



ROBOTTIJÄRJESTELMÄN DOKUMENTAATIO

Työstökeskuksen palvelurobotti

Jyri Valta
jyri.valta@edu.savonia.fi

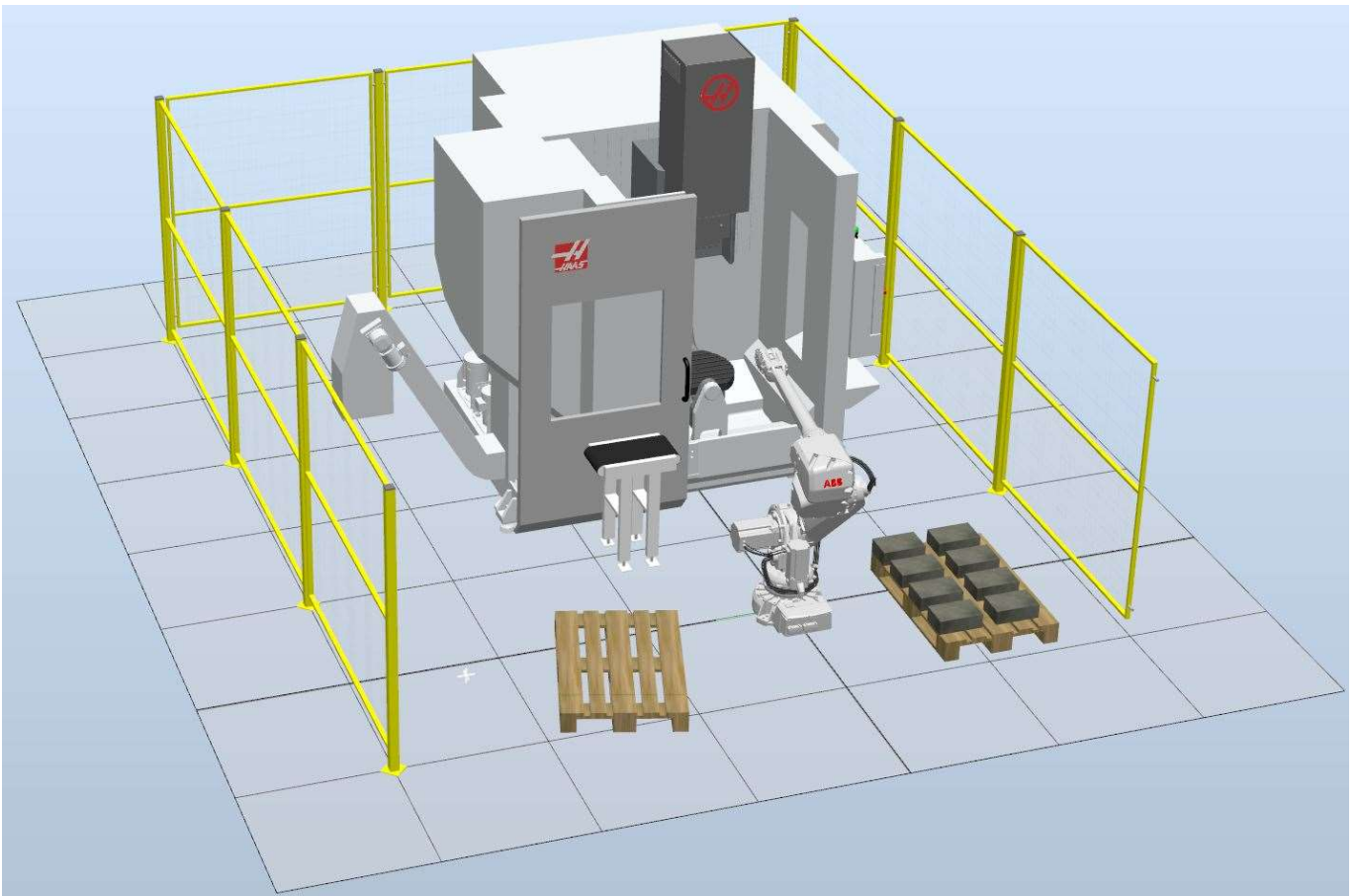
Johdanto

Tämä robottijärjestelmän dokumentaatio on osa Savonia-ammattikorkeakoulun avoimen AMK:n Digitaalisen valmistuksen perusteet -koulutusta, jonka yhtenä osa-alueena oli teollisuusrobottien ohjelmointi tuotantokäyttöön. Dokumentaatiossa kerron ensin lyhyesti järjestelmän toteutuksesta ja sen jälkeen kuvaan ohjelman toimintaa.

Järjestelmän toteutuksen pohjana oli tuotantosolu, joka sisältää ABB:n IRB 4600-teollisuusrobotin kuudella vapausasteella, 20 kg:n kantokyvyllä ja 2,5 metrin ulottuvuudella. Robotin controllerina toimii virtuaalinen IRC5-kontrolleri, joka oli robotin ohessa luotu asemaan valmiiksi.

Valmiiseen pohjaan kuului myös koneistuskeskus, jolle oli luotu tulosignaali oven sulkemiselle mekanismeineen sekä kappaleen valmistumisesta robotille kertova digitaalilähtö, myös koneistuskeskuksen logiikka oli ohjelmoitu valmiiksi. Mittauslaitteella ei ollut valmiita signaaleja, joten ne piti luoda sille itse, samoin robotin ja imukuppitarrainen väliset signaalit sekä robotille lähtö työstökeskuksen oven ohjaamiseen ja tulo kappaleen valmistumisesta kertovan signaalin vastaanottoon.

Soluun kuului myös kaksi kappaletta kuormalavoja, joista toisen päällä oli 4x2-matriisissa 8 kpl 200x300x100 mm palikoita, sekä aidat solun ympärillä.



Toteutus

Ensimmäiseksi mallinsin Robotstudion modeling-työkaluilla tyhjässä asemassa imukuppitarraimen, joka on vain 100 mm korkea, 100 mm halkaisijaltaan oleva sylinteri, jonka yläpintaan tein työkalupisteen. Työkalun tallensin kirjastoksi Save as library-toiminnolla, jotta saan sen tuotua työkalupisteineen projektitiedostoon.

Seuraavaksi mallinsin Solidworksilla yksinkertaisen pöydän, jonka runko on 50x50x4 mm putkipalkkia, kansilevyksi mallinsin 20 mm paksun, 600x600 mm laatan, josta tein sheet metal-työkaluilla 1,5 mm paksuisen kuoren, jonka yläpinta on poistettu. Näin sain tehtyä altaan, jonka liitin Solidworks-kokoonpanona pöydän rungon päälle, kokonaisuuden tallensin geometriatiedostoksi .sat-muotoon.

Pöydän tarkoitus on esittää simulaatiossa osanpesukonetta, se ei osaa tehdä mitään.

Seuraavana vaiheena oli luodun työkalun tuonti projektiin, joka käy helposti Robotstudion import library-toiminnolla. Oletuksena Robotstudio tuo uudet komponentit maailman koordinaatiston origoon, josta sen sai liitettyä robotin työkalulaippaan raahaamalla työkaluobjektin aseman layout-näkymässä robotin objektin päälle, ja vahvistamalla että halutaan päivittää objektin sijainti.

Ennen ”pesukoneen” tuomista projektiin mietin sille sopivan paikan. Loogisinta on sijoittaa pesukone työstökeskuksen oven välittömään läheisyyteen, jotta kappaleet saa vietyä koneistuksesta suoraan pesuun, mikäli ne on määritetty pestäviksi. Siispä siirsin mittauslaitetta pituutensa verran vasemmalle, eli maailman koordinaatiston x-akselista pois päin y-suunnassa. Näin sain tehtyä pesukoneelle sopivan paikan keskuksen oven viereen, jottei kappaleita tarvitsisi siirrellä edestakaisin työskentelyalueella. Tällä saadaan työkiertoa nopeammaksi koska turhat liikkeet jäävät pois.

Sitten tarvitsi vain tuoda .sat-muotoinen geometriatiedosto projektiin.

Ulkoisilla mallinnusohjelmilla luotujen objektien tuomisessa asemaan voi kappale ilmaantua lähes miten päin tahansa, koska eri ohjelmistojen koordinaattijärjestelmät saattavat olla eri suuntiin. Niin kävi tässäkin tapauksessa, ja alun perin pöytä ilmestyi projektin maailman koordinaatiston origoon kyljellään, puoliksi lattian sisään uponneena. Pöydän sai kuitenkin helposti käännettyä oikeaan asentoon klikkaamalla objektia layout-ikkunassa hiiren oikealla painikkeella, valitsemalla Position -> Rotate, ja kääntämällä sitä x-akselilla 90 astetta. Sitten siirsin pöydän oikealla paikalleen mittauslaitteen viereen.

Seuraavaksi oli vuorossa robotin, työkalun sekä mittalaitteen lähtö- ja tulosaalien luominen, sen jälkeen tein työkalusta smart componentin, jonka liitin työkaluun, jotta sitä voidaan ohjata robotin lähdoilla. Robotin antaessa työkalun di_Kiinnita-tuloon signaalin työkalu tarttuu linesensor-komponentin tunnistamaan objektiin, kun signaali vapautetaan tarrain päästää irti kappaleesta.

Mittalaitteelle tein yksinkertaisen logiikan, jossa robotin antaessa signaalin mittalaitteen di_Mittaa-tuloon mittalaite arpoo satunnaislukugeneraattorilla osaluvun nollan ja yhden väliltä. Mikäli osaluku on lähempänä lukua 1, mittalaitteen lähtö do_Hylsy aktivoituu, ja antaa robotille signaalin, että kappale ei ole hyväksytty, näin robotti osaa viedä kappaleen hylsyyn.

Toteutuksen seuraava vaihe oli workobjectien eli koordinaatistojen luominen aseman eri osille, tein koordinaatistot kappaleiden haku- sekä valmistuotelavoille, hylsyynolle, työstökeskukselle, pesurille sekä mittalaitteelle. Kuormalavojen koordinaatistot ovat suhteessa lavoihin samansuuntaiset, eli alkupiste on työstökeskuksen puoleisen päädyn vasen nurkka. Näin haku- ja vientipisteiden paikan muutos toimii molemmilla lavoilla samalla tavalla. Lavojen paikkapisteissä on se ero, että valmistuotelavan vientipistettä on käännetty 90 astetta suhteessa hakupisteeseen, näin lavan pinta-ala saadaan paremmin hyödynnettyä. Sijoitettuna poikittain suhteessa lavaan kappaleita mahtuu vain 12, mutta pitkittäin kuormattuna 16.

Sitten tarvitsi vain tehdä ohjelmaan paikkapisteet, jotka liitin oikeisiin workobjecteihin ja synkronoida aseman sisältö rapid-koodiin. Loppu oli kiinni ohjelman koodaamisesta rapid-editorissa, koodin toiminta on täysin kommentoitu ohjelmaan, joten en ala käymään sitä erikseen läpi tässä dokumentaatiossa.

Pyrin ohjelmoinnissa noudattamaan yleisiä ohjelmoinnin periaatteita, eli käytin arvojen tallentamiseen muuttujia ja vakioita, joihin viitataan funktioiden sisällä, näin jonkin arvon muuttuessa koko ohjelmaa ei tarvitse kirjoittaa uusiksi koska ohjelma sijoittaa viittauksen paikalle kyseisen arvon.

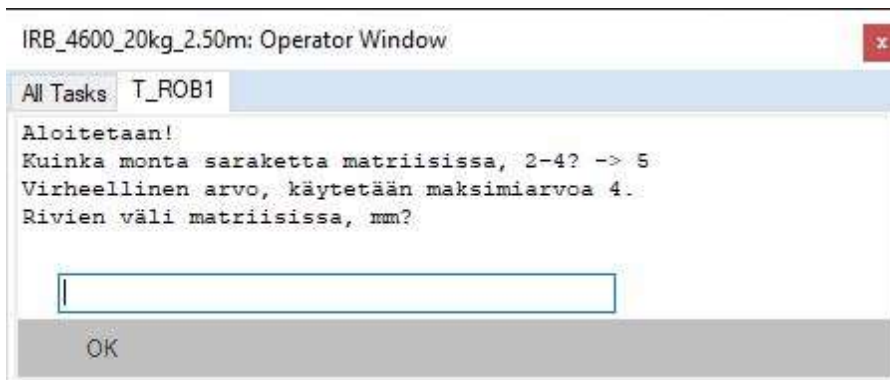
Tein myös suurimmasta osasta ohjelman toiminnoista erillisiä aliohjelmia, joita kutsutaan tarpeen mukaan, näin varsinainen pääohjelma pysyy siistinä ja ohjelmankulku on helposti muokattavissa. Jotkut aliohjelmat voivat myös tarpeen mukaan käyttää muita aliohjelmia osana omaa suoritustaan, näin samaa koodia ei tarvitse kirjoittaa montaa kertaa uudelleen.

Tein ohjelman käyttöparametrien syöttöön yksikertaisen virheenkorjauksen, jotta valmistuotelavan matriisista ei voi tehdä liian isoa tai sijoittaa kappaleita ”sisäkkäin” antamalla negatiivisia arvoja kappaleiden välille, pesun ja mittauksen parametrit korjataan myös automaattisesti, mikäli käyttäjä syöttää negatiivisia tai pyydettyä skaalaa suurempia lukuja.

Järjestelmän toiminta on kerrottu seuraavalta sivulta alkaen.

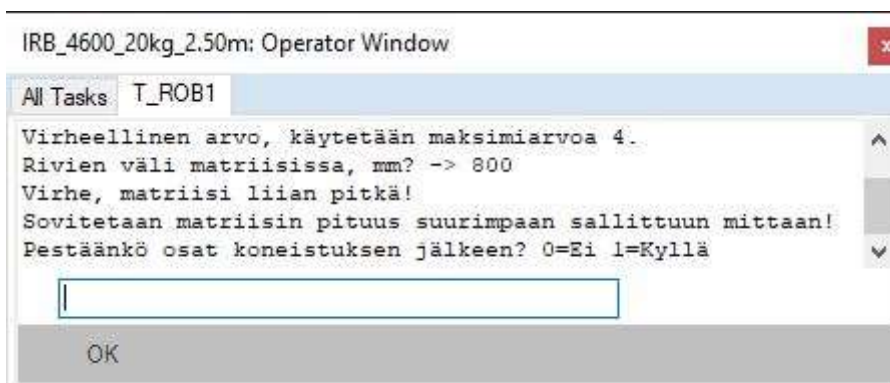
Järjestelmän toiminta

Simulaation alkutilassa robotti on ajettuna kotipisteeseen, valmistuotelava on tyhjä ja hakulavalla on 8 kappaletta. Ennen ohjelman aloittamista tulee kontrollerin Operator window-näkymä avata, koska parametrien syöttö ja palautteen anto käyttäjälle tapahtuu tekstipohjaisen käyttöliittymän kautta. Ensimmäiseksi ohjelma tulostaa näytölle tervehdystekstin, jonka jälkeen käyttäjältä pyydetään ohjelman parametrit yksi asetus kerrallaan. Ensimmäinen asetus on valmistuotelavan matriisin sarakkeiden määrä, tämä asetus määrää kuinka monta kappaletta sijoitetaan vierekkäin kuormalavan lyhyelle sivulle. Mikäli käyttäjän antama arvo ei ole pyydetyllä alueella tulostetaan näytölle virheilmoitus, jossa kerrotaan käytettävän virheellistä arvoa lähinnä olevaa raja-arvoa.



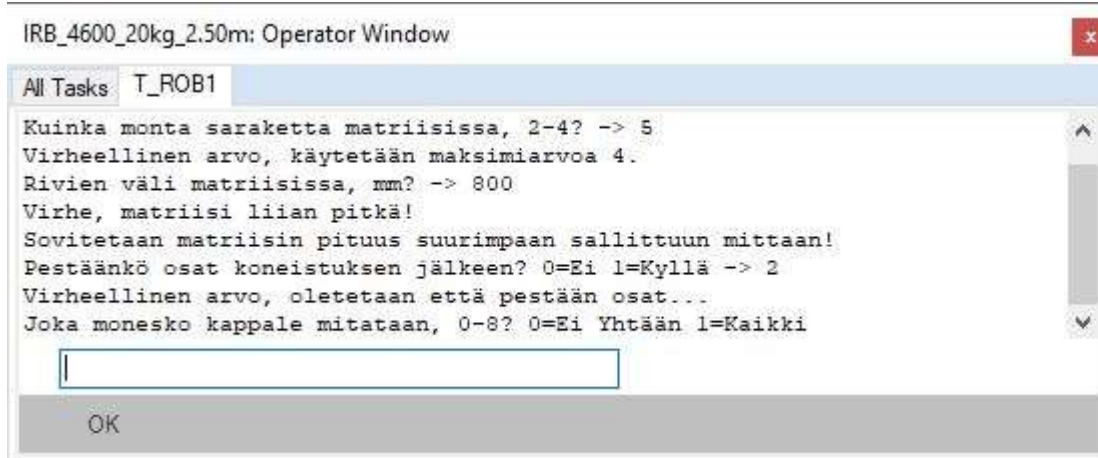
Ohjelma myös asettaa automaattisesti sarakkeiden väliksi 0 mm, mikäli sarakkeiden määrä on 4, näin käyttäjältä ei tarvitse turhaan kysyä sarakkeiden väliä, vaan ohjelma siirtyy suoraan kysymään matriisin rivien väliä millimetreinä. Jos annetaan sarakkeiden määräksi 2 tai 3, ohjelma pyytää käyttäjää syöttämään sarakkeiden välin millimetreinä. Mikäli käyttäjä antaa negatiivisen arvon muutetaan sarakkeiden väliksi 0 mm, jolloin kappaleet ovat kiinni toisissaan. Jos arvo on liian suuri, muutetaan se siten että matriisista tulee kuormalavan levyinen.

Seuraavaksi kysytään matriisin rivien väliä millimetreinä.



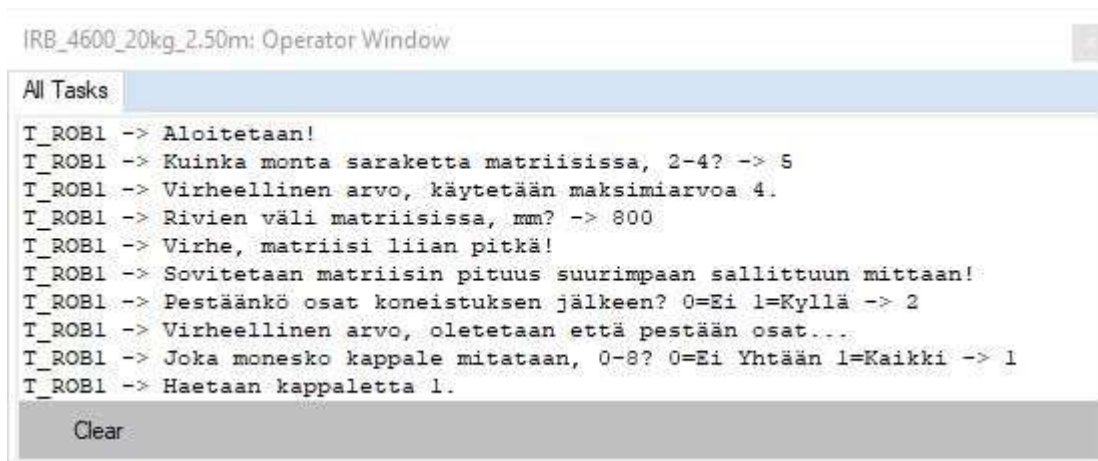
Mikäli rivien väliksi syötetään negatiivinen arvo, tilalle muutetaan taas 0 mm, liian suurella arvolla muutetaan arvoksi suurin sallittu mitta, jolla koko matriisi mahtuu lavalle.

Sitten ohjelma kysyy, halutaanko kappaleet pestä koneistuksen jälkeen, sallitut vastaukset 0 tai 1. Virheelliset arvot korjataan lähimpään sallittuun arvoon.

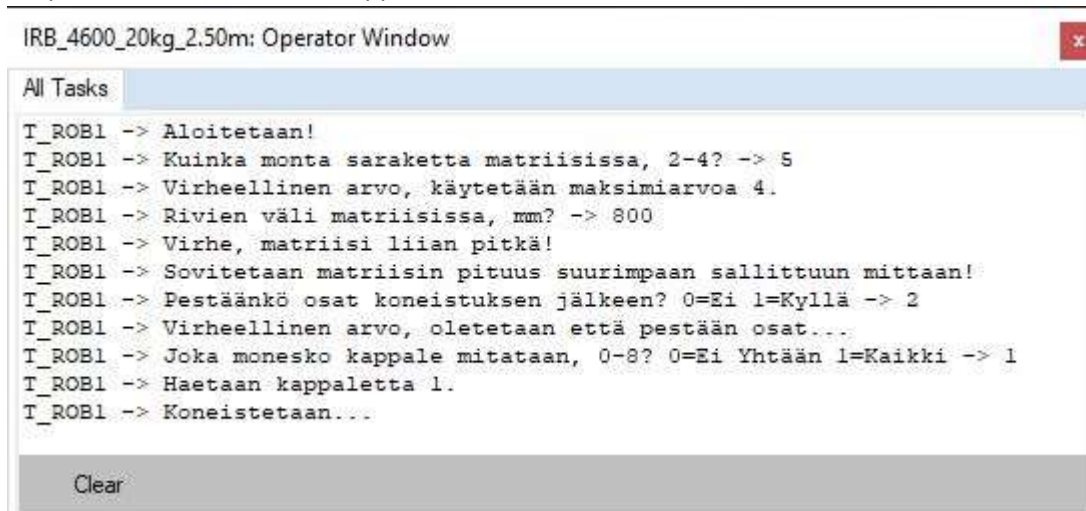


Viimeinen asetus on mitattavien kappaleiden väli, ts. joka monesko kappale mitataan. Negatiivisella syötteellä arvoksi asetetaan 0, liian suurilla arvoilla muuttuuaan tallennetaan ennalta määritetyn vakion nKappaleet arvo, jolloin mitataan vain sarjan viimeinen kappale.

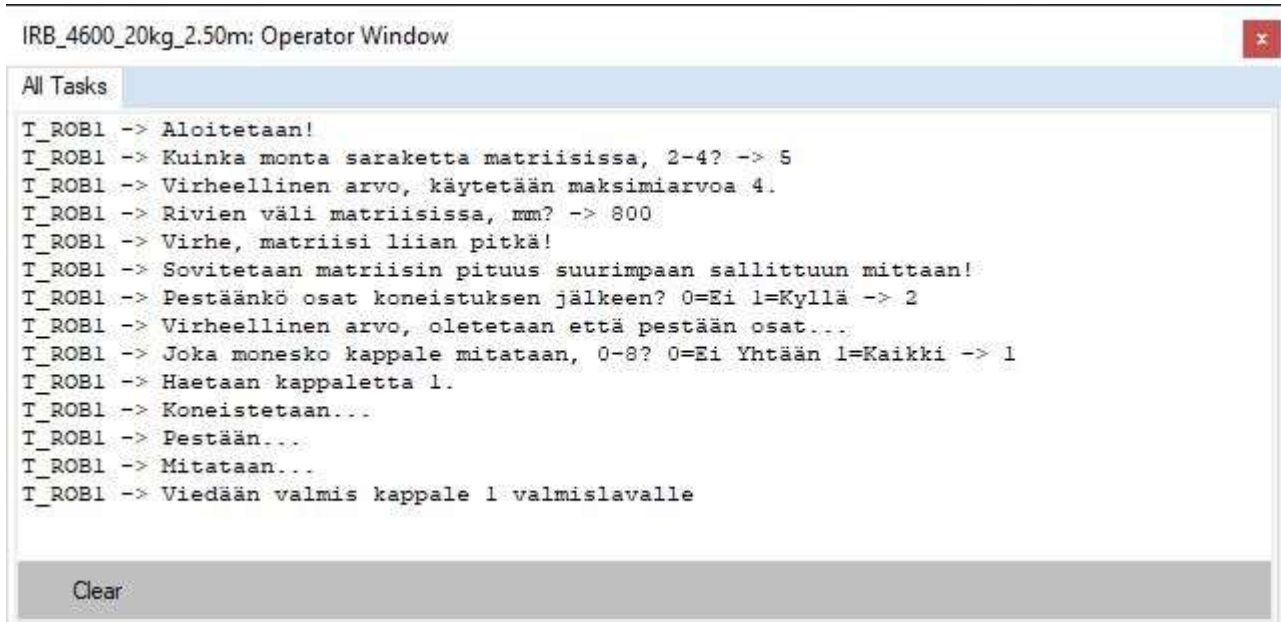
Kun viimeinen arvo on syötetty, robotti lähtee hakemaan ensimmäistä kappaletta, kulloinkin meneillään olevasta työvaiheesta kerrotaan näytöllä.



Vietyään kappaleen työstökeskukseen robotti antaa keskukselle signaalin sulkea oven ja jää oven ulkopuolelle välipisteeseen odottamaan kappaleen valmistumista.



Työstökeskuksen annettua signaalin kappaleen valmistumisesta robotti vapauttaa oven sulkemiskäskyn ja odottaa yhden sekunnin, jotta työstökeskuksen ovi ehtii aueta kokonaan, jonka jälkeen käydään poimimassa kappale ja viedään se seuraavaan vaiheeseen. Jos pesun sekä mittauksen arvoiksi on määritetty 0, kappaleet viedään suoraan koneistuksesta valmistuotelavalle.

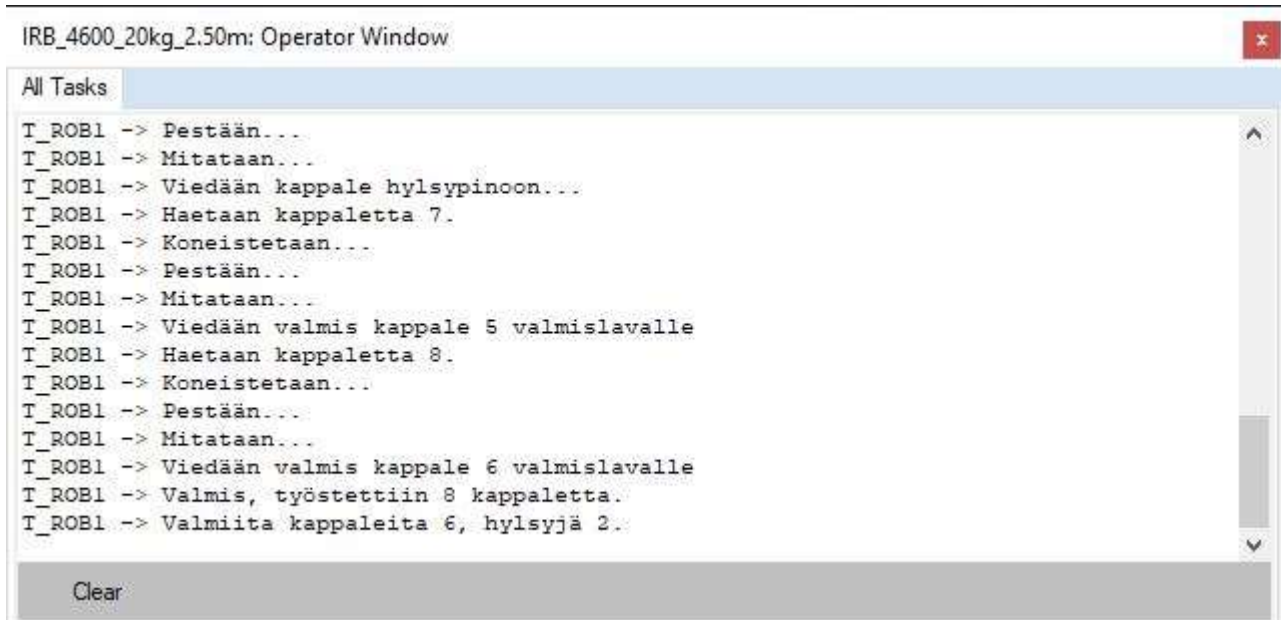


```

IRB_4600_20kg_2.50m: Operator Window
All Tasks
T_ROB1 -> Aloitetaan!
T_ROB1 -> Kuinka monta saraketta matriisissa, 2-4? -> 5
T_ROB1 -> Virheellinen arvo, käytetään maksimiarvoa 4.
T_ROB1 -> Rivien väli matriisissa, mm? -> 800
T_ROB1 -> Virhe, matriisi liian pitkä!
T_ROB1 -> Sovitetaan matriisin pituus suurimpaan sallittuun mittaan!
T_ROB1 -> Pestäänkö osat koneistuksen jälkeen? 0=Ei 1=Kyllä -> 2
T_ROB1 -> Virheellinen arvo, oletetaan että pestään osat...
T_ROB1 -> Joka monesko kappale mitataan, 0-8? 0=Ei Yhtään 1=Kaikki -> 1
T_ROB1 -> Haetaan kappaletta 1.
T_ROB1 -> Koneistetaan...
T_ROB1 -> Pestään...
T_ROB1 -> Mitataan...
T_ROB1 -> Viedään valmis kappale 1 valmislavalle
Clear

```

Molemmilla kuormalavoilla on välipisteet, joiden kautta robotti kulkee, pisteet ovat riittävän korkealla, jotta robotti ei törmää mihinkään. Uutta kappaletta hakiessaan robotti liikkuu työstökeskuksen välipisteen kautta, jottei se törmäisi työstökeskukseen. Myös kappaletta mittauksesta hylsyypinoon viedessään robotti käyttää keskuksen välipistettä, jottei se törmäisi pesukoneeseen. Työstökeskuksen edessä poikittain liikuttaessa käytetään lineaariliikkeitä jottei törmättäisi työstökeskukseen.



```

IRB_4600_20kg_2.50m: Operator Window
All Tasks
T_ROB1 -> Pestään...
T_ROB1 -> Mitataan...
T_ROB1 -> Viedään kappale hylsyypinoon...
T_ROB1 -> Haetaan kappaletta 7.
T_ROB1 -> Koneistetaan...
T_ROB1 -> Pestään...
T_ROB1 -> Mitataan...
T_ROB1 -> Viedään valmis kappale 5 valmislavalle
T_ROB1 -> Haetaan kappaletta 8.
T_ROB1 -> Koneistetaan...
T_ROB1 -> Pestään...
T_ROB1 -> Mitataan...
T_ROB1 -> Viedään valmis kappale 6 valmislavalle
T_ROB1 -> Valmis, työstettiin 8 kappaletta.
T_ROB1 -> Valmiita kappaleita 6, hylsyjä 2.
Clear

```

Mittauksessa hylätyt kappaleet viedään hylsyypinoon ja haetaan uusi kappale, kunnes ohjelmoitu laskuri on täynnä, ohjelman lopuksi robotti ajetaan kotipisteeseen ja kerrotaan käyttäjälle valmiiden kappaleiden sekä hylsyjen määrä.