

Ismo Kela

## **3D-TULOSETTU ARDUINO-POHJAINEN USB-C-KITARAOHJAIN**

# **3D-TULOSETTU ARDUINO-POHJAINEN USB-C-KITARAOHJAIN**

Ismo Kela  
Opinnäytetyö  
Kevät 2023  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

---

Tekijä: Ismo Kela

Opinnäytetyön nimi: 3D-tulostettu Arduino-pohjainen USB-C-kitaraohjain

Työn ohjaaja: Olli Himanka

Työn valmistumislukukausi ja -vuosi: Kevät 2023

Sivumäärä: 27

---

Tämän opinnäytetyön tarkoituksena oli dokumentoida Arduino-mikrokontrolleria käyttävän USB-kitaraohjaimen valmistus ja kuvata ohjaimen toimintaa, ohjelmistoa ja käytettyjä komponentteja. Ohjaimen valmistukseen kuului ohjaimen rungon suunnittelu, tulostaminen 3D-tulostimella, ohjaimen kokoaminen komponenteista sekä Arduino-mikrokontrollerin ohjelmointi.

Työn toteutuksessa hyödynnettiin sekä koulutuksen että harrastusten kautta opittuja taitoja. Ohjaimen 3D-malleja muokattiin Blender-ohjelmistolla ja tulostettiin Oulun ammattikorkeakoulun (Oamk) tiloissa Prusa i3 MK3S -tulostimella. Ohjaimen toiminnan mahdollistava mikrokontrollerin ohjelmisto tehtiin Arduino IDE -ohjelmalla ja ajettiin SparkFun Pro Micro USB-C -mikrokontrolleriin.

Työn tuloksena on valmis USB-kitaraohjain, joka toimii emuloimalla näppäimistöä ja lähettämällä näppäimistöpainalluksia suoraan tietokoneeseen sisäänrakennetun USB-yhteyden kautta. Kitaraohjaimen käyttötarkoituksena on toimia peliohjaimena Clone Hero -tietokonepelille, jossa pelaaja soittaa pelissä olevia musiikkikappaleita painelemalla ohjaimen painikkeita oikeaan aikaan rytmisissä kuvaruudulle ilmestyvien kuvakkeiden kanssa.

---

Asiasanat: 3D-tulostus, Arduino, USB-C, Mikrokontrollerit, Kitarat, Clone Hero

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Information Technology, Option of Software Development

---

Author: Ismo Kela  
Title of thesis: 3D printed Arduino-based USB-C guitar controller  
Supervisor: Olli Himanka  
Term and year when the thesis was submitted: Spring 2023  
Number of pages: 27

---

The purpose of this thesis was to document the manufacturing of a USB guitar controller using an Arduino microcontroller as well as describing the functions, software and the components of the guitar controller. The process of manufacturing the controller involved designing and 3D printing the chassis, assembling the controller with the required components and programming the Arduino-microcontroller to function on the PC.

Skills learned through both education and hobbies were utilized in the implementation of this thesis. 3D models for the controller were modified using Blender software and printed in the premises of Oulu University of Applied Sciences using a Prusa i3 MK3S printer. The program for the controller was created using Arduino IDE and then uploaded to SparkFun Pro Micro USB-C microcontroller.

The product of this thesis is a finished, functioning USB guitar controller that works by emulating a keyboard and sending keypresses to the PC directly using the microcontrollers built-in USB functionality. The purpose of the guitar controller is to act as a game controller for the PC game Clone Hero, in which players play the songs in the game by pressing buttons on the controller at the correct time in rhythm with icons appearing on screen.

---

Keywords: 3D-printing, Arduino, USB-C, Microcontrollers, Guitars, Clone Hero

# SISÄLLYS

1	JOHDANTO .....	6
2	TYÖN TAUSTA JA TAVOITTEET .....	7
3	LAITTEISTO JA TEKNOLOGIAT .....	8
3.1	SparkFun Qwiic Pro Micro USB-C -mikrokontrollerialusta .....	8
3.2	ATmega32U4-mikrokontrolleri .....	8
3.3	AVR-mikrokontrolleriperhe .....	9
3.4	RISC-suunnittelu .....	10
3.5	USB-C-liitäntä .....	10
3.6	Kailh Low Profile Choc -näppäimistökytkimet .....	10
4	TYÖHÖN LIITTYVÄ OHJELMISTO JA KÄSITTEET .....	11
4.1	3D-tulostaminen ja siihen liittyvä ohjelmisto .....	11
4.2	Blender-ohjelmisto .....	11
4.3	Clone Hero -tietokonepeli .....	12
4.4	Guitar Hero -konsolipeli .....	12
5	OHJAIMEN VALMISTUKSEN TYÖVAIHEET .....	13
5.1	3D-mallit tulostusta varten .....	13
5.2	Osien tulostaminen ja kokoaminen .....	16
5.3	Tarvittavat komponentit .....	17
5.4	Komponenttien juotokset ja kytkentäkaavio .....	18
5.5	Arduinon ohjelmointi .....	18
6	OHJAIMEN TOIMINTA JA OHJELMISTO .....	20
6.1	Keyboard-kirjasto .....	20
6.2	Arduino-koodi .....	21
6.3	Ohjaimen testaus Clone Hero -pelissä .....	23
7	POHDINTA .....	24
	LÄHTEET .....	25

# 1 JOHDANTO

Clone Hero on ilmainen, vuonna 2017 julkaistu rytmivideopeli, jonka pelattavuus ja perusidea ovat lähes identtisiä alun perin RedOctanen ja Harmonixin yhteistyönä kehittämän Guitar Hero -pelisarjan kanssa. Clone Heron tärkeimpiä ominaisuuksia on mahdollisuus pelata yhteisön luomia kappaleita, minkä ansiosta se on pysynyt suosiossa. Pelin julkaisu on saanut aikaan uutta kiinnostusta genreä kohtaan ja myös virallisten kitarahjainten kysyntä on tästä syystä kasvanut. Virallisten ohjainten valmistus on kuitenkin loppunut ja käytettyjä ohjaimia on vain rajallinen määrä, minkä vuoksi monet ovat alkaneet valmistaa ohjaimia itse.

Tämän opinnäytetyön tarkoituksena on dokumentoida 3D-tulostamalla toteutetun kitarahjaimen valmistus. Työn tavoitteena on valmistaa toimiva, käyttökelpoinen prototyyppi ohjaimesta ja tuottaa dokumentaatio ohjaimen kokoamisesta, komponenttien kytkemisestä sekä ohjaimen ohjelmistosta.

Työn alussa käsitellään keskeisiä käsitteitä ja teknologioita sekä kerrotaan ohjaimen valikoituista komponenteista. Tämän jälkeen käydään läpi työn tausta ja tavoitteet. Seuraavaksi kuvataan itse ohjaimen valmistuksen työvaiheet, jonka jälkeen kerrotaan ohjaimen toiminnasta ja ohjelmistosta, sekä testataan ohjaimen toimivuus pelissä. Lopuksi pohditaan työn onnistumista sekä mahdollisia parannuskohteita.

Opinnäytetyö suoritetaan osana Oulun ammattikorkeakoulun tieto- ja viestintätekniikan tutkinto-ohjelmaa. Työ valmistui kevään 2023 aikana.

## 2 TYÖN TAUSTA JA TAVOITTEET

Tämän opinnäytetyön tavoitteena oli valmistaa toimiva prototyyppi valmiista kitarahjaimesta, joka toimii Sparkfunin Qwiic Pro Micro USB-C -mikrokontrollerialustalla ja yhdistyy tietokoneeseen USB-C-kaapelilla. Ohjain toimii käytännössä emuloimalla näppäimistöä ja käyttää painikkeinaan Dongguan Kaihua Electronicsin valmistamia mekaanisia Kailh Low Profile Choc -näppäimistökytkimiä. Opinnäytetyö toimii kitarahjaimen valmistuksen ja kokoamisen dokumentaationa.

Opinnäytetyön idea syntyi siitä, kun yritin etsiä käytettyä Guitar Hero -kitarahjainta verkosta tuloksetta. Käyttökelpoisista yksilöistä pyydettiin liian kovaa hintaa ja postikulut maksoivat usein jopa enemmän kuin itse tuote, kun tuotteet olisi tilattava ulkomailta. Törmäsin kuitenkin etsinnöissäni Josh Designsin YouTube-videoon 3D-tulostetun kitarahjaimen valmistuksesta ja ajattelin, että voisin itsekin valmistaa oman ohjaimen ja tehdä tästä valmistusprosessista opinnäytetyöni. Ohjain olisi samalla parempi kuin alkuperäiset Guitar Hero -ohjaimet, sillä se käyttäisi mekaanisia näppäimistökytkimiä, joissa on parempi tuntuma ja jotka kestävät käyttöä paremmin.

### 3 LAITTEISTO JA TEKNOLOGIAT

Tässä luvussa käsitellään kitarahjaimessa käytettävää laitteistoa ja niihin liittyviä teknologioita. Kitarahjaimessa käytettävät osat on valittu hyvän saatavuuden ja modernien ominaisuuksien vuoksi, kuten esimerkiksi mikrokontrollerialustassa käytetty USB-C-liitin.

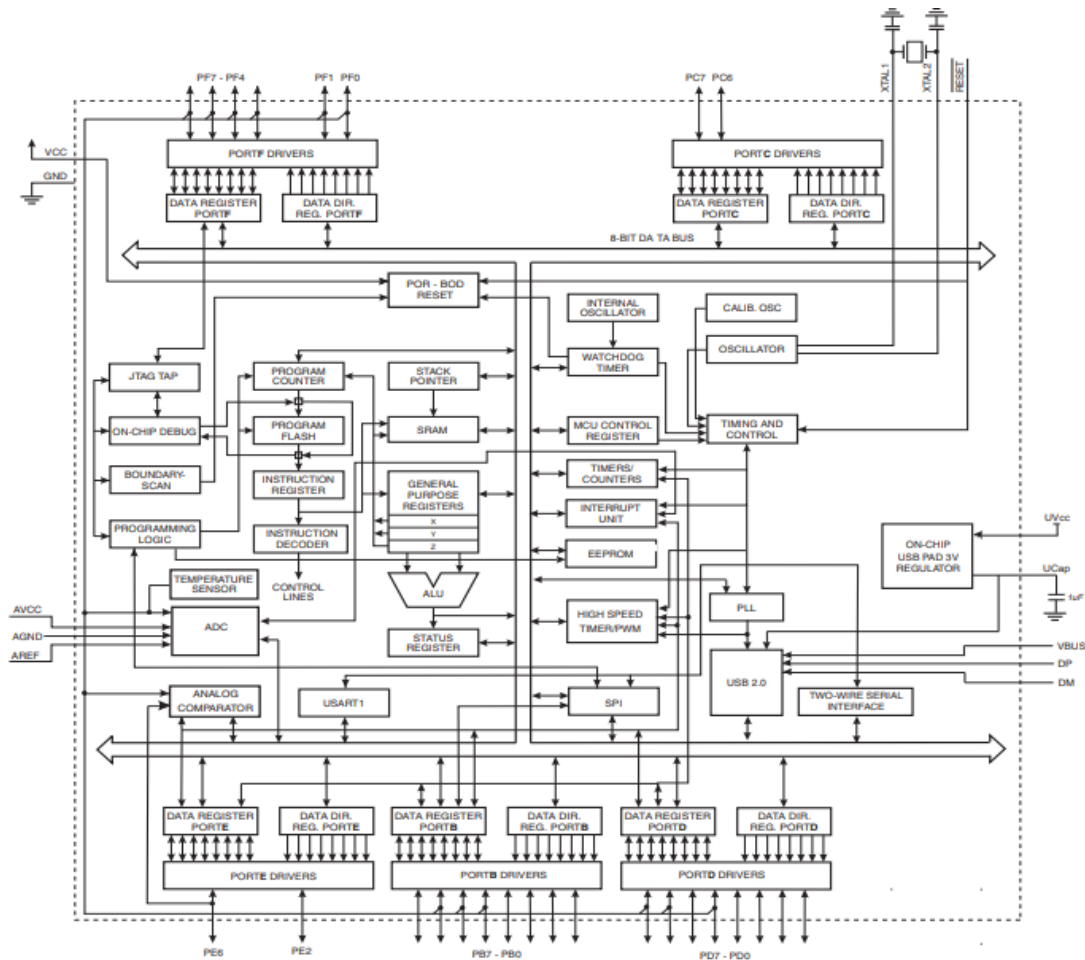
#### 3.1 SparkFun Qwiic Pro Micro USB-C -mikrokontrollerialusta

SparkFun Qwiic Pro Micro USB-C on pieni Arduinon Pro Microon pohjautuva mikrokontrollerialusta ATmega32U4-ohjaimella. Se saa tarvittavan virtansa USB-liitäntän kautta. Eroavaisuuksia alkuperäiseen Arduino Pro Microon on muun muassa reset-nappi ja Qwiic-liitin, joka mahdollistaa SparkFunin yhteensopivien laitteiden käyttämisen tämän liittimen kautta ilman tarvetta juotoksille. ATmega32U4-mikrokontrollerissa on sisäänrakennettu USB-yhteys, joten mikrokontrolleri ei tarvitse toissijaista suoritinta USB-datansiirtoon liitetulle laitteelle. Mikrokontrolleri myös käyttää alkuperäisen Pro Microon Micro-USB:n sijaan nykyaikaisempaa USB-C-liitäntää. (1).

#### 3.2 ATmega32U4-mikrokontrolleri

ATmega32U4 (kuva 1) on matalan virrankulutuksen 8-bittinen CMOS AVR RISC -pohjainen mikrokontrolleri sisäänrakennetulla USB-yhteydellä. Se noudattaa täysin USB 2.0 -spesifikaatiota. Suorittimen AVR-ytimessä on 32 yleiskäyttöistä rekisteriä, jotka ovat suoraan yhteydessä suorittimen aritmeettisloogiseen yksikköön, mikä mahdollistaa pääsyn kahteen erilliseen rekisteriin yhdellä käskyllä, joka suoritetaan yhdessä kellojaksossa. Tämä tekee suorittimesta kooditehokkaamman ja mahdollistaa jopa kymmenkertaisen suorituskyvyn perinteisiin CISC-mikrokontrollereihin verrattuna. (2; 3).





KUVA 1. ATmega32U4-mikrokontrollerin lohkaavio (2.)

### 3.3 AVR-mikrokontrolleriperhe

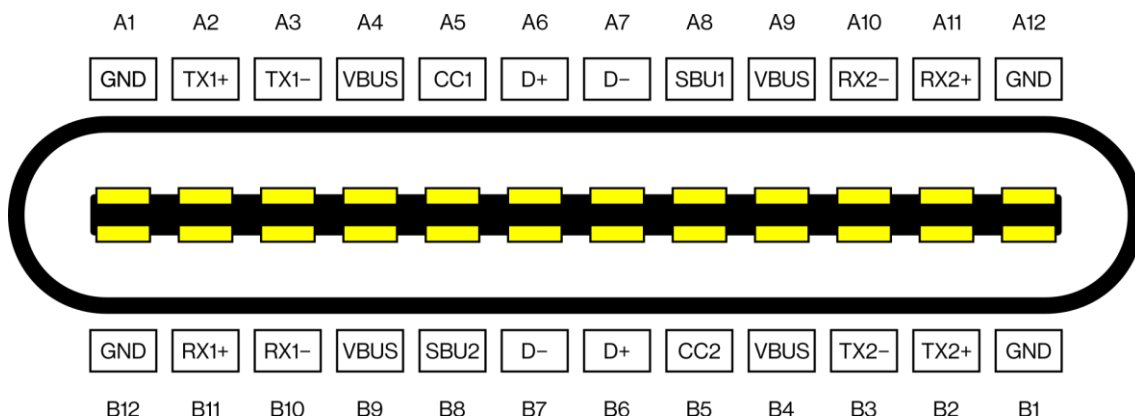
AVR on alun perin Atmelin ja nykyisin Microchipin mikrokontrolleriperhe, johon kuuluu laaja valikoima 8-bittisiä mikrokontrollereita, jotka ovat tulleet suosituksi erityisesti Arduino-laitteissa. Microchip osti Atmelin vuonna 2016. AVR-mikrokontrollerit jaetaan kolmeen ryhmään niiden ominaisuuksien sekä muistin määrän mukaan: ATmega, AT90x ja ATtiny. Näistä ainoastaan ATmegassa on sisäänrakennettu USB-ohjain. Kaikki AVR-mikrokontrollerit sisältävät ainakin RISC-ytimen, FLASH-ohjelmamuistia, SRAM-käyttömuistia sekä useita I/O-väyliä. (4).

### 3.4 RISC-suunnittelu

RISC eli Reduced Instruction Set Computer on suoritinarkkitehtuurien suunnittelufilosofia, jossa yksittäiset käskyt pyritään pitämään yksinkertaisina ja nopeasti suoritettavina. Laajasti varsinkin mobiililaitteissa käytetty ARM-arkkitehtuuri perustuu RISC-ajatteluun. RISC-suorittimien käskyjen yksinkertaisuuden vuoksi niitä tarvitaan suurempi määrä saman tehtävän suorittamiseen, mikä lisää tarvittavan ohjelmamuistin määrää. (5).

### 3.5 USB-C-liitäntä

USB-C (kuva 2) on 24-nastainen datan ja virran siirtoon käytettävä USB-liitäntä. Se on standardoitu IEC-standardissa IEC 62680-1-3:2021. USB-C-liitin on symmetrinen, joten sen voi asettaa laitteen kummin päin tahansa. USB-C:n suurin tiedonsiirtonopeus on yleisimmin käytetyssä USB 3.0 -standardissa 20 gigabittiä sekunnissa ja uusimmassa USB4 Version 2:ssa 80 gigabittiä sekunnissa. Suurten tiedonsiirtonopeuksien ansiosta sillä voi välittää myös DisplayPort-standardin mukaista videokuvaa. (6; 7.)



KUVA 2. USB-C-liittimen pinnit (6.)

### 3.6 Kailh Low Profile Choc -näppäimistökytkimet

Kailh Low Profile Choc on Dongguan Kaihua Electronicsin valmistama matalan profiilin mekaaninen näppäimistökytkin (8). Tässä työssä käytettävät näppäimistökytkimet ovat lineaarinen ja suhteellisen kevyt Choc Red ja napsahtava (clicky) Choc White. Kummankin kytkimen painallukseen tarvittava voima on 50 gf (gram-force).

## 4 TYÖHÖN LIITTYVÄ OHJELMISTO JA KÄSITTEET

Tässä luvussa käsitellään kitarahajaimen valmistuksessa käytettäviä ohjelmistoja ja menetelmiä sekä avataan työhön liittyviä käsitteitä.

### 4.1 3D-tulostaminen ja siihen liittyvä ohjelmisto

3D-tulostaminen on esineiden valmistamista 3D-tulostimella. Yleisimmin 3D-tulostukseen käytetään jonkinlaista muovia. Tulostettava materiaali johdetaan tulostuspäähän, joka sulattaa materiaalin alustalle, johon se jähmettyy ohueksi kerrokseksi muodostaen kerros kerrokselta kokonaisen esineen. 3D-tulostin tulostaa halutun esineen yhteensopivan 3D-mallin perusteella. 3D-malli pilkotaan ohjelmallisesti kerroksiksi ja muutetaan tulostimen ymmärtämään G-koodimuotoon. (9; 10).

Prusa Slicer on avoimen lähdekoodin sovellus, jolla muutetaan 3D-malleja tulostuskelpoiseksi G-kooditiedostoiksi. Ohjelma leikkaa siihen ladatut 3D-mallit yksittäisiksi tulostettaviksi kerroksiksi ja määrittelee tulosteen kiinteiden osien tiiviiden ja tarvittavat tukirakenteet. (11).

G-koodi on matalan tason komentokieli, jota käytetään esimerkiksi CNC-työstökoneiden ja 3D-tulostimien ohjaukseen. G-koodi kertoo koneelle mitä toimintoja sen tulisi suorittaa ja missä järjestyksessä sekä suoritettavien toimintojen koordinaatit. (12).

### 4.2 Blender-ohjelmisto

Blender on ilmainen avoimen lähdekoodin GPL-lisensioitu 3D-mallinnuksen ohjelmisto. Se kehitettiin alun perin NEO-GEO-nimisen mainostoimiston sisäiseen käyttöön IRIX-käyttöjärjestelmälle. Nykyisin sen kehityksestä vastaa Blender Foundation -säätiö. Blenderin tärkeimpiin ominaisuuksiin kuuluu suuri valikoima laajennuksia, joilla voidaan muokata ohjelman käyttökokemusta sekä lisätä ominaisuuksia ja toimintoja helpottamaan työskentelyä. Blender tukee myös 3D-mallien vientiä ja tuontia muihin tiedostomuotoihin ja ohjelmiin. (13).

### **4.3 Clone Hero -tietokonepeli**

Clone Hero on Ryan Fosterin kehittämä, vuonna 2017 julkaistu rytmivideopeli. Peli on toiminnaltaan ja perusidealtaan lähes identtinen Guitar Heron kanssa. Clone Heron tärkeimpiä ominaisuuksia on mahdollisuus pelata yhteisön luomia kappaleita. Clone Hero siirtyi pois beta-vaiheesta 29.11.2022, kun pelin versio v1.0.0.4080 julkaistiin. (14; 15; 16).

### **4.4 Guitar Hero -konsolipeli**

Guitar Hero on alun perin Harmonixin ja sittemmin Neversoftin kehittämä rytmivideopelisarja, jossa pelaajat käyttävät muun muassa kitaran ja rumpujen mallisia peliohjaimia soittaakseen pelissä olevia kappaleita. Pelaaminen tapahtuu painamalla ohjaimessa olevia painikkeita samanaikaisesti, kun painiketta vastaava kuvio liukuu ruudussa oikealle kohdalle rytmissä musiikin tahtiin. (17).

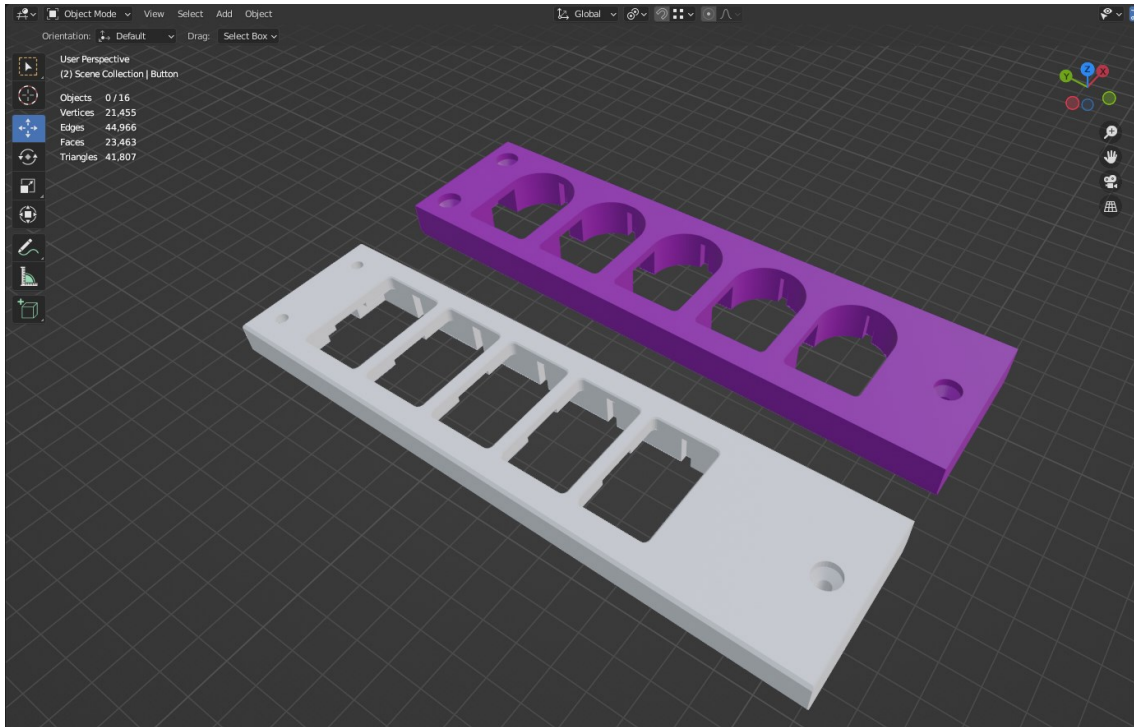
## 5 OHJAIMEN VALMISTUKSEN TYÖVAIHEET

Tässä luvussa käsitellään ohjaimen valmistuksen vaiheita sekä 3D-malleihin tehtyjä muutoksia ja mahdollisia parannuksia. Lisäksi luvussa listataan ohjaimen kokoamiseen tarvittavat osat ja komponentit sekä selvennetään komponenttien kytkentää.

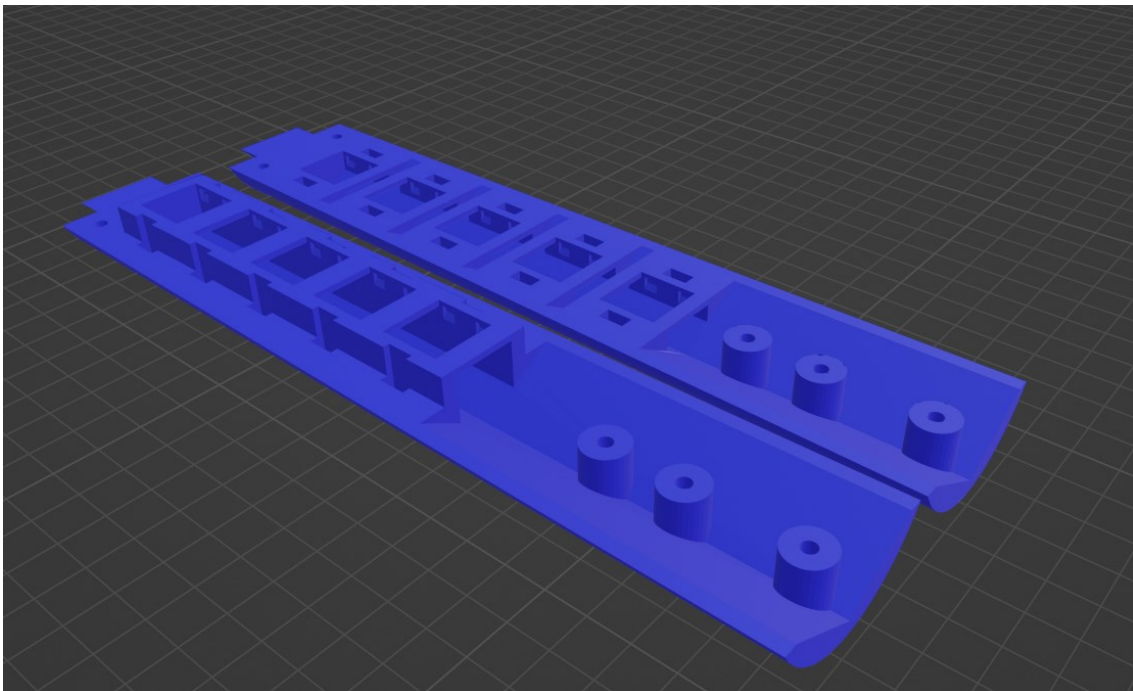
### 5.1 3D-mallit tulostusta varten

Tässä työssä käytettävät 3D-mallit ovat käyttäjän Josh Designs mallintamia ja ne ovat saatavilla ilmaiseksi Printables-verkkosivulta (18). Näihin 3D-malleihin on tehty pieniä muutoksia ja lisäyksiä Blender-3D-mallinnussovellusta käyttäen, jotta ne sopisivat paremmin työssä käytettäviin komponentteihin. Alkuperäiset 3D-mallit ovat Creative Commons Attribution 4.0 International Public License -lisenssin alaisia, mikä sallii mallien muokkaamisen, jakamisen ja kaupallisen käytön, kunhan mallien alkuperäinen tekijä on mainittu (19).

Tässä työssä käytetään alkuperäisten 3D-tulostettujen otelaudan painikkeiden sijasta näppäimistökytkimiin suoraan kiinnittyviä näppäinhattuja, joten otelaudan 3D-malleja muokattiin niin, että näppäimistökytkimet sijoittuvat lähemmäksi toisiaan (kuva 3). Otelaudan alemmaa puolikasta muokattiin myös, että kytkimiä saadaan lähemmäksi otelaudan pintaa matalampien näppäinten takia (kuva 4).



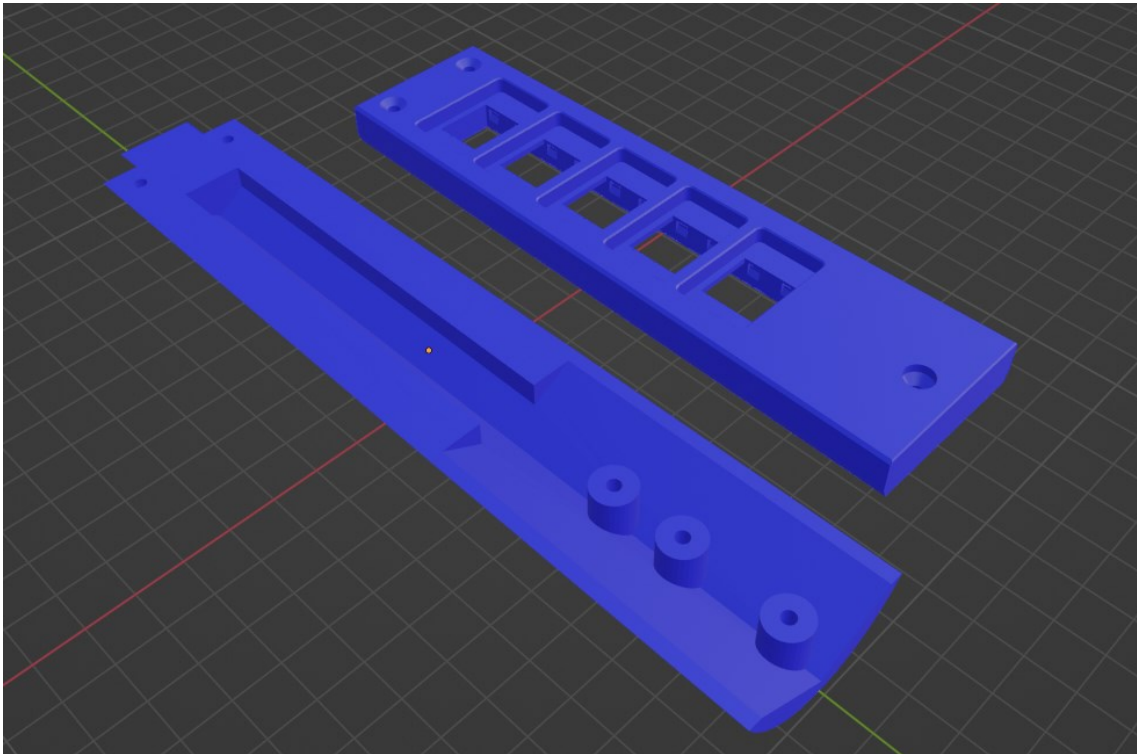
*KUVA 3. Violetti otelauta on alkuperäinen ja valkoinen muokattu*



*KUVA 4. Alkuperäinen matala kappale ja muokattu uusi kappale.*

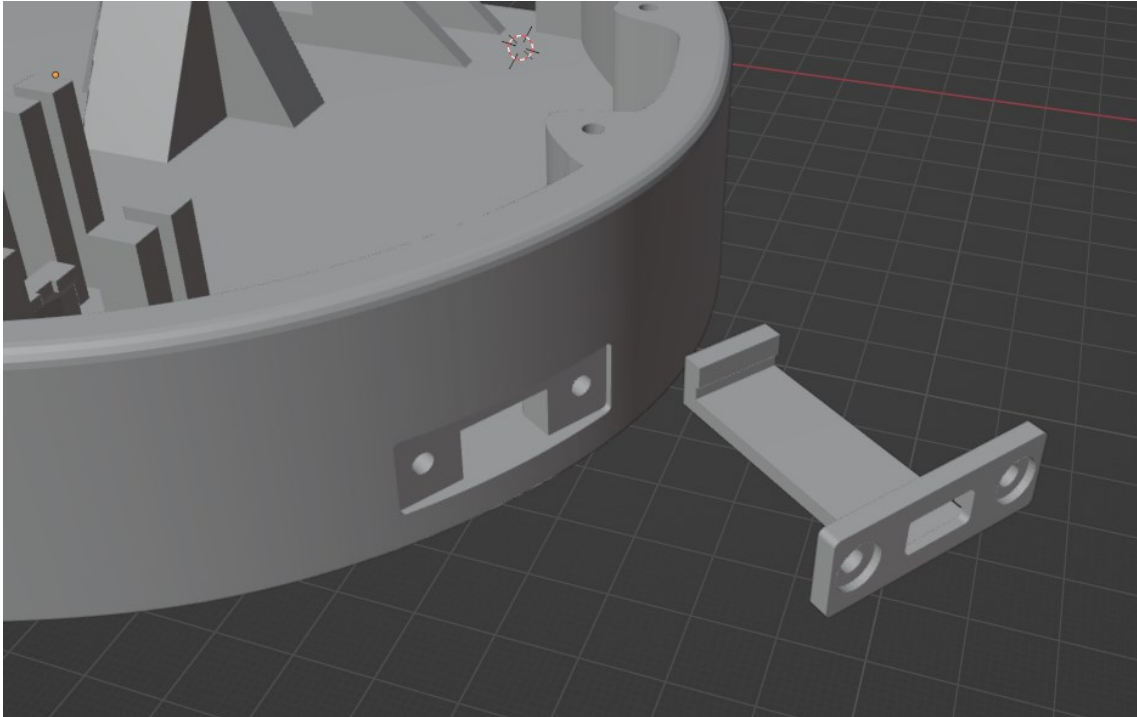
Otelaudasta ja sen alemmasta puoliskosta suunniteltiin myös uusi paranneltu versio (kuva 5), joka olisi helpottanut 3D-tulostusta vähentämällä vaadittujen tukirakenteiden määrää, mutta se ei ehtinyt

tulostuksiin mukaan. Tässä versiossa kytkinten kiinnityskohta on otelaudan ylemmässä osiossa, mikä helpottaisi myös kytkinten juottamisessa.



*KUVA 5. Otelaudan parannellut versiot*

Myös kitarahjaimen runkoa on muokattu ja siihen on lisätty kiinnityskohta erilliselle pidikkeelle (kuva 6), johon ohjaimessa käytetty mikrokontrolleri kiinnittyy. Mikrokontrollerin pidikettä on muokattu niin, että sen kiinnitykseen voidaan käyttää samoja M3-ruuveja, joita käytetään ohjaimen muiden osien kiinnitykseen.



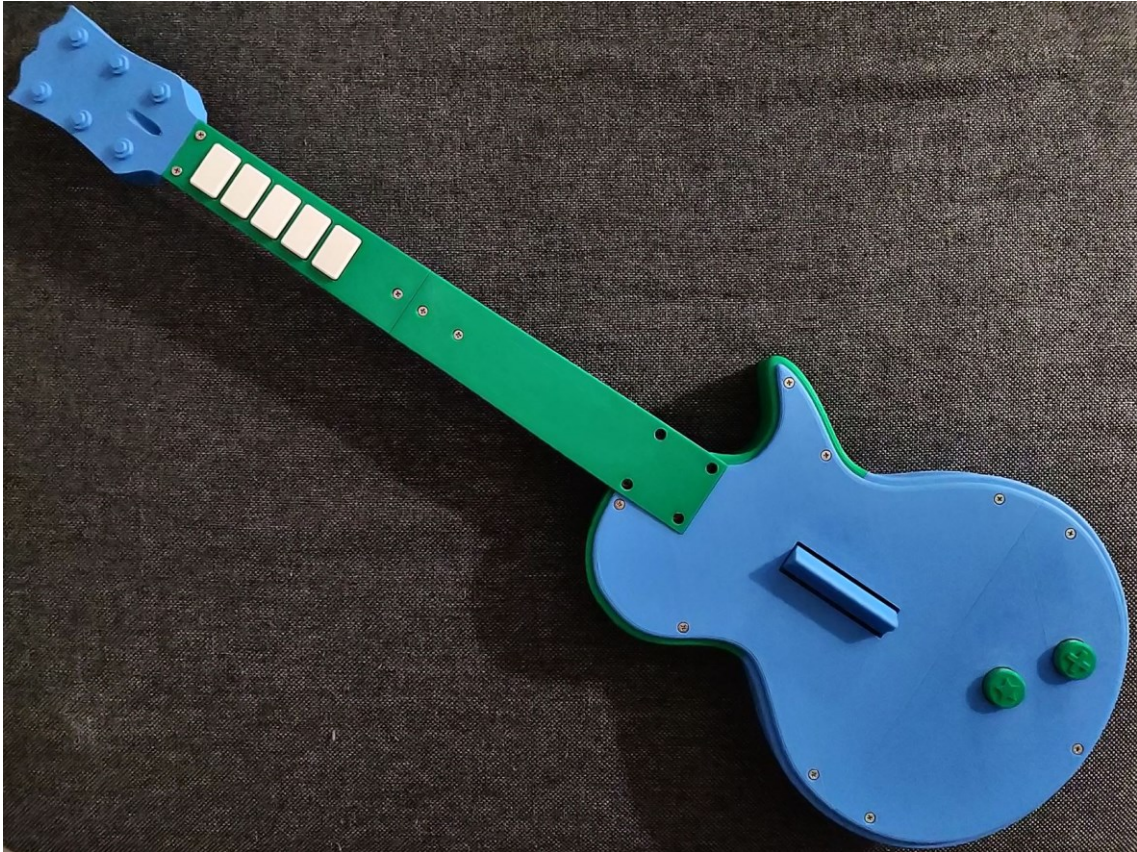
*KUVA 6. Kitaran runkoon lisätty kiinnityskohta ja siihen kiinnittyvä mikrokontrolleripidike.*

## **5.2 Osien tulostaminen ja kokoaminen**

Kitaraohjaimen (kuva 7) osat tulostettiin Oamkin elektroniikkalaboratoriossa Prusa i3 MK3S 3D-tulostimella. Tulostettavat 3D-mallit tuotiin muistitikulta Prusa Slicer -ohjelmaan, jolla mallit saatiin aseteltua tulostusalustalle ja muutettua 3D-tulostimen ymmärtämäksi G-kooditiedostoksi. Tämä tiedosto vietiin ohjelmasta SD-kortille, jolta tulostin lukee sen ja tulostaa sen mukaisesti. Ennen tulostusta on tärkeä varmistaa, että tulostusalusta ja tulostuspää ovat puhtaat.

Kitaraohjaimen osien on suunniteltu kiinnittyvän toisiinsa M3-ruuveilla ja -muttereilla. Näitä tarvittiin tässä työssä kumpiakin 23 kappaletta. Muttereita varten 3D-malleihin on tehty kuusikulmaiset syvennykset, joihin mutterit asettuvat ja pysyvät paikallaan ruuveja kiristäessä.





KUVIO 7. Valmis kitaraohjain

Useita 3D-tulostettuja osia joutui hiomaan ja viilaamaan, jotta ne asettuisivat tarpeeksi hyvin paikalleen.

### 5.3 Tarvittavat komponentit

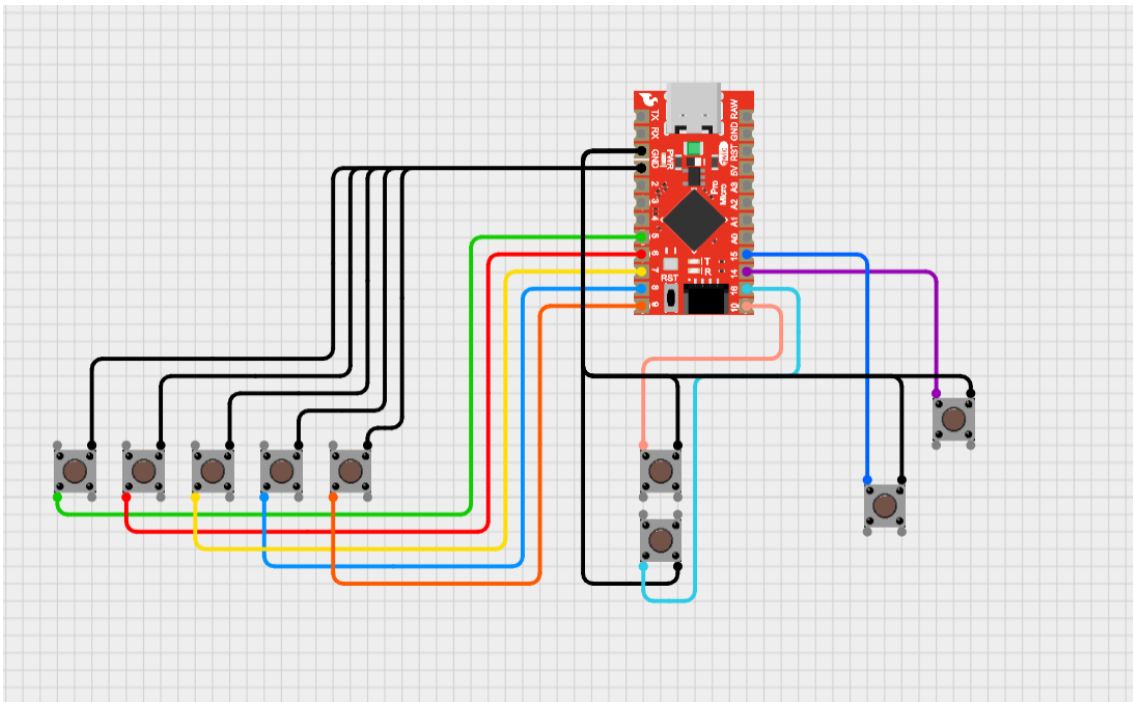
Kitaraohjaimen tarvittavat komponentit listattuna:

- Arduino Pro Micro -tyyppinen mikrokontrolleri (tässä tapauksessa Sparkfun Qwiic Pro Micro – USB-C)
- Kailh Low Profile Choc -näppäimistökytkimiä 9 kappaletta (7 punaista ja 2 valkoista)
- Näppäinhatut kytkimille 5 kappaletta (MBK Low Profile 1.5u)
- Uppokantaisia M3-ruuveja 23 kappaletta (pituus 20 mm ja 25 mm tarpeen mukaan)
- M3-kuusiomuttereita 23 kappaletta
- Kytkentälankaa (tässä tapauksessa vanhoista Ethernet-kaapeleista.)

Sparkfunin valmistama Qwiic Pro Micro USB-C valikoitui ohjaimen mikrokontrolleriksi hyvän saatavuuden ja nykyaikaisen USB-C-liitännän vuoksi. Arduino Pro Microon pohjautuvan mikrokontrollerit ovat myös laajasti käytössä mekaanisissa näppäimistöissä, joten se soveltuu ohjaimen hyvin, sillä ohjaimen on tarkoitus emuloida näppäimistöä.

#### 5.4 Komponenttien juotokset ja kytkentäkaavio

Komponenttien kytkentä (kuva 8) oli yksinkertaista, sillä jokaisessa näppäimistökytkimessä on vain kaksi pinniä, joista toinen kytketään mikrokontrollerissa digitaaliseen datapinniin ja toinen maadoituspinniin. Maadoituspinnejä mikrokontrollerissa on ohjaimen fyysisen rakenteen johdosta käytävissä kaksi, sillä kitaran runko peittää osan mikrokontrollerista. Näppäinten maahan menevät johdot täytyi täten juottaa toisiinsa ennen mikrokontrollerin maadoituspinniin yhdistämistä.



KUVA 8. Komponenttien kytkentä

#### 5.5 Arduinon ohjelmointi

Ohjelmointi tapahtui Arduino IDE:llä. Ohjaimessa käytettävää mikrokontrolleria varten täytyi ladata SparkFunin oma Board Manager -paketti, jotta Arduino IDE tunnistaisi sen oikein. Tämä tapahtui syöttämällä paketin URL-linkki Arduino IDE:n asetuksissa kohtaan "Additional Board Manager

URLs”, jotta ohjelma osaa hakea ja asentaa paketin. Tämän jälkeen ohjelman Boards-valikosta valittiin sinne ilmestynyt SparkFun Pro Micro. Koodin mikrokontrolleriin ajamista varten tarvitsi myös selvittää, mihin COM-porttiin laite on kytketty, ja valita oikea portti Arduino IDE -ohjelmassa.

Koodista tuli hyvinkin yksinkertainen, sillä ohjain käytännössä emuloi näppäimistöä. Mikrokontrollerin suorittimen sisäänrakennetun USB-yhteyden ansiosta Keyboard-kirjastoa käytettäessä tietokone tunnistaa laitteen näppäimistönä ja toimii siten laitteessa kuin laitteessa. Mikrokontrolleri toimii plug & play -periaatteella ja asentaa tarvittavat ajurit automaattisesti laitteen yhdistäessä tietokoneeseen. Kitaraohjaimen Arduino-koodi valmistui yksinkertaisuutensa vuoksi hyvinkin nopeasti ja toimi suoraan ilman suurempia ongelmia.

## 6 OHJAIMEN TOIMINTA JA OHJELMISTO

Kitaraohjain toimii emuloimalla näppäimistöä käyttäen Arduinon Keyboard-kirjastoa (20; 21; 22). Pro Micron ATmega32U4-mikrokontrollerin sisäänrakennetun USB-yhteyden ansiosta mikrokontrolleri pystyy näppäimistöä emuloimalla lähettämään näppäimistöpainalluksia suoraan tietokoneelle, ilman että toisen ohjelman tarvitsisi lukea painalluksia tietokoneen päässä sarjaportilta.

### 6.1 Keyboard-kirjasto

Arduinon Keyboard-kirjasto mahdollistaa 32U4- ja SAMD-pohjaisten mikrokontrollerialustojen näkyvän natiivina näppäimistönä tai hiirenä yhdistetyssä tietokoneessa. Keyboard-kirjasto sisältää seuraavat funktiot:

- Keyboard.begin()
- Keyboard.end()
- Keyboard.press()
- Keyboard.print()
- Keyboard.println()
- Keyboard.release()
- Keyboard.releaseAll()
- Keyboard.write().

Tässä opinnäytetyössä käytettävät funktiot ovat begin-, press- ja release-funktiot.

Keyboard.begin()-funktio aloittaa näppäimistön emuloinnin, jolloin tietokone, johon mikrokontrolleri on yhdistetty, tunnistaa sen näppäimistönä. Funktioon voi halutessaan määrittää myös käytettävän näppäimistöasettelun. Vakiona näppäimistöasettelu on Keyboardlayout\_en\_US. Näppäimistön emulointi jatkuu, kunnes kutsutaan Keyboard.end()-funktioita.

Keyboard.press()-funktio asettaa kutsuttaessa halutun näppäimen painetuksi ja pitää sitä painettuna, kunnes kutsutaan joko Keyboard.release()- tai Keyboard.releaseAll()-funktioita. Haluttu näppäin määritellään funktion parametrinä, jonka datatyypin tulee olla char. Parametrinä voidaan käyttää joko haluttua merkkiä lainausmerkeissä tai merkkiä vastaavaa ASCII-koodia.

Keyboard.release() -funktio vapauttaa halutun näppäimen painetusta tilasta. Kuten press-funkti-  
ossa, haluttu näppäin määritellään char-tyyppisellä parametrilla.

Keyboard.releaseAll() -funktio vapauttaa kaikki painettuna olevat näppäimet.

Keyboard.write() -funktio lähettää halutun char-parametrilla määritellyn näppäinpainalluksen tietokoneelle, eli siinä on käytännössä press- ja release-funktiot samassa.

Keyboard.print() -funktio toimii muuten samalla tavalla, mutta sillä voi lähettää useamman merkin kerralla. Print-funktioon kelpaa parametriksi char- tai string-tyyppinen muuttuja.

## 6.2 Arduino-koodi

Arduino-koodi valmistui nopeasti ja siitä tuli varsin yksinkertainen. Koodissa käytetään Arduinon Keyboard-kirjastoa, joka mahdollistaa näppäimistön näppäinpainallusten lähettämisen tietokoneeseen suoraan ATmega32U4 ja SAMD-pohjaisten mikrokontrollerialustojen USB-portin kautta.

Koodin määrittelyosiossa (kuva 9) lisätään käytettäväksi Keyboard-kirjasto, jonka jälkeen määritellään mikrokontrollerissa käytettävät pinnit sekä näppäinpainallusten jälkeen syötettävä viive. Sen jälkeen määritellään painikkeista, painikkeiden tiloista sekä syötettävistä näppäimistömerkeistä joukot, jotka ovat kaikki samassa järjestyksessä. Käytettävät näppäimistömerkit tässä koodissa on numeronäppäimet 1–5, Enter, Backspace, Right Shift ja Delete. Nämä valikoituivat yksinkertaisesti siksi, että ne olivat jo määritelty itse pelissä käytettäväksi näppäimiksi.

Setup-osiossa (kuva 10) käydään läpi kaikki käytettävät pinnit ja muutetaan niiden tilaksi INPUT\_PULLUP, jolloin pinnit käyttävät mikrokontrollerin sisäänrakennettua vetovastusta ja ne nostetaan oletuksena HIGH-tilaan (23). Tällöin kun ohjaimen näppäintä painetaan, vaihtuu sitä vastaava pinni LOW-tilaan, mitä koodissa käytetään rekisteröimään näppäinpainallus. Setup-osion lopussa kutsutaan Keyboard-luokan Keyboard.begin()-funktiota, joka aloittaa näppäimistön emuloinnin.

Arduino koodin Loop-osiossa (kuva 11) käydään for-loopissa toistuvasti läpi käytettävät pinnit joukosta ja katsotaan, onko pinni HIGH- vai LOW-tilassa. Jos pinnin tila on LOW ja edellinen tila oli HIGH, kutsutaan silloin `Keyboard.press()`-funktioita, joka emuloi näppäimistön näppäimen painamista pohjaan. Tämän jälkeen syötetään aikaisemmin määritelty viive. Jos taas pinnin tila on luetaan HIGH ja edellinen tila oli LOW, kutsutaan funktiota `Keyboard.release()`, joka vapauttaa aiemmin painetun painikkeen. Nämä toiminnallisuudet on toteutettu if-lausekkeilla. Loopin päätteeksi kirjoitetaan pinnin edellistä tilaa kuvaavaan muuttujaan loopin alussa luettu pinnin tila.

```
#include <Keyboard.h>

// Define buttons and corresponding pins
#define BUTTON_COUNT 9

#define PIN_1 5 // Green button, keyboard 1
#define PIN_2 6 // Red button, keyboard 2
#define PIN_3 7 // Yellow button, keyboard 3
#define PIN_4 8 // Blue button, keyboard 4
#define PIN_5 9 // Orange button, keyboard 5

#define PIN_SU 10 // Strum up, keyboard Enter
#define PIN_SD 16 // Strum down, keyboard Backspace
#define PIN_ST 14 // Start button, keyboard Right Shift
#define PIN_SP 15 // Star power, keyboard Delete

#define DEBOUNCE_TIME 20

byte buttons[9] = {PIN_1, PIN_2, PIN_3, PIN_4, PIN_5, PIN_SU, PIN_SD,
PIN_ST, PIN_SP};

// Button state arrays
byte buttonState[BUTTON_COUNT] = {0};
byte buttonStateLast[BUTTON_COUNT] = {1};

// Define an array of keyboard keys
char keyboardKeys[BUTTON_COUNT] = {
  '1', '2', '3', '4', '5', KEY_RETURN, KEY_BACKSPACE, KEY_RIGHT_SHIFT,
KEY_DELETE
};
```

KUVA 9. Koodin määrittelyosio

```

void setup() {
  // Serial for debug
  Serial.begin(115200);
  // Set all buttons pin mode to INPUT
  // and set them to HIGH
  for(int i = 0; i < BUTTON_COUNT; i++) {
    pinMode(buttons[i], INPUT_PULLUP);
  }
  // Begin keyboard emulation
  Keyboard.begin();
}

```

KUVIO 10. Koodin setup-osio

```

void loop() {
  // Loop through pins to read inputs
  for(int i = 0; i < BUTTON_COUNT; i++) {
    buttonState[i] = digitalRead(buttons[i]);
    if(buttonState[i] == 0 && buttonStateLast[i] == 1) {
      Keyboard.press(keyboardKeys[i]);
      delay(DEBOUNCE_TIME);
    }
    if(buttonState[i] == 1 && buttonStateLast[i] == 0) {
      Keyboard.release(keyboardKeys[i]);
      delay(DEBOUNCE_TIME);
    }
    buttonStateLast[i] = buttonState[i];
  }
}

```

KUVIO 11. Koodin loop-osio

### 6.3 Ohjaimen testaus Clone Hero -pelissä

Kitaraohjaimen testaus toteutettiin yhdistämällä ohjain tietokoneeseen ja käyttämällä sitä käyttötarkoituksen mukaisesti peliohjaimena Clone Hero -pelissä. Ohjain toimii pelissä käytännössä täysin samalla tavalla kuin tavallinen näppäimistö, jollaisella peliä myös usein pelataan. Pelin asetuksissa määritellään pelaamiseen käytettävät painikkeet painamalla ohjaimen näppäintä, kun peli pyytää määrittämään tietyille toiminnolle käytettävän painikkeen. Määrittelyn jälkeen ohjain on valmis käytettäväksi pelissä. Ohjainta testattaessa näppäinpainallusten syötössä ei havaittu tuntuvaa viivettä eikä ohjaimen painallukset menneet sekaisin vauhdikkaassakaan pelaamisessa.

## 7 POHDINTA

Työn aiheena oli valmistaa 3D-tulostamalla kitaraohjain käytettäväksi Clone Hero -tietokonepelissä. Yleisesti ottaen työ sujui hyvin. Opinnäytetyölle asetetut tavoitteet saavutettiin onnistuneesti ja työn tuloksena on toimiva ja laadukkaan oloinen 3D-tulostettu kitaraohjain. Kitaraohjaimen osien 3D-tulostus sujui pitkälti ongelmitta eikä tulosteissa tullut mitään varsinaisia virheitä. Kaksi ohjaimen osaa jouduttiin tulostamaan uudelleen tulostusfilamentin loputtua kesken.

Joitain parannuskohteita kuitenkin ilmeni työn edetessä. Otelaudan parannellut mallit olisivat helpottaneet otelaudan näppäinten juottamista, sillä näppäimet olisi voinut kiinnittää otelaudan yläpuoliskoon valmiiksi jo ennen juottamista. Mallit olisivat myös tulostuneet paremmin ja otelauta olisi ollut hieman jäməkampi. Valmistuneessa versiossa tuntuu pientä taipuisuutta.

Myös mikrokontrolleripidike aiheutti haasteita juottamisvaiheessa, sillä mikrokontrolleri täytyi kiinnittää pidikkeeseen ja asettaa paikoilleen ennen juottamista mittaan leikattujen johtojen vuoksi. Kitaran runko myöskin peitti osan mikrokontrollerin pinneistä. Lisäksi mikrokontrolleripidike halkesi, kun siihen kiinnitti ruuveja, ja se täytyi korjata liimalla.

Joitain juotoksia joutui tekemään uusiksi, mikä saattoi myös johtua liittämiseen käytetyistä vanhoista Ethernet-kaapeleiden sisällä olevista monisäikeisistä johdoista. Johdot soveltuivat muuten työhön hyvin pienen kokonsa vuoksi, mutta monisäikeiset johdot olivat hauraita ja niillä oli hieman vaikea saada hyvä juotos.

Seuraavaan versioon ohjaimesta käytettäisiin jo valmiiksi paranneltuja osia sekä suunniteltaisiin ainakin mikrokontrollerin kiinnitys ja sijainti ohjaimen uusiksi.



## LÄHTEET

1. SparkFun. Qwiic Pro Micro USB-C (ATmega32U4) Hookup Guide. Hakupäivä 1.11.2022. <https://learn.sparkfun.com/tutorials/qwiic-pro-micro-usb-c-atmega32u4-hookup-guide>.
2. Atmel 2010. ATmega16U4/32U4 Datasheet. SparkFun. Hakupäivä 15.12.2022. <http://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ATMega32U4.pdf>.
3. Microchip 2022. ATmega32U4. Documentation. Hakupäivä 11.11.2022. <https://www.microchip.com/en-us/product/ATmega32U4>.
4. Wikipedia 2022. Atmel AVR. Hakupäivä 15.12.2022. [https://fi.wikipedia.org/wiki/Atmel\\_AVR](https://fi.wikipedia.org/wiki/Atmel_AVR).
5. Wikipedia 2022. RISC. Hakupäivä 15.12.2022. <https://fi.wikipedia.org/wiki/RISC>.
6. Wikipedia 2022. USB-C. Hakupäivä 11.11.2022. <https://en.wikipedia.org/wiki/USB-C>.
7. Wikipedia 2022. USB. Hakupäivä 15.12.2022. <https://fi.wikipedia.org/wiki/USB>.
8. Perkins, Stephen 2022. Mechanical keyboard switches: An in-depth guide. Android Police. Hakupäivä 16.11.2022. <https://www.androidpolice.com/mechanical-keyboard-switches-types-explainer/>.
9. Wikipedia 2022. Kolmiulotteinen tulostus. Hakupäivä 15.12.2022. [https://fi.wikipedia.org/wiki/Kolmiulotteinen\\_tulostus](https://fi.wikipedia.org/wiki/Kolmiulotteinen_tulostus).
10. Luovina 2019. Mikä on 3D-tulostin ja kuinka se toimii? Hakupäivä 28.11.2022. <https://luovina.fi/mika-on-3d-tulostin-ja-kuinka-se-toimii/>.
11. Prusa 2021. Prusa Slicer – General Info. Prusa Research. Hakupäivä 15.12.2022. [https://help.prusa3d.com/article/general-info\\_1910](https://help.prusa3d.com/article/general-info_1910).

12. Ekaran, Sammy 2019. What Is a G-Code File and How Can You Prepare One? MUO. Hakupäivä 28.11.2022. <https://www.makeuseof.com/what-is-g-code-file-how-can-you-prepare-one/>.
13. Wikipedia 2022. Blender (ohjelmisto). Hakupäivä 15.12.2022. [https://fi.wikipedia.org/wiki/Blender\\_\(ohjelmisto\)](https://fi.wikipedia.org/wiki/Blender_(ohjelmisto)).
14. Stubbs, Mike 2018. The spirit of Guitar Hero lives on in a bizarre community-made clone. Eurogamer. Hakupäivä 11.11.2022. <https://www.eurogamer.net/meet-the-community-reviving-classic-guitar-hero-gameplay-in-their-own-clone>.
15. Wikipedia 2022. Clone Hero. Hakupäivä 11.11.2022. [https://en.wikipedia.org/wiki/Clone\\_Hero](https://en.wikipedia.org/wiki/Clone_Hero).
16. Clone Hero 2022. Clone Hero v1 is out! Hakupäivä 15.12.2022. <https://clone-hero.net/2022/11/29/v1-release.html>.
17. Wikipedia 2022. Guitar Hero. Hakupäivä 16.11.2022. [https://en.wikipedia.org/wiki/Guitar\\_Hero](https://en.wikipedia.org/wiki/Guitar_Hero).
18. Josh Designs 2022. Les Paul Clone Hero Controller. Printables. Hakupäivä 11.11.2022. <https://www.printables.com/model/227773-les-paul-clone-hero-controller>.
19. Creative Commons 2022. Attribution 4.0 International (CC BY 4.0). Hakupäivä 15.12.2022. <https://creativecommons.org/licenses/by/4.0/>.
20. Arduino. Keyboard-referenssisivu. Hakupäivä 10.12.2022. <https://www.arduino.cc/reference/en/language/functions/usb/keyboard/>.
21. Arduino-Libraries 2022. Keyboard Library for Arduino. Github. Hakupäivä 10.12.2022. <https://github.com/arduino-libraries/Keyboard>.
22. Rcb5126 2020. Arduino\_Keyboard. Github. Hakupäivä 10.12.2022. [https://github.com/rcb5126/Arduino\\_Keyboard](https://github.com/rcb5126/Arduino_Keyboard).

23. The Robotics Backend 2022. Arduino INPUT\_PULLUP Explained (pinMode). Hakupäivä 10.12.2022. [https://roboticsbackend.com/arduino-input\\_pullup-pinmode/](https://roboticsbackend.com/arduino-input_pullup-pinmode/).
24. Söderby, Karl 2022. Keyboard Presses with Arduino Micro. Arduino Docs. Hakupäivä 10.12.2022. <https://docs.arduino.cc/tutorials/micro/keyboard-press>.