



Henri Pakanen

PIKAMALLINNUSTYÖKALU NC-TYÖSTÖKESKUKSEEN

PIKAMALLINNUSTYÖKALU NC-TYÖSTÖKESKUKSEEN

Henri Pakanen
Opinnäytetyö
Kevät 2013
Kone- ja tuotantotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Kone- ja tuotantotekniikka, koneautomaatio

Tekijä: Henri Pakanen

Opinnäytetyön nimi: Pikamallinnustyökalu NC-työstökeskukseen

Työn ohjaaja: Jari Viitala

Työn valmistumislukukausi ja -vuosi: kevät 2013 Sivumäärä: 57 + 10 liitettä

Opinnäytetyössä suunniteltiin Fanuc-ohjauksella toimivaan NC-työstökeskukseen pikamallinnustyökalu ja testattiin sen toimivuutta. Työ tehtiin Pireplan Oy:lle Kurikkaan. Työn tavoitteena oli tutkia pikamallinnuksen toimivuutta ja mahdollisuuksia NC-ohjatussa työstökeskuksessa sekä suunnitella NC-työstökeskukseen soveltuva pikamallinnustyökalu, jolla voidaan valmistaa kappaleita suoraan 3D-mallista.

Työ oli tuotekehitystyö tyyppinen. Mekanismin suunnittelua ja toimivuutta varten työtä tehdessä perehdyttiin yleisimpiin pikamallinnustekniikoihin, edullisiin pikamallinnuslaitteisiin, avoimeen koodiin perustuvan Arduino-mikrokontrollerin ohjelmointiin ja G-koodilla ohjattuihin NC-koneisiin sekä G-koodiin. Aihepiiriin perehtymisen jälkeen laite päädyttiin toteuttamaan FDM-pursotustekniikalla sen yksinkertaisuuden takia. Pikamallin valmistukseen tarvittavien G-koodien generointiin käytettiin MakerBotin vapaata koodia hyödyntävää ReplicatorG-ohjelmaa. Jotta NC-työstökeskus ymmärtää ReplicatorG-ohjelman muodostamaa G-koodia, on koodiin tehtävä pieniä muutoksia, minkä vuoksi tehtiin myös kääntöohjelma PC-laitteelle. Itse pursotuspäätä ohjattiin Arduino-mikrokontrollerilla ja siihen tehtiin oma ohjausohjelmansa.

Laitteen toimintaa testattaessa valmistettiin prototyyppi, johon kuului mekaniikkasuunnittelu pursotuspään kiinnityksestä koneistuskeskukseen, tarvittavan ohjauselektronikan kasaus ja pursotuspään sekä Robodrillin hallintaan tarvittavat ohjelmat. Prototyypin kiinnitysmekanismi valmistus ja prototyypin testaus suoritettiin OAMK:n tiloissa Robodrill α -T10B-koneistuskeskuksella.

Työn tuloksena syntyi prototyyppi NC-työstökoneeseen mekaanisesti liitettävästä pursotuspäästä, joka saatiin toimimaan kohtalaisesti työstökoneessa. Pursotuspää ja työstökeskuksen tasot saatiin liikkumaan synkronoidusti ja toivotulla tavalla, mutta parametrien säädöt vaatisivat vielä enemmän pohdintaa. Suunnittelun tuloksena syntyivät myös 3D-mallit pursotuspään kiinnitysrakenteesta, tekniset piirustukset, sähkökaavio elektronikasta sekä ohjelmakoodit. Vaikkei pikamallinnustyökalu toiminut aivan asetettujen tavoitteiden mukaisesti, on se kuitenkin hyvä lähtökohta vastaavien toimintojen ja laitteiden suunnitteluun NC-laitteisiin.

Asiasanat: NC, pikamallinnus, FDM, Arduino, numeerinen ohjaus

ALKULAUSE

Tämän insinööriyön tilaajana toimi Pireplan Oy, jonka toimipaikka sijaitsee Kurikassa. Oppilaitoksen ohjaavana opettajana toimi lehtori Jari Viitala. Haluaisin kiittää edellä mainittuja, sekä erityisesti kiittää yliopettaja Eero Korhosta mielenkiinnosta opinnäytetyötäni kohtaan.

Oulussa 14.5.2013

Henri Pakanen

SISÄLLYS

TIIVISTELMÄ	3
SISÄLLYS	5
MERKKIEN SELITYKSET	7
1 JOHDANTO	8
2 PIKAMALLINNUSTEKNIikka (RAPID PROTOTYPING)	9
2.1 Yleistä	9
2.2 Pikamallinnuksen periaate	9
2.3 Pikamallinnustekniikat	11
2.4 FDM (Fused Deposition Modelling)	11
3 NC-TYÖSTÖKONEET	13
3.1 NC-koneiden koodi	13
3.1.1 Valmistelevat G-koodit	14
3.1.2 Kytkevät M-koodit	14
3.2 Robodrill	14
4 MAKERBOT JA ARDUINO	15
4.1 Avoin koodi	15
4.2 Tuotteet	15
4.3 Stepstruder-pursotuspää	18
4.4 Arduino	18
5 MEKANIikkaVAALINTOJEN JA KOMPONENTTIEN HAHMOTUS	20
5.1 Pursotuspää	20
5.2 Pursotuspään rakenteen suunnittelu	20
5.3 Pursotuspään ohjaus	22
5.4 Robodrillin XYZ-tasojen ohjaus	23
5.5 Kättely tietokoneen ja muiden laitteiden välillä	23
6 PURSOTUSPÄÄN SOVITUS ROBODRILLIIN	25
6.1 Laitekaavio	25
6.2 Elektroniikka	26
6.2.1 Elektroniikan komponentit	27
6.2.2 Lopullinen elektroniikka	28
6.3 Pursotuspään kiinnitys robodrilliin	32

7 ROBODRILLIN JA PURSOTUSPÄÄN OHJAAMINEN	35
7.1 Pursotuspään ohjelmointi	35
7.2 PC-laitteen ohjelma	41
7.2.1 STL-tiedostosta G-koodiin	42
7.2.2 PC-laitteen ohjelman toiminta	45
8 LOPPUTULOKSET JA PROTOTYYPIN TESTAUS	47
8.1 Prototyypin testaus	47
8.2 Prototyypin testauksen tulokset	49
8.3 Pursotinpäähän kiinnityksen mekaniikkasuunnittelu	50
9 YHTEENVETO	53
LÄHTEET	55
LIITTEET	57

MERKKIEN SELITYKSET

ABS	Akryylinitriilibutadienistyreeni, muovityyppi
CAD	Computer Aided Design, tietokoneavusteinen suunnittelu
CAM	Computer Aided Manufacturing, tietokoneavusteinen valmistus
DMLS	Direct Metal Laser Sintering, pikamallinnustekniikka
FDM	Fused Deposition Modeling, pikamallinnustekniikka
HDPE	Korkeatiheyksinen polyeteeni, muovityyppi
MJM	Multijet Modeling Systems, 3D-tulostustekniikka
MOSFET	Metal-oxide-semiconductor field effect transistor, tehotransistori
PC	Polykarbonaatti, muovityyppi
PLA	Polylaktidi, muovityyppi
PVA	Polyvinyyli, muovityyppi
RP	Rapid production, pikamallinnus
SLS	Selective Laser Sintering, RP-tekniikka

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on suunnitella ja toteuttaa uudentyyppinen työkalu NC-työstökeskukseen. Työkalun avulla numeerisesti ohjattua työstökoneetta voitaisiin käyttää myös pikamallien valmistamiseen. Vaikka pikamallinruslaitteet ovat vuosien varrella halventuneet ja markkinoille on ilmestynyt paljon edullisen hintaluokan pikamallinruslaitteita, ovat varteen otettavat vaihtoehdot oikeaan valmistamiseen vieläkin useiden tuhansien eurojen hintaluokassa. Työn tilaaja Pireplan Oy halusikin sovelluksen, jolla vanhasta Fanuc-ohjausta hyödyntävästä puuntyöstökeskuksesta saataisiin erillisellä työkalupäällä myös pikamallinnukseen soveltuva laite. Alkuperäisenä ideana on käyttää Makerbotin ilmaista ReplicatorG-ohjelmaa 3D-mallin NC-koneistuskeskukselle sopivaksi G-koodiksi muuntamiseen.

Pikamallinnuksen käyttö on viime vuosina kasvanut nopeasti, ja se on tuotu myös yksityisten henkilöiden saataville edullisempien ja kotiin sopivien laitteiden ansiosta. Pikamallinnuksessa tietokoneella luodusta 3D-mallista voidaan valmistaa kolmiulotteinen kappale, joka vastaa tietokoneella luotua 3D-mallia. Se on nopeuttanut tuotekehitysprosessia huomattavasti verrattuna perinteisiin menetelmiin, mutta sillä voidaan myös valmistaa käyttövalmiita kappaleita, jos vaatimukset eivät ole liian suuret.

Työn tavoitteena on ideoida NC-työstökeskukseen sopiva pikamallinnustyökalu, suunnitella sen kiinnitysmekanismi, toteuttaa pikamallinnustyökalun ohjaus, muokata ReplicatorG-koodin NC-työstökeskukseen sopivaksi ja tehdä laitteelle sopiva ohjauselektronikka. Laitteesta tehdään prototyyppi, jota voidaan testata Oulun ammattikorkeakoulun tiloissa Robodrill-koneistuskeskuksella. Opinnäytetyötä koskien on myös tehty lähtötietomuistio, joka on liitteenä 1.

2 PIKAMALLINNUSTEKNIikka (RAPID PROTOTYPING)

Pikamallinnustekniikka on kasvanut viime vuosina nopeasti, ja siitä on tullut tärkeä osa valmistettavien tuotteiden tuotekehitysprosessia. Yksi tuotekehityksen kulmakivistä on prototyyppien suunnittelu, jotta mahdolliset virheet, muotoilu ja mekaaninen toimivuus saadaan selvitettyä. (1, s. 1.)

Valmistettavien tuotteiden tuotekehitysprosessiin kuuluva aika on noussut elintärkeään asemaan kilpailun koventuessa (1, s. 1). Fyysisen prototyypin tekeminen on perinteisin keinoin usein aikaa vievä prosessi, minkä vuoksi pikamallinnustekniikasta on tullut nopeasti yksi suosituimmista menetelmistä prototyyppien valmistukseen. (2, s. 16 - 17.)

2.1 Yleistä

Pikamallinnustekniikka on suunnittelun, prototyyppien ja valmistamisen työkalu, jolla voidaan valmistaa tietokoneella muodostetusta 3D-mallista vastaava fyysinen kappale. Sillä voidaan valmistaa kappaleita ilman koneistamista, muotteja tai valamista, jolloin se on yksi parhaista aputyökaluista suunnittelun ja tuotekehityksen prototyyppien valmistamiseen. Asianmukaisilla jälkikäsitteilyillä ja tarkoilla koneilla jopa toiminnallisten kappaleiden valmistaminen on mahdollista. Kappale voi sisältää monimutkaisia geometrisia muotoja, joita perinteisillä valmistusmenetelmillä ei pystyttäisi tekemään. (1, s. 13.)

Suurin pikamallinnustekniikan tuoma etu on valmistamisen helppous. Kun suunnitteludata on saatavilla, voidaan vastaava fyysinen kappale luoda automaattisesti pikamallinnuslaitteistolla, jolloin vältetään käsityöltä sekä säästetään huomattavasti aikaa perinteisiin valmistusmenetelmiin verrattuna. (1, s. 13.)

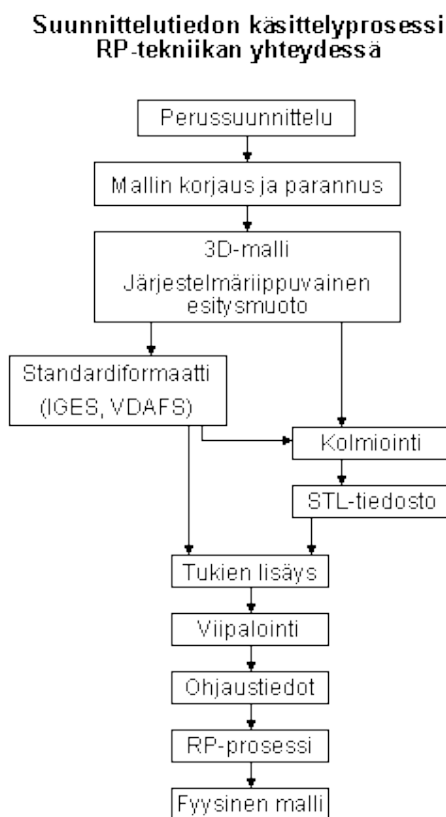
2.2 Pikamallinnuksen periaate

Huolimatta pikamallinnustekniikoiden eroavaisuudesta toisiinsa, ne kaikki käyttävät samaa peruskaavaa, jonka perusteella pikamalli luodaan (1, s. 11). Ensiksi malli tai komponentti suunnitellaan käyttäen CAD-CAM-ohjelmistoa. Mallin tulee olla pinnaltaan suljettu, jotta sitä voidaan hyödyntää pikamallinnuksessa. 3D-CAD-suunnittelun tuloksena syntynyt CAD-

järjestelmän sisäisessä muodossa oleva malli ei kuitenkaan ole yhteensopiva pikamallinnuslaitteiston kanssa sellaisenaan, vaan vaatii muokkausta ollakseen yhteensopiva. (3.)

Jotta tiedosto saataisiin yhteensopivaksi pikamallinnuslaitteiston kanssa, on se yleensä käännettävä STL-tiedostoksi. STL-tiedostossa 3D-mallin pinta jaetaan kolmioiksi, jotka määrätyn toleranssin sisällä kuvaavat kappaleen pintaa. (1, s. 12.)

STL-tiedoston tekemisen jälkeen STL-tiedosto viipaloidaan ja muutetaan G-koodikäskyiksi tarkoitukseen soveltuvalla ohjelmalla, esimerkiksi MakerBotin ReplicatorG-ohjelmalla. Ohjelma viipaloi tiedoston poikkileikkauksiksi, joiden paksuus riippuu pikamallinnuslaitteiston tulostettavan kerroksen paksuudesta. Malli rakennetaan näistä kaksiulotteisista poikkileikkauksista viipale kerrallaan. Malli muodostuu, kun jokainen poikkileikkaus kovetetaan tai rakennetaan menetelmästä riippuen edellisen päälle (1, s. 12). Kuvasta 1 nähdään kaaviona kappaleen edistymisen suunnittelusta valmiiksi kappaleeksi.



KUVA 1. RP-tekniikalla valmistettavan kappaleen käsittelyprosessi (3)

2.3 Pikamallinnustekniikat

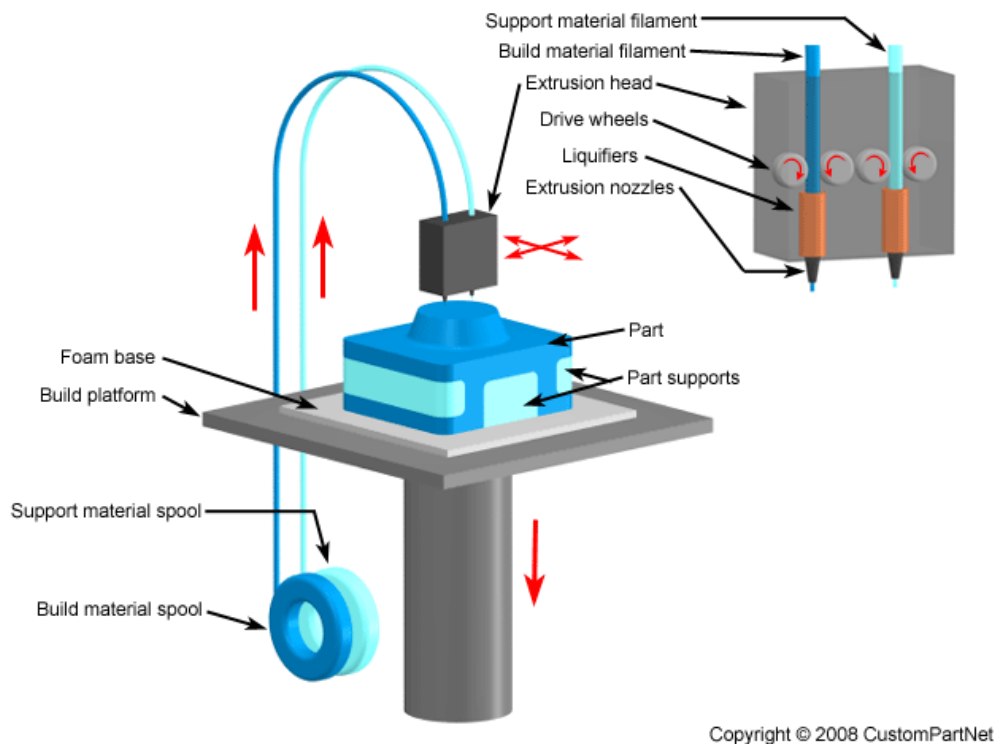
Pikavalmistuskoneiden yleistyessä myös erilaisia pikavalmistustekniikoita on syntynyt useita. Vaikka lähes joka valmistajan koneissa on joitain eroja toisiinsa nähden, voidaan tekniikat jakaa karkeasti pääryhmiin:

- sulaa materiaalia pursottaviin (FDM, MJM)
- kerroksia päällekkäin leikkaaviin ja laminoiviin (LOM)
- stereolitografisiin, valokovettuvaa hartsia kerros kerrokselta laserilla/UV-valolla kovettaviin (stereolitografia, MLS)
- muovi-, kipsi - tai metallipulveria sintraavat (SLS DMLS). (4, s. 24 - 55.)

Koska opinnäytetyö toteutetaan FDM-tekniikkaa hyväksi käyttäen, muita tekniikoita ei tässä opinnäytetyössä käsitellä.

2.4 FDM (Fused Deposition Modelling)

FDM-valmistusmenetelmässä malli aikaansaadaan pursottamalla sulakammiossa puolijuoksevaksi lämmitettyä termoplastista, eli lämpömuovattavaa materiaalia suuttimen kautta kerroksittain työtasolle. Materiaali jäähtyy nopeasti, jolloin se kiinnittyy edelliseen kerrokseen. Pursotuspää liikkuu X-Y-tasossa määrättyllä nopeudella pursottaen materiaalia kerroksen ääriviivojen mukaisesti. Z-tason suuntainen liike tehdään liikuttamalla rakennustason korkeutta suuttimeen verrattuna. Jokaisen kerroksen valmistuessa rakennustaso laskeutuu kerrospaksuuden verran alaspäin. (4, s. 35 - 36.) Osassa FDM-tekniikalla toimivista laitteista rakennustaso liikkuu X-Y-tasossa ja pursotinpää Z-tasossa, mutta toimintaperiaate on kuitenkin täysin sama. (Kuva 2.)



KUVA 2. FDM-tekniikan toimintaperiaate (5)

Materiaalina FDM-laitteissa käytetään yleisimmin ABS- ja PC-muoveja sekä näiden sekoitusta PC-ABC-muovia, mutta myös PLA-muovi on lisännyt suosiotaan FDM-laitteiden materiaalina. Yleisesti kalliimmista järjestelmissä on kaksikätkäinen pursotusjärjestelmä, jossa primääristä mallimateriaalia käytetään mallin geometrian luomiseen ja sekundaarista materiaalia käytetään mallin tukirakenteisiin. Tukimateriaali on yleensä vesiliukoista tai muuten helposti poistettavissa. Halvemmissa laitteistoissa pursotuspäitä on vain yksi, jolloin sekä malligeometria sekä tukirakenteet muodostuvat samasta materiaalista. Tässä tapauksessa tukirakenne on poistettava manuaalisesti tai koneellisesti. Ontot ja tyhjän päälle muodostuvat kohdat mallista tarvitsevat yleensä tukirakenteen. (1, s.142 - 143.)

3 NC-TYÖSTÖKONEET

NC-lyhenne tulee sanoista *numerical control* ja merkitsee numeerisesti ohjattua. Työstökoneetta voidaan pitää numeerisesti ohjattuna, kun se suorittaa kappaleen koneistuksessa tarvittavat liikkeensä automaattisesti. Numeerisella ohjauksella saadaan siirrettyä koneistajan vaativat tarkkailu- ja laskentatyöt sekä ruumiilliset työt, kuten työkalujen vaihtaminen, automaatiolaitteiden tehtäviksi. (6, s. 7 - 8.)

3.1 NC-koneiden koodi

NC-koneita ohjataan yleensä G-koodilla. G-koodissa eri *NC-osoitteilla* eli alkukirjaimilla on eri merkitykset. Esimerkiksi O merkitsee ohjelman numeroa ja F syöttönopeutta. NC-koodit ovat kansainvälisesti standardoituja, joten NC-osoitteet ovat koneissa yleensä vakiot. (6, s. 81.)

NC-osoitteista muodostuu *NC-sanoja*, kun niihin lisätään tarkentava numeroarvo. NC-ohjelman yhtä riviä kutsutaan *NC-lauseeksi*. Kuvassa 3 on esiteltyä G-koodilla muodostettu täydellinen ohjelma selityksineen. (6, s. 81.) Tärkeimmät G-koodit voidaan jakaa karkeasti kahteen tyyppiin, valmisteleviin G-koodeihin sekä kytkeviin M-koodeihin. (6, s. 81.)

```
o1120 - ohjelmanumero
G00 T0101 - valitaan työkalu 1 (rouhintaterä)
G96 S120 M3 - valitaan vakioastuamisnopeus 120 m/min ja käynnistetään kara
M8 - lastuamisnestepumppu päälle
G00 Z0. - liikutaan Z-akselilla nolapisteeseen (pikaliikkeellä)
G00 X41. - liikutaan X-akselilla halkaisijaan 41 mm (pikaliikkeellä)
G01 X-0.5 F0.3 - liikutaan X-akselilla halkaisijaan -0,5 mm (syöttöliike, syöttö 0,3 mm/r)
G01 X41. Z1. - liikutaan X-akselilla halkaisijaan 41 mm ja Z-akselilla 1 mm irti
G71 U3. - rouhintatyökierto, lastun vahvuus 3 mm
N10 G00 X20. - työkierron liikerataa
G01 Z-20. - työkierron liikerataa
N11 G01 X41. - työkierron liikerataa
G28 U0.W0. M9 - terän vienti koneen X-referenssi pisteeseen ja lastuamisneste pois päältä
G00 T0303 - valitaan työkalu 3 (silitysterä)
G96 S150 M3 - valitaan vakioastuamisnopeus 150 m/min ja käynnistetään kara
G00 Z1. M8 - siirrytään 1 mm irti kappaleesta (pikaliike) ja lastuamisneste päälle
G00 X41. - siirrytään halkaisijaan 41 mm (pikaliike)
G70 P10 Q11 S150 F0.15 - silitystyökierto, alkaa riviltä 10 ja loppuu riville 11
G28 U0. M9 - terän vienti koneen X 0 pisteeseen ja lastuamisneste pois päältä
G28 W0. - terän vienti koneen Z0 pisteeseen
M30 - ohjelman lopetus, ja paluu ohjelman alkuun
```

KUVA 3. G-koodi selityksineen (7)

3.1.1 Valmistelevat G-koodit

Varsinaiset G-koodit ovat valmistelevia koodeja, ja ne luokitellaan niiden modaalisuuden perusteella. G-koodi on modaalinen, kun se on voimassa niin kauan, kunnes toinen samaan ryhmään kuuluva G-koodi annetaan, tai ei-modaalinen, jos se on voimassa vain niissä lauseissa, joissa se on annettu. (6, s. 81.)

3.1.2 Kytkevät M-koodit

M-koodit ovat kytkeviä koodeja, eli ne ovat tyypillisesti ON/OFF-tyyppisiä komentoja. Esimerkiksi karan käynnistys myötäpäivään voidaan tehdä M03-sanalla ja se vastapäivään M04-sanalla. M-koodit eroavat eri koneilla huomattavasti enemmän toisistaan kuin G-koodit, joten M-koodit kannattaa aina tarkistaa valmistajan ohjekirjasta. (6, s. 85.)

3.2 Robodrill

Robodrill α -T10B on Fanucin valmistama vertikaalinen NC-koneistuskeskus jossa on 10-paikkainen työkalukaruselli. Se käyttää Fanucin omaa ohjausta, joka on vanhemmissa työstökoneissa yleinen. (8, s. 1.) Myös työn tilaajana toimiva Pireplan Oy käyttää Fanucin ohjauksella toimivaa puuntyöstökeskusta. Koska Pireplan Oy:n toimitilat ovat toisella paikkakunnalla, käytetään Oulun seudun ammattikorkeakoulun laboratoriotiloista löytyvää Robodrill α -T10B -koneistuskeskusta prototyyppi- ja testausalustana. Fanucin ohjaus käyttää normaaleja G-koodeja ohjauksessaan.

Keskeinen syy Robodrillin käyttämiseen prototyyppitestauksessa on mahdollisuus remote- eli kaukoajo-tilaan, jolloin koneistuskeskusta voidaan ohjata RS-232-C-väylää käyttäen PC-laitteella. Koska Robodrillin oma sisäinen muisti on kohtalaisen lyhyt ja pikamallin valmistamiseen tarvittava koodi sisältää yleensä kymmeniä tuhansia rivejä koodia, on ulkoinen ohjaus ja koodin jakaminen osiin pakollista, jotta laite toimisi. (8, s. I-16.)

4 MAKERBOT JA ARDUINO

MakerBot Industries on New Yorkissa vuonna 2009 perustettu yritys, joka tuottaa 3D-pikamallikoneita. Yrityksen perustivat Bre Pettis, Adam Mayer ja Zach Smith. Smith oli myös yksi RepRap Research Foundationin perustajajäsenistä. RepRap Research Foundation on voittoa tavoittelematon ryhmä joka perustettiin kehittämään varhaista tutkimusta avoimella koodilla toimivissa 3D-tulostimia. MakerBot Industries käyttää hyväkseen tätä tutkimustyötä ja yrityksen tavoitteena on tehdä avoimeen koodiin perustuvia 3D-tulostimia kotikäyttöön kohtuullisella hinnalla. (9.)

4.1 Avoin koodi

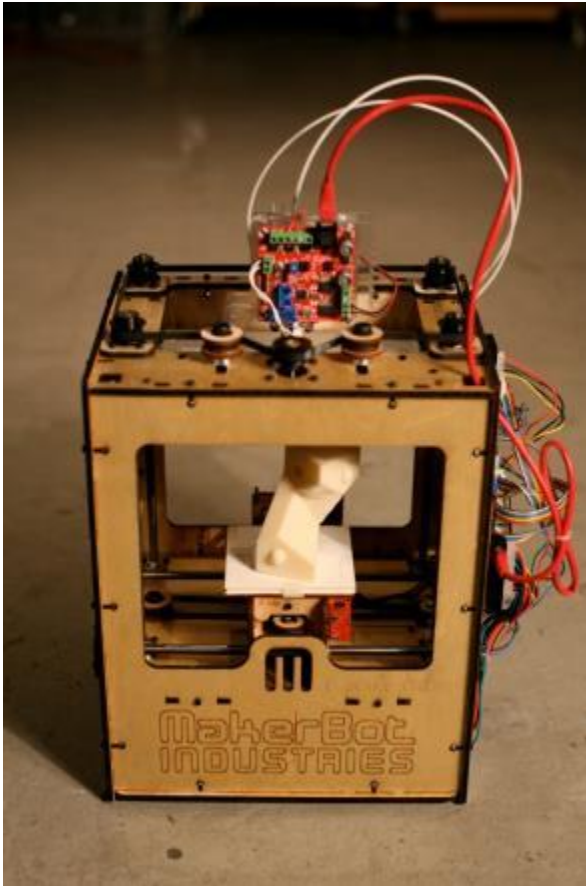
Avoimella koodilla tarkoitetaan sitä, että jokaisella käyttäjällä on mahdollisuus tutustua ohjelman tuottamaan lähdekoodiin ja muokata sitä omiin tarpeisiinsa sopivaksi. Kaikilla ohjelman käyttäjillä on samat oikeudet sekä oikeus jakaa myös muokattua tai muokkaamatonta ohjelman koodia eteenpäin. (10.) Suurin osa MakerBotin kehitystyöstä perustuukin aktiivisiin harrastajiin, jotka kehittävät ohjelmaa eteenpäin.

4.2 Tuotteet

MakerBotin tuotteet toimitetaan yleisesti rakennussarjoina. Tuotteet on suunniteltu niin, että niiden kasaaminen onnistuu keneltä vain, joka hallitsee perusteet teknisistä töistä. Nykyään MakerBot toimittaa myös valmiiksi kasattuja tuotteita, mutta rakennussarjat ovat hinnaltaan edullisempia. Yrityksen nykyisillä tuotteilla voidaan tulostaa ABS-, HDPE-, PLA- sekä PVA-muoveja. (9.)

Cupcake CNC oli MakerBotin ensimmäinen 3D-tulostin. Sen lähdetiedostot olivat ladattavissa internetistä, jolloin kuka vain pystyi rakentamaan omansa alusta alkaen. Tasojen säätö on toteutettu hihnavedolla ja laakeroiduilla kierretangoilla. X- ja Y-tasot liikkuvat rakennustasossa ja pursotinpää liikkuu Z-tasossa. Juuri vapaasta jakamisesta ja avoimesta koodista johtuen suurin osa tuotekehityspalautteesta sekä ideoista saatiin suoraan käyttäjiltä. Tuotannossa

ollessaan Cupcake CNC:stä tehtiin useita eriä joissa oli pieniä parannuksia edellisiin verrattuna. Kuvassa 4 on nähtävissä Cupcake CNC. (9.)



KUVA 4. Cupcake CNC -pikamallinusslaite (9)

MakerBotin seuraava tuote oli Thing-O-Matic. Siinä oli useita parannuksia verrattuna aikaisempaan Cupcake CNC:hen, kuten lämmitetty pursotusalusta, uudestaan suunniteltu Z-taso ja päivitetty elektroniikka. Kuvassa 5 voidaan nähdä Thing-O-Matic-pikamallinusslaite. (9.)

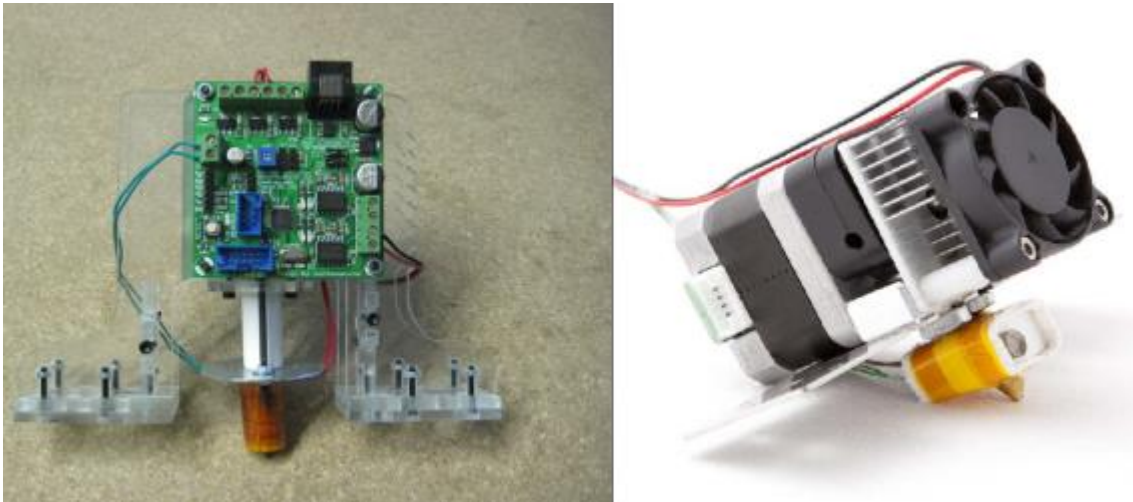


KUVA 5. Thing-O-Matic-pikamallinnuslaite (9)

Viimeisin MakerBotin lanseeraama pikamallinnuslaite on Replicator. Sen suurimpiin uudistuksiin kuului yli kaksinkertaistunut tulostuspinta-ala (225 x 145 x 150 mm), kaksi pursotuspäätä joiden ansiosta voidaan pursottaa kahta väriä ja päivitetty elektroniikka joka toi mukanaan LCD-näytön sekä ohjauspaneelin jolloin parametrien säätö on mahdollista ilman tietokonetta. Replicatorista on ilmestynyt myös kaksi päivitettyä versiota, Replicator 2 ja Replicator 2X, joihin on tehty pieniä parannuksia. (9.)

4.3 Stepstruder-pursotuspää

Stepstruder on MakerBotin kehittämä pursotuspääyksikkö, jota MakerBot käyttää omista pikamallinnuslaitteissaan. Ensimmäiset mallit kulkivat nimellä Plastruder. Stepstruder-pursotuspäitä on kehitetty samaa tahtia kun MakerBot on kehittänyt pikamallinnuslaitteitaan. Merkittävimpiä päivityksiä matkan varrella ovat siirtyminen DC-moottoreista askelmoottoreihin, sulatuskammion uudelleensuunniteltu parempi muoto, pienempi suutin, elektroniikan siirtäminen irralliseksi osaksi pursotuspäystä sekä huomattavasti kompaktimpi koko. MakerBot myy pursotuspäitään myös rakennussarjoina, joten koko pikamallinnuslaitetta ei tarvitse ostaa. Kuvassa 6 voidaan nähdä Plastruder MK4 ja Stepstruder MK8, joista huomaa selvästi kehityksen. (9.)



KUVA 6. Plastruder MK4- ja Stepstruder MK7 -pursotuspäät (11)

4.4 Arduino

Arduino on avoimeen laitteistoon ja koodiin perustuva yhdelle piirilevylle rakennettu mikrokontrolleri-alusta, jolla on myös oma vapaa ohjelmointiympäristönsä. Laitteen tekniikka perustuu avoimeen laitteistoon joka on kasattu 8-bittisen Atmel AVR -mikrokontrollerin ympärille. Arduinon mikropiirillä on sekä input-että output-pinnit johon voidaan lisätä erilaisia antureita, moottorinohjausta, LED-valoja ja muita tarvittavia elektroniikan komponentteja. Arduino-kontrollerit voidaan ostaa joko rakennussarjana tai tee se itse-sarjana. Piirikaaviot ovat myös vapaasti saatavissa, joten arduino-piirilevyn itse rakentaminen

komponenteista on myös mahdollista. (12.) Kuvassa 7 on esimerkkinä Arduino Uno-piirilevy.



KUVA 7. Arduino Uno -piirilevy (11)

Arduinon ohjelmointiympäristö koostuu normaalista ohjelmointikielen kääntäjäs-
tä ja käynnistyslataimesta jolla ohjelma saadaan mikrokontrollerille.

Ohjelmointiympäristössä ohjelmointikieli on C-kieleen perustuva, muttei kuiten-
kaan täysin vastaa C-kieltä. (12.)

5 MEKANIikkaVALINTOJEN JA KOMPONENTTIEN HAHMOTUS

Koska opinnäytetyön tuloksena on tarkoitus saada täysin uudenlainen työkalusovellus käyttöön NC-koneistuskeskukseen, siihen liittyy monia yhteensopivuus-ongelmia sekä mekaniikkasuunnittelua, jotta pursotuspää kiinnitys saadaan tukevaksi, mutta silti helposti irroitettavaksi. Projektin alkuvaiheessa pohdittiin järkevimpiä ja helpoimpia ratkaisuja työssä esiintyviin ongelma-kohtiin.

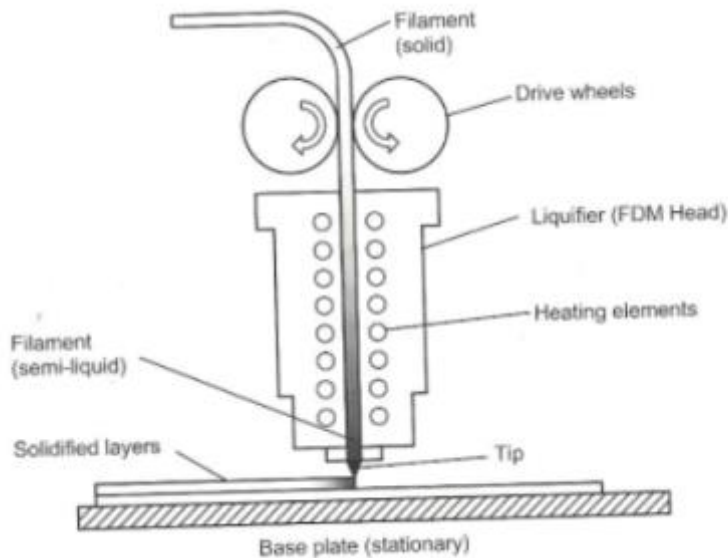
Ensimmäisenä asiana mietintälistalla oli koko systeemin yleinen kartoitus ja se, mitä komponentteja kokoamiseen tarvitaan ja minkälaisilla ohjelmilla pikamallinnustyökalu saadaan toimimaan NC-koneistusympäristössä. Seuraavassa pohditaan systeemin kriittisimpien osien toteutusta.

5.1 Pursotuspää

Jotta NC-koneistuskeskusta voidaan käyttää pikamallinnuskoneena, on siinä oltava työkalu, joka mahdollistaa pikamallien teon. Vaikka erilaisia pikamallinnustekniikoita on paljon, FDM-tekniikka on käytännössä ainoa jota voidaan soveltaa kyseiseen työhön sen yksinkertaisuuden, edullisuuden sekä helpohkon integroinnin takia. FDM-tekniikan määritelmä ja toimintaperiaate on selitetty tämän raportin kohdassa 2.4.

5.2 Pursotuspään rakenteen suunnittelu

FDM-tekniikassa itse pursotuspään rakenne on yksinkertainen. Se koostuu yksinkertaisimmillaan langansyöttökoneistosta, sulakammion sekä suuttimesta. Lisäksi tarvitaan sulakammiota lämmittävät elementit sekä lämpöanturikomponentti jotta sulakammion lämpötilaa voidaan seurata. Kuvassa 8 on kuvattuna pursotuspään peruseriaate ja perusosat.



KUVA 8. Pursotuspään komponentit ja toimintaperiaate (1, s. 143)

Ensisijainen pohdinnan aihe pursotuspään suhteen oli se, rakennetaanko ja suunnitellaanko pursotuspää itse vai hyödynnetäänkö jo markkinoilla olevien valmistajien valmiita pursotuspäitä.

Jos pursotuspää suunnitellaan ja rakennetaan itse, se edellyttää seuraavia asioita

- mekanismin suunnittelu
- syöttömekanismin osien valmistus tai ostaminen, hinta 40 - 80 e
- sopivan askelmoottorin ostaminen, hinta 20 – 30 e
- suuttimen ostaminen, hinta 15 e
- lämmitysvastuksen ja lämpötila-anturin ostaminen, hinta 20 e
- sulakammion valmistus tai ostaminen, hinta 20 – 40 e.

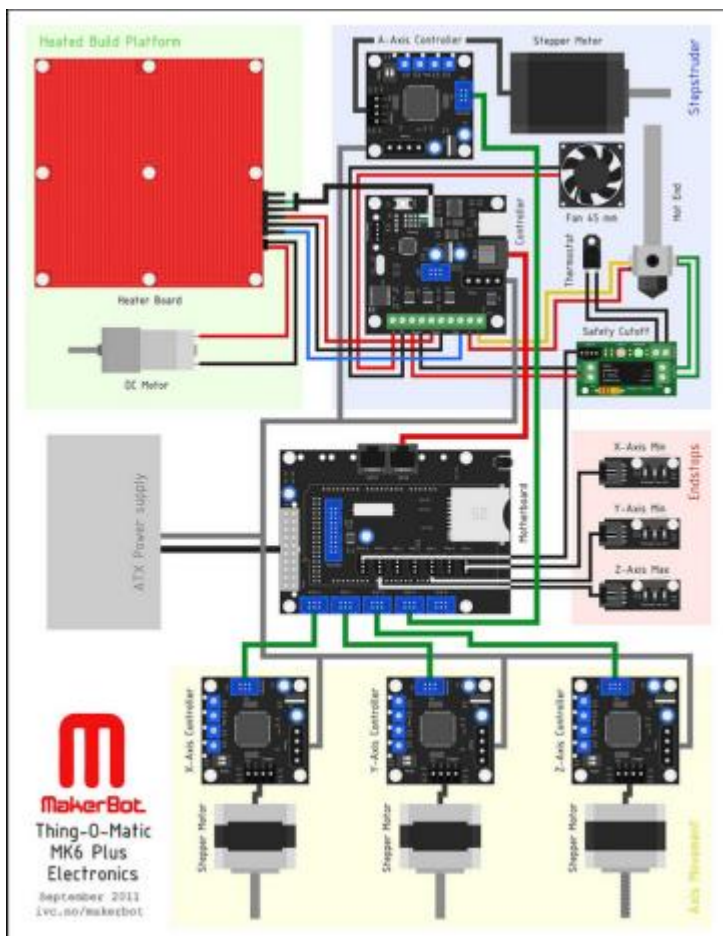
Hinta-arvioihin ei ole laskettu suunnittelusta ja rakentamisesta aiheutuvia kuluja. Toisena vaihtoehtona on Makerbotin omissa pikamallinnuskoneissa käytetty Stepstruder-pursotinpää. Se on rakenteeltaan monen kehitysmallin jälkeen kehittynyt hyvin toimivaksi. Uusimman MK7-mallin pursotuspään rakennussarjan saa noin 200 euron hintaan.

Pohdinnan jälkeen pursotuspään omatoiminen suunnittelu ja valmistus todettiin turhaksi Stepstruder-pursotuspään toimivan rakenteen ja edullisen kokonais-

hinnan vuoksi. Lisäksi koululla on Stepstruder MK6+ -pursotinpää, jolla voidaan tehdä alustavia kokeiluja ja testauksia uutta pursotinpäää odotellessa.

5.3 Pursotuspään ohjaus

MakerBotin alkuperäinen ohjaus tapahtuu useammalla avoimeen laitteistoon perustuvalla mikrokontrollerilla. Pursotuspään ohjaus MakerBotin alkuperäisellä elektroniikalla olisi helpoin tapa hallita pursotuspäätä, mutta se vaatisi koko MakerBotin elektroniikkapaketin ostamista, jossa on myös paljon ylimääräistä elektroniikkaa, joita ei opinnäytetyössä tarvita. Alkuperäinen MakerBotin laitekaavio voidaan nähdä kuvasta 9. Koska XYZ-tasojen ohjaus hoidetaan suoraan pc-laitteelta Robodrilliin, ovat esimerkiksi XYZ-tasojen mikrokontrollerit täysin turhia. Tämän vuoksi pursotuspään ohjaus päätettiin tehdä omalla Arduino-mikrokontrollerilla ja siihen kehitetään itse ohjelma, jolloin kulut saadaan minimoitua.



KUVA 9. Makerbotin elektroniikka ja kytkentäkaavio (13)

Pursotuspään Arduino-ohjelmassa tarvitaan lämpötilavastuksen hallinta sekä pursotuspään ON/OFF-komennot. Myös lämpötila-anturin seuraaminen tulee toteuttaa Arduino-ohjelmalla, jolloin tiedetään, että pursotuspää on lämmennyt tarpeeksi ollakseen valmis pursottamaan.

5.4 Robodrillin XYZ-tasojen ohjaus

3D-mallin muuntaminen NC-koneille kelpaavaksi G-koodiksi päätettiin tehdä MakerBot Industriesin vapaaseen koodiin perustuvalla ReplicatorG-ohjelmalla. Se viipaloi ja muuttaa 3D-mallinnusohjelmasta saatavan STL-tiedoston sopivaksi G-koodiksi. Fanucin ohjauksella toimivassa NC-koneessa G-koodit eivät kuitenkaan ole täysin samanlaisia kuin ReplicatorG:n tuottama G-koodi.

Koska koodit poikkeavat toisistaan on kehitettävä myös ohjelma, joka tekee Fanucin vaatimat pienet muutokset G-koodiin. Ennen koodin syöttöä koneistukeskukseen se on ajettava ensin tämän ohjelman läpi. Ohjelman on myös pystyttävä jakamaan käskyt kahteen osaan, pursotuspäälle meneviin koodeihin sekä XYZ-tasojen hallitseviin koodeihin Robodrillille. Ohjelman on pidettävä myös huolta siitä, että molemmat näistä kahdesta käskytettävästä pysyvät samassa tahdissa.

5.5 Kättely tietokoneen ja muiden laitteiden välillä

Ongelmalliseksi tilanteen tekee Robodrillin rajoittunut käyttömuisti, jonka vuoksi koko koodia ei voida syöttää kerralla työstökoneeseen. PC-laitteen on tästä syystä yhdistettynä RS-232-C-kaapelin välityksellä Robodrilliin ja Robodrillia käytetään remote-tilassa. Koodi on pilkottava pienemmiksi osiksi ja aina kun Robodrill on suorittanut edelliset komennot, se ilmoittaa PC-laitteelle valmiudesta ottaa vastaan lisää koodia. Tätä kutsutaan kättelyksi. Kättelyllä tarkoitetaan käytännössä sitä, että tiedonsiirron osapuolet kertovat toisilleen, koska lähetetty tieto on käsitelty ja tiedon lähettämistä voi jatkaa.

Kättely tarvitaan myös pursotuspäätä ohjaavan Arduinon sekä PC-laitteen välille. Arduinon ohjelman on ilmoitettava PC-laitteelle, milloin se on lämmennyt riittävään lämpötilaan ja on pursotusvalmiudessa. Robodrill eikä pursotuspää

saa lähteä kumpikaan liikkeelle, ennen kuin molemmat ovat valmiudessa ja samassa vaiheessa.

6 PURSOTUSPÄÄN SOVITUS ROBODRILLIIN

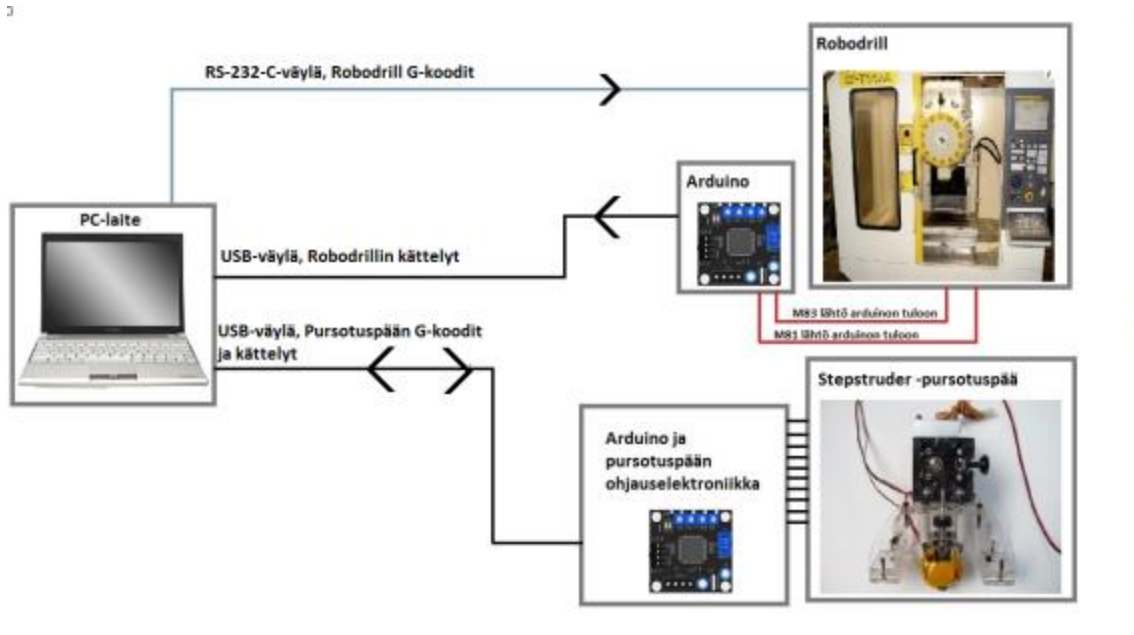
Alkuperäisen suunnitelman mukaisesti pursotuspääksi päätettiin hankkia MakerBot Industriesin valmistama Stepstruder MK7 -pursotuspää. Sen saa ostettua noin 200 euron hintaan, joten oman pursotuspään suunnittelemista ja valmistamista ei pidetty kannattavana tai hyödyllisenä. MakerBot on kuitenkin juuri siirtymässä pikamallinnuslaitteissaan uuteen Stepstruder MK8 -pursotuspäähän, jonka vuoksi MK7-pursotuspään valmistus oli lopetettu eikä MK8-pursotuspäätä vielä saanut ostettua erillään.

Koululta löytyi kuitenkin Jani-Pekka Moilasen (14) tekemästä opinnäytetyöstä Robotisoitu pikamallilaite jäänyt Stepstruder Mk6+ -pursotuspää, jota päätettiin käyttää myös tämän työn prototyypivaiheessa.

6.1 Laitekaavio

Työssä käytetään Robodrill-koneistuskeskusta, MakerBotin valmistamaa Stepstruder MK6+-pursotuspäätä, laitteiden ohjelmoinnissa Arduino-mikrokontrolleripiirejä ja laitteiden hallitsemiseen käytetään PC-laitetta, joka antaa sarjaportin kautta käskyt sekä pursotuspäälle että Robodrillille.

Suurimpana ongelmana oli saada koneistuskeskukselta kättelyviestin PC-laitteelle, koska Robodrill itse ei pysty RS-232-C-väylän kautta lähettämään viestiä takaisin. Robodrillistä löytyy kuitenkin kaksi vapaata ON/OFF-tyyppistä lähtöä, joita voidaan käyttää kytkevillä M83- ja M81-koodeilla. Näitä lähtöjä voidaan hyödyntää Robodrillin ja PC-laitteen välisessä kättelyssä kun lisätään kokoonpanoon vielä yksi Arduino-mikrokontrolleri, joka seuraa näiden kahden lähdön tiloja ja ilmoittaa PC-laitteelle niistä. Kun G-koodiin lisätään tietyn välein M81 koodi, lähtö M81 menee ON-tilaan noin 100 ms ajaksi, jolloin tieto koodin etenemisestä tähän pisteeseen saakka saadaan välitettyä arduinon kautta PC-laitteelle. Kuvassa 10 on koko laitekaavio sekä väylien tehtävät selostettuna. Nuolista voidaan nähdä, mihin suuntaan tieto liikkuu missäkin väylässä



KUVA 10. Laitekaavio

G-koodiohjelman alussa annetaan pursotuspään lämmitysvastukselle arvo. Tämän jälkeen PC-ohjelma jää odottamaan pursotuspäätä ohjaavalta Arduinolta M6ok-tulostetta sarjaportista, joka annetaan kun lämpöanturin lukema vastaa annettua lämpötilan arvoa. Kun M6ok on tulostettu sarjaporttiin, PC-laitteen ohjelma voi jatkaa G-koodia eteenpäin. Näin toimii pursotuspään kättely.

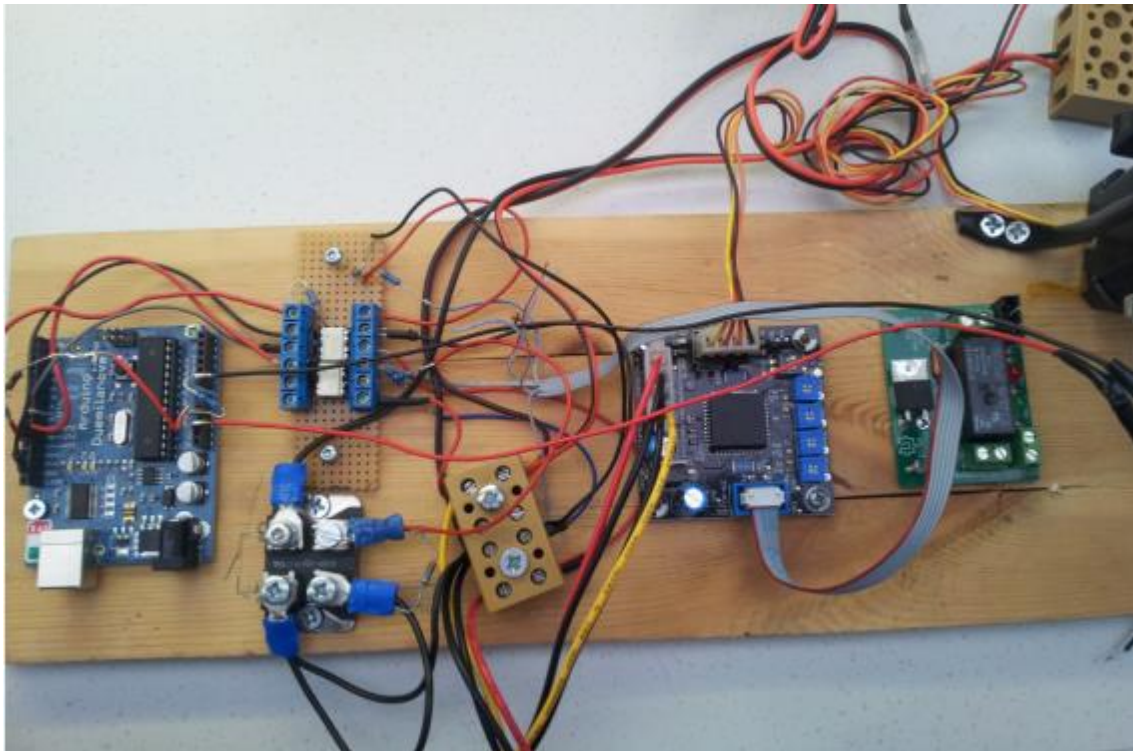
6.2 Elektroniikka

Pursotuspään tarvitsema elektroniikka päätettiin sijoittaa prototyypivaiheessa levyille, jolloin elektroniikka olisi helposti poistettavissa kun pursotuspäätä ei käytetä työkaluna. Koska pursotuspäätä käytettäessä esimerkiksi leikkuunestettä tai muita elektroniikkaa vahingoittavia tekijöitä ei ole, voitiin elektroniikkapaneeli sijoittaa Robodrill-koneistuskeskuksen koneistustilaan suojaseinämän sisäpuolelle. Elektroniikkapaneeli kiinnittyy paikalleen turvakopin sisäpuolelle kahden tehokkaan magneetin avulla, jolloin paneeli ei pääse liikkumaan sekä se on helposti ja nopeasti siirrettävissä.

6.2.1 Elektroniikan komponentit

Koululla olevaan Stepstruder-pursotuspäähän oli opinnäytetyötä aloittaessa kasattu jo jollain lailla toimiva elektroniikka. Kuvassa 11 on alkuperäinen elektroniikka ja seuraavassa on lista alkuperäisestä elektroniikasta:

- PC-laitteen 5V/12 V virtalähde
- Arduino Duemilanove -mikrokontrolleri pursotinpään ohjaukseen
- Arduino Uno-mikrokontrolleri Robodrillin kättelyiden hoitamiseen
- MakerBot Stepper motor Driver v. 3.3 -askelmoottoriohjain
- 2 kpl CNY74-2H -optoeristimiä
- STE180N10 -tehotransistori
- EPCOS B57560G104+ -lämpöanturi.



KUVA 11. Alkuperäinen elektroniikka

Pursotuspään ohjaukseen käytetään Arduinon Duemilanove-mikrokontrolleria. Arduino-mikrokontrollereista on selitetty tarkemmin tämän opinnäytetyön kohdassa 4.4.

Virtalähteenä käytettiin normaalia PC-laitteen virtalähdettä, josta saadaan sekä 5V- että 12V -jännitettä.

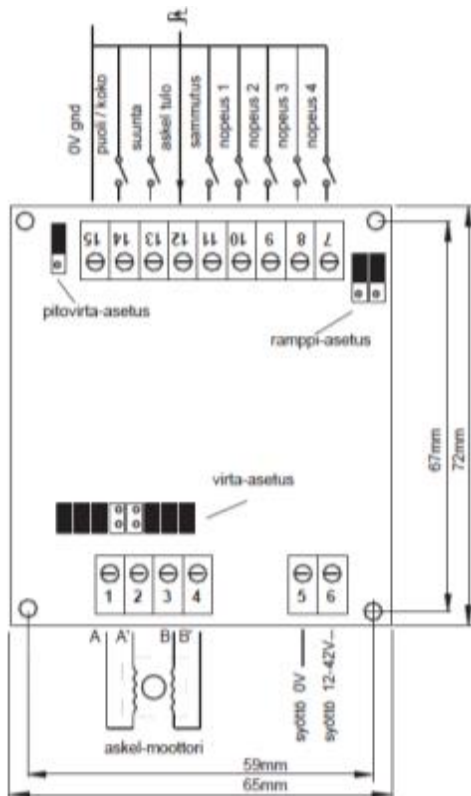
Stepstruder MK6+ käyttää langansyötössä askelmoottoria. Askelmoottorit vaativat yleisesti erillisen askelmoottoriohjaimen, jos tätä ei ole integroituna askelmoottoriin. Ensimmäisissä testiversioissa askelmoottoriohjaimena käytettiin MakerBotin omaa Stepper Motor Driver 3.3 -ohjainta.

Optoeristimiä käytetään erottamaan ohjaava ja ohjattava virtapiiri toisistaan. Tässä tapauksessa siis arduinon virtapiiri erotetaan ohjattavasta pursotuspään virtapiiristä. MOSFET-tehotransistoria taasen käytetään lämmitysvastuksen ohjauksessa, jolloin suuria virtoja voidaan ohjata pienellä jännitteellä, kuten tässä tapauksessa Arduinon 5V-lähdöllä. Lämpöanturina käytettiin analogista PCOS B57560G104+ -lämpöanturia.

6.2.2 Lopullinen elektroniikka

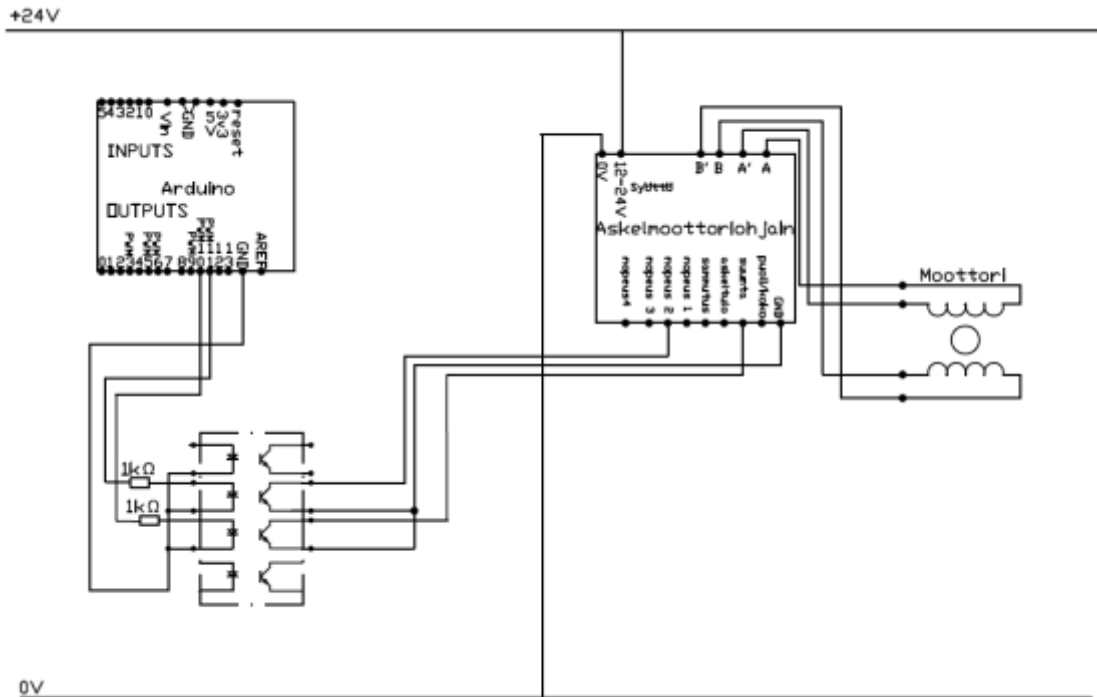
Alkuperäisessä testiversiossa askelmoottoriohjaimena käytettiin MakerBotin omaa Stepper Motor Driver 3.3 -ohjainta. Stepper Motor Driver 3.3 -ohjaimessa heikkoutena oli kuitenkin se, ettei itse ohjain sisältänyt minkäänlaista nopeudensäätöä tai kiihdytysrampin toimintoja vaan nämä toiminnot olisi sisällytettävä Arduinon ohjelmaan joka ohjaa pursotuspäätä.

Koululta löytyi myös EM-121 -ohjain, jossa taajuutta, eli tässä tapauksessa moottorin pyörimisnopeutta, voidaan säätää suoraan neljällä ennakkoon asetetulla ohjaustulolla. Ohjaimesta voi myös helposti säätää kiihdytysrampit ja virran määrää pelkillä hyppyyliittimillä. Kuvassa 12 on EM-121 -ohjaimen kytkentäkaavio, jossa on myös nähtävillä ramppiasetukset ja virta-asetukset.



KUVA 12. EM-121-ohjaimen kytkentäkaavio (15)

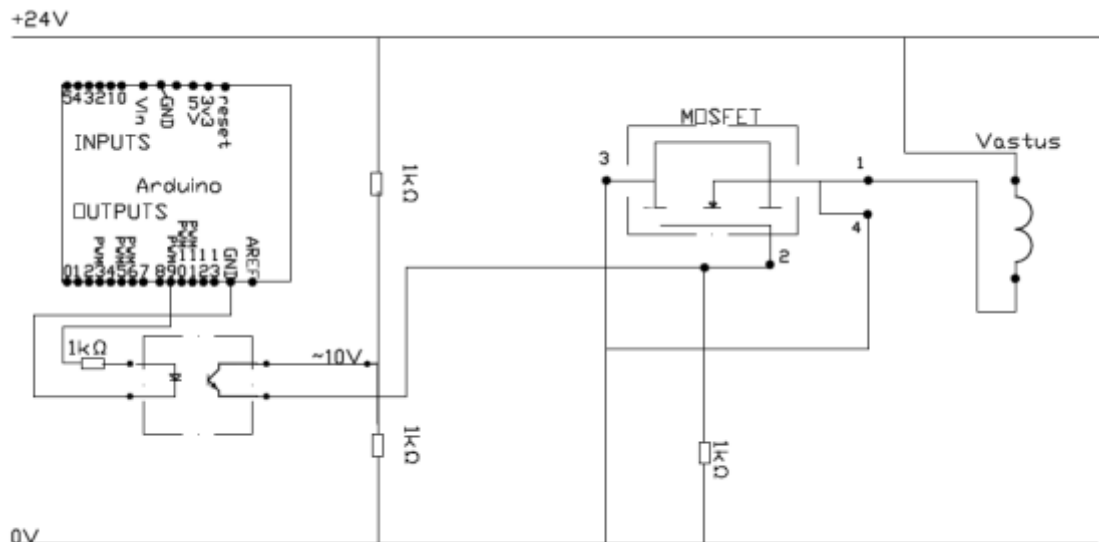
Näiden ominaisuuksien vuoksi lopulliseen kokoonpanoon valittiin EM-121 -askelmoottorihjain, jotta Arduinon ohjelmaa saatiin muokattua mahdollisimman yksinkertaiseksi ja toimivaksi ja askelmoottorin säädöt saadaan hoidettua ohjaimella. Kuvassa 13 on ohjaimen sähkökaavio kyseisessä opinnäytetyössä.



KUVA 13. Moottorinohjauksen sähkökaavio

Moottorinohjaukselle on kyseisessä opinnäytetyössä Arduinolla kaksi lähtöä, suunta ja moottorin eteenpäin ajo. Output-pinni 10 on moottorin eteenpäin ajo ja pinni 11 on moottorin suunnan ohjaukselle. Ohjauksikäskyn lähdettyä Arduinolta se kulkee optoeristimelle, joka avaa ohjattavan puolen. ohjattavalla puolella askelmoottorinohjaimen ohjaustulo kytketään maahan virrattomaksi, jolloin käsky kulkee moottorille ohjaimen kautta.

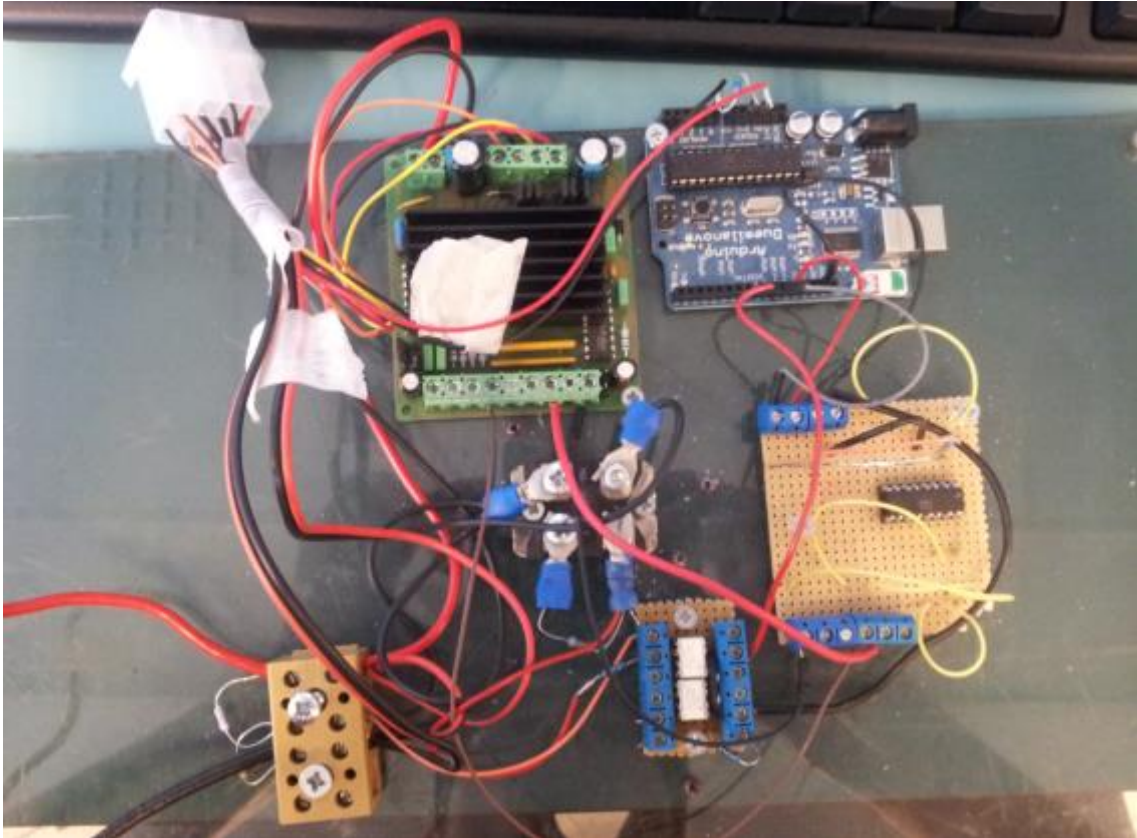
Koska EM-121-ohjaimen suositeltu jännitealue on 12 - 40V, jäi alkuperäinen tietokoneen virtalähde hieman liian tehottomaksi. Tämä huomattiin askelmoottorin puuttavana vääntönä jolloin askelmoottori jäi tykyttämään paikalleen eikä vetänyt lankaa. Virtalähde vaihdettiin 24 V jännitteellä ja 5 A virralla toimivaan Festo-virtalähteeseen. Virtalähteen vaihdolla saatiin myös huomattavasti nopeampi lämmitysvastuksen lämpeneminen, joten pursotuspään valmiusaikaa saatiin pienennettyä. Kuvassa 14 on MOSFET-tehotransistorin sekä vastuksen sähkökaavio.



KUVA 14. Vastuksen sähkökaavio

Lämmitysvastusta ohjataan Arduinon PWM output-pinnillä 9. PWM-pulssileveysmodulaatiosta kerrotaan enemmän sivulla 42 kohdassa 8.1. Ohjauksikäskyn lähdettyä arduinolta se kulkee optoeristimen kautta, joka avaa ohjattavan puolen. Kun MOSFET saa ohjausjännitteen, se päästää maan lävitseen 3-1 välillä ja vastus alkaa lämmitä. Pursotuspään lämmitysvastukselle optoeristimeltä lähtevä MOSFETin ohjausjännite piti alentaa noin 10 V jännitteeseen vastussillalla, 24V jännitteen ollessa liikaa.

Kuvassa 15 on esillä lopullinen elektroniikkakokoonpano prototyypilevyllä. Johdotus pursotinpäälle katkaistiin liittimellä, jotta pursotuspään paikalleen asentaminen olisi mahdollisimman helppoa. Myös optoerotin vaihdettiin lopulliseen versioon Liteonin LTV847 16-pinniseen optoerottimeen, jolloin kaikki kytkennät voidaan hoitaa yhdellä optoeristimellä. Koko sähkökaavio on tämän opinnäytetyön liitteenä 2.



KUVA 15. Lopullinen elektroniikkaprototyyppi

Lopulliseksi prototyyppivaiheen elektroniikaksi muodostuivat seuraavat komponentit:

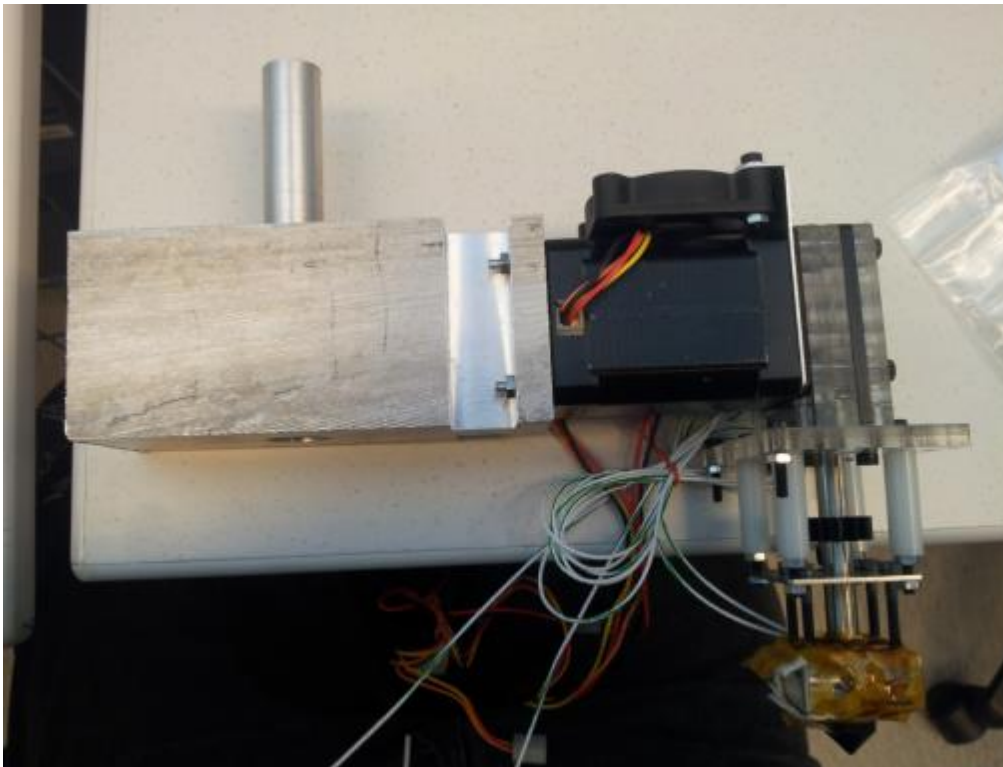
- Festo Didactic 24V 5A:n virtalähde
- Arduino Duemilanove -mikrokontrolleri pursotinpään ohjaukseen
- Arduino Uno -mikrokontrolleri Robodrillin kättelyiden hoitamiseen
- EM-121 -askelmoottorihjain
- Liteon LTV847 -optoeristin
- STE180N10 -tehotransistori
- EPCOS B57560G104+ -lämpöanturi.

6.3 Pursotuspään kiinnitys robodrilliin

Pursotuspään kiinnityksen osalta harkittiin muutamia vaihtoehtoja. Pursotuspään kiinnityksen tuli olla tukeva, jotta pursottaessa pursotussuutin pysyy varmasti paikallaan. Pursotuspäätä ei voida kiinnittää suoraan työkalupaikalle, koska johdotuksen ja suuren koon takia Robodrillin automaattista työkalunvai-

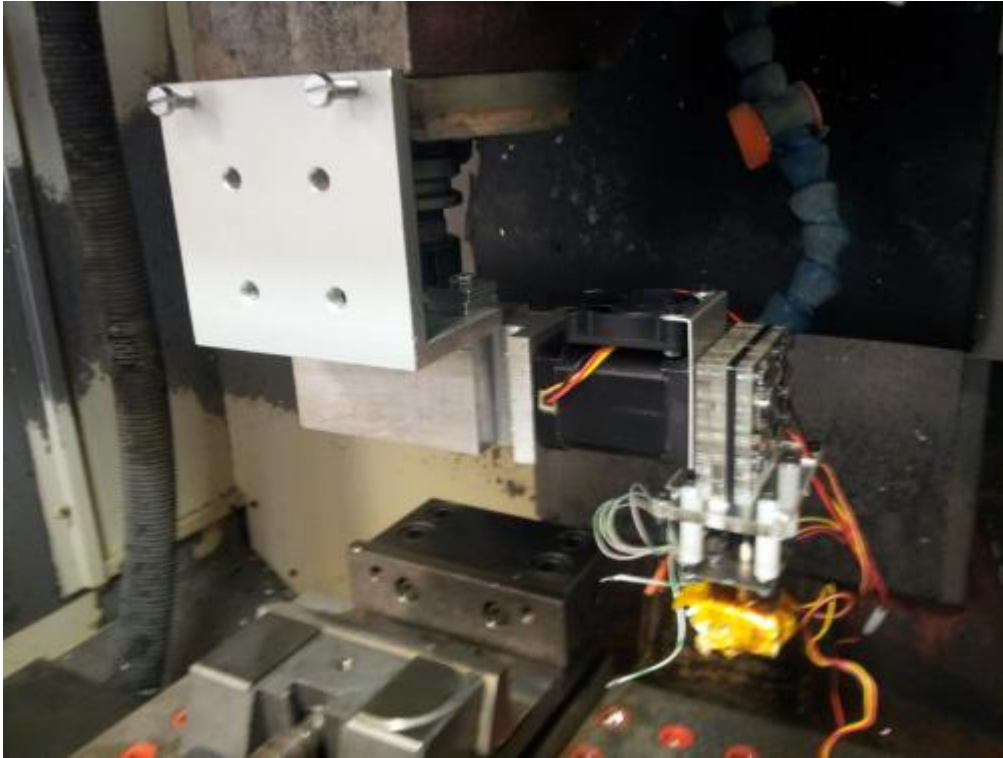
toa ei voida käyttää. Myös Robodrillin suojakopissa olevan tilan käyttö tuli saada mahdollisimman alhaiseksi.

Parhaimmaksi ratkaisuksi nähtiin kiinnittäminen sopivaan työkalukaraan kiinnittävään 16 mm metallitankoon. Metallitanko voidaan pitää paikallaan työkalupaikalla ja se saadaan tarvittaessa työkalua vaihtamalla esille. Prototyypivaiheessa kiinnitin päätettiin koneistaa koulun laboratoriosta löytyneestä sopivasta alumiinipalasta. Alumiinipalaan porattiin reikä 16 mm tangolle ja sen sivuilta koneistettiin sen verran metallia pois, että pursotuspään askelmootorin päästä tulevat kierretangot voidaan kiristää kiinnittimeen. Alumiinista sorvattiin 16 mm tanko, joka ahdistussoviteella liitettiin kiinnittimeen. Kuvassa 16 on kiinnittimen perusosa kiinnitettynä pursotuspäähän. Lopulliseen versioon metallitangosta tulee irroitettava.



KUVA 16. Kiinnitin ilman tukiprofiilia

Yläpintaan porattiin myös kaksi reikää, jolloin kiinnittimeen saatiin kiinnitettyä alumiiniprofiili jolla pursotuspää saatiin tuettua Robodrillin runkoa vasten. Tuenta tarvittiin koska työkalun pyörimisen estävän komennon jälkeenkin työkalu pääsee liikkumaan tarpeeksi aiheuttaakseen liikkumista kiinnikkeessä. Kuvassa 17 on prototyypikiinnitin kokonaisuudessaan.



KUVA 17. Kiinnitin tukiprofiileineen Robodrillin työkalupaikalla

7 ROBODRILLIN JA PURSOTUSPÄÄN OHJAAMINEN

Yksi opinnäytetyön haastavimmista osista oli kehittää sekä Robodrillille että pursotuspäälle sopivat ohjelmat, jolla niitä molempia voidaan ohjata yhtäaikaaisesti. Kättely on pakollinen osa molempien ohjausta, koska sekä Robodrillin että pursotuspään käskyjen pitää tulla samassa tahdissa.

7.1 Pursotuspään ohjelmointi

Pursotuspään ohjelmointiin käytettiin alkuperäisen suunnitelman mukaisesti Arduino-mikrokontrolleria, johon kehitettiin oma ohjelmakoodi pursotuspään ohjaamiseksi. Pursotuspään ohjelman on huolehdittava pursotuksen ON/OFF-komennot, lämmitysvastuksen lämmityskomennot sekä lämpötila-anturin seuranta jonka mukaan lämmitystä pidetään päällä. Seuraavassa on lista pursotuspäälle vaadittavista toiminnoista:

- M101 pursottimen moottorin ON-komento
- M103 pursottimen moottorin OFF-komento
- M102 pursottimen moottorin ON-komento, suunnan vaihto
- TXXX lämpötilan asetus
- M106ok ohjelma kirjoittaa sarjaporttiin M106ok kun pursotuspään lämpötila on oikea.

Koko laitteen komennot tulevat sarjaportin kautta, joten Arduinon input-tuloilla ei voitu hoitaa Arduinon ohjausta, vaan ohjelman on osattava poimia käskyt tekstimuotoisina suoraan sarjaportista. Tämän vuoksi ohjelmaan täytyi kehittää koodi, joka osaa lukea komennon ensimmäisen kirjaimen, jonka perusteella komennot voidaan eritellä. Tämä tapahtuu käyttämällä ASCII-merkkejä. ASCII on 7-bittinen tietokonemerkistö, joka sisältää ensisijaiset numerot, kirjaimet ja joitain väli- ja erikoismerkkejä. 7-bittisyys tarkoittaa sitä, että ASCII-merkistössä on 128 muistipaikkaa. Jokaisella muistipaikalla on siis oma merkkinsä. (16.) Kuvassa 18 on se osa koodia, jossa ohjelma reagoi, jos sarjaporttiin syötetään T-kirjain.

```

if (Serial.available() > 3) //Jos portissa on tavuja, luetaan sisältö tavu kerrallaan
{
  tulevatavu = Serial.read();
  merkki=tulevatavu;

  Serial.write(tulevatavu);

  //Heitetään perään vielä rivinvaihto LF eli ASCII-merkki 10
  //Komentojen perässä käytetään usein sitä ja/tai merkkiä 13=CR

  Serial.write(10);

  if (tulevatavu==84) // Jos tuleva merkki on t eli ASCII 84
  {
    Serial.println("Pursottimen °C"); //Pursottimen muuttujalle temperature annetaan seuraavat kolme tavua merkin T jälkeen
    ensitavu = Serial.read();
    toinentavu = Serial.read();
    kolmastavu = Serial.read();
    temperature=(ensitavu-48)*100+(toinentavu-48)*10+(kolmastavu-48)*1; //merkkien muuttaminen normaaliksi luvuksi
    //Serial.print(ensitavu); //ja tallentaminen muistipaikalle temperature
    //Serial.print(toinentavu);
    //Serial.print(kolmastavu);
    Serial.println(temperature); //tulestetaan saatu lämpötilan arvo
  }
}

```

KUVA 18. Lämpötila-säädön koodi

T on ASCII-merkistössä numero 84. Kun T-kirjain on löytynyt sarjaportista, ohjelma lukee sen jäljessä tulevat kolme tavua, joista saadaan lämpötilamuuttujalle temperature arvo. ASCII-merkeissä numerot alkavat merkistä 48, joten vähentämällä täydestä ASCII-merkistä 48, saadaan oikea numero. Sadat kerrotaan sadalla, kymmenet kymmenellä ja yksiköt yhdellä. Kun nämä lasketaan yhteen, saadaan aikaan 3-numeroinen luku. Esimerkiksi ohjelman poimiessa komennon T225, se tietää T-kirjaimen tarkoittavan komennon olevan lämpötilalle ja seuraavat kolme numeroa osataan laskea sen muuttujan arvoksi.

Pursotuspään pursotuskäskyjä hallitseva osa ohjelmakoodia toimii lähes samalla tavalla kuin lämpötila-arvon vastaanottava osa koodia. Se reagoi jos sarjaporttiin syötetään M-kirjain. Pursotuspään käskyjä säätelevä osa koodista on kuvassa 19.

```

if (tulevatavu==77) // jos tuleva tavu on M-kirjain, niin:
{
  //Serial.println("Pursottimen M-komento");
  neljastavu = Serial.read();
  viidestavu = Serial.read();
  kuudestavu = Serial.read();
  pursotinmuuttuja=((neljastavu-48)*100+(viidestavu-48)*10+(kuudestavu-48)*1); //lukeman muuttaminen normaaliksi luvuksi
  //Serial.print(ensitavu); //ja tallentaminen muistipaikalle temperature
  //Serial.print(toinentavu);
  //Serial.print(kolmastavu);
  // Serial.println("pursotinmuuttuja: ");
  // Serial.println(pursotinmuuttuja);
}

if (pursotinmuuttuja==101) //jos muuttujan pursotinmuuttuja-arvo on 101, niin:
{
  digitalWrite(suuntaPin, LOW); //suuntalähtö on OFF-tilassa
  pursotin=1; //pursotin=1 eli pursotin pursottaa
  Serial.println("M1ok"); //tulostetaan Mlok
}

if (pursotinmuuttuja==102) //jos muuttujan pursotinmuuttuja-arvo on 102, niin:
{
  digitalWrite(suuntaPin, HIGH); //suuntalähtö on ON-tilassa (pyörii toiseen suuntaan)
  pursotin=1; //pursotin=1 eli pursotin pursottaa
  Serial.println("M2ok"); //tulostetaan Mlok
}

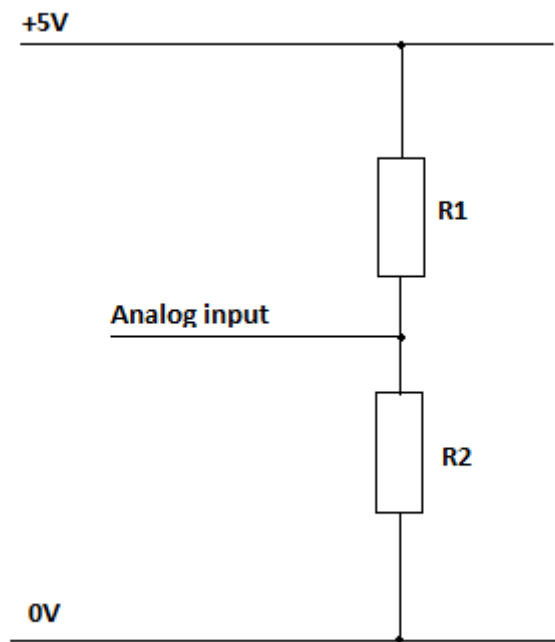
if (pursotinmuuttuja==103) //jos muuttujan pursotinmuuttuja-arvo on 103, niin:
{
  pursotin=0; //pursotin pois päältä
  Serial.println("M3ok"); //tulostetaan Mlok
}

```

KUVA 19. M-komentojen koodi

Ensimmäisen IF-lauseen ehtona on se, että merkki M pitää syöttää sarjaporttiin. Jos M on syötetty, ohjelma poimii sen perässä tulevat kolme tavua ja muuttaa sen takaisin numeroarvoksi, sekä tallentaa sen pursotinmuuttuja-muuttujan arvoksi. Käskyjä on kuitenkin kolme, M101, M102 ja M103. Tätä varten on tehty erilliset IF-lausekkeet jokaiselle eri komennolle. Jos pursotinmuuttuja-muuttujan arvoksi on annettu esimerkiksi 101, on ensimmäinen IF-lause voimassa ja moottori pyörii eteenpäin ja pursotin on päällä. Taasen esimerkiksi pursotinmuuttuja-muuttujan ollessa 103 on viimeinen IF-lause voimassa ja pursotin on pois päältä.

Jotta pursotuspään lämpötilaa voitaisiin säätää ja seurata, täytyi ohjelmaan kehittää myös kaava, jonka avulla lämpötila-anturin resistanssiarvot saadaan muutettua asteita vastaaviksi. Tällöin pursotuspäälle voidaan antaa suoraan lämpötila asteina. Kuvassa 20 on anturin sähkökaavio.



KUVA 20. Lämpötila-anturin sähkökaavio

Lämpötila-anturi on kuvassa R1 ja R2 taasen on 1kΩ vastus. Arduinolle lähtevä analogiatulo on näiden välissä. Tästä halutaan tietää resistanssin arvo joka tulee analogiatulon kautta Arduinolle. Resistanssi saadaan laskettua soveltamalla jännitteen jaon kaavaa 1 (17, s. 121.)

$$U_2 = \frac{R_2}{R_1 + R_2} * U_1$$

KAAVA 1

U_2 = lopullinen jännite(V)

U_1 = alkuperäinen jännite (V)

R_1 =resistanssi (Ω)

R_2 =resistanssi (Ω)

Muokkaamalla saadaan kaava, joka antaa analogiatuloon resistanssin arvon. Arduinossa on 10-bittinen analogiatulo, jonka suurin arvo on siis 1024. Kun kaavaan sijoitetaan alkuperäisen jännitteen tilalle 1024 ja lopullisen jännitteen tilalle analogiatulon arvon, saadaan kaava 2.

$$\text{Analogiatulo} = \frac{R_2}{R_1 + R_2} * 1024$$

KAAVA 2

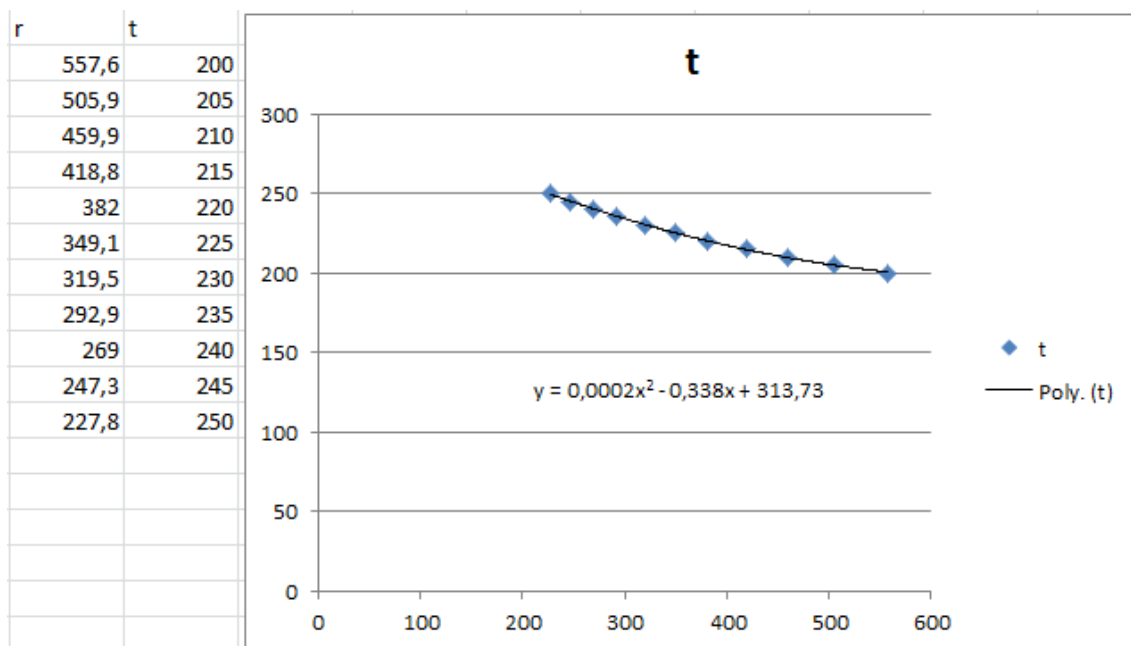
1024= 10-bittinen analogiatulo $10^2=1024$

Analogiatulo= Arduinon analogiatulon arvo

Kaava 2 haluttuun muotoon sijoitettuna

$$R1 = \frac{R2(1024 - \text{Analogiatulo})}{\text{Analogiatulo}}$$

Lämpötila-anturin teknisissä tiedoissa on taulukko, johon on listattuna tiettyjä lämpötiloja vastaavat resistanssit. Lämpötila-anturin tekninen tiedosto on tämän opinnäytetyön liitteenä 10. Näistä taulukoista saatiin tehtyä Excel-ohjelmalla käyrä ja käyrän yhtälö, jolla saadaan laskettua resistanssin arvo asteina. Kuvassa 21 on sekä käyrä että sen yhtälö ja kuvassa 22 on lämpötilaa koskeva osa koodia.



KUVA 21. Lämpötila-anturin käyrä ja yhtälö

```

tempAnalog = analogRead(analogPin); // tempAnalog-muuttujan arvo luetaan analogiatulosta
delay(100);
R1=(1000*(1024-tempAnalog)/tempAnalog); //R1-arvon laskeminen
vertailuarvo=180; //vertailuarvo-muuttuja 180
if(R1<1000) vertailuarvo=(0.0002*R1*R1-0.338*R1 + 313.73); //jos R1<1000 niin vertailuarvo lasketaan kaavalla
//Serial.println(vertailuarvo);
suhde=(vertailuarvo/temperature)*100; //suhde on vertailuarvo jaettuna annetulla lämpötilan arvolla

if (suhde<95) //jos suhde<95 on lämmityksen ohjearvo 255
{
ohje=255;
//Serial.println("heat 100%");
}
if (95<suhde && suhde<105)//jos suhde on 95-105 on lämmityksen arvo kaavan ohje=20*(105-suhde) mukainen
{
ohje=20*(105-suhde);
//Serial.println("heat 80%");
}
if (suhde>105)//jos suhde on yli 105 on lämmityksen ohjearvo 0
{
ohje=0;
Serial.println("heat off%");
}
if (suhde>100)//jos suhde on yli 100, on pursotuspää pursotusvalmis
{
targettemp=1;
}
analogWrite(vastusPin,ohje); //PWM-lähtöön annetaan edellisten IF-lauseiden mukaan lämmityksen ohjearvo

if(suhde<95) //jos suhde on alle 95, pursotuspää ei ole pursotusvalmiudessa
{
targettemp=0;
}

if(targettemp==1)//Kun pursotin on pursotusvalmiudessa, annetaan sarjaporttiin teksti M6ok
{
Serial.println("M6ok");
}

```

KUVA 22. Lämpötilan säätöä koskeva koodi

Ensin analogiatulosta luetaan arvo muuttujalle tempAnalog. TempAnalog-muuttujan avulla lasketaan arvo R1-vastukselle, eli resistanssin arvo lämpöanturissa. Kun vastuksen resistanssi on selvillä, lämpötila voidaan laskea käyrästä saadulla yhtälöllä jolloin saadaan vertailuarvo-muuttujan resistanssia vastaava lämpötila.

Anturin lämpötilan vastaavuus resistanssiin ei kuitenkaan ole lineaarinen, joten käyrään otettiin mukaan vain kriittisin lämpötila-alue 200 - 250 celsiusasteeseen. Tämän vuoksi muilla alueilla kuin 180 - 260 °C vertailuarvo-muuttujan antama lukema ei ole luotettava. Siksi vertailuarvolle annetaan arvo 180, kunnes resistanssin R1 arvo on alle 1000. Tämän jälkeen Resistanssin arvo on tarpeeksi lähellä oikeaa lämpöaluetta, ja sen tuloksia voidaan pitää luotettavina.

Tämän jälkeen vertailuarvosta ja annetusta temperature-arvosta lasketaan prosentuaalinen suhde, jonka avulla ohjataan pursotinpään vastuksen lämpiämistä.

Vastuksen ohjauksessa käytetään PWM-lähtöä, eli pulssileveysmodulaatiota. Se on modulointitapa, jossa kuormaan menevää jännitettä säädetään muuttamalla pulssisuhdetta niin, että lähtösignaalin keskiarvo yhden värähtelyjakson ajalta laskettuna on sama kuin modulointisignaalin arvo. (18.) Käytännössä tämä tarkoittaa sitä, että lähtöä voidaan pitää koko ajan päällä, mutta sen tehoa voidaan säädellä välillä 0 - 255, jossa 255 on maksimi ja 0 minimi.

Kun suhde on alle 95 %, on lähtö täysin auki, eli ohjearvoksi on annettu 255. Suhteen ollessa välillä 95–105 % saadaan ohjearvo kaavan 4 mukaan.

$$\text{ohje} = 20 * (105 - \text{suhde})$$

KAAVA 4

Esimerkiksi suhteen ollessa 103 % on lähdön PWM-arvo 40, eli teho on noin 16 % täydestä tehosta. Analogwrite-komennolla annetaan arduinon lähdölle käsky lämmittää vastusta ohjearvon mukaisella teholla.

Suhteen ylittäessä 100 % rajan saa muuttuja targettemp arvokseen numeron 1. Kun targettemp-muuttujan arvo on 1, täyttyy IF-ehto ja sarjaporttiin tulostetaan "M6ok", joka on pursotuspään kättelyn tapa sanoa, että pursotuspää on valmis pursotukseen. Jos suhde taas menee alle 95 % rajan, on targettemp=0 ja IF-ehto sulkeutuu. M6ok-ilmoitusta odotetaan aina G-koodista tulevan M6-lämmityskäskyn jälkeen. Koko ohjelmakoodi kommentoituna on tämän opinnäytetyön liitteenä 8.

7.2 PC-laitteen ohjelma

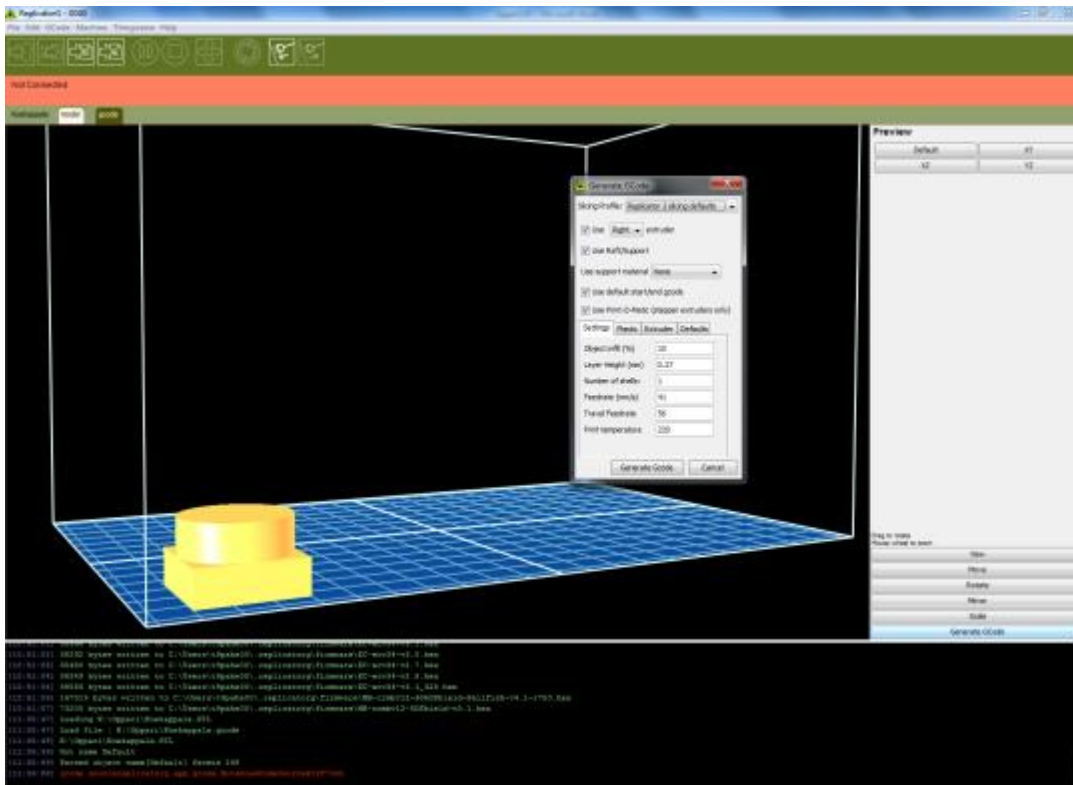
PC-laitteen ohjelmalla oli useampia tehtäviä kuin pursotuspään ohjauksella. Ohjelman tuli osata jakaa G-koodi niin, että pursotuspäälle ja Robodrillille menevät vain niitä koskevat komennot, poistaa tarvitsemattomat osat G-koodista, ja myös molempien laitteiden kättelyt olivat ohjelman hallinnassa. Kuten aiemmin jo laitekaavio-otsikon alla todettiin, Robodrill ei suoraan voi lähettää RS-232-C-väylää pitkin kättelyitä PC-laitteelle, joten kättely oli ratkaistava ylimääräisellä Arduinolla ja Robodrillistä löytyvillä kahdella vapaalla output-lähdöllä, M81 ja M83. Robodrillin käyttäjä voi itse käyttää vapaasti näitä kahta lähtöä.

Tiedonsiirto päätettiin tehdä pelkän M81-koodin avulla, joka lisätään koodiin aina ennen pursotuspään komentoja ja tietyin riviväleihin koodissa. M81-koodi voidaan syöttää esimerkiksi 20 rivin välein, jolloin vältetään mahdollinen Robodrillin muistin loppuminen. Kun Robodrill suorittaa M81-koodin, se kytkee lähdön M81 transistorin johtavaksi noin 100 ms ajaksi ja Robodrillin lähtöjä vahtiva Arduino lähettää USB-sarjaporttiinsa viestin M81. Kun viesti M81 on tullut PC-laitteelle, G-koodin lähettämistä jatketaan. Näin varmistetaan että Robodrill on varmasti suorittanut kaikki liikkeet ja kulkee samaa tahtia pursotuspään kanssa. Alla on kuvailtu kappaleen kulku STL-tiedostosta pursotuspäälle ja Robodrillille kelpaavaksi G-koodiksi.

PC-laitteen ohjelma on luotu Microsoft Visual Basic-ohjelmalla ja sen päätekijänä on ollut koulun yliopettaja Eero Korhonen. PC-laitteen ohjelman koodi löytyy myös tämän opinnäytetyön liitteenä 9.

7.2.1 STL-tiedostosta G-koodiin

Pikamallinnusprosessi alkaa aina kolmioidusta STL-tiedostosta. Kun valitulla 3D-mallinnusohjelmalla on luotu halutunlainen kappale, se pitää kääntää STL-muotoon. Kun STL-muotoinen tiedosto on luotu, voidaan se viipaloida ja muuttaa sopivaksi G-koodiksi valitulla ohjelmalla. Kuvassa 23 on opinnäytetyössä käytetty MakerBotin ReplicatorG-ohjelma generointivaiheessa.



KUVA 23. G-koodin generointi ReplicatorG-ohjelmalla

Ensimmäisenä vaiheena STL-tiedosto tuodaan ohjelmaan. Se asetellaan sopivasti siihen kohtaan pursotuspöytää, mihin se halutaan. Kun kappale on oikealla paikalla ja oikein skaalattu, voidaan G-koodin generointi aloittaa. ReplicatorG kysyy yleisiä parametrejä, kuten tulostuslämpötilaa ja kerrospaksuutta. Kun parametrit on annettu, voidaan G-koodi generoida. Kuvassa 24 on lyhyt osa ReplicatorG-ohjelmalla luotua koodia.

```

(**** beginning of start.txt ****)
(This file is for a MakerBot Thing-O-matic with)
(an automated build platform)
(This file has been sliced using skeinforge 35)
(**** begin initialization commands ****)
G21 (set units to mm)
G90 (set positioning to absolute)
M108 S255 (set extruder speed to maximum)
M104 S225 T0 (set extruder temperature)
M109 S125 T0 (set heated-build-platform temperature)
(**** end initialization commands ****)
(**** begin homing ****)
G162 Z F500 (home Z axis maximum)
G161 X Y F2500 (home XY axes minimum)
G92 Z80 ( ---- set Z axis maximum ---- )
G92 X-57.5 Y-57 (set zero for X and Y)
(**** end homing ****)
(**** begin pre-wipe commands ****)
G1 X52.0 Y-57.0 Z10.0 F3300.0 (move to waiting position)
M6 T0 (wait for toolhead parts, nozzle, HBP, etc., to reach temperature)
M101 (Extruder on, forward)
G04 P5000 (wait t/1000 seconds)
M103 (Extruder off)
(**** end pre-wipe commands ****)
(**** end of start.txt ****)
(<extruderInitialization>)
(<decimalPlacesCarried> 3 </decimalPlacesCarried>)
(<flowrateCubicMillimetersPerSecond> 5.866 </flowrateCubicMillimetersPerSecond>)
(<layerThickness> 0.36 </layerThickness>)
(<extrusionPerimeterWidth> 0.468 </extrusionPerimeterWidth>)
(<outsideExtrudedFirst> True </outsideExtrudedFirst>)
(<feedrateMinute> 2046.0 </feedrateMinute>)
(<operatingFeedratePerSecond> 34.1 </operatingFeedratePerSecond>)
(<orbitalFeedratePerSecond> 34.1 </orbitalFeedratePerSecond>)
(<supportFlowrate> 255.0 </supportFlowrate>)
(<travelFeedratePerSecond> 55.0 </travelFeedratePerSecond>)
(<extrusionwidth> 0.522 </extrusionwidth>)
(<fillInset> 0.4167 </fillInset>)
(<infillBridgewidthoverExtrusionwidth> 1.0 </infillBridgewidthoverExtrusionwidth>)
(<procedureDone> carve </procedureDone>)
(<procedureDone> inset </procedureDone>)
(<procedureDone> fill </procedureDone>)
(<procedureDone> multiply </procedureDone>)
(<procedureDone> speed </procedureDone>)
(<procedureDone> comb </procedureDone>)
(<procedureDone> clip </procedureDone>)
(<procedureDone> oozebane </procedureDone>)
(<procedureDone> fillet </procedureDone>)
(<procedureDone> raftless </procedureDone>)
(<procedureDone> export </procedureDone>)
(</extruderInitialization>)
M108 S255.0
(<layer> 0.18 )
(<surroundingLoop>)
(<boundaryPoint> X-1.706 Y-2.734 Z0.18 )
(<boundaryPoint> X-17.516 Y-2.734 Z0.18 )
(<boundaryPoint> X-17.516 Y-15.766 Z0.18 )
(<boundaryPoint> X-1.706 Y-15.766 Z0.18 )
(<perimeter>)
G1 X-40.0 Y40.0 Z0.18 F3300.0
M101
G1 X-40.0 Y-2.73 Z0.18 F900.0
G1 X-3.98 Y-2.73 Z0.18 F900.0
G1 X-17.43 Y-2.73 Z0.18 F900.0
G1 X-17.52 Y-2.83 Z0.18 F900.0
G1 X-17.52 Y-15.68 Z0.18 F900.0
G1 X-17.43 Y-15.77 Z0.18 F900.0
G1 X-1.8 Y-15.77 Z0.18 F900.0
G1 X-1.71 Y-15.68 Z0.18 F900.0
G1 X-1.71 Y-2.83 Z0.18 F900.0
G1 X-1.8 Y-2.73 Z0.18 F900.0
G1 X-3.7 Y-2.73 Z0.18 F900.0
G1 X-3.85 Y-2.81 Z0.18 F900.0
M103
(</perimeter>)
(</surroundingLoop>)
(<surroundingLoop>)
(<boundaryPoint> X-15.337 Y-8.247 Z0.18 )
(<boundaryPoint> X-14.158 Y-6.624 Z0.18 )
(<boundaryPoint> X-12.25 Y-6.004 Z0.18 )
(<boundaryPoint> X-10.342 Y-6.624 Z0.18 )
(<boundaryPoint> X-9.163 Y-8.247 Z0.18 )
(<boundaryPoint> X-9.163 Y-10.253 Z0.18 )
(<boundaryPoint> X-10.342 Y-11.876 Z0.18 )

```

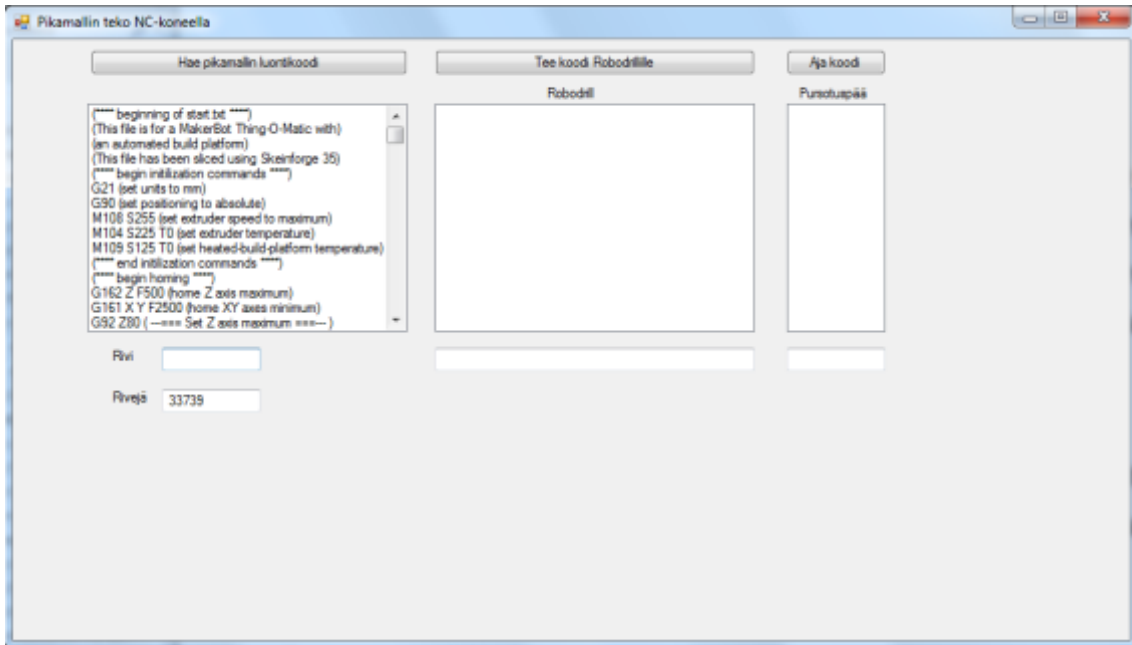
KUVA 24. ReplicatorG-ohjelmalla saatu G-koodi

Koodista huomataan, että ReplicatorG:n luomassa koodissa on myös paljon niin sanotusti turhia käskyjä, jotka vaikuttavat vain MakerBotin omissa tuotteissa.

ReplicatorG-ohjelman muodostama G-koodi tallennetaan tekstitiedostoksi, josta se voidaan siirtää PC-laitteen kääntöohjelmaan.

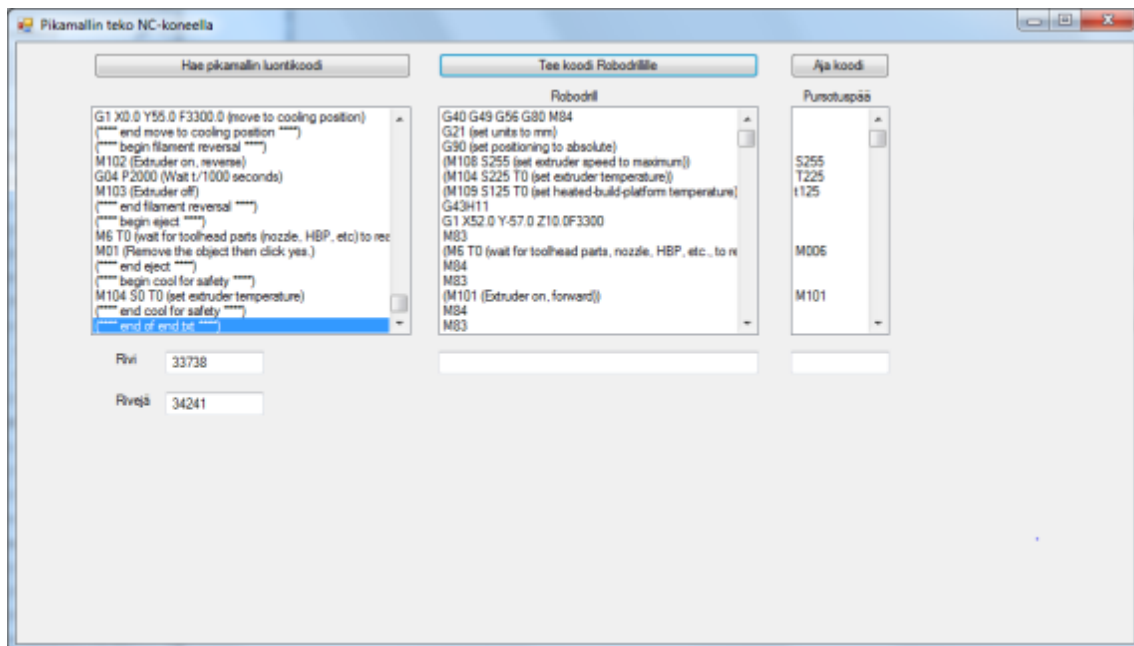
7.2.2 PC-laitteen ohjelman toiminta

Kun STL-tiedosto on saatu muutettua G-koodiksi ja G-koodi on tekstitiedostona, voidaan PC-laitteen ohjelmaa alkaa käyttämään. Kuvassa 25 on kuva ohjelmasta ja siihen haetusta G-koodista.



KUVA 25. PC-laitteen ohjelma ja siihen haettu G-koodi

Ohjelmaan on nyt haettu koodi, jossa on rivejä 33739. Seuraavana vaiheena napsautetaan *Tee koodi Robodrilille* -painiketta, jolloin ohjelma käy koko G-koodin läpi, poistaa turhat rivit ja jakaa Robodrilille ja pursotuspäälle menevät G-koodit erilleen. Kuvassa 26 koodi on läpikäytynä.



KUVA 26. G-koodi läpikäytynä

Kun koodi on jaettu, on G-koodi valmis ajettavaksi laitteistolle. Kuvasta nähdään, että suuri määrä turhia rivejä on myös poistunut. Ohjelmassa on myös ruutu, joka seuraa pursotuspäältä sarjaporttiin tulostettavia viestejä.

8 LOPPUTULOKSET JA PROTOTYYPIN TESTAUS

Työn lopputuloksena saatiin pursotinpään kiinnitysmekanismin tekniset piirustukset ja 3D-malli, sähkökaavio pursotuspään ohjauselektronikalle, ohjelmat Arduino- ja PC-ohjaukseen sekä toimiva prototyyppi testauskäyttöön. Lisäksi työssä ehdittiin testata valmistettua prototyyppiä, joka oli tärkeä osa kyseistä opinnäytetyötä.

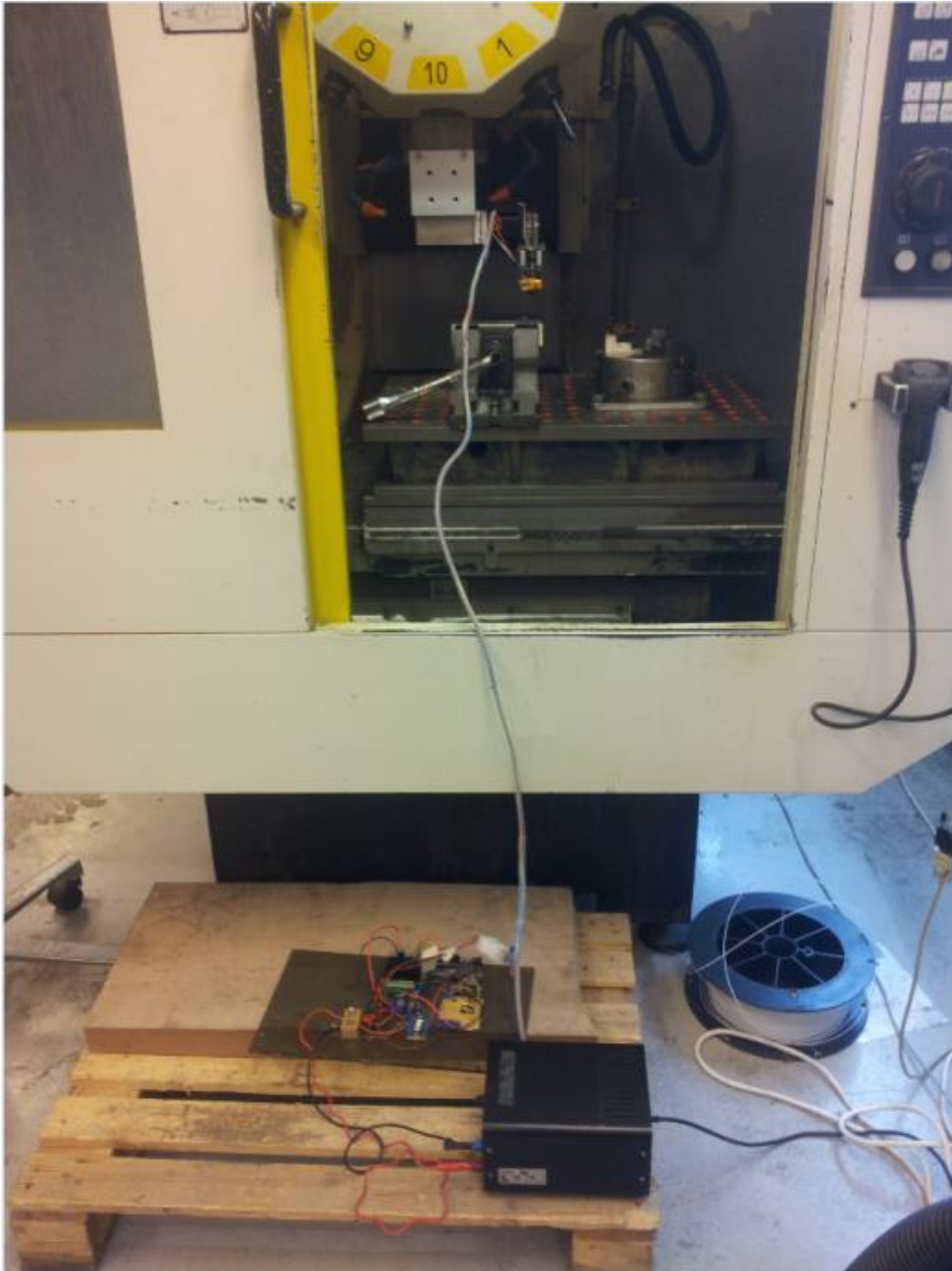
8.1 Prototyypin testaus

Kun pursotuspäälle tarvittava kiinnitysmekanismi, elektroniikkakomponentit ja sähkökytkennät sekä tarvittavat ohjelmat olivat valmiina, voitiin pursotuspäätä kokeilla Robodrill-koneistuskeskuksessa. Laitteet kytkettiin kuvan 10 laitekaavioon mukaan.

Ennen testausta toimitettiin seuraavat asiat:

- Robodrillin siirto remote-tilaan
- Robodrillin nollapisteen siirtäminen sopivalle kohdalle pursotusalustaa
- Z-lähtötason säätö
- pursotuspään virtalähteen kiinnittäminen
- USB- ja RS-232-C -väylien kiinnitys PC-laitteeseen
- pursotuspään kiinnitys työkalupaikalle.

Kuvassa 27 nähdään prototyyppitestauksen kokoonpano.



KUVA 27. Prototyypivaiheen testaus

Prototyypivaiheessa elektroniikka ja materiaalikerä sijoitettiin kaapin ulkopuolelle, johtuen lähinnä suuresta virtalähteen koosta ja puuttuvasta materiaalikerän telineestä. Lopullisessa versiossa nämä sijoitetaan suojakopin sisäpuolelle. Kun kaikki komponentit oli saatu paikalleen, voitiin laitetta testata.

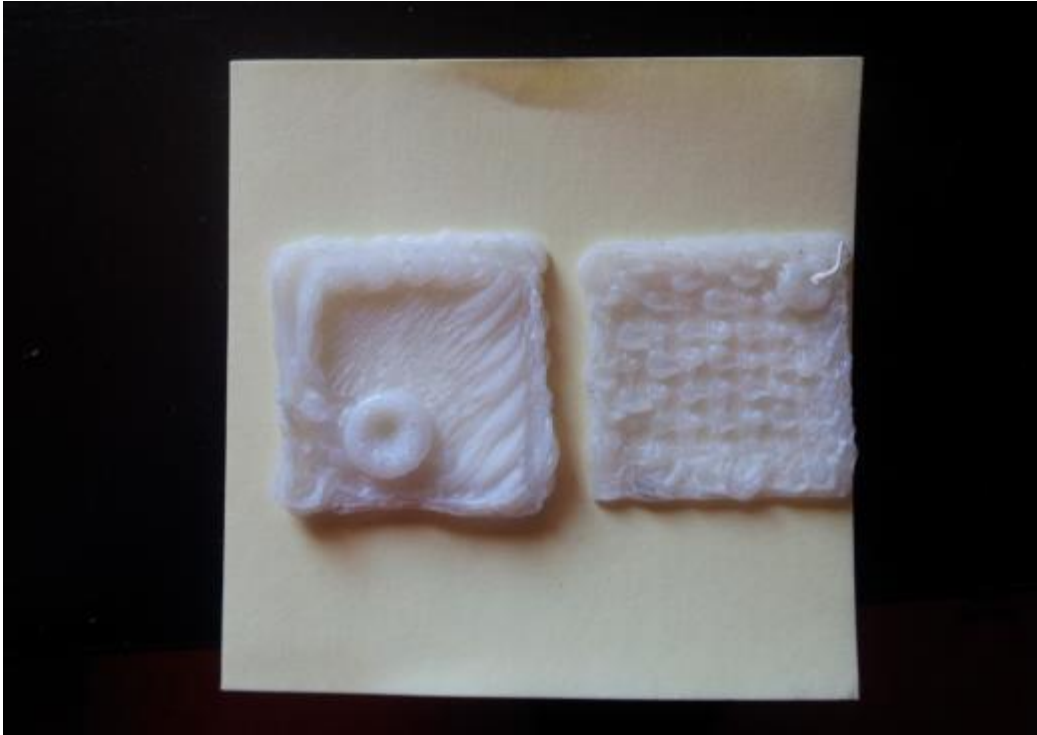
8.2 Prototyypin testauksen tulokset

Prototyyppiä testattiin yksinkertaisella 30 x 30 mm neliökappaleella, jolloin nähtäisiin miten varsinainen pursotus onnistuu. Kappale ajettiin ensin MakerBotin ReplicatorG-ohjelman läpi, joka suoritti kappaleen viipaloinnin ja muuntamisen G-koodiksi. Tämän jälkeen koodi siirrettiin PC-laitteelle, jonka sarjaporttiin oli liitetty pursotuspään komponentit. PC-laitteella koodi ladattiin tekstitiedostosta käänös-ohjelmaan, joka muokkaa G-koodin Robodrill-yhteensopivaksi sekä erottelee erilleen pursotuspäälle ja Robodrillille menevät komennot.

Pursotuspään ja Robodrillin yhtäaikainen ohjaus toimi kiitettävästi, ja molemmat pysyivät toimivan kättelyn ansiosta hyvin synkronoinnissa. Pursotuspään moottorin ohjaus, lämmityksen ohjaus ja pursotuksen valmiudesta ilmoitus toimivat hyvin. Myös pursotuksen ON/OFF-komennot toimivat toivotulla tavalla.

Suurimmaksi ongelmaksi prototyyppiä testatessa osoittautui Robodrillin hitaus peräkkäisissä liikkeissä. Pikamallinnuslaitteissa pursotusnopeus on yleensä vakio, ja tasojen nopeutta säätämällä saadaan oikea määrä pursotettua ainetta. Vaikka Robodrillille tässä tapauksessa annetaan sama syöttönopeus kuin annettaisiin vastaavalle MakerBotin pikamallinnuslaitteellekin, se joutuu käyttämään hidastus- ja kiihdytysramppeja jokaisen liikkeen alussa ja lopussa sen liikkuvien tasojen suuren massan takia. Myös seuraavaa käskyä odottaessa Robodrillilla aiheutuu pieni viive. Näistä syistä pursotuspää ehtii pursottaa liikaa sulaa ABS-muovia käänös- ja reunakohdissa, jolloin rakenteesta ei tule tasainen. Kun osassa paikkaa sulaa muovia on enemmän, alkaa pursotin pursottaa epätasaisesti eikä kunnollista jälkeä saada.

Pursotusnopeutta koetettiin muutamalla eri nopeudella, mutta tämä ei juuri auttanut ongelmaan. Hitaammalla nopeudella saatiin hieman parempaa jälkeä, koska sulaa muovia ei ehtinyt kertyä reuna-alueille yhtä paljon. Kuvassa 28 nähdään vasemmalla nopeammalla ja oikealla hitaammalla nopeudella pursotetut koekappaleet. Koekappaleita ei voitu jatkaa loppuun asti koska kappaleen pinnanmuodosta tuli liian epätasainen ja pursotuspään suutin alkoi törmätä muoviiin.



KUVA 28. Testipursotuksia joista vasen nopealla, oikea hitaalla nopeudella

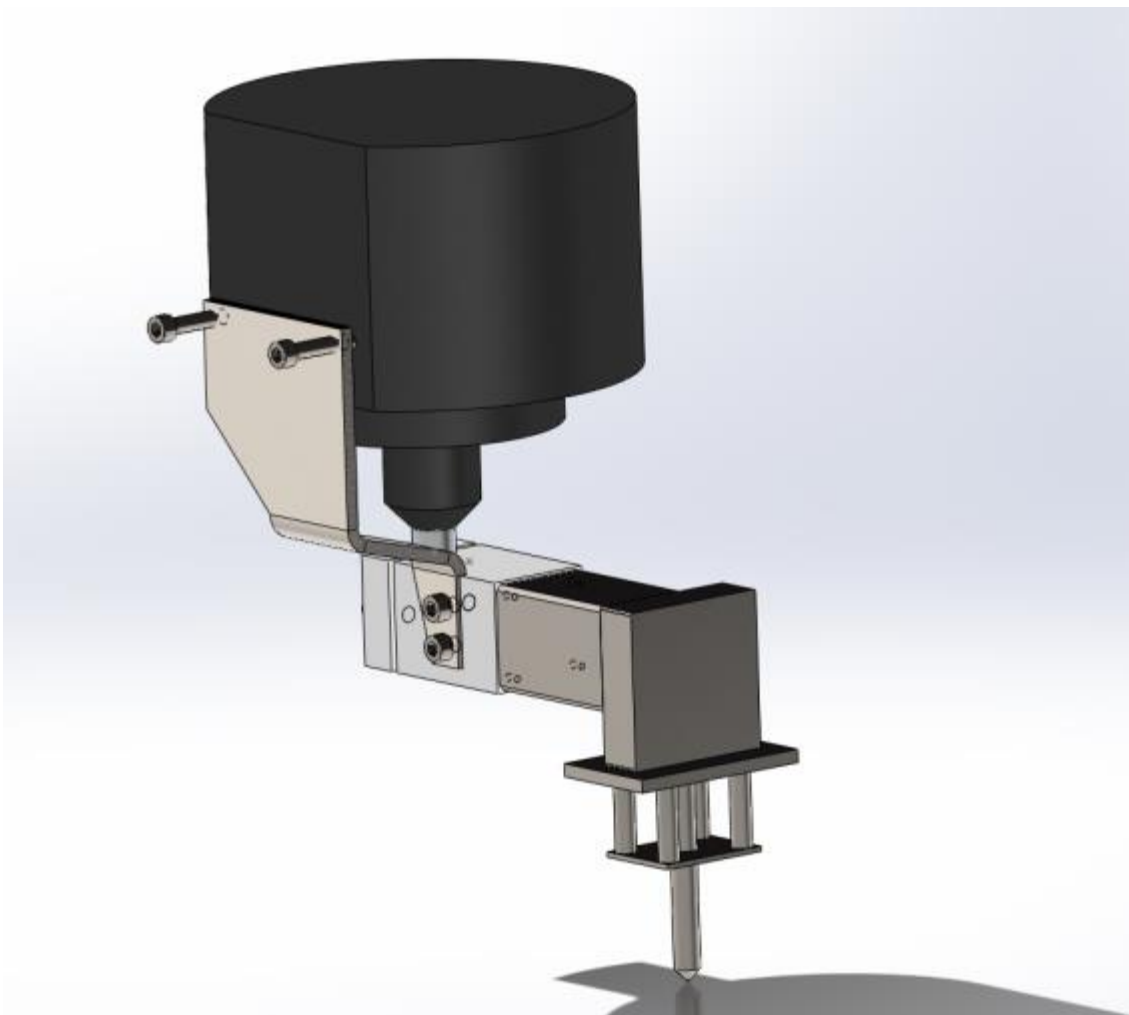
Ongelmaan ei keksitty ratkaisua, jolla robodrill olisi saatu liikkumaan sujuvammin. Jokainen pysäytys ja hidastus kerää liikaa sulaa muovia yhteen kohtaan, jolloin lopputuloksesta ei tule tarpeeksi tasainen. Nopeusalueen säädöllä ei saatu juurikaan parannettua tulosta, koska virhe tapahtuu Robodrillin tasojen ollessa paikallaan, eikä silloin, kun tasot liikkuvat.

Ratkaisuna ongelmaan olisi pursotusta säätelevän askelmoottorin parametrien ja ohjelman säätäminen niin, että pursotusnopeus säätäisi myös suhteessa syöttönopeuteen. Kun liike alkaa tulla loppuunsa, pursotusnopeuden pitäisi myös hidastua ja pysähtyä vastaavaksi hetkeksi kuin Robodrill pysähtyy liikkeiden välillä. Tähän ei kuitenkaan enää ehditty perehtyä, koska sen toiminnan ohjelmallisesti tekeminen ja parametrien testaaminen olisi vienyt toisen opinnäytetyön verran aikaa.

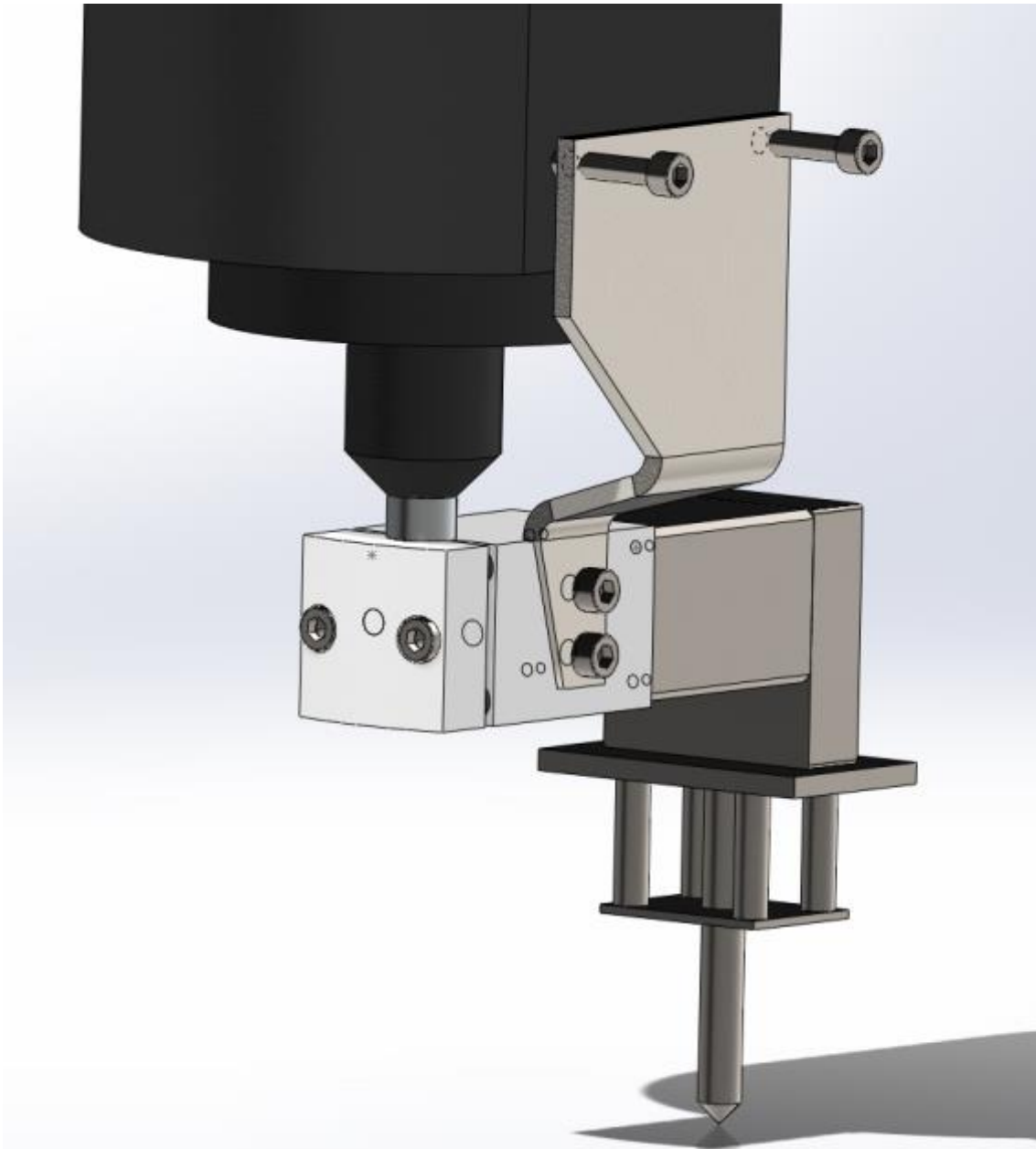
8.3 Pursotinpään kiinnityksen mekaniikkasuunnittelu

Pursotinpään kiinnitykseen tehtiin prototyyppivaiheessa alumiinista koneistamalla nopea versio, jotta testaukset saatiin tehtyä. Osana opinnäytetyötä suunniteltiin myös kunnollinen kiinnitinsarja pursotuspäälle. Pursotuspään kiin-

nittimen 3D-malli on esillä kuvissa 29 ja 30 ja sen tekniset piirustukset ovat liitteinä 3–7.



KUVA 29. Pursotuspään kiinnitysmekanismi



KUVA 30. Pursotuspään kiinnitysmekanismi

Pursotuspään kiinnitysmekanismi suunniteltiin valmistettavaksi FDM-tekniikkaa käyttävällä pikamallinnuskoneella. Koska itse kiinnitysmekanismi on vain pursotuspään paikallaan pitävä rakenne, eikä siihen kohdistu rasituksia, on aivan turhaa koneistaa kiinnitysmekanismia metallista tai kovemmasta materiaalista. Materiaalina käytetään ABS-muovia. Työkalupaikalle voidaan jättää 16 mm tappi, johon kiinnitin voidaan helposti pultata kiinni M6-ruuveilla. Samoin tukilevy on suoraan pultattavissa kiinnittimen kylkeen M6-ruuveilla. Tuennan säätö tehdään kierteitettyissä rei'issä olevilla M6-ruuveilla. Tukilevy valmistetaan särmämällä S235-teräksestä.

9 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli tutkia ja valmistaa pikamallinnustyökalu NC-työstökeskukseen, jolloin NC-työstökeskusta voitaisiin käyttää pikamallinnuslaitteena pelkällä työkalun vaihdolla. Työ tehtiin Pireplan OY:lle. Pikamallinnuslaitteet ovat tulleet nykypäivänä huomattavasti yleisemmiksi ja niiden hinta on alentunut, mutta vastaavaa NC-työstökeskusta hyödyntävää pikamallinnustyökalua ei markkinoilla ole.

Työ aloitettiin tutustumalla erilaisiin pikamallinnustekniikoihin, valmiiden pikamallinnuslaitteiden rakenteisiin, G-koodiin sekä Arduinoon ja muihin mikrokontrolloreihin. Koneistuskeskusympäristön rajoitteiden takia ainoaksi vaihtoehdoksi pikamallinnustyökalun tekniikaksi jäi FDM eli pursottava tekniikka. Tekniikan valinnan jälkeen tehtiin alustava suunnitelma koko laitekaaviosta, erilaisista tarvittavista komponenteista sekä pursotuspään mekaanisesta kiinnityksestä Robodrillin työkalupaikalle.

Kun tarvittava kokoonpano elektroniikkoineen oli koottu, voitiin Arduinolle kehittää ohjelma, jolla pursotuspäätä ohjattiin. Pursotuspään ohjelmoinnin tekeminen vaati perehtymistä C++-ohjelmointiin. Eero Korhonen ohjelmoi PC-laitteelle tulevan ohjelman, joka muokkasi G-koodin Robodrillille sopivaksi, sekä jakoi käskyt erikseen Robodrillille ja pursotuspäälle meneviin komentoihin. Ohjelma sisälsi myös kättelyt, joiden avulla pursotuspään ja Robodrillin liikkeet saatiin synkronoitua. Elektroniikkapuolella jouduttiin myös tekemään pieniä muutoksia, joita olivat askelmoottorin ohjaimen vaihto sekä virtalähteen vaihto 24 V jännitteellä toimivaan. Prototyypin testausta varten valmistettiin alumiinista koulun tiloissa alustava kiinnitysmekanismi, jolla voitiin testata prototyypin toimivuutta.

Prototyyppi saatiin testattua ja sen toiminta oli Robodrillin hitaahkoja liikkeitä lukuun ottamatta hyvää. Robodrill sekä pursotinpää tottelevat käskyjä mainiosti ja pysyvät hyvässä synkronoinnissa kättelyistä johtuen. Itse tulostettavan mallin laatu jäi kuitenkin heikoksi Robodrillin omien syöttöparametrien vuoksi. Robodrillin X- ja Y-tasaille saatiin säädettyä sama syöttö- eli liikuntanopeus kuin Markerbotin tasolle, mutta koska liikkuvien tasojen massa on Robodrillissa suuri, se

ei pysty tekemään tarpeeksi nopeita peräkkäisiä liikkeitä, vaan se joutuu pysähtyilemään liikkeiden välillä. Syöttönopeudessa on myös kiihdytys- ja hidastusrampit, joiden takia Robodrill kiihdyttää ja hidastaa kaikkien liikkeiden alku- ja loppupäässä.

Työtä tehtäessä saatiin kuitenkin hyvää pohjaa ja tietotaitoa NC-työstökeskukseen sovellettavista uusista työkaluista, joita ohjataan samanaikaisesti NC-laitteen kanssa. Vaikka laite ei pikamallinnuskäytössä ollut vielä prototyyppivaiheessa täysin toimiva, voidaan siitä parametreja säätämällä saada vielä täysin funktionaalinen laite. Opinnäytetyö toimii myös hyvänä pohjana muunlaisille työkaluille, jotka voidaan liittää NC-työstökeskukseen. Esimerkiksi ylimääräisen kallistusakselin lisääminen koneistukseen voidaan tehdä tasoon käyttäen askelmoottoreita kallistukseen, Arduinoa ohjaukseen sekä PC-laitetta kättelyiden ja ohjelman hallitsemiseen. Myös erillisen työkalupään ohjaus on opinnäytetyön pohjalta helppoa, esimerkiksi kappaleiden saumoihin pursotettava liima tai tiiviste voitaisiin tehdä koneistaessa työkalupäällä.

Opinnäytetyö oli kokonaisuudessaan antoisa, mielenkiintoinen ja monipuolinen. Opinnäytetyötä tehdessä jouduin perehtymään mekaniikkasuunnitteluun, ohjelmointiin, NC-laitteiden ohjaukseen sekä elektroniikan komponentteihin.

LÄHTEET

1. Chua, CK 2010. Rapid prototyping, third edition. Signapore: World Scientific Publishing CO.
2. Liu, Frank W 2008. Rapid prototyping and engineering applications. New York: CRC press. Saatavissa:
http://www.wosco.org/books/Informatics/Rapid_Prototyping_and_Engineering_Applications.pdf Hakupäivä 17.1.2013
3. CAD-mallista pikamalliksi. 2003. Oulun seudun ammattikorkeakoulu, tekniikan yksikkö. Saatavissa:
<http://www.oamk.fi/tekniikka/pikamallitekniikka/CAD- ja RP-tekniikka.php?PHPSESSID=a64de4d542060fd4d92678db55fb236d> . Hakupäivä: 17.1.2013
4. Syrjälä, S. 1997. Rapid prototyping - Mallien, prototyyppien ja työkalujen pikavalmistus. Teknologian kehittämiskeskus: Teknologia katsaus 52/97.
5. Fused Deposition Modeling. 2008. CustomPartNet. Saatavissa:
<http://www.custompartnet.com/wu/fused-deposition-modeling> . Hakupäivä: 24.1.2013
6. Pikkarainen, Eero 1999. NC-tekniikan perusteet. Helsinki: Hakapaino OY
7. Numeerinen ohjaus. 2013. Wikipedia. Saatavissa:
http://fi.wikipedia.org/wiki/Numeerinen_ohjaus. Hakupäivä 7.2.2013.
8. Fanuc Robodrill α -T10B For Europe Operator's manual. 1994. FANUC LTD.
9. MakerBot Industries. 2013. Wikipedia. Saatavissa:
http://en.wikipedia.org/wiki/MakerBot_Industries . Hakupäivä 25.3.2013.
10. Avoin lähdekoodi. 2013. Wikipedia. Saatavissa:
http://fi.wikipedia.org/wiki/Avoin_l%C3%A4hdekoodi . Hakupäivä 25.3.2013.

11. Stepstruder MK7. Robosavvy. Saatavissa: http://robosavvy.com/store/images/StepstruderMK7_main.jpg. Hakupäivä 14.5.2013.
12. Arduino. 2013. Wikipedia. Saatavissa: <http://en.wikipedia.org/wiki/Arduino> . Hakupäivä 24.4.2013.
13. Thing-O-Matic MK6 Plus Electronics. Flickr. Saatavissa: <http://www.flickr.com/photos/makerbot/6211601628/>. Hakupäivä 4.2.2013.
14. Moilanen, Jani-Pekka 2012. Robotisoitu pikamallilaite. Oulu: Oulun seudun ammattikorkeakoulu, tekniikan osasto. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2012052910543>. Hakupäivä 23.5.2013.
15. EM-121 Askelmoottorihjain 40V 1A. Tuoteseloste. Turku: Electromen Oy.
16. ASCII. 2013. Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/ASCII>. Hakupäivä 15.5.2013.
17. Mäkelä, Mikko – Soininen, Lauri – Tuomola, Seppo – Öistämö, Juhani 2012. Tekniikan kaavasto. Tampere: Tammertekniikka.
18. Pulssileveysmodulaatio, PWM. 2013. Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/Pulssinleveysmodulaatio> . Hakupäivä 15.5.2013.

LIITTEET

Liite 1 Lähtötietomuistio

Liite 2 Sähkökaavio

Liite 3 Kiinnitin, pääkoonpano

Liite 4 Takakiinnitin

Liite 5 Etukiinnitin

Liite 6 Tappi

Liite 7 Tukilevy

Liite 8 Pursotuspään Arduinon koodi

Liite 9 PC-laitteen ohjelmakoodi

Liite 10 Lämpöanturin tekninen tiedosto

LÄHTÖTIETOMUISTIO

Tekijä Henri Pakanen _____

Tilaaja Pireplan Oy _____

Tilaajan yhdyshenkilö ja yhteystiedot Pasi Viitala 0407253800 _____


Työn nimi Pikamallinnustyökalun kehittäminen _____

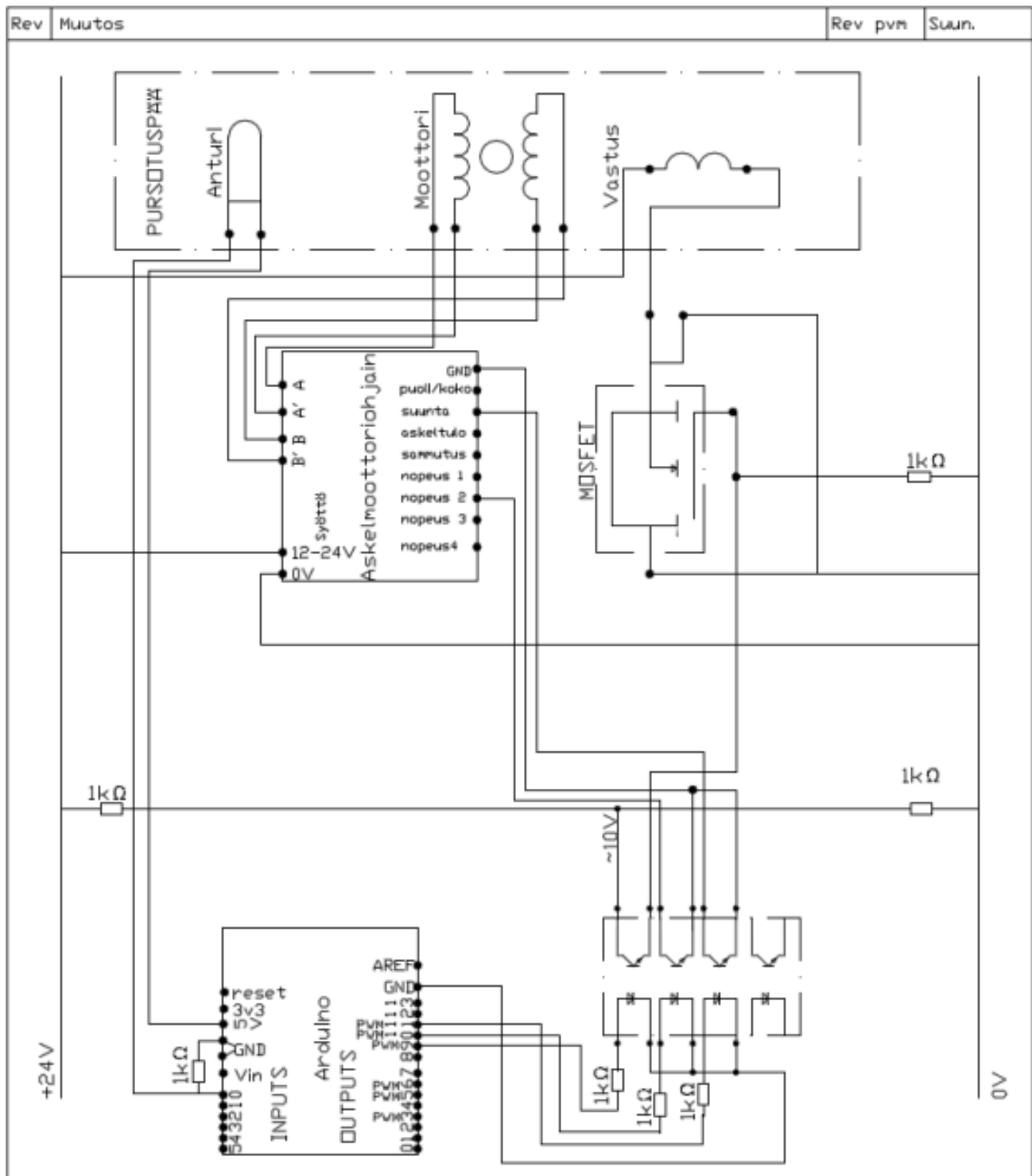
Työn kuvaus Kehittää Rep-Rap / Makerbot pikamalliteknoologiaan perustuva työkalu, jolla voidaan tulostaa pikamalleja työstökoneympäristössä. Laitteen prototyypivaiheessa käytetään ympäristönä Oamk:n robodrill-NC-jyrsinkonetta.

Työn tavoitteet

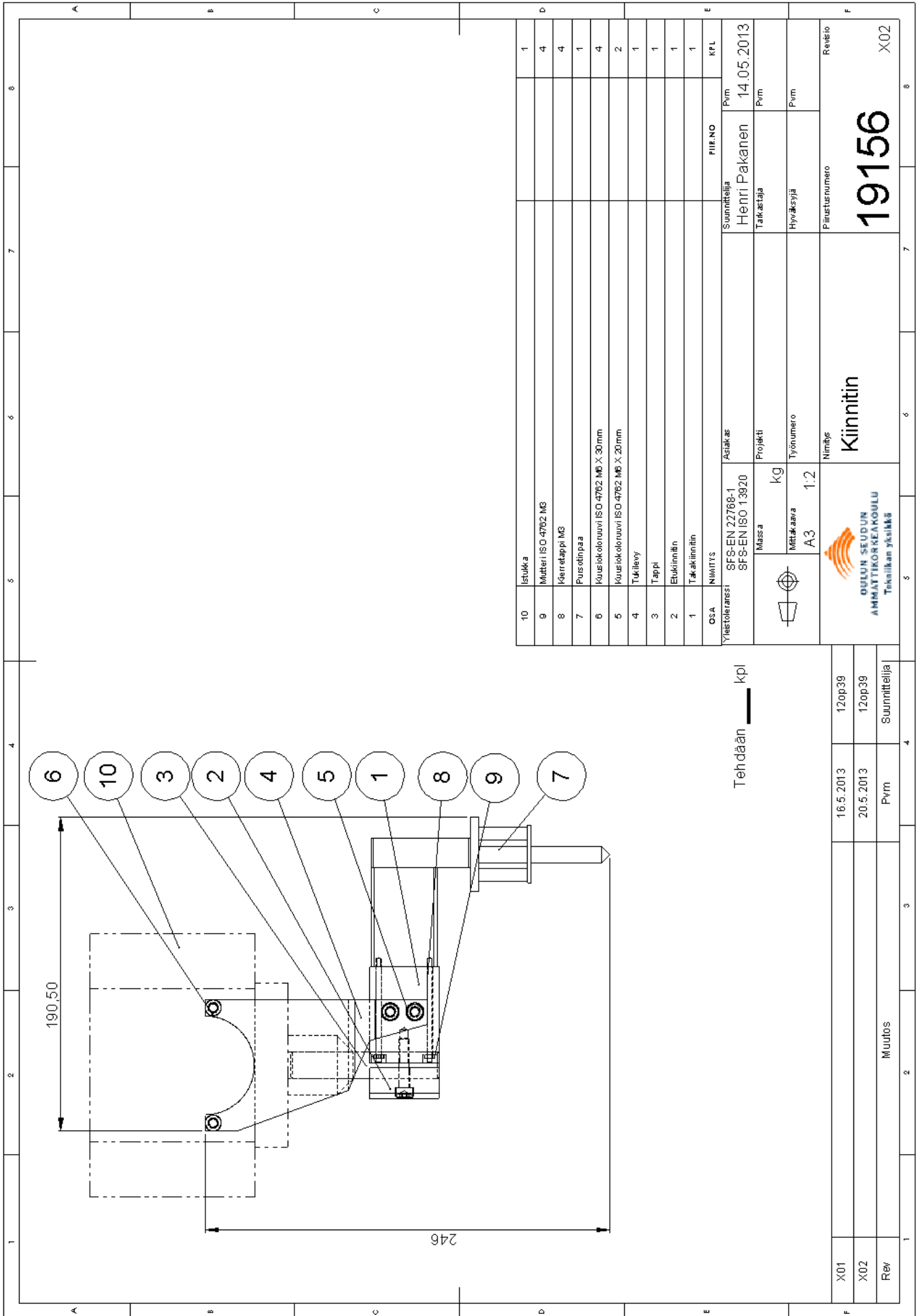
Saada suunniteltua ja testattua pikamallipää siten, että voidaan todeta menetelmän käyttökelpoisuus. Mekaniikkasuunnittelu tehdään 1.vaiheessa Robodrill ympäristöön sekä huomioidaan yleiskäyttöisyys. Pursotuspään ohjaus arduinolla. Replicator koodin yhteensopivuuden testaus Fanuc ohjauksen suhteen. Pursotuspään sekä Fanuc ohjauksen kommunikointi / kättelyn kuvaaminen.

Tavoiteaikataulu Työ alkaa 1.1.2013- 1.6.2013 _____

Päiväys ja allekirjoitukset 15.1.2013  _____ Henri Pakanen _____

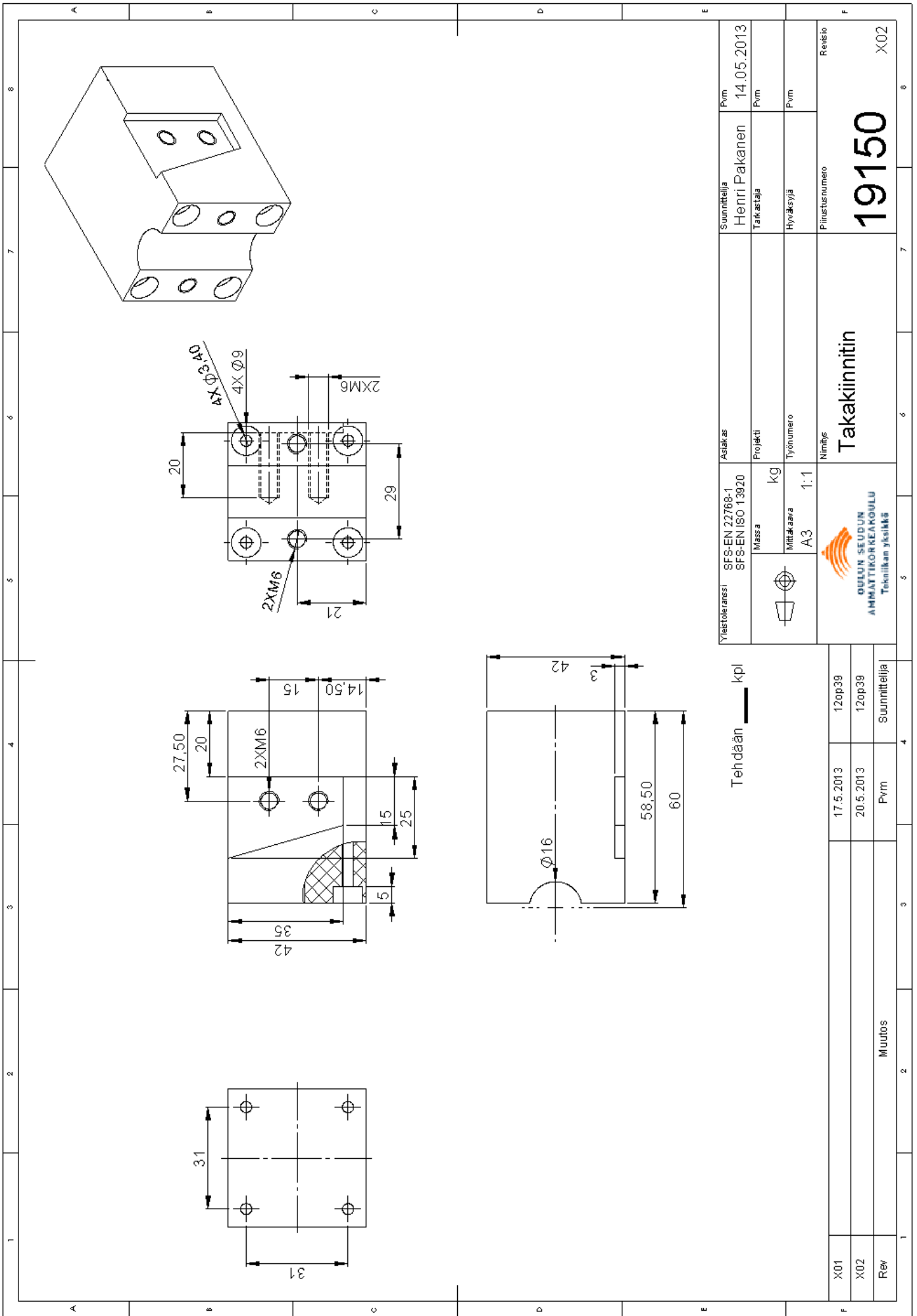


Rev	Muutos	Rev pvm	Suun.
Osa Piirustus nro.		Osan nimitys	Standardi, muoto, mitat, malli
Yleistoleranssit		Mittakaava	Laatu
		 Liittyy	Nimitys Pursotuspään sähkökaavio
Piirt.	16.05.13	HP	Prirustusnumero
Suun.	16.5.13	HP	
Tark.		JV	
OAMK Tekniikan Yksikkö			



Tehdään kpl

X01	16.5.2013	12op39
X02	20.5.2013	12op39
Rev	Pvm	Suunnittelija
	Muutos	

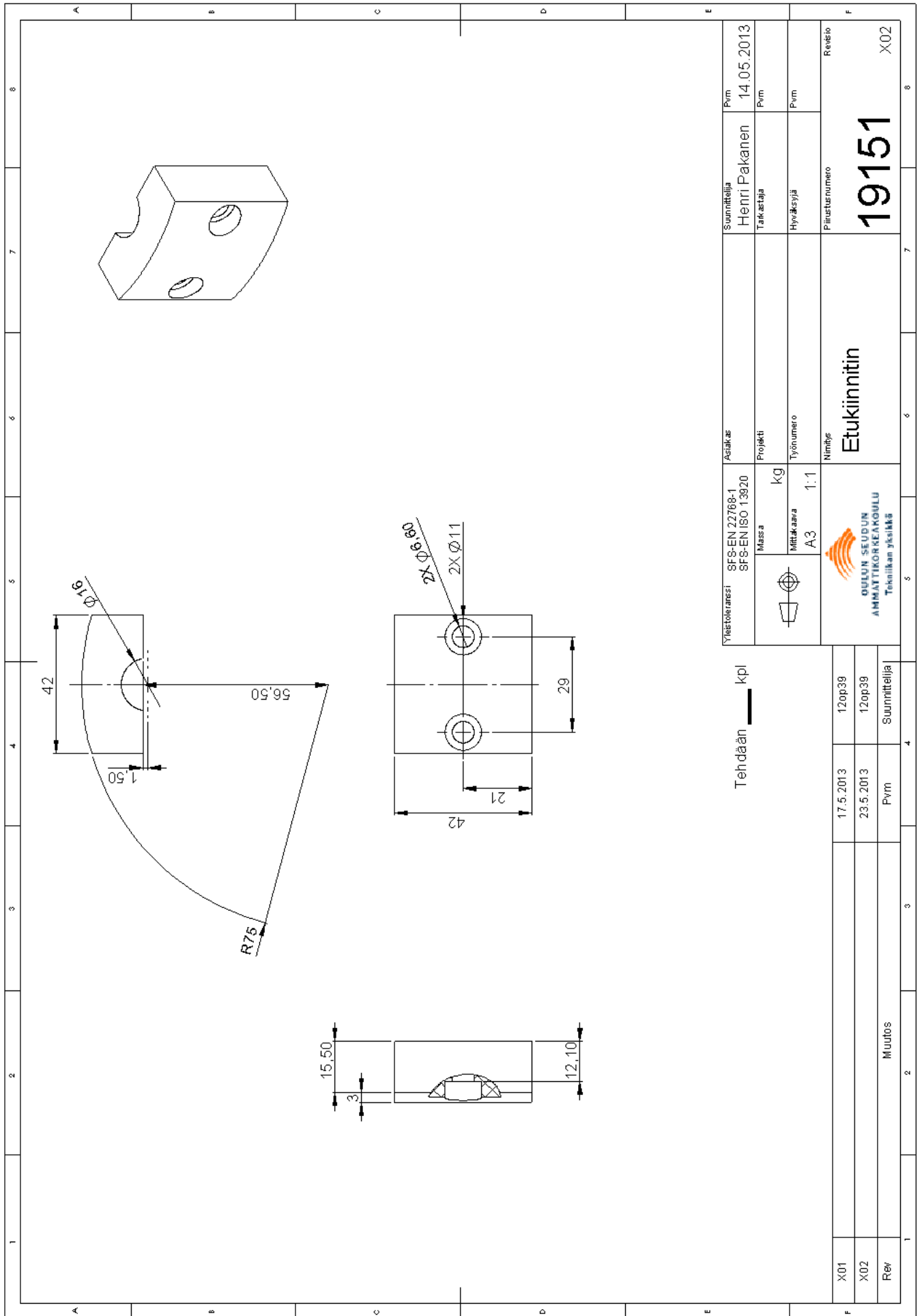


Tehdään _____ kpl

Ylietoleiäsi	SFS-EN 22768-1 SFS-EN ISO 13920	Asiakas	Suunnittelija	Pvm
			Henri Pakanen	14.05.2013
		Projekti	Talustaja	Pvm
		Työnumero	Hyväksyjä	Pvm
		Nimitys	Pilustusnumero	Revisio
			19150	X02



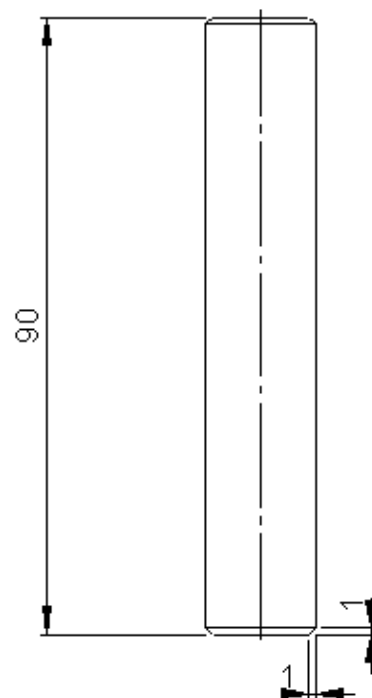
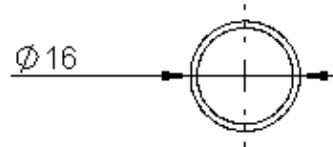
X01	17.5.2013	12op39	
X02	20.5.2013	12op39	
Rev		Pvm	Suunnittelija
			Muutos



Yleistoleranssi	SFS-EN 22768-1 SFS-EN ISO 13820	Asiakas	Suunnittelija Henri Pakanen	Pvm	14.05.2013
Massa	kg	Projekti	Tarkastaja	Pvm	
Mittakaava	A3	Työnumero	Hyväksyjä	Pvm	
	1:1	Nimitys	Pilustusnumero	Revisio	X02
 OULUN SEUDUN AMMATTIKORKEAKOULU Tekniikan yksikkö			19151		
			Etukiinnitin		

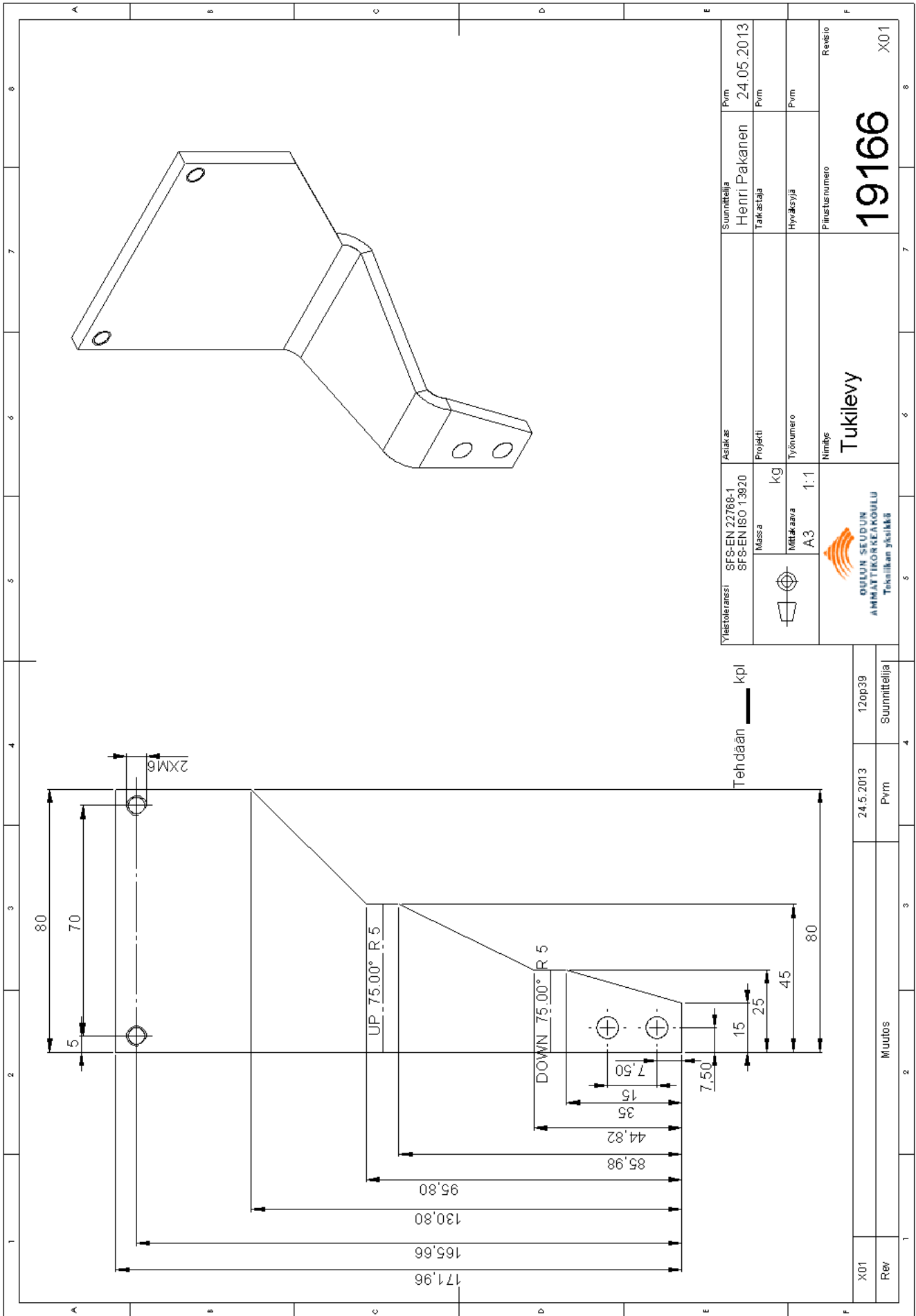
Tehdään _____ kpl		
X01	17.5.2013	12op39
X02	23.5.2013	12op39
Rev	Muutos	Suunnittelija

X01		17.5.2013	12op39
Rev	Muutos	Pvm	Suunnittelija



Tehdään _____ kpl

Yleistoleranssi SFS-EN 22768-1 SFS-EN ISO 13920		Asiakas	Suunnittelija Henri Pakanen	Pvm 14.05.2013
	Massa kg	Projekti	Tarkastaja	Pvm
	Mittakaava A4 1:1	Työnumero	Hyväksyjä	Pvm
		Nimitys Tappi	Piirustusnumero 19152	Revisio



Tehdään kpl


```
//Kaikki kaksoiskenoviivalla alkavat tekstit ovat kommentteja eivätkä vaikuta ohjelman toimintaan. Ohjelmassa on myös komentoja, jotka on asetettu kommentteiksi.
```

```
//Aluksi määritellään muuttujat
```

```
int tulevatavu,ensitavu,lukuarvo,toinentavu,kolmastavu,
int neljastavu,viidestavu,kuudestavu,temperature=0,keskiarvo=0, kerrat=0, pursotin=0;
int pursotinmuuttuja=0,targettemp=0;
int analogPin=0;
int vastusPin=9, moottoriPin=10, suuntaPin=11; // Output-pinnien määrittely
float R1=0;
float suhde=0,tempAnalog=0,vertailuarvo=0;
char merkki;
```

```
void setup()
```

```
{
pinMode(vastusPin, OUTPUT); // Asetetaan vastusPin Outputiksi
pinMode(moottoriPin, OUTPUT); // Asetetaan moottoriPin Outputiksi
pinMode(suuntaPin, OUTPUT); // Asetetaan suuntaPin Outputiksi
Serial.begin(9600); // Alustetaan sarjaliikenne 9600 baudiin eli 9600 bittiin sekunnissa
}
```

```
void loop() //Tätä luuppia kierretään
```

```
{
```

```
//digitalWrite(suuntaPin, HIGH); //Askelmoottorin suunnan kääntäminen tarvittaessa
```

```
tempAnalog = analogRead(analogPin); // TempAnalog-muuttujan arvo luetaan analogiatulosta
```

```
delay(100);
```

```
R1=((1000*(1024-tempAnalog))/tempAnalog); // R1-arvon laskeminen
```

```
vertailuarvo=180; // Vertailuarvo-muuttuja 180
```

```
if(R1<1000) vertailuarvo=(0.0002*R1*R1-0.338*R1 + 313.73); // Jos R1<1000 niin vertailuarvo lasketaan kaavalla
```

```
//Serial.println(vertailuarvo);
```

```
suhde=(vertailuarvo/temperature)*100; // Suhde on vertailuarvo jaettuna annetulla lämpötilan arvolla
```

```
if (suhde<95) //Jos suhde<95 on lämmityksen ohjearvo 255
```

```
{
```

```
ohje=255;
```

```
//Serial.println("heat 100%");
```

```
}
```

```
if (95<suhde && suhde<105) // Jos suhde on 95-105 on lämmityksen arvo kaavan ohje=20*(105-suhde) mukainen
```

```
{
```

```
ohje=20*(105-suhde);
```

```
//Serial.println("heat 80%");
}
if (suhde>105) // Jos suhde on yli 105 on lämmityksen ohjearvo 0
{
ohje=0;
Serial.println("heat off%");
}

if (suhde>100) //Jos suhde on yli 100, on pursotuspää pursotusvalmis
{
targettemp=1;
}

analogWrite(vastusPin,ohje); // PWM-lähtöön annetaan edellisten IF-lauseiden mukaan lämmit-
tyksen ohjearvo

if(suhde<95) // Jos suhde on alle 95, pursotuspää ei ole pursotusvalmiudessa
{
targettemp=0;
}

if(targettemp==1) // Kun pursotin on pursotusvalmiudessa, annetaan sarjaporttiin teksti M6ok
{
Serial.println("M6ok");
}

if (Serial.available() > 3) // Jos portissa on tavuja, luetaan sisältö tavu kerrallaan
{
tulevatavu = Serial.read();
merkki=tulevatavu;
Serial.write(tulevatavu);
Serial.write(10);
if (tulevatavu==84) // Jos tuleva merkki on t eli ASCII 84

{
Serial.println("Pursottimen °C"); // Pursottimen muuttujalle temperature annetaan seuraavat
kolme tavua merkin T jälkeen
ensitavu = Serial.read();
toinentavu = Serial.read();
kolmastavu = Serial.read();
temperature=((ensitavu-48)*100+(toinentavu-48)*10+(kolmastavu-48)*1);
//Merkkien muuttaminen normaaliksi luvuksi ja tallentaminen muistipaikalle temperature
//Serial.print(ensitavu);
//Serial.print(toinentavu);
//Serial.print(kolmastavu);
Serial.println(temperature); //tulostetaan saatu lämpötilan arvo
}
}
```

```
    if (tulevatavu==77) // Jos tuleva tavu on M-kirjain, niin:
    {
//Serial.println("Pursottimen M-komento");
neljastavu = Serial.read();
viidestavu = Serial.read();
kuudestavu = Serial.read();
pursotinmuuttuja=((neljastavu-48)*100+(viidestavu-48)*10+(kuudestavu-48)*1);
//Lukeman muuttaminen normaaliksi luvuksi ja tallentaminen muistipaikalle temperature
//Serial.print(ensitavu);
//Serial.print(toinentavu);
//Serial.print(kolmastavu);
// Serial.println("pursotinmuuttuja: ");
// Serial.println(pursotinmuuttuja);
    }

    if (pursotinmuuttuja==101) // Jos muuttujan pursotinmuuttuja-arvo on 101, niin:
    {
digitalWrite(suuntaPin, LOW); // Suuntalähtö on OFF-tilassa
pursotin=1; // pursotin=1 eli pursotin pursottaa
Serial.println("M1ok"); // Tulostetaan M1ok
    }

    if (pursotinmuuttuja==102) // Jos muuttujan pursotinmuuttuja-arvo on 102, niin:
    {
digitalWrite(suuntaPin, HIGH); // Suuntalähtö on ON-tilassa (pyörii toiseen suuntaan)
pursotin=1; // Pursotin=1 eli pursotin pursottaa
Serial.println("M2ok"); // Tulostetaan M1ok
    }

    if (pursotinmuuttuja==103) // Jos muuttujan pursotinmuuttuja-arvo on 103, niin:
    {
pursotin=0; // pursotin pois päältä
Serial.println("M3ok"); // Tulostetaan M1ok
    }

    if (pursotin==1 && targettemp==1) // Jos pursotin- ja targettemp-muuttujien arvo 1, pursotus
    voi alkaa
    {
digitalWrite(moottoriPin, HIGH); // Pursotin päällä
    }

    if (pursotin==0 || targettemp==0) // Jos pursotin- TAI targettemp-muuttuja on 0, ei purso-
    tusta
    {
digitalWrite(moottoriPin, LOW); // Pursotin pois päältä
    }
}
```

D:\NCrapid\NCrapid\Form1.vb

1

```
Public Class Form1
    Dim pitaapysayttaa = 0, lammitysmenossa = 0, saaAjaa = 0, puskuri = 20
    Dim Mvastaus As String
    Dim Pvastaus As String
    Dim Mkysymys As String

    Private Sub Hae_Click(sender As System.Object, e As System.EventArgs) Handles Hae.Click
        Dim Tiedosto As String
        ListBox1.Items.Clear()
        ListBox1.Visible = False

        Tiedosto = ""
        OpenFileDialog1.InitialDirectory = "c:\" 'asetetaan polku mistä haetaan
        OpenFileDialog1.Filter = "All files (*.*)|*.*" 'asetaan filtteri
        OpenFileDialog1.FilterIndex = 1
        OpenFileDialog1.RestoreDirectory = True
        If OpenFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK Then

            Tiedosto = OpenFileDialog1.FileName
            FileOpen(1, Tiedosto, OpenMode.Input) ' Avaa tiedosto
            While Not EOF(1) ' Loopataan tiedoston loppuun
                ListBox1.Items.Add(LineInput(1)) ' luetaan rivi listboxiin

            End While
            FileClose(1) ' file kiinni
            TextBox4.Text = Str(ListBox1.Items.Count)

        End If 'Löytyikö koodia

        ListBox1.Visible = True
    End Sub

    Private Sub TeeKoodi_Click(sender As System.Object, e As System.EventArgs) Handles TeeKoodi.Click
        If ListBox1.Items.Count > 0 Then

            Dim Rivi, Uusirivi, jono, vanhaZ, vanhaF, merkki As String
            Dim Spaikka, Glkasketty, puskuri, Glmenossa, M8lmenossa As Integer
            ListBox1.Visible = False
            ' ListBox2.Visible = False
            ListBox2.Items.Clear()
            ListBox3.Items.Clear()
            ListBox2.Items.Add("(RobodrillOhjelma)")
            ListBox3.Items.Add("")
            vanhaZ = ""
            ListBox2.Items.Add("G40 G80")
            ListBox3.Items.Add("")
            ListBox2.Items.Add("G49 G56")
            ListBox3.Items.Add("")
            vanhaZ = ""
            vanhaF = ""
            puskuri = Val(Trim(TextBox7.Text))
            Glkasketty = 0
            Glmenossa = 0
            M8lmenossa = 1

            For i = 0 To ListBox1.Items.Count - 1

                ListBox1.SelectedIndex = i
                TextBox1.Text = Str(i)
            End For
        End If
    End Sub
End Class
```

D:\NCrapid\NCrapid\Form1.vb

2

```

    TextBox1.Refresh()

    Rivi = ListBox1.Text

    If InStr(Rivi, "G1 ") > 0 Then
        Glmenossa = Glmenossa + 1
        If Glmenossa > puskuri Then
            ' ListBox2.Items.Add("M81" + Str(M81menossa))
            ' ListBox3.Items.Add("")
            ListBox2.Items.Add("M81")
            ListBox3.Items.Add("")
            Glmenossa = 0
            M81menossa = M81menossa + 1
        End If

        Spaikka = InStr(Rivi, "Z") 'etsitään Z ja vältetään toistoa
        Uusirivi = Rivi
        If Glkasketty = 0 Then 'liikkeiden aloitus
            ListBox2.Items.Add("G43H11")
            ListBox3.Items.Add("")
        End If

        Glkasketty = 1
        If Spaikka > 0 Then 'rivillä on Z
            Uusirivi = Mid(Rivi, 1, Spaikka - 1)

            jono = ""
            For j = Spaikka + 1 To Len(Rivi)
                merkki = Mid(Rivi, j, 1)
                If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then
Exit For 'tuli välilyönti tai rivinloppu tai sulku
                If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono +
merkki
                Next j

                If jono <> vanhaZ Then Uusirivi = Uusirivi + "Z" + jono
                vanhaZ = jono
            End If 'oliko Z

        Spaikka = InStr(Rivi, "F") 'etsitään Feedrate ja vältetään toistoa

        If Spaikka > 0 Then 'rivillä on Feedrate
            jono = ""
            For j = Spaikka + 1 To Len(Rivi)
                merkki = Mid(Rivi, j, 1)
                If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then
Exit For 'tuli välilyönti tai rivinloppu tai sulku
                If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono +
merkki
                Next j
                jono = Trim(Str(Int(Val(jono)))) 'muutetaan kokonaisluvuksi
                If jono <> vanhaF Then Uusirivi = Uusirivi + "F" + jono
                vanhaF = jono
            End If 'oliko F

            ListBox2.Items.Add(Uusirivi)
            ListBox3.Items.Add("")
        End If 'oliko G1

        If InStr(Rivi, "G04") > 0 Then
            Glmenossa = 0

            ListBox2.Items.Add("M81")
            ListBox3.Items.Add("")
            ListBox2.Items.Add("(" + Rivi + ")")
            ListBox3.Items.Add("")

            ListBox3.Items.Add("")

```

D:\NCrapid\NCrapid\Form1.vb

3

```

End If

If InStr(Rivi, "G21") > 0 Then
    ListBox2.Items.Add(Rivi)
    ListBox3.Items.Add("")
End If
If InStr(Rivi, "G90") > 0 Then
    ListBox2.Items.Add(Rivi)
    ListBox3.Items.Add("")
End If
If InStr(Rivi, "M6") > 0 Then
    Glmenossa = 0

    ListBox2.Items.Add("M81")
    ListBox3.Items.Add("")
    ListBox2.Items.Add("(" + Rivi + ")")
    ListBox3.Items.Add("M006")
    ' ListBox2.Items.Add("M84")
    ListBox3.Items.Add("")
End If

If InStr(Rivi, "M101") > 0 Then
    Glmenossa = 0

    ListBox2.Items.Add("M81")
    ListBox3.Items.Add("")
    ListBox2.Items.Add("(" + Rivi + ")")
    ListBox3.Items.Add("M101")
    ' ListBox2.Items.Add("M84")
    ListBox3.Items.Add("")
End If

If InStr(Rivi, "M102") > 0 Then
    Glmenossa = 0
    ListBox2.Items.Add("M81")
    ListBox3.Items.Add("")
    ListBox2.Items.Add("(" + Rivi + ")")
    ListBox3.Items.Add("M102")

    ' ListBox2.Items.Add("M84")
    ListBox3.Items.Add("")
End If
If InStr(Rivi, "M103") > 0 Then
    Glmenossa = 0
    ListBox2.Items.Add("M81")
    ListBox3.Items.Add("")
    ListBox2.Items.Add("(" + Rivi + ")")
    ListBox3.Items.Add("M103")

    ' ListBox2.Items.Add("M84")
    Glmenossa = 0
    ListBox3.Items.Add("")
End If
If InStr(Rivi, "M104") > 0 Then
    Glmenossa = 0
    ListBox2.Items.Add("(" + Rivi + ")")
    Spaikka = InStr(Rivi, "S")
    jono = "000"
    For j = Spaikka + 1 To Len(Rivi)
        merkki = Mid(Rivi, j, 1)

        If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then Exit For
    For 'tuli välilyönti tai rivinloppu tai sulku

        If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono + Trim
    (merkki)
    Next
    jono = Mid$(jono, Len(jono) - 2, 3)

```

D:\NCrapid\NCrapid\Form1.vb

4

```

        ListBox3.Items.Add("T" + Trim(jono))
    End If
    If InStr(Rivi, "M108") > 0 Then
        Glmenossa = 0
        ListBox2.Items.Add("(" + Rivi + ")")
        Spaikka = InStr(Rivi, "S")
        jono = ""
        For j = Spaikka + 1 To Len(Rivi)
            merkki = Mid(Rivi, j, 1)
            If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then Exit
For 'tuli välilyönti tai rivinloppu tai sulku

                If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono + merkki
            Next
            ListBox3.Items.Add("S" + Trim(Str(Int(Val(Trim(jono))))))
        End If
        If InStr(Rivi, "M109") > 0 Then
            Glmenossa = 0
            ListBox2.Items.Add("(" + Rivi + ")")
            Spaikka = InStr(Rivi, "S")
            jono = ""
            For j = Spaikka + 1 To Len(Rivi)
                merkki = Mid(Rivi, j, 1)
                If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then Exit
For 'tuli välilyönti tai rivinloppu tai sulku

                    If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono + merkki
                Next
                ListBox3.Items.Add("t" + Trim(jono))
            End If

        Next i
        ListBox2.Visible = True
        ListBox1.Visible = True
        TextBox4.Text = Str(ListBox2.Items.Count)
    End If 'oliko listaikkuna 1:ssä tavaraa
End Sub

Private Sub Aja_Click(sender As System.Object, e As System.EventArgs) Handles Aja.Click

    'Menettelytapa: Kun ajetaan NC-koodia aina M101 jne-koodirivin kohdalla syötetään
koodi pursotuspäälle
    'Samoin purtotuspäälle syötetään lämmön ohjearvo muodossa T220, eli T-kirjaimen
perässä ohjelämpö Celsiusasteina
    'Lämmityskäskyn M6 jälkeen jäädään odottamaan pursotuspäältä ilmoitusta lämmön
saavuttamisesta M6ok
    'Aina ennen noita M-koodeja lähetetään Robodrillille M81 ja jäädään odottamaan
Robodrillin Arduinolta vastausta,
    'että Robodrillin ohjelma on edennyt kyseiseen vaiheeseen. Kun Robodrill suorittaa
komennon M81, se kytkee
    'yhden lähtönsä transistorin johtavaksi noin 100 millisekunniksi. Tämä lähtö on
kytketty Arduinon tuloon.
    'Kun Robodrilliin kytketyn Arduinon ohjelma havaitsee tulon kytkettyneeksi, se
lähettää loogiseen USB-sarjaporttiinsa
    'viestin M81
    'Viestin M81 tultua ohjelman lähettämistä Robodrillille ja pursotuspäälle jatketaan
    'Pursotuspäälle menevät komennot ovat muotoa MXXX, jossa XXX on on lukuja vastaavina
ASCII-merkkeinä, M101 jne
    'Pursotuspää odottaa, että sille on tulossa vähintään 4 merkkiä ja lukee sitten M-
kirjainta seuraavat32 lukua
    'ja muodostaa niistä omaan käyttöönsä luvun
    'Lämmityksen kohdalla pursotuspää kaiuttaa M6ok, kun tavoite on saavutettu.
    'Tietokone lähettää Robodrillille G1-koodia korkeintaan ohjelman käynnistysvaiheessa
    'parametritiedostossa määritellyn määrän rivejä, esim 30, sitten M81-komennon ja jää
odottamaan kuittausta,että
    'ohjelma on edennyt M81-koodiin saakka. Näin vältetään Robodrillin muistin ylivuoto
    'Kun NC-ohjelmassa tulee suoritettavaksi viive G04, se suoritetaan PC:ssä tpahtuvana
NC-ohjelman

```

D:\NCrapid\NCrapid\Form1.vb

5

```
'syötön keskeyttämisenä viiveen ajaksi
ListBox2.SelectedIndex = 0
saaAjaa = 1
ListBox3.SelectedIndex = 0

Timer2.Enabled = True
```

```
End Sub
```

```
Private Sub Form1 FormClosing(sender As Object, e As System.Windows.Forms.
FormClosingEventArgs) Handles Me.FormClosing
    SerialPort1.Close()
    SerialPort2.Close()
    SerialPort3.Close()
```

```
End Sub
```

```
Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.
Load
```

```
    Dim tiedostonimi, portti1, portti2, portti3, aika1, aika2, aika3, rivimaara As String

    tiedostonimi = "C:\RoboRapid.txt"
    FileOpen(1, tiedostonimi, OpenMode.Input) ' Avaa tiedosto
    portti1 = LineInput(1) ' luetaan rivi
    portti2 = LineInput(1) ' luetaan rivi
    portti3 = LineInput(1) ' luetaan rivi
    aika1 = LineInput(1) ' luetaan rivi
    aika2 = LineInput(1) ' luetaan rivi
    aika3 = LineInput(1) ' luetaan rivi
    rivimaara = LineInput(1)
    TextBox7.Text = rivimaara
    FileClose(1) ' file kiinni
    SerialPort1.PortName = portti1
    SerialPort2.PortName = portti2
    SerialPort3.PortName = portti3
    SerialPort1.Open() 'robodrill
    SerialPort2.Open() 'robodrillin arduino
    SerialPort3.Open() 'pursotuspään arduino
    Timer1.Interval = aika1 'ohjelman etenemisviive
    Timer2.Interval = aika2 'Robodrillin vastausviive
    Timer3.Interval = aika3 'Pursotuspään vastausviive
    Timer2.Enabled = True
```

```
End Sub
```

```
Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.
Tick
```

```
    Dim Rivi, merkki, jono As String
    Dim Ppaikka As Integer
    ' MsgBox("Aloitetaan")
    pitaapysayttaa = 0
    Timer1.Enabled = False
    ' Timer1.Stop()

    If ListBox2.Items.Count > ListBox2.SelectedIndex + 1 Then

        ListBox2.SelectedIndex = ListBox2.SelectedIndex + 1
        ListBox3.SelectedIndex = ListBox2.SelectedIndex
        Rivi = ListBox2.Text
        TextBox2.Text = Rivi

        If InStr(Rivi, "(") > 1 Then
            Rivi = Mid(Rivi, 1, InStr(Rivi, "(") - 1)
        End If

        TextBox2.Text = Rivi
        TextBox2.Update()
```


D:\NCrapid\NCrapid\Form1.vb

6

```

    TextBox3.Text = ListBox3.Text
    TextBox3.Update()

    If (InStr(Rivi, "M83") > 0 Or InStr(Rivi, "M81") > 0) Then
        pitaapysayttaa = 1
        While SerialPort2.BytesToRead > 0

            Mvastaus = SerialPort2.ReadByte

        End While
        Timer2.Enabled = True
    End If

    If InStr(Rivi, "M6") > 0 Then
        pitaapysayttaa = 1
        lammitysmenossa = 1
    End If

    If InStr(Rivi, "G04") > 0 Then 'viive
        pitaapysayttaa = 1
        ' Rivi = TextBox2.Text
        Ppaikka = InStr(Rivi, "P")
        jono = ""
        For j = Ppaikka + 1 To Len(Rivi)
            merkki = Mid(Rivi, j, 1)
            If Asc(merkki) = 32 Or Asc(merkki) = 13 Or Asc(merkki) = 40 Then Exit For
            'tuli välilyönti tai rivinloppu tai sulku
            If Asc(merkki) > 42 And Asc(merkki) < 58 Then jono = jono + merkki
        Next j
        Timer4.Interval = Val(jono)
        Timer4.Enabled = True

    End If

    If InStr(Rivi, "(") = 0 Then 'ei ole kommenttirivi
        ' Rivi = TextBox2.Text
        If InStr(Rivi, "M81") > 0 Then Mkysymys = "M81"
        If InStr(Rivi, "M83") > 0 Then Mkysymys = "M83"
        For i = 1 To Len(Rivi)
            merkki = Mid(Rivi, i, 1)
            SerialPort1.Write(merkki)
        Next
        SerialPort1.WriteLine("")
    End If

    If TextBox3.Text <> "" Then SerialPort3.Write(TextBox3.Text)
    If TextBox3.Text <> "" Then SerialPort3.WriteLine(TextBox3.Text)
    ' MsgBox(Str(pitaapysayttaa))
    If pitaapysayttaa = 0 Then
        Timer1.Enabled = True
        ' Timer1.Start()

    End If
End If
End If 'oliko listassa rivejä
End Sub

Private Sub Timer2_Tick(sender As System.Object, e As System.EventArgs) Handles Timer2.
Tick
    'Lukee Arduinojen tilaa
    Timer2.Enabled = False
    Timer1.Enabled = False
    Mvastaus = SerialPort2.ReadExisting
    TextBox5.Text = Mvastaus
    If (InStr(Mvastaus, "Q") > 0) And Mkysymys = "M81" Then pitaapysayttaa = 0

```

D:\NCrapid\NCrapid\Form1.vb

7

```
    If (InStr(Mvastaus, "S") > 0) And Mkysymys = "M83" Then pitaapysayttaa = 0

    Pvastaus = SerialPort3.ReadExisting
    TextBox6.Text = Pvastaus
    If (InStr(Pvastaus, "M6ok") > 0) Then
        lammitysmenossa = 0
    End If

    'If pitaapysayttaa = 1 Then Timer2.Enabled = True
    Timer2.Enabled = True
    If lammitysmenossa = 0 And pitaapysayttaa = 0 And saaAjaa = 1 Then Timer1.Enabled = True
End Sub

Private Sub Timer4_Tick(sender As System.Object, e As System.EventArgs) Handles Timer4.Tick
    Timer4.Enabled = False
    Timer1.Enabled = True

End Sub

End Class
```



Electronic Parts and Components

DATA SHEET NTC

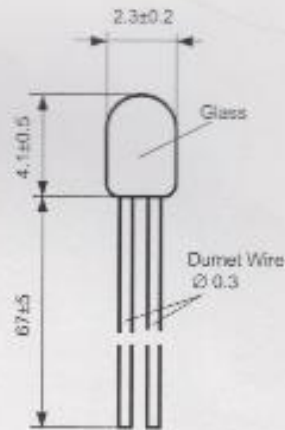
G560/100k/+

Application

Temperature measurement for high temperature and short response times.

Version

NTC-thermistor sealed in glass body.
 Dumet wires: \varnothing 0.3 mm
 lead length: 67 ± 5 mm
 headdiameter: 2.3 ± 0.2 mm
 headlength: 4.1 ± 0.5 mm



Dimensions in [mm]

Data

Climatic Category (IEC 60068-1)
 Lower category temperature
 Upper category temperature

: 55/300/56
 [°C]: -55
 [°C]: 300

Rated resistance R_N // Tolerance $\Delta R/R$
 Rated temperature

R_N [Ω // %]: 100 000 // $\pm 1^1$
 T_N [°C]: 25

B-value : $B_{(0/100)}$ // Tolerance $\Delta B/B$
 R_{25}

B_N [K//%]: 4036 // ± 1
 [Ω]: 100 000

Dissipation factor (air)
 Thermal time constant (air)
 Heat capacity

δ_{th} [mW/K]: approx. 1.3
 τ_C [s]: approx. 15
 C_{th} [mJ/K]: approx. 20

Ordering code

: B57560G104+

Remarks

¹⁾ +: F for $\Delta R/R_N = \pm 1\%$; standard dumet wires
 H for $\Delta R/R_N = \pm 3\%$; standard dumet wires
 F2 for $\Delta R/R_N = \pm 1\%$; dumet wires, Ni-plated
 H2 for $\Delta R/R_N = \pm 3\%$; dumet wires, Ni-plated

				DATENBLATT / DATA SHEET	
				HEISSLEITER / NTC	
Stück	Mitteilung	Datum	Name		B57560G104+
bezeichnet	Info No.	Date	Sign.		
		25.09.01			2

Resistance Temperature Characteristics

R at 25°C 100000 [Ω]
 B(0/100) 4036 [K] ± 1 [%]
 RN at 25 °C 100000 [Ω] ± 1 [%]

T [°C]	R _{nom} [Ω]	R _{min} [Ω]	R _{max} [Ω]	ΔR/R _N [±%]	ΔT [±°C]	α [%/K]
-55	10723236	10101868	11381685	6,1	0,8	7,7
-50	7366689	6968648	7786687	5,7	0,8	7,4
-45	5132757	4874705	5403929	5,3	0,7	7,1
-40	3624165	3455040	3801187	4,9	0,7	6,9
-35	2591304	2479373	2708016	4,5	0,7	6,6
-30	1874913	1800190	1952543	4,1	0,6	6,4
-25	1371886	1321623	1423918	3,8	0,6	6,2
-20	1014545	980519	1049648	3,5	0,6	5,9
-15	757881	734728	781685	3,1	0,5	5,8
-10	571590	555777	587794	2,8	0,5	5,6
-5	435026	424202	446081	2,5	0,5	5,4
0	333964	326553	341508	2,3	0,4	5,2
5	258497	253434	263636	2,0	0,4	5,1
10	201659	198216	205142	1,7	0,4	4,9
15	158499	156178	160838	1,5	0,3	4,8
20	125468	123927	127016	1,2	0,3	4,6
25	100000	99000	101000	1,0	0,2	4,5
30	80223	79243	81206	1,2	0,3	4,4
35	64759	63830	65695	1,4	0,3	4,2
40	52589	51727	53460	1,7	0,4	4,1
45	42951	42161	43751	1,9	0,5	4,0
50	35272	34556	36000	2,1	0,5	3,9
55	29119	28473	29776	2,3	0,6	3,8
60	24161	23581	24752	2,4	0,7	3,7
65	20144	19626	20674	2,6	0,7	3,6
70	16874	16411	17348	2,8	0,8	3,5
75	14198	13786	14622	3,0	0,9	3,4
80	11998	11631	12376	3,2	0,9	3,3
85	10181	9854	10519	3,3	1,0	3,2
90	8674	8382	8976	3,5	1,1	3,2
95	7419	7158	7688	3,6	1,2	3,1
100	6369	6136	6610	3,8	1,3	3,0
105	5487	5279	5703	3,9	1,3	2,9
110	4744	4557	4937	4,1	1,4	2,9
115	4115	3948	4288	4,2	1,5	2,8
120	3581	3431	3737	4,4	1,6	2,8
125	3126	2991	3268	4,5	1,7	2,7

DATENBLATT / DATA SHEET

HEISSLEITER / NTC

a		25.09.01	
Ausg. Jahr	Abteilung Info No.	Datum Date	Name Sign.



B57560G104+

D. Page 4

Resistance Temperature Characteristics

R at 25°C 100000 [Ω]
 B(0/100) 4036 [K] \pm 1 [%]
 RN at 25 °C 100000 [Ω] \pm 1 [%]

T [°C]	R_nom [Ω]	R_min [Ω]	R_max [Ω]	$\Delta R/R_N$ [\pm %]	ΔT [\pm °C]	α [%/K]
130	2737	2616	2864	4,6	1,8	2,6
135	2404	2294	2518	4,8	1,8	2,6
140	2117	2018	2220	4,9	1,9	2,5
145	1869	1780	1963	5,0	2,0	2,5
150	1655	1574	1740	5,1	2,1	2,4
155	1469	1396	1546	5,2	2,2	2,4
160	1307	1241	1377	5,4	2,3	2,3
165	1166	1106	1230	5,5	2,4	2,3
170	1043	987,6	1101	5,6	2,5	2,2
175	934,5	884,1	987,5	5,7	2,6	2,2
180	839,3	793,3	887,8	5,8	2,7	2,1
185	755,4	713,3	799,9	5,9	2,8	2,1
190	681,3	642,8	722,2	6,0	2,9	2,0
195	615,8	580,4	653,3	6,1	3,0	2,0
200	557,6	525,1	592,1	6,2	3,1	2,0
205	505,9	476,0	537,7	6,3	3,3	1,9
210	459,9	432,3	489,2	6,4	3,4	1,9
215	418,8	393,3	445,8	6,5	3,5	1,9
220	382,0	358,5	407,0	6,6	3,6	1,8
225	349,1	327,3	372,2	6,6	3,7	1,8
230	319,5	299,3	341,0	6,7	3,8	1,8
235	292,9	274,2	312,9	6,8	4,0	1,7
240	269,0	251,6	287,5	6,9	4,1	1,7
245	247,3	231,2	264,6	7,0	4,2	1,7
250	227,8	212,8	243,9	7,1	4,3	1,6
255	210,1	196,1	225,1	7,1	4,5	1,6
260	194,1	181,0	208,1	7,2	4,6	1,6
265	179,5	167,3	192,6	7,3	4,7	1,5
270	166,3	154,9	178,5	7,4	4,8	1,5
275	154,2	143,5	165,7	7,4	5,0	1,5
280	143,2	133,2	154,0	7,5	5,1	1,5
285	133,2	123,8	143,2	7,6	5,2	1,4
290	124,0	115,2	133,4	7,6	5,4	1,4
295	115,5	107,3	124,5	7,7	5,5	1,4
300	107,8	100,0	116,2	7,8	5,7	1,4

DATENBLATT / DATA SHEET

HEISSLEITER / NTC



Resistance Temperature Characteristics

R at 25°C 100000 [Ω]
 B(0/100) 4036 [K] ± 1 [%]
 RN at 25 °C 100000 [Ω] ± 3 [%]

T [°C]	R_nom [Ω]	R_min [Ω]	R_max [Ω]	ΔR/R _N [±%]	ΔT [±°C]	α [%/K]
-55	10723236	9897790	11607065	8,2	1,1	7,7
-50	7366689	6827867	7940878	7,8	1,1	7,4
-45	5132757	4776226	5510938	7,4	1,0	7,1
-40	3624165	3385242	3876459	7,0	1,0	6,9
-35	2591304	2429285	2761640	6,6	1,0	6,6
-30	1874913	1763822	1991207	6,2	1,0	6,4
-25	1371886	1294924	1452115	5,8	0,9	6,2
-20	1014545	960710	1070431	5,5	0,9	5,9
-15	757881	719885	797164	5,2	0,9	5,8
-10	571590	544549	599434	4,9	0,9	5,6
-5	435026	415633	454914	4,6	0,8	5,4
0	333964	319956	348271	4,3	0,8	5,2
5	258497	248314	268857	4,0	0,8	5,1
10	201659	194212	209204	3,7	0,8	4,9
15	158499	153023	164023	3,5	0,7	4,8
20	125468	121423	129531	3,2	0,7	4,6
25	100000	97000	103000	3,0	0,7	4,5
30	80223	77642	82814	3,2	0,7	4,4
35	64759	62541	66995	3,5	0,8	4,2
40	52589	50682	54519	3,7	0,9	4,1
45	42951	41310	44618	3,9	1,0	4,0
50	35272	33858	36713	4,1	1,0	3,9
55	29119	27898	30366	4,3	1,1	3,8
60	24161	23105	25242	4,5	1,2	3,7
65	20144	19230	21084	4,7	1,3	3,6
70	16874	16080	17692	4,8	1,4	3,5
75	14198	13507	14911	5,0	1,5	3,4
80	11998	11396	12622	5,2	1,6	3,3
85	10181	9655	10727	5,4	1,7	3,2
90	8674	8213	9154	5,5	1,7	3,2
95	7419	7013	7841	5,7	1,8	3,1
100	6369	6012	6741	5,8	1,9	3,0
105	5487	5172	5816	6,0	2,0	2,9
110	4744	4465	5035	6,1	2,1	2,9
115	4115	3868	4373	6,3	2,2	2,8
120	3581	3362	3811	6,4	2,3	2,8
125	3126	2931	3331	6,6	2,4	2,7

DATENBLATT / DATA SHEET

HEISSLEITER / NTC

8	25.09.01		
Ausg. Nr.	Mitteilung Nr.	Datum	Name
Issue	Info No.	Date	Sgn.




B57560G104+

6/6

Resistance Temperature Characteristics

R at 25°C 100000 [Ω]
 B(0/100) 4036 [K] ± 1 [%]
 RN at 25 °C 100000 [Ω] ± 3 [%]

T [°C]	R_nom [Ω]	R_min [Ω]	R_max [Ω]	ΔR/R _N [±%]	ΔT [±°C]	α [%/K]
130	2737	2563	2920	6,7	2,5	2,6
135	2404	2248	2568	6,8	2,7	2,6
140	2117	1977	2264	7,0	2,8	2,5
145	1869	1744	2001	7,1	2,9	2,5
150	1655	1542	1774	7,2	3,0	2,4
155	1469	1368	1577	7,3	3,1	2,4
160	1307	1216	1405	7,4	3,2	2,3
165	1166	1083	1254	7,6	3,3	2,3
170	1043	967,6	1123	7,7	3,5	2,2
175	934,5	866,3	1007	7,8	3,6	2,2
180	839,3	777,3	905,4	7,9	3,7	2,1
185	755,4	698,9	815,7	8,0	3,8	2,1
190	681,3	629,8	736,5	8,1	4,0	2,0
195	615,8	568,7	666,2	8,2	4,1	2,0
200	557,6	514,5	603,8	8,3	4,2	2,0
205	505,9	466,4	548,3	8,4	4,3	1,9
210	459,9	423,5	498,9	8,5	4,5	1,9
215	418,8	385,3	454,7	8,6	4,6	1,9
220	382,0	351,2	415,1	8,7	4,8	1,8
225	349,1	320,7	379,6	8,8	4,9	1,8
230	319,5	293,3	347,7	8,8	5,0	1,8
235	292,9	268,7	319,0	8,9	5,2	1,7
240	269,0	246,5	293,2	9,0	5,3	1,7
245	247,3	226,5	269,8	9,1	5,5	1,7
250	227,8	208,5	248,7	9,2	5,6	1,6
255	210,1	192,1	229,6	9,3	5,8	1,6
260	194,1	177,4	212,2	9,3	5,9	1,6
265	179,5	163,9	196,4	9,4	6,1	1,5
270	166,3	151,7	182,0	9,5	6,2	1,5
275	154,2	140,6	169,0	9,6	6,4	1,5
280	143,2	130,5	157,0	9,6	6,6	1,5
285	133,2	121,3	146,1	9,7	6,7	1,4
290	124,0	112,8	136,1	9,8	6,9	1,4
295	115,5	105,1	126,9	9,8	7,1	1,4
300	107,8	98,01	118,5	9,9	7,2	1,4

				DATENBLATT / DATA SHEET	
				HEISSLEITER / NTC	
..B		25.09.01			
Augsp.	Mitteilung	Datum	Name		
Issue	Info No.	Date	Sign.	B57560G104+	
				B. Page 7	