



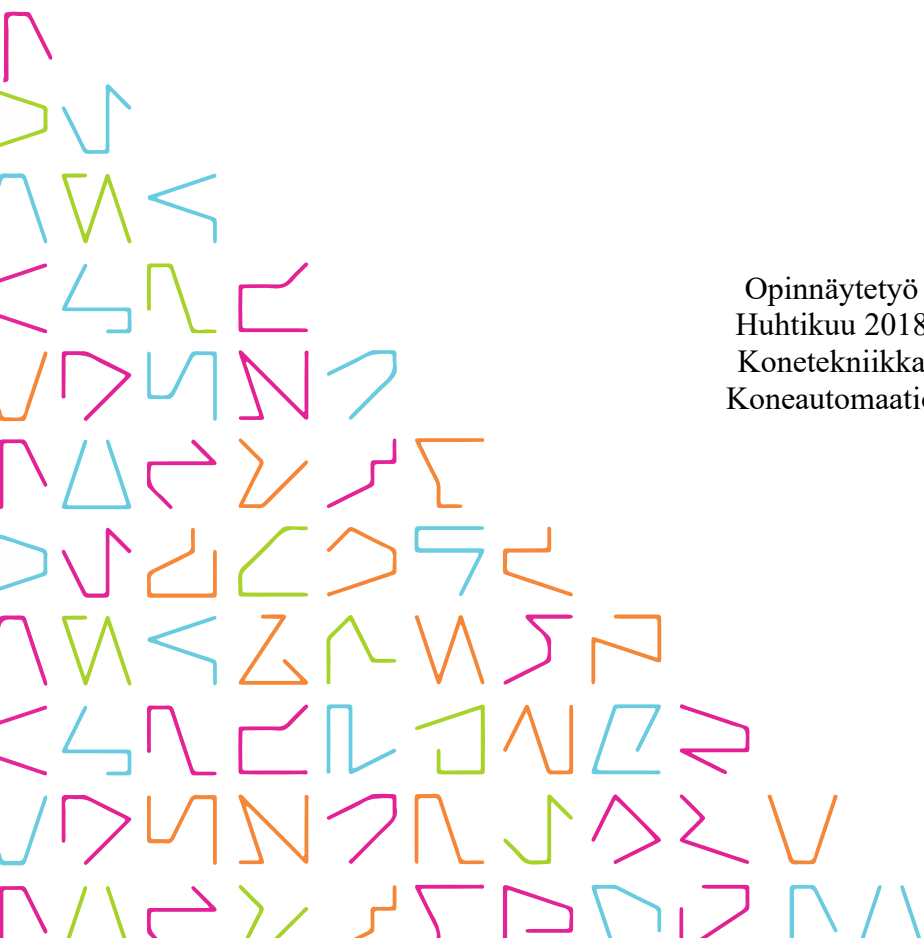
TAMPEREEN
AMMATTIKORKEAKOULU

ROBOTISOLUN OHJELMOINTI

messudemo

Jussi Harsunen

Opinnäytetyö
Huhtikuu 2018
Konetekniikka
Koneautomaatio



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikka
Koneautomaatio

Jussi Harsunen:
Robottisolun ohjelmointi
messudemo

Opinnäytetyö 55 sivua, joista liitteitä 24 sivua
Huhtikuu 2018

Tässä opinnäytetyössä käsitellään robottisolua joka rakennettiin Tampereen Konepaja- ja Nordic Welding Expo messuille 2018. Opinnäytetyö projektin tarkoitus oli ideoida robottisolua ja sen jälkeen toteuttaa se yhteistyössä toimeksiantajan Sasbotics Oy:n kanssa. Projektin jaettiin kahteen osaan josta tämä opinnäytetyö keskittyy robotin ohjelmointiin ja sen vaatimuksiin. Toisena osana projektissa oli PLC:n ohjelmointi (Kaattari, J. 2018 Robotisolun PLC:n ohjelmointi, messudemo).

Robotin ohjelmointi sisälsi monta eri vaihetta ennen itse ohjelmien tekoa. Opinnäytetyössä käydään läpi robotin käyttöönottoon liittyvät asiat ja lisäksi tärkeimpiä osia robotin ohjelmista.

Messuille tarkoitettu robottisolua saatiin valmiiksi asetettujen aikataulujen mukaisesti ja se toimi tarkoitetulla tavalla ilman ongelmia koko messujen ajan.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Mechanical Engineering
Machine Automation

JUSSI HARSUNEN:
Programming of robot cell
convention demo

Bachelor's thesis 55 pages, appendices 24 pages
August 2015

This thesis goes over the robot cell that was built for Nordic Welding Expo. The purpose of this thesis was to create a idea of a robot cell for the convention and ultimately execute the idea in collaboration with the commissioner Sasbotics LLC. The project was divided into two parts; robot programming and PLC programming. This thesis focuses on robot programming and PLC programming is described in Kaattari J. 2018 Programming the PLC of a robot cell, Convention Demo.

Programming of the robot holds various stages before teaching programs. This thesis goes over the commissioning of the robot and main points of robot programs.

The robot cell built for Nordic Welding Expo was finished in the set timelines and it functioned properly through the whole convention.

Key words: convention, robot, sasbotics, plc

SISÄLLYS

1	JOHDANTO.....	7
1.1	Lyhyesti opinnäytetyöstä	7
1.2	Sasbotics Oy	7
2	ROBOTTISOLU	8
2.1	Robottisolun ideointi.....	9
2.2	Suunnittelu	10
2.3	Simulointi.....	10
2.4	Robotti	11
2.5	Logiikka	12
3	MESSUSOLUN RAKENNUS.....	13
3.1	Paletointipöytä	14
3.2	Kuljetin	15
3.3	Sorvi.....	15
3.4	Robotin tarttuja	16
3.5	Solun sähköt.....	17
4	ROBOTTI.....	18
4.1	Käyttöönotto	18
4.1.1	Kalibroinnin tarkistus.....	18
4.1.2	I/O	18
4.1.3	Käyttäjakoordinaatitot ja työkalupisteet.....	19
4.2	Robotin ohjelmointi	22
4.2.1	Ohjelmointi	22
4.2.2	Rekisterit	23
4.2.3	Liikekäskyt.....	24
4.2.4	TP-ohjelmat.....	26
4.2.5	KAREL-ohjelmat	28
5	POHDINTA.....	30
	LÄHTEET.....	31
	LIITTEET	32
	Liite 1. Fanuc M-20iB/25 esite.....	32
	Liite 2. Beckhoff CX5130 esite 1 (3)	33
	Liite 3. Beckhoff CX5130 esite 2 (3)	34
	Liite 4. Beckhoff CX5130 esite 3 (3)	35
	Liite 5. Sähkökuvat 1(20)	36
	Liite 6. Sähkökuvat 2(20)	37
	Liite 7. Sähkökuvat 3(20)	38

Liite 8. Sähkökuvat	4(20)	39
Liite 9. Sähkökuvat	5(20)	40
Liite 10. Sähkökuvat	6(20)	41
Liite 11. Sähkökuvat	7(20)	42
Liite 12. Sähkökuvat	8(20)	43
Liite 13. Sähkökuvat	9(20)	44
Liite 14. Sähkökuvat	10(20)	45
Liite 15. Sähkökuvat	11(20)	46
Liite 16. Sähkökuvat	12(20)	47
Liite 17. Sähkökuvat	13(20)	48
Liite 18. Sähkökuvat	14(20)	49
Liite 19. Sähkökuvat	15(20)	50
Liite 20. Sähkökuvat	16(20)	51
Liite 21. Sähkökuvat	17(20)	52
Liite 22. Sähkökuvat	18(20)	53
Liite 23. Sähkökuvat	19(20)	54
Liite 24. Sähkökuvat	20(20)	55

ERITYISSANASTO

PLC	Programmable Logic Controller, ohjelmoitava logiikka
Roboguide	Fanuc:in simulointiohjelma robotille
UF	User Frame, Käyttäjakoordinaatisto
UT	User Tool, Työkalupiste
Input	Sisääntulo
Output	Ulostulo
Master	Ohjaava laite
Slave	Ohjattava laite
GUI	Graphical User Interface, Graafinen käyttöliittymä
EtherCat	Tiedonsiirrossa käytettävä väylätyyppi
IGS	3D tiedostotyyppi
TP	Teach Pendant, Ohjauskapula
KAREL	Fanuc robottien ohjelmointikieli

1 JOHDANTO

1.1 Lyhyesti opinnäytetyöstä

Tässä opinnäytetyössä käydään läpi messuille tarkoitetun robottisolun eri työvaiheet. Opinnäytetyön alussa käydään läpi projektin ideointi, suunnittelu ja tärkeimmät toimitaan liittyvät komponentit; Fanuc teollisuusrobotti ja Beckhoff PLC. Lopuksi käsitellään teollisuusrobotin ohjelmointia, sen osa-alueita ja se miten robotin yhdistettiin PLC-ohjaukseen.

Sasbotics Oy kustansi kaikki käytetyt materiaalit yhteistyökumppaneiden Ixtur, Omron ja Profican Oy:n kanssa. Lisäksi Sasbotics Oy:n henkilökunta teki osan solun rakennuksesta ja tuotti robottisolun sähköpiirrustukset jotka näkyvillä liitteistä 5 eteenpäin.

1.2 Sasbotics Oy

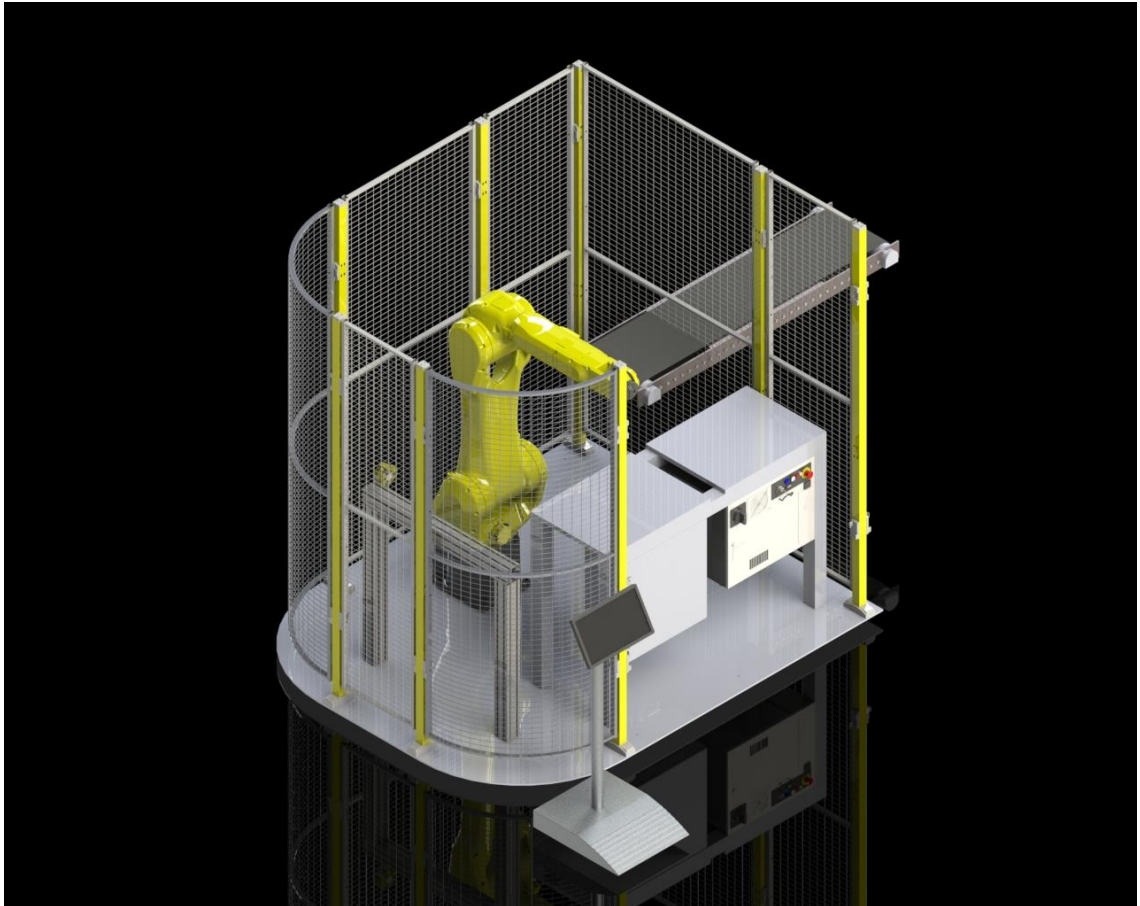
Sasbotics Oy perustettiin vuonna 2013 ja heidän toimialaan kuului kone- ja prosessisuunnittelu ja robottien huolto. Yritys on erikoistunut Fanuc-teollisuusrobotteihin, joille yritys tarjoaa huoltoa ja joita uusissa projekteissa.

Projektipuolella Sasbotics Oy suunnittelee, asentaa ja käyttöönottaa asiakkaiden tarpeisiin tehtyjä robottisoluja. Yrityksen vahvuutena on sen joustavuus projektien suorittamisessa ja loppuasiakkaalla heidän oma graafinen käyttöliittymänsä, joka mahdollistaa robotin joustavan käytön ilman laajaa tietämystä robotiikasta. Graafisen käyttöliittymän ansiosta käyttäjä pystyy muuttamaan robotille lähetettäviä asetuksia, täten mahdollistaen joustavan tuotannon. Lisäksi yritykselle on huolto-osasto, joka huoltaa ja korjaa Fanuc-teollisuusrobotteja.

Sasbotics Oy on ollut jatkuvassa taloudellisessa kasvussa perustamisen jälkeen joka tänä päivänä näkyy kasvaneessa työntekijämäärässä. Opinnäytetyön aloitus hetkellä yrityksessä työskentelee kahdeksan henkilöä.

2 ROBOTTISOLU

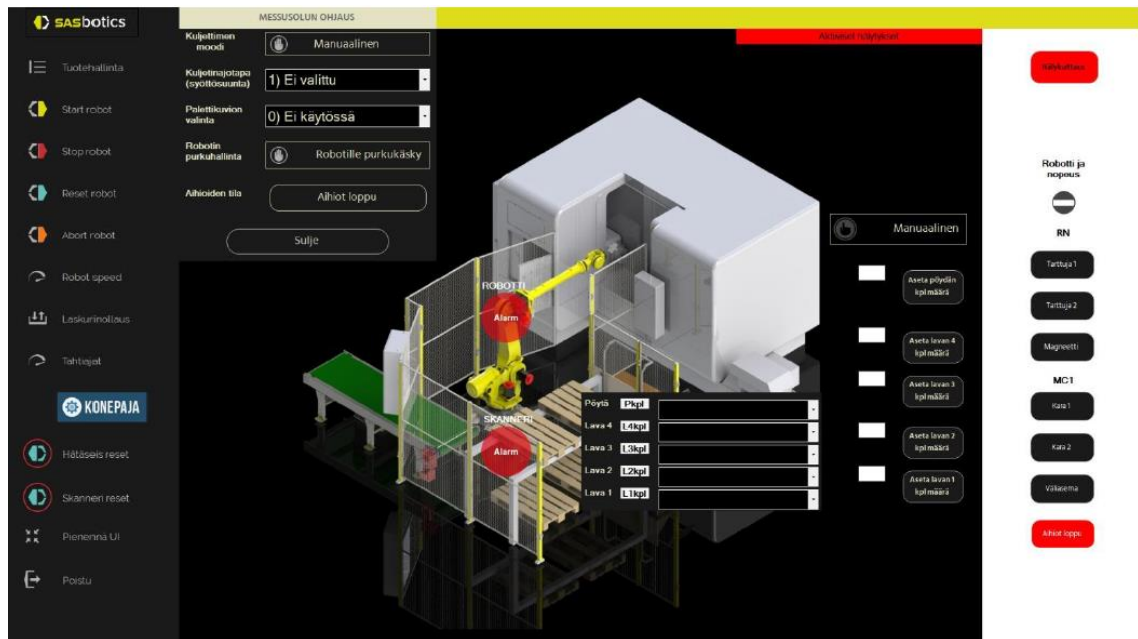
Messuille tehdyssä robottisolussa kuvataan modernia konepalvelusolua, jota ohjataan käyttöliittymästä. Kuvassa 1 on nähtävillä messuille tehdyn robottisolun 3D-malli.



KUVA 1. Messusolu

Messudemon työkierrossa kappaleet syötetään sisään käyttäen kuljetinta solun kyljessä. Robotti etsii ja poimii kappaleen kuljettimelta ja tarjoaa sen sorvia demonstroivaan laitteeseen. Tässä demo sorvissa kappale siirretään pakasta toiseen käyttäen lineaarijohdetta ja paineilmasylinteriä. Siirron jälkeen robotti lataa uuden kappaleen sorvinpakkaan, jolloin sorvissa on kaksi kappaletta samanaikaisesti. Sorvin panostuksen jälkeen robotti purkaa kappaleet toiselta pakalta kuvioon pöydälle. Kun kaikki kappaleet on ajettu pöydälle robotti palauttaa kappaleet parijonoon takaisin kuljettimelle, jonka jälkeen kierto alkaa alusta.

Messudemon hallinta tapahtui Sasbotics Oy:n tekemästä graafisesta käyttöliittymästä (KUVA 2).



KUVA 2. GUI

Robotin hallinta käyttöliittymästä tapahtuu reseptiikan avulla. Tuotteelle luodaan asetus-kortti, josta löytyy robotille oleelliset mitat käsiteltävästä kappaleesta ja tieto siitä miten tuotteeseen tartutaan. Käyttöliittymä mahdollistaa nopeat tuotevaihdot soluun ilman robotin uudelleen ohjelmoimista. Messuilla kävijät pystyivät muokkaamaan käyttöliittymältä robotin toimintaa rajoitetusti turvallisuuden takia.

2.1 Robottisolun ideointi

Robottisolun ideointi aloitettiin puhtaalta pöydältä ja tarkoituksena oli saada aikaiseksi sellainen idea, mikä nostaisi esiin mahdollisimman hyvin graafisen käyttöliittymän kuin myös yrityksen osaamisen. Solun ideoinnin alkuvaiheessa inspiraatiota haettiin käymällä vierailuilla yrityksissä johon Sasbotics Oy oli toimittanut robotteja tai kokonaisia robotisoluja. Vierailujen jälkeen jatkettiin tekemällä alustavia ehdotuksia yritykselle mitä messuilla voitaisiin tehdä. Huomioon tuli ottaa kustannuspuoli, näytettävyys ja turvallisuus.

Alustavia ideoita oli kokoonpanoa tekevästä roboteista. Ideat hylättiin nopeasti koska Sasbotics Oy oli erikoistunut tuottamaan räätälöityjä robottisoluja konepajayrityksille. Tämä

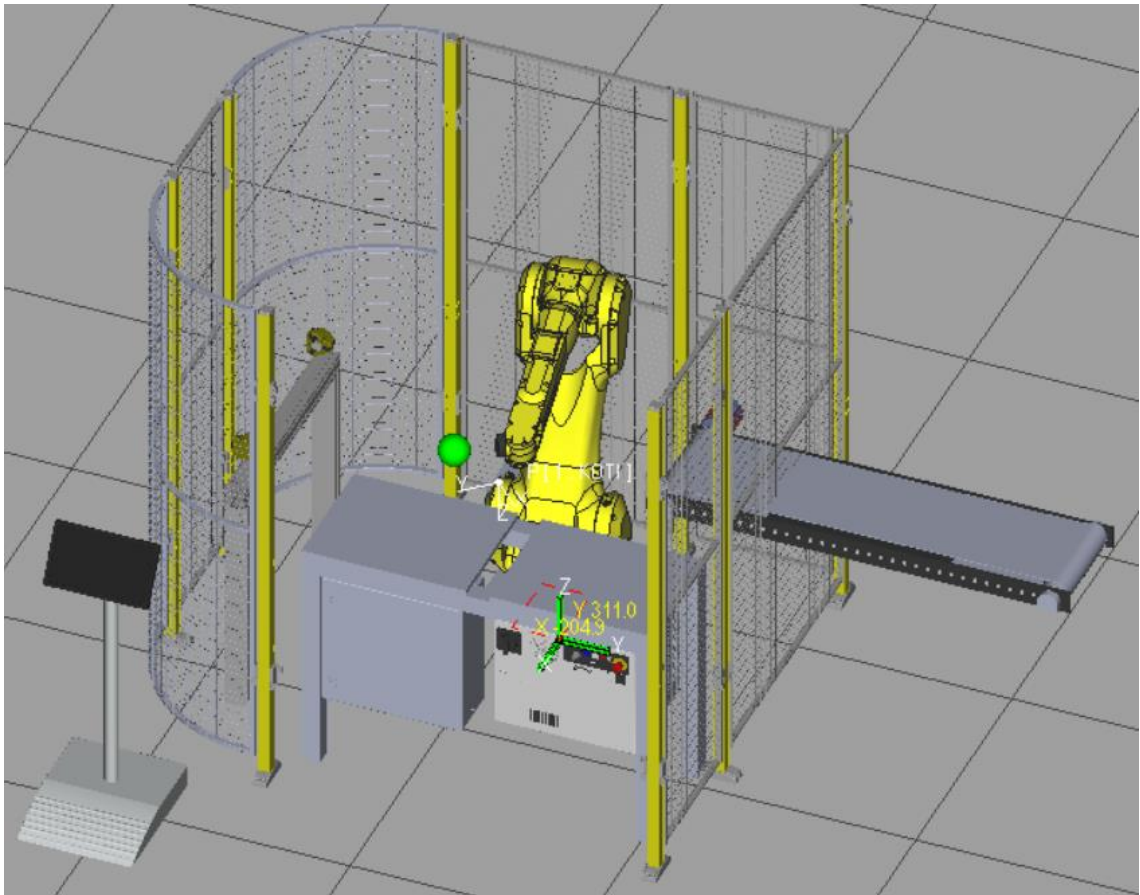
erikoisosaaminen palvelusoluista päädyttiin tuomaan esille messuilla. Messusolu demonstroi robotin toimintaa konepalvelusolussa.

2.2 Suunnittelu

Kun idea oli valmis ja hyväksytty aloitettiin solun suunnittelu. Messuilla oli käytössä $6,25m^2$ alue robottisolulle joten oikean sorvin paikalle tuominen ei ollut vaihtoehto pelkästään tilan puutteen takia. Messusoluun päätettiin täten valmistaa 2-karaista sorvia esittävä laite. Muita soluun sijoitettavia komponentteja oli kuljetin jolla kappaleet tuotiin soluun sekä purettiin solusta ulos. Kohdistuspöytä, jonka avulla kappaleet kohdistettiin kuljettimelta poiminnan jälkeen ja paletointipöytä johon kappaleet asetettiin ennalta määrättyyn kuvioon näennäisen sorvauksen jälkeen. Paletointipöydän alle sijoitettiin myös robotin ohjauskaappi ja sähkökaappi.

2.3 Simulointi

Ennen varsinaisten osien hankintaa robottisolu mallinnettiin käyttäen SolidWorks -ohjelmaa. Yrityksellä oli valmiit osakomponentit mallinnettuna kirjastossa, joten solun 3d-mallin rakennus simulointia varten tapahtui nopeasti. Kun robottisolun malli oli valmis käännettiin se IGES tiedostomuotoon, jota Fanucin ROBOGUIDE -ohjelma tukee. ROBOGUIDE -ohjelmassa pystyttiin realistisesti simuloimaan, kuinka suuren robotin soluun tarvittaisiin. Kuvassa 3 on kuvakaappaus ROBOGUIDE -ohjelmasta mihin on lisätty solun 3D-malli. Solun simulointi on tärkeä osa kun tarkastellaan riittääkö robotin ulottumat joka puolella sen työaluetta.



KUVA 3. ROBOGUIDE

Simulointia ajettiin kahdella eri koko luokan robotilla. Fanuc:n teollisuusrobottimalli M10-iA/10M oli ensimmäinen mitä hahmoteltiin solun sisään. M10-malli olisi ollut työskentely alueelta riittävän kokoinen soluun jos sen olisi asettanut korkealle jalustalle pöydän taakse. Simulointia päätettiin kuitenkin ajaa vielä M-20iB/25-mallilla jonka todettiin olevan optimi tämän kokoiseen soluun jolloin ylimääräinen korkea jalusta voitiin jättää hankinnoista pois.

2.4 Robotti

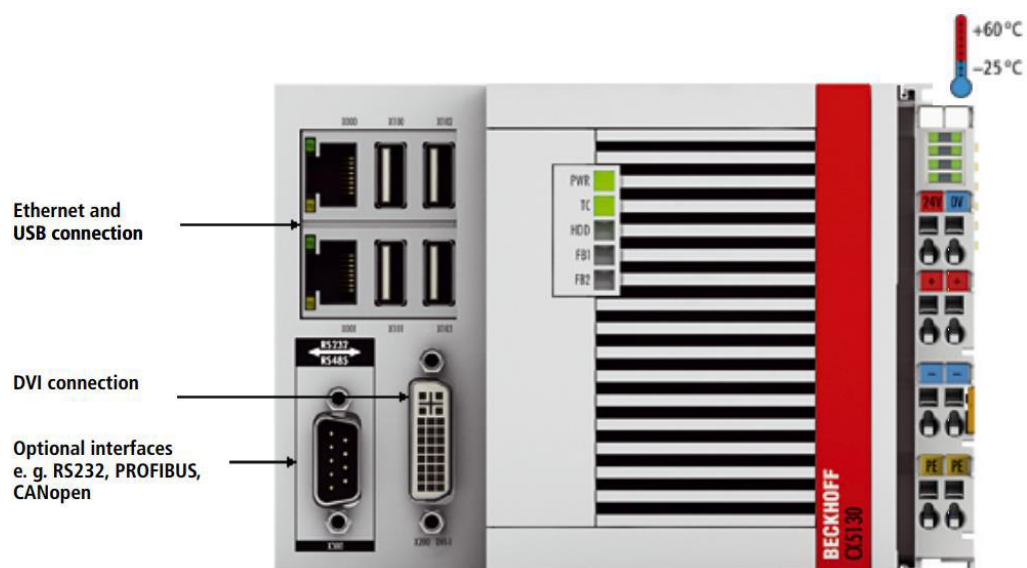
Fanuc M20iB/25 – teollisuusrobotti oli oikean kokoinen soluun, sen työskentelyalue oli tarpeeksi suuri jotta robotti pystyi liikkumaan ympäri solua. Lisäksi kyseinen malli oli saatavilla maahantuojalta heti. M-20iB/25 robotilla on 1853 mm ulottuvuus ja kappaleenkäsittelykykyä 25 kg asti. Lisäksi robotissa on nivelten sisäinen kaapelointi aina käsivarren neljännelle akselille asti ja IP67 suojausluokka jokaisen nivelen kohdalla (Liite 1.Fanuc 2017).



KUVA 4. M-20iB/25 (<https://www.fanuc.eu/fi/en/robots/robot-filter-page/m-20-series/m-20ib-25>)

2.5 Logiikka

Messusolu tehtiin käyttäen Beckhoff CX5130 logiikkaa (KUVA 5), johon oli alustettu Windows 10. Beckhoff on Sasbotics Oy:n pääsääntöisesti käyttämä logiikka valmistaja joten se oli luontainen valinta myös messusolun logiikaksi.



KUVA 5. Beckhoff CX5130 PLC (<https://www.beckhoff.fi/>)

3 MESSUSOLUN RAKENNUS

Kun robotti oli valittu ja solu 3D mallinnettu tarkoilla mitoilla, tiedettiin mitä osia soluun tarvittiin jolloin vältyttiin turhien komponenttien hankinnalta. Messualueella oli tiukka sääntö että mitään ei saa kiinnittää suoraan lattiaan. Tästä syystä messusolu rakennettiin suoraan paksun metallilevyn päälle. Metallilevyn alle lisättiin metallipalkit lisäämään jäykkyyttä ja helpottamaan solun siirtelyä.

Kun alusta oli valmis, nostettiin robotti paikalleen. Koska kyseessä oli työstökeskusta palvelevan robotin demo, tuli turva-alue rakentaa niin kuin se olisi tuotannossa oleva robotisolu. Robotin työalue rajattiin käyttäen Axelent-yrityksen valmistamia tolppia ja turvaverkkoja. Robotin etupuolelle verkkoon jätettiin kulkuväylä jota kuitenkin valvottiin jatkuvasti käyttäen Omron OS32C -skanneria(KUVA 6).



KUVA 6. Omron OS32C (<https://industrial.omron.fi/fi/products/os32c>)

Kohdistuspöytä (KUVA 7) asennettiin kuljettimen ja paletointipöydän väliseen tilaan. Kohdistuspöydässä robotti työntää kappaleen pöydänkulmassa olevaa kulmarautaa vasten, jolloin varmistetaan kappaleen sijainti joka kerta samaan pisteeseen.



KUVA 7. Kohdistuspöytä

3.1 Paletointipöytä

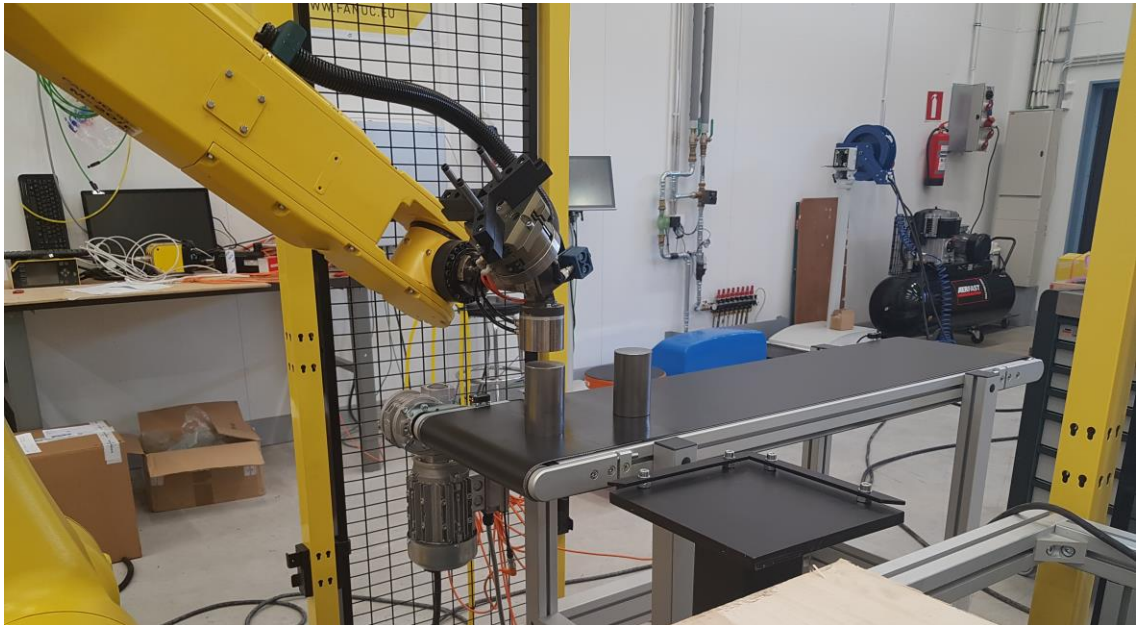
Kulkuväylän eteen asetettiin alumiiniprofiilista valmistettu paletointipöytä, jonka sisään asennettiin robotin ohjainkaappi ja solun sähkökaappi. Kuvassa 8 näkyy valmis paletointipöytä.



KUVA 8. Paletointipöytä

3.2 Kuljetin

Kuvassa 9 on nähtävillä kuljetin ja sen asemointiin liittyvää testiajoa. Robotin tuli pystyä kulkemaan kuljettimen etureunaa pitkin etsien kappaleita ja tämän jälkeen poimia kyseiset kappaleet magneetilla. Kuljettimen lopullinen asemointi päätettiin näistä testiajoista kerätyillä tiedoilla.



KUVA 9. Robotti kuljettimella

3.3 Sorvi

Soluun rakennettiin myös sorvia imitoiva laite, joka on esillä kuvassa 10. Laitteen runko rakennettiin alumiiniprofiiliputkesta jota oli helppo työstää tarvittuihin mittoihin. Sorvinpakoiksi asetettiin Schunk:n PZN Plus 100-1 kolmisormitarttajat.



KUVA 10. Sorvi

Sorvinpakat asennettiin lineaarijohteen päälle, tämän avulla mahdollistettiin vasemmanpuoleisen karan liikuttaminen suorassa linjassa. Liike karalla tuotettiin paineilmasylinterillä ja sorvin toimintoja ohjattiin PLC:ltä.

3.4 Robotin tarttuja

Robotille tehtiin kaksoistarttuja työkalu (KUVA 11). Tarttuvat asetettiin robotin laippaan nähden kohtisuoraan ja toisiinsa nähden pieneen kulmaan jolloin ultraäänianturi kyettiin helposti asentamaan tarttujien väliin. Tarttujiksi valitui yhteistyökumppani Ixtur:n MAP-40 magneettitarttuja ja Schunk PZN Plus 100-1 kolmisormitarttuja.



KUVA 11. Kaksoistarttuja

3.5 Solun sähköt

Sasbotics Oy suunnitteli solun sähköt ja toteutti sähkökeskuksen kokoonpanon kytkentöineen. Sähkökeskuksen kuvat löytyvät liitteistä 5-20 ja liitteessä 20 on kuva sähkökeskuksen komponenttien asetteluista keskuksen sisällä.

4 ROBOTTI

4.1 Käyttöönotto

Tässä osiossa käydään läpi työvaiheita mitä tarvitaan robotin valmisteluun PLC paritusta varten. Robotti toimitetaan Fanuc:n toimesta niillä ohjelmisto-optioilla mitä tilausvaiheessa on määritelty. Messudemo-robottiin tilattiin lisäksi muutama ohjelmisto-optio jälkikäteen.

4.1.1 Kalibroinnin tarkistus

Robotin käyttöönotossa tulee varmistaa sen kalibrointi. Fanuc:n roboteissa on rungon sisään sijoitetut paristot joiden tarkoitus on ylläpitää jokaisen akselin pulssikiekkujen asentotieto muistissa sähkökatkoksen tai kuljetuksen aikana. Robotin kalibrointi suoritetaan Fanuc:n tehtaalla, mutta johtuen pitkästä kuljetus- ja varastointi ajasta saattaa robotin muisti tyhjentyä johtuen paristojen jännitteen tippumisesta. Pulssikiekkujen asentotiedon tarkastus suoritetaan ajamalla jokainen akseli ohjelmallisesti nolla-asentoon. Kun robotti on nolla-asemassa tulisi tehtaalla asennettujen akseli merkintöjen osua toisiinsa.

Robotin jokaisesta akselistä löytyy painettu merkintä tai mittatarra joiden avulla on helppo määrittää silmämääräisesti onko pulssikiekkujen paikkatieto tallessa vai onko muisti jo tyhjentynyt.

4.1.2 I/O

Robotin rajapinta PLC:lle tehdään jakamalla digitaalisesta I/O -väylästä osa robotin ja PLC:n välillä. Fanuc tarjoaa monia väylämahdollisuuksia roboteilleen ja tähän sovellukseen valittiin Ethercat-väylä.

Väylän valinta aiheutti lisäohjelmisto vaatimuksen, joka tuli tilata Fanuc:lta. Ohjelmisto tilataan aina suoraan Fanuc:lta ja ohjelmiston lisäystä varten Fanuc tarvitsee robotin varmuuskopion. Fanuc lähettää ohjelmistopäivityksen pakettina, joka ladataan takaisin robottiin jolloin uusi ohjelmisto-optio saadaan käyttöön.

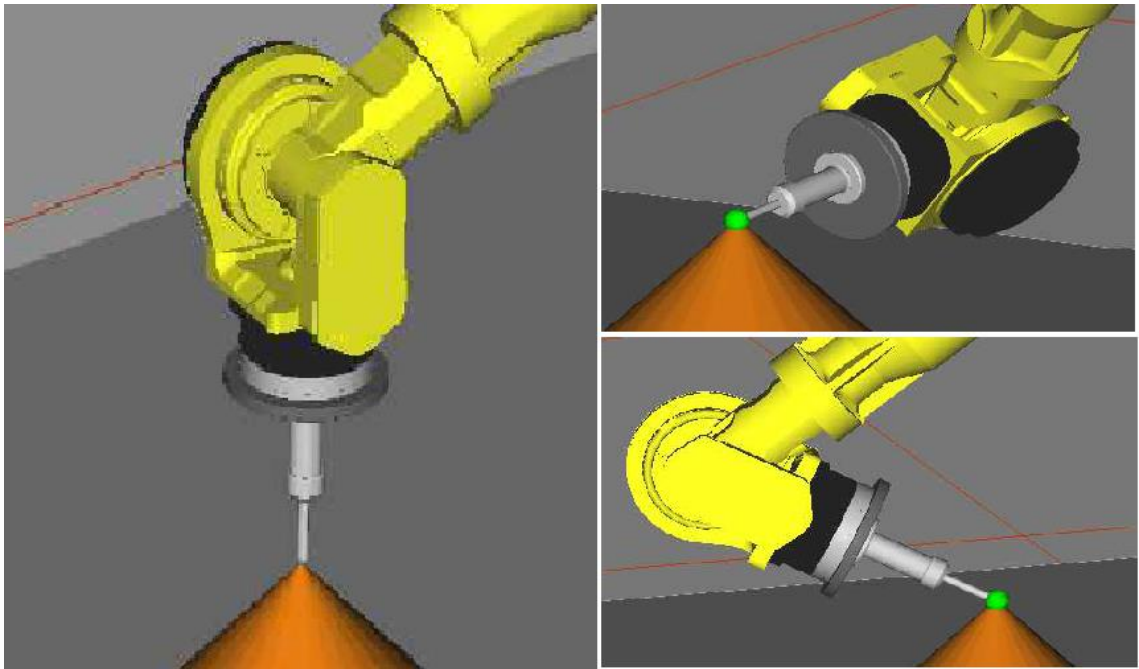
Fanucissa I/O-puoli jaetaan asettamalla tulot ja lähdöt haluttuun rack-osoitteeseen, jolloin robotin kontrolleri osaa ohjata signaalit tiettyä väylää pitkin. Kuvassa 12 on näkyvissä tämän robotin digitaalisen lähtö- ja tulopuolen asetukset. Lähtöpuolella bitit 1-400 on asetettu Ethercat Slave väylään ja bitit 401-465 robotin sisäiseen väylään.

I/O Digital Out						I/O Digital In					
#	RANGE	RACK	SLOT	START	STAT.	#	RANGE	RACK	SLOT	START	STAT.
1	DO[1- 400]	106	1	1	ACTIV	1	DI[1- 400]	106	1	1	ACTIV
2	DO[401- 465]	33	2	1201	ACTIV	2	DI[401- 465]	33	2	401	ACTIV
3	DO[466- 512]	0	0	0	UNASG	3	DI[466- 512]	0	0	0	UNASG

KUVA 12. I/O asetukset

4.1.3 Käyttäjäkoordinaatit ja työkalupisteet

Robottia ohjelmoidessa on tärkeää luoda työkalut sen mukaisesti. Robotille on mahdollista luoda useita UF eli käyttäjäkoordinaatistoja ja UT-pisteitä eli työkalun keskipisteitä. Robotin työkalupisteet opetettiin käyttäen Three Point-metodia, jolloin robotti ajetaan kolmesta erikulmasta samaan pisteeseen tallentaen jokaisen pisteen koordinaatit robotilla jolloin robotti laskee osoitetuista pisteistä työkalupisteeseen. Kuvassa 11 on esimerkki orientaatiot työkalupisteeseen opetukseen.



KUVA 13. Esimerkki orientaatiot

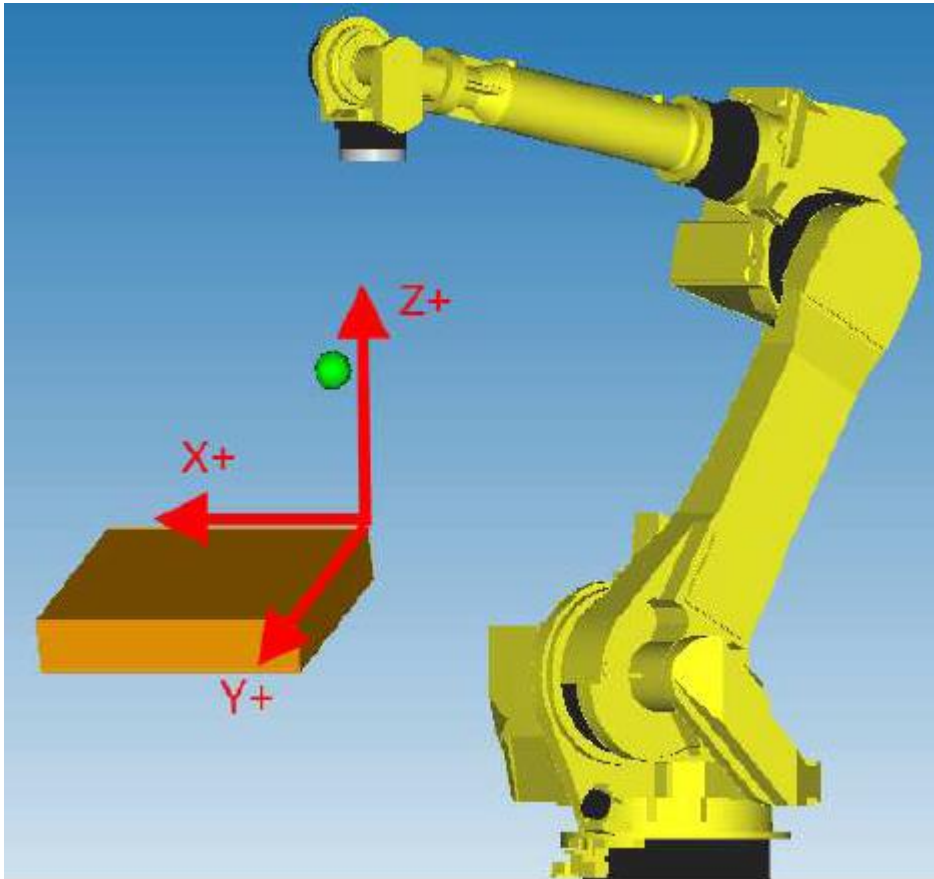
UT-pisteet luotiin jokaisen työkalun keskipisteeseen; magneettitarttujalle, kolmisormitarttujalle ja ultraäänianturille. Kuvassa 14 on näkyvillä työkalupisteiden opetusvalikko.

SETUP Frames				
Tool Frame	/ Direct Entry			10/10
X	Y	Z	Comment	
1	54.0	55.6	161.0	[UA]
2	0.0	0.0	0.0	[]
3	0.0	0.0	0.0	[]
4	0.0	0.0	0.0	[]
5	0.0	0.0	0.0	[]
6	0.0	0.0	0.0	[]
7	-139.1	65.3	120.9	[KOLMI_REAL]
8	57.0	-102.0	130.9	[MAGNET]
9	0.0	0.0	0.0	[]
10	0.0	0.0	0.0	[]

KUVA 14. UT-valikko

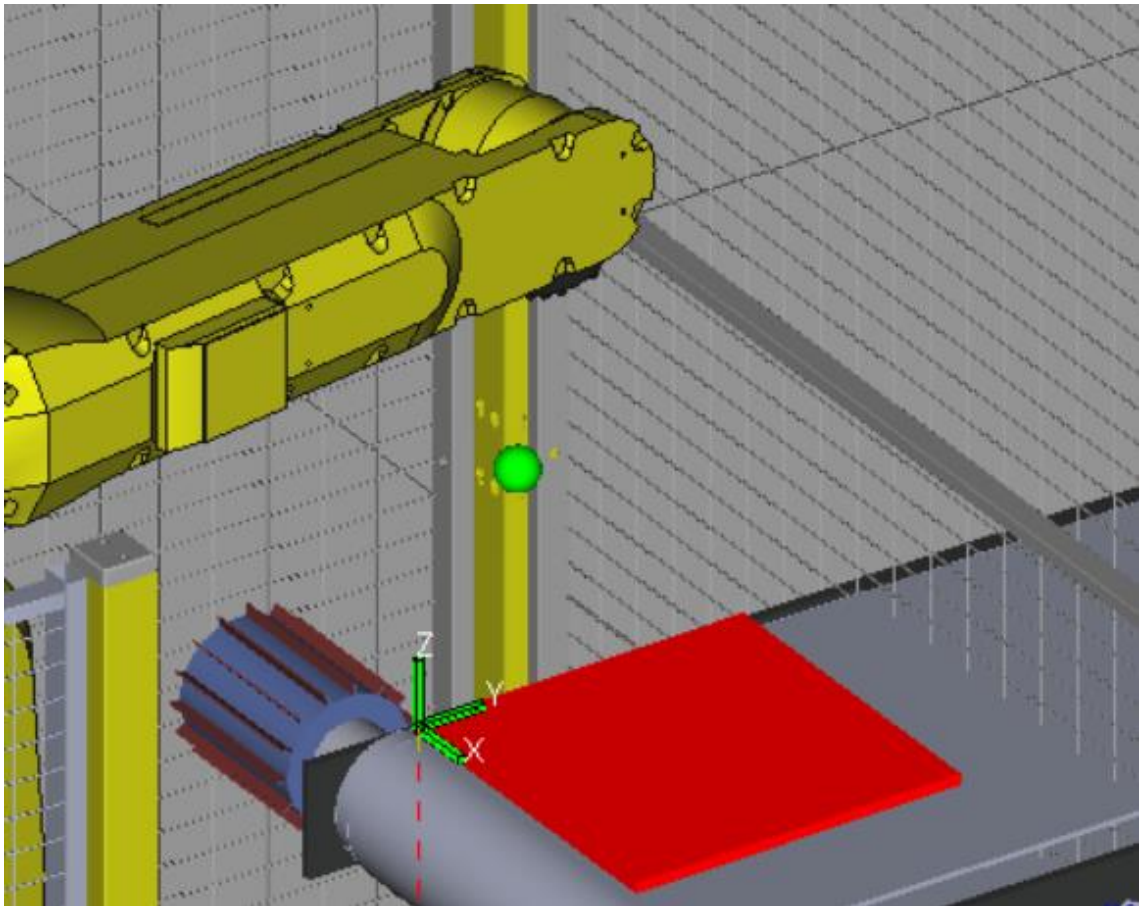
Työkalupisteiden opettamisen apuna käytettiin koneistettua piikkiä mikä mahdollistaa hyvin tarkan työkalupisteiden luomisen. Työkalupisteiden opettamisen jälkeen koneistettu-piikki siirrettiin kolmileukatarttujaan kiinni. Piikkiä käytettiin käyttäjäkoordinaatistojen opetukseen.

Messusoluun luotiin käyttäjäkoordinaatit käyttötarkoitusten mukaisesti; kuljetin, kohdistuspöytä, sorvin pakkojen välinen keskiakseli ja paletointipöytä. Käyttäjäkoordinaatit tojen luominen helpottaa robotin käsittelyä kyseisissä alueissa, erityisesti laskennallisten pisteiden osalta.



Kuva 15. Käyttäjäkoordinaatisto

Kuvan 15 mukaisen käyttäjäkoordinaatiston opetus tehdään robotin työkalulla osoittamalla ja tallentamalla kolme pistettä. Ensin tallennetaan koordinaatiston nollapiste, jonka jälkeen opetetaan erikseen koordinaatiston X-akselin positiivinen suunta ja Y-akselin positiivinen suunta. Opettaessa suuntia tulee olla tarkkana mihin suuntaan Z-akselin positiivisen suunnan halutaan osoittavan koska tämä on riippuvainen siitä mihin suuntaan X- ja Y-akselin suunnat on opetettu.



Kuva 16. Havainnekuva kuljettimen koordinaatisto

Kuljettimelle luotiin kuvan 16 mukainen koordinaatisto. Koordinaatiston nollapiste opetettiin kuljettimella sijaitsevan anturin linjalle kuljettimen vasempaan laitaan ja kuljettimen maton pintaan. Käyttäjäkoordinaatiston Z-akselin positiivinen suunta haluttiin osoittavan ylöspäin kuljettimen pinnasta. Tämä määritteli sen että X-akselin positiivinen suunta tuli osoittaa kuljettimen kulkusuuntaan nähden poikittain ja Y-akselin positiivinen suunta kuljettimen kulkusuunnan mukaisesti solusta ulospäin.

4.2 Robotin ohjelmointi

Tässä osiossa käydään läpi robotin ohjelmointiin liittyvät asiat ja miten messudemo robotin ohjelmointi suoritettiin tärkeimmiltä osin. Jokaista riviä koodia ei siis ole esittynä ainoastaan tärkeimpiä komentoja ja niiden tärkeimpiä ominaisuuksia.

4.2.1 Ohjelmointi

Fanuc robottien ohjelmointia voidaan tehdä kahdella eri koodikielellä jotka Fanuc on itselleen ottanut käyttöön. Yleisesti robotin liikeradat ja ohjelmasuoritukset tehdään TP-

ohjelmiin. TP -ohjelma on binäärimuodossa robotin ohjaimessa vaikkakin se näkyy robotin ohjaimessa helposti luettavana tekstinä. TP -koodikieli on suunniteltu yksinkertaiseksi, joka helpottaa ohjelmoinnin aloittamista. TP-koodikielessä on omat rajoitteensa sen yksinkertaisuuden takia. TP-ohjelmissa voidaan suorittaa kevyttä logiikkaohjelmointia ja hallita rekisteridataa jonka kautta pystytään laskemaan perusyhtälöitä kuten plus-, miinus-, kerto- ja jakolaskuja. Vaikeampiin sovelluksiin kuten trigonometrian laskemiseen tai samanaikaiseen ohjelma suorittamiseen robotista löytyy KAREL-koodikieli joka on TP:tä korkeampi koodikieli. KAREL pohjautuu Pascaliin ja sorvien ohjaukseen.

Robotin ohjelmointia voi suorittaa joko Roboguidessa tai suoraan ohjaukskapulasta TP-ohjelmien osalta. KAREL-ohjelmien lähdekoodi tulee kirjoittaa tekstimuodossa tietokoneella ja kääntää sitten robotille syötettävään PC-tiedostomuotoon.

Tehokkaaseen ohjelmointiin päästään purkamalla jokainen työvaihe yksittäiseksi ohjelmaksi ja tämän jälkeen ohjelmat ketjutetaan siltä osin mitkä aliohjelmat kuuluvat samaan toteutusvaiheeseen. Robotille luodaan MAIN-ohjelma, josta PLC:ltä tulevilla signaaleilla suoritetaan tietty ohjelmien ketju.

4.2.2 Rekisterit

Fanuc roboteissa on numeroarvojen tallentamiseen ja käsittelyyn kaksi erillistä rekisteriä. Rekisteri puolelle (KUVA 17) pystytään tallentamaan yksittäisiä numeroarvoja, joita taas voidaan käyttää apuna robotin ohjelmoinnissa.

DATA Registers		20/200
R[10:paikkag]=10	
R[11:viive]=1	
R[12:frame]=1	
R[13:tool]=1	
R[14:paikkagarvo]=1	
R[15:]=0	
R[16:]=0	
R[17:RIVI_P1]=0	
R[18:DISTANCE_P1]=0	

KUVA 17. Rekisteri muisti

Paikkarekisteri puolelle (KUVA 18) pystytään tallentamaan yhteen muistipaikkaan kuusi erillistä arvoa. Paikkarekistereitä voidaan käyttää robotin liikepisteiden koordinaatteina tai niihin voidaan syöttää arvoja, joilla hallinnoidaan robotin siirtymää OFFSET -komentoissa.

DATA Position Reg					
PR[3]	UF:F	UT:F		CONF:FUT	000
X	221.000	mm	W	0.000	deg
Y	67.000	mm	P	0.000	deg
Z	87.000	mm	R	0.000	deg
Position Detail					
PR[2:	KAANTO			=R	
PR[3:	MAG_POIMINTA			=R	
PR[4:	TART_POIMINTA			=R	
PR[5:				=*	
PR[6:	NOLLAUS_PR			=R	

KUVA 18. Paikkarekisteri

4.2.3 Liikekäskyt

Robottien liikekäskyjä on pääsääntöisesti kahta erityyppiä. Joint-liikkeessä robotti liikkuu vapaasti omien laskelmien mukaisesti tallennettuun pisteeseen käyttäen jokaista akseliaan. Linear-liikkeissä liikutaan suoraa linjaa kahden pisteen välillä. Pisteiden paikat voidaan tallentaa joko astekulmina akseleista tai karteesisina arvoina, jolloin paikan tallennus kannattaa sitoa harkinnalla tiettyyn käyttäjäkoordinaatistoon. Kuvassa 19 on avatuna paikkapisteseen tallentuvat tiedot.

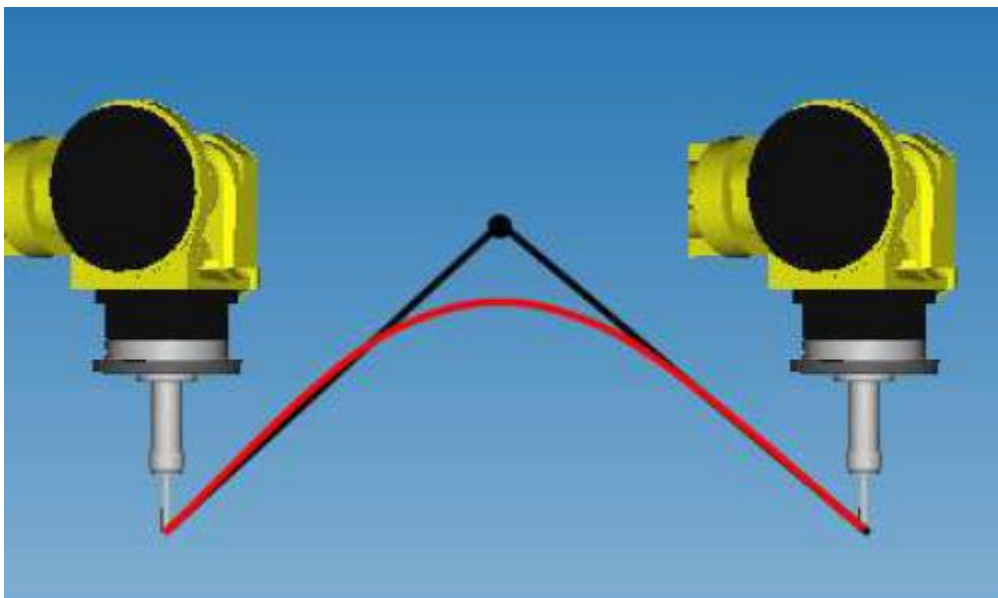
T_SORVILLE					
P[8]	UF:0	UT:7		CONF:NUT	000
X	321.666	mm	W	-177.438	deg
Y	-616.420	mm	P	.042	deg
Z	644.402	mm	R	-24.015	deg
Position Detail					
91:J	P[8:ABOVE_MC]			30%	FINE
92:					

Kuva 19. Paikkapisteen tiedot

Kuvassa 19 esitetty paikkapiste on tallennettu robottiin käyttäen karteesta tallennustapaa, jolloin koordinaattien yläpuolella näkyy myös määrittely siitä mihin käyttäjäkoordinaatistoon ja työkalupisteeseen paikka on sidottu.

Liikekäskyissä on paikkapisteen lisäksi määritelty millä nopeudella pisteeseen liikutaan ja kuinka tarkasti robotti paikoittaa itsensä pisteeseen ennen kuin se siirtyy ohjelman lukemisessa pidemmälle. Nopeudet määritetään Joint-liikkeille prosentteina mikä siis rajoittaa liikenopeuden haluttuun prosenttilukuun verrattuna maksiminopeuteen, kun ohjelma suoritetaan 100% nopeusohjeella. Linear-liikkeissä nopeus esitetään vaihtoehtoisesti millimetriä sekunnissa tai astetta sekunnissa.

Robotin paikotukseen liittyvä CNT -arvo on liikkeeseen sidottu arvo siitä kuinka lähelle robotti ajaa pistettä ennen kuin se siirtyy seuraavalle riville ohjelmassa. Esimerkkinä jos CNT-arvo on 100 robotti ajaa niin läheltä kyseistä pistettä kuin mahdollista ilman että robotin liikenopeus muuttuu. Kun taas jos CNT-arvo on ilmoitettu FINE niin robotti pysäyttää itsensä määritettyyn paikkapisteseen.



Kuva 20. CNT-arvo

Kuvassa 20 on robotille ohjelmoitu kolmen pistettä sisältävä liikerata. Kuvan musta viiva kuvaa reittiä mikä tapahtuu jos välipiste on asetettu FINE-tarkkuudella toteutuvaksi kun taas punainen viiva kuvaa reittiä jossa välipisteen paikoitus on CNT100-tarkkuudella. Muokatessa CNT-arvoja tulee huomioida se että kyseessä on liikenopeuteen sidonnainen

määrittäminen. Tästä johtuen kun ohjelman testiajo aloitetaan tulee se suorittaa ehdottomasti nopeusohjeen 100% arvolla jolloin nähdään todellinen oikaisu paikkapisteiden välillä.

4.2.4 TP-ohjelmat

Robotin liittäminen osaksi PLC:tä sidottiin digitaalisiin signaaleihin jotka välitettiin käyttäen ethercat -väylää laitteiden välillä. Logiikka asetettiin masteriksi ja robotti slaveksi joten robotille kirjoitettiin MAIN niminen TP -ohjelma joka mahdollisti tämän. Kuvassa 19 on esillä MAIN -ohjelman rakenne.

```

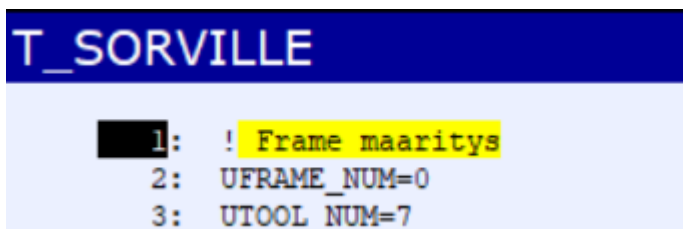
1: LBL[1: Kotiutus] ;
2: ;
3: UTOOL_NUM=8 ;
4: UFRAME_NUM=0 ;
5: J P[1:KOTI] 40% FINE ;
6: ;
7: / Tehtävän valinta ;
8: LBL[2: Tehtavat] ;
9: ;
10: IF DI[130:R_to_hihna]=ON AND DI[158:R_PURA]=OFF, CALL ETSINTA ;
11: ;
12: IF DI[141: Hae KPL hihnalta]=ON, CALL MAG_HAKU ;
13: ;
14: IF DI[154: Aihiot_loppu]=ON AND DI[132:R_to_kara2]=ON, CALL MAG_KARA2_PURKU ;
15: ;
16: IF DI[158:R_PURA]=ON AND DI[159:Kuvio_1]=ON, CALL PURKU_MAG_PAL1 ;
17: ;
18: IF DI[158:R_PURA]=ON AND DI[160:Kuvio_2]=ON, CALL PURKU_MAG_PAL2 ;
19: ;
20: / Tarkastus tuleeko pyyntoja ;
21: ;
22: IF (DI[130:R_to_hihna]=ON OR DI[141: Hae KPL hihnalta]=ON OR DI[154: Aihiot_loppu]=ON AND
DI[132:R_to_kara2]=ON OR DI[132:R_to_kara2]=ON OR DI[158:R_PURA]=ON AND DI[159:Kuvio_1]=ON OR DI[158:R_PURA]=
: ON AND DI[160:Kuvio_2]=ON) THEN ;
23: ;
24: JMP LBL[2] ;
25: ;
26: ELSE ;
27: ;
28: JMP LBL[1] ;
29: ;
30: ENDIF ;

```

KUVA 21. MAIN-ohjelma

Ohjelmassa MAIN robotti jatkuvasti lukee logiikalta tulevia signaaleja ja siirtyy niiden mukaisesti aliohjelmaan. Ohjelmaan on tehty silmukka, joka pitää robotin jatkuvasti aktiivisena. Kuvassa 21 rivillä yksi näkyvä LBL-komento on hyppy lausekkeen taso joka numeroidaan jolloin yksi ohjelma voi sisältää ainoastaan yhdessä kohtaa LBL-komennon tietyllä arvolla. Kuvan rivillä 28 näkyy komento JMP LBL-komento millä siirrytään LBL-komentoon jonka arvo on yksi. Käyttäen JMP LBL- ja LBL-komentoa yhdessä ehtolauseiden kanssa pystytään helposti luomaan ohjelmaan silmukka, jonka ansiosta robotti pysyy aktiivisena niin kauan kuin ylempi järjestelmä tuottaa sille uusia tehtäväpyyntöjä.

Tähän robottisoluun tehtiin jokaisesta työvaiheesta oma TP -ohjelma. Kuvassa 22 on esillä alustus T_SORVILLE-ohjelmasta, jossa kappale viedään solussa sijaitsevaan sorviin, ohjelman alussa tehdään määrittys liikepisteiden sidonnaisuudesta koordinaatistoihin.



```
T_SORVILLE
1: ! Frame maaritys
2: UFRAME_NUM=0
3: UTOOL_NUM=7
```

KUVA 22. Koordinaatistojen määrittys

Kuvan 22 rivillä kaksi sidotaan käyttäjäkoordinaatisto nolnaan ja rivillä kolme määritellään mikä työkalupiste on käytössä kyseisissä paikkapisteissä.

Robotille ohjelmoitiin tarkistuspisteitä ohjelmien sisälle joiden tarkoitus on varmistaa että robotilla on tosiaan lupa saapua alueelle tai suorittaa tietty toiminto. Kuvassa 23 on ote sorvin palvelussa tapahtuvasta kättelystä.

```

T_SORVILLE
91/118
73: WAIT DI[131:R_to_karal]=ON
74:
75: WAIT 2.00(sec)
76:
77: WAIT DI[151:SORVI_KOTONA]=ON
78:
79: UTOOL_NUM=7
80:
81:J P[8:ABOVE_MC] 30% CNT5
82:J P[9:MC_KARAL_RAMP] 30% CNT5
83:
84: WAIT DI[145:S_karal_auki]=ON
85:
86:L P[10:MC_KARAL ] 150mm/sec FINE
87: DO[131:R_at_karal]=ON
88: WAIT DI[145:S_karal_auki]=OFF
89: WAIT .50(sec)
90: RO[1:T_AUKI]=ON
91: DO[138:R-ladattu]=OFF
92:

```

KUVA 23. Kättely

Kuvassa 23 näkyvän ohjelman ylälaidassa on kaksi odotuskäskyä. Näistä ensimmäisessä robotti odottaa PLC:ltä tulevaa signaalia, joka antaa robotille luvan ajaa sorvin pakalle yksi. Toisessa odotuskäskyssä robotti odottaa että liikkuva pakka on varmasti saavuttanut kotiasemansa. Kun molemmat signaalit ovat kertaalleen toteutuneet robotti ajaa itsensä jättöpisteeseen ja antaa viestin PLC:lle että kappale on asetettuna leukoihin. Tämän jälkeen robotti jää odottamaan signaalia mikä kertoo robotille luvan päästää kappaleesta irti ja poistua paikalta.

4.2.5 KAREL-ohjelmat

KAREL-koodikieli on aloittelijalle hankalaa ja projektin aikataulu oli varsin tiukka, turvauduttiin toimeksiantajan valmiisiin ohjelmakirjastoihin. Näistä kirjastoista mesusolussa käytettiin kahta Sasbotics Oy:n kirjoittamaa KAREL-ohjelmaa; Nollaus ja PAIKKAG2. Nollaus -ohjelmassa robotti suoritti hallitun pysäytyksen, valitsi pääohjelman suoritettavaksi ja asetti itsensä pääohjelman ensimmäiselle riville pysähdys tilaan, ohjelman suoritus laukaistiin PLC:ltä tulevalta signaalilla.

Kappaleen etsintään liukuhihnalta liittyvä PAIKKAG2 -ohjelma suoritettiin robotin ohjauksessa taustaohjelmana. PAIKKAG2 -ohjelmassa robotti lähetti jatkuvasti työkalupisteen sijaintia suhteessa käyttäjäkoordinaatistoon PLC:lle. Tämän ohjelman avulla suoritettiin kappaleen etsintä liukuhihnalta. Kuvassa 24 nähdään miten Karel-ohjelma PAIKKAG2 alustetaan rekistereillä ja asetetaan käyntiin.

```
ETSINTA
3/61
1: DO[135:R_swipe_start]=OFF
2: DO[136:R_swipe_end]=OFF
3: UFRAME_NUM=1
4: UTOOL_NUM=1
5: R[11:viive]=1
6: R[12:frame]=1
7: R[13:tool]=1
8: R[14:paikkagarvo]=1
9:
10:J P[5:ASENTO] 40% CNT50
11: DO[130:R_at_hihna]=ON
12:J P[1:SWIPE_RAMP] 30% CNT100
13:J P[2:swipe start] 20% FINE
14: DO[135:R_swipe_start]=ON
15: RUN PAIKKAG2
16:
17:L P[3:swipe end] 60mm/sec FINE
18: DO[135:R_swipe_start]=OFF
19: WAIT .20(sec)
20: DO[136:R_swipe_end]=ON
```

KUVA 24. PAIKKAG2 alustus

Ohjelman riveillä 5-8 asetetaan PAIKKAG2 ohjelmaan viive sekunteina, käyttäjäkoordinaatisto missä liikutaan, työkalu minkä liikettä seurataan ja signaaliryhmä mitä pitkin tieto lähetetään PLC:lle. Tämän jälkeen rivillä 15 käynnistetään PAIKKAG2 ohjelma komennolla RUN.

5 POHDINTA

Robottisolu valmistui hyvin aikataulussa. Solulle määritetyt vaatimukset saatiin tehtyä hyvin ennen messuille menoa. Toimeksiantajan pyynnöstä solun toimintaa muutettiin ensimmäisen messupäivän jälkeen niin että se toimisi täysin itsenäisesti jatkuvana kehänä. Muutokset tehtiin messujen aukioloajan jälkeen, näytteilleasettajille tarkoitetun huoltoajan aikana.

Projektin robotin ohjelmoinnissa tuli tehtyä turhan vaikeita ratkaisuja asioiden tekemiseksi mutta näitä on helppo tulevaisuudessa välttää. Vaikeisiin ratkaisuihin päätyminen johtui kokemattomuudesta. Robotin ohjelmat tehtiin toimiviksi eikä niitä korjattu sen jälkeen kun ohjelma saatiin toimimaan riittävällä varmuudella. Ohjelmien optimointia ei siis tehty millään tasolla koska aikataulu oli toteutuksen puolesta ehdoton. Robotin ohjelmien teon haastavuus tuli myös esille siinä että kokemusta ulkoisesti ohjatusta robotista ei ollut ennen tätä projektia. Lisäksi logiikan ja robotin väliset kättelyt luotiin sitä tahtia mitä projektissa edettiin työvaiheissa, joten ohjelmien tekoa hidasti myös toinen osapuoli tässä projektissa.

Projekti antoi hyvän käsityksen siitä kuinka kompleksisiä palvelusolut ovat pinnan alla. Projekti kokonaisuudessaan oli onnistunut antaen vahvan tuntuman siihen miten tulevaisuudessa näitä asioita tulisi tehdä ja mihin asioihin tulisi panostaa esisuunnittelu-asteella enemmän.

LÄHTEET

Sasbotics. WWW-sivu. Luettu 10.1.2018

<http://sasbotics.fi/sasbotics/>

Fanuc. WWW-sivu. Luettu 5.2.2018

<http://www.fanuc.eu/fi/en/robots/robot-filter-page/m-20-series/m-20ib-25>

Fanuc. 2017. Esite. M-20iB/25

Beckhoff. CX5130. Esite. Luettu 1.2.2018

https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/Embedded_PC/CX5130.pdf

Omron. Os32C. Esite. Luettu 19.1.2018

https://assets.omron.eu/downloads/brochure/fi/v2/os32c_brochure_fi.pdf

LIITTEET

Liite 1. Fanuc M-20iB/25 esite

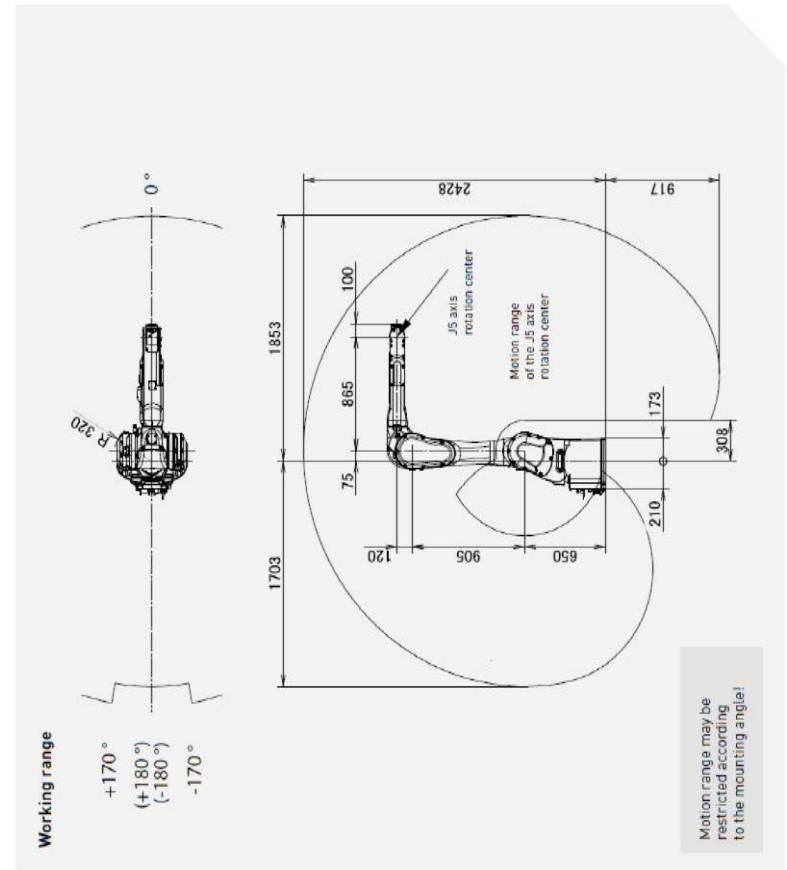
Controlled axes		Repeatability (mm)	Mechanical weight (kg)	Motion range [°]						Maximum speed [°/s]						J4 Moment/Inertia (Nm/kgm ²)	J5 Moment/Inertia (Nm/kgm ²)	J6 Moment/Inertia (Nm/kgm ²)
				J1	J2	J3	J4	J5	J6	J1	J2	J3	J4	J5	J6			
6		± 0.02*	210	340(360)	240	303	400	290	540	205	260	415	415	880	51/2.2	51/2.2	31/1.2	

M-20iB/25

Max. load capacity at wrist: **25 kg**

Max. reach: **1853 mm**

Robot	M-20iB/25
Robot footprint [mm]	383 x 343
Mounting position Floor	•
Mounting position Upside down	•
Mounting position Angle	•
Controller	R30iB
Open air cabinet	-
Mate cabinet	o
A-cabinet	•
B-cabinet	o
Pendant Touch	•
Electrical connections	
Voltage 50/60Hz 3phase [V]	380-575
Voltage 50/60Hz 1phase [V]	-
Average power consumption [kW]	1
Integrated services	
Integrated signals on upper arm In/Out	8/2
Integrated air supply	1
Environment	
Acoustic noise level [dB]	< 70
Ambient temperature [°C]	0-45
Protection	
Body standard/optional	IP67
Wrist & J3 arm standard/optional	IP67



MDS-00096-EN Technical information subject to changes without prior notice. All rights reserved. ©2017 FANUC Europe Corporation

• standard o on request - not available () with hardware and/or software option *Based on ISO9283

CX5130



i CX5130 | Embedded PC with Intel® Atom™ processor

The CX5130 has an Intel® Atom™ multi-core processor with a clock rate of 1.75 GHz. This makes genuine multi-core technology possible in the Embedded PC segment. The hardware interfaces in this new series are oriented and implemented identically to those of the existing CX5000 series. Two independent, Gigabit-capable Ethernet interfaces as well as four USB 2.0 and a DVI-I interface are available. A multitude of further connection options and gateway functions is created by an option interface, which can be pre-equipped ex factory, as well as the I/O level, which can optionally consist of either E-bus or K-bus terminals.

The CX5130 is characterised by low power consumption and fanless design.

Depending on the installed TwinCAT runtime environment, the CX5130 can be used for implementing PLC or PLC/Motion Control projects with or without visualisation. The execution of Motion Control applications with interpolating axis movements is also possible.

The extended operating temperature range from -25 to +60 °C enables the use in climatically demanding environments.

Like the CX5000, the CX5100 series has a compact design; a modular device with extension modules like in the CX2000 series is not available.

The order identifier is derived as follows:

CX5130-015T	Optional interfaces:
0 = no TwinCAT	CX5130-N020 = audio interface
1 = with TwinCAT 2 PLC runtime	CX5130-N030 = RS232, D-sub plug
2 = with TwinCAT 2 NC PTP runtime	CX5130-N031 = RS422/RS485, D-sub socket
3 = with TwinCAT 2 NC I runtime	CX5130-M310 = PROFIBUS master, D-sub socket, 9-pin
5 = TwinCAT 3 runtime (XAR)	CX5130-B310 = PROFIBUS slave, D-sub socket, 9-pin
0 = no operating system	CX5130-M510 = CANopen master, D-sub plug, 9-pin
1 = operating system Windows Embedded Compact 7	CX5130-B510 = CANopen slave, D-sub plug, 9-pin
2 = operating system Windows Embedded Standard 7 P 32 bit	CX5130-M930 = PROFINET RT, controller
3 = operating system Windows Embedded Standard 7 P 64 bit	CX5130-B930 = PROFINET RT, device, Ethernet (2 x RJ45 switch)
4 = Windows 10 IoT Enterprise LTSB 32 bit	CX5130-B931 = PROFINET IRT, device, Ethernet (2 x RJ45 switch)
5 = Windows 10 IoT Enterprise LTSB 64 bit	CX5130-B950 = EtherNet/IP slave, Ethernet (2 x RJ45 switch)
Since not all combinations make sense, the table "Ordering information" contains a breakdown of the permissible combinations.	CX5130-B110 = EtherCAT slave, EtherCAT IN and OUT (2 x RJ45)

Liite 3. Beckhoff CX5130 esite

2 (3)

Technical data	CX5130
Processor	Intel® Atom™ E3827, 1.75 GHz
Number of cores	2
Flash memory	slot for CFast card and for microSD card (cards are not included)
Internal main memory	4 GB DDR3 RAM (not expandable)
Persistent memory	integrated 1-second UPS (1 MB on CFast card)
Interfaces	2 x RJ45, 10/100/1000 Mbit/s, DVI-I, 4 x USB 2.0, 1 x optional interface
Diagnostics LED	1 x power, 1 x TC status, 1 x flash access, 2 x bus status
Clock	internal battery-backed clock for time and date (battery exchangeable)
Operating system	Microsoft Windows Embedded Compact 7, Microsoft Windows Embedded Standard 7 P or Microsoft Windows 10 IoT Enterprise LTSB
Control software	TwinCAT 2 runtime TwinCAT 3 runtime (XAR)
I/O connection	E-bus or K-bus, automatic recognition
Power supply	24 V DC (-15 %/+20 %)
Current supply E-bus/K-bus	2 A
Max. power loss	11 W (including the system interfaces)
Dimensions (W x H x D)	142 mm x 100 mm x 92 mm
Weight	approx. 960 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 20
Approvals	CE, UL, Ex, IECEx
TC3 performance class	performance (40); please see here for an overview of all the TwinCAT 3 performance classes

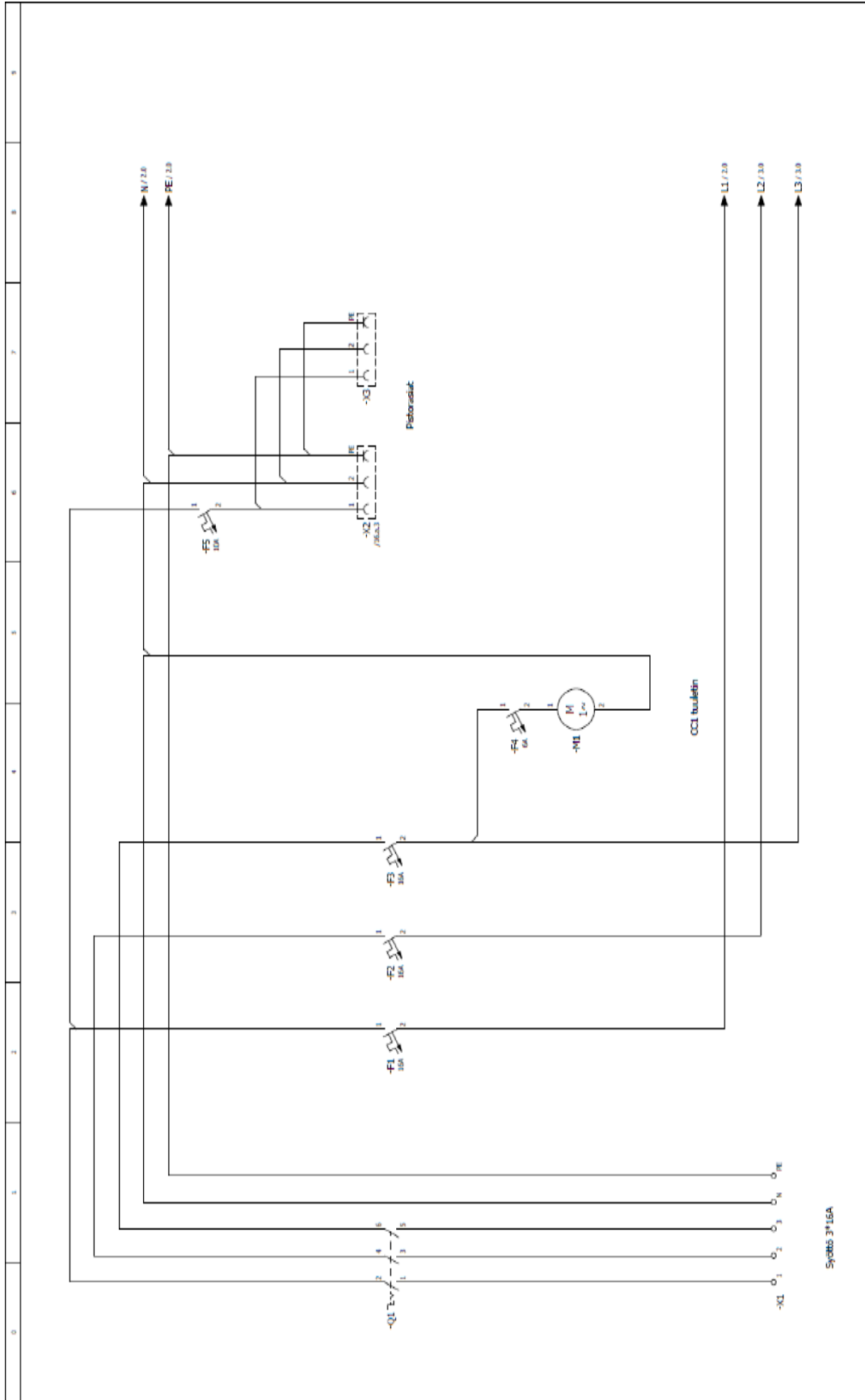
Ordering information	no operating system	Windows Embedded Compact 7	Windows Embedded Standard 7 P 32 bit	Windows Embedded Standard 7 P 64 bit	Windows 10 IoT Enterprise LTSB 32 bit	Windows 10 IoT Enterprise LTSB 64 bit	no TwinCAT	TwinCAT 2 PLC runtime	TwinCAT 2 NC PTP runtime	TwinCAT 2 NC I runtime	TwinCAT 3 run (XAR)
CX5130-0100	x	-	-	-	-	-	x	-	-	-	-
CX5130-0110	-	x	-	-	-	-	x	-	-	-	-
CX5130-0111	-	x	-	-	-	-	-	x	-	-	-
CX5130-0112	-	x	-	-	-	-	-	-	x	-	-
CX5130-0113	-	x	-	-	-	-	-	-	-	x	-
CX5130-0115	-	x	-	-	-	-	-	-	-	-	x
CX5130-0120	-	-	x	-	-	-	x	-	-	-	-
CX5130-0121	-	-	x	-	-	-	-	x	-	-	-
CX5130-0122	-	-	x	-	-	-	-	-	x	-	-
CX5130-0123	-	-	x	-	-	-	-	-	-	x	-
CX5130-0125	-	-	x	-	-	-	-	-	-	-	x
CX5130-0130	-	-	-	x	-	-	x	-	-	-	-
CX5130-0135	-	-	-	x	-	-	-	-	-	-	x
CX5130-0140	-	-	-	-	x	-	x	-	-	-	-
CX5130-0141	-	-	-	-	x	-	-	x	-	-	-
CX5130-0142	-	-	-	-	x	-	-	-	x	-	-
CX5130-0143	-	-	-	-	x	-	-	-	-	x	-
CX5130-0150	-	-	-	-	-	x	x	-	-	-	-
CX5130-0155	-	-	-	-	-	x	-	-	-	-	x

Accessories	
CX1900-0101	DVI-to-VGA passive adaptor for connecting a standard desktop VGA monitor to the CX system (singles out the VGA signals of the DVI-I interface).
CX2900-00xx	CFast cards: 2, 4, 8, 16, 32 GB CFast card
CX2900-0107	Device modification for CX5120, CX5130, CX5140 and CX9020 Embedded PCs according to the requirements for ATEX and IECEx certification. The modification is mandatory for the usage of CX5120, CX5130, CX5140 and CX9020 in hazardous areas, Zone 2/22. It includes the modification and repositioning of the device label as well as a mounting bracket installed ex works for mechanical locking of the connectors. Product labelling: ATEX: II 3 G Ex nA IIC T4 Gc and II 3 D Ex tc IIIC T135 °C Dc IECEx: Ex nA IIC T4 Gc and Ex tc IIIC T135 °C Dc Read the device documentation for use in hazardous areas carefully.

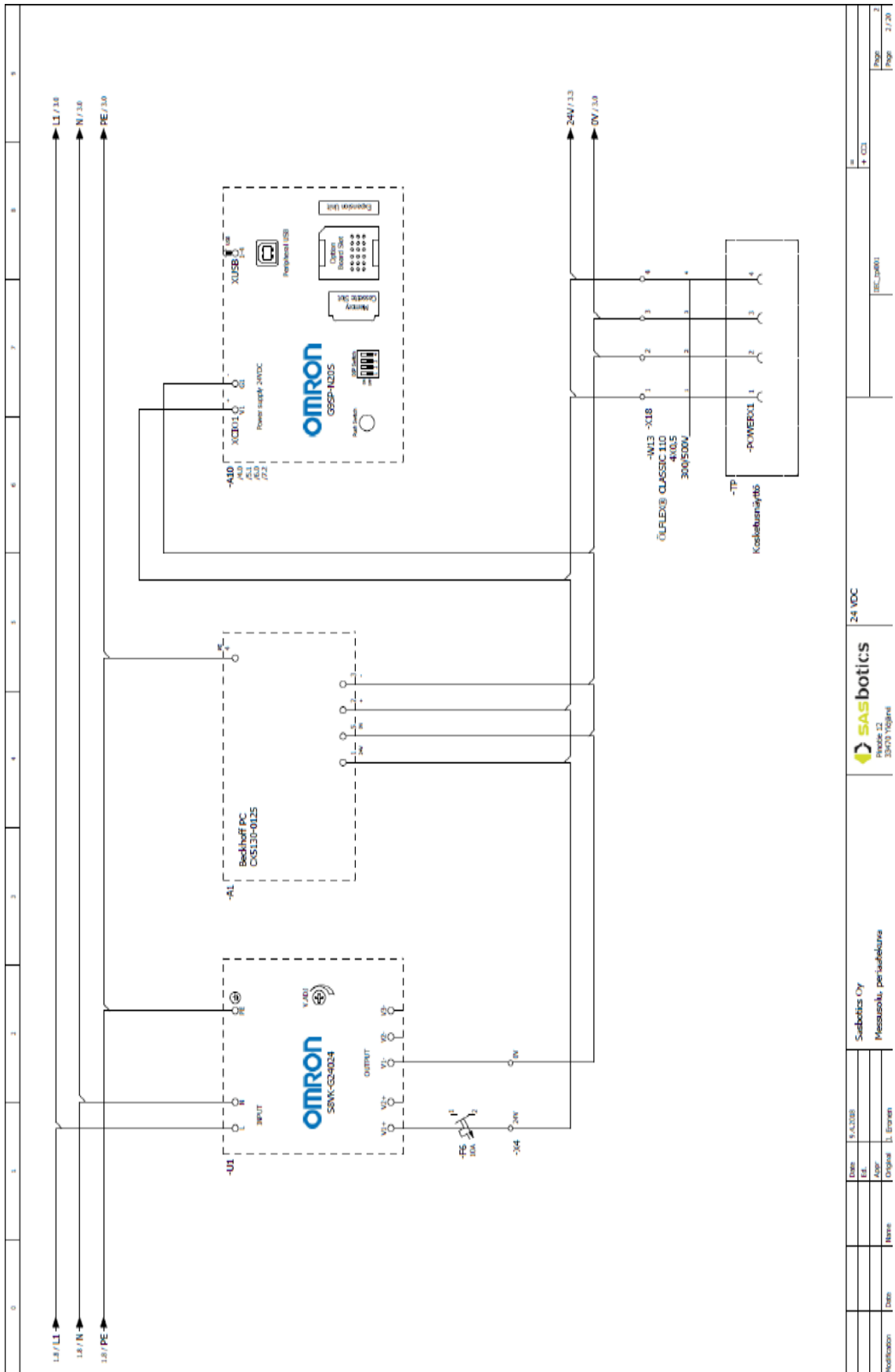


Optional interfaces	
CX5130-N010	DVI-D interface, additional DVI-D port for clone or extended desktop operation
CX5130-N011	DisplayPort interface, additional DisplayPort for clone or extended desktop operation
CX5130-N020	audio interface, 3 x 3.5 mm jack sockets, Line In, Mic In, Line Out or 5.1 Surround
CX5130-N030	RS232 interface, D-sub plug, 9-pin
CX5130-N031	RS485 interface, D-sub socket, 9-pin, configuration as an end point, without echo, termination on
CX5130-N031-0001	RS485 interface, D-sub socket, 9-pin, configuration as an end point, with echo, termination on
CX5130-N031-0002	RS485 interface, D-sub socket, 9-pin, configuration as drop point, without echo, termination off
CX5130-N031-0003	RS485 interface, D-sub socket, 9-pin, configuration as drop point, with echo, termination off
CX5130-N031-0004	RS422 interface, D-sub socket, 9-pin, configuration as full duplex end point, termination on
CX5130-B110	EtherCAT slave interface, EtherCAT IN and OUT (2 x RJ45)
CX5130-M310	PROFIBUS master interface, D-sub socket, 9-pin
CX5130-B310	PROFIBUS slave interface, D-sub socket, 9-pin
CX5130-M510	CANopen master interface, D-sub plug, 9-pin
CX5130-B510	CANopen slave interface, D-sub plug, 9-pin
CX5130-M930	PROFINET RT, controller interface, Ethernet (2 x RJ45)
CX5130-B930	PROFINET RT, device interface, Ethernet (2 x RJ45 switched)
CX5130-B931	PROFINET IRT, device interface, Ethernet (2 x RJ45 switched), in combination with TwinCAT 3 only
CX5130-B950	EtherNet/IP slave interface, Ethernet (2 x RJ45 switched)

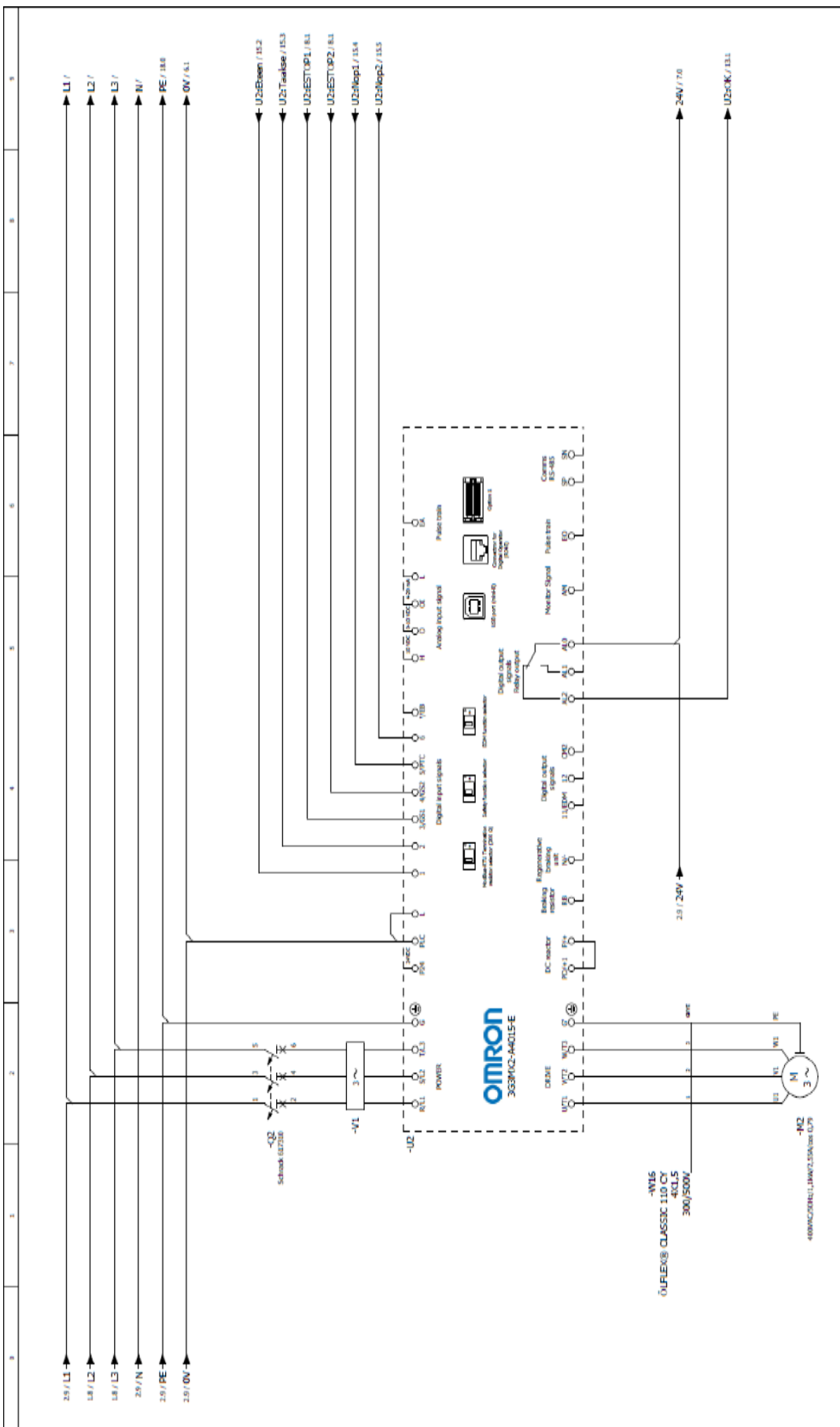
i Product announcement	CX5130-B931, CX5130-B950: estimated market release on request
-------------------------------	---------------------------------------------------------------

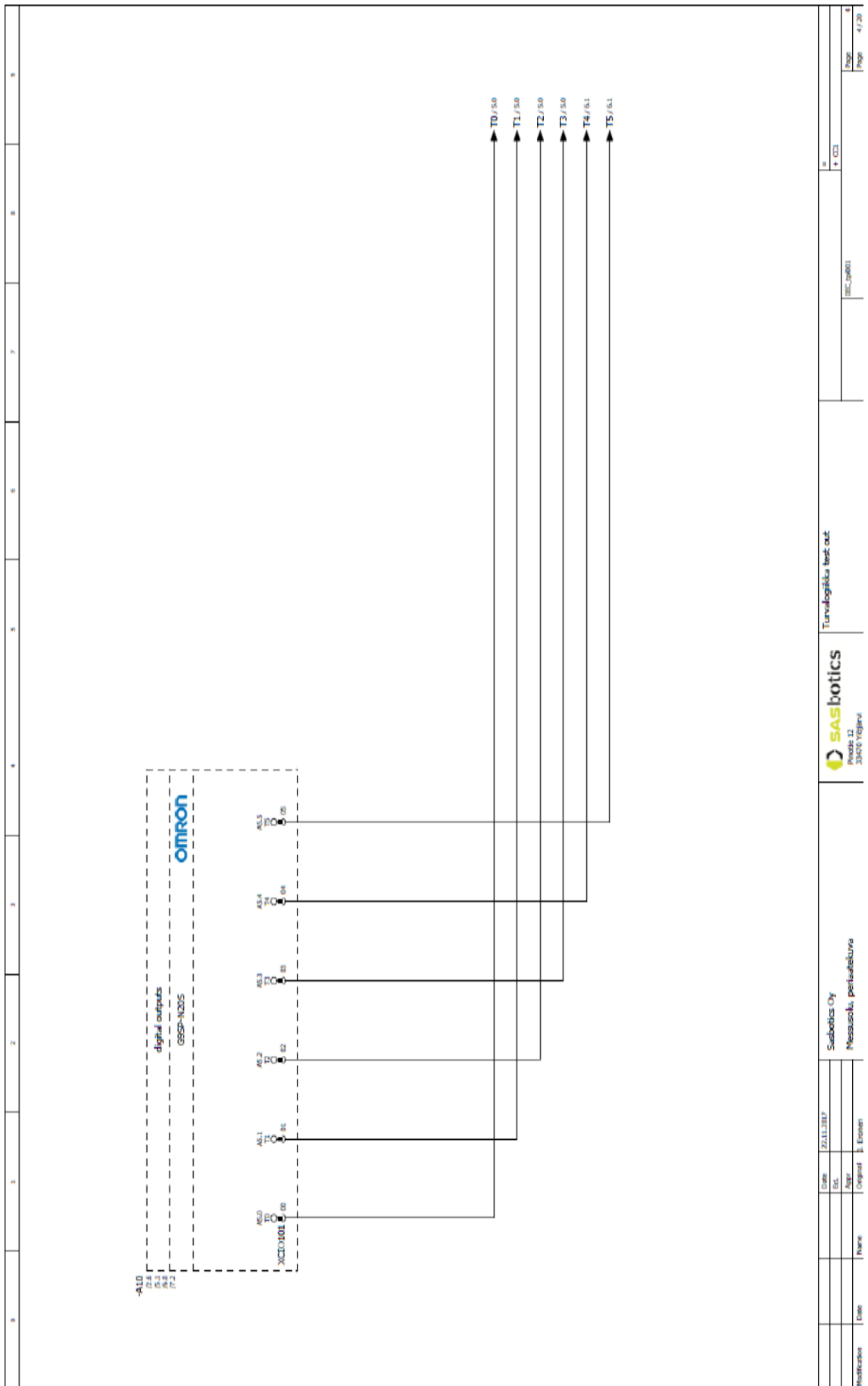


 Sasbotics Oy Messusuku, perälahtelina		400VAC	RE_0801	+	CCT	Page	1/20
Nimi: _____ Pvm: 22.11.2017 Laji: _____ Tilasto: Original Eräen	Syöttö 3*1GA						



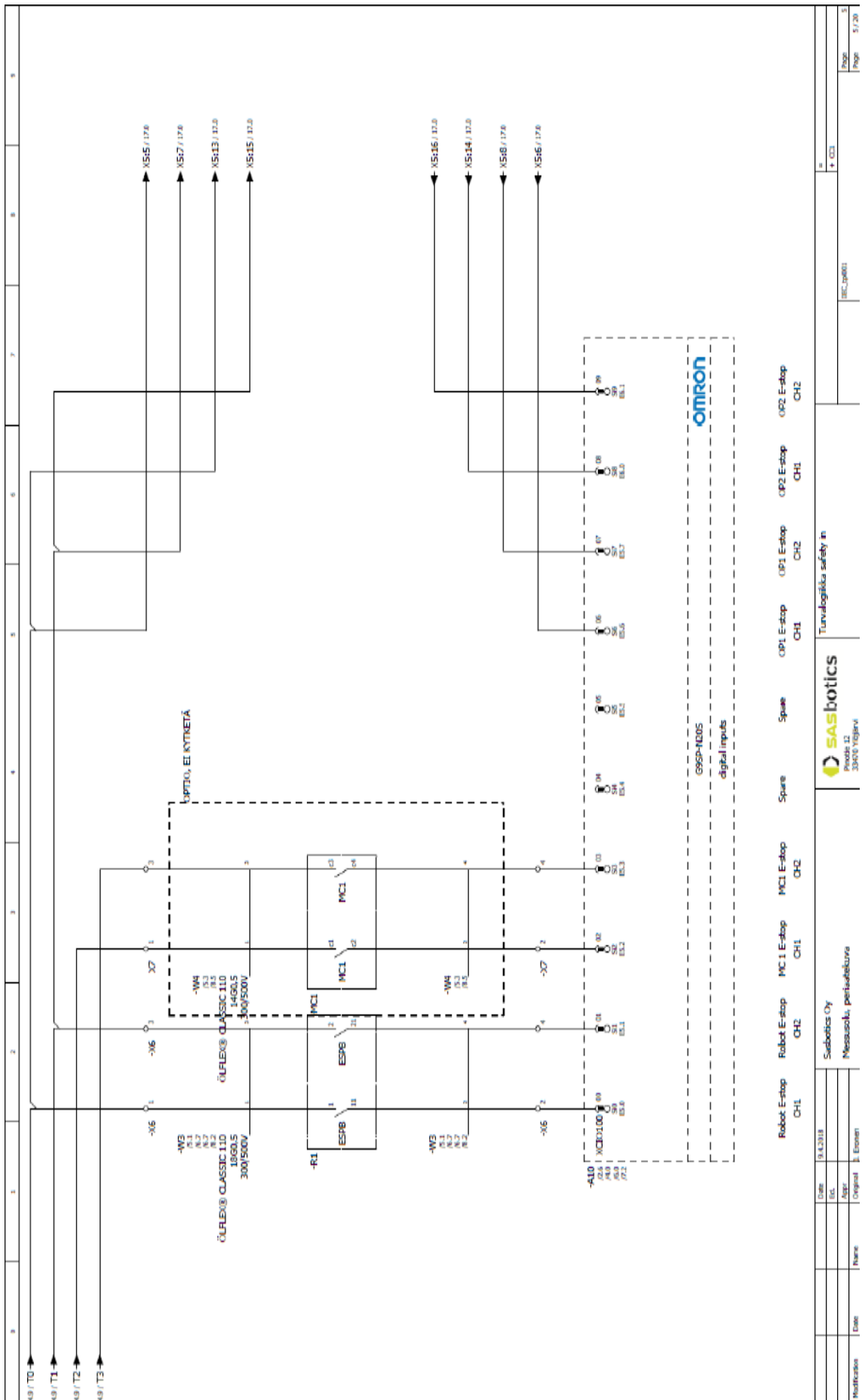
Revisio	1.0.2024	Sasbotics Oy	24 VDC	08C_2401	Page	2
EC		Mittausosasto			Page	2/20
Cor		Mittausosasto				
Original	1. Erä					





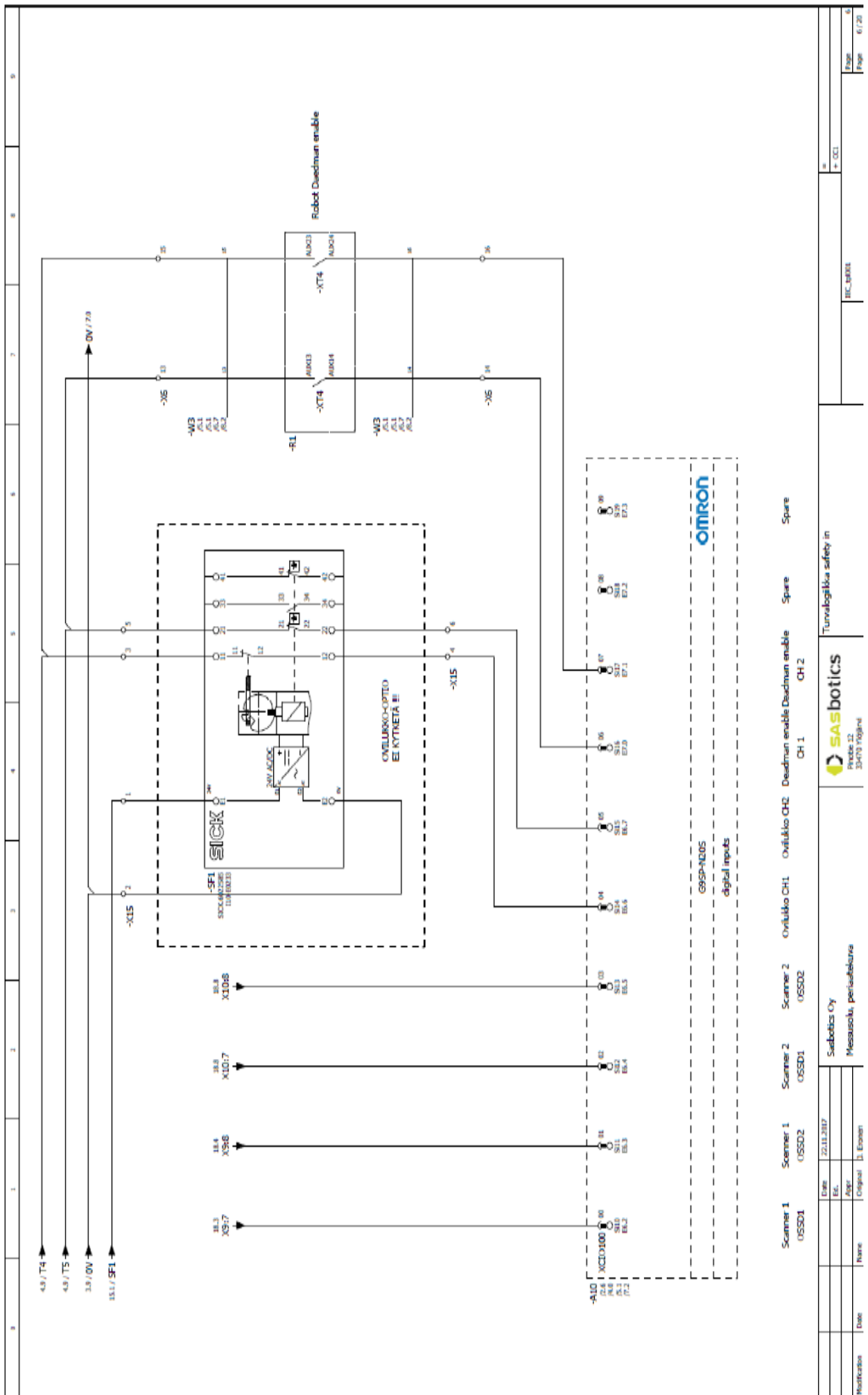
Liite 9. Sähkökuvat

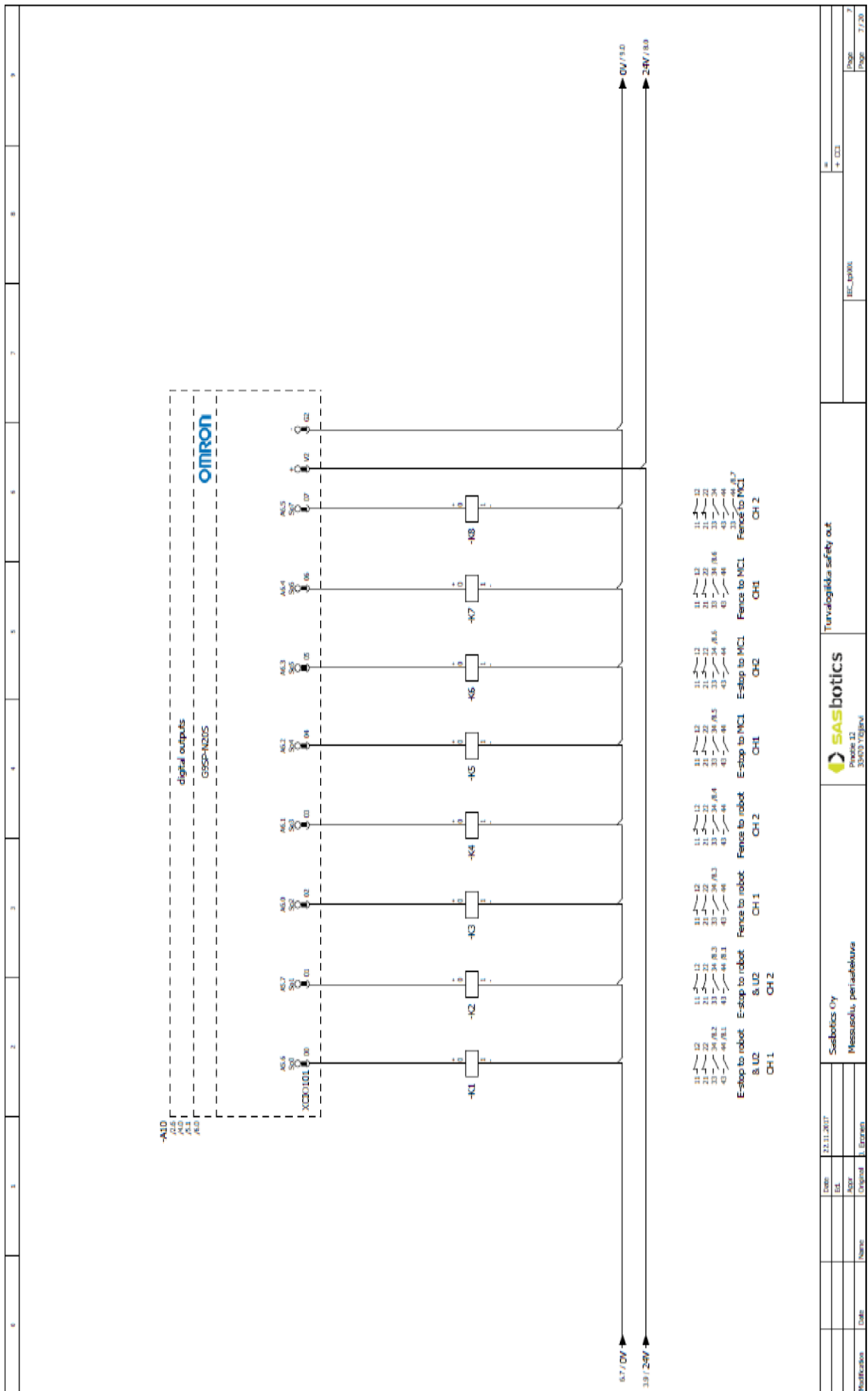
5(20)



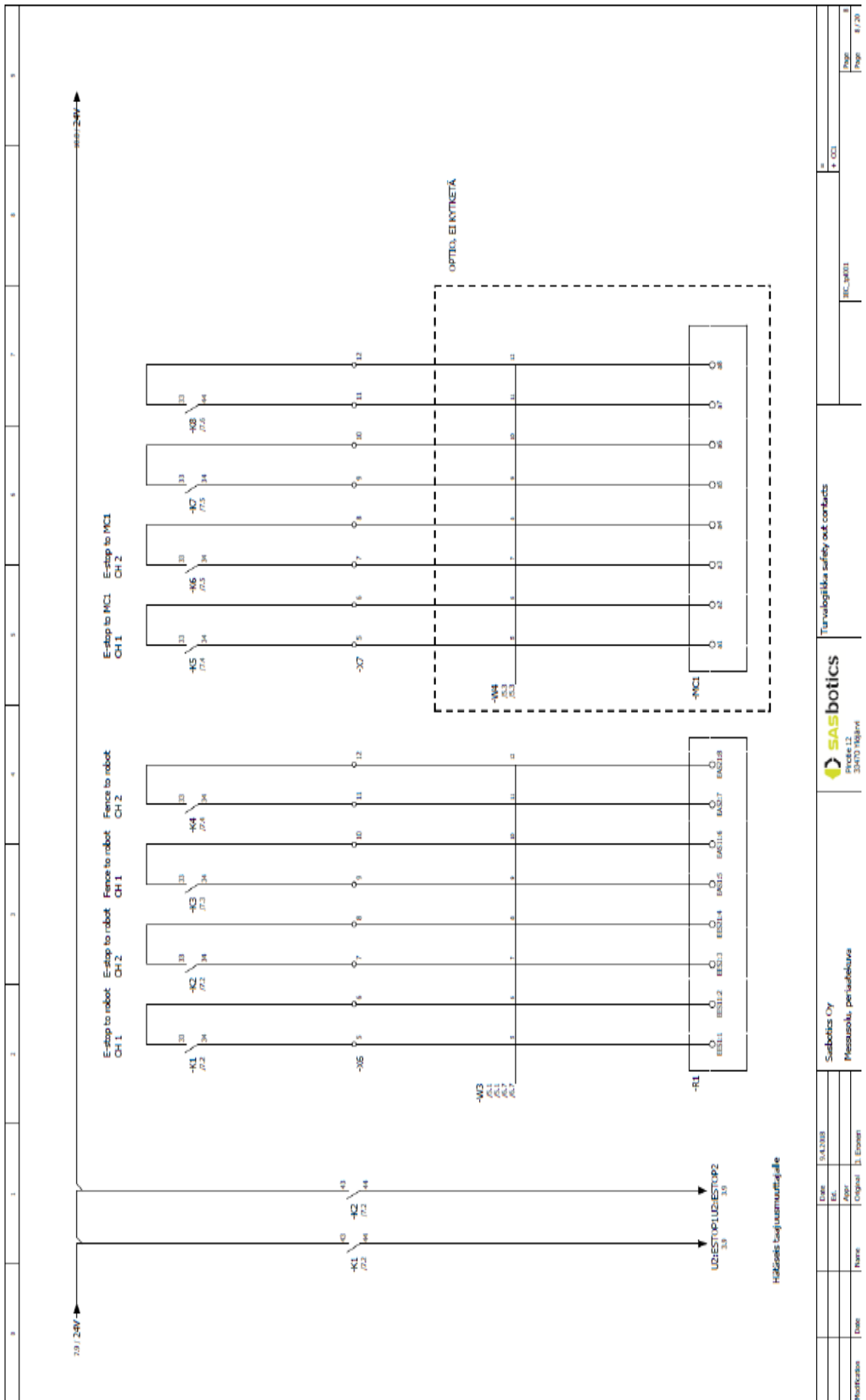
Liite 10. Sähkökuvat

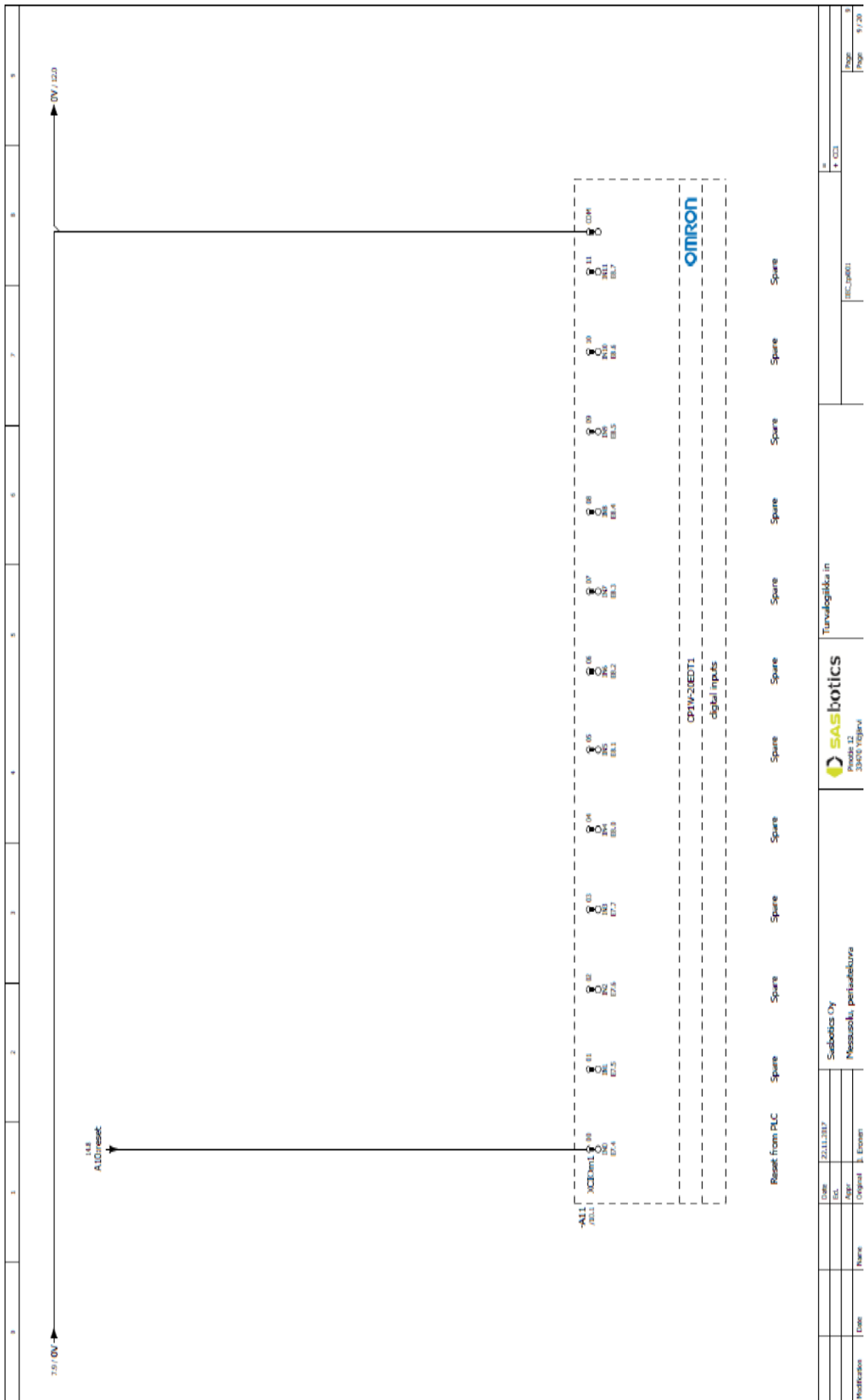
6(20)



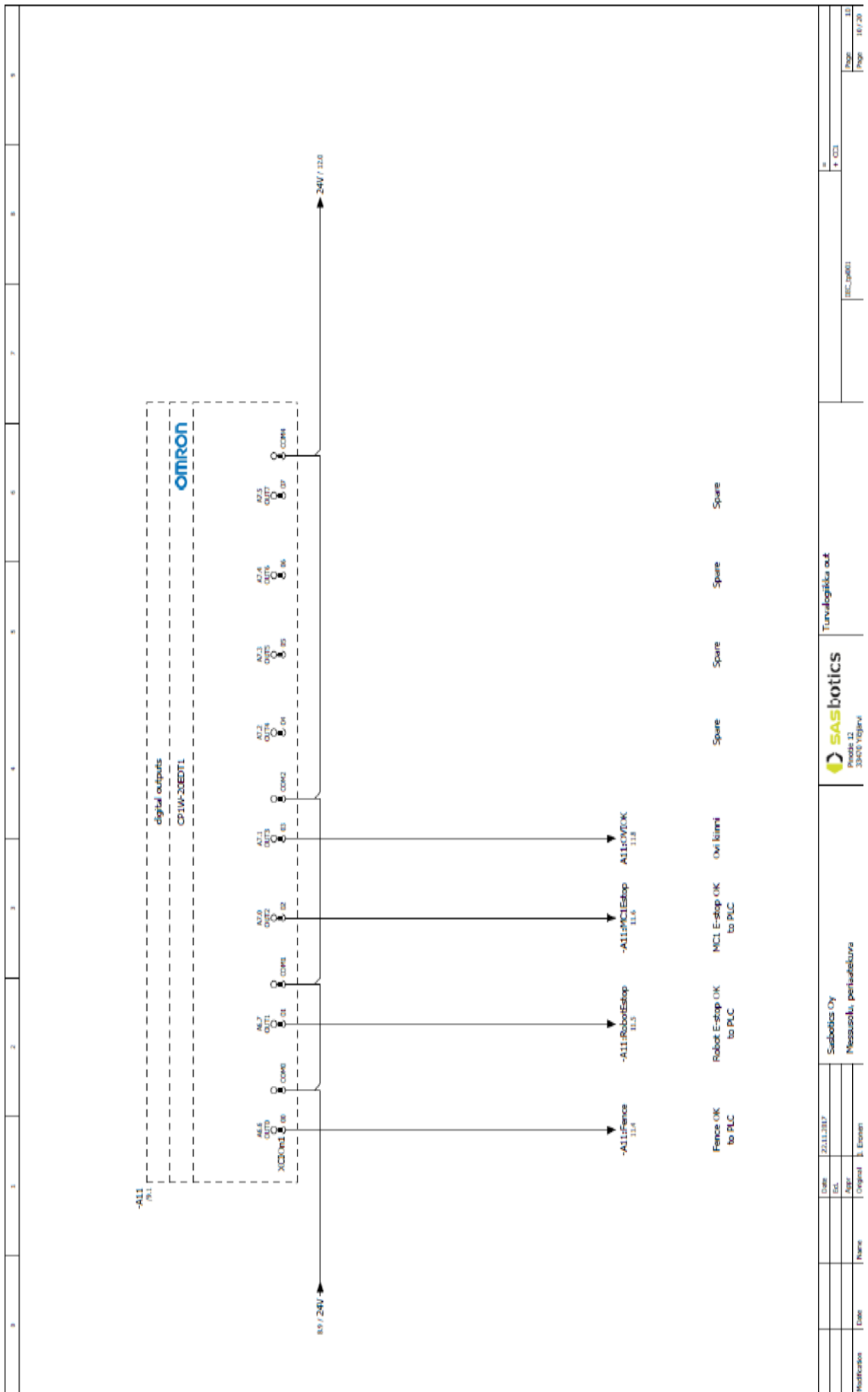


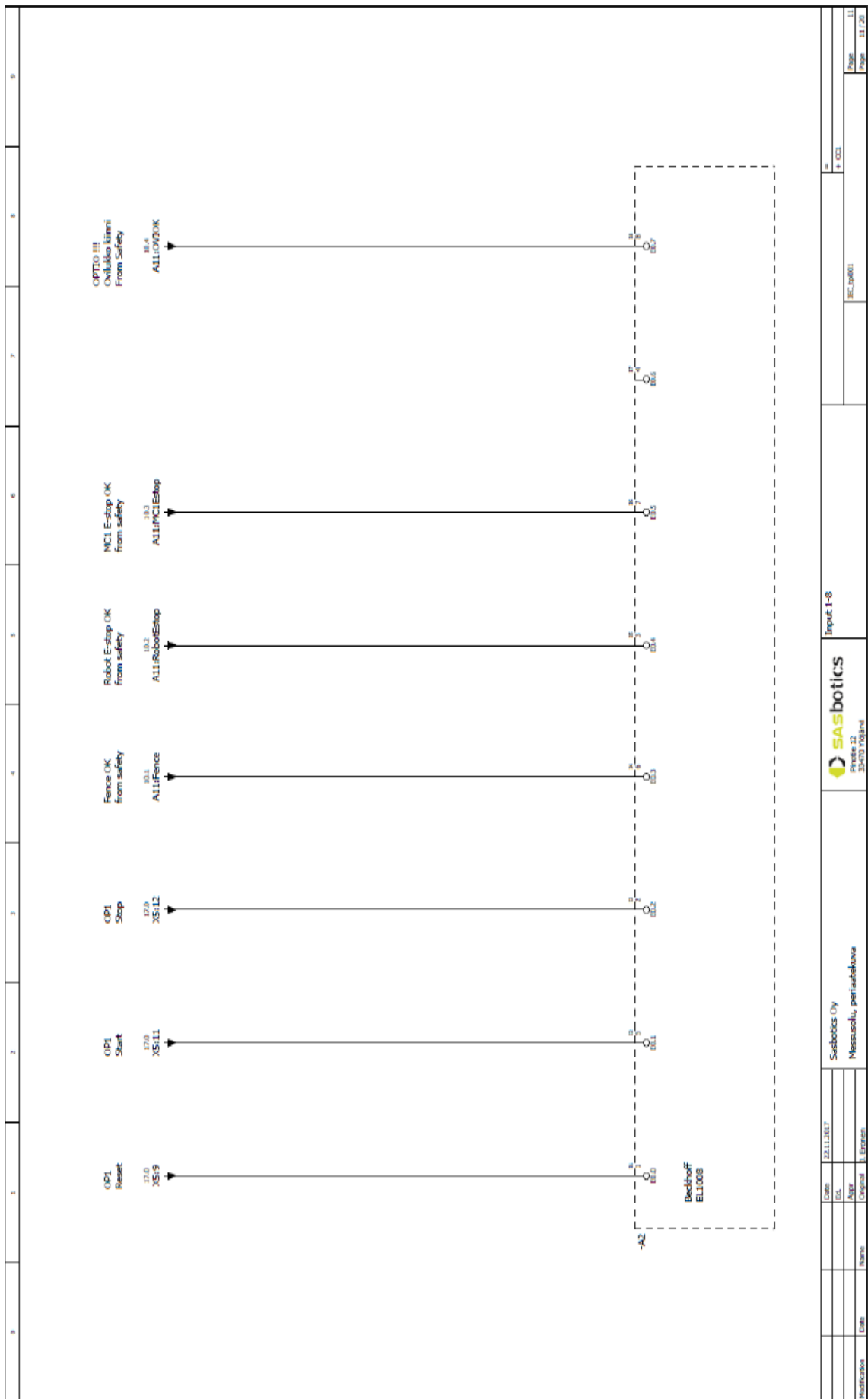
Rev	22.11.2017	SAABOTICS Oy	Turvallisuus safety out	Rev	4
Author		Messasolu, perustehtävä		Rev	1
Checker				Rev	2
Designer				Rev	3
Modifier				Rev	4
				Rev	5
				Rev	6
				Rev	7
				Rev	8
				Rev	9
				Rev	10
				Rev	11
				Rev	12
				Rev	13
				Rev	14
				Rev	15
				Rev	16
				Rev	17
				Rev	18
				Rev	19
				Rev	20



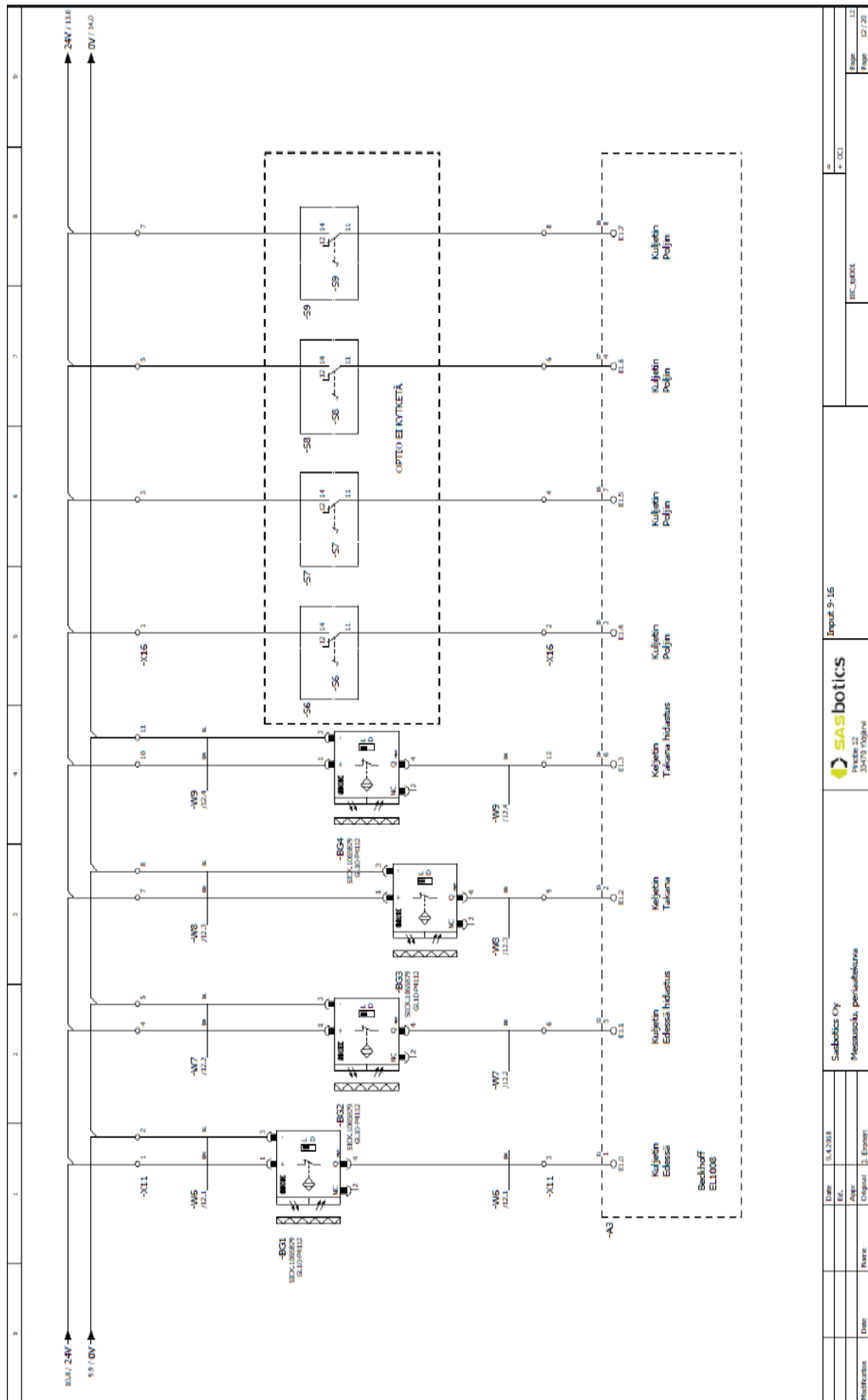


Modification	Date	Name	Original	E. Eronen
	22.11.2017			
	Doc.			
	App.			
Sasbotics Oy Messausloja, perinteiskivura				
Sasbotics Puhelin 12 30007163jvu				
Turvatekniikka in				
				REC_2001
				+ 033
				Page 9 / 20

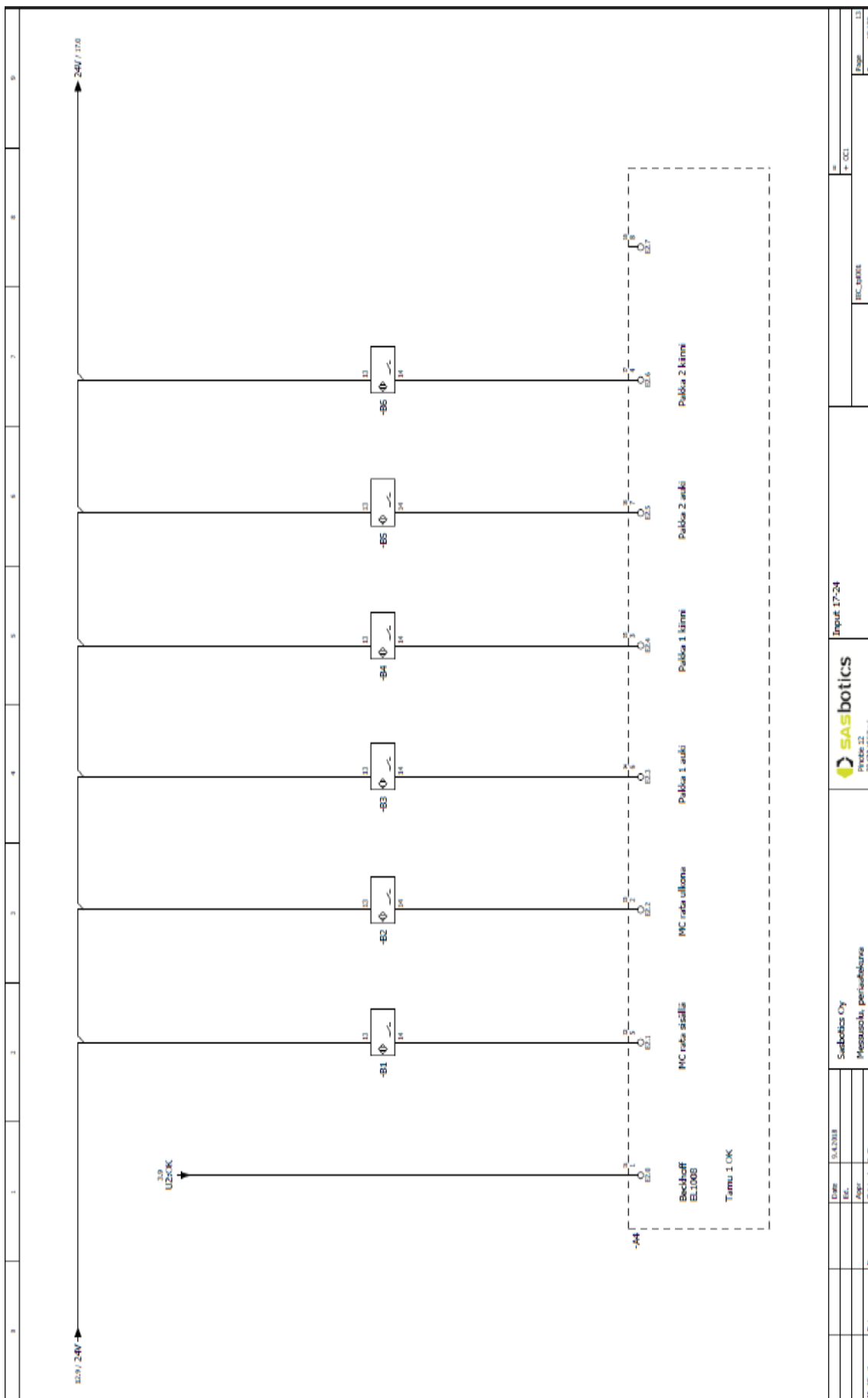




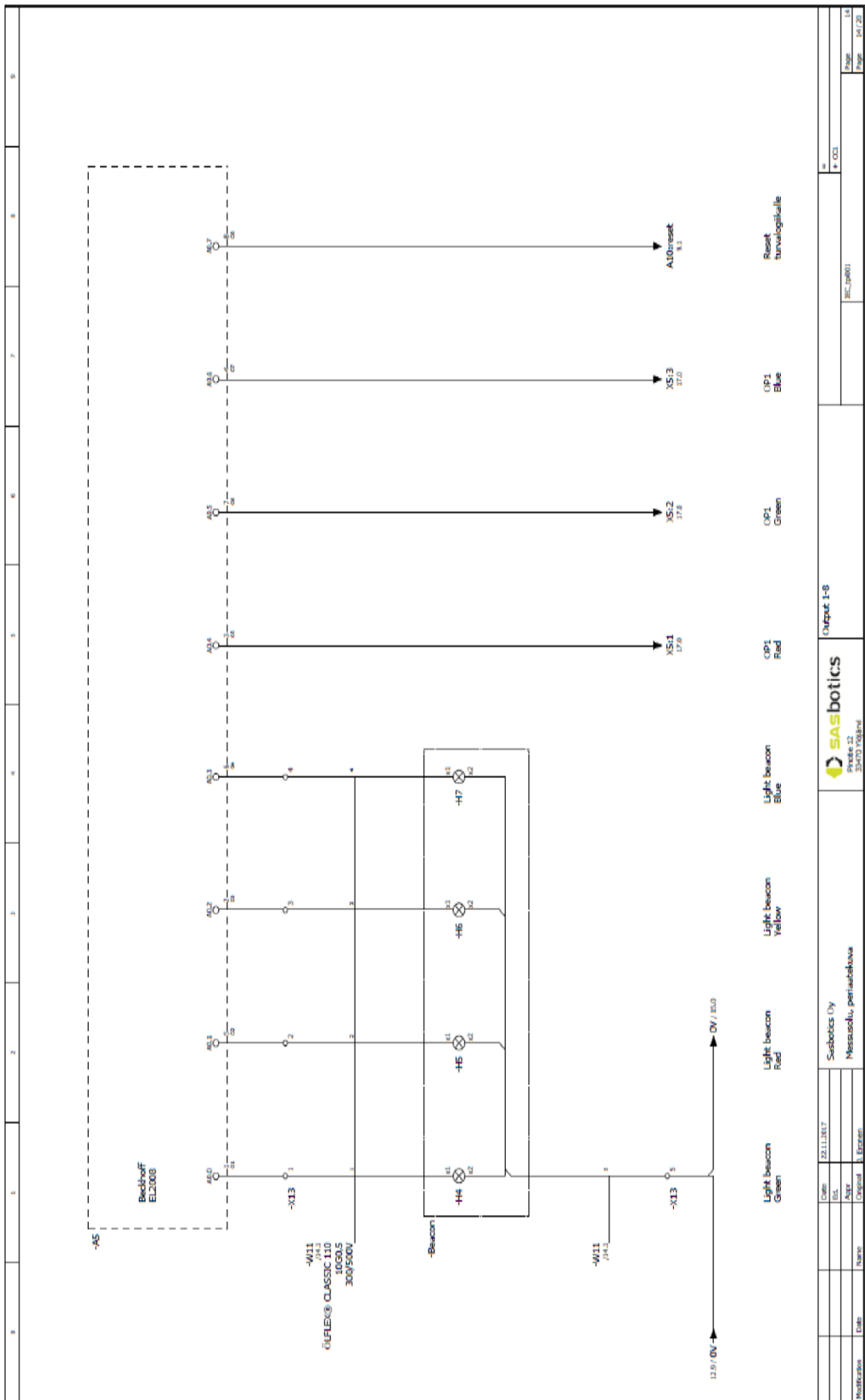
Modifikaatio	Date	22.11.2017	Sasbotics Oy	Input 1-8	REC:0001	Page 11/20
	Ulk.		Metsäsuomi, periaatekuvat			
	Nimi					
	Osasto					
	Eritys					



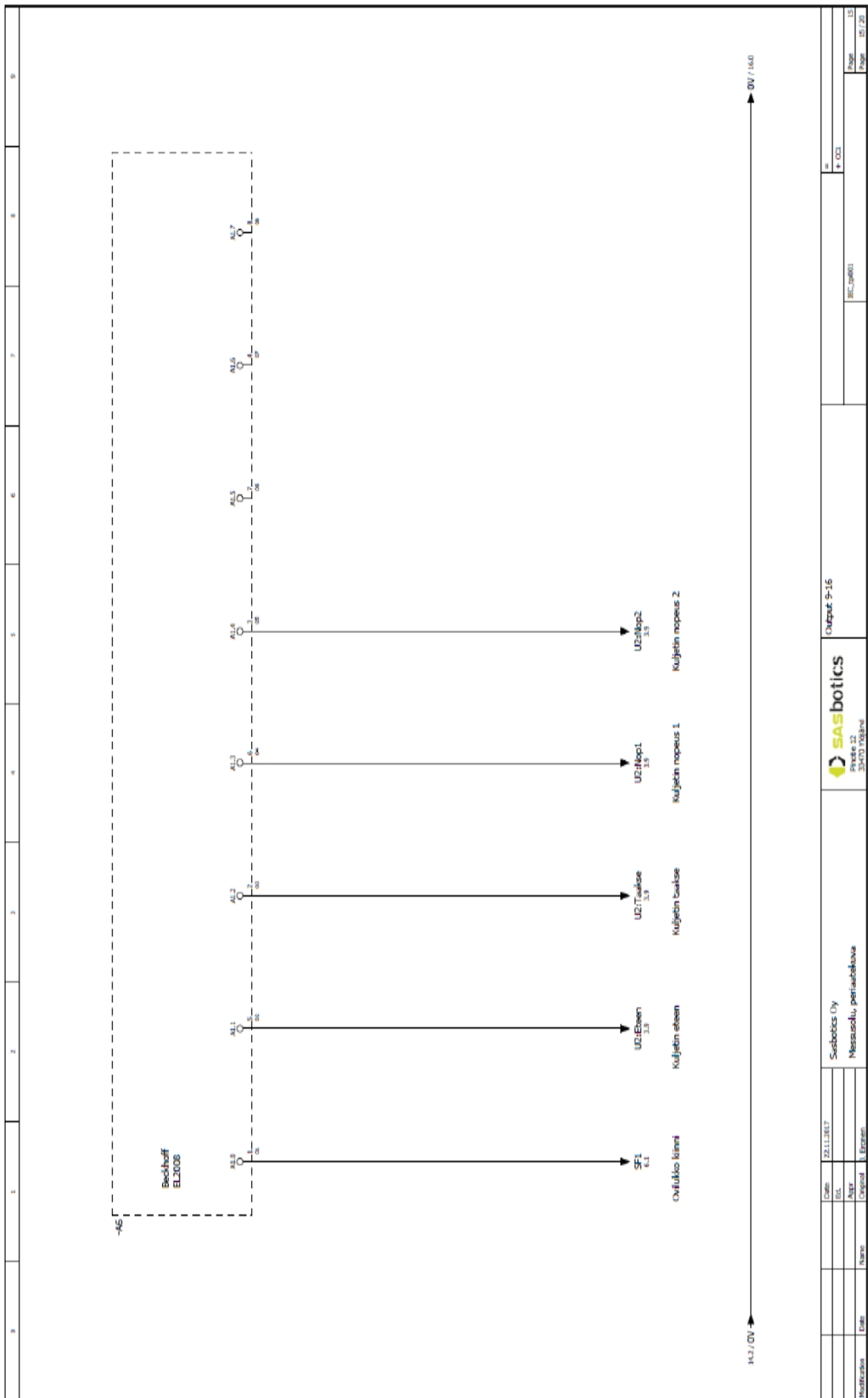
Drawn	S. L. 2018
Rev.	
Appr.	
Date	
Name	
Original	1. Eronen
 Sasbotics Oy Messausloja, perustajahuone	
Input 9-16	
REC-00000	
Page	12
Page	12 / 20



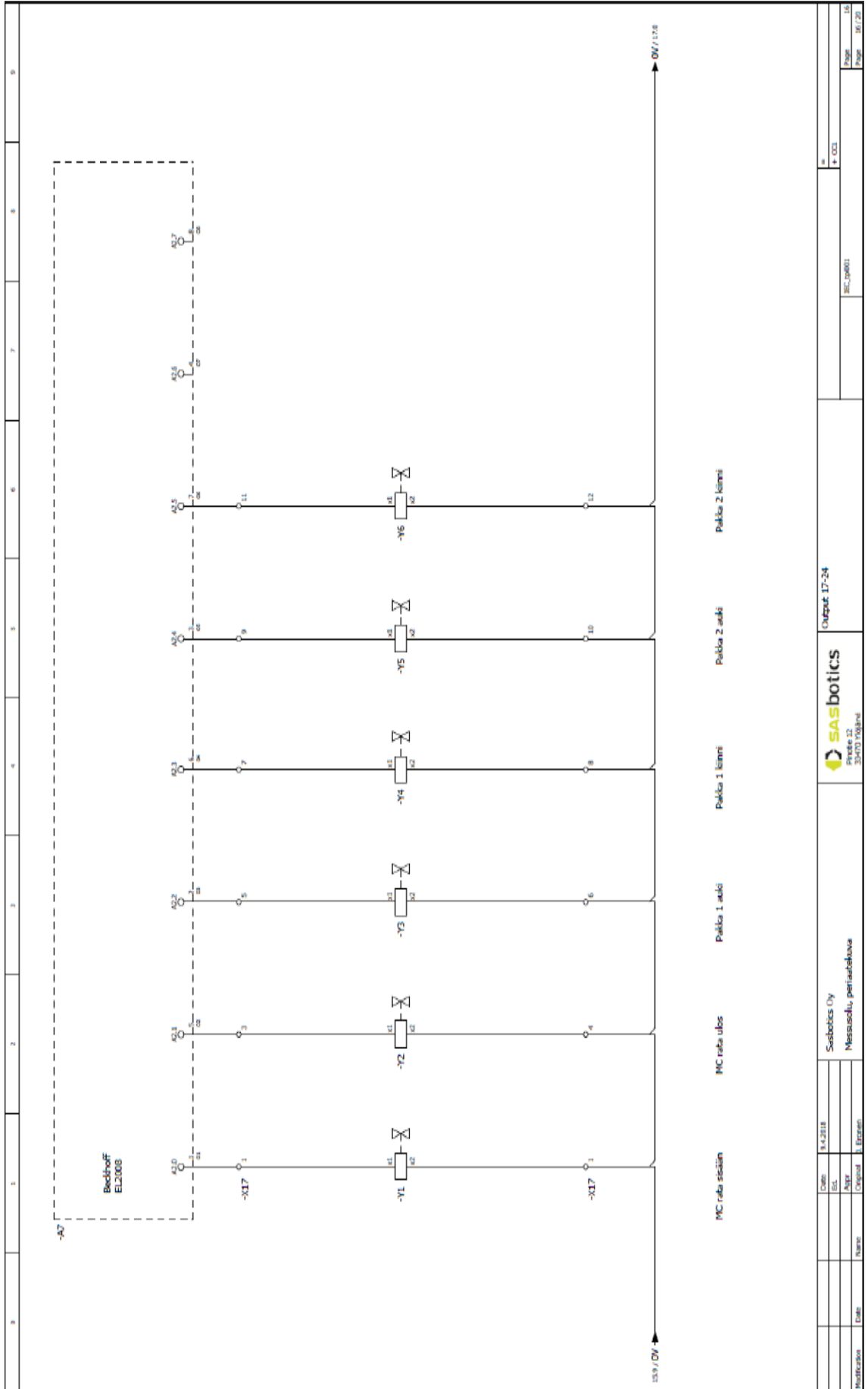
<table border="1"> <tr> <td>Date</td> <td>5.4.2018</td> </tr> <tr> <td>By</td> <td>Alper</td> </tr> <tr> <td>Original</td> <td><input type="checkbox"/> Eriksen</td> </tr> </table>	Date	5.4.2018	By	Alper	Original	<input type="checkbox"/> Eriksen	<p>Sasbotics Oy Metsäsuola, perusteluaine</p>	<p>Input 17-34</p>	<table border="1"> <tr> <td>Page</td> <td>13</td> </tr> <tr> <td>Page</td> <td>13 / 20</td> </tr> </table>	Page	13	Page	13 / 20
Date	5.4.2018												
By	Alper												
Original	<input type="checkbox"/> Eriksen												
Page	13												
Page	13 / 20												
<p>Modifications</p>		<p>SEC_00000</p>	<p>Page</p>										



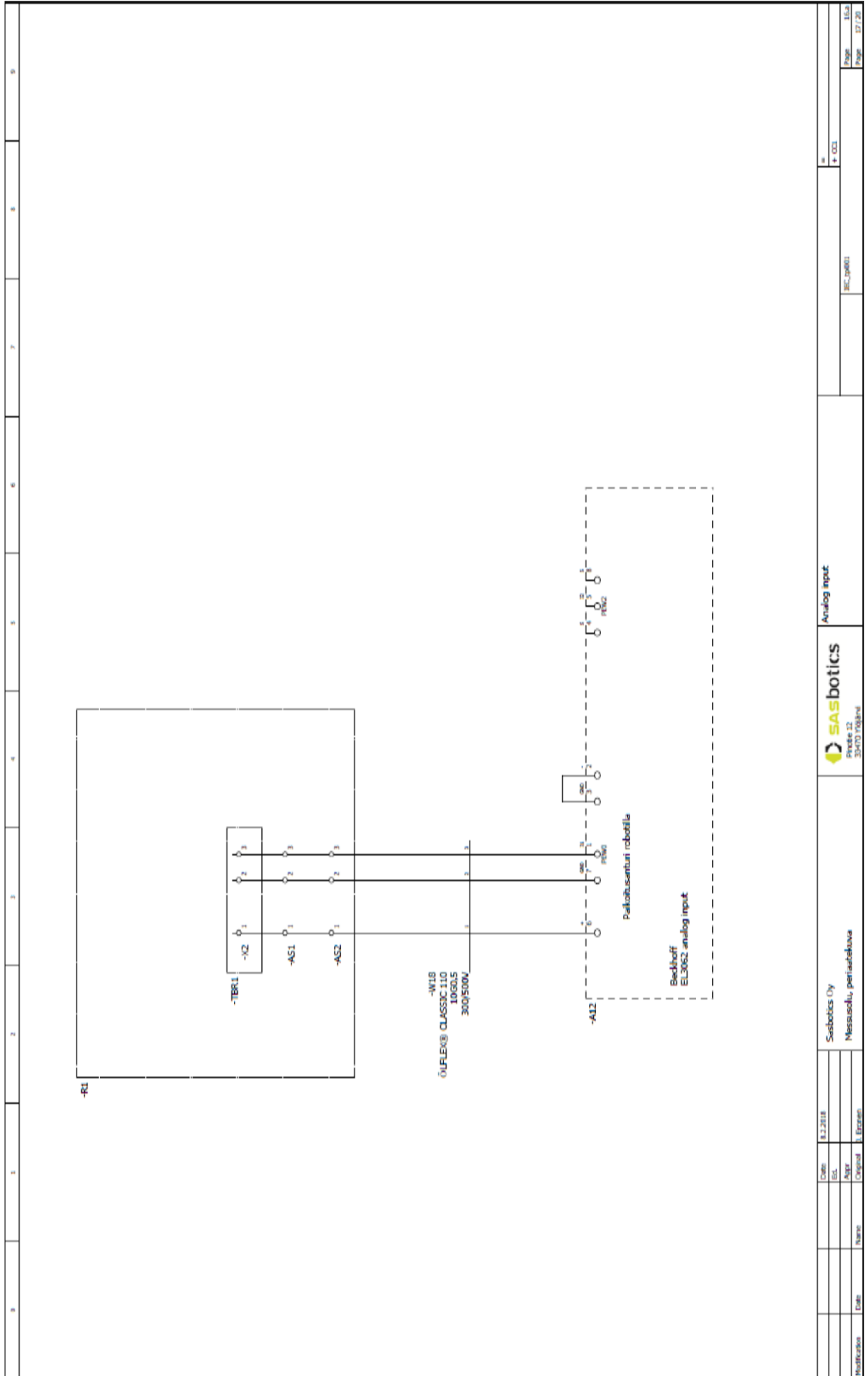
Modification	Date	Name	Original	1. Ertanen
	12.11.2017	Sasbotics Oy		
		Messuoliu, perisatekova		
			Output 1-8	
			REC_0001	
			Page	14 / 20
			Page	14 / 20



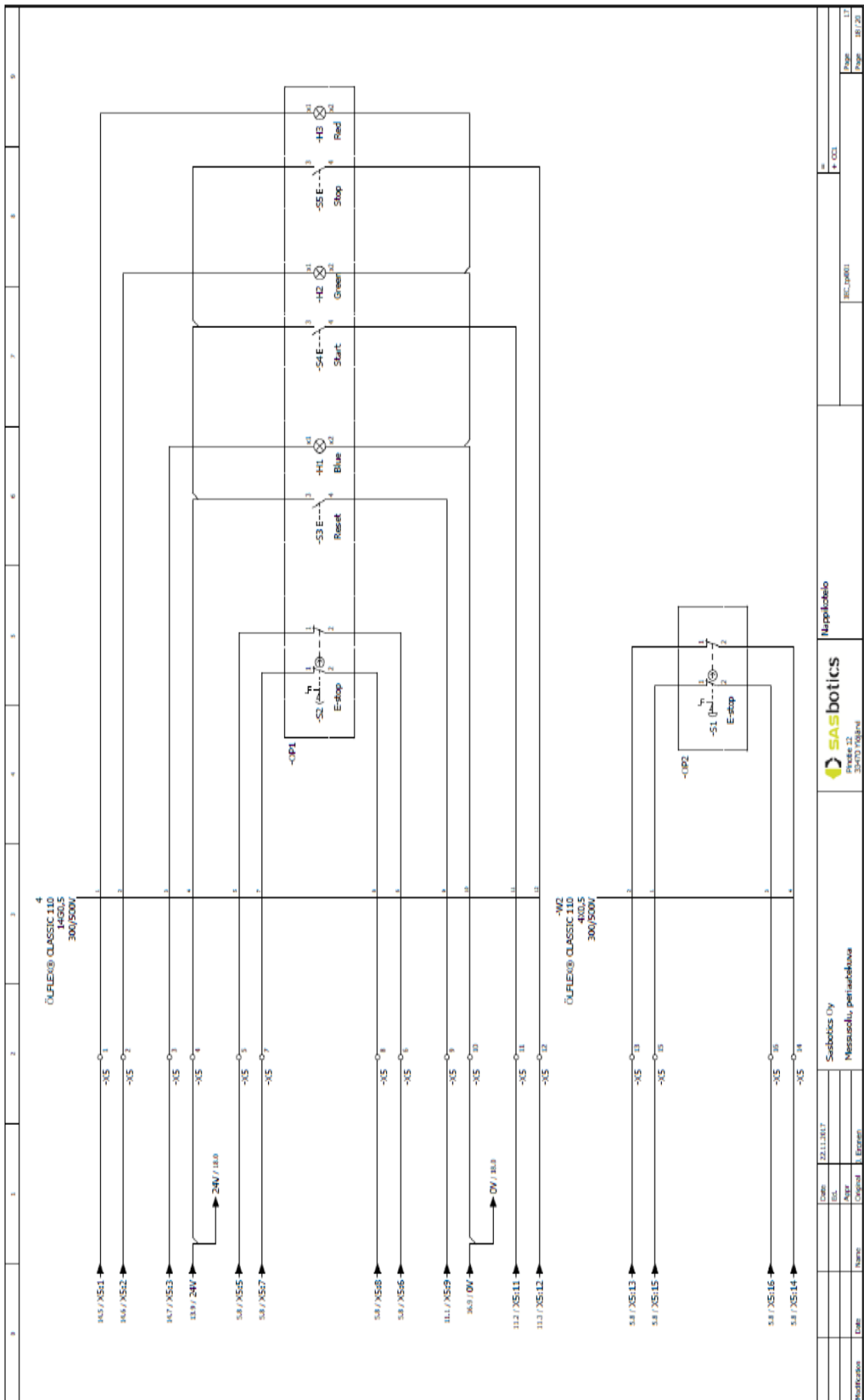
Modifikaatio	Date	Name	Version	Project	Page
				Output 9-16	15 / 20
				Page 15 / 20	



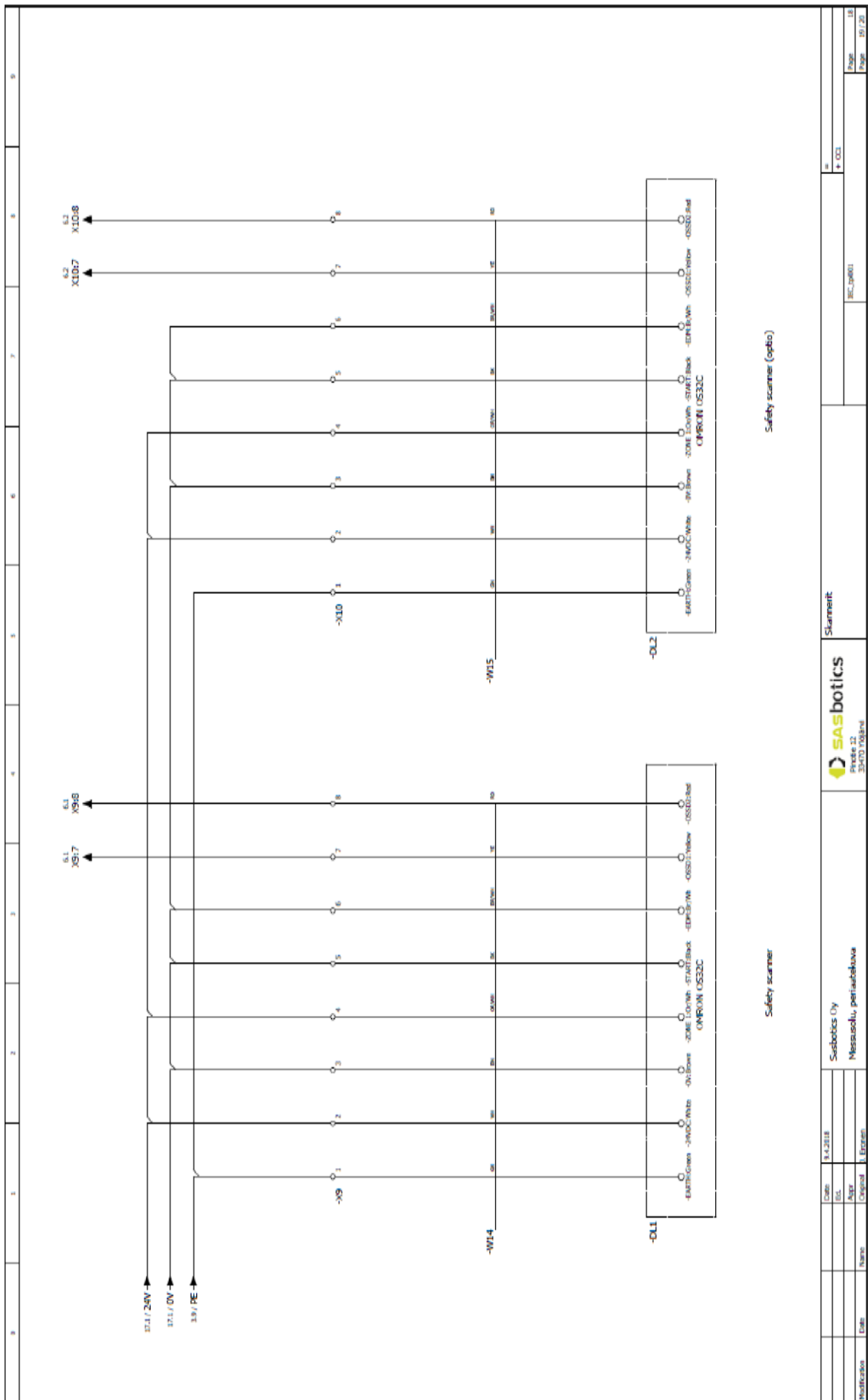
Revisio	Date	3.1.2018	Sasbotics Oy Metsänselän, peräntaivokkava	Output 17-34	KCC	Page 18 / 22
Author	Yrj					
Checker	Ermari					
Project	Name					



Modifikaatio	Date	Name	Version	Erityiset	Sasbotics Oy Metsäsoitu, perinteetoue	Analog input	EL3062	Page Page	17/20
--------------	------	------	---------	-----------	------------------------------------------	--------------	--------	--------------	-------



Modifikaatio	Itämeri	22.11.2017	Sasbotics Oy	Nurpikalela	Page 18 / 23
	Copyright		Messu- ja palvelutekniikka	REC_18001	



Modifikaatio	Date	11.1.2018	Sasbotics Oy	Skannarit	REC_0001	Page	19 / 23
	Date	01.01.2018	Sasbotics Oy			Page	19 / 23
	Author	Jyri	Määräsuoju, periaatekuva				
	Checked	Eero					

