



# MITSUBISHI RV-2AJ -ROBOTTIKÄDEN OTTAMI- NEN OPETUSKÄYTTÖÖN

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Kone- ja tuotantotekniikan koulutusohjelma	
Työn tekijä(t) Henri Liimatainen	
Työn nimi Mitsubishi RV-2AJ -robottikäden ottaminen opetuskäyttöön	
Päiväys 1.4.2016	Sivumäärä/Liitteet 21/8
Ohjaaja(t) Yliopettaja Risto Rönkä	
Toimeksiantaja/Yhteistyökumppani(t) Savonia-ammattikorkeakoulu	
Tiivistelmä <p>Tämän opinnäytetyön aiheena oli Mitsubishi RV-2AJ -robottikäden ottaminen opetuskäyttöön Savonia-ammattikorkeakoulun Opistotien kampuksen automaatiolaboratoriossa. Robottisolua on käytetty opetuskäytössä vain hyvin vähän. Työn tarkoituksena oli suunnitella ja toteuttaa yksinkertainen harjoitustyö, joka hyödyntää robotisolun laitteistoa mahdollisimman laajasti. Tavoitteena oli, että robottisolu otettaisiin tehokkaammin mukaan opetuskäyttöön.</p> <p>Tietoa robotiikasta saatiin alan kirjallisuudesta ja robottisolun laitteiston ominaisuuksista ja toiminnoista valmistajien käyttöohjekirjoista, mutta suurin osa tiedosta perustui käytännön kokemuksiin. Työ aloitettiin tutustumalla robottisolun laitteistoon sekä käytettävissä oleviin ohjelmistoihin, jotka olivat RoboExplorer sekä Intellect 1.5.1. Sen jälkeen suunniteltiin paletointiohjelma, johon yhdistettiin konenäöllä tapahtuva kappaleen tunnistaminen.</p> <p>Ethernet-ongelman vuoksi alkuperäinen tavoite ei täysin toteutunut. Opinnäytetyön tuloksena saatiin paletointiohjelma sekä ohjeet konenäön käyttöönottoa varten. Jatkossa näitä tietoja voidaan kokonaisuudessaan tai osittain hyödyntää opetuskäytössä.</p>	
Avainsanat robotti, opetuskäyttö, käyttöönotto, Mitsubishi	
julkinen	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Mechanical Engineering			
Author(s) Henri Liimatainen			
Title of Thesis Mitsubishi RV-2AJ Robot Arm, the Introduction to Educational Use			
Date	April 1, 2016	Pages/Appendices	21/8
Supervisor(s) Mr Risto Rönkä, Senior Lecturer			
Client Organisation /Partners Savonia University of Applied Sciences			
<p>Abstract</p> <p>The subject of this thesis was the introduction to educational use of Mitsubishi RV-2AJ robot arm. The robot cell was located in the automation laboratory on the Opistotie Campus of Savonia UAS. The robot cell had been used very little for educational purposes. The objective of this final year project was to design and implement a simple exercise, which makes use of the equipment of robot cell as widely as possible. This assignment was contributed to the idea that the robot cell would be more effectively involved in the educational use.</p> <p>Information about robotics were obtained from literature discussing this topic and information about the functions and features of the robot cell were obtained from manufacturers' operating manuals. However, most of the information used in the work is based on practical experience. The work was started by studying the equipment of the robot cell and learning how to use the available software, which were RoboExplorer and Intellect 1.5.1. After that, a palletizing program was created, which is combined with machine vision and product identification.</p> <p>Due to a problem with Ethernet connection, the original goal was not completely achieved. As a result of this project there was a simple palletizing program, as well as instructions for implementing machine vision. These results can then be used in teaching, using them as a whole or partially.</p>			
Keywords robot, educational use, introduction, Mitsubishi			
public			

## SISÄLTÖ

LYHENTEET JA MÄÄRITELMÄT .....	5
1 JOHDANTO .....	6
2 ROBOTIIKKA.....	7
2.1 Mikä robotti on?.....	7
2.2 Robottien käyttö tuotannossa .....	7
2.3 Robotiikan tulevaisuuden näkymät .....	7
3 ROBOTTISOLUN LAITTEISTO .....	8
3.1 Mitsubishi RV-2AJ .....	8
3.2 CR1-ohjainyksikkö.....	9
3.3 R28TB-opetusyksikkö .....	9
3.4 E410-operointipaneeli .....	10
3.5 E1070-operointiyksikkö .....	10
3.6 DVT-530-harmaasävykamera sekä DVT-542C-värikamera.....	10
3.7 IO-moduuli ja painonappi .....	11
4 HARJOITUSTYÖN SUUNNITTELU .....	13
5 HARJOITUSTYÖN TOTEUTUS .....	14
5.1 Prosessin kulku .....	14
5.2 Paletointiohjelman teko .....	14
5.3 Konenäkö ja kappaleiden tunnistus .....	15
5.4 Ongelmat .....	18
5.4.1 Kuljettimen pysähtyminen.....	18
5.4.2 Ethernet-yhteys kameraan .....	18
5.4.3 Valaistus.....	19
5.4.4 Heijastukset.....	19
5.4.5 Pyöreät kappaleet .....	19
6 PÄÄTÄNTÄ.....	20
LÄHTEET .....	21
LIITE 1: PALETOINTIOHJELMA.....	22
LIITE 2: PALETOINTIOHJELMAN KOMENTOJEN SELITYKSET.....	23
LIITE 3: OHJEET KONENÄKÖÄ VARTEN INTELLECT 1.5.1 -OHJELMALLE .....	24

## LYHENTEET JA MÄÄRITELMÄT

Robottisolu =	Järjestelmä, joka sisältää robotin, ohjausyksiköt sekä oheislaitteet ja turvavarustelut
Paletointi =	Tuotteen lajittelu ja siirto määrättyyn paikkaan tai järjestykseen
Emulointi =	Järjestelmän, ohjelman tai laitteen jäljittely
Input/Output =	Siirräntä, komponenttien välistä tiedonsiirtoa laitteistossa
Hyväksymis-/hylkäysraja =	Prosentuaalinen arvo, jolla kappale määritetään hyväksytyksi tai hylätyksi
Ethernet =	Lähiverkko
String-muuttuja =	Merkkijono, eli muuttuja tekstinä

## 1 JOHDANTO

Tämä opinnäytetyö on tehty Savonia-ammattikorkeakoululle Kuopion Opistotien kampukselle. Aiheena on Mitsubishi RV-2AJ -robottikäden ja sen työsoluun liittyvän laitteiston opetuskäyttöön ottaminen. Lisäksi käsitellään robotin ja kappaleiden tunnistukseen käytettävien kameroiden ohjelmistojä. Robotti sijaitsee Opistotien kampuksen robotiikkalaboratoriossa. Sitä ei ollut ennen tätä opinnäytetyötä käytetty opetuskäytössä Opistotien kampuksella.

Tavoitteena on suunnitella ja toteuttaa robottisolulle paletointiohjelma, jossa hyödynnetään kokenäköä, kuljetinta ja sen logiikkaa sekä varsinaista robottikättä. RV-2AJ-robottikäsi on hankittu Savonia-ammattikorkeakoulun Kuopion yksikköön opetuskäyttöön. Kokonaisuuteen kuuluu Mitsubishin RV-2AJ, CR1-ohjausyksikkö, E1070-operointiyksikkö, kosketusnäytöllinen E710-operointiyksikkö sekä ns. deadman-kytkimellä varustettu opetusyksikkö. Lisäksi laitteistoon kuuluu värikamera 542C, harmaasävykamera 530, IO-moduuli, painonappi ja kuljetin.

## 2 ROBOTIIKKA

### 2.1 Mikä robotti on?

Paineilmalla, sähköisesti tai hydraulisesti liikkuvaa, työkaluja tai kappaleita käsittelevää tietokoneohjattua laitetta kutsutaan yleisesti teollisuusrobotiksi. Teollisuusrobottien liikeradat ovat ohjelmallisesti muutettavissa. Manipulaattoriksi kutsutaan robottia, jonka akselit liikkuvat pelkästään ääriasennosta toiseen. Laite, jossa on vähintään yksi työkalu sekä vähintään kolme liikeakselia, jotka ovat vapaasti ohjelmoitavia, määritetään robotiksi. Myös vihivaunut eli automaattitrukit luetaan roboteiksi. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007.)

Robotteja käytetään nykyään paljon metalli-, elintarvike- ja muoviteollisuudessa. Lisäksi paperi- ja erityisesti lääketieteellisuudessa robotit ovat yleistyneet. Pääkäyttökohteita ovat teollinen hitsaus, kokoonpano, kappaleenkäsittely koneistuksessa ja ruiskupuristuksissa. Ihmiselle raskaissa ja vaarallisissa tehtävissä robottien käyttö on tavallista. Robottien käytön perustana on liikkeiden uudelleen ohjelmoitavuus. Tuotannon muuttuessa myös robottien ohjelmien on voitava muuttua, jotta robotti pääsee käsittelemään uusia tuotteita mahdollisimman nopeasti. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007.)

### 2.2 Robottien käyttö tuotannossa

Robottien avulla saavutetaan miehittämätön tai automaattinen tuotanto tehokkaammin. Tämä tarkoittaa parempaa käyttöastetta, koska käyttökatkokset on minimoitu tai kokonaan poistettu. Ihminen voidaan poistaa raskaista tai yksitoikkaisista tehtävistä laadun pysyessä tasaisena. Ihmisen kannalta epäergonomiset tai työturvallisuuden kannalta vaaralliset tehtävät ovat myös hyviä perusteita robottien käytön yleistymiseen. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007.)

### 2.3 Robotiikan tulevaisuuden näkymät

Robotteja käytetään jo nyt laajasti terveys- ja lääkealalla, teollisuudessa sekä palveluissa. Robottien käyttö teollisuudessa vähentää työpaikkoja, koska esimerkiksi yksinkertaiset liukuhihnatyöt voidaan automatisoida tehtäväksi robottien avulla. Mutta toisaalta robotiikan ja robottien lisääntyminen synnyttää myös uusia työpaikkoja. Täysin korvattavissa ihminen ei ole: Tietyissä töissä robotin laatu ei vastaa ihmisen tekemän työn laatua. Lisäksi robottien alkuvaiheen investointi on suuri. Nykyään robotisoitujen tehtaiden täytyy olla automatisaatioasteeltaan korkealla, jotta ne olisivat kannattavia.

### 3 ROBOTISOLUN LAITTEISTO

#### 3.1 Mitsubishi RV-2AJ

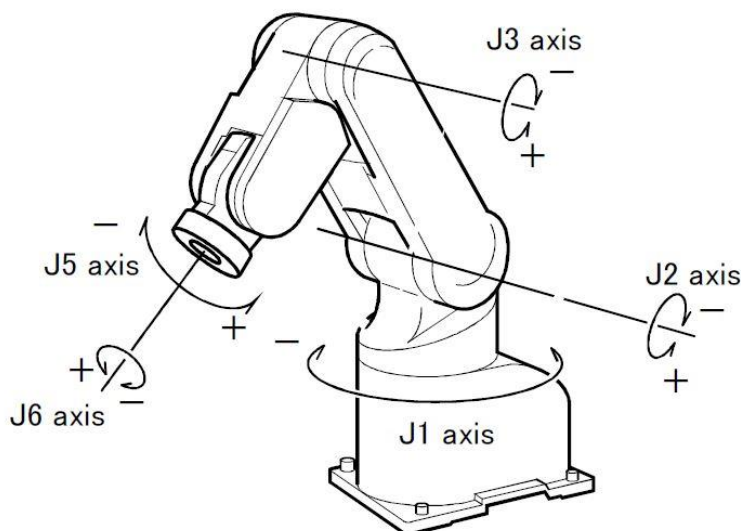
RV-2AJ-robottikäsi on Mitsubishiin kehittyneellä tekniikalla varustettu kompakti teollisuusrobotti. Tämä robotti vastaa käyttäjän tarpeeseen joustavuutta ja tarkkuutta vaativassa tuotannossa. Kohdesovelluksen mukaan työpiste voidaan varustaa robottikäden lisäksi erilaisilla lisälaitteilla. Näistä esimerkkeinä kuljetin, konenäkö sekä kääntyvä pöytä tai jalusta (kuva 1).



KUVA 1. Mitsubishi RV-2AJ -robottikäsi sekä robottisolu (Liimatainen 2016.)

RV-2AJ on vertikaalirakenteinen, joko lattialle tai roikkuvaksi asennettava viisiakselinen (J1, J2, J3, J5, J6) robottikäsi maksimissaan 2 kg:n kuormalle (kuva 2). Robotin ulottuvuus on maksimissaan 410 mm. Robotin alusta (J1) kääntyy 300 astetta, alin nivel (J2) 180 astetta, keskimäinen nivel (J3) 230 astetta, ulommainen nivel (J5) 180 astetta ja työkalua pyörittävä akseli (J6) 400 astetta asentotarkkuuden ollessa  $\pm 0,02$  mm. Verrattuna kuusiakseliseen robottiin tässä mallissa ei ole niveltä J4. Tässä robottisolussa nivel J4 voidaan korvata servomootorin avulla pyörivällä alumiiniekolla, joka tällä hetkellä sijaitsee kiinteän pöydän alapuolella. Pyörivä pöytä ei ole käytössä eikä sitä tässä työssä tarvita. (Mitsubishi 2006.)





5-axis type

KUVA 2. Robotin nivelet (Mitsubishi 2002.)

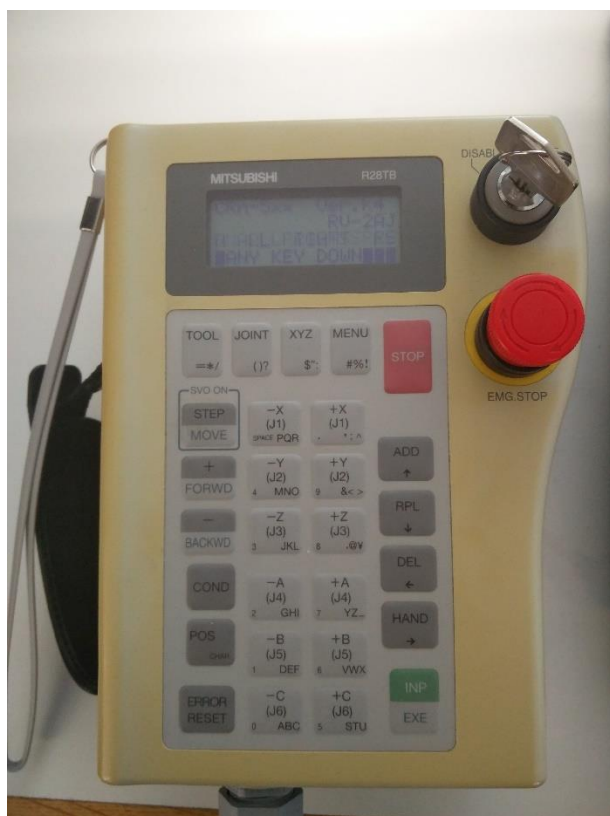
Robotin työkaluina ovat magneettiventtiiliohjatut, pneumaattiset tarttujat. Toinen työkaluista on kaksileukainen pihtitarttuja, jota ei tässä työssä käytetty. Sen sijaan käytin imukuppitarttujaa, joka toimii alipaineen avulla. Imukuppitarttuja on tähän työhön sopiva, koska kappaleet ovat kevyitä ja sileäpintaisia. Pihtitarttujassa kappaleen koko ja muoto rajoittavat työkalun käyttöä merkittävästi, mutta imukuppitarttujalla tämä ongelma voidaan sivuuttaa suoraan. Lisäksi kappaleen nostaminen on helpompaa, koska leukoja ei tarvitse ohjata tarkasti kappaleen ympärille ennen nostamista.

### 3.2 CR1-ohjainyksikkö

64-bittisellä prosessorilla varustettu CR1-ohjainyksikkö on robotin ohjausyksiköiden keskus. Ohjainyksikön ohjelmointikielenä on MELFA BASIC IV, ja se sisältää 88 kappaletta ohjelmamuistipaikkoja. Tämä ohjainyksikkö on varustettu Ethernet-väylällä, joten yksikköä voi hallita myös tietokoneen avulla. Ohjainyksiköllä voidaan valita, käynnistää, pysäyttää sekä muuttaa ohjelman suoritusnopeutta. Yksikössä on 16 sisäistä lähtöä ja tuloa. Lisäksi ohjainyksikkö on varustettu mm. hätäpysäytystoiminnolla, ovikytkimellä sekä törmäysanturilla. (Mitsubishi 2009.)

### 3.3 R28TB-opetusyksikkö

Laitesolun opetusyksikkö on Mitsubishin R28TB (kuva 3). Opetusyksikköä käytetään ohjelmakiertojen testauksessa sekä paikkapisteiden opetuksessa robotille. Opetusyksikkö on aktivoituna sen avainkytkimen ollessa Enable-asennossa sekä ohjausyksikön avainkytkimen ollessa Teach-asennossa. Robottia liikuteltaessa yksikön avulla on painettava ns. deadman-kytkintä. Kytkin on moniasentoinen ja se painetaan robottia liikuteltaessa puoleenväliin. Pohjaan painettuna robotti ei lähde liikkeelle. Opetusyksikön käyttö on turvallisempaa kuin suoraan ohjelmoimalla opetettujen pisteiden ja ohjelmien ajo, koska liikeradat käydään läpi opetusvaiheessa. (Mitsubishi 2009.)



KUVA 3. R28TB-opetusyksikkö (Liimatainen 2016.)

### 3.4 E410-operointipaneeli

E410 on kosketusnäyttöinen operointipaneeli, johon on mahdollista suunnitella eri käyttöliittymiä. Tässä laitesolussa liukuhihnan toimintaa ohjataan E410-operointipaneelilla. Ohjelmointi tapahtuu E-Designer-ohjelmistolla. Lisäksi paneelia käyttämällä voidaan ohjata robottisolussa olevia valoja.

### 3.5 E1070-operointiyksikkö

CR1-yksikköön kytketty E1070 on ohjelmoitavilla painikkeilla varustettu 6,5 tuuman TFT-väri näyttöinen operointipääte. Tällä yksiköllä voidaan valita ja ohjata valmiina olevia ohjelmia. Lisäksi E1070-yksiköllä voidaan ohjata valojen ja magneettiventtiilien käyttöä. Yksikkö on varustettu Ethernet-, sarjaportti- ja USB-liitännällä. (Mitsubishi 2005.)

### 3.6 DVT-530-harmaasävykamera sekä DVT-542C-värikamera

Laitesolun kamerat ovat yhdistettyinä Ethernet-liitännällä tietokoneeseen. Kameroiden resoluutio on 640 x 480 pikseliä. Niitä voidaan hyödyntää kappaleiden tunnistuksessa, mitaamisessa ja laskemisessa. Lisäksi niiden avulla pystytään vertaamaan kuvioita ja lukemaan viivakoodeja ja datamatriiseja. Molemmat kamerat on asennettu liukuhihnan ympärille ja niissä on omat LED-rengasvalot. Tässä työssä käytin vain 542C-väri näkökameraa (kuva 4). (Cognex® Corporation 2006.)



KUVA 4. DVT 542C -värinäkökamera (Liimatainen 2016.)

### 3.7 IO-moduuli ja painonappi

IO-moduuli ja painonappi eivät olleet alkuperäisessä robottisolussa mukana (kuva 5); ne lisättiin jälkikäteen helpottamaan ohjelmien ja laitteiston käyttöä. IO-moduuleja käytetään järjestelmien tulojen ja lähtöjen laajentamiseen. Moduuli tässä robottisolussa sisältää inputit 8 – 11 ja outputit 12–15. Maksimikuorma on 100 mA. Moduulissa on plus (+) ja miinus (-) -navat. Moduulin avulla voidaan kytkeä lisäominaisuuksia robotin vapaisiin tuloihin ja lähtöihin.

Painonappi on kytketty kuljettimeen. Kuljettimen viive voidaan aktivoida painamalla nappia tai nappi voidaan ohittaa, jolloin kuljetin toimii automaattisesti. Painonappi helpottaa ohjelmien testausta. Silloin kun anturin edessä ei ole tuotetta, kuljetin lähtee 5 s:n viiveellä liikkeelle. Kuljettimen anturin ollessa esteetön voidaan 5 s:n viive kytkeä manuaalisesti nappia painamalla.



KUVA 5. IO-moduuli ja painonappi (Liimatainen 2016.)

## 4 HARJOITUSTYÖN SUUNNITTELU

Harjoitustyön suunnittelussa täytyi ottaa huomioon robotin tuotantosolun laitteisto. Harjoitustyö ei voi olla liian monimutkainen eikä liian vaikeasti toteutettava, koska robottisolun on tarkoitettu oppilaskäyttöä varten. Tämä tarkoittaa sitä, että harjoitustyön tekemisessä ei saa kuluttaa liikaa aikaa. Pysin suunnittelemaan ja toteuttamaan yksinkertaisen ja nopeasti tehtävän harjoitustyön, joka sisältää mahdollisimman monta robottisolun ominaisuutta ja osa-aluetta.

Päätin sisällyttää harjoitustyöhön seuraavat osa-alueet: kappaleen muodon ja värin tunnistus konenäön avulla sekä kappaleen haku kuljettimelta ja paletointi. Näiden osa-alueiden ansiosta harjoitustyön tekijät pääsevät kirjoittamaan yksinkertaista Melfa Basic IV -ohjelmaa, lisäämään tunnistetietoja kameran ohjelmistoon ja lähettämään tiedot robotille. Kuljettimen logiikka on valmiiksi ohjelmoitu pysähtymään kappaleen osuessa sensorin valosäteeseen, joten logiikan ohjelmointi ei ole välttämätöntä.

Valmis ohjelma tulisi etenemään karkeasti seuraavalla tavalla: Erivärisiä kappaleita on asetettu liukuhihnalle sekalaisessa järjestyksessä. Kuljetin pysähtyy sensorin havaitessa kappaleen. Tässä vaiheessa konenäkö tunnistaa kappaleen muodon ja värin sekä pisteen, josta kappale noukitaan imukupilla. Kappale viedään tunnistetietojen perusteella pöydällä sijaitsevaan palettiin määritettyyn pisteeseen. Robottiohjelma tietää, monesko kappale kutakin väriä on paletoitu. Ohjelma on käynnissä niin kauan, kun kappaleita syötetään kuljettimelle tai kunnes kutakin väriä on paletoituna tarpeeksi. Kuljetin ei lähde liikkeelle, jos sen sensorin edessä on kappale. Jos sensori ei havaitse kappaleita, käynnistyy kuljetin 5 s:n viiveellä ja pysähtyy havaitessaan kappaleen sensorin edessä. Tunnistamisen jälkeen kappaleet noukitaan kuljettimelta ja viedään palettiin värin perusteella kahteen riviin. Kussakin rivissä on kolme kappaletta. Kappaleet viedään 50 millimetrin päähän toisistaan.

Harjoitustyössä aion käyttää jo olemassa olevia Feston punaisia, mustia sekä kromattuja sylinterinmuotoisia muovikappaleita (kuva 6). Kappaleiden halkaisija on 40 mm. Punaisten ja kromattujen kappaleiden korkeus on 25 mm. Mustat kappaleet ovat matalampia, 23 mm. Kappaleiden kannessa on kolme aukkoa ja ne ovat sisältä onttuja. Kappaleissa ei ole pohjaa.



KUVA 6. Työssä käytettävät kappaleet (Liimatainen 2016.)

## 5 HARJOITUSTYÖN TOTEUTUS

### 5.1 Prosessin kulku

Aloitin harjoitustyön valmistelemisen tutustumalla tarkemmin tässä työssä käytettävän robottisolun laitteistoon. Käytettävissä oli jo valmiiksi laadittua aineistoa, kuten käynnistysohjeet ja robottikäden liikuttaminen opetusyksikköä käyttäen. Robottisoluun tutustumisen ja ensimmäisten käyttökertojen aikana prosessia valvoi Savonian henkilökuntaan kuuluva laboratorioteknikko Reino Hyvönen.

Harjoitustyön tekemisen aloitin opettamalla robotille pisteet, joista kappale tulee noutaa ja minne se kuuluu viedä. Asetin kappaleen kuljettimelle siten, että kappaleen reuna sijaitti kuljettimeen piirrettyllä viivalla. Käynnistin kuljettimen ja annoin kappaleen pysähtyä sensorin eteen. Opetusyksikköä käyttäen liikutin robottikäden imukuppitarttujan kappaleen yläpuolelle sopivaan kohtaan kuljetinta. Näin määrittelin nostopisteen kappaleelle. Nostopisteen avulla kappale voidaan poimia aina samasta kohdasta. Imukuppityökalu poimii kappaleen ja kuljettaa sen määrättyyn pisteeseen yhden välipisteen kautta. Välipistettä käyttämällä vältetään mahdollisilta törmäyksiltä. Välipisteestä kappale vieään joko vasemmalle tai oikealle haluttuun palettiin. Tähän mennessä valmis ohjelma kuljetti ensin kuusi punaista kappaletta, sen jälkeen kuusi mustaa kappaletta.

Konenäköä varten kappaleista tarvittiin tunnistuskuvat, joita emuloimalla konenäkö oppi tuotteen muodon, värin sekä nostopisteen ja kulman. Kun kamera oli yhdistetty tietokoneelle, otin kuvasarjan, johon konenäkö vertasi ottamaansa kuvaa aina uutta tunnistusta tehdessään. Tunnistuskuvat tallensin tietokoneelle ja lisäsin tuotteeseen. Lopulta väri- ja paikkatiedot oli mahdollista kopioida emuloinnista fyysiseen konenäön ohjelmaan. Tässä vaiheessa kappaleita vaihtamalla ja ottamalla tunnistuskuvan konenäkö erotti kappaleen värin ja muodon noin 80 %:n tarkkuudella, kun hylkäysraja on 70 %.

### 5.2 Paletointiohjelman teko

Paletointiohjelman tekeminen oli tietokoneen ja RoboExplorerin avulla melko nopeaa. Tässä ohjelmassa käytin helposti ohjelmoitavia pisteitä ja komentoja. Robottikättä liikutin opetusyksiköllä ja tässä työssä suurimmaksi osaksi RoboExplorerin omalla työkalulla Pendant. Tämän avulla pystyin liikuttelemaan robottikättä tietokoneeseen kytketyn hiiren avulla.

Robotin paikoituspisteet tulee määrittää omaan tiedostoonsa. Kirjoitettu ohjelma poimii pistetiedot tästä tiedostosta ja käyttää niitä, kun ohjelmassa tulee käsky liikua tiettyyn pisteeseen. Määritettyjä pisteitä oli helppo muuttaa tarpeen vaatiessa joko ajamalla robotti käsin toiseen pisteeseen ja tallentamalla sen pisteen koordinaatit tai muokkaamalla manuaalisesti jo määritetyn pisteen koordinaatteja. Kappaleen asetin kuljettimelle sopivalle kohdalle, johon määritin aluksi nostopisteen. Tools-valikon alta valitsin TP Editor -välilehden, jonka avulla pisteitä määritetään. Tästä avautuvassa näkymässä oli taulukko, josta näkyy mm. pisteen numero, X- ja Y- sekä Z-koordinaatit. Nämä koordinaatit ovat absoluuttisia eli maailmakoordinaatteja. Kun robottikäden imukuppitarttuja oli liikutettu

kappaleen yläpuolelle siten että se kosketti vähän kappaleen pintaa, tallensin tämän pisteen pisteeksi P1. Tallentaminen tapahtui painamalla hiiren oikealla ja valitsemalla valikosta New Teach Point, uusi opetettu piste. Tämän jälkeen klikkasin pistettä P1 uudelleen hiiren oikealla ja valitsin valikosta Learn Robot's Position, opeta robotin sijainti. Samalla tavalla robottikättä liikuttaen tallensin välipisteen P2 pöydän yläpuolelle. Tämän pisteen vasemmalle puolelle tallensin pisteen P3, joka oli punaisten kappaleiden ensimmäinen laskupiste. Tämän pisteen alapuolelle tallensin 50 mm päähän pisteen P4, joka oli punaisten kappaleiden toinen rivi. Samalla tavalla määritin pisteen P2 oikealle puolelle pisteet P5 ja P6. Lopuksi tallensin pisteet tiedostoon Save As -komennolla. Tiedoston voi tallentaa haluttuun kansioon.

Itse ohjelman kirjoittaminen oli nopeaa, kun halutut pisteet oli tallennettu. Ohjelman kirjoittamisen aloitin valitsemalla Tools-valikon alta Program Editor. Tästä avautui ohjelma, johon oli mahdollisuus kirjoittaa robotin ohjelmaa rivi riviltä. Jokaisen rivin alkuun tulee laittaa rivin numero, esimerkiksi 10. Rivinumeroiden tulee olla kasvavassa järjestyksessä. Kirjottaessa ohjelmaa oli selkeintä laittaa rivinumerot kymmenen välein, koska jos ohjelmaan tulee lisättävää rivien väliin, voi rivin 10 ja 20 väliin tehdä vielä yhdeksän riviä lisää.

Kun ohjelma oli valmis, se tallennetaan Save As -kohdasta. Tallentaessa tulee kirjoittaa ohjelman nimen perään tiedostopäätte .MB4. Sen jälkeen ohjelma täytyi ajaa robotin muistiin hakemalla vasemmalta olevista tiedostoista juuri tallennettu ohjelma ja pistetiedosto. Nämä kannattaa tallentaa samaan kansioon, jotta ne on helpompi etsiä tietokoneen tallennustilasta. Vedin valmiin MB4-tiedoston vasemmalle robotin RV-2AJ päälle; näin ohjelma tuli näkyviin robotin alle. Sen jälkeen vedin pistetiedoston robotin alle muodostuneen ohjelman päälle. Tämä tarkoitti sitä, että robotti osasi lukea ohjelmasta pisteet ja tiesi niiden koordinaatit. Jos ohjelmassa olisi ollut virhe, esimerkiksi viallinen input-komento, robotti ei olisi hyväksynyt ohjelmaa ja olisi ilmoittanut virheestä siitä vikakoodilla. Vikakoodin voi lukea ja kuitata CR1-ohjainyksikössä.

Robotin RV-2AJ alla olevassa listassa näkyy valmiita ohjelmia, TD1, TD2, TG1 sekä TG2. Nämä ovat kaksileukaisen pihtitarttujan sekä imukupitarttujan haku- ja vientiohjelmat niille kuuluvista telineistä. Näitä ohjelmia ei saa muokata eikä poistaa!

### 5.3 Konenäkö ja kappaleiden tunnistus

Konenäköä käytin tässä työssä kappaleiden värin ja muodontunnistukseen. Nämä tiedot sekä nostopisteen koordinaatti ja kulma voidaan lähettää robotille Ethernet-yhteyden kautta. Pass/Fail, eli hyväksytty ja hylätty tieto voidaan lukea myös kameran ja robotin I/O-moduulien kautta.

Tunnistustietojen tallentamisen aloitin emuloimalla tunnistuskuvat. Tunnistuskuvat tallensin tietokoneelle haluamaani kansioon. Ensiksi avasin tietokoneen työpöydältä Intellect 1.5.1-ohjelman. Ohjelman avauduttua muodostin yhteyden kameraan Network Neighborhood-kohdasta. Valitsin kameran, jonka IP on 192.168.0.11. Video Display tulee olla päällä, jotta kameran kuva näkyy ohjelmassa. Jos kuva ei näy automaattisesti, se täytyy laittaa päälle View-valikon alta.

Ohjelman päänäkymästä laitoin päälle Inspect ja External Trigger-painikkeet sekä Image-kohdasta Play-painikkeen, jonka kuvake on musta nuoli oikealle. Inspect tarkoittaa, että kameran kuva näkyy Video Display-ikkunassa. External Triggerin avulla voin ottaa kuvia manuaalisesti. Nämä asetukset aktivoidaan sen takia, ettei kamera kuvaisi automaattisesti ja käyttäisi samalla liian paljon koneen prosessointikykyä jatkuvaan kuvaukseen.

Tämän jälkeen asetin halutut kappaleet kuljettimelle kohtaan, jossa ne näkyvät selkeästi kamerassa. Painoin Play-napin vasemmalta puolelta Next Image nappia ja kamera otti kuvan. Tämän kuvan tallensin hiiren oikealla haluttuun kansioon emuloitavaksi tunnistuskuvaksi. Tunnistuskuvia kannattaa ottaa useampi, jotta niistä muodostuisi laajempi kuvasarja johon fyysinen kamera vertaa ottaessaan kuvia.

Kun tunnistuskuvia oli tallennettu tarpeeksi, avasin System-riviltä Network Explorerin. Tämän jälkeen valitsin listasta Emulators ja etsin kameran DVT 542C, joka on tämän robottisolun värinäkökamera. Tämä valitsemalla saadaan kameran emulointitila päälle ja tunnistuskuvien lataaminen voitiin aloittaa. Ohjelma vertaa näitä kuvia kameralla otettaviin kuviin kappaleista. Tunnistuskuvien perusteella määritin muodon ja väritunnistustiedot kameralle. Tunnistuskuvat latsin avaamalla Image-valikon ja valitsemalla Configure Sequence. Tästä avautuvasta näkymästä klikkasin Browse for Images, jonka jälkeen valitsin kansion, johon tallensin tunnistuskuvat. Image Sequence-kohtaan valitsin All Images in Directory, jotta emulointi käyttää kaikkia kansion kuvia. Klikkasin Ok ja tunnistuskuvat oli ladattu emulaattoriin.

Tässä vaiheessa aloitin tunnistetietojen emuloinnin. Valitsin Toolbox-ikkunasta Area Positioning. Tällä etsin tunnistettavan kappaleen kuvasta. Kun Area Positioning oli valittuna, näkyi oikealla Video Display-ikkunassa kappaleiden tunnistuskuvat. Maalasin kappaleen läheltä sen reunoja hiiren vasemmalla näppäimellä. Tämän jälkeen avautui työkalun parametrivalikko. General-välilehden Name-kohtaan annoin työkalulle nimen, Kappale. Kappaleen voi nimetä haluamallaan tavalla. Nimessä ei voi käyttää ääkkösiä, koska ohjelma on englanninkielinen eikä tunnista niitä. Shape välilehdeltä voi tarvittaessa muokata Learn Region tai Search Region asetuksia, eli oppimisalue- ja etsimisalueasetuksia. Options välilehdeltä valitsin Select Task-kohdasta Object Locate. Tämä tarkoittaa sitä, että ohjelma tunnistaa kappaleen ja paikoittaa sille nostopisteen kappaleen pinnasta. Match Score kohtaan laitoin tunnistuksen hyväksymisrajana käytettävän prosenttiarvon 70%, joka oli vakioarvo. Model välilehdeltä klikkasin Relearn-näppäintä, näin kappaleen muoto ja nostopiste oli opetettu. Nostopiste näkyi Video Display-ikkunassa sinisenä ympyränä, jonka sisällä on plusmerkki ja ympyrän ulkopuolella viiva. Tämän viivan tulisi olla kohtisuoraan jotain pintaa vasten. Jos näytöllä on useampi kappale, voidaan opettaa kappaleen muoto ja nostopistetiedot toiselle kappaleelle siirtämällä Learn Region neliötä toisen kappaleen päälle ja painamalla Relearn painiketta uudelleen. Kun halutut kappaleet oli valittu ja opetettu, painoin Ok.



Seuraavaksi opetin kameralle värin tunnistamisen. Toolbox-ikkunasta valitsin Identification, jonka alta Color Identification. Kappaleen maalasin samalla tavalla Video Display ikkunasta. Tämän jälkeen avautuvaan ikkunaan annoin työkalulle nimen, Vari. Työkalun nimeämisessä ei voi käyttää tässäkään tapauksessa ääkkösiä. Training-välilehdeltä Model kohdasta painoin New Model ja Learn. Värille annoin tässä vaiheessa nimen, musta. Identification kohdasta valitsin Match Scoren, eli minimi tunnistamisprosentin. Annoin tälle arvoksi 70%, joka oli vakio. Lisäksi tarkistin, että haluttu väri oli valittu Select the Models to Identify-listasta. Koska tallensin toisen väritiedon, siirsin sinisen laatikon toisen värisen kappaleen päälle Video Display ikkunassa. Training välilehdeltä valitsin New Model ja painoin Learn. Annoin värille nimen, punainen. Identification välilehdeltä tarkistin, että molemmat värit oli valittu, musta ja punainen. Näin kamera haki näitä kahta väriä, kun värintunnistusta tehtiin kappaleille.

Seuraavaksi siirsin nämä työkalut varsinaiselle kameralle. Tool-valikon alta kopioin kappaleen ja värintunnistuksen työkalut yksitellen ja liitin kameralle. Valitsin työkalun ja painoin copy. Sitten valitsin System kohdasta Connections. Tässä ikkunassa näkyi sekä emuloitu kamera, että alussa valittu varsinainen kamera. Valitsin verkossa olevan kameran ja liitin työkalu. Tätä toistin, kunnes kaikki työkalut oli liitetty.

Seuraavaksi muunsin tunnistetiedot robotin String-muuttujaan luettavaan muotoon. Avasin Product-valikon ja valitsin Datalink Settings. Communications välilehdeltä painoin Datalink jonka jälkeen tyhjään ikkunaan hiiren oikealla New Item. Ikkunaan tuli näkyviin DataLink String 1. Tuplaklikkasin sen auki ja valitsin avautuneesta ikkunasta Tool Parameters-kohdasta tunnistustiedot, jotka oli määrä lähettää robotille. Tools-valikosta etsin kohdan Kappale, jonka alavalikosta etsin Pickpoint eli nostopisteen. Valitsin listaan X, Y sekä Angle, eli X-koordinaatin, Y-koordinaatin sekä nostokulman. Värien alta valitsin MatchScoren. Lopuksi lisäsin listan perään Line Termination kohdasta CR-LF ja painoin EOL. Tämä tarkoittaa lähetettävät datan rivinlopetusta. Lopuksi painoin Ok.

Tunnistuksen laitoin päälle Inspect, External Trigger, ja Play-nappia painamalla. External Trigger-painiketta käytin siksi että pystyin ottamaan manuaalisesti yhden kuvan kerrallaan ja tarkastella tuloksia ilman että kamera otti jatkuvasti kuvia. Alhaalla näkyvästä Result Table-ikkunasta tarkastelin kappaleiden tunnistuksen etenemistä. Hyväksytyt tunnistukset näkyivät vihreällä Pass-tekstillä ja hylätyt punaisella Fail-tekstillä.

System-valikon kohdasta Digital I/O:n avulla tarkastelin fyysisiä siirräntöjä. Nämä näkyivät Physical I/O valikosta. Kun tunnistetiedot olivat hyväksytyt vihreällä Pass-tekstillä, näkyi tämä tieto RoboExplorer-ohjelman IO Monitorissa. Tämä tieto kulkee kameran I/O-moduulin kautta robottikäden I/O-moduuliin. Kun väri ja muoto oli molemmat tunnistettu hyväksytysti, näkyi tieto RoboExplorer-ohjelman IO Monitorissa kohdassa Output 8.

## 5.4 Ongelmat

Harjoitustyössä tuli vastaan useita ongelmia, joista suurin osa liittyi konenäön ja paletoinnin yhdistämiseen. Lisäksi kuljettimen pysähtyminen aiheutti ongelmia. Ongelmien ratkaiseminen vaati todella suuria muutoksia alkuperäiseen suunnitelmaan.

### 5.4.1 Kuljettimen pysähtyminen

Alkuperäisen suunnitelman mukaisesti käytettävien pyöreiden kappaleiden kanssa ongelmaksi muodostui kuljettimen pysähtyminen kappaleen havaitsemisen jälkeen. Anturin havaitessa kappaleen kuljetin saa käskyn pysähtyä. Koska kuljettimen moottorissa ei ole jarrua, kuljetin "liukuu" pienen matkan ennen lopullista pysähtymistä. Kuljettimen pysähtyessä oli pyöreä kappale jo kerennyt liikua anturin ohitse, joten viive käynnistyi automaattisesti ja kuljetin lähti liikkeelle viiden sekunnin kuluttua. Alun perin tätä ongelmaa ei havaittu koska kuljettimen alkupäähän asennettu syöttökouru olikin painunut kuljettimen hihnaa vasten, lyhentäen jarrutusmatkaa. Tämä aiheutti sen, että kuljetin pysähtyi nopeammin ja pyöreä kappale jäi sopivasti anturin eteen. Vaihtoehtoina tämän ongelman ratkaisemiseksi oli, joko antaa syöttökourun hidastaa kuljetinta tai vaihtaa käytettäviä kappaleita. Jälkimmäinen vaihtoehto on järkevämpi, koska ei ole viisasta kuluttaa kuljettimen hihnaa tarkoituksellisesti rikki. Lisäksi jo tässä vaiheessa tehdyn paletointiohjelman paikkapisteiden sijainteja on helppo muokata.

### 5.4.2 Ethernet-yhteys kameraan

Ethernet-yhteydessä oli ongelmia alusta alkaen. Kameran oli tarkoitus lähettää robotille tieto kappaleen väristä sekä sijainnista. Nämä tiedot oli tarkoitus lukea robotin string-muuttujasta. Lähtevä data oli Melfa Basic IV -kielellä seuraavassa muodossa selityksineen:

```
OPEN "COM3:" AS#1      Avaa yhteys COM3
INPUT #1,PX,C1$        Lue paikkapiste PX sekä väritiedot C1$
CLOSE #1              Sulje yhteys
```

Paikkapisteessä PX on kappaleen X- ja Y-koordinaatti sekä kulmatieto. Väritiedossa C1\$ on kappaleiden värit musta, punainen sekä kromi.

Yhteyden luonnissa kamera DVT 542C toimi palvelimena ja käytti porttia 3247. Kameran IP oli kiinteä. Yhteyttä yritin saada auki eri porttia käyttämällä, mutta kameran ohjeissa kerrottiin portin 3247 olevan oikea. Monen yrityksen jälkeenkään yhteys ei avautunut, joten ajan säästämiseksi jätin Ethernet-yhteyden kokonaan pois. Lisäksi tietojen siirto käyttäen input- ja output-yhteyksiä vastaa mielestäni enemmän opintojani. Verkkoyhteyksien käyttö ei ole liittynyt opintoihini, joten niiden käyttö olisi ollut hankalaa ja aikaa vievää.

### 5.4.3 Valaistus

Konenäköä käytettäessä on oltava todella tarkka valaistuksen suhteen. Robottisolun katossa on kaksi valoa, loisteputki sekä tavallinen hehkulamppu. Molemmat olivat valoltaan keltaisia ja niiden valoteho oli heikko. Kirkkaisiin lampuihin verrattuna nämä rasittavat vähemmän käyttäjän silmiä. Lisäksi huoneessa on loisteputket katossa. Ikkunoista tuleva valo vaihteli sään mukaan. Kaikki nämä tekijät yhdessä johtivat todella vaihteleviin valaistusolosuhteisiin, mikä konenäköä käytettäessä ei ollut ollenkaan suotavaa. Kameran ottaessa tunnistuskuvia kappaleista syntyi varjoja, jotka hankaloittivat kappaleen tunnistamista. Vastaisuuden varalle robottisolua ja varsinkin konenäköä käytettäessä tulisi valaistusta parantaa. Kamerassa on led-salamavallo, jota lopulta en käyttänyt heijastuksien takia.

### 5.4.4 Heijastukset

Kuljettimen hihna on valkoinen ja pinnaltaan kiiltävä ja suhteellisen liukas. Kappaleita kuvattaessa kuvien laatu ja tarkkuus vaihtelivat huomattavasti pinnan heijastuksien takia salamaa käytettäessä. Säättämällä kameran kuvaa tummemmaksi ja muuttamalla kuvien valotusaikaa heijastuksia pystyi kyllä vähentämään, mutta vastaavasti tunnistettavat kappaleet tummentuivat. Tämän takia värien tunnistaminen hankaloitui. Lisäksi en ollut ottanut huomioon kromisista kappaleista heijastuvaa salamavalloa. Edellä mainittujen syiden vuoksi päätin, etten käytä kromattuja kappaleita ja poistin lopulta salamavalon kokonaan käytöstä. Myös valokuvan prosessointi tietokoneella hidastui ja pari kertaa ohjelman sammui, jos led-salamaa käytti otettaessa kuvaa kiiltävästä kappaleesta tai heijastavasta pinnasta.

### 5.4.5 Pyöreät kappaleet

Konenäköä käyttäessäni huomasin, että pyöreät kappaleet eivät soveltuneet työhön. Kappaleiden kannessa oli kolme reikää, jotka aiheuttivat ongelmia nostopistettä määritettäessä. Muodon tunnistusta tehtäessä automaattinen nostopiste ei ollut keskellä kappaletta ja lisäksi nostokulma oli väärä. Tätä ongelmaa ei synny käytettäessä neliskulmaisia, reiättömiä kappaleita. Neliskulmaisissa kappaleissa nostopiste asettuu keskelle kappaletta ja nostokulma on hyvä, kohtisuoran pinnan vuoksi. Näin ollen robotin nivelet eivät mene epäsuotuisaan asentoon kappaletta nostettaessa. Lopullisessa harjoitustyössä käytin lastulevystä leikattuja neliskulmaisia kappaleita, joista toisen puolen maalasin mattamustaksi ja toisen punaiseksi.

## 6 PÄÄTÄNTÄ

Opinnäytetyön tarkoitus oli perehtyä robotin ja ohjelmistojen käyttöön, paneelin ja logiikan ohjelmointiin sekä suunnitella ja toteuttaa harjoitustyö, jota voidaan käyttää osana opetusta. Opinnäytetyö antaa Savonia AMK:lle opetusta tukevaa aineistoa. Laati maani aineistoa käyttäen opiskelijat oppivat työskentelemään robotin parissa ja saavat käytännönläheistä kokemusta sekä robottisolusta että ohjelmistoista. Itselleni opinnäytetyö antoi vastaavia valmiuksia. Lisäksi robotiikasta saatava käytännön kokemus voi tulevaisuudessa edesauttaa työllistymistä koulutusta vastaaviin tehtäviin.

Robottisolua on käytetty opetuskäytössä todella vähän eikä varsinaisia harjoitustöitä vielä ole toteutettu. Omalta osaltani kokemusta roboteista ennen opinnäytetyön aloittamista ei juuri ollut. Mielestäni Savonia-ammattikorkeakoulu voisi järjestää enemmän käytännönläheistä opetusta robottien sekä ohjelmistojen käyttöön, koska uskon robotiikan olevan yhä merkittävämpää niin tekniikan kuin myös muilla aloilla. Opinnäytetyöstä teki haastavan lisäksi se, ettei Savonialla ollut kyseisen robottisolun käyttöön koulutettua henkilökuntaa. Perusasiat tosin olivat pääpiirteittäin samanlaiset kuin Savonian muiden robottien. Robotin perusteellisemmän käytön jouduin suurimmaksi osaksi opettelemaan itse.

Robotin ohjelmointi ja käyttö olivat suhteellisen helppoa yrityksen ja erehdyksen jälkeen, vaikkei työ toteutunut täysin suunnitellulla tavalla. Jouduin karsimaan työstä muutamia ominaisuuksia, koska en pystynyt niitä pitkäänkin pohdinnan ja yrittämisen jälkeen toteuttamaan. Eniten toteutuksessa jäi harmittamaan se, etten saanut Ethernet-yhteyttä konenäön ja paletointiohjelman välillä toimimaan. Yhteyden toteuttaminen olisi vaatinut ulkopuolisia resursseja, joita tässä työssä ei ollut käytettävissä. Alkuperäisestä suunnitelmansta poiketen opinnäytetyön tuloksena on erillinen paletointiohjelma sekä kappaleen tunnistus konenäön avulla. Alkuperäisestä tavoitteesta jäi puuttumaan näiden kahden yhdistäminen yhdeksi ohjelmakokonaisuudeksi Ethernet-ongelman takia. Laati miani ohjeita voi hyödyntää tulevaisuudessa joko sellaisenaan tai käyttää perustana vastaavanlaisiin harjoituksiin. Yhteysongelmien ratkaiseminen olisi tärkeä osa tämän opinnäytetyön mahdollista jatkokehitystä.

## LÄHTEET

KEINÄNEN Toimi, KÄRKKÄINEN Pentti, LÄHETKANGAS Markku, SUMUJÄRVI Matti 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. 1.painos. Helsinki: WSOY.

MITSUBISHI 2006. Mitsubishi Standard Specifications Manual RV-1A/RV-2AJ Series. Ohjekirja. [verkkodokumentti] [viitattu 29.2.2016] Saatavissa: <http://www.rixan.com/Portals/0/RV-1A-2AJ/1n2specs.pdf>

MITSUBISHI 2009. Mitsubishi Industrial Robot CR1/CR1B Controller INSTRUCTION MANUAL. Ohjekirja. [verkkodokumentti] [viitattu 29.2.2016] Saatavissa: <http://www.roboex.com/Mitsubishi%20Movemaster%20Robot%20Manual/3s.pdf>

MITSUBISHI 2005. E1070 Installation manual. Ohjekirja. [verkkodokumentti] [viitattu 29.2.2016] Saatavissa: <http://www.consys.ru/sites/default/files/documents/E1070.pdf>

COGNEX® CORPORATION 2006. Installation and User Guide For DVT® Vision Sensors. Ohjekirja. [verkkodokumentti] [viitattu 29.2.2016] Saatavissa: <http://www.hvssystem.com/documentations/Cognex/DVT/intellectManualEN.pdf>

## LIITE 1: PALETOINTIOHJELMA

```

10 M1=1
20 *LOOP
25 MVS P1,-50
30 DLY 0.1
40 MVS P1
50 M_OUT(9)=1
60 DLY 0.1
61 MVS P1,-100
62 DLY 0.1
70 MOV P2
80 DLY 0.1
85 IF M1>1 THEN P3.Y=P3.Y-60
90 MOV P3,-50
100 DLY 0.1
101 MOV P3
102 DLY 0.1
104 M_OUT(9)=0
105 DLY 0.1
106 MOV P3,-50
110 MVS P2
120 MOV P1,-100
130 M1=M1+1
140 IF M1<=3 THEN *LOOP
150 P3.Y=P3.Y+180

160 M2=1
170 *LOOP2
180 MVS P1,-50
190 DLY 0.1
200 MVS P1
210 M_OUT(9)=1
220 DLY 0.1
230 MVS P1,-100
240 DLY 0.1
250 MOV P2
260 DLY 0.1
270 IF M2>1 THEN P4.Y=P4.Y-60
280 MVS P4,-50
290 DLY 0.1
300 MOV P4
310 DLY 0.1
320 M_OUT(9)=0
330 DLY 0.1
340 MOV P4,-50
350 MVS P2
360 MOV P1,-100
370 M2=M2+1
380 IF M2<=3 THEN *LOOP2
390 P4.Y=P4.Y+180

400 M3=1
410 *LOOP3
420 MVS P1,-50
430 DLY 0.1
440 MVS P1
450 M_OUT(9)=1
460 DLY 0.1
461 MVS P1,-100
462 DLY 0.1
470 MOV P2
480 DLY 0.1
485 IF M3>1 THEN P5.Y=P5.Y-60
490 MOV P5,-50
500 DLY 0.1
501 MOV P5
502 DLY 0.1
504 M_OUT(9)=0
505 DLY 0.1
506 MOV P5,-50
510 MVS P2
520 MOV P1,-100
530 M3=M3+1
540 IF M3<=3 THEN *LOOP3
550 P5.Y=P5.Y+180

560 M4=1
570 *LOOP4
580 MVS P1,-50
590 DLY 0.1
600 MVS P1
610 M_OUT(9)=1
620 DLY 0.1
630 MVS P1,-100
640 DLY 0.1
650 MOV P2
660 DLY 0.1
670 IF M4>1 THEN P6.Y=P6.Y-60
680 MVS P6,-50
690 DLY 0.1
700 MOV P6
710 DLY 0.1
720 M_OUT(9)=0
730 DLY 0.1
740 MOV P6,-50
750 MVS P2
760 MOV P1,-100
770 M4=M4+1
780 IF M4<=3 THEN *LOOP4
790 P6.Y=P6.Y+180
800 END

```

## LIITE 2: PALETOINTIOHJELMAN KOMENTOJEN SELITYKSET

<b>10 M1=1</b>	Muuttujalle M1 annetaan arvo 1. M1 toimii tässä ohjelmassa laskurina.
<b>20 *LOOP</b>	Uudelleenkiertopiste. Tähän siirrytään rivin 140 ehtojen mukaisesti.
<b>25 MVS P1,-50</b>	Lineaarinen interpolaatio, liiku suoraviivaisesti pisteen P1 yläpuolelle 50 mm.
<b>30 DLY 0.1</b>	Viive, 0,1 sekuntia.
<b>40 MVS P1</b>	
<b>50 M_OUT(9)=1</b>	Aktivoi output 9, tässä tapauksessa imukupin imutoiminto päälle.
<b>60 DLY 0.1</b>	
<b>61 MVS P1,-100</b>	
<b>62 DLY 0.1</b>	
<b>70 MOV P2</b>	Nivel interpolaatio, liiku tämän hetkisestä pisteestä pisteeseen P2. Mielivaltainen liike
<b>80 DLY 0.1</b>	
<b>85 IF M1&gt;1 THEN P3.Y=P3.Y-60</b>	If-lause, jos M1 on suurempi kuin 1, P3-pisteen Y-arvosta vähennetään 60 mm.
<b>90 MOV P3,-50</b>	
<b>100 DLY 0.1</b>	
<b>101 MOV P3</b>	
<b>102 DLY 0.1</b>	
<b>104 M_OUT(9)=0</b>	Aktivoi output 9, tässä tapauksessa imukupin imutoiminto pois päältä.
<b>105 DLY 0.1</b>	
<b>106 MOV P3,-50</b>	
<b>110 MVS P2</b>	
<b>120 MOV P1,-100</b>	
<b>130 M1=M1+1</b>	Muuttujan M1 tämänhetkiseen arvoon lisätään 1.
<b>140 IF M1&lt;=3 THEN *LOOP</b>	Siirrytään ohjelman uudelleenkiertopisteeseen riville 20, jos M1 arvo on pienempi tai yhtä suuri kuin 3.
<b>150 P3.Y=P3.Y+180</b>	Palautetaan pisteelle P3 sen alkuperäinen arvo.
<b>160 END</b>	Ohjelman loppu.

## LIITE 3: OHJEET KONENÄKÖÄ VARTEN INTELLECT 1.5.1 -OHJELMALLE

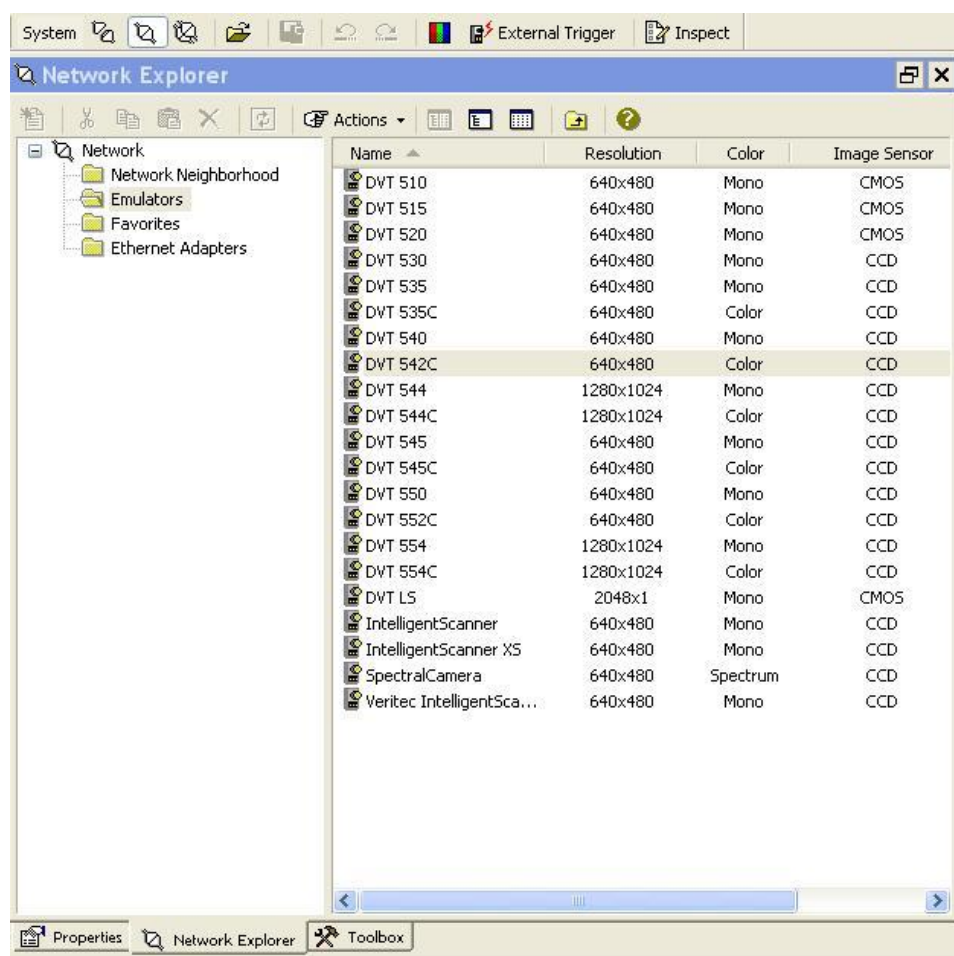
### Ohjelman käynnistäminen ja yhteyden muodostaminen kameraan

Avaa tietokoneen työpöydältä Intellect 1.5.1

Avaa Network Neighborhood

Valitse kamera, jonka IP on 192.168.0.11

Tarkista onko Video Display päällä, jos ei niin käynnistä se View-valikosta



### Tunnistuskuvien tallentaminen

Aktivoi seuraavat painikkeet: Inspect, External Trigger sekä Play-painike

Aseta kappale/kappaleet kuljettimelle kameran alle

Paina Next Image-painiketta

Tallenna kameran ottama kuva haluamaasi kansioon hiiren oikealla ja valitsemalla Save Image As

Tallenna kappaleesta/kappaleista useampia kuvia kuvasarjaa varten



## Kuvien haku emulointia varten

Avaa System-riviltä Network Explorer

Valitse Emulators-kohdasta kamera DVT 542C

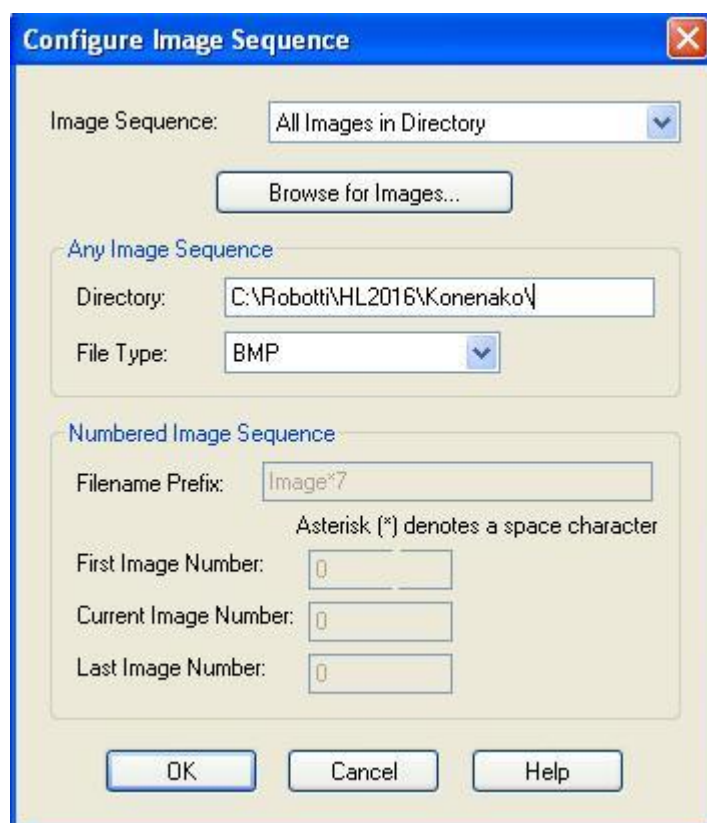
Avaa Image-valikko ja valitse Configure Sequence

Klikkaa Browse for Images

Valitse kansio, johon tallensit tunnistuskuvat

Klikkaa Image Sequence-kohdassa All Images in Directory

Paina Ok



## Muodon ja nostopisteen tunnistetietojen emulointi

Valitse Toolbox-ikkunasta Positioning, jonka alta Area Positioning

Maalaa kappale Video Display-ikkunasta

Nimeä työkalu nyt avautuvaan työkaluparametrivalikkoon General-välilehdelle kohtaan Name

Muokkaa Shape-välilehdeltä Learn Region tai Search Region asetuksia tarvittaessa

Valitse Options-välilehdeltä Select Task-kohdasta Object Locate

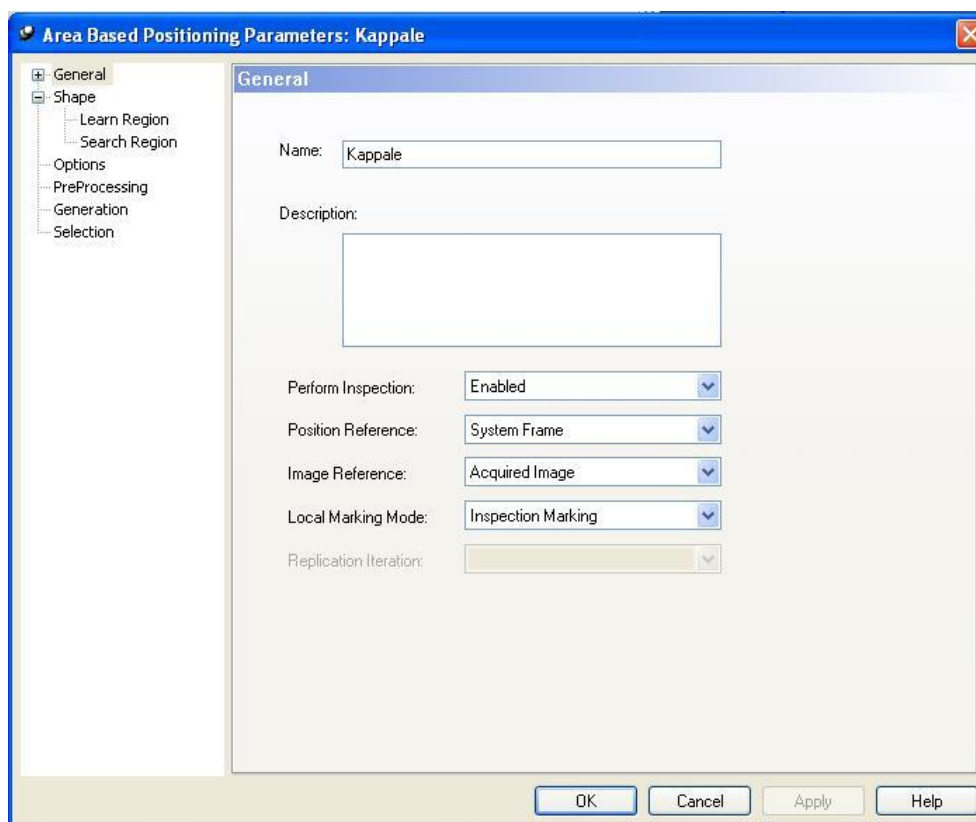
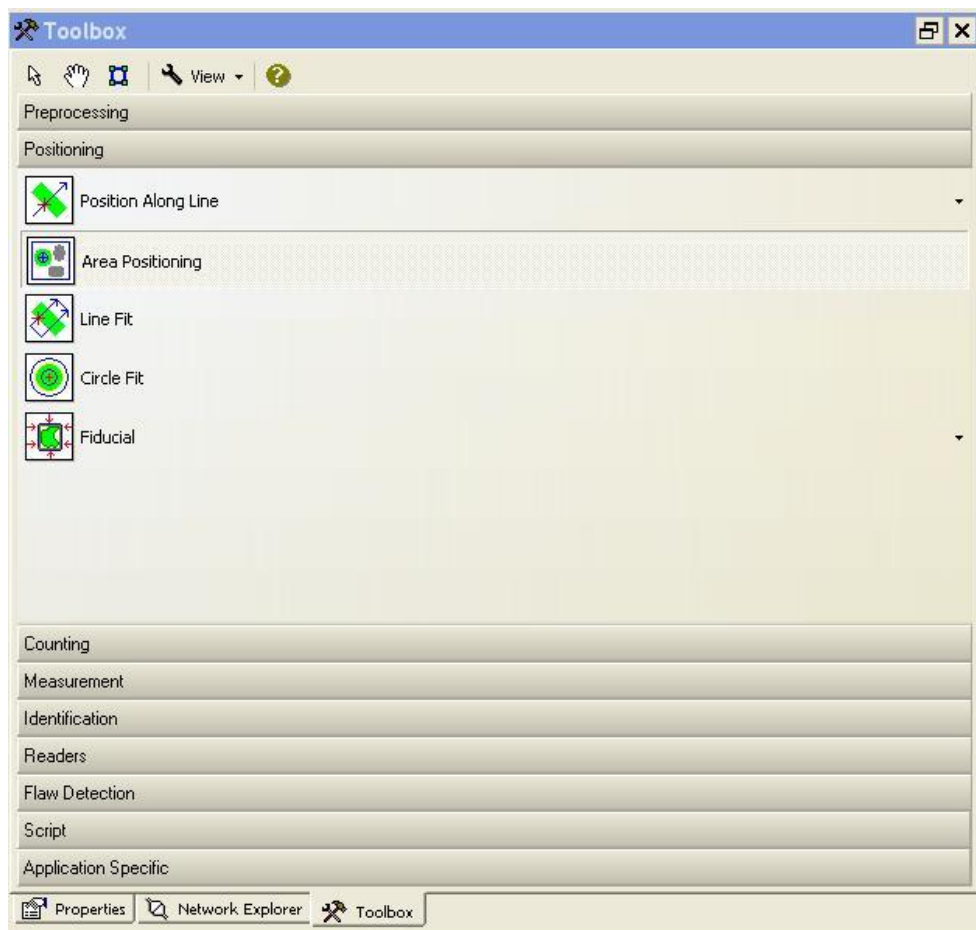
Aseta Match Score-kohtaan tunnistuksen hyväksymisrajan prosenttiarvo, vakiona 70%

Klikkaa Model-välilehdeltä Relearn

Tarkista Video Display ikkunasta että kappaleen muoto ja nostopiste ovat tunnistettuna

Tarvittaessa opeta useamman kappaleen tiedot siirtämällä Learn Region neliötä ja painamalla Relearn

Paina Ok kun halutut kappaleet on valittu ja opetettu



## Värintunnistamisen emulointi

Valitse Toolbox-ikkunasta Identification, jonka alta Color Identification

Maalaa kappale Video Display-ikkunasta

Nimeä työkalu avautuvaan työkaluparametrivalikkoon kohtaan Name

Avaa Training-välilehti ja mene kohtaan Model

Valitse New Model ja klikkaa Learn

Anna värille nimi

Valitse Identification

Aseta Match Score-kohtaan tunnistuksen hyväksymisrajan prosenttiarvo, vakiona 70%

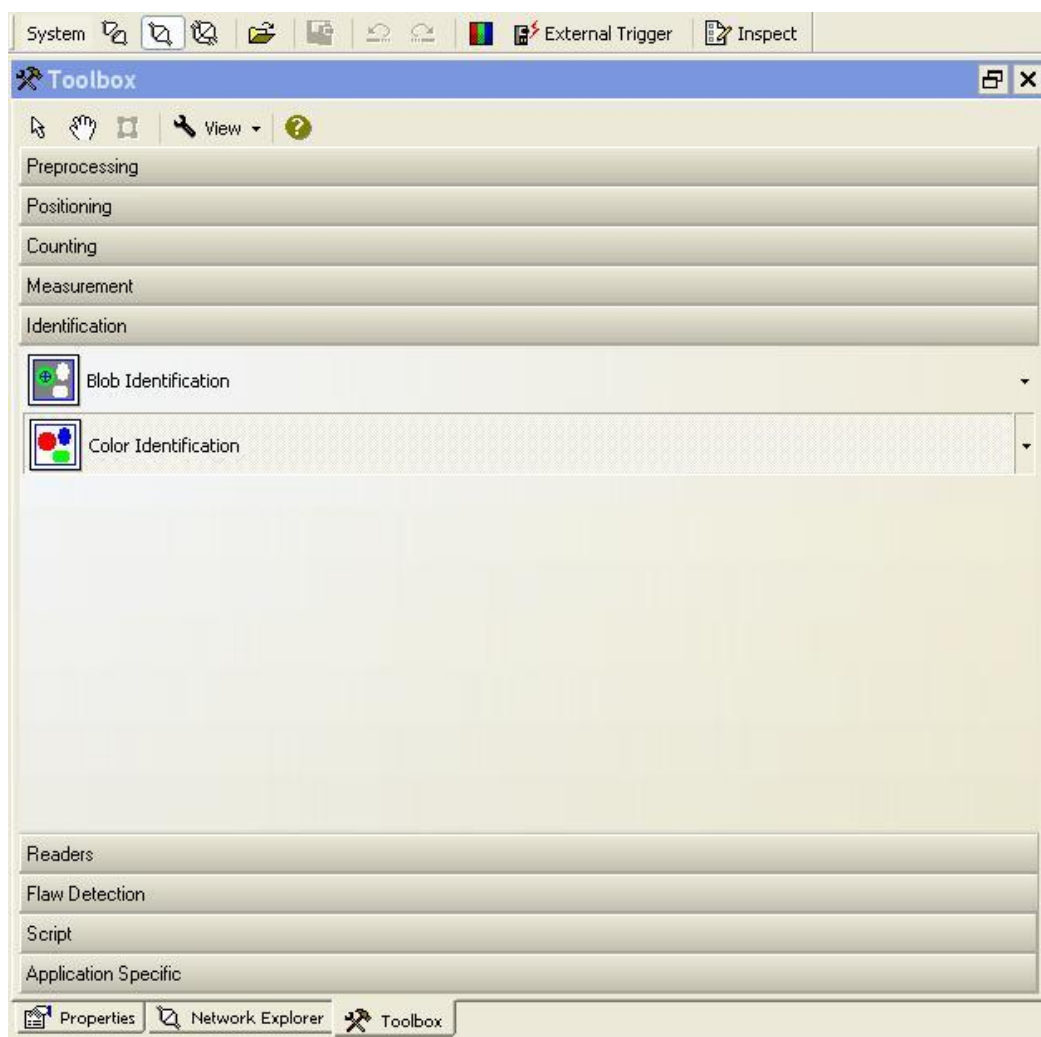
Jos tallennat muita väritietoja, siirrä sininen laatikko toisvärisen kappaleen päälle kohdasta Video Display

Avaa Training-välilehti ja mene kohtaan Model

Valitse New Model ja klikkaa Learn

Anna värille nimi

Tarkista että halutut värit löytyvät Select the Models to Identify-listalta



## Työkalujen kopiointi emulaattorista kameralle

Valitse Tools-valikosta haluttu työkalu ja paina Copy

Avaa System-kohdasta Connections

Valitse verkossa oleva värikamera

Liitä työkalu

Toista kaikkien työkalujen kohdalla

## Tunnistetietojen muuntaminen Ethernet-yhteyttä varten

Avaa Product-valikko

Valitse Datalink Settings

Etsi Communications-kohdan alta Datalink ja valitse se

Klikkaa avautuvaan ikkunaan hiiren oikealla ja valitse New Item

Tuplaklikkaa näkyviin tullutta kohtaa DataLink String 1

Etsi Tool-valikon alta kappaleen muodon ja nostopisteen tunnistustyökalu

Valitse Pickpoint

Lisää vasempaan listaan X, Y sekä Angle

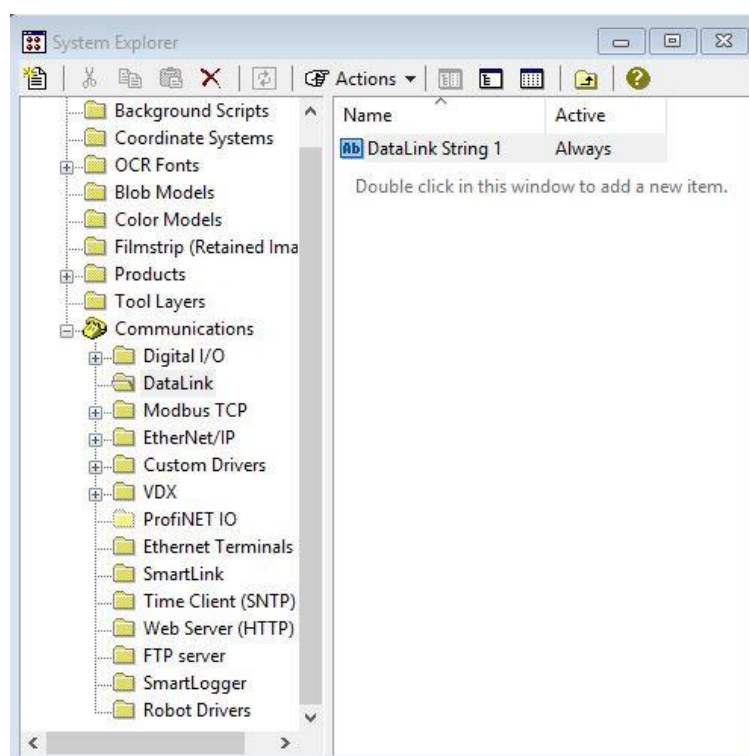
Etsi Tool-valikon alta kappaleen värintunnistustyökalu

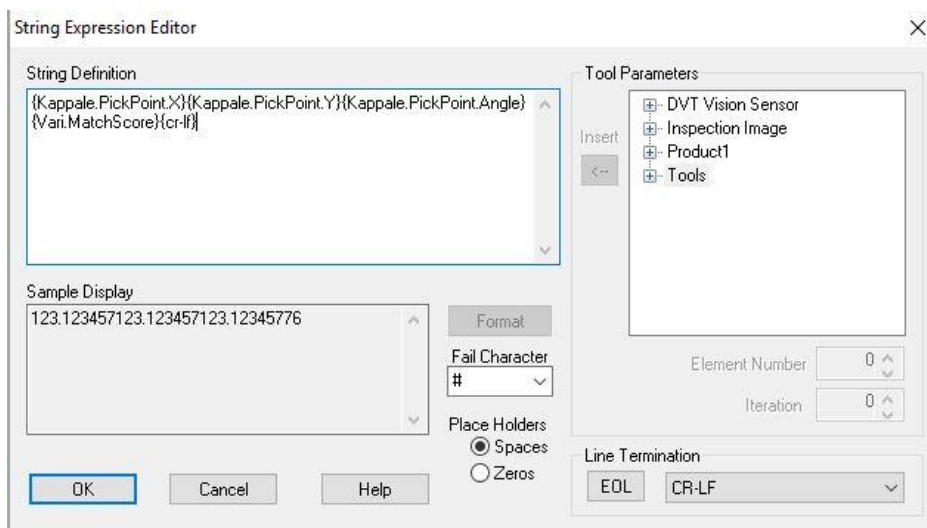
Lisää vasempaan listaan MatchScore

Mene kohtaan Line Termination

Valitse CR-LF ja paina EOL

Paina Ok





## Tunnistuksen päälle asettaminen

Aseta kuljettimelle tunnistettava kappale värikameran alapuolelle

Aktivoi seuraavat painikkeet: Inspect, External Trigger sekä Play-painike

Klikkaa Next Image painiketta

Tarkasta kameras ottaman kuvan tunnistaminen Results Table-ikkunasta ruudun alareunasta

Hyväksytyt tunnistetiedot näkyvät vihreällä Pass-tekstillä, hylätyt punaisella Fail-tekstillä

## Fyysisten siirräntöjen (I/O) tarkastelu

Avaa Systems ja Digital I/O

Avaa Physical I/O

Tunnistuksen onnistuttua kohta Pass muuttuu arvoon 1

