



OPETUSKÄYTTÖÖN SOVELTUVAN ROBOTTISOLUN KEHITTÄMINEN

LAHDEN AMMATTIKORKEAKOULU
Tekniikan ala
Kone- ja tuotantotekniikan koulutusohjelma
Tuotantopainotteinen mekatroniikka
Opinnäytetyö AMK
Kevät 2015
Kari Karjalainen

Lahden ammattikorkeakoulu
Kone- ja tuotantotekniikan koulutusohjelma

KARJALAINEN, KARI: Opetuskäyttöön soveltuvan robottisolun
kehittäminen

Tuotantopainotteisen mekatroniikan opinnäytetyö, 23 sivua, 27 liitesivua

Kevät 2015

TIIVISTELMÄ

Opinnäytetyö tehtiin Lahden ammattikorkeakoulun tekniikan laitokselle. Työn tarkoituksena oli kehittää robottisolu konenäön ja robotin ohjelmoinnin harjoittelua varten

Työhön kuului robottiin liitettävien lisälaitteiden ohjauskeskuksen suunnittelu ja valmistus, jonka sähkösuunnittelussa käytettiin CADS planner ohjelmistoa. Konenäkökameran ja robotin väliseen tiedon käsittelyyn suunniteltiin käyttöliittymä ja ohjelmisto hyödyntäen Visual Studio 2010 ohjelmistoalustaa ja C# ohjelmointikieltä. Konenäön tuottamaa paikkatietoa hyödyntävän ohjelmiston suunnittelu ja toteutus robotille tehtiin IVC- studio- ohjelmistolla.

Työn tuloksena valmistui uusi robotiikan oppimisympäristö, jossa on konenäkökameran tuottamaa paikkatietoa hyödyntävä ohjelmisto robotille sekä käyttöliittymä PC:lle. Lisäksi syntyi robotin I/O liitäntöihin kytkettävä keskus apulaitteiden liittämistä varten sekä ohjauskeskuksen kytkentäkaaviot.

Asiasanat: konenäkö, robotti, teollisuusautomaatio

Lahti University of Applied Sciences

Degree Programme in Mechanical and Production Engineering

KARJALAINEN, KARI: A Robot cell for a learning environment

Bachelor's Thesis in Production Oriented Mechatronics 23 pages, 27 pages of appendices

Spring 2015

ABSTRACT

This thesis was made for the faculty of technology at Lahti University of Applied Sciences. The aim was to develop a robot cell for the training of machine vision and robot programming.

The work included electrical design and manufacturing the robot control unit for connecting external equipment to the robot. Electrical design was done with the CADS Planner software. The user interface for processing information between the machine vision camera and the robot was designed using the Visual Studio 2010 software platform and the C# programming language. A software that utilizes location information generated by the machine vision was designed and implemented with IVC- studio software.

As a result of this study, a new robotics learning environment with software that uses location information produced by machine vision and interface for PC was created. An electric board to be connected to robot I/O connections for connecting auxiliary equipment was also built as well as control center wiring diagrams.

Key words: Machine Vision, Industrial robots, Industrial automation

SISÄLLYS

1	JOHDANTO	1
2	YLEISTÄ ROBOTTIYYPEISTÄ	2
2.1	Käyttöönotto	3
2.2	Paristovarmennus	3
2.3	Liitynnät	4
2.3.1	I/O -liityntä	5
2.3.2	Sarjaliityntä	7
2.4	Ohjelmointi	8
2.5	Tiedostotyypit	9
3	OHJAUSKESKUS	11
4	KONENÄKÖ	13
4.1	Yleistä	13
4.2	Liitynnät	14
4.3	Ohjelmointi	15
4.3.1	Ohjelman rakenne	16
4.3.2	Koordinaatisto	16
5	KÄYTTÖLIITTYMÄ	18
5.1	Liityntä	18
5.2	Ohjelmointi	19
6	YHTEENVETO	20
	LÄHTEET	21
	LIITTEET	23

1 JOHDANTO


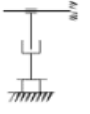

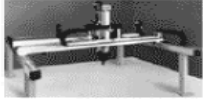
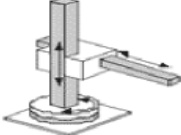
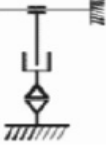

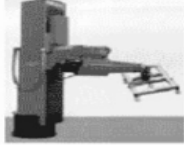
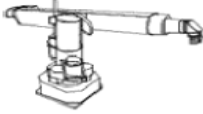
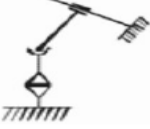


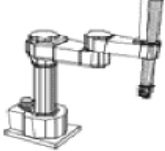
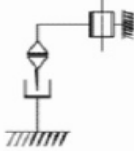



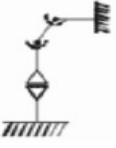

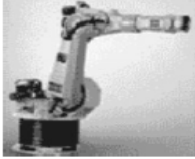




Lahden ammattikorkeakoulu (LAMK) on yksi Suomen suurimmista ja monialaisimmista korkeakouluista ja kuuluu itsenäisenä tulosyksikkönä 14 päijät-hämäläisen kunnan omistamaan kuntayhtymään. Tämän opinnäytetyön tavoitteena on tuottaa Lahden ammattikorkeakoulun tekniikan laitokselle opetuskäyttöön soveltuva robottisolun, jolla tullaan vahvistamaan aiemmin vähemmälle huomiolle jääneen konenäön opetusta muuten monipuolisessa robotiikan oppimisympäristössä.

Projekti alkoi Lahden ammattikorkeakoulun tekniikan laitoksen edustajien kanssa pidetyllä aloituspalaverilla, missä määriteltiin opetussolun tarpeet. Opinnäytetyö rajattiin koskemaan yhtä opetuspistettä kolmen solun ryhmästä. Työssä tuotetaan solun ohjauskeskus, sähköpiirustukset, paikoitustiedon tuottava ohjelma kameralle, käyttöliittymä PC:lle robotin ja kameran väliseen kommunikointiin sekä kameran tuottamaa paikoitustietoa hyväksikäyttävä ohjelma robotille.

Opinnäytetyön kohde oli mielenkiintoinen ja sitä tehdessäni pääsin hyödyntämään aikaisempaa työkokemusta erilaisista koneautomaatioon liittyvistä toimeksiannoista ja niiden läpiviemisestä.

2 YLEISTÄ ROBOTTITYYPEISTÄ

Teollisuusrobotit on kuvattu standardissa ISO 8373 ja sen määritelmän mukaan teollisuusrobotti sisältää vähintään kolme uudelleenohjelmoitavaa vapausastetta (akselia). Robotti voidaan sijoittaa joko kiinteäksi tai liikkuvaksi osaksi automaatioosovellusta. Monikäyttöisenä sitä voidaan mukauttaa eri sovelluksiin ilman fyysisiä muutoksia.

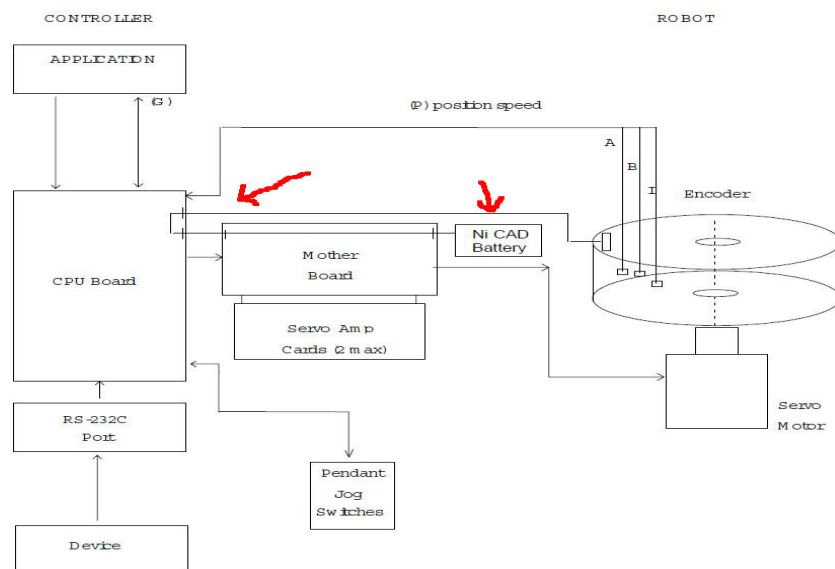
Robotti	Akselit		Esimerkki
	Rakenne	Kinemaattinen kaavio	
 <p>Suorakulmainen robotti</p>			
 <p>Sylinterirobotti</p>			
 <p>Napakoordinaatisto-robotti</p>			
 <p>SCARA-robotti</p>			
 <p>Kiertyvänivellinen robotti</p>			
 <p>Rinnakkaisrakenteinen robotti</p>			

KUVA 1. Erilaisia robottityyppejä (Muokattu lähteestä)

Tämän opinnäytetyön robottityyppinä on SCARA (Selective Compliant Assembly Robot Arm). (KUVA1) Tämän robottityyppin esitteli Sankyo Seiki, Pentel ja NEC vuonna 1981.

2.1 Käyttöönotto

Opinnäytetyöni kohteena olevan robotin moottori sekä encoderi-kaapelit ovat olleet irrotettuna kontrollerista varastoinnin aikana ja sisäisestä kytkennästä minkä vuoksi robotin kotipisteen tiedot ovat hävinneet (KUVA 2). Tämän takia käyttöönoton yhteydessä robotille täytyi tehdä niisanottu kotiinajo (ABS Homing) joko konfigurointiohjelmalla tai erillisellä käsiohjauslaitteella. Käsiohjauslaitetta ei ollut käytettävissä joten kotiinajo oli tehtävä 3000SPWin ohjelmalla.



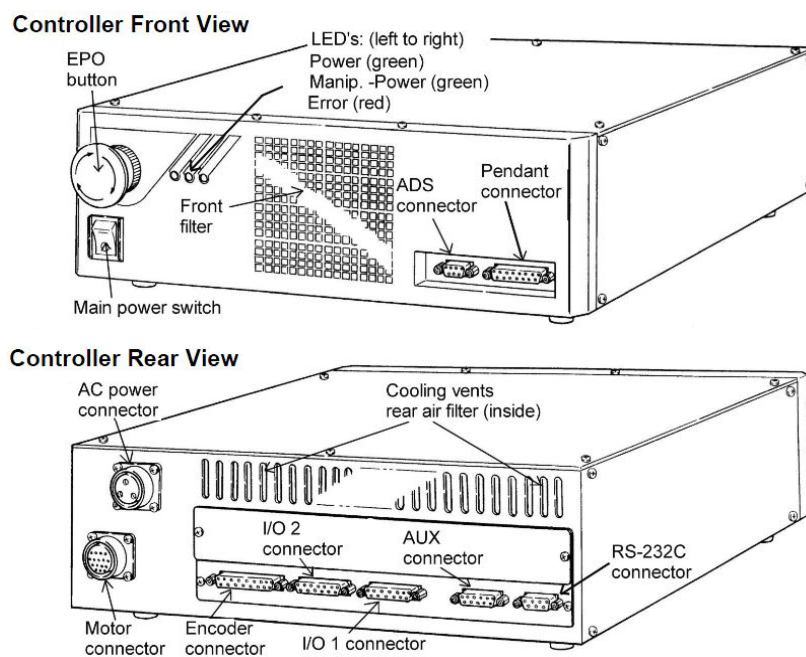
KUVA 2. Paristovarmennuksen kytkentä (Sankyo 2000a)

2.2 Paristovarmennus

Prossessorikortilla oleva SDRAM/DRAM-muisti on patterivarmennettu 4,8 V:n NiCad-paristolla. Patterivarmennusta tarvitaan robottiohjelmassa olevien muuttujien tallettamiseen sähkökatkon aikana. NiCad-paristolla varmistetaan myös encoderin absoluuttinen paikkatieto (ABS home): Valmistaja ilmoittaa NiCad-patterien elinkaareksi ilman latausta 14 vuorokautta ja 3 vuotta kontrollerin ollessa käytössä. Täyden kapasiteetin saavuttamiseen tarvitaan 24 tunnin latausaika.

2.3 Liitynnät

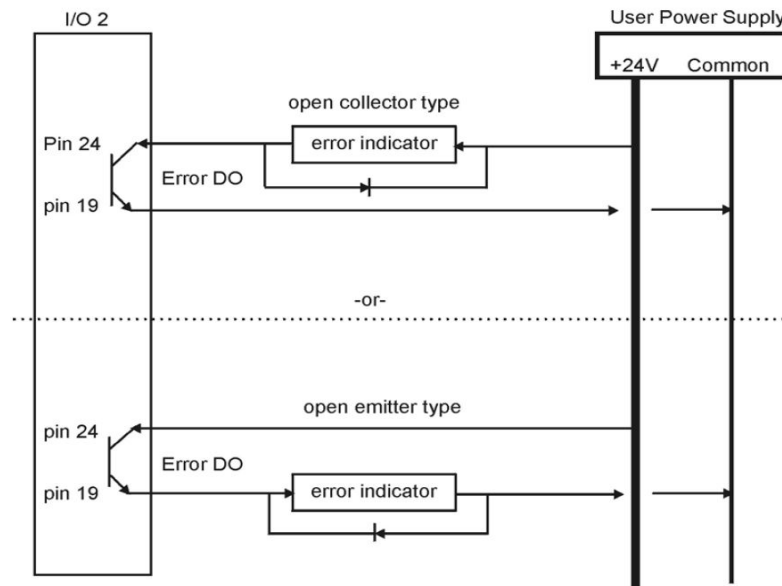
SC3000-sarjan kontrolleri (KUVA3) sisältää peruskonfiguraatiossa ADS-ohjelmointiportin, käsiohjainportin, moottoreiden jännitesyötön, encoderin liitännän, AUX-liitännän ARCnet-protokollaa käyttävien lisälaitteiden liittämiseen, I/O-liitännät sekä RS232C-portin sarjamuotoisen tiedon käsittelyyn.



KUVA 3. Liityntöjen sijainti (Sankyo 2000b)

2.3.1 I/O -liityntä

Standardikokoonpanossa kontrollerin tulot ja lähdöt on jaettu kahteen erilliseen I/O1- ja I/O2-liitäntään. Kytkennässä voidaan käyttää joko PNP tai NPN kytkentää. Esimerkki häiriösignaalin kytkennästä on esitetty kuvassa 4.



KUVA 4. Häiriösignaalin NPN ja PNP kytkentä. (Sankyo 2001a)

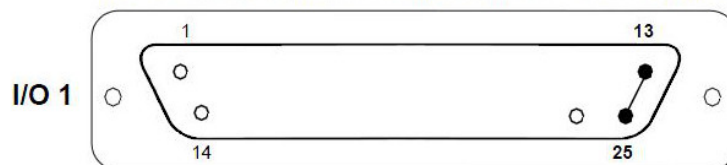
I/O-rajapintaa voidaan laajentaa (TAULUKKO 1) lisäkorteilla jopa 384:n I/O-tietoon saakka. Liittiminä I/O1- ja I/O2-porteissa käytetään D25 liittintä ja muissa I/O-liitynnöissä käytössä on D37-liitin.

TAULUKKO 1. Laajennuskortit (Sankyo 2001b)

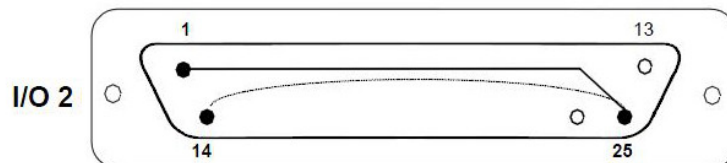
DI/DO Type	Board ID	Address Range	Definition
Standard I/O 24 points total	CPU	1-24	1-16 DI 17-24 DO
Expanded I/O AD1319 Feature or *AD1308 Feature 64 points total	Board 1	25-88	25-40 DI 41-56 DO 57-72 DI 73-88 DO
	Board 2	89-152	89-104 DI 105-120 DO 121-136 DI 137-152 DO
Expanded I/O AD1309 Feature 128 points total	Board 1	153 - 280	153 - 216 DI 217 - 280 DO
	Board 2	281 - 408	281 - 344 DI 345 - 408 DO
	Board 3	409 - 536	409 - 472 DI 473 - 536 DO

Käytettäessä kontrolleria ilman ulkoisia I/O-liityntöjä tulee I/O1 ja I/O2 jumpperoida kuvan 5 mukaisesti. Jumpperoinnilla mahdollistetaan kontrollerin toiminta ilman ulkoisia Häätä-Seis ja servojen virityspainikkeita.

The I/O 1 connector (male) should have pins 13 and 25 jumpered.



The I/O 2 connector (male) should have pins 1 and 25 jumpered.

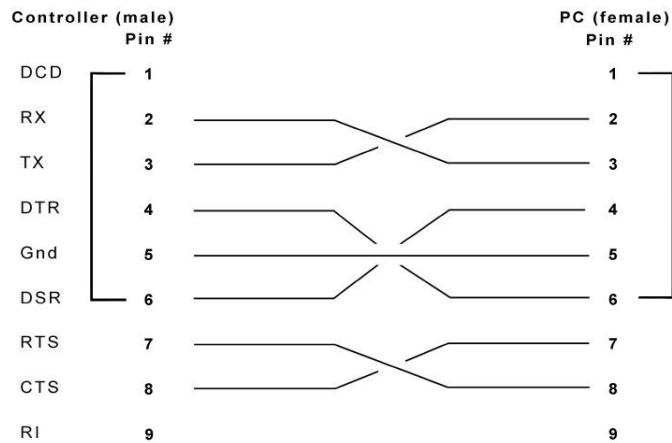


KUVA 5. I/O jumpperointi. (Sankyo 2000c)

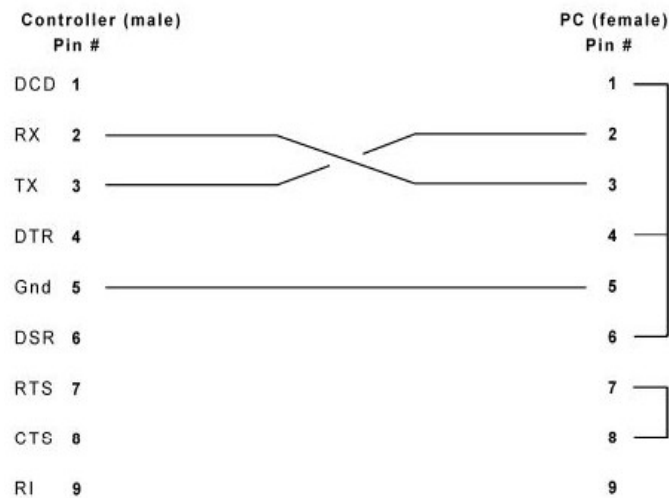
2.3.2 Sarjaliityntä

Ohjainyksikkö on varustettu kahdella sarjaliikennettä tukevalla portilla, joista ADS-porttia käytetään laitteen ohjelmointiin. Standardin RS-232C mukaista porttia käytetään lisälaitteiden ja ohjainyksikön väliseen kommunikointiin.

Ohjelmointi ja RS-232C-kaapelin valmistin robotin manuaalista löytyvän kytkennän mukaan (KUVA 6) ja (KUVA 7).



KUVA 6. ADS-ohjelmointikaapeli (Sankyo 2000d)



KUVA 7. RS-232C kaapeli (Sankyo 2000e)

2.4 Ohjelmointi

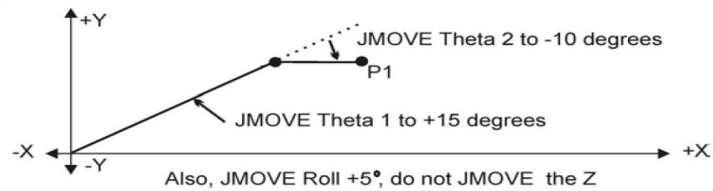
Ohjelmointikielenä Sankyon roboteissa käytetään Sankyo Seiki Manufacturing Company Ltd:n kehittämää Sankyo Structured Language/Enhanced (SSL/E) -ohjelmointikieltä.

```

PROG do_angle ( )
POSITION P1, P2 ;
P1 = -350, -250, 65, 0 ;
P2 = -275, -170, 65, 0 ;

MOVE ( P1) ;           // Move to point P1
SPEED (50) ;          // Speed is 50 %
JMOVE (<15,-10,,5>) ; // T1 moves to 15 °, T2 moves to -10 ° . and Roll
                        // moves to +5 °. // Z does not change.
MOVE ( P2) ;           // Move to point P2
END

```



KUVA 8. Esimerkki SSL/E ohjelmasta (Sankyo 2001a)

Ohjelmointityökaluna käytettiin Buzz versiota 2 (KUVA 9) joka vaatii robotti-valmistajan suositusten mukaan toimiakseen Windows 98-, Windows NT- tai Windows 2000 -käyttöjärjestelmän. Mainitut käyttöjärjestelmät ovat tulleet käyttöikänsä loppuun eikä sellaisen tietokoneen hankkiminen, missä se olisi ollut asennettuna, ollut järkevää, joten asensin robottiohjelmiston kehitysympäristön Windows XP -käyttöjärjestelmällä varustettuun tietokoneeseen, missä se toimikin suhteellisen luotettavasti.

```

Sankyo Buzz 2.0 - Stesti - [C:\Vpu\Stesti\ROBOTTI.SSL]
File Edit Debug Robot Window Help
//~ Robotti.ssl
//~ Paikoitetaan robotti kameralta tulleen paikkatiedon perusteella
//~ Muodostetaan 6x2 paletti
//~ Tehdään paletinvaihto
//~
TPOSITION TP[4]:
TPOSITION Pal[12]:
POSITION KOTI[1]={<-165.163.311.923.50.000.0.00>}; // Annetaan paikka joka ei ole kamera
// Tässä paletissa 6 saraketta ja 2 riviä
// Sarakeväli 30.0mm
// Riviväli 42.0mm
Real PalX=-330.467; // Paletin X arvo eka ruutu -330.467 488.432 -116.0
Real PalY=488.432; // Paletin Y arvo eka ruutu
Real PalZ=150.0; // Luovutuskorkeus paletissa
Real PalR=0; // Luovutuskulma paletissa
Real PalSV=30.0; // Paletin sarakeväli
Real PalRV=42.0; // Paletin riviväli
Real PalSN=0; // Paletin sarake numero
Compiling KAMERA
SSL/E Compiler, Version 3.00 beta1
Copyright 1988-1999 Sankyo Seiki Mfg. Co., Ltd.
File : KAMERA.TSK was created.
Compilation(s) Complete
For Help, press F1 Offline 1 1/167 INS

```

KUVA 9. Buzz 2:n käyttöliittymä

2.5 Tiedostotyypit

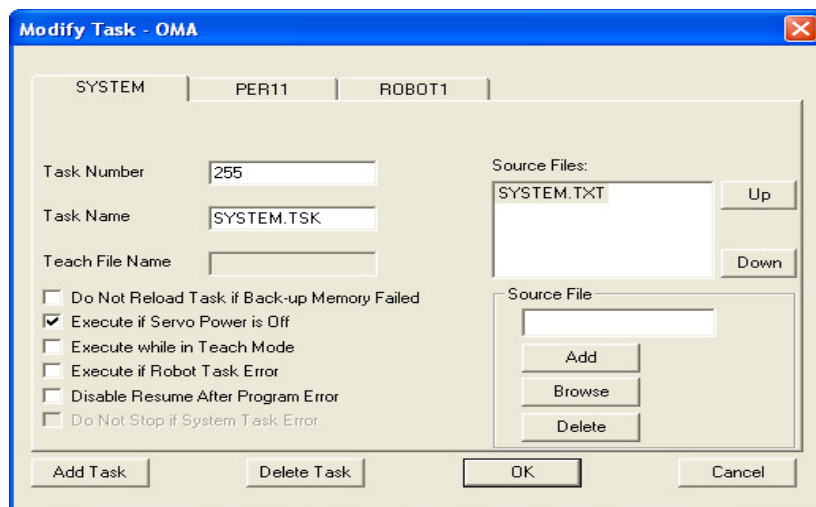
SSL/E-ohjelmissa käytetään viittä erilaista tiedostotyyppiä ja näiden nimeämisessä tulee käyttää DOS-käyttöjärjestelmän 8+3 nimeämistapaa.

Kontrollerin suorittamat ohjelmat on lajiteltu eri tehtäväkokonaisuuksiin, joita kutsutaan taskeiksi. System taskin tehtävänä on käynnistää, suorittaa ja pysäyttää robotin ohjelmia. Tätä taskityyppiä voi olla vain yksi kutakin kontrolleria kohden eikä se tue SSL/E-liikekäskyjä. System taski voidaan myös jättää käyttämättä; tälle taskille tulee antaa aina numero 255.

Robotin liikekäskyt esitellään ja suoritetaan Robot taskissa. Taskin nimenä voidaan käyttää mitä tahansa kahdeksanmerkkistä nimeä, mutta tiedoston tarkenteen tulee olla .TSK. Tässä taskissa voidaan suorittaa kaikkia SSL/E-komentoja.

Robottiin liitettävien lisälaitteiden ohjaamiseen on varattu kahdeksan taskia, joiden numerointi sijoittuu 1:n ja 18:n välille. Lisälaitetaskit voidaan nimetä vapaasti, mutta tiedoston tarkenteen tulee olla .TSK. Valmistaja viittaa manuaaleissaan tähän taskityyppiin merkinnällä PERIPHERAL Task.

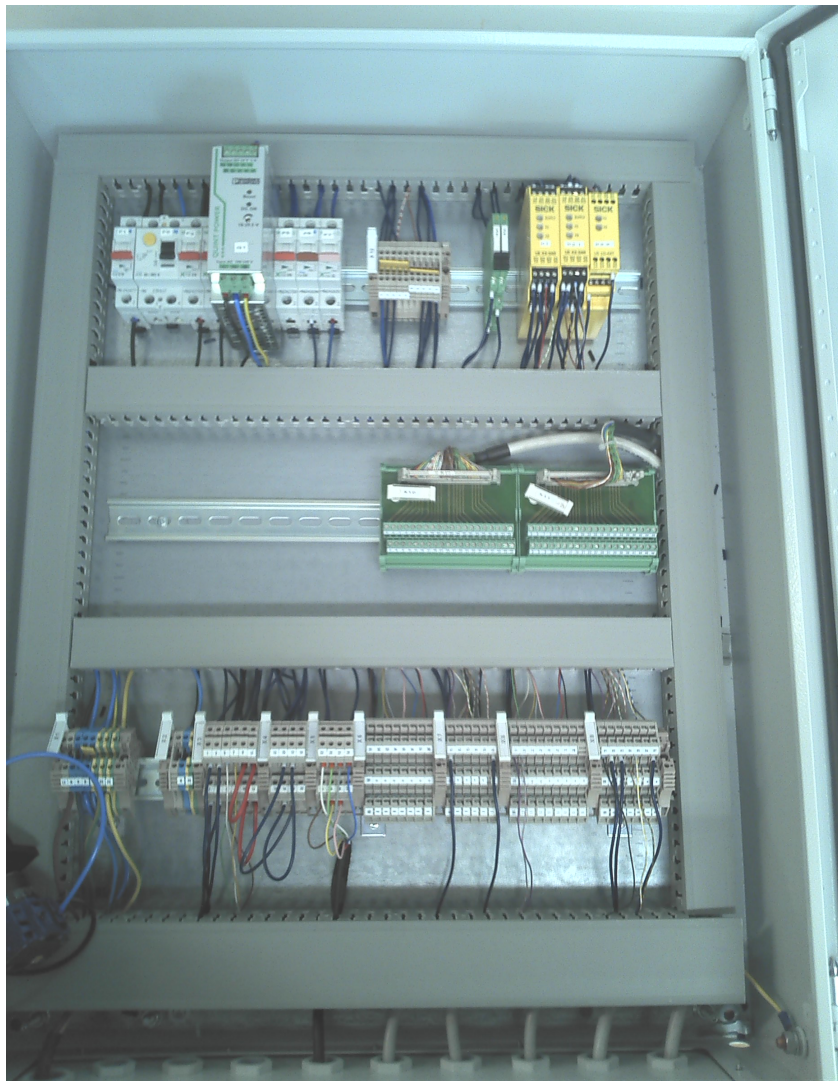
Paikkatiedon määrittelyyn voidaan käyttää TEACH Data -tiedostoa, joka tarvitaan TPOSITION käskyn yhteydessä. Tämä tiedoston takenne tulee olla .DAT ja tiedosto voidaan nimetä vapaasti, mutta suositeltavaa on käyttää samaa nimeä kuin robot taskissa.



KUVA 10. Taskien määrittely Buzz ohjelmassa

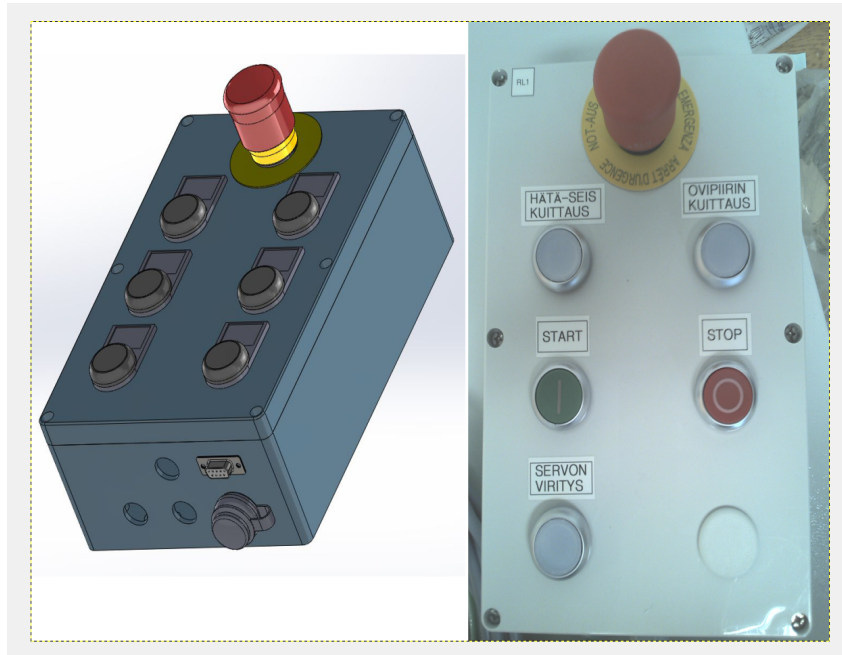
3 OHJAUSKESKUS

Koneiden suunnittelussa tulee ottaa huomioon voimassaolevat määräykset ja standardit. Konedirektiivi 2006/42/EY valmistui vuonna 2006 ja otettiin käyttöön 29.12.2009. Tämän pohjalta suunnittelin robotin keskusyksikköön liitettävän ohjauskeskuksen (KUVA 11) sekä siihen liittyvän ohjauspanelin(KUVA 12). Ohjauskeskusta voidaan käyttää esimerkiksi robottiin liitettävien apulaitteiden, kuten tarttujien ja kuljettimien, kytkemiseen sekä niiden tuottaman tilatiedon rajapintana.



KUVA 11. Ohjauskeskus

Ohjauspaneeliin tulevien käyttöpainikkeiden (KUVA 12) layout suunnittelun toteutin lataamalla kotelovalmistajan sekä painikkeita valmistavan yrityksen kotisivuilta komponenttikirjastot. Komponenttikirjastoja sekä Solidworks ohjelmaa hyväksikäyttäen syntyi mitoituskuvat ohjauspaneeliin sijoitettujen käyttöpainikkeiden ja johdotusten tarvitsemien aukkojen tekemistä varten.



KUVA 12. Käyttöpaneelin havainnekuva ja toteutus

Ohjauspaneelin tehtävänä on toimia käyttäjärajapintana, jota voidaan käyttää robotin työkierron käynnistämisen ja pysäyttämisen lisäksi servojen viritykseen sekä Hätä-Seis-toimintojen kytkemiseen ja kuittauksen.

4 KONENÄKÖ

Konenäkö on koneen kykyä nähdä ja tulkita näkemäänsä - aivan kuten ihmisen näkökyky on ihmisen kykyä nähdä ja tulkita näkemäänsä. Konenäköjärjestelmä sisältää muun muassa kameran, jolla muodostetaan kuva halutusta kohteesta, sekä tietokoneen tai muun laitteen, jossa suoritettavien algoritmien avulla kuvasta tehdään tarvittavat päätelmät. (Ropponen & Pulkinen 2009.)

4.1 Yleistä

Tämän opinnäytetyön yhteydessä oli käytettävissä SICK:n valmistama mustavalkokamera tyyppimerkinnältään IVC-2DM1121. Kuvakennon resoluutio tässä mallissa on 1024 x 768 ja riittää tähän käyttötarkoitukseen hyvin.

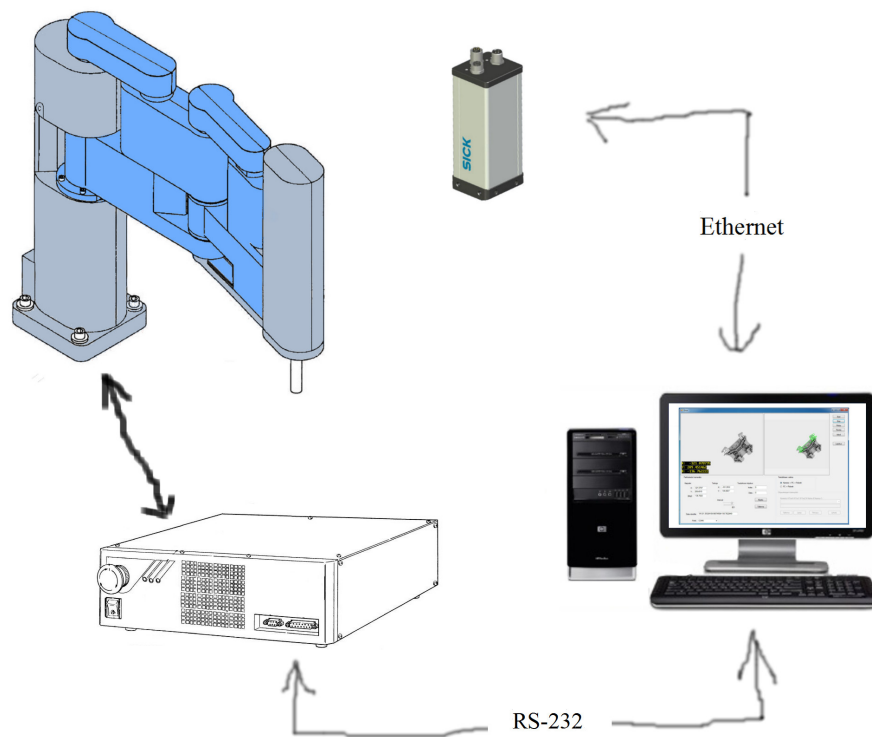


KUVA 13. Konenäkökamera IVC-2DM1121 (Sick 2014)

4.2 Liitynnät

Konenäön tuottaman informaation siirtämiseen sitä tarvitsevalle laitteelle on käytettävissä erilaisia menetelmiä. IVC-2DM1121-kamerassa informaation siirto on mahdollista Ethernet-, RS485- tai Power I/O -liitännän kautta.

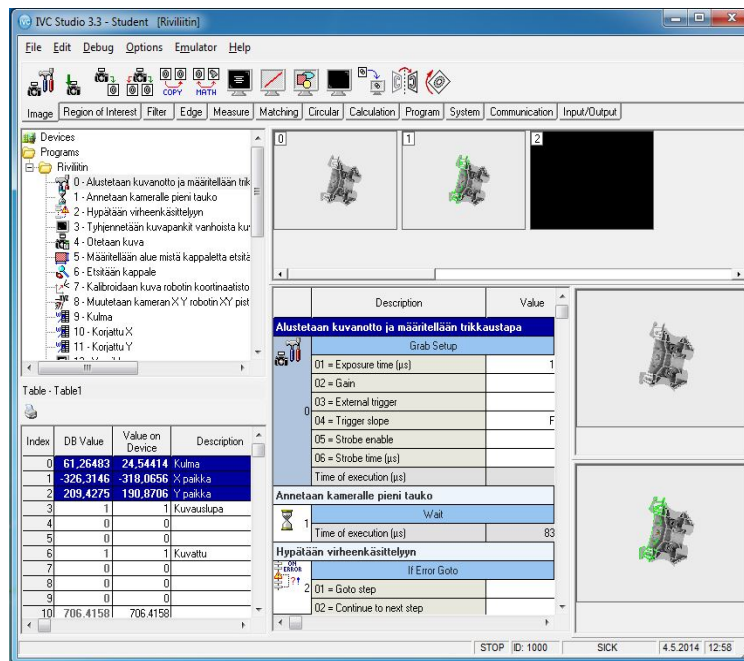
Kameran tuottaman informaation siirtämiseksi robotin käyttöön hyödynsin Ethernet-liitynnän tuomia mahdollisuuksia. Tässä liityntämallissa tietokoneessa oleva ohjelma lukee kameran muistiin tallennettuja tietoja Active-X-rajapintaa hyväksikäyttäen.



KUVA 14. Kameran liityntä PC:n kautta robotille

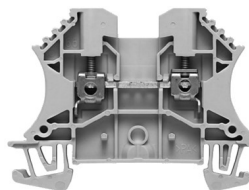
4.3 Ohjelmointi

Konenäkökameran hyödyntämiseen tarvitaan kameralle tapauskohtaisesti räätälöity ohjelma, jonka avulla kameras ottamaa kuvaa tulkitaan. Kuvaa tulkitsevan ohjelmiston kehittämiseksi eri konenäkökameroiden valmistajat tarjoavat omiin kameramalleihinsa suunniteltuja työkaluja, joilla haluttuja toimintoja kamerassa saadaan aikaiseksi. (KUVA 15.)



KUVIO 15. IVC Studio 3.3:n käyttöliittymä

Tässä opinnäytetyössä tehtävänäni oli laatia tietokoneelle asennetulla IVC Studio -ohjelmalla kameralle sovellus, jolla tutkittavasta kappaleesta saadaan X- ja Y-koordinaatit sekä kappaleen asento. Ohjelman teko ja testausvaiheen tutkittavaksi kappaleeksi valitsin sähkökeskuksien kalustamisessa käytettävän riviliittimen tyypiltään WDU2.5 (KUVA 16).

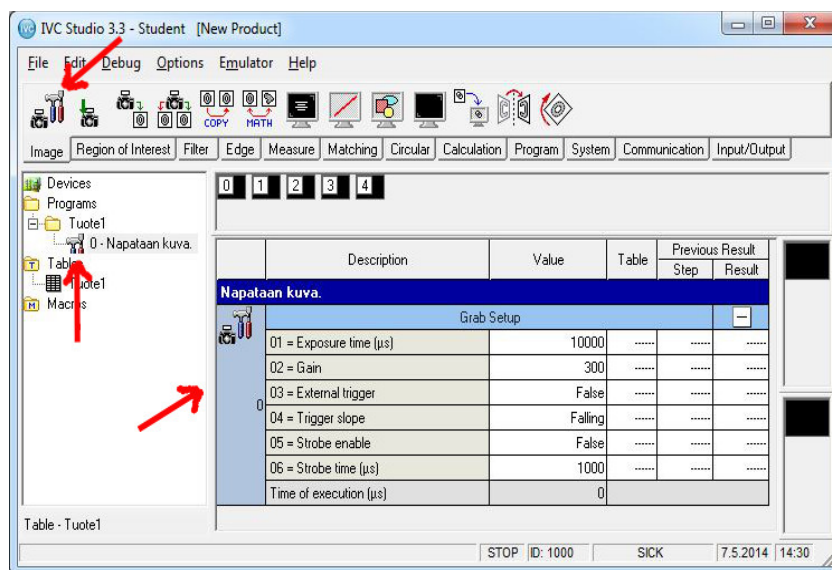


KUVA 16. Wdu2.5 riviliitin (Weidmüller 2014)

4.3.1 Ohjelman rakenne

Kuvaa tulkitseva ohjelma koostuu erilaisista komponenteista, joille voidaan määrittellä erilaisia syöttöarvoja, viittauksia taulukkoon tai toisiin komponentteihin. Taulukoita voidaan käyttää myös tietojen välittämiseen ulkopuolisille laitteille.

Ohjelman nimeämisen jälkeen määrittellään kuvan ottamiseen liittyvät asetukset Grap Setup -komponentilla. Käytettävä komponentti näkyy ohjelman rakennepuussa sekä siihen liittyvät parametrit omassa lohkoissaan. (KUVA 17.)

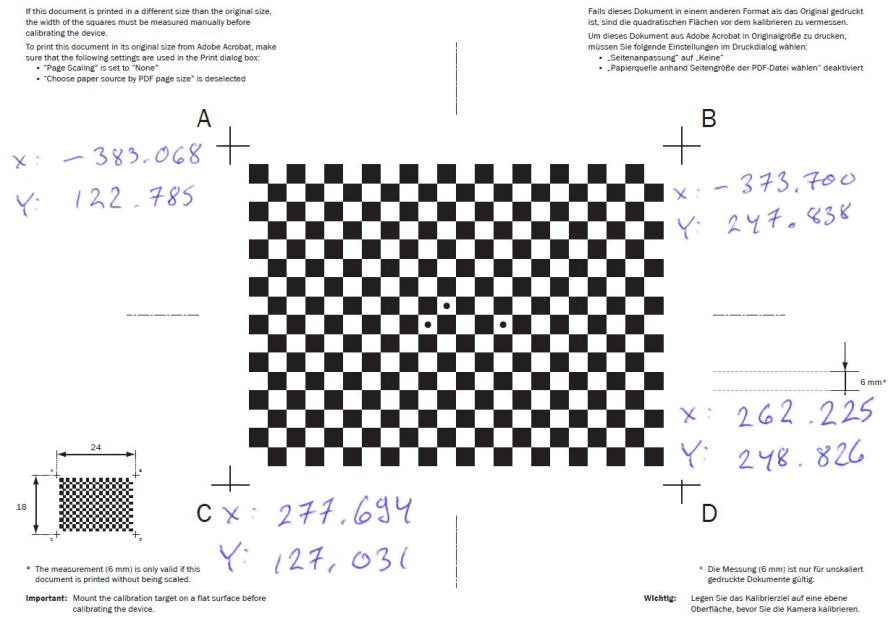


KUVA 17. Näkymä komponentin sijoittamisen jälkeen

4.3.2 Koordinaatisto

Kameran tulkitsemasta kuvasta saatu kappaleen paikkatieto ja kääntökulma täytyy muuttaa robotille sopivaan muotoon. Käytännössä tämä tarkoittaa sitä, että kameralle opetetaan robotin koordinaatisto, minkä jälkeen kamera laskee ja muuntaa kuvasta kappaleesta automaattisesti robotin ymmärtämät X- Y- -koordinaatit sekä kappaleen kääntökulman.

Koortinaatiston opettamisen aloitin tulostamalla IVC Studio ohjelmasta tähän tarkoitukseen suunnitellun kuvan, johon on merkitty A-, B-, C- ja D-pisteet (KUVA 18).

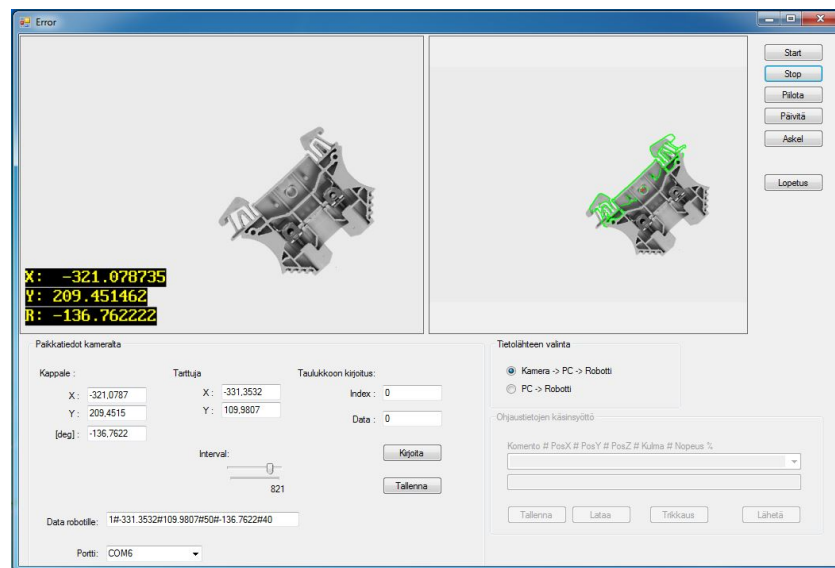


KUVA 18. Kameran kalibrointiintarkoitettu kuva (Sick 2014c)

Kalibrointikuvan sijoitin robotin työalueelle kohtaan, josta kuvattava kappale poimittaisiin robotin tarttujalla. Pisteiden A-D paikkatiedot selvitin siirtämällä robotin Z-akseliin käsin haluttuun kohtaan ja kirjaamalla ylös Buzz 2.0 -ohjelmasta senhetkiset X- ja Y-koordinaatit.

5 KÄYTTÖLIITTYMÄ

Käyttöliittymä on se laitteen, ohjelmiston tai minkä tahansa muun tuotteen osa, jonka kautta käyttäjä käyttää tuotetta. Esimerkiksi tietokoneohjelmassa käyttöliittymä tarkoittaa sitä ohjelman osaa, jonka käyttäjä näkee tietokoneen näytöllä, ja sitä tapaa (hiiri, näppäimistö), jolla hän käyttää ohjelmaa. (Wikipedia 2014c)



KUVA 19. Käyttöliittymä

5.1 Liityntä

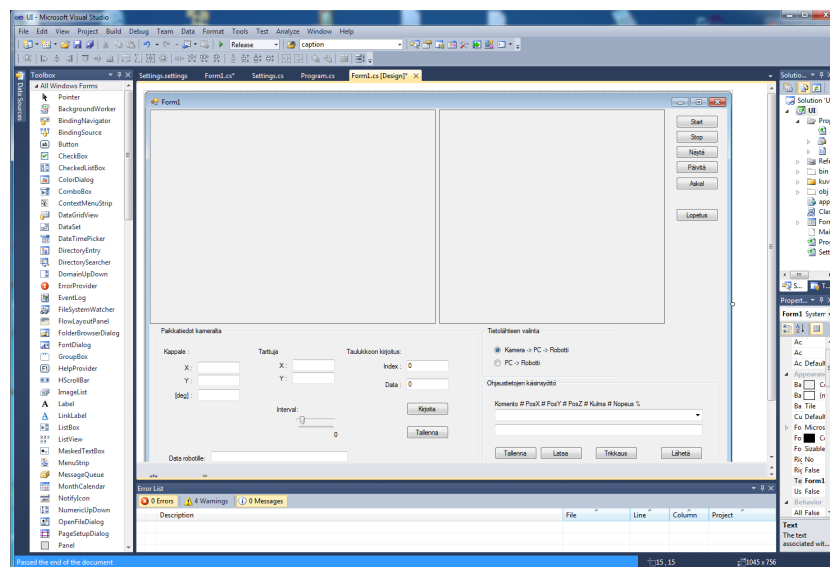
Rajapintana käyttöliittymän ja kameran välillä käytin kameranvalmistajan toimittamaa Active X -komponenttia. Tiedonsiirron käyttöliittymän ja robotin välillä toteutin Visual studion SerialPort-komponentin avulla.

ActiveX on toinen nimi Microsoftin Windows-käyttöjärjestelmässä käytetylle Component Object Model (COM) -tekniikalle. COM-komponentit ovat uudelleenkäytettäviä ohjelmistokomponentteja. (Wikipedia 2014a)

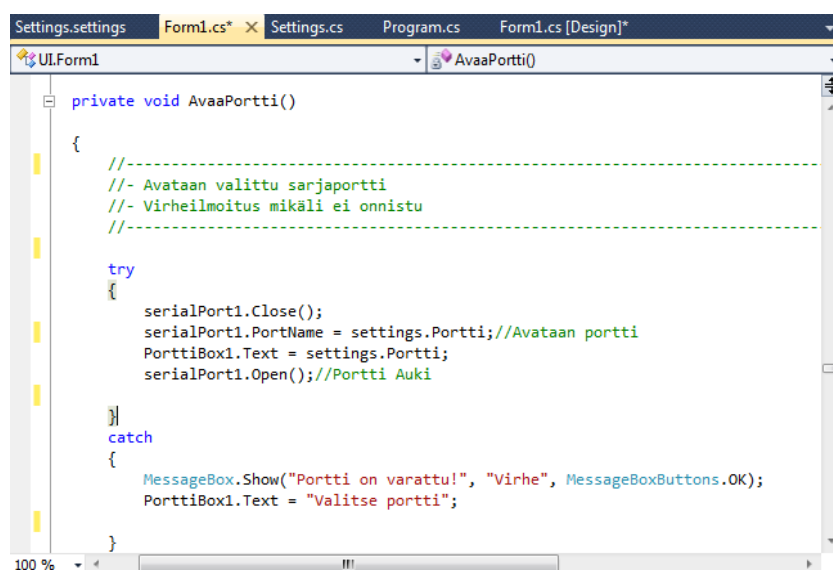
5.2 Ohjelmointi

Tilaaajan toiveesta käyttöliittymä toteutettiin Microsoft Visual studio -ohjelmankehitysympäristöön (KUVA 20) ja (KUVA 21) kuuluvalla C# Sharp -ohjelmointikielillä.

C# on Microsoft-yhtiön .NET-konseptia varten kehittämä ohjelmointikieli, joka julkaistiin kesäkuussa 2000. Kieli kehitettiin yhdistämään C++:n tehokkuus ja Java-kielen helppokäyttöisyys (Wikipedia 2014b).



KUVA 20. Microsoft Visual studio ohjelman käyttöliittymä



KUVA 21. C# Sharp esimerkkikoodia

6 YHTEENVETO

Opinnäytetyön tavoitteena oli tuottaa Lahden ammattikorkeakoulun tekniikan laitokselle opetuskäyttöön soveltuva robottisolun jota tullaan hyödyntämään opetuksen siirryttyä uusiin tiloihin.

Opinnäytetyön tuloksena syntyi robottisolun ohjauskeskus kytkentäkaavioineen, konenäköohjelmisto kameralle, käyttöliittymä PC:lle sekä niiden tuottamaa tietoa hyödyntävä paletointiohjelma robotille. Lopputulos oli mielestäni projektin aloituspalaverissa läpikäydyn määrittelyn mukainen. Robottisolun jatkokehittelyssä tulee tehtäväksi erilaisten tarttujien suunnittelu sekä konenäön tarvitseman erityyppisten valaistusratkaisujen tutkiminen.

LÄHTEET

Elektroniset lähteet:

Konedirektiivi 2006/42/EY. Koneturvallisuuden standardit [viitattu 16.11.2014].
Saatavissa: <http://www.sfs.fi>

Ropponen, S. & Pulkkinen, K. 2009 Konenäöllä paljon mahdollisuuksia teiden
ylläpidossa ja hoidossa [viitattu 16.11.2014].
Saatavissa: <http://www.tieyhdistys.fi/binary/file/-/id/8/fid/52/>

Wikipedia 2014c. [viitattu 16.11.2014]. Saatavissa:
<http://fi.wikipedia.org/wiki/Käyttöliittymä>

Wikipedia 2014a. [viitattu 16.11.2014]. Saatavissa:
<http://fi.wikipedia.org/wiki/ActiveX>

Wikipedia 2014b. [viitattu 16.11.2014]. Saatavissa:
http://fi.wikipedia.org/wiki/C_sharp

Kuvalähteet:

Kuva 1. Teollisuusrobottityypit [viitattu 16.09.2014] Muokattu lähteestä:
<http://www.themethodsengineer.com/2013/01/classication-and-types-of-robots.html#more>

Kuva 2. Sankyo. 2000a. SCARA Series Hardware Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 3. Sankyo. 2000b. SCARA Series Hardware Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 4. Sankyo. 2001a. SC3000 Series Installation and Specifications Manual
[viitattu 16.11.2014]. Saatavissa: Laitteiston hankinnan yhteydessä

Taulukko 1. Sankyo. 2001b. SC3000 Series Installation and Specifications Manual
[viitattu 16.11.2014]. Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 5. Sankyo. 2000c. SCARA Series Hardware Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 6. Sankyo. 2000d. SCARA Series Hardware Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 7. Sankyo. 2000e. SCARA Series Hardware Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 8. Sankyo 2001a. SSL/E Language Reference Manual [viitattu 16.11.2014].
Saatavissa: Laitteiston hankinnan yhteydessä

Kuva 13. Sick. 2014a. 2D vision IVC-2D, IVC-2D HighRes [viitattu 16.11.2014].
Saatavissa: <https://www.mysick.com/saqqara/wrapper.aspx?id=im0048448>

Kuva 16. Weidmüller. WDU 2.5 [viitattu 16.11.2014].
Saatavissa: http://catalog.weidmueller.com/catalog/Start.do?localeId=en_DE&ObjectID=1020000000

Kuva 18. Sick 2014c. IVC Studio 3.3. Calibration target IVC-2D [viitattu 16.11.2014]. Saatavissa: <https://www.mysick.com/eula.aspx?type=ZIP&path=saqqara/DC0001808&lang=en>

LIITTEET

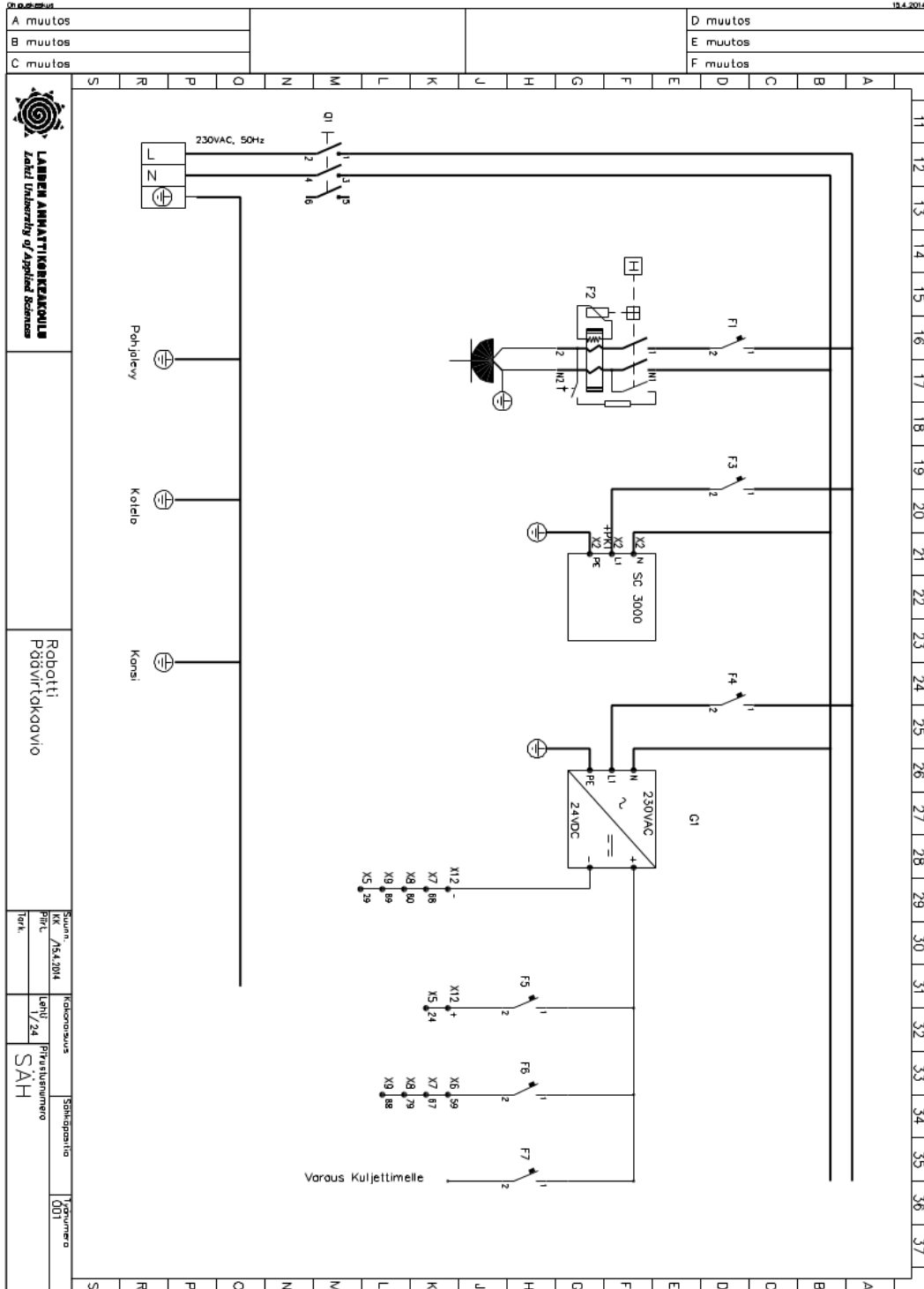
Liite 1. Kytkäkaaviot

Liite 2. Osaluettelo

Liite 3. Paletoinnin ohjelmalistaus

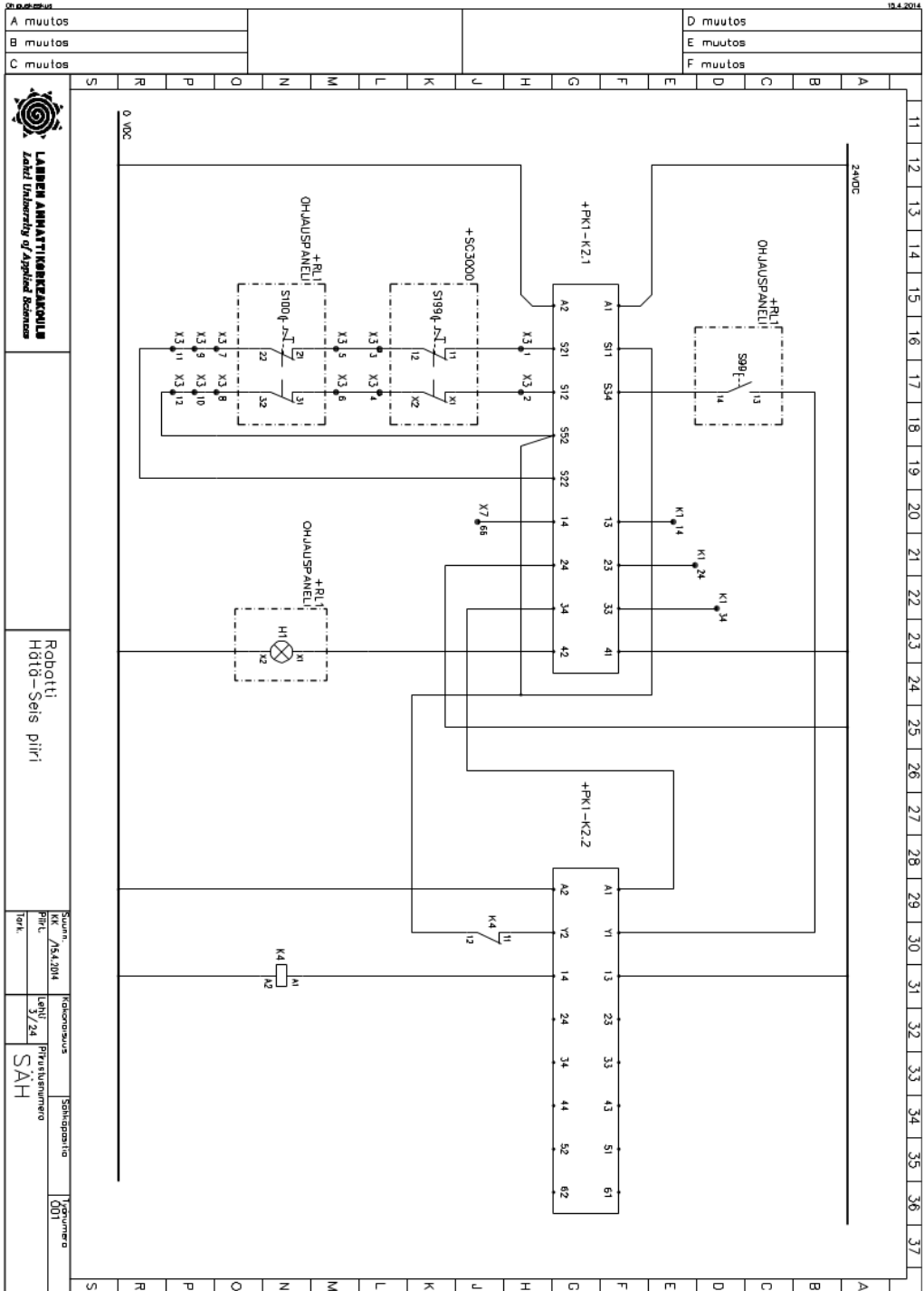
LIITE 1

Kytentäkaavio. 1/24



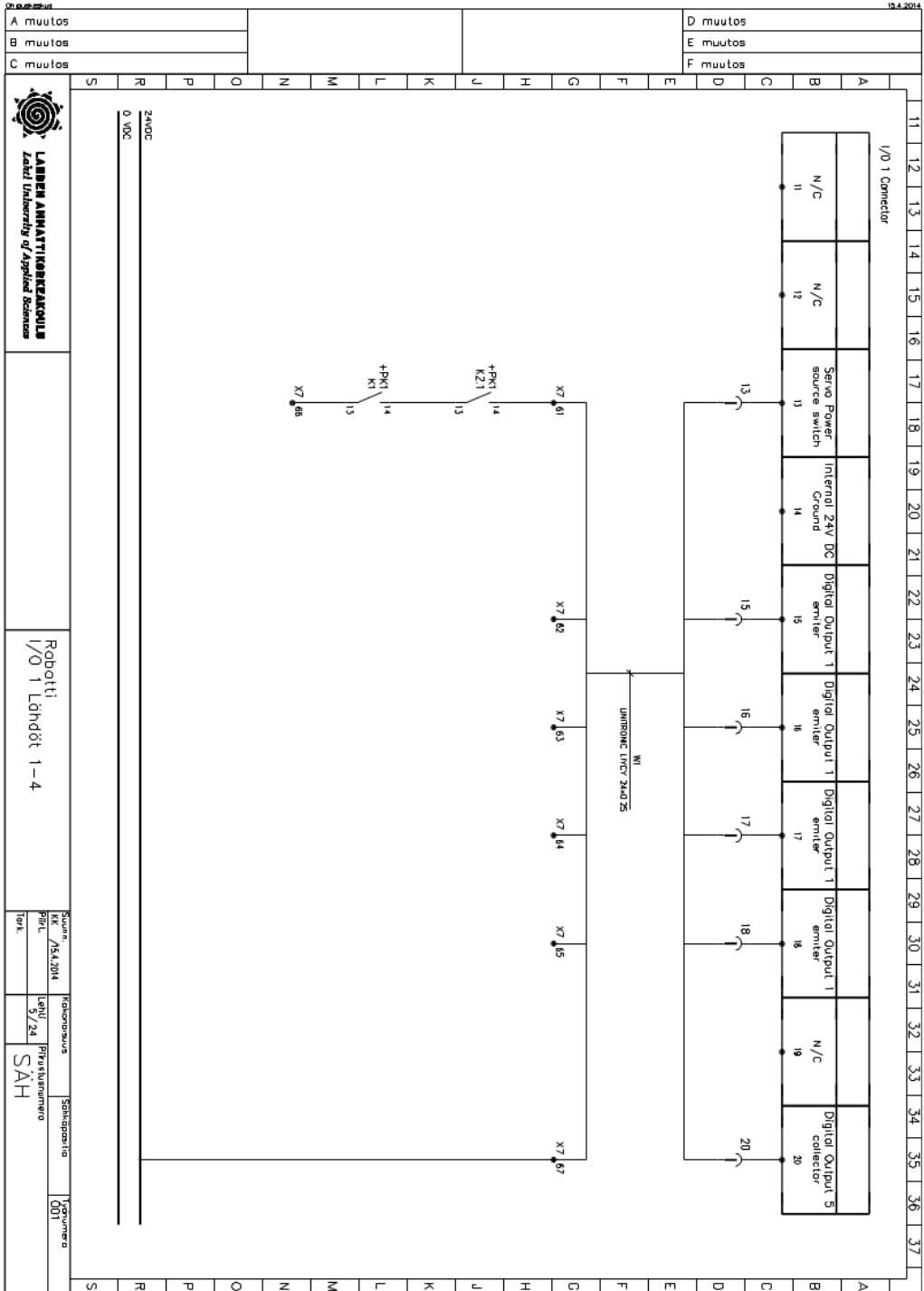
LIITE 1

Kytentäkaavio. 3/24



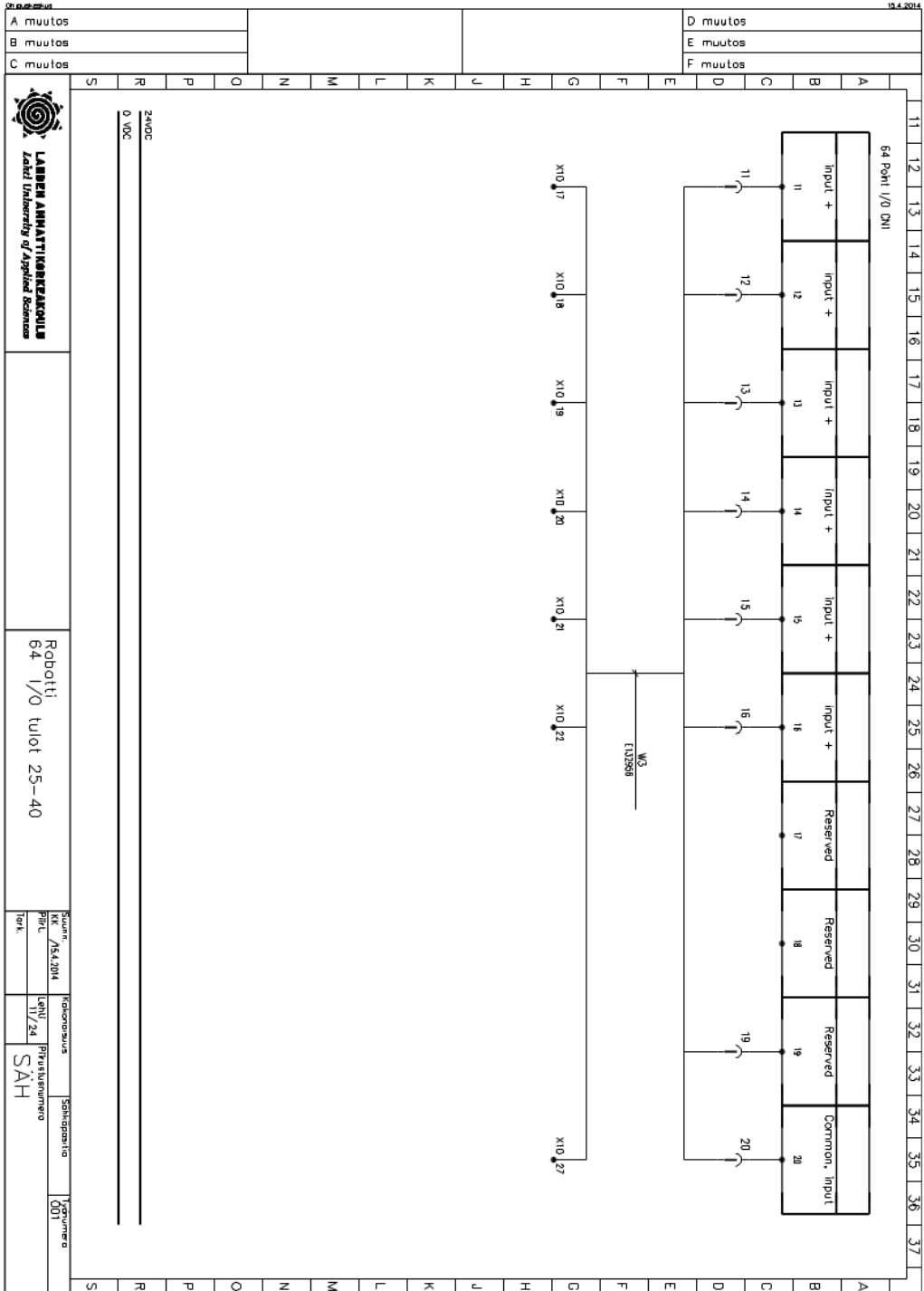
LIITE 1

Kytentäkaavio. 5/24



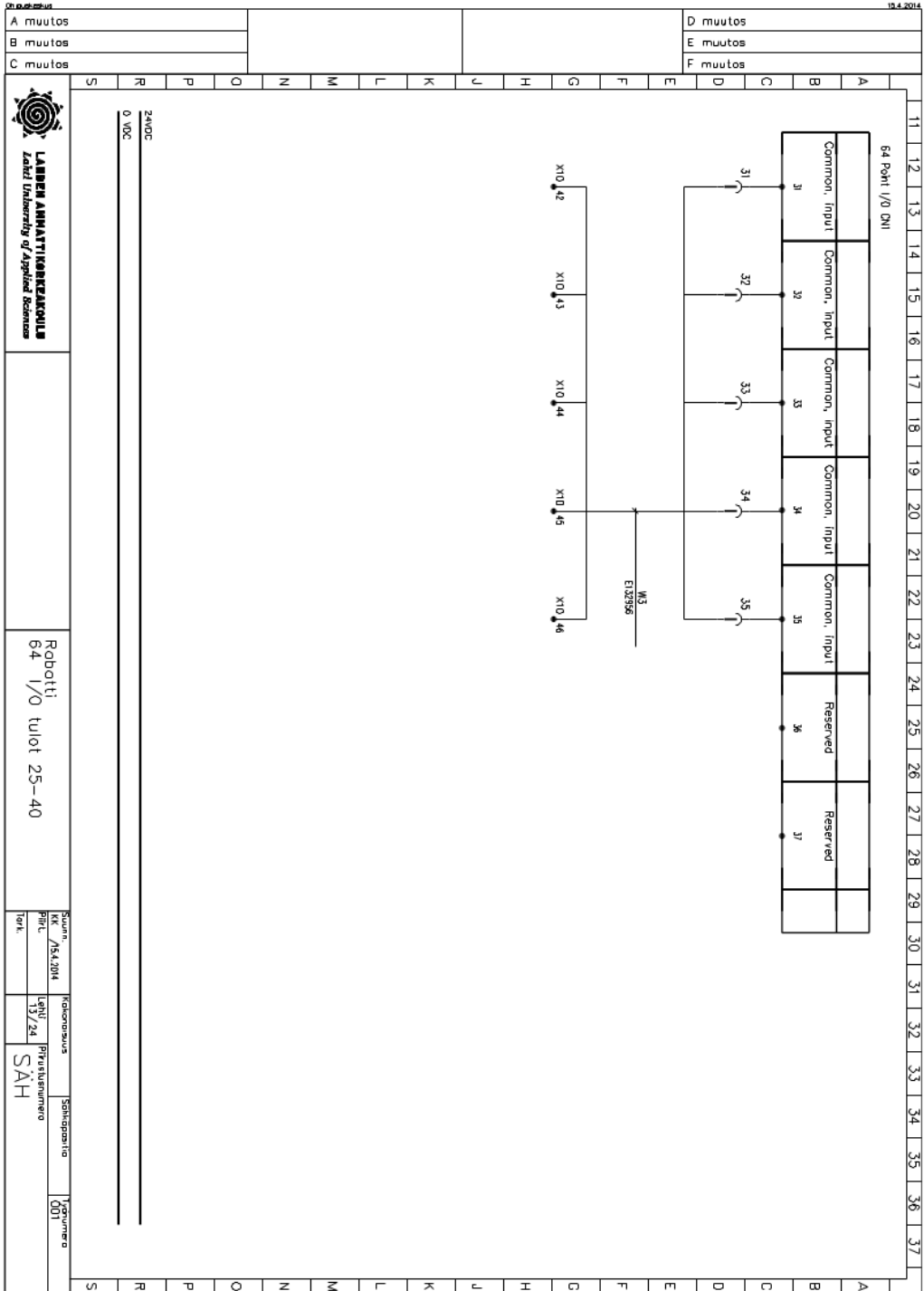
LIITE 1

Kytentäkaavio. 11/24



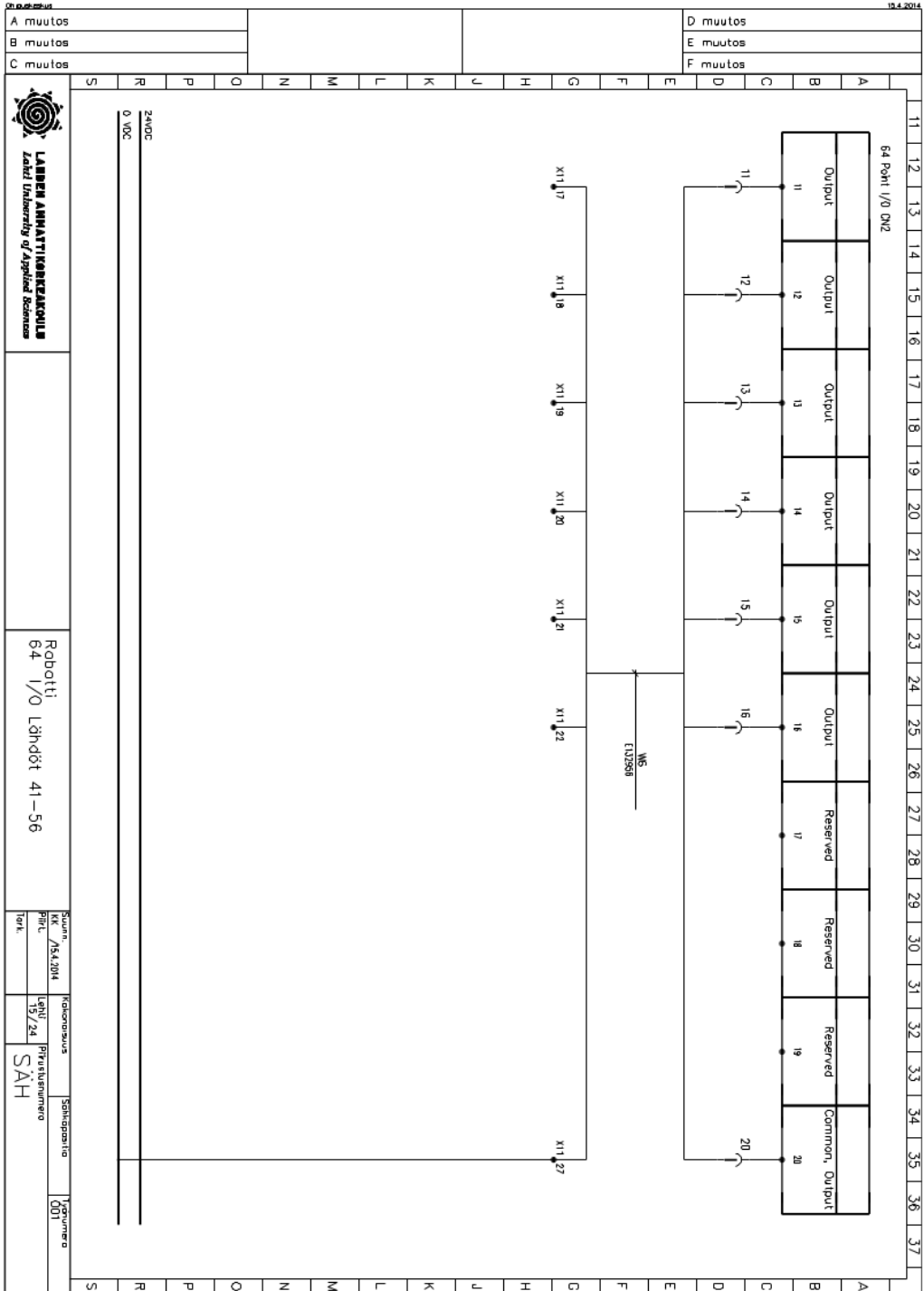
LIITE 1

Kytöntäkaavio. 13/24



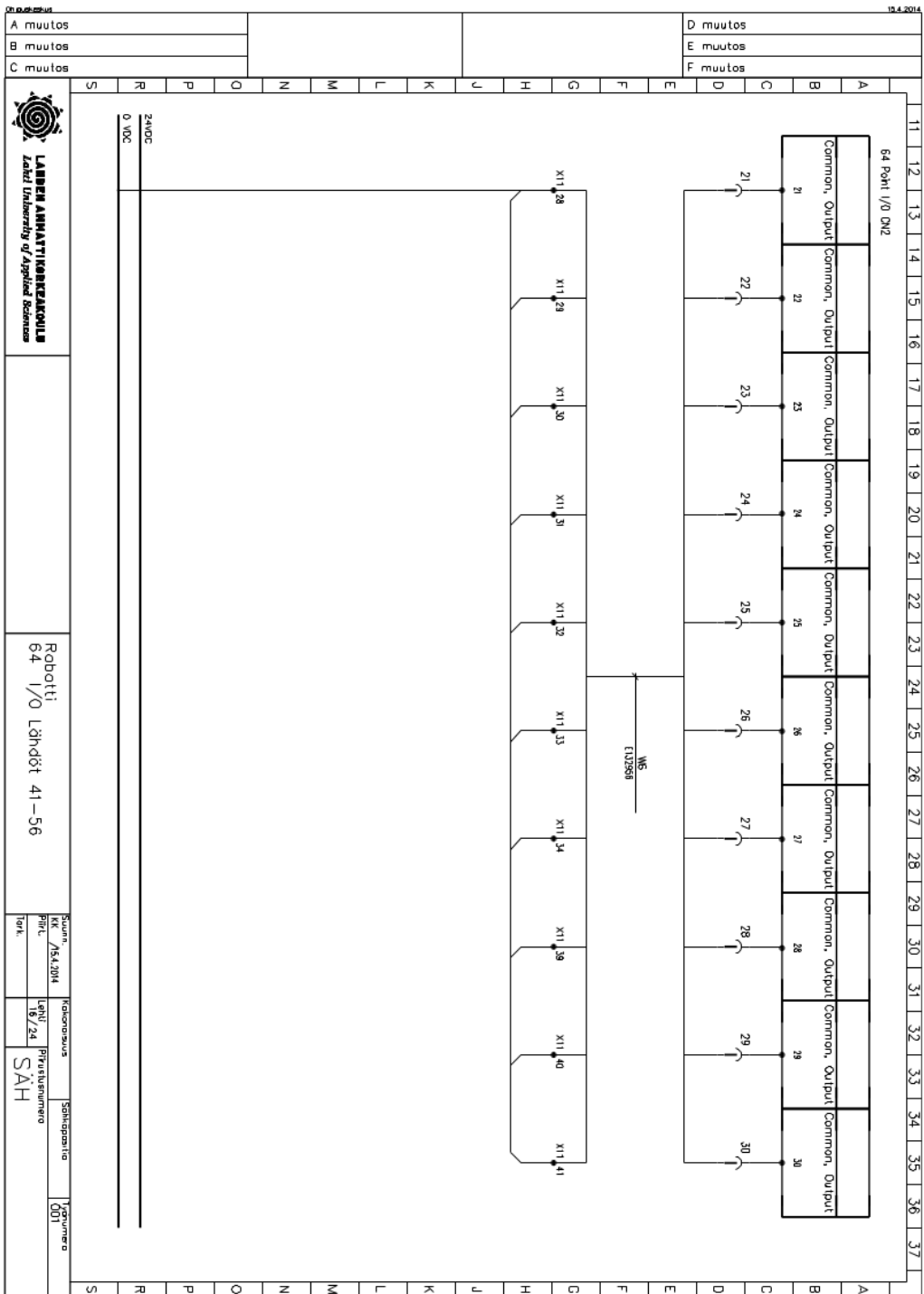
LIITE 1

Kytentäkaavio. 15/24



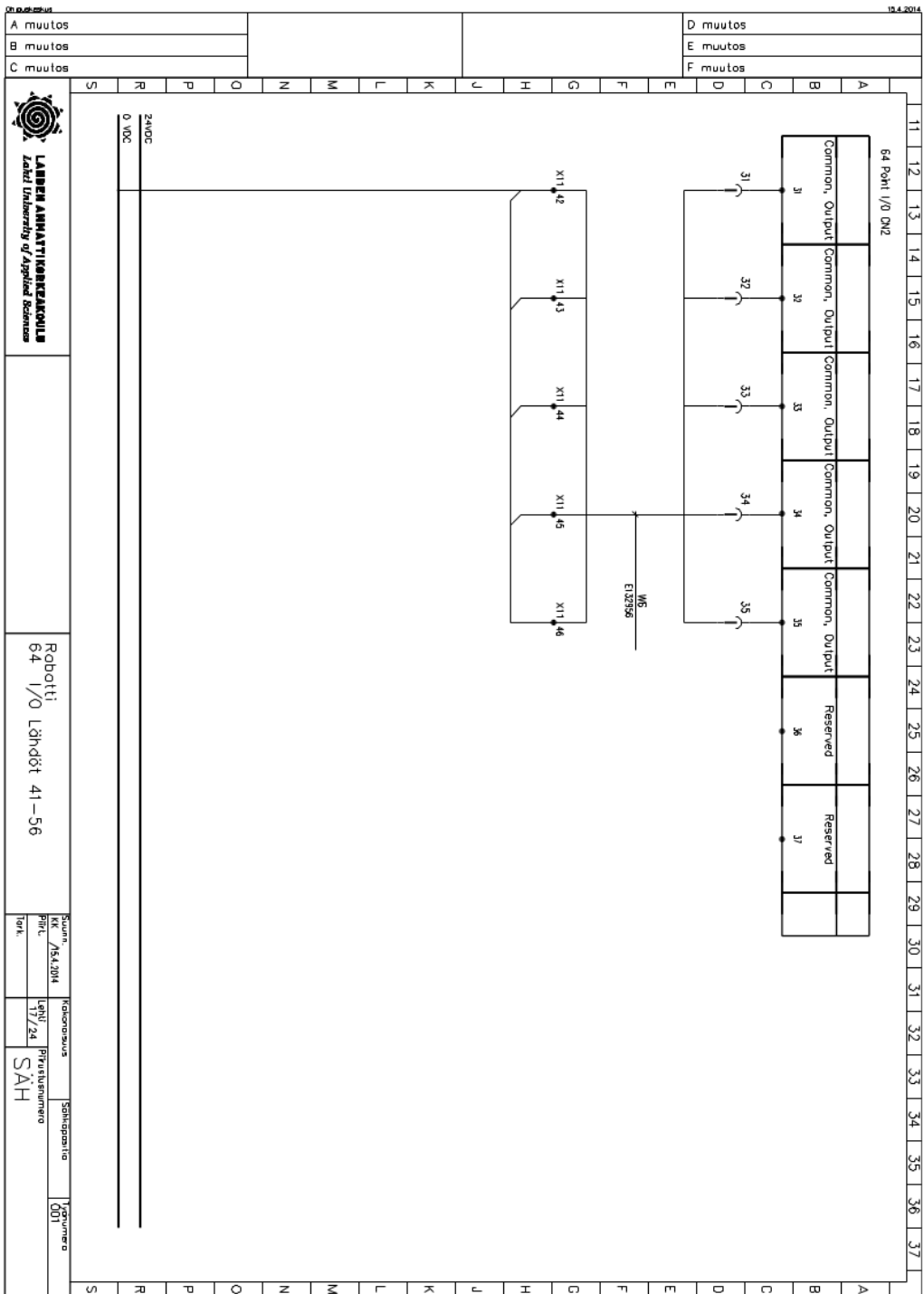
LIITE 1

Kytöntäkaavio. 16/24



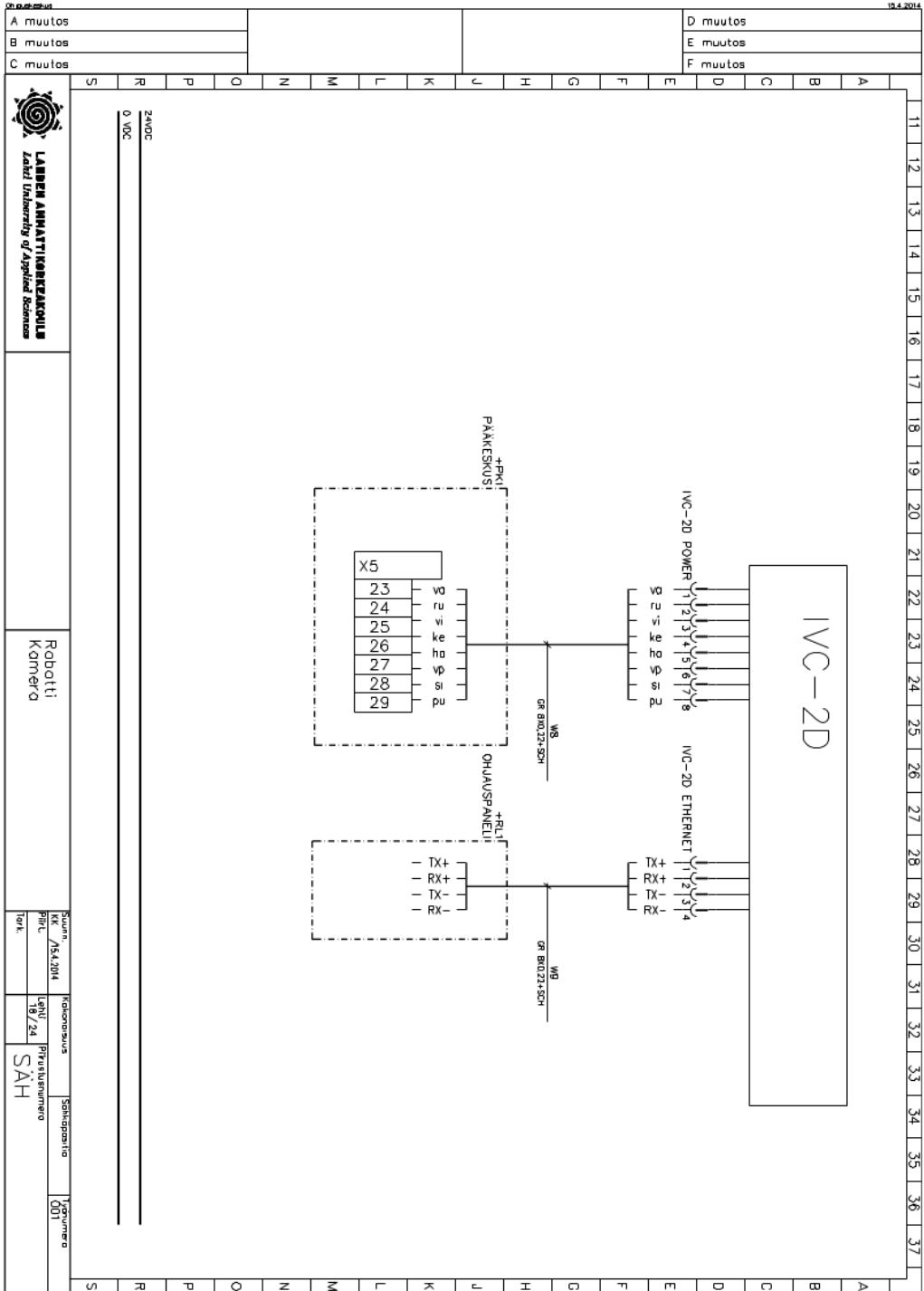
LIITE 1

Kytöntäkaavio. 17/24



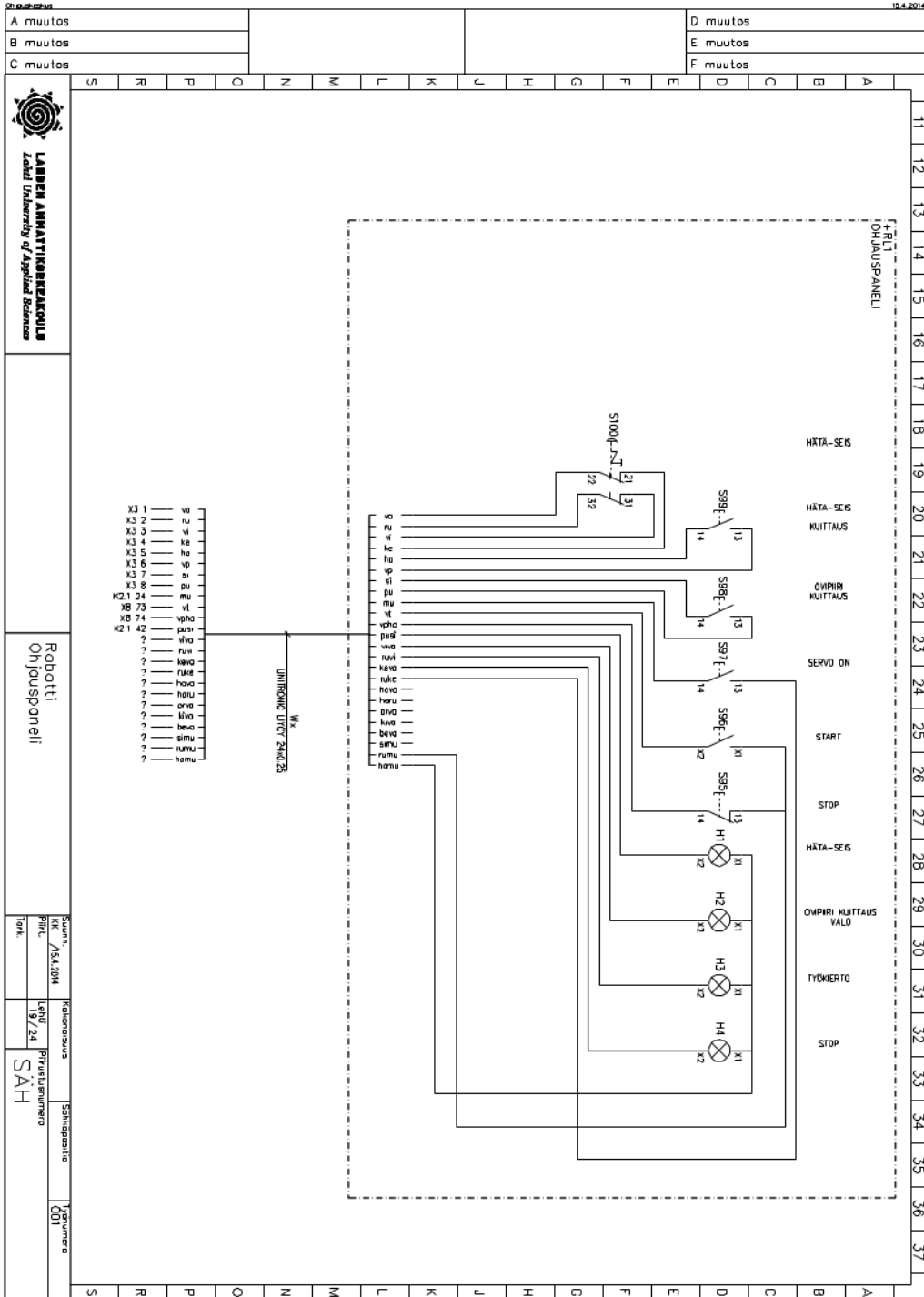
LIITE 1

Kytöntäkaavio. 18/24



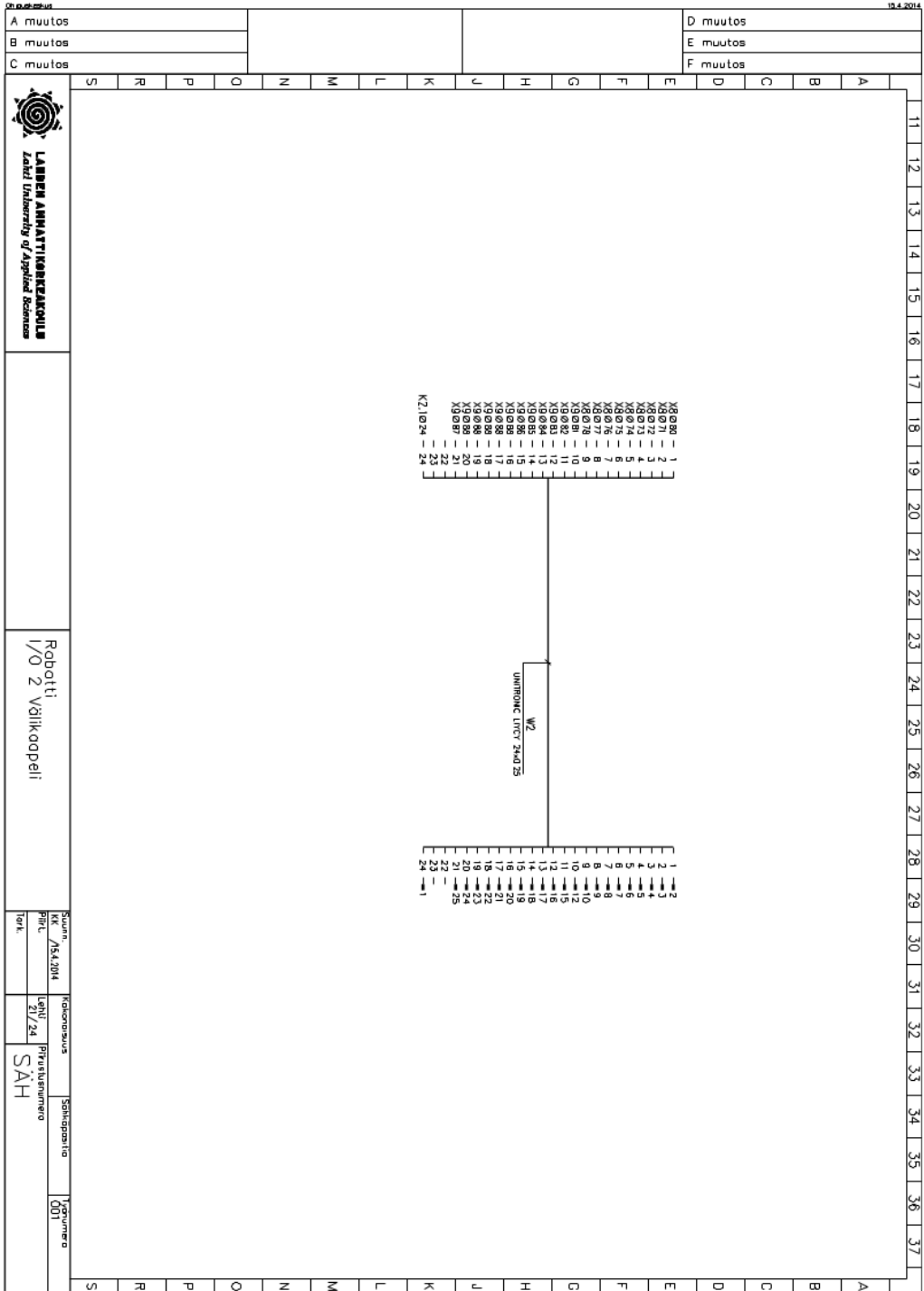
LIITE 1

Kytöntäkaavio. 19/24



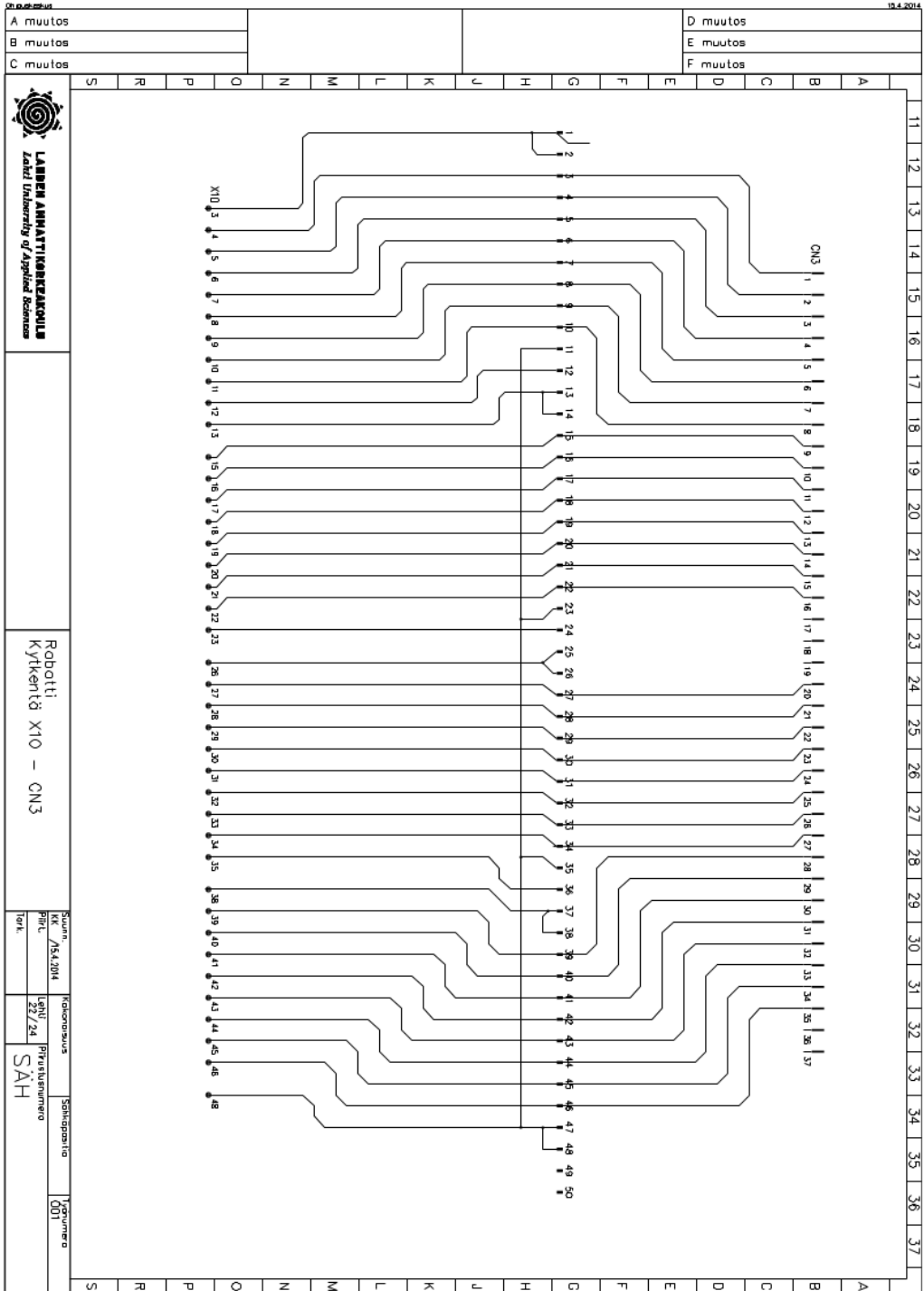
LIITE 1

Kytkentäkaavio. 21/24



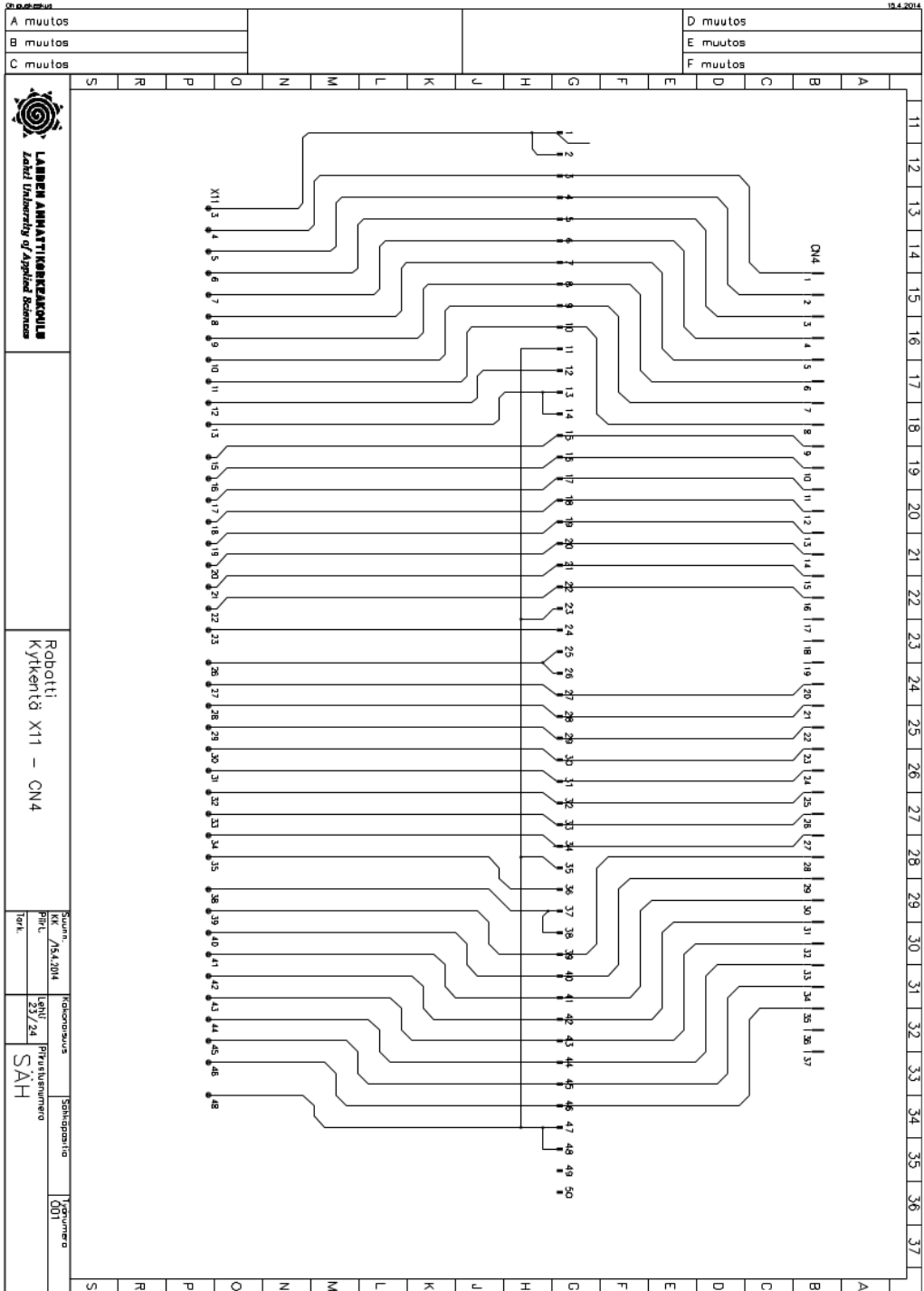
LIITE 1

Kytentäkaavio. 22/24



LIITE 1

Kytentäkaavio. 23/24

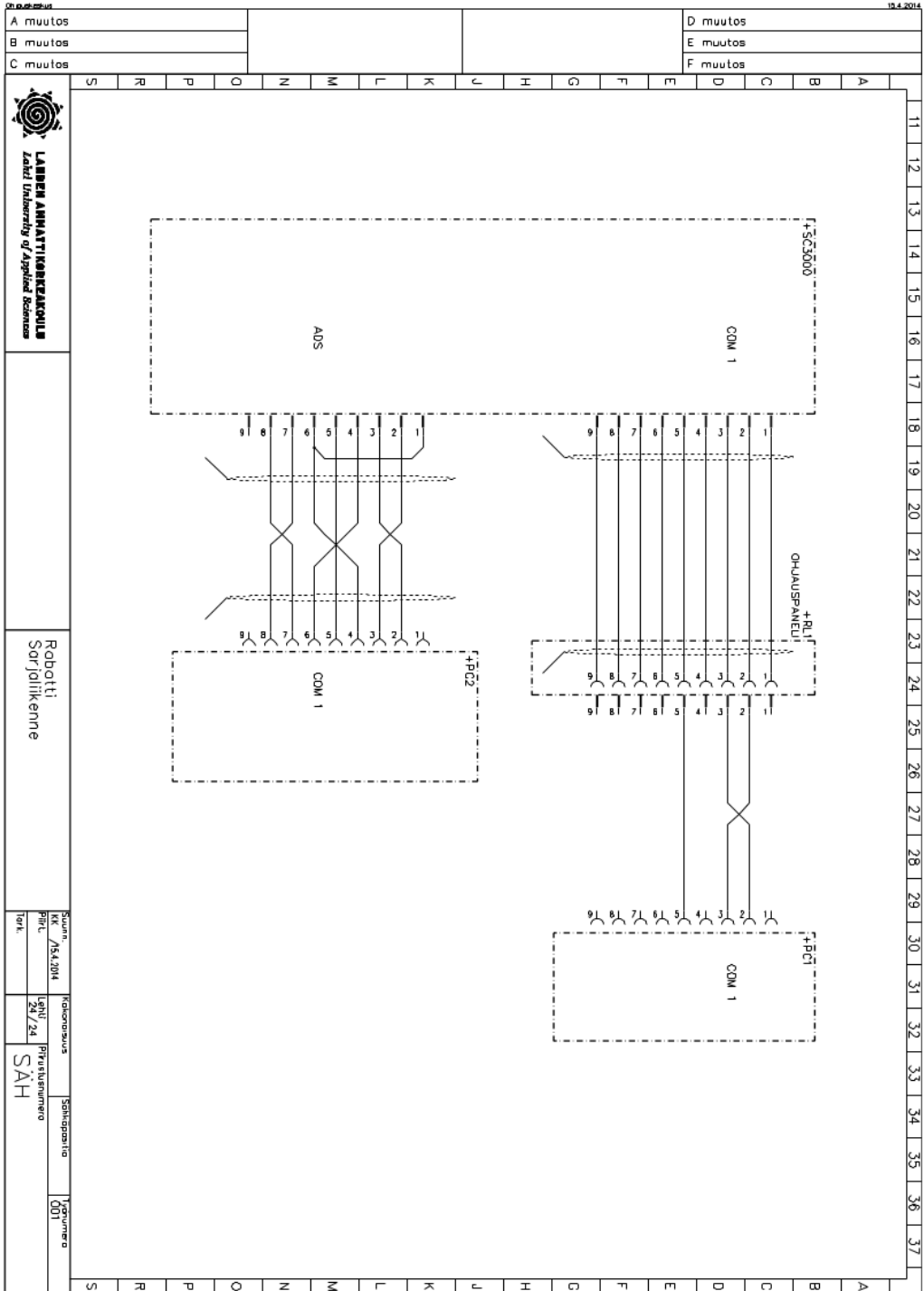


Rodotti:
Kytentä X11 – CN4

Kokonaismäärä	Kokonaismäärä
KK / 754/2014	SAH
Proj. / 24	Proj. / 24
Yhtymänumero	Yhtymänumero
001	001

LIITE 1

Kytöntäkaavio. 24/24



Liite 3 Paletoinnin ohjelmalistaus 1/2

LIITE 2

```
////////////////////////////////////
//- Robotti.ssl
//- Paikoitetaan robotti kameralta tulleen paikkatiedon perusteella
//- Muodostetaan 6x2 paletti
//- Tehdään paletinvaihto
////////////////////////////////////

POSITION KOTI[1]={<-165.163,311.923,50.000,0.00> };// Annetaan paikka joka ei ole kameras edessä

// Tässä paletissa 6 saraketta ja 2 riviä
// Sarakeväli 30.0mm
// Riviväli 42.0mm

Real PaIX=-330.467;// Paletin X arvo eka ruutu
Real PaIY=488.432;// Paletin Y arvo eka ruutu
Real PaIZ=150.0;// Luovutuskorkeus paletissa
Real PaIR=0;// Luovutuskulma paletissa

Real PaISV=30.0;//Paletin sarakeväli
Real PaIRV=42.0;//Paletin riviväli
Real PaISN=0;//Paletin sarake numero
Real PaIRN=0;//Paletin rivinnumero

Real Siirtyma;

real posX2;
real posY2;
real posZ2;
real posR2;//Kääntökulma asteina
real Nop2;//Nopeus Prosentteina

int X;

PROG Vaihto()
    //////////////////////////////////
    //Paletinvaihtoriitin tähän
    //////////////////////////////////
    DELAY(0.5);
    END //Paletinvaihdon loppu

PROG main()

//Resetoinnit ensimmäisellä kierroksella
CVAROUT(1,0);
CVAROUT(2,0);
CVAROUT(3,0);
CVAROUT(4,0);
CVAROUT(5,0);

PaISN=1;
PaIRN=0;
Siirtyma=0;

SPEED(10); //Asetetaan nopeudeksi 10% maksimista

QMOVE(KOTI[1]); //Ajetaan kotiin pois kameras edestä

alku: // Ohjelmarutiinin alku

//Siirretään Taskista11 tulleet tiedot paikallisiin muuttujiin
posX2=CVARIN(1);
posY2=CVARIN(2);
posZ2=CVARIN(3);
posR2=CVARIN(4);
Nop2=CVARIN(5);
```

Liite 3 Paletoinnin ohjelmalistaus 2/2

```
IF( posX2 !=0.0)// Tutkitaan onko tullut uutta paikkatietoa
{
  SPEED(Nop2);
  QMOVE(< , ,50, >); //Nostetaan Z-akseli ylös
  QMOVE(<-posX2,posY2,posR2>); //Siirytään kameran osoittamaan paikkaan yläpuolelle
  QMOVE(< , ,posZ2,>); //Siirytään hakukorkeuteen

  //Tarttujan toiminnot tähän
  //Tarttujan toiminnot tähän

  QMOVE(< , ,50, >); //Nostetaan Z-akseli ylös
  QMOVE(<PalX,PalY,PalR>); //Siirytään paletin ykkösrudun yläpuolelle
  QMOVE(< , ,PalZ, >); //lasketaan Z-akseli oikeaan korkeuteen
  //Tarttujan toiminnot tähän
  //Tarttujan toiminnot tähän
  DELAY(0.2); //Annetaan tarttujalle aikaa
  QMOVE(< , ,50, >); //Nostetaan Z-akseli ylös
  DELAY(0.1);
  QMOVE(KOTI[1]); //Siirytään kotiin odottamaan uutta paikkatietoa
  CVAROUT(1.0.0); //Nollataan muuttuja myöhempiä tutkimista varten
  PalX= PalX+PalSV; // Kasvatetaan X-siirtymää paletissa seuraavaa kierrosta varten
  PalSN=PalSN+1; //Kasvatetaan sarakelaskuria

  IF ( (PalSN==7) && (PalRN==0) )//Tuliko eka rivi täyteen TULI (PalSN==7||PalRN==0)
  {
    PalSN=0; //Sarake alusta
    PalRN=2; //Tokarivi käyttöön
    PalX=-330.467; //Annetaan paletin "ykkösrudun" paikka
    PalY=PalY-PalRV; //Sirretään Y-suuntaa rivivälin verran
    DELAY(0.5);
  }

  IF ( (PalSN==6) && (PalRN==2) ) // Tutkitaan onko paletti täysi
  {
    PalSN=0; //Alustetaan laskurit uutta palettia varten
    PalRN=0;
    PalX=-330.467; //Annetaan paletin "ykkösrudun" paikka
    PalY=488.432; //Sirretään Y-suuntaa rivivälin verran
    DELAY(0.5);
  }
  cycle alku; // Hypätään Silmukan alkuun
}
ELSE
{
  cycle alku; // Hypätään Silmukan alkuun
}
end //Main loppu
```