

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Edwin Weber 154992IALB

**SAGEDUSDETEKTSIOONIL PÕHINEV
REAALAJA DIGITAALNE
HELISÜNTESAATOR**

Bakalaureusetöö

Juhendaja: Julia Berdnikova
Tehnikateaduse
magister, teadur

Kaasjuhendaja: Sander Ulp
Tehnikateaduse
doktor, nooremteadur

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Edwin Weber

20.05.2019

Annotatsioon

Käesolevas bakalaureusetöös on kirjeldatud reaalaja polüfoonilise kitarrisüntesaatori ehitamist mikrokontrolleriga arendusplatvormi ja programmeeritava heligeneraatori baasil. Töö raames arendatud seade koosneb sisendsignaali eelvõimendist, mis sobitab sisendit mikrokontrolleri analoog-digitaalmuunduriga, mikrokontrollerist, programmeeritavast heligeneraatorist ja sellega ühendatud digitaal-analoogmuundurist, madalpääsfiltrist ning väljundsignaali eelvõimendist.

Signaali digitaliseerimise ja töötlemise otstarbeks oli valitud arendusplatvorm Teensy 3.6, mille tuumaks on arhitektuuril ARM Cortex-M4 põhinev Kinetis K66 seeria mikrokontroller. Mikrokontroller töötleb digitaliseeritud signaali tarkvaraliselt, teostades kiiret Fourier' teisendust (FFT) ning analüüsides signaali sageduslikku spektrit, kusjuures sageduste detekteerimise osas rakendatakse interpolatsiooni.

Sageduste detekteerimise järel toimub uue helisignaali süntees programmeeritava heligeneraatori poolt. Heligeneraatorina on kasutatud 9-kanalilist mikroskeemi Yamaha YM3812, mis on võimeline teostama sagedusmodulatsioonsünteesi. Seadme töö ajal on võimalik muuta Bluetooth-ühenduse kaudu sageduste detekteerimise tundlikkust ja sünteesitava heli parameetreid, nagu lainekuju ja mähisjoon.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 53 leheküljel, 9 peatükki, 19 joonist, 6 tabelit.

Abstract

Frequency detection-based real-time sound synthesizer

The following bachelor's thesis describes the development of a real-time polyphonic guitar synthesizer based on a microcontroller-based development platform and a programmable sound generator. As a practical part of this work, a device consisting of an input signal pre-amplifier, a microcontroller, a programmable sound generator, a low-pass filter and an output signal pre-amplifier was built.

For the signal digitizing and processing purposes, Teensy 3.6 development platform containing an ARM Cortex-M4 architecture-based Kinetis K66 family microcontroller was chosen. The microcontroller processes the digitized signal by performing Fast Fourier Transform (FFT) and analyzing its frequency spectrum. The frequency detection part of the task involves interpolation of detected frequencies and amplitudes.

After the frequencies are detected, a new sound signal is synthesized by a programmable sound generator, the Yamaha YM3812 9-channel sound chip capable of frequency modulation synthesis. It is possible to change frequency detection sensitivity and the parameters of synthesized sound, such as waveform and envelope, via Bluetooth connection while the device is operating.

The thesis is in Estonian and contains 53 pages of text, 9 chapters, 19 figures, 6 tables.

Lühendite ja mõistete sõnastik

ADC	<i>Analog-to-digital converter</i> , analoog-digitaalmuundur
ADSR	<i>Attack, Decay, Sustain, Release</i> , helisignaali mähisjoone faasid
ARM	RISC arhitektuuride seeria
Bluetooth	Standard seadmete omavahelise juhtmevaba side jaoks
Cortex	ARM mikroarhitektuuride seeria
CMSIS	<i>Cortex Microcontroller Software Interface Standard</i> , Cortex mikrokontrollerite tarkvaraliidese standard
DAC	<i>Digital-to-analog converter</i> , digitaal-analoogmuundur
DMA	<i>Direct memory access</i> , otsemällupöördus
DSP	<i>Digital signal processing</i> , digitaalne signaalitöötlus
EAGLE	Elektriskeemide ja trükkplaatide modelleerimise tarkvarapakett
FM	<i>Frequency modulation</i> , sagedusmodulatsioon
FPU	<i>Floating point unit</i> , ujukomaplokk
MIDI	<i>Musical Instrument Digital Interface</i> , muusikariistade digitaalne liides
OPL2 (OPL II)	<i>FM Operator Type-L II</i> , sagedusmodulatsiooni operaator, tüüp L, II versioon
OV	Operatsioonivõimendi
RISC	<i>Reduced Instruction Set Computer</i> , kärbitud käsustikuga arvuti
Teensy	USB-liidese kaudu programmeeritavate mikrokontrolleri arendusplatvormide seeria
USB	<i>Universal Serial Bus</i> , universaalne järjestikliides
YM3812	Firma Yamaha poolt toodetud programmeeritava heligeneraatori mikroskeem, enam tuntud kui OPL2

Sisukord

1 Sissejuhatus	9
2 Olemasolevate lahenduste ülevaade	11
3 Tehniline ülesanne	14
4 Seadme struktuur	16
4.1 Sisendsignaali eelvõimendi	19
4.2 Programmeeritav heligeneraator ja digitaal-analoogmuundur	21
4.3 Bluetooth-moodul	23
4.4 Väljundsignaali madalpääsfilter ja eelvõimendi	24
5 Sisendsignaali töötlemine	26
5.1 Signaali digitaliseerimine	26
5.2 Spektri analüüs	28
5.3 Sageduste edastamine heligeneraatorile	32
6 Programmeeritava heligeneraatori kasutamine	34
7 Seadme juhtimine	38
8 Tulemuste analüüs	42
9 Kokkuvõte	47
Kasutatud kirjandus	49
Lisa 1 – Funktsioon <i>findMaxValues</i>	52

Jooniste loetelu

Joonis 1. Arendusplatvorm Teensy 3.6 (väljaviikudega), vaade ülevalt ja alt.	15
Joonis 2. Seadme plokk skeem.	16
Joonis 3. Seadme elektriline skeem.	17
Joonis 4. Kokkupandud skeem makettplaadil.	19
Joonis 5. Programmeeritav heligeneraator YM3812.	21
Joonis 6. Bluetooth-moodul JDY-30, vaade eest ja tagant.	24
Joonis 7. Puhvri osa töötlev funktsioon <i>processADCInput</i> ning abifunktsioon <i>bufferCopyPart</i>	30
Joonis 8. Sageduse F -arvuks teisendamist teostav funktsioon.	32
Joonis 9. Heligeneraatori YM3812 kanali sagedust määrav funktsioon.	33
Joonis 10. Funktsiooni <i>YM3812::channelSetAmpl</i> lähtekood.	33
Joonis 11. ADSR-mähisjoone faasid [16], [35].	35
Joonis 12. Heligeneraatori YM3812 operaatorite väljundi võimalikud lainekujud [16].	36
Joonis 13. Näide andmevahetusest Bluetooth-liidese kaudu.	38
Joonis 14. Spektri maksimumide graafik (sisendiks on kitarril esitatud akord). Sinised ringid tähistavad muutmata spektris esinevaid maksimume ning punaste ristidega on tähistatud algoritmi töö tulemusena leitud maksimumid.	42
Joonis 15. Spektri maksimumide graafik (sisendiks on siinussignaali sagedusega 440 Hz). Sinised ringid tähistavad muutmata spektris esinevaid maksimume ning punaste ristidega on tähistatud algoritmi töö tulemusena leitud maksimumid.	43
Joonis 16. Ajaline viide sisend- ja väljundsignaali vahel. Horisontaaltelje ühele ruudule vastab 25 ms.	44
Joonis 17. Süntesaatori sisend- ja väljundsignaali graafik (polüfoonilises režiimis).	45
Joonis 18. Süntesaatori sisend- ja väljundsignaali graafik (monofoonilises režiimis). ...	45
Joonis 19. Sisendsignaali (üleval) ja väljundsignaali (all) spektrite graafikud.	46

Tabelite loetelu

Tabel 1. Skeemi komponentide maksumus.	18
Tabel 2. Mikroskeemi YM3812 kasutatavate väljaviikude loetelu [16], [17].....	22
Tabel 3. Heligeneraatori YM3812 operaatorite parameetrid [16].	35
Tabel 4. Heligeneraatori YM3812 operaatorite sageduste kordajate vastavused [31]...	36
Tabel 5. Heligeneraatori YM3812 kanali parameetrid [16].	37
Tabel 6. Seadme juhtimise käskude loetelu.....	39

1 Sissejuhatus

Esimesed efektiplokid, mille eesmärgiks oli muuta kitarriheli mingil viisil, ilmusid juba aastal 1938, mil hakati tootma elektrikitarris sisseehitatud mootoriga, mis muutis keelte pinget, tekitades vibraato efekti. Efektipedaalide võidukäik algas 1960. aastatel, kui elektriskeemides hakati laialt kasutama transistoreid [1].

Suureks sammuks, mis võimaldas kitarrisüntesaatorite loomist, oli sageduse pingeks teisendamise tehnoloogia leiutamine. Selle alusel hakkas firma *Roland* tootma alates 1977. aastast süntesaatorit GR-500 ja sellega ühilduvat kitarrimudelit GS-500, mis oli varustatud heksafoonilise ehk 6-kanalilise helipeaga. Helipea võimaldas iga keele kõla eraldi töötlemist, kuid muusikud pidid kitarris ühendama eraldiseisva süntesaatoriga [2].

Suurem osa 1970. aastatel toodetud süntesaatoritest (siin mõistetakse elektroonilisi klahvpille) põhines analoogsel skeemitehnikal, millel olid teatud puudused. Analoogetel süntesaatoritel ei olnud võimalust seadistuste salvestamiseks, need olid üldjuhul võimelised sünteesima ainult monofoonilist heli ja läksid kergesti häälest ära [3].

Sagedusmodulatsioonisüntees (FM-süntees), mida hakati katsetama 1960-ndate aastate lõpus, osutus soodsamaks helisünteesi võimaluseks, sest seda kasutatavat süntesaatorit oli võimalik teha digitaalseks, mis lahendas eelmainitud probleemid. FM-süntees võimaldas signaali moodustamist ainult valitud harmooniliste komponentidega, mis oli suureks eeliseks võrreldes aditiivse (liitmissünteesi) ja subtraktiivse sünteesiga (lahutussünteesiga), kus harmooniliste komponentide muutmiseks kasutatakse filtreerimist. Skeemitehnika lihtsustamine tegi võimalikuks ja soodsamaks süntesaatorite masstootmist, mis panustas nende laialdasse kasutamisse [3]. FM-sünteesi võib pidada vahepealseks etapiks enne seda, kui mälu mikroskeemide suurenenud maht võimaldas reaalse muusikariistade salvestuste kasutamist sünteesiks. Kitarrisüntesaatorite helisünteesi võimalused oluliselt laienesid tänu MIDI-liidese (*Musical Instrument Digital Interface*) leiutamisele 1980. aastatel, kusjuures hakati tootma ka seadmeid, mis teisendavad helisignaali kõrgust otse MIDI-käskudeks [2].

Enamik praegu saadavalolevaid kitarrisüntesaatoreid, mis on toodetud suurte helitehnikafirmade poolt, on huviliste jaoks kallid. Näiteks Saksamaa veebipoes Thomann pakutavate mudelite hinnad mai 2019 seisuga algavad ligikaudu 400 eurost. Üldjuhul see hind ei sisalda nende süntesaatoritega kasutamiseks mõeldud heksafoonilise helipea maksumust [4].

Soodsamaks lahenduseks oleks *guitar-to-MIDI* konverteri kasutamine, kuid see ei suuda iseseisvalt heli genereerida ning seda on vaja ühendada välise seadmega, mis töötleb MIDI-käskke. Tavaliselt selle seadme rolli täidab eraldiseisev süntesaator või arvuti. Kui vaadelda sünteesi jaoks arvuti kasutamist, siis tuleb tunnustada, et kaasaegsete arvutite arvutusvõime võimaldab signaali töötlemist tarkvaraliselt, ilma välise konverterita. Integreeritud helikaardi olemasolul kaob välise seadmete kasutamise vajadus üldse. Ometi mitte igal muusikul on võimalus esinemistel kasutada arvutit või süntesaatorit. Pikad vahemaad, mida tuleb läbida ühest linnast teisse reisival artistil, ei julgusta kallist ja mahukat varustust kaasas kandma. See põhjustab vajadust soodsa ja portatiivse helisüntesaatori arendamise järele, mille kasutamine ja transportimine oleks kerge ja mugav.

2 Olemasolevate lahenduste ülevaade

Käesolevas peatükis antakse lühiülevaade olemasolevate kitarrisüntesaatorite realisatsioonidest, nii kauplustes kättesaadavatest kui ka huviliste isetehtutest.

Kõige soodsamaks viisiks jääb arvuti kasutamine koos helikaardiga, kui need on juba saadaval, kuna praeguste arvutite kiiretoimelisus võimaldab reaajas helisignaali kvaliteetset töötlemist. Riistvara olemasolul sõltub lahenduse hind tarkvara hinnast, kusjuures eksisteerivad ka tasuta või vabavaralised lahendused, näiteks projekti *Fretted Synth* raames loodud pistikprogrammid [5].

- Sonuus G2M V3
 - Hind: 77 €.
 - Teisendab helisignaali MIDI-käskudeks, ühildub MIDI-liidesega; sobib kitarrile, basskitarrile, häälele ja muude muusikariistade jaoks; ei nõua erilise helipea paigutamist; monofooniline väljund.
 - Link: https://www.thomann.de/intl/ee/sonuus_g2m_v3.htm
- Seymour Duncan Fuzz – Analog Fuzz Synth Pedal
 - Hind: 368 €.
 - Analoofiltritel põhinev skeem mitmete efektidega.
 - Link:
https://www.thomann.de/intl/ee/seymour_duncan_fuzz_analog_fuzz_synth_pedal.htm
- Boss GP-10S
 - Hind: 399 € (komplekt Boss GP-10GK koos elektrikitarrile helipeaga – 479 €).

- Nõuab helipea paigaldamist kitarrile; ühildub USB- ja MIDI-liidestega; modelleerib reaalse muusikariistade helinat.
- Link: https://www.thomann.de/intl/ee/boss_gp_10s.htm
- Roland GR-55S
 - Hind: 519 € (komplekt Roland GR-55GK koos elektrikitarrilise helipeaga – 599 €, basskitarrilise helipea – 99 €)
 - 910 sisseehitatud helinat, millest võib kaks korraga valida; võimendite modelleerimine; ühilduvus USB- ja MIDI-liidestega; võimalus salvestada ja taasesitada kuni 20 sekundit audiot tänu *looper*’ile.
 - Link: https://www.thomann.de/intl/ee/roland_gr55s_black.htm
- Boss SY-300 Guitar Synthesizer
 - Hind: 666 €
 - Ei vaja erilist helipead; sisseehitatud efektiplokid (neist võib kuni 4 korraga kasutada), võimaldab signaali filtreerimist, võimendite modelleerimist ja madalsageduslike ostsillaatorite (*Low Frequency Oscillator*, LFO) rakendamist; töötab ka basskitarrilise ja teiste elektrooniliste muusikariistadega; ühilduvus USB- ja MIDI-liidestega.
 - Link: https://www.thomann.de/intl/ee/boss_sy_300_guitar_synthesizer.htm
- Into the Unknown – Guitar Synth kit
 - Hind: alates 43 €
 - Komplekt isetegemiseks; monofooniline väljund; seade sisaldab modulatsiooniefekte ja *octaver*’it.
 - Link: <https://www.musikding.de/Into-the-Unknown-Guitar-Synth-kit>

Kokkuvõtmiseks olid vormistatud valminud seadmele esitatavad nõuded. Süntesaator peab olema võimeline ise heli sünteesida, olema polüfooniline ning selle komponentide maksumus peab jääma alla eelmainitud toodete keskmise hinna.

3 Tehniline ülesanne

Töö teemaks oli valitud reaalaaja digitaalse helisüntesaatori välja töötamine. Valminud seadme peamiseks kasutusala on ette nähtud elektrikitarriga signaali töötlemine, kuid võimalik on ka teiste analoogsignaallikate kasutamine, näiteks mõni teine muusikariist, arvuti helikaart või telefoni kõrvaklapiväljund.

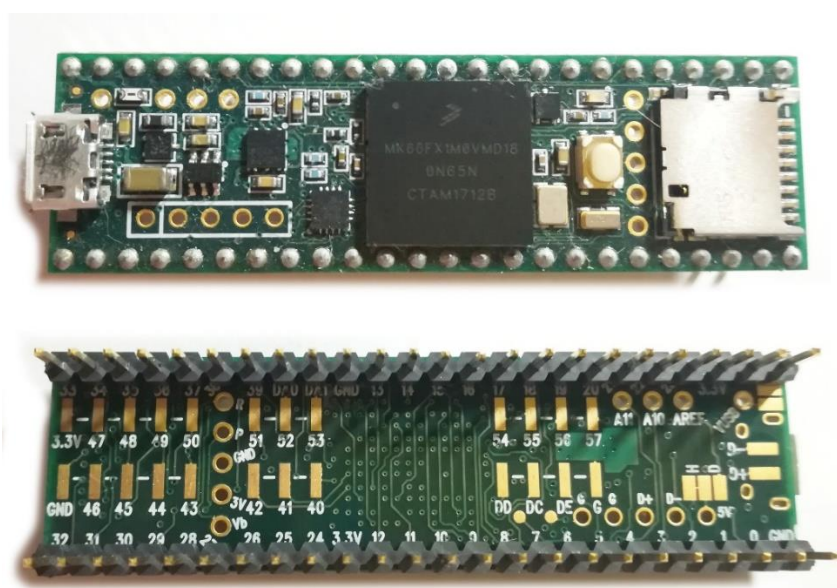
Eesmärgiks on pideva helisignaali digitaalne töötlemine reaajas ning uue helisignaali süntees samasuguse helikõrgusega ja teistsuguse, muudetava lainekujuga. See eeldab signaali amplituudspektri analüüsi, et oleks võimalik vastavate sagedustega heli sünteesi teostada. Määravaks parameetrik helitöötlemisel on signaali töötlemise aeg. Suur ajaline viide võib väljenduda signaali soovimatu faasinihke (juhul, kui väljundsignaal sisaldab mingil määral lähtesignaali) või kaja kujul. Ühe uuringu järgi viide üle 15 ms heli tekitamise ja taasesitamise vahel on juba inimese jaoks tajutav; kui see ületab 20 ms, võib tekkida raskusi tempos püsimisel, samas ka teiste muusikariistadega kaasamängimisel [6]. Seega seadme poolt tekitav viide peab olema minimaalne.

Samas peab helitöötlemise puhul sageduste eristamise täpsus olema piisavalt suur, et oleks võimalik eristada eri noote. Sellel põhjusel oli otsustatud keskenduda digitaalsele signaalitöötlemisele, kusjuures töötlemise protsess sõltub rohkem tarkvarast kui riistvarast. Juhul, kui on tarvis põhjalikke muudatusi signaali analüüsi protsessi sisse viia, on tarkvaras muudatuste tegemine palju lihtsam kui elektriskeemi ümberprojekteerimine.

Kuigi signaalitöötlemise otstarbeks eksisteerivad digitaalsed signaaliprotsessorid (DSP), mille käsustikud on optimeeritud helitöötlemiseks, nende soetamine tükikaupa iseseisvalt teostatava projekti otstarbeks on keeruline. Sellepärast otsustati valida mikrokontrolleri, mis oleks kergesti kättesaadav, hästi dokumenteeritud ja laialt kasutatud. Programmiteekide olemasolu oluliselt kergendab tööd mikrokontrolleriga, kõrvaldades vajadust laialt kasutatavaid funktsioone uuesti kirjutada ja soodustades programmeerimisvõtete õppimist läbi lähtekoodi analüüsi.

Autoril oli varasem kogemus Arduino Uno arendusplatvormiga, mis ei sobinud püstitatud ülesande täitmiseks väikese mälumahu ja ebapiisava kiiretoimelisuse tõttu. Uue platvormi otsimisel peeti oluliseks integreeritud analoog-digitaalmuunduri olemasolu ja ühilduvust Arduino Uno jaoks kirjutatud programmikoodiga.

Seega oli projekti jaoks valitud arendusplatvorm Teensy 3.6, mille tuumaks on arhitektuuril ARM Cortex-M4 põhinev mikrokontroller Kinetis MK66FX1M0VMD18 (seeria K66) taktsagedusega 180 MHz [7]. Selle peamiseks eeliseks on ujukomaplaki olemasolu, mis võimaldab teatud juhtudel tehteid ujukomaarvudega teostada kiiremini kui täisarvulisi operatsioone kasutades optimeeritud funktsioone [8].



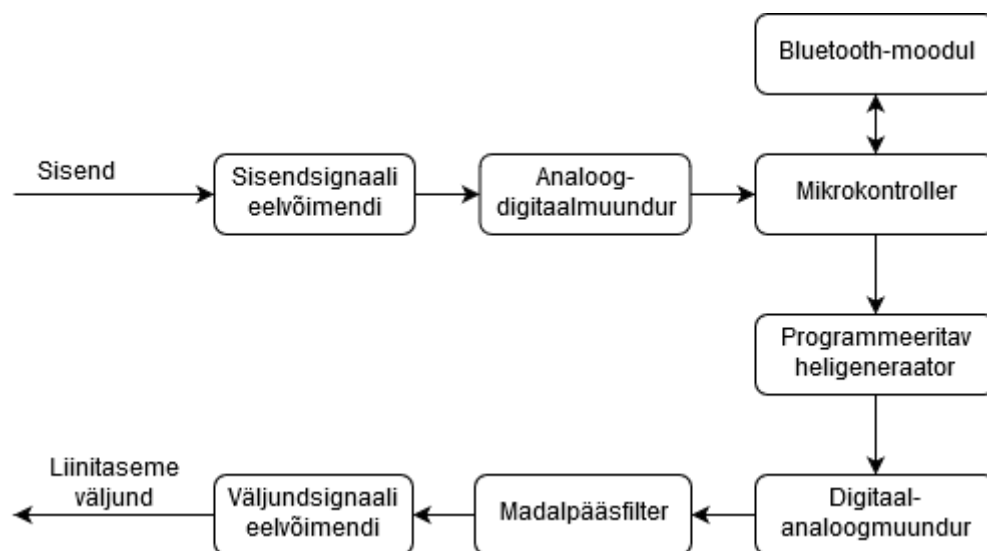
Joonis 1. Arendusplatvorm Teensy 3.6 (väljaviikudega), vaade ülevalt ja alt.

Samuti see mikrokontroller sisaldab kahte 16-bitist analoog-digitaalmuundurit, mis võimaldavad signaali diskreetimist kiirusega kuni 1 200 000 punkti sekundis (kuni 13-bitise resolutsiooniga) või kuni 461 467 punkti sekundis (16-bitise resolutsiooniga) [9]. Lisaks on saadaval kaks digitaal-analoogmuundurit, mis võimaldavad helisignaali genereerimist otse mikrokontrolleris, kuid käesoleva töö raames neid ei kasutata.

Signaali töötlemise järel on vaja teostada uue helisignaali sünteesi teatud sagedustega. Selleks oli otsustatud programmeeritava heligeneraatori kasutamise kasuks eelneva kogemuse põhjal ning eesmärgiga anda arendatavale seadmele omapärase „retrokõla“. Selle kasutamine skeemis on kirjeldatud peatükis 4.2 ning programmeerimine peatükis 6.

4 Seadme struktuur

Töö käigus oli välja töötatud püstitatud ülesannet täitev seade. Seda võib tingimisi jagada plokkideks, mille vahelised seosed on visualiseeritud allpool toodud plokk skeemina.



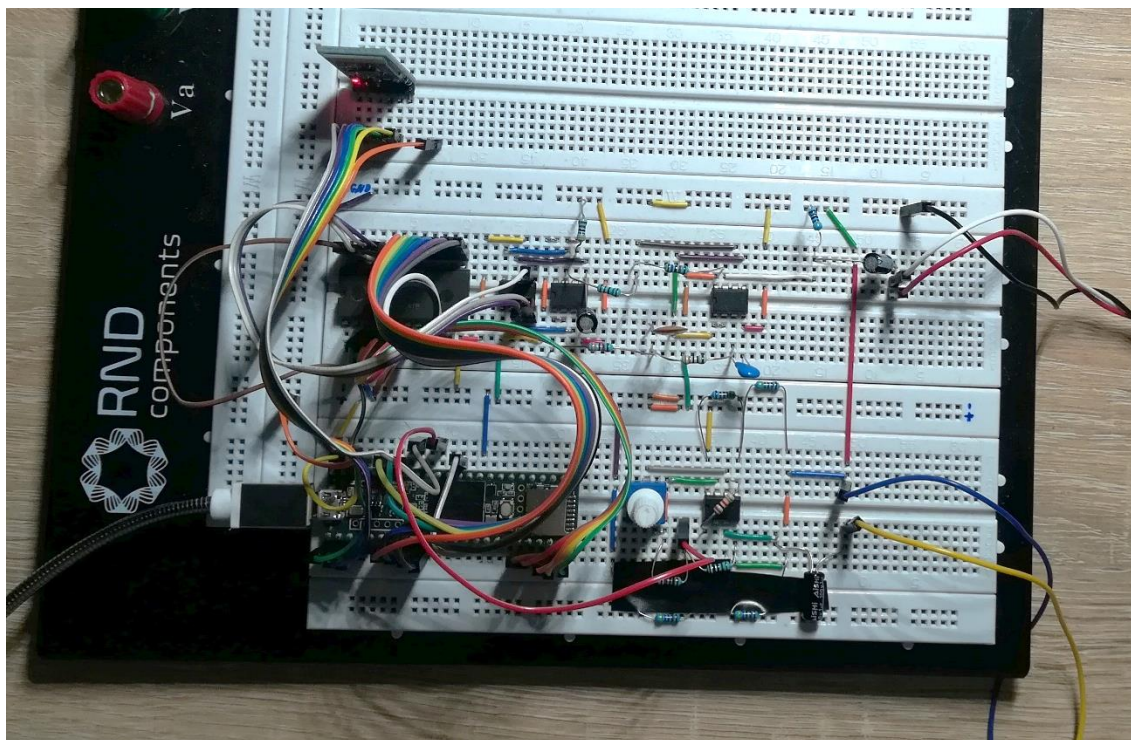
Joonis 2. Seadme plokk skeem.

Lähtuvalt sellest oli koostatud ja makettplaadil kokku pandud elektriline skeem, mille ülesehitust kirjeldatakse järgnevas alamtükkides. Elektrilise skeemi joonis oli tehtud modelleerimistarkvara EAGLE abil.

Tabel 1. Skeemi komponentide maksumus.

Tähistus	Komponent	Kogus (tk)	Kauplus	Hind (EUR)
R1, R2, R4, R5	Takisti, 47 k Ω , 0,25 W	4	Oomipood	0,80
R3	Takisti, 1 k Ω , 0,25 W	1	Oomipood	0,20
R6	Trimmer, 10 k Ω , horisontaalne, 100 mW	1	Oomipood	0,30
R7	Takisti, 20 k Ω , 0,6 W, 1%	1	Oomipood	0,20
R8	Takisti, 10 k Ω , 0,6 W, 1%	1	Oomipood	0,20
R9, R14	Takisti, 4,7 k Ω , 0,25 W	2	Oomipood	0,20
R10, R11	Takisti, 33 k Ω , 0,25 W	2	Oomipood	0,40
R12	Takisti, 15 k Ω , 0,6 W, 1%	1	Oomipood	0,20
R13	Takisti, 27 k Ω , 0,6 W, 1%	1	Oomipood	0,20
C1, C5	Elektrolüüt kondensaator, 1 μ F, 63 V	2	Oomipood	0,40
C2	Elektrolüüt kondensaator, 10 μ F, 100 V	1	Oomipood	0,20
C3	Keraamiline kondensaator, 510 pF, 50 V	1	Oomipood	0,20
C4	Keraamiline kondensaator, 220 pF, 50 V	1	Oomipood	0,20
IC1	Mikroskeem Yamaha YM3812	1	eBay.de	1,07
IC2	Mikroskeem Yamaha YM3014B	1	eBay.de	0,99
IC3, IC4, IC5	Operatsioonivõimendi NE5532P (20 tüki kaupa)	3	eBay.de	1,00
	Arendusplatvorm Teensy 3.6 (väljaviikudega)	1	YEInternational	50,70
	Bluetooth-moodul JDY-30	1	eBay.de	2,01
	Kokku			59,47

Kõige rohkem komponente kulus analoogse sisendsignaali sobitamiseks analoog-digitaalmuunduriga ning sünteesitud signaali sobitamiseks tarbijaga. Skeemi koostamine makettplaadil võimaldas arendusetapil veenduda selle töövõimelisuses ning vajadusel komponente asendada. Makettplaadil koostatud seadet näitab Joonis 4.



Joonis 4. Kokkupandud skeem makettplaadil.

Näidatud on skeem tervikuna, kusjuures sisendsignaali allikas on ühendatud kollase juhtme kaudu ja maandatud sinise juhtme kaudu (mõlemad paremas alumises nurgas). Järgnevates alampeatükkides vaadeldakse teatud skeemi osi ja nende vahelisi ühendusi.

4.1 Sisendsignaali eelvõimendi

Kitarri helipea koosneb püsimagnetist, mida ümbritseb mähis. Keele võnkumisel tekitab elektromagnetiline induktsioon elektrivoolu mähises. Sõltuvalt sellest, mis suunas keel liigub helipea suhtes, omab elektrikitarrist tulenev elektrisignaali nii positiivset, kui ka negatiivset pinget erinevatel ajalõikudel [10].

Kinetis K66 seeria mikrokontrollerites sisalduv analoog-digitaalmuundur (ADC, *analog-to-digital converter*) on siiski võimeline töötama ainult positiivsete pingeväärtustega. Seda võib järeldata andmelehes toodud muunduri madalama tugipinge suuruselt V_{REFL} , mis vastab digitaliseeritud signaali nivoole 0 ja võrdne maanduspotsiaaliga V_{SSA} . Kõrgema tugipinge suurus V_{REFH} jääb vahemikku alates 1,13 V kuni toitepinge suuruseni V_{DDA} , mis ei ületa 3,6 V [9]. Tarkvaraliselt aga saab kõrgemaks tugipingeks valida 1,2 V või 3,3 V. Ülaltoodust järeldatakse, et sisendsignaali on vaja ADC-ga sobitada, näiteks alaliskomponendiga liitmise teel.

Skeemis realiseeritud eelvõimendi põhineb mikroskeemil NE5532P (kuigi skeemil on tähistatud NE5532N, tähed erinevad tootja ja mikroskeemi korpuse järgi [11] [12]), mis sisaldab kahte operatsioonivõimendit. Operatsioonivõimendi väljundpinge vahemik on piiratud selle toitepingega [13]. Sellest järgnevalt väljundamplituudide vahemik jääb 0 V ja +5 V vahele, mis omalt poolt samuti põhjendab signaali alaliskomponendiga liitmise vajadust.

Esimene operatsioonivõimendi (skeemil tähistatud IC3A) on kasutatud pingejärgurina, mis vähendab koormust signaallialikale tänu sisendite suurele takistusele. Pingejärguri võimendustegur on 1, mis tähendab, et väljundpinge on võrdne sisendpingega.

Eralduskondensaator (tähistus C1) ei võimalda alalisvoolu sattumist signaallialikasse, kuid laseb läbi vahelduvsignaali. Takistid R1 ja R2 moodustavad pingejaguri. Võrdsete R1 ja R2 väärtuste puhul pingejagurilt saadav pinge poolitab toitepinget, mida näitab valem (1).

$$V_{R2} = V_{CC} \cdot \frac{R1}{R1 + R2} = 5 \text{ V} \cdot \frac{47 \text{ k}\Omega}{47 \text{ k}\Omega + 47 \text{ k}\Omega} = 2,5 \text{ V} \quad (1)$$

Mõõdetav pinge on aga arvutatust veidi madalam. See on põhjustatud nii takistite tolerantsist, mis on tegeliku takistuse kõrvalekalle nimitakistusest, kui ka operatsioonivõimendi sisendite tarbitavast voolust. Eeldusel, et sisendvool on konstantne, saab valides väiksema takistusega takisteid pingelangu vähendada, kuid see suurendab pingejagurit läbiva voolu.

Eelvõimendi teist astet moodustab operatsioonivõimendi invertteerivas lülituses (tähistus skeemil IC3B). See võimendab kasulikku signaali, andes kasutajale võimalust muuta seadme tundlikkust liiga nõrga või tugeva signaali puhul. Puuduseks on aga see, et madala toitepinge tõttu on oluliselt piiratud maksimaalne võimendustegur, mille puhul signaalis ei teki mittelineaarseid moonutusi. Kuna operatsioonivõimendi invertteerivasse sisendisse antav signaal sisaldab juba alaliskomponenti, mida ei pea võimendama, on mitteinvertteerivale sisendile rakendatud tugipinge, mida saadakse samuti võrdväärsetest takistitest R4 ja R5 koosneva pingejaguri abil. Siiski tuleb arvestada, et esimesest astmest tuleneva signaali alalispinge on tugipingest pisut väiksem, mistõttu kahe alalispinge vahet võimendatakse nagu vahelduvsignaali. Alaliskomponendi pinge väljundis kõrvaldatakse pärast signaali digitaliseerimist. Operatsioonivõimendi

sisendimpedants invertteerivas lülituses on ligikaudselt võrdne signaaliallika ja sisendi vahel asuva takisti väärtusega. Võimendustegur on väljendatav tagasiside ahela takistuse ja sisendtakistuse suhtega (valem (2)) [13].

$$G = \frac{R_f}{R_{in}} = \frac{R6}{R3} = \frac{10 \text{ k}\Omega}{1 \text{ k}\Omega} = 10 \quad (2)$$

Arvestades, et takisti R3 takistus on 1 k Ω ja R6 on potentsiomeeter takistusega 0 kuni 10 k Ω , on valemi (2) järgi võimendustegur muudetav vahemikus 0 kuni 10. Kui võimendustegur jääb alla 1, on väljundpinge amplituud sisendpinge omast väiksem, mis on kasulik, kui sisendsignaali on liiga tugev. Nagu invertteeriva lülituse nimetus ütleb, väljundi ja sisendi polaarsused on omavahel vastupidised, kuid see ei mõjuta signaali sageduskomponente. Takistitest R7 ja R8 koosnev pingejagur vähendab pinget kolmekordselt, jättes alaliskomponendi suuruseks 0,83 V. Eelvõimendi väljund on ühendatud mikrokontrolleri väljaviiguga 16, mis oli valitud analoog-digitaalmuunduri kasutamiseks. Edaspidi toimub signaali digitaliseerimine ja töötlemine, mida vaadeldakse peatükis 5.

4.2 Programmeeritav heligeneraator ja digitaal-analoogmuundur

Programmeeritav heligeneraator (PSG, *programmable sound generator*) on mikroskeem, mis genereerib helilaineid, sünteesides mitut lihtsamat lainekuju ning muutes nende amplituude mähisjoone järgi [14]. Käesolevas töös oli heligeneraatoriks valitud digitaalne mikroskeem Yamaha YM3812 (tuntud samuti kui OPL2 – *FM Operator Type-L II*, „sagedusmodulatsiooni operaator, tüüp L, II versioon“), mis oli enim levinud 1980. – 1990. aastatel tänu kasutamisele personaalarvutite helikaartides *AdLib* ja *Sound Blaster* ning mitmes klahvsüntesaatorite mudelites [15]. Eksisteeris võimalus ka teiste heligeneraatorite kasutamiseks, kuid selle mikroskeemi programmeerimine ja kasutamine on väga hästi dokumenteeritud.



Joonis 5. Programmeeritav heligeneraator YM3812.

Digitaalse heligeneraatori YM3812 nominaalne taktsagedus on 3,58 MHz, kusjuures lubatud vahemik on 2 kuni 4 MHz [16]. Arvestades, et mikrokontroller on võimeline taimeriga järgi teatud väljaviikude loogilist nivood ümber lülitada, kasutatakse taktsignaali allikana väljaviiku 23. Antud juhul heligeneraatori taktsageduseks on valitud 3,75 MHz, mida saadakse mikrokontrolleri siini taktsageduse (60 MHz) täisarvuga jagamise teel. Ülejäänud ühendused on kirjeldatud allpool (Tabel 2).

Tabel 2. Mikroskeemi YM3812 kasutatavate väljaviikude loetelu [16], [17].

YM3812 väljaviik	Kirjeldus	Mikrokontrolleri väljaviik	DAC väljaviik
1 (V _{SS})	Toitepinge	V _{IN}	–
3 (/IC)	Heligeneraatori lähtestamine (<i>reset</i>), aktiivne signaal 0	32	–
4 (A0)	Valib, kas kirjutatakse registri aadressi (0) või registri sisu (1)	31	–
5 (/WR)	Lubab (0) või keelab (1) andmete kirjutamist	30	–
6 (/RD)	Lubab (0) või keelab (1) andmete lugemist	29	–
7 (/CS)	Lubab (0) või keelab (1) kiibile pöördumist	28	–
10 (D0)	Andmesiini bitt 0	2 (pordi D bitt 0)	–
11 (D1)	Andmesiini bitt 1	2 (pordi D bitt 1)	–
12 (GND)	Maandus	GND	–
13 (D2)	Andmesiini bitt 2	2 (pordi D bitt 2)	–
14 (D3)	Andmesiini bitt 3	2 (pordi D bitt 3)	–
15 (D4)	Andmesiini bitt 4	2 (pordi D bitt 4)	–
16 (D5)	Andmesiini bitt 5	2 (pordi D bitt 5)	–
17 (D6)	Andmesiini bitt 6	2 (pordi D bitt 6)	–
18 (D7)	Andmesiini bitt 7	2 (pordi D bitt 7)	–
20 (SH)	Signaal digitaal-analoogmuunduri nihkeregistrile	–	3 (LOAD)
21 (MO)	Heligeneraatori väljund (digitaalne)	–	4 (SD)
23 (øSY)	Taktsignaali väljund digitaal-analoogmuunduri jaoks	–	5 (CLK)

YM3812 väljaviik	Kirjeldus	Mikrokontrolleri väljaviik	DAC väljaviik
24 (øM)	Heligeneraatori taktsignaali sisend	23	–

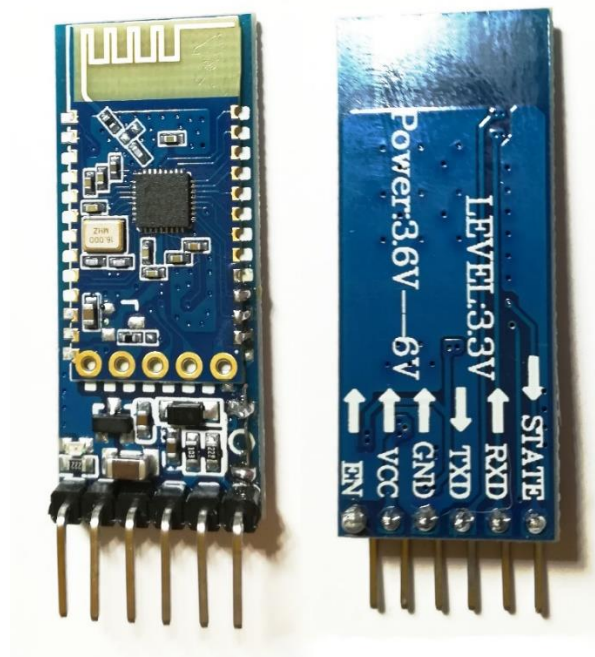
Erinevalt paljudest teistest heligeneraatorite mudelitest, YM3812 ei oma sisseehitatud digitaal-analoogmuundurit (DAC, *digital-to-analog converter*). Sellega kasutamiseks on ettenähtud DAC mikroskeem Yamaha YM3014B (tuntud ka kui Y3014B) [16]. Muunduri sisendiks on digitaalne signaal ujukomaformaadis, kusjuures 10 bitti moodustavad mantissi ja 3 bitti eksponendi osa [17]. Operatsioonivõimendi pingejärguri lülituses (tähistus IC4A) moodustab tugipinge allika digitaal-analoogmuunduri jaoks, mille väljundi keskväärtuseks on $\frac{1}{2}$ toitepingest [17] ehk 2,5 V (toitepinge 5 V puhul). Veel üks pingejärguri lülituses kasutatav operatsioonivõimendi (IC5A) moodustab puhvervõimendi muunduri väljundsignaali jaoks.

Heligeneraatori kasutamine ja eripärad on põhjalikumalt kirjeldatud peatükis 6.

4.3 Bluetooth-moodul

Igal süntesaatoril on üldjuhul olemas elemendid, mille abil seadet on võimalik juhtida. Tavaliselt need on mehaaniliste osadega komponendid, näiteks nupud või potentsiomeetrid. Töö raames oli soovitud skeemi lihtsaks jätta ruumi kokkuhoidmise eesmärgiga. Arvestades Bluetooth toega seadmete laia levikut, oli otsustatud mehaaniliste komponentide asemel kasutada juhtmevaba sidet Bluetooth-liidese kaudu.

Bluetooth-moodul JDY-30, mis on sarnane laiali kasutatavate moodulitega HC-05 ja HC-06, toetab protokollil SPP (*Serial Port Protocol*) ja imiteerib järjestikliidese ühendatud terminali. See ei ole otseselt vastutav sünteesi juhtimise eest, vaid edastab vastuvõetud andmed mikrokontrollerile, mis omakorda töötleb neid. Kohe pärast sisselülitamist moodul töötab vaikimisi käsurrežiimis (*command mode*) ning võtab vastu käsked järjestikliidese kaudu. Sellel etapil on võimalik muuta mooduli nime, paarimise PIN-koodi ning edastuskiirust. Ühendust initsieeriva seadme (*host-seadme*) rolli täidab arvuti, nutitelefon või muu Bluetooth-võimega seade. Kui side on tekkinud, moodul automaatselt lülitub ümber edastusrežiimisse (*transmission mode*). Seejärel võivad seadmed alustada andmevahetust omavahel [18].



Joonis 6. Bluetooth-moodul JDY-30, vaade eest ja tagant.

Moodul omab 6 väljaviiku, millest kaks (*VCC*, *GND*) ühendatakse toiteallikaga. Antud juhul toitepinget saadakse mikrokontrolleri 3,3-voldise stabilisaatori väljundist. Katsete järgi ei mõjuta toitepinge nominaalsest madalam väärtus mooduli töövõimet. Väljaviigud *RX* ja *TX* on ühendatud mikrokontrolleri järjestikliidesega, kusjuures mooduli „vastuvõttev“ kontakt *RX* tuleb ühendada mikrokontrolleri „saatva“ kontaktiga *TX* ja vastupidi. Väljaviik *STATE* võimaldab mooduli seisundi jälgimist: kui see on edastusrežiimis, siis sellelt väljaviigult on loetav loogiline väärtus „1“, millele vastab pinge 3,3 V. Väljaviik *EN* on mõeldud mooduli käsu- ja edastusrežiimi vahel käsitsi ümberlülitamiseks, kuid antud moodulis see funktsionaalsus ei toimi ning ilmselt see väljaviik on jäetud vanemate moodulitega mehaanilise ühilduvuse tagamiseks.

4.4 Väljundsignaali madalpääsfilter ja eelvõimendi

Filtri kasutamise eesmärgiks on signaali võimendamine või nõrgendamine teatud sageduste vahemikus. Antud juhul see on vajalik peamiselt kõrgsageduslikest komponentidest tekkiva müra eemaldamiseks.

Töös kasutatud väljundsignaali filter ja eelvõimendi on realiseeritud väikeste muudatustega arvutite *Commodore 64/128* välisseadme *Commodore SFX Sound Expander* skeemi järgi [19].

Genereeritud signaali filtreerimiseks on rakendatud aktiivset madalpääsfiltrit, mis on koostatud Sallen-Key topoloogia järgi kahest takistist, kahest kondensaatorist ja operatsioonivõimendist. Üks filtri poolus vähendab ülekandetegurit alates lõikesagedusest 20 dB võrra dekaadi kohta. Valitud topoloogia järgi on filtril kaks poolust ehk tegemist on teist järku filtriga, mis tähendab langust 40 dB võrra dekaadi kohta [13]. Filtri moodustavad komponendid on skeemil tähistatud vastavalt R10, R11, C3, C4 ja IC5A. Lõikesagedus, mille puhul signaali pinge väheneb ligikaudu 3 dB võrra ehk kahekordselt, on leitav valemi (3) järgi.

$$f_c = \frac{1}{2\pi \sqrt{R_1 R_2 C_1 C_2}} = \frac{1}{2\pi \sqrt{33 \text{ k}\Omega \cdot 33 \text{ k}\Omega \cdot 510 \text{ pF} \cdot 220 \text{ pF}}} \approx 14\,398 \text{ Hz} \quad (3)$$

Asetades valemisse nimetatud komponentide takistuste ja mahtuvuste suurused, saame lõikesageduseks 14 398 Hz. See katab suuremat osa inimese tajutavate sageduste vahemikust, mis keskmiselt on 20 kuni 20 000 Hz.

Filtreeritud helisignaali võimendatakse mitteinverteerivas lülituses oleva operatsiooni (IC5B) abil, et sobitada signaali lõppvõimendi liinisisendiga. Helisignaali koos alaliskomponendiga saadakse mitteinverteeriva sisendi kaudu ning tugipinge allikas on ühendatud unverteeriva sisendiga jadamisi oleva takisti R12 kaudu. On rakendatud negatiivne tagasiside, kusjuures tagasisideahelat moodustab takisti R13. Võimendustegur on leitav valemi (4) järgi [13].

$$G = \frac{R_f}{R_{in}} + 1 = \frac{R_{13}}{R_{12}} + 1 = \frac{27 \text{ k}\Omega}{15 \text{ k}\Omega} + 1 = 2,8 \quad (4)$$

Tingimusel $R_{12} = 15 \text{ k}\Omega$ ja $R_{13} = 27 \text{ k}\Omega$ saab arvutada, et võimendustegur on 2,8 korda. Lõpuks helisignaali läbib passiivset kõrgpääsfiltrit, mida moodustavad takisti R14 ja kondensaator C5, mille lõikesagedus on leitav valemi (5) järgi.

$$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 4,7 \text{ k}\Omega \cdot 1 \text{ }\mu\text{F}} \approx 33,86 \text{ Hz} \quad (5)$$

Asetades valemisse komponentide väärtused, saame lõikesageduseks 33,86 Hz, mis on sobiv alaliskomponendi ja mittekuuldavate madalate sageduste kõrvaldamiseks.

5 Sisendsignaali töötlemine

Signaali töötlemiseks oli arendatud mikrokontrolleri püsivara. Selle lähtekood oli kirjutatud programmeerimiskeeltes C ja C++ ning on kättesaadav *Git*'i repositooriumist aadressilt <https://gitlab.com/fastman1010/teensyfft/>. Projekti kirjutamise protsessis oli kasutatud integreeritud arenduskeskkond *Eclipse* koos tarkvarapaketi *PlatformIO*, mis lihtsustab tarkvarateekide haldamist, projekti kompileerimist ning püsivara üleslaadimist mikrokontrollerile. Seega projekti käivitamiseks on vajalik nimetatud tarkvara olemasolu ning projekti konfiguratsioonis tuleb määrata teekide asukohta.

Kuna signaali töötlemine eeldab sisendsignaali spektri analüüsi selles sisalduvate sageduste leidmiseks, kasutatakse kiiret Fourier' teisendust (FFT). Kiiretoimelisuse saavutamiseks kasutatakse koodis ulatuslikult CMSIS-DSP (*Cortex Microcontroller Software Interface Standard – Digital Signal Processing*) teegi funktsioone, mis on optimeeritud ujukomaploki kasutamiseks. Teegis on maksimaalseks lubatud FFT akna pikkuseks 4096 punkti [20], mis ongi püsivaras kasutatud (tähistatud koodis kui *FFT_LEN*).

Sisendsignaali pärineb elektrikitarrist või muust helisignaali allikast ning on pidevsignaali. Et oleks võimalik pidevsignaali töödelda digitaalselt, on vaja kõigepealt signaali digitaliseerida. Teisiti öeldes, on vaja alles jätta ainult osa signaalist, mille suhtes teatud algoritmi järgi digitaalset töötlust sooritatakse. Digitaalseks töötlemiseks kõlbliku signaali saamiseks tuleb pidevsignaali diskreetida ja kvanteerida [21]. Selleks on vaja kasutada analoog-digitaalmuundurit.

5.1 Signaali digitaliseerimine

Diskreetimisel on kasutatud otsemällupöödust (DMA, *Direct Memory Access*), mis võimaldab andmete edastamist otse sisend-väljundseadme ja mälu vahel ilma protsessori osaluseta. Süntesaatori püsivara kasutab DMA-d, et võimaldada signaali uue osa puhverdamist ning vanemate osade analüüsimist samaaegselt. Vastupidisel juhul signaali digitaliseerimine katkeb analüüsi ajaks ning osa signaalist kaob. DMA

initsialiseerimist teostav kood on võetud muudetud kujul teegi *Teensy Audio Library* failist *input_adc.cpp* [22].

Diskreetimissageduse määramiseks on kasutatud programmeeritavat viiteplokki (*programmable delay block*, PDB). Antud juhul PDB loenduri väärtus suureneb 1 võrra iga mikrokontrolleri siini taktiga. Analoo-digitaalmuundurile pöördumiste sagedust on võimalik määrata, seadistades loenduri mooduli väärtust. Mooduli väärtuse ületamisel tekib katkestus (*interrupt*), millega kaasneb analoo-digitaalmuunduri väljundi lugemine ja salvestamine mällu, ning loendamine algab uuesti nullist [23].

Maksimaalne võimalik analoo-digitaalmuunduri resolutsioon on 16 bitti [9], mis vastab dünaamilisele diapsoonile 96,32 dB [21]. Püsivaras on määratud diskreetimissagedus 32 000 Hz (tähistus *SAMP_RATE*). Arvestades, et Nyquisti teoreemi järgi diskreetimissagedus peab vähemalt kahekordselt ületama kõrgeimat signaalis sisalduvat sagedust [24], saame sisendsignaali ribalaiuseks 16 000 Hz, mis ületab paljude muusikariistade diapsooni ülempiiri [25].

Sisendsignaali salvestamiseks kasutatakse 16 osaks (tähistus *FFT_SHIFTS*) jagatud ringpuhvri *sampBuffer[FFT_SHIFTS][FFT_SHIFT_LEN]*, kusjuures ühe osa pikkus (tähistatud kui *FFT_SHIFT_LEN* ning on võrdne 256-ga) määrab FFT akende mittekattuvat. Ringpuhvri kõigi osade pikkus on võrdne FFT akna pikkusega. Arvestades eelmainitud, saab leida ühe akna osale vastavat ajaperioodi. See periood määrab maksimaalset võimalikku FFT akna töötlemise aega ning maksimaalset puhverdamisest tingitud viidet.

$$t = \frac{FFT_LEN}{FFT_SHIFTS} \cdot \frac{1}{SAMP_RATE} = \frac{4096}{16} \cdot \frac{1}{32000 \text{ Hz}} = \frac{256}{32000 \text{ Hz}} = 0,008 \text{ s} = 8 \text{ ms}$$

See tähendab, et signaali töötlemise järel tekkiv viide ei pea osutama oluliseks piiranguks muusikariista mängimisel (vt. peatükk 3). Kui puhvri osa on täitunud, tekitatakse tarkvaralist katkestust ning katkestusetöötaja (*Interrupt Service Routine*, ISR) kutsub välja töötlemist teostavat funktsiooni *processADCInput*, mis on kirjeldatud järgmises alamtükkis ja mille koodi näitab Joonis 7.

5.2 Spektri analüüs

Puhvri osa täitumisel tuleb teisendada salvestatud täisarvulised väärtused ujukomaarvulisteks. Ujukomaarvuline 32-bitine formaat (*float32_t*) sai valitud selle põhjusel, et CMSIS-DSP funktsioonid võimaldavad tööd ainult arvudega ujukomaformaadis või fikseeritud komaga formaadis (tuntud kui *Q format*) [20]. Teisendatavad väärtused võetakse asukohast mälus, millele viitab muutuja *adcBuffer* ning salvestatakse massiivi *sampBuffer[bufCurrent]*, kusjuures muutuja *bufCurrent* tähistab ringpuhvri osa, mis kirjutatakse üle. Seda muutujat võib pidada loenduriks mooduliga *FFT_SHIFTS*.

Seejärel funktsiooni *bufferCopyPart* abil moodustatakse mälus olevates signaali osadest FFT akna suurusega puhver, mis sisaldab viimati digitaliseeritud signaalilõike. Järgnevast võrrandist järeldub, et aknas sisalduvad väljavõtted vastavad viimase 128 millisekundi jooksul digitaliseeritud signaali osale.

$$t = \frac{FFT_LEN}{SAMP_RATE} = \frac{4096}{3200} = 0,128 \text{ s} = 128 \text{ ms}$$

Analüüsitava signaali osa lõpliku kestvuse ja diskreetsete omaduste tõttu tekib niinimetatud lekkimisefekt, mis avaldub spektrijoone jagunemises mitme sageduse vahel. Selle parandamiseks võib kasutada aknafunktsioone [26]. Valitud on Blackman-Harris aknafunktsioon, mida defineeritakse valemiga (6), kus n on väljavõtte number ning N on FFT akna pikkus [26].

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N}\right) + a_2 \cos\left(\frac{4\pi n}{N}\right) - a_3 \cos\left(\frac{6\pi n}{N}\right) \quad (6)$$

Koefitsiendid on valitud järgmiselt [27]:

$$a_0 = 0,35875; a_1 = 0,48829; a_2 = 0,14128; a_3 = 0,01168$$

Funktsioon *arm_mult_f32* korrutab omavahel kahe massiivi elemente ja salvestab tulemust mõnda massiivi [28], kusjuures antud juhul korrutatakse puhvris sisalduvaid väljavõtteid (*sampReal*) aknafunktsiooni väärtustega (*window*). Tulemust salvestatakse *sampReal* kohale.

Kuna kiiret Fourier' teisendust teostava funktsiooni *arm_cfft_radix4_f32* sisend peab olema kompleksarvuline [20], kuid digitaliseeritud signaal on reaalne, võib muuta

signaali kompleksarvuliseks, võttes imaginaarosa võrdseks nulliga [21]. Selleks kutsutakse välja funktsiooni *padRealToCompl*, mis kopeerib väärtused massiivist *sampReal* massiivi *specDoubleCompl*, asetades iga väärtuse taga nulli. Funktsioon *arm_cfft_radix4_f32* salvestab tulemust, mis on kahepoolne spekter komplekskujul, samasse massiivi *specDoubleCompl*.

Funktsioon *arm_cmplx_mag_f32* arvutab massiivis *specDoubleCompl* sisalduvate kompleksarvude moodulid ja salvestab massiivi *specSingleAbs*. Kuna maksimumide leidmiseks piisab ühepoolsest spektrist, tuleb leida vaid $FFT_LEN/2$ arvu moodulid. Kui moodulid on leitud, teostatakse normeerimist. Arvestades, et 16-bitise resolutsiooniga väljavõtte maksimaalne võimalik väärtus on $2^{16} - 1 = 65\,535$ ning FFT tulemus on skaleeritud FFT_LEN kordselt [20], jagatakse massiivis *specSingleAbs* sisalduvad väärtused nende arvude korrutisega ($FFT_LEN * 65535$).

```

void bufferCopyPart(float32_t *data, float32_t (*buf)[FFT_SHIFT_LEN],
uint8_t bufCurrent)
{
    uint8_t i;
    for (i = 0; i < FFT_SHIFTS; i++) {
        arm_copy_f32(buf[(i + bufCurrent) % FFT_SHIFTS], &data[i *
            FFT_SHIFT_LEN], FFT_SHIFT_LEN);
    }
}

void processADCInput(uint16_t *adcBuffer) {
    uint16_t i;
    uint8_t freqsUsed;
    float32_t sampReal[FFT_LEN];
    float32_t specDoubleCompl[FFT_LEN * 2];
    float32_t specSingleAbs[FFT_LEN / 2];
    static voiceState synthStates[VOICES];
    static voiceState freqStates[VOICES];

    for (i = 0; i < FFT_SHIFT_LEN; i++) {
        sampBuffer[bufCurrent][i] = (float32_t) adcBuffer[i];
    }
    bufCurrent = (bufCurrent + 1) % FFT_SHIFTS;

    bufferCopyPart(sampReal, sampBuffer, bufCurrent);

    arm_mult_f32(sampReal, window, sampReal, FFT_LEN);
    padRealToCompl(sampReal, specDoubleCompl, FFT_LEN);
    arm_cfft_radix4_f32(&fftInst, specDoubleCompl);
    arm_cmplx_mag_f32(specDoubleCompl, specSingleAbs, FFT_LEN / 2);
    arm_scale_f32(specSingleAbs, (1.0 / (FFT_LEN * 65535)), specSingleAbs,
FFT_LEN / 2);

    freqsUsed = findMaxValues(specSingleAbs, specDoubleCompl,
ym.frequencyCount, freqStates);
    playFrequencies(&ym, freqsUsed, freqStates, synthStates);
}

```

Joonis 7. Puhvri osa töötlev funktsioon *processADCInput* ning abifunktsioon *bufferCopyPart*.

Saadud spekter on nüüd analüüsiks ette valmistatud ning edaspidine töötlus toimub funktsioonis *findMaxValues* (kood on toodud Lisas 1), mis võtab vastu nii ühepoolset amplituudspektri kui ka kahepoolset kompleks-spektri. Arvestades, et digitaliseeritud signaal sisaldab alaliskomponenti, millele spektris vastab positsioon 0 (ehk sagedus 0 Hz), ühepoolsest spektrist lõigatakse välja kõik komponendid, mis vastavad sagedustele alla 30 Hz. Sellega välditakse mittekuuldavate sageduste genereerimist heligeneraatori poolt.

Seejärel tsüklis otsitakse ühepoolsest spektrist maksimumid, mille arv on määratav käsuga *poly*, mis on kirjeldatud peatükis 7. Maksimumide leidmiseks kasutatakse CMSIS-DSP teegi funktsiooni *arm_max_f32*, mis tagastab maksimumi väärtust ja asukohta massiivis. Kui maksimum on leitud, võrreldakse seda minimaalse lubatud amplituudi väärtusega *amplMin*. Juhul, kui leitud väärtus jääb alla *amplMin*, tsükkel lõpetatakse ning toimub funktsioonist väljumine. Vastasel juhul toimub interpoleerimine Quinni teise hinnangfunktsiooni (*Quinn's Second Estimator*) abil. Seda kirjeldavad järgmised avaldised (valem (7)), kus k on maksimumi asukoht spektris, X_i on amplituudi väärtus asukohal i ning k' on interpoleeritud maksimumi asukoht [29].

$$\left\{ \begin{array}{l} \tau(x) = \frac{1}{4} \log(3x^2 + 6x + 1) - \frac{\sqrt{6}}{24} \log \left(\frac{x + 1 - \sqrt{\frac{2}{3}}}{x + 1 + \sqrt{\frac{2}{3}}} \right) \\ ap = \frac{Re(X_{k+1}) \cdot Re(X_k) + Im(X_{k+1}) \cdot Im(X_k)}{Re^2(X_k) + Im^2(X_k)} \\ dp = -\frac{ap}{1 - ap} \\ am = \frac{Re(X_{k-1}) \cdot Re(X_k) + Im(X_{k-1}) \cdot Im(X_k)}{Re^2(X_k) + Im^2(X_k)} \\ dm = \frac{am}{1 - am} \\ d = \frac{dp + dm}{2 + \tau(dm^2) - \tau(dm^2)} \\ k' = k + d \end{array} \right. \quad (7)$$

Koodis interpoleeritud maksimumi asukohta salvestatakse muutujasse *kAppr*. selle korrutamisel murdega (*SAMP_RATE / FFT_LEN*) saame sagedust. Seejärel pannakse see leitud sageduste nimekirja. Juhul, kui on mitmekordselt leitud sama sagedus (lubatud kõrvalekalde on $\pm 1,5\%$), mis on võimalik lähedal asuvate maksimumide korral, kordusi ignoreeritakse.

Kui tegemist on esmakordselt esinenud sagedusega, tuleb teostada amplituudi interpoleerimist. Sellel juhul kasutatakse lineaarinterpoleerimist, mida kirjeldab valem (8) [30]. Antud juhul y on interpoleeritav amplituudi väärtus, x_0 ja x_1 on interpoleeritud maksimumi asukohale lähimad täisarvulised positsioonid ning y_0 ja y_1 on nende amplituudide suurused.

$$y = y_0 + (x - x_0) \left(\frac{y_1 - y_0}{x_1 - x_0} \right) \quad (8)$$

Et vältida sageduse korduvat leidmist ning lähestikku asuvate sageduste genereerimist, pärast amplituudi arvutamist positsiooni X_k väärtust võetakse võrdseks nulliga ning seda ümbritsevate positsioonide X_{k-1} ja X_{k+1} korrutatakse 0,5-ga. Kui vajalikud sagedused ja nendele vastavad amplituudid on spektrist leitud, tagastab funktsioon leitud sageduste arvu. Seejärel neid tuleb edastada heligeneraatorile, millega tegeleb funktsioon *playFrequencies*.

5.3 Sageduste edastamine heligeneraatorile

Heligeneraator YM3812 kasutab kanali sageduse määramiseks niinimetatud F -arvu (*F-Number*) mis on leitav valemiga (9), kus f on sagedus, mille jaoks vastava F -arvu otsitakse, *block* on sageduste ploki number, mis on laiemas mõttes sarnane oktaavi numbriga [31], ning f_s on diskreetimissagedus, mis on leitav kiibi taktsageduse 72-ga jagamise teel; nominaaltaktsageduse puhul see on 49 716 Hz [32].

$$F = \frac{f \cdot 2^{20-block}}{f_s} \quad (9)$$

F -arvu ja ploki numbri arvutamiseks on loodud funktsioon *YM3812::freqToFNum*, mille sisu näitab Joonis 8.

```
uint16_t YM3812::freqToFNum(float32_t freq) {
    uint32_t fnum;
    uint16_t result;
    uint8_t block = 0;

    fnum = freq * 1048576 / chipSampleRate; // 1048576 = 2^20
    while (fnum >= 1024 && block < 8) {
        fnum = fnum >> 1;
        block++;
    }
    result = fnum | (block << 10);
    return result;
}
```

Joonis 8. Sageduse F -arvuks teisendamist teostav funktsioon.

10 bitist koosnev F -arv säilitatakse heligeneraatori kahes registris (0xA0 – 0xA8 ning 0xB0 – 0xB8, kusjuures kanali number vastab positsioonile nendes vahemikes). Kaks vanemat bitti säilitatakse noorematest eraldi ning vanemaid bitte sisaldav register säilitab ka ploki numbri [16], mistõttu funktsioon tagastab F -arvu ja ploki numbri koos

16-bitise arvuna. Sageduse edastamist heligeneraatorile ja klahvi sisselülitamist teostab funktsioon *YM3812::voiceSetFreqAndKeyOn*.

```
void YM3812::voiceSetFreqAndKeyOn(float32_t freq, uint8_t voice) {
    uint16_t f = freqToFNum(freq);
    write(0xA0 + voice, f & 0xFF);
    write(0xB0 + voice, (state[0xB0 + voice] & 0xC0) | (f >> 8) | (1 << 5));
}
```

Joonis 9. Heligeneraatori YM3812 kanali sagedust määrav funktsioon.

Tasub arvestada, et heligeneraatori väljundi tugevus on määratav logaritmilise väärtusega [33], mistõttu spektri analüüsi osas arvutatud amplituudi tuleb teisendada, arvestades, et tegelikult tuleb seadistada signaali summutamise määra. Sellest on mainitud põhjalikumalt peatükis 6. Teisendamiseks on loodud funktsioon *YM3812::channelSetAmpl*.

```
void YM3812::channelSetAmpl(float32_t ampl, uint8_t chan) {
    chan = getChannelShift(chan);
    uint8_t attn, op1level, op2level;
    if (ampl >= amplMax) {
        attn = 0;
    } else {
        ampl = 20 * log10f(ampl / amplMax);
        if (isnan(ampl)) {
            attn = 63;
        } else {
            attn = (uint8_t) (ampl / (-0.75));
        }
    }
    op1level = cfg.op1tl + attn;
    if (op1level > 63) op1level = 63;
    op2level = cfg.op2tl + attn;
    if (op2level > 63) op2level = 63;

    if (cfg.conn == true) { // additive
        write(0x40 + chan, (state[0x40 + chan] & 0b11000000) | op1level);
        write(0x43 + chan, (state[0x43 + chan] & 0b11000000) | op2level);
    } else { // FM
        write(0x43 + chan, (state[0x43 + chan] & 0b11000000) | op2level);
    }
}
```

Joonis 10. Funktsiooni *YM3812::channelSetAmpl* lähtekood.

Järgmises peatükis vaadeldakse põhjalikumalt heligeneraatori YM3812 parameetreid ning nende omapärasid.

6 Programmeeritava heligeneraatori kasutamine

Mikroskeem Yamaha YM3812 sisaldab digitaalset 9-kanalilist programmeeritavat heligeneraatori. Selle püsivus on salvestatud veerand siinusfunktsiooni perioodist logaritmilisel kujul. Tänu sellele signaali genereeritakse, kasutades peamiselt sümmeetriat ja liitmistehet [34].

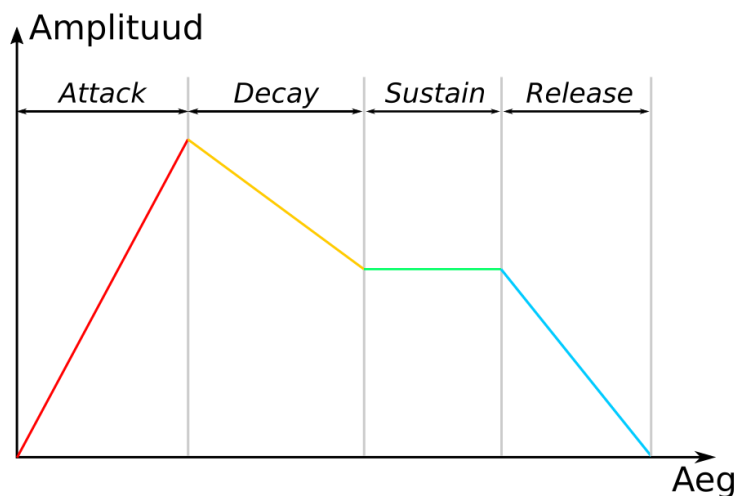
Iga kanali eest vastutavad kaks digitaalset lainegeneraatorit, mida nimetatakse operaatoriteks; seda terminit kasutatakse töös ka edaspidi. Kanali operaatorite paari on võimalik ühendada kahel viisil [16]. Esimene neist on operaatorite väljundite kokku liitmine, mille puhul on tegemist liitmis- ehk aditiivse sünteesiga (*additive synthesis*), mis on kirjeldatav valemiga (10), kus F on väljundi väärtus (indeksiks on ühendamise viis), I on amplituud ning w on harmooniline sagedus (indeksid vastavad operaatorite numbritele).

$$F_1 = I_1 \sin w_1 t + I_2 \sin w_2 t \quad (10)$$

Teiseks võimaluseks on sagedusmodulatsioon, kusjuures esimene operaator genereerib moduleerivat signaali ning teine operaator tekitab kande signaali [16]. See võimaldab keerukamate lainekujude genereerimist. Kanali väljund on sellel juhul kirjeldatav valemiga (11).

$$F_2 = I_2 \sin(w_2 t + I_1 \sin w_1 t) \quad (11)$$

Igal operaatoril (nii moduleerival kui ka kande signaali tekitaval) saab määrata *ADSR*-mähisjoont (*ADSR envelope*), mis sai oma nimetuse seda moodustavate faaside järgi. Heli puhkemise faas (*attack*) määrab maksimaalse signaali amplituudi saavutamise kiirust alates kanali sisselülitamisest ja laine sünteesi algusest. Sellele järgneb vaibumise faas (*decay*), mille jooksul amplituud väheneb kuni püsivuse helitasemeni (*sustain level*). Selle saavutamise järel on amplituud püsiv kuni kanali väljalülitamiseni. Viimasena tuleb kustumise faas (*release*), mis kirjeldab, kui kiiresti amplituud langeb nullini pärast kanali väljalülitamist [35]. Faaside järjestust illustreerib Joonis 11.



Joonis 11. ADSR-mähisjoone faasid [16], [35].

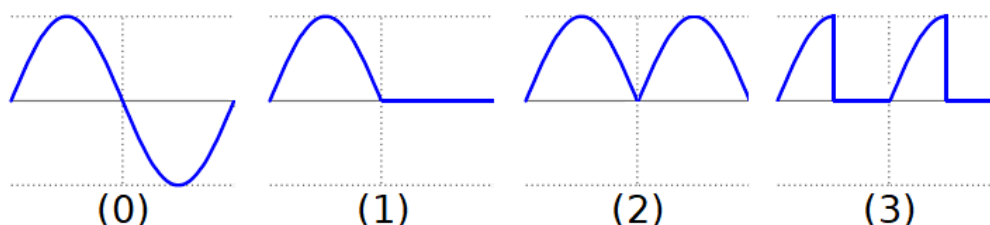
Operaatoritel on võimalik määrata ka muud parameetrid, mis on kirjeldatud allpool (Tabel 3). Tabelis on toodud ka lubatud täisarvuliste väärtuste vahemikud. Registre ja operaatorite seoste kohta võib leida infot allikast [33].

Tabel 3. Heligeneraatori YM3812 operaatorite parameetrid [16].

Parameeter	Kirjeldus
Amplituudmodulatsioon (<i>Amp Mod</i>), 1 bitt	Kui biti väärtus on 1, toimub amplituudmodulatsioon operaatori väljund signaali üle.
Vibraato, 1 bitt	Kui biti väärtus on 1, operaatori väljundsignaali sagedus muutub perioodiliselt väikses ulatuses, tekitades vibraato efekti.
Mähisjoone generaatori tüüp (<i>EG Type</i>), 1 bitt	Kui biti väärtus on 1, arvestatakse mähisjoone püsimise faasiga; vastasel juhul see jäetakse vahele ning vaibumisele koheselt järgneb kustumine.
Klahvi skaleerimine (<i>Key Scaling</i>), 1 bitt	Kui biti väärtus on 1, lubatakse klahvi skaleerimist, mis tähendab, et mida kõrgemat sagedust genereeritakse, seda rohkem väljundit summutatakse.
Sageduse kordaja (<i>Multiple</i>), 4 bitti	Määrab operaatori sageduse kordajat (vahemik 0–15).
Klahvi skaleerimise tegur (<i>Key Scaling Level</i>), 2 bitti	Määrab operaatori väljundi summutamist sõltuvalt sagedusest (vahemik 0–3).
Operaatori väljundi tugevus (<i>Total Level</i>), 6 bitti	Määrab operaatori väljundi summutamist (vahemik 0–63, samm 0,75 dB) Väärtus 0 vastab maksimaalsele helitugevusele ning 63 minimaalsele).

Parameeter	Kirjeldus
Puhkemise kiirus (<i>Attack Rate</i>), 4 bitti	Määrab operaatori maksimaalse amplituudi saavutamise kiirust kanali sisselülitamise hetkest (vahemik 0–15). Kui väärtus on 0, väljund puudub.
Vaibumise kiirus (<i>Decay Rate</i>), 4 bitti	Määrab operaatori vaibumise kiirust (vahemik 0–15).
Püsimise tase (<i>Sustain Level</i>), 4 bitti	Määrab operaatori maksimaalse amplituudi ja püsimise faasi amplituudi vahet (vahemik 0–15).
Kustumise kiirus (<i>Release Rate</i>), 4 bitti	Määrab operaatori signaali kustumist nullini kanali väljalülitamise hetkest (vahemik 0–15). Kui väärtus on 0, signaal ei kustu isegi pärast kanali väljalülitamist ning enne, kui kanal lülitatakse uuesti sisse ja algab uue heli süntees.
Lainekuju valik (<i>Waveform Select</i>), 2 bitti	Määrab operaatori väljundi lainekuju (vahemik 0–3).

Operaatori lainekuju on võimalik valida 4 variandist, mis on tuletatud siinusfunktsioonist. Need on näidatud ja nummerdatud allpool (Joonis 12). Tuletatud lainekujude kasutamiseks tuleb heligeneraatori registri 0x01 biti 5 väärtuseks seadistada 1, et kaotada ühilduvust varasema mikroskeemiga YM3526 [16].



Joonis 12. Heligeneraatori YM3812 operaatorite väljundi võimalikud lainekujud [16].

Lisaks tasub parameetrite seadmisel arvestada, et sageduse kordaja väärtus registris ei määra kordajat üksüheselt [31]. Vastavust kirjeldab Tabel 4.

Tabel 4. Heligeneraatori YM3812 operaatorite sageduste kordajate vastavused [31].

<i>MUL</i> väärtus	Kordaja
0	0,5
1	1
2	2
3	3
4	4
5	5

<i>MUL</i> väärtus	Kordaja
6	6
7	7
8	8
9	9
10	10
11	10
12	12
13	12
14	15
15	15

Ühtlase kõlavärvingu saamiseks seadistatakse selle töö raames kõik kanalid ja nende operaatorite paarid samasuguselt. Eranditeks on kanali sagedus ja operaatori väljundi tugevus, mida muudetakse sõltuvalt sisendsignaali vastava sageduskomponendi amplituudist antud ajahetkel. Juhul, kui on valitud operaatorite väljundite liitmine, tuleb muuta korraga kanali mõlema operaatori väljundite tugevust; sagedusmodulatsiooni puhul piisab teise operaatori väljundi tugevuse muutmisest [31].

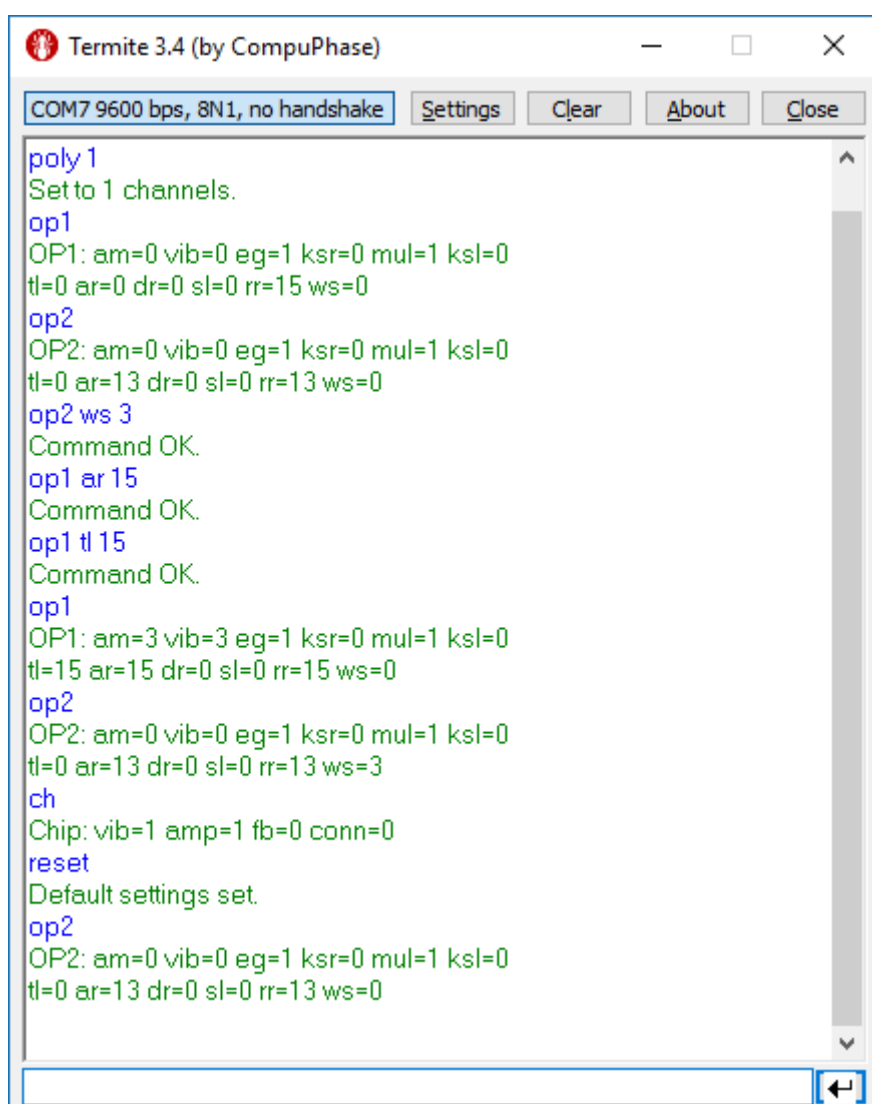
Tabel 5. Heligeneraatori YM3812 kanali parameetrid [16].

Parameeter	Kirjeldus
<i>F</i> -arv (<i>F-Number</i>), 8+2 bitti	Määrab genereeritavat fundamentaalsagedust. Seos sagedusega on kirjeldatud valemis (6).
Plokk (<i>Block</i>), 3 bitti	Määrab genereeritavat fundamentaalsagedust. Seos sagedusega on kirjeldatud valemis (6).
Klahv sees (<i>Key On</i>), 1 bitt	Lülitab kanali sisse/välja.
Tagasiside (<i>Feedback</i>), 3 bitti	Määrab kanali operaatori 1 tagasiside suurust.
Ühendus (<i>Connection</i>), 1 bitt	Määrab kanali operaatorite ühendamise viisi. Kui biti väärtus on 1, kasutatakse väljundite liitmist, vastasel juhul sagedusmodulatsiooni.

Neid parameetreid on võimalik seadistada, et muuta genereeritava heli kõlavärvingut. Seda vaadeldakse põhjalikumalt peatükis 7.

7 Seadme juhtimine

Nagu oli varem mainitud, töös on ettenähtud seadme juhtimine Bluetooth-ühenduse kaudu, mille jaoks oli loodud käskude süsteem. *Host*-seade võib käske saata alles pärast ühenduse loomist seadmega. Käske edastatakse inimloetavas formaadis, mis võimaldab nende sisestamist tekstiterminalis, nagu näitab Joonis 13.



```
Termitte 3.4 (by CompuPhase)
COM7 9600 bps, 8N1, no handshake Settings Clear About Close
poly 1
Set to 1 channels.
op1
OP1: am=0 vib=0 eg=1 ksr=0 mul=1 ksl=0
tl=0 ar=0 dr=0 sl=0 rr=15 ws=0
op2
OP2: am=0 vib=0 eg=1 ksr=0 mul=1 ksl=0
tl=0 ar=13 dr=0 sl=0 rr=13 ws=0
op2 ws 3
Command OK.
op1 ar 15
Command OK.
op1 tl 15
Command OK.
op1
OP1: am=3 vib=3 eg=1 ksr=0 mul=1 ksl=0
tl=15 ar=15 dr=0 sl=0 rr=15 ws=0
op2
OP2: am=0 vib=0 eg=1 ksr=0 mul=1 ksl=0
tl=0 ar=13 dr=0 sl=0 rr=13 ws=3
ch
Chip: vib=1 amp=1 fb=0 conn=0
reset
Default settings set.
op2
OP2: am=0 vib=0 eg=1 ksr=0 mul=1 ksl=0
tl=0 ar=13 dr=0 sl=0 rr=13 ws=0
```

Joonis 13. Näide andmevahetusest Bluetooth-liidese kaudu.

Tabel 6 kirjeldab püsivaras realiseeritud käske, mida võib kasutada seadmete parameetrite juhtimiseks, kusjuures nurksulgudes on antud võimalikud parameetrite väärtuste vahemikud. Meeldetuletuseks tuleb mainida, et kõik kanalid on käesoleva töö

raames seadistatud ühesuguselt, seega sõnauhendi „operaator 1“ all mõistetakse selle töö raames iga kanali moduleerivat operaatorit.

Tabel 6. Seadme juhtimise käskude loetelu.

Käsk	Kirjeldus	Vastus
<i>reset</i>	Heligeneraatori lähtestamine	<i>Default settings set.</i>
<i>min</i>	Tagastab minimaalse amplituudi väärtuse, mis on spektris sisalduva sageduse äratundmise piiriks	<i>amplMin=...</i>
<i>min [0.0...1.0]</i>	Seadistab minimaalse amplituudi väärtuse	Õnnestumise korral: <i>Set amplMin to [0.0...1.0].</i> Ebaõnnestumise korral: <i>Argument 2 must be a float.</i>
<i>max</i>	Tagastab maksimaalse amplituudi väärtuse, mille ületamisel heligeneraatori helitugevuse ei muutu	<i>amplMax=...</i>
<i>max [0.0...1.0]</i>	Seadistab maksimaalse amplituudi väärtuse	Õnnestumise korral: <i>Set amplMax to [0.0...1.0].</i> Ebaõnnestumise korral: <i>Argument 2 must be a float.</i>
<i>poly [0...9]</i>	Seadistab samaaegselt sünteesitavate sageduste maksimaalse arvu	Õnnestumise korral: <i>Set to [0...9] voices.</i> Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>spec</i>	Saadab spektri amplituude ning leitud sagedusi ja amplituude USB-ga seotud järjestikliidese kaudu	<i>Preparing spectrum values.</i>
<i>ch</i>	Tagastab heligeneraatori jooksvad üldised parameetrid	<i>Chip: vib=[0/1], amp=[0/1], fb=[0...7], conn=[0/1]</i>
<i>ch vib [0/1]</i>	Seadistab vibraato efekti sügavust: 0 – 7 tsendi, 1 – 14 tsendi	Õnnestumise korral: <i>Command OK.</i> Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>ch amp [0/1]</i>	Seadistab amplituudmodulatsiooni efekti sügavust: 0 – 1 dB, 1 – 4,8 dB.	Õnnestumise korral: <i>Command OK.</i> Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>ch fb [0...7]</i>	Seadistab operaatori 1 tagasiside sügavust.	Õnnestumise korral: <i>Command OK.</i> Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>ch conn [0/1]</i>	Seadistab operaatorite	Õnnestumise korral: <i>Command</i>

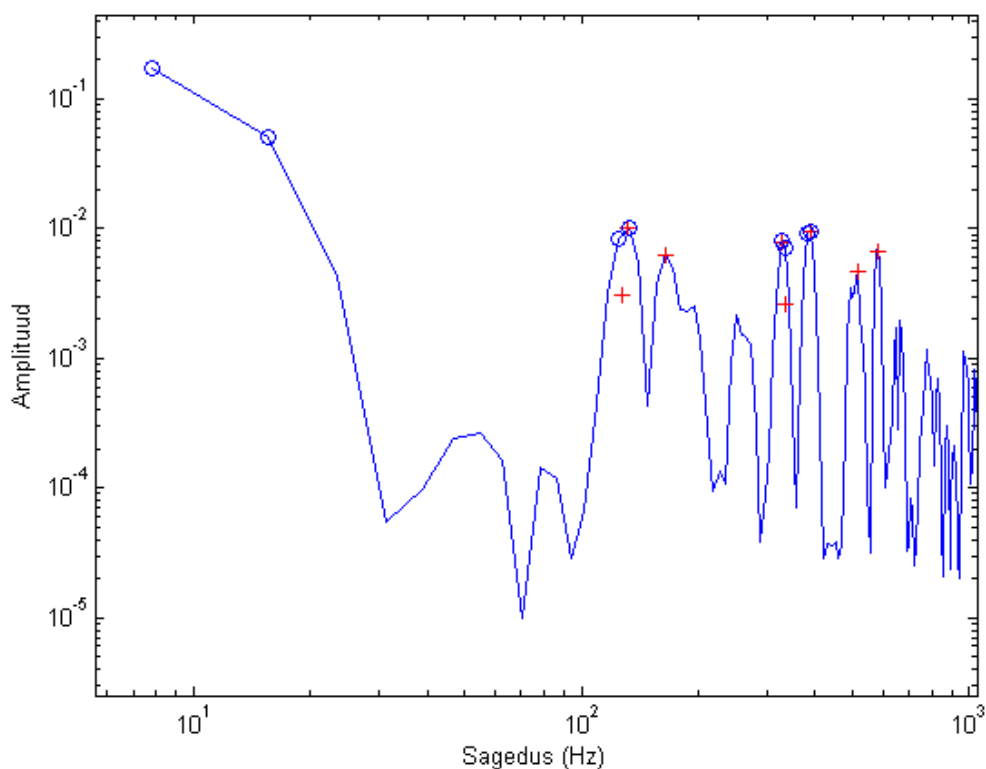
Käsk	Kirjeldus	Vastus
	ühendamise viisi: 0 – sagedusmodulatsioon, 1 – liitmine	OK. Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>op1</i>	Tagastab operaatori 1 jooksvad parameetrid	<i>OPI: am=[0/1], vib=[0/1], eg=[0/1], ksr=[0/1], mul=[0...15], ksl=[0...3], tl=[0...63], ar=[0...15], dr=[0...15], sl=[0...15], rr=[0...15], ws=[0...3]</i>
<i>op1 am [0/1]</i>	Lülitab välja/sisse amplituudmodulatsiooni operaatoril 1	Õnnestumise korral: <i>Command</i> OK. Ebaõnnestumise korral: <i>Argument 2 must be an integer.</i>
<i>op1 vib [0/1]</i>	Lülitab välja/sisse vibraato efekti operaatoril 1	
<i>op1 eg [0/1]</i>	Lülitab välja/sisse mähisjoone püsivuse faasi operaatoril 1.	
<i>op1 ksr [0/1]</i>	Lülitab välja/sisse klahvi skaleerimist operaatoril 1.	
<i>op1 mul [0...15]</i>	Määrab sageduse kordajat operaatoril 1.	
<i>op1 ksl [0...3]</i>	Määrab klahvi skaleerimise tegurit operaatoril 1.	
<i>op1 tl [0...63]</i>	Määrab operaatori 1 väljundi tugevust.	
<i>op1 ar [0...15]</i>	Määrab operaatori 1 väljundi puhkemise kiirust.	
<i>op1 dr [0...15]</i>	Määrab operaatori 1 väljundi vaibumise kiirust.	
<i>op1 sl [0...15]</i>	Määrab operaatori 1 väljundi puhkemise kiirust.	
<i>op1 rr [0...15]</i>	Määrab operaatori 1 väljundi maksimaalse amplituudi ja püsivuse faasi amplituudi vahet.	
<i>op1 ws [0...3]</i>	Määrab operaatori 1 väljundi lainekuju.	
Muu käsk (3-osaline)		
Muu käsk (1- või 2-osaline)		<i>Command not recognized.</i>

Käsk	Kirjeldus	Vastus
Muu käsk		<i>Invalid command size.</i>

Operaatori 2 seadistamise käsud on sarnased operaatori 1 omadega, sel juhul tuleb *op1* asemel sisestada *op2*.

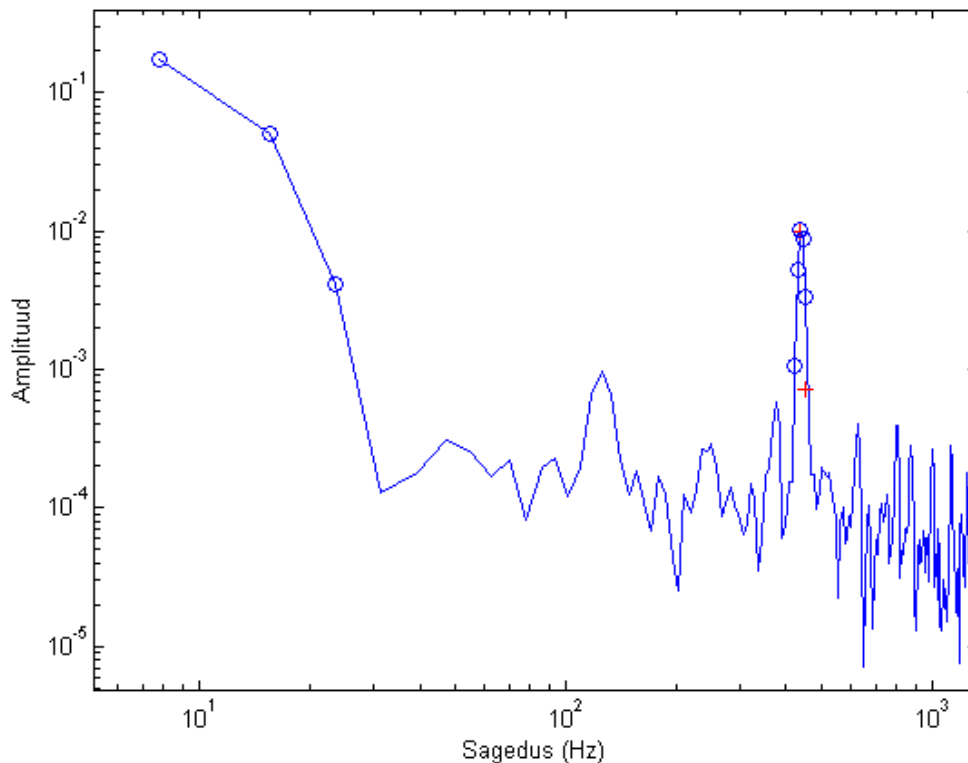
8 Tulemuste analüüs

Spektri analüüsi algoritmi testimiseks olid saadud spektraalkomponentide amplituudid ning joonestatud graafik MATLAB tarkvarapaketi. Võrdluseks olid leitud 9 absoluutset maksimumi muutmata spektris. Edaspidi graafikule olid lisatud püsivaras realiseeritud algoritmi järgi leitud ja interpoleeritud väärtused.



Joonis 14. Spektri maksimumide graafik (sisendiks on kitarril esitatud akord). Sinised ringid tähistavad muutmata spektris esinevaid maksimume ning punaste ristidega on tähistatud algoritmi töö tulemusena leitud maksimumid.

Muutmata kujul spektrist leitud 9 maksimumi on tähistatud siniste ringidega. Interpolatsiooni tulemused on tähistatud punaste ristidega; neid on antud graafikul alles 8, kuna ülejäänud amplituudid ei ületa tundlikkuse piiri. Kõige märgatavam on madalate sageduste ja aliskomponendi puudumine töötlemise järel resulteeruvas spektris. Samuti kui mitu maksimumi asuvad lähestikku, keskmine jääb kõige tugevamaks.



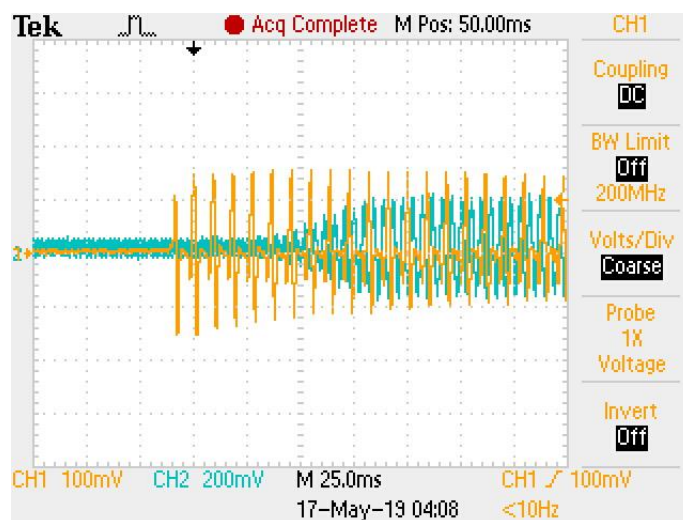
Joonis 15. Spektri maksimumide graafik (sisendiks on siinussignaali sagedusega 440 Hz). Sinised ringid tähistavad muutmata spektris esinevaid maksimume ning punaste ristidega on tähistatud algoritmi töö tulemusena leitud maksimumid.

Juhul, kui sisendile on antud siinussignaali, algoritm näitab ennast paremini. Lihtne maksimumide otsimine annab tulemuseks mitte ainult sisendsignaali sagedust, vaid ka selle ümbritsevaid sagedusi. Minimeerides korduste arvu, saime tulemuseks kahte punkti, millele vastavad sagedused said määratud kui 438,42 ja 448,64 Hz, kusjuures viimase amplituud oli mitmekordselt väiksem võrreldes esimesega.

Skeemi testimise osana oli mõõdetud selle voolutarbe, kusjuures toiteallika rolli täitis mikrokontrolleri väljaviikudega V_{IN} ja GND ühendatud toiteadapter. Multimeetri näit toitepinge 5 V juures jäi alla 100 mA. See tähendab, et on võimalik kaaluda patareide kasutamist süntesaatori toiteallikana. Seda rolli võib täita ka kaasaskantavate seadmete laadimiseks kasutatav akupank.

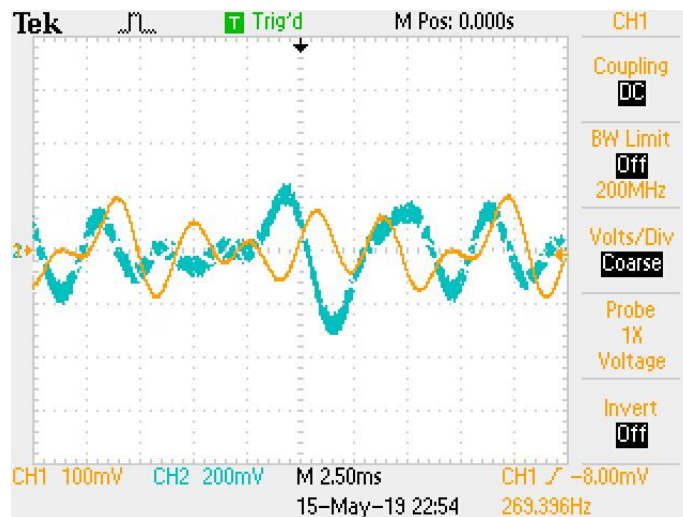
Kaudselt mõõdetud oli ka puhvri osa (suurusega 256 diskreeti) töötlemise aeg. Funktsioon *micros()*, mis tagastab mikrokontrolleri sisselülitamisest möödunud aega mikrosekundites, oli välja kutsutud funktsiooni *processADCInput* alguses ja lõpus. Tulemus olid moodustatud nende arvude vahel ja trükitud välja. Selgus, et kõigi 9 kanali kasutamisel keskmine töötlemise aeg jäi 6240 mikrosekundi ümbrusse ning

maksimaalseks ajaks oli ligikaudu 7220 mikrosekundit. Liites seda puhvris sisalduva signaali pikkusele vastava ajaga (8 ms), saame maksimaalseks viiteks 15,22 ms. Otsese ostsillograafi mõõtmise teel oli leitud, et viide võib mõnedel juhtudel osutada märgatavalt suuremaks – kuni 50 millisekundit, nagu näitab Joonis 16. Seega tulevikus on soovitud uurida viite vähendamise ja algoritmide optimeerimise võimalusi.



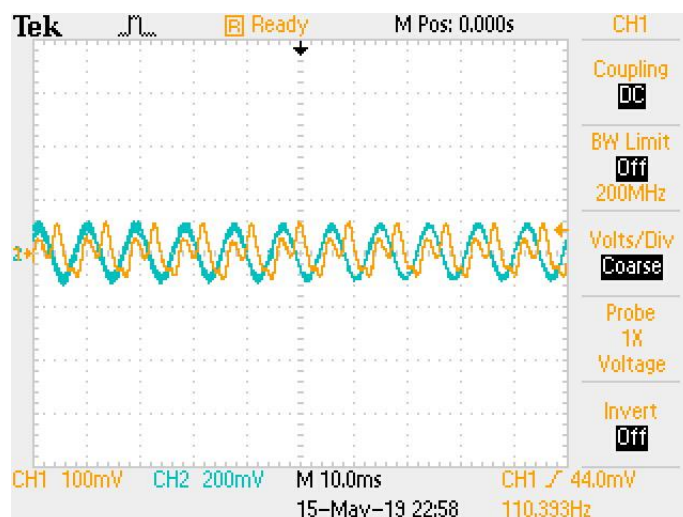
Joonis 16. Ajaline viide sisend- ja väljundsignaali vahel. Horisontaaltelje ühele ruudule vastab 25 ms. Näidiseks on toodud kahte signaali näitava ostsillograafi ekraanitõmmised, kusjuures sisendiks on elektrikitarrist tulenev signaal. Ekraanitõmmiste tegemiseks on kasutatud ostsillograafi Tektronix TDS 2022B. Joonistel kollane joon tähistab sisendsignaali ning sinine joon näitab väljundsignaali. Nendes näidistes kasutatakse vaid ühte operaatorit kanali kohta, mis genereerib siinussignaali.

Joonis 17 näitab kahte signaali polüfoonilises režiimis. Saab näha, et signaalid on omavahel nihkes töötlemisviide tõttu. Siiski on nähtav, et signaalide perioodid ligikaudselt ühtivad.



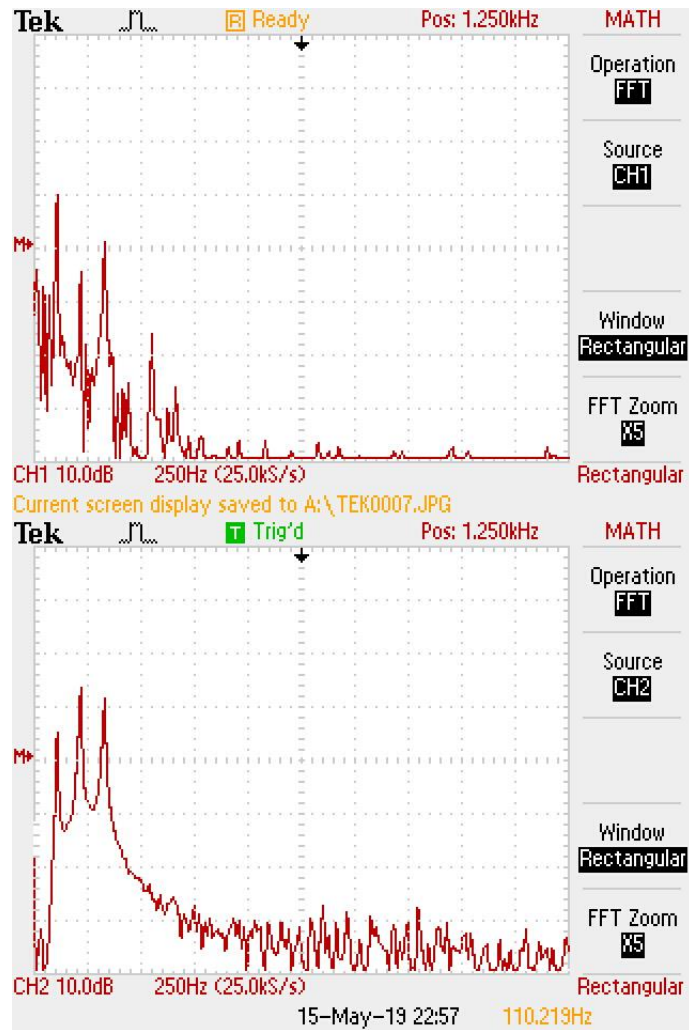
Joonis 17. Süntesaatori sisend- ja väljundsignaali graafik (polüfoonilises režiimis).

Joonis 18 on näide monofoonilisest režiimist. Monofoonilises režiimis kasutatakse vaid ühte heligeneraatori kanalit korraga. Antud näidisel väljundsignaali sagedus suure täpsusega langeb kokku sisendsignaali fundamentaalsagedusega.



Joonis 18. Süntesaatori sisend- ja väljundsignaali graafik (monofoonilises režiimis).

Joonis 19 näitab sisend- ja väljundsignaalide spektreid. Kuna mõlema graafiku samaaegne kuvamine ei olnud võimalik, need peegeldavad ühe noodi kõla erinevatel ajahetketel.



Joonis 19. Sisendsignaali (üleval) ja väljundsignaali (all) spektrite graafikud.

Siit on näha, et väljundsignaal sisaldab vähem kõrgemate sagedustega spektraalkomponente, sest neid ei sünteesita. Samuti tuleb tunnustada, et väljundi müra tase on suhteliselt kõrge. Osa müra tekib skeemi analoogosades varjestuse puudumise ja pikkade juhtmete kasutamise tõttu, mida on võimalik parandada trükkplaadi ja korpuse valmistamisel.

9 Kokkuvõte

Bakalaureusetöö eesmärgiks oli arendada reaalaja polüfoonilist helisüntesaatori mikrokontrolleri baasil ja luua juhtmevaba liidest selle juhtimiseks. Eesmärgi täitmiseks oli koostatud elektriskeem makettplaadil. Skeemi põhiliseks osaks oli valitud arendusplatvorm Teensy 3.6, milles sisalduv mikrokontroller näitas häid tulemusi kiiretoimelisuse poolest. Signaali sünteesiks oli edukalt rakendatud programmeeritav heligeneraator Yamaha YM3812. Mikrokontrolleri jaoks oli kirjutatud püsivara, mille lähtekood on saadaval aadressilt <https://gitlab.com/fastman1010/teensyfft/>.

Valminud seade sobib muusikalõikude ja heliefektide esitamiseks ja salvestamiseks, milleks tuleb ühendada kitarrisüntesaatori väljundi helivõimendi või lindistusseadmega. Süntesaatorit demonstreeriv video on saadaval veebilehelt aadressil <http://www.tud.ttu.ee/web/edwebe/synth/>.

Kõige olulisemaks piiranguks osutus kitarri keele võnkumine kui füüsikaline protsess, mille tagajärjel signaalis tekkivad kõrgemad harmoonilised sagedused, mis on fundamentaalsageduse kordsed. Juhul, kui mitu keelt võnkuvad samaaegselt, ei ole võimalik kõrgemate nootide fundamentaalsagedusi madalamate nootide harmoonilistest komponentidest. Lisaks tuleb mainida, et spektris üksteisele lähedal asuvate sageduste analüüs ei anna alati täpset tulemust, mis on eriti tunda signaali madalsageduslikus osas.

Seadme juhtmevaba juhtimine oli realiseeritud Bluetooth-ühenduse kaudu, milleks tuleb edastada käsud terminali kaudu teksti kujul. Sellega seotud potentsiaalne arendussuund seisneb nendele käskudele tugineva graafilise kasutajaliidese loomises, et tõsta kasutusmugavust. Teistest tulevikuplaanidest võib esile tuua teiste algoritmide katsetamist sageduste täpsemaks määramiseks. Samuti võib kaaluda elektriskeemi edasist arendamist voolutarbe ja müra taseme minimeerimiseks ning trükkplaadi valmistamist. Vajadusel on võimalik realiseerida spektri analüüsi tulemuste salvestamist mälukaardile hilisemaks taasesitamiseks, muusikaliste nootide määramist ja kirjapanemist ning signaali tarkvaralisele sünteesile üleminekut, mis kaotaks vajadust välise heligeneraatori kasutamiseks ja laiendaks signaalikuju muutmise võimalusi.

Olulise eelise võiks anda MIDI-liidese tugi, mis teeks helisüntesaatori ühilduvaks teiste digitaalsete heliseadmetega ning laiendaks võimalusi seadme juhtimiseks ja selle abil muusika esitamiseks.

Kasutatud kirjandus

- [1] A. Watts, „Musicology: A History Of The Guitar Effect Pedal,“ [Võrgumaterjal]. <http://www.mixdownmag.com.au/musicology-history-guitar-effect-pedal>. [Kasutatud 30.04.2019].
- [2] Musicradar, „The history of the guitar synth,“ 19.11.2015. [Võrgumaterjal]. <https://www.musicradar.com/news/guitars/the-history-of-the-guitar-synth-630650>. [Kasutatud 30.04.2019].
- [3] D. Earl, „A Brief History of FM Synthesis,“ 01.11.2013. [Võrgumaterjal]. <https://ask.audio/articles/a-brief-history-of-fm-synthesis>. [Kasutatud 29.04.2019].
- [4] Thomann, „Guitar/Bass Synthesizers,“ [Võrgumaterjal]. https://www.thomann.de/intl/ee/guitar_and_bass_synthesizers.html. [Kasutatud 02.05.2019].
- [5] Rekkerd.org, „Fretted Synth,“ 15.06.2010. [Võrgumaterjal]. <https://rekkerd.org/fretted-synth/>. [Kasutatud 19.05.2019].
- [6] A. Keltz, „Opening Pandora’s Box? The "L" word - latency and digital audio systems,“ [Võrgumaterjal]. <http://whirlwindusa.com/support/tech-articles/opening-pandoras-box/>. [Kasutatud 08.05.2019].
- [7] PJRC Store, „Teensy 3.6,“ [Võrgumaterjal]. <https://www.pjrc.com/store/teensy36.html>. [Kasutatud 29.04.2019].
- [8] T. Lorensen, „The DSP capabilities of ARM® Cortex®-M4 and Cortex-M7 Processors,“ 11.2016. [Võrgumaterjal]. https://community.arm.com/cfs-file/__key/communityserver-blogs-components-weblogfiles/00-00-00-21-42/7563.ARM-white-paper-_2D00_-DSP-capabilities-of-Cortex_2D00_M4-and-Cortex_2D00_M7.pdf. [Kasutatud 05.05.2019].
- [9] NXP Semiconductors, „Kinetis K66 Sub-Family 180 MHz ARM® Cortex®-M4F Microcontroller,“ 04.2017. [Võrgumaterjal]. <https://www.nxp.com/docs/en/data-sheet/K66P144M180SF5V2.pdf>. [Kasutatud 29.04.2019].
- [10] Amplified Parts, „Basic Electric Guitar Circuits 1: Pickups,“ [Võrgumaterjal]. <https://www.amplifiedparts.com/tech-articles/basic-electric-guitar-circuits-1-pickups>. [Kasutatud 04.05.2019].
- [11] Texas Instruments, „NE5532x, SA5532x Dual Low-Noise Operational Amplifiers,“ 01.2015. [Võrgumaterjal]. <http://www.ti.com/lit/ds/symlink/ne5532.pdf>. [Kasutatud 16.05.2019].
- [12] ON Semiconductor, „Internally Compensated Dual Low Noise Operational Amplifier,“ 09.2013. [Võrgumaterjal]. <https://www.onsemi.com/pub/Collateral/NE5532-D.PDF>. [Kasutatud 16.05.2019].
- [13] R. T. Paynter, Introductory Electronic Devices and Circuits, Electron Flow Version/Sixth Edition, New Jersey: Prentice Hall, 2003.
- [14] B. Troise, „Compositional Strategies For Programmable Sound Generators With Limited Polyphony,“ 16.07.2015. [Võrgumaterjal].

- <https://www.ludomusicology.org/2015/07/16/compositional-strategies-for-programmable-sound-generators-with-limited-polyphony/>. [Kasutatud 02.05.2019].
- [15] R. Atkinson, „Sound generators of the 1980s home computers,“ [Võrgumaterjal]. <http://www.atkinsoft.com/soundgenerators.html>. [Kasutatud 07.05.2019].
- [16] Nippon Gakki Co., Ltd, „YM3812: FM Operator Type-L II (OPL II),“ 04.1986. [Võrgumaterjal]. https://web.archive.org/web/20160519173446/http://dtech.lv/files_ym/ym3812.pdf. [Kasutatud 07.05.2019].
- [17] Yamaha Corporation, „YM3014B Serial Input Floating D/A Converter (DAC-SS),“ 1987. [Võrgumaterjal]. <http://www.atkinsoft.com/datasheets/YM3014B.PDF>. [Kasutatud 09.05.2019].
- [18] Shenzhen Xintai Micro Technology Co., Ltd., „JDY-30 SPP Bluetooth Module Manual Product Introduction,“ [Võrgumaterjal]. <http://www.ko4bb.com/getsimple/data/uploads/jdy-30/jdy-30-bluetooth-module.pdf>. [Kasutatud 07.05.2019].
- [19] R. Atkinson, „Schematic for the Commodore C64/C128 Sounder Expander cartridge. Both analogue and digital parts; reverse-engineered,“ 12.1987. [Võrgumaterjal]. <http://www.zimmers.net/anonftp/pub/cbm/schematics/cartridges/c64/sfx/sfx-sch.gif>. [Kasutatud 07.05.2019].
- [20] Arm Ltd., „Complex FFT Functions,“ 18.03.2019. [Võrgumaterjal]. https://www.keil.com/pack/doc/CMSIS/DSP/html/group__ComplexFFT.html. [Kasutatud 12.05.2019].
- [21] I. Arro ja J. Derkatš, Digisignaali töötlemine, Tallinn: TTÜ Kirjastus, 2005.
- [22] P. Stoffregen, „Teensy Audio Library / input_adc.cpp,“ 03.02.2019. [Võrgumaterjal]. https://github.com/PaulStoffregen/Audio/blob/master/input_adc.cpp. [Kasutatud 17.05.2019].
- [23] NXP Semiconductors, „K66 Sub-Family Reference Manual,“ 04.08.2018. [Võrgumaterjal]. <https://www.nxp.com/docs/en/reference-manual/K66P144M180SF5RMV2.pdf>. [Kasutatud 17.05.2019].
- [24] B. A. Olshausen, „Aliasing,“ 10.10.2000. [Võrgumaterjal]. <http://www.rctn.org/bruno/npb261/aliasing.pdf>. [Kasutatud 17.05.2019].
- [25] Sound Engineering Academy, „Audio Frequency Range of Various Musical Instruments (Infographic),“ [Võrgumaterjal]. <http://www.seaindia.in/blog/audio-frequency-range-of-various-musical-instruments-infographic/>. [Kasutatud 17.05.2019].
- [26] R. Lyons, „Windowing Functions Improve FFT Results, Part I,“ 01.06.1998. [Võrgumaterjal]. <https://www.edn.com/electronics-news/4383713/Windowing-Functions-Improve-FFT-Results-Part-I>. [Kasutatud 17.05.2019].
- [27] MathWorks, „Minimum four-term Blackman-Harris window,“ [Võrgumaterjal]. <https://se.mathworks.com/help/signal/ref/blackmanharris.html>. [Kasutatud 17.05.2019].
- [28] Arm Ltd., „Vector Multiplication,“ 18.03.2019. [Võrgumaterjal]. https://www.keil.com/pack/doc/CMSIS/DSP/html/group__BasicMult.html. [Kasutatud 17.05.2019].
- [29] M. Donadio, „How to Interpolate the Peak Location of a DFT or FFT if the

- Frequency of Interest is Between Bins," 19.04.1999. [Võrgumaterjal].
<https://dspguru.com/dsp/howtos/how-to-interpolate-fft-peak/>. [Kasutatud 14.05.2019].
- [30] Arm Ltd., „Linear Interpolation," 18.03.2019. [Võrgumaterjal].
https://www.keil.com/pack/doc/CMSIS/DSP/html/group__LinearInterpolate.html. [Kasutatud 18.05.2019].
- [31] V. Arnost, „Programmer's Guide to Yamaha YMF 262/OPL3 FM Music Synthesizer," 23.11.2000. [Võrgumaterjal].
<https://www.fit.vutbr.cz/~arnost/opl/opl3.html>. [Kasutatud 30.04.2019].
- [32] Nerdly Pleasures, „Digital OPL2 and OPL3 Recording," 24.02.2015. [Võrgumaterjal]. <http://nerdlypleasures.blogspot.com/2015/02/digital-opl2-and-opl3-recording.html>. [Kasutatud 12.05.2019].
- [33] J. S. Lee, „Programming the AdLib/Sound Blaster FM Music Chips," 24.02.1992. [Võrgumaterjal]. <http://qzx.com/pc-gpe/adlib.txt>. [Kasutatud 30.04.2019].
- [34] M. Gambrell ja O. Niemitalo, „OPLx decapsulated," 20.04.2008. [Võrgumaterjal].
https://docs.google.com/document/d/18IGx18NQY_Q1PJVZ-bHywao9bhsDoAqoIn1rIm42nwo/. [Kasutatud 10.05.2019].
- [35] habr, „Программирование&Музыка: ADSR-оггибающая сигнала. Часть 2," 05.10.2016. [Võrgumaterjal]. <https://habr.com/ru/post/311750/>. [Kasutatud 17.05.2019].

Lisa 1 – Funktsioon *findMaxValues*

```
uint32_t findMaxValues(float32_t *specAbs, float32_t *specCompl,
                      uint8_t numFreqs, channelState *freqStates) {
    bool freqExists;
    float32_t maxValue, kAppr, freqAppr, amplAppr, ap, dp, am, dm, d;
    uint8_t i, j;
    uint32_t k, x0, x1;
    uint32_t freqsUsed = 0;

    for (i = 0; i < ((float32_t) MIN_FREQ / SAMP_RATE * FFT_LEN); i++) {
        specAbs[i] = 0;
    }

    for (i = 0; i < numFreqs; i++) {
        freqExists = false;
        arm_max_f32(specAbs, FFT_LEN / 2, &maxValue, &k);
        if (maxValue > amplMin) {
            ap = (specCompl[(k + 1) * 2] * specCompl[k * 2] +
                  specCompl[(k + 1) * 2 + 1] * specCompl[k * 2 + 1]) /
                (powf(specCompl[k * 2], 2) + powf(specCompl[k * 2 + 1], 2));
            dp = -ap / (1 - ap);
            am = (specCompl[(k - 1) * 2] * specCompl[k * 2] +
                  specCompl[(k - 1) * 2 + 1] * specCompl[k * 2 + 1]) /
                (powf(specCompl[k * 2], 2) + powf(specCompl[k * 2 + 1], 2));
            dm = am / (1 - am);
            d = (dp + dm) / 2 + tau(dp * dp) - tau(dm * dm);
            kAppr = (float32_t) k + d;
            freqAppr = kAppr * ((float) SAMP_RATE / FFT_LEN);

            for (j = 0; j < freqsUsed; j++) {
                if (freqAppr >= (1.0 - MAX_DEV) * freqStates[j].freq &&
                    freqAppr <= (1.0 + MAX_DEV) * freqStates[j].freq) {
                    freqExists = true;
                    break;
                }
            }

            if (!freqExists) {
                x0 = floor(kAppr);
                x1 = ceil(kAppr);
                amplAppr = specAbs[x0] + (kAppr - x0) *
                    ((specAbs[x1] - specAbs[x0]) / (x1 - x0));
                freqStates[freqsUsed].ampl = amplAppr;
                freqStates[freqsUsed].freq = freqAppr;
            }
        }
    }
}
```

```
    freqStates[freqsUsed++].active = false;
}

specAbs[k] = 0;
specAbs[k - 1] *= 0.5;
specAbs[k + 1] *= 0.5;

} else {
    break;
}
}

return freqsUsed;
}
```