



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Petri Sinisalo

# Testityökalun suunnittelu ja toteutus panoraamaröntgenlaitteen terävyysalueen määrittämiseen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja Automaatiotekniikka

Insinöörityö

20.3.2019

Tekijä Otsikko	Petri Sinisalo Testityökalun suunnittelu ja toteutus panoraamaröntgenlaitteen terävyysalueen määrittämiseen
Sivumäärä Aika	43 sivua 20.3.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja Automaatiotekniikka
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Verification Engineer Janne Hirsimäki Lehtori Kristian Junno
<p>Tämän insinööritöön tavoitteena oli automatisoida panoraamaröntgenlaitteen terävän kerroksen määrittämisessä tarvittava viuhkafantomien liikuttaminen. Toteutetun laitteen avulla korvattiin aiemmin käytössä ollut manuaalinen ja epätarkka liikutuslaitteisto.</p> <p>Työssä suunniteltiin ja toteutettiin panoraamaröntgenlaitteen leukatukeen kiinnitettävä testilaitte, johon voidaan kiinnittää testifantomi ja jota voidaan liikuttaa tarkasti syvyysuunnassa vaaditulla tarkkuudella. Työssä myös tutustuttiin Raspberry Pi-tietokoneen toimintaan ja soveltuvuuteen mittausten automatisoinnissa.</p> <p>Testilaitteen suunnittelussa tutustuttiin ensin Raspberry Pi-tietokoneen toimintaan ja sen IO-porttien käyttöönottoon. Työssä opiskeltiin askelmoottorin toiminta, ohjaus sekä eri moottorityypit. Testilaitteeseen käytettävät osat pyrittiin valitsemaan yrityksen valmistamien tuotteiden osista, koska sillä turvattiin varaosien saanti pitkälle tulevaisuuteen. Laitteen ohjelmisto laadittiin Python-ohjelmointikielellä.</p> <p>Laitteen prototyyppinä rakennettiin kolme kappaletta, joista viimeinen jäi lopulliseksi laitteeksi. Ensimmäisellä prototyypillä testattiin liikkeen muodostumista ja tarkkuutta, toisella prototyypillä testattiin laitteen käyttöä röntgenlaitteen yhteydessä. Viimeinen prototyyppi toteutettiin kahden ensimmäisen oppien perusteella ja se todettiin toimivaksi.</p> <p>Lopputestauksessa laitetta verrattiin vaatimusmäärittämiin. Testilaitte saavutti vaaditun 0,1 mm:n liiketarkkuuden vaaditulla terävän kerroksen alueella. Laitteen aloituspaikan sijainnin toistotarkkuus oli vaaditussa 0,1 mm:n rajoissa.</p> <p>Käyttöönotossa testilaitteella tehtiin terävän kerroksen määrittäminen panoraamaröntgenlaitteelle, josta laite suoriutui vaatimusten mukaisesti.</p>	
Avainsanat	Panoraamaröntgenlaite, testityökalu, Raspberry Pi, Python

Author Title Number of Pages Date	Petri Sinisalo Design and implementation of a test tool for determining the sharpness layer of a panoramic x-ray device 43 pages 20 March 2019
Degree	Bachelor of Engineering
Degree Programme	Automation Technology
Professional Major	Automation
Instructors	Verification Engineer Janne Hirsimäki Lecturer Kristian Junno
<p>The aim of this engineering thesis was to automate the movement of the phantom required to determine the sharp layer of the panoramic x-ray device. The designed and produced device replaced the previously used manual and inaccurate movement equipment.</p> <p>A test fixture which can be attached to panoramic x-ray device chin rest was designed in this work. Test phantom can be attached to the test fixture and can be moved accurately to the depth direction. Also the Raspberry Pi computer was studied and its suitability was tested for the automation.</p> <p>The operation of Raspberry Pi and taking its IO-port in use was studied in the beginning. Different stepper motor types and their operation were studied to choose the best suitable motor for the test device. The components used in the test device were chosen from products being manufactured by the company to secure spare parts availability in the future. Device software was created with Python programming language.</p> <p>Three prototypes were built: the first prototype was tested for motion and accuracy, the second was used to test the compatibility of the device with an X-ray device and the third prototype was implemented on the basis of the first two and it became the final version.</p> <p>In the final test, the device was compared to the requirement. The accuracy of the device was measured. The test device reached the required 0,1 mm motion accuracy in the sharp-layer area. The accuracy to repeat the location of the home position was within the required 0.1 mm range.</p> <p>During the commissioning of the test device, a sharp layer was determined for the panoramic X-ray device and the device performed its task according to requirements.</p>	
Keywords	Panoramic x-ray unit, test tool, Raspberry Pi, Python

# Sisällys

## Lyhenteet

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Panoraamaröntgenlaite</b>	<b>1</b>
2.1	Yleistä	1
2.2	Geometriakalibrointi	4
2.3	Terävä kerros	5
2.4	Viuhkafantomi	6
<b>3</b>	<b>Työn määrittely</b>	<b>7</b>
<b>4</b>	<b>Suunnittelu</b>	<b>8</b>
4.1	Raspberry Pi tietokone	8
4.2	Moottorin ohjauskortti	13
4.3	Python-ohjelmointi	15
4.4	Askelmoottorit	18
<b>5</b>	<b>Toteutus</b>	<b>25</b>
5.1	Prototyypin rakentaminen	25
5.2	Leukatukeen asennettavan prototyypin rakentaminen	29
5.3	Testilaitteen rakentaminen	33
<b>6</b>	<b>Testaus</b>	<b>34</b>
<b>7</b>	<b>Yhteenveto</b>	<b>38</b>
	<b>Lähteet</b>	<b>40</b>

## Lyhenteet

ACK	<i>Acknowledgement</i> . Kuittausbitti, jolla ilmoitetaan virheetön paketti saapuneeksi.
DC	<i>Direct Current</i> . Tasavirta.
ELEC	<i>Open Embedded Linux Entertainment Center</i> . Kotiteatterilaitteille suunniteltu Linux pohjainen käyttöjärjestelmä, joka toimii myös Raspberry Pi tietokoneissa.
FAT	<i>File allocation table</i> . Microsoftin kehittämä tiedostojärjestelmä, jota käytetään edelleen muistikorttien tiedostojärjestelmänä.
FET	<i>Field Effect Transistor</i> . Kanavatransistori, jossa varauksenkuljettajien liikettä ohjataan puolijohdekanavassa vaikuttavan sähkökentän avulla.
GPIO	<i>General Purpose I/O</i> . Yleiskäyttöinen pinni mikrokontrollereissa ja mikroprosessoreissa, joka voidaan ohjelmoida joko signaalin vastaanottajaksi tai lähettäjäksi.
I2C	<i>Inter-Integrated Circuit</i> . Yksinkertainen kaksisuuntainen ohjaus- ja tiedonsiirtoväylä.
I/O	<i>Input / output</i> . Tiedon signalointi tietokonelaitteiston komponenttien välillä.
IoT	<i>Internet of Things</i> . Esineiden ja asioiden internet.
NOOBS	<i>New Out Of Box Software</i> . Raspberry tietokoneen käyttöjärjestelmän asennusohjelma.
NXP	<i>Next eXPerience</i> . NXP Semiconductors on alankomaalainen puolijohdetta valmistava suuryritys.
OPTG	<i>OrthoPanTomoGrafia</i> . Kasvojen alaosan kuvantamiseen käytetty kerroskuvaustekniikka.
OSMC	<i>Open source media center</i> . Kodin viihdekeskuksen käyttöjärjestelmä.
PWM	<i>Pulse Width Modulation</i> . Modulointitapa, jossa jännitteen pulssisuhdetta muutetaan niin, että lähtösignaalin keskiarvo yhden värähtelyjakson ajalta laskettuna on sama kuin modulointisignaalin arvo.

QC	<i>Quality Control.</i> Laadun valvonta.
RISC OS	<i>Reduced Instruction Set Computer Operating System.</i> Acorn Computers Ltd:n kehittämä käyttöjärjestelmä ARM-pohjaisille kehitysalustoille.
SCL	<i>Serial clock Line.</i> I2C-väylän kello linja.
SD	<i>Secure Digital.</i> Digikameroissa, älypuhelimissa ja kämmentietokoneissa yleisesti käytetty muistikorttityyppi.
SDA	<i>Serial data Line.</i> I2C-väylän data linja.
SPI	<i>Serial Peripheral Interface.</i> Synkroninen sarjaliikenneväylä, jota käytetään sulautetuissa järjestelmissä lyhyen matkan viestintään.
SSID	<i>Service Set Identifier.</i> Langattoman lähiverkon verkkotunnus.
TMJ	<i>TemporoMandibular Joint.</i> Leukanivel.
V&V	<i>Verification and Validation.</i> Toisistaan riippumattomia menettelyjä, joita käytetään yhdessä tarkastettaessa, että tuote, palvelu tai järjestelmä täyttää vaatimukset ja käyttötarkoituksen.

## 1 Johdanto

Tämän insinööriyön tavoitteena on suunnitella ja toteuttaa KaVo Kerr Group Finlandin V&V-tiimille testityökalu panoraamaröntgenlaitteen terävyysalueen määrittämiseen. Työn tarkoituksena on kehittää panoraamaröntgenlaitteen leukatukeen kiinnitettävä laite, joka liikuttaa viuhkafantomia tarkasti syvyysuunnassa. Eri syvyyksiltä otetuista röntgenkuvista pystytään määrittämään panoraamaröntgenlaitteen terävyysalueen paikka ja sen laajuus. Laitteen ohjaamiseen halutaan käyttää Raspberry Pi-tietokonetta. Raspberry Pi-tietokoneen käyttöön mittausten automatisoinnissa halutaan tutustua, koska laite on edullinen, siihen on helppo liittää ohjattavia laitteita ja sen avulla voidaan helposti automatisoida testausta.

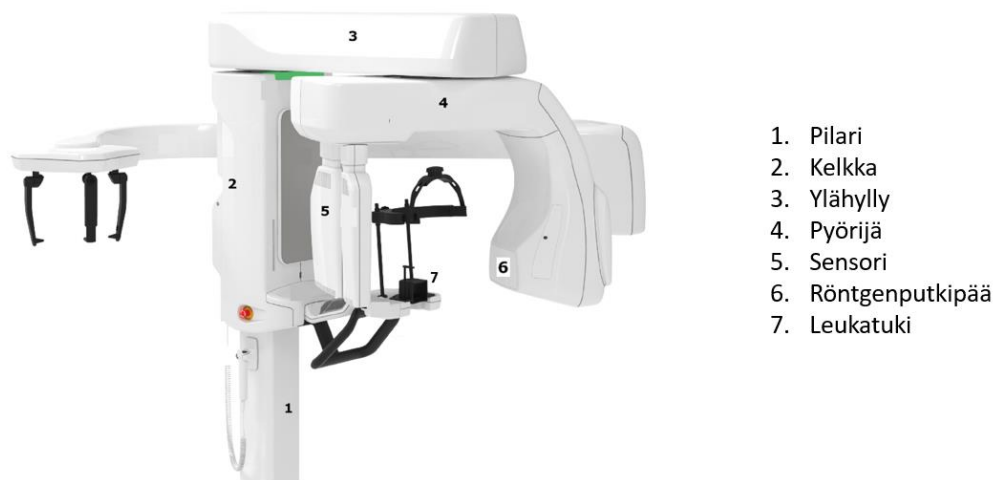
KaVoKerr Group Finland (jatkossa KavoKerr) on osa yhdysvaltalaisista tiede- ja teknologiayhtiö Danaheria. KaVoKerr tunnetaan myös nimellä Palodex Group ja se perustettiin vuonna 1964 tuotteistamaan suomalaisen hammaslääkärin, professori Yrjö Paateron keksintöä hampaiden panoraamakuvantamisesta. KaVoKerr on yksi Suomen suurimmista terveysteknologian vientiyrityksistä. Yrityksen sijaintipaikka on Tuusula, jossa sijaitsee kuvantamisen tuotekehitys, tuotanto sekä tekninen tuki. KaVoKerr Group Finland on yksi Lean benchmark-yrityksistä Suomessa ja sen toimintaa ohjaa Lean filosofian pyrkimys jatkuvaan parantamiseen. [1.]

## 2 Panoraamaröntgenlaite

### 2.1 Yleistä

Nykyaikaisella panoraamaröntgenlaitteella voidaan tuottaa laadukkaita digitaalisia kuvia hampaista, leukanivelistä sekä pään ja kaulan alueesta. Kuvien ottamiseen tarvitaan työasema, joka on liitetty röntgenlaitteeseen sekä hampaiden kuvantamisohjelmisto kuvien käsittelyyn ja hallintaan. Panoraamaröntgenlaitteen kuvausohjelmat ovat perus- ja lasten panoraama, osapanoraamat edellisiin, bitewing sekä lateraalisuunnan ja posterior-anterior-suunnan TMJ (leukanivel). [2, s. 7.]

Panoraamaröntgenlaitteen pääosat ovat pilari, kelkka, ylähylly, pyörijäosa, sensori ja röntgenputkipää. Pilari on laitteen perusrunko, joka pultataan pohjalevystä lattiaan ja takaosastaan seinään. Pilari tulee asentaa tarkasti pystysuoraan. Kelkka liikkuu pilarissa ylä- ja alasuunnassa, ja sen avulla röntgenlaite säädetään oikealle korkeudelle potilaan pituuden mukaisesti. Ylähylly on nivelöity kelkan yläosaan, ja se liikkuu kuvantamisen aikana. Pyörijäosa on kiinnitetty ylähyllyyn. Pyörijäosa liikkuu kuvauksen aikana syvyysuunnassa ja pyörii potilaan pään ympäri. Pyörijän pääosat ovat röntgenputkipää ja digitaalinen sensori röntgensäteen ilmaisuun. Panoraamaröntgenlaitteen pääosat ovat esitetty kuvassa 1. [2, s.13.]

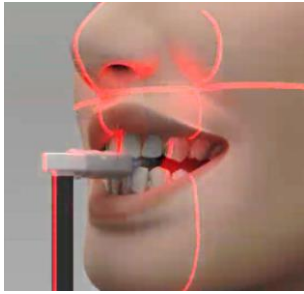


1. Pilari
2. Kelkka
3. Ylähylly
4. Pyörijä
5. Sensori
6. Röntgenputkipää
7. Leukatuki

Kuva 1. Panoraamaröntgenlaitteen pääosat [2, s. 13].

Panoraamaröntgenlaitteen panoraamaohjelmaa käytettäessä potilas asetetaan laitteeseen niin, että potilaan leuka asettuu laitteen leukatukea vasten ja potilaan otsa laitteen otsatukea vasten. Potilas puree purutikkua siten, että ylä- ja alahampaat asettuvat purutikun uraan. Asettelyn apuna käytetty lateraalilaservalo keskitetään yläkulmahampaan keskelle. Tarkoituksena on saada hampaiden etualue ns. tarkkaan kerrokseen, joka on kuvausgeometrian terävän kerroksen keskialueella ja ohuimmillaan hammaskaaren etuhampaiden kohdalla. [3.] Kuvassa 2 on esitetty potilaan asettelussa käytetty purutikku.





Kuva 2. Potilaan asettelussa käytetty purutikku [4].

Kuvantamisparametrit valitaan ja kuvaus käynnistetään. Röntgenkuvauksen aikana pyörjäosan putkipää generoi röntgensäteen, joka suuntautuu ulos röntgenputkipään ikkunasta. Säde kavennetaan kollimaattorilla kapeaksi keilaksi. Kapea röntgensädekeila suuntautuu potilaan leuan alueelle. Osa säteilystä heijastuu, osa absorboituu ja osa läpäisee kuvattavan alueen. Potilaan läpäisseet säteet tallentuvat kallon vastakkaisella puolella sijaitsevaan digitaaliseen sensoriin. Panoraamakuvauksessa pyörjäosa pyörähtää kuvauksen aikana reilun puoliympyrän mittaisen alueen jatkuvasti säteilyttäen. Näin saadaan koko hampaiston alueen kattava röntgenkuva potilaasta yhdellä kuvauksella. [3.] Kuvassa 3 on esitetty tyypillinen hammasalueen peruspanoraamaröntgenkuva.



Kuva 3. Hammasalueen peruspanoraamaröntgenkuva [2, s. 22].

## 2.2 Geometriakalibrointi

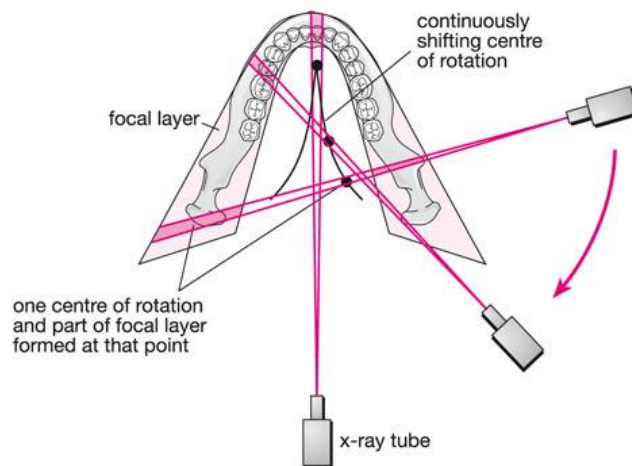
Panoraamaröntgenlaitteen asennuksen jälkeen laitteelle tehdään tarvittavat kalibroinnit, jotta voidaan varmistaa laitteen toimintakunto. Yksi tarvittavista kalibroinneista on geometriakalibrointi. Geometriakalibroinnissa laitteen leukatukeen potilaan tilalle asennetaan ympyrälieriön muotoinen kuulafantomi, johon on upotettu metallikuulia eri tasoihin. Kuulafantomi kuvataan, laitteen ohjelma mittaa röntgenkuvista kuulien etäisyydet ja laskee laitteen pyörjäosan pyörähdyskeskipisteen tarkan paikan, joka tallennetaan laitteen muistiin. Pyörjä ajetaan lähtöpisteeseensä ennen valotusta. Tämä toimenpide toistetaan tarvittaessa niin monta kertaa, että haluttu pyörimiskeskipisteen paikka saavutetaan. Kuvassa 4. on esitetty kuulafantomi asennettuna röntgenlaitteen leukatukeen. [3.]



Kuva 4. Geometriakalibroinnissa käytettävä kuulafantomi asennettuna röntgenlaitteen leukatukeen [3, s. 77].

### 2.3 Terävä kerros

Panoraamakuvauksessa pyörijäosan pyöriessä kapea röntgensäde läpäisee potilaan kallon. Säde kulkee pyörimiskeskustien läpi ja aktivoi sensorin pikseleitä pyörimisliikkeen suhteessa. Sensorin ilmaisema säteily tallennetaan säteen aktivoimalta alueelta. Näin saadaan koottua laaja-alainen kuva koko hammaskaaresta. Tätä kuvantamistapaa kutsutaan tomografiakerroskuvaukseksi eli orthopantomografiaksi (OPTG). Pyörimiskeskustien ja sensorin välistä tiettyä geometristä aluetta kutsutaan teräväksi alueeksi. Panoraamaröntgenkuvauksessa kuvauskehikon pyörimisen lisäksi kuvauskehikon pyörimiskeskustettä muutetaan kuvauksen aikana. Tämä johtuu siitä, että ihmisen hammaskaari ei ole ympyrän kaaren muotoinen, vaan lähinnä puolikkaan ellipsin muotoinen. Tomografiakerroskuvauksen ja pyörimisliikkeen vaikutuksesta vain pieni alue edellä mainitusta terävästä alueesta näkyy röntgenkuvassa terävän. Kuvauksen aikana kuvattavaa aluetta pyritään pitämään koko ajan samalla etäisyydellä pyörimiskeskustien ja sensorin välillä. Kun kuvattu kohde sijaitsee terävän alueen keskivaiheilla, sensoria pyyhkivä säde kulkee samalla nopeudella, kun kuvattavaa kerrosta pyyhkivä säde. Tällöin kohde piirtyy tarkkana sensoriin. [5, s. 21, 23–24, 29.] Kuvassa 5 on esitetty pyörimiskeskustien liike panoraamakuvauksen aikana. Kuvausgeometrian takia terävä alue on ohuimmillaan hammaskaaren etualueella.

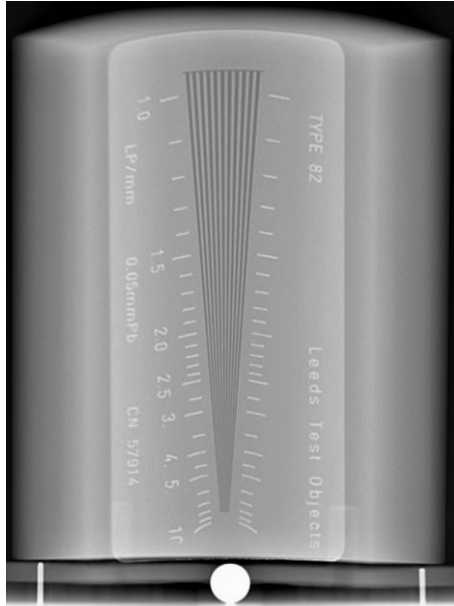


Kuva 5. Pyörimiskeskustien liike panoraamakuvauksen aikana ja terävyysalueen muodostuminen [6].

Terävän kerroksen sijoittuminen oikeaan kohtaan pyörimiskeskusteen etualueella tarkastetaan siten, että röntgenlaitteen leukatukeen asetetaan viuhkafantomi ja otetaan panoraamaröntgenkuva. Saadusta kuvasta määritetään silmämääräisesti viivaparierotuskyky ja saatua tulosta verrataan vaatimusmäärittäisiin. Jos vaadittu viivaparierotuskyky saavutetaan, on laitteen kalibrointi oikein. Jos vaadittua viivapiirtotarkkuutta ei saavuteta, liikutetaan viuhkafantomia askeleittain eteen- tai taaksepäin ja otetaan siirtojen jälkeen röntgenkuva. Röntgenkuvista nähdään, paraneeko vai huononeeko piirtotarkkuus ja näin terävyysalueen paikka saadaan määritettyä. [3.]

## 2.4 Viuhkafantomi

Viuhkafantomi on metallinen levy, johon on syövytetty eri levyisiä uria eri etäisyyksille toisistaan. Viuhkafantomissa urat ovat yläosassa leveämpiä ja etäämmällä toisistaan. Alempana urat ovat kapeampia ja lähempänä toisiaan. Niiden avulla otetuista röntgenkuvista voidaan tutkia röntgenlaitteen paikkaerotuskykyä. Mitä ohuemmat viivat erottuvat röntgenkuvasta sitä tarkempi on röntgenlaitteen kuva. Kuvassa 6. on esitetty viuhkafantomilla otettu röntgenkuva.



Kuva 6. Viuhkafantomilla otettu röntgenkuva

### 3 Työn määrittely

Tässä insinööriyössä automatisoidaan viuhkafantomien siirto syvyysuunnassa. Työssä kehitetään laite, joka liikuttaa tietokoneen avulla viuhkafantomia tarkasti eteen tai taaksepäin. Laitetta ohjataan manuaalisesti painonapeilla ja laite voidaan myös kytkeä osaksi suurempaa automaattista testausjärjestelmää. Työssä laaditaan tarvittavat ohjelmakoodit.

Seuraavassa ovat vaatimukset kehitettävälle laitteelle:

- Laitteen tulee olla röntgenlaitteen leukatukeen kiinnitettävissä, ja laite tulee voida poistaa röntgenlaitteesta.
- Laitteeseen tulee voida asentaa viivapari- sekä viuhkafantomi, ja näitä tulee voida liikuttaa eteenpäin ja taaksepäin röntgenlaitteen leukatuesta 0,5 mm:n askelin.
- Laitteen liikkeen tarkkuus terävän kerroksen alueella tulee olla parempi kuin 0,1 mm.
- Laite ei saa aiheuttaa röntgenkuvaan heijastuksia tai häiriöitä.
- Viuhkafantomien asennuskorkeus tulee olla lähellä QC-kuvauksessa käytettävän viivaparifantomien korkeutta. Sallittu poikkeama ei saa olla suurempi kuin 5 mm.
- Laitteen etulevyyn tulee asettaa 3 painonappia, joista laitteen kelkkaa voidaan siirtää aloituspaikkaan, 0,5 mm eteen- ja 0,5 mm taaksepäin.
- Laitteen tulee voida palauttaa aina aloituspaikkaan riippumatta siitä missä kohtaa kelkka kullakin hetkellä sijaitsee.
- Laitteessa tulee olla ethernet-yhteys laitteen liittämiseksi osaksi suurempaa automaattista testausjärjestelmää.
- Käynnistettäessä liikutusohjelma, laitteen tulee käynnistyä automaattisesti suoraan ilman näyttöä, näppäimistöä ja hiirtä.
- Laitteen ohjelmiston tulee käynnistyttyään kalibroida kelkka ja palauttaa kelkka aloituspaikalleen. Painettaessa plus-painokytöntä laitteen kelkan tulee liikkua 0,5 mm eteenpäin ja painettaessa miinus-painokytöntä laitteen kelkan

tulee liikkua 0,5 mm taaksepäin. Painettaessa home-kytkintä laitteen kelkan tulee kalibroida kelkka ja palauttaa kelkka aloituspisteeseen.

- Laitteen ohjelmiston tulee sisältää kutsuttavissa olevia Python-ohjelmia, joiden avulla laitteen kelkka voidaan siirtää aloituspisteeseen, 0,5 mm eteen- ja 0,5 mm taaksepäin.
- Laitteen käynnistys ja sammutus tulee voida hoitaa hallitusti.

## 4 Suunnittelu

Työtä määritettäessä päätettiin, että kehitettävä laite asennetaan röntgenlaitteen leukatukeen ja aiemmin käytössä olleesta kolmijalasta luovutaan. Näin saadaan tarkempi asemointi kehitettävälle työkalulle. Päätettiin käyttää laitteen ohjaamiseen Raspberry Pi (jatkossa Raspberry)-tietokonetta, koska laitteen ominaisuudet haluttiin selvittää ja tutkia samalla laitteen mahdollista käyttöä muissa automatisointikohteissa. Raspberryn ohjelmointiin käytettiin Python-ohjelmointikieltä ja liike toteutettiin yrityksen tuotteissa käytettyjen askelmoottorin ja lineaarijohteiden avulla.

### 4.1 Raspberry Pi tietokone

Raspberry on brittiläisen tutkijan Eben Uptonin kehittämä yksinkertainen, pienikokoinen ja edullinen luottokortin kokoinen tietokone. Kehitystyön tavoitteena oli luoda tietokone, jonka jokainen voisi hankkia ja jonka avulla jokainen voisi helposti kehittää ohjelmointitaitoja. Hyväntekeväisyysjärjestö Raspberry Pi Foundation perustettiin jakamaan ohjelmointitaitoja ja -tietoutta. Raspberry-tietokone myydään ilman näppäimistöä, hiirtä ja näyttöä. Kuvassa 7. on esitetty Raspberry-tietokone myyntipakkauksessaan. [8.]



Kuva 7. Raspberry tietokone myyntipakkauksineen [7].

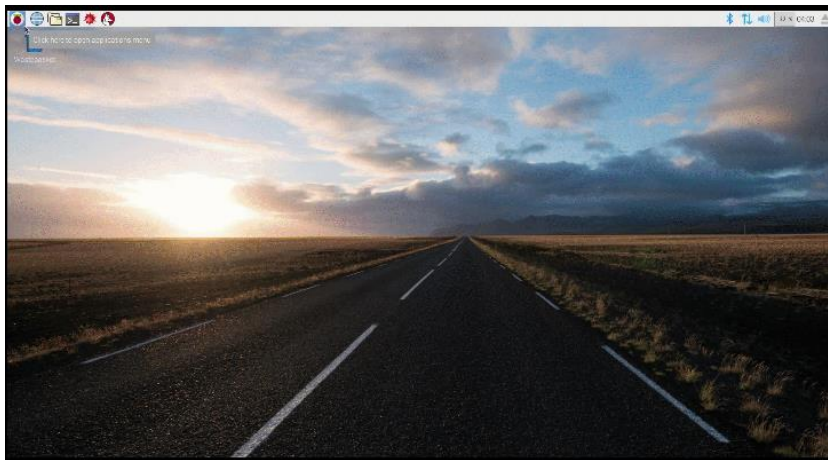
Raspberrysssä ei ole kovalevyä eikä valmiiksi asennettua käyttöjärjestelmää. Soveltuvia käyttöjärjestelmiä on useita ja käyttöjärjestelmä ladataan verkosta SD-muistikortille. Ilmainen Linux-pohjainen Rasbian-käyttöjärjestelmä on valmistajan ilmoittama virallinen käyttöjärjestelmä, joka toimii kaikilla eri malliversioilla. Muita suosittuja käyttöjärjestelmiä Raspberrylle ovat OSMC, OpenELEC, Windows IoT Core, and RISC OS. Rasbian-tietokoneita on tällä hetkellä viittä eri mallia, joista vanhin, Pi A+ julkaistiin 2012. Pi B+ julkaistiin 2014, Pi2 B julkaistiin 2015 ja uusin malli, Pi3 B julkaistiin 2016. Taulukossa 1 on esitetty eri Rasbian mallien tekniset tiedot. [7, 8, 9.]

Taulukko 1. Rasbian Pi-tietokoneiden mallien tekniset tiedot [8].

	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Raspberry Pi Model B+	Raspberry Pi Model A+	Raspberry Pi Zero & Zero W
Processor	Broadcom	Broadcom	Broadcom	Broadcom	Broadcom
Chipset	BCM2837 64-bit quad-core processor	BCM2837 64-bit quad core processor	BCM2835 32-bit single-core processor	BCM2835 32-bit single-core processor	BCM2835 32-bit single-core processor
Processor Speed	1.2 GHz	900 MHz	700 Mhz	700 Mhz	1 GHz
RAM	1 GB	1 GB	512 MB	256 MB	512 MB
Storage	MicroSD	MicroSD	MicroSD	MicroSD	MicroSD
Output	4 USB ports	4 USB ports	4 USB ports	1 USB port	1 micro-USB port
GPIO	40-pin	40-pin	40-pin	40-pin	40-pin
Ethernet Port	Yes	Yes	Yes	No	No
Wi-Fi	Yes	No	No	No	No (Zero)/Yes (Zero W)
Bluetooth	Yes	No	No	No	No (Zero)/Yes (Zero W)

## Käyttöönotto

Raspberryn käyttöönotto on yritetty tehdä mahdollisimman helpoksi. Järjestön virallisilla verkkosivuilla on kuvalliset ohjeet, kuinka asennus tapahtuu. Ensin tietokoneeseen kytketään monitori, näppäimistö ja hiiri. Valmistajan suosittama Linux-pohjainen Rasbian-käyttöjärjestelmä ladataan joko valmistajan verkkosivuilta tai tilataan esiasennettu SD-kortti. Valmistaja on kehittänyt käyttöjärjestelmän verkkolatausta varten NOOBS latausjärjestelmän. NOOBS tulee sanoista new out of box software. Sen avulla käyttöjärjestelmä voidaan ladata ilman Linux-osaamista. Käyttöjärjestelmän lataukseen tarvitaan vähintään 8 GB:n SD-muistikortti, joka tulee alustaa. Alustusta varten ladataan SD-formatter ohjelma, jonka avulla kortti alustetaan FAT32-formaattiin. Raspberryn sivuilta ladataan ensin NOOBS-käyttöjärjestelmän sisältämä zip-tiedosto. Pakattu tiedosto puretaan ja tiedostot tallennetaan muistikortille, joka asennetaan Raspberryn. Kun Raspberryn kytketään virta, avautuu valikkoikkuna, josta valitaan Rasbian-käyttöjärjestelmä ja asennus alkaa. Käyttöjärjestelmän asennuksen valmistuttua langaton verkkoyhteys muodostetaan syöttämällä verkon SSID-tunnus ja salasana. Raspberrissä on oma selainohjelma, jolla pääsee verkkoon. [10, 11.] Kuvassa 8. on esitetty Rasbian käyttöjärjestelmän työpöytä-ikkuna.



Kuva 8: Rasbian käyttöjärjestelmän työpöytäikkuna.

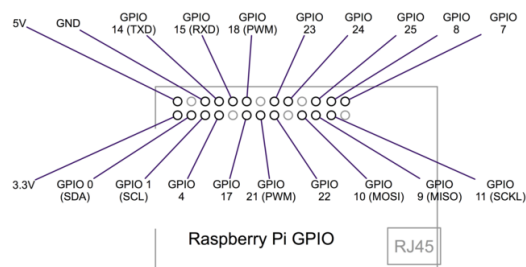


## IO-portit

Jotta Raspberry-tietokoneeseen saadaan kytkettyä ulkopuolisia laitteita, laitteessa on tarjolla USB-liitinten lisäksi 8 GPIO-linjaa, sarjaportti, I2C- ja SPI-väylät sekä IS2-lähtö [12]. Tässä työssä käytetään ledien ohjaamiseen ja kytkintietojen lukemiseen GPIO-linjoja sekä moottorinohjauskortin ohjaamiseen I2C-väylää.

## GPIO

GPIO on lyhenne sanoista *general purpose I/O*, jolla tarkoitetaan mikrokontrollerin porttia, joka voidaan ohjelmoida joko lähettämään tai vastaanottamaan dataa. Raspberry:ssä on 40-jalkainen riviliitin, jossa on 17 GPIO-porttia käytettävissä. Kahdesta portista saadaan 5 V:n - ja lopuista 3,3 V:n jännitetasoa. Lukumoodissa portit voidaan asettaa ohjelmallisesti alkutilaan ”1” tai ”0” sisäänrakennettujen ylös- tai alasvetovastuksien avulla. [13.] Kuvassa 9 on esitetty Raspberryn GPIO porttien pinnijärjestys.



Kuva 9. Raspberryn GPIO-porttien pinnijärjestys [14].

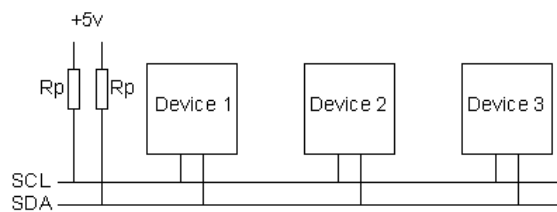
GPIO-porttien luku- ja kirjoitustoimintoa varten Raspberry:ssä otetaan käyttöön sisäänrakennettu Python-kirjasto ”Rpi.GPIO”. Tämä tapahtuu kirjoittamalla komentoikkunaan esimerkkikoodissa 1 esitetyt komennot. [15.]

```
>>> sudo apt-get install python-dev
>>> sudo apt-get install python-rpi.gpio
```

Esimerkkikoodi 1. Komennot GPIO-kirjaston käyttöönottoon [15].

## I2C-sarjaliikenneväylä

Philipsin kehittämä I2C-sarjaliikenneväylä perustuu kahteen signaaliin, kello- ja datasiignaliin. Kellosignaali merkitään SCL ja datasiignali SDA. Perustilassa väyliä pidetään ylös vetovastuksien avulla loogisessa ”1”-tilassa. Raspberryn GPIO-littimessä pinni 0 on varattu I2C-väylän datasiignaalille ja pinni 1 on varattu I2C-väylän kellosignaaliin. Kuvassa 10 on esitetty I2C-sarjaliikenneväylän rakenne. [16.]



Kuva 10. I2C sarjaliikenneväylän rakenne [16].

Väylän laitteet erotetaan toisistaan osoitteiden avulla. Isäntälaitte ohjaa väylän toimintaa, ja muut ovat orjalaitteita. Isäntälaitte kellottaa väylää, ja jokaisella orjalaitteella on oma osoitteensa, jonka avulla isäntälaitte erottaa laitteet toisistaan. Isäntälaitte lähettää väylään orjalaitteen osoitteen, ja orjalaitte vastaa ACK-bitillä tunnistessaan kutsun. Dataa lähetetään 8-bitin ryhmissä. Kun kellosignaali laskee alas, lähetys alkaa. Ensin lähetetään 7-bittinen orjalaitteen osoite, jonka perään lisätään yksi bitti, joka kertoo, luetaanko vai vastaanotetaanko. Tämän jälkeen datasiirto alkaa. Datasiirto loppuu kellosignaalin ylös nostoon. [16.] I2C-väylän käyttöönotto Raspberrytietokoneessa toteutetaan kirjoittamalla komentoikkunaan esimerkkikoodissa 2 esitetyt komennot [17].

```
>>> sudo apt-get install -y python-smbus
>>> sudo apt-get install -y i2c-tools
>>> sudo raspi-config
```

Esimerkkikoodi 2. Komennot I2C väylän käyttöönottoon [17].

Koodien syötön jälkeen Raspberryn konfigurointi-ikkuna avautuu. Ikkunasta avataan ”Interfacing options” ja valitaan ”I2C”, joka hyväksytään. Tämän jälkeen ohjelma kysyy, halutaanko I2C-väylä ottaa käyttöön aina käynnistyksen yhteydessä, joka hyväksytään. Tämän jälkeen Raspberry tulee käynnistää uudelleen. I2C-väylän toiminnan voi tarkastaa esimerkkikoodissa 3 esitetyllä komennolla, jolloin Raspberry skannaa liitetyt kortit ja palauttaa löydettyjen korttien osoitteet. [17.]

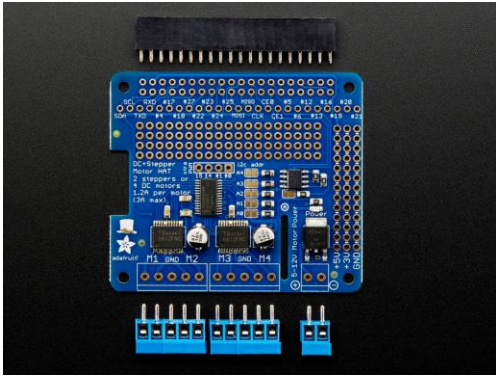
```
>>> sudo i2cdetect -y 1
```

Esimerkkikoodi 3. Komento I2C-väylän toimivuuden tarkastamiseen [17].

#### 4.2 Moottorin ohjauskortti

Moottorin ohjauskortiksi valittiin Adafruitin valmistama DC and Stepper Motor HAT -kortti. Kortti on halpa, yksinkertainen ja sopii jokaiseen Raspberry-tietokoneeseen, jossa on 40-jalkainen GPIO-liitin. Korttia varten on laadittu valmiiksi Python-kielinen ohjelmakirjasto moottorin ohjaukseen. Kortti toimitetaan pintaliitoskomponentit ladottuna, mutta liittimet irrallaan, joten ne tulee juottaa itse. Moottorin ohjauskortti kytkeytyy I2C-väylän kautta Raspberry-tietokoneeseen ja väylärakenne sallii maksimissaan 32 orjakortin asentamisen samaan väylään. Yksi moottorin ohjauskortti kykenee ohjaamaan kahta askelmoottoria, tai neljää DC-moottoria. Tällöin voidaan ohjata maksimissaan 64 askelmoottoria tai 128 DC-moottoria. Tässä työssä ohjataan vain yhtä askelmoottoria, joten yksi kortti riittää. [18.]

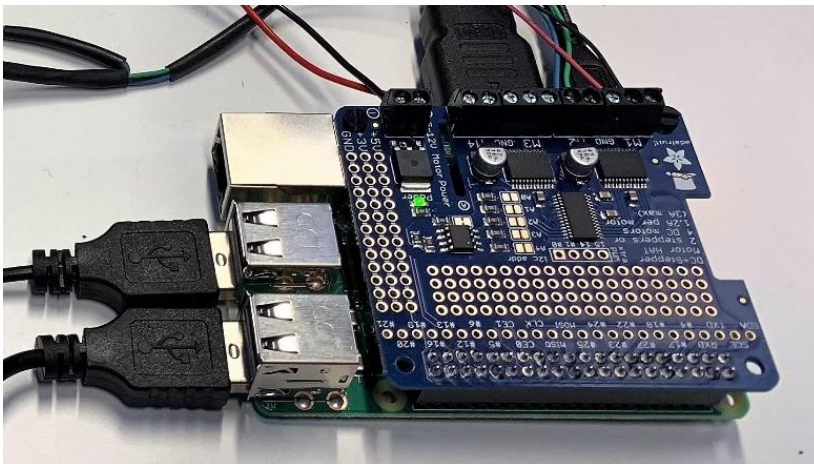
Moottorin ohjauskortin väyläohjaimena on NXP:n 16-kanavainen PCA9685-ohjauspiiri. Jokaiselle kanavalle on oma 12-bittinen PWM-ohjain, jonka taajuutta voidaan yksilöllisesti säätää välillä 24–1525 Hz. Päälläolojakso (Duty cycle) on säädettävissä 0–100 %:n välillä. [19.] Moottorin ajamiseen käytetään Toshiba TB6612FNG-ohjainpiiriä, joka pystyy syöttämään virtaa 1,2 A moottoria kohti. Moottorin ohjauskortissa on sisäänrakennettu lämpötilasuojaus, käyttöjännitteen polariteettisuojaus sekä hieman tilaa omille kytkennöille. Moottorin ohjauskortti kykenee ohjaamaan kahta 5–15 V:n askelmoottoria, jotka voivat olla joko unipolaarisia tai bipolaarisia, joissa on yksi kela, kaksoiskäämi, lomitettu tai mikroaskel. [18, 20.] Kuvassa 11 on esitetty Adafruitin valmistama DC and Stepper Motor HAT kortti.



Kuvassa 11. DC and Stepper Motor HAT kortti [22].

### Asennus

Moottorin ohjauskortin asennus on tehty helpoksi, asennustyö alkaa liittimien juottamisella, jonka jälkeen kortti painetaan kiinni Raspberryn 40-jalkaiseen korttiliittimeen. Kulmiin ruuvataan pakkauksessa mukana tulleet muovitukiholkit ja käyttöjännitteet kytketään. Moottorinohjauskortille tulee olla oma virtalähde, sillä ohjattavat moottorit voivat aiheuttaa häiriötä verkkoon päin eikä Raspberry-tietokonetta siksi suositella kytkettäväksi yhteiseen virtalähteeseen moottorikortin kanssa. [20.] Kuvassa 12 moottorin ohjauskortti on asennettuna Raspberry-tietokoneeseen.



Kuva 12. Moottorin ohjauskortti asennettuna Raspberry-tietokoneeseen.

Moottorinohjauskortin Python-ohjelmakirjasto ladataan verkosta GitHub-palvelusta. GitHub on verkkosivusto, joka tarjoaa paikan Git-versionhallintaa käyttäville ohjelmakehitysprojekteille. Git on versionhallintaohjelmisto ja koostuu suuresta joukosta pienempiä sovelluksia, joista kukin toteuttaa yksittäisiä toimintoja. GitHub tarjoaa graafisen käyttöliittymän, ohjelmistovirheiden seurannan ja tehtävien hallintatyökalut ilmaiseksi ohjelmavarastoille. [21.]

Raspberry-tietokoneen ympärille on muodostunut kaupallisia yrityksiä, jotka valmistavat ja myyvät lisälaitekortteja asennettavaksi Raspberry-tietokoneeseen ja näiden korttien ohjelmakirjastot ovat GitHub-tyyppisillä verkkosivustoilla kaikkien vapaasti ladattavina ja muokattavina. Tässä työssä käytetyn moottorinohjauskortin ohjelmakirjastot löytyvät GitHubista nimellä ”Adafruit CircuitPython MotorKit”. [22.]

Moottorinohjauskortin ohjelmakirjasto asennetaan verkosta, joten verkkoyhteys tulee olla muodostettuna. Asennus tapahtuu syöttämällä Raspberryn komentoikkunaan esimerkikoodin 4 mukainen komento. Lataus etenee suhteellisen hitaasti tehokkaiisiin tietokoneisiin verrattuna. Latauksen valmistuttua kaikki vaadittavat asennustoimenpiteet on suoritettu ja moottorin ohjauskortti on valmis käyttöön. [23.]

```
>>> sudo pip3 install adafruit -circuitpython-motorkit
```

Esimerkkikoodi 4. Komento, jolla moottorin ohjauskortin kirjasto asennetaan [23].

### 4.3 Python-ohjelmointi

Python-ohjelmointikieli julkaistiin vuonna 1990, ja sen nimi perustuu 70-luvun suosittuun tv-sarjaan Monty Pythonin lentävä sirkus. Sen kehitti Alankomaista kotoisin oleva ohjelmistosuunnittelija Guido van Rossum. Python on tulkattava ohjelmointikieli, jota pidetään helppona oppia, koodi on selkeää ja helposti luettavaa. Pythonista on useita eri versioita, ja työn kirjoitushetkellä uusien saatavilla oleva on joulukuussa 2016 julkaistu versio 3.6. [24.]

Python-koodauksen aloittaminen onnistuu vasta-alkajaltakin, sillä verkosta löytyy lukemattomasti esimerkikioodeja, joita opiskelemalla oman koodin kirjoittaminen onnistuu

helposti. Myös erilaisia verkkokursseja on runsaasti tarjolla. Tuskin koskaan suora toisen koodin kopioiminen toimii omassa laitteistossa, mutta niitä opiskelemalla ja muuntamalla omaan käyttöön sopivaksi saadaan toimivaa koodia aikaiseksi. Python-ohjelmaa laadittaessa ensin ladataan tarvittavat kirjastot. Python sisältää laajan standardikirjaston, joka muodostuu kokoelmasta moduuleita eri tarkoituksiin. [25.] Tässä luvussa esitetään ne Python-kielen toiminnot, joita tässä työssä on käytetty.

Python-ohjelmakoodia laadittaessa ensin ladataan tarvittavat ohjelmakirjastot. Tämä tapahtuu import-komennon avulla. Esimerkiksi matemaattisille funktioille on math-kirjasto. Esimerkkikoodissa 5 ladataan ensin math-kirjasto, ja ohjelma pyörittää for-silmukan avulla 10 kertaa antaen joka kierroksella muuttujalle "i" uuden yhtä suuremman arvon. Muuttujan neliöjuuri lasketaan ja joka kierroksen laskutoimituksen tulos näytetään. [26.]

```
import math
for i in range(0, 10):
    print("Luvun", i, "neliöjuuri on", math.sqrt(i))
```

Esimerkkikoodi 5. Malliesimerkki Python ohjelmakirjaston käytöstä [26].

Ehtorakenteet Pythonissa tehdään if/else-komennon avulla. Ehtorakenteen aloittaa if-sana, jonka jälkeen tulee ehto. Jos ehto on tosi, ohjelma suorittaa ehtorakenteen sisällä olevat rivit. Jos ehto on epätosi, ohjelma hyppää ehtorakenteen yli. Esimerkkikoodissa 6 on esitetty ohjelma, joka kysyy henkilön nimeä. Jos syötetty nimi toteuttaa ehdon, tulostetaan "Hellurei" -teksti. Jos taas nimi ei toteuta ehtoa, tulostetaan "Hoi, sankari" -teksti. [27.]

```
nimi = input("Nimi: ")
if nimi != "Henrikki":
    print("Hellurei!")
else:
    print("Hoi, sankari!")
```

Esimerkkikoodi 6. Malliesimerkki ehtorakenteen käytöstä [27].

Python-kielen toistorakenteet ovat while-silmukka, jossa ohjelma toistaa samaa koodia niin kauan, kuin jokin ehto on voimassa, sekä for-silmukka, jossa ohjelma käy läpi joukon tietoja ja suorittaa jokaisen kohdalla saman koodin. Esimerkkikoodissa 7 ohjelma kysyy käyttäjältä niin kauan tunnussanaa, kunnes haluttu tunnussana saadaan. Kun haluttu tunnussana saadaan, tulostetaan teksti ”Tervetuloa”. [28.]

```
sana = ""
while sana != "python":
    sana = input("Kirjoita tunnussana: ")
print("Tervetuloa!")
```

Esimerkkikoodi 7. Malliesimerkki toistorakenteen käytöstä [28].

Askelmoottorin ajoa varten ladataan käytettävän moottorinohjauskortin kirjasto käyttöön. Askelmoottoria ajetaan edellä mainitulla for-silmukalla. Esimerkkikoodissa 8 esitetystä ohjelmakoodista ajetaan 7500 askelta eteenpäin kaksivaiheohjauksella. For-silmukka pyörittää 7500 kertaa, ja jokaisella pyörähdyksellä askelmoottoria ajetaan yksi askel.

```
for i in range(7500):
    Kit.stepper1.onestep(direction=stepper.FORWARD, style=stepper.DOUBLE)
```

Esimerkkikoodi 8. Malliesimerkki askelmoottorin ohjauksesta for-silmukkaa käyttäen.

Ulkoisen kytkimen tietoa voidaan lukea add event detect-toiminnon avulla. Esimerkkikoodissa 9 esitetystä ohjelmakoodista ensin määritetään, että kytkin 1 on johdotettu IO-porttiin numero neljä. Add event detect-toiminto odottaa portista neljä jännitteen nousevaa reunaa ja tämän toteutuessa suorittaa def-funktion mukaisen toiminnan. Jotta kytkimen luku on jatkuvaa, käytetään while-toistorakennetta, joka jatkuvasti 0,3 sekunnin välein pyörittää ohjelman niin kauan kun ohjelma sammutetaan.

```

kytkinl=4
def Kytkinl(channel):
    print("kytkinl painettu"):
GPIO.add_event_detect(4,GPIO.RISING,callback=Kytkinl,bouncetime=100)
while True:
    sleep(0.3)

```

Esimerkkikoodi 9. Malliesimerkki kytkintiedon lukemisesta add event detect toimintoa käyttäen.

Rajakytkimen lukuun voidaan käyttää esimerkiksi if-komentoa. Tällöin ensin määritetään kytkimen IO-portti ja lisätään rajakytkimen luku askelmoottorin ajo-ohjelmakoodiin esimerkkikoodin 10 mukaisesti. Break-komento pysäyttää ajon.

```

Rajakytkinl=5
for i in range(7500):
    Kit.stepper1.onestep(direction=stepper.FORWARD,style=stepper.DOUBLE)
    if GPIO.input(Rajakytkinl)==1:
        break

```

Esimerkkikoodi 10. Malliesimerkki rajakytkimen käytöstä osana askelmoottorin ajo-ohjelmakoodia.

Ajastimen rakennus toteutetaan lataamalla Timer-kirjasto. Esimerkkikoodissa 11 on esitetty yksinkertainen ajastin. Ajastimen ajaksi asetetaan 10 sekuntia. Ajastimen käynnistää t.start()-komento. Kun ajastimen asetusaika on kulunut, suoritetaan timeout-funktion toiminta.

```

from threading import Timer

def timeout():
    print("Aika täyttyi")

t = Timer(10, timeout)
t.start()

```

Esimerkkikoodi 11. Malliesimerkki ajastimen toteutuksesta.

#### 4.4 Askelmoottorit

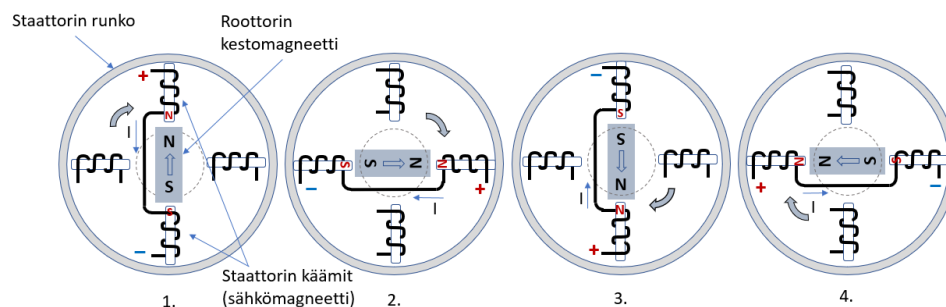
Askelmoottori muuttaa sähköenergian pyöriväksi mekaaniseksi liikkeeksi. Askelmoottori on harjaton synkroninen moottori ja se jakaa pyörivän liikkeen useisiin vaiheisiin. Askelmoottori



pyörii askelkulmissa, mikä tarkoittaa, että yhdellä askeleella moottori pyörähtää tietyn verran. Yleisimmin käytetyissä askelmoottoreissa on yksi pyörähdys jaettu 200 askeleeseen. Tästä saadaan yhden askeleen askelkulmaksi 1,8 astetta. Pyörivää osaa kutsutaan roottoriksi ja kiinteää runko-osaa staattoriksi. Askelmoottoreita on kolme päätyyppiä, kestopagneetti- (PM), reluktanssi- (VR) ja hybridiaskelmoottori (HB). [29.]

### Kestomagneettiaskelmoottori (PM)

Kestomagneettiaskelmoottori koostuu kestopagneettisesta pyörijäosasta sekä sähkömagneeteista, jotka on käämitty runko-osaan roottorin ympärille. Kuvassa 13 on havainnollistettu kaksivaiheisen kestopagneettiaskelmoottorin toimintaa. Kun moottorin pystykäämeihin tuodaan tasavirtaa, muodostuu kestopagneetteihin magneettiset navat N ja S. Sähkömagneettisen voiman vaikutuksesta roottorin kestopagneetti pyörähtää siten, että samanmerkkiset navat ovat vastakkain. Kun virta johdetaan pystykäämiin sijaan vaakakäämiin, moottori pyörähtää 90 astetta, jolloin samanmerkkiset navat ovat taas vastakkain. Syötettäessä energiaa aina seuraavaan käämiin saadaan aikaan saadaan pyörivä liike. [29.]

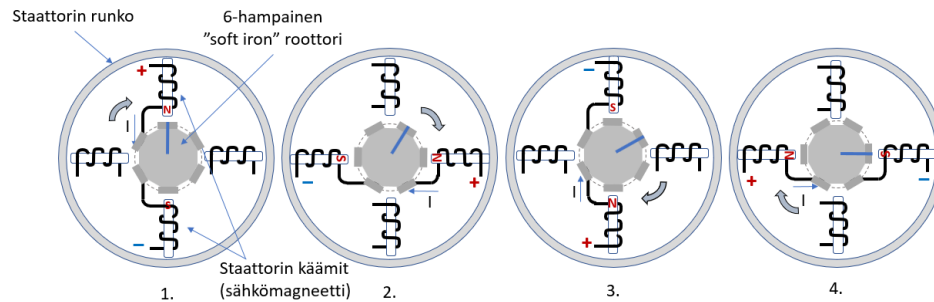


Kuva 13. Kaksivaiheisen kestopagneettiaskelmoottorin toimintaperiaate. [29.]

### Reluktanssiaskelmoottori (VR)

Reluktanssiaskelmoottori koostuu meltorautaisesta hammastetusta pyörijäosasta sekä sähkömagneeteista, jotka on käämitty runko-osaan staattorin ympärille. Meltorauta on helposti magnetisoituvaa matalahiilipitoista rautaa. Moottorin toiminta perustuu siihen, että magneettivirta

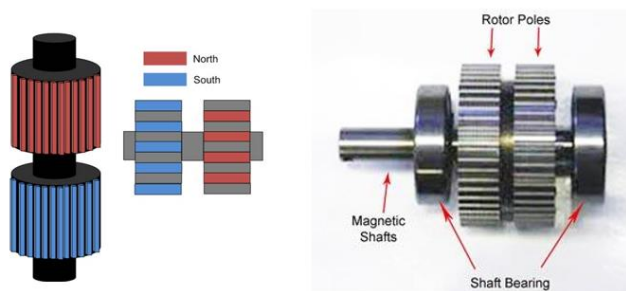
pyrkii kulkemaan sieltä, mistä se helpoiten pääsee. Kun staattorin käämpariin johdetaan tasavirtaa, roottori magnetisoituu ja se roottorin hammas, joka on lähimpänä staattorin sähkömagneetin napoja, pyörähtää niitä vasten. Silloin magneettivirralla on lyhin mahdollinen matka kulkea. Kuvassa 14 on havainnollistettu reluktanssiaskelmoottorin toimintaa. [30].



Kuva 14. Reluktanssiaskelmoottorin toimintaperiaate. [31.]

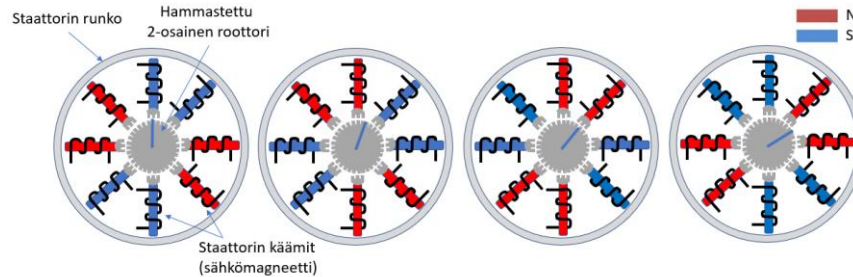
### Hybridiaskelmoottori (HR)

Hybridiaskelmoottori on yhdistelmä kestopagneetti- ja reluktanssiaskelmoottoreista. Siinä roottori koostuu kahdesta meltorautaisesta hammastetusta navasta, joiden sisällä on akselin suuntainen kestopagneetti. Kestomagneetin sijoittaminen akselin suuntaisesti aiheuttaa sen, että toinen roottorin napa asettuu pohjois- (N) ja toinen etelänavaksi (S). Kuvassa 15 on esitetty hybridiaskelmoottorin kaksiosainen roottori. [31.]



Kuva 15. Hybridiaskelmoottorin kaksiosainen roottori [29.]

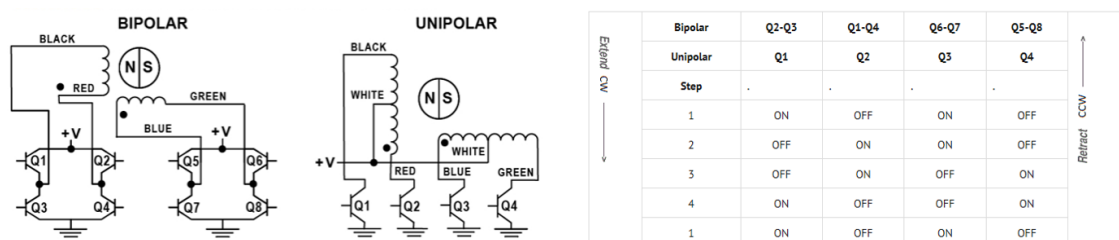
Staattorin sähkömagneetteja virroitetaan pareittain kuvan 16. mukaisesti. Tällöin moottori saadaan pyörimään myötäpäivään. Hybridiaskelmoottorin edut ovat suuri vääntömomentti, korkea liikeresoluutio ja nopeus. [31.]



Kuva 16. Hybridiaskelmoottorin toimintaperiaate. [31.]

### Unipolaariset ja bipolaariset askelmoottorit

Askelmoottoreiden sähkömagneettien käämityksiä on kahdenlaisia, unipolaarisia ja bipolaarisia. Bipolaarisessa kytkennässä on käytössä yksi käämi per vaihe ja unipolaarisessa kytkennässä on kaksi käämiä per vaihe. Bipolaarisella kytkennällä saadaan suurempi vääntömomentti, koska siinä on koko käämi energisoituneena. Bipolaarisen kytkennän ohjaus on vaikeampaa, ja se tarvitsee erillisen ohjauskortin. Unipolaarisessa kytkennässä vain puolet käämistä on energisoituneena, jolloin vääntömomentti on pienempi. Tällöin moottorin ohjaus on helpompaa, koska siinä ei tarvitse muuttaa virran suuntaa ohjauskortilla navan polarisuuden vaihtamiseksi. [29; 33.] Kuvassa 17 on esitetty askelmoottorin unipolaarinen ja bipolaarinen kytkentä.

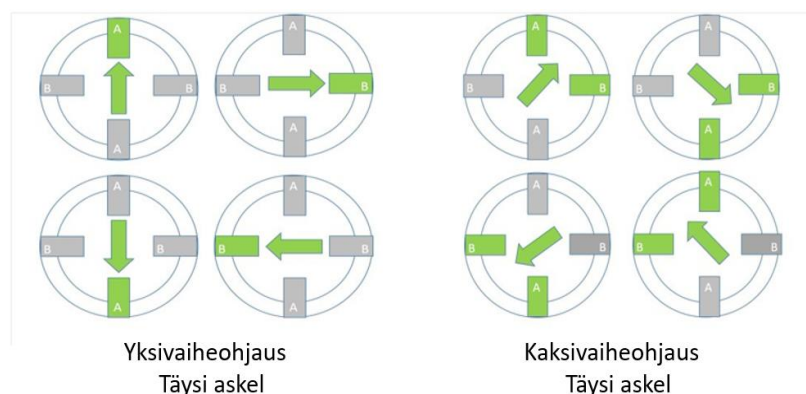


Kuva 17. Askelmoottorin unipolaarinen ja bipolaarinen kytkentä [33].

## Askelmoottorin ohjaaminen

Askelmoottoreita ohjataan joko täydellä-, puoli- tai mikroaskeleella. Askelmoottorit tarvitsevat ohjaukskortin, joka soveltuu kyseisen askelmoottorin ohjaukseen. Bipolaarisille ja unipolaarisille moottoreille on tyypillisesti omat korttinsa. Askelmoottoria ohjataan tyypillisesti ilman takaisinkytkentää. Tällöin paikkatieto haetaan tunnetusta paikasta esim. rajakytkimestä ja liikuttaessa eteenpäin ajettuja askelia laskemalla tiedetään, missä paikassa liikutettava kohde kulloinkin on.

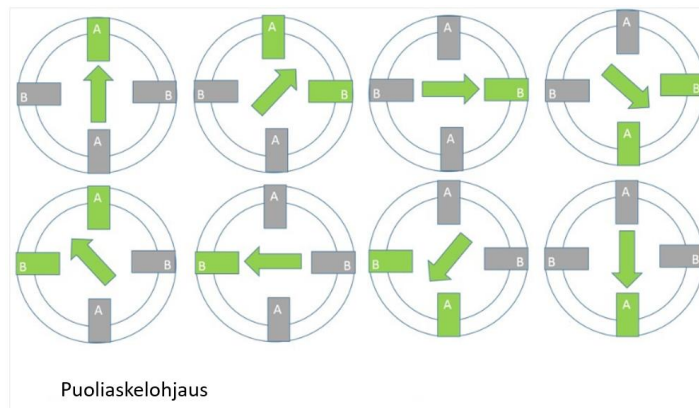
Täysiaskelohjauksessa on kaksi eri moodia, yksivaiheohjaus ja kaksivaiheohjaus. Yksivaiheohjauksessa tasavirta ohjataan yhteen käämipariin kerrallaan. Tämän kytkennän etuna on pienempi virrankulutus kaksivaihe ohjaukseen nähden. Kaksivaiheohjauksessa virroitetaan kaksi vierekkäistä käämiä samalla kertaa. Tämän kytkennän etuna on 30–40 % suurempi vääntömomentti, mutta kytkentä vie kaksi kertaa enemmän tehoa. Kuvassa 18 on esitetty askelmoottorin täysiaskelohjaus. [34.]



Kuva 18. Askelmoottorin täysiaskelohjaus [34].

Puoliaskelohjauksessa askelmoottoria virroitetaan siten, että vuorotellen yksi vaihe virroitetaan ja seuraavaksi kaksi vierekkäistä vaihetta virroitetaan. Virrat pidetään päällä ensimmäisessä vaiheessa ja samaan aikaan seuraava vaiheistetaan. Sitten ensimmäisestä vaiheesta katkaistaan virrat. Tämä ohjaustapa aiheuttaa sen, että moottori kääntyy puolittaisia askelia. Puoliaskelohjaus on yhdistelmä täysiaskelohjauksen yksi- ja kaksivaiheohjausta. Sillä saadaan 15 % pienempi

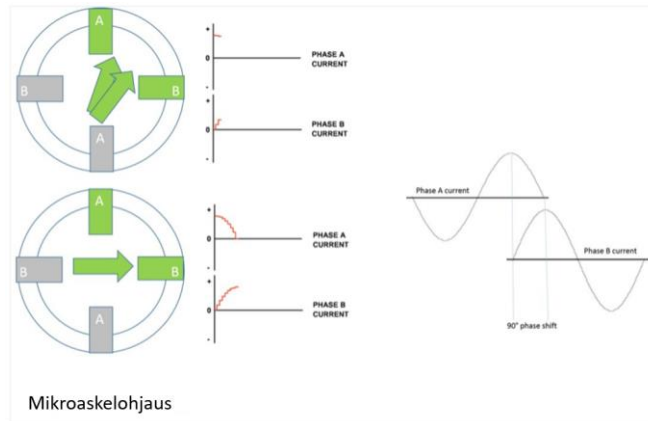
vääntömomentti kuin täyden askeleen kaksivaiheohjauksesta. Puoliaskelohjaus on esitetty kuvassa 19. [34.]



Kuva 19. Askelmoottorin puoliaskelohjaus [34].

### Mikroaskelohjaus

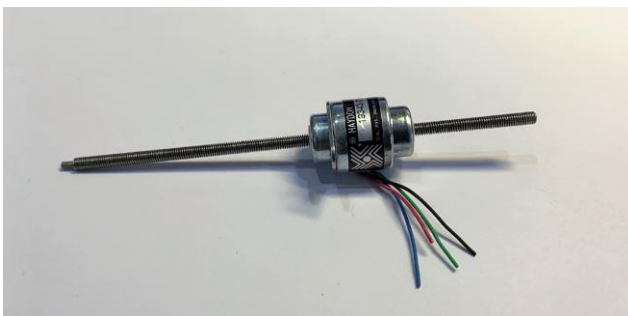
Mikroaskelohjauksessa syötetään 90 asteen vaihesiirrosta olevaa sinimuotoista virtaa moottorin kahteen vierekkäiseen vaiheeseen. Ohjaamalla virran suuntaa ja amplitudia molemmissa käämityksissä resoluutio kasvaa ja moottorin ominaisuudet paranevat, jolloin saadaan vähemmän tärinää ja tasaisempi toiminta. Kun virta kasvaa yhdessä käämissä, se pienenee toisessa käämissä, mikä johtaa tasaiseen askeleen etenemiseen ja vääntömomentin pysymiseen muuttumattomana. Kuvassa 20 on esitetty mikroaskelohjauksen toimintaperiaate. [34].



Kuva 20. Askelmoottorin mikroaskelohjauksen toimintaperiaate [34].

### Työssä käytetty askelmoottori

Tässä insinöörityössä käytetty askelmoottori on Haydon Kerk Motion Solutionsin G4 19000-sarjan ”non captive actuator”-askelmoottori. Askelmoottorin erikoisuus on, että siinä roottorin keskellä on kierteinen tarkkuusmutteri, joka on laakeroitu moottorin kuoreen. Tarkkuusmutterin läpi on asennettu johdinruuvi. Roottorin pyöriessä lineaarinen liike johdetaan suoraan mutterin läpi johdinruuviin. Tämä lähestymistapa yksinkertaistaa lineaaristen sovellusten suunnittelua ja mahdollistaa suuren tarkkuuden. Askelmoottori on bipolaarinen eli siinä on yksi käämi per vaihe. Yhden askeleen askelkulma on 7,5 astetta ja yksi askel liikuttaa kierretankoa 0,013 mm. Käämitysjännite on 5 V. Moottori on pienikokoinen. [32.] Kuvassa 21 on esitetty työssä käytetty askelmoottori.



Kuva 21. Insinöörityössä käytetty Haydon Kerk Motion Solutionsin G4 19000-sarjan askelmoottori.

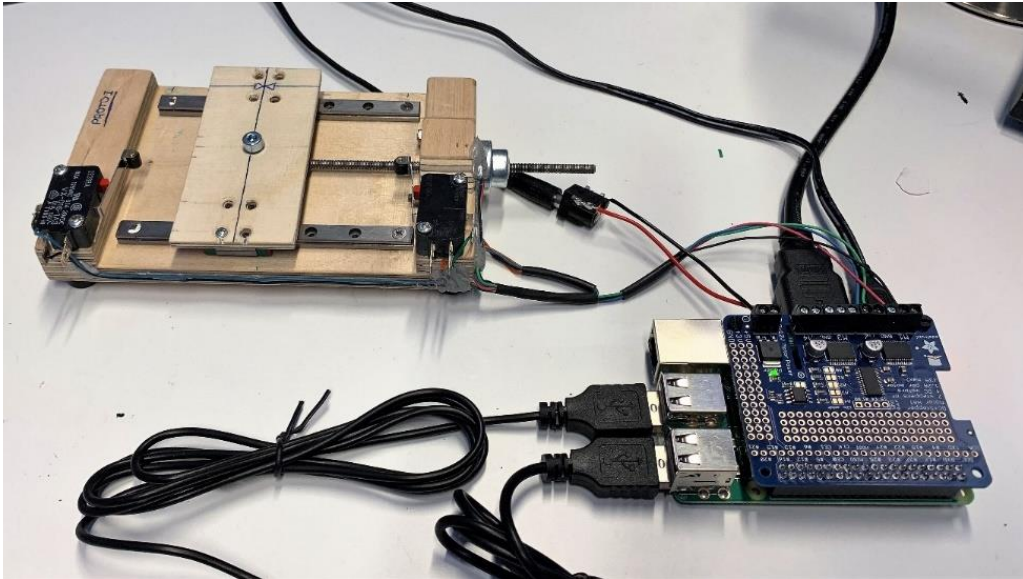
## 5 Toteutus

### 5.1 Prototyypin rakentaminen

Rakennettava testilaite koostuu Raspberry-tietokoneesta ja siihen liitettävästä moottorinohjauskortista, ohjelmakoodista, lineaarikiskoista kelkkoineen, rajakytkimistä, askelmoottorista sekä runko-osasta. Koska Raspberry-tietokoneen käytöstä ja sen avulla moottorin ohjaamisesta ei ollut aikaisempaa kokemusta, päätettiin tehdä ensin yksinkertainen prototyyppi, jolla testattiin liikkeen muodostamista ja toistotarkkuutta.

Työ aloitettiin Raspberry-tietokoneen ja moottorinohjauskortin hankinnalla. Raspberryyhin ladattiin vaadittavat ohjelmistot sekä asennettiin moottorin ohjauskortti. Tarvittavien komponenttien osalta tavoitteena oli käyttää tarveosina yrityksen valmistamien laitteiden osia, joita myydään myös varaosina. Tämä siksi, että varaosille on tällöin taattu pitkä saatavuus ja vauriotilanteessa voidaan taata nopea varaosien saatavuus. Laitteen liike toteutettiin pienten lineaarikiskojen ja niihin asennettavien välyksettömien ja kuulalaakeroitujen liukukelkkojen avulla. Liukukelkkojen päälle asetettiin yhtenäinen pohjalevy, johon kiinnitettiin teline viuhkafantomia varten.

Prototyyppi rakennettiin vanerilevyille, johon muotoiltiin paikat askelmoottorille ja rajakytkimille. Lineaarikiskojen asentaminen riittävän tarkasti samaan linjaan oli työlästä. Moottori johdotettiin moottorinohjauskortin ruuviliittimiin ja rajakytkimet johdotettiin Raspberryn GPIO-portteihin 4 ja 18, jotka määritettiin lukumoodiin ja alasetovastuksilla maihin lepotilassa. Liukukelkkoihin kiinnitetty pohjalevy rakennettiin vanerista ja se mitoitettiin siten, että rajakytkimet osuivat pohjalevyyn pysäyttäen liikkeen. Kuvassa 22 on esitetty testilaitteen prototyyppi.

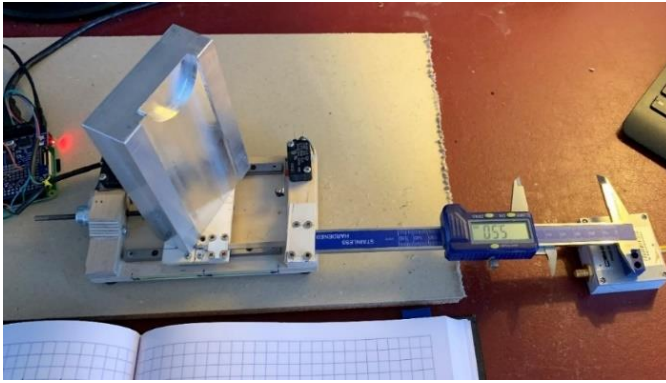


Kuva 22. Testilaitteen prototyyppi

Proton rakentamisen jälkeen alkoi Python-ohjelman opettelu. Ensin tarvittavat kirjastot ladattiin ohjelmaan ja sitten opeteltiin moottorin askelten syöttö. Moottori totteli hyvin ohjauskoodeja ja rajakytkinten tieto saatiin luettuja oikea-aikaisesti. Laitteen testauksessa pohjalevyn päälle asennettiin laitteen kokoon nähden raskas, noin 0,5 kg:n painoinen alumiinilevy kuormaksi, jotta voitiin todeta, hukkuiko moottorin askelia liian suuren kuorman vaikutuksesta. Laitteen liiketarkkuutta testattiin kiinnittämällä digitaalinen työntömitta laitteeseen siten, että työntömitan runko kiinnitettiin laitteen runkoon ja työntömitan liikkuva sisäosa kiinnitettiin laitteen kelkkaan. Työntömitan valmistaja lupaa työntömitalle 0,03 mm:n mittaustarkkuuden, jonka todettiin olevan riittävä tutkittaessa, päästäänkö vaadittuun 0,1 mm:n tarkkuuteen. [35.]

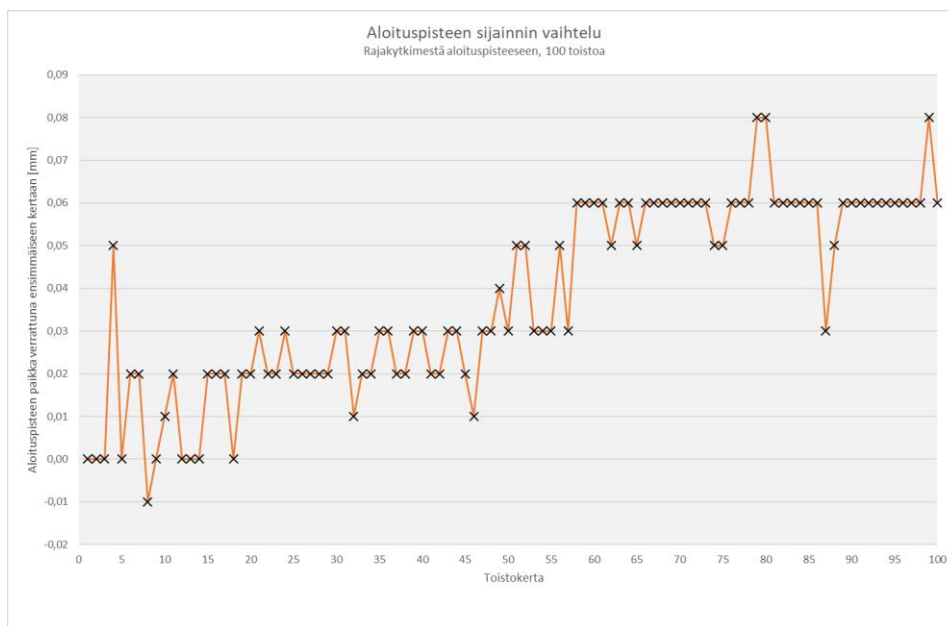
Laitetta testattiin siten, että liikkeen keskikohta määritettiin ja tämä kohta asetettiin kotipaikaksi. ”Home”-komennolla laite ensin etsii moottorin puoleisen rajakytkimen ja liikkuu siitä eteenpäin aloituspaikkaan. Tätä komentoa toistettiin sata kertaa peräkkäin. Ensimmäisellä kerralla, kun laite oli saavuttanut kotipaikan, työntömitan asteikko nollattiin. Toistomittauksen aikana, kun kelkka saavutti kotipaikan, työntömitan lukema kirjattiin ylös. Näin kerättiin tietoa kotitilan paikan vaihtelusta 100 toiston aikana. Kuvassa 23 on kuva toistomittauksen suorittamisesta.





Kuva 23. Ensimmäisen prototyypin aloituspaikan tarkkuuden määrittäminen käynnissä.

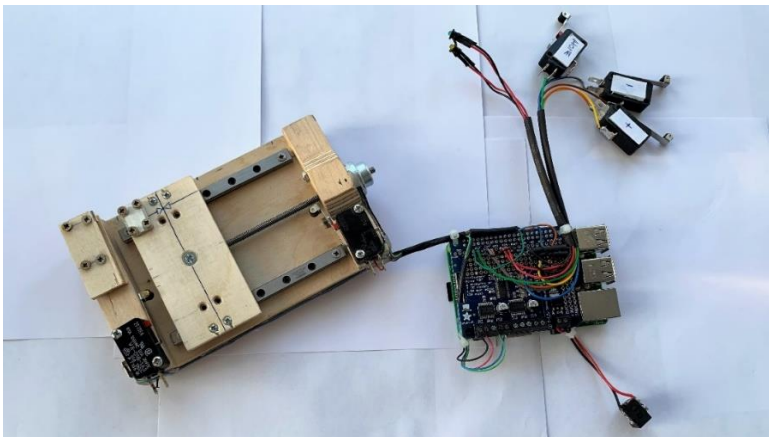
Toistotarkkuuden todettiin olevan riittävä eikä askeleita hukkunut, vaikka kelkan päällä pidettiin ylisuurta painoa kuormittamassa kelkkaa. Mittalaite nollattiin vain testin alussa, jolloin mittalaitteen epätarkkuus ja hystereesi vaikuttavat huonontavasti mittaustuloksen. Kuten kuvassa 24 esitetyistä mittaustuloksista voidaan todeta, kotipaikan sijainnin tarkkuus pysyi koko testin aikana alle vaaditun 0,1 mm:n. Tällöin mittalaitteen epätarkkuutta eikä hystereesiä tarvitse huomioida.



Kuva 24. Toistomittauksen tulokset

Mittauksen aikana askelmootorin huomattiin kuumenevan, ja tälle alettiin etsiä syytä. Liikettä testattiin myös suunnanmuutoksen osalta. Testi tehtiin siten, että kelkkaa ajettiin 1 mm:n välein ensin 10 mm eteenpäin kotipaikasta, suuntaa muutettiin ja ajettiin vastakkaiseen suuntaan 20 mm:n matkan, suuntaa muutettiin ja ajettiin 10 mm takaisin kotipaikkaan. Tämä toistettiin neljä kertaa. Mittaustuloksista huomattiin, että suunnan vaihdon yhteydessä joitain askelia hukkuu moottorin läpi kulkevan ruuvin pieneen välykseen. Tämä korjattiin ohjelmallisesti lisäämällä askelia moottorin komentoon aina, kun suunnan muutos tehtiin.

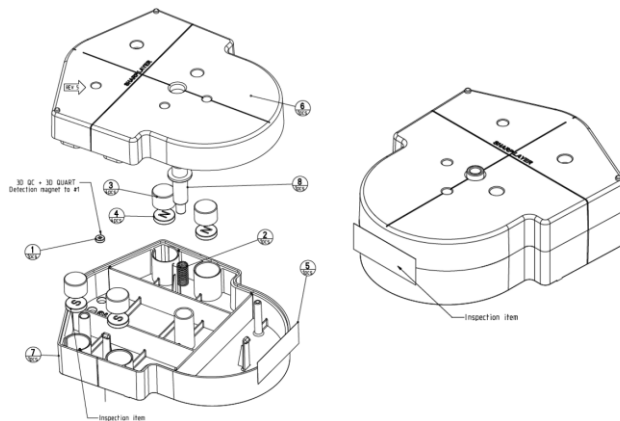
Prototyyppi todettiin toimivaksi, joskin edellä mainitut huomiot tuli korjata. Lopullisen laitteen tuli toimia ilman näyttöä, näppäimistöä tai hiirtä, joten laitteeseen oli saatava omat käyttökytkimet aloituspaikan eli kotipaikan hakemiseen sekä eteen- ja taaksepäin liikutukseen. Myös merkkivalot haluttiin ilmaisemaan sekä moottorin liikettä että laitteen ohjelman käynnissä oloa. Näitä varten otettiin käyttöön GPIO-portit. Portti 17 varattiin kotipaikan hakemiseen, portti 22 varattiin eteenpäin- ja portti 27 taaksepäin liikkeelle. Portti 24 määritettiin liikkeen merkkivalolle ja portti 25 ohjelman käynnissä olon merkkivalolle. Kuvassa 25 johdotukset asennettuna.



Kuva 25. Kytkimet ja merkkivalot asennettuna prototyyppiin

## 5.2 Leukatukeen asennettavan prototyypin rakentaminen

Seuraavaksi aloitettiin lopullisen laitteen suunnittelu. Laitteen runkona päätettiin käyttää geometriakalibroinnissa käytettävän kuulafantomien leukatukeen kiinnitettävää runko-osaa, joka on esitetty kuvassa 26.

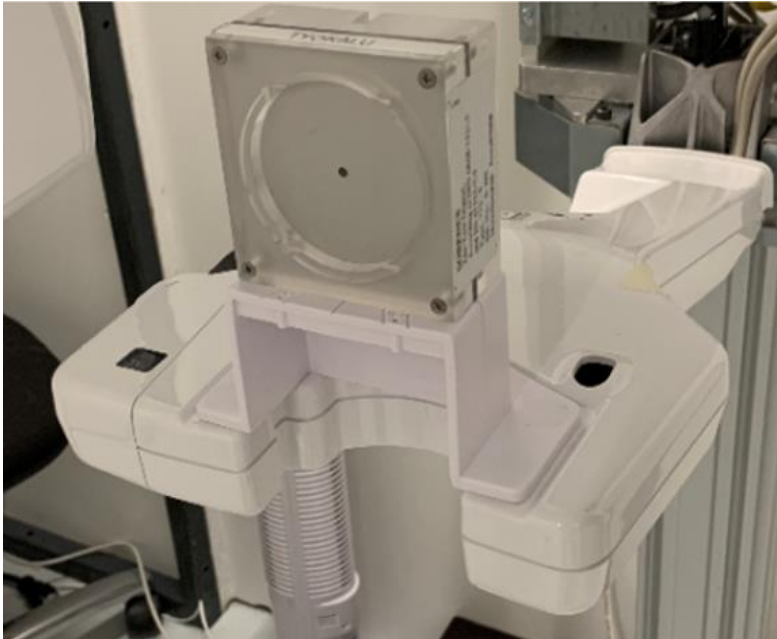


Kuva 26. Kuulafantomien runko-osa

Laitetta tehtiin kaksi kappaletta, harjoittelu- ja lopullinen malli. Tämä tehtiin siksi, että moottorin pelättiin sijoittuvan liian korkealle ja aiheuttavan heijastuksia röntgenkuvaan. Tämän tutkimiseksi päätettiin ensin tehdä toimiva harjoituskappale, johon asennettiin Raspberry Pi-tietokone ja moottorin ohjauskortti rungon sisään. Tietokone ja moottorin ohjauskortin todettiin mahtuvan rungon sisään pienin muutoksin ja vielä siten, että haluttuihin liittimiin saatiin kaapelit kytkettyä. Linearikiskot asennettiin rungon kanteen, mihin myös moottori upotettiin. Rajakytkimet vaihdettiin pienempiin ja kaikki johdotukset tehtiin rungon sisään.

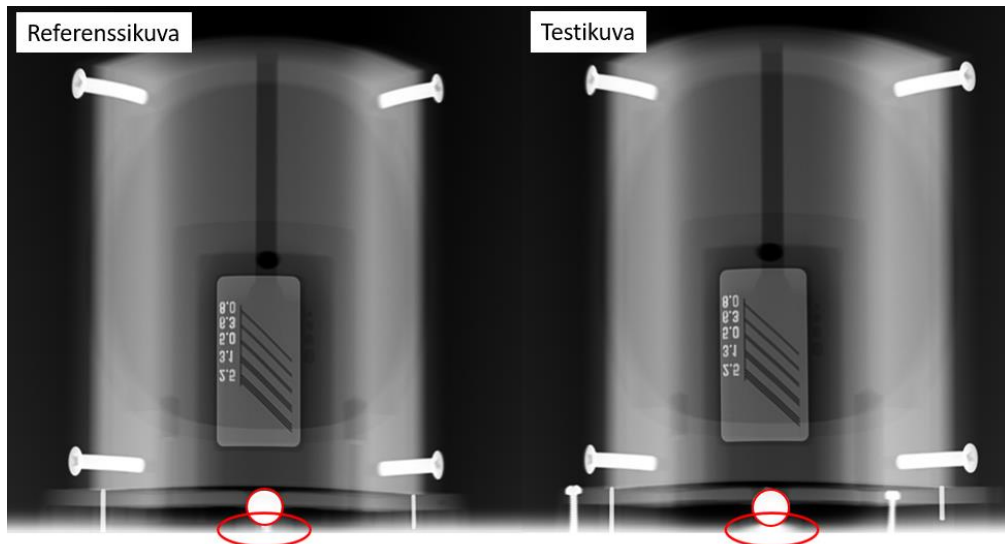
Raspberryn asetuksia muutettiin siten, että kun siihen kytki virrat, niin Python-ohjelma lähti käyntiin automaattisesti ilman näppäimistöä, hiirtä ja näyttöä. Laitteen etulevyyn asennettiin vihreä led-merkkivalo ilmaisemaan, että ohjelma on käynnissä ja keltainen led-merkkivalo ilmaisemaan moottorin ajoa. Laitetta testattiin ensimmäisen kerran lopullisessa käyttöympäristössään. Ensin röntgenlaitteen leukatukeen asennettiin kuvanlaatumittauksissa käytetty viivaparifantomi omalla telineellään, jonka avulla otettiin referenssiröntgenkuva. Sitten

viivaparifantomi asennettiin rakennettuun testilaitteeseen ja röntgenkuva otettiin. Kuvassa 27 viivaparifantomi on asennettuna kuvanlaatukalibroinnissa käytettävään telineeseen.



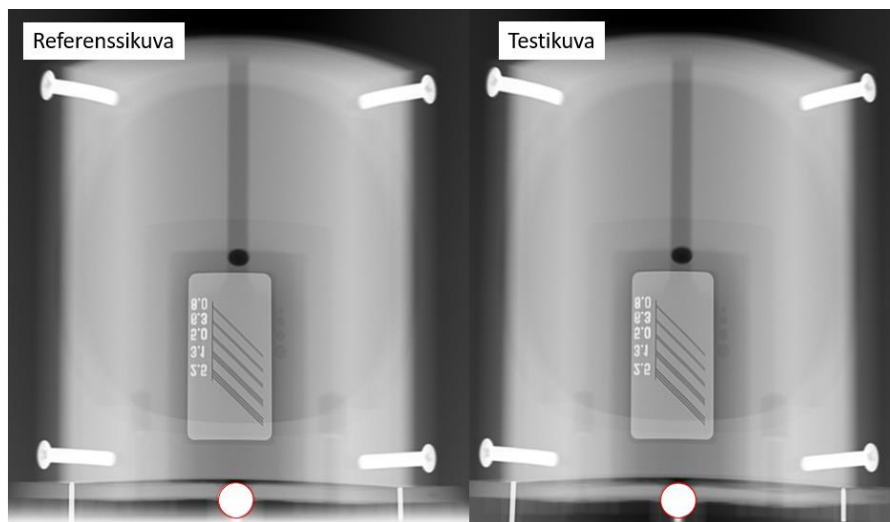
Kuva 27. Viivaparifantomi asennettuna kalibroitintelineeseen.

Saatuja kuvia verrattiin keskenään ja erityistä huomiota kiinnitettiin kuvan alareunassa näkyvään 6 mm:n teräskuulan ympäristöön sekä kuulan pyöreyyteen, sillä moottorista mahdollisesti aiheutuvat heijastukset muuttaisivat juuri tätä aluetta. Kuvassa 28 on esitetty ensimmäisen testin tuloksena saadut röntgenkuvat. Kuten kuvasta näkyy, testilaitteen kanssa otetussa kuvassa kuulan alareuna peittyy varjostuksesta. Moottoria oli saatava alemmaksi tai fantomia vastaavasti ylemmäksi.



Kuva 28. Ensimmäinen vertailuröntgenkuva, josta näkyy, että testikuvan kuulan alareuna peittyy moottorin metallikuoren aiheuttamasta heijastuksesta.

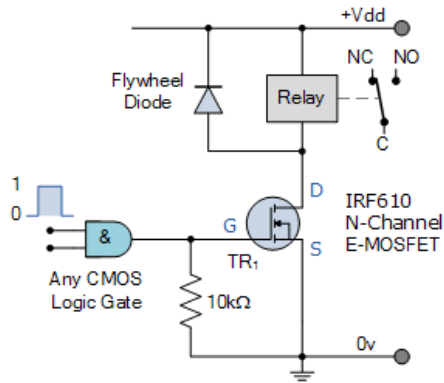
Seuraavaksi laitteen runkoa madallettiin ja jalustalevyä nostettiin samassa suhteessa siten, että asennettavan viivaparifantomien korkeus pysyi vaaditun 5 mm:n tarkkuudella samana kuin vakiokalibrointijalustalla. Sama testi toistettiin. Testikuvasta voitiin havaita, että heijastukset olivat poissa ja kuulat aivan pyöreitä. Kuvassa 29 on esitetty röntgenkuvat muutostyön jälkeen.



Kuva 29. Vertailuröntgenkuva, josta ilmenee, että testikuvan kuula näkyy pyöreänä ilman heijastuksia.

Testin perusteella katsottiin voitavan jatkaa nykyisellä suunnitelmalla ja keskityttiin seuraavaan haasteeseen, moottorin kuumentumiseen. Todettiin, että moottori ei ehdi kuumentua ajon aikana, vaan pidossa. Askelmoottorin ollessa pidossa virta kulkee jatkuvasti moottorin käämin läpi. Työssä käytetty vakiovirtaohjauskortti antaa 1,2 A:n virtaa moottorille. Häviöiden vaikutuksesta moottori lämpenee. Kehittyneimmissä moottorinohjauspiireissä moottorin virtaa pystytään rajoittamaan ohjauspiirin avulla, mutta halvoissa ja yksinkertaisissa moottorinohjauspiireissä tätä mahdollisuutta ei ole, jolloin ohjaimen syöttämä maksimivirta kulkee jatkuvasti käämien läpi pitotilassa. Vaihtoehtoina oli vaihtaa moottorin ohjauskortti laadukkaampaan tai sitten keksiä jotakin, millä moottorin virrat saataisiin rajoitettua. Tiedonhaun perusteella sellaisia moottorinohjauskortteja, jotka sopisivat suoraan Raspberryn päälle asennettavaksi, joissa olisi vaadittu virranrajoitusominaisuus, jotka tottelisivat I2C-väylää ja jotka mahtuisivat käytettävän kotelon sisään, ei löytynyt. Päätettiin pitäytyä nykyisessä kortissa ja keksiä muu tapa moottorin lämpenemisen rajoittamiseen.

Yksinkertaisin tapa on katkoa moottorin käämien virtapiirit silloin, kun moottori ei liiku. Käytettävän askelmoottorin rakenne on sellainen, jossa moottorin runko on asennettu kiinteästi laitteen runkoon ja moottorin läpi kulkeva johdinruuvi on asennettu kiinteästi kelkkaan. Tällöin ainoa liikkuva osa on moottorin sisällä oleva roottori. Liikuteltavat massat ovat pieniä eikä ole pelkoa, että kelkka liikahtaisi esimerkiksi tärähdyksen voimasta, koska kelkan liikkumiseen tarvitaan roottorin pyörivää liikettä. Tällöin ei ole tarvetta pitää pitovirtoja kytkettyinä silloin, kun laitetta ei ajeta ja tämän perusteella päätettiin katkoa moottorin virtapiiriä releellä. Yksi GPIO-portti otettiin käyttöön releen ohjaamiseen, portin ja releen ohjauskäämin väliin asetettiin FET-kanavatransistori sekä suojadiodi. Ohjelmaan rakennettiin ajastinpiiri, joka katkaisi askelmoottorin virtapiirin liikkeen loputtua ja kytki virrat takaisin liikutuskäskyn saavuttua. 0,3 sekunnin viive lisättiin napin painalluksen jälkeen, jotta rele ehtii mukaan eikä askelia hukata releen asettamisen aikana. Testeissä todettiin kytkennän toimivan eikä moottori käynyt kuumana. Kuvassa 30 on esitetty releen ohjauskytkentä. [36.]



Kuva 30. Releen ohjauskytkentä [36].

Seuraava ongelma oli laitteen sammuttaminen ja sen jälkeen uudelleen käynnistäminen. Kun laitteeseen ei enää tarvittu näyttöä, näppäimistöä tai hiirtä, oli ratkaistava, kuinka laite sammutettaisiin. Laitetta ei voida sammuttaa vetäisemällä muuntaja seinästä, vaan tietokone tulee sammuttaa hallitusti. Raspberryyn on valmis Python-ohjelmakomento laitteen sammuttamiseen, joten laitteen takaosaan lisättiin kaksiosainen palautuva vaihtokytkin tätä varten. Sammutustoimintoon varattiin GPIO-portti 5, joka luki kytkimen tilaa. Kun nouseva reuna havaittiin, ajettiin sammutuskomento. Laitteen uudelleen käynnistämiseksi haluttiin käyttää samaa liukukytkintä. Raspberry saadaan käynnistymään, kun reset-nappia painetaan. Tähän tarkoitukseen Raspberryssä on paikat johtimille, jotta ulkoista kytkintä voidaan käyttää. Nämä johtimet kytkettiin vaihtokytkimen toiseen linjaan siten, että kytkimen painallus oikosulkee nämä johtimet ja tietokone käynnistyy.

### 5.3 Testilaitteen rakentaminen

Lopullisen laitteen runkona päätettiin myös käyttää kuulafantomien leukatukeen kiinnitettävää runko-osaa. Osaa muokattiin tarkoitukseen sopivaksi madaltamalla koteloa niin paljon kuin mahdollista, jotta Raspberry moottorinohjainkortteineen mahtui juuri ja juuri sisään. Lopullinen laite on kopio aiemmasta prototyypistä, mutta rakennettu uusista osista ja suurempaa tarkkuutta käyttäen. Liukukelkkojen päälle asennettava pohjalevy tehtiin alumiinista jäykkyyden lisäämiseksi. Moottori asennettiin niin alas kuin mahdollista, jotta röntgenkuvaan ei tule häiriöitä.

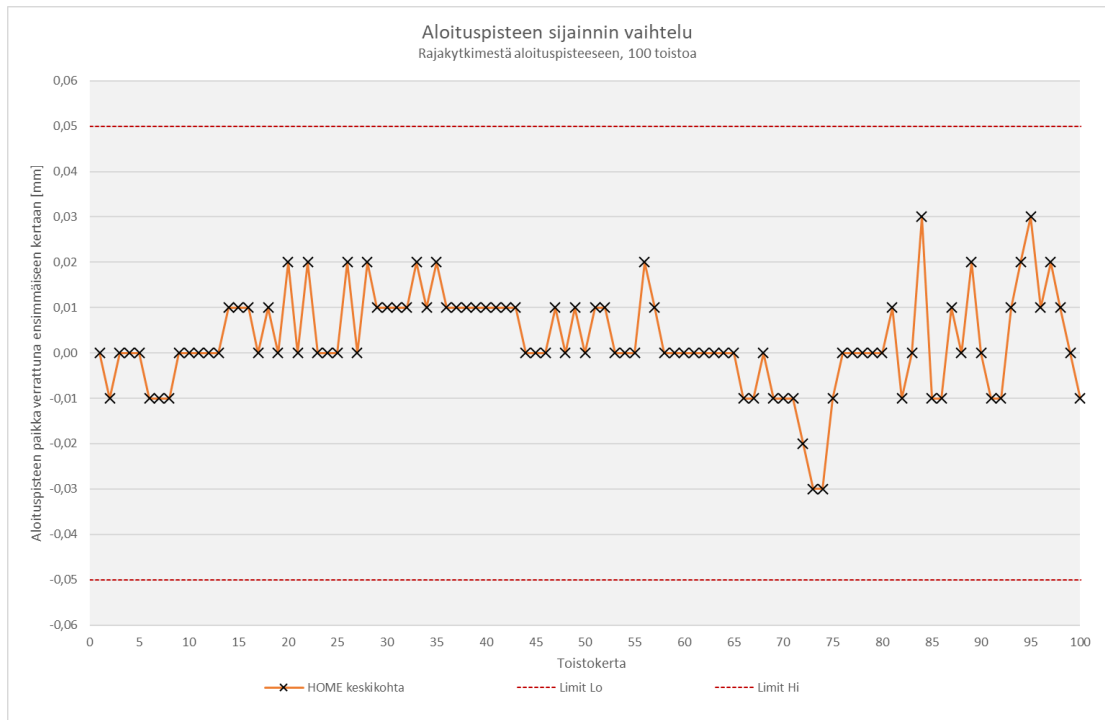
Sammutus/käynnistys-kytkin asennettiin laitteen takareunaan ja muut kytkimet merkkivaloineen laitteen etureunaan. Moottorin kiinnitys toteutettiin käyttäen valmiita osia. Näin toimien saatiin viuhkafantomien alareunan ja moottorin yläreunan etäisyyttä kasvatettua 4 mm:iin ja eliminoitua metallista aiheutuvien heijastusten vaikutus röntgenkuvaan.

## 6 Testaus

Kun laite saatiin valmiiksi ja todettiin toimivaksi, aloitettiin lopullinen testaus. Testilaitetta testattiin suhteessa työn aloituksessa asetettuihin vaatimuksiin. Vaatimusmääritysten mukaisesti laitteen tarkkuuden tulisi olla parempi kuin 0,1 mm. Aloituspisteen oletetaan olevan terävän kerroksen keskellä. Terävän kerroksen leveys etuhampaiden kohdalla on tyypillisesti noin 10–15 mm, joten laitteen liikkeen tulee olla vaadituissa rajoissa vähintään tuolla matkalla. Testilaitteelle rakennettiin puusta tukeva jigi, johon testilaitte asennettiin. Jigin runkoon asennettiin korotuspala, johon kalibroitu työntömitta kiinnitettiin runko-osasta. Työntömitan luisti kiinnitettiin testilaitteen kelkkaan.

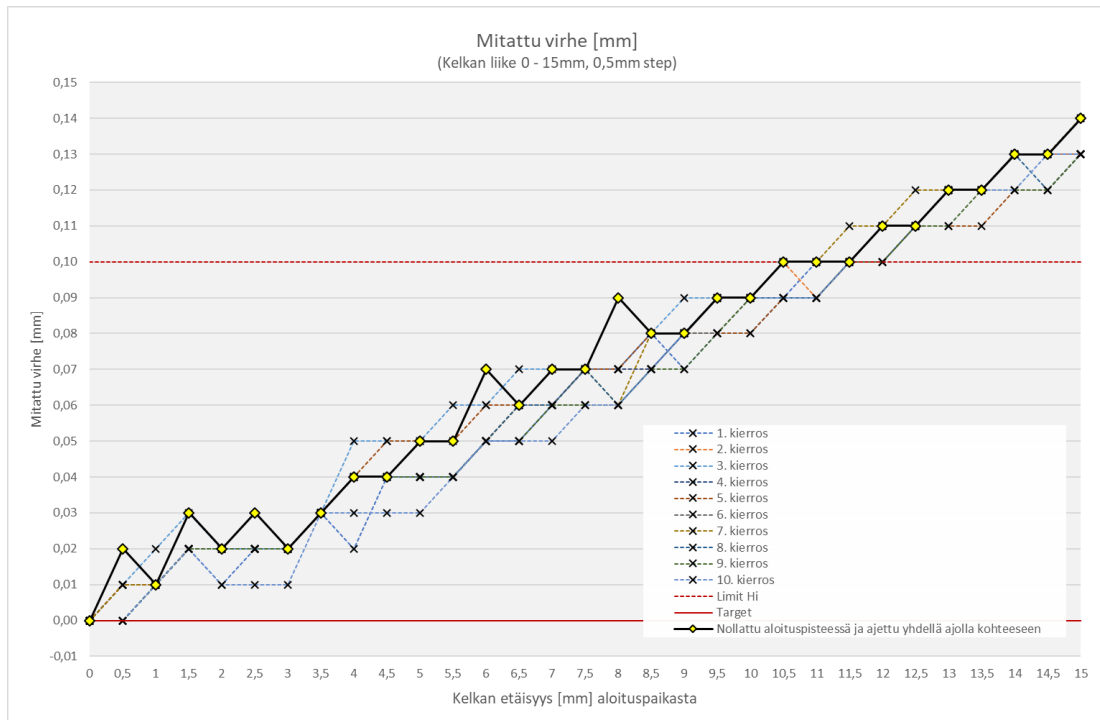
Ensin testattiin aloituspisteen paikan tarkkuus. Tämä mitattiin siten, että kelkka ajettiin ensin rajakytkimeen (Raja1) ja sieltä aloituspaikkaan. Ensimmäisellä kerralla työntömitan näyttö nollattiin ja sen jälkeen liike toistettiin sata kertaa. Aina, kun kelkka saavutti aloituspaikan, työntömitan lukema kirjattiin. Aloituspaikan tarkkuudeksi saatiin 0,06 mm. Testissä saavutettiin 0,1 mm:n vaatimus liikkeen tarkkuudelle. Kuvassa 31 on esitetty mittaustulokset.





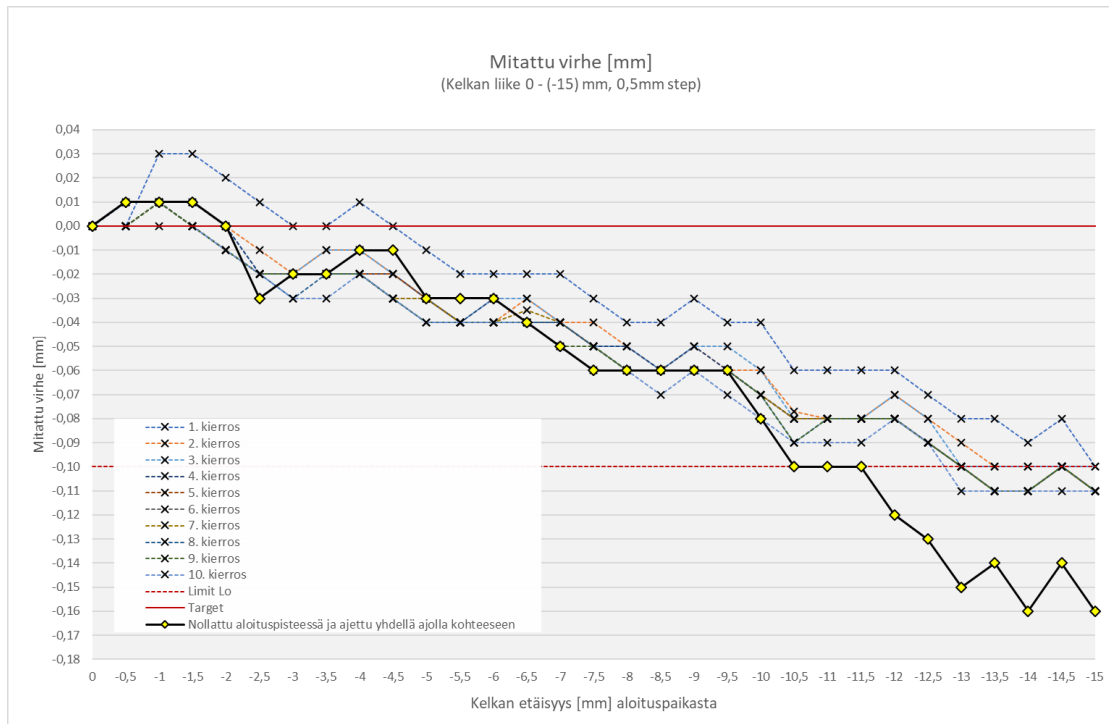
Kuva 31. Aloituspisteen paikan tarkkuuden mittaustulokset.

Seuraavaksi mitattiin liikkeen tarkkuus, kun liike aloitettiin aloituspaikasta. Testissä aloituspaikasta liikuttiin 0,5 mm:n askelin 15 mm:n etäisyyteen saakka. Aloituspisteessä työntömitan näyttö nollattiin ja joka liikkeen jälkeen mittalaitteen näyttö kirjattiin. Testi toistettiin kymmenen kertaa. Tuloksista havaittiin, että vaaditulla 0,1 mm:n tarkkuudella voitiin liikuttaa kelkkaa aloituspisteestä 11 mm eteenpäin. Mittaus toistettiin vielä siten, että kelkka ajettiin ensin aloituspisteeseen ja siitä suoraan loppupisteeseen jokaisen pisteen kohdalla. Tämä tarkoittaa sitä, että esimerkiksi 15 mm:n pisteessä ajettiin kelkka ensin aloituspisteeseen ja siitä suoraan 15 mm:n kohtaan. Mittaustuloksia verrattiin ja tuloksista havaittiin, aiheuttiko työntömitta mittausrvirhettä liikuttaessa askeleittain. Mittaustulokset seurasivat askelmittauksen tuloksia, joten voitiin todeta, että työntömitta ei aiheuta kasvavaa virhettä mittaustuloksiin. Kuvassa 32 on esitetty kelkan liikkeen tarkkuusmittauksen tulokset, kun kelkkaa liikutettiin aloituspaikasta eteenpäin.



Kuva 32. Kelkan liikkeen tarkkuusmittauksen tulokset, kun kelkkaa liikutettiin aloituspaikasta eteenpäin.

Edellinen mittaus toistettiin siten, että kelkkaa ajettiin 15 mm taaksepäin 0,5 mm:n askelin. Tuloksista havaittiin, että vaaditulla 0,1 mm:n tarkkuudella voitiin liikuttaa kelkkaa aloituspisteestä 11,5 mm taaksepäin. Tulosten mukaan kelkan liikealue vaatimusten mukaisella tarkkuudella on +/- 11 mm eli kokonaisuudessaan 22 mm. Kuvassa 33 on esitetty kelkan liikkeen tarkkuusmittauksen tulokset, kun kelkkaa liikutettiin aloituspaikasta taaksepäin.



Kuva 33. Kelkan liikkeen tarkkuusmittauksen tulokset, kun kelkkaa liikutettiin aloituspaikasta taaksepäin

Vaatimusten mukaisesti kelkkaan tulee voida asentaa sekä viivapari- että viuhkafantomi eikä näiden asennuskorkeus saa olla yli 5 mm korkeammalla, kuin QC-mittauksessa käytetyn fantomin asennuskorkeus. Kelkkaan voidaan asentaa molemmat fantomit ja asennuskorkeuden mitattu ero on 4,3 mm, joten vaatimus toteutui.

Vaatimusten mukaisesti laitteen ohjelmiston tulee käynnistyä automaattisesti laite käynnistettäessä ilman näyttöä, näppäimistöä ja hiirtä. Tämä toteutettiin siten, että Raspberryn Python-ohjelman polku lisättiin rc.local-tiedostoon. Raspberryn käynnistyessä kaikki ohjelmat, joiden sijaintipolku on rc.local-tiedostossa, käynnistyvät.

## 7 Yhteenveto

Insinööriyössä suunniteltiin ja toteutettiin panoraamaröntgenlaitteen leukatukeen kiinnitettävä testilaite, jonka avulla röntgenkuvan terävä alue voidaan määrittää. Laitteeseen voidaan kiinnittää viuhkafantomi tai viivaparifantomi ja fantomia voidaan liikuttaa tarkasti syvyysuunnassa vaaditulla tarkkuudella. Toteutetun laitteen avulla korvattiin aiemmin käytössä ollut manuaalinen ja epätarkka liikutuslaitteisto. Työn keskeinen tavoite oli aloittaa terävän kerroksen mittaamisen automatisointi. Terävän kerroksen mittauksen automatisointi vaatii automaattisen fantomin liikutuksen, röntgenkuvan analysoinnin konenäön avulla ja röntgenlaitteen ohjaamisen. Opinnäytetyössä toteutettiin automaattinen fantomin liikutus.

Työssä käytettiin Raspberry Pi-minitietokonetta, jonka käyttöön automatisoinnissa tutustuttiin. Työssä myös opiskeltiin Python-ohjelmointikieltä ja sen avulla toteutettiin tarvittavat ohjelmakoodit. Askelmoottorin toimintaan perehdyttiin ja laitteen liike toteutettiin opitun mukaisesti laitteeseen parhaiten sopivan moottorin avulla.

Testilaitteen prototyyppejä rakennettiin kolme kappaletta. Ensimmäisen prototyypin avulla testattiin moottorin soveltuvuutta laitteeseen sekä liikkeen tarkkuutta. Moottorin todettiin soveltuvan hyvin liikkeen aikaansaamiseen sekä asennettavuuden että liikkeen tarkkuuden puolesta. Moottorin todettiin myös jaksavan liikuttaa vaadittavaa kuormaa. Toisen prototyypin avulla testattiin laitteen yhteensopivuutta panoraamaröntgenlaitteen kanssa. Lisäksi tutkittiin aiheuttako laite häiriöitä röntgenkuvaan. Laitteen perusrunkona käytettiin valmiina olevaa kuulafantomien runko-osaa, joka muokattiin käyttötarkoitukseen sopivaksi. Moottori asennettiin aluksi liian korkealle, jolloin röntgenkuvaan tuli heijastuksia. Tämä korjattiin madaltamalla laitteen runko-osaa ja korottamalla kelkan päällä olevaa fantomin jalustaosaa. Näin saatiin häiriöt röntgenkuvasta eliminoidua. Kolmas prototyyppi rakennettiin niin huolella, että siitä muodostui lopullinen testilaite.

Lopullisesta laitteesta mitattiin aloituspisteen toistotarkkuus ja liikutus aloituspisteestä askeleittain sekä eteen- että taaksepäin. Toistotarkkuusmittaus suoritettiin ajamalla kelkka sata kertaa rajakytkimeen ja takaisin aloituspaikkaan. Mittauksista voitiin nähdä, että toistotarkkuus oli parempi kuin vaadittu 0,1 mm. Toistotarkkuus mitattiin myös lähettäessä aloituspisteestä

sekä eteenpäin että taaksepäin. Mittaustulosten mukaan kelkan liikealue vaatimusten mukaisella tarkkuudella on +/- 11 mm, mikä on suurempi kuin vaatimuksena ollut terävän kerroksen paksuus. Suuria ongelmia työn toteuttamisessa ei ollut. Vähäisiä haasteita oli moottorin liiallisen lämpenemisen estäminen sekä laitteen hallittu sammutus ja sammutuksen jälkeinen käynnistäminen. Moottorin lämpeneminen hoidettiin katkomalla moottorin virtoja silloin, kun moottoria ei käytetty. Laitteen sammuttamiseksi laitteeseen asennettiin virtakytkin, jota ohjelma luki. Sammutusvipua käännettäessä ohjelma suorittaa sammutuskomennon ja laite siirtyy lepotilaan. Sammutuksen jälkeinen käynnistys hoidettiin saman vipukytkimen toisella asennolla, joka käynnisti laitteen maadoittamalla Raspberryn resetointipinnan, jolloin laite käynnistyy.

Laitteen käyttöönotossa testilaitteen avulla määritettiin panoraamaröntgenlaitteen terävän kerroksen paikka ja laajuus. Laite suoriutui tehtävästään suunnitellusti. Mittauksen aikana havaittiin, että vaatimusten määrittelyssä olisi pitänyt osata lisätä laitteen vaatimuksiin myös näyttö, joka kertoo, missä kohdassa kelkka kullakin hetkellä on. Koska näyttöä ei ollut, mittausten aikana tuli pitää kirjaa kelkan etäisyydestä aloituspisteeseen, jotta mittaus ei mene sekaisin. Laitetta tullaan käyttämään osana suurempaa automaatiojärjestelmää ja paikkatiedon laskeminen on helppo lisätä laitteen ohjelmaan, mutta se tehdään tämän insinööriyön ulkopuolella.

Insinööriyössä toteutettiin kokonainen laite alkusuunnittelusta lopputestaukseen ja käyttöönottoon. Työn edetessä opittiin paljon uusia asioita mm. askelmoottorin toiminnasta ja ohjaamisesta, Python-ohjelmoinnista, elektroniikasta sekä röntgenlaitteen toiminnasta. Oppiminen ja kokonaisuuden ymmärtäminen lisääntyi. Työn edetessä joutui miettimään ja selvittämään eri asioita, joita ei osannut työn alkaessa ottaa huomioon. Esimerkiksi Python-ohjelmoinnissa oli varsin yksinkertaista saada laitteen kelkka liikkumaan, mutta kelkka piti myös saada pysähtymään rajakytkimeen ja liike piti toimia takaisinpäin rajakytkimen aktivoitua. Ohjelman laatimisessa piti varautua myös käyttäjän toimintaan, kuten liikutusnapin painaminen liikkeen aikana. Poikkeustiloihin varautuminen toi paljon lisäivejä ja testausta ohjelmakoodiin. Myös lopputestaus ja liikkeen hienosäätö vastaamaan tilaajan tarpeita oli opettavaista. Kaiken kaikkiaan tämän insinööriyön tekeminen oli erittäin mielekästä.

## Lähteet

- 1 Yritys. 2019. Verkkoaineisto. Kavokerr. <<https://www.kavokerr.com/fi-fi/yritys>>. Luettu 29.1.2019.
- 2 ORTHOPANTOMOGRAPH™ OP 3D User and Installation Manual. 2019. Verkkoaineisto. KavoKerr. <<https://www.kavo.com/en-uk/dental-x-ray-machines-and-diagnostics/op-3d-extraoral#docs>>. Luettu 15.1.2019.
- 3 Isoviita. Reijo. V&V Senior Validation Specialist. KavoKerr Group Finland. Keskustelu. 14.2.2019.
- 4 KaVo ORTHOPANTOMOGRAPH™ OP 3D - Bitewing Patient Positioning. 2019. Verkkoaineisto. KavoKerr. <<https://www.youtube.com/watch?v=5QqvGIVWuv4>>. Luettu 12.3.2019.
- 5 Töysä, A. 2019. Bitewing kuvausohjelman kehitystyö ekstraoraaliseen hammasröntgenlaitteeseen. Diplomityö. Tampereen teknillinen yliopisto.
- 6 Panoramic Equipment and Imaging. 2019. Verkkoaineisto. Pocket Dentistry <<https://pocketdentistry.com/3-panoramic-equipment-and-imaging/>>. Luettu 20.2.2019.
- 7 Raspberry Pi 3 Model B+. 2019. Verkkoaineisto. Vadelma Pii. <<https://vadelmapii.com/tuote/raspberry-pi-3-model-b-tietokone-0>>. Luettu 3.1.2019.
- 8 What is a Raspberry Pi? Verkkoaineisto. The Pi. <<https://thepi.io/what-is-a-raspberry-pi/>>. Luettu 3.1.2019
- 9 The best operating system for your Raspberry Pi projects. 2019. Verkkoaineisto. Lifehacker.com. <<https://lifehacker.com/the-best-operating-systems-for-your-raspberry-pi-projec-1774669829>>. Luettu 4.1.2019.
- 10 Get started with Raspberry Pi. 2019. Verkkoaineisto. Raspberry.org. <<https://www.raspberrypi.org/learning/software-guide/quickstart/>>. Luettu 5.1.2019.
- 11 Noobs. 2019. Verkkoaineisto. Raspberry.org. <<https://www.raspberrypi.org/downloads/noobs/>>. Luettu 5.1.2019.
- 12 RasPi. 2019. Verkkoaineisto. Hutasu.net <<https://www.hutasu.net/raspi/>>. Luettu 7.1.2019.

- 13 GPIO. 2019. Verkkoaineisto. Raspberry.org.  
<<https://www.raspberrypi.org/documentation/usage/gpio/>>. Luettu 10.1.2019.
- 14 The GPIO Connector. 2019. Verkkoaineisto. Adafruit.com  
<<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/the-gpio-connector>>. Luettu 15.1.2019.
- 15 Configuring GPIO. 2019. Verkkoaineisto. Adafruit.com.  
<<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-gpio>>. Luettu 10.1.2019.
- 16 Using the I2C bus. 2019. Verkkoaineisto. Robot electronics. <<https://robot-electronics.co.uk/i2c-tutorial>>. Luettu 11.1.2019.
- 17 Configuring I2C. 2019. Verkkoaineisto. Adafruit.com.  
<<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>>. Luettu 25.1.2019.
- 18 Adafruit DC & Stepper Motor HAT for Raspberry Pi - Mini Kit. 2018. Verkkoaineisto. Adafruit.com. <<https://www.adafruit.com/product/2348>>. Luettu 17.12.2018.
- 19 PCA9685. 2018. Verkkoaineisto. NXP. <<https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>>. Luettu 17.12.2018.
- 20 Powering motors. 2018. Verkkoaineisto. Adafruit.com.  
<<https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi/powering-motors>> Luettu 17.12.2018
- 21 GitHub. 2019. Verkkoaineisto. Wikipedia. < <https://fi.wikipedia.org/wiki/GitHub>>. Luettu 11.3.2019.
- 22 Adafruit CircuitPython MotorKit. 2018. Verkkoaineisto. GitHub.  
<[https://github.com/adafruit/Adafruit\\_CircuitPython\\_MotorKit](https://github.com/adafruit/Adafruit_CircuitPython_MotorKit)>. Luettu 19.12.2018.
- 23 Adafruit DC and Stepper Motor HAT for Raspberry Pi. 2019. Verkkoaineisto.  
<<https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi/installing-software> > Luettu 24.2.2019.
- 24 Python ohjelmointikieli. 2019. Verkkoaineisto. Metropolia.fi.  
<<https://wiki.metropolia.fi/pages/viewpage.action?pageId=16288055>>. Luettu 24.2.2019.

- 25 The Python standard library. 2019. Verkkoaineisto. Python.org. <<https://docs.python.org/3/library/>>. Luettu 24.2.2019.
- 26 Python-ohjelmointi: Osa 9 - Standardikirjasto. 2019. Verkkoaineisto. Ohjelmointiputka. <[https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3\\_09#tiedostojarjestelma](https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3_09#tiedostojarjestelma)>. Luettu 9.3.2019.
- 27 Python-ohjelmointi: Osa 3 - Ehtorakenteet. 2019. Verkkoaineisto. Ohjelmointiputka. <[https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3\\_03](https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3_03)>. Luettu 9.3.2019.
- 28 Python-ohjelmointi: Osa 4 - Toistorakenteet. 2019. Verkkoaineisto. Ohjelmointiputka. <[https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3\\_04](https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=python3_04)>. Luettu 9.3.2019.
- 29 Wireless communication. 2019. Verkkoaineisto. Rakesh Kumar Jha. What is a stepper motor? <<http://jharakeshnetworks.blogspot.com/2011/03/what-is-stepper-motor.html>>. Luettu 3.3.2019.
- 30 Turja, J. 2017. Askelmoottorin testaaminen ja soveltuvuus xy-nosturiin. Insinööriyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 31 Stepper Motor. 2019. Verkkoaineisto. Engineers Garage. <<https://www.engineersgarage.com/articles/stepper-motors?page=1>>. Luettu 4.3.2019.
- 32 Can-Stack Stepper. 2019. Verkkoaineisto. <<https://www.haydonkerkittman.com/products/linear-actuators/can-stack-stepper/20mm-19000>>. Luettu 4.3.2019.
- 33 What is the difference between - bipolar & unipolar stepper motor? 2019. Verkkoaineisto. Quora. <<https://www.quora.com/What-is-the-difference-between-bipolar-unipolar-stepper-motor>>. Luettu 4.3.2019.
- 34 Stepper motors and drives, what is full step, half step and microstepping? 2019. Verkkoaineisto. Design Spark. <<https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping>>. Luettu 4.3.2019.
- 35 Caliper cage. 2019. Verkkoaineisto. Promat. <<https://www.nordwest-promat.com/en/tools/bme/article/messschieber-3/>>. Luettu 4.3.2019.



- 36 Relay Switch Circuit. 2019. Verkkoaineisto. Electronics Tutorials.  
<<https://www.electronics-tutorials.ws/blog/relay-switch-circuit.html>>. Luettu 4.3.2019.