

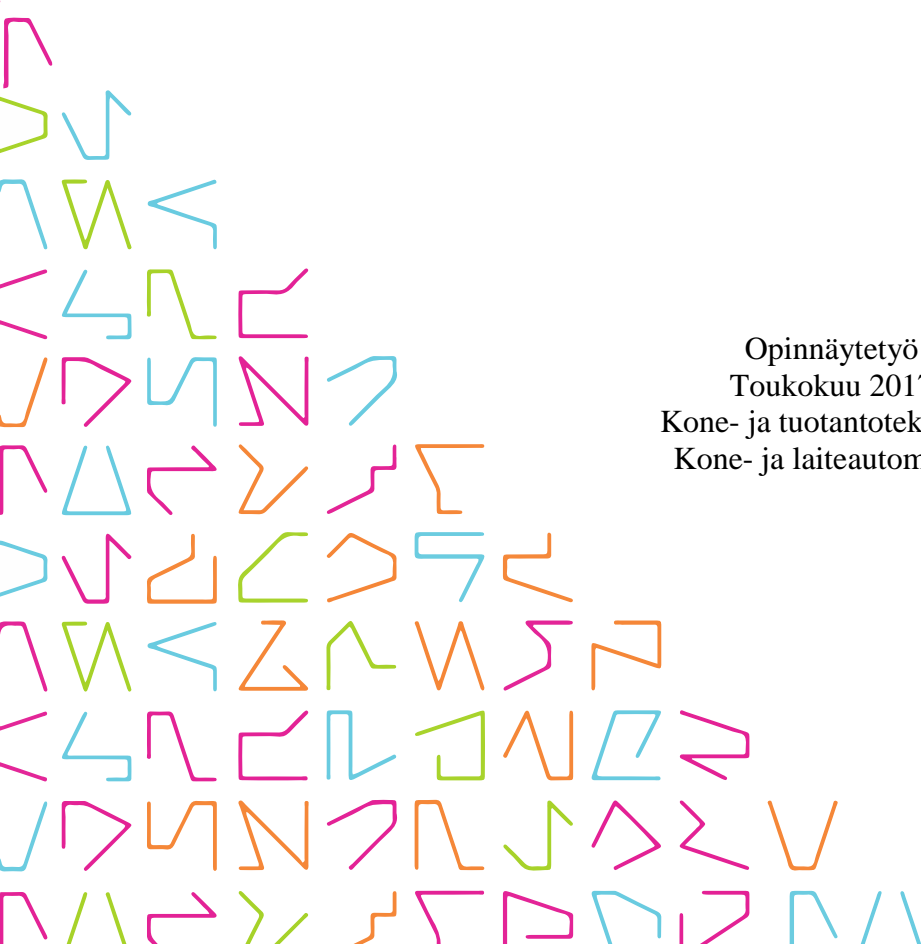


TAMPEREEN
AMMATTIKORKEAKOULU

KOPIOSORVIN SUUNNITTELU

Harri Salminen

Opinnäytetyö
Toukokuu 2017
Kone- ja tuotantotekniikka
Kone- ja laiteautomaatio



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Kone- ja tuotantotekniikka
Kone- ja laiteautomaatio

SALMINEN, HARRI:
Kopiosorvin suunnittelu

Opinnäytetyö 61 sivua, joista liitteitä 7 sivua
Toukokuu 2017

Tämän opinnäytetyön tavoitteena oli vuoden 2015 syksyllä opiskelijaprojektina alkaneen kopiosorvin suunnittelu, ohjausjärjestelmän kehittäminen sekä mekaanisen mallin luominen. Tarkoituksena oli valmistaa prototyyppi, jolla voitaisiin valmistaa kopioimalla kalastusvaappuja sekä muita pieniä puisia epäsymmetrisiä pyörähdyskappaleita. Kopiosorviprojektin suunnittelun ohjenuorana toimi vaatimusluettelo, jossa listattiin tärkeimmät saavutettavat toiminnalliset tavoitteet.

Työn toteutus aloitettiin olemassa olevia laitteita tutkimalla. Näiden pohjalta valittiin komponentit sekä runkomateriaalit, joiden uskottiin täyttävän laitteelle asetetut vaatimukset. Kopiosorvin suunnittelussa ja komponenttivalinnoissa haluttiin hyödyntää mahdollisimman paljon automaatiota. Tärkeimpänä komponenttina toimi Raspberry Pi -mikrotietokone, jolle luotiin Python-ohjelmointikielelle ohjausjärjestelmä valituille sähköisille komponenteille.

Työn tuloksia ei ole vielä nähtävissä, sillä kopiosorvin rakennus on edelleen kesken, mutta työn aikana on opittu paljon uusia asioita ohjelmoinnista sekä projektinhallinnasta. Työssä ei esitellä toteutettuja ohjelmointiratkaisuja, sillä ne halutaan pitää salassa mahdollista tuotemarkkinointia varten.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Mechanical and Production Engineering
Machine Automation

SALMINEN, HARRI:
Design of a Copy Lathe

Bachelor's thesis 61 pages, appendices 7 pages
May 2017

The aim of this thesis was to design a copy lathe, along with developing its control system and creating its mechanical model. The copy lathe project started out as a student project in fall 2015. The idea was to create a prototype that could produce fishing wobblers and other small asymmetrical wooden objects by using a copy of the original design. A list of functions required to achieve the desired goal was used as guidelines for designing the copy lathe.

The execution of the thesis started with inspecting the existing machines. Based on these machines, components and chassis materials that were believed to meet the machine requirements were selected. The design and the component selection for the lathe depended very much on its automation capabilities. The most important component was a micro-computer called Raspberry Pi, for which the control system for the selected electrical components was created.

The results of the thesis cannot be seen yet, because the building of the copy lathe is still under way, but in the process of the thesis a lot of new things have been learned about programming and project management. The programming solutions are not included in this thesis.

Key words: copy lathe, design, control system

SISÄLLYS

1	JOHDANTO.....	6
2	SORVAUS TYÖSTÖMENETELMÄNÄ.....	7
3	MEKATRONISET JÄRJESTELMÄT	8
	3.1 Tuotekehitys mekatroniikan näkökulmasta	9
	3.2 Mikrokontrollerit	10
	3.2.1 Arduino	12
	3.2.2 Raspberry Pi	13
	3.2.3 Raspberry Pin ohjelmointi	14
	3.3 Anturointi.....	17
	3.4 Harjattoman tasavirtamoottorin ohjaus.....	20
	3.5 Askelmoottorin ohjaus	23
4	KOPIOSORVIPROJEKTIN HISTORIA.....	26
	4.1 Alkuluonnostelu ja olemassa olevien tuotteiden tarkastelu	26
	4.2 Kopiointimenetelmän valinta.....	28
5	KOMPONENTTIEN VALINTA	30
	5.1 Sorvaustyökalu.....	31
	5.2 Moottorit ja niiden ohjaus	32
	5.3 Laseranturi	34
	5.4 Mikrotietokone Raspberry Pi sekä MCP3008-A/D-muunnin	36
	5.5 Käyttöliittymä	38
6	KOMPONENTTIEN TESTAUS JA OHJELMOINTI	40
	6.1 Moottoreiden testaus	40
	6.2 Laseranturin testaus ja MCP3008-mikrosirun ohjelmointi.....	42
	6.3 Anturidatan tallennus ja uudelleenkäyttö	45
7	TULOKSET JA JATKOSUUNNITELMAT	47
	LÄHTEET.....	53
	LIITTEET	55
	Liite 1. Vaatimusluettelo	55
	Liite 2. Kopiosorvin 3D-malli	56
	Liite 3. Laserleikkeiden valmistuspiirustukset.....	57
	Liite 4. Kopiosorvin komponenttien johdotusten Fritzing-kuva	61

LYHENTEET JA TERMIT

ADC	Analogia-digitaalimuunnin (eng. Analog-to-Digital Converter)
CAD	Tietokoneen avulla tapahtuva suunnittelu (eng. Computer-aided Design)
CLK	Kellotaajuus (eng. Clock frequency)
CS/CE	Sirun valinta/aktivointi (eng. Chip Select/Chip Enable)
DC	Tasavirta (eng. Direct Current)
GPIO	Yleis-I/O-liitäntä (eng. General Purpose Input/Output)
MISO	Mikrokontrolleri-isännän sisäänmeno, orjan ulostulo (eng. Master In, Slave Out), käytetään kahden mikrokontrollerin välisessä sarjakytkennässä
MOSI	Mikrokontrolleri-isännän ulostulo, orjan sisäänmeno (eng. Master Out, Slave In), käytetään kahden mikrokontrollerin välisessä sarjakytkennässä
PWM	Pulssinleveysmodulaatio (eng. Pulse-Width Modulation)
RPM	Kierrosnopeuden yksikkö (eng. Rounds Per Minute)

1 JOHDANTO

Tämä opinnäytetyö perustuu vuoden 2016 keväällä alkaneeseen kahden hengen opiskelijaprojektiin, jonka lopputuloksena oli tarkoitus rakentaa kopiosorvi kotikäyttöön. Projekti lähti liikkeelle, kun ilmeni tarvetta epäsymmetrisiä pyörähdyskappaleita kopioivan laitteen kehittämiseen käsityönä tapahtuneen tuotantoprosessin tilalle.

Suunnittelussa lähdettiin liikkeelle jo olemassa olevien koneiden ominaisuuksien ja toiminnan tutkimisesta. Näiden pohjalta haluttiin luoda uusi laite, joka antaisi mahdollisuuden hyödyntää kone- ja laiteautomaation suuntautumisessa saatuja oppeja mahdollisimman kustannustehokkaaseen hintaan. Valmistuksessa haluttiin turvautua mahdollisimman vähän koneistettaviin komponentteihin, jotta kokoonpano olisi vaivatonta ja vaatisi mahdollisimman vähän erikoisosaamista.

Työn tavoitteena oli toimivan rungon ja mekaniikan lisäksi myös itseohjelmoitava ohjausjärjestelmä, jonka toiminta olisi loogista ja helposti ymmärrettävää mahdollisten viikatilanteiden varalta. Koneen hallinnointiin haluttiin myös graafinen käyttöliittymä, joka tuli valitun mikrokontrollerin mukana valmiina.

Työn tuloksia ei ole vielä täysin nähtävillä, sillä koneen runko ja mekaniikka odottavat vielä kokoonpanoa. Koneen ohjausjärjestelmän ohjelmointi on lähes valmis, mutta siitä ei esitetä tässä opinnäytetyössä kuin pätkiä, sillä projektin tuloksena syntyvän koneen piensarjatuotantoa ja myyntiä harkitaan.

2 SORVAUS TYÖSTÖMENETELMÄNÄ

Sorvaus on yleinen lastuavan työstön menetelmä, jossa pituusakselinsa ympäri pyörivästä kappaleesta irrotetaan työkalulla eli sorvinterällä lastuja. Tässä työtavassa sorvinterä tekee syöttöliikkeen. (Ihalainen, Aaltonen, Aromäki & Sihvonen 2011, 150.) Sorvaus päätyi työtapana opiskelijaprojektin koneen työtavaksi yleisyytensä ja suhteellisen helpon toteuttamistapansa takia. Lisäksi useimmat kopiointiin tarkoitetut vertailulaitteet, joita tutkittiin suunnittelun alkuvaiheessa, toimivat juuri sorveina.

Sorvinterän liikeradan perusteella voidaan eri sorvaustyypeiksi määritellä

- lieriösorvaus
- tasosorvaus
- muotosorvaus
- poraus- ja sisäsorvaus
- kartiosorvaus (Ihalainen ym. 2011, 150).

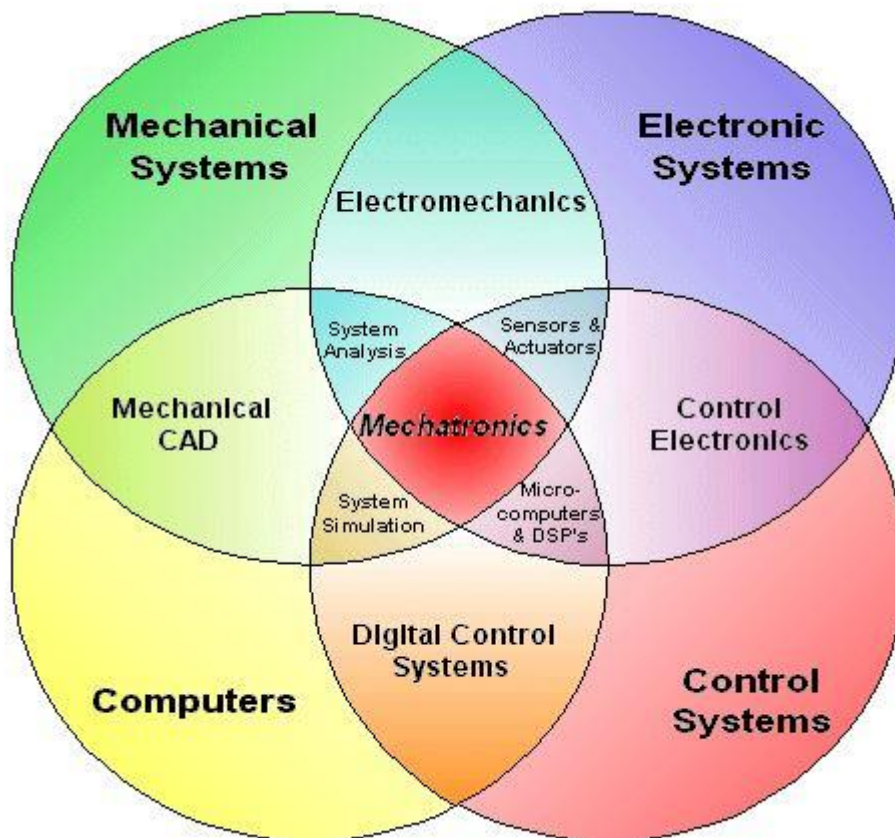
Tässä opinnäytetyössä lieriösorvaus oli lähimpänä laitteen sorvaustyyliä, tosin sorvissa kiinteän terän sijaan työstö hoidetaan pienellä käsijyrsimellä (kuva 1), jonka pyörimisnopeutta voitiin säätää materiaalin mukaan. Kaikissa laitteissa, joita muut ihmiset olivat rakentaneet, sorvaustyöstö tapahtui juuri pienen jyrsimen avulla.



Kuva 1. Laitteessa käytettävä työkalu. Työstöterä tulee kiinni moottorin päässä olevaan ”ruuviin”.

3 MEKATRONISET JÄRJESTELMÄT

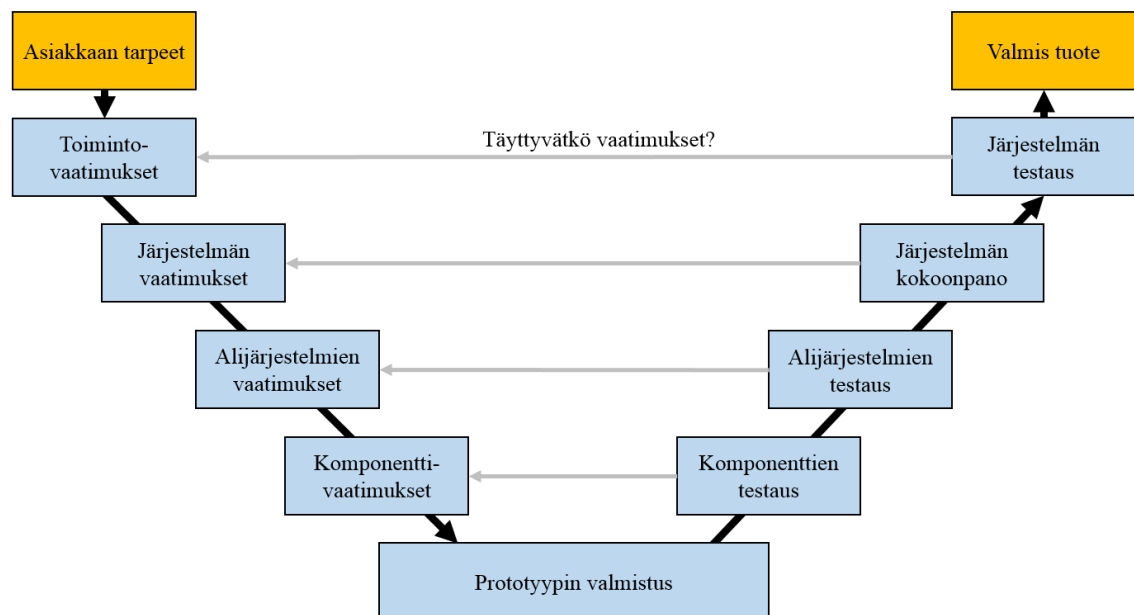
Mekatroniikka sanana tuli ensimmäisen kerran käyttöön jo 1960-luvulla Japanissa ja sillä tarkoitetaan tekniikan osa-aluetta, joka yhdistelee elektroniikka, automaatiota ja tietotekniikkaa perinteiseen koneenrakennukseen (kuva 2). Mekatroniikassa siis tuodaan jonkinlaista älyä osaksi mekaanista tuotetta. Hyviä esimerkkejä mekatroniikan hyödyntämisestä arkisemmassakin elämässä ovat nykyajan autot, kopionkoneet, juoma-automaatit tai robotit, joissa mikroprosessoripohjainen ohjausyksikkö ohjailee koneen sisällä olevia toimilaitteita, kuten servomoottoreita, tekemään haluttuja fyysisiä toimintoja laitteen ohjelmoinnin mukaisesti. (NC State University 2015.) Nyt käsitellään lyhyesti mekatronisen järjestelmän suunnitteluprosessia.



Kuva 2. Mekatroniikka yhdistää monia tekniikan osa-alueita (NC State University 201

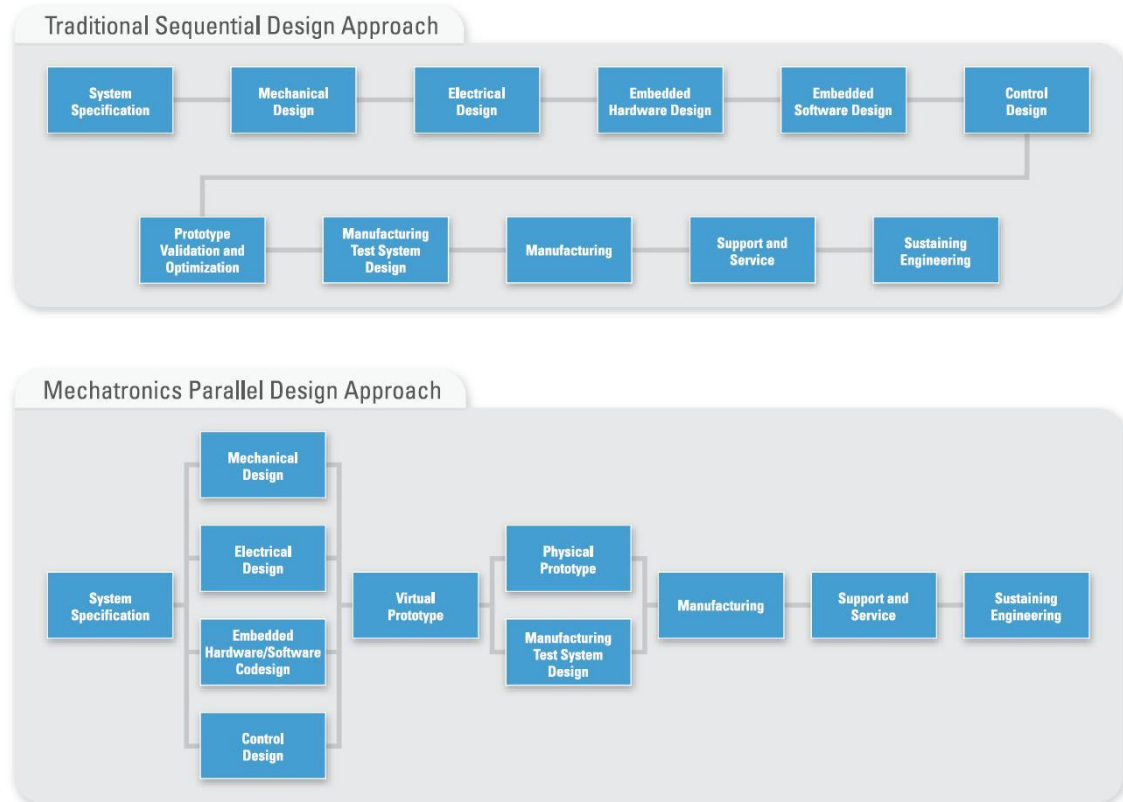
3.1 Tuotekehitys mekatroniikan näkökulmasta

Kun suunnitellaan uutta mekatronista laitetta, olisi hyvä käyttää mekatronista tuotekehitysmallia, joka pääpiirteissään on kuvattu kuvassa 3. Suunnittelu lähtee liikkeelle asiakkaan tarpeesta, niin myös tämän opinnäytetyön kohdalla. Asiakas ilmaisee vaatimuksensa laitteen toiminnan, hinnan, koon ja muiden ulkoisten piirteiden suhteen, ja niiden pohjalta aletaan suunnitella laitteen järjestelmää, joka edelleen jaetaan pienempiin alijärjestelmiksi ja niiden komponenteiksi. Alijärjestelmät voivat olla eri tekniikan osa-alueilta, ja niiden yhdisteleminen toimiviksi järjestelmiksi voi olla haastavaa erilaisten suunnitteluperiaatteiden ja -ohjelmistojen takia. (Künzel 2005.) Tämä vaatii hyvää kokonaisuuden hallintaa, sillä kunkin komponentin ja alijärjestelmän tulee myös täyttää järjestelmälle ja yleiselle toiminnalle asetetut vaatimukset.



Kuva 3. Mekatronisen tuotekehitysprosessin kuvaus

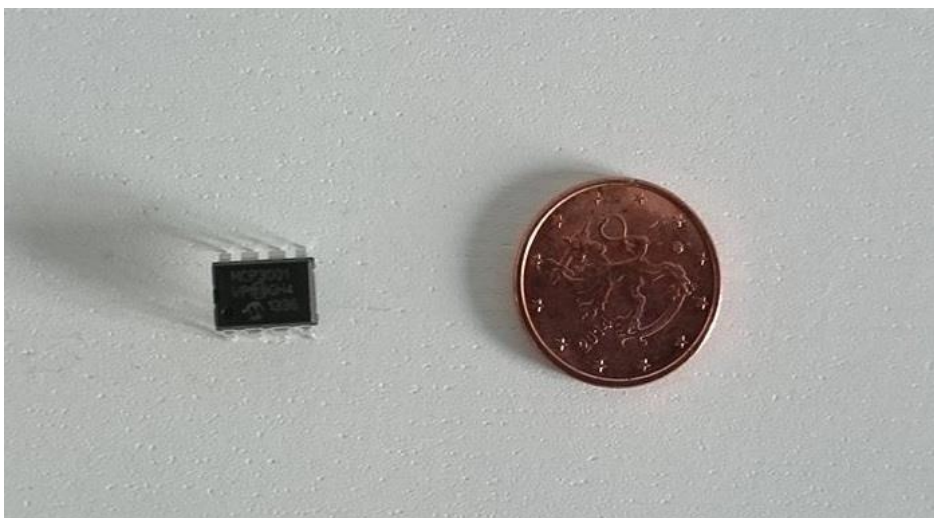
Verrattaessa perinteiseen tuotekehitysprosessiin, mekatronisessa tuotekehitysprosessissa syntyy monimutkaisempiakin tuotteita lyhyessä ajassa, koska samaan aikaan voidaan suunnitella ja simuloida useampia alijärjestelmiä (Künzel 2005). Tämä suunnitteluprosessien ero käy ilmi kuvasta 4. Perinteisessä mallissa kaikki osa-alueet tehdään järjestyksessä alusta loppuun ennen kuin voidaan aloittaa seuraava osio, koska seuraava osa-alue vaatii edellisen osan ratkaisujen olemassaoloa. Mekatronisessa mallissa voidaan simuloida ja suunnitteluohjelmistoja hyödyntämällä joitakin osioita suunnitella samanaikaisesti, mikä lyhentää suunnittelu-aikaa.



Kuva 4. Perinteisen ja mekatronisen tuotekehitysprosessin ero (Mathur 2012)

3.2 Mikrokontrollerit

Mikrokontrollerit ovat nimensä mukaisesti pieniä ohjaimia, joiden koko vaihtelee kolikkoa pienemmästä (kuva 5) hieman luottokorttia suurempaan. Niitä käytetään erilaisissa elektronisissa ja mekatronisissa sovelluksissa.



Kuva 5. Microchip Technology Inc.:n valmistama MCP3001-A/D-muunnin ja viiden sentin kolikko.

Mikrokontrolleri yhdistää mikroprosessorin kaikkeen tarpeelliseen, jotta kontrolleri voi toimia yhden piirilevyn tietokoneena. Mikrokontrolleri on ohjelmitavissa suorittamaan tiettyä toimintoa tai ohjelmaa. Mikrokontrollereita on useita tuhansia erilaisia, ja ne palvelevat jokainen omaa käyttötarkoitustaan. Mikrokontrollerit koostuvat ainakin seuraavista osista:

- CPU (Central Processing Unit) eli suoritin, joka suorittaa mikrokontrollerille ohjelmoidun ohjelman kertomia käskyjä
- ROM-muisti (Read-Only Memory) tai lukumuisti, jota prosessori hyödyntää ohjelman suorittamisessa
- RAM-muisti (Random Access Memory) eli keskusmuisti, johon tallennetaan data suorittimelta ohjelman suorittamisen ajaksi
- I/O-liitännät, joiden avulla mikrokontrolleri viestii ympäristönsä kanssa binäärimuodossa
- ajastinpiiri (Clock Source) suorittimen kellotaajuuden määrittämiseen. (Carryer, Ohline & Kenny 2011.)

Peruskomponenttiansa lisäksi mikrokontrollerissa voi olla myös lisäominaisuuksia, kuten

- Sarjaportti ulkoisten laitteiden kanssa kommunikointia varten, esimerkiksi USB (Universal Serial Bus)
- A/D-muunnin eli signaalimuunnos analogisesta voltti- tai ampeerisignaalista digitaaliseksi binääritiedoksi
- PWM- eli pulssinleveysmodulaatio, joka varsinkin yhdessä A/D-muuntimen kanssa on todella hyödyllinen ominaisuus
- LED-valoja tai LED-näyttö.

Lisäkomponenteilla mikrokontrollerin ominaisuuksia voidaan kasvattaa käyttötarpeen mukaan. Markkinoilla on myös mikrokontrollereita, joiden varustelutaso riittää sellaiseen useimpiin käyttötarkoituksiin. Tällaisia ovat esimerkiksi Arduinon nimeä kantavat erilaiset kehitysalustat tai Raspberry Pi,

Mikrokontrollereita löytyy nykypäivänä joka paikasta, ja joissain sovelluksissa, kuten autoissa, on useampi mikrokontrolleri hoitamaan eri tehtäviä, kuten vilkkujen toimintaa tai bensansyöttöä.

3.2.1 Arduino

Arduino on avoimen lähdekoodin mikrokontrolleripohjainen kehitysalusta, joka pohjautuu helposti käytettävään ohjelmistoon ja laitteistoon. Se voi kommunikoida muiden laitteiden kanssa analogisesti ja digitaalisesti tai pulssinleveysmodulaatiota hyödyntäen. Siitä löytyy myös sarjaliitännämahdollisuus niin USB-portin kuin SPI (Serial Peripheral Interface) -kytkennän muodossa.

Arduinoa ohjelmoidaan sen oman C-kieleen pohjautuvan ohjelmointikielen avulla käyttäen Arduino IDE (Integrated Development Environment) -ohjelmointiympäristöä. IDE:n helppokäyttöisyyden ja Arduinon nettisivuilta löytyvien helppotajuisten ohjeiden takia kokematonkin käyttäjä kykenee tuossa tuokiossa kirjoittamaan omaa koodiaan ja luomaan omia ohjelmia projekteihinsa. Arduino on kuitenkin sen verran moneen taipuva alusta, että myös kokeneet, vaativammat käyttäjät ovat siihen tyytyväisiä. Helppokäyttöisyytensä ja joustavuutensa ansiosta Arduinosta on tullut erittäin suosittu kehitysalusta varsinkin koulutusympäristössä, minkä lisäksi sen hinta on halpa, joten kaikilla on Arduinon kautta mahdollisuus tutustua ohjelmointiin ja omien sovellusten kehittämiseen. Arduinon suosiota lisää myös se, että se toimii kaikilla käyttöjärjestelmillä, eli Windowsilla, Linuxilla ja Macintosh OSX:llä. (Arduino 2017, What is Arduino?)

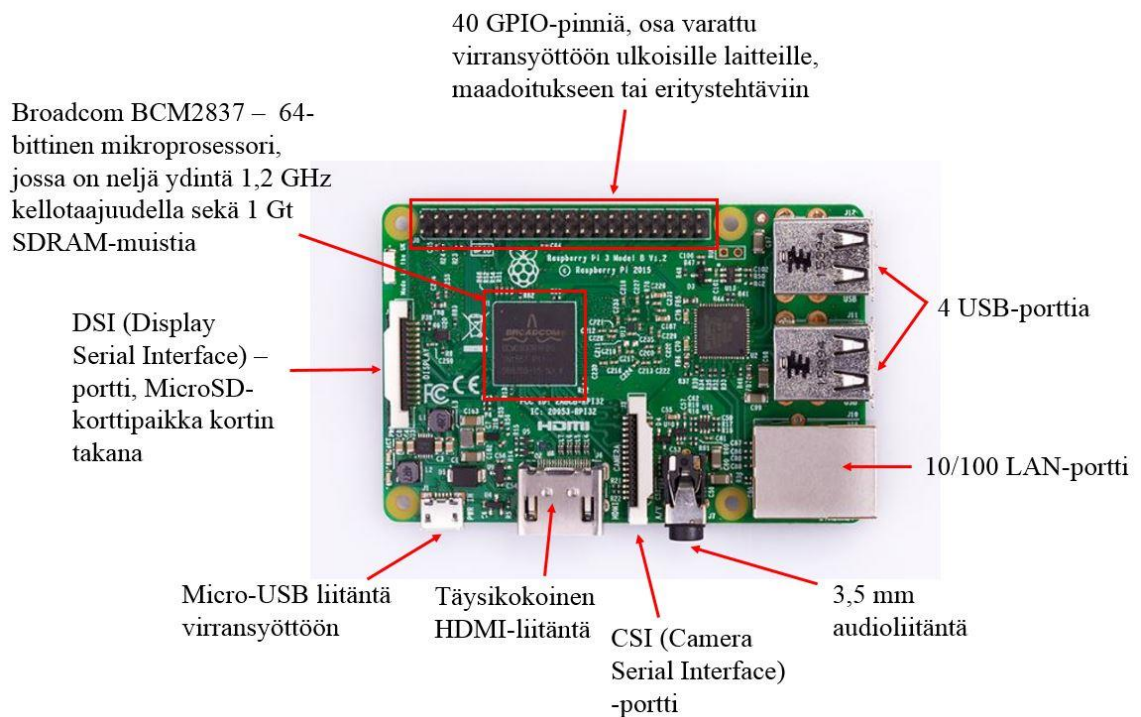
Arduinon mikrokontrollereita on useita eri malleja, mutta tässä opinnäytetyön aiheena olevassa projektissa hyödynnetään Arduino UNO -mikrokontrolleria (kuva 6). Arduino UNOn sydämenä toimii ATmega328-mikrokontrolleri, jonka Grbl-niminen avoimen lähdekoodin CNC-ohjelmisto vaatii toimiakseen. Grbl:stä kerrotaan lisää tämän opinnäytetyön luvussa 5.2.1.



Kuva 6. Arduino UNO -mikrokontrolleri (Arduino 2017. Arduino UNO)

3.2.2 Raspberry Pi

Raspberry Pi (kuva 7) ei oikeastaan ole vain mikrokontrolleri, se on pieni, hieman luotokorttia isompi tietokone. Se luotiin, kun haluttiin edullinen laite, jolla kyettäisiin kehittämään ohjelmointitaitoja ja mikrokontrollerien laitteiston ymmärrystä muuallakin kuin yliopistotasolla. Sen pieni koko ja hinta kuitenkin houkuttelivat sen käyttäjiksi myös kaikki elektroniikkaintoilijat ja yksityiset tuotekehittelijät, koska Raspberry Pillä saattoi tehdä projekteja, joihin tavallinen mikrokontrolleri ei kykene. Laitteen laitteisto on avoin omalle kehitystyölle lukuun ottamatta laitteen sydämenä toimivaa Broadcom BCM2837 -mikroprosessoria (Red Hat 2017.)



Kuva 7. Raspberry Pi Model 3 B -mikrotietokone (Raspberry Pi Foundation 2016)

Raspberry Pi on Linux-pohjainen tietokone, johon täytyy itse ladata käyttöjärjestelmä. Käyttöjärjestelmävaihtoehtoja on useita, mutta niistä Raspbian lienee suosituin. Se pohjautuu Debian-käyttöjärjestelmään. Toinen suosittu vaihtoehto on Pidora, joka taas pohjautuu Fedora-käyttöjärjestelmään. (Red Hat 2017.) Lataaminen käy melko vaivattomasti, jos vain omistaa Raspberry Pin tukeman MicroSD-muistikortin. Raspberry Pin nettisivuilla, osoitteessa www.raspberrypi.org, suositellaan aloittelijoille käyttöjärjestelmän asentamisessa nk. NOOBS -vaihtoehtoa (New Out Of the Box Software), joka siis on

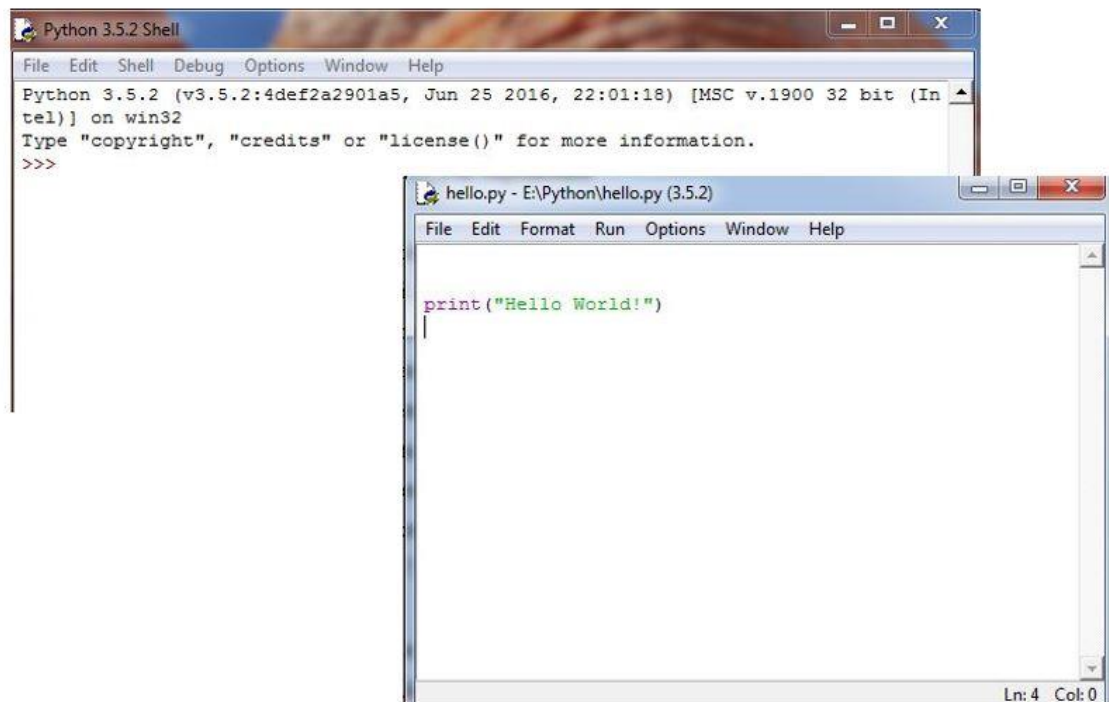
jälleenmyyjien myymä muistikortti, johon on valmiiksi asennettu Raspbian-käyttöjärjestelmä. Nettisivuilla on kuitenkin ohjeet myös manuaaliseen Raspbianin, kuten myös muiden käyttöjärjestelmävaihtoehtojen asennukseen.

Kun käyttöjärjestelmä on asennettu, voidaan Raspberry Piin kytkeä näyttö HDMI-kaapelilla ja näppäimistö ja hiiri USB-portteihin, ja alkuasetusten jälkeen Raspberry Pi toimii kuten mikä tahansa tietokone: sillä voi pelata, selata Internetiä, ohjelmoida tai luoda omia tekstidokumentteja. Laitteen suorituskyky on kuitenkin rajallinen, ja useamman ohjelman samanaikainen suorittaminen hidastaa laitetta merkittävästi. Tämä tuli ilmi opinnäytetyön aikana, kun Internet-selaimen aukeaminen kesti todella kauan useamman koodinpätkän ollessa auki työpöydällä.

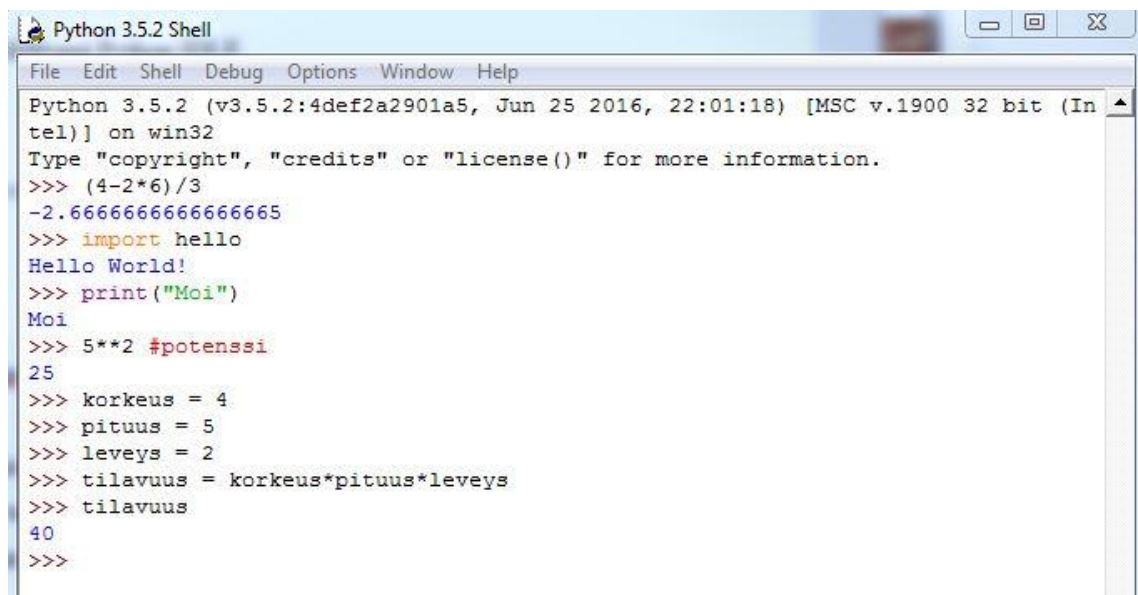
3.2.3 Raspberry Pin ohjelmointi

Raspberry Pin ohjelmointikieli on Python. Python on helppotajuinen, mutta täysiverinen ohjelmointikieli, joka on yleisesti käytössä Linux-ympäristössä. Sen suosio ohjelmointikielenä kasvoi paljon Raspberry Pin markkinoille tulon myötä. Se sopii niin aloittelijoille kuin erittäin vaativaan käyttöön sisältämänsä datatyypikirjon vuoksi. Python on tulkittava kieli: Python-kielellä kirjoitettu ohjelma pitää koota ja kääntää bittiedoksi, ennen kuin kone ymmärtää suorittaa ohjelmaa. Raspberry Pi:ssä kääntäjä tulee ainakin Raspbian-käyttöjärjestelmän mukana, joten ohjelmaa ei tarvitse erikseen ladata tietokoneelta mikrokontrollerille, kuten Arduinin tapauksessa. Tämä nopeuttaa huomattavasti alkuvaiheen testailua, kun ohjelmaa voi suorittaa kerta toisensa perään muutosten jälkeen reaaliajassa. (Python Software Foundation 2017, Whetting your appetite.)

Python-ohjelmia voidaan suorittaa suoraan niiden sijainnista, jos niistä on tehty suoritettavia (executable). Ohjelmia voidaan kirjoittaa tai muokata tekstieditorissa, kuten Notepad ++:ssa tai Pythonin omalla IDLE:llä (Integrated Development and Learning Environment), jonka käyttöliittymää on kuvattu kuvassa 8. Koska Python on valmiiksi tulkittu ohjelmointikieli, IDLE:tä voi käyttää vaikka laskimena. Sen kautta voi tulostaa tekstiä näytölle tai suorittaa ohjelmia. Kuvassa 9 on kuvattu näitä mahdollisuuksia.



Kuva 8. Pythonin käyttöliittymä Python IDLE



Kuva 9. IDLE:n käyttöä eri tarkoituksissa

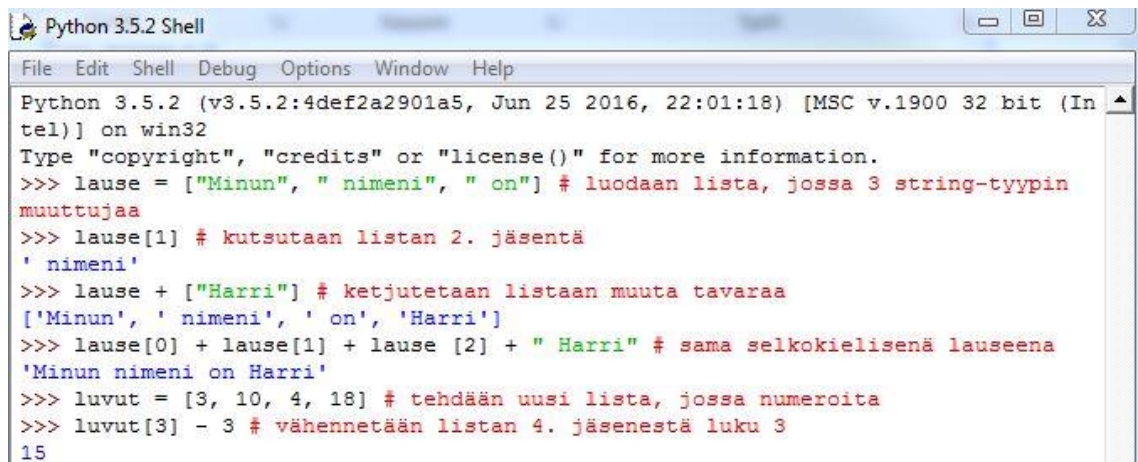
Pythonissa on useampi erilainen muuttujatyyppi. Muuttuja on nimitys erilaisille datatyypeille, joita näiden muuttujien nimissä voi tallentaa ja käyttää uudelleen ohjelman sisällä.

Näitä muuttujatyyppejä ovat ainakin:

- Int eli Integer, joka tallentaa kokonaislukuja
- Float, johon tallennetaan desimaalilukuja
- Boolean, johon tallennetaan totuusarvoja tosi tai epätosi (myös 1 ja 0)

- String, joka tallentaa kirjainjonoja, kuten ”Hyvää päivää”
- List, johon tallennetaan dataa tietyssä järjestyksessä. (The Python Book 2016, 10)

Int, float, boolean ja string katsotaan datan perustyypeiksi, list taas on yhdistetätyyppi. Lista kirjoitetaan pilkulla erotettujen arvojen listana, ja se voi sisältää erilaisia perusdatatyyppisiä, tosin yleensä yhdessä listassa käytetään vain yhtä datatyyppiä. Listan arvoja voidaan muuttaa, ja listan jäsenille voidaan tehdä laskutoimituksia, jos listan arvot ovat numeroita. (Python Software Foundation 2017, An Informal Introduction to Python.) Kuvassa 10 on esitelty erilaisia listan käyttömahdollisuuksia.



```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> lause = ["Minun", " nimeni", " on"] # luodaan lista, jossa 3 string-tyypin muuttujaa
>>> lause[1] # kutsutaan listan 2. jäsentä
' nimeni'
>>> lause + ["Harri"] # ketjutetaan listaan muuta tavaraa
['Minun', ' nimeni', ' on', 'Harri']
>>> lause[0] + lause[1] + lause [2] + " Harri" # sama selkokieლისენä lauseena
'Minun nimeni on Harri'
>>> luvut = [3, 10, 4, 18] # tehdään uusi lista, jossa numeroita
>>> luvut[3] - 3 # vähennetään listan 4. jäsenestä luku 3
15

```

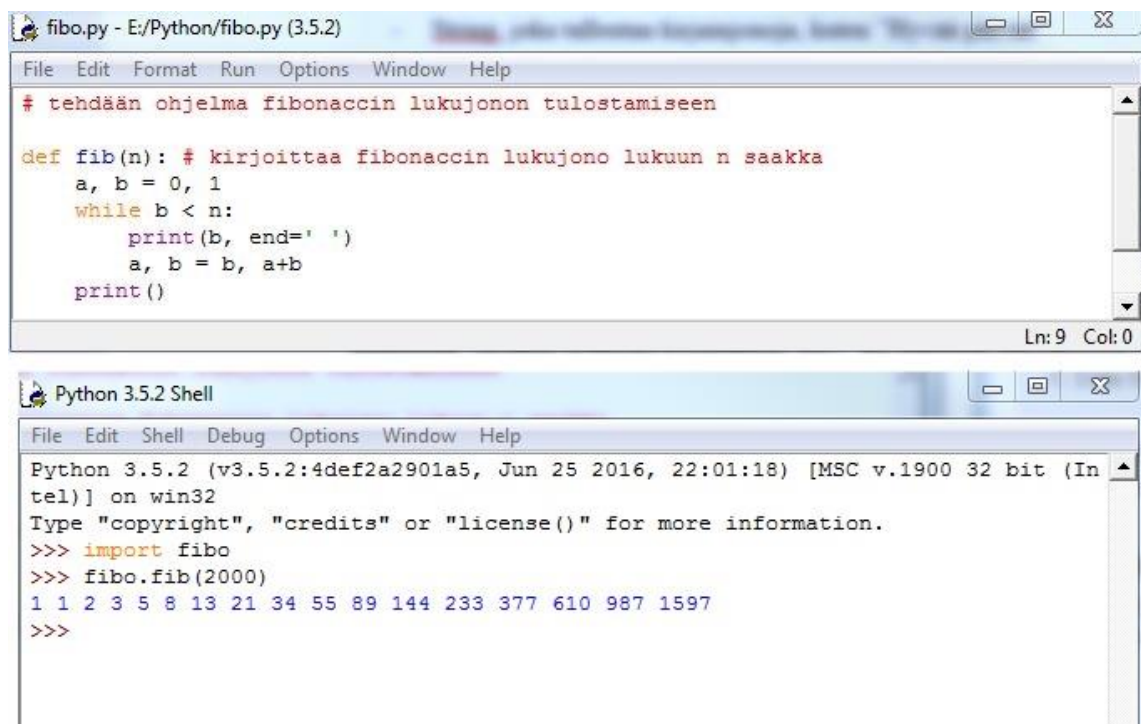
Kuva 10. Erilaisia listan käyttömahdollisuuksia

Python-ohjelmointikielessä on C-kielen tapaan erilaisia ohjausrakenteita, joilla voidaan vaikuttaa ohjelman kulkuun. Näitä rakenteita ovat *if*, *for* ja *while*. Niillä jokaisella on omat käyttötarkoituksensa, joita voisi kuvailla vaikka näin:

- *If*-rakenne on yleisin ohjausrakenne. Sillä voidaan kirjoittaa ehtoja ”jos ehto 1 toteutuu, tee näin”. *If*-rakenteessa voidaan antaa myös vaihtoehtoisia ohjelman etenemisreittejä *elif*- ja *else*-komennoilla.
- *For*-rakenne on hyödyllinen, kun tiettyä silmukkaa halutaan suorittaa jonkin määrätyn ajan, ”tee tätä niin kauan kunnes”. Vaikkapa *for* ($x < 10$) suorittaa silmukkaa niin kauan, kuin integer-muuttuja x :n arvo on alle 10.
- *While*-rakennetta käytetään, kun ohjelmasilmukkaa halutaan suorittaa ”niin kauan kuin ehto x on voimassa”, yleensä sen näkee ohjelmissa muodossa *while True*. (Python Software Foundation 2017, More Control Flow Tools.)

Kun käytetään näitä ohjausrakenteita, sisennysten oikea paikoitus on ehdottoman tärkeää. Tämän opinnäytetyön aikana tämän sai huomata muutaman kerran, kun ohjelman oikea suoritus oli yhdestä sisennyksestä kiinni.

Kun työskennellään Python IDLE:ssä, kaikki muuttujat ja funktiot, joita olet siinä määrittellyt, pyyhkiytyvät pois muistista, kun IDLE:n sulkee. Python-ohjelmointi on siitä kätevää, että ohjelmat voi kirjoittaa moduleiksi, joita sitten kutsutaan IDLE:ssä ja toistensa sisällä tarpeen mukaan. (Python Software Foundation 2017, Modules.) C-kielessä näitä moduleita kutsuttaisiin kirjastoiksi. Moduulit voivat sisältää määrittelyjä (definitions) tai luokkia (classes), jotka ovat Pythonin olio-ohjelmoinnin perusta, tai yksinkertaisuudessaan suoritettavia funktioita, joita haluaisit käyttää useammassa instanssissa ohjelmasi sisällä. Luokkia ja määrittelyjä voidaan kutsua modulin sisältä *import*-toiminnon avulla. Esimerkki modulin sisältämän määrittelyn kutsumisesta on kuvassa 11.



The image shows two windows from the Python IDLE environment. The top window, titled 'fibonacci.py - E:/Python/fibonacci.py (3.5.2)', contains the following Python code:

```
# tehdään ohjelma fibonaccin lukujonon tulostamiseen

def fib(n): # kirjoittaa fibonaccin lukujono lukuun n saakka
    a, b = 0, 1
    while b < n:
        print(b, end=' ')
        a, b = b, a+b
    print()
```

The bottom window, titled 'Python 3.5.2 Shell', shows the execution of the script:

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import fibo
>>> fibo.fib(2000)
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
>>>
```

Kuva 11. Esimerkki modulin käytöstä (Python Software Foundation 2017, Modules).

3.3 Anturointi

Mäkelä S.:n (2014) mukaan anturi on mittalaitteen osa, joka reagoi ympäristön kanssa esimerkiksi sähköjohtavuuden avulla. Antureiden avulla voidaan mitata fysikaalisia suureita, kuten kiihtyvyyttä tai lämpötilaa, sekä tunnistaa kemiallisia yhdisteitä.

Antureita voidaan jaotella niiden mittaaman suureen, kytkentätavan, koon, tunnistustavan ja mittaustavan perusteella. Anturit voidaan myös jaotella absoluutti- ja inkrementtiantureihin sen mukaan, säilyttävätkö ne referenssipisteensä, kun niistä sammutetaan virta. Antureiden antama signaali voi olla analoginen eli jännite- tai virtasignaali, jonka suuruus vaihtelee tietyllä välillä verrannollisesti mitattavaan suureeseen (esimerkiksi 4 – 20 mA). Signaali voi olla myös digitaalinen eli 0- ja 1-tason jännitepulsseista koostuva. (Mäkelä S. 2014.)

Taulukossa 1 on eritelty joitakin erilaisia anturityyppejä. Näistä yksinkertaisin lienee mekaaninen mikrokytkin. Kytkin on yleensä pakattu pieneen muovikoteloon, ja siitä löytyy nappi, rulla tai vipu, joka alas painamalla virta kulkee tai ei kulje kytkinjohtimen läpi, riippuen siitä on kytkin avautuva (NC, Normally Closed) vai sulkeutuva (NO, Normally Open). Sama jaottelu pätee myös muissa anturimalleissa, jotka lähettävät digitaalista signaalia.

Antureiden ominaisuuksien kuvaamiseen on olemassa monia termejä:

- Lineaarisuus: kuinka tarkasti ja suoraviivaisesti anturin signaali seuraa mitattavaa suuretta
- Toistotarkkuus: kuinka tarkkaan anturin antama lähtösignaali toistuu, kun mittausolosuhteet pidetään vakiona
- Resoluutio: lähtösignaalin askelmuutos, kun mitattava suure muuttuu jatkuvasti (kuinka tarkkoja arvoja anturilla voidaan mitata)
- Hystereesi: lähtösignaalin päälle ja pois kytkevän mittaussuureen asema-arvon ero
- Mittausalue: ne mitattavan suureen raja-arvot, joiden välillä anturin on tarkoitettu toimivan, esimerkiksi 0 - 100 °C.
- Ylikuormitettavuus: kykeneekö anturi toimimaan nimellismittausalueen ulkopuolella vioittumatta
- Ryömintä: Lämpötilan vaikutus mittaustulokseen. (Mäkelä S. 2014.)

Taulukko 1. Erilaisia anturityyppejä ja niiden ominaisuuksia sekä käyttöetuja

Tyyppi	Toimintaperiaate	Signaali	Edut
Rajakytkin / mikrokytkin	Mekaaninen voima, kosketus	Digitaalinen	Yksinkertainen ja edullinen
Induktiivinen	Anturi luo magneettikentän, josta tunnistettava metallinen esine indusoituu	Digitaalinen	Tunnistaa kappaleen koskettamatta, ei liikkuvia osia, helppoja ja nopeita asentaa
Kapasitiivinen	Anturissa kondensaattori, jonka kapasitanssi muuttuu, kun tunnistettava kappale tuodaan lähelle	Digitaalinen	Tunnistaa erilaisia materiaaleja, myös metallin
Magneettinen	Anturissa magneettiherkkä kosketin	Digitaalinen	Lian-, pölyn- ja kosteudensieto
Hall	Magneettivuossa tapahtuvan muutoksen havaitseminen, Hall-ilmio	Digitaalinen	Nopea toiminta, pitkä elinikä
Ultraääni	Mittaa ultraäänen kulkuaikaa kohteeseen	Analoginen / digitaalinen	Tunnistaa riippumatta valoisuudesta, lämpötilasta, väristä ja materiaalista
Laser	Mittaa lasersäteen kulkuaikaa kohteeseen	Analoginen / digitaalinen	Etäisyyden tunnistamisen tarkkuus

Tässä opinnäytetyössä oleellisin anturityyppi on laseranturi sen etäisyyden tunnistamistarkkuuden takia. Myös ultraäänianturi on ollut harkinnassa, mutta siitä luovuttiin epätarkkojen mittaustulosten takia.

3.4 Harjattoman tasavirtamoottorin ohjaus

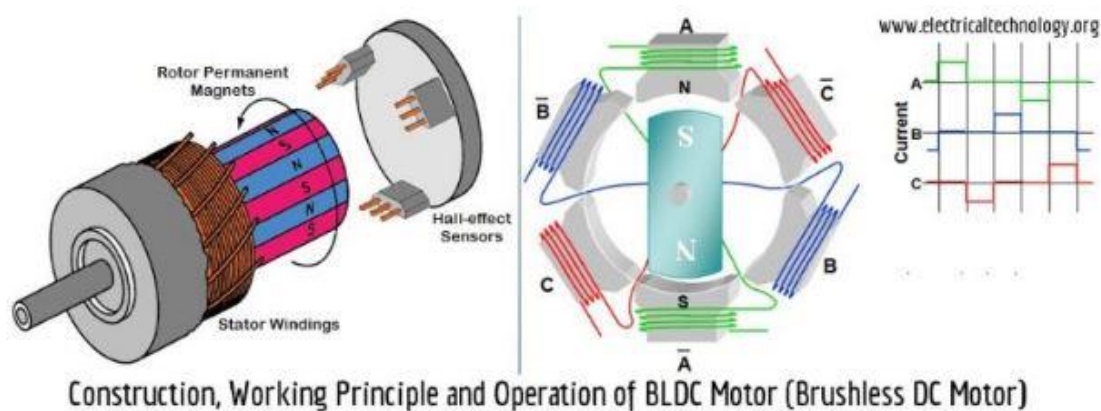
Harjattomat tasavirtamoottorit ovat sähkömoottoreiden alalaji, ja ne tarvitsevat toimiakseen tasavirtaa. Toisin kuin harjallisessa DC-moottorissa, harjattomassa moottorissa roottori (pyörivä keskiosa) on kestmagneetoitu ja staattorissa (ympäröivä paikoillaan pysyvä ulkokehä) sisältää käämityksen. Käämien lukumäärä vaihtelee, mutta yleinen lukumäärä on 6, jolloin ne muodostavat 3 käämitysparia (kuva 12). (Electrical Technology 2016, Brushless DC Motor.)



Kuva 12. Esimerkki harjattoman DC-moottorin käämityksestä (Electrical Technology 2016, Brushless DC Motor.)

Toisin kuin harjallisessa DC-moottorissa, jossa kommutointi eli virran suunnan muuttaminen roottorin pyörittämiseksi tehdään fyysisten harjojen avulla, harjattomassa DC-moottorissa käämipareille johdetaan vuorotellen sähkövirta, jolloin vastakkaisille puolille muodostuu hetkellinen pohjois- ja etelänapa. Napojen välille indusoituu magneettikenttä, joka saa roottorin pyörimään, kun roottorin kestmagneettikenttä pyrkii asettumaan samansuuntaiseksi syntyneen magneettikentän kanssa. Näin saadaan aikaan vääntömomentti moottorin akselille. Kun käämipareille sitten vuorotellen syötetään virtaa, roottori pysyy jatkuvassa liikkeessä. Pyörimissuuntaa tarkastellaan Hall-anturein, jotka on asettu staattorin ympärille. Moottorin rakennetta ja käämityksen virrankulun sekvenssikaa-viota on esitelty kuvassa 13. (Electrical Technology 2016.)

Harjattomat DC-moottorit ovat huoltovapaampia, hiljaisempia ja pitkäikäisempiä kuin harjalliset moottorit harjallisten kommutointiin tarvitsemien harjojen puuttumisen vuoksi. Harjattomilla moottoreilla voidaan myös saavuttaa suuremmat kierrosnopeudet.

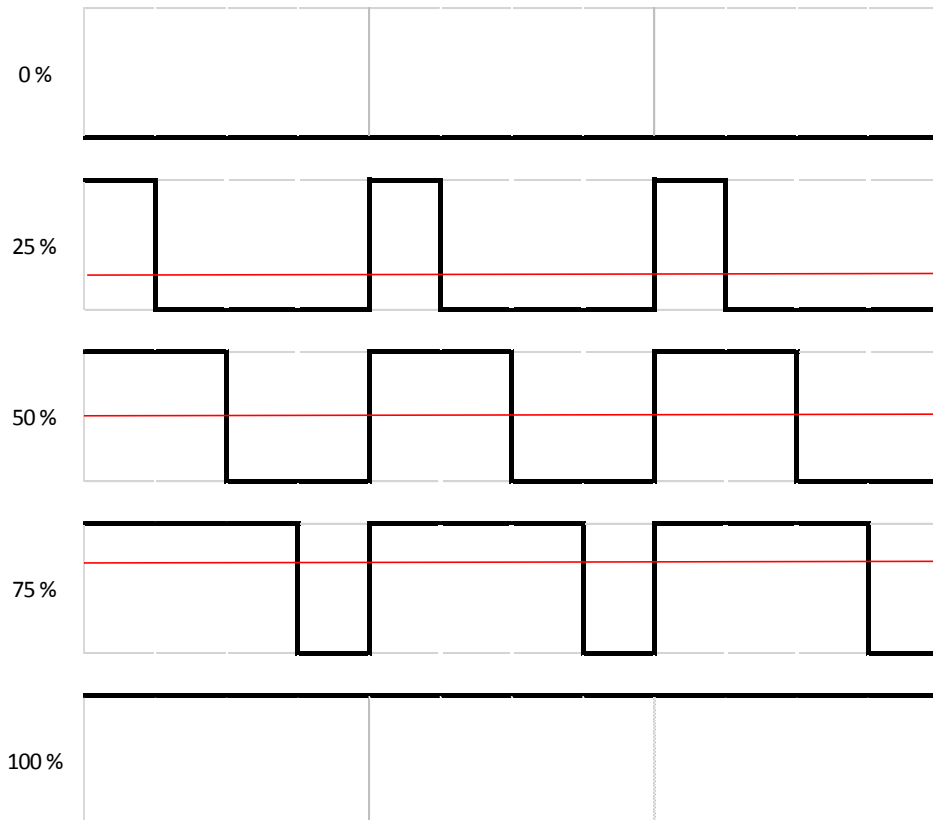


Kuva 13. Harjattoman DC-moottorin perusrakenne sekä sekvenssikaavio (Electrical Technology 2016)

Harjattomat DC-moottorit ovat kuitenkin harjallisia moottoreita kalliimpia, ja käämitys on altis korkeille lämpötiloille. Harjattomien DC-moottorien ohjaaminen on myös hankalaa, vaatien usein ohjainyksikön työstämään käämityksen virransyöttöä, Hall-anturin signaaleja ja pyörimissuunnan vaihtoa, nopeudensäädöstä puhumattakaan. (Electrical Technology 2016.)

Harjattoman DC-moottorin nopeutta voi säätää vaikkapa potentiometrillä, sillä kun syöttöjännite pysyy samana ja resistanssia kasvatetaan, niin käämitykselle menevä virta pienenee Ohmin lain mukaisesti. Pienempi virta indusoi pienemmän magneettikentän staattorin käämityksen kelalle, pienentäen magneettikentän voimakkuutta ja näin ollen myös kentän aiheuttamaa vääntömomenttia, hidastaen roottorin pyörimisnopeutta. (Mäkelä, M., Soininen, Tuomola & Öistämö 2012, 125-126.)

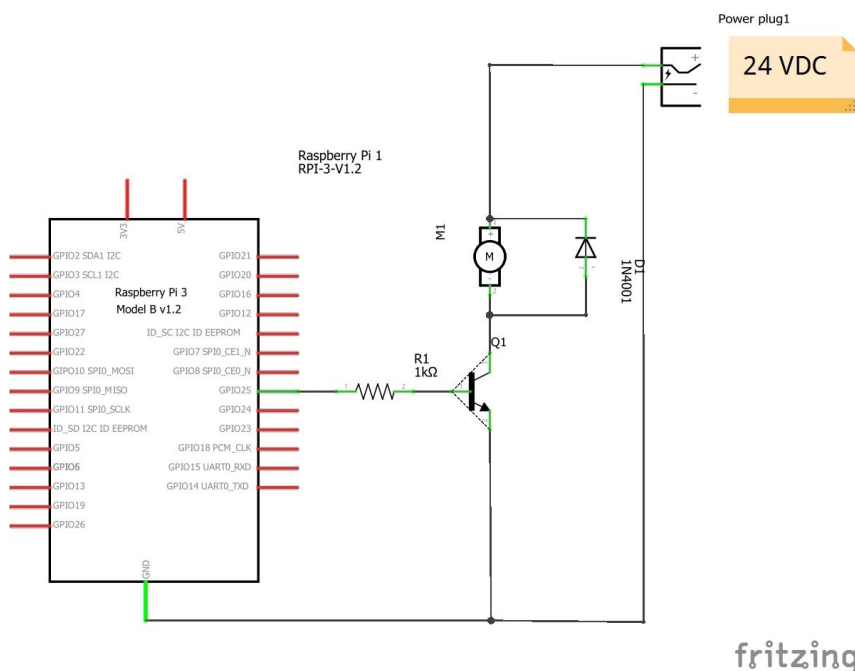
Jos kuitenkin halutaan ohjata harjattoman DC-moottorin nopeutta ohjelmallisesti esimerkiksi mikrokontrollerilla, tarvitaan pulssinleveysmodulaatiota (PWM). PWM hyödyntää digitaalisen ulostulon signaaleja muuntamalla signaalit analogisiksi. PWM:ssä digitaalilähtöä kytketään päälle ja pois tietyllä pulssisuhteella, jolloin syntyvä kanttiaalto antaa ulostulevalle jännitteelle arvon väliltä täysin kiinni = 0 V -> täysin auki = 5 V (esimerkiksi). (Kuphaldt 2017). Kuvassa 14 olevilla pulssisuhteilla saataisiin aikaan jännitteet 1,25 V (25 %), 2,5 V (50 %) tai 3,75 V (75 %).



Kuva 14. Pulssinleveysmodulaatiossa käytettävän pulssisuhteen ja syntyvän ulostulojännitteen yhteys.

DC-moottorin ohjaus ei kuitenkaan ole ihan näin yksinkertaista, sillä moottorin käämitysten vuoksi moottorissa on sekä induktanssia että resistanssia. Induktanssin takia kestää jonkin aikaa, että moottorille tuleva virta saavuttaa maksimiarvonsa $I = \frac{U}{R}$, sillä käämityksiin indusoitua sähkömagneettinen kenttä vastustaa virran muutosta. Jokaisella DC-moottorilla on aikavakio $\tau = \frac{L}{R}$, missä L on moottorin induktanssi ja R on moottorin resistanssi. Aikavakio kertoo, missä ajassa komponentin jännite saavuttaa $\frac{1}{e} \approx 37\%$ lopullisesta arvostaan. Virta saavuttaa maksimiarvonsa vasta useamman aikavakion kulumisen jälkeen induktanssin takia, mitä on kuvattu kuvassa 15. Tästä syystä PWM-signaalin taajuuden asettaminen liian suureksi heikentää moottorin vääntömomenttia, sillä vääntömomentti on riippuvainen käämeille tulevan virran suuruudesta. Pulssin päällä-osuuden tulisi olla vähintään 5τ , jotta ohjaussignaali olisi lähellä lineaarista. (Precision Microdrives Limited. 2017.) Maksimitaajuus olisi tällöin 1 % resoluutiolla $f_{max} = \frac{1}{5\tau \cdot 100}$. Liian alhaisella taajuudella taas käämitys sopeutuu kulkevaan virtaan ja roottori pysähtyy, kun staatorikäämitykseen indusoituvan magneettikentän muutokset eivät enää riitä pyörittämään roottoria.

Mikrokontrollereilla voi ohjata DC-moottoreita vain tiettyyn rajaan asti, sillä niiden pinnien ulostulovirta on rajallinen. Esimerkiksi Raspberry Pin GPIO-ulostulosta saa maksimissaan 15 mA ulostulovirran. Tämä ei riitä vähänkään järeämpien moottoreiden käytössä, vaan moottori saattaa ottaa paljon suuremman ohjausvirran kuin mitä GPIO-pinni kestää, ja Raspberry Pi voi vahingoittua, vaikka kortilla on sulake. Tästä syystä käytetään NPN-transistoria, jonka virrankestävyys on riittävä moottorin ottamaan ohjausvirtaan nähden. Transistori toimii kytkimenä Raspberry Pin ja moottorin välillä. Moottorin kanssa rinnan kannattaa kytkeä diodi, sillä moottorista tulevat induktiiviset jännitepiikit voivat ylikuumentaa tai vahingoittaa transistoria. (Raja 2017.) Kuvassa 15 on esitetty transistorilla varustetun DC-moottorin ohjauspiirin periaatetta.

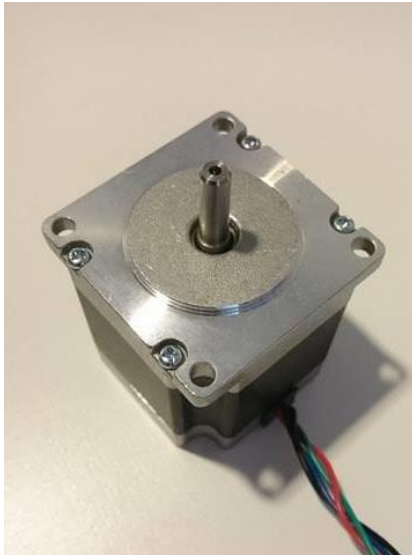


Kuva 15. Raspberry Pin DC-moottorin ohjaus transistorin avulla

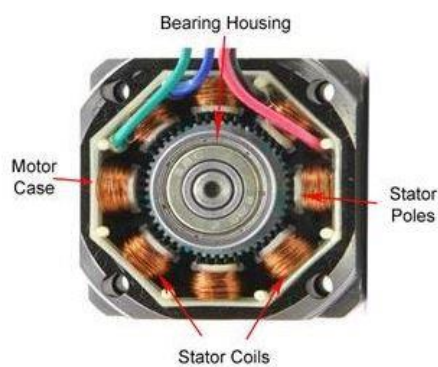
3.5 Askelmoottorin ohjaus

Askelmoottori on käytännössä harjaton DC-sähkömoottori, jota ohjataan sähköisillä impulsseilla. Toisin kuin tavallisissa sähkömoottoreissa, askelmoottorin kierrosnopeutta ohjaillaan ohjauspulssien määrää tihentämällä ja harventamalla. Tässä opinnäytetyössä käytetään niin kutsuttua hybridiaskelmoottoria (kuva 16), jossa roottori on kestmagnetoitu ja siinä on 2 erinapaista rautauista hammaspyörää (kuva 17). Myös staattorin käämityksessä on joka käämille hammastukset, ja askelten ottaminen perustuu erinapaisten hampaiden keskinäiseen vetovoimaan. Roottorin hampaat ovat aina hieman eri kohdassa seuraavaan käämiin/käämpariin nähden. Kun nyt syötetään virtaimpulssi käämille, roottorin

hampaat asettuvat sen käämin kanssa vastakkain (kokonainen askel). Käämeihin sitten syötetään virtaimpulssi vuoron perään, ja roottori liikkuu askeleittain eteenpäin. Käämejä voidaan myös virroittaa kaksi tai useampi yhtä aikaa, jolloin saadaan aikaan puolikkaita askelia tai sitä pienempiä askelia. Tämän takia askelmoottorit ovat erinomaisia sovelluksissa, jossa vaaditaan suurta asematarkkuutta. (Daware 2016.)



Kuva 16. Työssä käytetty Nema 23-askelmoottori.

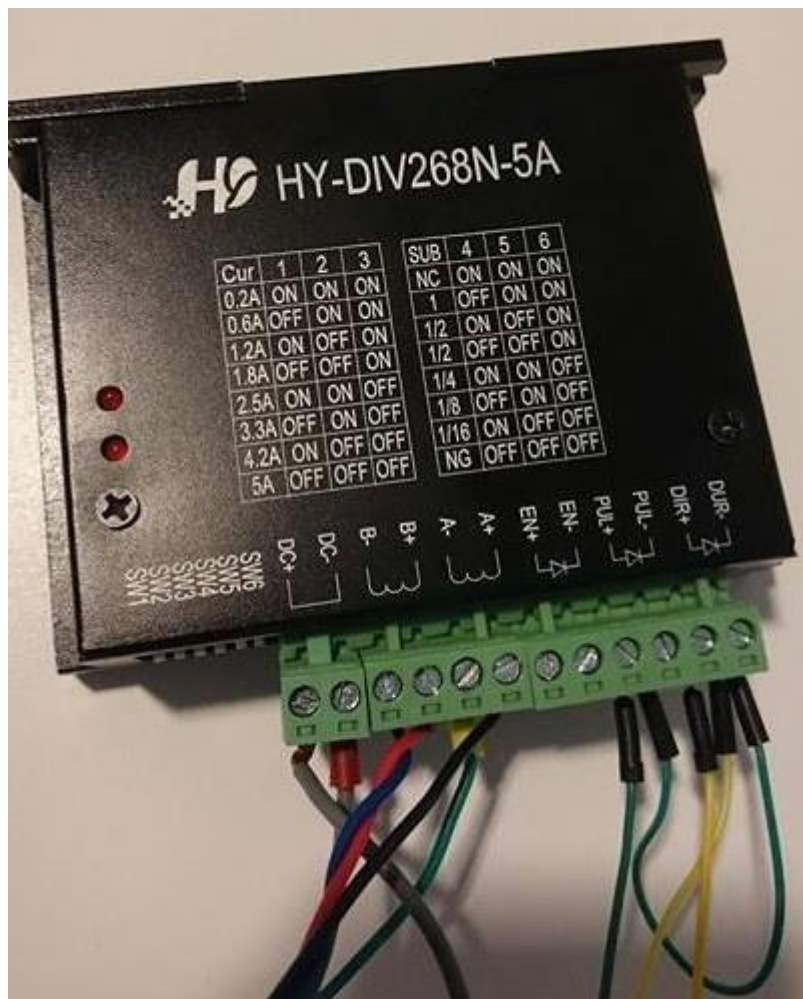


Kuva 17. Hybridiaskelmoottorin rakenne (Agnihotri 2012)

Suurin osa askelmoottoreista on kaksivaiheisia, mutta myös kolme- ja nelivaiheisia askelmoottoreita löytyy markkinoilta. Ne voivat olla joko unipolaarisia, jolloin jokaisessa navassa on 2 käämitystä kierrettynä eri suuntiin, jolloin pyörimissuuntaa voidaan vaihtaa suuntaamalla virta käämilangasta toiseen. Unipolaariset moottorit vaativat vähemmän ohjaimelta, sillä siihen ei tarvita komponenttia virran suunnan muutokseen. Bipolaarisissa moottoreissa jokaisessa navassa on yksi kierretty käämi, ja suunnan vaihtamiseksi pitää

virran suuntakin vaihtaa esimerkiksi H-sillalla. (Carrier, Ohline & Kenny 2011, 632-635.)

Koska askelmoottoreita ohjataan sähköisillä impulsseilla, niiden ajoittaminen on tärkeää sujuvan askelluksen aikaansaamiseksi. Useimmiten tähän ajoitukseen tarvitaan erillinen ohjain mikrokontrolleriohjaimen, kuten Arduinon tai Raspberry Pin lisäksi. Ohjaimesta riippuen tarvitaan mikrokontrollerilta ohjaustieto joka vaiheelle, tai vain pulssitaajuus ja suuntatieto. Joissakin on myös mahdollisuus valita askelmoottorille menevän pulssin virta (jolla voidaan vaikuttaa nopeuteen ja vääntömomenttiin) ja askelten laatu (kokonaiset askelet, puoliaskleet yms.) kuten kuvassa 18.



Kuva 18. Opinnäytetyössä käytetty askelmoottorin ohjain. Vasemmalla alhaalla ovat mikrokytkimet, joita päälle tai pois laittamalla saadaan virta- ja askelasetuksia muutettua.

4 KOPIOSORVIPROJEKTIN HISTORIA

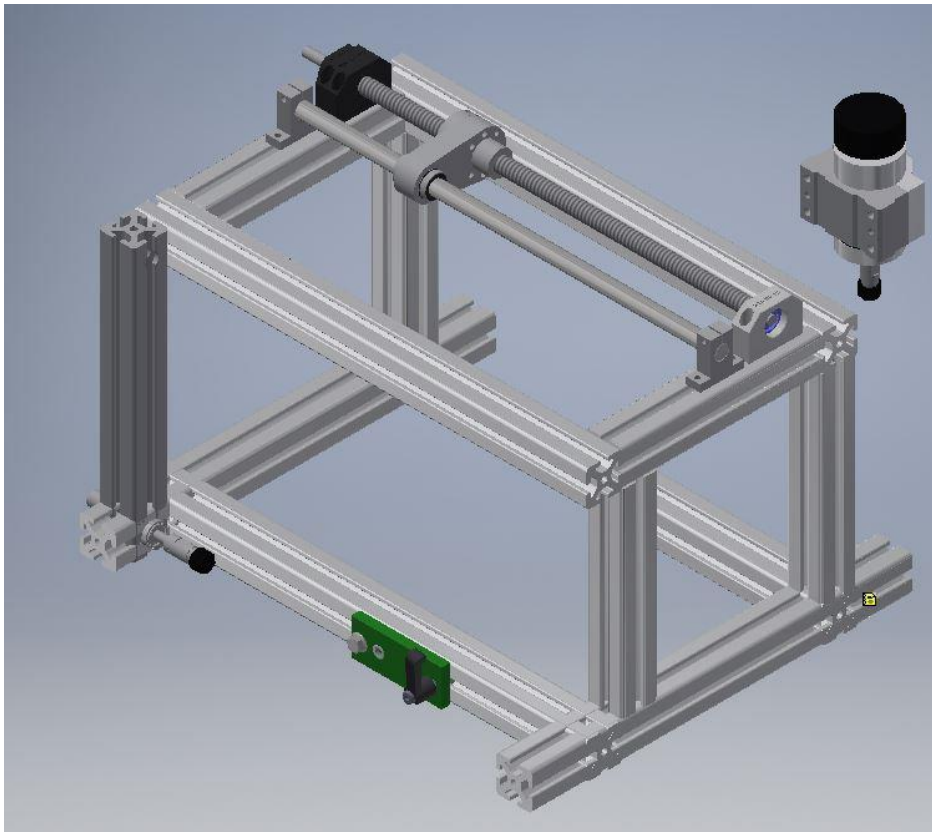
Kopiosorviprojekti lähti liikkeelle syksyllä 2015 kalastuksesta innostuneen opiskelijato-verin ehdotuksesta. Hän itse teki opinnäytetyönään sähkötakilaa vaappukalastukseen. Hän teki vaappuja itse puusta, ja toivoi että käsityönä syntyvien vaappujen sijaan voisi tehdä vaappuja kopioimalla olemassa olevien vaappumallien muotoja. Opiskelijaprojek-tia lähti TAMKIn puolesta vetämään Seppo Mäkelä. Sorvia lähdettiin suunnittelemaan liitteessä 1 olevan vaatimusluettelon pohjalta.

Alusta asti oli selvää, että valmistettavan sorvin tulisi olla opiskelijabudjettiin sopiva sekä yksinkertainen laite kotikäyttöön, mutta ajan myötä mieli muuttui ja tulevaisuuden tavoit-teena onkin tuottaa sorvia piensarjatuotantona myytäväksi erilaisilla Internet-palstoilla. Myös sorvin toiminta on ottanut suuria harppauksia monimutkaisempaan suuntaan ajan kuluessa. Tuotteen paino haluttiin myös pitää kohtuuden rajoissa, jotta sitä olisi helppo siirtää, ja koon piti olla sellainen, että se olisi sijoitettavissa vaikka keittiön pöydälle.

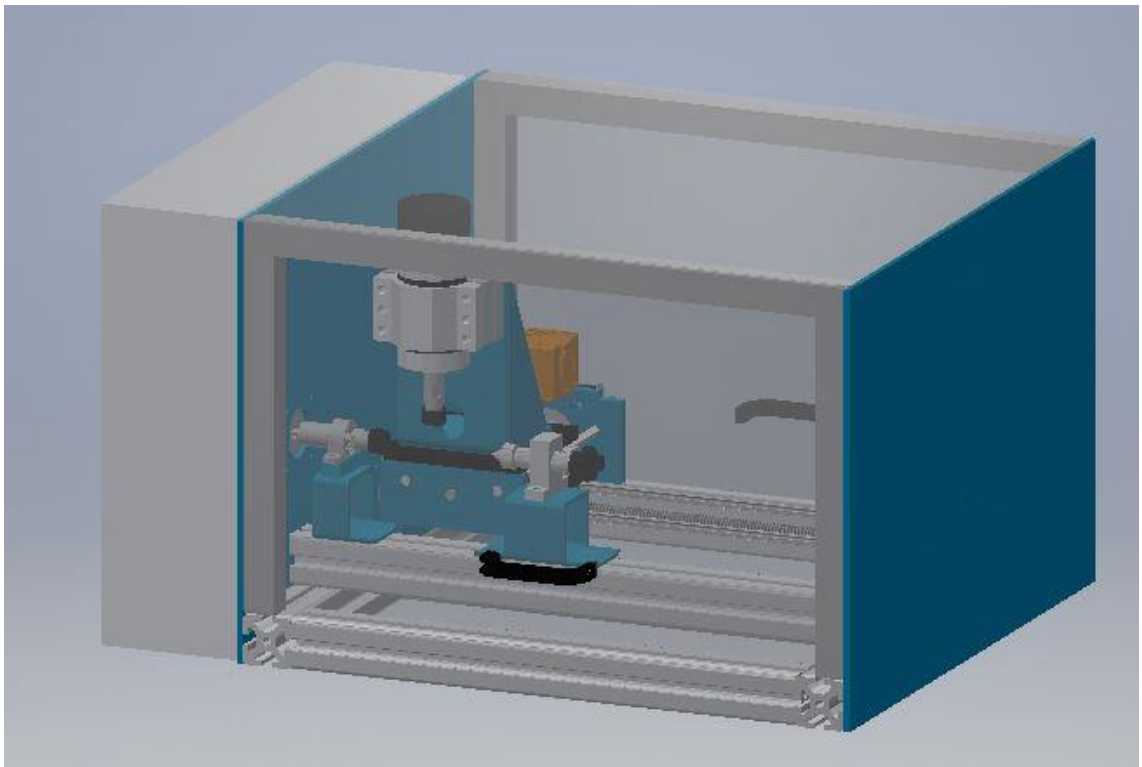
4.1 Alkuluonnostelu ja olemassa olevien tuotteiden tarkastelu

Aluksi lähdettiin tarkastelemaan olemassa olevia yksityishenkilöiden tekemiä tuotteita. Inspiraatiota löytyikin Internetin kalastusforumeilta runsaasti, ja Googlen kuvahaulla löy-tyi monia mielenkiintoisia malleja, joista lähdettiin etsimään ominaisuuksia, joita synty-vään laitteeseen haluttiin. Prosessin aikana huomattiin, että iso osa ihmisten laitteista oli reippaasti ylimitoitettuja ajatellen laitteiden käyttötarkoitusta, ja ohjenuorana luonnoste-lussa oli karsia laitteesta kaikki ylimääräinen, jotta laitteesta tulisi kokonaishinnaltaan mahdollisimman edullinen.

Kun tutkimustyötä oli tehty jonkin aikaa, tuotiin avuksi 3D-mallinnus CAD-ohjelmalla, jolla kyettiin hahmottamaan laitteen muotoa ja näkemään, mikä toimii ja mikä ei kompo-nenttien sijoittelussa. Mallinnuksessa pyrittiin ratkaisemaan osaongelma kerrallaan kohti toimivaa kokonaisuutta. Kuvassa 19 on alkuvaiheen luonnostelua rungosta ja työstettävän kappaleen sekä työkalun sijoittelusta. Siinä oikeastaan kaikki näkyvät komponentit hake-vat vielä omaa paikkaansa. Kuvassa 20 taas on tämänhetkisen prototyypin mallinne, josta suurempi kuva löytyy liitteestä 2 eri komponenttien sijainnin hahmottamisen paranta-miseksi.



Kuva 19. Alkuvaiheen mallinne. Tässä suunnitellaan rungon rakennetta ja työkalun sijoittelua sekä sen liikutteluun tarvittavan liikeruuvien paikkaa.



Kuva 20. Rakennettavan prototyypin tämänhetkinen mallinne. Runko ja komponentit ovat paikoillaan, laitteen ohjaus on vasemmassa reunassa olevan harmaan laatikon sisällä.

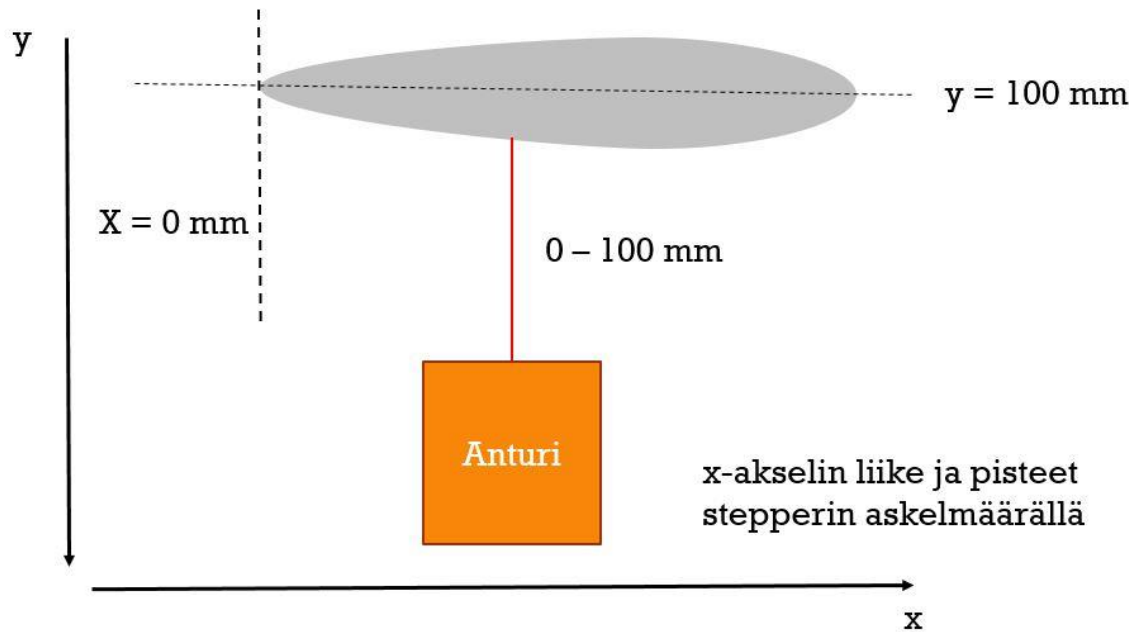
4.2 Kopiointimenetelmän valinta

Suurin osa yksityishenkilöiden tekemistä kopioivista sorveista hyödynsi mekaanista kopiointitapaa, kuten kuvassa 21. Metallista tai puista mallikappaletta seuraili tanko, joka oli kiinteästi yhteydessä työstökoneeseen, joka näin kopioi mallin muodon työstettävään puukappaleeseen. Projektin alkuvaiheessa lähdettiin suunnittelemaan tällaista sorvia, jonka pyörittävän moottorin ohjaus toteutettaisiin Arduinon ja H-siltakytkennän avulla.



Kuva 21. Kuvakaappaus mekaanisesti kopioivan sorvin toiminnasta (Vaappusorvi, Lauri M., YouTube 2015)

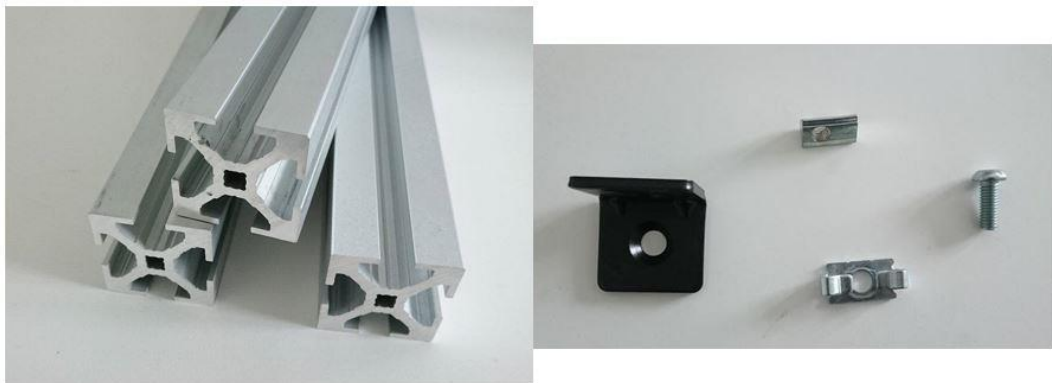
Syksyllä 2016 projekti sai kuitenkin radikaalin käänteeseen, kun päätettiin ryhtyä suunnittelemaan kopiointitapaa, jossa tallennettaisiin pisteitä mallikappaleesta laseretäisyysanturin avulla. Tätä toimintamallia on pyritty havainnollistamaan kuvassa 22. Pisteet tallennettaisiin ja niiden avulla ohjattaisiin askelmoottoreita pyörittämään kahta liikeruuvia, liikuttaen työstökoneita yhtenevässä koordinaatistossa. Toisin kuin kuvassa 21, työstökone siis liikkuu pystysuunnassa, kuten kuvassa 20 on havainnollistettu. Liitteessä 2 on kopiosorvin 3D-malli isommassa koossa helpottamaan yksittäisten komponenttien sijoittamisen hahmottamista.



Kuva 22. Havainnekuva mallikappaleen skannauksen toiminnasta. Y-akselin pisteet tulisivat anturilta ja x-suunnassa askelmoottorien ohjauksen perusteella. Koska Anturin minimimittausetäisyys on 200 mm, asetettiin anturin mittausalue välille 200 - 300 mm, jotta anturin mittausalue mahtuisi kopiosorvin sisälle.

5 KOMPONENTTIEN VALINTA

Komponenttien valinnassa lähdettiin liikkeelle runkomateriaalin valinnasta. Vaihtoehdot olivat käytännössä alumiiniprofiili ja teräksinen, hitsattava neliöputki. Rungon haluttiin olevan helposti koottava, joten alumiiniprofiilin todettiin olevan parempi vaihtoehto. Projektiin saatiin lahjoituksena profiilia Purso Oy:ltä, ja mahdolliset huolet materiaalin kestävydestä kaikkosivat nopeasti, sillä profiili osoittautui todella jämäkäksi. Tämän lisäksi hankittiin kiinnitystarvikkeita profiilin kappaleiden yhteen liittämiseen. Alumiiniprofiilin muotoa ja profiilikappaleiden liittämiseen tarkoitettuja tarvikkeita on kuvattu kuvassa 23.



Kuva 23. Projektissa käytettävä alumiiniprofiili ja sen palasten liittämiseen tarkoitettuja kiinnikkeitä.

Komponenttien valintaprosessi oli melko sujuva. Suurin osa komponenteista löytyi pienellä nettihaeskelulla, mutta varsinkin työstöjyrsimen ja laseranturin valinta ja hankinta tuottivat päänvaivaa. Määritteleviä tekijöitä komponenttien valinnassa olivat hinta, paino sekä toimintajännite, sillä järjestelmän haluttiin toimivan 24 voltilla, jota voitaisiin sitten alentaa komponenteille tarvittaessa.

X- ja y-suuntaiset liikkeet päätettiin toteuttaa trapetsikierretangoilla. Kuularuuvitkin olivat harkinnassa, mutta niiden korkea hinta jätti ne lopulta pois projektista. Trapetsikierretankoa katkaistaisiin 2 sopivan mittaista pätkää, joiden päihin sorvattaisiin itse päihin tulevat modifikaatiot projektitoverin työpaikalla.

Kopiosorvin rakentamisessa ei trapetsikierteen lisäksi selvitty täysin ilman koneistusta, sillä työkalua pitelevään telineeseen sekä askelmoottoreiden kiinnitykseen tarvittiin laserleikattuja metallikappaleita. Kappaleet teetettiin HTL Works Oy:n ammattilaisten käsittelyssä. Laserleikkeiden piirustuskuvat ovat liitteessä 3.

5.1 Sorvaustyökalu

Opinnäytetyön alussa, kuvassa 1, on kuvattu sorvaustyökaluksi valikoitunutta pienjyr-sintä, niin kutsuttu ”spindle motor”. Se on harjaton DC-moottori, jonka mukana tuli oh-jainlaatikko, joten sen ohjaaminen mikrokontrollerilla tai Raspberry Pillä on suhteellisen helppoa. Työkalu löytyi Ebay-nettikauppapaikasta, jossa on myös muita samankaltaisia tuotteita harrastelijakäyttöön, sekä myös kalliimmilla osilla ja ohjauspiireillä varustettuja versioita vaativammille asiakkaille.

Tämä kyseinen malli toimii +18 - +52 voltin DC-jännitteellä, ja moottorille lupailaan 400 W tehoa pyörittämiseen. Moottorin maksimipyörimisnopeus on 12000 kierrosta mi-nuutissa, ja sitä voi säätää joko mukana tulevalla potentiometrillä tai ohjelmallisesti mik-rokontrollerilla. Raspberry Pi:ssä PWM-mahdollisuus on toteutettavissa ohjelmallisesti, sillä GPIO-pinneissä ei PWM:ää ei suoralta kädeltä löydy, sillä PWM-piiri on audiolii-tännän käytössä. Onneksi kuitenkin tätä varten löytyy valmis moduli. Moottorissa on Hall-sensorit pyörimissuunnan havaitsemiseen, joten suunnan muutos ja reaaliaikainen pyörimisnopeuden tulostaminen onnistuvat ohjelmallisesti.

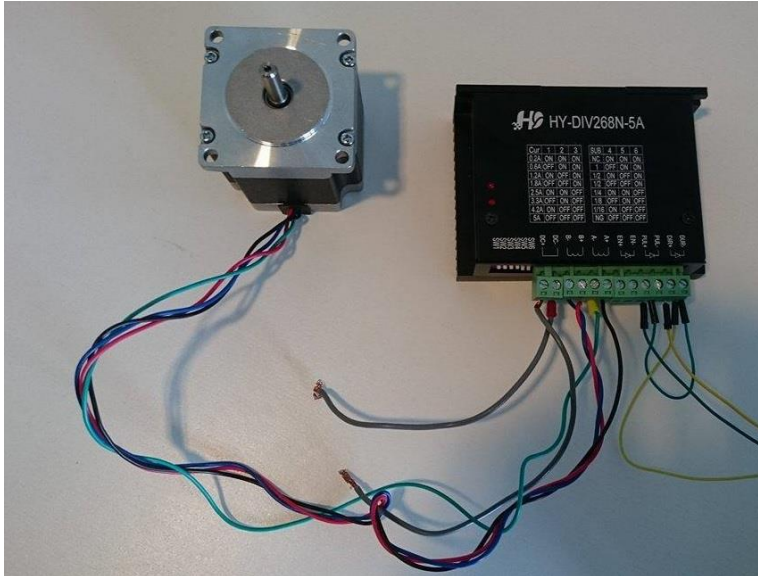
Jyrsinterä kiinnitetään moottorin akselille kuvan 24 mukaisella kiinnitysmekanismissa. Jousimainen kiristeosa menee mutterin ja akselin välissä olevaan koloon, ja mutteria ki-ristettäessä kiristyy jousiosa jyrsinterän akselin ympärille pitäen jyrsinterää paikoillaan.



Kuva 24. Vasemmalla oleva ”jousiosa” tulee oikealla olevan mutterin sisään, ja mutteria kiristäessä jousiosa kiristyy kiinnitettävän akselin ympärille.

5.2 Moottorit ja niiden ohjaus

Järjestelmän y- ja x- akselien suuntaisen liikkeen sekä kopioitavan ja työstettävän kappaleen pyörittämisen toteuttamiseksi päätettiin hankita kolme samanlaista askelmoottoria ja niille jokaiselle ohjainlaatikot. Projektissa päädyttiin käyttämään Nema 23-moottoreita, jotka 2,8 ampeerin virralla tuottaisivat 1,26 Nm:n vääntömomentin. Kuvassa 25 on yksi askelmoottoreista ohjainlaatikonsa kanssa.



Kuva 25. Projektissa käytetty askelmoottori ohjaimensa kanssa.

Ohjainlaatikot olivat siitä käteviä, että niiden kyljessä oli mikrokytkimet, joista saattoi lennosta vaihtaa moottorin vaiheille menevää virtaa sekä mikroaskelten määrää. Tämä helpotti työskentelyä testausvaiheessa, kun haettiin oikeaa konfiguraatiota pyörimisnopeuden ja yhden pyörähdysaskelmäärän välille, valmistajan ilmoittama 200 askelta/kierros -lukema kun ei tuntunut pitävän paikkansa.

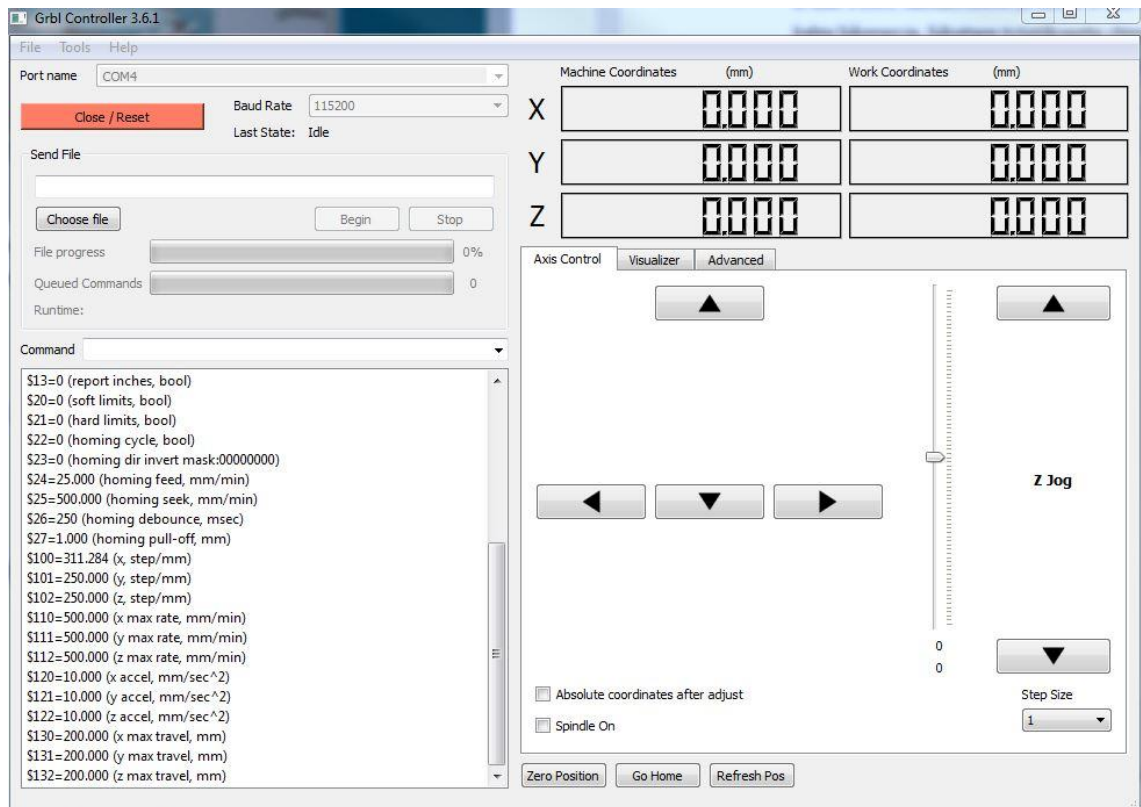
Projektin aikana todettiin, että projektissa olisi voinut käyttää myös kevytrakenteisempia Nema 17 -moottoreita esimerkiksi kappaleiden pyörittämisessä, sillä puukappaleiden pyörittämiseen ei tarvitse juuri kummoistakaan vääntömomenttia. Nema 17 -moottorit ovat hyvin yleisiä 3D-tulostinsovelluksissa, ja niiden toimivuudesta on runsaasti näyttöä. Nema 23 -moottorit palvelivat kuitenkin tarkoitustaan enemmän kuin hyvin, eikä niiden paino ollut oleellinen este niiden pitämiseksi mukana projektissa.

Askelmoottorien sekä työkalun liikkeiden ja pyörittämisen automatisoinnissa käytettiin hyväksi Grbl-ohjainta. Grbl on ilmainen, avoimen lähdekoodin CNC-ohjain, jota käytetään moottorien ja muiden liikkuvien osien ohjauksessa. Se on suunniteltu toimivaksi Atmega328p -mikrokontrollerissa, joka löytyy Arduino UNOsta. Grbl:ä käytetään paljon avoimen lähdekoodin 3D-tulostimissa ja ihmisten omista projekteista, jotka vaativat koneohjausta. (Github, Inc. 2017, Grbl: Home.)

CNC tulee sanoista Computer Numerical Control. Siinä muutetaan numeeriseen muotoon muotoja vaikkapa 3D-mallista, käytännössä vektoreiksi ja koordinaattipisteiksi koordinaatistossa. CNC-ohjelmaa voidaan myös ohjata niin sanotulla G-koodilla. G-koodi on numeerinen ohjelmointikieli (NC, Numerical Control), jossa kirjain- ja numeroyhdistelmällä ohjataan moottoreita tekemään haluttuja asioita. Komennot jaotellaan komentojen eteen tulevan kirjaimen mukaan, esimerkiksi G (general) -komentoihin ja M (miscellaneous) -komentoihin. G-komennot ovat yleisin komentotyyppi, niillä valitaan koordinaatisto ja liikutellaan moottoria. (Github, Inc. 2017, Grbl v.1.1 Commands.)

Grbl:llä haluttiin toteuttaa varsinainen kopiointi. Laseranturin avustuksella saadut pisteet tallennettaisiin G-koodin käskyiksi moottoreille liikkua x- ja y-akseleilla komennoissa määritelyihin pisteisiin.

Grbl tukee monta erilaista käyttöliittymäohjelmaa, joista tähän projektiin valikoitui Grbl Controller, sen yksinkertaisuuden vuoksi. Monissa CNC-ohjauksissa tarvitaan z-akselin liikesuunta mukaan ohjaukseen, mutta tässä projektissa se ei ole käytössä, joten mahdollisimman yksinkertaisen käyttöliittymän valinta oli järkevää. Lisäksi Grbl Controller toimii sekä Linuxissa että Windowsissa, ja sen sai kätevästi ladattua myös Raspberry Pihin. Käyttöliittymää on kuvattu kuvassa 26.



Kuva 26. Grbl Controller -ohjelman käyttöliittymä. Moottoreita voidaan ohjata itse nuolistalla tai syöttää komentoja komentokenttään. Ohjelmaan voidaan myös tuoda itse tehtyjä G-kooditiedostoja .nc-loppuisessa muodossa.

5.3 Laseranturi

Laseretäisyysanturin valinta ja hankinta tuottivat ylivoimaisesti eniten päänvaivaa, sillä useimmat tuotteet olivat liian kalliita, niiden mittausalue ei ollut sopiva tai ne olivat liiankin tarkkoja projektin tarpeisiin, mikä myös näkyi hinnassa. Lopulta kuitenkin Ifm Electronic Oy:n sivuilta löytyi sopivassa hintaluokassa olevia tuotteita, joiden tarkkuuskin todettiin riittäväksi kopiosorviin. Kun vielä mittausalue oli asetettavissa kopiosorvin fyysisten mittojen sisälle, niin Ifm:lle päätettiin lähettää tarjouskysely. Samoihin aikoihin saatiin TAMKista Seppo Mäkelältä lainaksi Omronin laseretäisyysanturi, joka kuitenkin osoittautui liiankin tarkaksi prototyyppiä ajatellen, ja sen hintakin oli tuhannen euron tietämällä, mikä ei ollut projektille asetetut opiskelijabudjetin rajoissa mitenkään mahdollinen hankinta.

Ifm:n edustaja Samuli Ilmonen vastasi lähetettyyn tarjouskyselyyn ja kysyi samalla, millaiseen projektiin anturi olisi tulossa. Saamiensa tietojen pohjalta hän ehdotti erilaista anturia (kuva 27), kuin mistä tarjouskysely oli alun perin lähetetty, eikä senkään hinta päättä

huimannut, varsinkin kun Samuli antoi anturista vielä opiskelija-alennusta. Se oli vieläpä tarkempi kuin ensimmäinen kysely anturi. Hän antoi myös vastaavan yrityksen testianturin projektia varten koekäyttöön, jotta anturin sopivuutta kopiosorvin mittausanturiksi voitaisiin testata. Testitulokset osoittivat anturin olevan oikein sopiva tähän projektiin, ja kaupat anturista tehtiin tammikuussa 2017.

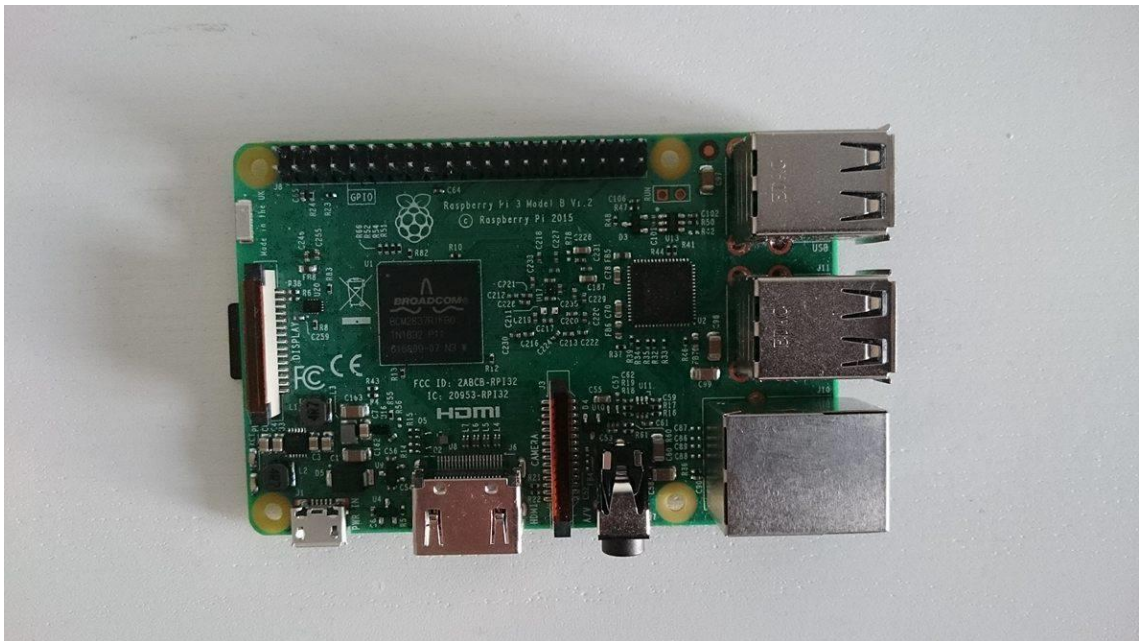


Kuva 27. Työssä käytetty Ifm Electronic Oy:n laseretäisyysanturi mallia 01D102.

Anturista saa analogista tietoa jännitearvona 0-10 V tai virtatietona 4-20 mA. Analogiselle signaalille voi asettaa mittausalueen minimi- ja maksimi-arvot, joiden väliltä jännite- tai virta-arvot asettuvat, kuitenkin huomioiden anturille ilmoitetut absoluuttiset minimi- ja maksimietäisyydet. Sen voi myös asettaa toimivaan tunnistavana anturina. Tunnistavalle signaalille on kaksi eri signaalikaapelia, joista toinen toimii myös analogisessa tilassa. Anturin mukana ei tullut virtakaapelia, mutta sopiva, nelinapainen kaapeli löytyi tamperelaisen Suomen Elektori Oy:n hyllystä. Kaapelin toisessa päässä olevissa johdoissa ei ollut riviliittimelle kytkemistä helpottavia liittimiä, joten ne asennettiin heti ennen testausten jatkamista.

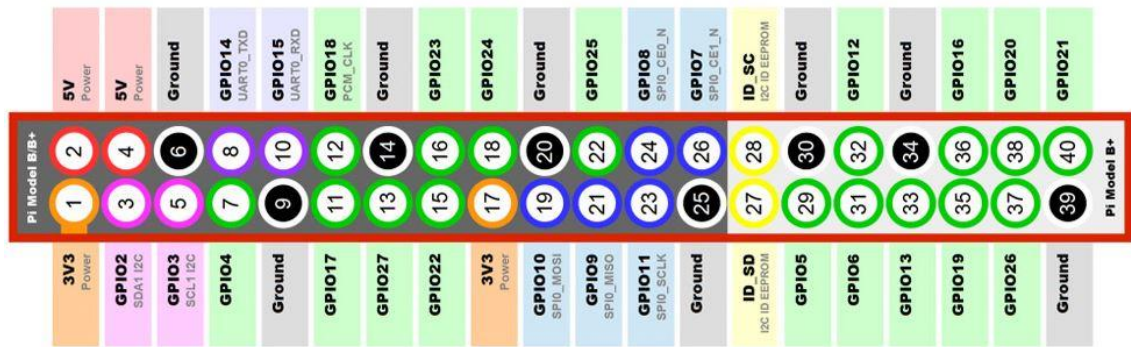
5.4 Mikrotietokone Raspberry Pi sekä MCP3008-A/D-muunnin

Projekti oli tarkoitus toteuttaa ensin Arduino UNO – tai Arduino MEGA -mikrokontrollerilla, mutta kopiosorvissa päädyttiin lopulta käyttämään mikrotietokone Raspberry Pin uusinta mallia Model 3 B (kuva 28). Se maksoi 49,90 euroa ja se oli paljon tehokkaampi kuin pelkkä mikrokontrolleri, ja siinä oli muistikorttipaikka anturidatan tallennukseen, HDMI-näyttöliitäntä ja USB-portit hiirtä ja näppäimistöä varten, joten sen käyttöönotto oli todella helppoa. Tulevaisuutta ajatellen Raspberry Pissä on myös sarjaliitännämahdollisuus näytölle HDMI-liitännän sijaan, joka lisää erilaisten kopiosorvin kanssa käytettävien näyttöjen määrää jonkin verran. Pienissä näytöissä kun HDMI-kaapeliliitäntä on kiiven alla, tai niiden hinta ylittää kipukynnyksen. Internet-yhteys hoidettiin USB-porttiin liitettävällä langattoman verkon sovittimella.



Kuva 28. Projektissa käytetty Raspberry Pi Model 3 B

Raspberry Pi Model 3 B:ssä on 40 I/O-pinniä, joista osa on varattu ulkoisten laitteiden virroittamiseen, maadoitukseen sekä erilaisiin sarjaliitännävaihtoehtoihin, joista erityisesti SPI tuli tutuksi projektin aikana. Osa erikoiskäytössä olevista pinneistä voi toimia myös yleisessä I/O-käytössä. Kuvassa 29 on esitetty eri pinnien funktioita.

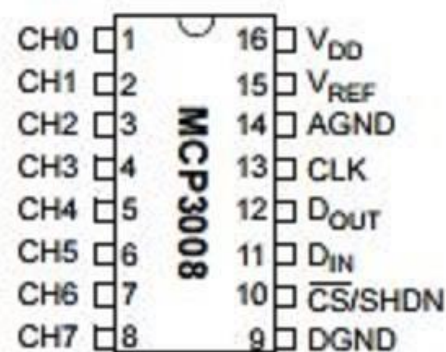


Kuva 29. Raspberry Pi Model 3 B:n I/O-pinnien järjestys (Jameco Electronics 2017)

Ohjelmissa pinnien käytössä on olemassa 2 eri moodia: BCM ja BOARD, jotka käyttävät eri pinninumeroita. BOARD-asetuksen pinnit ovat kuvassa 28 ympyröiden sisällä, BCM-pinniosoitteet GPIO-kirjainten perässä.

Tämän projektin kannalta vakavin puute Raspberry Pi:ssä oli kuitenkin analogia-digitaalimuuntimen puuttuminen. Toisin kuin Arduinossa, Raspberry Pi:ssä ei yhdessäkään pinnissä ole analogia-konversion mahdollisuutta. Tätä varten piti hankkia erillinen ADC-mikrokontrolleri, joka sitten yhdistettiin Raspberry Pi:hin SPI-sarjaliitännän kautta.

Projektin aikana kokeiltiin pariakin mikrokontrollerivaihtoehtoa, mutta lopulta päädyttiin käyttämään Microchipin MCP3008-mikrokontrolleria, sillä sen käyttöön löytyi erinomainen dokumentaatio Raspberry Pi:n kanssa käytettäväksi. Kuvassa 30 on MCP3008-mikrokontrollerin liitännöiden sijoittelu, ja taulukossa 2 liitännöiden kuvaus.



Kuva 30. MCP3008-mikrokontrollerin pinnien järjestys ja nimet (Microchip Technology Inc. 2008).

MCP3008:n voidaan liittää analogiasisääntulo 8 eri lähteestä. Sen analogia-digitaalikonversio on 10-bittinen, eli analogiasignaali voidaan pilkkoa 2^{10} osaan, jolloin digitaali-päässä signaalin arvo voi saada arvot väliltä 0-1023. MCP3008 vertailee analogista jännitesignaalia referenssijännitteeseen, joka tässä projektissa oli sama kuin mikrokontrollerille Raspberry Pi:lle tuleva syöttöjännite. Vertailujännitteen kuuluisi ideaalitulanteessa olla sama kuin analogiasignaalin maksimiarvo, mutta digitaalista signaalia voidaan käsitellä vastaavan todellista tilannetta, jos näin ei ole.

Taulukko 2. MCP3008-mikrokontrollerin pinnien nimien merkitys (Microchip Technology Inc. 2008).

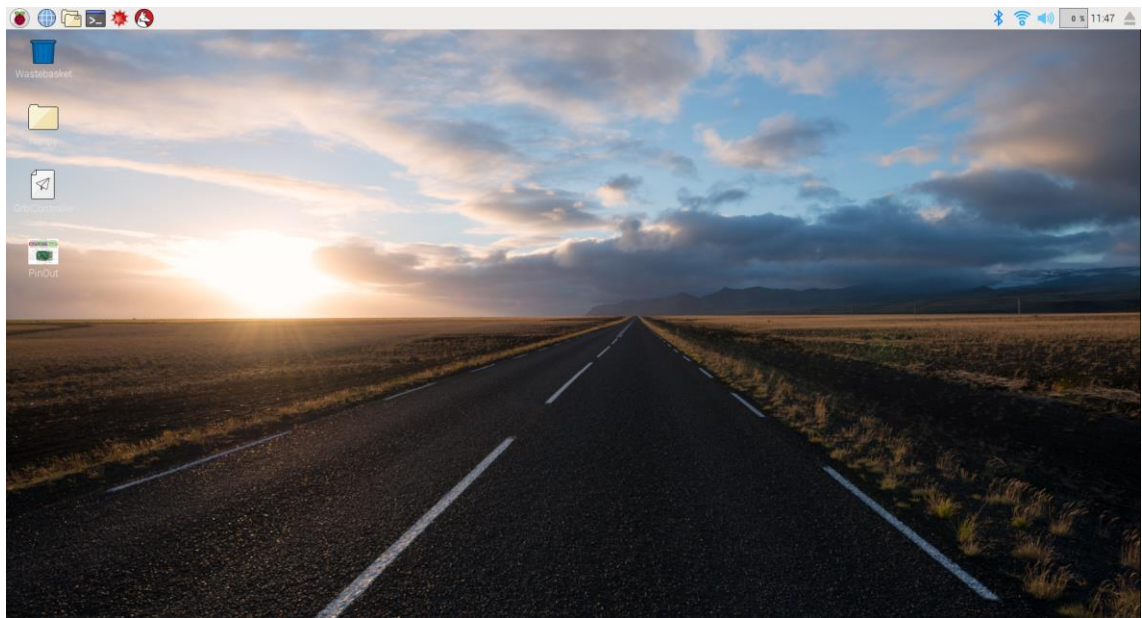
MCP3008	Symbol	Description
PDIP, SOIC		
1	CH0	Analog Input
2	CH1	Analog Input
3	CH2	Analog Input
4	CH3	Analog Input
5	CH4	Analog Input
6	CH5	Analog Input
7	CH6	Analog Input
8	CH7	Analog Input
9	DGND	Digital Ground
10	$\overline{\text{CS}}/\text{SHDN}$	Chip Select/Shutdown Input
11	D _{IN}	Serial Data In
12	D _{OUT}	Serial Data Out
13	CLK	Serial Clock
14	AGND	Analog Ground
15	V _{REF}	Reference Voltage Input
16	V _{DD}	+2.7V to 5.5V Power Supply
–	NC	No Connection

5.5 Käyttöliittymä

Raspberry Pi:hin oli tarjolla useampi erilainen käyttöjärjestelmä toimittamaan graafisen käyttöliittymän virkaa, mikä kopiosorviin välttämättä haluttiin. Raspberry Pi:hin on olemassa useita eri käyttöjärjestelmävaihtoehtoja, joista Raspbian on ”virallinen” Raspberry Pi:n käyttöjärjestelmä. Se on runsaimmin dokumentoitu ja tarjoaa parhaat työkalut ohjelmointia ajatellen, joten sen käyttämiseen oli helppo päätyä.

Raspbian asennetaan erilliselle MicroSD-muistikortille itse tai sitten ostetaan jo luvussa 3.2.2. mainittu NOOBS-paketti, joka on 8 gigatavun muistikortti, johon Raspbian on jo asennettu valmiiksi. Projektia varten ostettiin suuri 16 gigatavun muistikortti, joten asennus suoritettiin itse. Asennus oli helppoa Raspberry Pin nettisivuilla, osoitteessa www.raspberrypi.org, olevien ohjeiden avulla, ja käyttöönotto oli vaivatonta.

Ohjelmointia varten tarvittiin pari pakettia, jotka ladattiin Raspberry Pin Linux-terminaalin kautta, kunhan Internet-yhteys oli ensin muodostettu. Paketit asennettiin komennolla *sudo apt-get install (paketin nimi)*, eikä dokumentoinnin määrässä jälleen ollut valittamista, jos jokin meni vikaan. Python-ohjelmointiin tarvittava ohjelma Raspbianissa on valmiiksi asennettuna.



Kuva 31. Rasbian-käyttöjärjestelmän työpöytänäkymä.

Linux-järjestelmiin tottuneet ovat varmasti kuin kotonaan Raspbianin parissa. Windows-käyttäjälle käyttöliittymä vaati hieman totuttelua. Työpöytänäkymä (kuva 31) on kuitenkin selkeä ja valikkonäkymä looginen, joten opettelu ei ollut vaikeaa. Ainoa haasteellinen osuus oli terminaalin käytön opettelu, Windowsin terminaalia kun harvemmin tarvitsee.

6 KOMPONENTTIEN TESTAUS JA OHJELMOINTI

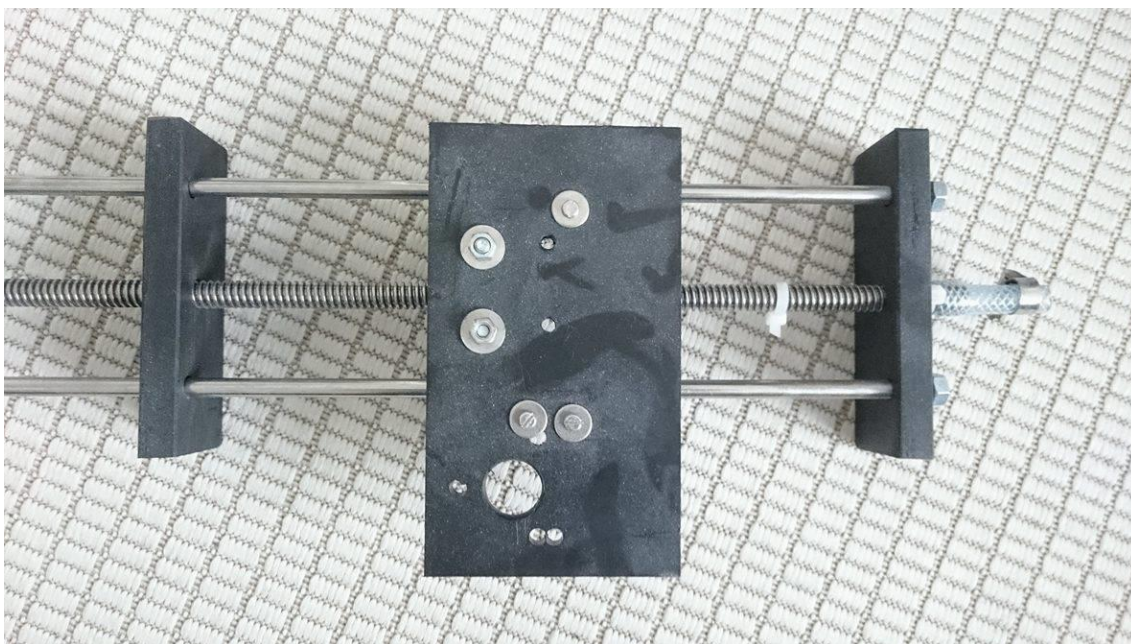
Komponenttien testaus on jatkunut koko kevään 2017, ja jatkuu edelleen. Kunkin komponenttien ohjaamiseen tarvittavia ohjelmapätkiä ei tässä opinnäytetyössä julkaista, koska suunnitelmissa on myydä projektin tuloksena syntyvää laitetta eteenpäin muille yksityishenkilöille.

Raspberry Pin suosion takia Internet on pullollaan neuvoja, ohjeita, tutoriaaleja sekä valmiita ohjelmapätkiä mitä erilaisimpiin projekteihin, mutta välillä komponenttien ohjauksen tuottaminen aiheutti todellista päänvaivaa. Onneksi Python- ja Raspberry Pi -forumeilla on runsaasti asiantuntevia ihmisiä, joilta voi kysyä lähestulkoon mitä tahansa. Hyvänä esimerkkinä tästä projektin aikana lähetettiin kysymys nettipalstalle, kun oma ymmärrys ei enää riittänyt. Vastaus tuli tunnin kuluessa, ja vastauksen avulla ohjaus toimi. Hauskaksi asian tekee se, että kysymys kirjoitettiin kello 11 yöllä.

Avoimen lähdekoodin ohjelmien ja valmiiden moduulien yleinen latauspaikka oli GitHub, jonne yksityishenkilöt ja organisaatiot lataavat tuotoksiaan yleiseen ja avoimeen leveykseen. GitHubista löytää myös valmiiksi mallinnettuja komponentteja esimerkiksi Fritzing-ohjelmaan, joka on kätevä ohjelma johdotusten mallintamiseen.

6.1 Moottoreiden testaus

Askelmoottorien väännön riittämisestä ei oltu aivan varmoja, vaikkakin optimistisia. Moottorin toimintaa päätettiin testata Arduino UNOn ja Arduinon oman askelmoottorikirjaston avulla. Testiä varten rakennettiin mallikelkka, jota liikutti projektissa käytettävä M12x3 -trapetsikierretanko. Testissä käytetty kelkka on kuvattu kuvassa 32. Kelkkaan asetettiin 2,5 levypaino, joka löytyi kotioloista käsipainosarjan takia. Osoittautui, että askelmoottorilla ei ollut minkäänlaista ongelmaa liikutella sellaista kuormaa edes takaisin, vaikutti jopa siltä, että kuormaa olisi voinut hyvin lisätä. Koska kelkan päälle tuleva rasite tulle suurimmaksi osin työkalulta, joka painaa 1,1 kg, arvioitiin näiden askelmoottorien olevan riittävän voimakkaita tähän projektiin.



Kuva 32. Askelmoottorin testaukseen käytetty testikelkka. Askelmoottorin akseli tuli kiinni oikeassa reunassa näkyvään letkunpätkään kiristimellä.

Pythonilla moottoreiden ohjaus ohjelmallisesti oli melko yksinkertaista, sillä moottorien ohjainlaatikot hoitivat käämien virroituksen tahdistuksen. Ohjelmaan piti vain kirjoittaa, mitkä GPIO-pinnit olivat suuntatiedon ja pulssitiedon käytössä, ja määrittellä viive pulssien välille. Viiveen määrittely oli melko tarkkaa puuhaa, sillä liian pieni viive johti askelten hukkaamiseen ja toimintakyvyn heikkenemiseen. Joillain arvoilla askelmoottorien akselit eivät edes pyörineet. Liian suuren viiveen asettaminen taas ei ollut järkevää, sillä askelmoottorit eivät toimisi optimaalisesti. Onneksi kuitenkin nettisivuilta löytyi ohjeita oikean viiveen asettamiseen askelmoottorityypistä riippuen, sekä valmiita ohjelmapätkiä, kun omat koodaustaidot eivät riittäneet.

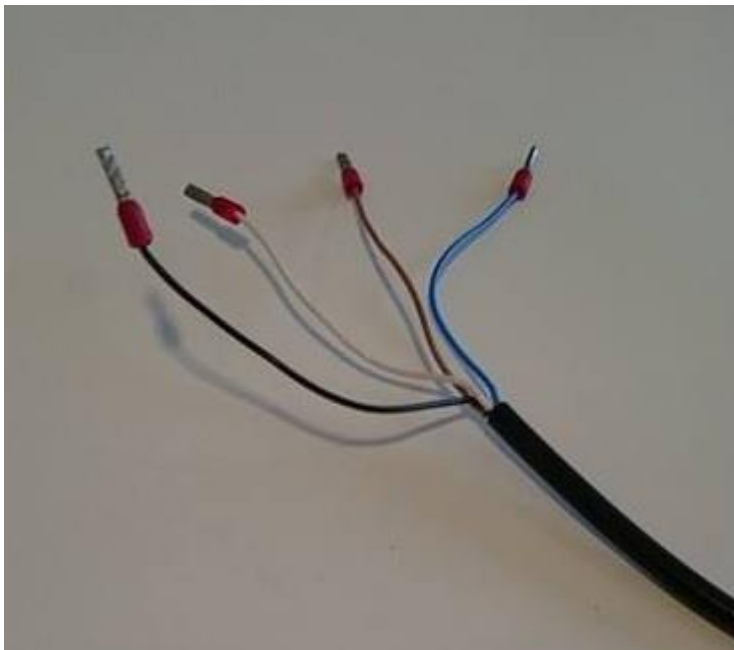
Ohjelmaan määriteltiin pulssien eli askelten määrä, joilla x-akselin suunnassa kelkkaa liikuttava askelmoottori sekä kappaletta pyörittävä askelmoottori tekisivät haluttuja asioita laseranturin mittapisteiden tallennusta eli skannausta ajatellen. Kappaletta pyörittävä moottori määriteltiin pysähtymään 10 kertaa yhden kokonaisen pyörähdyksen aikana, ja jokaisella kerralla tallennettaisiin laseranturin arvo y-akselin koordinaatiksi myöhemmin tapahtuvaa kopiointia varten. Pysäytykset toteutettiin *for*-loopilla *for i in xrange(0, mittauspisteet)*, missä mittauspisteet-muuttujalle annettiin arvo 10. Mittauspisteiden määrä pyörähdykselle olisi näin muutettavissa, jos todettaisiin kopiointilaadun olevan liian ”kulmikas”. Tämä tosin vaatisi askelmäärän uudelleenkalibrointia.

X-akselin suunnassa kelkkaa liikuttavan moottorin ohjaus taas liikkui pisteiden ottamisen jälkeen tietyn matkan eteenpäin, ja pisteiden kerääminen toistuisi. Näin toimittaisiin, kunnes koko kopioitava kappale olisi käyty läpi. Kappaleen molemmin puolin, kiinteästi pidikkeissä kiinni olisi rajakytkimet, jotka aloittaisivat ja lopettaisivat skannauksen ennen kuin kelkka osuisi johonkin. Tätä varten ohjelmaan rakennettiin *if*-ehtolauserakenne.

Grbl:n kanssa testaaminen on vielä alkuasteella. Grbl Controller on onnistunut suorittamaan läpi itse kirjoitetun .nc loppuisen tiedoston, joka sisälsi vain liikkumiskäskyjä xy-koordinaatistossa. Se liikutteli askelmoottoreita asianmukaisessa järjestyksessä. Ensimmäiset kokeilut skannaustiedoston kanssa eivät sujuneet hyvin, mutta ongelma oli luultavasti huonosti kirjoitettu G-koodin alkumäärittely.

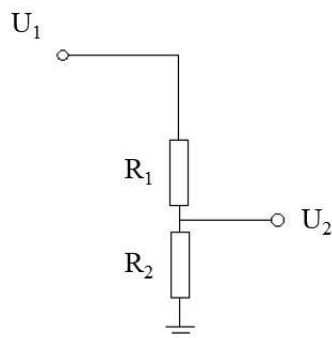
6.2 Laseranturin testaus ja MCP3008-mikrosirun ohjelmointi

Laseranturin johdossa (kuva 33) oli 4 johtoa, joista 2 oli varattu anturin virroittamiseen ja toiset 2 eri signaalityypeille. Tässä projektissa tarvittiin analogiasignaaliin kykenevää valkoista johtoa, joka liitettäisiin ADC-siruun signaalinmuunnosta varten.



Kuva 33. Anturin virtajohdon päät. Sininen ja ruskea ovat anturin virroittamiseen, musta signaalikaapeli kappaleen tunnistustoimintoa varten. Valkoinen johto toimii sekä analogiatilassa että tunnistamismoodissa.

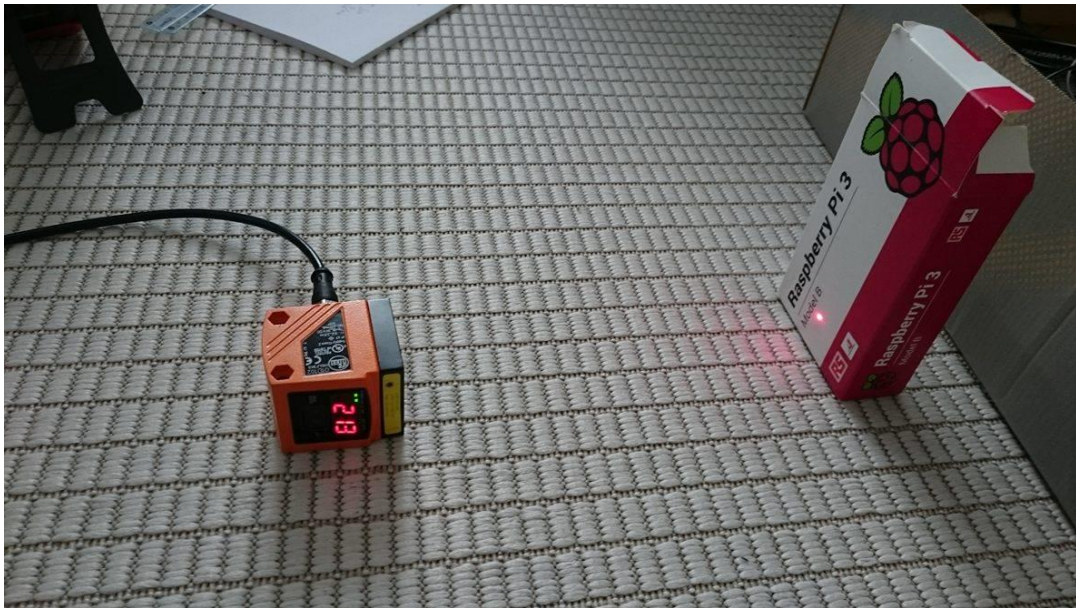
Anturin analogiasignaalksi valittiin jännitevaihtoehto, joka siis tuotti 0-10 voltin jännitteen annettujen mittausalueen minimi- ja maksimiarvojen mukaan. Signaalin jännite piti kuitenkin alentaa, koska sekä ensitesteissä käytetty Arduino että MCP3008 toimivat vain välillä 0-5 V. Signaaliarvot puolitettiin jännitteenjakokaavaa $U_2 = \frac{R_2}{R_1 + R_2} \cdot U_1$ hyödyntäen (Mäkelä, M., Soininen, Tuomola & Öistämö 2012, 121), jonka periaatetta on kuvattu kuvassa 34.



Kuva 34. Jännitteenjakokaavan periaatekuva.

Kaksi sopivan suuruista vastusta asetettiin sarjaan, ja niiden välistä otettiin jännitesignaali eteenpäin mikrokontrollerille. Arduinolle tämä toimi sellaisenaan, mutta MCP3008-mikrokontrollerille jännitesignaali alennettiin aina 2,5 V asti, koska toisin kuin Arduinossa, Raspberry Pin GPIO-pinnit eivät input-tilassa kestä kuin maksimissaan 3,3 voltin jännitteen. Tästä syystä myös MCP3008:lle tuleva vertailujännite Raspberry Piltä on myös 3,3 V. Jännite-ero oli toki kompensoitavissa MCP3008:n ohjelmoinnissa, kun digitaalinen maksimiarvo laskettaisiin uudelleen ja otettaisiin huomioon koodissa $\left(\frac{2,5 V}{3,3 V} \cdot 1023 = 775\right)$, mutta optimitilanteessa signaalin maksimiarvo ja vertailujännite olisivat samat.

Laseranturilta ohjelmallisesti saatu mittatieto oli verrattavissa anturin nestekidenäytön lukemaan (kuva 35). Aluksi ohjelmallisesti tuotettu arvo antoi mielivaltaisia lukemia, ja syyn selvittäminen tuotti paljon päänvaivaa. Ratkaisu löytyi lopulta maadoituksista, kun MCP3008:n analoginen ja digitaalinen maadoitus yhdistettiin Raspberry Pin maadoitukseen ja virtalähteen miinukseen. Tämän jälkeen luvut ohjelmaa ja nestekidenäytön välillä olivat yhtenevät.



Kuva 35. Laseretäisyysanturin ensimmäisiä kokeilukertoja. Anturin nestekidenäytön arvo sekä ohjelmallisesti saatu arvo olivat hyvin koherentit.

MCP3008:n ohjelmoinnissa hyödynnettiin Adafruitin (www.adafruit.com) valmista kirjastoa, sillä omat ohjelmointitaidot eivät riittäneet mikrokontrollerilta tulevan bittitiedon käsittelyyn, joista MCP3008 datalehdellä kerrotaan (Microchip Technology Inc. 2008).

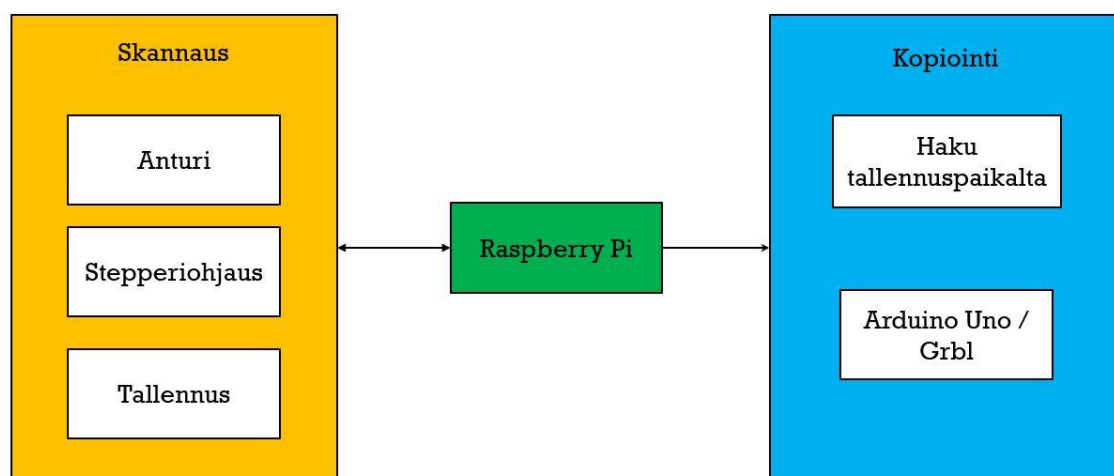
Kuten luvussa 5.4. mainittiin, MCP3008 kytketään kiinni Raspberry Pin SPI-sarjaliitännätpinneihin. SPI ei ole Raspberry Pi:ssä oletusarvoisesti käytössä, vaan se pitää itse käydä asetuksista laittamassa päälle. MCP3008-mikrokontrollerissa ei ole merkintöjä eri liitännöjen sijainnista, vaan ne pitää tarkistaa mikrokontrollerin datalehdeltä. Muistamista helpottaa pieni lovi mikrokontrollerin toisessa päässä. Tämän jälkeen MCP3008 kytketään kiinni Raspberry Pi:hin kuvan 36 mukaisesti:

- MCP3008 V_{REF} ja V_{DD} kiinni Raspberry Pin 3,3 jännilähteeseen (ruskea)
 - MCP3008 AGND ja DGND (analogia- ja digitaalimaadoitus) kiinni sekä Raspberry Pin maadoitukseen että yhdessä anturisignaalin maadoituksen kanssa kiinni virtalähteen miinukseen (musta)
 - MCP3008 CLK kiinni Raspberry Pin CLK/GPIO11-pinniin (punainen)
 - MCP3008 D_{OUT} kiinni Raspberry Pin MISO/GPIO21-pinniin (violetti)
 - MCP3008 D_{IN} kiinni Raspberry Pin MOSI/GPIO10-pinniin (oranssi)
 - MCP3008 CS/SHDN kiinni Raspberry Pin CE0/GPIO8-pinniin (keltainen).
- (DiCola 2016.)

Tiedostojen nimeäminen osoittautui hankalaksi, sillä yllä esitetyllä tavalla jokaisella kopiointikerralla täytyy tiedostonimi käydä vaihtamassa ohjelman sisällä. Tämän takia ohjelmaan tehtiin erillinen *nimi = raw_input("Syötä tiedostonimi:")* -komento, joka siis pyytää käyttäjää syöttämään nimen kopioitavan kappaleen pistedatatieostolle. *Open*-komennon käyttäminen ei näin kuitenkaan suoraan, vaan ensin tiedostopolusta, -nimestä ja -päätteestä tehdään yhtenäinen string-muuttuja, joka sitten annetaan *open*-komennon sulkujen sisään yhdessä tiedostomoodin kanssa. Tallennuksen toimivuutta testattiin ensin normaalina tekstitiedostona, jonka jälkeen siirryttiin käyttämään Grbl-ohjaimen vaatimaa .nc-päätettä.

7 TULOKSET JA JATKOSUUNNITELMAT

Tuloksista mainittakoon ensimmäisenä, että kopiosorvi ei ole vielä valmis. Mekaniikka on mallinnettu, mutta kokoonpanoa ei ole vielä suoritettu. Projekti siis jatkuu opinnäytetyön tekemisen jälkeenkin. Ohjelmallisesti erilaiset ohjelman palaset ovat toimintakykyisiä, mutta palasten yhdistämisessä kokonaisuudeksi on tuottanut jonkin verran ongelmia. Toimintalogiikka on kuitenkin ja komponenttien tehtävät ovat kuitenkin selkeästi mielessä (kuva 37).

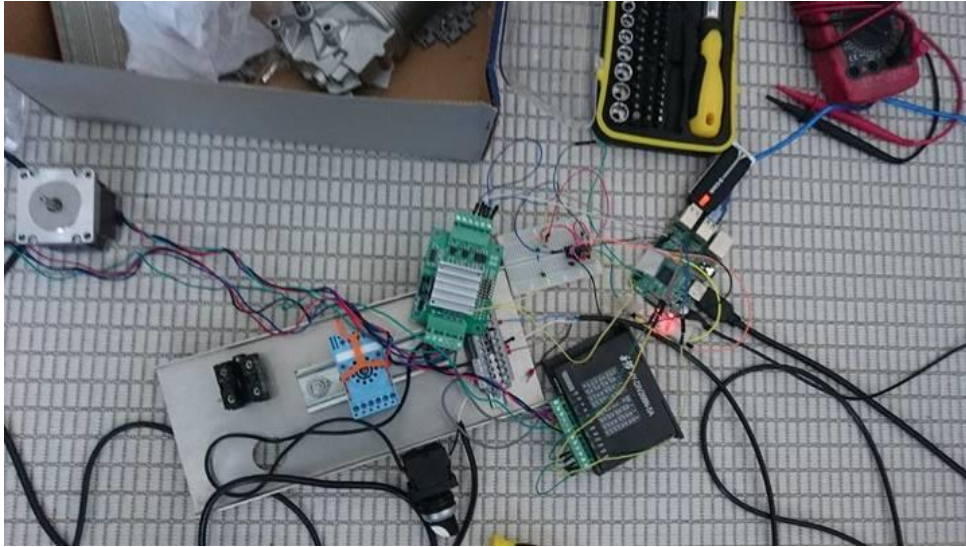


Kuva 37. Kopiosorvin eri toimintojen havainnollistamista, ja mitkä komponentit toimintoihin osallistuvat

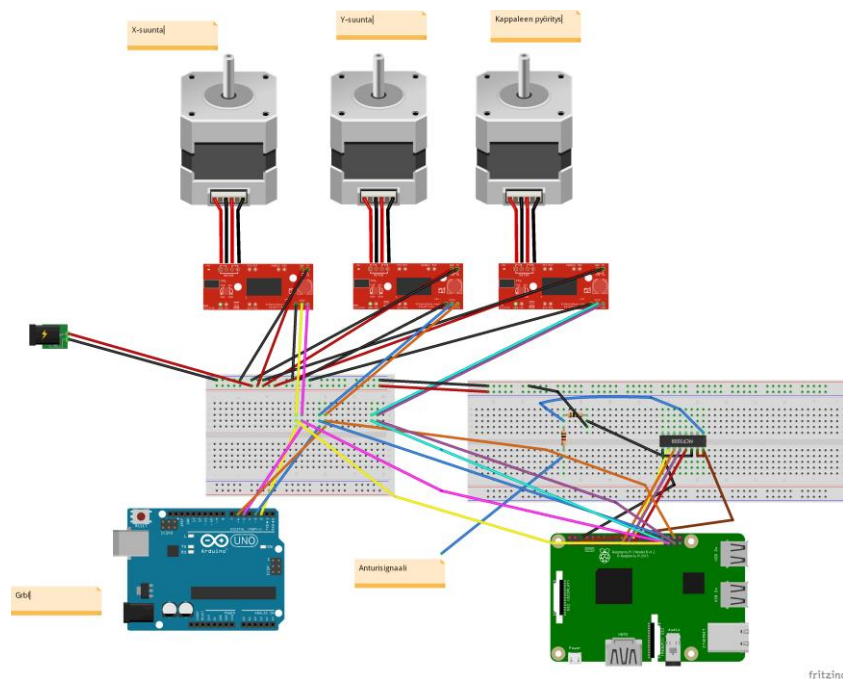
Mitä aikataulutukseen tulee, on kopiosorviprojekti edennyt hyvin hitaasti, ottaen huomioon, että se alkoi jo syksyllä 2015. Rakentamisaikataulua ovat kuitenkin hidastaneet erinäiset muutokset, kuten kopiointitavan muuttuminen mekaanisesta laseranturia hyödyntäväksi, sekä ilmenneet puutteet kuten Raspberry Pin ADC-ominaisuuden puuttuminen. Myös opinnäytetyön kirjoittamisaikataulu ei ole pitänyt paikkansa, vaan se on tehty uudelleen kolmesti. Tulevaisuutta ajatellen ajankäytön hallinnassa on siis merkittävästi parannettavaa.

Projektin parissa työskentelyn opettavaisuutta ei kuitenkaan voi kiistää. Työssä on jouduttu soveltamaan paljon opintojen aikana tutuksi tulleita asioita, ja Python kielen opettelu ja erinäisten komponenttien sekä ohjelmakoodien yhteensovittaminen kuuluvat selkeästi automaatiotekniikan piiriin.

Projektin sähköisten komponenttien määrän kasvaessa johtoja risteilee sinne tänne, kuten kuvassa 38. Ongelmaan ei kuitenkaan ole välitöntä ratkaisua, joten projektia varten tyydyttiin piirtämään Fritzing-ohjelmalla omaan käyttöön johdotuskuva (kuva 39) muistin-virkistykseksi, jotta pinniosoitteita ei tarvitsisi kaivaa joka kerta eri lähteistä. Tämä kuva on myös liitteenä 4 isommassa koossa eri komponenttien hahmottamiseksi. Askelmootoreiden ohjaimina käytetään kuvassa EasyDriver-mikrokontrollereita, jotka toiminnaltaan vastaavat projektissa käytettyjä ohjaimia.



Kuva 38. Testausta kahden askelmootorin, anturin, Raspberry Pin ja Grbl-kirjaston sisältävän Arduino UNOn kanssa. Johtojen määrä on melkoinen.



Kuva 39. Fritzing-kuva johdotuksesta. Hahmottaminen on edelleen vaikeaa.

Kopiosorvin markkinoinnin suhteen ollaan varovaisen toiveikkaita. Suomen nettiforumeilla ei juuri tässä projektissa hyödynnettyä lasertekniikkaa nähdä, ja sen oletetaan olevan suurin kilpailuetu. Lisäksi projektissa tuotetun kopiosorvin hinta tulee olemaan sellainen, että muiden lasertekniikkaa hyödyntävien sorvien markkinoilla kilpailuetu on todellinen, sillä niiden hinnat liikkuvat tuhansissa euroissa. Valmiin kopiosorvin kustannukset tulevat liikkumaan reilun tuhannen euron tietämissä, kuten taulukon 3 kustannuslaskelmasta voidaan havaita.

Taulukko 3. Kopiosorviprojektin tähänastiset kustannukset

Osa	Hinta (€)	Huom.
Spindle	139	
Askelmoottorit	72,68	
Laseranturi	209,25	
6 m alumiiniprofilia	ilmainen	
Askelmoottorien ohjaimet	31,83	
Sakarakytkimet	18,57	
Laakerit	21,4	
Laserleikkeet	20	
Huonekaluputki	---	Ehkä ilmainen
Pleksit	---	Ei hankittu
Muoviosat	---	Ei hankittu
Kosketusnäyttö	128,41	
Arduino	9,44	
Raspberry Pi	49,9	Kaverin oma
Käynnistys/stopnapit	---	Ei hankittu
Kappaleen pitimet yht.	77,29	
Kelkat	---	Ei hankittu
Piuhat	19	
Imuriletku/alunen	---	Ei hankittu
Kahvat	---	Ei hankittu
MCP3008	7	
Johdetarvikkeet	102,92	
Yht. €	906,69	

Liitteessä 1 olevan vaatimusluettelon tavoitteita tarkastellessa osa tavoitteista on saavutettu, osa taas vaatii vielä kehitystyötä. Vaatimukset ovat toteutuneet seuraavasti:

- Suurin osa komponenteista täyttää ehdottomasti vaatimukset edullisuudesta ja keveydestä, tosin tärkeimpien komponenttien, kuten laseranturin ja sorvaustyökalun kohdalla, edullisuudesta oltiin valmiita tinkimään. Ne saatiin kuitenkin yllättävän edullisesti.
- Automatisointivaatimukset on täytetty, ja tulevaisuudessa automatisointia voidaan kehittää pidemmälle, jos sille tulee tarvetta.
- Turvallisen käytön suhteen vaatimuksien täyttymistä ei vielä kunnolla voida sanoa, sillä niiden täyttymistä voidaan arvioida vasta prototyypin valmistuttua. Suunniteltujen turvallisuusratkaisujen arvioidaan kuitenkin riittävän täyttämään asetetut vaatimukset.
- Raspbian-käyttöjärjestelmä on hetken opetteluun jälkeen erittäin helppokäyttöinen, ja tallennetut tiedostot löytyvät vaivatta, joten käyttäjäystävällisyyteen liitetyt vaatimukset täyttyivät osittain. Kosketusnäyttöä ei vielä ole, mutta sen hankkimista harkitaan, kuten myöhemmin tässä luvussa ilmenee.
- Mekaanisen kokoonpanon helppoudessa ei aivan päästy tavoitteisiin, mutta koneistettavia osia on niin vähän, että tämän ei katsottu olevan haitta.
- Markkinoille kopiosorvi ei ole vielä ole, mutta kuten edellä todettiin, sen menestymisestä ollaan toiveikkaita. Hinta on tällaisen tason sorville oikein edullinen, ja markkinointia varten on hyödynnettävissä erilaisia forumeita ja sosiaalista mediaa. Opinnäytetyön aikana ihmiset ovat osoittaneet kiinnostustaan laitetta kohtaan, joten kysyntääkin laitteelle on.
- Vaapun kiinnityksen suunnittelu on vielä kesken, joten vaatimusta ei tähän mennessä ole saavutettu, mutta toteutuksen oletetaan olevan toimiva.

Tulevaisuuden kirkkain tavoite on saada mekaaninen prototyyppi kasaan, jotta kopiosorvin toimintaa voi testata käytännössä. Ensin testataan moottoreiden ja anturin yhteistyötä, jonka jälkeen tuodaan työkalu osaksi testiprosessia. Kun kaikki näyttäisi toimivan kuten on suunniteltu, ryhdytään testaamaan varsinaisen työstön toimivuutta, ja tehdään parannuksia ohjelmointiin testitulosten mukaisesti.

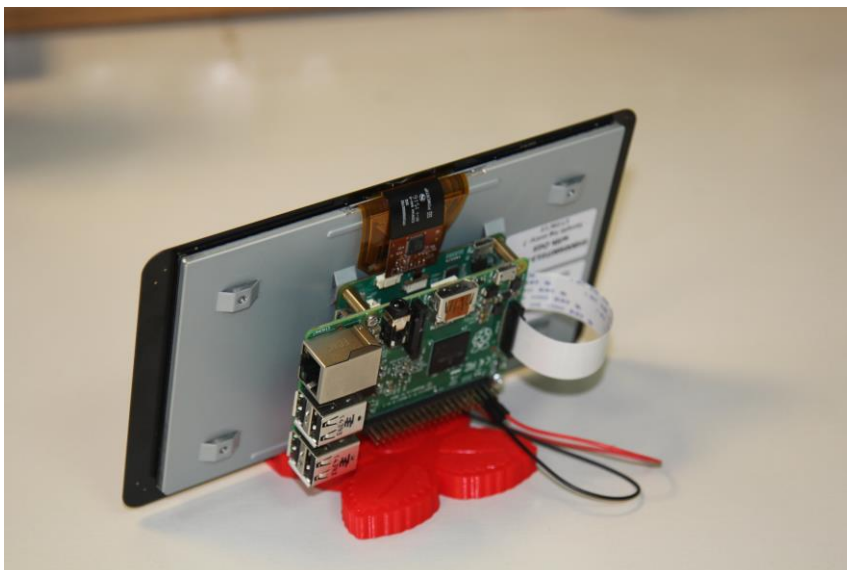
Kopiosorvin käyttöliittymä halutaan pitää graafisena, mutta tähän asti Raspberry Pin näyttönä on käytetty televisiota. Kopiosorvin näyttöratkaisuksi potentiaalisin on Raspberry Pin viralliseksi näytöksi nimetty 7 tuuman kapasitiivinen kosketusnäyttö (kuvat 40

ja 41). Se voidaan liittää Raspberry Piin DPI (Display Parallel Interface) - ja DSI (Display Serial Interface) -tekniikoita hyödyntäen. Näytön ominaisuudet ovat seuraavat:

- 800x480 RGB resoluutio 60 Hz virkistystaajuudella
- 24-bittinen väripaletti
- 10 pisteen kapasitiivinen kosketusnäyttö
- 70 asteen katselukulma
- metallinen tausta, jossa kiinnityskohdat Raspberry Piä varten. (Hollingworth 2015).



Kuva 40. Virallinen 7” Raspberry Pi -kosketusnäyttö (Hollingworth 2015)



Kuva 41. Kosketusnäyttö takaapäin. Havainnollistava kuva siitä, miten Raspberry Pi voidaan kiinnittää näytön taakse käsikäyttöisiä sovelluksia silmällä pitäen. (Hollingworth 2015)

Taustan kiinnitystapa voisi mahdollistaa sen, että näyttöä voisi käyttää kuin käsiohjainta robotiikassa Bluetooth- tai Wifi-yhteyden avulla. Ennen kaikkea näyttö kuitenkin palvelisi paremmin kopiosorvin käytössä kuin nykyinen järjestely, sillä kopiosorvin käyttäjät tuskin haluavat raahata televisiota tai tietokoneen näyttöä mukanaan kopiosorvin kanssa.

Vaapun kiinnittäminen kopioitavaksi on vielä suunnittelupöydällä. Vaappu haluttaisiin kiinnittää koukkulenkkeistään, mutta sopivaa ratkaisua ei tunnu löytyvän. Tämän hetkinen suunnitelma on tehdä oma viritys 8 mm metallitangon pätkästä, jonka saisi toisesta päästä kiinni vastaavanlaiseen kiristysmutteriin kuin kopiosorvin työkalussakin on, ja toiseen päähän tehdään poran ja metallisahan avustuksella kiristämällä sulkeutuva mekanismi, jonka inspiraationa on toiminut perhovieheiden tekoon tarkoitettu teline. Työstettävän puukappaleen kiinnitys puolestaan tapahtuisi piikein.

LÄHTEET

Agnihotri, N. 2012. Stepper Motors or Step Motors. Luettu 8.5.2017. <https://www.engineersgarage.com/articles/stepper-motors>

Arduino. 2017. Arduino UNO. Luettu 6.5.2017. <http://www.arduino.org/products/boards/arduino-uno>

Arduino. 2017. What is Arduino? Luettu 6.5.2017. <https://www.arduino.cc/en/Guide/Introduction>

Carrier, J. E., Ohline, R. M. & Kenny, T. W. 2011. Introduction to Mechatronic Design (International ed.). Boston, Mass.: Pearson.

Daware, K. 2016. Basics Of Stepper Motor. Luettu 8.5.2017. <http://www.electrical-leasy.com/2014/10/stepper-motor.html>

DiCola, T. 2016. MCP3008. Päivitetty 2.9.2016. Luettu 9.5.2017. <https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008>

Electrical Technology. 2016. Brushless DC Motor: Construction, Working Principle & Applications. Päivitetty 27.5.2016. Luettu 8.5.2017. <http://www.electricaltechnology.org/2016/05/bldc-brushless-dc-motor-construction-working-principle.html>

Github, Inc. 2017. Grbl: Home. Päivitetty 1.5.2017. Luettu 9.5.2017. <https://github.com/gnea/grbl/wiki>

Github, Inc. 2017. Grbl v1.1 Commands. Päivitetty 11.3.2017. Luettu 9.5.2017. <https://github.com/gnea/grbl/wiki/Grbl-v1.1-Commands>

Hollingworth, G. 2015. The eagerly awaited Raspberry Pi display. Päivitetty 8.9.2015. Luettu 9.5.2017. <https://www.raspberrypi.org/blog/the-eagerly-awaited-raspberry-pi-display/>

Ihalainen, E., Aaltonen, K., Aromäki, M. & Sihvonen, P. 2011. Valmistustekniikka. 14.painos. Helsinki: Hakapaino Oy

Ifm Electronic Oy. 01D102-anturin datalehti. Päivitetty 13.2.2012. Luettu 10.5.2017. <http://www.ifm.com/products/fin/ds/O1D102.htm>

Jameco Electronics. 2017. Raspberry Pi pinout diagram: circuit notes. Luettu 9.5.2017. <http://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>

Kuphaldt, T. R. 2017. Lessons in Electric Circuits, volume III, Chapter 11: DC Motor Drives: Pulse Width Modulation. Luettu 8.5.2017. <https://www.allaboutcircuits.com/textbook/semiconductors/chpt-11/pulse-width-modulation/>

Künzel, G. 2005. The Mechatronics Design Process. Julkaistu 14.9.2005. Luettu 20.5.2017. http://www.slpk.sk/eldo/2005/010_05/13.pdf

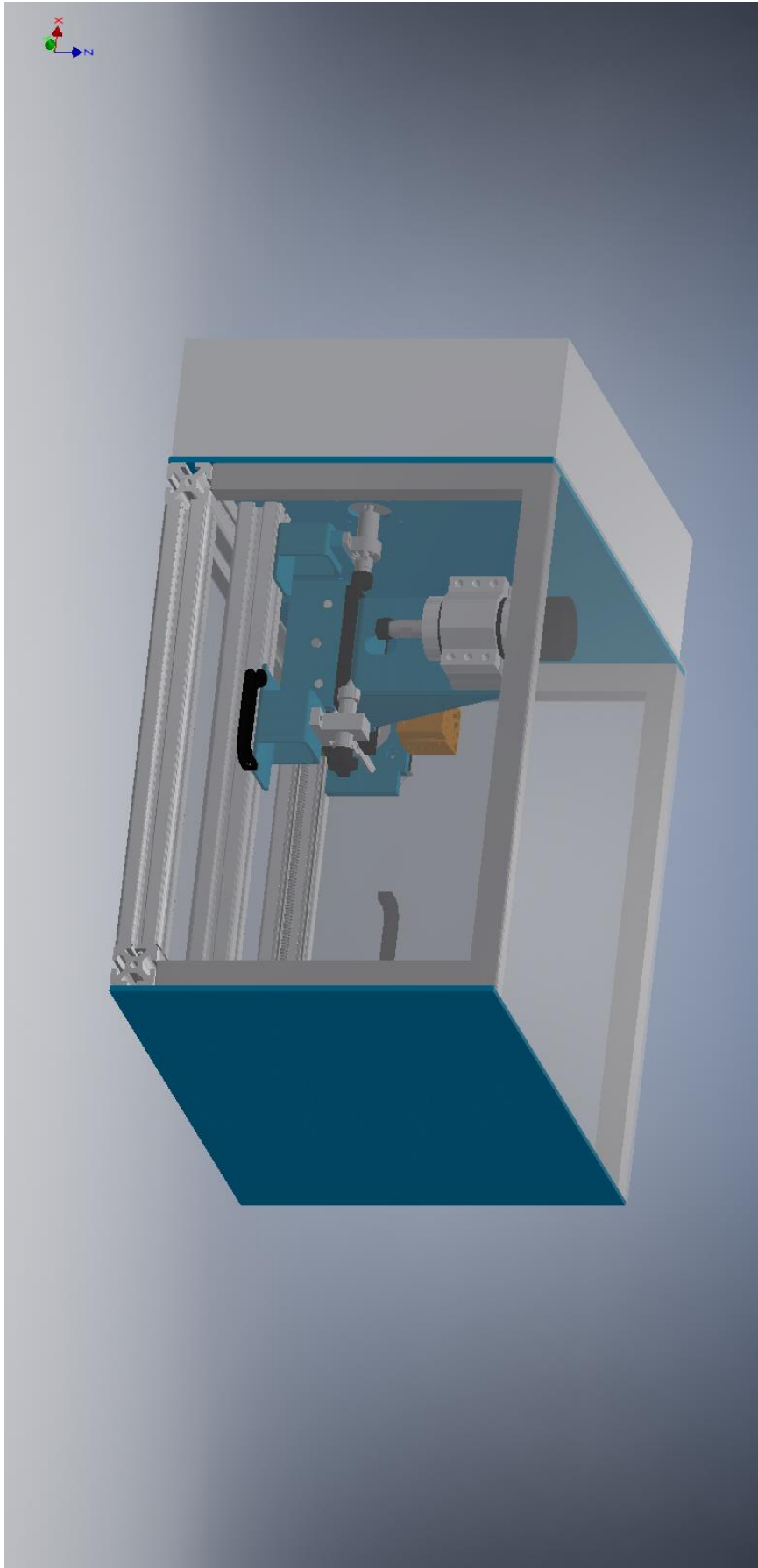
- Mathur, N. 2012. Mechatronics – Five Design Challenges and Solutions for Machine Builders. Julkaistu 1.6.2012. Luettu 6.5.2017. <http://www.ni.com/newsletter/50145/en/>
- Microchip Technology Inc. 2008. MCP30043008 – datasheet. Päivitetty 3.12.2008. Luettu 9.5.2017. <http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>
- Mäkelä, M., Soininen, L., Tuomola, S. & Öistämö, J. 2012. Tekniikan kaavasto. 10. painos. Tampere: Tammertekniikka
- Mäkelä, S. 2014. Anturitekniikka ja koneautomaation komponentit. Anturitekniikka ja koneautomaation komponentit.pptx. Muokattu 27.1.2014. Tuntiopettaja.
- NC State University. 2015. What is Mechatronics? Päivitetty 12.11.2015. Luettu 6.5.2017. <http://www.engr.ncsu.edu/mechatronics/what-mech.php>
- Precision Microdrives Limited. 2017. AB-022 : PWM Frequency For Linear Motion Control. Luettu 23.5.2017. <https://www.precisionmicrodrives.com/application-notes/ab-022-pwm-frequency-for-linear-motion-control>
- Python Software Foundation. 2017. An Informal Introduction to Python. Päivitetty 26.3.2017. Luettu 7.5.2017. <https://docs.python.org/3/tutorial/introduction.html>
- Python Software Foundation. 2017. Modules. Päivitetty 26.3.2017. Luettu 8.5.2017. <https://docs.python.org/3/tutorial/modules.html>
- Python Software Foundation. 2017. More Control Flow Tools. Päivitetty 4.5.2017. Luettu 8.5.2017. <https://docs.python.org/3/tutorial/controlflow.html>
- Python Software Foundation. 2017. Whetting Your Appetite. Päivitetty 26.3.2017. Luettu 7.5.2017. <https://docs.python.org/3/tutorial/appetite.html>
- Raja, D. 2017. DC Motor Control with Raspberry Pi. Luettu 23.5.2017. <https://circuitdigest.com/microcontroller-projects/controlling-dc-motor-using-raspberry-pi>
- Raspberry Pi Foundation. 2016. Raspberry Pi Model 3 B. Luettu 6.5.2017. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Red Hat. 2017. What is a Raspberry Pi? Luettu 7.5.2017. <https://opensource.com/resources/what-raspberry-pi>
- The Python Book. 2016. 4. painos. Bournemouth, Iso-Britannia: Future Publishing Ltd. Lehti.
- Vaappusorvi. 2015. Kuvakaappaus vaappusorvin toiminnasta. Tekijä: Lauri M. YouTube 2015. Katsottu 9.5.2017. <https://www.youtube.com/watch?v=mpQfdJ9VO24>

LIITTEET

Liite 1. Vaatimusluettelo

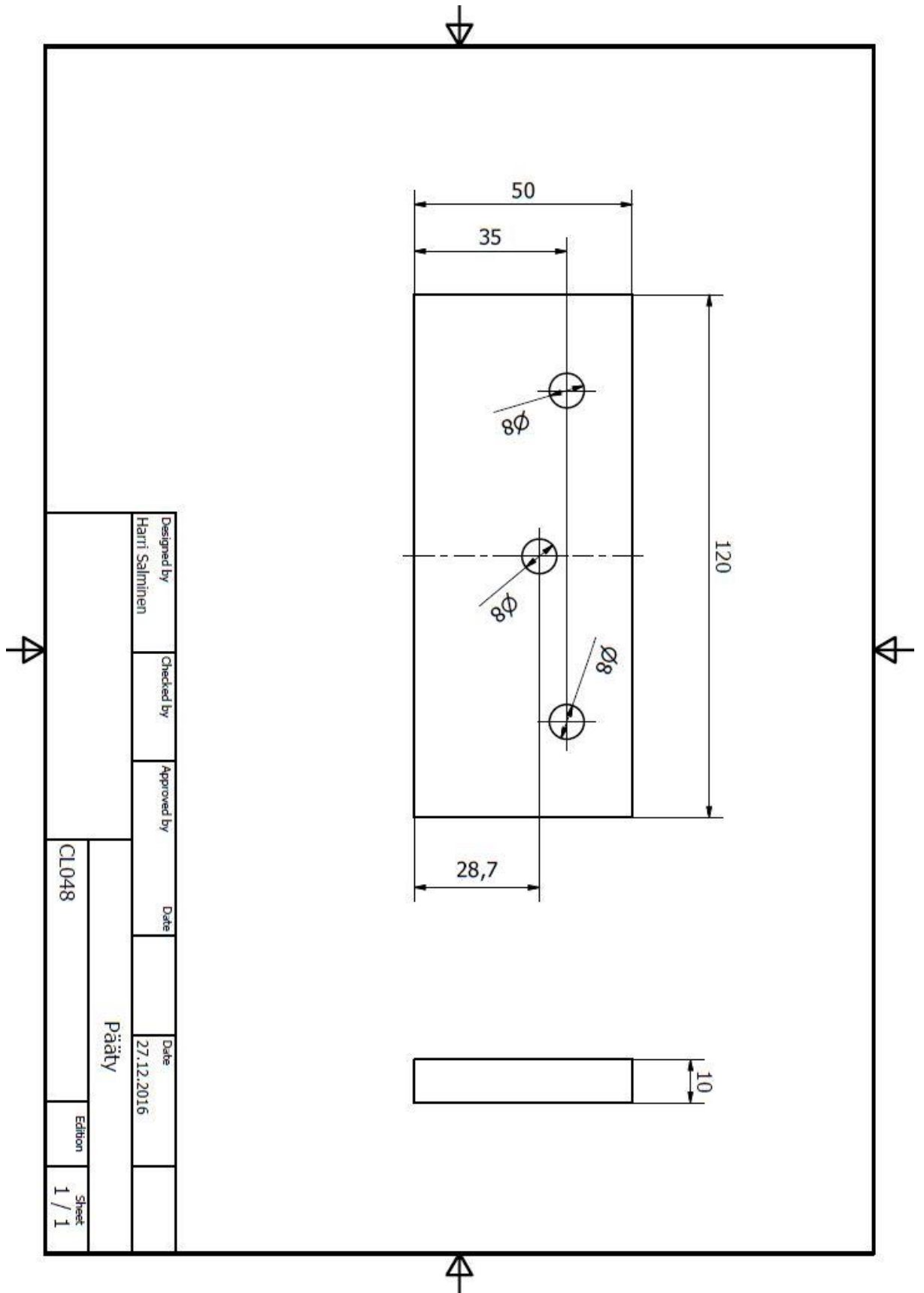
1. Edullinen
 - komponentit mahdollisimman edullisia, mutta toimivia
2. Kevyt
 - pienikokoinen, yhden ihmisen liikuteltavissa
 - mahtuu pieneen tilaan
3. Pitkälle automatisoitu
 - anturoitu
 - kopiointi ohjelmallinen
4. Turvallinen käyttö
 - rajakytkimet työkalulle
 - magneettikytkimet luukkuihin
 - hätäseis-kytkin
5. Käyttäjäystävällinen
 - graafinen käyttöliittymä
 - helppotajuinen toiminta
 - kosketusnäyttö
6. Helppo mekaaninen kokoonpano
 - ei paljon koneistettavia osia
7. Markkinoitavissa
 - kilpailukykyinen hinta
 - valmiit kauppapaikat Internetissä
 - kysyntää oltava/luotava
8. Vaapun kiinnittäminen helppoa
 - kiinnitys tapahduttava vaapun lenkeistä

Liite 2. Kopiosorvin 3D-malli

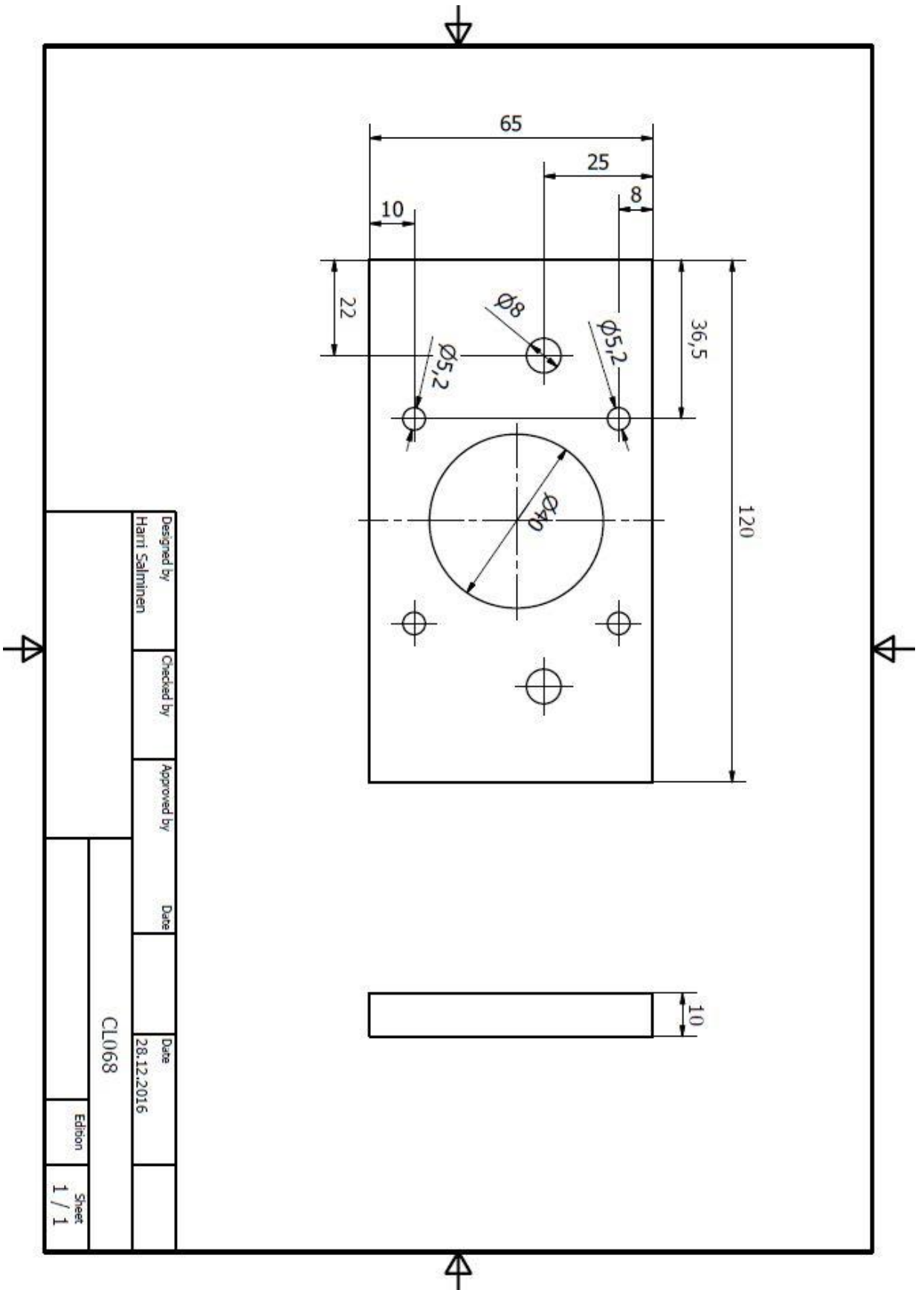


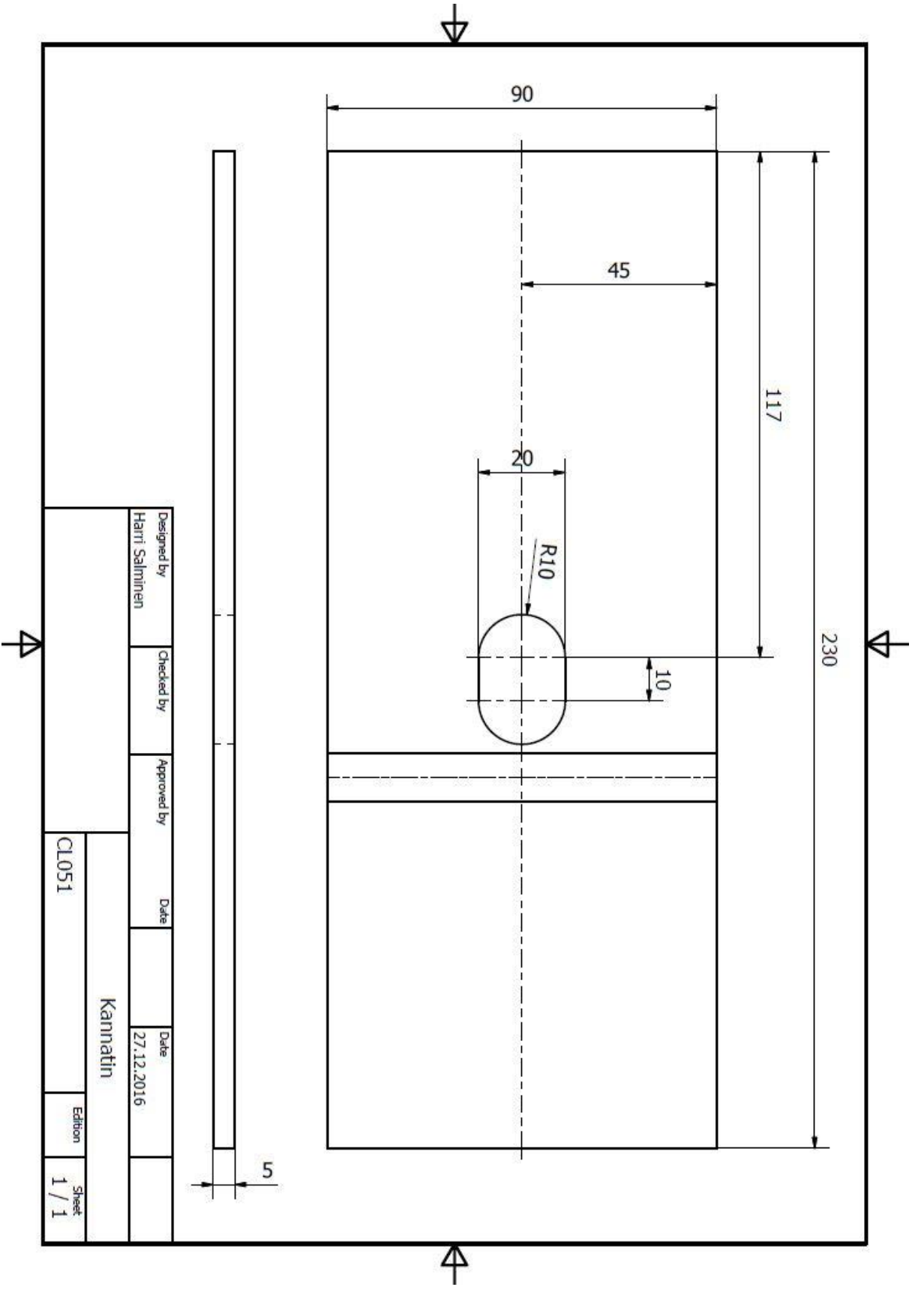
Liite 3. Laserleikkeiden valmistuspiirustukset

1 (4)



(Jatkuu)





Liite 4. Kopiosorvin komponenttien johdotusten Fritzing-kuva

