



CEAD 3D-tulostusrobotin värilisäyslaitteen prototyyppi

Janrik Hermunen

OPINNÄYTETYÖ
Huhtikuu 2025

Konetekniikan tutkinto-ohjelma
Tuotekehitys

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikan tutkinto-ohjelma
Tuotekehitys

HERMUNEN, JANRIK:
CEAD 3D-tulostusrobotin väriläyslaitteen prototyyppi

Opinnäytetyö 62 sivua, joista liitteitä 24 sivua
Huhtikuu 2025

Tampereen ammattikorkeakoululla on tullut tarve valmistaa koulun nykyiseen 3D-tulostussoluun väriaineen lisäysominaisuus. Tarve on syntynyt, kun todettiin käyttäjän manuaalisen materiaalin lisäämisen olevan ajan ja resurssien käytön kannalta järjetöntä. Haluttiin, että tämä vaihe voidaan automatisoida tulevaisuutta varten.

Suunniteltu laite perustuu Ida-Maria Forsmanin vuonna 2024 toteuttamaan ehdotukseen väriaineen lisäyslaitteesta. Tämän ehdotuksen pohjalta toteutettiin mitoitettu ja ohjelmoitu prototyyppi toimimaan todisteena projektin jatkamisen järkevyydestä. Forsmanin ehdotuksen tueksi haettiin vaihtoehtoisia ratkaisuja ja mekaanisten komponenttien jumiintumista jouduttiin pohtimaan. Sähkö- ja automaatiototeutukselle oli tehty ehdotus, joka oli myös testattu. Värigranulaateille tehtiin mittaukset niiden painojen selvittämiseksi. Annostelun tarkkuuden kannalta mitaus oli tärkeä vaihe. Mittauksen pohjalta voitiin aloittaa mekaanisen annostelulevyn reikien sovittaminen.

Sähkötekniisiin ratkaisuihin perehdyttiin käyttäen Tampereen ammattikorkeakoulun automaatiotekniikan laboratorion prototyyppisähköalustoja. Prototyyppi yhdistettiin sähkökytkentään. Kytkenällä todennettiin ohjelmoinnin toimivuus mekaanisen kokoonpanon kanssa. Ohjelmisto, jonka väriaineen lisäyslaite tarvitsi, toteutettiin käyttäen Arduino Uno R3 -mikro-ohjainta. Ohjauksen kirjoittamiseen tarvittiin Arduino Cloud-verkkopalvelua, jossa koodin sai kirjoitettua ja testattua. Ohjauksen koodit on selitetty, ja ne sisältyivät lopputulokseen. Ohjelmat pyrittiin toteuttamaan mahdollisimman yksinkertaisesti, jotta muutosten tekeminen myöhemmässä vaiheessa onnistuu.

Tuloksena syntyi annostelulevylle testattu ja muovista valmistettu prototyyppi vaadituilla ominaisuuksilla. Lopputuloksena annettiin myös vaihtoehtoinen toteutustapa käyttäen annosteluruuvia. Virallisen mallin kustannuksia pohdittiin. Esille tuotiin valmistukseen tarvittava vahvempi materiaali, esimerkiksi alumiini. Työhön sisältyy valmistukseen vaadittavat piirustukset annostelulevyn ja annosteluruuvien prototyypeille.

Asiasanat: granulaatti, arduino, askelmoottori, l298n, ohjelmointi, prototyyppi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Mechanical Engineering
Product Development

HERMUNEN, JANRIK:
Color adding machines prototype for CEAD 3D printing robot

Bachelor's thesis 62 pages, appendices 24 pages
April 2025

The purpose of this thesis was to design and make a colour adding machine for Tampere University of Applied Sciences. Adding colour manually was too time-consuming and a waste of resources, which called for an automated solution.

The design was based on Ida-Maria Forsman's suggestion in the year 2024 about how to feed additional material into the school's 3D printing robot cell. Based on her proposal, a prototype was built to fit the designated purpose. Different angles were considered to determine which material adding method was to be used. A problem with material getting stuck in between the machine's moving parts was identified, and solutions were designed. As a result, an electrical diagram and an operating program were made for the designed machine. The machine was assessed, and it was confirmed that it functioned as intended. The granules that were used as the colouring agent had to be measured. After measuring the granules, the holes of the sorting wheel could be fitted.

An electrical solution was made using the school's automation laboratory's proto circuit boards. The mechanical model was connected to an electric circuit so that the program that was designed to operate the prototype could be assessed. The program was made on Arduino Uno R3 microcontroller using Arduino's Cloud coding service. The platform allowed testing and storing the written code. The program was included in the thesis. The program was made to be simple to edit and change to use on different applications.

As a result, the sorting wheels prototype was manufactured. The prototype was manufactured from plastic with adjusted sorting capability. Also, an alternative method for supplying the added material was designed using a screw feeder. The overall component costs were considered. The final model should be produced from aluminum for smoother operation. The mechanical drawings of the models were included in the project as well as the electrical circuit diagram that was used.

Key words: granule, arduino, stepper motor, l298n, coding, programing, prototype

SISÄLLYS

1	JOHDANTO.....	6
2	TUTKIMUSKYSYMYS	7
3	MEKAANINEN SUUNNITTELU	11
	3.1 Vaatimusten asetus.....	11
	3.2 Annostelulevy.....	11
	3.3 Annosteluruuvi	13
4	SÄHKÖSUUNNITTELU	15
	4.1 Piirinkomponenttien valinta	15
	4.2 Piirikaavio	16
	4.2.1 Askelmoottorin kytkentä.....	18
	4.2.2 Jännitteen pudotuspiiri.....	18
	4.2.3 Kytkimet	19
5	OHJAUS	20
	5.1 Annostelulevyn ohjaus	21
	5.2 Annosteluruuvien ohjaus	27
6	LOPPUTULOS	28
7	POHDINTA	31
8	JOHTOPÄÄTÖKSET	33
	8.1 Mekaniikka	33
	8.2 Sähkö.....	33
	8.3 Ohjaus	33
	8.4 Lopputulos	35
	LÄHTEET	36
	LIITTEET	39
	Liite 1 Nema 23 Askelmoottorin Datalehti	39
	Liite 2 Nema 17 askelmoottorin Datalehti.....	40
	Liite 3 A4988 Datalehdestä otteita	41
	Liite 4 L298N Datalehdestä otteita	44
	Liite 5 Oteita Arduino Unon Datalehdestä	48
	Liite 6 Ote Painekytkimen Datalehdestä	50
	Liite 7 Annostelulevyn ohjelma.....	51
	Liite 8 Ruuvien ohjelma	54
	Liite 9 Annostelulevyn piirustukset	57
	Liite 10 Annosteluruuvien piirustukset.....	60
	Liite 11 Arvioita värinlisäjäyksikön komponenttien kustannuksista ...	62

LYHENTEET JA TERMIT

TAMK	Tampereen ammattikorkeakoulu
PLC	Programmable logic controller
GND	Maa (Ground) 0 V
A0	Analogianasta 0 (Analog pin 0)
D1	Digitaalinnasta 1 (Digital pin 1)
INPUT / IN	Sisääntulo(nasta)
OUTPUT / OUT	Lähtö(nasta)
CW	Clockwise (Myötäpäivään)
CCW	Counterclockwise (Vastapäivään)
ABS	Acrylonitrile butadiene styrene (3D-tulostusmateriaali)
PLA	Polylactic acid (3D-tulostusmateriaali)

1 JOHDANTO

Tampereen ammattikorkeakoulun käytössä on ABB:n robottisolu, johon on integroitu CEAD:n 3D-tulostuspursotinmoduuli. Moduulin ohjaus tapahtuu sen omalla järjestelmällä, joka toimii yhdessä ABB:n robottiin ajettujen tulostusohjeiden kanssa. Moduulin materiaali on varastoitu siiloon, josta sitä syötetään paineilmalla pursottimen käyttöön.

Materiaalina käytetään tavallisesti luonnonväristä komposiittigranulaattia. Tampereen ammattikorkeakoululle on tullut tarve lisätä solun 3D-tulosteisiin väriä. Värin lisäys kyettiin hoitamaan manuaalisesti, mutta se ei ollut käytännöllinen ratkaisu. Automatisointi oli järkevämpi tapa. Ida-Maria Forsman esitteli oman suunnitteluehdotuksensa vuonna 2024. Tavoitteena oli saada annostelija, jota voidaan ajaa ABB:n solun avulla. Mekaanisen mallin pohjalta oli tarkoitus toteuttaa prototyyppi.

Forsman oli kerännyt dataa tulostimen toiminnasta. Nämä tiedot tarkastettiin ja sen pohjalta aloitettiin uuden laitteen valmistus. Granulaatit, joita tulostin käyttää väriaineena, mitattiin. Mittausten pohjalta tehtiin laskelmat tilaajan antamien käyttöannosmäärien mukaan. Mekaanisesti laitteen tuli kyetä annostelemaan väriainetta mahdollisimman tarkasti. Annostelun tuli olla luotettavaa. Toteutuksessa suositettiin yksinkertaisia ratkaisuja laitteen kustannuksien minimoimiseksi. Laitteen tuli olla myös mahdollisimman helppo huoltaa. Osien vaihtamisen kannalta pyrittiin mahdollisimman yksinkertaiseen kokoonpanoon.

Tampereen ammattikorkeakoulu tarvitsi mekaaniset piirustukset, sähkökaavion sekä kytkentälogiikan, että toimivan koodin laitteen ohjaukseen. Prototyyppi toteutettiin aluksi PLA-muovista 3D-tulostamalla ja virallisempi malli tulostettiin ABS-muovista. Prototyypin ohjaukseen suunniteltiin koodi, jota voidaan jatkossa soveltaa muihinkin tarkoituksiin.

2 TUTKIMUSKYSYMYKSIÄ

Laitteen toiminnan kannalta oli ehdottoman tärkeää selvittää CEAD:n materiaalin annostelijan siirtämät väriainemateriaalien määrät. Väriainetta aiottiin annostella 1 %, 5 % ja 10 %, jotka suhteutettiin tulostuksessa käytettyyn materiaaliin määrään. Ida-Maria Forsman oli kerännyt dataa 3D-tulostimen toiminnasta. Se on esitetty taulukossa 1. Tuloksista selviää laitteen oman materiaalin annostelun olevan epätarkkaa.

TAULUKKO 1. 3D-tulostimen materiaalin määrä eri annoskerroissa.

Tulokset perustuvat Ida-Maria Forsmanin mittauksiin.

SYÖTTÖKERTA	MATERIAALIN MÄÄRÄ (g)
1	44
2	17,8
3	27,1
4	30,6
5	26,4
6	28,7
7	31,6

Näistä tuloksista Forsman totesi, että syöttökerroissa yksi ja kaksi oli ongelmia, joten niitä ei otettu huomioon keskiarvoa laskettaessa. Keskiarvoksi Forsman sai 28,9 grammaa. Mittauksen varmistuksen yhteydessä selvisi, että yleinen käyttöannos on noin 30 grammaa. Poikkeavuudessa oli todennäköisesti takana puristusventtiilin toiminnallinen epätarkkuus. Epätarkkuus saattoi johtua puristimen vajaasta kiristyksestä tai liian lyhyestä avautumisajasta. Myös tekninen vika signaaleissa, jotka ohjaavat avautumista, saattoi aiheuttaa vajaatoimintaa. Prototyypin suunnittelun yksinkertaistamiseksi päätettiin käyttää robotin annostelussa 30 gramman arvoa. Myöhemmässä vaiheessa voidaan harkita laajemman mittauksen toteuttamista. Sen pohjalta voidaan tehdä tarkempia laskelmia. Niin voidaan vähentää värisävyjen eroavuutta tulostettavien kerrosten välillä.

Kun annostelumäärät olivat selvillä, lähdettiin selvittämään värigranulaattien yksilöpainoja. Granulaattien painot mitattiin käyttäen Tampereen ammattikorkeakoulun kemian laboratorion tarkkaa vaakaa, joka on esitetty kuvassa 1. Vaa'an tarkkuus oli herkkä, jolloin ilmavirtaus ja staattinen sähkö muuttivat vaa'an tuloksia.



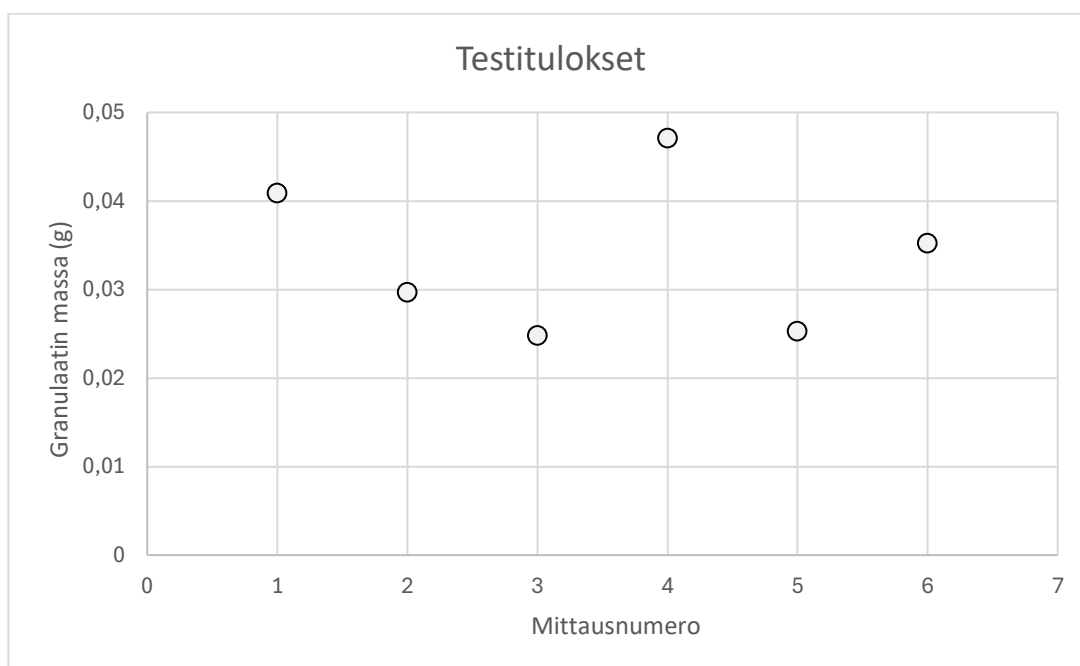
Kuva 1. Vasemmalla on granulaattien punnitukseen käytetty vaakaa. Oikealla on tarkempi kuva vaa'an näytöstä (Kuvat: Janrik Hermunen).

Mittauksessa käytettiin kuutta satunnaisesti valittua värigranulaattinäytettä. Nämä on tilastoitu taulukkoon 2. Tuloksista nähtiin, että väriaineen painot eivät olleet yhteneväisiä.

TAULUKKO 2. Mittaustulokset. (Janrik Hermunen)

Musta väriainegranulaatti	Mitatut painot (g)
1	0,0409
2	0,0297
3	0,0248
4	0,0471
5	0,0253
6	0,0352

Eroavuutta satunnaisten mittaustuloksien välillä oli melkein kaksinkertaisesti. Taulukon 2 tulokset näkyvät graafisesti kuviosta 1. Granulaattien painojen hajonnan takia piti valita, miten lähestytään prototyypin annostelun tarkkuutta. Jos myöhemmin todetaan, että väriaineen annostelun tarvitsee olla tarkempaa, on järkevää toteuttaa laajempi ja kattavampi mittaus. Esimerkiksi sadalla satunnaisella granulaatilla toteutettu mittaus voi valaista minkälainen tilastollinen jakauma eri granulaateista syntyy. Koska prototyyppi toteutettiin vastaamaan vain yhden tyyppistä ja -muotoista granulaattia, tulevaisuudessa on järkevää tehdä mittaukset myös muille vaihtoehdoille.



Kuvio 1. Granulaattien mittatustulokset.

Annosmääristä saatiin tuloksia, jotka on kirjattu taulukkoon 3. Sen mukaan tehtiin alustava valinta tarkkuudesta. Prototyypille sopiva tarkkuus oli noin 10 granulaattia jokaisessa annoksessa. Annostelu toteutettiin keskiarvon mukaan. Tarkemmalla ja kattavammalla mittauksella voidaan päästä luotettavampiin arvoihin.

Taulukko 3. Lasketut granulaattien annostelumäärät. (Janrik Hermunen)

Keskiarvo painoannokset				
gramma (g)	3,0	1,5	0,3	0,1
Annostelumäärä	88,67	44,33	8,87	2,96
Pyöristetty ylöspäin	89	45	9	3
Pyöristetty alaspäin	88	44	8	2
Maksimi painoannokset				
gramma (g)	3,0	1,5	0,3	0,1
Annostelumäärä	63,69	31,85	6,37	2,12
Pyöristetty ylöspäin	64	32	7	3
Pyöristetty alaspäin	63	31	6	2
Minimi painoannokset				
gramma (g)	3,0	1,5	0,3	0,1
Annostelumäärä	120,97	60,48	12,10	4,03
Pyöristetty ylöspäin	121	61	13	5
Pyöristetty alaspäin	120	60	12	4

3 MEKAANINEN SUUNNITTELU

3.1 Vaatimusten asetus

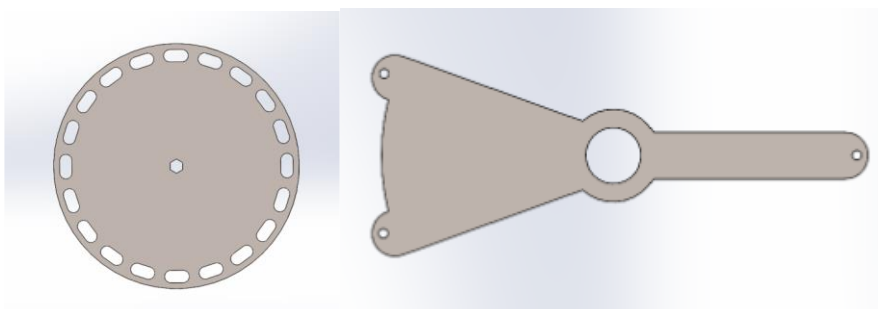
Annostelu toteutettiin aluksi mittausten mukaisissa määrissä. Alkuperäiset määrät olivat 1 %, 5 % ja 10 % 3D-tulostimen 30 grammasta pursotettavaa materiaalia.

Uusien komponenttien oli oltava mahdollisimman yksinkertaisia sekä 3D-tulostaa, että tarvittaessa työstää metallista. Suunnitellun laitteen jalanjälki minimoitiin, jotta se olisi yhteensopiva asentaa samassa tilassa olemassa olevien laitteiden kanssa.

Paineilmalla toimiva materiaalinkuljetin rajoitti toimintaa. Oli pystyttävä minimoimaan säiliöön päätyvä ohivirtaus. Kun laitetta ei käytetty, tuli säiliön olla suljettuna.

3.2 Annostelulevy

Annostelulevy perustuu Forsmanin ehdotukseen, joka on esitetty kuvassa 2. Kuvassa on 20-reikäinen pyöreä levy, jonka tarkoitus on jakaa tasaisia määriä granulaaiteja sen koloihin. Levyn paikallaan pitämiseksi oli suunniteltu Y-mallinen kappale. Kuvassa 2 oikealla Y-mallinen suoja vähentää ilman takaisinvirtausta tukkiessaan annostelureiän, johon levy siirtää materiaalia.

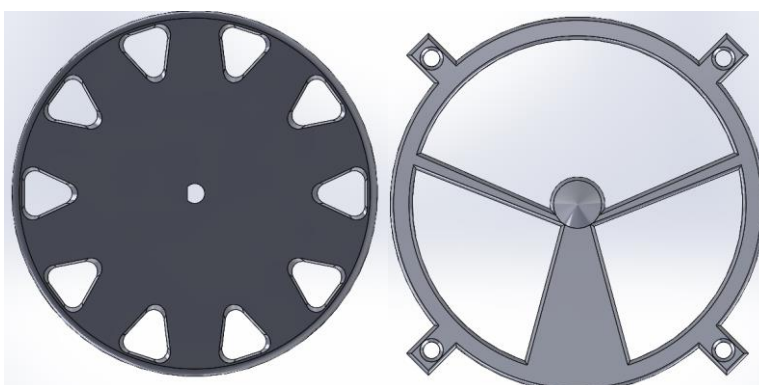


Kuva 2. Vasemmalla on annostelulevy.

Oikealla on annostelulevyn suoja (Kuvat: Ida-Maria Forsman).

Forsmanin mekaanista suunnittelua muokattiin sopiviin mittoihin. Annostelulevyn kolot mitoitettiin saatujen tulosten mukaan. Kuvassa 3 vasemmalla on uusi versio annostelulevystä, joka on sovitettu siten, että koloihin mahtuu noin 10 granulaattia.

Levyn jumittuminen ja materiaalin joutuminen seinämien väliin oli todettu ongelmaksi Forsmanin opinnäytetyössä. Tämän takia Y-mallinen suoja muutettiin malliin, joka estää levyn ja pohjan seinämien väliin joutuvat granulaatit. Uusi malli on kuvassa 3 oikealla.



Kuva 3. Vasemmalla on annostelulevyn uudelleen suunniteltu versio. Oikealla on annostelulevyn suojan uusi versio. (Kuvat: Janrik Hermunen).

Säiliön pohjasta kuvassa 4 tehtiin yhteensopiva isommalle Nema 23 -askelmoottorille ja Nema 17 -askelmoottorille. Niiden tekniset tiedot löytyvät liitteistä 1 ja 2.



Kuva 4. Väriläslaitteen säiliön pohja (Kuva: Janrik Hermunen).

3.3 Annosteluruuvi

Toinen mahdollinen toteutustapa oli käyttää niin sanottua annosteluruuvia. Menetelmä perustui materiaalin siirtämiseen kierteen avulla, joka on mitoitettu halutun aineenkuljetusmäärän mukaisesti. Mitoittaminen vaati testausta, jota ei ehditty tehdä.

Laitteen toiminnasta kerrotaan David Milliksen kirjassa (Pneumatic conveying design guide, kolmas painos, 2015). Kuvassa 5 on esitetty eräänlainen kaupallinen annosteluruuvia käyttävä ratkaisu.

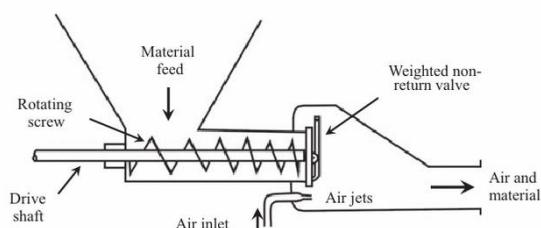


FIG. 5.16

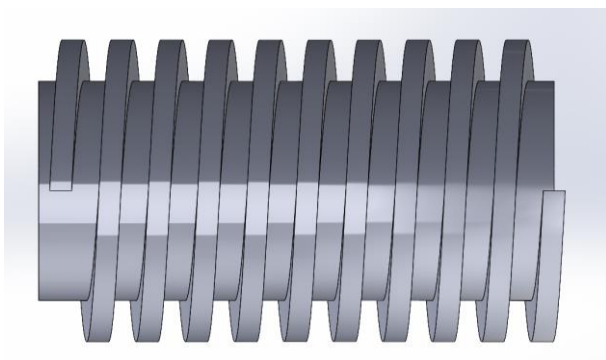
Commercial type of screw feeder

Kuva 5. Kaupallinen ruuviannostelija (David Millis, 2015)

Materiaalin syöttö ajetaan ruuville säiliöstä, josta ruuvi siirtää sen paineilma-
jettimeen. Ruuvien syöttö on tasaista ja hallittua. Ruuvi toimii paineistetuissa jär-
jestelmissä ja tyhjiöjärjestelmissä, Millis kertoo. Ruuvien toiminta vähentää itses-
sänsä paineen ja ilmavirran kulkua takaisin. Kun ruuvi on täynnä tavaraa, ilmavir-
ran pääsy takaisinpäin on estetty.

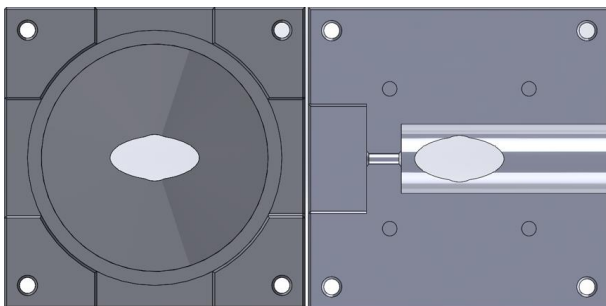
Milliksen esimerkissä käytetään tämän lisäksi painovoimalla toimivaa yksisuun-
taista venttiiliä. Ruuvi työntää materiaalin venttiilin läpi, ja kun tavaraa ei liiku,
venttiili sulkeutuu.

Kuvassa 6 on ruuviannostelijan prototyypin 3D-malli. Mallissa on 10 kierrosta, jotka vastaavat edellä esitettyä kymmenenreikäistä annostelulevyä. Ruuvien etuna on muun muassa sen helppo ajettavuus. Koska ruuvi siirtää tavaraa kokonaisilla kierroksilla, on moottorin ohjelmointi entistä helpompaa. Moottoria tarvitsee ajaa vain 360 asteen tarkkuudella.



Kuva 6. 10-kierteinen ruuviannostelija (Kuva: Janrik Hermunen).

Kuvassa 7 on värilisäyslaitteen säiliön pohjaan tehdyt muutokset ruuvia varten. Pohjaan jouduttiin tekemään ura, johon ruuvi voitiin sijoittaa. Säiliön pohjasta suunniteltiin kovera, jotta granulaatit valuisivat reiästä läpi.



Kuva 7. Ruuviannostelijan säiliön pohja (Kuva: Janrik Hermunen).

4 SÄHKÖSUUNNITTELU

4.1 Piirinkomponenttien valinta

Värilisäyslaitteen annostelijalle oli tarkoitus saada aikaan aksiaalista liikettä joko annostelulevylle tai annosteluruuvulle. Liikkeen aikaansaamiseksi tarvitaan moottori. Jotta laite toimisi mahdollisimman tarkoilla annostelumäärillä, tuli valitun moottorin olla askelmoottori.

Askelmoottorin etuja tavalliseen sähkömoottoriin verrattuna on mahdollisuus ohjata sitä pulsseilla askel kerrallaan. Liitteessä 1 on datalehti, joka kertoo valitun kaksinapaisen Nema 17 -askelmoottorin ominaisuudet. Pohdinnassa oli myös kaksinapainen Nema 23 -askelmoottori, jonka datalehti on liitteessä 2. Isomman moottorin väännön ansiosta jumiintumisen riski vähenee.

Nema 23 ohjaamiseen vaaditaan suuremmalle teholle mitoitettut komponentit, joita ei prototyyppiä tehdessä käytetty. Liitteessä 1 on esitetty moottorin arvot. Nema 17 toimii 2 ampeerilla ja moottorille suositeltiin 12 voltia. Prototyypin toteutuksen kannalta 12 voltin kytkentä oli helppo toteuttaa ja 12 voltille mitoitettuja komponentteja lainattiin Tampereen ammattikorkeakoulun automaatiotekniikan laboratoriosta. Nema 17 mahdollisti prototyypin valmistamisen projektille annetussa aikaikkunassa.

Askelmoottorin ohjaus vaati ajurin. Valinnassa käytettiin vaihtoehtoja, jotka olivat käytössä Tampereen automaatiotekniikan laboratoriossa. Vaihtoehtoina olivat A4988 (liite 3) ja L298N (liite 4). Ajurina päädyttiin käyttämään L298N:ää, koska sen toiminnallisuus ja toimintakapasiteetti mahdollistivat eri kokoisten moottorien käytön ilman erillistä säätöä. A4988 ajurissa virta piti rajoittaa jokaiselle moottorille erikseen sopivaksi, mikä todettiin turhaksi prototyypin toimintaa testattaessa. L298N avulla voitiin ajaa Nema 17 -askelmoottoria sellaisenaan ilman mitään muutoksia.

Koska ajureille on annettava komentoja, tarvittiin ohjainta. Valinnassa päädyttiin pohtimaan PLC:tä tai Arduinoa. PLC vaatii hieman enemmän vaivaa. PLC on

tarkoitettu teollisempiin ratkaisuihin. Lopullista laitetta tehdessä kannattaa harkita PLC:tä, koska se tukee 24 voltin jännitteitä ja voi ajaa suurempia moottoreita. PLC:llä on mahdollista toteuttaa suuria laitteistojen yhdistyksiä. Näistä eduista riippumatta, koska prototyyppiä voitiin käyttää pienemmällä moottorilla ja jännitteellä, voitiin toteutus tehdä Arduinolle. Käytetyn Arduinon datalehdessä löytyy otteita liitteestä 5. Arduino on pienempi komponenttina ja sen kytkentä voidaan toteuttaa helposti käyttäen laboratorion testikytkentäalustoja ja säädettäviä laboratorion jännitelähteitä.

Arduinon rajoite on sen käyttöjännite, koska Arduino käyttää 12 voltia ja yleinen jännite, jota robotti tukee, on 24 V. Tästä johtuen kytkentää ei voitu toteuttaa suoraan. Arduinon toinen haaste on sen sisääntulonastojen jänniteraja, joka on vain 5 V. Normaalisti 5 V on sopiva arvo, mutta robotti käyttää 10 voltia analogialähdöissä. Tämän takia jouduttiin piiriin tekemään jännitteen pudotus niin, että jännite puolittui ennen Arduinon analogianastaa.

Ohjauksessa aiottiin käyttää painekytkintä, joka mahdollistaa toiminnan robotin täyttösyklin kanssa. Painekytkintä ei ole lisätty piiriin. Kytkimen tarkoitus on huomata paineen laskeminen pneumaattisessa materiaalien kuljetusputkessa. Painekytkimen datalehti on liitteessä 6. Koska halutaan minimoida laitteen sisälle kulkeutuva paineistettu ilma, on tärkeää, että laite operoi vain silloin, kun paine on laskenut halutun tason alle.

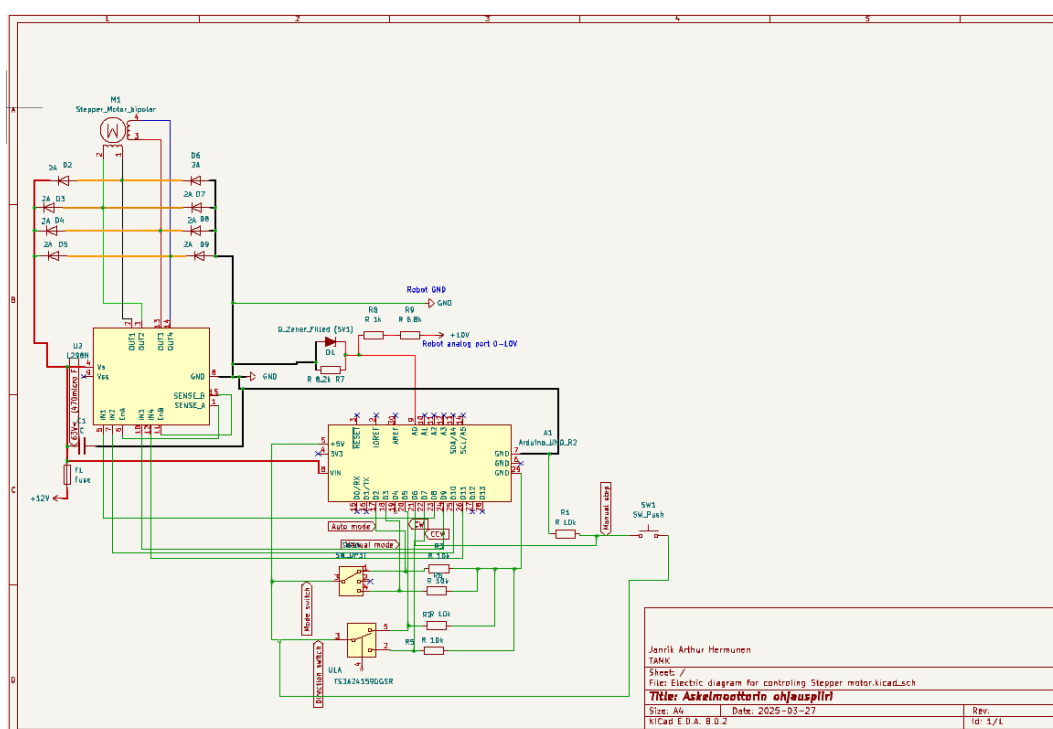
4.2 Piirikaavio

Mekaanisten komponenttien ohjaukseen vaadittava kytkentä on nähtävissä kuvassa 8. Moottori on kaksinapainen askelmoottori, joka on kytketty L298N:n lähtönastoihin. Kytkennässä moottorin mahdollinen takaisin palaava jännite voi aiheuttaa hetkellisiä yli- tai alijännitteitä. Jännitteiden piikkiarvot voidaan tasata puolijohdekytkennällä. Jotta moottorin ja Arduinon jännitteet pysyisivät vakaina, käytetään 470 μ F kondensaattoria, kuten liitteestä 4 Figure 10 nähdään.

Arduinon A0-nastan arvoa on tarkoitus ohjata robotin analogianastalla. Siitä lähtevä signaali on 0 ja 10 voltin välillä, joten se pitää pudottaa sopivaksi Arduinolle.

Arduino tukee 3,3 ja 5 voltin jännitteitä. Pudotus onnistuu kytkemällä 6,8 k Ω vastus sarjaan 1 k Ω vastuksen kanssa ja lopuksi nämä liitetään 8,2 k Ω vastukseen, joka on rinnan Arduinon analogianastan sisäresistanssin kanssa. Vastusten välistä saatiin mitattua jännitearvoksi noin 4,35 voltia.

Piiri suojataan jännitepiikeiltä liittämällä 5,1 V zenerdiodi Arduinon ja maan väliin. Zenerdiodi rajoittaa jännitteen, jolloin Arduinon analogianasta säilyy ehjänä vika-tilanteissa.



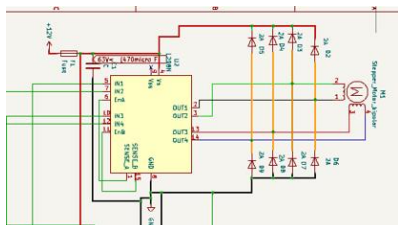
Kuva 8. Ohjauksen sähkökytkennän piirikaavio (Kuva: Janrik Hermunen).

Käsin valittava ohjelma tehtiin kytkimellä, jossa oli kolme asentoa. Asennot olivat neutraali, automaattiajo robotilla ja käsiajo eli manuaaliasetus, jonka avulla vika-tilanteissa painamalla kytkintä voidaan moottorin akselia kääntää haluttuun asemaan.

Moottorin pyörimiseen valittavaa suuntaa ohjattiin digitaalinaldoilla viisi ja seitsemän. Aktiivinen digitaalinalasta valittiin kaksiasentoisella kytkimellä.

4.2.1 Askelmoottorin kytkentä

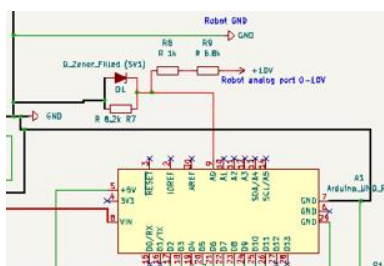
Askelmoottorinkytkeä on esitetty tarkemmin kuvassa 9. Se on suunniteltu käyttäen L298N:n ajuria. Tämä tehtiin seuraten esimerkikykentää liitteessä 4 Figure 10. Kytkenässä on otettu huomioon sähkömoottorin toiminta generaattorina. Kun sähkömoottori vaihtaa suuntaa, käämeihin indusoituu jännitettä, joka palaa ajurin nastoille.



Kuva 9. Askelmoottorin kytkentäkaavio (Kuva: Janrik Hermunen).

4.2.2 Jännitteen pudotuspiiri

Robotilta tuleva säätösignaali käyttää 0–10 voltin suuruista jännitettä. Robotin säätöjännite pudotettiin alle 5 volttiin kolmen sarjaan kytketyn vastuksen avulla, joiden arvot ovat 6,8 k Ω , 1 k Ω ja 8,2 k Ω . Kuvassa 12 esitellään jännitteen pudotuspiiriä.



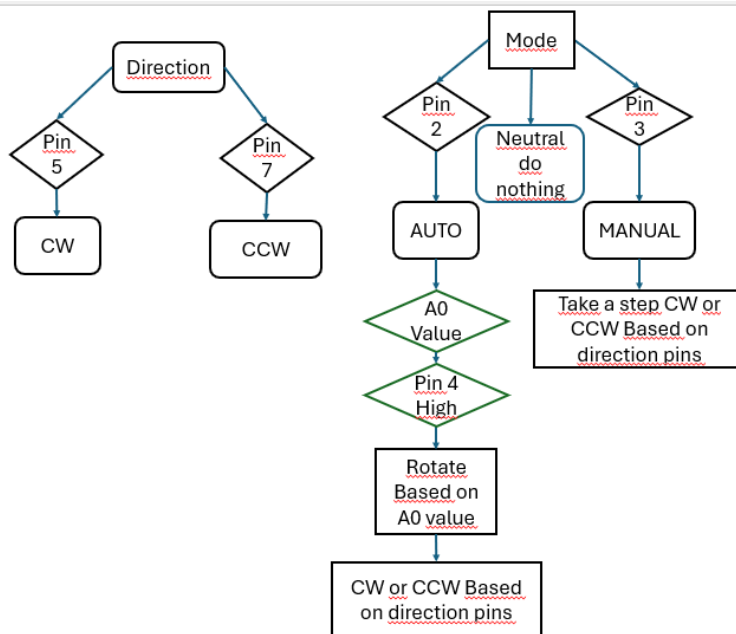
Kuva 10. Arduinon jännitteen pudotuspiiri (Kuva: Janrik Hermunen).

Arduinon sisäresistanssin noin 108k Ω rinnankytkennästä 8,2 k Ω vastuksen kanssa muodostuu 7,6 k Ω . Kytkeä muodostaa 7,8 k Ω ja 7,6 k Ω sarjan kytkennän. Tarvittavien vastusten arvot selvitettiin käyttäen 10 k Ω säätövastusta sarjassa 4,7 k Ω vastuksen kanssa. Mittauksessa havaittiin, että tarvittavat vastusarvot ovat 7,1 k Ω ja 7,6 k Ω rinnan Arduinon sisäresistanssin kanssa. Jotta jännite puolittuisi on molempien vastusten kokonaisresistanssin oltava yhtä suuri. Tästä

5 OHJAUS

Ohjaus toteutettiin Arduino Uno 3 -mikropiirillä. Ohjauksen lähteenä käytettiin Arduinon verkkosivuja ja valmiita esimerkkejä useilta eri verkkosivuilta. Linkit kaikkiin käytettyihin lähteisiin löytyvät lähdeluettelosta. Ohjauskoodi toteutettiin ilman valmiita kirjastoja.

Ohjauskoodin tarkoitus on ohjata moottorin liikettä laskettujen annosmäärien arvojen mukaan. Chat GPT -tekoälyä käytettiin koodin tarkastuksessa. Sen avulla tehtiin korjauksia ja se antoi kommentteja virheistä. Perusohjaamisen kannalta tärkein lähde oli How To Mecathronics-kanavan video Stepper Motors and Arduino - The Ultimate Guide, jossa esitettiin askelmoottorin ohjauksen perusteita Arduinolla. Kuva 12 on ohjauskoodin kulun vuokaavio.



Kuva 12. Ohjauksen vuokaavio (Kuva: Janrik Hermunen).

Lopputuloksena syntyi kaksi koodia, yksi annostelulevylle ja toinen annosteluruuville. Ensin tehtiin levyn ohjauskoodi, joka on liitteestä 7. Kun ensimmäinen ohjauskoodi oli tehty, toteutettiin annosteluruuvien ohjauskoodi liitteessä 8 esitetyllä tavalla.

5.1 Annostelulevyn ohjaus

Aluksi annettiin moottorin ohjauksen nastoille tunnus motorPin x, jossa x on moottorin nastan numero. Määritetyille nastoille merkittiin fyysinen vastakappale Arduinon digitaalinasoista. Numerot ohjauskoodissa ovat kahdeksan, kymmenen, yhdeksän ja yksitoista. Vastaavat ajurin nastat lisättiin kommentoimalla In x, jossa x on ajurin sisääntulojen numero yhdestä neljään. Nimeäminen perustuu yleiseen tapaan merkitä nastoja.

```
// Define motor control pins (L298N Driver)
const int motorPin1=8;// IN1
const int motorPin2=10;// IN2
const int motorPin3=9;// IN3
const int motorPin4=11;// IN4
```

Kun on asetettu askelmoottorin ohjaukseen vaaditut nastat, määritetään seuraavaksi kytkimien ja signaalien välittämiseen tarkoitettut nastat. A0-nasta määritettiin säätövastukselle potentiometerPin. Kun se kytketään robotin analogianastaan, robotti muuttaa jännitettä säätövastuksen sijaan. Nastojen nimeämisessä on käytetty niiden toiminnallisuuden mukaisia termejä. ControlPin ohjaa moottorin toimintaa kytkimellä. Lopullisessa värilisyislaitteessa on tarkoitus asentaa painekytkin controlPin nastaan.

```
// Define input pins
const int potentiometerPin=A0;// Potentiometer
const int controlPin= 4;// Trigger motor movement
const int autoModePin=2;// Enable auto mode
const int manualModePin=3;// Enable manual mode
const int buttonPin=6;// Manual step button
const int clockwisePin=5;// Clockwise direction
const int counterclockwisePin=7;// Counterclockwise direction
```

Kaikille ominaisuuksille, joita sähkökytkennässä käytettiin, määritettiin omat muuttujat ja sitä vastaava fyysinen nasta. Määritetyt nastat ovat analogianasta 0 (A0) ja digitaalinasat Dx, joissa x on nastan numero. Analogianastassa tulee koodia kirjoittaessa nimetä se a-kirjaimella, jotta ohjelma tunnistaa sen analogianastaksi. Digitaalinasoille sen sijaan ei tule antaa etuliitettä D, joten ne on vain numeroitu tässä tapauksessa neljästä seitsemään. Kaikille nastoille annoin selityksen sähkökytkennän selkeyttämiseksi.

Kun oli määritetty kaikki nastat, ohjelmalle luotiin muuttujalista. Ohjauskoodi hyödyntää muuttujalista suorittaessaan komentoja. Ensin analogianastan (A0) muuttujalle potentiometerValue varattiin arvo. Sen alkuarvoksi asetettiin nolla, jotta moottori ei lähde liikkeelle ennenaikaisesti. Analogianastan arvoa käytetään myöhemmässä vaiheessa koodia.

Toisena muuttujana on stepsToMove, joka määrittää kuinka monta askelta moottori ottaa. Kolmantena muuttujana on isAutoMode, joka säättää moottorin ajotilaa. Painikkeen aseman seuraamiseen luotiin muuttuja lastButtonState. Moottorin tyhjäkäytön vähentämiseksi määritettiin muuttuja viisi motorEnabled, joka ohjaa, onko moottori päällä vai pois.

```
// Variables
int potentiometerValue=0;
int stepsToMove=0;
bool isAutoMode=false;
bool lastButtonState=HIGH;
bool motorEnabled=false;
int stepDelay=5;// Fine-tuned delay for smoother movement
```

Kun ohjelma kutsuu disableMotor(), se asettaa moottorin ohjauksen pois päältä. Yksitellen jokainen nasta asettuu loogiseen nollatilaan (LOW) ja lopuksi moottori kytkeytyy pois, kun motorEnabled määritetään epätodeksi.

```
void disableMotor()
{
    digitalWrite(motorPin1,LOW);
    digitalWrite(motorPin2,LOW);
    digitalWrite(motorPin3,LOW);
    digitalWrite(motorPin4,LOW);
    motorEnabled=false;
}
```

Seuraavaksi määritettiin kaikkien käytössä olevien nastojen sisääntulo- ja lähtö-
asemat. Kytkimille varatut nastat merkittiin vakioarvolla INPUT_PULLUP, joka
tarkoittaa Arduinon sisäisen ylösvetovastuksen päälle kytkemistä.

```
void setup()
{
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  pinMode(controlPin, INPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(clockwisePin, INPUT_PULLUP);
  pinMode(counterclockwisePin, INPUT_PULLUP);
  pinMode(autoModePin, INPUT_PULLUP);
  pinMode(manualModePin, INPUT_PULLUP);
  Serial.begin(9600);
}
```

Laitteen toiminnallisuuden valinta automaatin ja käsikäytön välillä tapahtuu alla
olevien komentojen mukaisesti. Ensin ohjelma tarkastaa automaattiajolle varatun
nastan muuttujan arvon autoModePin. Jos nasta on aktiivinen eli looginen yksi,
se suorittaa automaattiajon komennot. Jos käsikäytölle merkitty nasta on aktiivi-
nen manualModePin, ohjelma valmistautuu toteuttamaan käsiajolle varatun ko-
mentosarjan. Kaikissa muissa tapauksissa koodi on ohjelmoitu asettamaan
moottori pois päältä disableMotor() komennolla.

```
void loop()
{
  if (digitalRead(autoModePin)==HIGH)
  {
    isAutoMode = true;
    autoMode();
  }
  else if (digitalRead(manualModePin)==HIGH)
  {
    isAutoMode=false;
    manualMode();
  }
  else
  {
    disableMotor();// Ensure motor is off when no mode is active
  }
}
```

Automaattiajon koodi on mitoitettu toimimaan 0–4 voltin arvoilla. Ohjauskoodi
käskee moottoria kääntymään, jotta haluttu määrä reikiä kohtaa annostelulle
tehdyn reiän. Esitelty mekaanisessa suunnittelussa. Koska reikiä on kymmenen

ja askelia on 200, toteutuu reikien paikoittaminen jakamalla 200 askelta kymmenellä reiällä. Näin jokaisen reiän etäisyys toisistaan on 20 askelta. Automaatti-asetus tarkastaa suuntanastojen signaalin, josta se lukee moottorin akselin kääntösuunnan. Lopuksi moottori kytketään pois päältä.

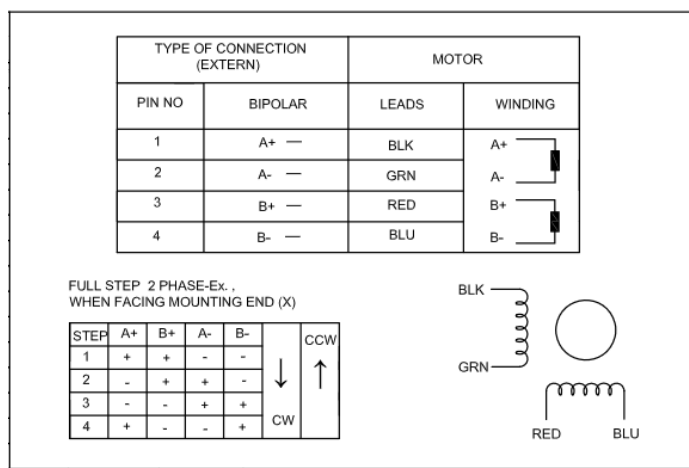
```
void autoMode()
{
    potentiometerValue=analogRead(potentiometerPin);
    stepsToMove=map(potentiometerValue,0,(1023*4)/5,0,200);
    // Adjusted for 0-4V range
    stepsToMove=(stepsToMove/20)*20;// Round to nearest 20

    bool moveForward=digitalRead(clockwisePin)==HIGH;
    if (digitalRead(counterclockwisePin)==HIGH)
    {
        moveForward=false;// Override if counterclockwise is pressed
        if (digitalRead(controlPin)==HIGH)
        {
            moveStepper(stepsToMove, moveForward);
            delay(500);
        }
        else
        {
            disableMotor();// Turn off motor when not in use
        }
    }
}
```

Manuaaliohjaus eli käsiajo toteutuu silloin, kun isAutoMode ei ole totta ja kytkintä painetaan. Liikkumiseen valittu suunta vaikuttaa käsiajoon samalla tavalla kuin automaattiasetukseen.

```
void manualMode()
{
    bool currentButtonState=digitalRead(buttonPin);
    bool moveForward = digitalRead(clockwisePin)==HIGH;
    if (digitalRead(counterclockwisePin)==HIGH)
    {
        moveForward=false;
        if (lastButtonState==HIGH && currentButtonState==LOW)
        {
            // Detect button press
            Serial.println("Manual button pressed, moving four steps");
            moveStepper(4, moveForward);
            // Move forward or backward based on switch
            delay(200);// Debounce delay
        }
    }
    lastButtonState=currentButtonState;
}
```

Askeleet on määritetty 4x4-matriisilla. Matriisien toiminta perustuu pulssien ajoon oikeissa ajoituksissa. Pulssien ohjaamiseen löytyy taulukko askelmoottorien datalehdistä liitteistä 1 ja 2 stepperonline verkkosivuilta. Valituille moottoreille toimiva täysillä askeleilla ohjattu matriisi löytyy alla olevasta koodista. Suunnan vaihto toimii siten, että 4x4 matriisi kirjoitetaan alkamaan neljännestä askeleesta alaspäin. Matriisii siis käännetään ympäri, jotta askeleet otetaan takaperin, lähde löytyy liitteistä 1 ja 2 sekä stepperonline verkkosivuilta. Kuvassa 13 on esitetty olennaisin tieto matriisin toteutukseen.



Kuva 13. Pulssituksien taulukko askelmoottorille (Kuva: stepperonline)

Matriisissa on ensin A+ ja A- vierekkäin käämi 1 ja B+ ja B- vastaavalla tavalla käämi 2. Matriisista, joka on esitetty kaavassa 1 näkyy ohjauskoodin matriisin käämin napoja vastaavat paikat samalla tavalla nimettynä kuin kuvassa 13.

$$[A + \quad A- \quad B + \quad B-] \quad (1)$$

```
void moveStepper(int steps, bool moveForward)
{
    int stepSequence[4][4]=
    {
        {HIGH, LOW, HIGH, LOW}, //Step 1
        {LOW, HIGH, HIGH, LOW}, //Step 2
        {LOW, HIGH, LOW, HIGH}, //Step 3
        {HIGH, LOW, LOW, HIGH} //Step 4
    };

    int stepSequenceBackward[4][4]=
    {
        {HIGH, LOW, LOW, HIGH}, // Reverse Step 4
        {LOW, HIGH, LOW, HIGH}, // Reverse Step 3
        {LOW, HIGH, HIGH, LOW}, // Reverse Step 2
        {HIGH, LOW, HIGH, LOW} // Reverse Step 1
    };
}
```

Moottorin akselin pyöriminen toteutuu for-silmukalla. Silmukka ohjaa askelmoottorin toimintaa ajamalla matriisien mukaisia pulsseja moottorin käämin napoihin. Tätä silmukkaa koodi toistaa, kunnes saavutaan haluttuun asemaan. Idea käyttää for-silmukkaa, joka on peräisin HowToMechatronics YouTube kanavan vuonna 2023 julkaistulta videolta Stepper Motors and Arduino - The Ultimate Guide. Koodi on tehty Githubin verkkosivujen ja Arduinon omien verkkosivujen esimerkkien pohjalta. Chat GPT-tekoälyä käytettiin selvittämään toimivuuden ongelmia. Ongelma oli, kun moottori otti askelia neljä kerrallaan yhden sijaan. Lopulta ongelma ratkaistiin jakamalla steps muuttuja neljällä. For-silmukan rakenteen selkeyttämiseen saatiin apua koneautomaatiotekniikan opiskelijoilta.

```
for (int i=0;i<steps/4;i++)
{
  for (int j=0;j<4;j++)
  {
    if (moveForward)
    {
      digitalWrite(motorPin1, stepSequence[j][0]);
      digitalWrite(motorPin2, stepSequence[j][1]);
      digitalWrite(motorPin3, stepSequence[j][2]);
      digitalWrite(motorPin4, stepSequence[j][3]);
    }
    else
    {
      digitalWrite(motorPin1, stepSequenceBackward[j][0]);
      digitalWrite(motorPin2, stepSequenceBackward[j][1]);
      digitalWrite(motorPin3, stepSequenceBackward[j][2]);
      digitalWrite(motorPin4, stepSequenceBackward[j][3]);
    }
    delay(stepDelay); // Fine-tuned delay for smoother operation
  }
}
```

5.2 Annosteluruuvien ohjaus

Annosteluruuvien ohjaus on tehty periaatteessa saman koodin pohjalle, joten alla vain muutokset. Ensimmäinen poikkeus on muuttujalistassa, jossa on määritetty moottorin kokonaisen kierroksen askeleet `stepsPerTurn`.

```
const int stepsPerTurn=200;// Full stepper motor turn
```

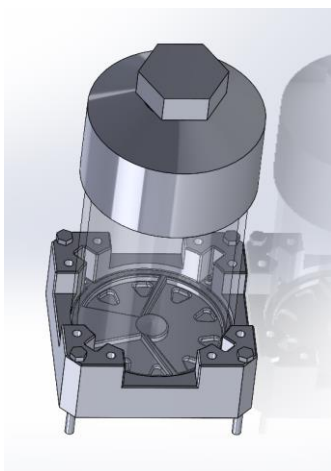
Annosteluruuvien koodin seuraava ero on, kuinka automaattiasetus pyörittää ruuvia. Yksi kierros vastaa materiaalin yhtä annosta ja ruuvien tulee olla oikein mitoitettu. Prototyypin ruuvissa on kymmenen kierrettä. Tämän takia ohjauskoodi on sovitettu kääntymään yhdestä kymmeneen kierrokseen. Ohjauskoodi muuntaa analogiasignaalin arvon lähimpään täyteen kierrokseen. Tämän lisäksi manuaalinen asetus kääntää moottorin akselia yhden askeleen sijaan yhden kierroksen.

```
void autoMode()
{
    potentiometerValue=analogRead(potentiometerPin);
    stepsToMove=map(potentiometerValue,0,(1023*4)/5, stepsPerTurn,
        stepsPerTurn*10);// Scale for 1-10 turns
    stepsToMove=(stepsToMove/stepsPerTurn)*stepsPerTurn;
    // Round to nearest full turn

    bool moveForward=digitalRead(clockwisePin)==HIGH;
    if (digitalRead(counterclockwisePin)==HIGH)
    {
        moveForward=false;// Override if counterclockwise is pressed
        if (digitalRead(controlPin)==HIGH)
        {
            moveStepper(stepsToMove, moveForward);
            delay(500);
        }
        else
        {
            disableMotor();// Turn off motor when not in use
        }
    }
}
```

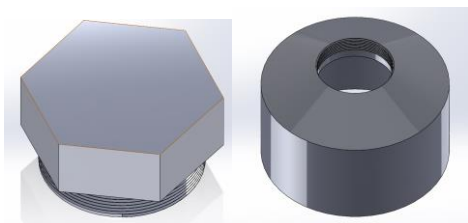
6 LOPPUTULOS

Kuvassa 14 on valmis annostelulevylle tehty värilisäyslaitteen kokoonpano. Kokoonpanossa on mukana kierteillä varustettu korkki. Korkki on vasemmalla ja materiaalisäiliön kansi on oikealla kuvassa 15. Säiliön ulkohalkaisija on 110 mm ja sisähalkaisija on 100 mm seinämän paksuuden ollessa 5 mm. Säiliön materiaalin on suunniteltu olevan läpinäkyvää akryyliä. Läpinäkyvyys mahdollistaa säiliön materiaalitilavuuden tarkistamisen silmämääräisesti tai myöhemmin sensorilla. Säiliö on asetettu annostelulevyn suojan päälle, joka on vasemmalla kuvassa 16. Annostelulevy on suojan alla. Annostelulevyjä tehtiin kaksi, yksi isommalle moottorille mitoitettuna ja toinen pienemmälle oikealla kuvassa 17. Laitteen pohjaan tehtiin kiinnitysreiät sekä Nema 17- että Nema 23 -askelmoottoreille. Ne ovat nähtävissä kuvasta 16. Ruuvien reiät on sovitettu M3- ja M5-ruuveille.



Kuva 14. Annostelulevyn mekaaninen kokoonpano (Kuva: Janrik Hermunen).

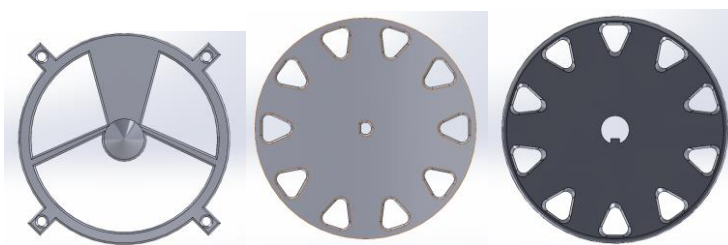
Korkissa kuvassa 15 on vasemmalla M8 kierteet. Jos korkki valmistetaan 3D-tulostamalla, tarkistettava onnistuuko kierteen tulostaminen. Säiliön kannen vastaavat kierteet korkille oikealla kuvassa 15.



Kuva 15. Vasemmalla on kierteellä varustettu korkki.

Oikealla materiaalisäiliön kansi (Kuva: Janrik Hermunen).

Annostelulevyn suojan viimeisin versio on vasemmalla kuvassa 16. Annostelulevyt on mitoitettu kahdelle moottorin akselikoolle. Toiminnaltaan ne ovat samantyyppiset ja geometrialtaan yhtenevät. Ainoa ero on akseleiden kiinnitys.



Kuva 16. Vasemmalla annostelulevyn suoja

Oikealla annostelulevyt (Kuva: Janrik Hermunen).

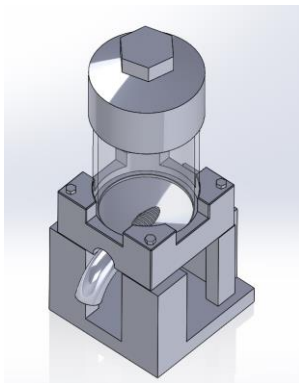
Pohja toteutettiin 3D-tulostamalla, jotta voidaan käyttää M4 kierreholkkeja. Ne voidaan kolvilla painaa suoraan materiaaliin. Valmiit piirustukset ovat liitteessä 9. Moottoreiden reiät ja kaikkien päälle tulevien osien paikat näkyvät kuvassa 16.



Kuva 17. säiliön pohja (Kuva: Janrik Hermunen).

Sähköjen kiinnitykseen on tarkoitus tehdä M6 ruuvien avulla irrallinen pohja, jonka voi kiinnittää suoraan CEAD-materiaalin käsittelylaitteen valmiisiin ruuvi-paikkoihin.

Kuvassa 18 on vaihtoehtoinen toteutus annosteluruuville. Kokoonpanossa säiliö, kansi ja korkki ovat samat kuin annostelulevyn mallissa. Kuvassa 18 on myös nähtävissä yksi mahdollinen pohjan idea. Annosteluruuvien mallin piirustukset on esitelty liitteessä 10.



Kuva 18. Annosteluruuvien mekaaninen kokoonpano (Kuva: Janrik Hermunen).

7 POHDINTA

Levysyöttimen prototyyppiä testattiin Tampereen ammattikorkeakoulun automaatiotekniikan laboratoriossa. Testissä käytettiin laboratorion säädettäviä jännitelähteitä ja valmiita kytkentälevyjä. Sähköpiiri rakennettiin aluksi käyttäen säädettävää vastusta, jolla simuloitiin robotin 0–10 voltin analogiaulostulon toimintaa.

Ohjelman lopullisen skaalaamisen kannalta oli tärkeää selvittää kuinka se toimi jännitteen pudotuspiirin kanssa. Kytkeä toteutettiin uudelleen kytkemällä A0-nasta jännitteen pudotuspiirin kautta säädettävään jännitelähteeseen. Moottorin liikettä seurattaessa havaittiin sen toimivan toivotulla tavalla.

Kun laite tullaan ottamaan käyttöön, on tärkeää kalibroida ja tilastoida, mikä jännite vastaa mitään liikettä. Ohjauskoodi on toteutettu niin, että moottori kääntää levyä jännitteen säädön mukaan.

Esimerkkikoodi on skaalattu niin, että neljä volttia vastaa 200 askelta ja nolla volttia nollaa askelta. Ohjelma on säädetty 11 eri liikeasteelle 0–200 välillä ja askeleet ovat 20 välein. Laskennallisesti tämä tarkoitti, että prototyypin koodi maksimissaan teki yhden täyden kierroksen, joka vastasi noin 10 % annosta.

Testattaessa materiaalin kanssa havaittiin jumiintumistilanteiden olevan yhä ongelma Nema 17 -moottorille. Ehdotuksena on palata käyttämään esimerkiksi Nema 23 -moottoria kuten alussa oli ehdotettu. Isomman moottorin suuremmalla väännöllä voidaan vähentää jumiintumista. Nema 23 tarvitsee teollisemman ajurin ja suuremman jännitelähteen.

Todellisen laitteen valmistuksessa on selvitettävä, mitkä osat halutaan valmistaa esimerkiksi alumiinista. Jumiintumiset saattavat vähentyä vaihtamalla prototyyppissä käytetty ABS-muovista valmistettu annostelulevy granulaattia kovempaan materiaaliin. Annostelulevyn reikien koon ja muodon muuttaminen vaikuttaa toimintaan. On myös järkevää selvittää näiden muuttamisen vaikutus laitteen toimintaan.

Yksi tavoitteista oli selvittää kuinka halvalla voi teettää toiminnallisen värinlisäyslaitteen. Tätä varten selvitettiin eri elektroniikkaosien hintoja liitteessä 11. Arvioksi tuli noin 73 €. Kustannuksissa ei ole otettu huomioon posti- ja toimituskuluja tai tulleja. Painekeytkimen hintaa ei saatu selville, joten se on jätetty pois. Nema 17 hinta on noin 20 € ja Nema 23 noin 54 €. Kyseessä on arvio 29.3.2025 netistä tehdyn selvityksen mukaan.

Kokonaishintaan vaikuttaa, jos osia tehdään alumiinista. Alumiinin ostohinta vaihtelee eri toimittajien välillä. Työstöhinnat voivat nostaa kustannuksia riippuen siitä tehdäänkö koneistus Tampereen ammattikorkeakoululla vai ulkoistetaanko se. Työstön kannalta harkittavana olevat osat ovat pohja, annostelulevy ja annostelulevynsuoja.

Todennäköisesti annostelulevy ja annostelulevynsuoja tullaan työstämään alumiinista materiaalin kovuuden takia. Näin voidaan vähentää laitteen jumiintumista. Pohja voidaan tehdä alumiinista, mutta se ei ole välttämätöntä, sillä siihen ei kohdistu juurikaan voimia, jotka rasittaisivat 3D-tulostettua osaa. Moottorin akseli ei saa osua säiliön pohjaan.

Mahdollisia muutoksia suunnitteluun voi joutua tekemään, kun valitaan T-liitännäinen putki, jonka kautta granulaatit annostellaan 3D-tulostussolun pneumaattiseen materiaalin siirtokanavaan.

Ruuvisyöttimen prototyyppiä ei toteutettu tässä yhteydessä. Se on kuitenkin toimiva idea, jos kaivataan toisenlaista menetelmää. Ruuvisyötin vaatii testaamista kierteen mitoittamisessa. Todennäköisesti sitä pitää ajaa Nema 23 -askelmoottorilla kuten levysyötintäkin. Ruuvin materiaali lienee parasta tehdä alumiinista, jotta se kestää käyttöä ja moottorin tuottamia voimia.

Lopputuloksena syntyi toimiva prototyyppi levymäisestä materiaalin syöttölaitteesta. Sen testaaminen onnistuttiin toteuttamaan laboratorioympäristössä. Jos aikaa olisi ollut enemmän, olisin voinut valmistaa ruuvisyöttimestä prototyypin.

8 JOHTOPÄÄTÖKSET

8.1 Mekaniikka

Testituloksista havaitaan, kuinka epätarkkaa annostelu voi olla riippumatta siitä käytetäänkö annosteluruuvia vai -levyä, koska granulaatit ovat kooltaan ja muotoeroiltaan suuria. Koeajolla voidaan selvittää materiaalin kokoerosta johtuvaa todellisen syötön toteutumaa. Värin annostelussa tärkeintä on yhdenmukainen värjäys, koska raidallisen tulosteen teko ei ole tämän laitteen alkuperäinen tarkoitus.

8.2 Sähkö

Periaatteeltaan isomman moottorin ajaminen on toteutettavissa esitetyllä piirillä, jos ajuri vaihdetaan teollisempaan malliin. Kun halutaan vaihtaa esimerkiksi Nema 23 -askelmoottoriin, tarvitsee moottorin ohjaimen olla joko isommille virroille mitoitettu tai askelmoottoria voidaan ajaa esimerkiksi releillä. Kaikki komponentit, jotka ovat kiinni moottorissa ja suuremmassa jännitteessä, pitää vaihtaa asianmukaisiin komponentteihin. Piirin logiikka on toiminnaltaan sama moottorin koosta riippumatta.

Moottorista palaava jännite voidaan tasata kytkemällä diodeja jokaiselle johtimelle. Tässä tapauksessa diodit on kytketty 12 volttiin. Diodit eli puolijohteet mahdollistavat jännitteen siirtymisen vain toiseen suuntaan. Vastaavalla tavalla myös jännite voi alittaa 0 V, joten toinen diodisetti on kytketty maahan. Maahan kytketyt puolijohteet pyrkivät tasaamaan jännitteen arvoon nolla.

8.3 Ohjaus

Ohjauksen päivittämisen tai mahdollisten muutosten kannalta on järkevää tutustua materiaaliin, jota käytettiin koodin kirjoittamisen tukena. Githubin tietokannat askelmoottoreiden ohjauksen koodaamisesta ovat hyvä ensiaskel Arduinon omien sivujen ohella.

Ohjaukseen luotu koodi toimii eri kokoisille 2-napaisille askelmoottoreille. Iso rajoitus, joka tulee vaatimaan muutoksen, on askelmäärän muuttaminen. Molemmat ohjelmat on kirjoitettu toimimaan 200 askeleen moottorilla ja askeleet on tehty seuraamaan täysiä askelia. Kun tarkkuutta halutaan lisätä, voi moottorille aktivoida mikroaskeleita. Tämä lisää kokonaisaskelmäärää, koska jokainen askel jaetaan pienempiin osiin. Myös vaihteiston lisääminen muuttaa askelmäärää. Jos esimerkiksi lisätään 1:15 planeettavaihteet kuten liitteessä 1, tulee askelia lisää.

Uusi askelmäärä voidaan laskea verrannolla kaavalla 2, jossa x ja y ovat askelten määrät täydessä kierroksessa, ja a sekä b ovat askelmääriä vastaavia kierroslukumääriä ennen vaihteistoa.

$$\frac{1}{200} = \frac{15}{y} \Rightarrow y = 3000 \text{ askelta} \quad (2)$$

Moottorissa, jossa on 1:15 vaihteisto, tarvitsee moottorin akselin pyöriä 15 kertaa 200 askelta, jotta vaihteiston akseli pyörii yhden kierroksen. Tämä tulee huomioida ohjauksen koodia muutettaessa, jos moottoria vaihdetaan.

Prototyypin piiri toteutettiin käyttäen säätövastusta ja tasaista 5 voltin jännitettä. Periaatteessa ei ole väliä, säädetäänkö jännitettä suoraan robotin kautta vai vastuksella. Lopputulos koodissa on sama (testattu säädettävällä laboratorion jännitelähteellä).

Koodissa oli yksi ongelma. Koska askeleet otettiin neljän askeleen ryhminä, jouduttiin yhden askeleen ottamiseksi laittamaan manuaaliasetuksen liike neljäksi askeleeksi. Ongelma ratkaistiin jakamalla steps-muuttuja neljällä.

8.4 Lopputulos

Laitteen pohja tehtiin laatikkomaiseksi, jotta laitteita tulevaisuudessa voitaisiin yhdistää toisiinsa ja sijoittaa rinnan. Prototyypivaiheessa ei testattu moniajota useammalla samanlaisella kytkennällä, mutta sen pitäisi olla mahdollista pienellä koodimuutoksella.

Arduinoja voi kytkeä toisiinsa niin, että robotin digitaalilähdöistä valitaan mikä yksikkö on aktiivinen. Se toteutetaan ajamalla looginen 1 (HIGH) signaali valitun laitteen Arduinon digitaalinaan. Lopputuloksen pohjalta luotan, että suunniteltu laite saadaan rakennettua ja otettua käyttöön tarvittaessa.

Kiinnityksessä on huomioitava sähköjen asentaminen. Jos aiotaan käyttää eri kokoisia moottoreita, tarvitaan säädettävä kytkentäalusta. Alusta voidaan toteuttaa 3D-tulostamalla. Siinä voi olla kääntyvät korvakkeet, joilla voi lukita piirilevyt paikoilleen. Toinen vaihtoehto sähköjen kiinnitykseen on käyttää kierreholkkeja tekemällä ruuviliitos reunoista esimerkiksi aluslevyillä.

Jos kiinnitys halutaan toteuttaa käyttäen tulostussolun olemassa olevia ruuvipaikkoja, tämä voidaan toteuttaa esimerkiksi taivutetusta alumiinista tai teräksestä (vaatii lujuuslaskennan, kun tiedetään lopullisen version ja granulaattien paino säiliössä) tehdyllä kulmaraudalla. Toinen mahdollinen kiinnitysvaihtoehto on tehdä pellistä taivutettu aihio, johon värilisäyslaite voidaan kiinnittää esimerkiksi pohjasta.

LÄHTEET

3D Jake Kierreosat haettu 10.2.2025

<https://www.3djake.fi/3djake/kierreosat-50-kpl>

Allegro Micro Systems LLC 2014. DMOS Microstepping Driver with Translator And Overcurrent Protection.

https://www.pololu.com/file/0j450/a4988_dmos_microstepping_driver_with_translator.pdf

AndreaLombardo (2023). L298N Library. Haettu 11.2.2025

<https://github.com/AndreaLombardo/L298N>

Arduino (2025). Analog Read Serial haettu 16.3.2025

<https://docs.arduino.cc/built-in-examples/basics/AnalogReadSerial/>

Arduino 2025. Arduino® UNO R3

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

Arduino kotisivut 31.3.2025

<https://www.arduino.cc/>

Arduino. (2025). Arduino and Stepper Motor Configurations. Haettu 9.2.2025.

<https://docs.arduino.cc/learn/electronics/stepper-motors/>

Arduino. (2025). Stepper. Haettu 9.2.2025.

<https://docs.arduino.cc/libraries/stepper/>

Components 101. (2021). L298N Motor Driver Module. Haettu 5.2.2025.

<https://components101.com/modules/l293n-motor-driver-module>

Cooper Bussmann. (2005). Overcurrents and Voltage Ratings.

<https://www.eaton.com/content/dam/eaton/products/electrical-circuit-protection/fuses/protective-device-ratings/bus-ele-voltage-rating.pdf>

HowToMechatronics 2023. Stepper Motors and Arduino - The Ultimate Guide. Katsottu 6.2.2025

https://www.youtube.com/watch?v=7spK_BkMJys&ab_channel=HowToMechatronics

IAC Pneumatic Conveying System Parts and Components. Haettu 18.3.2025

<https://iac-intl.com/parts/pneumatic-material-handling-components/>

IAC Pneumatic Conveying Systems: Designing the Feed and Discharge for Maximum Production Rates. Haettu 18.3.2025

<https://news.iac-intl.com/pneumatic-conveying-systems-feed-and-discharge>

Last Minute Engineers. Control Stepper Motor with L298N Motor Driver & Arduino. Haettu 11.2.2025

<https://lastminuteengineers.com/stepper-motor-l298n-arduino-tutorial/>

laurb9 2023. StepperDriver. Haettu 11.2.2025

<https://github.com/laurb9/StepperDriver>

Mills Dip Tech (Eng), CEng. Pneumatic Conveying Design Guide. 3rd ed., Elsevier Science & Technology, 2015,

<https://doi.org/10.1016/C2014-0-02678-0>.

https://andor.tuni.fi/permalink/358FIN_TAMPO/176jdvt/cdi_proquest_ebookcentral_EBC4091005

Stepper Online

<https://www.omc-stepperonline.com/nema-17-bipolar-1-8deg-26ncm-36-8oz-in-0-4a-12v-42x42x34mm-4-wires-17hs13-0404s1>

RS-online Toggle Switches guide Haettu 19.3.2025

<https://uk.rs-online.com/web/content/discovery/ideas-and-advice/toggle-switches-guide>

RS-online Toggle Switches Haettu 19.3.2025

<https://uk.rs-online.com/web/c/switches/toggle-switches-slide-switches/toggle-switches/?applied-dimensions=4294535985>

Science Direct Screw Feeder: Haettu 18.3.2025

<https://www.sciencedirect.com/topics/engineering/screw-feeder>

Stepper Online (2018) (Nema 23) 23HS30-2804S-PG15 <https://www.omc-stepperonline.com/download/23HS30-2804S-PG15.pdf?srsId=AfmBOopAdFfUiC-CmgJfDnq2b6fgrvFW4KUkiE7b7dwSrknMzEYyGMbE>

STMicroelectronics 2023. Dual full-bridge driver L298 Haettu 31.3.2025

<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.st.com/resource/en/datasheet/l298.pdf>

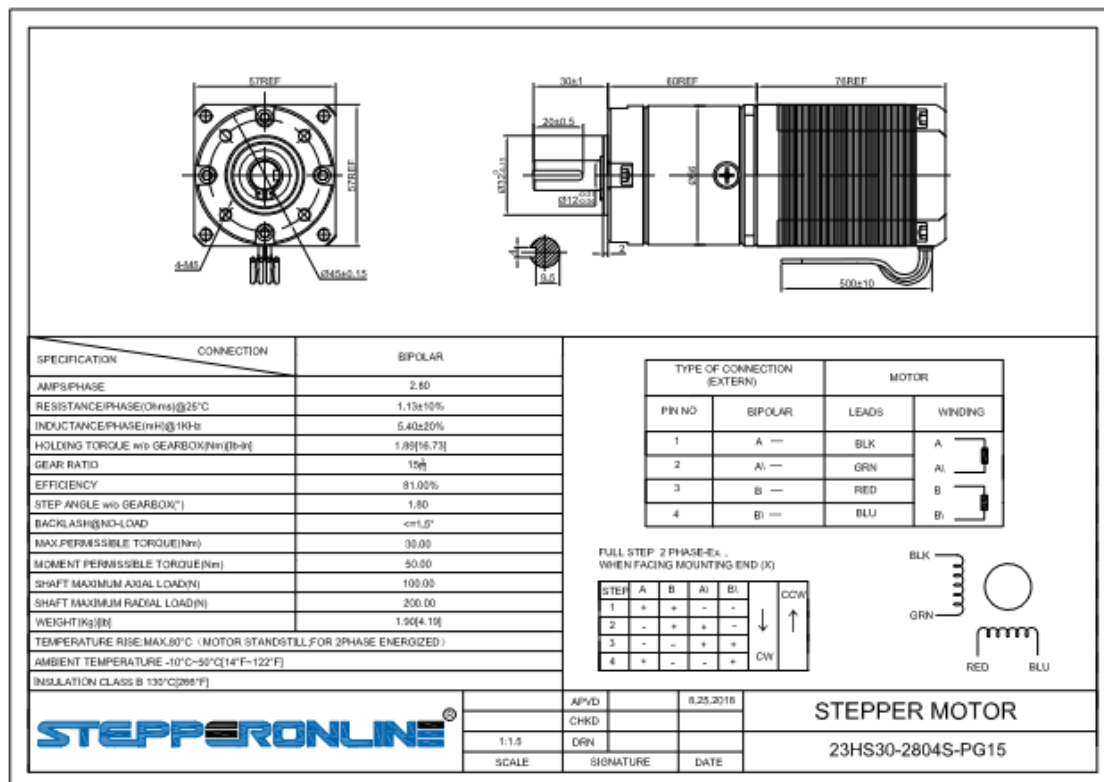
Swe-Check The Fuse Bible 22 nd Edition

https://www.swe-check.com.au/pdfs/fuse_bible_complete.pdf

LIITTEET

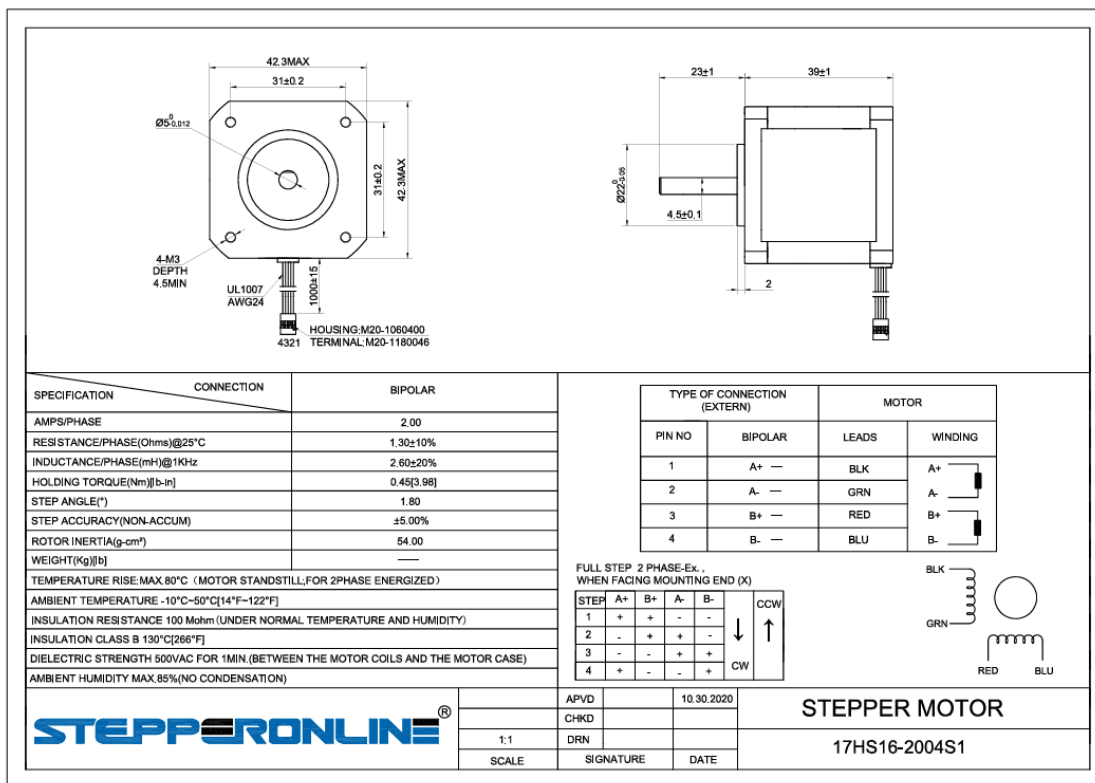
Liite 1 Nema 23 Askelmoottorin Datalehti

<https://www.omc-stepperonline.com/nema-23-stepper-motor-bipolar-l-56mm-w-gear-ratio-15-1-planetary-gearbox-23hs22-2804s-pg15>



Liite 2 Nema 17 askelmootorin Datalehti

<https://www.omc-stepperonline.com/nema-17-bipolar-1-8deg-26ncm-36-8oz-in-0-4a-12v-42x42x34mm-4-wires-17hs13-0404s1>



https://www.pololu.com/file/0j450/a4988_dmos_microstepping_driver_with_translator.pdf

**A4988**

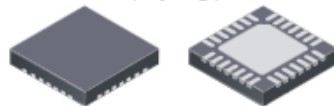
DMOS Microstepping Driver with Translator And Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $1/2$, $1/4$, $1/8$, and $1/16$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Approximate size

Description

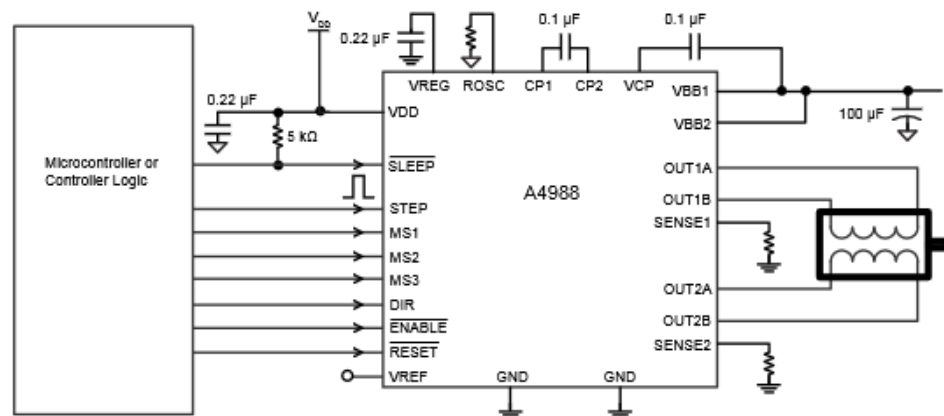
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ± 2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page...

Typical Application Diagram



A4988***DMOS Microstepping Driver with Translator
And Overcurrent Protection*****Description (continued)**

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix -T), with 100% matte tin plated leadframes.

Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{DD}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		-0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		-0.3 to 5.5	V
Motor Outputs Voltage			-2.0 to 37	V
Sense Voltage	V_{SENSE}		-0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	-20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	T_{stg}		-55 to 150	°C



Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

A4988**DMOS Microstepping Driver with Translator
And Overcurrent Protection**ELECTRICAL CHARACTERISTICS¹ at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. ²	Max.	Units
Output Drivers						
Load Supply Voltage Range	V _{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V _{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R _{DS(ON)}	Source Driver, I _{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I _{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V _F	Source Diode, I _F = –1.5 A	–	–	1.2	V
		Sink Diode, I _F = 1.5 A	–	–	1.2	V
Motor Supply Current	I _{BB}	f _{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I _{DD}	f _{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	V _{IN(1)}		V _{DD} ×0.7	–	–	V
	V _{IN(0)}		–	–	V _{DD} ×0.3	V
Logic Input Current	I _{IN(1)}	V _{IN} = V _{DD} ×0.7	–20	<1.0	20	μA
	I _{IN(0)}	V _{IN} = V _{DD} ×0.3	–20	<1.0	20	μA
Microstep Select	R _{MS1}	MS1 pin	–	100	–	kΩ
	R _{MS2}	MS2 pin	–	50	–	kΩ
	R _{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	V _{HYS(IN)}	As a % of V _{DD}	5	11	19	%
Blank Time	t _{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t _{OFF}	OSC = V _{DD} or GND	20	30	40	μs
		R _{OSC} = 25 kΩ	23	30	37	μs
Reference Input Voltage Range	V _{REF}		0	–	4	V
Reference Input Current	I _{REF}		–3	0	3	μA
Current Trip-Level Error ³	err _I	V _{REF} = 2 V, %I _{TRIPMAX} = 38.27%	–	–	±15	%
		V _{REF} = 2 V, %I _{TRIPMAX} = 70.71%	–	–	±5	%
		V _{REF} = 2 V, %I _{TRIPMAX} = 100.00%	–	–	±5	%
Crossover Dead Time	t _{OT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold ⁴	I _{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T _{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T _{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V _{DDUVLO}	V _{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	V _{DDUVLOHYS}		–	80	–	mV

¹For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.²Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.³V_{ERR} = [(V_{REF}/8) - V_{SENSE}] / (V_{REF}/8).⁴Overcurrent protection (OCP) is tested at T_A = 25°C in a restricted range and guaranteed by characterization.

Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Liite 4 L298N Datalehdestä otteita

<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.st.com/resource/en/datasheet/l298.pdf>

1(4)



L298
Pin description

3 Pin description

Figure 2. Pin configuration

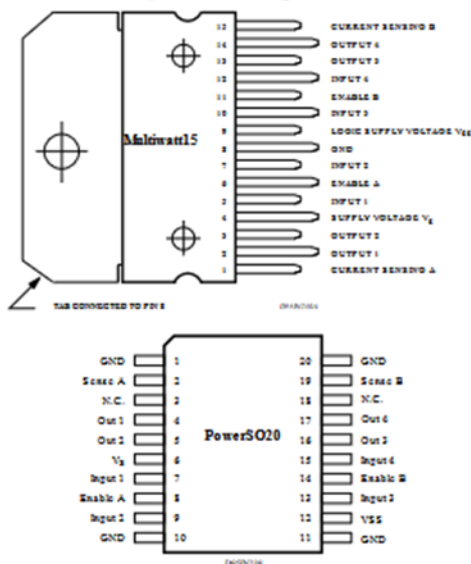


Table 3. Pin function

MW.15	Power SO	Name	Function
1, 15	2, 19	Sense A, Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2, 3	4, 5	Out 1, Out 2	Outputs of the bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_s	Supply voltage for the power output stages. Anon-inductive 100nF capacitor must be connected between this pin and ground.
5, 7	7, 9	Input 1, Input 2	TTL compatible inputs of the bridge A.
6, 11	8, 14	Enable A, Enable B	TTL compatible enable input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1, 10, 11, 20	GND	Ground.
9	12	VSS	Supply voltage for the logic blocks. A 100nF capacitor must be connected between this pin and ground.
10, 12	13, 15	Input 3, Input 4	TTL compatible inputs of the bridge B.
13, 14	16, 17	Out 3, Out 4	Outputs of the bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3, 18	N.C.	Not connected



4 Electrical characteristics

Table 4. Electrical characteristics

 (V_S = 42 V; V_{SS} = 5 V; T_J = 25 °C; unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V _S	Supply voltage (pin 4)	Operative condition	V _{IH} +2.5		46	V
V _{SS}	Logic supply voltage (pin 9)		4.5	5	7	V
I _S	Quiescent supply current (pin 4)	V _{en} = H; V _I = L; I _L = 0		13	22	mA
		V _{en} = H; V _I = H; I _L = 0		50	70	mA
		V _{en} = L; V _I = X			4	mA
I _{SS}	Quiescent current from V _{SS} (pin 9)	V _{en} = H; V _I = L; I _L = 0		24	36	mA
		V _{en} = H; V _I = H; I _L = 0		7	12	mA
		V _{en} = L; V _I = X			6	m
V _{IL}	Input low voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{IH}	Input high voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low voltage input current (pins 5, 7, 10, 12)	V _I = L			-10	μA
I _{IH}	High voltage input current (pins 5, 7, 10, 12)	V _I = H ± V _{SS} -0.6V		30	100	μA
V _{enL}	Enable low voltage (pins 6, 11)		-0.3		1.5	V
V _{enH}	Enable high voltage (pins 6, 11)		2.3		V _{SS}	V
I _{enL}	Low voltage enable current (pins 6, 11)	V _{en} = L			-10	μA
I _{enH}	High voltage enable current (pins 6, 11)	V _{en} = H ± V _{SS} -0.6V		30	100	μA
V _{CSat(H)}	Source saturation voltage	I _L = 1 A	0.95	1.35	1.7	V
		I _L = 2 A		2	2.7	V
V _{CSat(L)}	Sink saturation voltage	I _L = 1 A ⁽¹⁾	0.85	1.2	1.6	V
		I _L = 2 A ⁽¹⁾		1.7	2.3	V
V _{CSat}	Total drop	I _L = 1 A ⁽¹⁾	1.80		3.2	V
		I _L = 2 A ⁽¹⁾			4.9	V
V _{sens}	Sensing voltage (pins 1, 15)		-1 ⁽²⁾		2	V
T ₁ (V _I)	Source current turn-off delay	0.5 V _I to 0.9 I _L ⁽³⁾ , ⁽²⁾		1.5		μs
T ₂ (V _I)	Source current fall time	0.9 I _L to 0.1 I _L ⁽³⁾ , ⁽²⁾		0.2		μs
T ₃ (V _I)	Source current turn-on delay	0.5 V _I to 0.1 I _L ⁽³⁾ , ⁽²⁾		2		μs
T ₄ (V _I)	Source current rise time	0.1 I _L to 0.9 I _L ⁽³⁾ , ⁽²⁾		0.7		μs
T ₅ (V _I)	Sink current turn-off delay	0.5 V _I to 0.9 I _L ⁽³⁾ , ⁽²⁾		0.7		μs
T ₆ (V _I)	Sink current fall time	0.9 I _L to 0.1 I _L ⁽³⁾ , ⁽²⁾		0.25		μs
T ₇ (V _I)	Sink current turn-on delay	0.5 V _I to 0.9 I _L ⁽³⁾ , ⁽²⁾		1.6		μs
T ₈ (V _I)	Sink current rise time	0.1 I _L to 0.9 I _L ⁽³⁾ , ⁽²⁾		0.2		μs
f _C (V _I)	Commutation frequency	I _L = 2 A		25	40	KHz
T ₁ (V _{en})	Source current turn-off delay	0.5 V _{en} to 0.9 I _L ⁽³⁾ , ⁽²⁾		3		μs
T ₂ (V _{en})	Source current fall time	0.9 I _L to 0.1 I _L ⁽³⁾ , ⁽²⁾		1		μs



5 Application information

(Refer to the Section 1 Block diagram)

5.1 Power output stage

The L298 integrates two power output stages (A ; B).

The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

5.1.1 Input stage

Each bridge is driven by means of four gates, managed by following inputs:

- In1, In2 and EnA
- In3, In4 and EnB

The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge.

All the inputs are TTL compatible.

5.2 Suggestions

A non inductive capacitor, usually of 100 nF, is required between both V_S and V_{SS} to ground, as near as possible to GND pin. When the bulk capacitor of the power supply is too far from the IC, a second smaller one must be placed near the L298.

The sense resistor, low inductance, must be grounded near the negative pole of V_S that must be near the GND pin of the IC.

Each input must be connected to the source of the driving signals by means of a very short path.

Before to Turn-ON the supply voltage and before to Turn it OFF, the enable inputs must be driven to the low state.

5.3 Applications

Figure 8 shows a bidirectional DC motor control schematic diagram for which only one bridge is needed.

The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ ns) that must be chosen of a V_F as low as possible at the worst case of the load current.

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 A must never be exceeded.

When the repetitive peak current needed from the load is higher than 2 A, a paralleled configuration can be chosen (See Figure 9).

This solution can drive up to 3 A in dc operation and until 3.5 A of a repetitive peak current.

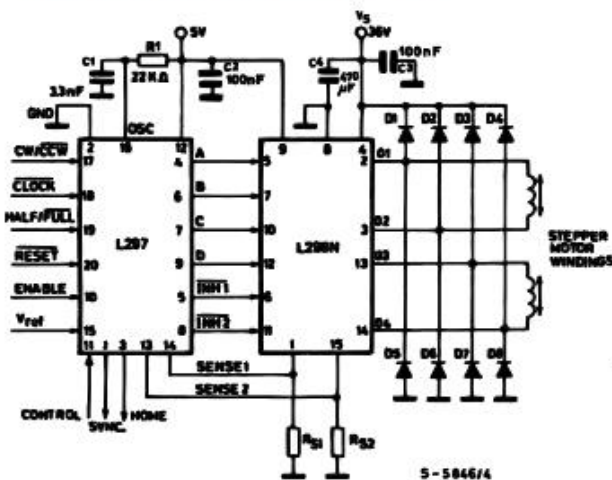
On Figure 10 it is shown the driving of a two phase bipolar stepper motor; the needed signals to drive the inputs of the L298 are generated, in this example, by the IC L297.

Figure 11 shows an example of P.C.B. designed for the application of Figure 10.

Figure 12 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the IC L6506.

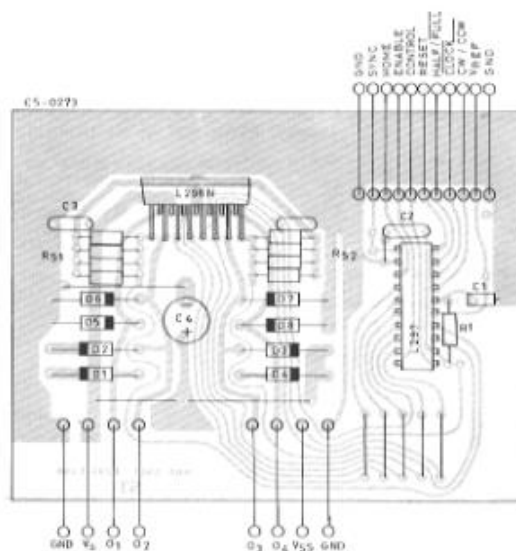


Figure 10. Two phase bipolar stepper motor circuit



Note: $R_{S1} = R_{S2} = 0.5 \Omega$.

Figure 11. Suggested printed circuit board layout for the circuit of fig. 10 (1:1 scale)



Liite 5 Oteita Arduino Unon Datalehdestä

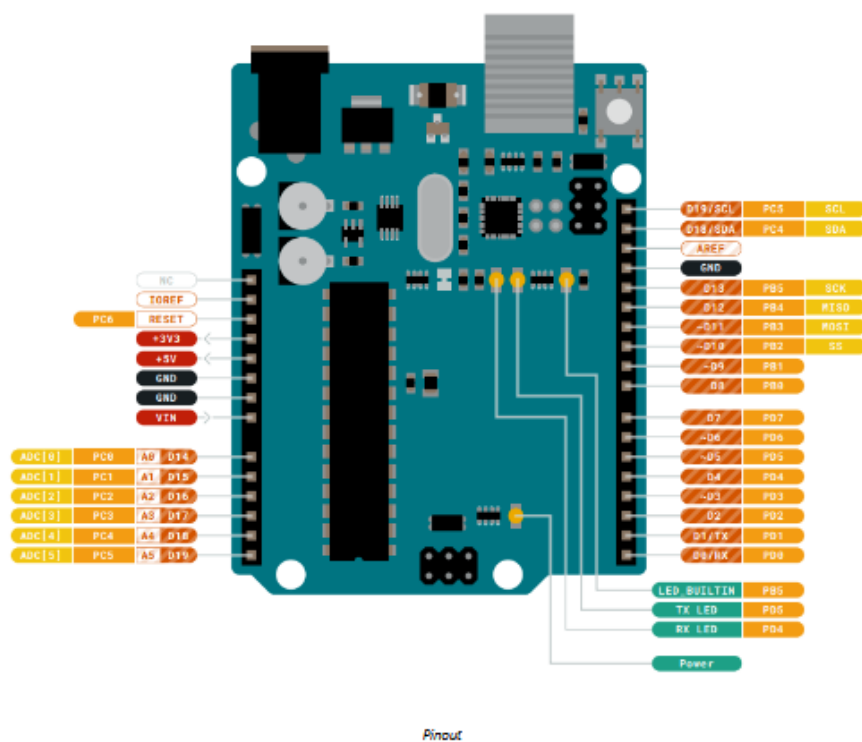
<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

1(2)



Arduino® UNO R3

5 Connector Pinouts





5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Liite 6 Ote Painekeytkimen Datalehdessä

PS1000-TF22054EN



ORIGINAL INSTRUCTIONS

Instruction Manual

Air Checker: Electronic Pressure Switch
PS1000 / PS1100 / PS1200



The intended use of the compact pressure switch is to measure, monitor and display pressure and provide an output signal.

1 Safety instructions

These safety instructions are intended to prevent hazardous situations and/or equipment damage. These instructions indicate the level of potential hazard with the labels of "Caution," "Warning" or "Danger." They are all important notes for safety and must be followed in addition to International Standards (ISO/IEC) and other safety regulations.

ISO 4414: Pneumatic fluid power - General rules relating to systems.
ISO 4413: Hydraulic fluid power - General rules relating to systems.
IEC 60204-1: Safety of machinery - Electrical equipment of machines. (Part 1: General requirements).
ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots.

- Refer to product catalogue, Operation Manual and Handling Precautions for SMC Products for additional information.
- Keep this manual in a safe place for future reference.

Caution	Caution indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
Warning	Warning indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
Danger	Danger indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.

Warning

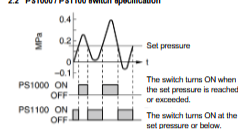
- Always ensure compliance with relevant safety laws and standards.
- All work must be carried out in a safe manner by a qualified person in compliance with applicable national regulations.
- This product is class A equipment intended for use in an industrial environment. There may be potential difficulties in ensuring electromagnetic compatibility in other environments due to conducted or radiated disturbances.
- Refer to the operation manual or catalogue on the SMC website (URL: <https://www.smcworld.com>) for more safety instructions.

2 Specifications

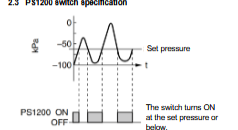
2.1 General specifications

	PS1000	PS1100	PS1200
Model No.	PS1000	PS1100	PS1200
Switch output	Pressure $\neq$ Set pressure: ON	Pressure $\neq$ Set pressure: ON	Pressure $\neq$ Set pressure: ON
Max. operating pressure	1 MPa	500 kPa	500 kPa
Setting pressure range	-0.1 to 0.45 MPa	-0.1 to 0.4 MPa	-100 to 0 kPa
Applicable fluids	Air, non-corrosive gas, non-flammable gas		
Indicator light	Switch ON: Red LED turns ON		
Temp. characteristics	$\pm 3\%$ F.S.		
Repeatability	$\pm 1\%$ F.S.		
Hysteresis	4% F.S. or less	10% F.S. or less	
Load voltage	12 to 24 VDC $\pm 10\%$, ripple (p-p) 10% or less		
Load current	5 to 40 mA		
Leakage current	1 mA max.		
Internal voltage drop	5 V max.		
Operating temperature range	0 to 60°C (no condensation)		
Insulation resistance	2 M Ω or more at 500 VDC (between live parts and case)		
Withstand voltage	1000 VAC, 1 minute (between live parts and case)		
Port sizes	R06: $\phi 6$ reducer R07: $\phi 1/4$ reducer		
Weight	5 g (excluding lead wire)		
Enclosure rating	IP40		
Lead wire	Oil resistant vinyl cable 2 cores, $\phi 2.5$, 3 m long Conductor cross section 0.18 mm ² Insulator O.D. 0.85 mm		
Material of wetted parts	Sensor: Silicon, Body: PBT, O-ring: HNBR		

2.2 PS1000 / PS1100 switch specification



2.3 PS1200 switch specification



3 Installation

3.1 Installation

Warning

Do not install the product unless the safety instructions have been read and understood.

3.2 Piping

Caution

- Before connecting piping make sure to clean up chips, cutting oil, dust etc.

3.3 Environment

Warning

- Do not use in an environment where corrosive gases, chemicals, salt water or steam are present.
- Do not use in an explosive atmosphere.
- Do not expose to direct sunlight. Use a suitable protective cover.
- Do not install in a location subject to vibration or impact. Check the product specifications.
- Do not mount in a location exposed to radiant heat.

3.4 Lubrication

Caution

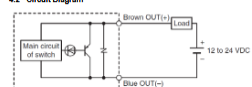
- SMC products have been lubricated for life at manufacture, and do not require lubrication in service.
- If a lubricant is used in the system, use turbine oil Class 1 (no additive), ISO VISO. Once lubricant is used in the system, lubrication must be continued because the original lubricant applied during manufacturing will be washed away.

4 Wiring

4.1 Wiring

- Connections should be made with the power supply turned OFF.
- Use a separate route for the product wiring and any power or high voltage wiring. Otherwise, malfunction may result due to noise.
- Incorrect wiring may cause damage to the pressure switch, breakdown and malfunction. Confirm the colour of the wires with the operation manual before wiring.
- Wiring applying repeated bending and tensile stress to the lead wire can break the circuit.
- The recommended bend radius of the lead wire is 6 times the outside diameter of the sheath, or 33 times the outside diameter of the insulation material, whichever is larger.
- Avoid defective insulation (crossed lines with other circuit, ground fault, defective insulation between terminals, etc.) with the wiring. Excessive current can flow through the pressure switch, which may cause damage.
- If the pressure switch is turned on with no load connected to the switch, over current will flow, causing the pressure switch to break instantly.
- The pressure switch has no reverse connection protection for the brown (+) and blue (-) of the power supply line.
- Do not exceed the maximum allowable load (24 VDC/40 mA) specified.

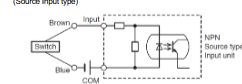
4.2 Circuit Diagram



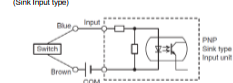
4 Wiring (continued)

4.3 Wiring diagram

Example of connection to a PLC (sequence controller). (Source: Input type)



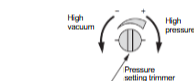
(Sink Input type)



5 Setting

- Adjust the calibration adjustment to set the ON pressure.
- Rotate clockwise to increase the set pressure. For setting vacuum pressure rotate anticlockwise.
- For setting, use a flat blade screwdriver suitable for a trimmer. Rotate lightly to adjust.
- The rotation angle of the trimmer is 220°.

There is a stop provided to prevent the trimmer from rotating beyond its limits. Rotation beyond the limits can damage the trimmer. Adjust the trimmer gently within the rotation angle.

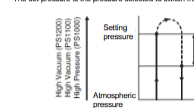


Caution

- Do not use a large screwdriver as this may damage the trimmer groove.
- Gently turn the trimmer and do not exceed the trimmer adjustment range.

5.1 Hysteresis

Hysteresis is the pressure difference between the ON and OFF pressure. The set pressure is the pressure selected to switch from OFF to ON.



5 How to Order

Refer to the operation manual or catalogue on the SMC website (URL: <https://www.smcworld.com>) for How to order information.

Liite 7 Annostelulevyn ohjelma

1(3)

```

// Define motor control pins (L298N Driver)
const int motorPin1 = 8; // IN1
const int motorPin2 = 10; // IN2
const int motorPin3 = 9; // IN3
const int motorPin4 = 11; // IN4

// Define input pins
const int potentiometerPin = A0; // Potentiometer
const int controlPin = 4; // Trigger motor movement
const int autoModePin = 2; // Enable auto mode
const int manualModePin = 3; // Enable manual mode
const int buttonPin = 6; // Manual step button
const int clockwisePin = 5; // Clockwise direction
const int counterclockwisePin = 7; // Counterclockwise direction

// Variables
int potentiometerValue = 0;
int stepsToMove = 0;
bool isAutoMode = false;
bool lastButtonState = HIGH;
bool motorEnabled = false;
int stepDelay = 5; // Fine-tuned delay for smoother movement

void disableMotor() {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    motorEnabled = false;
}

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);
    pinMode(controlPin, INPUT);
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(clockwisePin, INPUT_PULLUP);
    pinMode(counterclockwisePin, INPUT_PULLUP);
    pinMode(autoModePin, INPUT_PULLUP);
    pinMode(manualModePin, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    if (digitalRead(autoModePin) == HIGH) {
        isAutoMode = true;
        autoMode();
    } else if (digitalRead(manualModePin) == HIGH) {
        isAutoMode = false;
        manualMode();
    } else {

```

```

    disableMotor(); // Ensure motor is off when no mode is active
}
}

void autoMode() {
    potentiometerValue = analogRead(potentiometerPin);
    stepsToMove = map(potentiometerValue, 0, (1023 * 4) / 5, 0,
200); // Adjusted for 0-4V range
    stepsToMove = (stepsToMove / 20) * 20; // Round to nearest 20

    bool moveForward = digitalRead(clockwisePin) == HIGH;
    if (digitalRead(counterclockwisePin) == HIGH) moveForward =
false; // Override if counterclockwise is pressed

    if (digitalRead(controlPin) == HIGH) {
        moveStepper(stepsToMove, moveForward);
        delay(500);
    } else {
        disableMotor(); // Turn off motor when not in use
    }
}

void manualMode() {
    bool currentButtonState = digitalRead(buttonPin);
    bool moveForward = digitalRead(clockwisePin) == HIGH;
    if (digitalRead(counterclockwisePin) == HIGH) moveForward =
false;

    if (lastButtonState == HIGH && currentButtonState == LOW) { //
Detect button press
        Serial.println("Manual button pressed, moving four steps");
        moveStepper(4, moveForward); // Move forward or backward based
on switch
        delay(200); // Debounce delay
    }
    lastButtonState = currentButtonState;
}

void moveStepper(int steps, bool moveForward) {
    int stepSequence[4][4] = {
        {HIGH, LOW, HIGH, LOW}, // Step 1
        {LOW, HIGH, HIGH, LOW}, // Step 2
        {LOW, HIGH, LOW, HIGH}, // Step 3
        {HIGH, LOW, LOW, HIGH} // Step 4
    };

    int stepSequenceBackward[4][4] = {
        {HIGH, LOW, LOW, HIGH}, // Reverse Step 4
        {LOW, HIGH, LOW, HIGH}, // Reverse Step 3
        {LOW, HIGH, HIGH, LOW}, // Reverse Step 2
        {HIGH, LOW, HIGH, LOW} // Reverse Step 1
    };

    for (int i = 0; i < steps/4; i++) {
        for (int j = 0; j < 4; j++) {
            if (moveForward) {

```

```
        digitalWrite(motorPin1, stepSequence[j][0]);
        digitalWrite(motorPin2, stepSequence[j][1]);
        digitalWrite(motorPin3, stepSequence[j][2]);
        digitalWrite(motorPin4, stepSequence[j][3]);
    } else {
        digitalWrite(motorPin1, stepSequenceBackward[j][0]);
        digitalWrite(motorPin2, stepSequenceBackward[j][1]);
        digitalWrite(motorPin3, stepSequenceBackward[j][2]);
        digitalWrite(motorPin4, stepSequenceBackward[j][3]);
    }
    delay(stepDelay); // Fine-tuned delay for smoother operation
}
}
```

Liite 8 Ruuvín ohjelma

1(3)

```

// Define motor control pins (L298N Driver)
const int motorPin1 = 8; // IN1
const int motorPin2 = 10; // IN2
const int motorPin3 = 9; // IN3
const int motorPin4 = 11; // IN4

// Define input pins
const int potentiometerPin = A0; // Potentiometer
const int controlPin = 4; // Trigger motor movement
const int autoModePin = 2; // Enable auto mode
const int manualModePin = 3; // Enable manual mode
const int buttonPin = 6; // Manual step button
const int clockwisePin = 5; // Clockwise direction
const int counterclockwisePin = 7; // Counterclockwise direction

// Variables
int potentiometerValue = 0;
int stepsToMove = 0;
bool isAutoMode = false;
bool lastButtonState = HIGH;
bool motorEnabled = false;
int stepDelay = 5; // Fine-tuned delay for smoother movement
const int stepsPerTurn = 200; // Full stepper motor turn

void disableMotor() {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    motorEnabled = false;
}

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);
    pinMode(controlPin, INPUT);
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(clockwisePin, INPUT_PULLUP);
    pinMode(counterclockwisePin, INPUT_PULLUP);
    pinMode(autoModePin, INPUT_PULLUP);
    pinMode(manualModePin, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    if (digitalRead(autoModePin) == HIGH) {
        isAutoMode = true;
        autoMode();
    } else if (digitalRead(manualModePin) == HIGH) {
        isAutoMode = false;
        manualMode();
    } else {

```

```

    disableMotor(); // Ensure motor is off when no mode is active
}
}

void autoMode() {
    potentiometerValue = analogRead(potentiometerPin);
    stepsToMove = map(potentiometerValue, 0, (1023 * 4) / 5, stepsPerTurn, stepsPerTurn * 10); // Scale for 1-10 turns
    stepsToMove = (stepsToMove / stepsPerTurn) * stepsPerTurn; // Round to nearest full turn

    bool moveForward = digitalRead(clockwisePin) == HIGH;
    if (digitalRead(counterclockwisePin) == HIGH) moveForward = false; // Override if counterclockwise is pressed

    if (digitalRead(controlPin) == HIGH) {
        moveStepper(stepsToMove, moveForward);
        delay(500);
    } else {
        disableMotor(); // Turn off motor when not in use
    }
}

void manualMode() {
    bool currentButtonState = digitalRead(buttonPin);
    bool moveForward = digitalRead(clockwisePin) == HIGH;
    if (digitalRead(counterclockwisePin) == HIGH) moveForward = false;

    if (lastButtonState == HIGH && currentButtonState == LOW) { // Detect button press
        Serial.println("Manual button pressed, moving one turn");
        moveStepper(stepsPerTurn, moveForward); // Move one full turn
        delay(200); // Debounce delay
    }
    lastButtonState = currentButtonState;
}

void moveStepper(int steps, bool moveForward) {
    int stepSequence[4][4] = {
        {HIGH, LOW, HIGH, LOW}, // Step 1
        {LOW, HIGH, HIGH, LOW}, // Step 2
        {LOW, HIGH, LOW, HIGH}, // Step 3
        {HIGH, LOW, LOW, HIGH}  // Step 4
    };

    int stepSequenceBackward[4][4] = {
        {HIGH, LOW, LOW, HIGH}, // Reverse Step 4
        {LOW, HIGH, LOW, HIGH}, // Reverse Step 3
        {LOW, HIGH, HIGH, LOW}, // Reverse Step 2
        {HIGH, LOW, HIGH, LOW}  // Reverse Step 1
    };

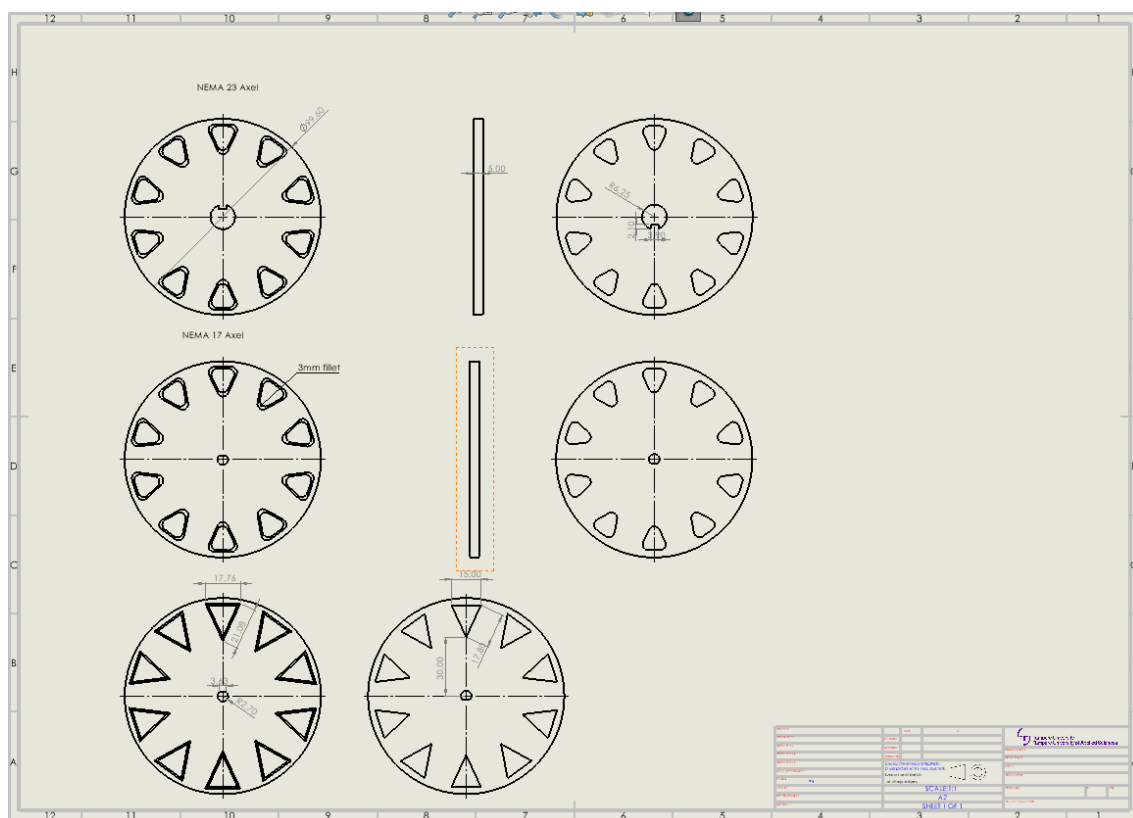
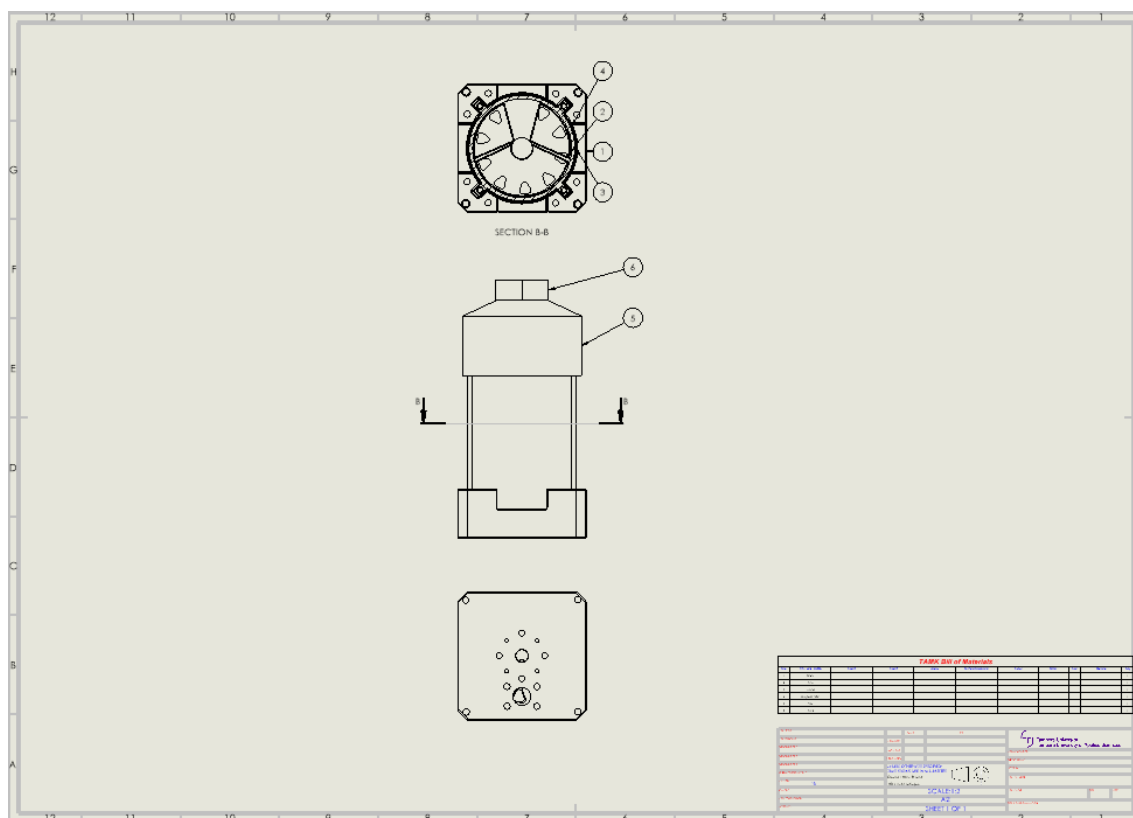
    for (int i = 0; i < steps / 4; i++) {
        for (int j = 0; j < 4; j++) {
            if (moveForward) {

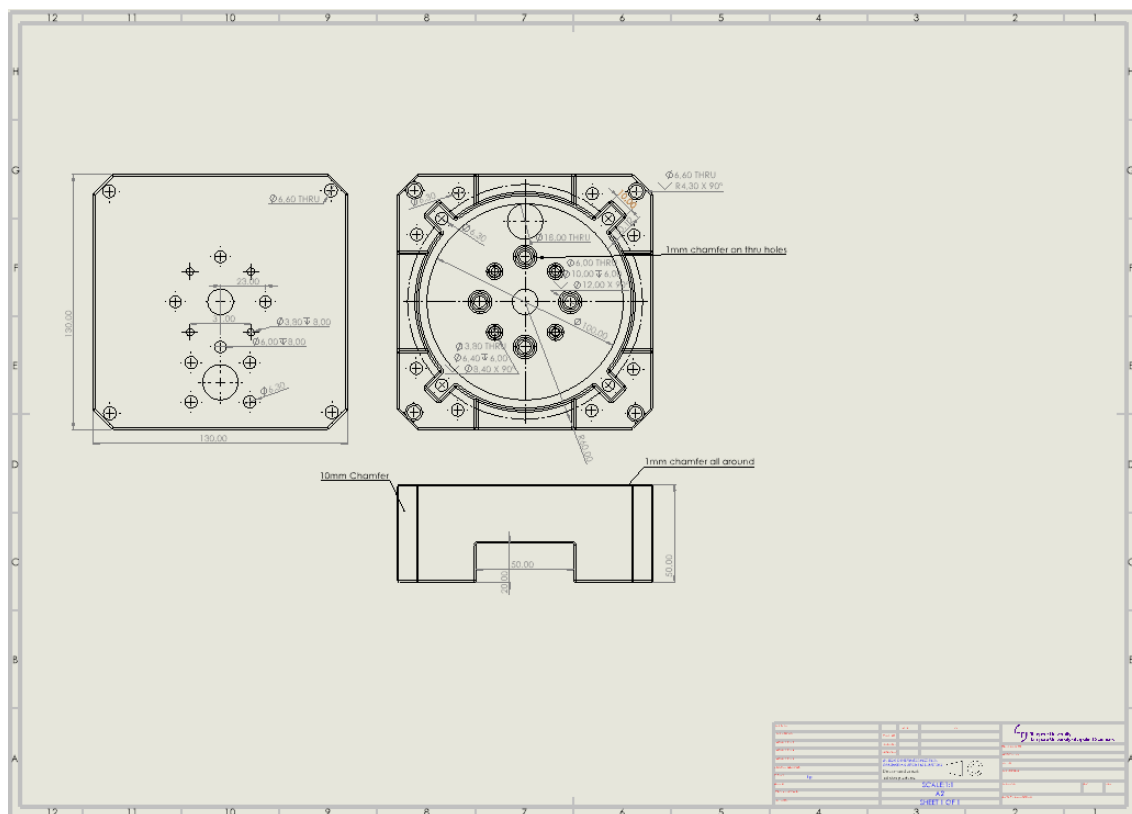
```

```
        digitalWrite(motorPin1, stepSequence[j][0]);
        digitalWrite(motorPin2, stepSequence[j][1]);
        digitalWrite(motorPin3, stepSequence[j][2]);
        digitalWrite(motorPin4, stepSequence[j][3]);
    } else {
        digitalWrite(motorPin1, stepSequenceBackward[j][0]);
        digitalWrite(motorPin2, stepSequenceBackward[j][1]);
        digitalWrite(motorPin3, stepSequenceBackward[j][2]);
        digitalWrite(motorPin4, stepSequenceBackward[j][3]);
    }
    delay(stepDelay); // Fine-tuned delay for smoother operation
}
}
```

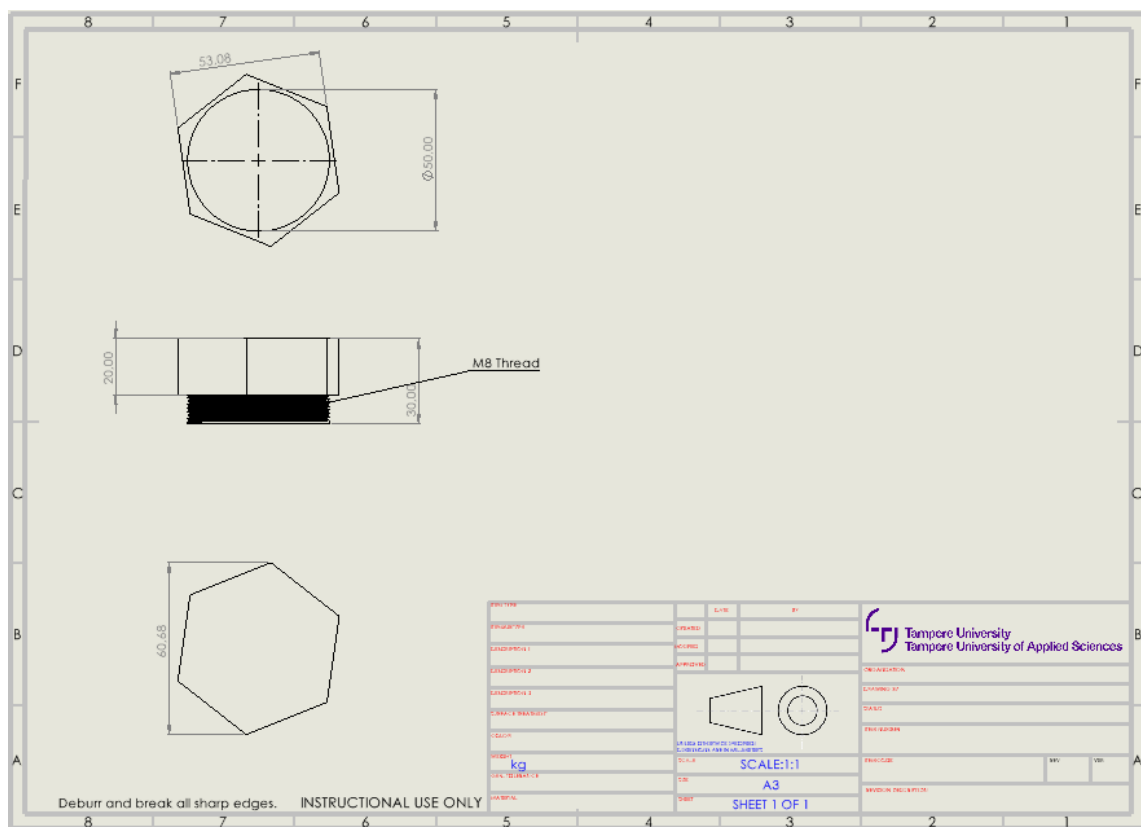
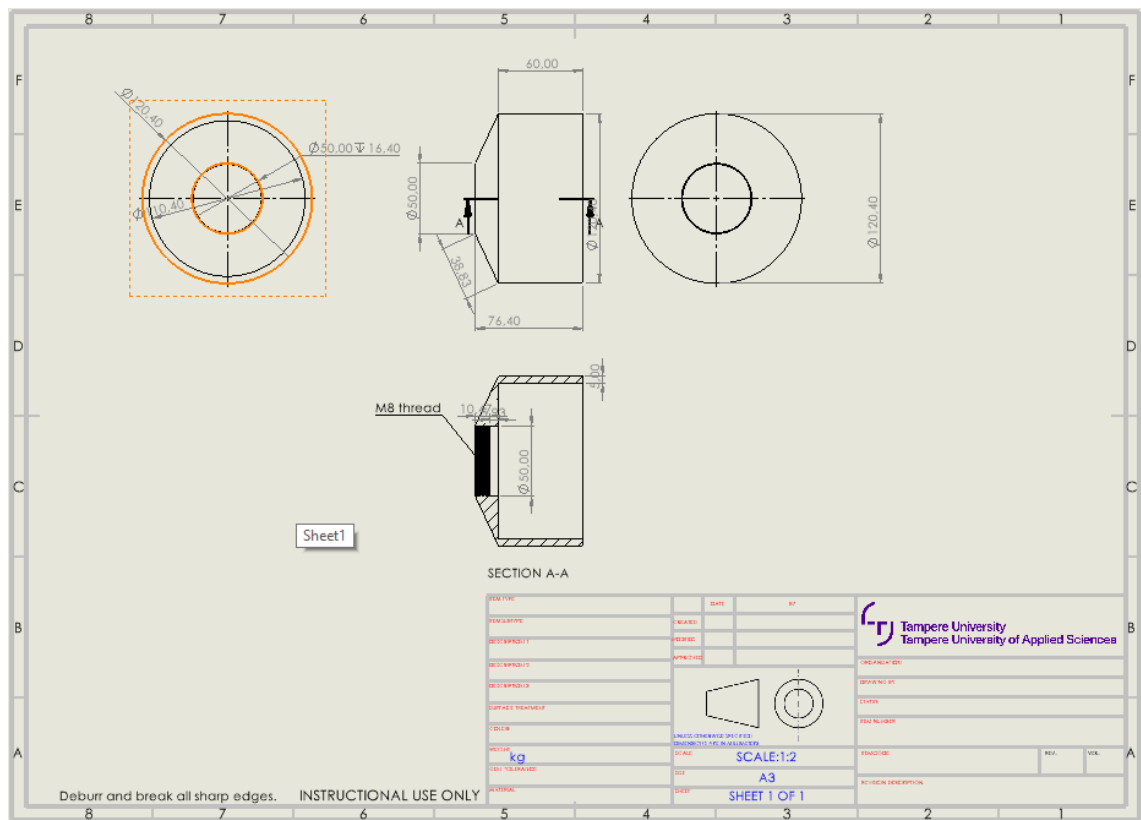
Liite 9 Annostelulevyn piirustukset

1(3)



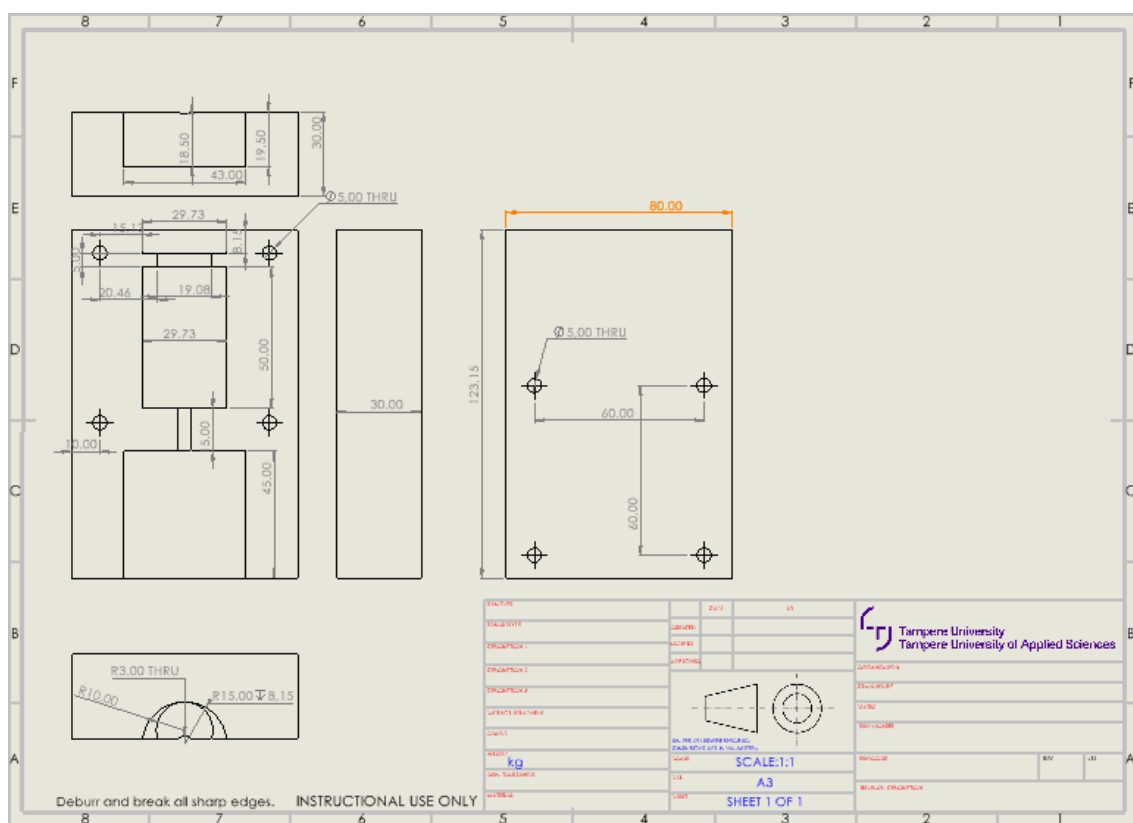
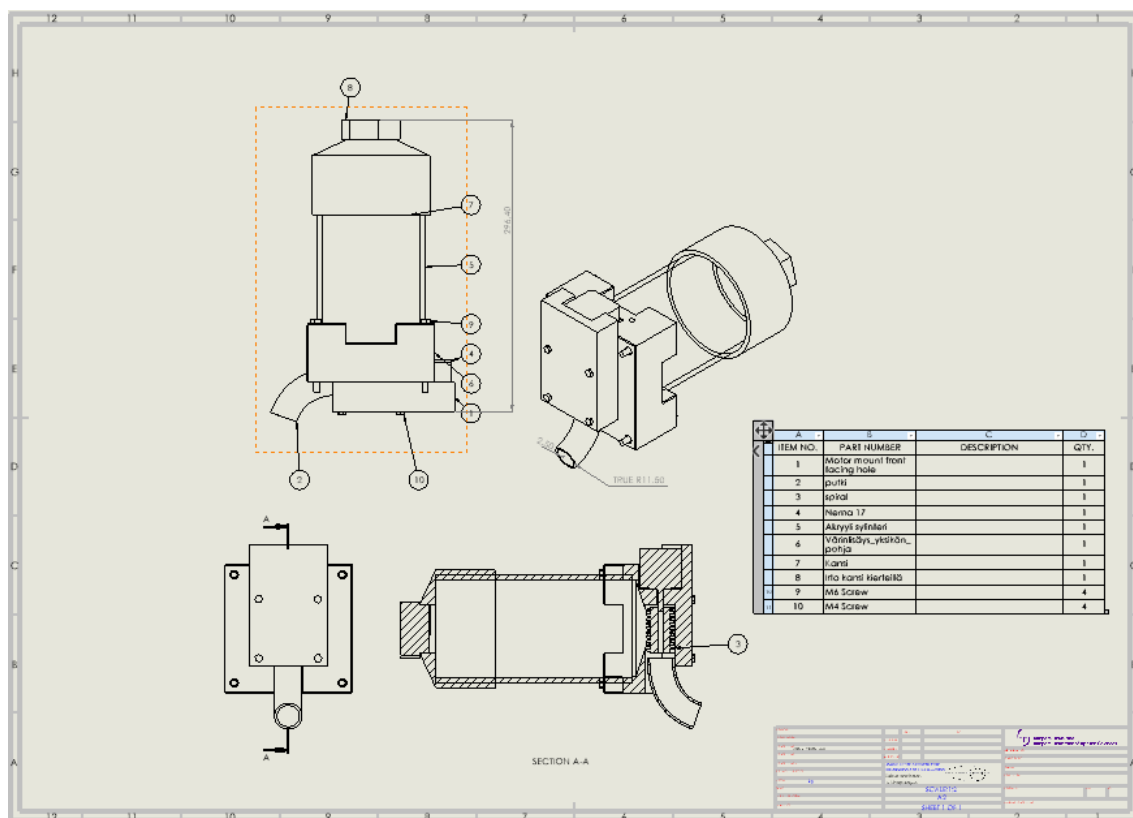


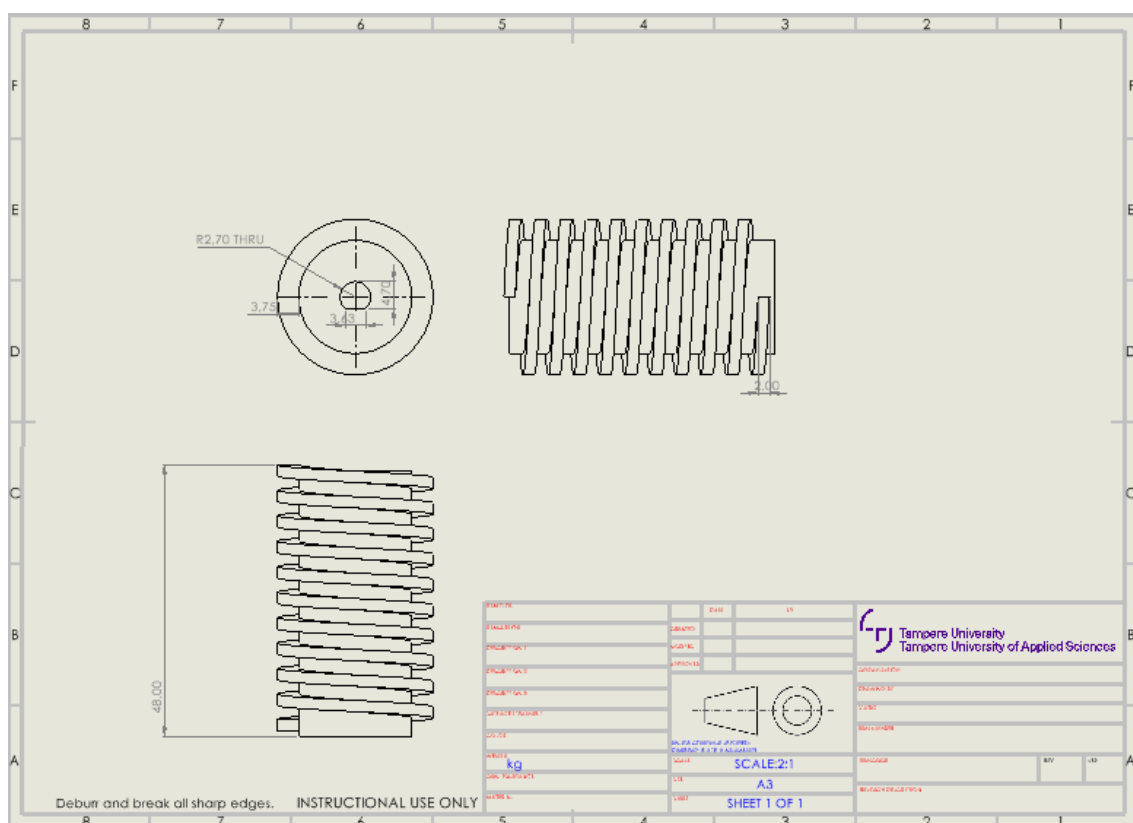
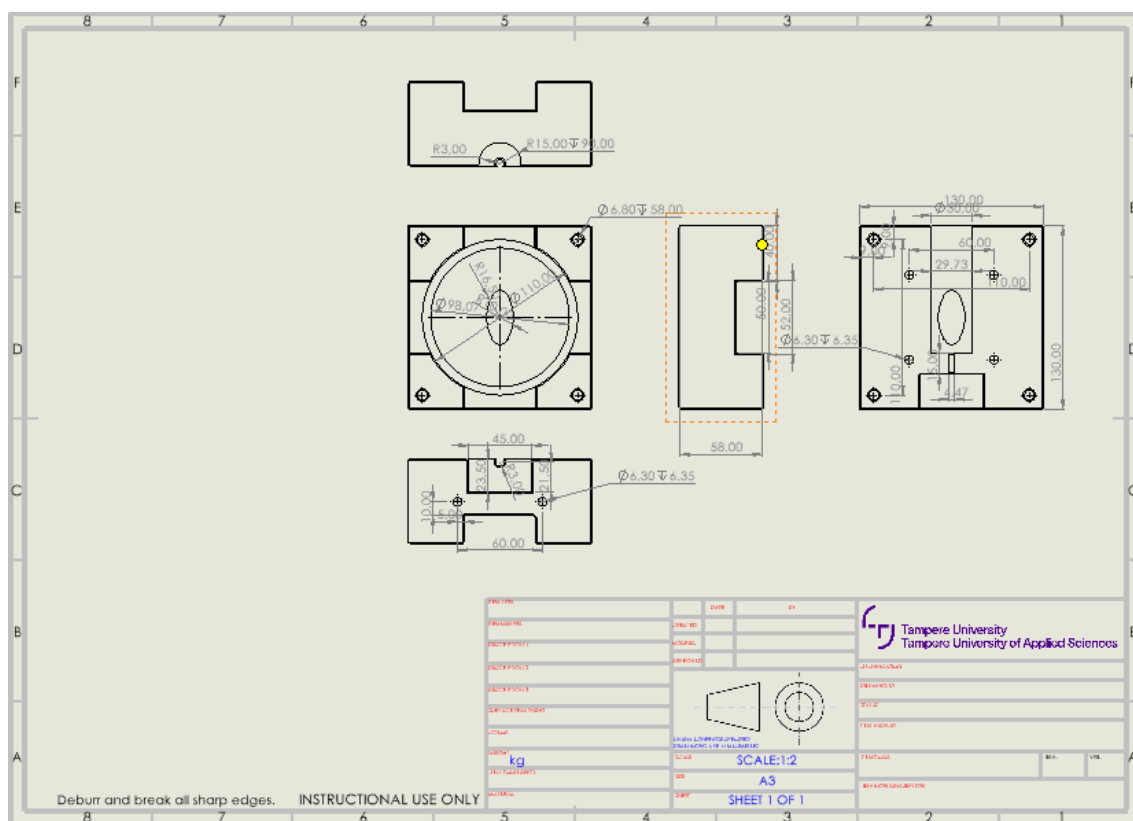
3(3)



Liite 10 Annosteluruuvien piirustukset

1(2)





Liite 11 Arvioita värinlisääjäyksikön komponenttien kustannuksista

	Price	Date	More detales	Source
L298N	8,17 €	29.3		https://www.partco.fi/fi/saehkoemeka-niikka/moottorit/moottoriohjaimet/25737-mot-mdl298n.html
Arduino UNO R3	30,12 €	29.3		https://store.arduino.cc/en-fi/products/arduino-uno-rev3?_gl=1*_nzmom5*_up*MQ..*_ga*Nzly-MzcwMjcxljE3NDMyMzg2MjM.*_ga_NEXN8H46L5*MTc0MzIzO-DYyMS4xLjEuMTc0MzIzO-DYzMi4wLjAuMTY1NzI1MzY4Mg..
Resistors and diodes	3,40 €	29.3	approx. each component costs cents	
switches	2,00 €	29.3	approx. each component costs cents	
ABS	4,03 €	29.3	130g of material used 31€/kg	https://www.3djake.com/3djake/niceabs-black
Powersuply 12V or 24V	25,00 €	29.3	Price may vary based on where they are bought	
PS1*00 preasure switch		29.3	Pricing not avilabe	https://www.smc.eu/fi-fi/products/ps1-00-electronic-pressure-switch~31443~cfg?partNumber=PS1000-R06L-Q
Total	72,72 €			