



Teollisuuskeskuksen kunnostaminen

Joni Aivio

Opinnäytetyö, AMK

Kesäkuu 2025

Insinööri (AMK), Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Aivio, Joni

Teollisuuskeskuksen kunnostaminen

Jyväskylä: Jyväskylän ammattikorkeakoulu. Kesäkuu 2025, 75 sivua

Sähkö- ja automaatiotekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Opinnäytetyön tavoitteena oli toimeksiantajana toimineen Jyväskylän ammattikorkeakoulun sähkötekniikan laboratoriotiloissa sijaitsevan teollisuuskeskuksen kunnostaminen. Tavoite koostui keskuksen nykytilan selvittämisestä, käyttäjäkokemuksen parantamisesta sekä pohjan luomisesta jatkokehitysmahdollisuuksille. Teollisuuskeskusta ei ollut voitu hyödyntää täysimääräisesti osana opetusta siinä ilmenneiden puutteiden vuoksi. Toimeksiantajalla oli tarve saada opetuslaitteistona toimiva teollisuuskeskus takaisin opetuskäyttöön koko sen tarjoamalla laajuudella.

Opinnäytetyö toteutettiin tutkimuksellisena kehittämistyönä, joka aloitettiin laitteistoon ja sen dokumentointiin perehtymisellä. Puutteellisen dokumentoinnin ja vanhentuneen laitekannan vuoksi päädyttiin tuotamaan uudet dokumentit ja uusimaan vanhentuneet laitteet sekä suunnittelemaan uusi sovellusohjelma. Sovellusohjelmaa luotiin parhaan saatavilla olevan tiedon pohjalta, ohjelman toimivuutta testattiin laitteiston avulla ja muokattiin toimimaan halutulla tavalla.

Työn tuloksina saatiin ajantasainen dokumentointi teollisuuskeskuksesta, toimiva ja helppokäyttöinen sovellusohjelma ja käyttöliittymä, sekä hyvä pohja keskuksen jatkokehitykselle. Keskuksen nykytila oli alkuselvityksen perusteella oletettua huonompi, joka laajensi tehtävää työtä. Käyttäjäkokemuksen parantamiseksi järjestelmän käyttöä yksinkertaistettiin sovellus- ja käyttöliittymäsuunnittelulla. Lisäksi tulevaisuuden kehityshankkeita helpottamaan kerättiin keskuksen liittyvä materiaali talteen useaan paikkaan.

Suuria haasteita opinnäytetyössä aiheuttivat iäkkästä laitekannasta johtuvat yllättävät laiterikot, jotka hidastivat työn suorittamista ja toivat lisäselvitystä vaativia uusia vikoja. Viallisten laitteiden aiheuttamat viat saatiin kuitenkin selvitettyä ja teollisuuskeskus järjestelmänä toimimaan toimeksiantajaa tyydyttävällä tasolla. Kehitysehdotuksina esitettiin dokumentoinnin versionhallinnan ajan tasalla pitämisen lisäksi työasemaa keskuksen yhteyteen helpottamaan laboratoriotöiden suorittamisen lisäksi tulevaa kehitystyötä.

Avainsanat (asiasanat)

TIA Portal, sovellusohjelmointi, oikosulkumoottori, momentti

Muut tiedot (salassa pidettävät liitteet)

-

Aivio, Joni

Renovation of the industrial center

Jyväskylä: JAMK University of Applied Sciences, June 2025, 75 pages

Degree Programme in Electrical and Automation Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The aim of the thesis was to renovate the industrial center located in the electrical engineering laboratory premises of the Jyväskylä University of Applied Sciences, which was the client. The aim consisted of investigating the current state of the center, improving the user experience and creating a basis for further development opportunities. The industrial center had not been able to be fully utilised as part of teaching due to the shortcomings it had encountered. The client needed to get the industrial center, which functions as a teaching facility, back into teaching use to the full extent it offers.

The thesis was carried out as a research development work, which began with familiarization with the hardware and its documentation. Due to the insufficient documentation and outdated equipment, it was decided to produce new documents and renew outdated equipment, as well as design a new application program. The application program was created based on the best available information, the functionality of the program was tested using the hardware and modified to work in the desired way.

The results of the work included up-to-date documentation of the industrial center, a functional and easy-to-use application program and user interface, and a good basis for the center's further development. The current state of the center was worse than expected based on the initial survey, which expanded the work to be done. To improve the user experience, the use of the system was simplified through application and user interface design. In addition, to facilitate future development projects, material related to the center was saved in several places.

The thesis presented major challenges due to unexpected equipment failures due to the aging equipment, which slowed down the work and introduced new defects requiring further investigation. However, the defects caused by the faulty equipment were resolved and the industrial center system was able to function at a level satisfactory to the client. Development proposals included keeping the documentation version control up to date, as well as a workstation connected to the center to facilitate future development work in addition to laboratory work.

Keywords/tags (subjects)

TIA Portal, application programming, induction motor, torque

Miscellaneous (Confidential information)

-

Sisältö

| | | |
|----------|--|-----------|
| 1 | Johdanto | 5 |
| 1.1 | Työn tausta..... | 5 |
| 1.2 | Tavoitteet | 5 |
| 1.3 | Aiheen rajaus..... | 6 |
| 2 | TIA Portal | 6 |
| 2.1 | TIA Portal yleisesti | 6 |
| 2.2 | Toimintolohko | 6 |
| 2.3 | Toiminto | 7 |
| 2.4 | Toimintolohkokaavio..... | 7 |
| 2.5 | Strukturoitu ohjauskieli..... | 7 |
| 3 | Kenttäväylä | 8 |
| 3.1 | Kenttäväylä yleisesti..... | 8 |
| 3.2 | Profibus DP..... | 8 |
| 4 | ABB ACSM1-04 ja ABB ACSM1-204 | 9 |
| 4.1 | Suora momentinsäätö (DTC) | 10 |
| 4.2 | DriveStudio & DriveSPC..... | 10 |
| 5 | Oikosulkumoottori | 12 |
| 5.1 | Oikosulkumoottori yleisesti | 12 |
| 5.2 | Suora käynnistys (DOL) ja suunnanvaihto..... | 13 |
| 5.3 | Tähtikolmiokäynnistys..... | 14 |
| 5.4 | Pehmokäynnistin..... | 16 |
| 5.5 | ABB:n älykkäät moottorinohjaimet..... | 16 |
| 5.6 | Taajuusmuuttaja | 17 |
| 6 | Kuormitustyytit | 18 |
| 6.1 | Vakiomomentti..... | 18 |
| 6.2 | Neliöllinen momentti | 19 |
| 6.3 | Vakioteho | 20 |
| 7 | Teollisuuskeskus | 21 |
| 8 | Toteutus | 24 |
| 8.1 | Tutkimuksellinen kehittämistyö..... | 24 |
| 8.2 | Tutkimuskysymykset | 25 |
| 8.3 | Eettiset periaatteet | 25 |
| 8.4 | Projektin alkukartoitus..... | 26 |

| | | |
|-----------|--|-----------|
| 8.5 | Ohjelmoitava logiikka ja käyttöliittymäpaneeli | 26 |
| 8.6 | Sovellusohjelmointi | 27 |
| 8.7 | Käyttöliittymäsuunnittelu | 29 |
| 9 | Käyttöönotto | 29 |
| 9.1 | Laitteiden asennus ja kytkentämuutokset | 30 |
| 9.2 | Suorien moottorilähtöjen testaus | 32 |
| 9.3 | Väylän pystytys ja vianetsintä | 35 |
| 9.4 | Väylälaitteiden testaus | 37 |
| 9.5 | Kuormitusohjelmien testaus | 39 |
| 9.5.1 | Vakiomomenttiohjeen testaus | 41 |
| 9.5.2 | Neliöllisen momenttiohjeen testaus | 43 |
| 9.5.3 | Vakioteho momenttiohjeen testaus | 45 |
| 9.5.4 | Järjestelmän testaus | 46 |
| 10 | Pohdinta | 47 |
| | Lähteet | 50 |
| | Liitteet | 52 |
| | Liite 1. Käyttöliittymäpaneelin kotisivunäkymä | 52 |
| | Liite 2. Käyttöliittymäpaneelin ML1-3 sivunäkymä | 53 |
| | Liite 3. Käyttöliittymäpaneelin ML4 sivunäkymä | 54 |
| | Liite 4. Käyttöliittymäpaneelin ML5 sivunäkymä | 55 |
| | Liite 5. Kuormitustyyppin valinta koodissa | 56 |
| | Liite 6. Jännitteenjakelu 400V & 230V | 57 |
| | Liite 7. Jännitteenjakelu 24V | 58 |
| | Liite 8. Moottorin syöttö & virranmittaus | 59 |
| | Liite 9. Jännitteenmittaus | 60 |
| | Liite 10. Tähtikolmiokäynnistin | 61 |
| | Liite 11. Suunnanvaihtolähtö | 62 |
| | Liite 11. Pehmokäynnistin | 63 |
| | Liite 12. Älykäs moottorinohjain | 64 |
| | Liite 13. Taajuusmuuttajalähtö | 65 |
| | Liite 14. Generaattorin pääpiiri | 66 |
| | Liite 15. Generaattorin ohjauspiiri | 67 |
| | Liite 16. Tulopiirikaavio DI0.0–0.7 | 68 |
| | Liite 17. Tulopiirikaavio DI1.0–1.5 | 69 |
| | Liite 18. Tulopiirikaavio DI8.0–8.7 | 70 |

| | |
|--|----|
| Liite 19. Lähtöpiirikaavio DO0.0–0.7 | 71 |
| Liite 20. Lähtöpiirikaavio DO1.0–1.1 | 72 |
| Liite 21. Lähtöpiirikaavio DO8.0–8.7 | 73 |
| Liite 22. Piirikaavio analogiatulot | 74 |
| Liite 23. Väyläkaavio..... | 75 |

Kuviot

| | |
|---|----|
| Kuvio 1. Function Block Diagram (FBD). (Simatic S7-1200 Programmable controller 2012) | 7 |
| Kuvio 2. Esimerkki strukturoidun ohjauskielen kirjoitusasusta. (Simatic S7-1200 Programmable controller 2012) | 8 |
| Kuvio 3. Kenttäväylätologioita. (What is fieldbus? 2019)..... | 8 |
| Kuvio 4. Profibus DP topologia. (Chabauty n.d.)..... | 9 |
| Kuvio 5. Suoran momentinsäädön täydellinen lohkokaaavio (Tekninen opas Nro1 2001, 26) ... | 10 |
| Kuvio 6. DriveStudio parametrivalikko. (DriveStudio User manual 2012)..... | 11 |
| Kuvio 7. DriveStudio monitorointinäyttö. (DriveStudio User manual 2012)..... | 11 |
| Kuvio 8. DriveSPC sovellusohjelmapiirros..... | 12 |
| Kuvio 9. Suora käynnistys. (Pehmokäynnistinopas 2011, 13) | 14 |
| Kuvio 10. Suoran käynnistuksen momentti- ja virtakäyrä. (Pehmokäynnistinopas 2011, 13) ... | 14 |
| Kuvio 11. Tähtikolmiokäynnistin. (Pehmokäynnistinopas 2011, 14)..... | 15 |
| Kuvio 12. Tähtikolmiokäynnistimen momentti- ja virtakäyrät. (Pehmokäynnistinopas 2011, 15)15 | |
| Kuvio 13. Tähtikolmiokäynnistimen voimansiirto- ja virtahuiput pumppukäytössä. (Pehmokäynnistinopas 2011, 15)..... | 15 |
| Kuvio 14. Pehmokäynnistin. (Pehmokäynnistinopas 2011, 18)..... | 16 |
| Kuvio 15. UMC100.3 älykäs moottorinohjain. (Universal Motor Controller 100.3 n.d.)..... | 17 |
| Kuvio 16. Taajuusmuuttaja. (Pehmokäynnistinopas 2011, 17) | 18 |
| Kuvio 17. Vakiomomenttisovelluksen tyypilliset teho- ja momenttikäyrät. (Tekninen opas Nro7 2001, 20) | 19 |
| Kuvio 18. Neliöllisen momentin sovelluksen tyypilliset momentti- ja tehokäyrät. (Tekninen opas Nro7 2001, 20) | 20 |
| Kuvio 19. Vakiotehosovelluksen tyypilliset teho- ja momenttikäyrät. (Tekninen opas Nro7 2001, 21) | 21 |
| Kuvio 20. Moottori ja generaattori | 22 |
| Kuvio 21. Moottorilähdöt..... | 22 |
| Kuvio 22. Kuormamoottorin taajuusmuuttaja..... | 23 |
| Kuvio 23. Uusi ohjelmoitava logiikka | 31 |

| | |
|--|----|
| Kuvio 24. Uusi käyttöliittymäpaneeli edestä | 31 |
| Kuvio 25. Uusi käyttöliittymäpaneeli takaa | 32 |
| Kuvio 26. Moottorilähtöjen lukitukset..... | 34 |
| Kuvio 27. Tähtikolmiolähtö ohjelmaloikka | 34 |
| Kuvio 28. Suunnanvaihtolähtö ohjelmaloikka..... | 35 |
| Kuvio 29. Pehmokäynnistinlähtö ohjelmaloikka | 35 |
| Kuvio 30. UMC22-FBP kenttäväyläliitin (FBP FieldBusPlug 2008) | 37 |
| Kuvio 31. Älykäs moottorinohjain ohjelmaloikka..... | 38 |
| Kuvio 32. Taajuusmuuttajalähdön ohjelmaloikka | 39 |
| Kuvio 33. Kuormitusohjelmien ohjelmaloikka..... | 40 |
| Kuvio 34. Kuormitusohjelmien käynnistys ehdot | 41 |
| Kuvio 35. Momenttisäädön ohjausketju. (ACSM1 Motion Control-ohjelma 2015). | 43 |
| Kuvio 36. Momenttiohjeen asetusarvon etumerkin hallinta..... | 45 |
| Kuvio 37. Vakioteho momenttiohjeen laskenta. | 46 |

1 Johdanto

1.1 Työn tausta

Jyväskylän ammattikorkeakoulun sähkötekniikan laboratoriossa sijaitsee sähkökeskus, joka antaa esimerkin teollisuuden moottorikeskuksesta. Keskuksella voidaan käytännön harjoituksia teke- mällä tutustua monipuolisesti erilaisiin oikosulkumoottorin käynnistystapoihin sekä oikosulku- moottorin käyttäytymiseen erilaisten kuormitustyyppien alaisena. Laitteistolla tehtyjen harjoitus- ja laboratoriotöiden avulla opiskelijalle havainnollistuu oikosulkumoottorin oikeanlaisen käynnis- tystavan valinnan tärkeys, moottorin kuormitustyyppin mukaan. Opetusympäristönä keskuksen ar- voa lisää toiminnallisten harjoitusten lisäksi sen kouluympäristöön laaja laitemäärä. Keskuksen li- sähyöty opetusympäristönä toiminnallisten harjoitusten lisäksi, on kouluympäristöön laaja laitemäärä. Tämän seurauksena päästään tutkimaan suhteellisen suurta määrää piirikaavioita ja tunnistamaan ja paikantamaan fyysisesti kaavioissa esiintyviä laitteita.

Teollisuuskeskuksen opetuksellinen arvo oli tunnistettu suureksi, mutta sitä oli voitu hyödyntää opetuksessa vain rajallisesti, sillä keskuksessa oli ilmennyt sen käyttöä rajoittavia vikoja sekä kehi- tyskohteita. Koska keskus oli rakennettu 2010-luvun alkupuolella, alkoi myös laitekanta olla jo iä- käästä ja laitetuki oli loppunut ohjelmoitavasta logiikasta ja käyttöliittymäpaneelistä. Myöskään pii- rikaavioiden versionhallinta ei ollut pysynyt vuosien saatossa ajan tasalla. Keskuksen koko potentiaali haluttiin päästä hyödyntämään opetuksessa, joten sen kehitystarve tuli toimeksianta- jalle ajankohtaiseksi.

1.2 Tavoitteet

Toimeksiantajan nykytilanne oli, ettei teollisuuskeskusta voitu hyödyntää opetuksessa siinä laajuu- dessa kuin toivottiin. Vaikka keskusta oli käytetty säännöllisen epäsäännöllisesti opetuksessa, siinä määrin kuin se oli ollut mahdollista keskuksessa ilmenneiden ongelmien vuoksi, ei toimeksiantaja- kaan tuntenut laitteistoa tai sen nykytilaa kovin syvällisesti. Tiedettiin siis keskuksessa olevan vi- koja ja kehityskohteita, mutta niiden täydestä laajuudesta ei ollut täyttä varmuutta. Alkutietojen perusteella pystyttiin määrittämään selkeäksi tarpeeksi vain ohjelmoitavan logiikan ja käyttöliitty- mäpaneelin uusiminen, sekä kenttäväylän vianetsintä.

1.3 Aiheen rajaus

Opinnäytetyön selkeä rajaaminen osoittautui vaikeaksi. Toimeksiantaja esitti kohtuullisen vähimmäisvaatimuksen, että ohjelmoitava logiikka ja käyttöliittymäpaneeli vaihdetaan ja kenttäväylän vikatila pyritään selvittämään. Toimeksiantajan rajaus perustui siihen, että nämä olivat selkeät, määriteltävissä olevat tarpeet. Kaikki keskukselle tehty kartoitus- ja kehitystyö olisi kuitenkin parannusta nykytilanteeseen, ja edistäisi keskuksen kehitystyön jatkoa. Opinnäytetyössä jouduttiin kuitenkin pohtimaan kriittisesti, puutteellisen ja mahdollisesti edelleen toimimattoman järjestelmän luovuttamista toimeksiantajalle, ja kuinka tämä on ristiriidassa ammattietiikan kanssa. Päätettiin siis pyrkiä siihen, että teollisuuskeskus saatetaan kokonaisuudessaan toimintakuntoon ja dokumentointi ajantasaistetaan, jotta toimeksiantaja pääsee laitteistoa hyödyntämään.

2 TIA Portal

2.1 TIA Portal yleisesti

Siemensin ohjelmointityökalu Totally Integrated Automation Portalin tarkoituksena on mahdollistaa kaikki automaatiosovellussuunnittelussa tarvittavat toiminnallisuudet yhdellä työkalulla. TIA Portalista löytyy esimerkiksi sovellussuunnitteluun Simatic Step 7, valvomo- ja ajonäyttöjen suunnitteluun Simatic WinCC, taajuusmuuttajien parametrisointiin Simatic Startdrive. Näiden lisäksi TIA Portal sisältää liikkeenohjaukseen, turvallisuuteen ja moniin lisätoiminnallisuuksiin liittyviä työkaluja. (Totally Integrated Automation Portal – Always ready for tomorrow n.d.).

2.2 Toimintolohko

Function Block eli toimintolohko on koodilohko, joka toteuttaa ohjelmoitua toiminnallisuutta, sekä tallentaa syklisesti esimerkiksi siihen liitettyjen muuttujien arvoja eli dataa. Tallennus suoritetaan toimintolohkon kutsumisen yhteydessä instanssidatalohkoon, jossa toimintolohkon muuttujien arvot, hetkellisiä muuttujia lukuun ottamatta, säilyvät toimintolohkon suorittamisen jälkeenkin. Toimintolohkojen sisälle voidaan luoda aliohjelmaa toimintoilla ja toisilla toimintolohkoilla ja sitä voidaan kutsua useissa kohdissa ohjelmaa, useita kertoja, joten ohjelman rakenne voidaan muodostaa toimintolohkoilla. (Simatic S7-1200 Programmable controller 2012).

2.3 Toiminto

Function eli toiminto on koodilohko, joka ei tallenna muuttujiensa arvoja instanssidatalohkoon. Haluttaessa tallentaa dataa toimintoa käytettäessä, on käytettävä globaalia datalohkoa, johon data tallennetaan tai kutsuttava toiminnolla toimintolohkoa, joka tallentaa muuttujadatan omaan instanssidatalohkoonsa. Toiminto soveltuu hyvin käytettäväksi esimerkiksi matemaattisiin laskutoimituksiin, sillä sen avulla toteutetut laskuoperaatiot eivät aiheuta ohjelmoitavalle logiikalle tallentamisesta aiheutuvaa rasitusta. (Simatic S7-1200 Programmable controller 2012).

2.4 Toimintolohkokaavio

Function Block Diagram eli toimintolohkokaavio on Boolean algebrassa esiintyviin loogisiin symboleihin pohjautuva graafinen ohjelmointikieli (Simatic S7-1200 Programmable controller 2012). Toimintolohkokaavioita on helpohko ymmärtää ilman koodausosaamista, sillä se pohjautuu loogisiin operaatioihin ja graafinen esitystapa auttaa hahmottamaan asioiden yhteyksiä ja sidonnaisuuksia koodissa (kuvio 1).



Kuvio 1. Function Block Diagram (FBD). (Simatic S7-1200 Programmable controller 2012)

2.5 Strukturoitu ohjauskieli

Structured Control Language eli strukturoitu ohjauskieli (kuvio 2) on Siemensin S7-sarjan ohjelmoitavissa logiikoissa käytettävä ohjelmointikieli. Strukturoidun ohjauskielen etu verrattuna toimintolohkokaaviolla toteutettuun koodiin on sen kompakti esitystapa ja suoraviivaiset operaatiot, lisäksi strukturoitua ohjauskieltä voi kirjoittaa millä tahansa tekstieditorilla ja viedä tämän tiedoston TIA Portal projektiin käytettäväksi. Strukturoidulla ohjauskielellä koodit, jotka muuten vaatisivat laajoja ja monimutkaisia toimintolohkokaavioita, saadaan tiiviisti toteutettua suoraviivaisilla koodiriveillä. (Simatic S7-1200 Programmable controller 2012)

```

"C" := #A+#B;
"Data_block_1".Tag := #A;
IF #A > #B THEN "C" := #A;
"C" := SQRT (SQR (#A) + SQR (#B));

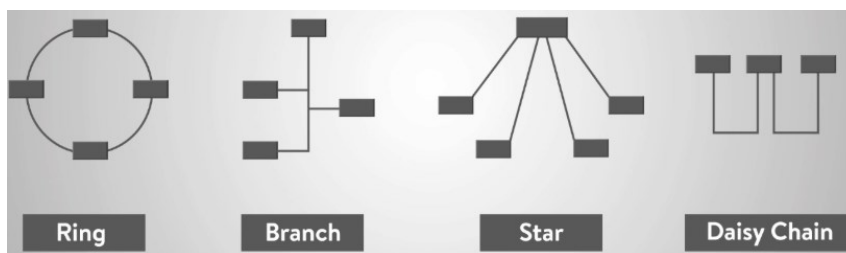
```

Kuvio 2. Esimerkki strukturoidun ohjauskielen kirjoitusasusta. (Simatic S7-1200 Programmable controller 2012)

3 Kenttäväylä

3.1 Kenttäväylä yleisesti

Kenttäväylällä tarkoitetaan IEC 61158 standardoituja protokollia. Kenttäväyläprotokollia ovat esimerkiksi ControlNet, Modbus, Profibus ja EtherCAT. Ennen kenttäväylien kehittämistä laitteiden yhdistämiseen tiedonsiirtoa varten käytettiin RS232 sarjaväylää, joka mahdollisti vain kahden laitteen välisen kommunikaation. Kenttäväylien myötä kenttälaitteita voitiin alkaa kytkeä yhteen erilaisin topologioin (kuvio 3) ja kaapeloinnin määrä saatiin vähenemään merkittävästi, kun yhdellä väyläkaapelilla voidaan yhdistää kentällä olevat laitteet esimerkiksi sähkötilassa sijaitsevaan ohjelmoitavaan logiikkaan. (What is Fieldbus? 2019.)

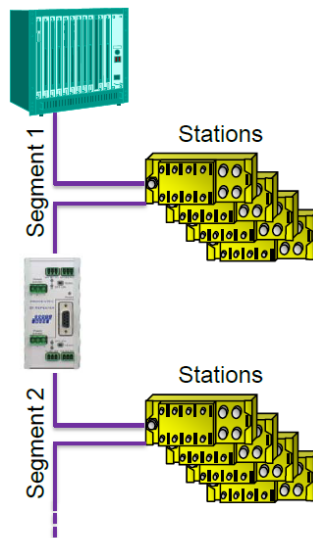


Kuvio 3. Kenttäväylätopologioita. (What is fieldbus? 2019)

3.2 Profibus DP

Opinnäytetyössä käytettiin Profibus DP:tä (Process Fieldbus Decentralized Peripherals) eli hajautettua I/O:n prosessikenttäväylää, jonka suosiota eniten käytetyksi kenttäväyläratkaisuksi on edesauttanut sen yhteensopivuus eri valmistajien laitteiden välillä, sekä käyttökelpoisuus haastavissakin kenttäväyläsovelluksissa. DP väylän fyysinen toteutus toimii RS485 standardilla, joka

mahdollistaa pitkätkin kaapelointimatkat ympäristöissä, joissa on sähköisiä häiriöitä, sekä useiden laitteiden viestinnän samassa väylässä. Profibus DP mahdollistaa tiedonsiirron nopeudella 9 kb/s – 12 Mb/s ja sen suurin sallittu kaapelipituus on 1200 m, tulee kuitenkin huomioida tiedonsiirtonopeuden laskevan kaapelipituuksien kasvaessa. Profibus toimii segmentteittäin (kuvio 4), joissa voi olla 31 osoitetta eli laitetta ja segmenttien määrää voidaan kasvattaa toistimilla siten, että laitemäärä voi olla 125. Segmentti tarkoittaa päättyvien kaapeleiden välistä väylän osaa. (Chabauty n.d.)



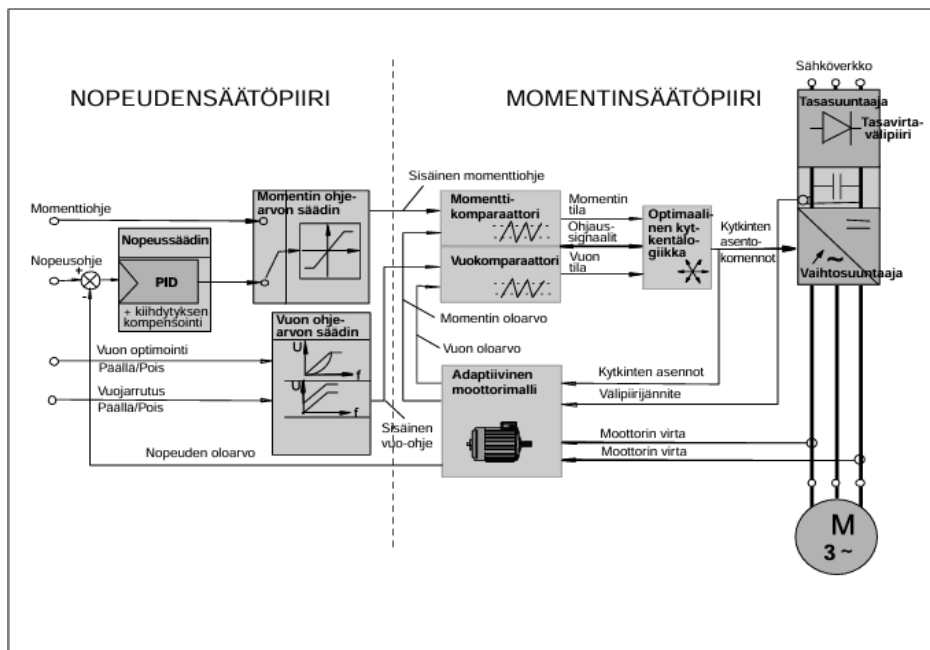
Kuvio 4. Profibus DP topologia. (Chabauty n.d.)

4 ABB ACSM1-04 ja ABB ACSM1-204

Teollisuuskeskuksessa, johon opinnäytetyö kohdistuu, olevat taajuusmuuttajat ovat ABB high performance machinery drive- tuoteperheeseen kuuluvia ACSM1-sarjan taajuusmuuttajia. Nämä taajuusmuuttajat on suunniteltu vaativien sovellusten liikkeenhallintaan sekä momentin- ja nopeudensäätöön. ACSM1-sarjan taajuusmuuttajissa on suora momentinsäätö (DTC) ja niillä voidaan ohjata erilaisia moottoreita takaisinkytkennällä tai ilman takaisinkytkentää. Verkkoon syöttävänä eli regeneroivana taajuusmuuttajana on ACSM1-204 taajuusmuuttaja, jonka kanssa käytetään WFU-suodinmoduulia verkkopuolen yliaaltojen hillitsemiseen. (ABB High performance machinery drives 2010, 4–8.)

4.1 Suora momentinsäätö (DTC)

Suora momentinsäätö (kuvio 5) on perinteiset takaisinkytketyt ja takaisinkytkemättömät pulssinleveyssäätö-käytöt korvaamaan kehitetty vaihtovirtakäyttötekniikka. Suoralla momentinsäädöllä ovat moottorin momentti ja staattorin vuo ensisijaisia säätösuureita, jotka tunnustusajon aikana mitattujen arvojen perusteella luodusta tarkasta moottorin mallinnuksesta saadaan. Tämä moottorimalliksi kutsuttu mallinnus lähettää todellisen momentin ja staattorin vuon ohjaussignaaleina, jonka lisäksi moottorimalli laskee akselin nopeuden. Suoran momentinsäädön vasteaika on kymmenesosa perinteiseen pulssinleveyssäätöön verrattuna eikä moottorimallin tarkkuuden vuoksi tarvita erillistä asentoanturia tai takometriä. (Tekninen opas Nro1 2001, 12, 22–23)

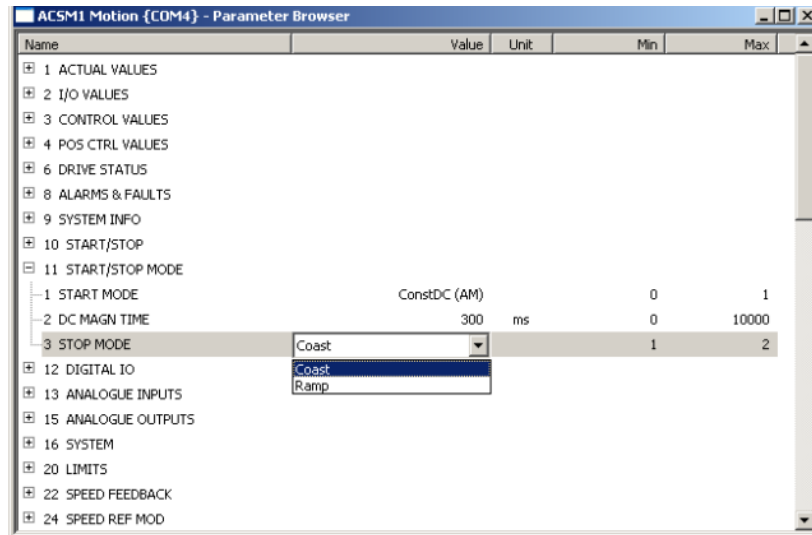


Kuvio 5. Suoran momentinsäädön täydellinen lohkokaavio (Tekninen opas Nro1 2001, 26)

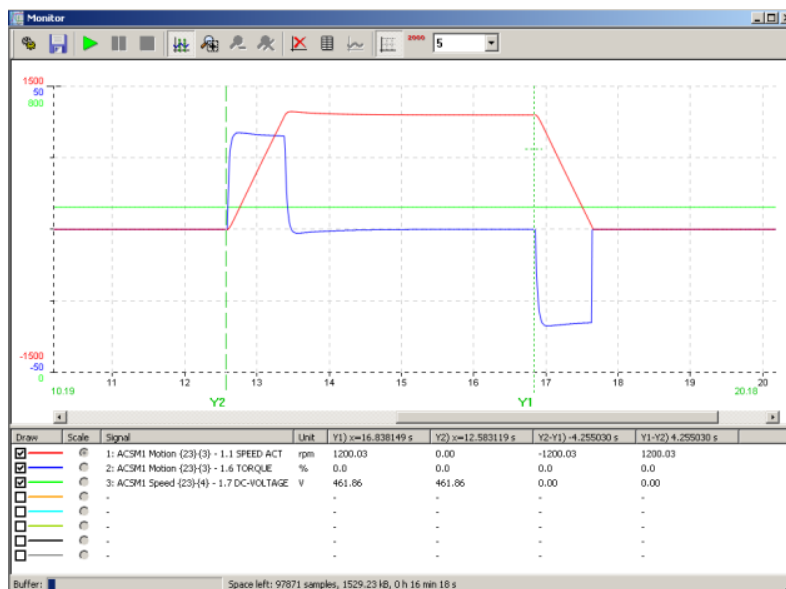
4.2 DriveStudio & DriveSPC

DriveStudio on ABB:n ACSM1-sarjan, sekä muiden saman tyyppisten ABB:n taajuusmuuttajien käyttöönottoon, parametointiin, hallintaan ja valvontaan käytettävä Windows pohjainen sovellus. Sen avulla taajuusmuuttajan hallinnointi on huomattavasti helpompaa kuin taajuusmuuttajan omalla käyttöliittymäpaneelilla, kun parametrit saadaan tietokoneen näytölle auki (kuvio 6), jolloin käyttäjä ei joudu selaamaan parametrejä yksittäin. Myös taajuusmuuttajan toiminnan seu-

raaminen haluttujen suureiden, kuten virran tai momentin mukaan, on DriveStudiolla yksinkertaista monitorointi näytön avulla (Kuvio 7), jolla voidaan piirtää kuvaajia näistä suureista. (DriveStudio User manual 2012, 6, 29 & 42.)



Kuvio 6. DriveStudio parametrivalikko. (DriveStudio User manual 2012)

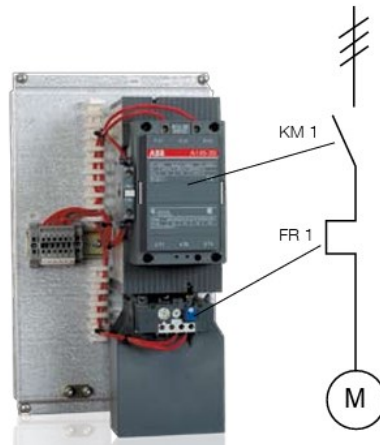


Kuvio 7. DriveStudio monitorointinäyttö. (DriveStudio User manual 2012)

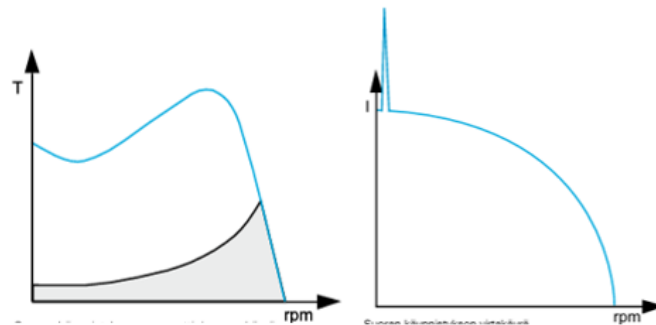
molemmat paketit koostuvat toisistaan eristetyistä ohuista sähkölevyistä. Oikosulkumoottorin toiminta perustuu virralliseen johtimeen magneettikentässä sekä sähkömagneettiseen induktioon. Staattorikämmityksen napaluku vaikuttaa staattorin magneettikentän pyörimisnopeuteen. Esimerkiksi 50Hz:n taajuudella kaksinapaiseksi käännetty moottori pyörii 3000 rpm, nelinapainen 1500 rpm ja kuusinapainen 1000 rpm. Staattorikämmityksen magneettikenttä siis pyörähtää syöttöjännitteen yhden jakson aikana kaksinapaisena yhden kierroksen, nelinapaisena puoli kierrosta ja kuusinapaisena kolmasosa kierroksen. Oikosulkumoottoria nimitetään myös epätahtimoottoriksi, joka tulee siitä, että roottori pyörii staattorin magneettikenttää hitaammin. Oikosulkumoottorin kuormitus vaikuttaa sen toimintaan siten, että kuormituksen kasvaessa roottorin pyörimisnopeus hidastuu, jolloin roottorisauvoihin indusoituvat jännite ja virta kasvavat, josta seuraa moottorin mekaanisen vääntömomentin sekä moottorin jättämän kasvu. Kuormitusta pienennettäessä tapahtumat ovat päinvastaiset. Sähkömoottori tulisikin sähkökäytön mitoituksen yhteydessä nähdä momenttilähteenä. (Ahoranta 2020, 254–258.)

5.2 Suora käynnistys (DOL) ja suunnanvaihto

Kontaktorista ja lämpö- tai ylikuormitusreleestä koostuva suora käynnistys (kuvio 9) on kompakti ja yksinkertainen tapa käynnistää moottori, jonka takia se onkin yleisin käytetty moottorin käynnistystapa. Suoran käynnistystyksen haittana on suuri käynnistysvirta, joka on noin 6–8 kertaa moottorin nimellisvirran suuruinen (kuvio 10). Suoran käynnistystyksen sopivuutta suunniteltuun sovellukseen on pohdittava tarkkaan, sillä suoran käynnistystyksen aiheuttama suuri käynnistysmomentti saattaa aiheuttaa tarpeetonta rasitusta laitteistolle ja sen osille. Käyttökohteiksi sopivat sovellukset, joissa suuri käynnistysmomentti on tarpeellinen kuten kuljettimet ja syöttölaitteet. (Pehmökäynnistinopas 2011, 13.) Suunnanvaihtokytkennässä suora käynnistinkytkennässä olevan kontaktorin rinnalle kytketään toinen kontaktori, jolla vaihdetaan minkä tahansa kahden vaiheen paikkaa keskenään, jonka seurauksena moottorin pyörimissuunta vaihtuu. Kontaktorien yhtäaikaista vetämistä tulee estää oikosululta välttymiseksi, sekä huolehtia moottorin akselin pysähtyneenä oleminen laiterikkojen ehkäisemiseksi.



Kuvio 9. Suora käynnistys. (Pehmokäynnistinopas 2011, 13)

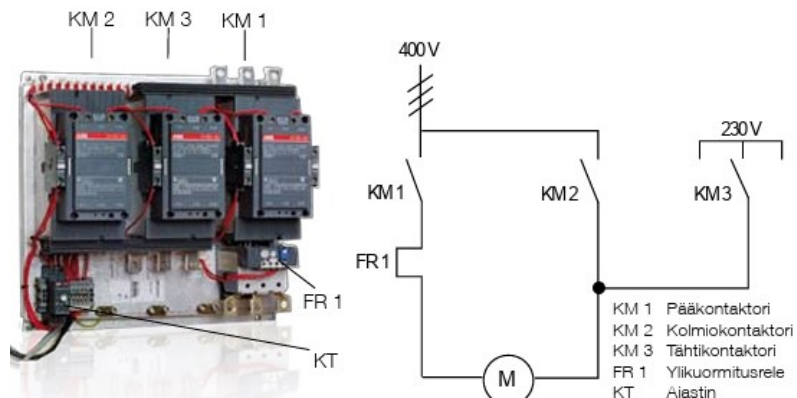


Kuvio 10. Suoran käynnistyksen momentti- ja virtakäyrä. (Pehmokäynnistinopas 2011, 13)

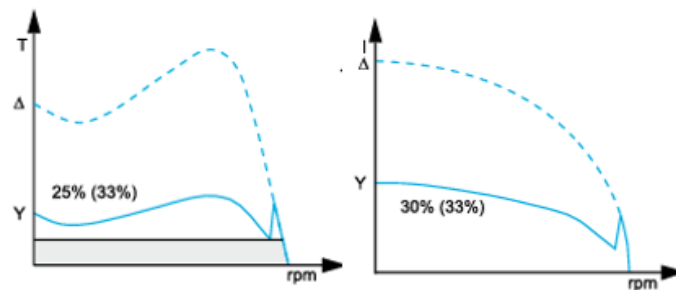
5.3 Tähtikolmiokäynnistys

Tähtikolmiokäynnistin (Y/D) (kuvio 11) koostuu yleensä pääkontaktorista, tähtikontaktorista, kolmiokontaktorista, sekä ajastimesta ja ylikuormitusreleestä. Tähtikolmiokäynnistimen periaatteena on vaihtaa moottorin käämityksien kytkentää moottorin kiihdytyksen aikana tähtikytkennän ja kolmiokytkennän välillä, jolloin vältetään moottorin käynnistyshetken suuri virtapiikki. Moottoria pidetään tähtikytkennässä, kunnes se on saavuttanut riittävän pyörimisnopeuden, jonka jälkeen kytkentä vaihdetaan kolmioon. Tämä siirto aiheuttaa voimansiirto- ja virtahuippuja (kuvio 13). Kytkennän vaihtamisen ajankohta käynnistyshetkestä määritetään ajastimella. Tähtikytkennässä moottorin virta on 1/3 kolmiokytkennän virrasta, mutta tulee huomioida myös momentin olevan tällöin noin 1/3 kolmiokytkennän momentista (kuvio 12). Tähtikolmiokäynnistin ei siis sovellu käynnistystavaksi sovelluksiin, joissa käynnistysmomentti on suuri, sillä moottorin momentti on liian pieni käyntinopeuden saavuttamiseksi ennen kolmiokytkentään vaihtamista. Käyttökohteita

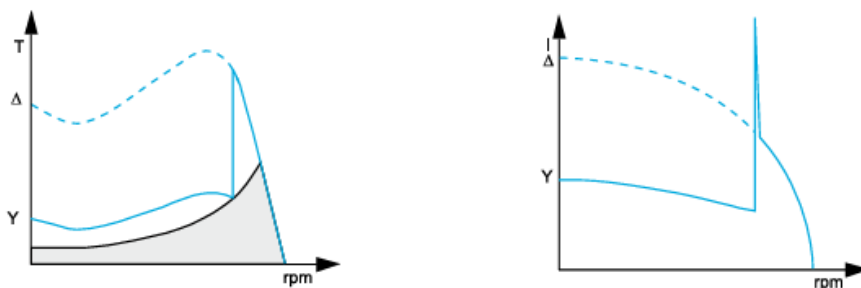
voivat olla esimerkiksi puhaltimet, joissa kuormitusmomentti kasvaa neliöllisesti suhteessa pyörimisnopeuteen. (Pehmokäynnistinopas 2011, 14–15.)



Kuvio 11. Tähtikolmiokäynnistin. (Pehmokäynnistinopas 2011, 14)



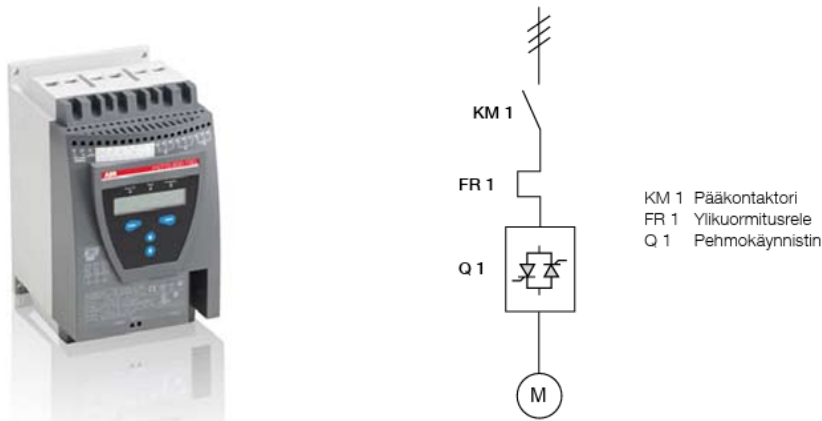
Kuvio 12. Tähtikolmiokäynnistimen momentti- ja virtakäyrät. (Pehmokäynnistinopas 2011, 15)



Kuvio 13. Tähtikolmiokäynnistimen voimansiirto- ja virtahiput pumppukäytössä.
(Pehmokäynnistinopas 2011, 15)

5.4 Pehmökäynnistin

Pehmökäynnistin (kuvio 14) on laite, jolla moottorin käynnistysvirtaa rajoitetaan tähtikolmiokäynnistintä edistyksemmemmin puolijohdetekniikkaa hyödyntäen, jolloin sähköverkon jännitteenalennemilta vältytään. Pehmökäynnistin eroaa toimintaperiaatteeltaan taajuusmuuttajasta, joka muuttaa taajuutta tai nopeutta, rampittamalla eli nostamalla jännitettä vähitellen lähtöjännitteestä täyteen jännitteeseen. Tällöin kiihdytys on hallittu ja momenttia voidaan säätää moottorin kuormasta riippumatta. Pehmökäynnistimen käytöllä voidaan pidentää laitteiden elinkaarta, kun välteään tarpeetonta mekaanista rasitusta. Pehmökäynnistimen pehmopysäytystoiminto on erityisen hyödyllinen paineiskujen välttämiseen putkistoissa, sekä hihnakäyttöjen hallittuun pysäyttämiseen. (Pehmökäynnistinopas 2011, 18.)



Kuvio 14. Pehmökäynnistin. (Pehmökäynnistinopas 2011, 18)

5.5 ABB:n älykkäät moottorinohjaimet

ABB:n älykkäät moottorinohjaimet (kuvio 15) mahdollistavat yksi- ja kolmivaiheisten moottoreiden ohjauksen joustavasti käyttötarpeen mukaan. Moottorinohjain voidaan konfiguroida toimimaan usealla eri tavalla, kuten esimerkiksi suorakäynnistimenä, tähtikolmiokäynnistimenä tai ylikuormitusreleenä. Sitä voidaan ohjata digitaalituloilla, sen omalla käyttöliittymäpaneelilla tai kenttäväylällä. Moottorin suojaukselle on suuri määrä parametrejä, joiden avulla suojaus voidaan räätälöidä laajuudeltaan sovelluksen tarpeen mukaiseksi. Moottorin tilasta saadaan kattavasti tietoa, jota voidaan sovelluskohtaisesti valikoida valvottavaksi halutulla laajuudella, esimerkiksi kenttäväylän kautta. Valvonnalle on varattu neljä sanaa digitaaliselle diagnoosidatalle, sekä viisi sanaa

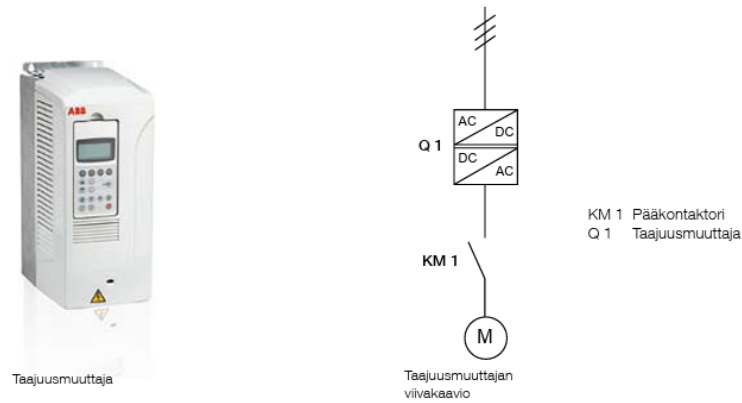
joihin käyttäjä voi parametroida haluamansa mittausdatan laajasta prosessiarvostasta. (Universal Motor Controller UMC100.3 2024, 70–94, 122, 128–130 & 132.)



Kuvio 15. UMC100.3 älykäs moottorinohjain. (Universal Motor Controller 100.3 n.d.)

5.6 Taajuusmuuttaja

Taajuusmuuttaja (kuvio 16) on tehoelektronikkalaite, joka muuttaa sähkömoottorin jännitteen taajuutta ja sitä kautta moottorin pyörimisnopeutta. Taajuusmuuttajan kaksi pääosaa ovat sähköverkosta tulevan vaihtovirran tasavirraksi muuttava tasasuuntaaja, sekä tasavirran vaihtovirraksi, jota voidaan säätää halutusti, muuttava vaihtosuuntaaja. Taajuusmuuttajaa käyttämällä saadaan moottorin nimellismomentti käytettäväksi heti käynnistyksestä lähtien ja myös virta on suunnitteen nimellisvirran suuruinen. Taajuusmuuttaja on energiatehokas ratkaisu, sillä prosessia voidaan ajaa aina juuri tarpeellisella nopeudella, jolloin energiaa ei mene hukkaan. Esimerkiksi virtausta ei tarvitse kuristaa venttiilillä, vaan moottorin pyörimisnopeutta säädetään tarpeen mukaan sopivaksi. Taajuusmuuttaja on erittäin hyvä ratkaisu sovelluksiin, joissa nopeutta on säädeltävä jatkuvasti. Taajuusmuuttajan suurimpana haittana ilmenee sen taajuudenvaihtamisesta ja siniaallon luomisesta sähköverkkoon aiheutuvat harmoniset yliaallot. (Pehmökäynnistinopas 2011, 16–17.)



Kuvio 16. Taajuusmuuttaja. (Pehmökäynnistinopas 2011, 17)

6 Kuormitustyypit

6.1 Vakiomomentti

Vakiomomentti-kuormitustyypissä (kuvio 17) kuormamomentin määrä pysyy vakiona, mutta teho kasvaa pyörimisnopeuden kasvaessa suoraan verrannollisesti pyörimisnopeuteen nähden kaavan 1 mukaan. Toisin sanoen kuormitusmomentti ei muutu, mutta kuorman nopeuden kiihdyttäminen kasvattaa vaadittua sähkötehoa. Koska vakiomomentti-kuormitustyypissä käynnistyshetken momentti on nimellismomentin suuruinen, ei esimerkiksi tähtikolmiokäynnistin sovellu tällaisen kuormitustyypin käynnistystavaksi. Muun muassa ruuvikompressorit ja kuljettimet ovat esimerkkejä tyypillisistä vakiomomenttisolvelluksista. (Tekninen opas Nro7 2001, 20.)

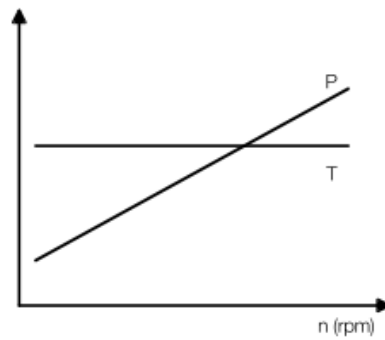
$$T = Tr \left(\frac{n}{nr} \right)^0 \quad (1)$$

missä T = momentti

Tr = nimellismomentti

n = hetkellinen nopeus

nr = nimellisnopeus



Kuvio 17. Vakiomomenttisovelluksen tyypilliset teho- ja momenttikäyrät. (Tekninen opas Nro7 2001, 20)

6.2 Neliöllinen momentti

Yleisimmin esiintyvä kuormitustyyppi, jonka tyypillisiä sovelluksia ovat pumput ja puhaltimet. Neliöllisen momentin kuormitustyyppissä (kuvio 18) momentti kasvaa neliöllisesti ja teho kuutiollisesti verrannollisesti nopeuteen kaavan 2 mukaan. Kuormitusmomentin ollessa käynnistyshetkellä pieni, voidaan neliöllisen momentin kuormitustyyppin sovelluksissa käyttää myös tähtikolmiokäynnistintä, vaikkei se voimansiirto- ja virtahuippujen vuoksi ole esimerkiksi pumppujen käynnistykseen sopivin ratkaisu. (Tekninen opas Nro7 2001, 20.)

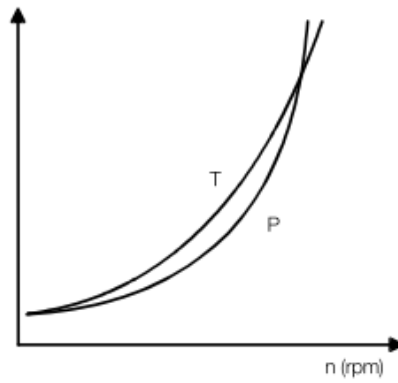
$$T = Tr \left(\frac{n}{nr} \right)^2 \quad (2)$$

missä T = momentti

Tr = nimellismomentti

n = hetkellinen nopeus

nr = nimellisnopeus



Kuvio 18. Neliöllisen momentin sovelluksen tyypilliset momentti- ja tehokäyrät. (Tekninen opas Nro7 2001, 20)

6.3 Vakioteho

Tavallisesti vakiotehokuormitustyyppi (kuvio 19) on kuormitustyyppinä, kun materiaalia rullataan ja sen läpimitta muuttuu rullauksen aikana, kuten rullaimissa. Tässä kuormitustyyppissä teho pysyy vakiona, mutta momentti on kääntäen verrannollinen pyörimisnopeuteen kaavan 3 mukaan. Kuormitustyyppinä vakioteho sovellus vaatii suuren käynnistysmomentin, jonka jälkeen momentti lähtee pienenemään nopeuden kasvaessa, jolloin tähtikolmiokäynnistin ei sovellu tällaisen sovelluksen käynnistämiseen. (Tekninen opas Nro7 2001, 21.)

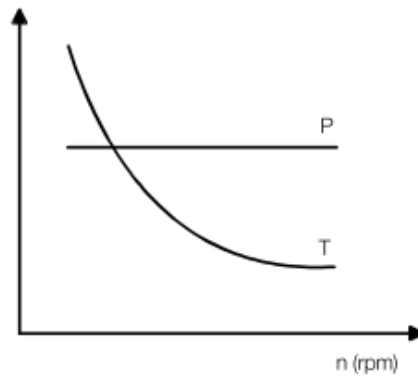
$$T = Tr \left(\frac{n}{nr} \right)^{-1} \quad (3)$$

missä T = momentti

Tr = nimellismomentti

n = hetkellinen nopeus

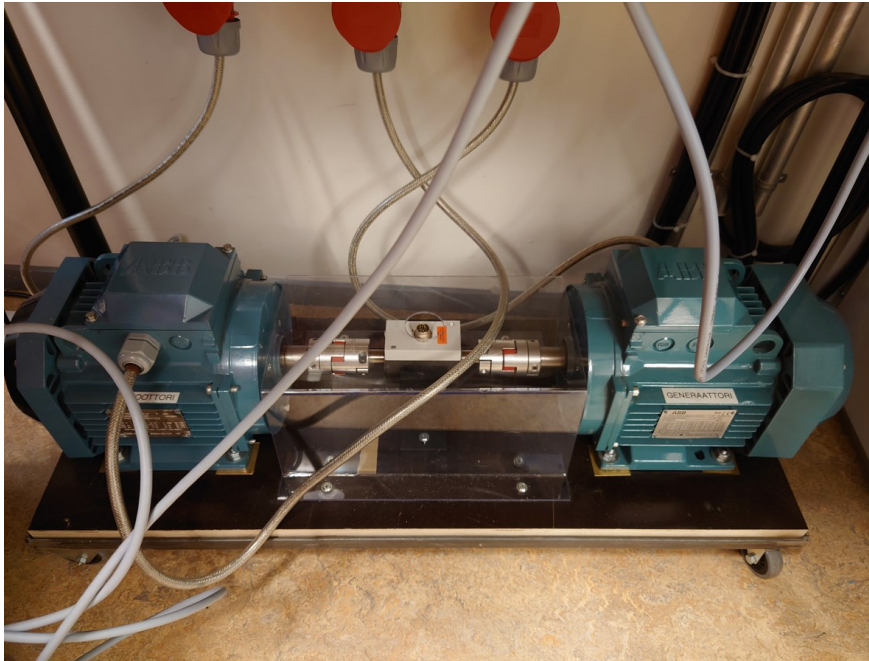
nr = nimellisnopeus



Kuvio 19. Vakiotehosovelluksen tyypilliset teho- ja momenttikäyrät. (Tekninen opas Nro7 2001, 21)

7 Teollisuuskeskus

Toimeksiantajan laboratoriotiloissa sijaitsevan, opetuskäytössä toimivan teollisuuden moottorikeskusta mallintavan sähkökeskuksen kehittäminen oli tullut ajankohtaiseksi. Keskuksessa on viisi erilaista lähtöä, joilla sähkömoottori voidaan käynnistää. Keskuksen lähdöillä ohjataan sähkömoottoria, joka on kytketty akselistaan kiinni generaattorina toimivaan kuormamoottoriin, jonka tarkoitus on jarruttaa moottorin pyörimistä käyttäjän asettamalla kuormitusmomentilla (kuvio 20). Käynnistinvaihtoehtoina ovat Tähti-kolmiökäynnistin, suunnanvaihtokäynnistin, pehmokäynnistin, älykäs moottorinohjain sekä taajuusmuuttaja (kuvio 21). Moottorilähtöjen lisäksi keskuksessa on taajuusmuuttaja kuormamoottorin ohjaamiseen (kuvio 22), joka mahdollistaa kuormitustyyppien ja kuormitusmomenttien simuloimisen, sekä kuormamoottorin tuottaman jarrutusenergian takaisin sähköverkkoon syöttämisen mahdollistava taajuusmuuttaja ja verkkosuodin (kuvio 23).



Kuvio 20. Moottori ja generaattori



Kuvio 21. Moottorilähdöt



Kuvio 22. Kuormamoottorin taajuusmuuttaja



Kuvio 23. Verkkoon jarruttava taajuusmuuttaja & verkkosuodin

Moottorilähtöjen ohjaukseen käytetään ohjelmoitavaa logiikkaa keskuksen ovelle olevan käyttöliittymäpaneelin kautta, josta käyttäjä voi valita moottorin käynnistystavan eli millä lähdöllä moottori käynnistetään. Älykäs moottorinohjain, taajuusmuuttajalähtö sekä kuormamoottorin taajuusmuuttaja ja verkkoon jarruttava taajuusmuuttaja on liitetty kenttäväylään, joka keskuksessa on Profibus DP. Väylän kautta on toteutettu moottorinohjaimen sekä taajuusmuuttajalähdön ohjaaminen, sekä moottorin virran seuranta. Moottorin syöttöjohtimet kulkevat älykkään moottorinohjaimen läpi, joka mittaa moottorin ottamaa virtaa ja virtatieto lähetetään kuormamoottoria ohjauvalle taajuusmuuttajalle. Kuormamoottorin ohjaukseen käytetty taajuusmuuttaja tukee mahdollisuutta tehdä sovellusohjelmointi sen omalla DriveSPC-ohjelmalla ja kuormamoottorin ohjaus olikin toteutettu siten, että taajuusmuuttajalle oli tehty kolme erilaista kuormitusohjelmaa, joilla kuormamoottoria voitiin ohjata.

Kuormitusohjelma vaihtoehtoina olivat vakiomomentti, neliöllinen momentti sekä vakioteho. Kuormitusohjelman vaihtaminen tapahtui siten, että taajuusmuuttajaan kytkettiin tietokone ja eri ohjelma ladattiin taajuusmuuttajaan. Moottorin käyttäytymisen seuranta toimi siten, että tietokone, jolla oli DriveStudio-ohjelma, kytkettiin kiinni kuormamoottoria ohjaavaan taajuusmuuttajaan. DriveStudion kautta käyttäjä pystyi tarkkailemaan kuormamoottoria ohjaavan taajuusmuuttajan parametrejä, joista tärkeimpinä momenttia, pyörimisnopeutta ja tehoa sekä kenttäväylästä saatavaa tietoa moottorin virrasta. DriveStudion kautta käyttäjällä oli myös mahdollisuus muuttaa kuormitusmomentin asetusarvoa.

8 Toteutus

8.1 Tutkimuksellinen kehittäminen

Tämän opinnäytetyön toteutustavaksi päädyttiin valitsemaan kehittämistutkimus, sillä se koettiin soveltuvimmaksi toimeksiannon luonteen vuoksi, sekä tavoitteiden saavuttamiseksi ja tutkimuskypsyksiin vastaamiseksi. Tutkimusmenetelmänä kehittämistutkimus yhdistää tutkimusta ja kehittämistä syklisessä prosessissa, johon sisältyy sekä teoreettisia että kokeellisia vaiheita. Kehittämistä toteutetaan olemassa olevassa ympäristössä olevaan ongelmaan, jolloin tutkimuksen tarve perustuu todelliseen tosielämän ongelmaan. Tutkimus alkaa ongelman analysoimisella, jolla pyri-

tään selkeyttämään kehittämistavoitteet. Kehittämistavoitteiden selkeydyttyä voidaan laatia tutkimusta kohti tavoitteiden saavuttamista ohjaava alustava kehittämissuunnitelma, jota päivitetään tutkimuksen edetessä. (Pernaa 2013.)

8.2 Tutkimuskysymykset

Tämän tutkimuksen tavoitteena on tutkimuksellista kehittämistyötä tutkimusmenetelmänä käyttäen kartoittaa toimeksiantajan laitteiston lähtötilanne ja kehittää sitä toimeksiantajaa palvelevaan suuntaan. Tutkimuksella pyritään löytämään vastaukset seuraaviin tutkimuskysymyksiin:

1. Mikä teollisuuskeskuksen nykytilanne on?
2. Kuinka teollisuuskeskuksen käyttöä voidaan sujuvoittaa?
3. Miten järjestelmän jatkokehitykselle luodaan vahva pohja?

Tutkimus pyrkii koostamaan tilannekuvan toimeksiantajan järjestelmästä, jotta sen kunnosta oltaisiin ajan tasalla, tulevien kunnossapito- ja kehitysprojektien suunnittelun helpottamiseksi. Tutkimuksen tuloksena syntyvien teknisten ratkaisujen tulisi palvella harjoitus- ja laboratoriotöiden joutavaa suorittamista. Tunnistetaan kehitystyön luonne jatkumona ja pyritään löytämään tätä tukevia toimintatapoja.

8.3 Eettiset periaatteet

Opinnäytetyössä noudatettiin Jyväskylän ammattikorkeakoulun eettisiä periaatteita (Hautamäki, Hyvätti, Kirjalainen, Kokko, Korhonen, Liimatainen, Laitinen-Väänänen, Norvapalo & Törn-Laapio 2024), sekä tutkimuseettisen neuvottelukunnan ohjeistusta, josta erityisesti pyrittiin huolelliseen työn tekemiseen, antamaan muiden työlle siitä kuuluva arvo, sekä rehelliseen toimintaan (Hyvä tieteellinen käytäntö ja sen loukkausepäilyjen käsittely Suomessa 2023, 12–14). Lähteiden monipuolisuuteen on pyritty, siinä määrin kuin se opinnäytetyön realiteettien asettamissa rajoissa on ollut mahdollista. Kestävän kehityksen edistämiseksi pyrittiin hyödyntämään mahdollisimman paljon vanhoja ja käytettyjä laitteita ja materiaaleja.

8.4 Projektin alkukartoitus

Työ alkoi keskuksen tutustumisella toimeksiantajan kanssa, sekä keskustelulla mikä nykytilanne on ja mitä kehitystarpeita ja toiveita toimeksiantajalla on. Ensitetona oli, että toimeksiantajalla oli tarve keskuksen ohjelmoitavan logiikan ja sen käyttöliittymäpaneelin vaihtamiselle. Keskustelussa selvisi keskuksen kenttäväylän ja taajuusmuuttajien kanssa olleen ongelmia ajan saatossa, sekä kuormitusohjelman vaihtamisen olevan hieman epäkäytännöllistä ja sisältävän mahdollisuuden käyttäjävirheiden tapahtumiseen, jotka voivat aiheuttaa haasteita laboratoriotöiden sujuvalle suorittamiselle. Toimeksiantaja määritteli suuntaa antavasti vaateet käyttöliittymäpaneelin toiminnallisuuksille, joiden perusteella käyttöliittymäsuunnittelulle saatiin pohja. Alustavien keskusteluiden jälkeen tutustuttiin toimeksiantajalta saatuihin keskuksen liittyviin dokumentteihin ja ohjelmiin ja pyrittiin muodostamaan ymmärrys järjestelmän toiminnasta sekä tulevan työn laajuudesta.

8.5 Ohjelmoitava logiikka ja käyttöliittymäpaneeli

Koska alustavana opinnäytetyön aiheena oli teollisuuskeskuksen logiikan vaihto, aloitettiin varsinainen työ tutkimalla keskuksen silloista logiikkaa ja käyttöliittymäpaneelia. Ensimmäisenä suunnitelmana oli kartoittaa mahdollisuus hyödyntää olemassa olevia laitteita ja tehdä vain muutoksia sovellusohjelmaan ja käyttöliittymään. Teollisuuskeskuksen ohjelmoitavana logiikkana toimi ABB:n 1. sukupolven AC500 CPU mallia PM571-ETH ja käyttöliittymäpaneelina CP430 BP-ETH.

Tarkoituksena oli tutustua olemassa olevaan sovellusohjelmaan ja käyttää sitä soveltuvilta osin mallina uudelle ohjelmalle. Koska toimeksiantajalla ei ollut enää tallessa minkäänlaisia tiedostoja logiikassa olevasta sovellusohjelmasta eikä logiikassa ollut muistikorttia asennettuna, yritettiin logiikkaan muodostaa yhteys, jotta ohjelma voitaisiin ladata tietokoneelle sen tutkimista varten. Kun logiikkaan ei yrityksistä huolimatta onnistuttu saamaan yhteyttä Ethernetillä eikä sarjaväylällä, käännyttiin ABB:n teknisen tuen puoleen, josta selvisi, että kyseisen 1. sukupolven CPU ei tue lähdekoodin lataamista sen sisäiseen muistiin lainkaan ja sekä logiikan, että käyttöliittymäpaneelin tuki ohjelmiston ja laitteiston osalta on päätynyt, joten logiikasta ei ollut mahdollista ladata ohjelmaa ulos. Olemassa olevaa logiikkaa tai käyttöliittymäpaneelia ei siis voitu enää hyödyntää, vaan ne oli vaihdettava uudempiin laitteisiin ja sovellussuunnittelu oli toteutettava kokonaisuudessaan uusiksi.

Seuraavaksi siirryttiin valitsemaan uutta ohjelmoitavaa logiikkaa sekä käyttöliittymäpaneelia. Toimeksiantajan kanssa keskusteltiin uuden logiikan ja käyttöliittymän valmistajan merkityksellisyydestä ja päädyttiin vaihtamaan laitteet Siemensin valmistamaan S7-1200 CPU:hun (kuvio 23) sekä KTP1000 Basic PN käyttöliittymäpaneeliin (kuvio 24), sillä nämä olivat heti saatavilla toimeksiantajan varastossa. Vain väylämoduuli CM1243-5 jouduttiin hankkimaan. Valintaan vaikutti keskeisesti myös toimeksiantajan organisaatiosta löytyvä Siemensin TIA Portal-ohjelmointiympäristöön liittyvä osaaminen sekä olemassa olevat lisenssit kyseiseen ohjelmistoon.

Uusien laitteiden valitsemisen jälkeen purettiin vanha logiikka sekä käyttöliittymäpaneeli pois keskuksesta ja asennettiin uusi logiikka keskuksen. Logiikan vaihdon yhteydessä uusittiin logiikan johdotus, sillä ne eivät ylettäneet uuden logiikan kytkentäliittimille. Käyttöliittymäpaneelin vaihtamiseksi jouduttiin keskuksen ovelle olevaa vanhan käyttöliittymäpaneelin reikää suurentamaan, sillä uusi näyttö oli fyysisesti huomattavasti vanhaa suurempi. Asennusten jälkeen piirikaaviot päivitettiin ajan tasalle.

8.6 Sovellusohjelmointi

Opinnäytetyön toimeksiantajan kanssa käydyn aloituspalaverin perusteella sovellus- ja käyttöliittymäsuunnittelua varten oli saatu pohjatietoja, joiden perusteella suunnittelu voitiin aloittaa. Keskustelussa käytiin läpi mitä käyttöliittymäpaneelilta pitäisi pystyä ohjaamaan ja seuraamaan. Koska kuormitusohjelmat oli toteutettu DriveSPC-ohjelmalla tekemällä kolme erilaista kuormitussovellusta, jotka ladattiin suoraan taajuusmuuttajaan, sen mukaan millaisen kuormitussovelluksen mukaan taajuusmuuttajan halutaan kuormamootoria ohjaavan, oli kuormitussovelluksen vaihtaminen epäkäytännöllistä. Lisäksi tietokone, jolla kuormitussovelluksien tekoon käytetty DriveSPC ja Kuormamootorin käyttäytymisen seurantaan käytetty DriveStudio oli asennettuna, toimi Windows XP-käyttöjärjestelmällä, oli itsessään jonkinasteinen tietoturvariski. Päätettiin siis siirtää myös kuormitusohjelmat toteutettavaksi logiikan kautta käyttäjäkokemuksen ja tietoturvan parantamiseksi.

Sovellusohjelmoinnin osuus opinnäytetyöstä oli ennakoarvioiden mukaan työn työläin ja haastavin osuus, jonka haastavuutta lisäsi, ettei vanhoja ohjelmia voitu hyödyntää uuden suunnittelussa. Ohjelmointi aloitettiin luomalla TIA Portaliin projekti, johon lisättiin työssä käytettävä logiikka, käyttöliittymäpaneeli, väylämoduuli sekä digitaalisten tulojen ja lähtöjen lisämoduuli. Seuraavaksi

ladattiin ABB:n sivuilta kenttäväylässä oleville laitteille General Description File-tiedostot eli laitekohtaiset kuvaustiedostot. Tiedostojen latausten jälkeen suoritettiin Profibus DP-kenttäväylän konfigurointi, jossa väylälle määritettiin tiedonsiirtonopeus, väylälaitteille väyläosoitteet sekä saatiin jokaiselle laitteelle oma muistialue logiikkaan.

Koska ohjelman tekeminen jouduttiin aloittamaan täysin tyhjästä, päädyttiin suunnittelemaan ensin moottorilähtöjen ohjaukset ja lukitukset, sillä niiden suunnittelusta oli suoraviivaisinta aloittaa. Ohjelmalohkoja testattiin TIA Portalin simulointityökalulla suunnittelun lomassa. Kun ohjelmalohkojen toiminta vastasi simuloinnin perusteella tahdottua, siirryttiin suunnittelemaan seuraavaa ohjelmalohkoa. Tähtikolmio, suunnanvaihtolähtö, pehmokäynnistin sekä taajuusmuuttaja olivat yksinkertaisia toteutettavia ja niiden suunnittelu oli helppoa. Älykäs moottorinohjain vaati lähdeistä eniten selvitystyötä, sillä se oli laitteena vieras. Laitteen ohjeopasta tutkimalla selvisi moottorinohjaimen tukevan neljää valvontatavua ja kahta käskytavua. Oppaassa oli myös kuvattu tavujen sisältö yksittäisten bittien osalta. Tavut 0 ja 1 sisältävät laitteen tilojen valvonnan, tavuja 2 ja 3 käytetään moottorivirran lukemiseen. Käskytavuilla 0 ja 1 voidaan ohjata moottorinohjaimen digitaalisia lähtöjä. Kun ohjaimen tavujen sisältö oli selvitetty, täytyi seuraavaksi selvittää, kuinka tavut voidaan purkaa ohjelmassa yksittäisiksi biteiksi niiden tilojen valvontaa ja muuttamista varten, sekä kuinka yksittäiset bitit saadaan kerättyä takaisin tavuiksi, sillä moottorinohjaimelle tuli lähettää käskyt tavuina eikä yksittäisinä bitteinä. TIA Portalin lohkokirjastosta löytyi juuri tähän tarkoitukseen tehdyt lohkot, joilla myös moottorinohjaimen ohjaukset ja valvonnat saatiin tehtyä.

Kun moottorilähdöt oli saatu rakennettua ohjelmaan ja testattua simulointityökalulla, sillä laajuudella kuin se oli mahdollista, siirryttiin suunnittelemaan kuormitusohjelmia. Kuormitusohjelmien suunnittelu aloitettiin selvittämällä maksimivääntömomentti, jolla moottoria voidaan jarruttaa. Momentin selvittäminen oli keskeistä, sillä generaattorina toimiva kuormamoottori on teholtaan moottoria suurempi. Tämä momentti saatiin laskemalla moottorin nimellismomentti moottorin kilpiarvojen perusteella kaavan 4 mukaan. Kuormitusohjelmat toteutettiin siten, että vakiomomentille, neliölliselle momentille ja vakioteholle on oma ohjelmalohko, joka suorittaa toteutuvan ohjausviestin suuruuden laskentaa, oman kaavansa mukaan. Vakiomomentti noudattaa kaavaa 1, neliöllinen momentti kaavaa 2 ja vakioteho kaavaa 3. Käyttäjä voi käyttöliittymäpaneelin kautta

asettaa kuormitusmomentin väliltä 0–100 %, jolla moottoria jarrutetaan sekä valita mitä kuormitusohjelmaa käytetään, jolloin valitun kuormitusohjelman ohjelmalohkon tuottama ohjausviesti välitetään kuormamoottoria ohjaavalle taajuusmuuttajalle.

$$Tn = \frac{9550 * P}{nr} \quad (4)$$

missä Tn = nimellismomentti

P = moottorin nimellisteho

nr = nimellisnopeus

8.7 Käyttöliittymäsuunnittelu

Kun sovellussuunnittelu oli edennyt niin, että moottorilähtöjen ja kuormitusohjelmien toteutus oli hahmoteltu, alkoi käyttöliittymän hahmottelu. Käyttöliittymää varten oli olemassa toimeksiantajan alustavia vaatimusmäärittelyitä, mutta koska kehitysideoita ja tarkennuksia tiedettiin tulevan työn edetessä, oli sovittu, että käyttöliittymän osalta opinnäytetyöntekijä lähettää toimeksiantajalle kuvia käyttöliittymäsuunnittelun edetessä ja toimeksiantaja antaa palautetta siitä. Työssä käytetyn käyttöliittymäpaneelin ollessa Siemensin 1. sukupolven paneeli, siihen ei ollut mahdollista tehdä ponnahdusikkuna mallisia toimintoja, joten päädyttiin suunnittelussa alustavasti sijoittamaan moottorilähdöt, kuormitusohjelmien valinta sekä moottoreiden arvoja näyttävä graafinen kuvaaja kaikki käyttöliittymän eri sivuille.

9 Käyttöönotto

Sovellus- ja käyttöliittymäsuunnittelun edistyttyä siihen valmiusasteeseen, että ne olivat valmiita testattavaksi oikeilla laitteilla, siirryttiin laboratorioon teollisuuskeskukselle valmistelemaan keskuksen uuden sovellusohjelman käyttöönottoon.

Ensimmäiseksi oli asennettava uusi käyttöliittymäpaneeli ja ohjelmoitava logiikka ja tehtävä tarvittavat kytkentämuutokset. Uusien laitteiden asentamisen ja kytkemisen jälkeen voitiin aloittaa niiden käyttöönotto. Kun ohjelmoitava logiikka ja käyttöliittymäpaneeli oli saatu käyttöönotettua, aloitettiin uuden sovellusohjelman testaus digitaalisten tulojen ja lähtöjen osalta. Tulojen ja lähtöjen testauksen jälkeen mahdolliset kytkentävirheet oli saatu karsittua pois ja varmistuttua siitä, että sovellusohjelma toimii niiltä osin halutunlaisesti.

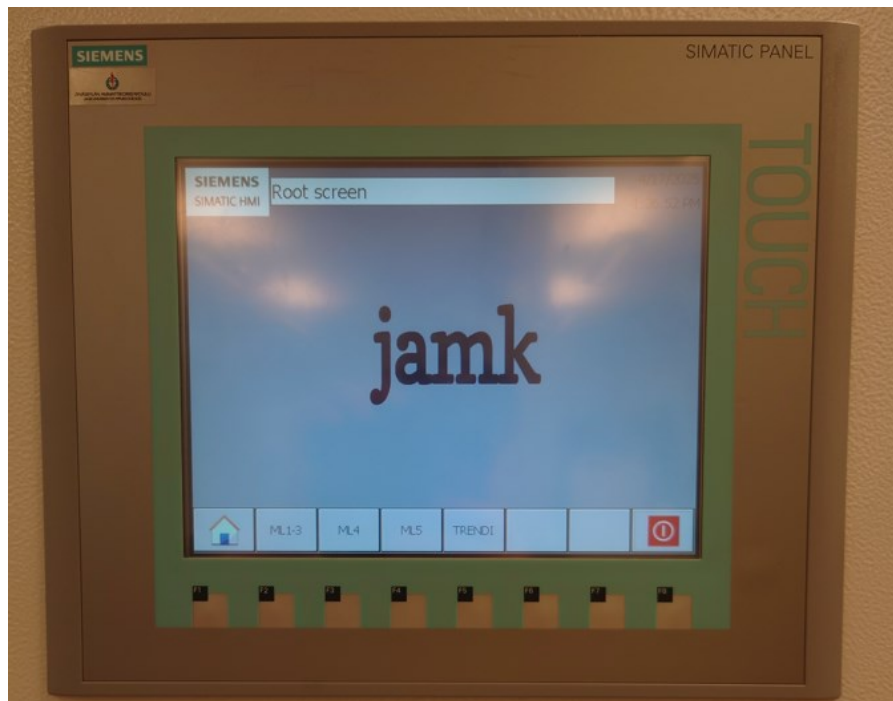
Seuraavaksi siirryttiin väylälaitteiden testaukseen. Tiedettiin alkutietoina, että väylässä oli jotain vikaa ja sitä oli ennen opinnäytetyötä tutkittu, mutta ei saatu selvitettyä syytä. Väylän vianetsintään kulutettiin myös opinnäytetyössä paljon aikaa ja käyttöönotto väylälaitteiden osalta jouduttiin tekemään osissa, ottaen käyttöön vain sellaiset laitteet, joiden ollessa väylässä pysyy väylä pysyvässä. Sitä mukaa kun vianetsintä eteni ja vika saatiin paikannettua ja korjattua voitiin väylään lisätä uusi laite. Kun väylä saatiin toimimaan, voitiin aloittaa myös kuormitusohjelmien testaus. Tässä käyttöönoton vaiheessa sovellusohjelmaan tehtiin paljon muutoksia ja törmättiin ongelmiin kuormamootoria ohjaavan taajuusmuuttajan kanssa. Seuraavissa kappaleissa käyttöönoton vaiheita käsitellään tarkemmin.

9.1 Laitteiden asennus ja kytkentämuutokset

Uusien laitteiden valitsemisen jälkeen purettiin vanha logiikka sekä käyttöliittymäpaneeli pois keskuksesta ja asennettiin uusi logiikka keskukseen. Logiikan vaihdon yhteydessä uusittiin logiikan johdotus, sillä ne eivät ylettäneet uuden logiikan kytkentäliittimille (kuvio 23). Käyttöliittymäpaneelin vaihtamiseksi jouduttiin keskuksen ovesa olevaa vanhan käyttöliittymäpaneelin reikää suurentamaan, sillä uusi näyttö oli fyysisesti huomattavasti vanhaa suurempi. Aukon suurentamisen jälkeen näyttö asennettiin oveen ja johdotettiin (kuviot 24 & 25).



Kuvio 23. Uusi ohjelmoitava logiikka



Kuvio 24. Uusi käyttöliittymäpaneeli edestä



Kuvio 25. Uusi käyttöliittymäpaneeli takaa

Ohjelmoitavan logiikan osalta ei tarvinnut tehdä muuta kuin ladata ohjelma siihen. Käyttöliittymäpaneelin kanssa ilmeni ongelma sitä käynnistettäessä, sillä paneeli oli jumissa käynnistysnäyttötilassa. Paneelin eheyttä ei ollut osattu epäillä ennen ensimmäistä käynnistystä sillä se oli ollut toisessa käytössä, josta se oli toimivana purettu ja siirretty varastoon säilytykseen. Paneeliin yritettiin saada yhteys tietokoneella siinä onnistumatta. Asiaa selviteltiin ja lopulta Siemensin teknisen tuen neuvoilla päädyttiin kokeilemaan paneelin palauttamista tehdasasetuksille ja uudelleen alustamista. Paneeli saatiin toimimaan ja sille pystyttiin antamaan IP-osoite sekä lataamaan ohjelma.

9.2 Suorien moottorilähtöjen testaus

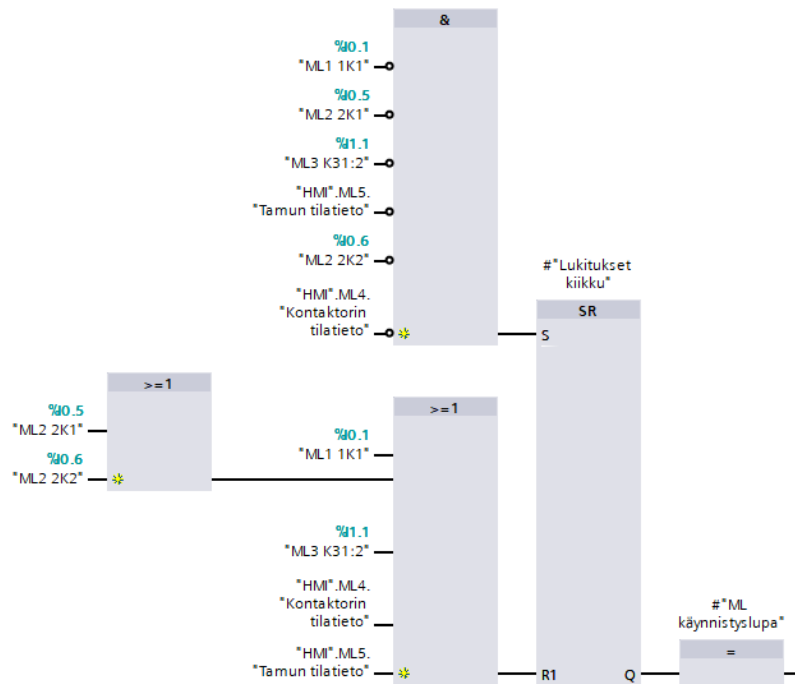
Kun vaihdetut logiikka ja käyttöliittymäpaneeli oli saatu otettua käyttöön, aloitettiin sovellusohjelman testaaminen logiikan digitaalisten tulojen ja lähtöjen osalta. Testaamisen ensimmäinen vaihe toteutettiin pelkällä 24V DC-ohjausjännitteiden kanssa ilman pääjännitteitä, jotta mahdolliset virheet sovellussuunnittelussa eivät aiheuta tuhoa laitteille. Koska vanhojen piirikaavioiden ajantasaisuudesta ei ollut varmuutta, mutta niitä oli käytetty kytkentämuutoksissa pohjana, olisi näin löydetty myös siitä johtuvat mahdolliset kytkentävirheet turvallisesti ja nopeasti.

Tulojen testaus alkoi niitä yksittäin kentän päästä päälle pakottamalla. Kytkinvarokkeita käännettiin kiinni ja kontaktoreja painettiin kiinni ruuvimeisselillä, jonka jälkeen tarkistettiin, että logiikassa syttyy oikeaa tuloa vastaava merkkivalo. Samalla tuloja valvottiin myös TIA Portalissa, jotta nähtiin, ettei sovellussuunnittelussa ole nimetty tulobittejä väärin tai käytetty niitä vahingossa väärässä paikassa. Kun tulot oli testattu, siirryttiin lähtöihin. Lähtöjä pakotettiin päälle TIA Portalilla ja tarkistettiin, että oikea rele tai kontaktori vetää.

Kun oltiin testausten perusteella varmistuttu kytkentöjen olevan kunnossa, siirryttiin testaamaan sovellusohjelman moottorinohjauksen ehtojen ja lukitusten toimintaa (kuvio 26). Koska teollisuuskeskuksessa samaa moottoria ohjataan usealla eri käynnistimellä, oli sovellussuunnittelussa kiinnitettävä erityistä huomiota siihen, ettei moottoria pysty ohjaamaan muilla lähdöillä silloin, kun jokin muu lähtö ohjaa moottoria. Lisäksi tuli huolehtia, ettei pyörimissuuntaa pääse vaihtamaan silloin kun moottori pyörii.

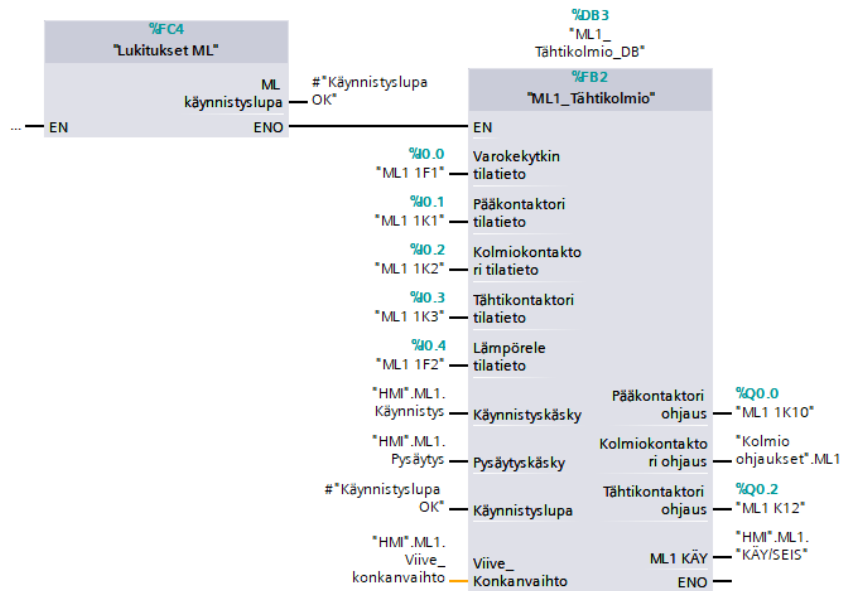
Kun oli testattu sovellusohjelman toiminta moottorilähtöjen ohjauksien ja lukitusten osalta voitiin keskukseseen kytkeä pääjännitteet päälle ja testata sovellusohjelmaa käynnistämällä moottoria moottorilähdöillä, jotka eivät ole väylässä (kuviot 27, 28 & 29). Sovellusohjelman moottorinohjauslohkoissa olevia ajastimia säädettiin vastaamaan paremmin sekä moottorin pysähtymisaikaa, että kiihdytysaikaa nimelliskohteeseen.

Network 1: Ehdot lähdön vaihdolle / päälleohjaukselle
 Comment



Kuvio 26. Moottorilähtöjen lukitukset

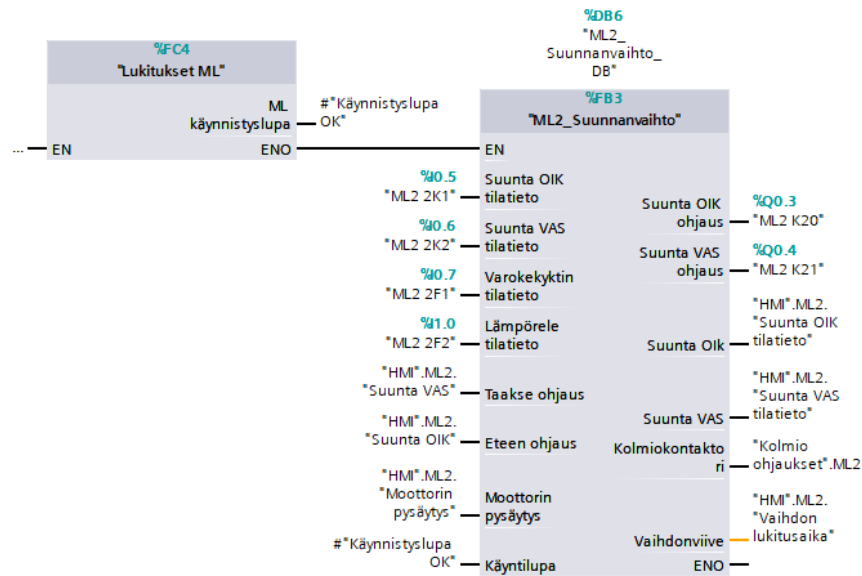
Network 1: ML1 Tähtikolmio lähtö
 Comment



Kuvio 27. Tähtikolmiolähtö ohjelmalohko

Network 2: ML2 Suunnanvaihto lähtö

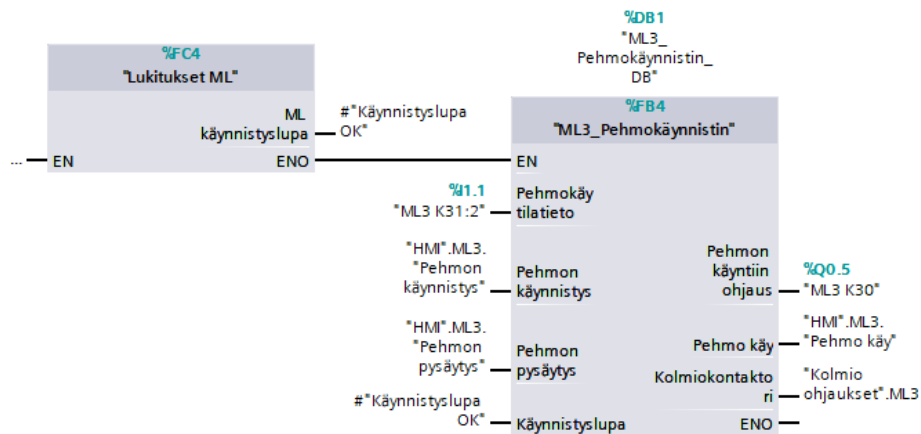
Comment



Kuvio 28. Suunnanvaihtolähtö ohjelmalohko

Network 3: ML3 Pehmökäynnistin lähtö

Comment



Kuvio 29. Pehmökäynnistinlähtö ohjelmalohko

9.3 Väylän pystytys ja vianetsintä

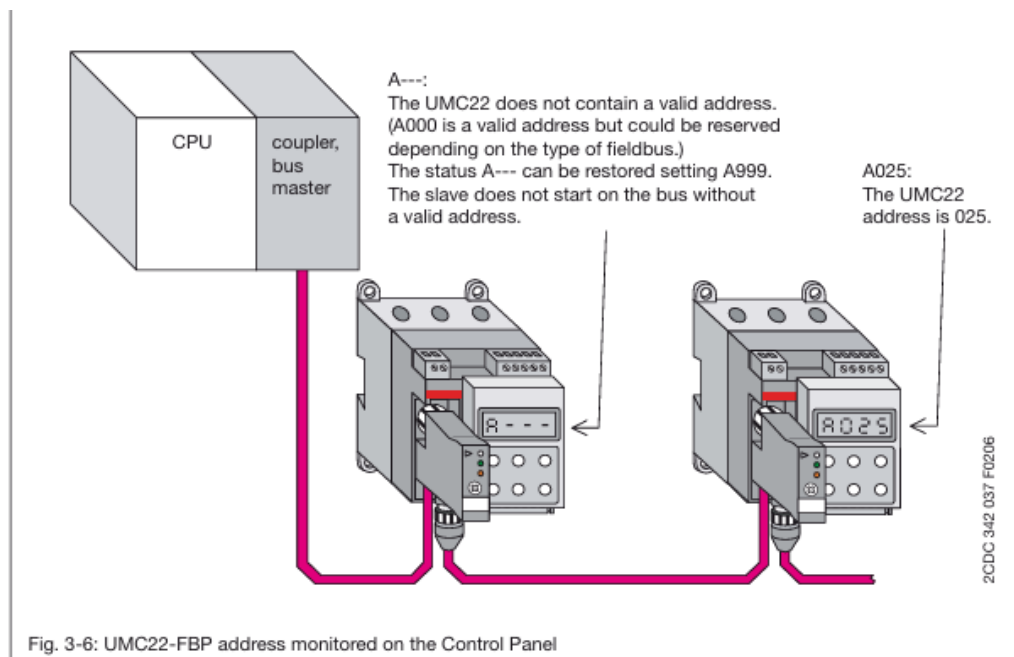
Suorien moottorilähtöjen testauksen jälkeen aloitettiin väylän pystyttäminen. Väylässä osattiin odottaa olevan jotain vikaa, muttei sen tarkemmin tiedetty millainen vika. Väylään kuuluviin laitteisiin kytkettiin jännitteet ja siirryttiin parametroimaan niitä väylään.

Ensimmäisenä parametroidiin älykäs moottorinohjain UMC22-FBP ja parametroidin jälkeen korjattiin sen tila- ja käskysanojen tavut oikeaan järjestykseen sovellusohjelmassa. UMC22-FBP:n relelähtöjä ohjattiin päälle sovellusohjelman puolelta ja todettiin tilasanan bittien muuttuvan ja tiedonsiirron toimivan, joten siirryttiin seuraavaan laitteeseen. Seuraavaksi liitettiin kuormamoottorin taajuusmuuttaja ja verkkoon jarruttava taajuusmuuttaja väylään. Parametrejä asetettaessa selvisi, ettei kuormamoottoria ohjaavan taajuusmuuttajan eikä verkkoon jarruttavan taajuusmuuttajan kenttäväyläsovittimelle ollut asetettu mitään väyläparametrejä eli ne eivät olleet osana kenttäväylää lainkaan. Sovittimille asetettiin oikeat tiedonsiirtoparametrit ja molemmat taajuusmuuttajat alkoivat TIA Portalilla näkyä väylässä ja tilasanojen perusteella todettiin tiedonsiirron toimivan.

Moottorilähtönä toimivan ACS355 taajuusmuuttajan kenttäväyläsovittimen parametrit olivat valmiiksi asetettuina, joten sen osalta tarkistettiin vain oikea väyläosoite. Väylälaitteiden kenttäväylään parametroidin jälkeen havaittiin väylän toiminnassa olevan jotain kummallista. TIA Portalista voitiin seurata, kuinka väylälaitteet häviävät väylältä erilaisin kombinaatioin. Välillä väylässä oli vain älykäs moottorinohjain, välillä vain taajuusmuuttajat, välillä kaksi taajuusmuuttajaa ja välillä älykäs moottorinohjain ja jokin kolmesta taajuusmuuttajasta.

Vian etsintä aloitettiin tarkistamalla jokaisen profibus -liittimen oikea kytkentä, yleinen kunto sekä alavetovastusten ja kytkimien eheys. Päätteet oli tehty moitteettomasti ja liittimet olivat täysin ehjiä. Taajuusmuuttajien kenttäväyläsovittimet olivat ehjiä ja hyvin kiinnitettyjä. Älykäs moottorinohjain ei käyttänyt normaalia profibus -liitintä, vaan sen kenttäväyläsovittimessa oli tulevaa väylää varten kiinteä kaapeli valetulla 5-napaisella M12-liittimellä, sekä paikka 5-napaiselle M12-liittimelle lähtevää väylää varten (kuvio 30). Koska älykkään moottorinohjaimen kenttäväyläsovitin tarvitsi ulkoisen jännitesyötön, oli lähtevä kaapeli jouduttu kytkemään riviliittimille tarvittavaa jännitesyöttöä varten ja myös väylää jatkettu siinä eteenpäin. Koska väyläkaapeli oli jouduttu jatkamaan kyseenalaisella tavalla, eikä väyläkaapeloinnin uusiminen ollut suuri tehtävä, päädyttiin uusimaan koko väyläkaapelointi. Samalla muutettiin kaapelointijärjestystä siten, että älykäs moottorinohjain siirrettiin väylän viimeiseksi, jolloin se on päättyvä laite, eikä väylää tarvitse enää jatkaa riviliittimillä.

Muutostöiden jälkeen älykkään moottorinohjaimen kenttäväyläsovitin ei enää saanut väyläisäntään yhteyttä lainkaan ja kaikki muut väylässä olevat orjalaitteet menivät vikatilaan moottorinohjaimen ollessa osana väylää. Moottorinohjain kytkettiin ainoana laitteena ja suoraan omalla kaapelilla väyläisäntänä toimivaan logiikkaan, mutta yhteyttä ei saatu. Moottorinohjaimen kenttäväyläsovitin todettiin rikkoutuneeksi ja se poistettiin väylästä, jonka jälkeen muut väylälaitteet alkoivat toimia normaalisti. ABB:ltä saatiin lahjoituksena uusi moottorinohjain UMC100.3, joka asennettiin keskukseen ja parametroitiin osaksi väylää.

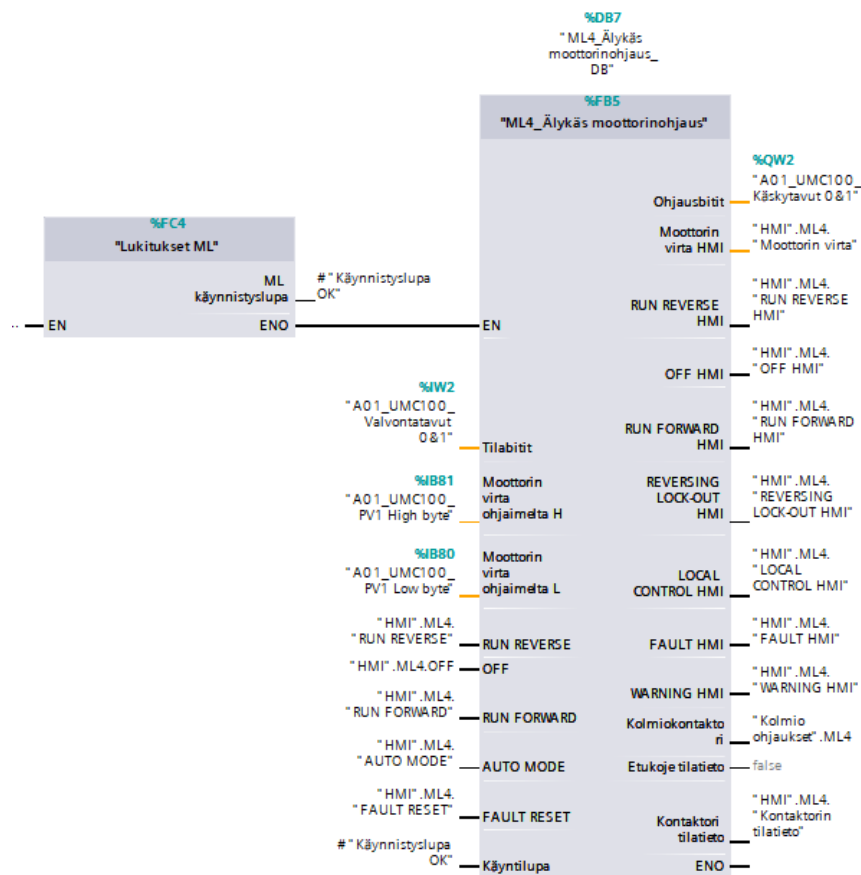


Kuvio 30. UMC22-FBP kenttäväyläliitin (FBP FieldBusPlug 2008)

9.4 Väylälaitteiden testaus

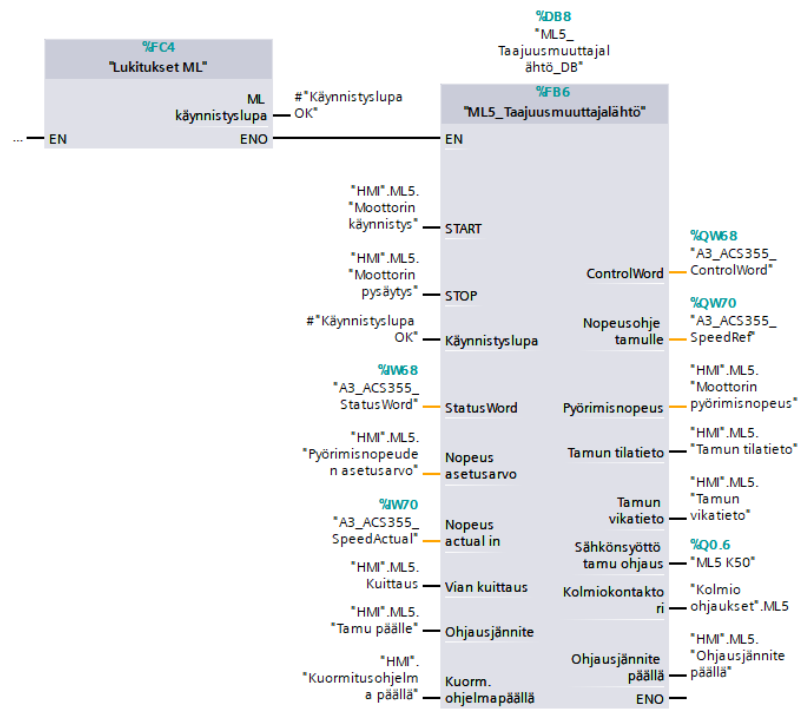
Kun väylä oli saatu pystytettyä, voitiin aloittaa sovellusohjelman testaaminen myös väylälaitteiden osalta. Testaaminen aloitettiin moottorilähdöistä, sillä näitä tarvittiin moottorin virran ja pyörimisnopeuden seurantaan. Taajuusmuuttajälähtö oli myös käytännössä välttämätön kuormitusohjelmien testauksessa, sillä moottorin pyörimisnopeutta ei ollut mahdollista säätää muilla lähdöillä. Koska älykäs moottorinohjain oli vikaantunut ja korvaavan laitteen saapumista jouduttiin odottamaan, priorisoitiin taajuusmuuttajälähdön toimintakuntoon saattaminen, jotta kuormitusohjelmia päästään testaamaan. Lisäksi, kun tiedettiin korvaavan moottorinohjaimen malli, voitiin verrata

ohjainten käyttöohjeita ja näin tarkistaa tarvitseeko sovellusohjelman älykkään moottorinohjaimen ohjelmalohkoon (kuvio 31) tehdä muutoksia vaihdoksen vuoksi. Tässä tapauksessa muutoksia ei tarvinnut tehdä. Taajuusmuuttajälähdön kanssa ei ollut ongelmia ja moottoria voitiin heti pyörittää, eikä sovellusohjelman taajuusmuuttajälähdön ohjelmalohkoon (kuvio 32) tehty tässä vaiheessa muita muutoksia kuin korjattiin nopeusohjeen ja nopeuden oloarvotiedon skaalaukset oikein. Kun moottorin ohjausten osalta oltiin valmiita, siirryttiin kuormamoottorin ohjausten testaukseen, jota kuvataan seuraavassa kappaleessa.



Kuvio 31. Älykäs moottorinohjain ohjelmalohko

Network 5: ML5 taajuusmuuttaja lähtö
 Comment

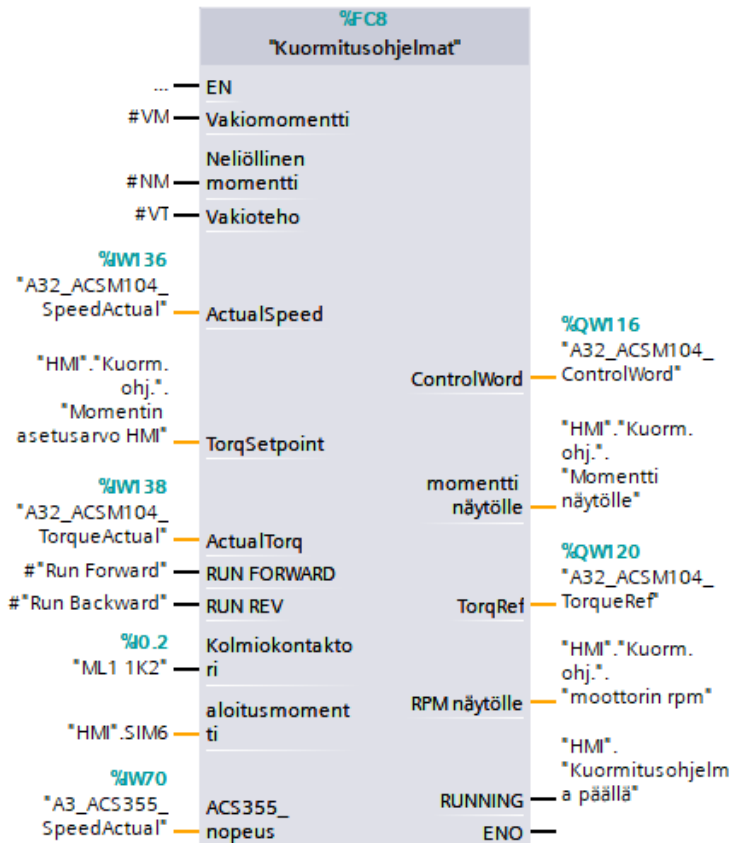


Kuvio 32. Taajuusmuuttajälähdön ohjelmalohko

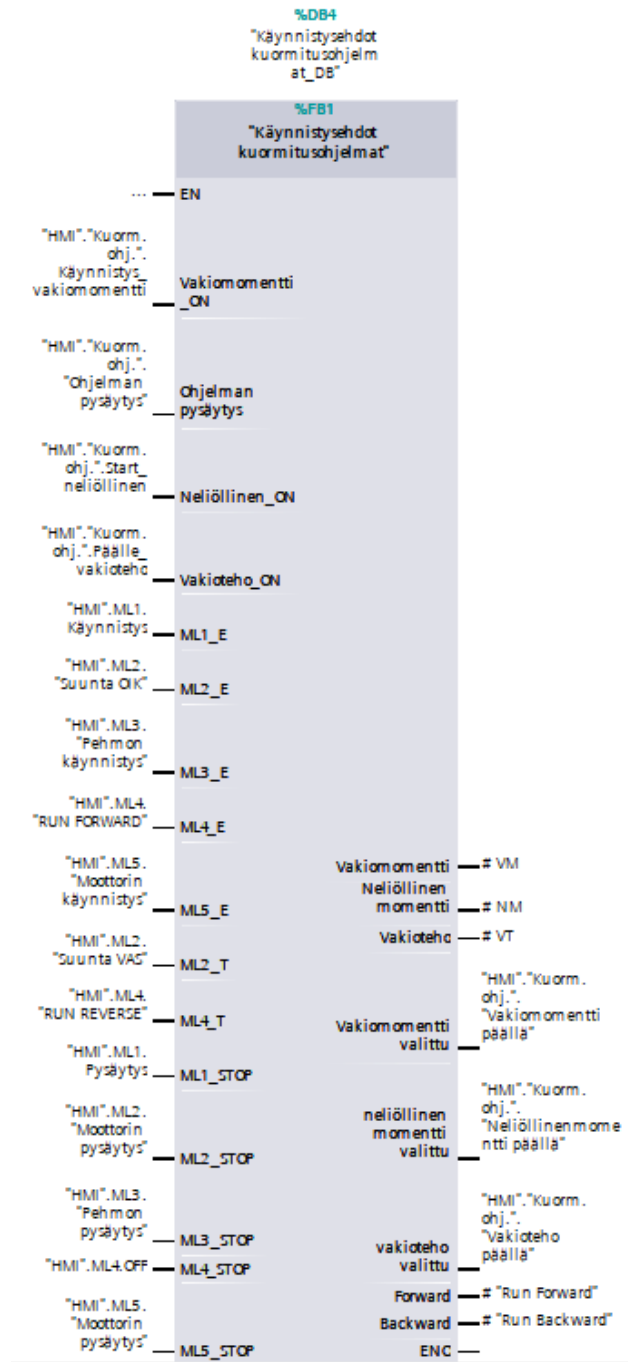
9.5 Kuormitusohjelmien testaus

Kuormitusohjelmista oli suunnitteluvaiheessa rakennettu TIA Portaliin erilliset toimintolohkot toimintolohkokaavioilla ja simuloimalla testattu momenttiohjeen muodostavien kaavojen toimintaa. Kuormitusohjelmista oli siis tehty melko alkeelliset pohjat, joissa oli vain taajuusmuuttajan tila- ja käskysanat, sekä ohje- ja oloarvosanat. Käyttöönottovaiheessa kuormitusohjelmia muokattiin siten, että ne siirrettiin omista toimintolohkoistaan yhteen toimintoon (kuvio 33). Nopeasti selvisi, että strukturoitu ohjauskieli oli kaikkein yksinkertaisin tapa muodostaa kuormitusohjelmat, sillä silloin välttyään monimutkaisilta toimintolohkokaavioilta. Omista toimintolohkoista samaan toimintoon siirtymiseen päädyttiin, sillä kuormitusohjelmat sisältävät enimmäkseen tietotyypin muunnoksia sekä laskentaa. Jokaisen kuormitustyyppin mukaista momentin ohjearvoa lasketaan toiminnossa jatkuvasti, mutta vain yhtä niistä kirjoitetaan väylään kuormamoottorin taajuusmuuttajalle, käyttäjän valinnan mukaan (ks. liite 5).

Koska toiminnossa ei ole muistia rakennettiin kuormitusohjelmien käynnistysehdot yhteen toimintolohkoon (kuvio 34), jota kutsuttiin kuormitusohjelmien toiminnossa. Kuormitusohjelmien testauksen yhteydessä myös käyttöliittymäpaneelin ulkoasu sai lopullisen muotonsa, kun järjestelmää testatessa selkiytyi mitä toimintoja on järkevää sijoittaa millekin sivulle (ks. liitteet 1, 2, 3 &4). Kuormitusohjelmien testauksen etenemistä käsitellään seuraavissa kappaleissa.



Kuvio 33. Kuormitusohjelmien ohjelmalohko.



Kuvio 34. Kuormitusohjelmien käynnistyshdot

9.5.1 Vakiomomenttiohjeen testaus

Kuormitusohjelmien testaus aloitettiin vakiomomentti-momenttiohjeella, sillä se oli kuormitustyypeistä yksinkertaisin. Taajuusmuuttajalla laitettiin moottori pyörimään maltillisella pyörimisnopeudella, jonka jälkeen käynnistettiin kuormamoottoria ohjaavan ACSM1 taajuusmuuttajan ohjaus ja

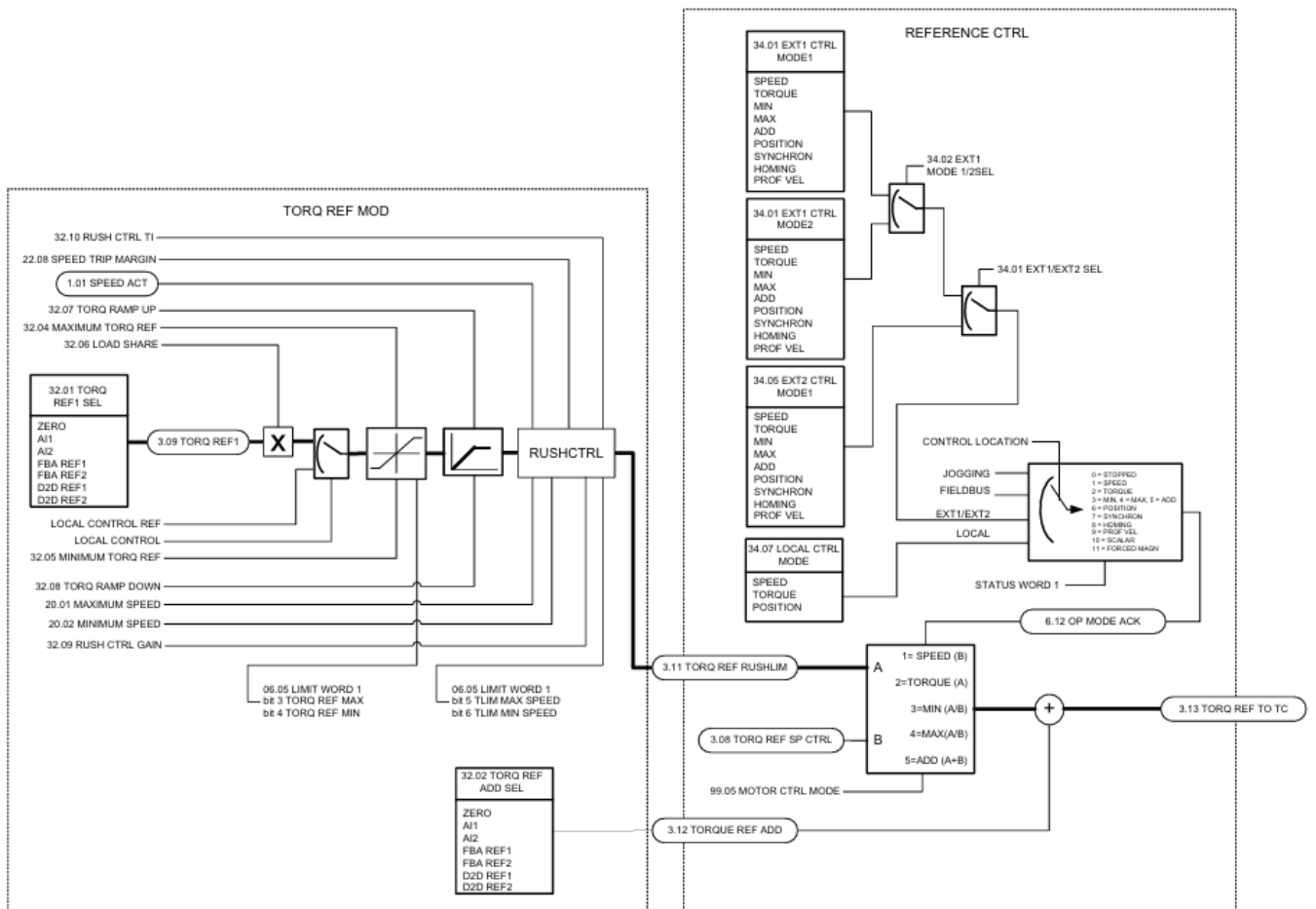
annettiin kuormitusohjelmalla sille vakiomomenttiohjetta. Huomattiin, että ACSM1 ei tottele annettua momenttiohjetta eikä ylipäänsä mene momenttiohjaustilaan, sen nopeuden oloarvotieto ei anna järjellistä lukemaa ja taajuusmuuttaja menee vikatilaan ja ilmoittaa vikakoodin, joka vastaa pyörimisnopeuden sallitun huippuarvon ylitystä.

Ensimmäisenä kytkettiin tietokone kiinni ACSM1:een ja avattiin DriveStudio parametrien monitorointia ja muokkaamista varten. Tarkistettiin taajuusmuuttajan parametreistä, että ne oli asetettu momenttiohjaukselle kenttäväylän kautta sopiviksi (kuvio 35) ja sallittu pyörimisnopeus on asetettu vastaamaan moottorin nimellisnopeutta. Seuraavaksi ACSM1:llä suoritettiin tunnistusajo, jonka tyypiksi valittiin paikallaan ajo, sillä kuormamoottorin akselia ei saatu helposti vapautettua moottorin akselista. Kun tunnistusajo oli tehty ja selvitystyötä jatkettiin, huomattiin että ACSM1 saa kyllä kenttäväylästä momenttiohjeen parametriin 3.09 (kuvio 35) ja oikean ohjaustavan parametriin 34 (kuvio 35), mutta se on silti parametrin 6.12 (kuvio 35) mukaan nopeusohjaustilassa. Tämän jälkeen heräsi epäily, että taajuusmuuttajassa sisällä oleva aiemmin kuormitusohjelmana toiminut DriveSPC-sovellusohjelma ajaa yli kenttäväylän kautta annetuista ohjeista, joten tutkittiin tätä sovellusohjelmaa.

Sovellusohjelmasta ei löydetty suoraa syytä miksi järjestelmä ei toimi, mutta koska tätä sovellusohjelmaa ei enää tarvittu nykyisessä järjestelmässä, päätettiin taajuusmuuttajaan ladata tehdasohjelma sen tilalle. Kun näistä toimenpiteistä huolimatta taajuusmuuttajaa ei saatu momenttiohjaustilaan, tarkastettiin parametrejä DriveStudiolla vielä tarkemmin. Moottori ohjattiin päälle taajuusmuuttajalähdöllä ja kuormamoottori ACSM1:llä ja DriveStudiosta havaittiin, ettei kuormamoottorin pyörimisnopeus vastaa moottorin pyörimisnopeutta, vaikka moottorin pyörimisnopeutta muutetaan. Lopulta käynnistys- ja pysäytysparametreistä löytyi parametri 10.15 JOG ENABLE, jolle oli annettu arvo C.TRUE, joka pakotti taajuusmuuttajan nopeusohjaustilaan ja ajamaan vakionopeutta. JOG ENABLE parametroidiin arvoon C.FALSE, jonka jälkeen ACSM1 saatiin momenttiohjaustilaan ja vakiomomentti-momenttiohje saatiin toimimaan. Koska nyt ACSM1 ei enää mennyt välittömästi vikatilaan, pyrittiin selvittämään, oliko myös tämä vika korjaantunut JOG ENABLEN parametri muutoksella. Nostettiin moottorin pyörimisnopeutta asteittain lähelle sen nimellisnopeutta, jolloin kuormamoottori alkoi ääntämään kummallisesti, ja silloin ACSM1 meni jälleen vikatilaan ja ilmoitti nopeusrajojen ylityksestä. Edelleen myöskään nopeuden oloarvotieto ei vastannut todellisuutta. Tässä vaiheessa tarkistettiin vielä kerran ACSM1:een asetetut moottori-,

nopeus, ja momenttiparametrit. Kun parametrien oli varmistettu olevan kuten pitää, päätettiin suorittaa moottorin tunnistusajo uudelleen, tällä kertaa normaalina tunnistusajona. Kun tunnistusajo oli suoritettu ei taajuusmuuttajaa enää mennyt vikatilaa ja nopeuden oloarvotieto alkoi toimimaan. Nyt voitiin siirtyä testauksessa seuraavaan kuormitusohjeeseen.

Momenttisäädön ohjausketju



Kuvio 35. Momenttisäädön ohjausketju. (ACSM1 Motion Control-ohjelma 2015).

9.5.2 Neliöllisen momenttiohjeen testaus

Kun vakiomomenttiohjeen testauksen aikana oli saatu ACSM1:n kanssa olleet ongelmat selvitettyä ja korjattua, sekä testattu sovellusohjelma vakiomomenttiohjeen osalta siirryttiin neliöllisen momenttiohjeen testaukseen. Moottoria pyöritettiin jälleen taajuusmuuttajalähdöllä ja kuormamoottoria ohjaavalle ACSM1:lle kirjoitettiin väylän kautta 10 % asetusarvoa momenttiohjeelle. Moottorin pyörimisnopeutta nostettiin aina nimellisnopeuteen asti ja todettiin momenttiohjeen kasvavan

neliöllisesti. Seuraavaksi kasvatettiin myös momenttiohjetta, mutta nyt huomattiin, että kun moottorin pyörimisnopeutta lähdettiin nostamaan taajuusmuuttajälähdöllä lähelle moottorin nimellisko nopeutta, meni taajuusmuuttajälähdön ACS355-taajuusmuuttaja vikatilaan. Vikakoodi paljasti syyksi tässäkin tapauksessa nopeusrajojen ylityksen.

Syytä lähdettiin selvittämään etsimällä raja-arvot, joilla ACS355 menee vikatilaan. Asetettiin ACS355 nopeusohjeen asetusarvoksi 20 % moottorin nimellisko nopeudesta ja lähdettiin nostamaan ACSM1:n momenttiohjeen asetusarvoa kirjaten ylös ACSM1:n nopeuden oloarvotiedon, virran, sekä momentin arvot. Havaittiin, että vikatila ilmaantuu, kun ACSM1:n momenttiohjetta nostetaan yli 20 %:n maksimimomentista. Ylös kirjattuja arvoja tutkimalla nähtiin, että ACSM1:n nopeuden oloarvotieto on kasvanut, kun momenttiohjetta on nostettu. Nopeuden oloarvotiedon olisi pitänyt laskea, kun momenttiohjetta kasvatetaan, sillä ACS355 ohjaa moottoria staattisella nopeudella, jolloin kuorman kasvaessa moottorin pyörimisko nopeus laskee. Moottorin ja kuormamoottorin ollessa akseleistaan yhteen kytkettyinä (kuvio 20) tulisi tämän pyörimisko nopeuden laskun näkyä myös ACSM1:n nopeuden oloarvotiedossa, sillä moottorit pyörivät samalla pyörimisko nopeudella.

Selvisi siis, että kuormamoottori ei jarruta moottoria, vaan kiihdyttää sitä. Kuvioista 35 nähdään, kuinka ACSM1 toteuttaa momenttisäätöä. Siinä parametri 3.09 on taajuusmuuttajalle annettava momentin asetusarvo ja parametri 1.01 on nopeuden oloarvo. Muut kuviossa 35 esiintyvät parametrit ovat kiinteästi asetettuja, mutta näiden kahden parametrin arvo voi olla positiivinen tai negatiivinen, riippuen käyttäjän asettamasta asetusarvosta ja moottorin pyörimisko suunnasta. Jotta kuormamoottori jarruttaisi moottorin pyörimistä, tulee nopeuden oloarvon ja momentin asetusarvon olla etumerkiltään erilaisia. Koska moottorin pyörimisko suuntaa voidaan yhdellä moottorilähdöistä vaihtaa, päädyttiin järjestelmän käyttämisen yksinkertaistamiseksi siihen, että käyttäjän asettama momentin asetusarvo rajataan sovellusohjelmassa pelkkiin positiivisiin arvoihin ja ohjelmallisesti toteutetaan asetusarvon etumerkin määrittäminen (kuvio 36). Ohjelmamuutosten jälkeen ACS355 ei enää mennyt vikatilaan, moottoreiden pyörimisko nopeus laski momentin asetusarvoa nostettaessa ja neliöllinen momenttiohje toimi koko momentin asetusarvo- sekä pyörimisko nopeusalueella. Kaikkiin edellä kuvattuihin ACSM1-taajuusmuuttajan vianselvitystöihin saatiin asiantuntijatukena ABB:n Antti Kuposelta. Neliöllisen momenttiohjeen toimintaan oltiin tyytyväisiä ja voitiin siirtyä viimeisen kuormitusohjeen testaukseen.

```

1  (* ACSM1-04 OHJAUS*)
2
3  // ASETUSARVOJEN & ACTUAL ARVOJEN RAJOITUKSET, TIETOTYYPPIMUNNOKSET & SKAALAUKSET
4  #LIM_Torqsp := LIMIT(MN := 0, IN := #TorqSetpoint, MX := 100.0); // rajoitetaan momentin asetusarvo 0-100%
5
6  #"momentin asetusarvo Nm" := #LIM_Torqsp * 0.0732007; // skaalataan momentin asetusarvo 0-7,32007Nm

24 //Momenttiohjeen etumerkin käsittely => acsm1:lle annetaan positiivinen ohje jos speed actual on negatiivinen, negatiivinen jos se on positiivinen
25 IF #”RUN FORWARD” = TRUE THEN
26   #MomentinSP_etumerkki := 1.0 * #"momentin asetusarvo Nm";
27 END_IF;
28
29 IF #”RUN REV” = TRUE THEN
30   #MomentinSP_etumerkki := -1.0 * #"momentin asetusarvo Nm";
31 END_IF;
32

```

Kuvio 36. Momenttiohjeen asetusarvon etumerkin hallinta

9.5.3 Vakioteho momenttiohjeen testaus

Vakiomomentin ja neliöllisen momentin mukaisten momenttiohjeiden testauksessa oli saatu kaikki ilmenneet viat korjattua järjestelmästä, joten vakiotehoa noudattavan momenttiohjeen testauksessa voitiin keskittyä puhtaasti sovellusohjelman toimintaan.

Taas moottoria pyöritettiin taajuusmuuttajälähdön ACS355:llä ja kuormamoottoria ohjaavalle ACSM1:lle annettiin momentin asetusarvo. Vakiotehon tapauksessa momentin tulisi kasvaa pyörimisnopeuden laskiessa kaavan 3 mukaisesti. Momentin asetusarvo asetettiin 100 %:n ja moottorin pyörimisnopeutta lähdettiin laskemaan nimellisnopeutta vastaavasta 100 %:sta kohti nolaa ja seurattiin keskuksen käyttöliittymäpaneelilta momentin oloarvotiedon muutosta. Momentin oloarvotieto kasvoi pyörimisnopeuden laskiessa, mutta kun moottoria pyörittävälle ACS355:lle annettavaa nopeuden asetusarvoa laskettiin, huomattiin 20 %:n asetusarvolla ACS355 ei enää pysty pyörittämään moottoria kuormamoottoria vasten, vaan se menee ylivirrasta johtuvaan vikatilaan.

Taajuusmuuttajälähdön toimintolohkoon tehtiin sovellusohjelmassa muutos, jolla rajoitettiin minimi pyörimisnopeus 22 %:n vakioteho- tai vakiomomentti momenttiohjeita käytettäessä ja jatkettiin testausta. Enää ACS355 ei mennyt ylivirrasta johtuvaan vikatilaan, mutta kun momentin oloarvotiedon muuttumista seurattiin tarkemmin, se muuttui kyllä oikeansuuntaisesti muttei vaikuttanut noudattavan vakiotehon kaavaa. Asia päätettiin selvittää asettamalla momentin asetusarvoksi 100 % ja lähtemällä laskemaan moottorin pyörimisnopeutta 100 %:sta alaspäin ja kirjaa-

malla 25 %:n välein pyörimisnopeus ja momentin oloarvotieto ylös. Mittausten tuloksista piirrettiin Excel-taulukkolaskenta ohjelmalla kuvaaja ja havaittiin kuvaajan olevan lineaarinen. Vakiotehon momenttiohjeen laskentaa tarkistettiin sovellusohjelmassa ja korjattiin sieltä laskennassa olleet virheet, jonka jälkeen momenttiohjeen laskenta oli kuvion 37 mukainen ja se alkoi noudattaa oikeanlaista säätökäyrää.

```

35 //Vakiotehonmomenttiohjeen laskenta alkaa
36 #Torq_(Sp/TN)/100 := #Torq_Sp/TN" ^ 100.0; // kerrotaan asetusarvon ja nimellismomentin suhdeluku 100:lla => saadaan prosentuaalinen arvo kertoimena käytettäväksi
37
38 #Torq_(Sp/100)^100 := #Torq_Sp/TN" ^ 100.0; // Kerrotaan asetusarvon ja nimellismomentin suhdetta sadalla => 0-100
39
40 #Vakiotehomomenttityhteen := #Torq_(Sp/100)^100" ^ #Torq_Sp/TN"; // Kerrotaan momentti kertoimet yhteen
41
42 #nr/n" := #moottorin nimellis rpm" / #ActualSpeed 0-100%; // jaetaan nimellismoisuus(100%) skaalatulla todellisella nopeudella(0-100%)
43
44 // Varmistetaan että nimellisen ja todellisen nopeuden suhdeluku on positiivinen
45 IF #nr/n" < 0.0 THEN
46   #nr/r itseisarvo" := -1 ^ #nr/n";
47 ELSE
48   #nr/r itseisarvo" := #nr/n";
49 END_IF;
50
51 #Nopeuskerroin := (#nr/r itseisarvo" - #nr/n_MIN" ) / ( #nr/n_MAX" - #nr/n_MIN" ); // skaalataan nopeuskerroin
52
53 #VakiotehoMomentOut := #Nopeuskerroin ^ #Vakiotehomomenttityhteen; // muodostetaan vakiotehon momenttiohje kertomalla nopeustekijällä momenttitekijää
54
55 #VakiotehoRefscaled := 100.0 ^ #VakiotehoMomentOut; // skalataan vakiotehon momenttiohje 0-100 => 0-10000
56
57 #Torq_Vakioteho := REAL_TO_INT(#VakiotehoRefscaled); // muutetaan skalattu vakiotehon ohje integeriksi
58 //Vakiotehon momenttiohjeen laskenta loppuu

```

Kuvio 37. Vakioteho momenttiohjeen laskenta.

9.5.4 Järjestelmän testaus

Kun kuormitusohjelmat oli saatu toimimaan halutunlaisesti taajuusmuuttajalähtöä moottorin pyörittämiseen käyttäen, siirryttiin testaamaan myös muita moottorilähtöjä kuormitusohjelmien kanssa. Nopeasti huomattiin, että sovellusohjelmaan on asetettava ehtoja kuormamoottorin ohjaukselle. ACSM1:n noudattaessa nopeusohjauksen sijaan momenttiohjausta lähtee kuormamoottorin pyörimisnopeus kiihtymään välittömästi kohti asetettua nopeusrajaa (kuvio 35) sitä suuremmalla kiihdytyksellä, mitä suurempi momenttiohje ACSM1:lle on annettu. Oli siis tehtävä sovellusohjelmaan ehtoja (kuvio 34) joilla estetään momenttiohjeen kirjoittaminen ACSM1:lle, ennen kuin käyttäjä on valinnut käytettävän moottorilähdön. Sovellusohjelmaan tehtiin myös muutos, jolla kuormamoottorilta katkaistaan ohjauksikäsky, jos moottori pysäytetään ennen kuormamoottoria niiden molempien pyöriessä, jotta myös kuormamoottori pysähtyy eikä lähde kiihdyttämään pyörimisnopeuttaan kohti asetettua maksiminopeutta.

Tähtikolmiokäynnistimen, suunnanvaihtolähdön ja pehmokäynnistimen testauksen jälkeen myös uusi älykäs moottorinohjain saatiin asennettua keskukseseen ja voitiin aloittaa sen testaus. Moottorinohjaimelle ladattiin TIA Portaliin uusi laitekohtainen kuvaustiedosto, ja konfiguroitiin laite väylään. Moottorinohjain parametroidiin toimimaan suorakäynnistimenä ja testattiin käynnistää sillä moottoria. Tia Portalista valvomalla voitiin huomata, että tila- ja käskytavujen bitit muuttuvat kyllä, mutta moottorinohjaimen tilatavun mukaan se oli käynnistyskäskystä huolimatta pysähdystilassa eikä moottori näin ollen lähtenyt pyörimään. Parametrejä tutkimalla huomattiin, että digitaalitulot pakottavat ohjaimen pysähdystilaan, joten ne parametroidiin pois käytöstä, jonka jälkeen moottori saatiin pyörimään. Koska älykästä moottorinohjainta käytetään järjestelmässä myös moottorin virran valvontaan, huomattiin sen lähettämän virtatiedon poikkeavan moottorin kilpi-tietojen perusteella lasketusta virrasta. Moottorinohjaimen parametreista korjattiin virtakerroin oikein, jonka jälkeen virranmittaus alkoi toimimaan oikein.

Kun kaikki moottorilähdöt ja kuormitusmomenttityypit oli saatu toimimaan ja järjestelmään oltiin tekemässä viimeisiä testauksia ennen toimeksiantajalle luovutusta, hajosi taajuusmuuttajalähdön ACS355 taajuusmuuttaja, kun sille kytkettiin syöttöjännite. Uutta taajuusmuuttajaa ei ehditty hankkimaan enää tässä vaiheessa työtä, joten toimeksiantajan kanssa sovittiin, että keskus luovutetaan ilman taajuusmuuttajalähtöä ja taajuusmuuttajan asentaminen ja liittäminen järjestelmään toteutetaan erillisenä projektina.

10 Pohdinta

Työn tavoitteena oli vastata työn tutkimuskysymyksiin, eli selvittää keskuksen tilanne, sujuvoittaa keskuksen käyttöä opetuksessa sekä edistää keskuksen jatkokehittämisen lähtökohtia. Työn kannalta ensimmäinen tutkimuskysymys eli keskuksen nykytilanteen selvittäminen oli työn kannalta tärkein, sillä se määrittä työn laajuuden. Kysymykseen lähdettiin etsimään vastausta tutustumalla keskukseseen paikan päällä laboratorioissa, sekä tutkimalla olemassa olevia dokumentteja ja ohjelmia. Selvisi, että keskuksen kehitystarve oli oletettua huomattavasti suurempi. Dokumentoinnin versiointi ei ollut ajan tasalla, olemassa oleviin logiikkaohjelmiin ei päästy käsiksi lainkaan, sekä kenttäväylälaitteiden parametroinnissa oli puutteita.

Toisen tutkimuskysymyksen tavoite oli selvittää keskuksen käytön sujuvoittamista. Käytön suurin epäkäytännöllisyys havaittiin liittyvän kuormitustyyppien vaihtamiseen. Kuormitustyyppiä vaihtamiseen täytyi erikseen ladata tietokoneelta kuormitustyyppin ohjelma taajuusmuuttajaan. Tämä aiheutti haasteita laboratoriotöiden suorittamisessa niille varatussa aikataulussa. Toteutustapa lisäsi inhimillisen virheen riskiä, esimerkiksi taajuusmuuttajan parametrien sekoittamisesta. Havaittujen ongelmien poistamiseksi päädyttiin toteuttamaan kuormitustyyppit siten, että ne siirrettiin ohjelmoitavan logiikan sovellusohjelmaan. Kuormitustyyppien vaihtaminen ja hallinta siirrettiin toteutettavaksi käyttöliittymäpaneelilta. Toimenpiteillä helpotetaan laitteiston käyttöä, jonka seurauksena käyttäjälle jää enemmän aikaa varsinaisen laboratoriotyön suorittamiseen.

Kolmannen tutkimuskysymyksen tavoitteena oli luoda järjestelmän jatkokehitykselle vahva pohja. Tavoitteeseen pääsemiseksi luotiin keskuksen piirikaavioista ajan tasainen versio, ohjelmoita logiikka ja käyttöliittymäpaneeli uusittiin, sovellusohjelma tehtiin uusiksi ja koodia kommentoitiin mahdollisimman kattavasti. Kaikista keskuksen liittyvistä dokumenteista ja ohjelmista koottiin kansio, joka tallennettiin muistitikulle ja useamman eri henkilön tietokoneelle.

Suurimpia hankaluuksia työssä tuottivat alkuperäisen keskuksen toiminnan ja tilan heikko tuntemus. Tämä ratkaistiin tutustumalla keskuksen, sekä olemassa oleviin dokumentteihin. Koska kaikkea keskuksen toimintaan liittyvää tietoa ei ollut tallessa, jouduttiin puuttuvat dokumentit ja ohjelmat luomaan uusiksi. Lisäksi iäkkäät laitteet hajosivat odottamattomasti työn edetessä, josta seurasi uusia vikoja. Koska laiterikkoihin ei voitu etukäteen varautua, ne täytyi tilannekohtaisesti selvittää ja korvata rikkoutuneet laitteet mahdollisuuksien mukaan. Edellä mainittujen seikkojen vuoksi työn laajuuden sekä toteutukseen käytettävän ajan arviointi etukäteen oli erittäin haastavaa. Järjestelmä saatiin toimimaan toimeksiantajaa tyydyttävällä tasolla ja puuttuvat ominaisuudet ja laitteet päätettiin toimeksiantajan kanssa jättää tämän opinnäytetyön ulkopuolelle. Ensimmäisenä jatkokehityksen kohteena on taajuusmuuttajalähdön toimintakuntoon saattaminen, kun taajuusmuuttaja on hankittu. Keskuksen jatkuvan kehittämisen edistämiseksi ehdotettiin työaseman järjestämistä keskuksen välittömään läheisyyteen. Jatkokehitystä ajatellen ehdotettiin myös, että kirjataan ylös keskuksen käytön myötä esiin nousevia parannuskohteita ja -ideoita.

Työ toteutettiin soveltavana tutkimuksena eli tutkimuksellisenä kehittämistyönä. Tutkimuksella kehittämistyöllä ratkaistiin olemassa olevassa ympäristössä ilmennyt ongelma käytäntöä ja

teoriaa soveltamalla. Tämä tarkoitti ongelman paloittelua pienempiin osiin, joita ratkottiin yksitellen ja erikseen. Ongelmien ratkaisemiseksi hyödynnettiin aineiston analysointia, konkreettista kehittämistoimintaa sekä tutkimusmenetelmien soveltamista. Tutkimuksellisen kehittämistyön pohjana pyrittiin käyttämään mahdollisimman monipuolisesti eri lähteitä, mutta sitä rajoitti työn kohteena olevien laitteiden keskittyminen kahdelle valmistajalle. Työn tuloksia voidaan hyödyntää keskuksen jatkokehityksen pohjana. Konkreettinen kehitystoiminta tarkoitti tässä työssä ohjelmakoodin tekemistä parhaan sen hetkisen ymmärryksen perusteella, jonka jälkeen ohjelman toimintaa testattiin. Mikäli ohjelma ei toiminut toivotulla tavalla, ohjelmakoodia muokattiin ilmenneen ongelman perusteella toivottuun suuntaan. Työssä käytettiin pääosin kvalitatiivisia menetelmiä ohjelman toimivuutta testattaessa, mutta niihin yhdistettiin kvantitatiivisia menetelmiä tarkistettaessa ohjelman tuottaman momentin asetusarvon oikeellisuutta.

Lähteet

ABB High performance machinery drives. 2010. ACSM1-sarjan taajuusmuuttajat 0,75–160 kW tuoteluettelo. Viitattu 22.10.2024. https://library.e.abb.com/public/6f55f54ae9b66610c12576f8002757fe/ACSM1catalogREVE_FI.pdf.

ACSM1 Motion Control-ohjelma. 2015. Ohjelmointiopas. Viitattu 5.5.2025. https://library.e.abb.com/public/21cbe5f081a0409faff29ba088e1d5b8/FI_ACSM1_motion_control_FW_H_A4.pdf

Ahoranta, J. 2020. Sähkötekniikka. 15.–18. p. Helsinki: Sanoma Pro.

Chabauty, D. N.d. Profibus Design and Good Practices. Viitattu 22.5.2025. <https://emersonexchange365.com/products/control-safety-systems/m/deltav-mediagallery/3904/download>.

DriveStudio User manual. 2012. Käyttöohje. Viitattu 20.5.2025. <https://search.abb.com/library/Download.aspx?DocumentID=3AFE68749026&LanguageCode=en&DocumentPartId=1&Action=Launch>.

DriveSPC User manual. 2009. Käyttöohje. Viitattu 20.5.2025. <https://search.abb.com/library/Download.aspx?DocumentID=3AFE68836590&LanguageCode=en&DocumentPartId=1&Action=Launch>.

FBPA-01 PROFIBUS DP adapter module. 2024. Käyttöohje. Viitattu 4.5.2025. <https://search.abb.com/library/Download.aspx?DocumentID=3AFE68573271&LanguageCode=en&DocumentPartId=1&Action=Launch>.

FBP FieldBusPlug. 2008. Käyttöohje. Viitattu 20.4.2025. <https://library.e.abb.com/public/d8a83af3e7630601c12574030037a954/2CDC135001D0210.pdf>.

Hautamäki, J., Hyvätti, S., Kirjalainen, E., Kokko, M., Korhonen, K., Liimatainen, L., Laitinen-Väänänen, S., Norvapalo, K. & Törn-Laapio, A. 2024. Eettiset periaatteet. Jyväskylän ammattikorkeakoulun eettisten periaatteiden ohjeistus. Viitattu 23.5.2025. <https://www.jamk.fi/fi/media/41520>.

Hyvä tieteellinen käytäntö ja sen loukkausepäilyjen käsittely Suomessa. 2023. HTK-ohje. Viitattu 31.5.2025. https://tenk.fi/sites/default/files/2023-03/HTK-ohje_2023.pdf.

Pernaa, J. 2013. Kehittämistutkimus tutkimusmenetelmänä. Kehittämistutkimusta tutkimusmenetelmänä käsittelevä artikkeli Helsingin yliopiston verkkosivuilla. Viitattu 23.5.2025. <https://helda.helsinki.fi/server/api/core/bitstreams/fd4fcd23-2d7c-474a-a426-438c93075ff3/content>.

Pehmokäynnistinopas. 2011. Tekninen opas. Viitattu 21.5.2025. chrome-extension://efaidnbmn-nibpcajpcglclefindmkaj/https://library.e.abb.com/public/d11f99611045fef8c125796e00473a8a/OPAS%20Pehmokaynnistys%20FI12_01.pdf.

Simatic S7-1200 Programmable controller. 2012. Järjestelmä manuaali. Viitattu 18.5.2025. chrome-extension://efaidnbnmnnibpcajpcglclefindmkaj/https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf.

Tekninen opas Nro1. 2001. Opas suorasta momentinsäädöstä. Viitattu 20.5.2025. <chrome-extension://efaidnbnmnnibpcajpcglclefindmkaj/https://library.e.abb.com/public/d0212381cc29489d96bb455cf16fa101/Tekninenopasnro1.pdf?x-sign=clnS1tTvkW5XrMPzxW3xPIQ02DreFbl1RXcu/ciB4LDIQ/Q5/wFDmqszTP3HjLMA>.

Tekninen opas Nro7. 2001. Opas sähkökäytön mitoittamiseen. Viitattu 20.5.2025. <https://search.abb.com/library/Download.aspx?DocumentID=3BFE64444841&LanguageCode=fi&DocumentPartId=1&Action=Launch>.

Totally Integrated Automation Portal – Always ready for tomorrow. N.d. TIA Portal tietopaketti Siemensin verkkosivuilla. Viitattu 16.5.2025. <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>.

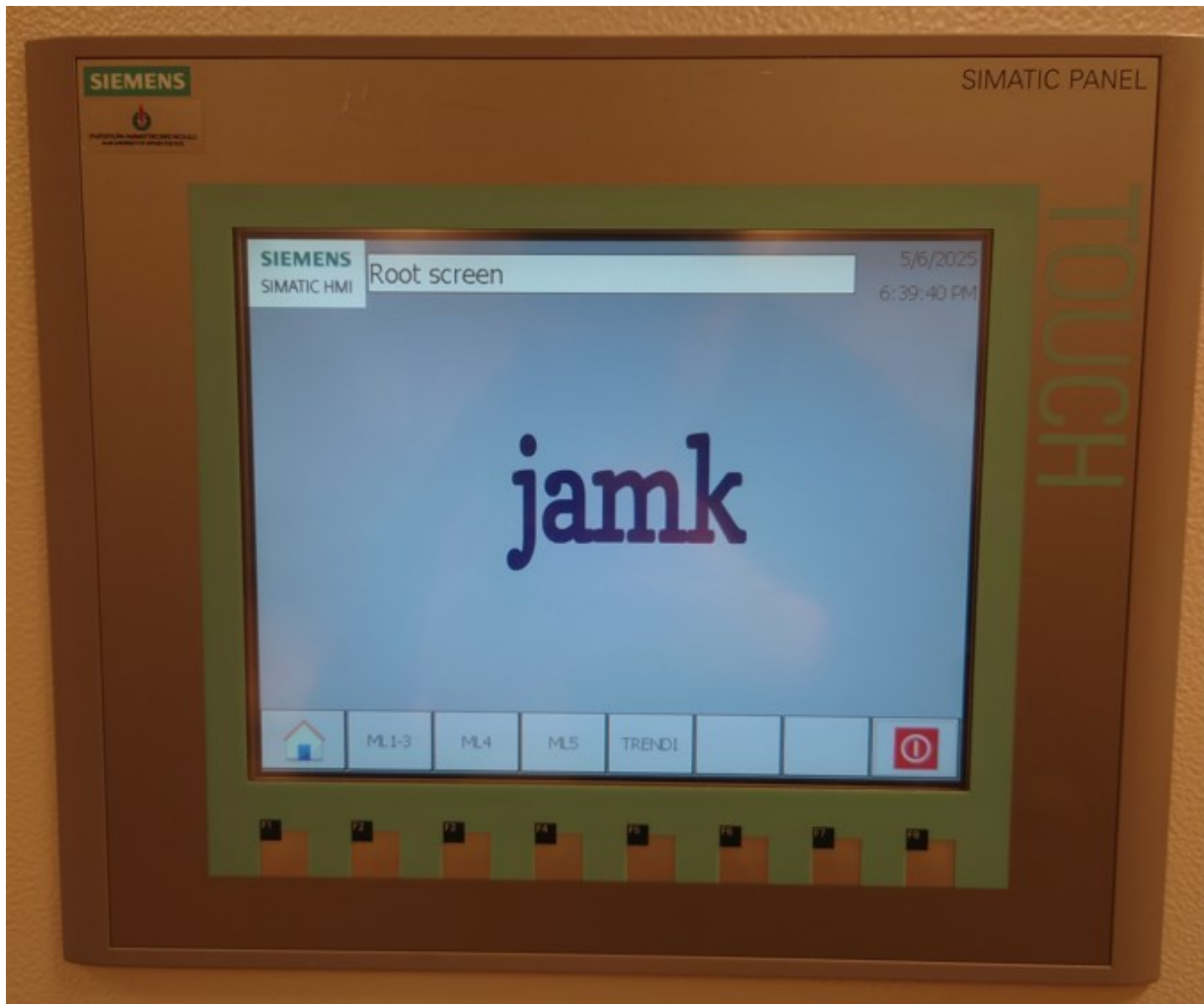
Universal Motor Controller UMC100.3. 2024. Käyttöohje. Viitattu 23.5.2025. <https://search.abb.com/library/Download.aspx?DocumentID=2CDC135032D0204>.

Universal Motor Controller 100.3. N.d. Esite moottorien älykkästä ohjaamisesta ABB:n verkkosivuilla. Viitattu 23.5.2025. <https://new.abb.com/low-voltage/fi/tuotteet/moottoriohjaimet/umc/umc100>.

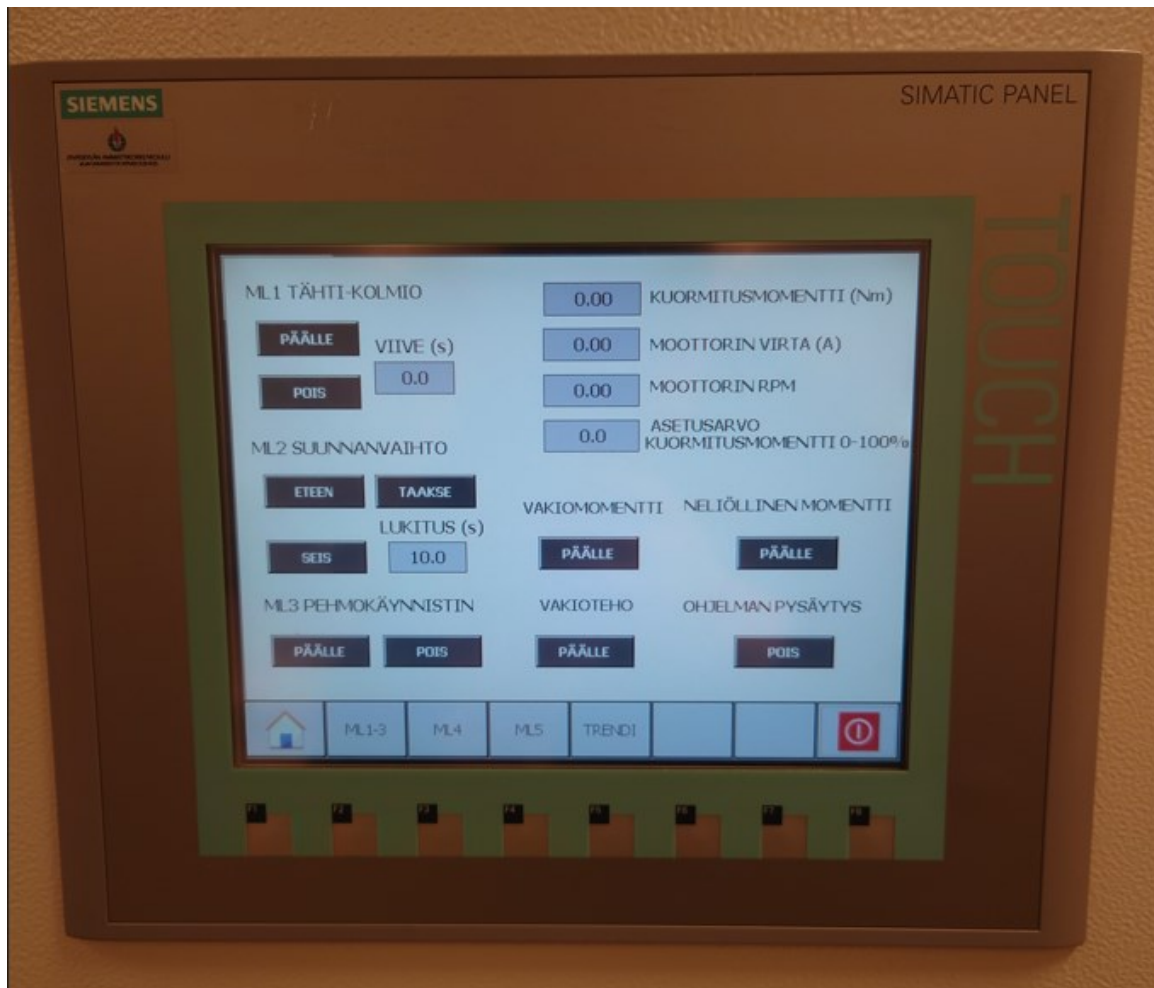
What is fieldbus?. 2019. Blogiteksti kenttäväylystä. Viitattu 22.5.2025. <https://www.realpars.com/blog/fieldbus>.

Liitteet

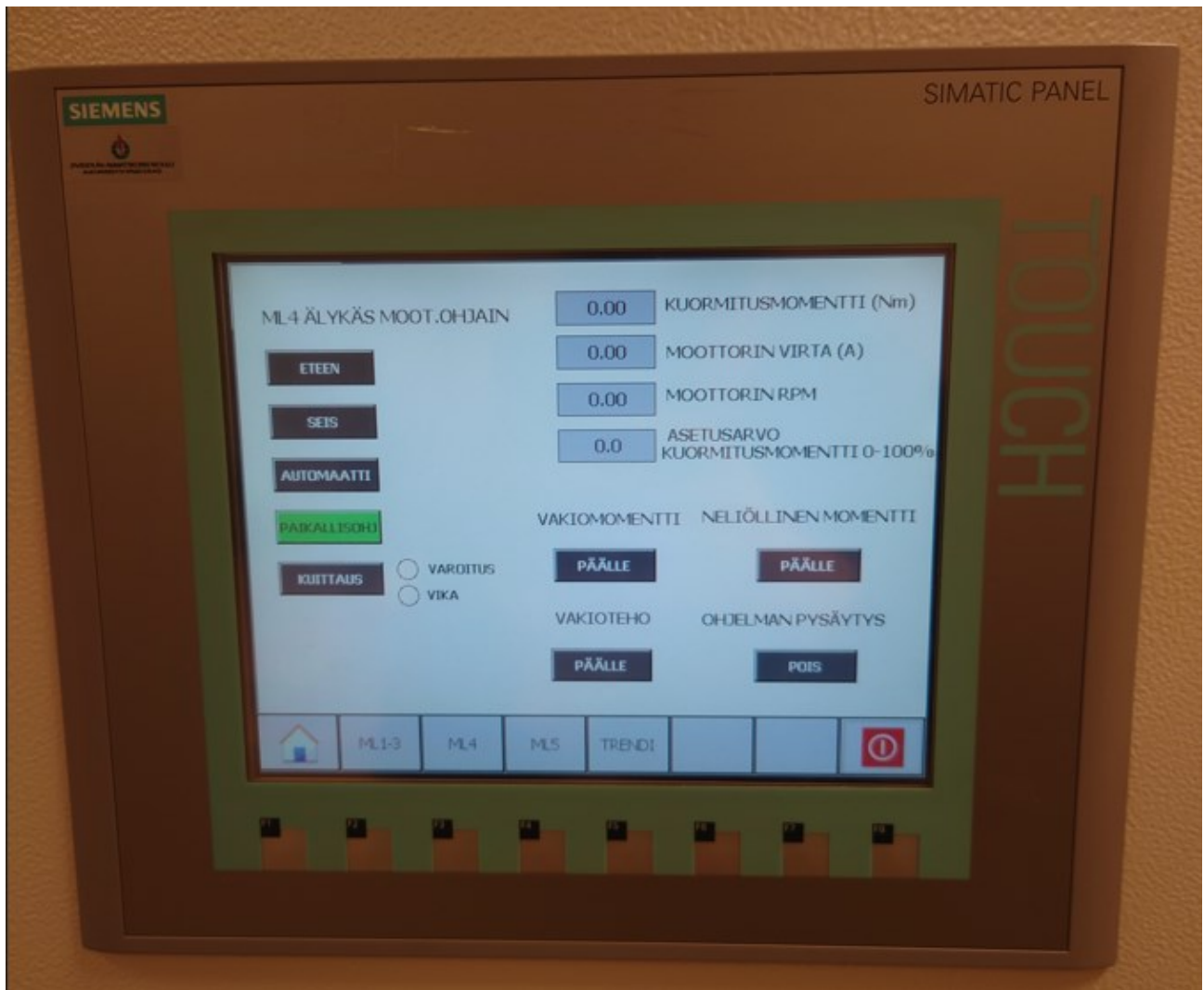
Liite 1. Käyttöliittymäpaneelin kotisivunäkymä



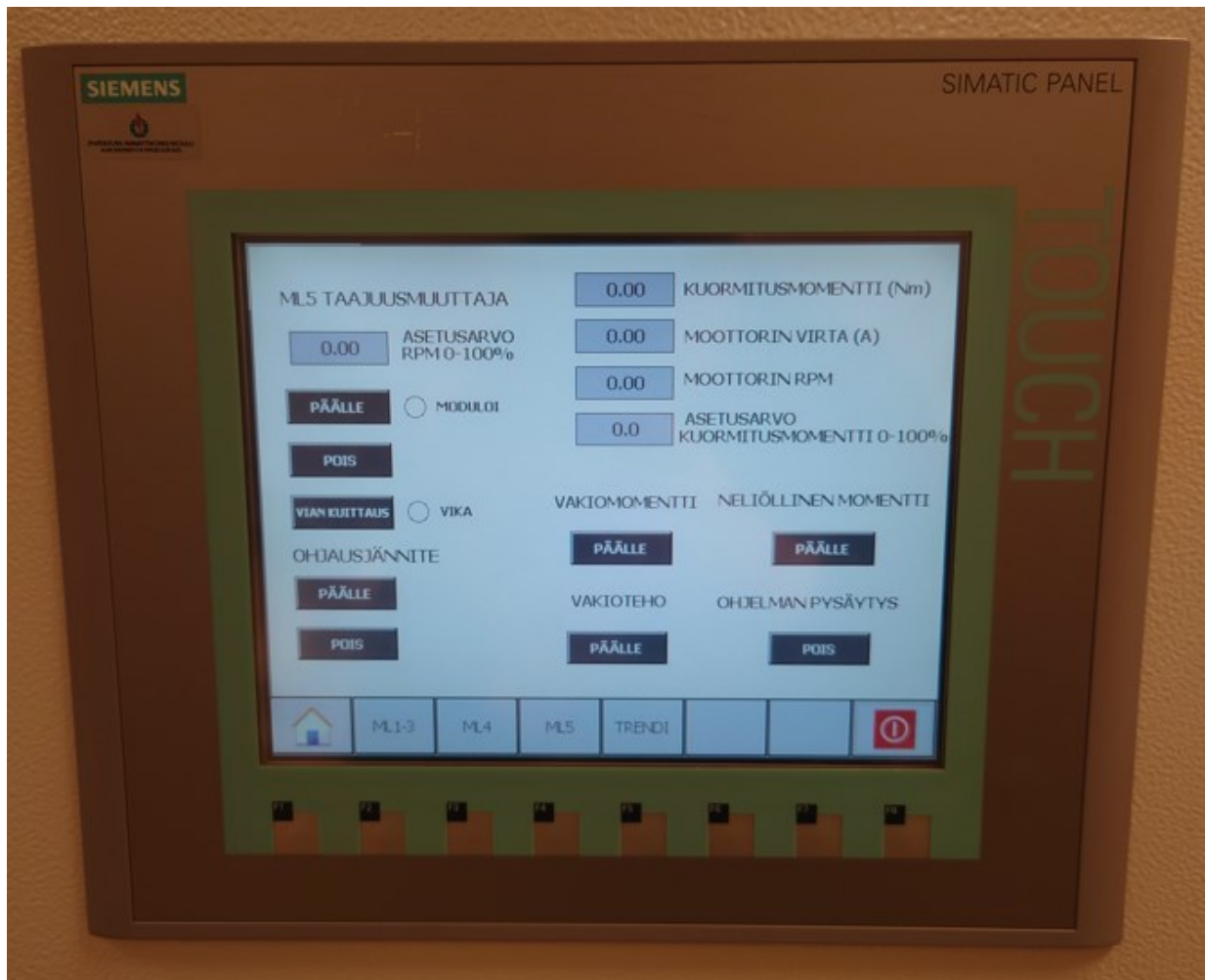
Liite 2. Käyttöliittymäpaneelin ML1-3 sivunäkymä



Liite 3. Käyttöliittymäpaneelin ML4 sivunäkymä



Liite 4. Käyttöliittymäpaneelin ML5 sivunäkymä



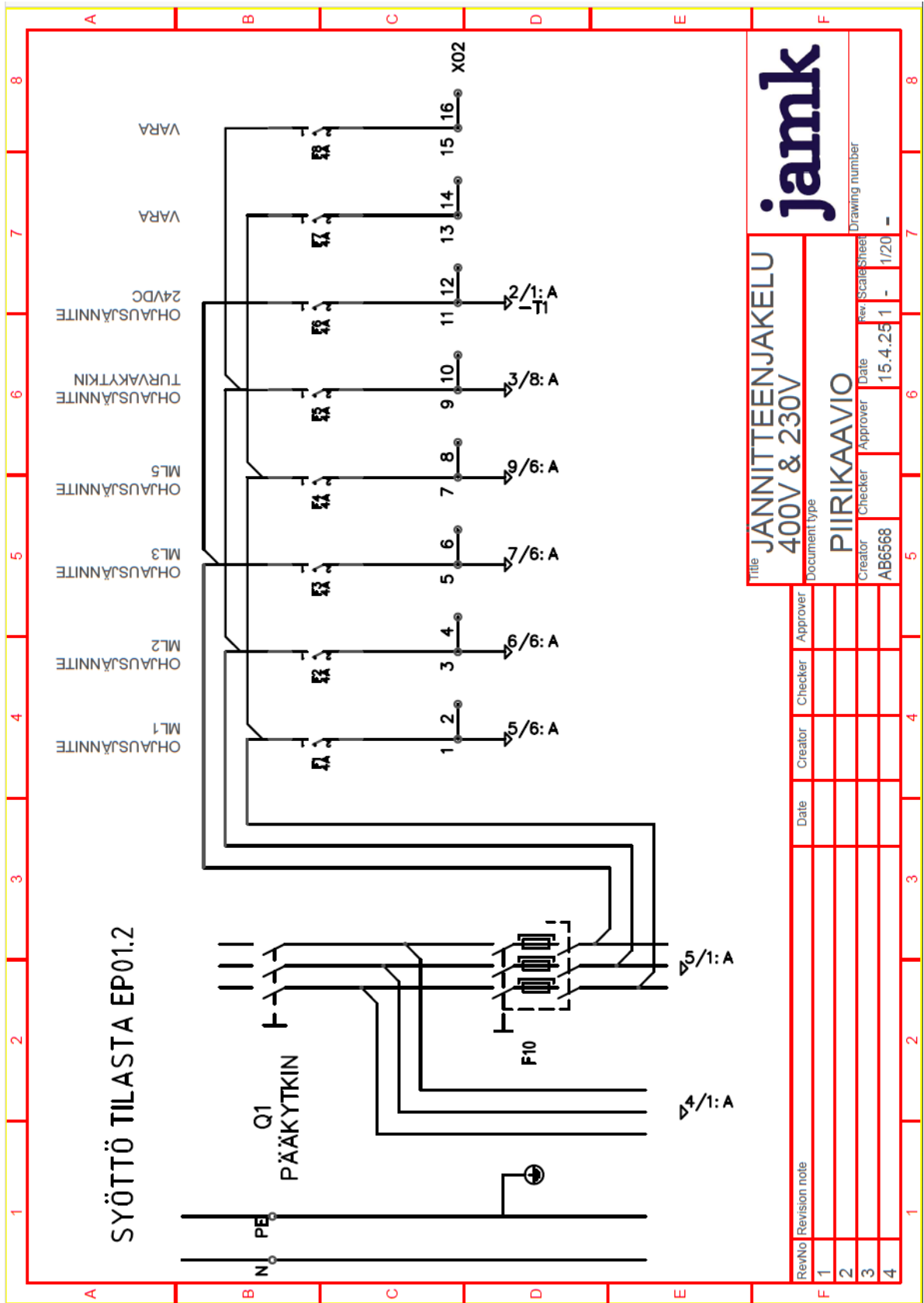
Liite 5. Kuormitustyyppin valinta koodissa

```

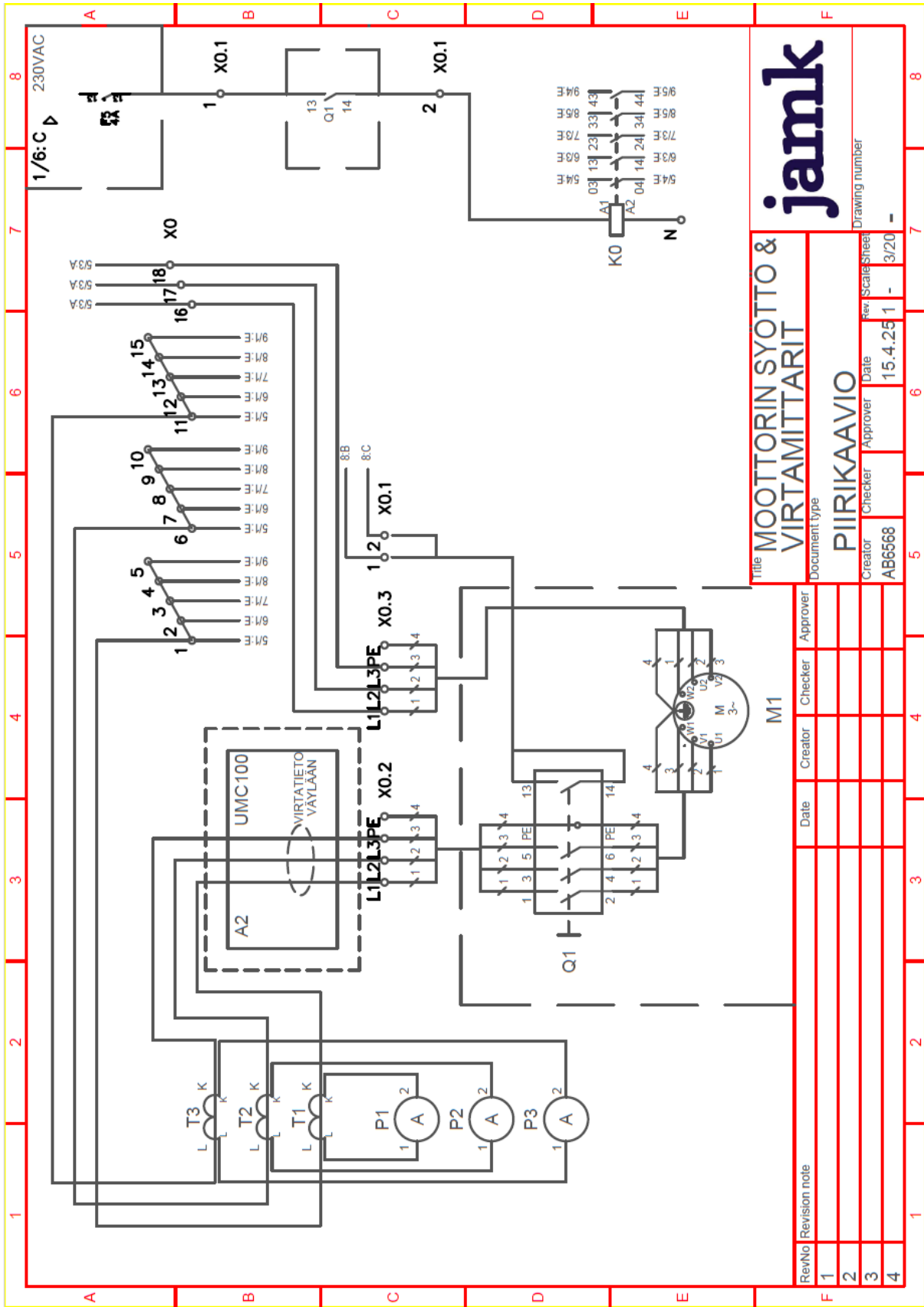
75
76 □ (^
77 Kuormitusohjelmien valinta & momenttioshjeiden käsittely
78 □ ^)
79
80 □ IF
81 //Vakiomomentti kuormitusohjelma
82 #Vakiomomentti = TRUE AND #Kolmiokontaktori = TRUE THEN // Vakiomomentti täytyy olla valittuna ja kolmiokontakan vetää
83
84 #Vakiomomentti_Temp := #MomentinSP_etumerkki * #Vakiomomentin kerroin; // kerrotaan momentin asetu-arvoa (0-7,32Nm) kertoimella 1366.107
85
86 #Torq_Vakiomomentti := REAL_TO_INT(#Vakiomomentti_Temp); // käännetään vakiomomentin momenttioshje integeriksi
87
88 #TorqRef := #Torq_Vakiomomentti; // viedään vakiomomentin mukainen momenttioshje väylään kirjoitettavaksi
89
90 #Kuormitusohjelma päällä := TRUE; // triggeri tamun controlwordille
91
92 ELSIF
93
94 //Neliöllinen momentti kuormitusohjelma
95 #Neliöllinen momentti = TRUE AND #Kolmiokontaktori THEN // kun neliöllinen momentti on valittuna ja kolmiokontakan vetää
96
97 #SpeedRatio := #ActualSpeed 0-100% / 100.0; // otetaan todellisen nopeuden ja nimellisen nopeuden suhdeluku
98
99 #SpeedSqr := SQR_REAL(#SpeedRatio); // korotetaan nopeuksien suhde toiseen potenssiin
100
101 #Neliöllinen momenttioshje_Temp := #SpeedSqr ^ #Torq_(Sp/TW)/100; // muodostetaan neliöllinen momenttioshje kertomalla nopeustekijällä momenttitekijää
102
103 #Torq_Neliöllinen := REAL_TO_INT(#Neliöllinen momenttioshje_Temp); // käännetään momenttioshje integeriksi
104
105 #Torq_NeliöllinenScaled := #Torq_Neliöllinen ^ 100; // skaalataan ohje 0-100% => 0-10000
106
107 #TorqRef := #Torq_NeliöllinenScaled; // viedään neliöllinen momenttioshje väylään kirjoitettavaksi
108
109 #Kuormitusohjelma päällä := TRUE; // triggeri controlwordille
110
111 ELSIF
112 // Vakioteho kuormitusohjelma
113 #Vakioteho = TRUE AND #Kolmiokontaktori = TRUE THEN // kun vakioteho on valittu ja kolmiokontakan vetää
114
115 #TorqRef := #Torq_Vakioteho; // laskettu vakiotehon momenttioshje viedään väylään kirjoitettavaksi
116
117 #Kuormitusohjelma päällä := TRUE; // triggeri controlwordille
118 END_IF;
119
120 //Kun kuormitusohjelma suoritetaan kirjoitetaan controlwordiin käynnistys sana, ellei ohjelmaa ole valittu CW:hen ajetaan pysäytys
121 □ IF #Kuormitusohjelma päällä = TRUE OR #RUN FORWARD = TRUE OR #RUN REV = TRUE THEN
122 #ControlWord := 3199;
123 #RUNNING := TRUE;
124 ELSIF #RUN FORWARD = FALSE OR #RUN REV = FALSE THEN
125 #ControlWord := 1150;
126 #RUNNING := FALSE;
127 END_IF;
128

```

Liite 6. Jännitteenjakelu 400V & 230V



Liite 8. Moottorin syöttö & virranmittaus



jamk

MOOTTORIN SYÖTÖ & VIRTAMITTARIT

Document type

PIIRIKAAVIO

Drawing number

AB6568

Date

15.4.25

Scale

1 - 3/20

RevNo

Revision note

1

2

3

4

Date

Checker

Approver

Creator

Checker

Approver

Date

Checker

Approver

Creator

Checker

Approver

Date

Checker

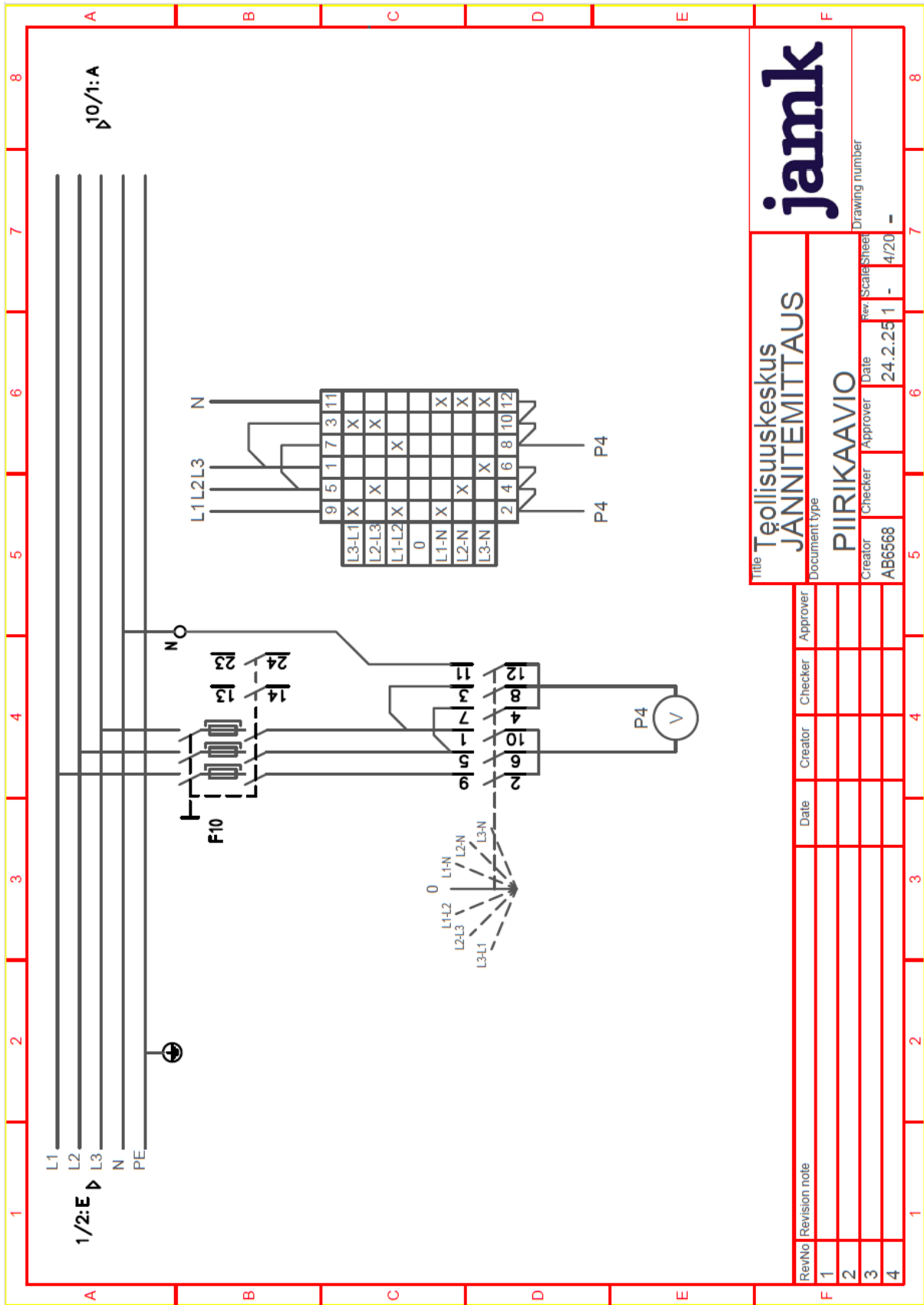
Approver

Creator

Checker

Approver

Liite 9. Jännitteenmittaus



jamk

Title: **Teollisuuskeskus JÄNNITEMITTAUS**

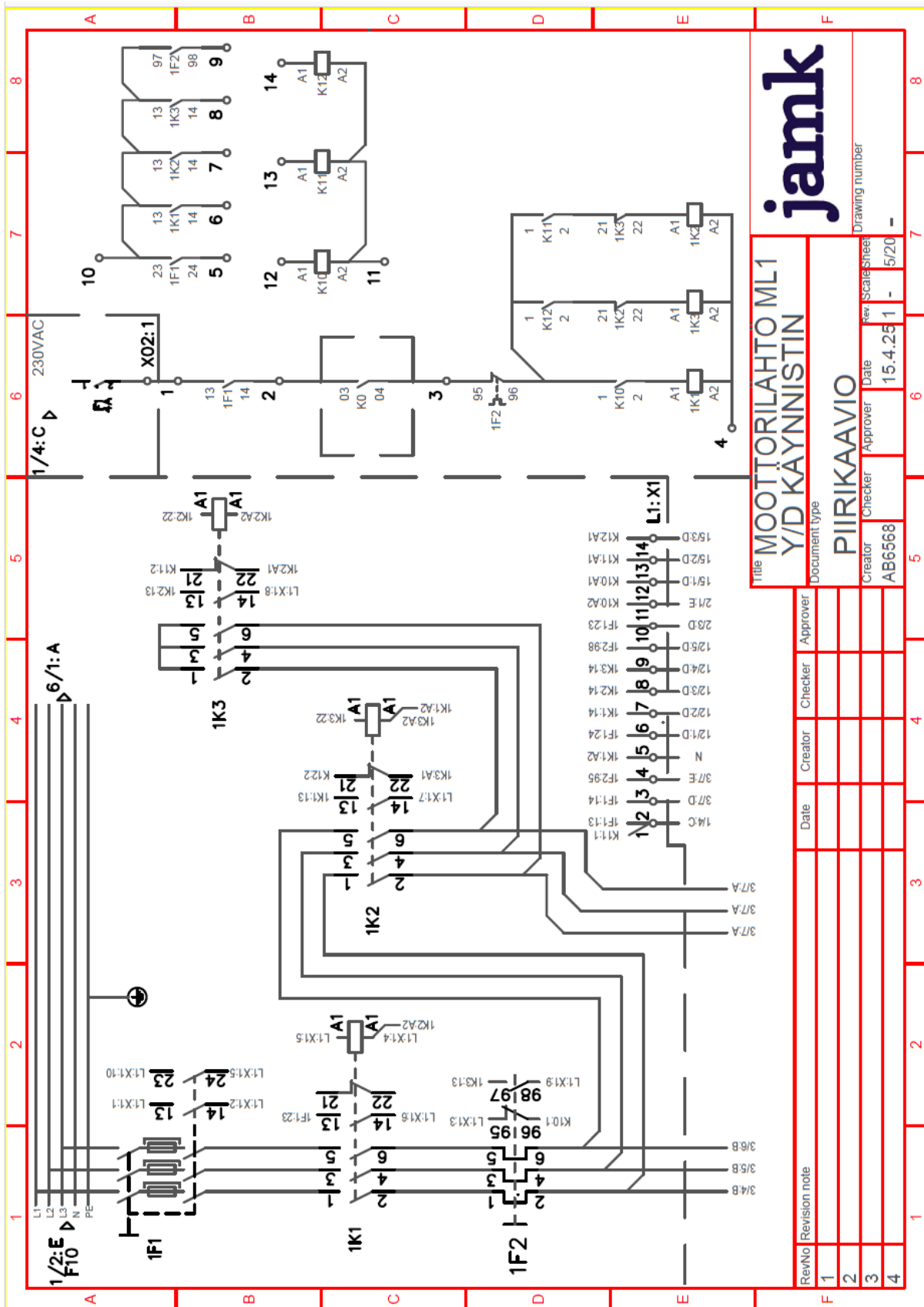
Document type: **PIIRIKAAVIO**

| | | | | | | |
|---------|---------|----------|---------|-----|-------|-------|
| Creator | Checker | Approver | Date | Rev | Scale | Sheet |
| AB6568 | | | 24.2.25 | 1 | - | 4/20 |

Drawing number: -

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

Liite 10. Tähtikolmiökäynnistin



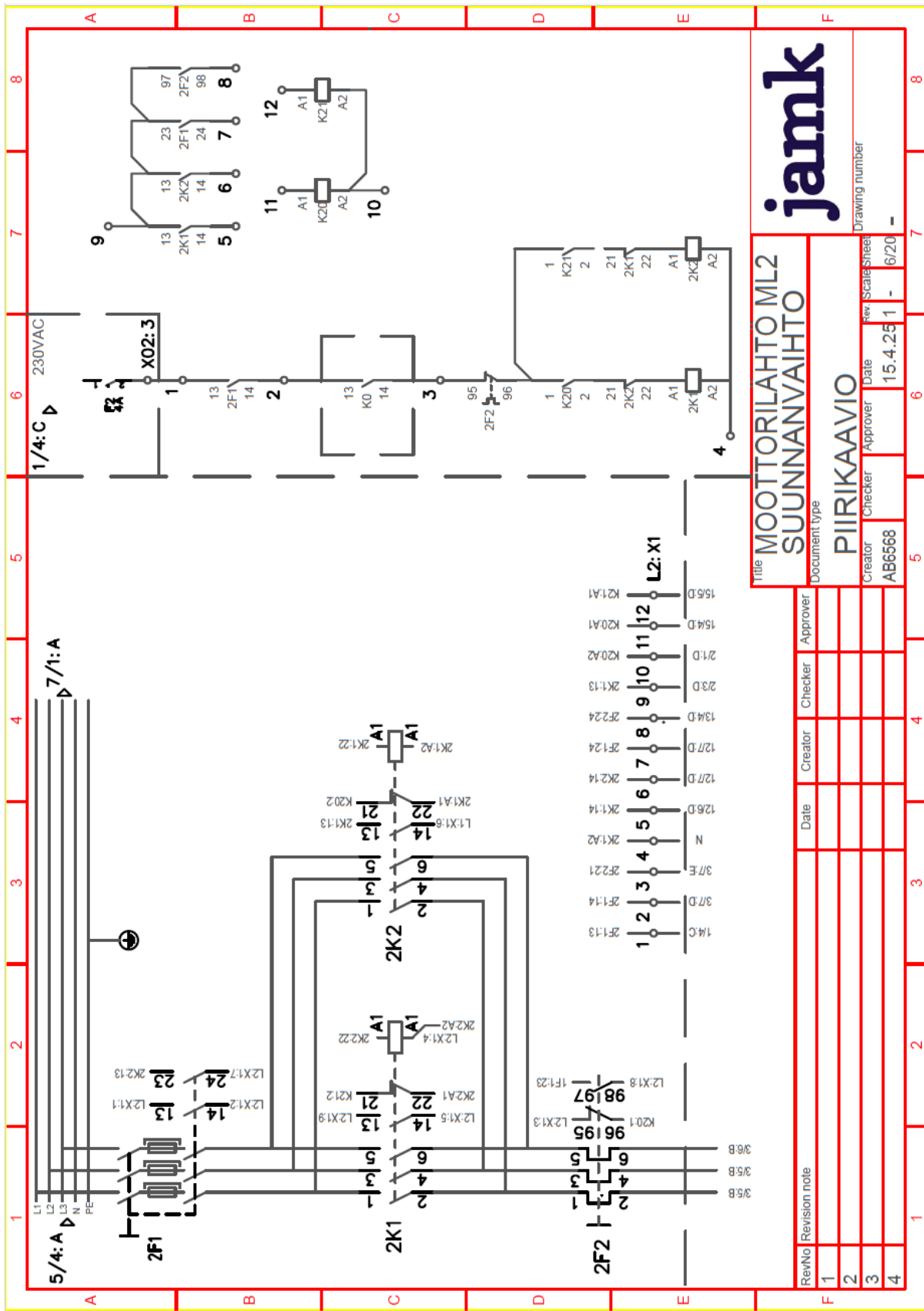
MOOTTORILAHTO ML1
Y/D KÄYNNISTIN

Document type: **PIIRIKAAVIO**

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

| | | | | | | |
|----------------|--------|----------|---------|---|------|---|
| Creator | AB6568 | Date | 15.4.25 | 1 | 5/20 | - |
| Checker | | Approver | | | | |
| Drawing number | | | | | | |

Liite 11. Suunnanvaihtolähtö



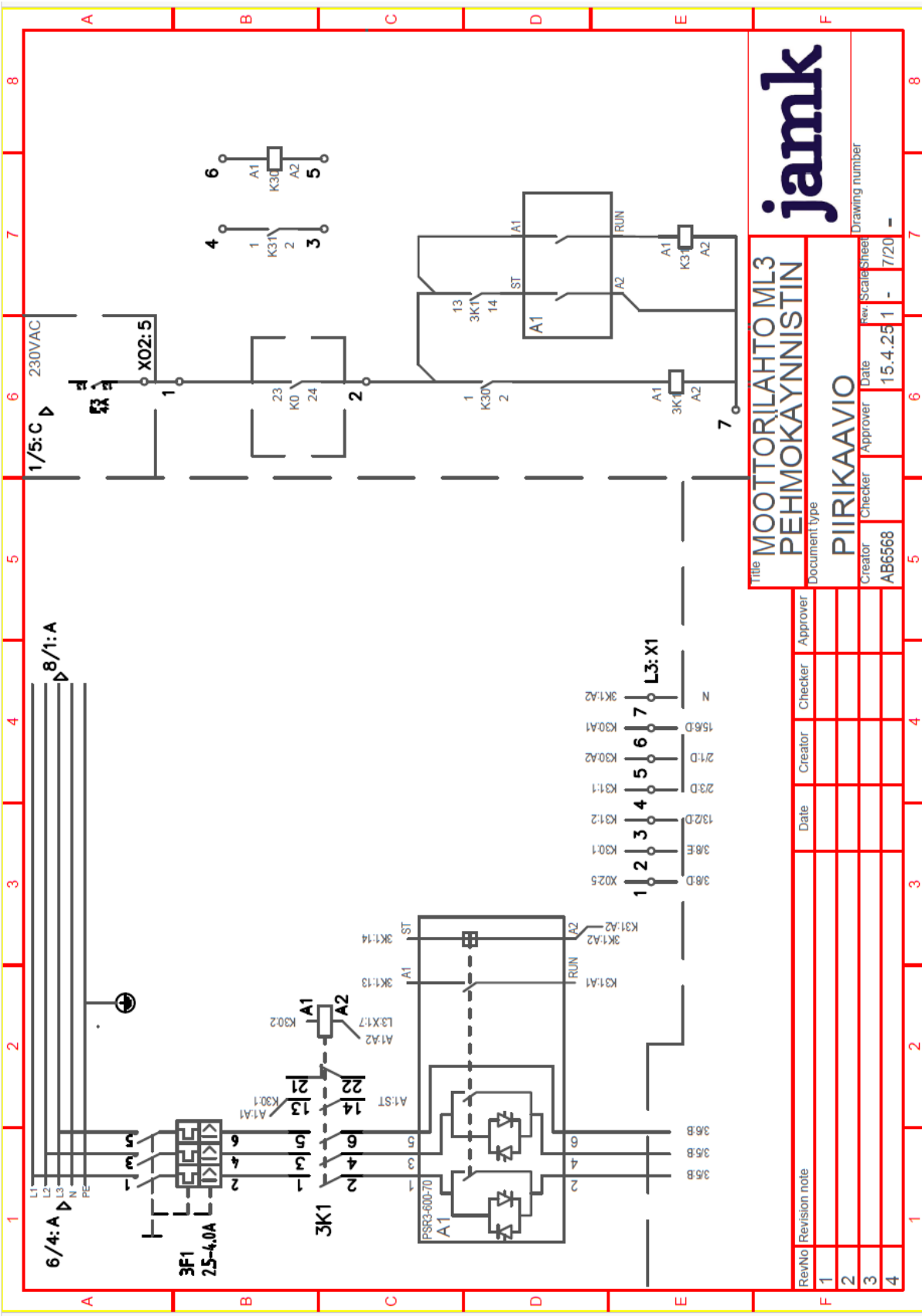
jamk

Title: MOOTTORILAHTO ML2
 SUUNNANVAIHTO
 Document type: PIIRIKAAVIO

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

| | | | | | | | | | | | |
|---------|--------|------|---------|-----|---|-------|---|-------|------|----------------|---|
| Creator | AB6568 | Date | 15.4.25 | rev | 1 | Scale | 1 | Sheet | 6/20 | Drawing number | - |
|---------|--------|------|---------|-----|---|-------|---|-------|------|----------------|---|

Liite 11. Pehmokäynnistin



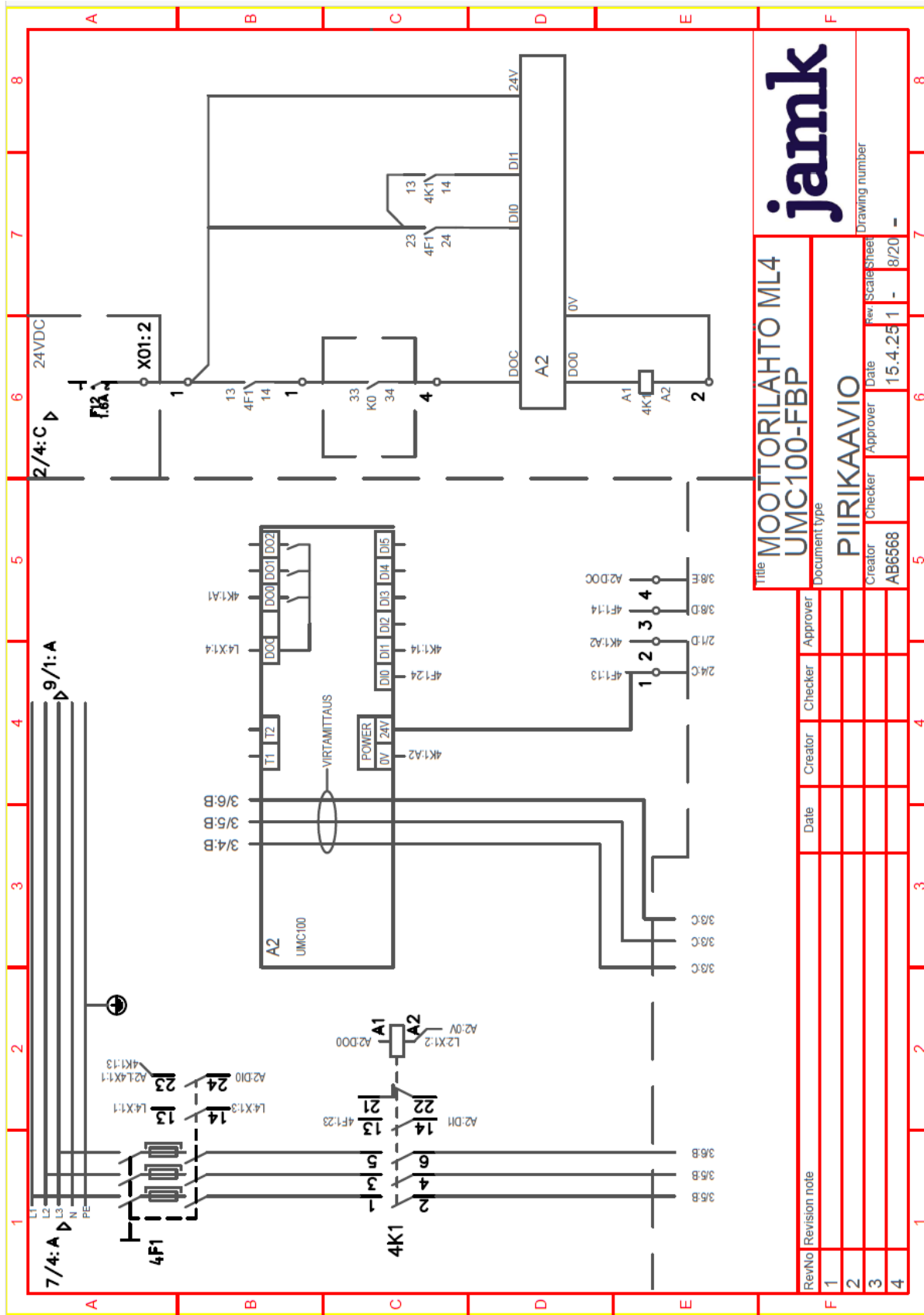
jamk

Title: MOOTTORILAHTO ML3
 PEHMOKÄYNNISTIN
 Document type: PIIRIKAAVIO

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

| Creator | Checker | Approver | Date | Rev | Scale | Sheet | Drawing number |
|---------|---------|----------|---------|-----|-------|-------|----------------|
| AB6568 | | | 15.4.25 | 1 | - | 7/20 | - |

Liite 12. Älykäs moottorinohjain



jamk

Title
MOOTTORILAHTO ML4
UMC100-FBP

Document type

PIIRIKAAVIO

Creator AB6568

Checker

Approver

Date 15.4.25

Rev. Scale/Sheet 1 - 8/20

Drawing number

-

Rev/No

Revision note

1

2

3

4

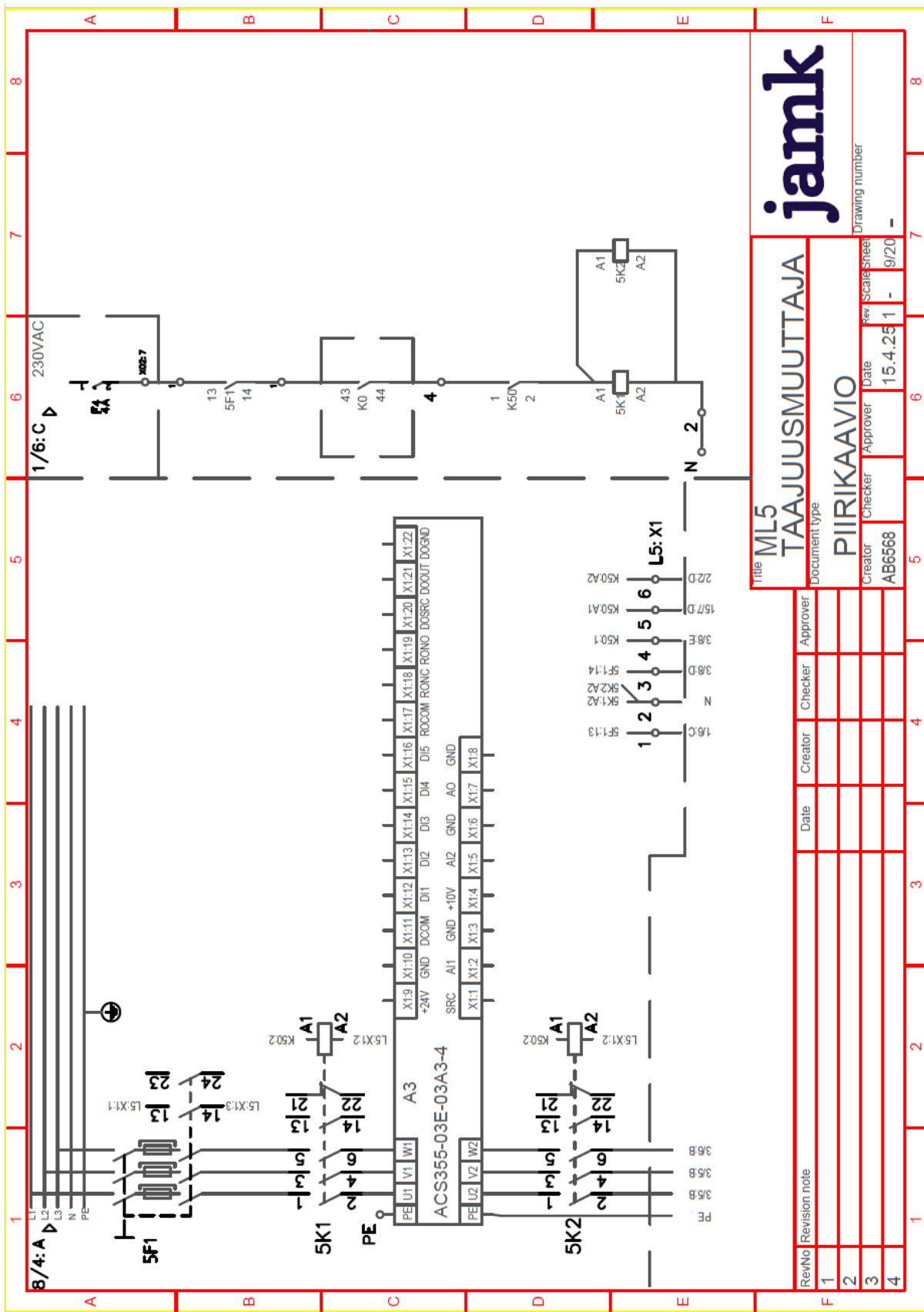
Date

Approver

Checker

Creator

Liite 13. Taajuusmuuttajälähtö

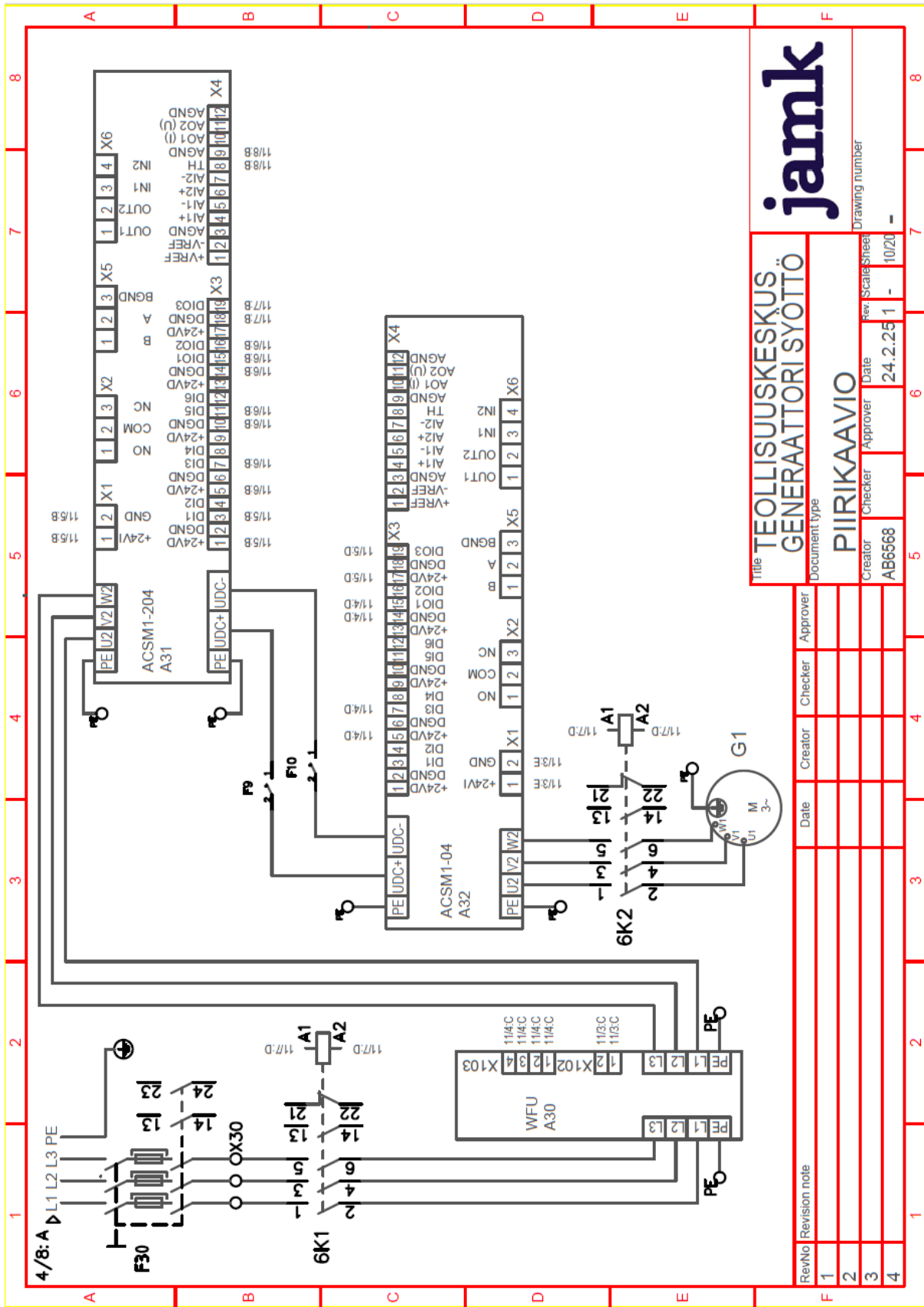


| | | | |
|----------------|----------|-----------------|------------------|
| Title | | ML5 | |
| Document type | | TAAJUUSMUUTTAJA | |
| PIIRIKAAVIO | | Creator | AB6568 |
| Checker | Approver | Date | 15.4.25 1 - 9/20 |
| Drawing number | | - | |

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |



Liite 14. Generaattorin pääpiiri



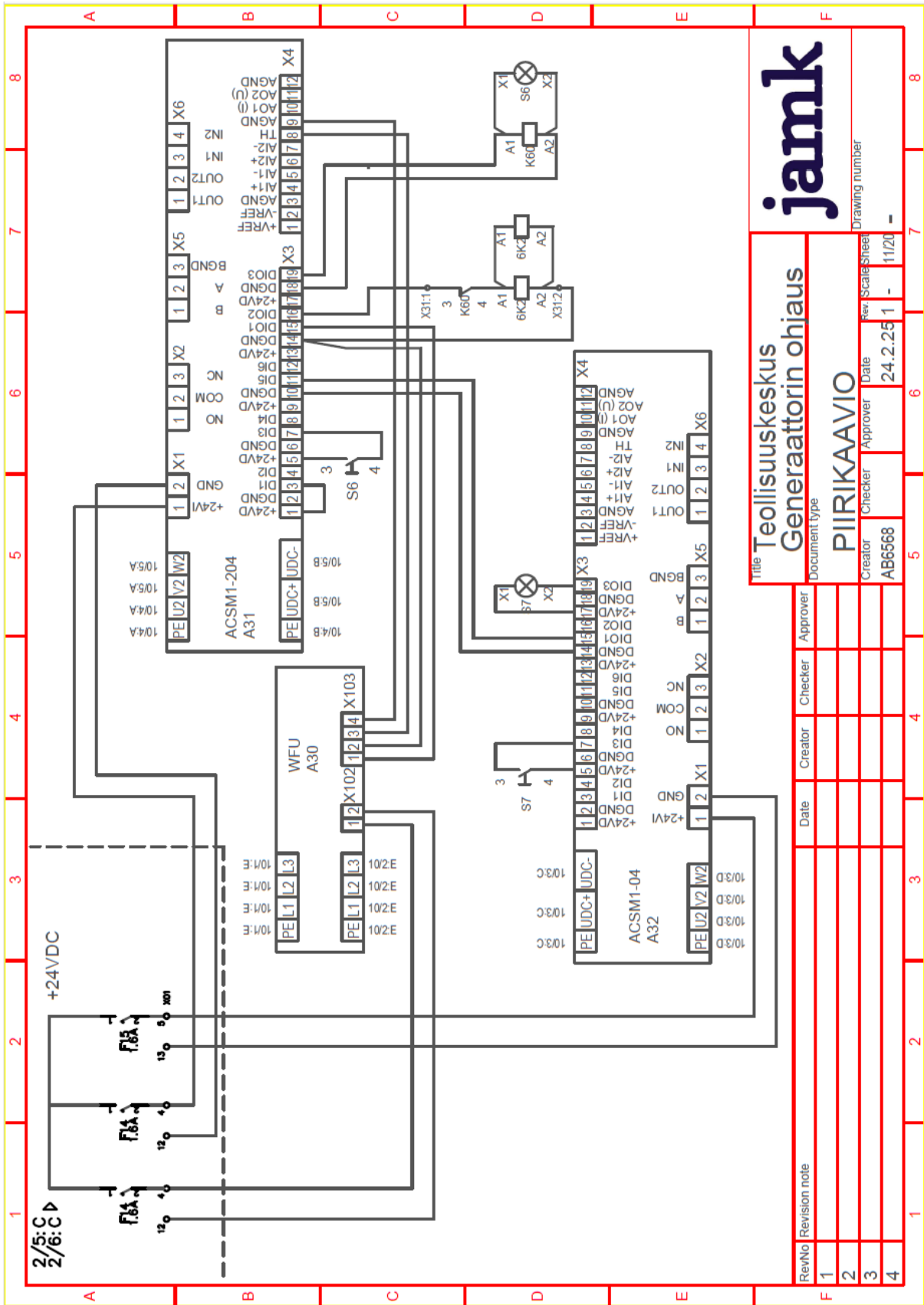
jamk

**TEOLLISUUSKESKUS
GENERAATTORISYÖTÖ**
Document type

PIIRIKAAVIO
Creator AB65568
Date 24.2.25
rev 1
Scale Sheet 10/20
Drawing number -

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

Liite 15. Generaattorin ohjauspiiri



jamk

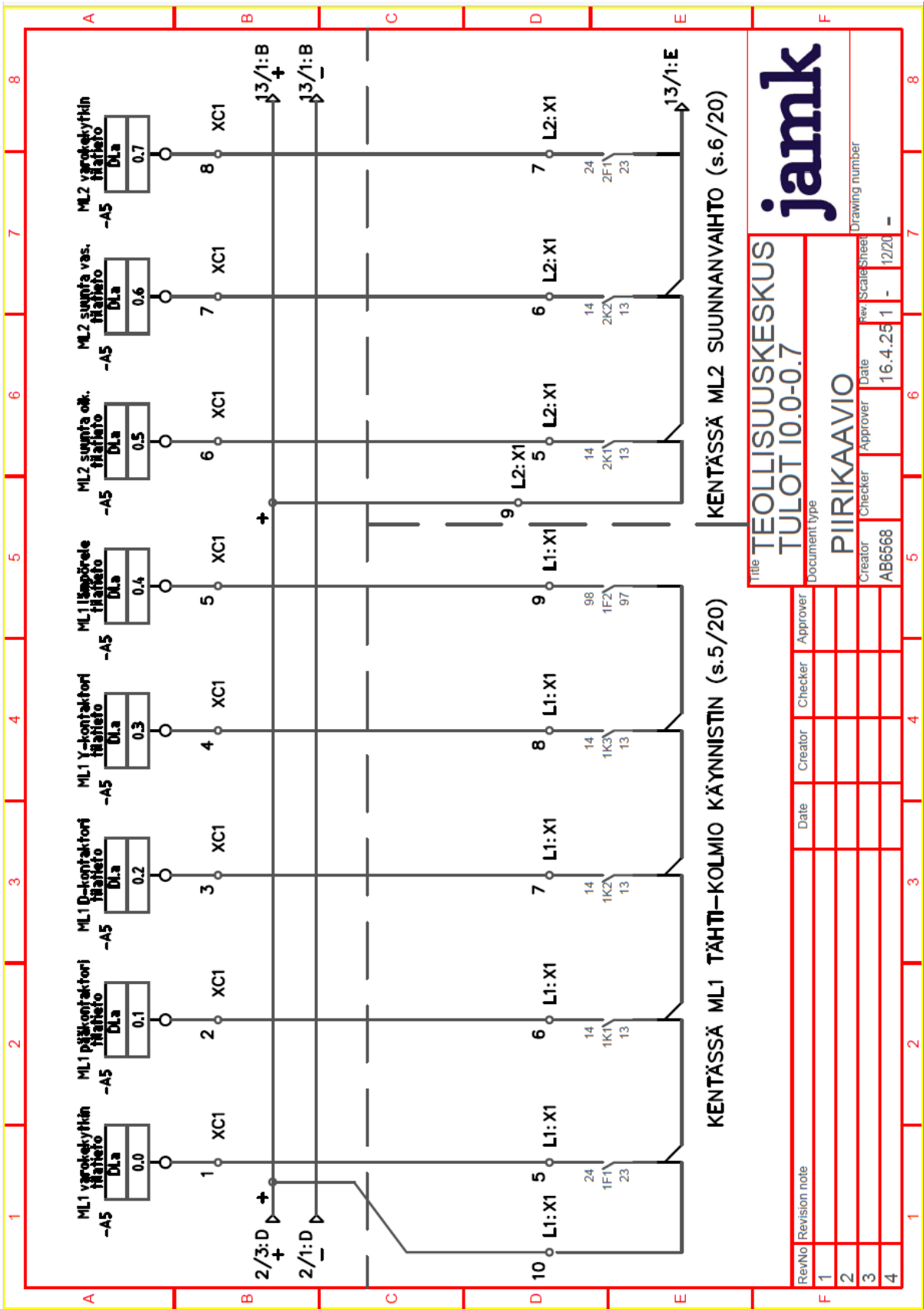
Title: Teollisuuskeskus
Generaattorin ohjaus

Document type: PIIRIKAAVIO

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

| Creator | Checker | Approver | Date | Rev. Scale | Sheet | Drawing number |
|---------|---------|----------|---------|------------|-------|----------------|
| AB6568 | | | 24.2.25 | 1 | 11/20 | - |

Liite 16. Tulopiirikaavio DI0.0–0.7



TEOLLISUUSKESKUS
TULOT 10.0-0.7
 Document type

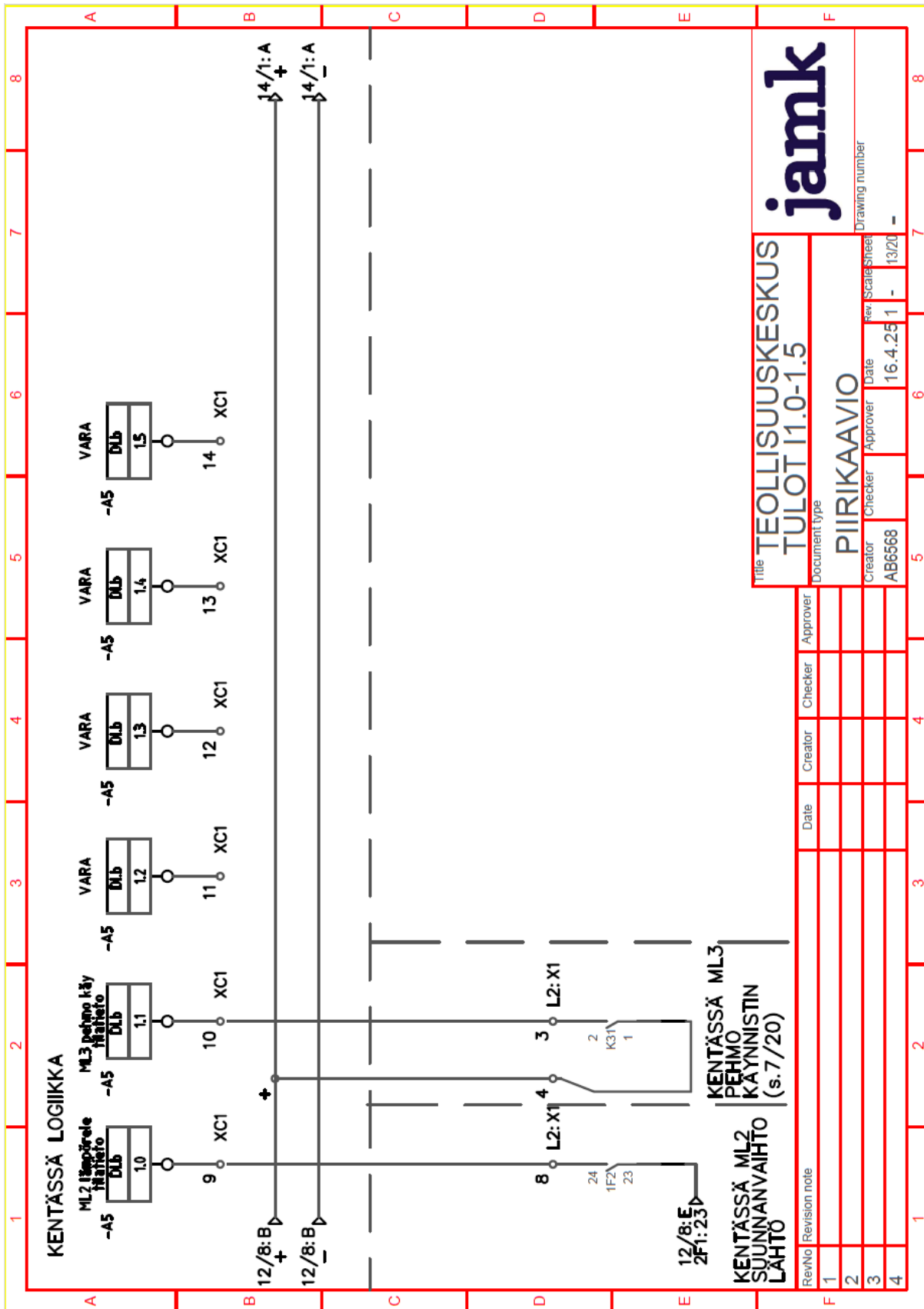
PIIRIKAAVIO
 Drawing number

Creator: AB6568
 Date: 16.4.2011
 Checker: -
 Approver: -
 Rev: 1
 Scale: -

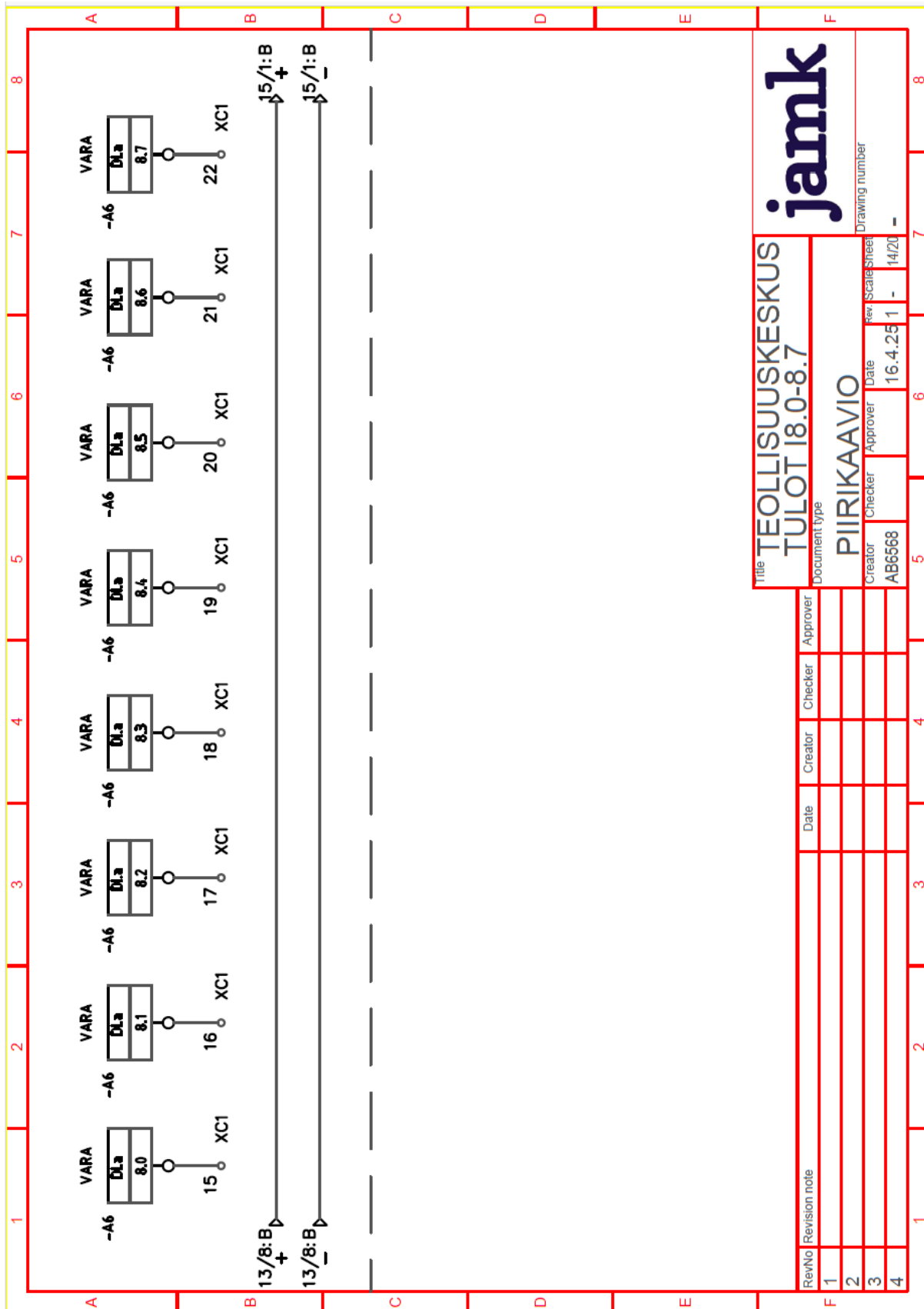
| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |



Liite 17. Tulopiirikaavio DI1.0–1.5

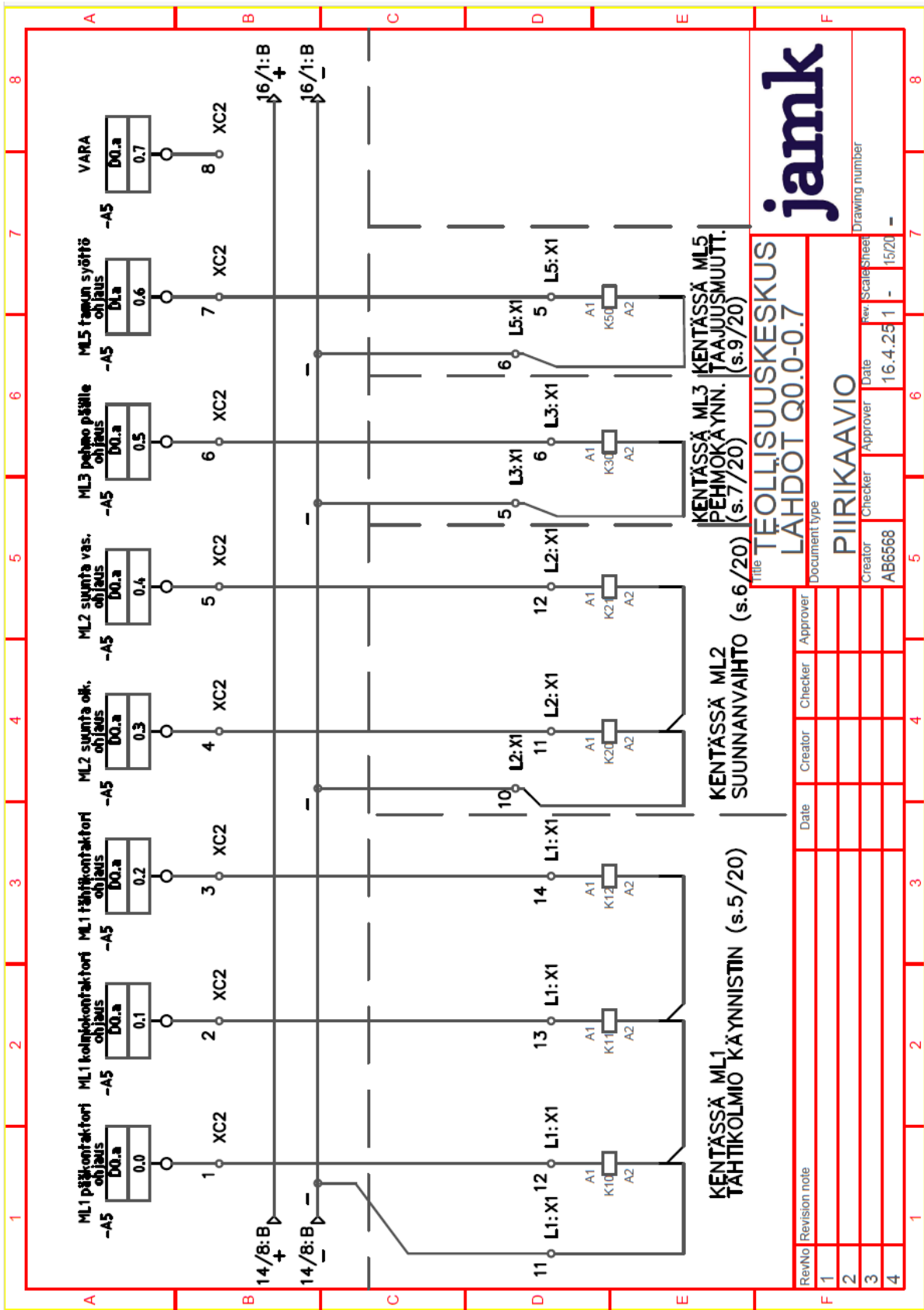


Liite 18. Tulopiirikaavio DI8.0–8.7



| | | | | | | | |
|--------|---------------|------------------------------------|---------|---------------|----------|----------------|---------|
| Title | | TEOLLISUUSKESKUS TULOT 18.0-8.7 | | Document type | | PIIRIKAAVIO | |
| Rev/No | Revision note | Date | Creator | Checker | Approver | Creator | Checker |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| | | Date | 16.4.25 | 1 | 14/20 | Drawing number | |
| | | Scale | | | | | |

Liite 19. Lähtöpiirikaavio DO0.0–0.7



File: TEOLLISUUSKESKUS LAHDOT Q0.0-0.7

Document type: PIIRIKAAVIO

| | | | | | | | | | | | | | | | |
|---------|--------|---------|--|----------|--|------|---------|-----|---|-------|--|-------|---|-------|-------|
| Creator | AB6568 | Checker | | Approver | | Date | 16.4.25 | Rev | 1 | Scale | | Sheet | 1 | Total | 15/20 |
|---------|--------|---------|--|----------|--|------|---------|-----|---|-------|--|-------|---|-------|-------|

Drawing number: -

| RevNo | Revision note | Date | Creator | Checker | Approver |
|-------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

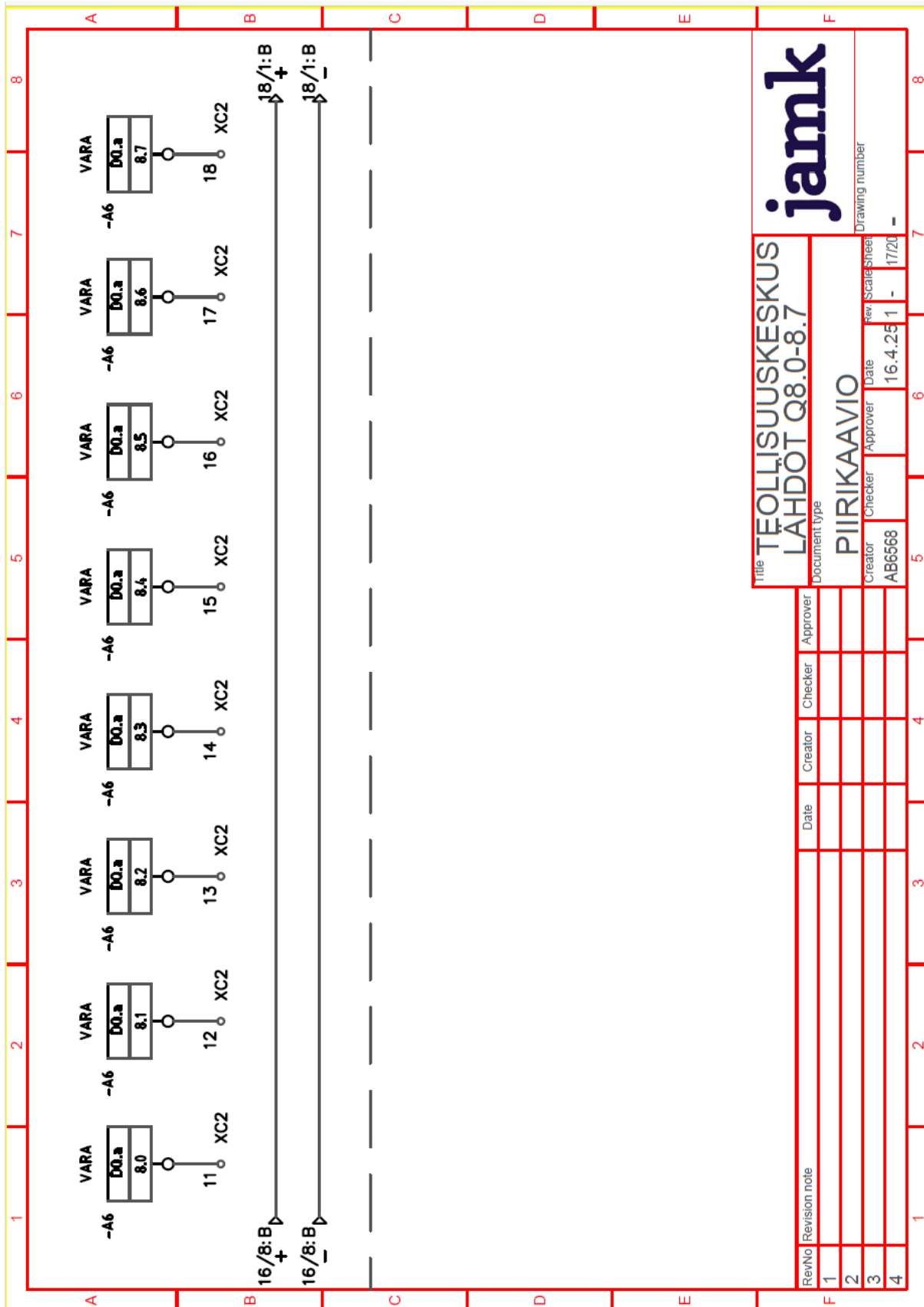
KENTÄSSÄ ML1 TAHTIKOLMIO KÄYNNISTIN (s.5/20)

KENTÄSSÄ ML2 SUUNNANVAIHTO (s.6/20)

KENTÄSSÄ ML3 PEHMOKÄYNN. TAAJUUSMUUTT. (s.7/20)

KENTÄSSÄ ML5 KENTÄSSÄ ML5 TAAJUUSMUUTT. (s.9/20)

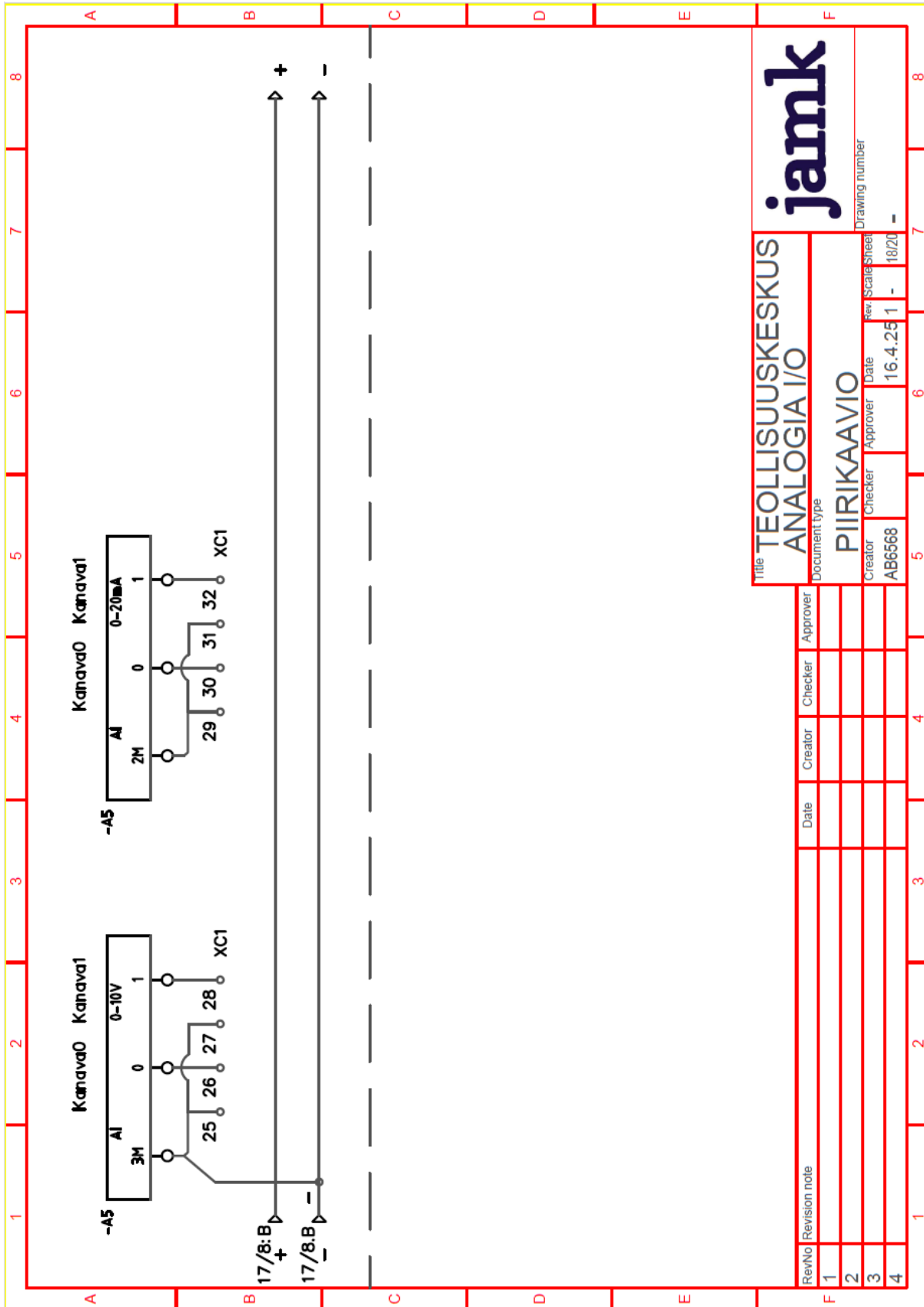
Liite 21. Lähtöpiirikaavio DO8.0–8.7



| | | | | | |
|---------------|---------|------------------|-----------|---------------------|-------|
| Title | | TEOLLISUUSKESKUS | | jamk | |
| Document type | | LÄHDÖT Q8.0-8.7 | | Drawing number | |
| PIIRIKAAVIO | | Date | | 16.4.25 1 - 17/20 - | |
| Creator | Checker | Approver | Date | rev | Scale |
| AB6568 | | | 16.4.25 1 | 1 | |

| Rev/No | Revision note | Date | Creator | Checker | Approver |
|--------|---------------|------|---------|---------|----------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

Liite 22. Piirikaavio analogiatulot



| | | | | | | | |
|-------------|---------------|------------------|---------|--------------|----------|---------------|-------------|
| Title | | TEOLLISUUSKESKUS | | ANALOGIA I/O | | Document type | |
| PIIRIKAAVIO | | Creator | AB6568 | Checker | Approver | Date | 16.4.25 |
| RevNo | Revision note | Date | Creator | Checker | Approver | rev | Scale/Sheet |
| 1 | | | | | | 1 | 18/20 |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| | | Drawing number | | | | - | |



Liite 23. Väyläkaavio

