



Jani Österberg

**MATKAPUHELIMEN NÄYTÖN ISKUNKESTÄVYYSTESTIN
SUUNNITTELU JA TOTEUTUS TEOLLISUUSROBOTIN AVULLA**

**MATKAPUHELIMEN NÄYTÖN ISKUNKESTÄVYYSTESTIN
SUUNNITTELU JA TOTEUTUS TEOLLISUUSROBOTIN AVULLA**

Jani Österberg
Opinnäytetyö
Kevät 2012
Automaatiotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma, projektoinnin suuntautumisvaihtoehto

Tekijä(t): Jani Österberg
Opinnäytetyön nimi: Matkapuhelimen näytön iskunkestävyydestin suunnittelu ja toteutus teollisuusrobotin avulla
Työn ohjaaja(t): Heikki Kurki
Työn valmistumislukukausi ja -vuosi: Kevät 2012
Sivumäärä: 50 + 5 liitettä

Tämä opinnäytetyö on Nokia Oyj:n toimeksiannosta toteutettu matkapuhelimien tuotekehityksen testauksen menetelmäkehitystyö. Opinnäytetyön aiheena on matkapuhelimille tehtävän ball drop -testin suunnittelu ja toteutus teollisuusrobotin avulla. Ball drop -testissä teräskuula pudotetaan matkapuhelimen näytölle eri kohtiin eri korkeuksilta puhelimen näytön iskunkestävyyden selvittämiseksi. Vanhassa ball drop -testauslaitteistossa suurimmat ongelmat liittyivät testausjärjestelyn tarkkuuteen ja toistettavuuteen. Uusi testausjärjestely päätettiin toteuttaa teollisuusrobotin avulla sen erinomaisen paikoitustarkkuuden vuoksi. Päättävänä tässä menetelmäkehitystyössä on parantaa testissä käytettävän kuulun osumapisteiden tarkkuutta ja toistettavuutta. Lisäksi työssä pyritään nostamaan kyseisen testin automaatioastetta.

Opinnäytetyössä suunniteltiin ja piirrettiin kuvat testauslaitteistoon tarvittavista osista. Laitteiston osat valmistettiin pikamallitekniikalla. Työssä valittiin testausjärjestelyyn tarvittavat toimilaitteet sekä esitettiin niiden kytkennät ja testauslaitteiston lopullinen kokoonpano. Testauslaitteiston suunnittelun ja toteutuksen jälkeen työssä luotiin testauksessa käytettävälle robotille testausjärjestelyn toimintasekvenssin mukainen sovellus. Sovelluksen ohjelmointi tehtiin RAPID-kielellä.

Opinnäytetyön tuloksena saatiin luotua kokonaan uusittu testauskokonaisuus, joka koostui uudesta testauslaitteistosta sekä käytettävästä teollisuusrobotista sovelluksineen. Robotilla toteutettu testausjärjestely paransi teräskuulan osumapisteiden tarkkuutta ja toistettavuutta huomattavasti. Lisäksi testauksen automatisoinnin kehittäminen helpotti operaattorin työtä. Laitteiston osien ja sovelluksen suunnittelussa huomioitiin uuden testauslaitteiston käyttö useamman eri puhelinmallin testauksessa tulevaisuudessa. Opinnäytetyön loppupuolella pohdittiin testausjärjestelyn turvallisuutta sekä mahdollisuuksia testauslaitteiston jatkokehittämiseen kuulun osumapisteiden tarkkuuden parantamiseksi entisestään.

Asiasanat: automaatio, matkapuhelimet, ohjelmointi, robotit, testaus.

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Automation Engineering, Option of Project Engineering

Author(s): Jani Österberg

Title of thesis: Design and Implementation of Ball Drop Test with Industrial Robot

Supervisor(s): Heikki Kurki

Term and year when the thesis was submitted: Spring 2012

Pages: 50 + 5 appendices

This Bachelor's thesis is a mobile phone test method development work to Nokia Corporation's R&D department. Topic of the thesis is design and implementation of the ball drop test with industrial robot. In the ball drop test a steel ball is dropped to different locations on the display of the mobile phone. The ball is dropped from different heights to determine impact resistance of the phone. Accuracy and reproducibility of the test were the main problems in the previous ball drop test setup. The new test setup was done with industrial robot for its excellent positioning accuracy. The main objectives of this thesis is to improve accuracy and reproducibility of the steel ball impact location points. Another objective of the thesis is to increase the degree of automation of the test.

The layouts of the components of ball drop test assembly were designed in this thesis work. The hardware components was made by rapid prototyping technology. Pneumatic and electrical components were chosen and connected. After design and implementation of the hardware components, the control application of the industrial robot was designed. The application was made with RAPID programming language.

The result of the thesis was a completely improved ball drop test equipment, which consisted of the new hardware components and the industrial robot with designed application. The ball drop test with industrial robot improved the accuracy and reproducibility of the steel ball impact points. In addition, the new automated test sequence made testing easier to operators. In the future, it is possible to test several different phone models with the new ball drop test equipment. The safety of industrial robot in the ball drop test was reflected in the end of this Bachelor's thesis. In addition to this, the opportunities to improve accuracy of impact points more in future were assessed.

Keywords: automation, mobile phones, programming, robotics, testing

ALKULAUSE

Tämä opinnäytetyö on testauslaitteiston menetelmäkehitystyö, joka on tehty Nokia Oyj:lle Oulun Peltolan yksikköön. Haluan kiittää Nokia Oyj:tä mahdollisuudesta toteuttaa omasta ideasta syntynyt menetelmäkehitysprojekti.

Haluan kiittää EXT-Test Engineer Heikki Ahvensalmea avusta testauslaitteiston mekaanisten osien piirtämisessä. Lisäksi esitän kiitokseni opinnäytetyön valvojalle HW Testing Specialist Sami Järvelälle, opinnäytetyön ohjaajalle yliopettaja Heikki Kurjelle sekä kaikille niille, jotka ovat auttaneet minua tämän opinnäytetyön valmistumisessa.

Oulussa 25.5.2012

Jani Österberg

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
MERKKIEN SELITYKSET JA SANASTO	8
1 JOHDANTO	9
2 BALL DROP -TESTI	10
3 ABB IRB 1400 -TEOLLISUUSROBOTTI	12
3.1 Koordinaattijärjestelmät	14
3.2 Ohjelmointi	15
3.3 I/O-järjestelmä	16
4 TESTAUSLAITTEISTON SUUNNITTELU JA TOTEUTUS	19
4.1 Lähtötilanne	19
4.2 Vaatimukset testauslaitteistolle	20
4.3 Testauslaitteiston osien suunnittelu	23
4.3.1 Jigi	23
4.3.2 Jalusta	24
4.3.3 Luiska	25
4.3.4 Anturipidike	27
4.3.5 Osien yhdistäminen ja valmistus	27
4.4 Laitevalinnat	28
4.4.1 Teräskuulan tunnistaminen	28
4.4.2 Pneumaattiset laitteet	29
4.5 Kytkennät	31
4.5.1 Sähköiset kytkennät	31
4.5.2 Pneumaattiset kytkennät	32
4.6 Valmis testauslaitteisto	33
5 ROBOTIN OHJELMOINTI	37
5.1 Osumapisteiden määrittely	37
5.2 Robotin toimintasekvenssi	39
5.3 Ohjaava sovellus	40

6 TYÖN ARVIOINTI	42
6.1 Tarkkuus ja toistettavuus	42
6.2 Turvallisuus	44
6.3 Muutokset otettaessa uusi puhelinmalli testaukseen	45
7 POHDINTA	46
LÄHTEET	48
LIITTEET	50

MERKKIEN SELITYKSET JA SANASTO

AI	Analoginen tulo
AO	Analoginen lähtö
DI	Digitaalinen tulo
DO	Digitaalinen lähtö
I/O	Järjestelmän tulot/lähdöt
Jigi	Puhelimelle suunniteltu alusta
KK	Kytkentäkotelo, kenttäkotelo
Nurkkapiste	Robotin xyz-koordinaatiston piste, jossa robottiin kiinnitetty imukuppi on puhelimen näytön nurkkapisteen yläpuolella
Osumapiste	Piste puhelimen näytöllä, mihin teräskuula putoaa
Paikoituspiste	Robotin sovellukseen tallennettu piste, josta selviää robotin paikoitus ja asento
Poimintapiste	Robotin xyz-koordinaatiston piste, josta robotti poimii teräskuulan
Pudotuspiste	Robotin xyz-koordinaatiston piste, josta teräskuula putoaa puhelimen näytölle
Rutiini	Robotille luotu ohjelma
TKP, (TCP)	Työkalukeskipiste, jonka paikoituksen perusteella robotin aseman koordinaatit näytetään käyttäjälle

1 JOHDANTO

Tämän opinnäytetyön aiheena on matkapuhelimille tehtävän ball drop -testin kehittäminen. Tehtävänä työnä on suunnitella ja toteuttaa teollisuusrobotin avulla testausjärjestely kyseistä testiä varten. Työ tehdään Nokia Oyj:n Oulun yksikön toimeksiannosta.

Ball drop -testiä käytetään matkapuhelimien näyttöjen iskunkestävyyden määrittämiseen puhelimen tuotekehityksen yhteydessä suoritettavassa testauksessa. Ball drop -testissä teräskuula pudotetaan puhelimen näytölle ennalta määrättyihin kohtiin eri korkeuksilta ja havaitut viat kirjataan ylös. Suurimpina ongelmina vanhassa ball drop -testauslaitteistossa oli teräskuulan osumapisteiden tarkkuus ja pudotuskohtien toistettavuus. Myös kuulan aiheuttamien iskujen voimakkuuksien vaihtelu aiheutti ongelmia, koska puhelin pääsi liikkumaan iskujen voimasta.

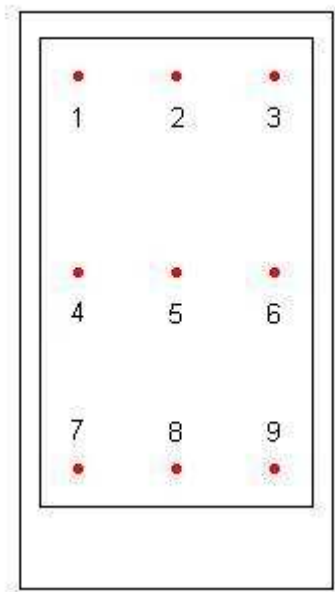
Ball drop -testi suunniteltiin toteutettavaksi Nokia Oyj:n testauslaboratoriossa olevan teollisuusrobotin avulla, jotta pudotusten tarkkuus ja toistettavuus paranevat. Lisäksi testausjärjestelyn automatisointi helpottaa operaattorin työtä.

Tässä opinnäytetyössä suunnitellaan ja valmistetaan uuden testauslaitteiston osat, tehdään tarvittavat laitevalinnat ja kytkennät sekä kootaan uusi testauslaitteisto valmiiksi testauskokonaisuudeksi. Tämän lisäksi olennaisena osana opinnäytetyötä on ball drop -testissä käytettävälle robotille suunniteltava sovellus. Sovellus suunnitellaan ball drop -testin suorittamiseksi mahdollisimman automatisoidusti. Robottisovelluksen ohjelmointi tapahtuu RAPID-kielellä.

2 BALL DROP -TESTI

Ball drop -testin tarkoituksena on määrittää matkapuhelimen näytön iskunkestävyyttä. Testissä iskun aiheuttaa halkaisijaltaan 20 mm:n ja massaltaan 32,5 g:n painoinen teräskuula, joka päästetään vapaasti putoamaan matkapuhelimen näytölle. Kuula pudotetaan yhdeksään osumapisteeseen neljältä eri korkeudelta. Yhteensä testissä siis suoritetaan 36 pudotusta. Pudotukset aloitetaan aina alimmasta pudotuskorkeudesta.

Teräskuulan pudotuspisteistä kahdeksan on määritelty 5 mm:n päähän matkapuhelimen näytön reunasta ja yksi keskelle näyttöä. Kuvassa 1 on esitettynä kosketusnäytöllisen matkapuhelimen pudotuspisteet ball drop -testissä. Testissä on tavoitteena päästä pudotuspisteen määrittämisessä tarkkuuteen ± 1 mm.



KUVA 1. Teräskuulan osumapisteet kosketusnäytöllisen matkapuhelimen näytölle

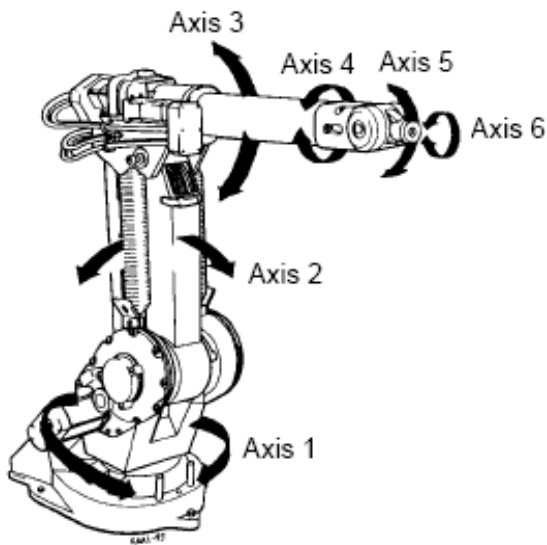
Testissä alimmaksi pudotuskorkeudeksi on määritelty 47 cm. Seuraava korkeus on 63 cm. Kolmas pudotuskorkeus on 78 cm ja viimeinen 94 cm. Jokaiselta pudotuskorkeudelta pudotetaan kuula kaikkiin yhdeksään pisteeseen.

Yleensä ball drop -testissä näyttövikojen tarkastus tehdään jokaisen pudotuksen jälkeen. Toisinaan testi kuitenkin voidaan tehdä siten, että viat tarkastetaan ainoastaan jokaisen pudotuskorkeuden jälkeen eli yhdeksän pudotuksen välein. Tällöin puhelimen testaus nopeutuu huomattavasti.

Testissä tulisi käyttää jonkinlaista jigiä, joka pitää puhelimen testin aikana mahdollisimman hyvin paikallaan. Jigin käyttö parantaa näytön eri pisteiden iskunkestävyyden vertailtavuutta, koska iskun osuessa näyttöön ei puhelin pääse liikkumaan iskun voimasta.

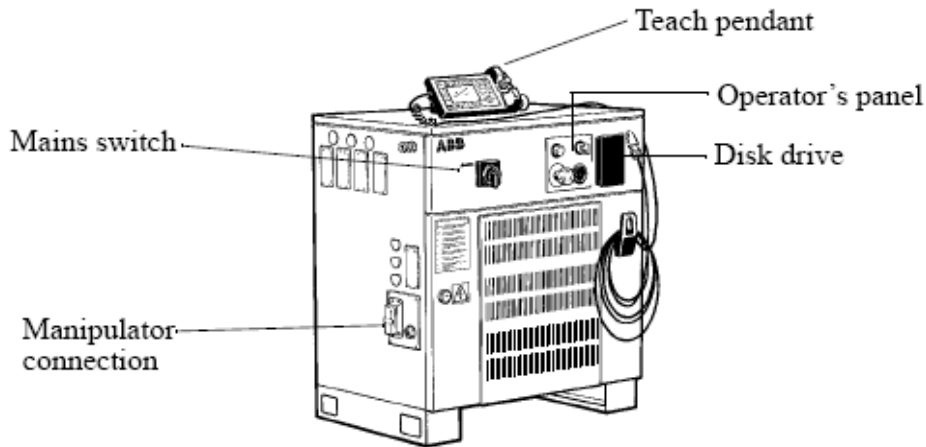
3 ABB IRB 1400 -TEOLLISUUSROBOTTI

ABB:n valmistama teollisuusrobotti IRB 1400 koostuu kahdesta pääosasta, manipulaattorista ja ohjainkaapista. Manipulaattori on robotin liikkuva osa, jolla käyttäjän haluamat toiminnot suoritetaan. Manipulaattorin päähän voidaan asentaa erilaisia työkaluja ja tarttujia. IRB 1400 -robotin manipulaattori on kuusiakselinen ja jokaista akselia voidaan liikuttaa kahteen suuntaan. (Kuva 2.) (5, luku 4, s. 7–8.)



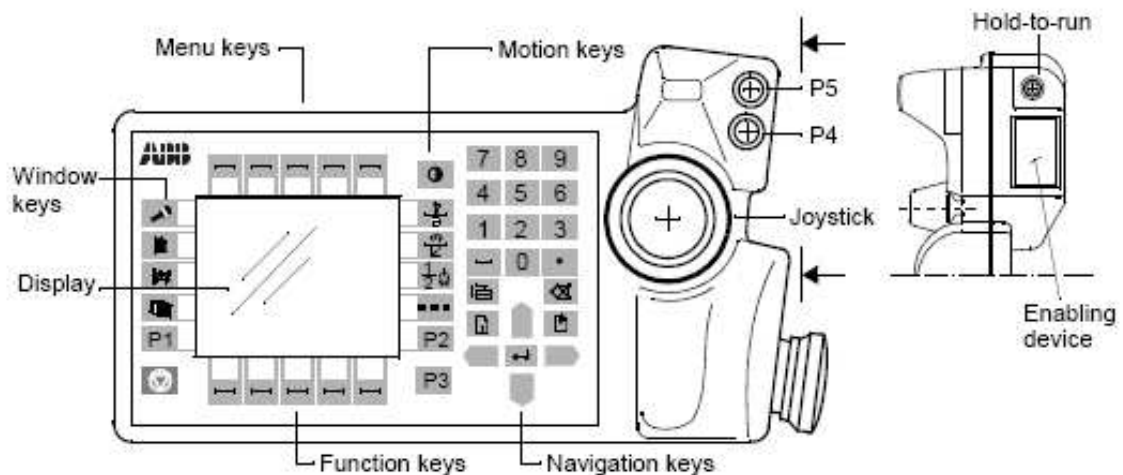
KUVA 2. IRB 1400 -robotin manipulaattori, akselit ja niiden liikesuunnat (2, s. 5)

IRB 1400 -robotin toinen pääosa eli ohjainkaappi sisältää robotin ohjaukseen tarvittavat hallintalaitteet ja komponentit. Hallintalaitteina ohjauskaapissa toimivat ohjelmointiyksikkö, käyttöpaneeli ja pääkytkin. Lisäksi ohjainkaapissa on levyasema sovellusten tallentamista ja siirtämistä varten. Ohjainkaapin sisälle on sijoitettu mm. muuntaja, taajuusmuuttaja, laitteiston tehonsyöttöyksikkö, robotin päätietokone ja I/O-tietokone sekä signaalien välittämiseen tarvittavat I/O-kortit. (Kuva 3.)(6, luku 1, s. 2; 2, s. 168–169.)



KUVA 3. IRB 1400 -robotin ohjainkaappi (2, s. 167)

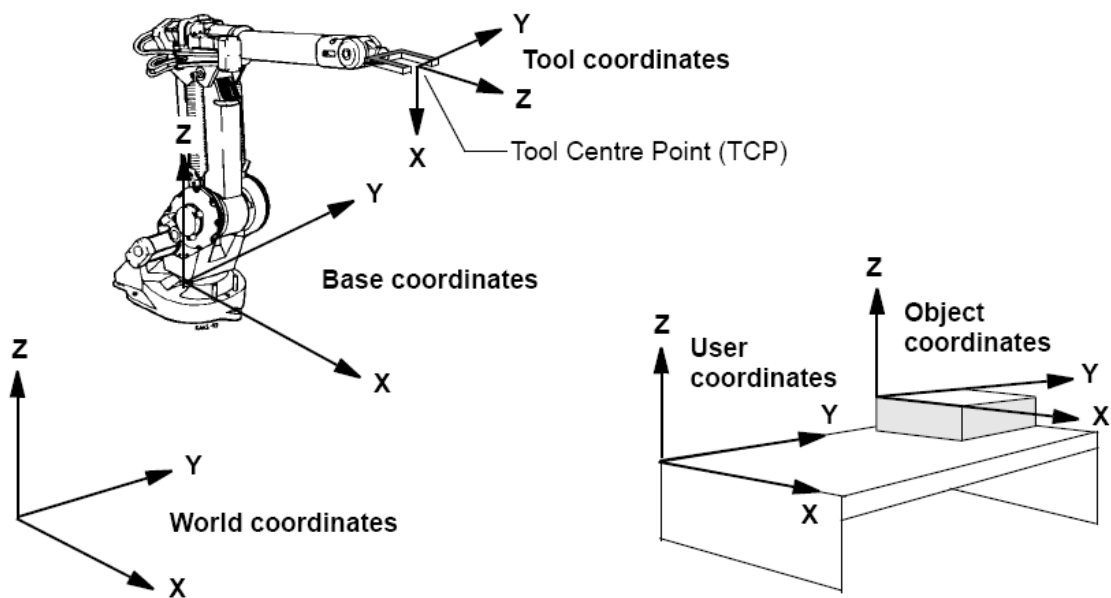
Tärkein robotin hallintalaitteista on ohjelmointiyksikkö, joka toimii pääasiallisena rajapintana käyttäjän ja robotin välillä. Robotin ohjelmointi, testaus ja käyttö suoritetaan ohjelmointiyksikön avulla. Ohjelmointiyksikön ohjaussauvalla voidaan ohjata robottia manuaalillassa sallintalaitteen ollessa painettuna. Manuaaliltilan lisäksi robotissa on valittavissa automaattitila. Ohjelmointiyksikössä on lisäksi ikkunanäppäimet, valikkonäppäimet, liikkeen valintanäppäimet, ohjausnäppäimet, numeeriset näppäimet, käyttäjän määrittelemät näppäimet P1–P5 sekä funktionäppäimet, joille voidaan ohjelmoida robotin odottamia käskyjä käyttäjältä. (Kuva 4.)(5, luku 4, s. 10–11.)



KUVA 4. IRB 1400 -robotin ohjelmointiyksikkö (2, s. 33)

3.1 Koordinaattijärjestelmät

IRB 1400 -robotissa kaikki robotin paikoitukset määritellään xyz-koordinaatistojen avulla suhteessa käytössä olevaan koordinaattijärjestelmään. Robottia siirretään koordinaattijärjestelmissä työkalukeskipisteen (TKP) perusteella, joka on oletusarvoisesti robotin manipulaattorin päässä olevan laipan keskellä. Käytössä oleva koordinaattijärjestelmä voi olla suhteessa muihin koordinaattijärjestelmiin ketjutettuna. Näitä koordinaattijärjestelmiä on robotin käyttöoppaassa esitelty kaksitoista. Osa näistä on käyttäjän määriteltävissä ja osa on sisällytetty robotin konfiguraatioon. Kuvassa 5 on esitettyä robotin tärkeimpiä koordinaatiojärjestelmiä. (5, luku 10, s. 3; 6, luku 3, s. 4.)



KUVA 5. IRB 1400 -robotin koordinaatiojärjestelmiä (2, s. 25)

Kaikki robotille ohjelmoidut paikoitukset ovat aina suhteutettuna kehysketjun kautta maailmakoordinaatistoon, joka on määritelty robotin konfiguroinnin yhteydessä. Tavallisesti maailmakoordinaatisto konfiguroidaan robotin asennuksen yhteydessä yhdensuuntaiseksi peruskoordinaatiston kanssa. Jos käytetään useampaa robottia esimerkiksi tuotantolinjalla, robottien maailmakoordinaatistot konfiguroidaan samansuuntaisiksi riippumatta robotin asennosta. Robotin peruskoordinaatisto taas on määritelty robotin manipulaattorin mukaan. X-akselin on määrätty osoittavan suoraan eteenpäin robotin manipulaattorista katsoen ja z-akselin suoraan ylöspäin. (Kuva 5.) (5, luku 10, s. 3; 7, s. 121–122.)

Robotissa on mahdollista käyttää myös esimerkiksi työkalukoordinaatistoa, käyttäjäkoordinaatistoa tai työkohdekoordinaatistoa. Käyttäjäkoordinaatistoksi voidaan määritellä esimerkiksi pöytä tai linjasto, jossa työkohde sijaitsee. Käyttäjäkoordinaatisto määritellään suhteessa maailmakoordinaatistoon. Työkohteelle itselleen voidaan määritellä työkohdekoordinaatisto, joka on suhteessa käyttäjäkoordinaatistoon. Työkalukoordinaatisto taas määritellään suhteessa robotin manipulaattorin ranteen koordinaatistoon. (Kuva 5.) (5, luku 10, s. 3.)

Robottia on mahdollista liikuttaa myös akseli kerrallaan. Akseli kerrallaan tapahtuvat robotin liikkeet määritellään kulma-asteina robotin kalibrointiasentoon nähden. Robotin kalibrointiasennossa kaikki akselit on ajettu manuaaliajolla arvoon 0,0 astetta. (6, luku 3, s. 5.)

3.2 Ohjelmointi

ABB:n IRB 1400 -robotin ohjelmointi tehdään tavallisesti käyttäen ohjelmointiyksikköä. Luotava sovellus voi sisältää pääohjelman ja aliohjelmia. ABB:n ympäristössä käytetään näistä ohjelmista nimitystä rutiinit. Rutiinin alussa sovellus käsittelee rutiinin määrittelyt, jotka sisältävät rutiinin nimen, rutiinityypin, tarkoituksen, funktion datatyyppin ja parametrin. Lisäksi rutiini sisältää sovelluksessa tarvittavan datan, käskyt ja virheenkäsittelyn. (5, luku 8, s. 9; 7, s. 83.)

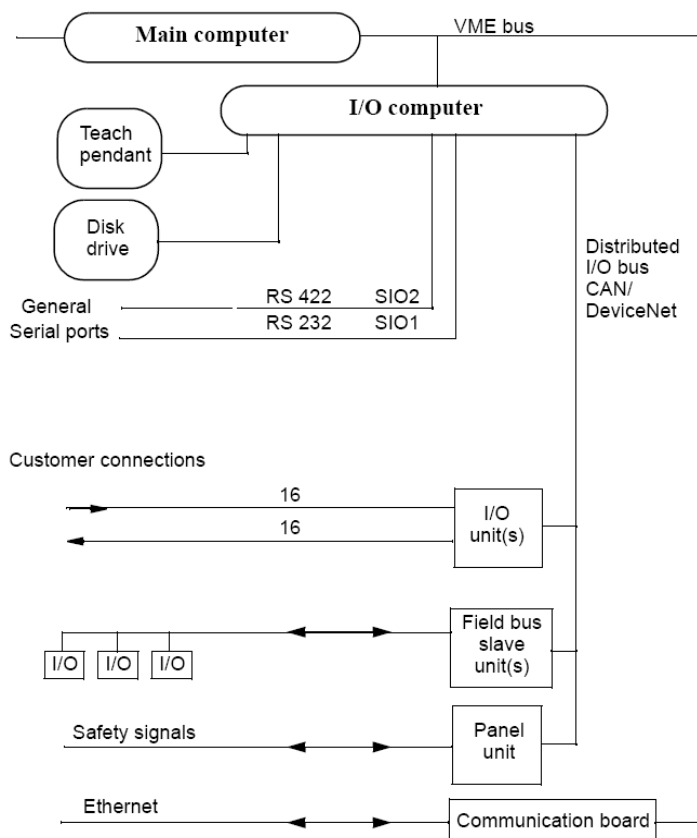
ABB:n robottien sovellusten luomisessa käytetään RAPID-ohjelmointikieltä. RAPID-kielellä tapahtuva ohjelmointi on käskylistamuotoista ohjelmointia ja sovelluksen suoritus etenee käsky kerrallaan. Robotille annetuilla käskyillä hallitaan robotin liikkumista, lähtöjen asettamista haluttuihin arvoihin, tulojen odottamista, datan muuttamista ja hyppäyksiä ohjelman sisällä. Robotin ohjelmointiyksiköltä valitaan käytettävät käskyt ja määritellään tarvittavat käskyihin liittyvät argumentit. Argumentit voivat sisältää numeerisia arvoja, tekstijonoja, dataa, funktiokutsuja tai lausekkeita. (5, luku 8, s. 13–14; 5, luku 9, s. 1.)

Sovelluksia voidaan testata ohjelmoinnin aikana. Ohjelman suoritus kuitenkin pysähtyy epätäydellisen käskyn kohdalla. Ohjelmaa voidaan ajaa kolmella eri tavalla: jatkuvana, yksi ohjelmakierros kerrallaan tai askelittain eteen- ja taaksepäin. Valmis ohjelma voidaan tallentaa disketille ja avata tietokoneella tekstitiedostona. Tällöin sovelluksen muokkaaminen on helpompaa kuin ohjelmoin-

tiyksiköllä tapahtuva muokkaus. Uuden ohjelman luomiseen tekstitiedostomuotoa ei kuitenkaan kannata käyttää, koska mahdollisia kirjoitusvirheitä ei välttämättä huomata sovellusta tehtäessä. Ohjelmointiyksikön avulla tapahtuvassa ohjelmoinnissa voidaan valita valmiita käskyrakenteita, jolloin uuden ohjelman luominen on helpompaa kuin käskyjen kirjoittaminen merkki kerrallaan. (5, luku 8, s. 21–23.)

3.3 I/O-järjestelmä

IRB 1400 -robotin I/O-järjestelmää hallitsee robotin I/O-tietokone. I/O-tietokone on liitetty VME-väylän avulla robottia ohjaavaan päätietokoneeseen. Robotti on mahdollista liittää ethernet-verkkoon RJ 45 -liittimellä. Myös ethernet-verkon tietoliikenneliityntä kommunikoi robotin päätietokoneen kanssa käyttäen VME-väyliä. VME-väyliä voidaan käyttää eri prosessorien väliseen kommunikointiin. (Kuva 6.)(2, s. 60; 8.)



KUVA 6. IRB 1400 -robotin I/O-järjestelmä (2, s. 175)

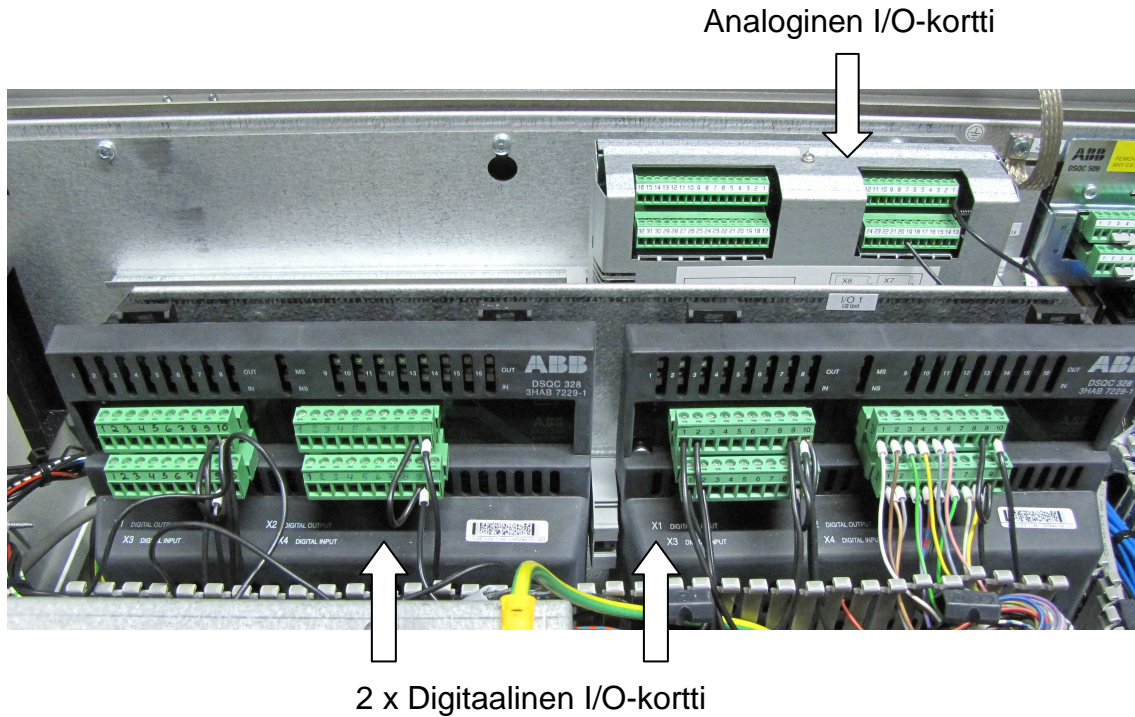
Robotin I/O-tietokone hallitsee sekä robotin sisäisten, että robottiin ulkopuolelta liitettyjen I/O-signaalien liikennettä. Robotin liikkeenohjaukseen tarkoitettu ohjelmointiyksikkö ja sovellusten tallentamiseen ja avaamiseen käytetty levyase- ma ovat liitettyinä suoraan I/O-tietokoneeseen. I/O-tietokoneeseen on liitettyinä myös kaksi sarjaporttia (RS 422 ja RS 232) sarjaliikennettä varten. Näistä RS 232 on perinteisempi esimerkiksi tietokoneissa yleisesti käytetty porttityyppi, kun taas RS 422 on teollisuuskäyttöön tarkoitettu sarjaportti, joka mahdollistaa väylässä nopean tiedonsiirron pitkilläkin etäisyyksillä. Lisäksi se mahdollistaa maksimissaan 10 dataa vastaanottavaa laitetta, kun taas perinteisemmässä RS 232:ssa vastaanottavia laitteita voi olla vain yksi. (Kuva 6.)(9.)

I/O-signaalien liikennöinti toimilaitteilta, robotin manipulaattorilta ja käyttöpaneelilta I/O-tietokoneelle tapahtuu pääasiassa CAN-väylän avulla. CAN-väylän käyttö mahdollistaa hajautettujen I/O-yksiköiden käytön robotin I/O-signaalien hallinnassa. CAN-väylä onkin alun perin suunniteltu hajautettujen ohjausjärjestelmien reaaliaikaiseen tiedonsiirtoon. CAN-väylät yleistyivät aluksi autoissa ja sen jälkeen työkoneissa, teollisuuslaitteissa ja roboteissa. (Kuva 6.)(10, s. 1.)

CAN-väylän avulla robotti käyttää hajautettuja I/O-yksiköitä. Ohjainkaapissa sijaitseva käyttöpaneeli on yksi näistä. Käyttöpaneeliyksikön kautta kulkevat robotin turvallisuuteen liittyvät signaalit I/O-tietokoneelle. CAN-väylään on myös mahdollista liittää hajautettuja kenttäväyläyksiköitä. Tällöin kenttäväyläyksikkö tekee signaalinmuunnoksen kenttäväyläsignaalin ja CAN-väylän välillä. Kenttäväyliä avulla voidaan robottiin liittää erilaisia I/O-laitteita. IRB 1400 -robotin ohjainkaapissa on valmiina liittimet kenttäväyläyksiköille Allen-Bradley Remote I/O, Interbus-S Slave ja Profibus DP Slave. Lisäksi robotissa on valmis liityntä kenttäväylälle, jolla on mahdollista toteuttaa synkronoidut anturiliitynnät kuljettimen seurantaan. (Kuva 6.)(2, s. 59.)

IRB 1400 -robotissa on käytössä myös perinteistä I/O:ta käyttäviä yksiköitä, I/O-kortteja, jotka myös ovat liitettyinä I/O-tietokoneeseen CAN-väylän avulla (kuva 6). Robottiin on mahdollista liittää sekä analogisia, että digitaalisia tulo- ja lähtökortteja, tai kombikortteja, joissa on sekä analogista, että digitaalista I/O:ta. Korttipaikkoja on robotin ohjainkaapissa neljä. Tässä insinööriyössä käytetyssä robotissa on neljästä korttipaikasta käytetty kolmea. I/O-korteista kaksi on digi-

taalisia, ja ne sisältävät kumpikin 16 digitaalista tulosignaalikanavaa ja 16 digitaalista lähtösignaalikanavaa. Yksi I/O-kortti on analoginen, ja se sisältää neljä analogista tulokanavaa sekä neljä analogista lähtökanavaa. (Kuva 7.)(2, s. 48–49.)

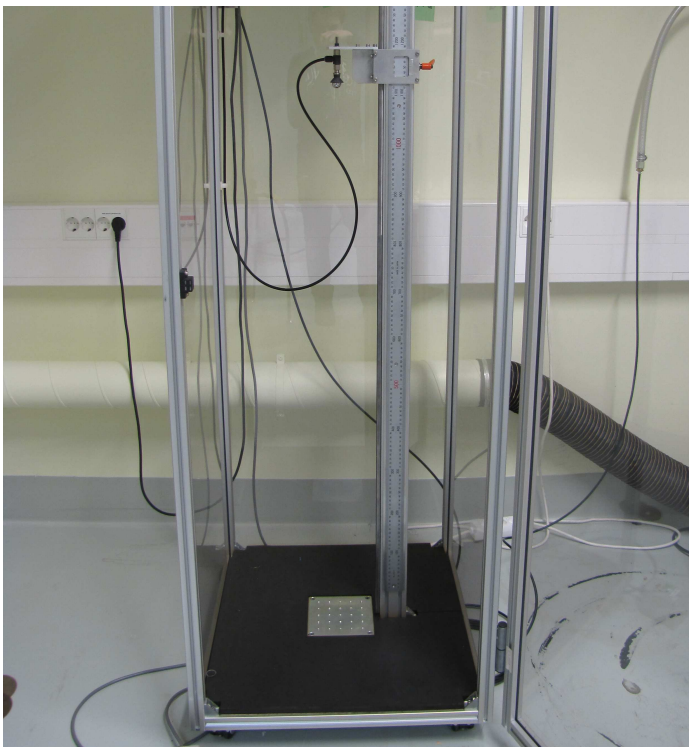


KUVA 7. IRB 1400 -robotin ohjainkaappiin asennetut I/O-kortit

4 TESTAUSLAITTEISTON SUUNNITTELU JA TOTEUTUS

4.1 Lähtötilanne

Idea ball drop -testin kehittämistä lähti liikkeelle kesällä 2011, kun olin tekemässä kyseistä testiä silloin käytössä olleella testauslaitteistolla. Alkuperäisessä testausjärjestelyssä teräskuula asetettiin imukuppiin, johon oli johdettu alipaine kuulan paikallaan pitämiseksi. Alipaine poistettiin nappia painamalla, ja teräskuula putosi vapaasti alla olevan puhelimen näytölle. (Kuva 8.)

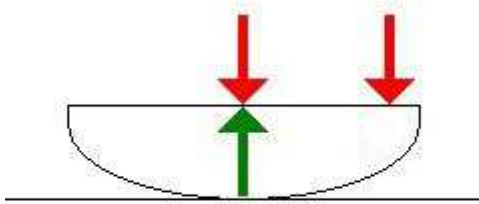


KUVA 8. Alkuperäinen ball drop -testauslaitteisto

Alkuperäisen testausjärjestelyn suurin ongelma oli kuulan osumapisteen tarkkuuden määrittämisessä (1). Testattava laite asetettiin testauskaapin pohjalle ja iskeytymispiste arvioitiin silmämääräisellä tarkkuudella (kuva 8).

Toinen ongelma alkuperäisessä laitteistossa oli kuulanpudotustestin heikko toistettavuus. Silmämääräisesti arvioitu kuulan osumakohta ei ollut välttämättä lähelläkään samaa osumakohtaa toista laitetta testattaessa. Tämä aiheutti ongelmia samanmallisten laitteiden testitulosten vertailtavuudessa. (1.)

Kolmas ongelma alkuperäisessä ball drop -laitteistossa oli iskuvoiman kohdistumisessa. Jos testattavan laitteen takaosa on muotoilultaan kupera, aiheuttaa kuulan iskeytyminen näyttöön erisuuruisen iskuvoiman riippuen siitä, tuleeko isku keskelle näyttöä vai näytön reunalle. Erisuuruiset iskuvoimat johtuvat siitä, että iskun tullessa keskelle näyttöä puhelin ei pääse liikkumaan iskun suuntaan. Iskun tullessa näytön reuna-alueille laite pääsee liikkumaan iskun voimasta iskun suuntaan, ja näin puhelimeen kohdistunut isku on heikompi puhelimen näytön kannalta tarkasteltuna. (Kuva 9.) (3.)

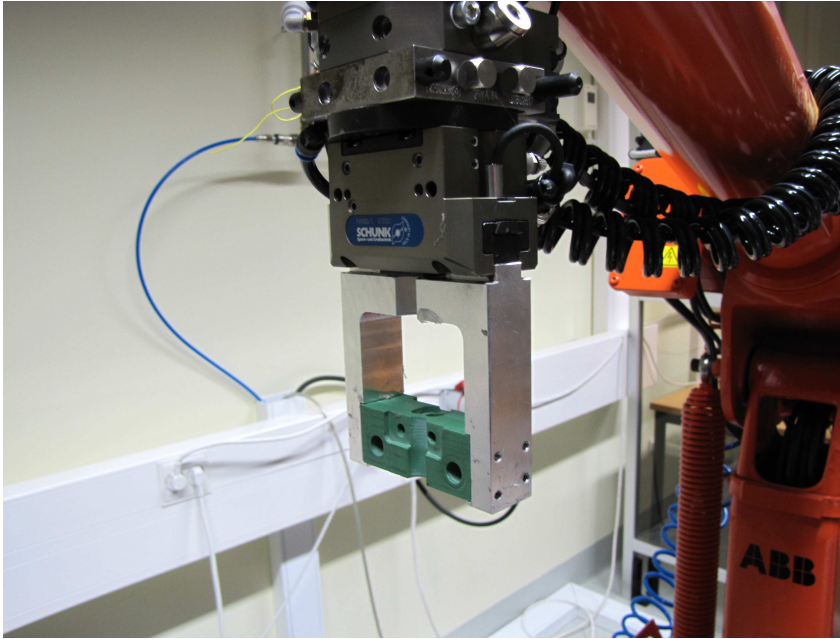


KUVA 9. Teräskuulan osuman aiheuttamat voimat ja tukivoima takaosaltaan kuperan puhelimen näyttöön

4.2 Vaatimukset testauslaitteistolle

Ball drop -testin tarkkuuden ja toistettavuuden parantamiseksi ryhdyttiin Nokia Oyj:lle suunnittelemaan insinööriyönäni teollisuusrobotin avulla toteutettavaa testiä. Teollisuusrobotin valinta testauksen apuvälineeksi oli luonnollinen, koska robotti oli valmiiksi testauskäytössä laboratoriossa. ABB:n valmistaman IRB 1400 -teollisuusrobotin paikoituksen tarkkuus on erittäin hyvä, joten se soveltui hyvin pudotuspisteiden määrittelyyn. Robotin käyttöaste ei ollut erityisen suuri, joten testauslaitteiston rakentaminen sen yhteyteen ei häirinnyt robotin muuta käyttöä.

Robotin työkaluna käytettiin pihtitarttujaa, josta robottisovelluksessa käytettiin nimitystä *tool0*. Tarttujan leukojen välissä oli sopivaksi muotoiltu vihreä muoviosa, johon oli helppo kiinnittää lisäosia esimerkiksi pulttien avulla. Robotin työkalulla teräskuula haluttiin viedä tiettyihin pisteisiin testattavan puhelimen yläpuolelle ja pudottaa kuula laitteen näytölle mahdollisimman tarkasti. (Kuva 10.)



KUVA 10. Robotin pihitarttujatyökalu tool0

Robotissa käytettyyn työkaluun kiinnitetään imukuppi, joka alipaineella ottaa kuulan poimintapisteestä ja vie aina halutun osumapisteen yläpuolelle oikealle korkeudelle. Testattavan laitteen täytyy pysyä tiukasti paikoillaan huolimatta teräskuulan aiheuttamasta iskusta. Tämän vuoksi tarvitaan puhelinta varten muotoiltu jigi. Jigin ympärille täytyy suunnitella hieman kallistettu luiska, jota pitkin teräskuula vierii pudotuksen jälkeen aina samaan pisteeseen robotin poimittavaksi. Kuulan saapuminen poimintapisteeseen tunnistetaan anturilla tai kytkimellä, joka kertoo robotille kuulan olevan valmiina poimittavaksi. Lisäksi testausjärjestelyyn voidaan liittää paineilmasylinteri, joka painaa puhelimen näyttön lukitusnäppäintä. Lukitusnäppäimen painaminen sytyttää puhelimen näyttöön taustavalon, jolloin iskun aiheuttamien mahdollisten näyttövikojen havaitseminen helpottuu.

Alun perin testauslaitteiston kiinteät mekaaniset osat jigi ja luiska ajateltiin kiinnitettävän ensin toisiinsa ja sitten robotin edessä olevaan pöytään, jossa on alumiininen uralevy kiinnitystä varten. Robotin työkalu imukuppeineen ei kuitenkaan yltänyt korkeimpaan pudotuskorkeuteensa, jos testattava puhelin sijaitsi pöydän korkeudella. Niinpä testattava laite jouduttiin laskemaan lähelle lattian tasoa, jotta haluttuihin pudotuskorkeuksiin päästiin. Jigi päätettiin kiinnittää lattialle asetettavan jalustan päälle, johon luiskakin suunniteltiin kiinnitettävän. Luis-

kaan ajateltiin liitettävän anturinpidike, joka sijoitetaan kuulan poimintapisteen kohdalle.

Ball drop -testin robottitestausjärjestely täytyi suunnitella siten, että sillä voidaan tulevaisuudessa testata useita eri puhelinmalleja aiheuttaen mahdollisimman vähän muutoksia testauslaitteiston mekaniikkaan ja robotin sovellukseen. Testausjärjestely päätettiin tehdä aluksi Nokia 603 -puhelimille sopivaksi. Nokia 603 -puhelimien käyttöjärjestelmänä toimii Symbian Belle. (Kuva 11.)



KUVA 11. Nokia 603 -matkapuhelin (4)

Nokia 603 -puhelimien yläosassa sijaitsee lukitusnäppäin, josta näytön taustavalo saadaan päälle. Nokia 603 on takaosan muotoilultaan kupera, joten jigin käyttäminen ball drop -testin yhteydessä oli välttämätöntä luotettavien ja vertailukelpoisten tulosten saavuttamiseksi.

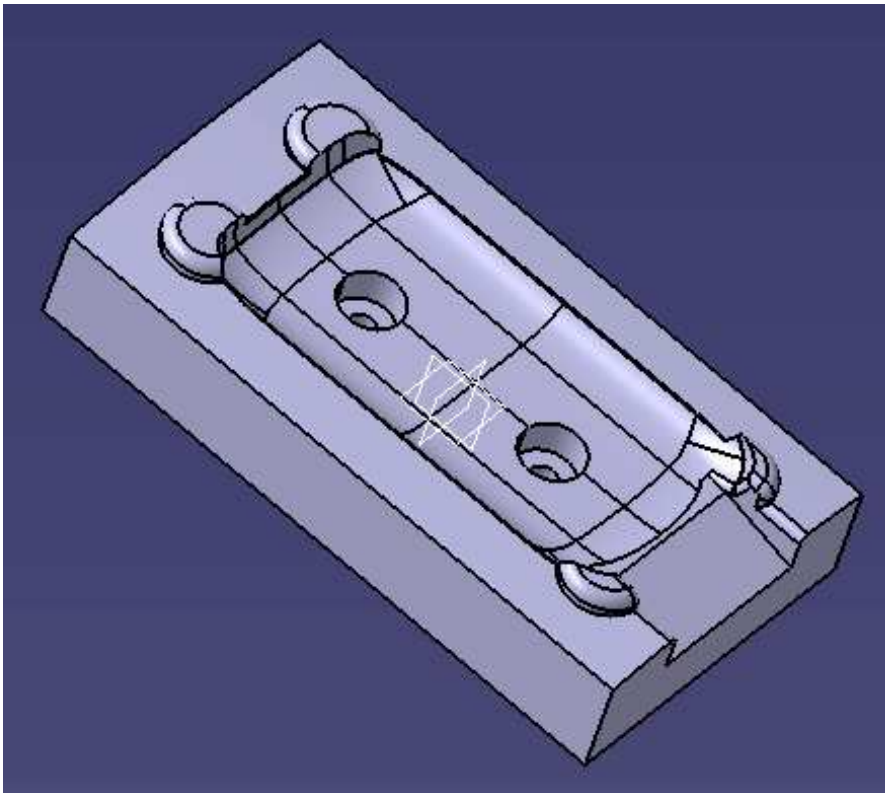
Nokia 603 -laitteen iskunkestävyyttä testattiin alkuperäisellä ball drop -testauslaitteistolla teräskuulan kimpoamisen ja pudotuksen jälkeisten liikkeiden tutkimista varten. Testauksen aikana havaittiin, että teräskuula jäi usein pudotuksen jälkeen puhelimen päälle. Tämän takia robotin työkaluun päätettiin kiinnittää osa, joka pyyhkäisee puhelimen näytön yli työntäen kuulan luiskaan, jos kuula ei saavu poimintapisteelle tietyn ajan kuluessa.

4.3 Testauslaitteiston osien suunnittelu

Ball drop -testauslaitteistoon suunnitellut mekaaniset osat suunniteltiin ja piirrettiin Catia V5 -ohjelmalla, joka tukee 3D-tyyppistä mallinnusta. Luodut kuvat tallennettiin STL-tiedostomuodossa.

4.3.1 Jigi

Jigin tehtävänä oli pitää testattava laite mahdollisimman hyvin paikoillaan testauksen aikana. Jigin ulkomitat suunniteltiin siten, että samalle alustalle voidaan jatkoissa mallintaa myös muiden testattavien puhelinmallien muodot. Jigin ulkomittojen tuli riittää isommankin puhelinmallin testaamiseen. Jigin yläosa on pyöristetty loivasti, jotta kuula ei pudotessaan jäisi jigin päälle, vaan vierisi alas luiskaan. Jigiin on tehty neljä korotettua kohtaa, joihin testattavan puhelimen muodon nurkat sijoitettiin. Nokia 603 -puhelimien takaosan muodot siirrettiin jigiin. Työstettävien muotojen syvyys täytyi arvioida silmämääräisesti. (Kuva 12.)

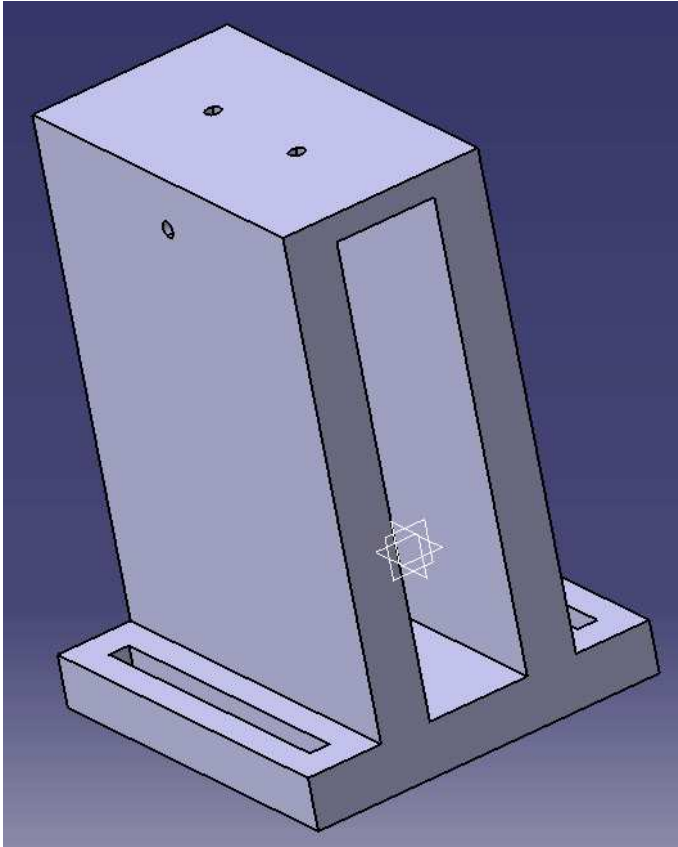


KUVA 12. Jigi Nokia 603 -puhelinmallille

Jigin päätyyn, puhelimen yläpäähän tehtiin alaspäin viettävä leikkaus, jolla oli kaksi tarkoitusta. Ensimmäinen syy oli paineilmasylinteri, joka painaa puhelimen yläosassa olevaa lukitusnäppäintä. Sylinteri tarvitsi riittävästi tilaa toimiakseen. Toisaalta on hyvä ottaa huomioon, että puhelinta voidaan joutua lataamaan kesken testauksen, ja laturin liitin on myös puhelimen yläosassa. Jigiin tehtiin myös kaksi reikää, joiden kautta jigi voitiin kiinnittää jalustaan pulteilla ja mutteilla. (Kuva 12.)

4.3.2 Jalusta

Jigille tarvittiin myös jalusta, jotta testattava puhelin tulisi mahdollisimman korkealle. Jalustan ulkomittoja suunniteltaessa tuli ottaa huomioon testauksessa käytettävän robotin rajalliset liikeradat. Puhelin voitiin sijoittaa korkeintaan 28 cm korkeudelle maasta, jotta robotin työkalu yltää korkeimpaan pudotuskorkeuteensa. Jalustasta tehtiin hieman matalampi, jotta robotin ei tarvitse liikkua toimialueensa ääri rajoilla ja mahdolliset muutokset mekaniikkaan ovat tulevaisuudessa helpompia. Jalustan yläosan mitat ovat samat jigin ulkomittojen kanssa. (Kuva 13.)

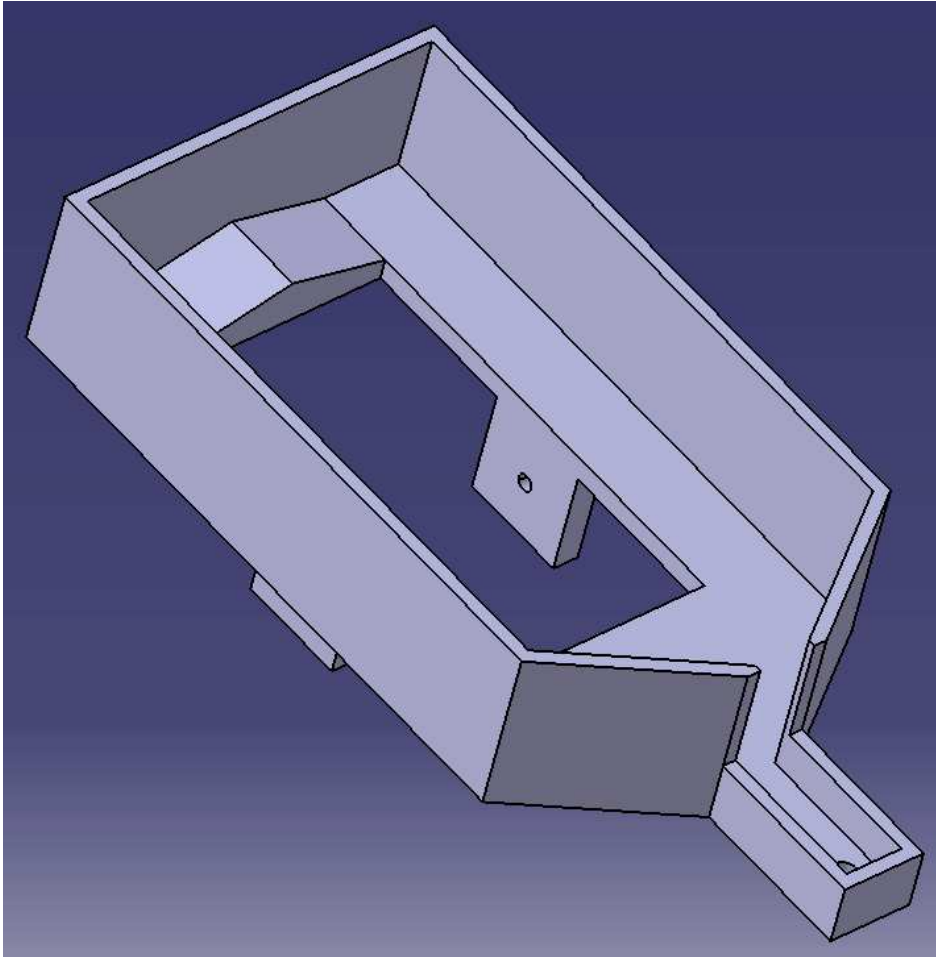


KUVA 13. Testauslaitteiston jalusta

Jalustan pohja suunniteltiin leveämmäksi kuin sen yläosa, jotta testausalusta olisi tukevampi. Lisäksi pohjaan tehtiin pitkittäiset reiät jalustan kiinnittämiseksi lattiaan. Jalustan päälle suunniteltiin kaksi reikää jigien kiinnittämistä varten. Lisäksi jalustan kylkiin tehtiin reiät jigien ja jalustan ympärille tulevan luiskan kiinnittämistä varten. Jalustasta suunniteltiin ontto, jotta muiden osien kiinnitys onnistuisi pulttien avulla. (Kuva 13.)

4.3.3 Luiska

Luiskan ainoa tehtävä testausjärjestelyssä oli johtaa teräskuula pudotuksen jälkeen aina samaan pisteeseen poimittavaksi. Luiskan keskelle suunniteltiin aukko, joka oli leveydeltään sama kuin jigi ja jalusta, mutta pituudeltaan 10 mm näitä pitempi. Kun luiska kiinnitettiin kahdesta korvakkeestaan pulteilla jalustaan, saatiin luiskaa kallistettua hieman poimintapisteen suuntaan, koska aukko oli hieman jalustaa pitempi. Kallistuksella saatiin teräskuula vierimään luiskan kaapeaan päähän poimintapisteeseen. (Kuva 14.)

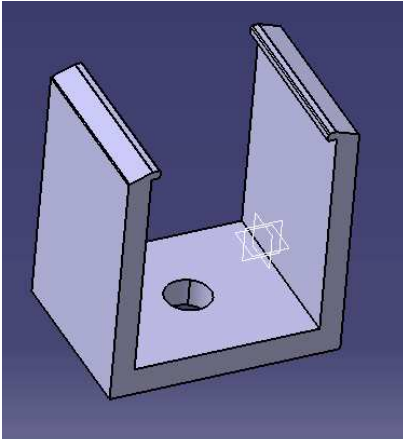


KUVA 14. Luiska

Luiskan yläpäähän tehtiin pienet kallistukset, joiden tehtävänä oli ohjata kuula luiskan reunoille, jos kuula vierii testattavan laitteen päältä pois kyseisestä päädyistä. Luiskan reunoista tehtiin melko korkeat, koska teräskuula voi kimmota puhelimen näytöltä jonkin verran. Poimintapisteen kohdalla reunat tehtiin hieman teräskuulan halkaisijaa matalammiksi, jotta kuulan poiminta onnistui robottiin asennetun imukupin avulla. Teräskuulan poimintapisteen alapuolelle suunniteltiin halkaisijaltaan sopivan kokoinen reikä kuulan tunnistavaa anturia varten. Kun reikä tehtiin juuri kuulan säteen etäisyydelle luiskan päädyistä, teräskuula jäi aina pudotuksen jälkeen reiän päälle samaan poimintapisteeseen. Tämä oli erityisen tärkeää teräskuulan osumapisteiden toistettavuuden kannalta. (Kuva 14.)

4.3.4 Anturinpidike

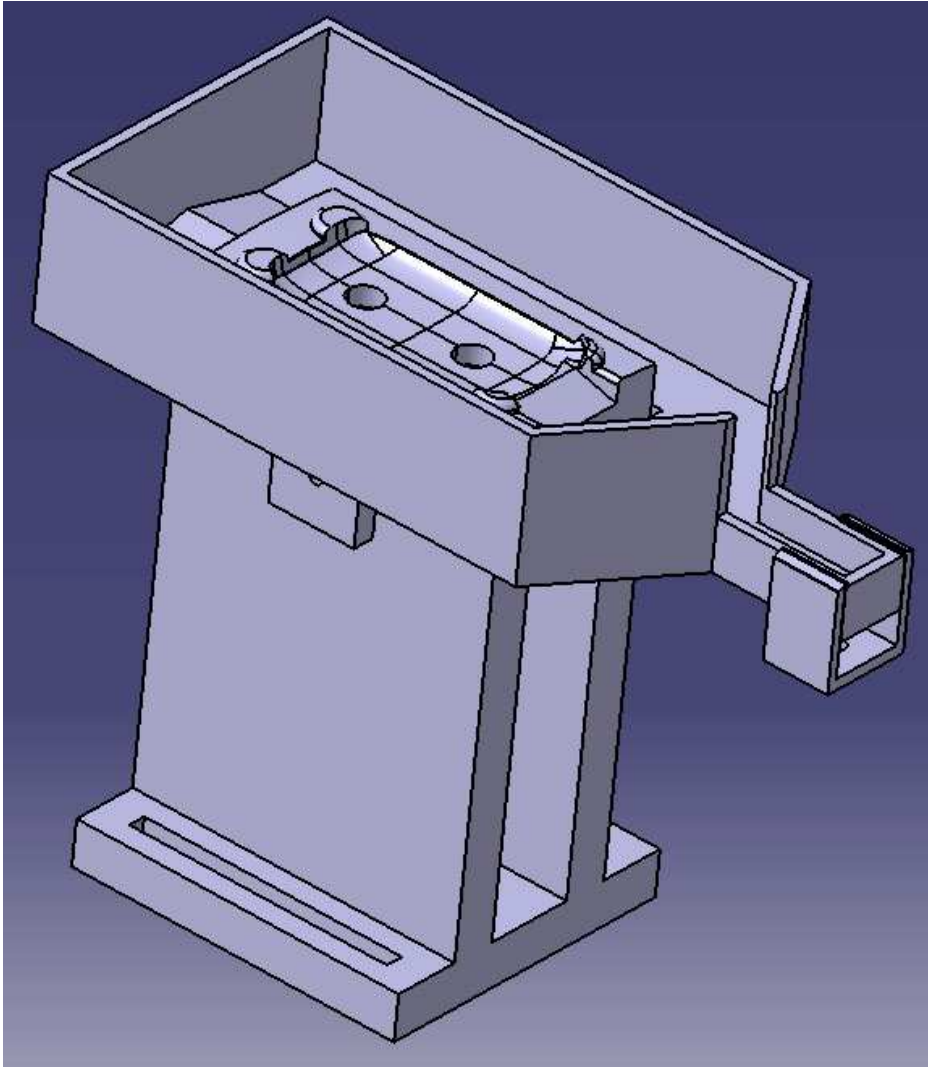
Valitulle anturille täytyi suunnitella myös pidike, koska sopivaa pidikettä oli hankala löytää valmiina. Pidikkeeseen suunniteltiin halkaisijaltaan sopivan kokoinen reikä anturia varten. Pidikkeen sivuille suunniteltiin pienet ulokkeet, jotka pitivät pidikkeen kiinni luiskassa. Ulokkeiden tuli olla mahdollisimman pienet, jotta ne eivät olisi imukupin tiellä robotin poimiessa teräskuulan. (Kuva 15.)



KUVA 15. Anturinpidike

4.3.5 Osien yhdistäminen ja valmistus

Yksittäisten osien suunnittelun ja mallinnuksen jälkeen suunnitellut osat liitettiin yhteen virtuaaliseksi malliksi laitteistokokonaisuuden tarkastelua varten (kuva 16). Tällä voitiin varmistaa, että osien mitat olivat oikeat toistensa suhteen. Testauslaitteistolle suunnitellut mekaaniset osat päätettiin valmistaa Nokia Oyj:n omalla pikamallinnuskoneella. Kyseisellä pikamallinnuskoneella ei voida valmistaa juurikaan suunniteltuja osia suurempia kappaleita (15). Kustannussyistä oli järkevintä käyttää yrityksen omaa laitteistoa hyödyksi osien valmistamisessa.



KUVA 16. Testauslaitteistoon suunnitellut osat yhdistettynä virtuaaliseksi malliksi

4.4 Laitevalinnat

4.4.1 Teräskuulan tunnistaminen

Pudotuksen jälkeen teräskuulan oli määrä vieriä poimintapisteeseen robotin poimittavaksi uutta pudotusta varten. Teräskuulan tunnistamista varten tarvittiin anturi, joka lähettää signaalin robotin I/O-tietokoneelle teräskuulan ollessa poimintapisteessä.

Anturin tuli olla induktiivinen, sillä induktiivinen anturi tunnistaa metalleja. Lisäksi induktiiviset anturit ovat mekaanisesti erittäin kestävä, sillä ne eivät koske lainkaan tunnistettavaan materiaaliin. Anturin tuli olla fyysiseltä kooltaan melko pie-

ni, jotta se sopisi luiskassa olevaan reikään, johon teräskuula vierisi poimittavaksi. Vastaavasti anturin kytkentäetäisyyden tuli olla riittävä. Anturin tuli olla tyypiltään 3-johtiminen. Robotin I/O-kortit toimivat 24 voltin tasajännitteellä, joten se tuli huomioida anturin valinnassa. Anturin lähdön tuli olla PNP-tyyppinen, koska myös robotin I/O-korttien tulot ovat PNP-tyyppisiä. (13; 18, luku 12, s. 7-2.)

Testauslaitteistossa käytettäväksi induktiiviseksi anturiksi valittiin Omronin valmistama anturi E2A-S08KN04-WP-B1 2M (kuva 17). Anturi on varustettu M8-kierteellä. Kytkentäetäisyys kyseisellä anturilla on 4 mm, mikä on riittävä teräskuulan vieressä poimintapisteeseen. Anturi on myös PNP-tyyppinen ja kolmi-johtiminen. (Liite 1.)(14.)



KUVA 17. Valittu induktiivinen anturi.

4.4.2 Pneumaattiset laitteet

Teräskuulan poimimista varten tarvittiin tyhjiögeneraattori, joka muuttaa magneettiventtiileiltä tulevan paineilman alipaineeksi. Tyhjiögeneraattorin toiminta perustuu tulevan paineilman virtaukseen, joka ohjataan generaattorin kyljistä ulos. Tämä aiheuttaa alipaineen päävirtauksen alapuolelle liitettyyn paineilmalietkuun ja sitä kautta tyhjiömuriin. Tyhjiögeneraattorin tuli olla maksimimuvoiltaan riittävä, jotta tyhjiömuri saa pidettyä teräskuulan tiiviisti paikallaan. Imuvoiman suuruuteen vaikuttaa luonnollisesti myös syötettävän ilmanpaineen suuruus. Tyhjiömurin tuli olla halkaisijaltaan lähellä teräskuulan halkaisijaa (20 mm), jotta kuulan poiminta onnistuisi mahdollisimman helposti, ei-

vätkä tyhjiöimurin reunat osuisi kuulaa poimittaessa luiskan laitoihin. Toisaalta tyhjiöimurin halkaisija tuli olla riittävän suuri, jotta imupintaa olisi mahdollisimman paljon.

Tyhjiögeneraattoriksi valittiin SMC:n valmistama generaattori malliltaan EZH10BS-06-06. Sen maksimi-imuteho on 24 l/min. Valittu tyhjiögeneraattori kehittää maksimissaan 88 kPa:n alipaineen. Tämä riittää hyvin teräskuulan paikoillaan pitämiseksi. (Liite 2.)(16.)

Tyhjiöimuriksi valittiin SMC:n valmistama imuri malliltaan ZPT20UN-B01 (liite 3). Tyhjiöimurissa on kiinni jatkopala, johon voitiin liittää paineilmaletku. Tyhjiöimurin imukupin halkaisija on teknisten tietojen mukaan 23 mm (17). Kuvassa 18 on esitettynä valittu tyhjiöimuri jatkopaloineen liitettynä paineilmaletkulla tyhjiögeneraattoriin.



KUVA 18. Valittu tyhjiöimuri ja tyhjiögeneraattori

Testauslaitteistossa tarvittiin myös paineilmasylinteri, jonka tehtävänä oli painaa robotin antamasta käskystä näytön lukitusnäppäintä, jolloin näytön taustavalo syttyy ja näyttövikojen havaitseminen helpottuu. Oleellista paineilmasylinterin valinnassa oli sylinterin koko ja riittävä paine lukitusnäppäimen painamista varten. Käytettäväksi paineilmasylinteriksi valittiin SMC:n valmistama sylinteri mallia CJPB6-15H6 (kuva 19)(liite 4).



KUVA 19. Valittu paineilmasylinteri ja pidike

Testauslaitteisto tuli suunnitella siten, että sitä voitaisiin käyttää myöhemmin myös muiden puhelinmallien testauksessa. Näytön lukitusnäppäin ei sijaitse aina samassa paikassa eri puhelinmalleissa, joten paineilmasylinterin pidikettä ei voitu sijoittaa pysyvästi tiettyyn paikkaan. Sylinterin pidike päätettiin tehdä alumiinista ja kiinnitys jalustaan tehtiin kuumaliiman avulla (kuva 19).

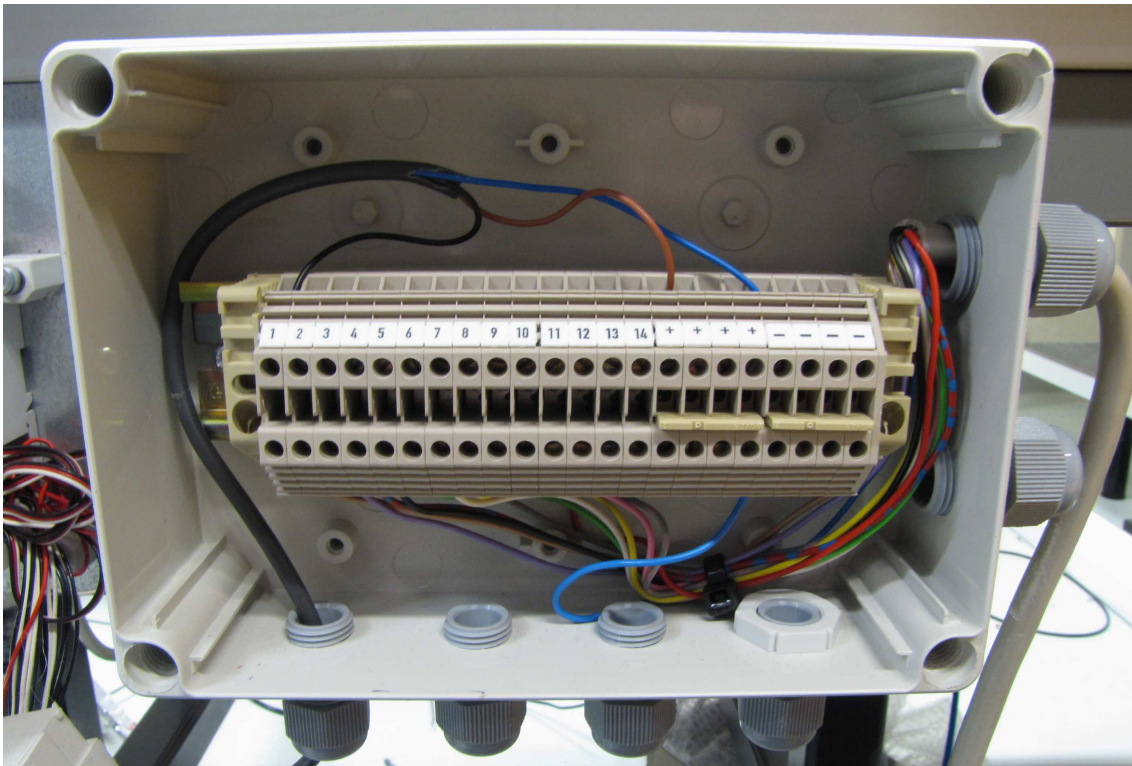
4.5 Kytkenät

4.5.1 Sähköiset kytkennät

Valitun induktiivisen anturin lähdön tuli olla PNP-tyyppinen. Tämä johtui siitä, että robotin ohjainkaapissa sijaitsevien digitaalisten I/O-korttien signaalissa kuorma muodostuu +24 V:n ja nollapotentiaalivälille, eli I/O-kortitkin ovat PNP-tyyppisiä (18, luku 12, s. 7-2). Näin ollen anturin tunnistuksessa teräskappaleen syntyvä 24 voltin potentiaaliero signaalijohtimen ja nollajohtimen välille.

Valittu induktiivinen anturi toimi siis digitaalisena tulona robotin automaation näkökulmasta. Robotin kontrollerilta oli valmiiksi tuotu järjestelmän käyttöjännite ja nollapotentiaali sekä osa robotin digitaalisista tuloista robotin työskentelyalueelle kytkentäkoteloon KK2. Valittu kolmijohtiminen induktiivinen anturi liitettiin kytkentäkotelossa liittimiin +24 V (ruskea), 0 V (sininen) ja signaalijohdin

KK2.1:een (musta). Valitun anturin signaalin lukemiseen robotti käytti automaatiopositiota *diT9*. (Kuva 20.)



KUVA 20. Anturin kytkentä kytkentäkotelo KK2:ssa

4.5.2 Pneumaattiset kytkennät

Testauslaitteistoon tarvittiin pneumaattisia kytkentöjä tyhjiömuria ja paineilmasylinteriä varten. Kumpaakin ohjattiin digitaalisina lähtöinä. Kuusi robotin digitaalista lähtöä oli tuotu valmiiksi robotin työskentelyalueelle kytkentäkotelon KK2 viereen magneettiventtiileille. Kyseiset magneettiventtiilit ohjasivat venttiileihin liitettyjä pneumaattisia lähtöjä robotin I/O-tietokoneelta saatujen signaalien mukaan. (Kuva 21.)



KUVA 21. Magneettiventtiilit digitaalilähtöjen pneumatiikan ohjaukseen

Magneettiventtiilien pneumaattisista lähdöistä paineilma johdettiin modulaarisille paineensäätimille, joihin oli kiinnitetty manometrit. Paineensäätimien avulla voitiin säätää ilmanpaineet käytön mukaan sopiviksi magneettiventtiilien ollessa auki. Paineen säätö sopivaksi oli tärkeää, jotta tyhjiömuri aiheuttaisi riittävästi alipainetta teräskuulan paikoillaan pitämiseksi ja paineilmasylinteri painaisi riittävällä voimalla lukitusnäppäintä, mutta ei kuitenkaan rasittaisi sitä turhaan.

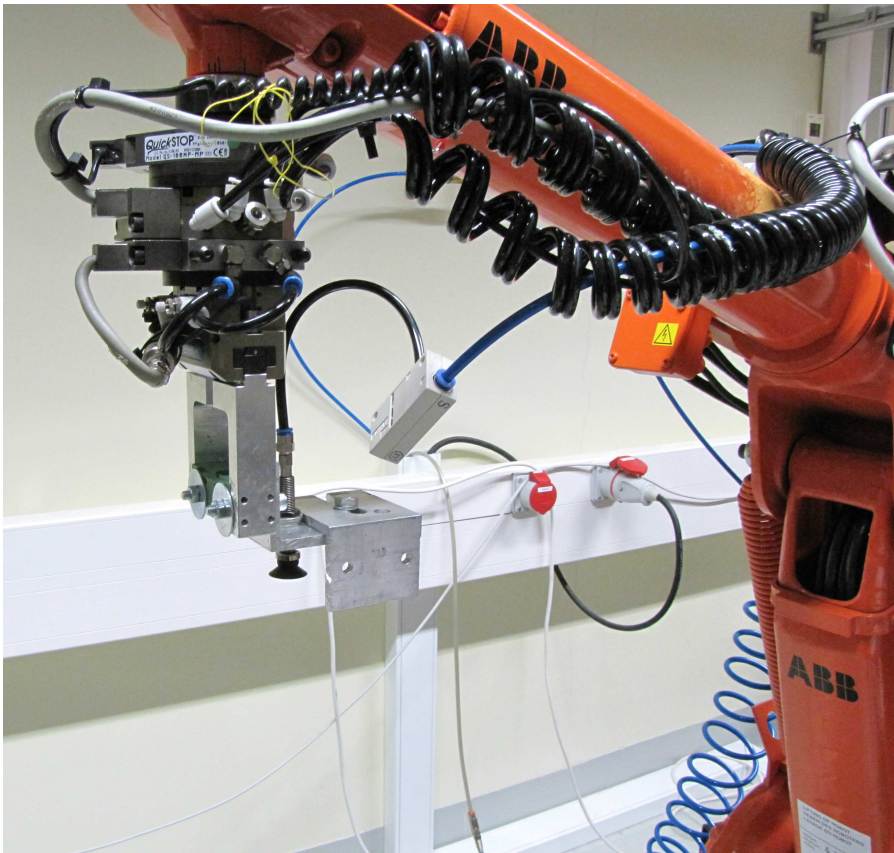
Paineensäätimiltä paineilmaletkut johdettiin toimilaitteille. Magneettiventtiili V1 liitettiin paineensäätimen ja tyhjiögeneraattorin kautta robotin manipulaattoriin kiinnitettyyn imukuppiin. Robotin sovelluksessa kyseisen digitaalilähdön automaatiopositio on *doVent1*. Magneettiventtiiliin V3 kiinnitettiin paineensäätimen kautta testausjalustaan kiinnitetty paineilmasylinteri. Robotin sovelluksessa paineilmasylinteriä ohjataan automaatioposition *doVent3* avulla.

4.6 Valmis testauslaitteisto

Lopullinen testauslaitteisto koostui kahdesta kokonaisuudesta, testaukseen käytetystä teollisuusrobotista imukuppeineen sekä suunnitellusta testausalustasta,

johon testattava laite asetettiin. Nämä kaksi testaukseen tarvittavaa osaa yhdistettiin yhdeksi testauskokonaisuudeksi automaation avulla.

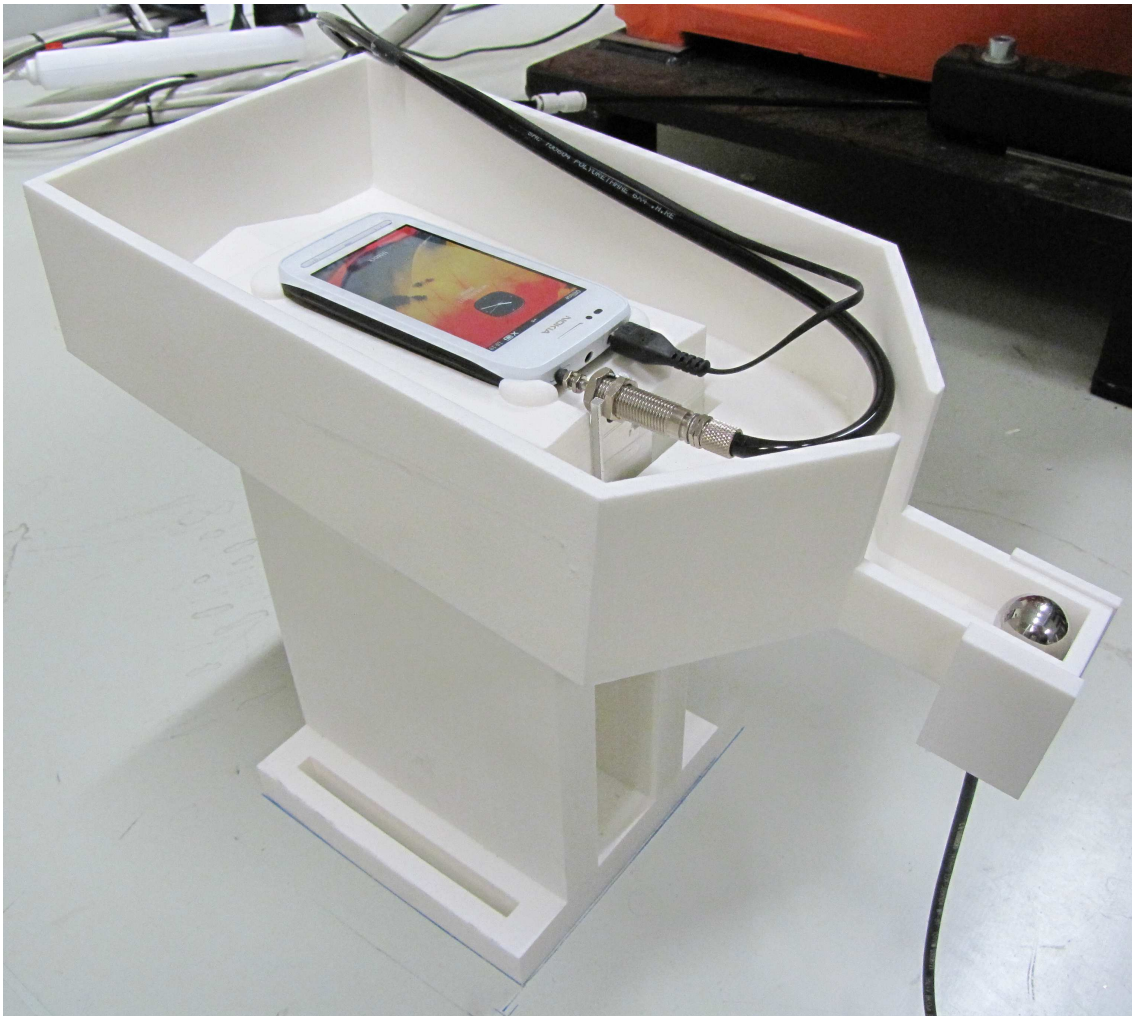
Teollisuusrobotin pihtitarttujassa kiinni olleeseen muoviosaan kiinnitettiin alumiininen kulmapala, johon teräskuulan poimintaan käytetty imukuppi asennettiin. Pihtitarttujaan kiinnitettyyn kulmapalaan liitettiin myös toinen kulmapala. Tämän alumiiniosan tehtävänä oli pyyhkäistä testattavan puhelimen näytön yli ja työntää teräskuula luiskaan, jos kuula jäisi pudotuksen jälkeen näytön päälle. (Kuva 22.)



KUVA 22. Imukuppi robotin tarttujaan asennettuna

Catia V5 -ohjelman avulla suunnitellut ja mallinnetut osat valmistettiin Nokia Oyj:n omalla pikamallikoneella. Osien liittäminen toisiinsa onnistui ongelmitta. Jigi ja luiska kiinnitettiin jalustaan pulteilla ja muttereilla. Anturinpidike pysyi hyvin paikallaan pienien ulokkeiden avulla, kun valittu anturi oli asennettu pidikkeeseen. Puhelimen näytön lukitusnäppäintä painavan paineilmasylinterin pidi-

ke kiinnitettiin kuumaliimalla jalustaan. Puhelimen lataaminen testauksen aikana onnistui jigin avoimen päädyn ansiosta. (Kuva 23.)



KUVA 23. Valmis testausalusta

Testausalustan oikea paikoitus robottiin nähden oli erityisen huomionarvoinen asia testauskokonaisuutta suunniteltaessa. Testausalusta täytyi asettaa mahdollisimman tarkasti robotin maailmakoordinaatiston mukaisten x- ja y-akselien suuntaisesti. Tämä johtui siitä, että ball drop -testin osumapisteitä määriteltäessä pudotuspisteet ohjelmoitiin robotille x-, y- ja z-akselien suuntaisina etäisyyksinä näytön nurkkapisteeseen nähden.

Alusta saatiin asetettua oikeaan asentoon kiinnittämällä robotin tarttujaan kynä ja liikuttamalla sitä robotin maailmakoordinaatiston mukaisten x- ja y-akselien suuntaisesti lattian tasossa. Näin saatiin piirrettyä lattiaan x- ja y-akselien suun-

taiset viivat, joiden mukaan testausalusta voitiin asettaa mahdollisimman tarkasti oikeaan asentoon. Jalustan kiinnitys lattiaan oli tarkoitus tehdä pulteilla. Robottikonaisuus tullaan kuitenkin siirtämään lähitulevaisuudessa toiseen paikkaan, joten ei ollut perusteltua asentaa jalustaa vielä kiinteästi lattiaan. (Kuva 23.)

5 ROBOTIN OHJELMOINTI

5.1 Osumapisteiden määrittely

Teräskuulan iskeytyminen puhelimen näyttöön mahdollisimman tarkasti oli olennaisena vaatimuksena insinööriyössä. Osumapisteiden määrittäminen tehtiin yhden näytön nurkkapisteen (sovelluksessa *pnurkka*) perusteella. Nurkkapiste *pnurkka* on se robotille määritelty piste, jossa robotin manipulaattoriin kiinnitetyn imukupin keskikohta on täsmälleen puhelimen näytön nurkan yläpuolella. Pudotuspisteet robotille annettiin maailmakoordinaatiston x-, y- ja z-akselien suuntaisina etäisyyksinä robotille määriteltyyn nurkkapisteeseen nähden. Tästä johtuen osumapisteiden määrittämisen kannalta oli erityisen tärkeää, että puhelinta pitävä alusta oli mahdollisimman hyvin x- ja y- akselien suuntaisesti kiinnitettyinä lattiaan.

Puhelimen näytön nurkkapisteeksi valittiin näytön oikea alakulma, koska tällöin etäisyyksien määrittäminen nurkkapisteistä x- ja y-koordinaateilla voitiin tehdä positiivisina lukuina käytössä olevassa robotin maailmakoordinaatistossa. Teräskuulan osumapisteet laskettiin puhelimen näytön koon mukaan etäisyyksinä näytön nurkkapisteestä *pnurkka*. (Kuva 24.)



KUVA 24. Teräskuulan osumapisteiden x- ja y-suuntaiset etäisyydet näytön nurkkapisteestä

Näytön nurkkapisteen määrittäminen robotille tehtiin siten, että robotin työkalussa kiinni ollut imukuppi ohjattiin puhelimen näytön päälle niin, että imukupin kaksi reunaa osuivat kumpikin täsmälleen puhelimen näytön reunaan. Pisteen x- ja y-koordinaatit kirjattiin ylös, ja kun imukupin halkaisija tiedettiin, voitiin x- ja y-koordinaattien arvoista vähentää puolet imukupin halkaisijasta. Näin saatiin selville se maailmakoordinaatiston x-y-koordinaatiston piste, jossa imukupin keskikohta on täsmälleen puhelimen näytön nurkkapisteen yläpuolella.

Teräskuulan pudotuskorkeudet määriteltiin siten, että imukupissa kiinni ollut teräskuula tuotiin puhelimen näytön päälle niin, että kuula kosketti näyttöä. Tämän jälkeen kirjattiin ylös robotin z-suuntainen sijainti ja laitettiin se nurkkapis-

teen *pnurkka* z-arvoksi. Näin saatiin nurkkapisteen *pnurkka* avulla määritettyä myös pudotuskorkeudet z-akselin suuntaisina etäisyyksinä puhelimen näytöstä.

5.2 Robotin toimintasekvenssi

Robotin päätoimintoja testin aikana voidaan kuvata sekvenssinä. Robotti suoritti sekvenssin läpi jokaisella kuulun pudotuskerralla ja palasi takaisin sekvenssin alkuun. Testauksen aloituksen jälkeen sovellus suoritti testin aikana 36 kertaa sekvenssinsä ja ilmoitti tämän jälkeen testin olevan päättynyt. Robotin toimintasekvenssi, jota liitteessä 5 oleva sovellus noudattaa, on päätoiminnoiltaan seuraavanlainen.

- Robotti jäykistää työkalun *tool0* ohjelmallisesti paineilmalla.
- Robotti ajaa imukupin kotiasemaan poimintapisteen yläpuolelle. Robotti kysyy käyttäjältä, aloitetaanko testi.
- Robotti tarkastaa, onko kuula poimintapisteessä. Jos on, siirrytään eteenpäin.
- Robotti poimii kuulun imukuppiin. Jos kuula on poistunut poimintapisteestä, siirrytään eteenpäin. Jos kuula on yhä poimintapisteessä, yritetään se poimia aina uudelleen.
- Robotti ajaa imukupin takaisin kotiasemaan.
- Robotti siirtyy yhteen 36:sta pudotuspisteestä riippuen siitä, monesko ohjelmasilmukka on suorituksessa.
- Robottiin liitetty paineilmasyylinteri painaa näytön taustavalon päälle puhelimen lukitusnäppäimestä ja samalla painaa puhelinta jiggin reunaa vasten.
- Robotti pudottaa kuulun.
- Robotti vapauttaa lukitusnäppäintä painavan paineilmasyylinterin.
- Jos kuula ei saavu viidessä sekunnissa poimintapisteelle, robotin työkaluun kiinnitetty pyyhkäisijä liikkuu näytön yli, jolloin kuula putoaa luiskaan ja vierii poimintapisteeseen.
- Robotti ajaa imukupin kotiasemaan.
- Jos suorituksessa ei ole pudotuskorkeuden viimeinen pudotuspiste, sovelluksen suoritus siirtyy sekvenssin alkuun.

- Pudotuskorkeuden viimeisen pudotuspisteen jälkeen robotti kysyy käyttäjältä, jatketaanko seuraavaan pudotuskorkeuteen. Jos jatketaan, sovelluksen suoritus siirtyy sekvenssin alkuun.
- Viimeisen pudotuskorkeuden viimeisen pudotuspisteen jälkeen robotti ilmoittaa testin päättyneen. Ohjelman suoritus päättyy.

5.3 Ohjaava sovellus

Robottia ohjaavan sovelluksen ohjelmointi tehtiin RAPID-kielellä. Ohjelmointi tehtiin robotin ohjelmointiyksikköä käyttäen. Käskyt sovellukseen valittiin ohjelmointiyksikön käskyvalikoista. Jokaiselle käskylle määriteltiin tarvittavat parametrit. Liitteessä 5 on esitetty robottia testin aikana ohjaava sovellus kokonaisuudessaan. Sovelluksen testaus suoritettiin useaan kertaan ohjelmoinnin yhteydessä.

Luodun ohjelman myöhemmän muokkauksen kannalta oli järkevintä toteuttaa sovellus siten, että toistuvia käskyjä tulisi mahdollisimman vähän. Mahdollisimman tiivis sovellus oli olennaista sovelluksen myöhemmän muokkauksen helpottamiseksi. Sovellusta tullaan muokkaamaan jatkossa, kun muita puhelinmaljeja otetaan testattavaksi toteutetulla testauskokonaisuudella. Sovelluksen tiivis rakenne tuotti hankaluuksia ohjelman selkeyden kannalta. Ohjelmointiyksiköllä tapahtuvassa ohjelmoinnissa ei ole mahdollisuutta liittää kommenttitekstejä käskyihin, joten näinkään ei voitu ohjelmaa selkeyttää. Kommenttitekstejä voitiin lisätä valmiiseen sovellukseen tietokoneella tekstitiedostomuodossa, kuten liitteessä 5 on tehty vihreällä värillä. Tästä ei kuitenkaan ole hyötyä, jos ohjelmaa tarkastellaan ohjelmointiyksiköllä.

Sovelluksen alussa on määriteltynä sovelluksessa käytetty data. Robotin paikoitukseen käytetyt pisteet ja robotin asento paikoituksessa nähdään *CONST robotarget* -määrittelyissä. Sopivat *robotarget*-paikoituspisteet määriteltiin ajamalla robotti manuaalijolla haluttuihin paikkoihin. Ennen paikoituspisteiden määrittämistä robotin työkalu imukuppeineen ajettiin robotin akseli kerrallaan sellaiseen asentoon, että robotin työkalu oli maailmakoordinaatiston mukaisesti x-, y- ja z-akselien suuntaisesti. Tämän jälkeen robotin työkalua liikutettiin ainoastaan maailmakoordinaatiston akselien suuntaisesti paikoituspisteisiin. Näin voitiin

varmistaa, että työkalun asento oli kaikissa paikoituspisteissä sama. Lisäksi datan määrittelyissä on esitelty ohjelmointiyksikön funktionäppäimien avulla valittavat muuttujat. (Liite 5.)

Testin aikana robottia ohjaava sovellus päätettiin tehdä yhdellä päärutinilla. Alirutiinien käyttämisestä luovuttiin, koska pudotuspisteiden valinta sovelluksen suorituksen aikana oli yksinkertaisempaa toteuttaa kahdella sisäkkäisellä for-silmukalla. Sovelluksen alkupuolella suoritettiin teräskuulan poiminta, jonka jälkeen robotin imukuppi kuului siirtyä oikeaan pudotuspisteeseen. Ulompi for-silmukka määritteli pudotuskorkeuden ja sisempi määritteli sen, kuinka monennen osumapisteen yläpuolelle robotti teräskuulan liikuttaa. Näiden for-silmukoiden laskureina olevia arvoja i ja j käytettiin if-lausekkeissa, jotka määrittivät, mihin pudotuspisteeseen imukupin tuli milloinkin siirtyä for-silmukoiden suoritusnumeroiden mukaisesti. (Liite 5.)

Kun teräskuula oli oikeassa pudotuspaikassa, painoi paineilmasylinteri puhelimesta näytön taustavalon päälle, minkä jälkeen teräskuula putosi näytölle. Jos kuula jäi näytön päälle, pyyhkäisi robotti sen luiskaan ja odotti, että kuula saapui poimintapisteelle valmiina seuraavaa pudotusta varten. (Liite 5.)

6 TYÖN ARVIOINTI

6.1 Tarkkuus ja toistettavuus

Tässä insinööriyössä olennaisena osana oli parantaa ball drop -testin iskupisteiden tarkkuutta ja toistettavuutta verrattuna vanhaan testausjärjestelyyn. Iskkeytymispisteiden todellista tarkkuutta on erittäin vaikea määrittää. Iskukohdan tarkkaan määrittämiseen ei löytynyt mitään sopivaa tapaa, joten teräskuulan osumapisteiden tarkkuus arvioitiin itse. (1.)

Teräskuulan vieressä aina samaan kohtaan poimintapisteellä olevan reiän päälle, kuula asettuu täsmälleen samaan kohtaan imukupissa, joten tämän ei pitäisi olla ongelma toistettavuuden kannalta. Teräskuulan osumapisteen toistettavuuteen vaikuttaa ainoastaan robotin asemoinnin toistettavuus. Robotti siirtyy ohjelmoituun pisteeseen keskimäärin 0,05 mm:n tarkkuudella (11). Tämä tarkkuus on siis robotista aiheutuvan epätarkkuuden keskimääräinen suuruus testausjärjestelyssä.

Toinen osumapisteiden tarkkuuteen vaikuttava seikka liittyy siihen, kuinka tarkkaan puhelimen näyttö on robotin maailmakoordinaatiston mukaisten x- ja y-akselien suuntaisesti. Epätarkkuus tässä riippuu siitä, kuinka tarkkaan jalusta on saatu asetettua lattiaan piirrettyjen x- ja y-akselien suuntaisten viivojen mukaan. Toinen epätarkkuuden aiheuttaja puhelimen suuntauksessa voi olla epätarkkuudet jalustan pikamallinnuksessa, eli onko esimerkiksi suoraksi piirretty jalustan reuna todella suora.

Kolmas tekijä pudotuspisteen tarkkuuden kannalta liittyy puhelimen näytön nurkkapisteen määrittämiseen. Osumapisteen tarkkuuden kannalta on olennaista, kuinka hyvällä tarkkuudella nurkkapiste saatiin määritettyä imukupin avulla ja puhelimen näytön reunojen perusteella. Tällainen virhe on systemaattinen, koska kaikki osumapisteen määrittämisen perusteella.

Oman arvioni mukaan kaksi jälkimmäistä tekijää ovat olennaisia osumapisteen tarkkuuden kannalta. Robotin asemoinnin toistettavuuden keskimääräinen virhe on niin pieni (0,05 mm), että se voidaan jättää huomioimatta. Puhelimen

näytön saamisesta x- ja y-akselien suuntaiseksi sekä näytön nurkkapisteen määrittämisestä aiheutuvien virheiden suuruudeksi arvioisin yhteenlaskettuina maksimissaan 2 mm.

Näiden kahden eniten virhettä aiheuttavan tekijän parantamisen mahdollisuuksia pohdittiin työn loppuvaiheessa. Nurkkapisteen määrittämisestä aiheutuvasta epätarkkuudesta voitaisiin päästä, jos myös robotin työkaluun liitetty osa, joka pitelee imukuppia, suunniteltaisiin ja valmistettaisiin samalla tavalla kuin testausalustan osat. Tällöin imukupin pidikkeeseen voitaisiin suunnitella tappi, joka pudotuspisteiden määrittelyä varten ohjattaisiin jigiin suunniteltuun reikään (12). Imukupin pidikkeen ja jigin välinen asento voitaisiin määrittellä, jos pidikkeessä olisi kaksi tappia ja jigissä kaksi reikää. Tällöin jigin ja imukupin pidikkeen välinen asento tiedettäisiin tarkasti. Tässä järjestelyssä puhelimen näytön osumapisteiden ja imukupin väliset etäisyydet tiedettäisiin jo jigin suunnitteluvaiheessa ja pudotuspisteiden paikoituksen arvot voitaisiin laskea robotin näyttämistä koordinaateista, kun tapit on ajettuna reikiin.

Puhelimen näytön suuntaaminen robotin maailmakoordinaatiston x- ja y-akselien suuntaisesti voi aiheuttaa epätarkkuutta kuulan osumapisteisiin. Tätä epätarkkuutta voi olla mahdollista parantaa käyttämällä työkohteen koordinaattijärjestelmää. Tässä työssä esitellyssä testausjärjestelyssä käytettiin robotin maailmakoordinaatistojärjestelmää, jossa x-, y- ja z-akselit määriteltiin robotin kalibrointiasennon mukaan ja testausalusta pyrittiin asettamaan mahdollisimman tarkasti näiden akselien suuntaisesti. Robotille on kuitenkin mahdollista määrittää työkohde, jolloin testausalustalle voitaisiin määrittellä oma työkohdekoordinaatisto näyttämällä robotille kahta x-akselin suuntaista pistettä ja kahta y-akselin suuntaista pistettä (5, luku 10, s. 38). Tämä menetelmä on kuitenkin hankalampi robottia ohjelmoitaessa. Pudotusten tarkkuuden kannalta olennainen kysymys on, saadaanko testausalustan x- ja y-akselien suuntaiset pisteet riittävän tarkasti määritettyä vai piirretäänkö lattiaan maailmakoordinaatiston x- ja y-akselien suuntaiset viivat. Tässä työssä päädyttiin jälkimmäiseen.

Jos robotti joudutaan jostain syystä kalibroimaan uudelleen, on mahdollista, että kalibroinnin epätarkkuus aiheuttaa systemaattisen virheen pudotuspisteiden määrittelyssä. Tästä johtuen puhelimen näytön nurkkapisteen sijainti kannattaa

tarkastaa aina uuden kalibroinnin jälkeen. Robottia sammutettaessa on aina syytä huomioida oikea sammutustapa. Robotti tulee sammuttaa ensin ohjelmointiyksiköstä löytyvältä *service*-välilehdeltä *shut down* -komennolla, minkä jälkeen ohjelmointiyksikkö ilmoittaa, kun pääkytkimestä voidaan katkaista virta. Jos virta katkaistaan pääkytkimestä suoraan ilman *shut down* -komentoa, robotti voidaan joutua kalibroimaan uudelleen.

6.2 Turvallisuus

Robotilla toteutettua testausta aloitettaessa on aina syytä huomioida, missä asennossa robotin manipulaattori on ja onko robotin työkalulla esteetön pääsy sovelluksessa määriteltyyn kotiasemaan. Tämä on olennaista, jotta robotin manipulaattorille tai siinä kiinni olevalle työkalulle ei aiheutuisi törmäysvaaraa. Käytössä oleva robotti on varustettu törmäyksen havaitsevalla toiminnolla, joka pysäyttää törmäyksen sattuessa robotin liikkeen.

Robotin manipulaattorin ympärille on asennettu läpinäkyvä muovinen seinä, jossa on ovi. Robotti havaitsee, jos ovi on auki. Robottia ei voida ajaa automaattiajolla oven ollessa auki. Manuaaliajolla robottia voidaan ohjata ja sovellusta suorittaa oven ollessa auki, mutta tällöin käyttäjän on pidettävä ohjelmointiyksikön liikkeen sallintalaite painettuna, mikä parantaa myös turvallisuutta manuaaliajossa.

Tässä insinööriyössä suunnitellussa testausjärjestelyssä käyttäjän täytyy tavallisesti jokaisen kuulan pudotuksen jälkeen tarkastaa laitteen mahdollinen vikaantumisen. Alkuperäinen ajatus oli, että paineilmasylinterin painettua näytön valon päälle voidaan viat tarkastaa. Testausalusta jouduttiin asentamaan kuitenkin niin kauas käyttäjästä, että robotin ollessa automaattiajolla ei vikoja voida havaita. Näin ollen testaus suoritetaan manuaaliajolla ohjelmointiyksikkö kädessä, ja operaattori tarkastaa viat testauslaitteiston vieressä. Testaus voidaan kuitenkin suorittaa myös automaattiajolla, mutta tällöin operaattori joutuu laitetta testatessaan menemään jokaisen pudotuksen ajaksi pois robotin luota ja laittamaan oven kiinni. Ratkaisuna tähän voisi olla tarkka videokuva puhelimen näytöstä, josta viat voitaisiin havaita riittävällä tarkkuudella. Tällöin pudotukset voitaisiin suorittaa automaattiajolla ja testaus nopeutuisi.

6.3 Muutokset otettaessa uusi puhelinmalli testaukseen

Suunnitellulla ja toteutetulla testauslaitteistolla on tulevaisuudessa tarkoitus testata myös muita puhelinmalleja. Puhelinmallin vaihto ei kuitenkaan onnistu ilman muutoksia testauslaitteiston mekaniikkaan ja robotin sovellukseen.

Otettaessa uutta puhelinmallia testattavaksi suurimmat mekaaniset muutokset tehdään jigiin. Tässä työssä suunnitellulle jigille voidaan lisätä uuden puhelinmallin muodot. Uuden mallin muotoja liitettäessä kannattaa lisätä Catia-ohjelmassa hieman välystä *offset*-käskyllä, jotta jigistä ei tulisi liian ahdas puhelimelle. Myös avoin osa jigin päätyyn puhelimen laturia varten kannattaa muistaa. Puhelinmallin ja käyttäjän tarpeiden mukaan voidaan näytön taustavalon päälle painavan paineilmasylinterin paikka ja sen aiheuttamat muutokset miettiä uudelleen.

Kun tarvittavat muokkaukset testauslaitteiston mekaniikkaan on tehty, voidaan siirtyä robottia ohjaavan sovelluksen muokkaukseen. Jokaista puhelinmallia varten kannattaa tallentaa robotille oma ohjelmansa, joka on muokattu alkupe-
räisestä sovelluksesta. Sovelluksen muokkauksen aluksi kannattaa kaikki sovel-
luksessa käytettävät *robotarget*-pisteet tarkastaa, koska puhelin voi olla uudessa
jigissä hieman eri paikalla kuin aiemmin. Puhelimen näytön nurkkapiste *pnurkka*
tulee määritellä uudelleen. Lisäksi jokaisen osumapisteen x- ja y-suuntaiset
etäisyydet nurkkapisteestä *pnurkka* täytyy asettaa uudelleen, koska puhelimen
näyttöjen koko ei ole aina sama.

7 POHDINTA

Tämän opinnäytetyön päätarkoituksena oli suunnitella ja toteuttaa matkapuhelimen näytön iskunkestävyydesti teollisuusrobotin avulla. Alkuperäisessä ball drop -testauslaitteistossa suurimmat ongelmat olivat teräskuulan osumapisteen tarkkuudessa ja toistettavuudessa. Kolmas ongelma alkuperäisessä testauslaitteistossa oli, että testattava puhelin pääsi liikkumaan teräskuulan iskun voimasta, mikä aiheutti ongelmia iskun voimakkuuden vaihteluina. Opinnäytetyön päätavoitteena oli näiden ongelmakohtien parantaminen. Lisäksi yhtenä tavoitteena opinnäytetyössä oli ball drop -testin automaatioasteen nosto.

Opinnäytetyössä suunnitellun ja toteutetun testausjärjestelyn mekaaniset osat jigi, jalusta, luiska ja anturipidike valmistettiin Nokia Oyj:n omalla pikamallikoneella. Laitevalinnoissa valittiin testauksessa käytettävät laitteet: anturi, tyhjiögeneraattori, tyhjiömuri ja paineilmasylinteri. Valitut laitteet kytkettiin osaksi robotin automaatiota. Työn loppuvaiheessa robotille luotiin koko testaussekvenssiä ohjaava sovellus.

Testausjärjestelyssä otettiin huomioon laajasti käytettävissä olevat mahdollisuudet testauksen automatisoinnin toteuttamisessa. Testauslaitteiston mekaanisten osien suunnittelussa oli keskeistä huomioida kaikki testin toimintaan ja teräskuulan liikkeisiin vaikuttavat ominaisuudet.

Testin tarkkuus ja toistettavuus paranivat huomattavasti, kun testauksessa käytettiin teollisuusrobotia apuvälineenä. Muotoilultaan sopivaksi suunniteltu jigi antoi puhelimelle tasapainoisen tuen teräskuulan iskeytyessä puhelimen näyttöön. Näin teräskuulan iskujen voimakkuuden vaihtelun aiheuttanut puhelimen liikkumiseen liittynyt ongelma saatiin poistettua. Teollisuusrobotilla toteutettu ball drop -testi helpotti myös operaattorin työtä, sillä koko 36 pudotusta sisältänyt testi automatisoitiin siten, että ainoastaan näyttövikojen visuaalinen tarkastus jäi operaattorin tehtäväksi.

Uuden ball drop -testin testausjärjestelyn tarkkuutta on vielä mahdollista kehittää eteenpäin tässä opinnäytetyössä toteutetusta järjestelystä. Mahdollisuuksiin parantaa uuden teollisuusrobotilla toteutetun testin tarkkuutta otettiin kantaa jo

luvussa 6. Uudenlaista toimintamallia luotaessa on myös oleellista ottaa turvallisuuden liittyvät näkökulmat huomioon.

Tässä opinnäytetyössä suunniteltua ja toteutettua testausjärjestelyä on mahdollista hyödyntää myös muissa Nokia Oyj:n toimipisteissä. Opinnäytetyössä suunniteltua testausalustaa ja jiggiä on mahdollista hyödyntää myös muissa testeissä, joissa vaaditaan puhelimen pysymistä paikallaan tai halutaan käyttää teollisuusrobotia apuna.

Opinnäytetyö oli kokonaisuudessa monipuolinen automaation, mekaniikan suunnittelun sekä projektinhallinnan näkökulmasta tarkasteltuna. Omasta ideasta lähtenyt testauksen menetelmäkehitystyö palveli hyvin sekä Nokia Oyj:n tuotekehityksen tarpeita, että oman ammatillisen kehittymisen tarpeita.

LÄHTEET

1. Järvelä, Sami 2011. Specialist, HW testing, Nokia Oyj. Keskustelut 31.8.2011.
2. Product On-line Manual IRB 1400. 1998. ABB Flexible Automation. Saatavissa: <http://rab.ict.pwr.wroc.pl/irb1400/prod14.pdf>. Hakupäivä 5.2.2012.
3. Korhonen, Ari 2012. Re: Kysymys insinööriyöhön liittyen. Sähköpostikeskustelu 24.2.2012.
4. Nokia 603. 2012. Saatavissa: <http://mea.nokia.com/find-products-en/devices/nokia-603>. Hakupäivä 24.2.2012.
5. Käyttöopas BaseWare OS 4.0. ABB Service Oy.
6. Robottikoulutus Ohjelmoinnin peruskurssi. ABB Robotics.
7. RAPID Reference manual. 1998. ABB Flexible Automation AB. Saatavissa: <http://rab.ict.pwr.wroc.pl/irb1400/overviewrev1.pdf>. Hakupäivä 5.2.2012.
8. Lahtinen, Raimo 1998. Damatic XDi järjestelmän väylien rakenne. Saatavissa: <http://robo.cop.fi/kurssit/lahtinen/VAYLA.HTM>. Hakupäivä 23.4.2012.
9. Ruohomaa, Timo – Toppila, Rauno 1998. Yhteiskäytäntö RS-422 ja RS-485. Saatavissa: <http://www.pori.tut.fi/~tiru/rs.htm>. Hakupäivä 23.4.2012.
10. Alanen, Jarmo 2000. CAN – ajoneuvojen ja koneiden sisäinen paikallisväylä. Saatavissa: http://www.oamk.fi/~eero/ko/Opetus/Ohjausjarjestelmat/CAN/CAN-perusteet_AlaseenMateriaalia.pdf. Hakupäivä 23.4.2012.
11. IRB 1400 Industrial Robot. 2004. ABB. Saatavissa: http://www.antenen.com/htdocs/downloads/files/files_dl/IRB%201400.pdf. Hakupäivä 25.4.2012.
12. Harju-Konno, Pasi 2012. Senior Engineer, HW Validation, Nokia Oyj. Keskustelut 19.3.2012.

13. Honkanen, H 2003. Anturit. Saatavissa:
http://gallia.kajak.fi/opmateriaalit/yleinen/honHar/ma/ELE_A%20N%20T%20U%20R%20I%20T.pdf. Hakupäivä 30.4.2012.
14. Cylindrical Proximity Sensor. 2003. Omron. Saatavissa:
<https://www1.elfa.se/data1/wwwroot/assets/datasheets/03757408.pdf>. Hakupäivä 27.2.2012.
15. Leinonen, Ilkka 2012. Senior Engineer, Mechanics Prototyping, Nokia Oyj.
Puhelinkeskustelu 18.1.2012.
16. Vacuum Ejector Series ZH. 2006. SMC. Saatavissa:
https://www1.elfa.se/data1/wwwroot/assets/datasheets/qzZH_dat_e.pdf.
Hakupäivä 1.5.2012.
17. Vacuum Pad Series ZP. 2002. SMC. Saatavissa:
https://www1.elfa.se/data1/wwwroot/assets/datasheets/sdZP_dat_e.pdf. Hakupäivä 1.5.2012.
18. Product Manual IRB 1400. ABB Flexible Automation.

LIITTEET

Liite 1 Induktiivinen anturi, Omron

Liite 2 Tyhjiögeneraattori, SMC

Liite 3 Tyhjiömuri, SMC

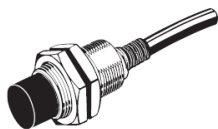
Liite 4 Paineilmasyylinteri, SMC

Liite 5 Robottia ohjaava sovellus

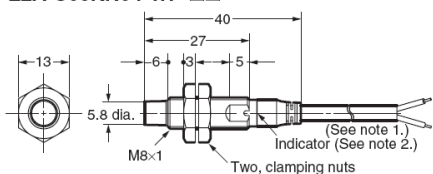
E2A-S08KN04-WP-B1 2M

E2A-S08□N04/M08□N04

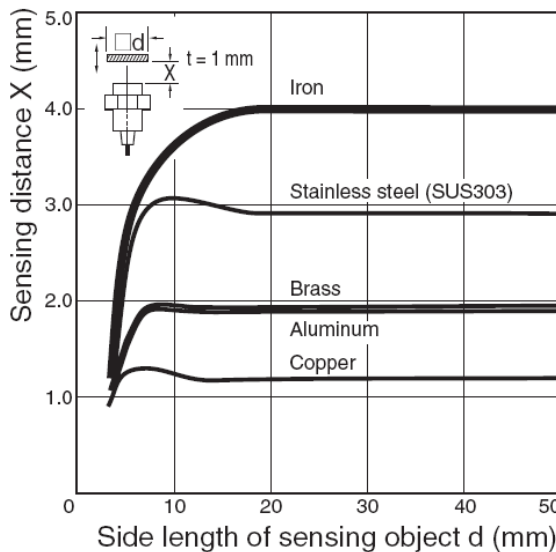
Pre-wired Models (Non-shielded)



E2A-S08KN04-WP-□□



Note 1. 4-dia. vinyl-insulated round cable with 3 conductors (conductor cross section: 0.3 mm²; insulator diameter: 1.3 mm); standard length: 2 m
 2. Operation indicator (yellow)



Operation

PNP Output

Operation mode	Model	Timing chart	Output circuit
NO	E2A-□□-□-B1		<p>Note 1: With M8 connector models, there is no output reverse polarity protection diode.</p> <p>M12 Connector Pin Arrangement (See note 2.)</p> <p>M8 Connector Pin Arrangement</p> <p>Note 2: Terminal 2 of the M12 connector is not used.</p>

Tekniset tiedot	
Toimintalämpötila maks.	+70 °C
Pituus	40 mm
Kytettäisyys maks.	4 mm
Lähtö	PNP, sulkeva
Seinaläpiviivetti	ei upotettu
Versio	3-johtiminen DC
Malli	M8
Kotelo materiaali	Jaloteräs
Liitäntä	PVC-kaapeli 2 m
Suojaustuokka	IP 67/69K
Käyttöjännite	10...32 VDC
Kuormitusvirta	max.200 mA
Kytettäajuus maks.	1000 Hz
Käyttöjännite min.	10 VDC
Käyttöjännite maks.	32 VDC
Toimintalämpötila	-40...+70 °C
Toimintalämpötila min.	-40 °C

Tiedot	
Valmistaja	Omron
Valmistajan tyyppi	E2A-S08KN04-WP-B1 2M
Alkuperämaa	CN
Alkuperäispakkauksen koko	0
Mitat (pakkaus)	120X120X18 mm
Paino	60.1 g
Luettelosivu / Luettelonumero	1571/59
Tullinimike	8536508099
UIHSPSC (v5.03)	30211910
RoHS-info	RoHS-direktiivin mukainen

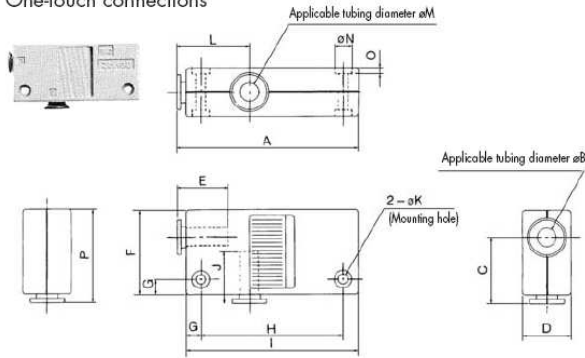
EZH10BS-06-06



Vacuum Ejector Series ZH

Dimensions Box Type (Built-in silencer): ZH□B^S□-□□

One-touch connections



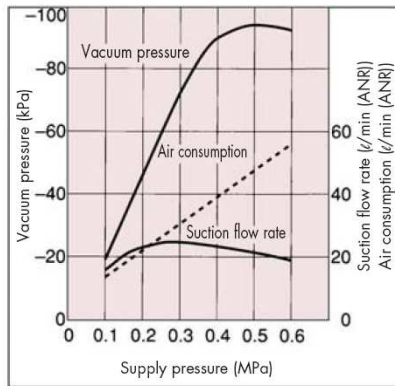
Model	A	øB	C	D	E	F	G	H
ZH05BS-06-06	60	6	22	16	12.8	28	5	47
ZH05BL-06-06	60	6	22	16	12.8	28	5	47
ZH07BS-06-06	60	6	22	16	12.8	28	5	47
ZH07BL-06-06	60	6	22	16	12.8	28	5	47
ZH10BS-06-06	63	6	23	18	12.8	29	5	50
ZH10BL-06-06	63	6	23	18	12.8	29	5	50
ZH13BS-08-10	78	8	27.5	23	13.7	35	7	61
ZH13BL-08-10	78	8	27.5	23	13.7	35	7	61

Model	I	J	øK	L	øM	øN	O	P
ZH05BS-06-06	57	12.8	3.2	24	6	5.8	2	31
ZH05BL-06-06	57	12.8	3.2	24	6	5.8	2	31
ZH07BS-06-06	57	12.8	3.2	24	6	5.8	2	31
ZH07BL-06-06	57	12.8	3.2	24	6	5.8	2	31
ZH10BS-06-06	60	12.8	3.2	26	6	5.8	2	32
ZH10BL-06-06	60	12.8	3.2	26	6	5.8	2	32
ZH13BS-08-10	75	15.3	4.2	28	10	7.5	3	38.5
ZH13BL-08-10	75	15.3	4.2	28	10	7.5	3	38.5

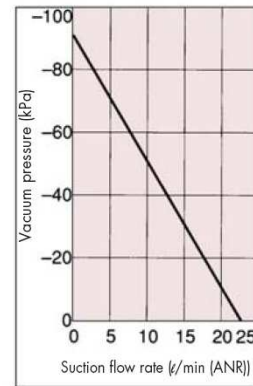
ZH10□□

Max. vacuum pressure: -88 kPa

Exhaust Characteristics



Flow Characteristics



Tekniset tiedot

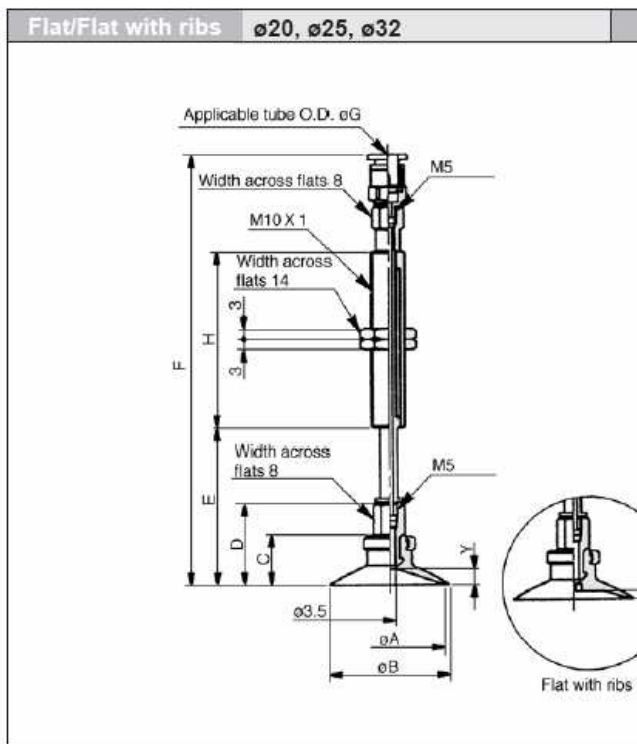
Korkeus	29 mm
Leveys	63 mm
Syvyys	18 mm
Imuteho maks.	24 l/min
Suuttimen ø	1.0 mm
Tyhjö maks.	-88 kPa
Mitat L x K x S	63 x 29 x 18 mm
Letkun ø liittämä VAC	6 mm
Letkun ø liittämä SUP	6 mm
Ilmankulutus	46 l/min

Tiedot

Valmistaja	SMC
Valmistajan tyyppi	EZH10BS-06-06
Alkuperämaa	JP
Alkuperäispakkauksen koko	1
Mitat (pakkaus)	70x35x25 mm
Paino	70 g
Luettelosivu / Luettelonumero	1770/59
Tullinimike	8467920000
UNSPSC (v5.03)	31250000
RoHS-info	RoHS-direktiivin mukainen



ZPT20UN-B01



Flat style/Flat with ribs

Model	A	B	C	D	Y	
					Flat	Flat with ribs
ZPT20 $\frac{1}{2}$ □□□□-0□-A10	20	23	14	23	4	1.7
ZPT25 $\frac{1}{2}$ □□□□-0□-A10	25	28			4	1.8
ZPT32 $\frac{1}{2}$ □□□□-0□-A10	32	35	14.5	23.5	4.5	2.3

Dimensions by stroke

Model	ø20, ø25			ø32			H
	E	F		E	F		
		G: ø4	G: ø6		G: ø4	G: ø6	
ZPT□□ $\frac{1}{2}$ □□□□10-0□-A10	34.5	84.5	85.5	35	85	86	51
ZPT□□ $\frac{1}{2}$ □□□□20-0□-A10	44.5	122.5	123.5	45	123	124	
ZPT□□ $\frac{1}{2}$ □□□□30-0□-A10	54.5	132.5	133.5	55	133	134	
ZPT□□ $\frac{1}{2}$ □□□□40-0□-A10	64.5	168.5	169.5	65	169	170	77
ZPT□□ $\frac{1}{2}$ □□□□50-0□-A10	74.5	178.5	179.5	75	179	180	

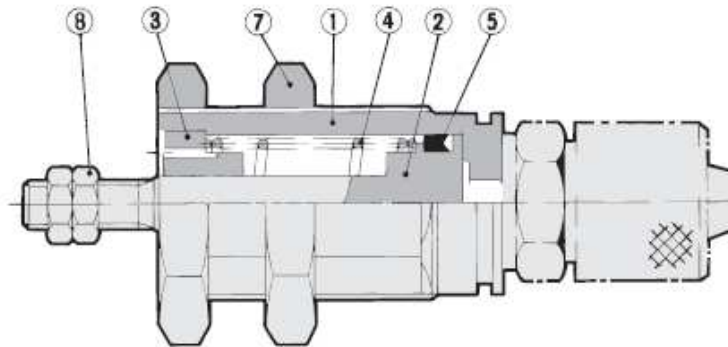
Tekniset tiedot	
Väri	musta
Toimintalämpötila maks.	60 °C
Materiaali	NBR
Tyhjiöllitäntä	Sisäkierte Rc1/8
Imurin Ø	20 mm
Toimintalämpötila	0...60 °C
Toimintalämpötila min.	0 °C
Kovuus	50 Shore A55°

Tiedot	
Valmistaja	SMC 
Valmistajan tyyppi	ZPT20UN-B01
Alkuperämaa	JP
Alkuperäispakkauksen koko	1
Mitat (pakkaus)	35X25X25 mm
Paino	16 g
Luettelosivu / Luettelonumero	1771/59
Tullinimike	8467920000
UHSpsc (v5.03)	31250000
RoHS-info	RoHS-direktiivin mukainen

CJPB6-15H6

Construction (Not able to disassemble.)

Panel mounting style

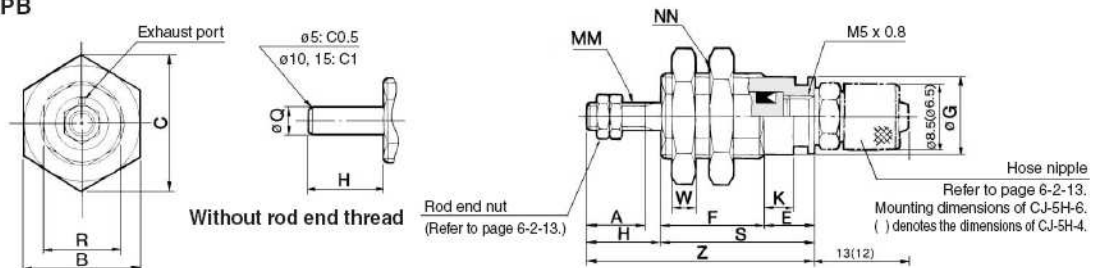


Component Parts

No.	Description	Material	Note
①	Cover	Brass	Electroless nickel plated
②	Piston	Stainless steel	
③	Collar	Oil-impregnated sintered alloy	ø6, ø10 Phosphor bronze
④	Return spring	Piano wire	Zinc chromated
⑤	Piston seal	NBR	
⑥	Gasket	NBR	Special product (O-ring) for embedded style
⑦	Mounting nut	Brass	Electroless nickel plated
⑧	Rod end nut	Steel	Nickel plated

Panel Mounting Style

CJPB



Bore size (mm)	A	B	C	E	F			G	H	K	MM	NN	R	S			W	Z			Q
					5 st	10 st	15 st							5 st	10 st	15 st		5 st	10 st	15 st	
6	7	12	13.9	6	12.5	19.5	26.5	8.5	9	3.5	M3 x 0.5	M10 x 1.0	9	18.5	25.5	32.5	3	27.5	34.5	41.5	3
10	10	19	22	6	14.5	21	28	12	12	3.5	M4 x 0.7	M15 x 1.5	13	20.5	27	34	4	32.5	39	46	5
15	12	27	31	7	16.5	22.5	29	19	14	4.2	M5 x 0.8	M22 x 1.5	20	23.5	29.5	36	5	37.5	43.5	50	6

```
%%%
  VERSION:1
  LANGUAGE:ENGLISH
%%%

MODULE BALLDROP
// Paikoituksen määrittelyt robotin liikkeitä varten.
  CONST robtarget ppyyh2=[[124.42,-867.23,409.97],[0.000226,0.000416,-1,-0.000421],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget ppyyh1=[[13.21,-867.22,409.98],[0.000226,0.000426,-1,-0.000422],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget pkoti=[[298.33,-822.33,425.64],[0.000213,0.000474,-1,-0.000349],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget ppoim=[[298.34,-822.34,346.01],[0.000195,0.000432,-1,-0.000323],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget pnurkka=[[63.11,-844.56,393.27],[0.000139,0.000369,-1,-0.000372],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
// Muuttujat, valinta ohjelmointiyksikön funktionäppäimillä.
  VAR num jatko:=0;
  VAR num aloitus:=0;

// Päärutiinin aloitus.
  PROC main()
// Työkalan jäykistys paineilmalla, analogialähtö 0-10 voltia.
  SetAO aoQstop,7;
// Ohjelmointiyksikön näytön tyhjennys.
  TPErase;
// Ohjelmointiyksikön näytölle kirjoittaminen.
  TPWrite "Siirrytaan kotiasemaan.";
// Käskey robotin liikuttamiseksi.
  MoveL pkoti,v500,fine,tool0;
  TPErase;
// Ohjelmointiyksikön funktionäppäimien käyttö.
  TPreadFK aloitus,"Aloitetaanko testi?",stEmpty,stEmpty,stEmpty,stmpty,"Aloita";
  TPErase;
  TPWrite "Aseta kuula poimintapaikkaan.";
// Ulompi for-silmukka, jossa i:n arvo määrittelee pudotuskorkeuden.
  FOR i FROM 1 TO 4 DO
// Sisempi for-silmukka, jossa j:n arvo määrittelee osumapisteen.
  FOR j FROM 1 TO 9 DO
// Odotetaan induktiivisen anturin tuloa, kuula poimintapisteessä.
  WaitDI diT9,1;
// While-rakenteella poimitaan kuulaa. Jos kuula ei poistunut poimintapisteestä, yritetään aina uudelleen.
  WHILE diT9=1 DO
    MoveL ppoim,v500,fine,tool0;
    SetDO doVent1,1;
    WaitTime 1;
    MoveL pkoti,v500,fine,tool0;
  ENDWHILE
  TPErase;
// Pudotuskorkeuden ja pudotuspisteen valinta i:n ja j:n avulla.
  IF i=1 THEN
    IF j=1 THEN
      TPWrite "Pudotuskorkeus 47cm, Pudotusnro 1";
// Pudotuspaikka määritellään etäisyyksinä näytön nurkkapisteestä.
    MoveL Offs(pnurkka,73,39.4,470),v500,fine,tool0;
  ENDIF
  IF j=2 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 2";
```

```
    MoveL Offs(pnurkka,73,22.2,470),v500,fine,tool0;
ENDIF
IF j=3 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 3";
    MoveL Offs(pnurkka,73,5,470),v500,fine,tool0;
ENDIF
IF j=4 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 4";
    MoveL Offs(pnurkka,39,39.4,470),v500,fine,tool0;
ENDIF
IF j=5 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 5";
    MoveL Offs(pnurkka,39,22.2,470),v500,fine,tool0;
ENDIF
IF j=6 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 6";
    MoveL Offs(pnurkka,39,5,470),v500,fine,tool0;
ENDIF
IF j=7 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 7";
    MoveL Offs(pnurkka,5,39.4,470),v500,fine,tool0;
ENDIF
IF j=8 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 8";
    MoveL Offs(pnurkka,5,22.2,470),v500,fine,tool0;
ENDIF
IF j=9 THEN
    TPWrite "Pudotuskorkeus 47cm, Pudotusnro 9";
    MoveL Offs(pnurkka,5,5,470),v500,fine,tool0;
ENDIF
ENDIF
IF i=2 THEN
    IF j=1 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 1";
        MoveL Offs(pnurkka,73,39.4,630),v500,fine,tool0;
    ENDIF
    IF j=2 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 2";
        MoveL Offs(pnurkka,73,22.2,630),v500,fine,tool0;
    ENDIF
    IF j=3 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 3";
        MoveL Offs(pnurkka,73,5,630),v500,fine,tool0;
    ENDIF
    IF j=4 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 4";
        MoveL Offs(pnurkka,39,39.4,630),v500,fine,tool0;
    ENDIF
    IF j=5 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 5";
        MoveL Offs(pnurkka,39,22.2,630),v500,fine,tool0;
    ENDIF
    IF j=6 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 6";
        MoveL Offs(pnurkka,39,5,630),v500,fine,tool0;
    ENDIF
    IF j=7 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 7";
        MoveL Offs(pnurkka,5,39.4,630),v500,fine,tool0;
    ENDIF
    IF j=8 THEN
```



```
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 8";
        MoveL Offs(pnurkka,5,22.2,630),v500,fine,tool0;
    ENDIF
    IF j=9 THEN
        TPWrite "Pudotuskorkeus 63cm, Pudotusnro 9";
        MoveL Offs(pnurkka,5,5,630),v500,fine,tool0;
    ENDIF
ENDIF
IF i=3 THEN
    IF j=1 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 1";
        MoveL Offs(pnurkka,73,39.4,780),v500,fine,tool0;
    ENDIF
    IF j=2 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 2";
        MoveL Offs(pnurkka,73,22.2,780),v500,fine,tool0;
    ENDIF
    IF j=3 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 3";
        MoveL Offs(pnurkka,73,5,780),v500,fine,tool0;
    ENDIF
    IF j=4 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 4";
        MoveL Offs(pnurkka,39,39.4,780),v500,fine,tool0;
    ENDIF
    IF j=5 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 5";
        MoveL Offs(pnurkka,39,22.2,780),v500,fine,tool0;
    ENDIF
    IF j=6 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 6";
        MoveL Offs(pnurkka,39,5,780),v500,fine,tool0;
    ENDIF
    IF j=7 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 7";
        MoveL Offs(pnurkka,5,39.4,780),v500,fine,tool0;
    ENDIF
    IF j=8 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 8";
        MoveL Offs(pnurkka,5,22.2,780),v500,fine,tool0;
    ENDIF
    IF j=9 THEN
        TPWrite "Pudotuskorkeus 78cm, Pudotusnro 9";
        MoveL Offs(pnurkka,5,5,780),v500,fine,tool0;
    ENDIF
ENDIF
IF i=4 THEN
    IF j=1 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 1";
        MoveL Offs(pnurkka,73,39.4,940),v500,fine,tool0;
    ENDIF
    IF j=2 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 2";
        MoveL Offs(pnurkka,73,22.2,940),v500,fine,tool0;
    ENDIF
    IF j=3 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 3";
        MoveL Offs(pnurkka,73,5,940),v500,fine,tool0;
    ENDIF
    IF j=4 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 4";
```

```
        MoveL Offs(pnurkka,39,39.4,940),v500,fine,tool0;
    ENDIF
    IF j=5 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 5";
        MoveL Offs(pnurkka,39,22.2,940),v500,fine,tool0;
    ENDIF
    IF j=6 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 6";
        MoveL Offs(pnurkka,39,5,940),v500,fine,tool0;
    ENDIF
    IF j=7 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 7";
        MoveL Offs(pnurkka,5,39.4,940),v500,fine,tool0;
    ENDIF
    IF j=8 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 8";
        MoveL Offs(pnurkka,5,22.2,940),v500,fine,tool0;
    ENDIF
    IF j=9 THEN
        TPWrite "Pudotuskorkeus 94cm, Pudotusnro 9";
        MoveL Offs(pnurkka,5,5,940),v500,fine,tool0;
    ENDIF
ENDIF
// Näytön taustavalo päälle paineilmasylinterillä.
SetDO doVent3,1;
WaitTime 1;
// Teräskuulan pudotus.
SetDO doVent1,0;
WaitTime 2;
SetDO doVent3,0;
WaitTime 3;
// Jos kuula jää näytön päälle, pyyhkäistään robotilla näytön yli.
IF diT9=0 THEN
    MoveL ppyyh1,v500,fine,tool0;
    MoveL ppyyh2,v500,fine,tool0;
ENDIF
MoveL pkoti,v500,fine,tool0;
// Sisempi for-silmukka päättyy.
ENDFOR
TPErase;
// Kysytään testin jatkumisesta, vastaus funktionäppäimillä.
IF i=1 THEN
    TPReadFK jatko,"Jatketaanko seuraavaan pudotuskorkeuteen? (63c
m)","Alkuun",stEmpty,stEmpty,stEmpty,"Jatka";
ENDIF
IF i=2 THEN
    TPReadFK jatko,"Jatketaanko seuraavaan pudotuskorkeuteen? (78c
m)","Alkuun",stEmpty,stEmpty,stEmpty,"Jatka";
ENDIF
IF i=3 THEN
    TPReadFK jatko,"Jatketaanko seuraavaan pudotuskorkeuteen? (94c
m)","Alkuun",stEmpty,stEmpty,stEmpty,"Jatka";
ENDIF
IF i=4 THEN
    TPReadFK jatko,"Testi on paattynyt.",stEmpty,stEmpty,stEmpty,s
tEmpty,"OK";
ENDIF
TPErase;
// Testin keskeytys funktionäppäimen valinnalla.
IF jatko=1 THEN
    EXIT;
```

```
ENDIF  
// Ulompi for-silmukka päättyy.  
ENDFOR  
ENDPROC  
ENDMODULE
```