D#) ja toinen sävelen äänenvoimakkuuden eli amplitudin asteikolla 0–127. Asteikko on siksi 0–127, koska ensimmäinen bitti on määritelty kertomaan, onko kyseessä data- vai statustavu (11). Kun painetusta koskettimesta päästetään irti, lähettää koskettimisto Note Off -viestin. Note Off -viestin datatavu sisältää myös tiedot painetusta sävelestä ja voimakkuudesta, mutta normaalisti tämän viestin informaatio jätetään huomioimatta (7).

#### **3.1.2 Koskettimeen kohdistuva paine (Aftertouch)**

Joissakin MIDI-koskettimissa on paineominaisuus. Tämä tarkoittaa sitä, että kun pohjaan painettua kosketinta painetaan lisää, koskettimisto tunnistaa tämän paineen. Tätä ominaisuutta voidaan hyödyntää esim. silloin, kun halutaan soitettulle sävelelle aiheuttaa tehokeinomaista huojuntaa (vibrato). Ominaisuutta käytetään myös silloin, kun halutaan moniäänisyyttä. Esimerkiksi jos koskettimistolla painetaan kosketin pohjaan, joka tuottaa pianoäänen, niin lisäämällä painetta jo painettuun koskettimeen voidaan muodostaa pianoäänen rinnalle viuluääni. Aftertouch-viestit sisältävät kaksi datatavua, joista ensimmäinen kertoo, mikä elementti on kyseessä (vibrato, moniäänisävel) ja toinen kertoo, kuinka paljon painetta kohdistuu koskettimelle asteikolla 0–127 (7).

### 3.1.3 Sävelkorkeuden taivutus (Pitch Bend)

Sävelkorkeuden taivutus tarkoittaa soitetun sävelen korkeutta manipuloimista Pitch Bend -rullalla, joka löytyy monista MIDI-koskettimistoista (Kuva 14). Sävelkorkeuden manipulointi rullalla kuulostaa liukuvulta siirtymiseltä sävelkorkeudesta toiseen. Pitch Bend -viestit sisältävät kaksi datatavua, joista molemmat määrittävät arvot, mihin suuntaan ja kuinka paljon rullaa on väännetty asteikolla 0–127. Kaksi datatavua on tarpeellista, jotta siirtymä sävelestä toiseen kuulostaa mahdollisimman tarkalta ja liukuvulta (7).



KUVA 14. Pitch Bend -rulla oikealla

### 3.1.4 Ohjelmanvaihto (Program Change)

Program Change -viesteillä määritetään soitinääni, jota halutaan käyttää tietyllä midi-kanavalla. Nämä viestit tarvitsevat yhden tavun, jolla kerrotaan käytettävä soitinääni (Kuva 2) (7).

### 3.1.5 Ohjauksenvaihto (Control Change)

Control Change -viestejä käytetään, kun halutaan käyttää midi-laitteen lukuisia ohjaustoimintoja halutulla tavalla (8). Näitä ovat esimerkiksi erilaiset efektit, joissa ääntä ohjataan kaiuttimesta toiseen. Control Change -viesteihin menee yksi statusavu ja kaksi datatavua. Statusavu ilmaisee käytettävän kanavan numeron, ensimmäinen datatavu käytettävän ohjaustoiminnon ja toinen datatavu ohjaustoiminnon vahvuuden, eli esimerkiksi sen, kuinka nopeaa ääni liikkuu kaiuttimesta toiseen (7).

### 3.1.6 Kentän vaihto (Bank Select)

Bank Select -toimintoa käytetään, kun halutaan laajentaa käytettävää soitinkarttaa. Esimerkiksi jos käytetään soitinääntä nro. 0, joka on akustinen piano (Kuva 2), Bank Select -toiminnolla voidaan valita kyseisen äänen tilalle erilaiselta kuulostava akustinen piano. Bank Select -toiminto löytyy ohjauksenvaihdolla (Control Change) ja se on määritelty ohjaimiksi nro. 0 (Controller 0). Bank Select -toiminnon avulla jokaiselle 128 äänelle (Kuva 2) voidaan luoda 16,384 lisä-äänikenttää (7).

### 3.1.7 Rekisteröidyt ja ei-rekisteröidyt parametrit (RPN / NRPN)

Registered Parameter Number (RPN) ja Non-Registered Parameter Number (NRPN) ovat numeerisia toimintoja, joilla pystyy muokkaamaan syntetisaattorin rekisteröityjä ja ei-rekisteröityjä parametrejä. Muokkaus onnistuu valitsemalla ensiksi halutun parametrin numeron ohjaimilla 98 ja 99 tai 100 ja 101 ja sen jälkeen muokkaamalla parametrin arvoja ohjaimilla 6, 96 tai 97. Ohjaimet 98 ja 99 vaihtavat NRPN:n vähiten ja eniten merkitsevää bittiä, kuten ohjaimet 100 ja 101 RPN:n. Ohjaimilla 96 ja 97 lisätään ja vähennetään parametrin data-arvoja ja ohjaimella 6 valitaan parametrin muokkaus käyttöön. Rekisteröidyt parametrit ovat parametrejä, joille eri valmistajat antavat samat käyttötarkoitukset. Esimerkiksi Pitch Bend -rullan (Kuva 14) käytölle on määritelty samat RPN-parametrit valmis-

tajasta riippumatta. Parametrejä muokkaamalla saadaan rullan toiminnasta herkempää tai hitaampaa. Ei-rekisteröidyt parametrit taas riippuvat valmistajasta, koska niitä ei ole määritelty (7).

### **3.2 Kanavatilaviestit (Channel Mode Messages)**

Kanavatilaviestit määräävät, miten syntetisaattori reagoi tulevaan MIDI-dataan. Viestejä ohjataan Control Change -ohjaimilla nro 121–127 ja niiden toiminnot on kerrottuna alla.

#### **3.2.1 Ohjauksien nollaus (nro.121)**

Ohjaimella nro 121 voidaan nollata kaikkien ohjaimien arvot tehdasasetuksiin (7).

#### **3.2.2 Paikallinen hallinta (Local Control) Päälle/Pois (nro.122)**

Tällä ohjaimella voidaan määrittää, missä tilassa Local Control -toiminto on. Esimerkiksi jos käytetään syntetisaattoria, jossa on myös koskettimisto (Kuva 15), voidaan koskettimiston ja syntetisaattorin toiminnot erottaa Local Control Off -käskyllä. Näin niiden toiminnot eivät enää ole riippuvaisia toisistaan ja syntetisaattorin ääniä voidaan ohjata jollain toisella laitteella, kuten tietokoneella (7).



*KUVA 15. Syntetisaattori koskettimistolla*

### **3.2.3 Nuottien mykistys (All Notes Off) (nro.123)**

All Notes Off -ohjaimella voidaan mykistää MIDI-soitin. Jos All Notes Off -tila on päällä syntetisaattorissa, ei koskettimen painaminen tuota enää ääntä (7).

### **3.2.4 Omni-tila Päällä/Pois (nro.124 ja 125)**

Omni-tila määrittää, miten MIDI-laitteen kanavat reagoivat tulevaan dataan. Jos Omni-tila on päällä, kaikki kanavat reagoivat. Jos tila on pois päältä, vain yksi kanava reagoi. Omni-tilan päällä olemista voidaan verrata televisioon, joka näyttää kaikki kanavat yhtä aikaa (10,7).

### **3.2.5 Mono- ja Poly-tilat (nro 126 ja 127)**

Poly-tilassa MIDI-soittimella vastaanotetut Note On -viestit soitetaan moniäänisesti. Tämä tarkoittaa sitä, että koskettimistolla pystytään soittamaan niin monta ääntä kuin halutaan. Esimerkiksi jos on valittu pianoääni ja poly-tila on päällä, niin

voidaan painaa kaikki koskettimet pohjaan ja kuulla jokaisen koskettimen luoma pianoääni. Mono-tilassa taas voidaan soittaa yhtä ääntä kerralla. Esimerkiksi jos mono-tila on päällä ja painetaan jokainen kosketin pohjaan, kuulet vain ja ainoastaan sen äänen, minkä painoit ensiksi pohjaan (7).

### **3.3 Järjestelmäviestit (System Messages)**

Järjestelmäviestit jaetaan kolmeen alhaalla olevaan alakategoriaan: yleisiin, reaaliaikaisiin ja yksinoikeutettuihin järjestelmäviesteihin (7).

#### **3.3.1 Yleiset järjestelmäviestit (System Common Messages)**

Yleisiin järjestelmäviesteihin kuuluu neljä viestityyppiä, jotka on selitetty alla.

##### **3.3.1.1 MIDI-aikakoodin neljäs kehys (MTC Quarter Frame)**

MTC Quarter Frame -viestit ovat osa MIDI-aikakooditietoa. Viesteillä voidaan synkronoida eli samanaikaistaa MIDI-laitteiden toiminta (7).

##### **3.3.1.2 Kappaleen valinta (Song Select)**

Song Select -viesteillä voidaan vaihtaa nauhoitettua MIDI-kappaletta, joka on MIDI-protokollalla ohjelmoitu sävellys. MIDI-kappaleita nauhoitetaan yleensä rumpukoneilla ja sekvenssereillä, jotka ovat MIDI-yhteensopivia (7).

##### **3.3.1.3 Kappaleen kohdan osoitus (Song Position Pointer)**

Song Position Pointer -viesteillä voidaan määrittää ajallisesti mistä kohtaa MIDI-kappale alkaa. Näitä viestejä voidaan käyttää vain niiden MIDI-laitteiden kanssa, jotka tunnistavat reaaliaikaisia järjestelmäviestejä (System Real Time Messages) (7).

##### **3.3.1.4 Yksinoikeutetun viestin lopetus (End Of Exclusive)**

End Of Exclusive eli EOX-viesteillä merkataan, kun yksinoikeutettu järjestelmäviesti (System Exclusive Message) loppuu (7).

### **3.3.2 Reaaliaikaiset järjestelmäviestit (System Real Time Messages)**

Reaaliaikaisiin järjestelmäviesteihin kuuluvat alla selitetyt kuusi viestityyppiä.

#### **3.3.2.1 Ajastuskello (Timing Clock)**

Timing Clock -viestillä määritetään kappaleen nopeus eli tempo. Nopeus ilmaistaan iskuina per minuutti (beats per minute) (7).

#### **3.3.2.2 Aloitus, jatkaminen ja lopetus (Start,Continue &Stop)**

Start-, continue- ja stop-viesteillä ohjataan MIDI-kappaleen toiston aloitusta ja lopetusta (7).

#### **3.3.2.3 Aktiivinen tunnustelu (Active Sensing)**

Active Sensing -viestit pitävät huolen, että jos MIDI-laitteen MIDI-kaapeli irtoaa performanssin aikana ja Note Off -viestit, jotka ilmoittavat kun kosketinta ei enää paineta, eivät pääse perille, niin nuotit eivät jää soimaan vaan ne katkaistaan Active Sensing -viesteillä (7).

#### **3.3.2.4 Järjestelmän nollaus (System Reset)**

System Reset -viestillä voidaan nollata ja alustaa käytettävä MIDI-laite (7).

### **3.3.3 Yksinoikeutetut järjestelmäviestit (System Exclusive Messages)**

Yksinoikeutettuja järjestelmäviestejä käytetään, kun halutaan lähettää valmistajan määrittämää dataa MIDI-laitteiden välillä. Tämä data voi sisältää tietoa korjausparametreista tai ääninäytteistä. Yksinoikeutetut järjestelmäviestit sisältävät valmistajan oman tunnusnumeron (ID), jota seuraa mikä tahansa määrä datatavuja. Yksinoikeutettujen järjestelmäviestien datalähetys katkaistaan End Of Exclusive eli EOX-viesteillä (7).

## 4 MIDIN TULEVAISUUS

Uusi julkaisematon versio MIDI-protokollasta nimeltään HD-protokolla (High-Definition Protocol) tai HD-MIDI on ollut työn alla jo vuodesta 2005 (12). HD-MIDI:n keskeneräisyys johtunee uuden protokollan laajuudesta ja yhteensopivuusongelmista. HD-protokollan tuleviin ominaisuuksiin kuuluu:

- Yhteensopivuus vanhan protokollan (MIDI 1.0) kanssa varmistamaan laajamittaisen omaksumisen.
- Plug & Play -verkko-ominaisuudet helpottamaan USB:n ja Ethernetin välistä kommunikointia.
- Nopeampi datan siirto
- Tuhansia uusia MIDI-kanavia, ohjaimia ja parametrejä.
- MIDI-viestien päivitys, lisäys ja yksinkertaistaminen.
- Aikaleimaominaisuus MIDI-viesteihin ajoituksen parantamiseen.
- Yhteensopivuus uusien ohjainteknologien kanssa.
- Laajentumisvaraa tulevaisuutta varten, jotta uusi protokolla ei vanhene ennenaikaisesti (14,13).
- Tarkemmat sävelkorkeus- ja artikulaatio-ohjaukset, joilla mm. saadaan MIDI-kielisoittimien ulosanti kuulostamaan autenttisemmalta (15, 13).

Muuta uutta MIDI:n tulevaisuudessa on Web-MIDI -ohjelmointirajapinta, jonka on suunnitellut W3C-järjestön (World Wide Web Consortium) webbiaudior ryhmä (The Web Audio Working Group). Web-MIDI:n avulla käyttäjä pystyy käyttämään MIDI-laitteita nettiselaimen kautta (16). Jokainen laite, joka tukee nettiselaimen käyttöä (tietokone, tabletti), pystyy käyttämään kyseistä ohjelmointirajapintaa. Esimerkiksi jos jokin Web-MIDI -selainohjelma tuottaa jonkun efektin (kaiku), voi käyttäjä yhdistää oman MIDI-syntetisaattorin nettiselainta tukevaan laitteeseen, aukaista kyseisen efektin tuottavan selainohjelman ja käyttää efektiä hyödyksi halutun äänen kanssa. MIDI-ohjaimella voi myös halutessa pelata selainpelejä, jotka tukevat MIDIä. Web-MIDI:n etuihin kuuluu muun muassa:



- Toimii millä tahansa laitteella, jolla on pääsy Internetiin selaimen kautta.
- Kun MIDI-laite yhdistetään tietokoneeseen, tablettiin, älypuhelimeen tai mihin tahansa selainta tukevaan laitteeseen esimerkiksi USB:llä tai Wi-Fi:llä, on kyseinen MIDI-laite toiminnassa saman yhteyden/liitäntän kautta Web-MIDI:ssä.
- Web-MIDI -ohjelmat on aina päivitetty uusimpaan versioon.
- Web-MIDI -ohjelmia ja dataa voi käyttää missä vain, jossa on pääsy Internetiin.
- Web-MIDI -ohjelmia ja dataa voi käyttää, muokata ja jakaa käyttäjien välillä Internetin palvelujen avulla (23).



KUVA 16. Web-MIDI-ohjelma, jota ohjataan MIDI-koskettimistolla

## 5 YHTEENVETO

Tässä opinnäytetyön ensimmäisen vaiheen työssä käytiin läpi MIDI:n historiaa, käyttötarkoituksia, protokollan toimintaperiaate MIDI-soittimissa ja MIDI:n tulevaisuutta. Työn pääpaino oli MIDI-protokollan toimintaperiaatteen selvittämisessä, mikä onnistui minusta hyväksyttävästi. Protokollan toiminta perustuu erilaisiin MIDI-viesteihin, jotka ohjaavat MIDI-laitteiden järjestelmän ja kanavien toimintaa tavujen avulla. Toimintaperiaatteen selvittäminen oli erittäin mielenkiintoista ja opin itsekin uusia asioita. Olisin halunnut kirjoittaa enemmän kappaleeseen neljä uudesta HD-protokollasta, mutta siitä ei löytynyt vielä tarpeeksi tietoa, koska se on ollut tätä työtä tehdessä keskeneräinen.

## 6 LÄHTEET

1. Swift, Andrew 1997. A brief Introduction to MIDI. *SURPRISE* (Imperial College of Science Technology and Medicine). Viitattu 21.1.2016
2. Huber, David Miles. 1991. *The MIDI Manual*. Carmel, Indiana. SAMS. Viitattu 21.1.2016
3. Ghassaei, Amanda 2013. What is MIDI? Instructables. Viitattu 21.1.2016. Saatavissa: <http://www.instructables.com/id/What-is-MIDI/?ALLSTEPS>
4. Vinikow, Brad 1988. *The Mideo Series Part 1: Untangling MIDI*. VHS ja Youtube-videosarja. Fountain Valley. Video Impact Marketing. Viitattu 21.1.2016. Saatavissa: <https://www.youtube.com/watch?v=CJCe6mu-TIGI&list=PL57158B8833B1E6BF>
5. MIDI PRODUCTION. Layon, Ken 2000. *Understanding and Using MIDI – Part 1. General Concepts and Setting Up Your MIDI System*. Recording Engineer's Quarterly. Viitattu 21.1.2016. Saatavissa: <http://recordingeq.com/EQ/req0600/midi1.htm>
6. MIDI Association. *About MIDI*. Viitattu 4.2.2016. Saatavissa: <http://www.midi.org/about-midi/table3.shtml>
7. MIDI Association. *About MIDI-PART 3: Midi messages*. Viitattu 11.2.2016. Saatavissa: <https://www.midi.org/articles/about-midi-part-3-midi-messages>
8. MIDI Association. *Control Change Messages (Data Bytes)*. Viitattu 11.2.2016. Saatavissa: <https://www.midi.org/specifications/item/table-3-control-change-messages-data-bytes-2>
9. MIDI Association. *Expanded Messages List (Status Bytes)*. Viitattu 18.2.2016. Saatavissa: <https://www.midi.org/specifications/item/table-2-expanded-messages-list-status-bytes>

10. Hass, Jefferey. Chapter Three: MIDI. Viitattu 18.2.2016: Saatavissa: [http://www.indiana.edu/~emusic/etext/MIDI/chapter3\\_MIDI7.shtml](http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI7.shtml)
11. Stack OverFlow –foorumi. Aihe "Why Does MIDI Offer 127 Notes" Viitattu 10.3.2016. Saatavissa: <http://stackoverflow.com/questions/5106758/why-does-midi-offer-127-notes>
12. Battino, David. "Finally: MIDI 2.0" O'Reilly Digital Media Blog. Viitattu 10.3.2016.
13. Synthopia. MIDI Manufacturers Association Says New HD Protocol 'Has Reached A Milestone' Viitattu 10.3.2016. Saatavissa: <http://www.synthtopia.com/content/2015/01/16/new-midi-hd-protocol-has-reached-a-milestone/>
14. MMA HD Protocol Announcement. MIDI Manufacturers Association. Viitattu 17.3.2016.
15. General Meeting for MIDI developers by MMA. Pro-Music-News. Viitattu 17.3.2016. Saatavissa: <http://pro-music-news.com/html/01/e20105mm.htm>
16. MIDI Association. Midi History:Chapter 5-Midi Is Born 1980-1983. Viitattu 17.3.2016. Saatavissa: <https://www.midi.org/articles/midi-history-chapter-5>
17. Digital Sound & Music. 6.2.5 Non- Musical Applications for MIDI. Viitattu 7.4.2016. Saatavissa: <http://digitalsoundandmusic.com/6-2-5-non-musical-applications-for-midi/>
18. Patchman Music. Wind Controller FAQ. Viitattu 7.4.2016. Saatavissa: <http://www.patchmanmusic.com/WindControllerFAQ.html>
19. MIDI Association. Alternative Controllers: Part 5 Hand-Held And Struck Controllers, And Controllers For Microtonal Tunings. Viitattu 14.4.2016.

- Saatavissa: <https://www.midi.org/articles/alternative-controllers-part-5-hand-held-and-struck-controllers>
20. Huber, David Miles. "The MIDI Manual". Carmel, Indiana: SAMS, 1991. Viitattu 14.4.2016.
21. Sound On Sound. MIDI Basics, Part 2: Sequencing. Viitattu 21.4.2016. Saatavissa: [https://www.soundonsound.com/sos/1995\\_articles/sep95/midibasics2.html](https://www.soundonsound.com/sos/1995_articles/sep95/midibasics2.html)
22. Cakewalk. MIDI Machine Control (MMC). Viitattu 21.4.2016. Saatavissa: <https://www.cakewalk.com/Documentation?product=SONAR%20X2&language=4&help=Synchronizing.13.html>
23. MIDI Association. About Web-MIDI. Viitattu 28.4.2016. Saatavissa: <https://www.midi.org/articles/about-web-midi>



Ville Varanka

**PROTOTYPING OF AN ELECTRICAL TEST EQUIPMENT FOR  
PRODUCTION**

**PROTOTYPING OF AN ELECTRICAL TEST EQUIPMENT FOR  
PRODUCTION**

Ville Varanka  
Bachelor's Thesis  
Autumn 2017  
Information Technology  
Oulu University of Applied Sciences

## ABSTRACT

Oulu University of Applied Sciences  
Information Technology

---

Author: Ville Varanka

Title of the bachelor's thesis: Prototyping of an Electrical Test Equipment for Production

Supervisor: Vesa Kajanus & Veijo Väisänen

Number of pages: 20 + 1 appendices

Term and year of completion: Autumn 2017

---

The objective of this bachelor's thesis was to design, implement and test an electrical testing equipment prototype for production.

The progression of the thesis started by deciding the main components for the prototype circuitry simulation done virtually by LT Spice (2). After the simulation, the logic was programmed to the prototype, which was controlled by Raspberry Pi 3 (3). Lastly the prototype's test circuitry was assembled and tested.

The prototype functioned as commissioned by the orderer of this thesis. The prototype has been taken to a further development, so it will be a part of the production's test equipment series in the future.

---

Keywords: electrical, test, equipment, production



## **PREFACE**

This work was commissioned by MoveSole Ldt and it was done in their premises.

I would like to thank Vesa Kajanus and Markus Suvanto from MoveSole for their assistance in the work and Spektikor Company for lending their microscope and mini soldering iron.

Oulu, 20.11.2017  
Ville Varanka

**TABLE OF CONTENTS**

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
LIST OF ABBREVIATIONS	6
1 INTRODUCTION	8
2 DESIGN OF THE PROTOTYPE	10
2.1 Deciding main components	10
2.2 Simulation	12
2.3 Programming a logic for DAC	14
3 ASSEMBLY AND TEST OF THE SYSTEM	16
4 CONCLUSION	20
5 REFERENCES	21
6 APPENDICES	22

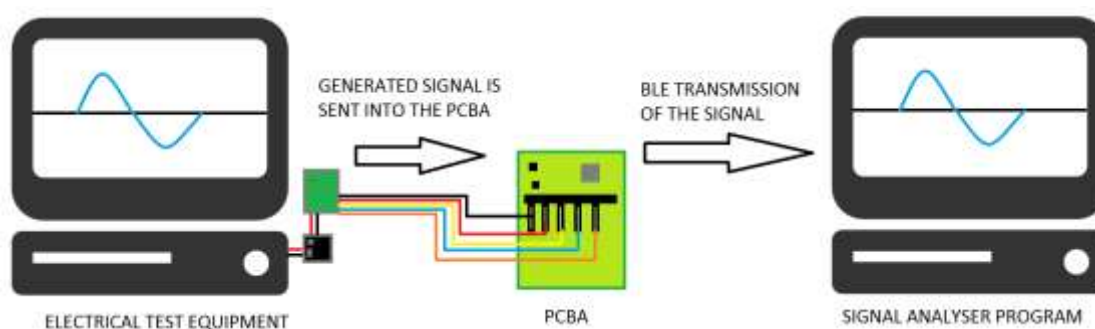
**LIST OF ABBREVIATIONS**

Bash	Bourne Again Shell
$\overline{\text{CLR}}$	Asynchronous Clear Input
CPU	Central Processing Unit
$\overline{\text{CS}}$	Serial Interface Chip Select/Load Input
DAC	Digital to Analog Converter
DIN	Serial Interface Data Input
GND	Ground
GPIO	General Purpose Input/Output
INV	Center Tap of Internal Scaling Resistors
LSB	Least Significant Bit
Ltd.	Limited company
MSB	Most Significant Bit
OP AMP	Operational Amplifier
PCBA	Printed Circuit Board Assembly
Rfb	Feedback Resistor

SCLK	Serial Interface Clock Input
V	Voltage
V <sub>dd</sub>	Supply Voltage
V <sub>out</sub>	Output Voltage
V <sub>ref</sub>	Reference Voltage

## 1 INTRODUCTION

The object of this thesis was to design, implement and test an electrical testing equipment prototype for production. The function of the test equipment is to test electrically the functionality of the Smart Insole's PCBA. The test equipment sends sine wave into PCBA's testing pads. If the wave measured from PCBA's output is identical to the original wave sent into PCBA's testing pads, the PCBA is working correctly (Figure 1).



*FIGURE 1. Desired functionality of the test system.*

Basically, the electrical test equipment is a programmable signal generator with multiple outputs. A signal generator is an electronic device that generates repeating or non-repeating analog signals in different shapes, like in sine, square or triangle form. Signal generators are used for designing, testing, troubleshooting and repairing electrical devices.

This thesis was done for MoveSole Ltd., whose main product is StepLab. StepLab is a package, which consists of the Smart Insole System (Figure 2) and other additional material for the right usage. The Smart Insole System is a measuring device for receiving instant force and timing information from selected points under the feet for the foot function and gait analysing. The product is intended for medical professionals and for short term home use purposes under supervision.



*FIGURE 2. MoveSole's Smart Insole System*

## 2 DESIGN OF THE PROTOTYPE

The designing of an electrical test equipment prototype is described in a chronological order via subsections below.

### 2.1 Deciding main components

The first problem was to decide what kind of signal generator would be best for the attended usage. The PCBA has seven test pads and each one needs a sine wave as an input. Conventional signal generators are expensive and they rarely have multiple outputs. They also rarely have any logic for the input segmentation.

It was decided that a 12-bit LTC2642 DAC (Figure 3) (1) will be used with the Raspberry Pi 3 model B single-board computer (Figure 4) (3).

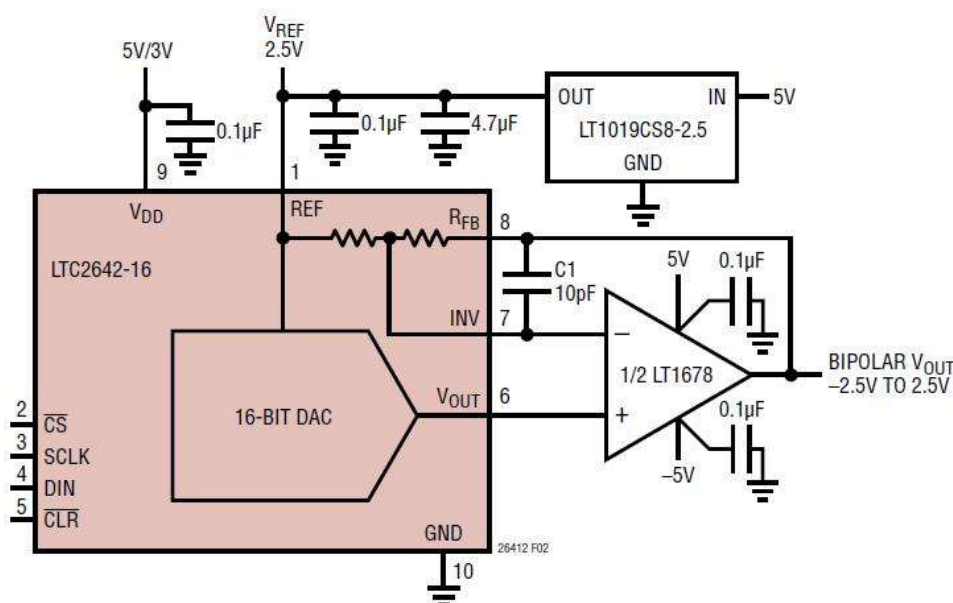


FIGURE 3. 16-bit LTC2642 DAC. Although 16 bits, the mechanism is identical to 12-bit DAC



FIGURE 4. Raspberry Pi 3 model B, a single-board computer

The Raspberry controls the DAC's inputs via GPIOs and can control at least seven DACs, which would be optimal for PCBA having seven test pads. When 12 bits are sent into the DAC, it calculates and outputs a voltage corresponding the received bit number, as seen in Table 1 (1). This output voltage is amplified, bipolarized and sent to PCBA's test pad.

TABLE 1. 12-bit Bipolar Offset Binary Code Table (LTC2642-12).

DIGITAL INPUT BINARY NUMBER IN DAC LATCH	ANALOG OUTPUT ( $V_{out}$ )
MSB      LSB	
1111 1111 1111 XXXX	$V_{ref} (2047/2048)$
1000 0000 0001 XXXX	$V_{ref} (1/2048)$
1000 0000 0000 XXXX	0V
0111 1111 1111 XXXX	$-V_{ref} (1/2048)$
0000 0000 0000 XXXX	$-V_{ref}$



## 2.2 Simulation

After the main components were decided, the circuit model of DAC had to be tested via simulation. This also enabled to test if the components were right for this purpose before they would be bought.

For simulation, LT Spice (2) was used. The main focus of the simulation was to simulate the logic and timing behind DAC that it would be understood and the test circuit model functionality would be proved (Figure 5).

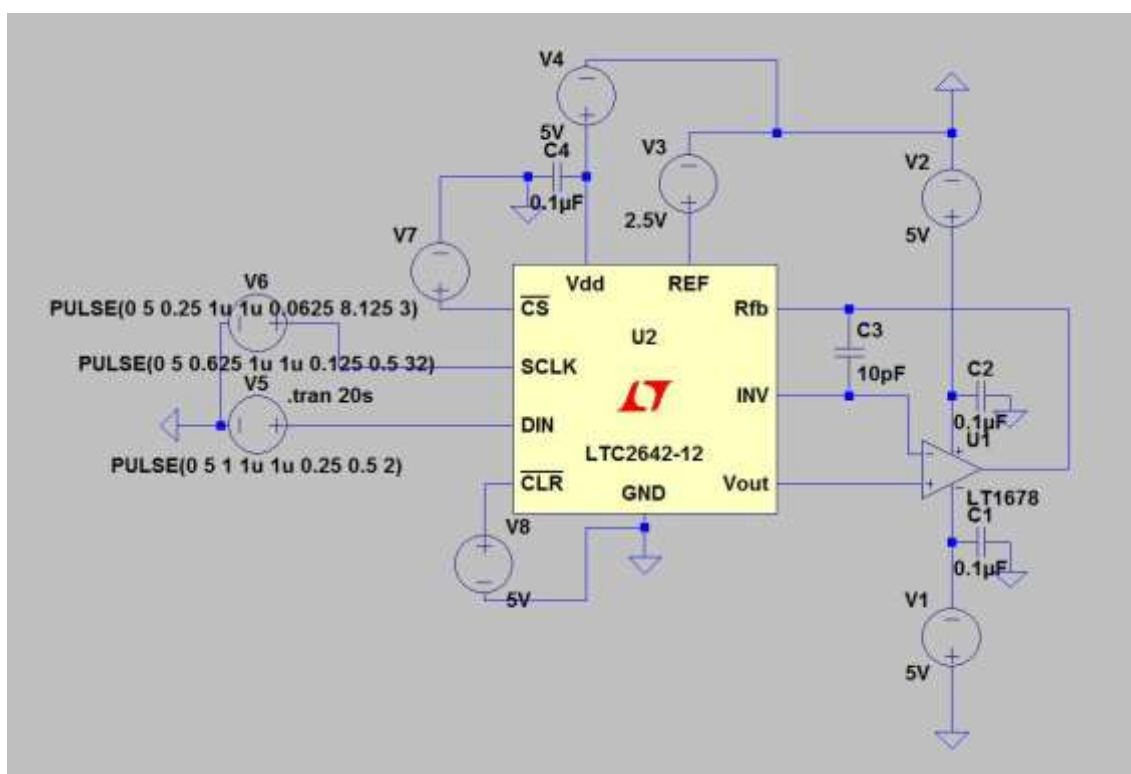


FIGURE 5. Circuit model for simulation

As seen in timing diagram for DAC in Figure 6. (1), the timing and logic are as follows:

- When  $\overline{\text{CS}}$  is low, SCLK is enabled for shifting in data on DIN. When  $\overline{\text{CS}}$  is taken high, SCLK is disabled, the 16-bit input word is latched and the DAC is updated

- Data from DIN is sent via rising edge of SCLK

It should be noticed that for a 12-bit DAC, the bits after LSB are “don’t-care” bits, thus it does not matter if they are high or low.

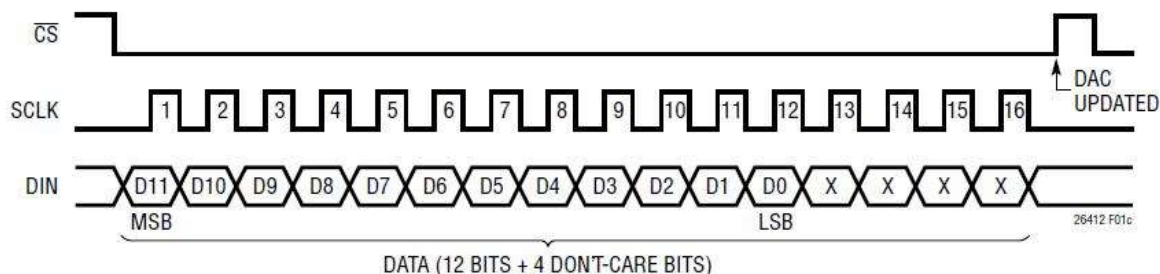


FIGURE 6. Timing diagram

The logic of timing diagram was tested via a virtual circuit model (Figure 5) done by the LT Spice. In the model, the  $V_{ref}$  is 2.5 V, therefore if DIN would receive a bit number 1111 1111 1111 XXXX,  $V_{out}$  from OP AMP would be 2.5 V and if the bit number was 0000 0000 0000 XXXX,  $V_{out}$  from OP AMP would be -2.5 V as described in Table 1 in the previous chapter.

When in the simulation DIN received bit numbers 0110 0000 0000 XXXX and 0000 0000 0000 XXXX and  $V_{out}$  from OP AMP was -0.625 V and -2.5 V (Figure 7), it confirmed that the simulation was a success and the DAC model worked as it should work.

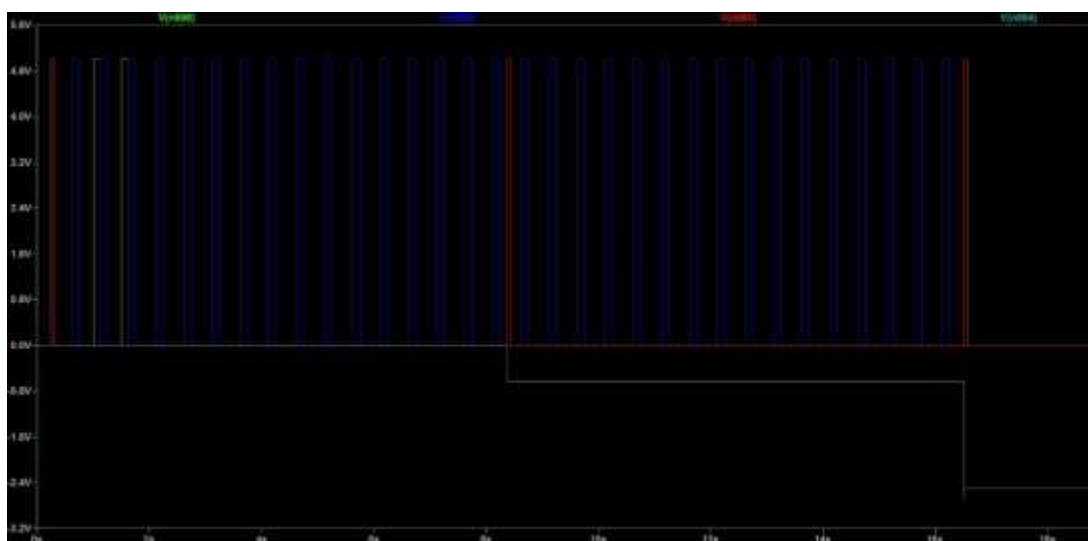


FIGURE 7. Simulation output (BLUE = SCLK, GREEN = DIN, RED =  $\overline{CS}$  and LIGHT BLUE = OP AMP  $V_{out}$ ).

### 2.3 Programming a logic for DAC

The next step was to make a program for Raspberry Pi 3 to control the DAC input ports, which are DIN, SCLK and  $\overline{CS}$ .

The first approach was to program three separate programs by Python (4) for controlling DIN, SCLK and  $\overline{CS}$ , which are controlled by Bash (5). The Bash shells were supposed to start and stop three Python scripts at the same time from a command (Figure 8). The scripts controlled Raspberry's GPIO pins, which output three different pulse wave signals into the DAC's inputs. The pulse waves for each program were programmed according to the DAC's timing diagram (Figure 6).

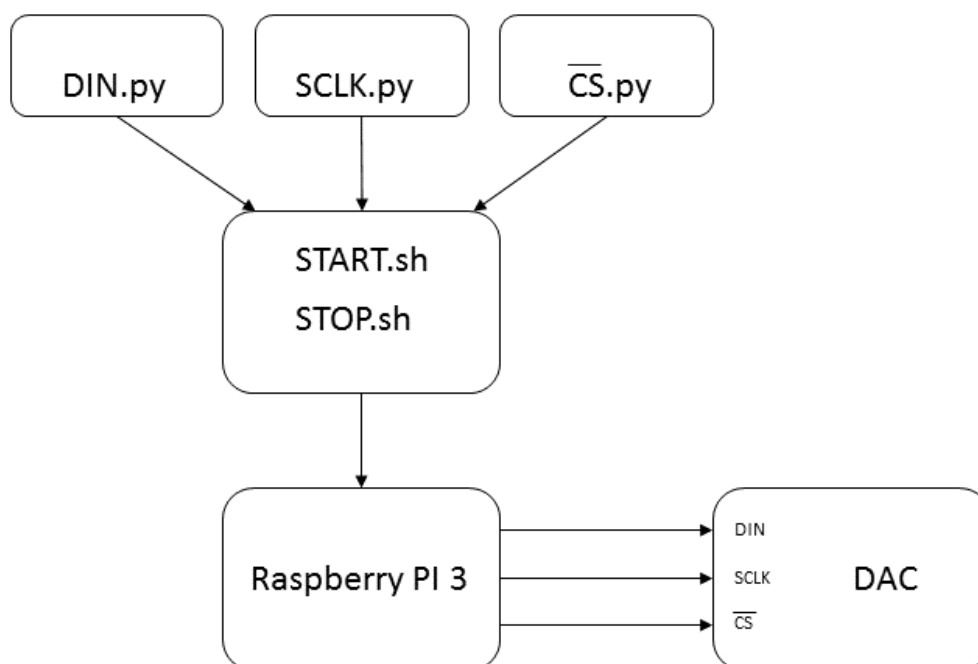
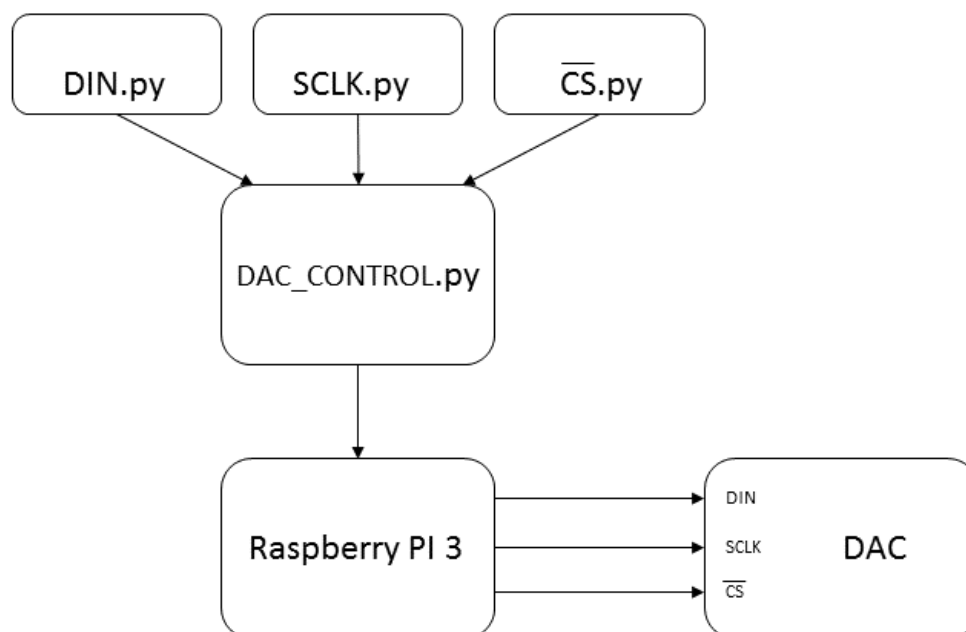


FIGURE 8. First attempt for program logic

It was noticed while testing the program that the timing between Python scripts was off. The scripts started as expected but after some time their timing was completely off. While investigating the problem, it was realized that the differences in timing were caused by the faulty architecture. It was not possible to run more than one Python script simultaneously with precise timing, because Raspberry is doing other processes in the background. These processes use the same clock as Python scripts, thus when more than one script are started to run at the same time, the processes will randomly jump between them, messing up the timing.

The solution for this problem was to make only one Python script for DAC controlling, thus the timing could not go wrong (Figure 9). The Bash shells were no longer needed because there was only one Python script to run. The final Python script after combining three scripts can be seen in Appendix 1.

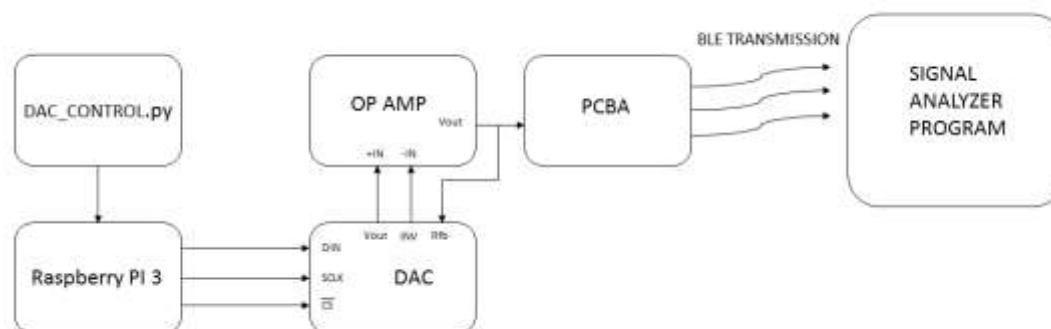


*FIGURE 9. Second and final attempt for program logic*

### 3 ASSEMBLY AND TEST OF THE SYSTEM

After all the steps for designing were done, it was time to assemble the prototype and test the entire system.

The architect for the system test environment can be seen below in Figure 10. The Raspberry controls the input pins of DAC. The DAC generates a sine wave, which is sent to the OP AMP for amplifying and bipolarity. The OP AMP sends the modified signal to PCBA, which sends it further wirelessly via BLE transmission to a signal analyzer program. If the signal received by the program is identical to the signal generated by the DAC, the system works.

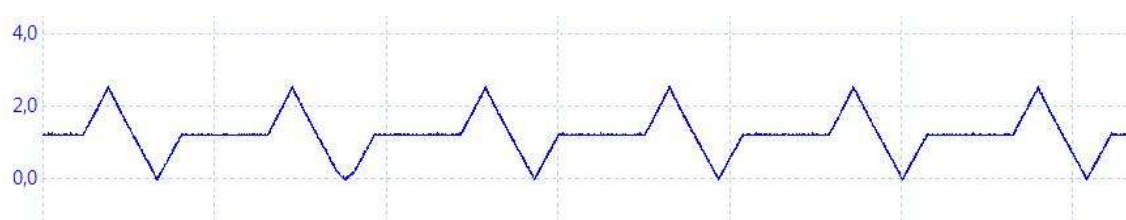


*FIGURE 10. System test environment*

The first thing needed to build for the test environment was a breadboard model, where Raspberry, DAC and OP AMP would be connected. DAC and OP AMP needed a separate board where their test wirings would be soldered and connected to the main breadboard where Raspberry's wirings would be located. For building the test wirings for DAC and OP AMP, a microscope and a mini soldering iron were used because the pins on DAC and OP AMP are in a microscopic size. There were some problems in the soldering process because of the unattachment of the wires on to the pins due to microscopic difficulties but in time the test wirings for DAC and OP AMP were finished.

The next step was to test that the DAC outputs the right signal. A PC oscilloscope named PicoScope (6) was used to plot the signal on the computer. As seen in Figure 11, the signal from DAC's  $V_{out}$  output was correct although it was not a sine wave yet. When the DAC's  $V_{out}$  was being tested at the time, the DAC control program (Figure 9) did not have a sinewave function. It was added later into the code (Appendix 1), as can be seen later in Figure 13. For the time the signal was generated by just counting numbers from 0 to 4096 (0000 0000 0000 XXXX to 1111 1111 1111 XXXX). Because of that, it made a triangular wave.

The signal does not either go under 0 V. This is a correct behaviour for DAC's  $V_{out}$  because for going bipolar from -2.5 to 2.5 V, the DAC's  $V_{out}$ , INV and Rfb must be connected into the OP AMP, as was seen earlier in Figure 3. in chapter 2.1. The bipolar output comes from the OP AMP's  $V_{out}$ .



*FIGURE 11.  $V_{out}$  signal from DAC*

When the functionality of the DAC was tested and proved to be working correctly, it was time to connect the OP AMP into the DAC and test the OP AMP's  $V_{out}$  signal. As can be seen in Figure 12, the signal became bipolar and went from -2.5 to 2.5 V as it should have been. This proved that Raspberry, DAC and OP AMP worked correctly together.

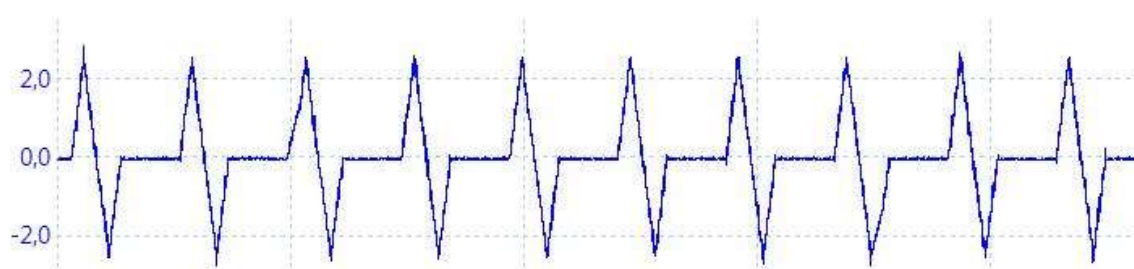


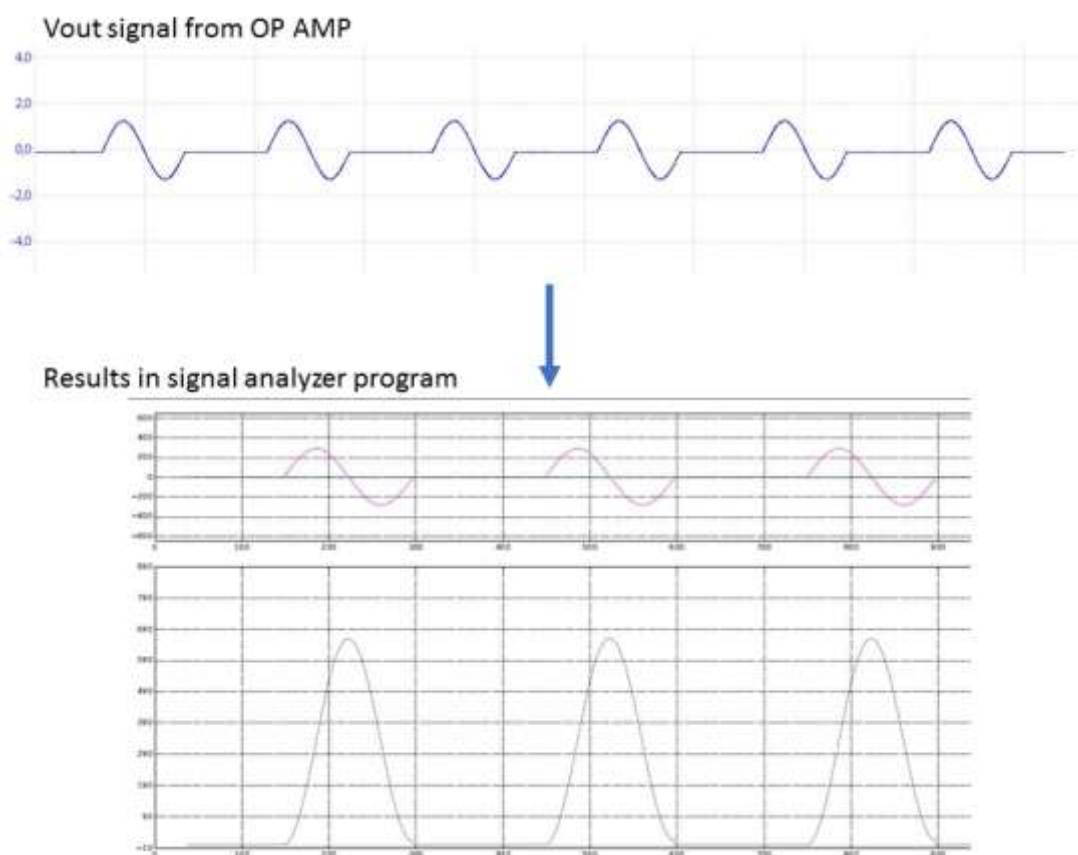
FIGURE 12.  $V_{out}$  signal from OP AMP

The next steps were to add a sinewave function into the DAC control program, connect the PCBA into the OP AMP and test the entire system.

When the system was tested the first time as a whole, the signal analyzer program did not seem to show a right wave form. It was noticed that the sine wave in the analyzer program went out of bounds. This was because the signal received by the PCBA was too powerful. It needed to be weakened via a voltage slicing.

After the slicing, the signal in the analyzer program was plotted out correctly although the signal was still little bit too powerful (Figure 13). This can be fixed via more powerful voltage slicing in the future.

In the end, the system test was a success and proved that the electrical test equipment prototype worked as planned.



*FIGURE 13. Results of testing the entire system. The bottom wave illustrates the force generated by PCBA from the received sinewave*



## 4 CONCLUSION

The main object of this thesis was to design, implement and test an electrical test equipment prototype. The prototype was a success and it proved that the electrical test equipment for testing PCBA is a plausible idea.

If MoveSole Ltd. wants to use this design in their future testing, some steps need to be addressed. A circuit board needs to be designed and built to control seven DACs and OP AMPS. The circuit can be controlled via one Raspberry PI 3. The voltage received by PCBA from OP AMP must be in a correct range, so that it can be measured correctly via a signal analyzer program. The Python script, which controls the DAC(s), must be changed to C-language code so that it does not use too much Raspberry's CPU capacity when seven DACs are being controlled. The Python script shown in Appendix 1 takes 10-13% of Raspberry's CPU capacity.

## 5 REFERENCES

24. Linear Technology LTC2641/LTC2642 datasheet. 2007. Date of retrieval 25.10.2017. <http://cds.linear.com/docs/en/datasheet/26412fd.pdf>
25. Linear Technology LT Spice. 2017. Date of retrieval 26.10.2017. <http://www.linear.com/designtools/software/#LTspice>
26. Raspberry Pi 3 model B. 2017. Date of retrieval 31.10.2017. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
27. Python, a programming language. 2017. Date of retrieval 31.10.2017. <https://www.python.org/>
28. Bash, a command processor. 2016. Date of retrieval 31.10.2017. <https://www.gnu.org/software/bash/>
29. Pico Technology. 2017. Date of retrieval 7.11.2017. <https://www.picotech.com/>

## 6 APPENDICES

Appendix 1 Program code for DAC control

```

1 import RPi.GPIO as GPIO
2 import time
3 import math
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 GPIO.setup(14,GPIO.OUT)
7 GPIO.setup(15,GPIO.OUT)
8 GPIO.setup(4,GPIO.OUT)
9
10 def dinFunction(): #Function for DIN table
11     sequence1 = []
12     sequence2 = []
13     bitnumber = []
14     sinewave = []
15     delay = [0,0]
16     dontcare = [1,1,0]
17
18     for y in range(0,100,1): #Generate a sinewave table and switch float numbers into integers
19         t = (2*math.pi/100)*y
20         z = int(2048*(math.sin(t))*2048)
21         if (z > 4096):
22             z = 4095
23         sinewave.extend([z])
24
25     sinewave.extend([2048])
26
27     for y in sinewave: #Switch integers into 12-bits
28         del bitnumber[:]
29         del sequence1[:]
30         binary = [int(x) for x in format(y, '012b')]
31         for z in binary:
32             if z == 0:
33                 bitnumber.extend([0,0,0])
34             else:
35                 bitnumber.extend([1,1,0])
36         sequence1 = bitnumber + dontcare*4 + delay
37         sequence2.extend(sequence1)
38
39     return sequence2
40
41 def sclkFunction(): #Function for SCLK table
42     delay = [0,0]
43     clock = [0,1,0]
44     sequence = (clock*16 + delay)*101
45     return sequence
46
47 def csFunction(): #Function for CS table
48     delay = [0]
49     chipselect = [1,0]
50     sequence = (delay*48 + chipselect)*101
51     return sequence
52
53 sclk = sclkFunction()
54 din = dinFunction()
55 cs = csFunction()
56
57 while (True):
58
59     for i in range(0, len(cs)): #Control GPIOs with the values of tables
60         sample_time = time.time()
61
62         GPIO.output(4,sclk[i])
63         GPIO.output(14,din[i])
64         GPIO.output(15,cs[i])
65
66         sleep = (200 - (time.time() - sample_time))/1000000
67         time.sleep(sleep)
68
69     time.sleep(0.5) #Gap between waves

```