

TAMPEREEN AMMATTIKORKEAKOULU
Kone- ja tuotantotekniikan koulutusohjelma
Kone- ja laiteautomaatio

Tutkintotyö

Anu Heinonen

**GX IEC DEVELOPER FX –OHJEKIRJA OHJELMOINTIIN
JA OHJELMAN TESTAAMISEEN**

Työn ohjaaja
Työn teettäjä
Tampere 2006

Tekniikan lisensiaatti Olavi Kopponen
TAMK, valvojana laboratorioinsinööri Seppo Mäkelä

TAMPEREEN AMMATTIKORKEAKOULU

Kone- ja tuotantotekniikan koulutusohjelma

Kone- ja laiteautomaatio

Heinonen, Anu

GX IEC Developer FX -Ohjekirja ohjelmointiin ja ohjelman testaamiseen

Tutkintotyö

62 sivua + 23 liitesivua

Työn ohjaaja

Tekniikan lisensiaatti Olavi Kopponen

Työn teettäjä

Tampereen ammattikorkeakoulu,
valvojana laboratorioinsinööri Seppo Mäkelä

Helmikuu 2006

Hakusanat

logiikka, ohjekirja, ohjelmointi, Mitsubishi, FX2N, E-Designer, Beijer E300

TIIVISTELMÄ

GX IEC Developer FX on automaatioalalla käytössä oleva tietokoneohjelma. Sen avulla suunnitellaan ja luodaan ohjelmia, jotka ladataan käyttöä varten logiikalle. Logiikka on ohjelmamuistina toimiva tietokone, jonka toiminta perustuu siihen ladattavaan ohjelmaan. Se ohjaa ladatun ohjelman avulla erilaisia toimilaitteita, esimerkiksi venttiileitä, moottoreita ja sylintereitä.

Tämä tutkintotyö tehtiin Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan koulutusohjelman kone- ja laiteautomaation suuntautumisvaihtoehdon opetuskäyttöä varten. Aiheena oli laatia GX IEC Developer FX -ohjelmalle ohjekirja, joka antaa tietoa itse ohjelmasta ja sen käytöstä. Laatimisperiaatteena oli aikaansaada ohjeistus, josta lukijan on helppo löytää etsimänsä asiat. Ohjekirjasta löytyy perustietoa ohjelmasta, sen käyttöliittymästä sekä käytettävistä osoitteista ja operaatioista. Se on myös ohjeistus ohjelmoimisesta, ohjelman testaamisesta sekä ohjelman lataamisesta Mitsubishin FX2N-logiikalle. Ohjekirjasta löytyy lisäksi tietoa GX IEC Developer FX -ohjelman rinnalla käytettävästä E-Designer-ohjelmasta, jonka avulla voidaan ohjelmoida Tampereen ammattikorkeakoulussa opetuskäytössä olevaa graafista Beijer E300 -operointipäätettä.

TAMPERE POLYTECHNIC

Mechanical and Production Engineering

Machine Automation

Heinonen, Anu

GX IEC Developer FX -An Instruction Book for programming and project testing

Engineering Thesis

62 pages + 23 appendices

Thesis Supervisor

Olavi Kopponen Lic. Tech.

Commissioning Company

Tampere Polytechnic,

Supervisor Laboratory Engineer Seppo Mäkelä

February 2006

Keywords

logic, manual, programming, Mitsubishi, FX2N, E-Designer, Beijer E300

ABSTRACT

GX IEC Developer FX is a computer program for automation applications. It is used for creating and designing special programs, which can be downloaded to the logic. The logic itself is a computer, the function of which is based on this downloaded program. Utilizing the downloaded program, the logic controls various instruments and components, such as valves, motors and cylinders for example.

This engineering research report was made for the educational use of Tampere Polytechnic. The department of Machine Automation has begun to use an automation program, GX IEC Developer, as a part of the education. The purpose of this report was to devise a manual that informs the reader how to use the program. The manual was designed to give basic information about the program, for example its user interface, addresses and the functions it contains. It also includes instructions for programming, project testing as well as downloading the project into Mitsubishi FX2N-logic. There is also a chapter for E-Designer, through which the Beijer E300-graphical display can be programmed.

ALKUSANAT

Tahdon kiittää Tampereen ammattikorkeakoulun kone- ja laiteautomaation suuntautumisvaihtoehtoa haastavasta ja mielenkiintoisesta tutkintotyöaiheesta. Ohjekirjan kirjoittaminen opetuskäyttöä varten on ollut antoisaa ja opettavaista. Tieto tutkintotyöni tarpeellisuudesta on osaltaan innostanut minua työnteon aikana. Erityiskiitoksen haluan antaa työni valvojalle, laboratorioinsinööri Seppo Mäkelälle, joka yhteistyöllään on avustanut minua koko tutkintotyöprosessin ajan ja antanut käyttööni tarvittavan materiaalin.

Tampereella 23.2.2006

Anu Heinonen

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

ALKUSANAT

SISÄLLYSLUETTELO	5
LYHENTEET	7
1 JOHDANTO	8
2 OHJEKIRJAN KIRJOITTAMINEN	9
3 IEC 61131-3 -STANDARDI	10
4 OHJEKIRJA: ”FX2N-LOGIIKAN OHJELMOINTI GX IEC DEVELOPER FX -OHJELMALLA” ..	11
4.1 LOGIIKKA	11
4.2 MUISTIOHJELMOITAVA OHJAUS	11
4.3 SYKLINEN OHJELMANKÄSITTELY.....	12
4.4 OPETUSLAITESALKKU - YLEISKUVA.....	13
4.5 GX IEC DEVELOPER FX –OHJELMA	15
4.5.1 Yleiskuva	15
4.5.2 Ohjelman käynnistys	16
4.5.3 Uuden projektin luominen.....	17
4.5.4 Ohjelman alustaminen	18
4.5.5 Yleismuuttujien listaaminen	20
4.5.6 Ohjelman luominen.....	21
4.5.7 Aputoiminta.....	24
4.6 OSOITTEET.....	25
4.6.1 Bitti, tavu, puolitavu, sana ja kaksoissana	25
4.6.2 I/O-osoitteet	26
4.6.3 Erikoismuistipaikat	27
4.6.4 Virhekoodit.....	27
4.6.5 Nopeat laskurit.....	27
4.7 OPERAATIOIT	29
4.7.1 Peruskäskyt	29
4.7.2 Vertailijat	31
4.7.3 Ajastimet	32
4.7.4 Laskurit	33
4.7.5 Aritmetiikkäkäskyt.....	36
4.7.6 Master Control / Master Control Reset	37
4.7.7 Tavutiedon siirto (MOV) I/O-avaruuteen	37
4.8 POU	38
4.9 TASK	38
4.10 TOIMILOHKON LUOMINEN	39
4.11 LUVUT	42
4.11.1 BCD-luvut	42
4.11.2 Datatyypit.....	43
4.12 SIMULOINTI	44
4.12.1 Projektin kääntäminen.....	44
4.12.2 Offline-simulaattori.....	46
4.13 OHJELMAN LATAAMINEN LOGIIKALLE	47
4.14 OHJELMAN MONITOROINTI	48
4.15 E-DESIGNER	48
14.15.1 E-Designer -ohjelma.....	48
4.15.2 Ohjelman käynnistys ja esiasetukset	49
4.15.3 Sivujen luominen.....	50

4.15.4 Digitaalisymbolit.....	52
4.15.5 Ohjelman lataaminen operointipäätteelle.....	56
5 TULOKSET	58
6 TULOSTEN TARKASTELU	59
LÄHTEET.....	61
LIITTEET	
1 Erikoisapumuistit	
2 Virhekoodit	
3 E-Designer-ohjelman graafiset symbolit	
4 Harjoitustehtävät	

LYHENTEET

b	Bit, suom. bitti
B	Byte, suom. tavu
BCD	Binary Coded Decimal, suom. binäärikoodattu desimaalilukuesitys
CPU	Central Processing Unit, suom. keskusyksikkö
D	Dataregister, suom. datarekisteri
FB	Function Block, suom. toimilohko
FBD	Function Block Diagram, suom. toimilohkokaavio
FUN	Function, suom. funktio
INT	Integer, suom. kokonaisluku
I/O	Input/Output, suom. tulot /lähdöt
IL	Instruction List, suom. käskylista
LD	Ladder Diagram, suom. relekaavio
LTT	Ladder Logic Test Tool
M	Memory, suom. muisti
MC	Master Control, suom. pääkoskettimen asetus
MCR	Master Control Reset, suom. pääkoskettimen nollaus
PLC	Programmable Logic Controllers, suom. ohjelmoitavat logiikat
POU	Program Organization Unit, suom. rakenneyksikkö
R	Reset
REAL	Reaaliluku
SFC	Sequential Function Chart, suom. sekvenssikaavio
S	Set
W	Word, suom. sana

1 JOHDANTO

GX IEC Developer FX -ohjelma on automaatioalalla käytettävä ohjelma, jonka avulla voidaan luoda graafisia projekteja ja ladata ne logiikkaan. Ohjelma on laajalti suunnittelukäytössä eri automaatioyrityksissä sekä opetuskäytössä tekniikan alalla. Myös Tampereen ammattikorkeakoulun kone- ja laiteautomaation suuntautumisvaihtoehto on ottanut ohjelman opetuskäyttöön aiemmin käytetyn automaatio-ohjelman rinnalle.

Kuten muut kehittyneet automaatio-ohjelmat, GX IEC Developer FX on monipuolinen ohjelma ja vaatii käyttäjältään perehtymistä ohjelmaan. Tampereen ammattikorkeakoulussa oppilailla on käytössään ohjelmamanuaaleja ja paljon muuta tietoa ohjelmasta. Ongelmana on se, että tietoa on paljon ja se on hajallaan eri lähteissä. Jotta ohjelman oppiminen ja opettaminen koulussa helpottuisi, tietoa oli yhdistettävä.

Tämän tutkintotyön tavoitteena on laatia ohjelman perusasiat sisältävä ohjekirja sekä luoda tiivis kokonaisuus sen sisältämistä toiminnoista. Opetuksen kannalta tärkeä tieto oli eroteltava muusta tietomäärästä ja koottava yhteen. Ohjekirjan aiheenrajaus ja jäsentely on muodostettu työn teettäjän kanssa yhteistyössä. Tavoitteena oli tehdä ohjekirjasta mahdollisimman kattava ja johdonmukainen. Toistoja työssä on pyritty välttämään.

2 OHJEKIRJAN KIRJOITTAMINEN

Opetuskäyttöön tarkoitettussa materiaalissa yhdistyvät sekä tutkimus että opetus. On tärkeää, että kirjoittajalla on sekä tietoa aiheestaan että taitoa käyttää kieltä. Kirjoittajan on myös hyvä varautua siihen, että lukijakuntana saattaa olla sellaisia henkilöitä, joille kirjoittaja ei ole teostaan alun perin tarkoittanut. /1/

Opetusmateriaali pyritään tekemään paljolti lukijan tarpeet huomioonottaen. Selkeys, tekstin tiiviys, selkeä kappalejako sekä sujuva eteneminen aiheesta toiseen ovat ominaisuuksia, jotka tekevät tieteellisestä tekstistä helposti luettavan. Lukijan kannalta on myös tärkeää, että kirjoittaja tuntee työnsä tarkoituksen. Kun kirjoittaja tietää, mitä teokselta odotetaan, aiheen rajaus sujuu onnistuneesti. /1/

3 IEC 61131-3 -STANDARDI

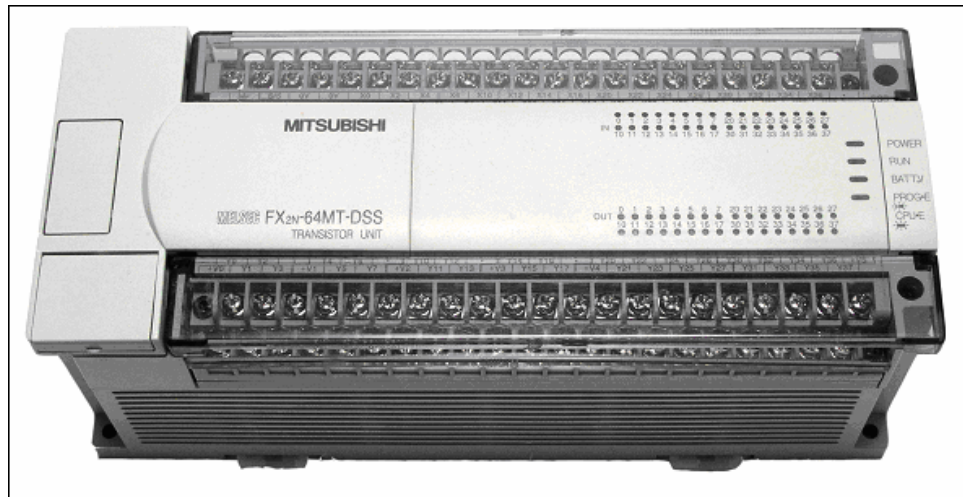
Suunniteltava oppimateriaali perustuu kansainvälisesti hyväksytyyn IEC 1131-3 -standardiin. Tämä logiikoiden väliseen tiedonsiirtoon perustuva standardi kehitettiin 1990-luvulla yhtenäistämään logiikkaohjelmoinnin kuvaustapaa, parantamaan ohjelmoinnin laatua sekä nopeuttamaan ohjelmointiprosessia. Jokaisen logiikkavalmistajan on tarkoitus ottaa kyseinen standardi käyttöön lähitulevaisuudessa, jolloin käytettävät ohjelmamallit ja datatyypit mukautuvat toisiinsa yhä enemmän. /7;12;13/

IEC 61131-3 -standardi on muistiohjelmoitavan ohjauksen (Programmable Logic Controllers, PLC) uusi standardi, joka koostuu PLC-ohjelmissa käytettävistä rakennesista sekä viidestä ohjelmointikielestä. Nämä ohjelmointikielet ovat keskenään yhteensopivia, joten niitä voidaan käyttää samassa ohjelmassa. Kie-listä käskylista (Instruction List, IL) ja rakenteinen teksti (Structured Text, ST) ovat tekstimuotoisia. Graafiselta muodoltaan ovat tikapuukaavio (Ladder Diagram, LD), funktiolohkokaavio (Function Block Diagram, FBD) sekä sekvenssiohjelmointi (Sequential Function Chart, SFC). SFC-kieli on kehitetty ylemmän tason ohjelmointia varten, eikä se ole varsinainen kieli, vaan joukko graafisia kaavioita, joiden avulla logiikkaohjelmaa voidaan kuvata. /2;7;12;13/

4 OHJEKIRJA: ”FX2N-LOGIIKAN OHJELMOINTI GX IEC Developer FX -ohjelmalla”

4.1 Logiikka

Logiikkaa (kuva 1) verrataan usein tietokoneeseen, jonka avulla voidaan ohjata automaatioprosessin toimilaitteita. Logiikan toiminta perustuu sen muistiin laadittavaan ohjelmaan, joka voidaan laatia tietokoneen tai ohjelmointilaitteen avulla. Tämä ohjelma määrää logiikan tulojen avulla lähtöjen tilat, jotka vuorostaan ovat yhteydessä käytettäviin toimilaitteisiin. Toisin sanoen logiikka saa tuloporttiansa avulla tietoa automaatioprosessista ja ohjaa automaatiojärjestelmää lähtöjensä avulla. Logiikan tuloporttiin voidaan kytkeä esimerkiksi painike, jota painettaessa logiikan lähtöporttiin kytketty moottori käynnistyy. /25/



Kuva 1 FX2N-logiikka

4.2 Muistiohjelmoitava ohjaus

Ohjelmoitavat logiikat eli muistiohjelmoitava ohjaus (Programmable Logic Controllers, PLC) on kehitetty tehostamaan ja helpottamaan ohjaustehtävien suorittamista. Sen avulla voidaan toteuttaa monimutkaisempia ohjaustehtäviä kuin aiemmin käytetyn johdotusohjelmointiohjauksen avulla. Esimerkiksi virheiden korjaaminen ja etsiminen toteutuvat sujuvasti PLC-ohjauksen avulla, sillä sen toiminta perustuu laadittavaan ohjelmaan. Virheen löydyttyä korjaus suoritetaan ohjelmaan, eikä johdotusten purkamista tai uudelleenjohdotusta tarvitse tehdä, toisin kuin johdotusohjelmointiohjauksen vianetsinnässä. PLC-ohjauksen oh-

jelma kytketään tulojen ja lähtöjen avulla järjestelmään, minkä jälkeen se tallennetaan ohjelmoitavan logiikan sisäiseen muistiin. Tämän jälkeen keskusyksikkö pystyy lukemaan tulosignaaleja, suorittamaan käskyt sekä asettamaan lähtöjen tilat, jotka siirretään kaapelia pitkin käytettävälle toimilaitteelle. Tällaisia ohjattavia toimilaitteita voivat olla esimerkiksi kontaktorit, venttiilit sekä mekaaniset kytkimet. /3;6/

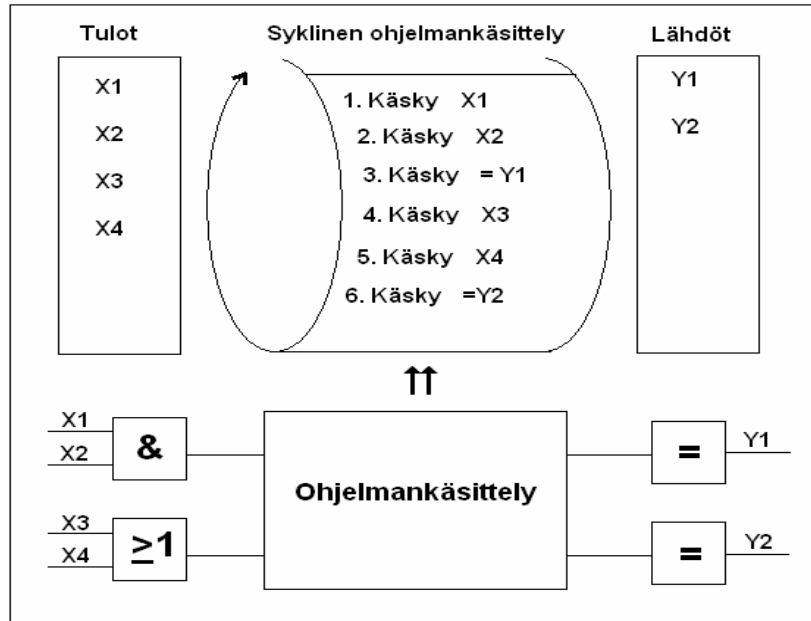
Muistiohjelmoitavaan ohjaukseen tarvittavia komponentteja ovat:

1. automaatiojärjestelmä
2. ohjelmointilaite
3. yhdistyskaapeli. /6/

Mitsubishi Electric FX -sarjasta opetuskäytössä on kompakti automaatiojärjestelmä, jossa laitteiston pääyksikkö koostuu perusyksiköstä sekä kompaktista laajennusyksiköstä. Perusyksikkö sisältää keskusyksikön (CPU) ja ohjelma-muistin. Sekä perusyksikkö että kompakti laajennusyksikkö sisältävät yhdysrakenteisen verkkolaitteen (24 VDC) sekä tulojen ja lähtöjen riviliitinrintamat. Kummassakin on käytettävissä joko rele- tai transistorilähtötyyppejä. /3;6/

4.3 Syklinen ohjelmankäsittely

Kun perinteinen johdotusohjelmointiohjaus suorittaa ohjaustoiminnot rinnakkain, suorittaa muistiohjelmoitava ohjaus ohjelman peräkkäin ja toistaa sen syklisesti (kuva 2). Syklisessä ohjelmankäsittelyssä käydään kokonainen ohjelma- kierros kerralla läpi, minkä jälkeen tulokset siirretään lähtöihin Y. Ohjelman kokonaiskäsittely jakautuu siis kolmeen vaiheeseen: tulojen käsittelyyn, ohjelman käsittelyyn sekä lähtöjen käsittelyyn. Tulojen käsittelyssä tulosignaaleja käytetään tulotietoina ja ne tallentuvat tulojen kuvarekisteriin. Ohjelmankäsittely-vaiheessa ohjelmoitava logiikka lukee kyseiset tilat tulojen kuva-rekisteristä. Nämä tilatiedot tallentuvat järjestyksessä operandien kuva-rekisteriin. Lähtöjen käsittelyvaiheessa kuvarekisteritieto siirtyy lähtömuistiin, jonka kautta käytettävät toimilaitteet saavat toimintatietonsa. Syklisessä ohjelmankäsittelyssä suoritetaan siis edellä mainitut kokonaiskäsittelyvaiheet peräkkäin ja toistuvasti. /3/



Kuva 2 Syklinen ohjelmankäsittely

4.4 Opetuslaitesalkku - yleiskuva

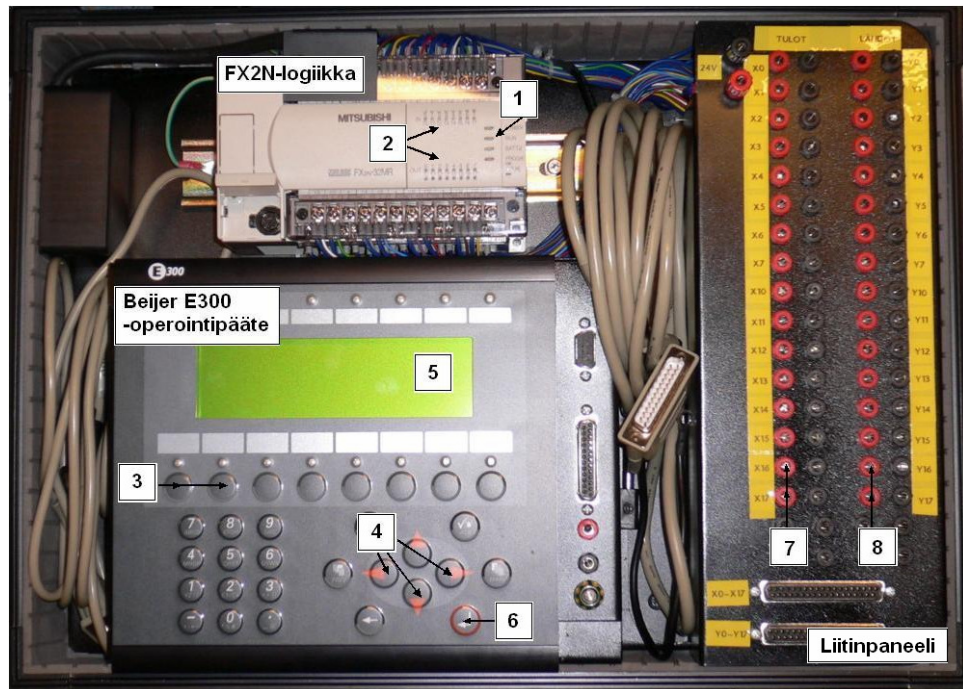
Opetuslaitesalkku on kokoonpano opetukseen käytettävistä laitteista. Se on koottu helpottamaan opetuslaitteiden siirrettävyyttä sekä I/O-liittimien kytke- mistä. Opetuspaketin kytkeminen toimilaitteisiin voi tapahtua joko banaaniliit- timillä tai 37-napaisilla D-liittimillä. /8/

Opetuslaitesalkun sisältö on seuraava:

- 1) FX2N-logiikka
- 2) Beijer E300 -operointipäätte
- 3) liitinpaneeli. /8/

Opetuslaitesalkussa oleva logiikka (kuva 3) kuuluu Mitsubishin FX2N-sarjaan. Logiikan tyyppimerkinnässä oleva luku, kuten FX2N 64-MR, ilmaisee logii- kassa olevien I/O-liittimien lukumäärän. Perusyksiköissä näitä liittimiä voi olla 16-256 kappaletta. Luvun perässä oleva kirjain M tarkoittaa perusyksikköä. Tämä tunnus kompaktille laajennusyksikölle on vastaavasti E. R-kirjain tarkoi- ttaa relelähtötyyppiä ja T-kirjain tarkoittaa transistorilähtötyyppiä. Perusyksikkö ja kompakti lisälaite on kiinnitetty toisiinsa DIN-kiskokiinnikkeen ja nauhaka-

peliliittimen avulla. DIN-kiskossa olevien reikien avulla moduulien kiinnittäminen käy suoraan ja logiikka on näin ollen helposti muunneltavissa. Perusyksikössä olevat merkkivalot (1) ilmaisevat POWER-, RUN-, BATT.V- ja CPU-E-toimintoja. Tulo- ja lähtövalot (2) syttyvät vain tulojen ja lähtöjen ollessa päällekytkettyinä. FX2N-logiikka on verkkovirralla toimiva sekä hyvin nopea ja helppokäyttöinen logiikka. /3;8;16;18/



Kuva 3 Opetuslaitesalkku ja sen osat

Irrotettavissa oleva Beijer E300 -operointipääte on asennettu salkkuun helpottamaan logiikan käyttöä. Operointipäätettä ohjataan E-Designer-ohjelman avulla, joten se kytketään logiikan ja verkkovirran lisäksi tietokoneeseen. Operointipääte sisältää funktiopainikkeet (3), nuolipainikkeet (4) sekä taustavalaistun LCD-näytön (5). Toimintojen hyväksyminen tapahtuu enter-painikkeella (6). /8/

Liitinpaneelissa on liittimet sekä I/O-liittimiä että liittimiä erikoismoduuleja varten. I/O-liittimiä ovat tulot ja lähdöt. Liitinpaneelin vasemmanpuoleiset punaiset liittimet ovat tuloja (7). Vastaavasti oikeanpuoleiset punaiset liittimet ovat lähtöjä (8). Tulojen ja lähtöjen numerointi on toteutettu ylhäältä alas nousevasti (X0-X17, Y0-Y17). Alimmaisat liittimet (1-8) ovat erikoismoduuleja var-

ten. Liitinpaneelissa on lisäksi 4 kappaletta 37-napaista D-liitintä. Näistä tuloiksi on merkitty IN1 ja IN2, ja lähdöiksi OUT1 ja OUT2. /8/

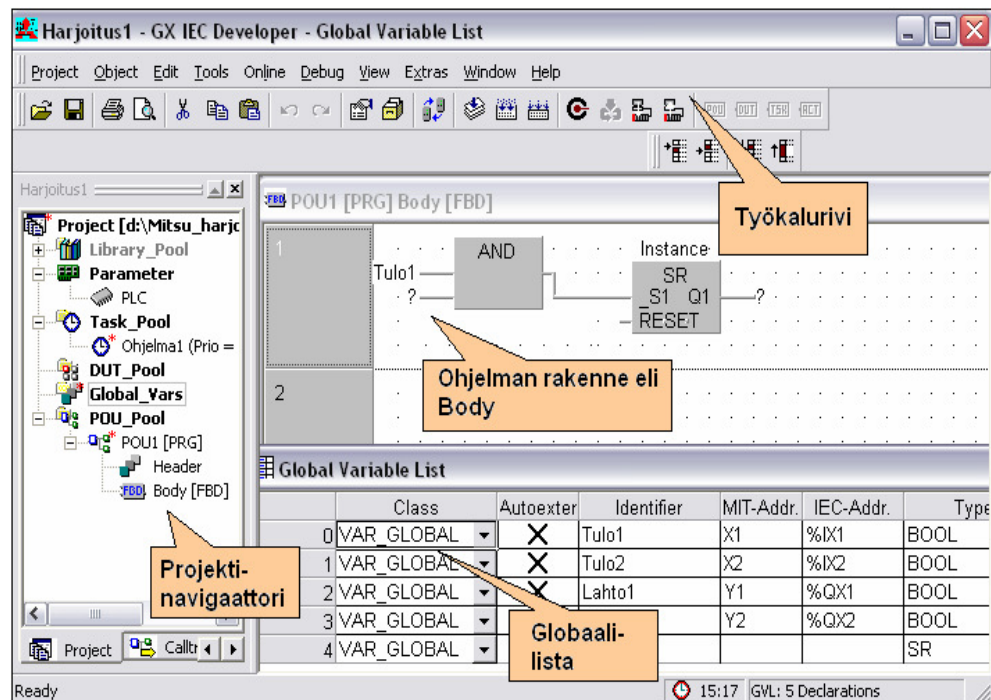
4.5 GX IEC Developer FX –ohjelma

4.5.1 Yleiskuva

GX IEC Developer FX on IEC 61131-3 -standardiin perustuva ohjelma. Tämä tarkoittaa sitä, että ohjelmalla on IEC 1131-3 -standardin mukainen rakenne sekä standardiin perustuvat ohjelmanosat. Ohjelma käyttää myös standardiin asetettuja ohjelmointikieliä, joita ovat käskylista (IL), relekaavio (LD), toimilohkokaavio (FBD), sekvenssikaavio (SFC), MELSEC-käskylista (MELSEC IL) ja rakenteinen teksti (ST)./14;15/

Ohjelma soveltuu pienille ja suurille projekteille. Monipuolisen ohjelmointieditorin ansiosta sillä voidaan toteuttaa myös hyvin monimutkaisia projekteja. GX IEC Developer -ohjelmaan on asennettuna offline-simulaattori, joka mahdollistaa luotujen ohjelmien testaamisen pelkän ohjelman avulla. Varsinaisessa ohjelman suorittamisessa voidaan käyttää kaikkia Mitsubishi Electric -logiikoita, joita ovat FX-, A-, QnA- ja Q-logiikat. /14;15/

Kuvassa 4 on esitetty ohjelmaikkunan rakenne ja valikot. Ikkunan vasemmassa reunassa on projektinavigaattori, josta löytyy kuvat kaikille objektityypeille. Työkalurivi löytyy ohjelmaikkunan ylälaidasta. Global Variable List eli globaalilista sisältää tiedot muuttujista, niiden osoitteista sekä datatyypeistä. Lista on käytettävissä kaikissa ohjelmanosissa. Ohjelman rakenne eli body, jonne varsinainen ohjelma kirjoitetaan, on jaettuna virtapiireihin helpottamaan ohjelmointia. Sekä body että globaalilista saadaan avattua projektinavigaattorista. /12;15/



Kuva 4 Ohjelman käyttöliittymän rakenne /15;31/

4.5.2 Ohjelman käynnistys

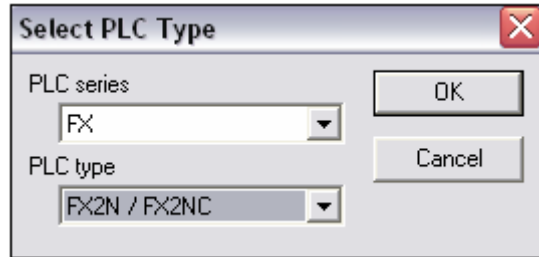
Käynnistä ohjelma Windows-perustilasta joko GX IEC Developer 6.10 -pikaku-
vakkeesta tai valitsemalla *Start* → *Programs* → *MELSOFT Application* →
GXIEC Developer 6.10 → *GX IEC Developer*. Tämän jälkeen aukeaa GX IEC
Developer -ikkuna (kuva 5). Paina OK-painiketta, jotta voit jatkaa. /2;7/



Kuva 5 GX IEC Developer -ikkuna

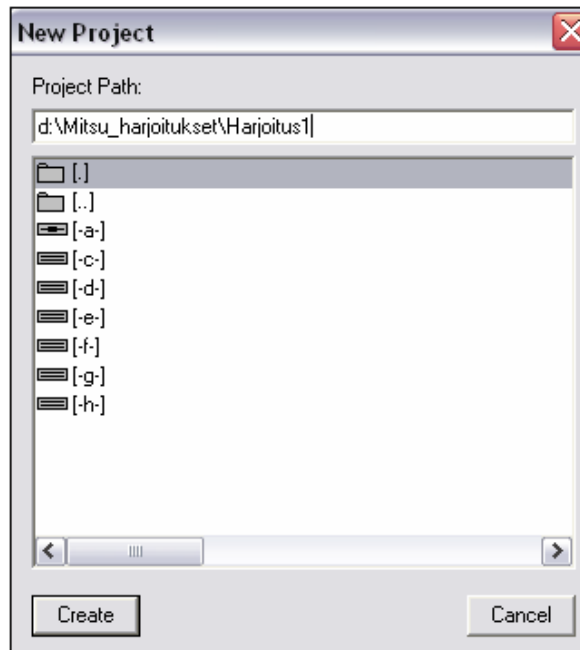
4.5.3 Uuden projektin luominen

Valitse ylävalikosta *Project* → *New*, jolloin *Select PLC Type* -ikkuna avautuu (kuva 6). Valitse haluttu PLC-tyyppi (PLC type) ja paina OK. /2;7/



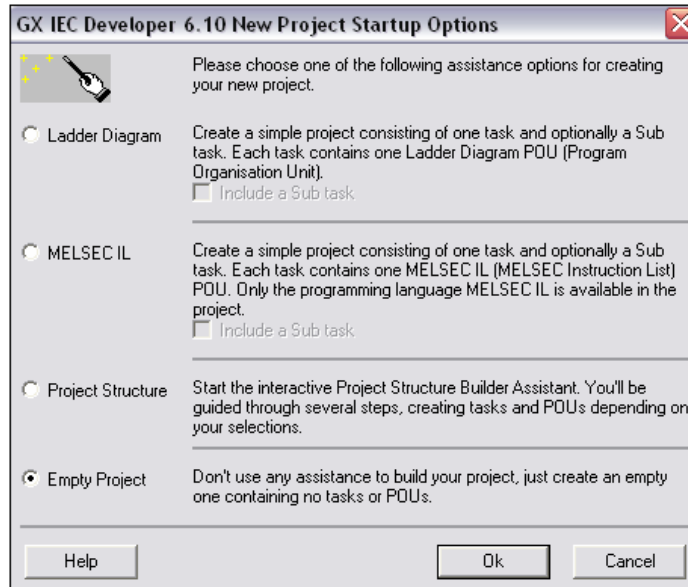
Kuva 6 PLC-tyypin valintaikkuna

Kirjoita avautuvaan *New Project* -ikkunaan (kuva 7) projektille nimi: esimerkiksi *Harjoitus 1*. Luodaan uusi projekti *Create*-painikkeella.



Kuva 7 Projektin nimeäminen

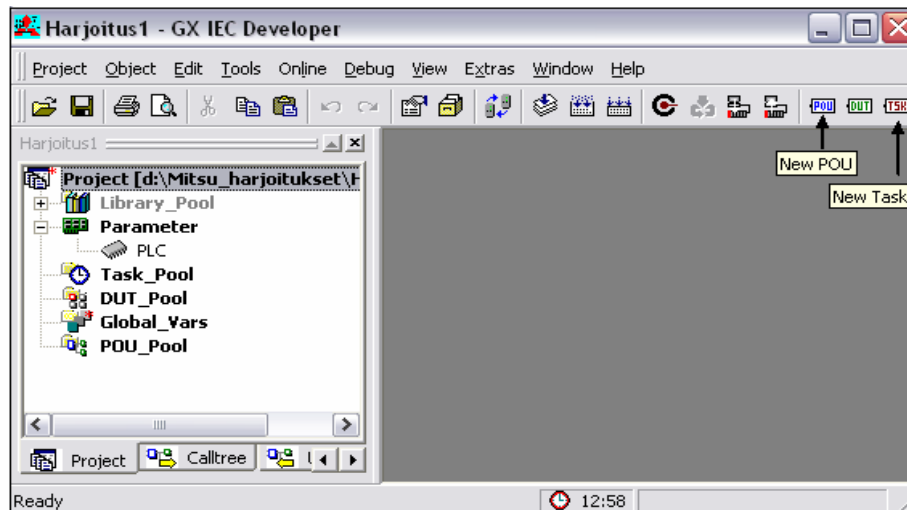
Valitse avatuvasta ikkunasta (kuva 8) *Empty Project* ja paina OK. Tämän jälkeen ohjelma saattaa ilmoittaa: "Attention: The 'OnlineProgram Change' functionality is limited for this CPU type!" Tämä välihuomautus kuitataan painamalla OK. /2;7/



Kuva 8 Tyhjän projektin valitseminen

4.5.4 Ohjelman alustaminen

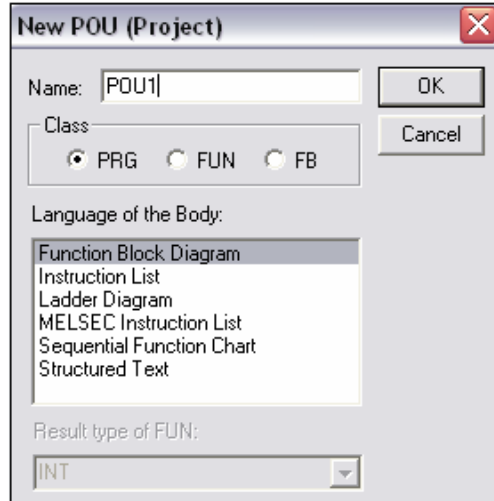
Edellisten kohtien suorittamisen jälkeen uusi projekti on luotu, ja ohjelmaikkuna on kuvan 9 mukainen. Projektin rakenne näkyy projektinavigaattorissa ikkunan vasemmassa reunassa. /2;7/



Kuva 9 Varsinainen ohjelmaikkuna

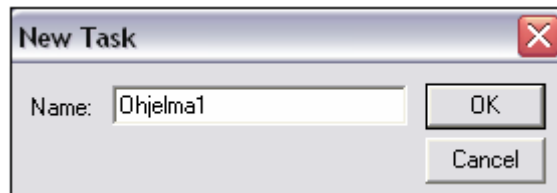
Seuraavaksi muodostetaan rakenneyksikkö POU. Valitse työkalurivistä painike *New POU* (kuva 9). Toiminto löytyy myös ylävalikosta seuraavasti: *Object* → *New* → *POU*. Nimitään tämä rakenneyksikkö ja valitaan haluttu esitystapa (kuva 10), joka voi olla Program (PRG), function (FUN) tai function block (FB).

Hyväksy valinnat painamalla OK. Tällöin projektinavigaattoriin ilmestyy merkintä POU1 [PRG], joka ilmaisee rakenneyksikön nimen sekä valitun esitystavan. /2;7/



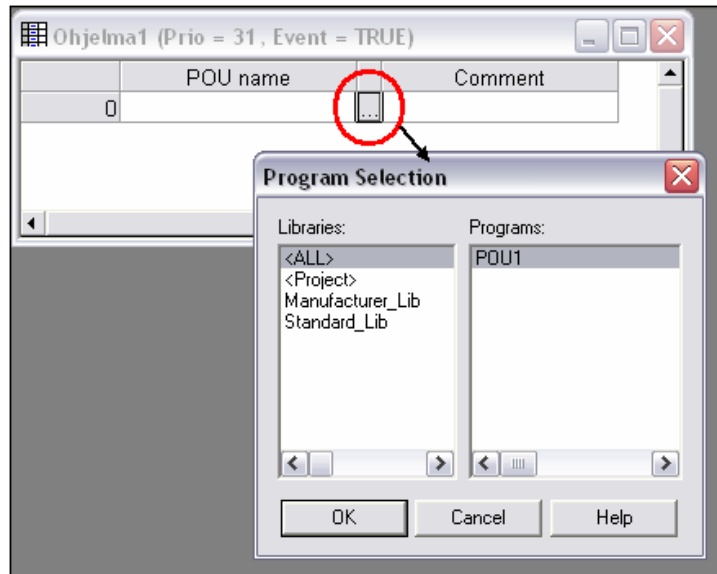
Kuva 10 POU-rakenneyksikön nimeäminen

Tehtävän asettelu suoritetaan työkalurivin painikkeella *New Task* (kuva 9) tai valitsemalla ylävalikosta seuraavasti: *Object* → *New* → *Task*. Nimeä tehtävä (kuva 11) ja paina OK. /2;7/



Kuva 11 Tehtävän nimeäminen

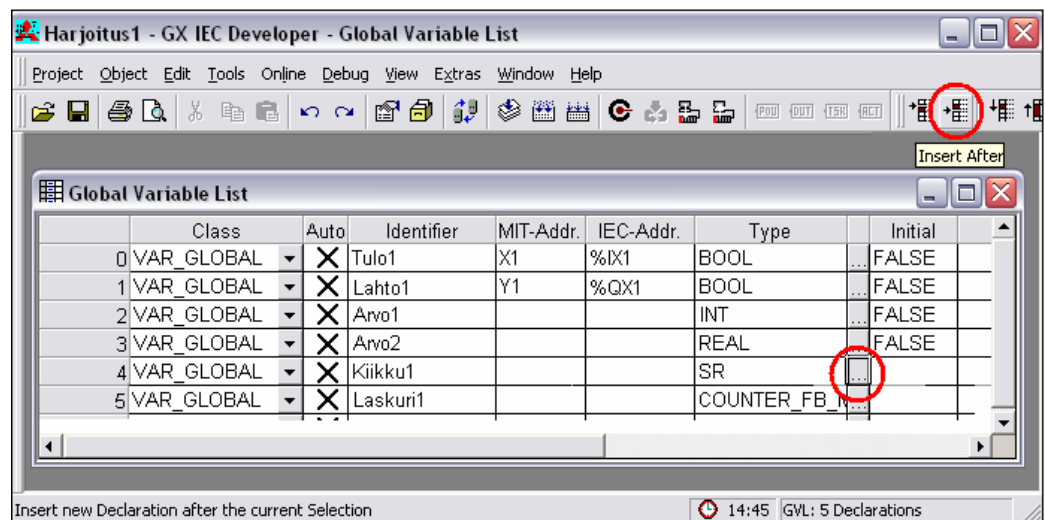
Näpätä hiirellä projektin rakennepuusta Task Pool -kohtaa ja valitse nimetty tehtävä (Ohjelma1) kaksoisnäpäyttämällä sitä. Avautuvasta ikkunasta painetaan POU name -kohdasta painiketta, jossa on kolme pistettä (kuva 12). Näytölle avautuu Program Selection -ikkuna, josta valitaan ohjelmaksi POU1 ja painetaan OK. Tallenna ohjelma tämän jälkeen ja sulje ikkuna. /2;7/



Kuva 12 POU-ohjelman valitseminen

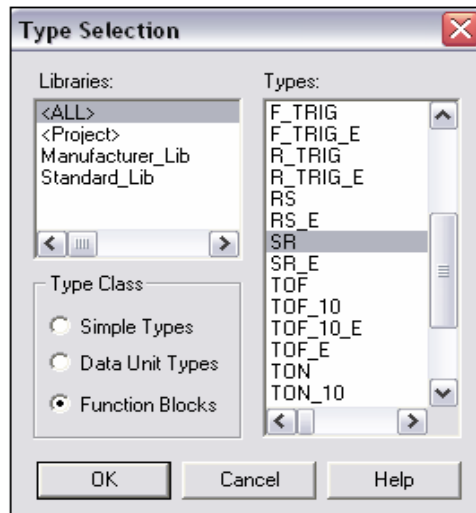
4.5.5 Yleismuuttujien listaaminen

Ohjelman alustaminen on suoritettu edellisillä toimenpiteillä. Seuraavaksi listataan muuttujat globaalilistaan ja annetaan niille I/O-osoitteet. Valitse projektinavigaattorista Global_Vars kaksoisnäpättämällä, jolloin ruudulle avautuu Global Variable List (kuva 13). Kirjoita Identifier -sarakeeseen muuttujalle nimi: esimerkiksi Tulo 1. Kirjoita tämän jälkeen MIT-Address-sarakeeseen absoluuttinen osoite (esim. X1) ja paina enter, jolloin rivin muut sarakkeet täyttyvät. Seuraava rivi luodaan painikkeella *Insert After*, joka luo uuden rivin edellisen rivin alle. /2;/



Kuva 13 Globaalilista ja muuttujat

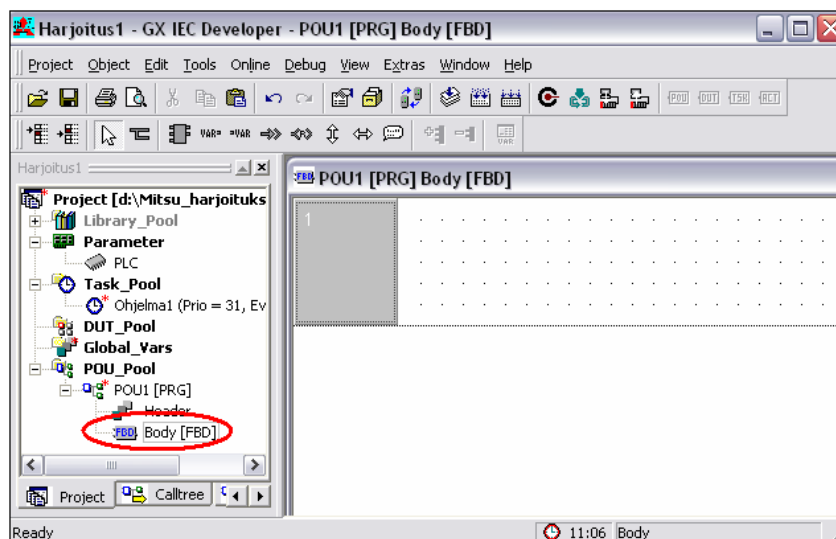
Esimerkiksi kiikuille, ajastimille, laskureille ja arvoille tulee vaihtaa oikeanlainen datatyyppi. Tämä tapahtuu painamalla Type-sarakkeessa olevaa painiketta. Painikkesta avautuu Type Selection -ikkuna (kuva 14), josta valitaan haluttu datatyyppi, ja painetaan OK. Paina rasti Autoextern-sarakkeen jokaiselle muuttujan riville, jolloin rivit kopioituvat automaattisesti jokaisen POU-rakenneyksikön muuttujalistaan. Lopuksi ohjelma tallennetaan. /2;7/



Kuva 14 Datatyyppin valinta

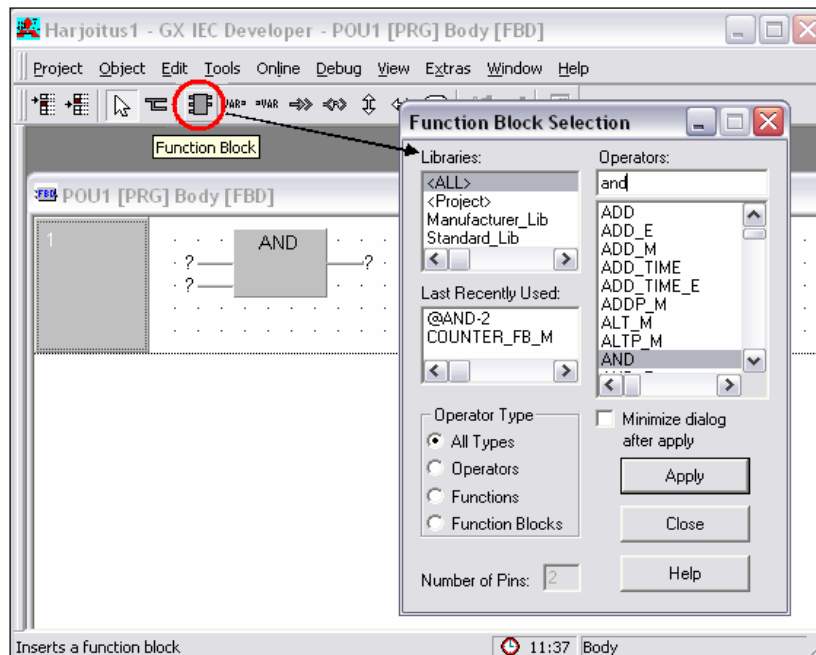
4.5.6 Ohjelman luominen

Kaksoisnäpätä projektinavigaattorissa olevan POU_Pool-kohdan alla olevaa Body [FBD]-kohtaa (kuva 15). Ohjelma aukaisee editointi-ikkunan, johon varsinainen ohjelma voidaan luoda. /2;7/



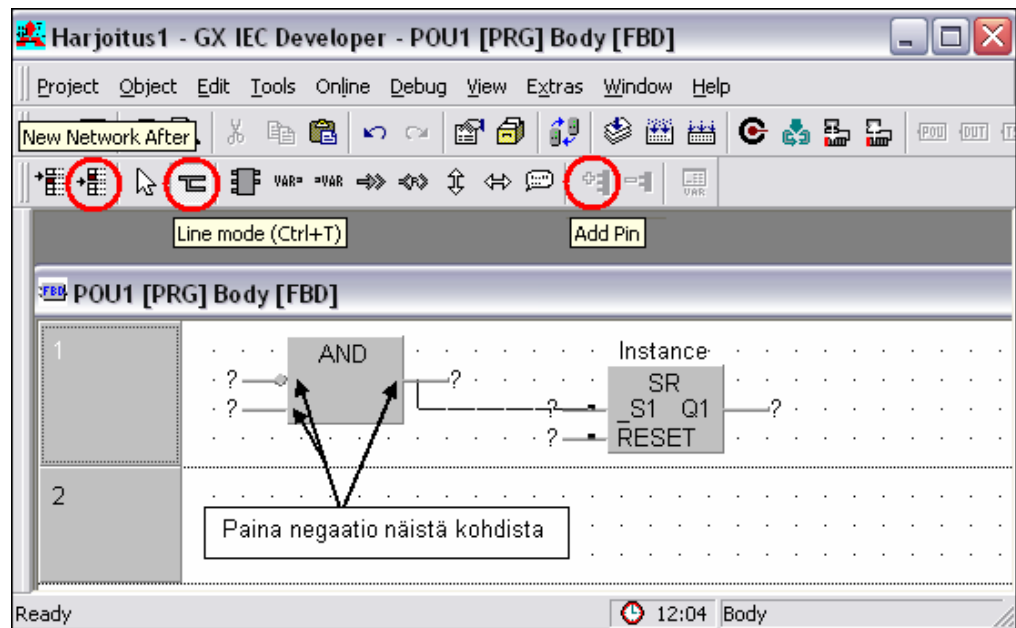
Kuva 15 Ohjelman editointi -ikkuna

Function Block -painikkeesta aukeaa Function Block Selection -ikkuna (kuva 16), josta voidaan valita halutut operaatiot virtapiireihin. Tämä tapahtuu valitsemalla operaatio listalta (AND), ja hyväksymällä se Apply-painikkeella. Vie seuraavaksi kursori ohjelmointialueelle, ensimmäiseen virtapiiriin, ja näpäytä hiiren vasenta painiketta, jolloin operaatio ilmestyy ohjelmointialueelle. /2;7/



Kuva 16 Operaation valinta

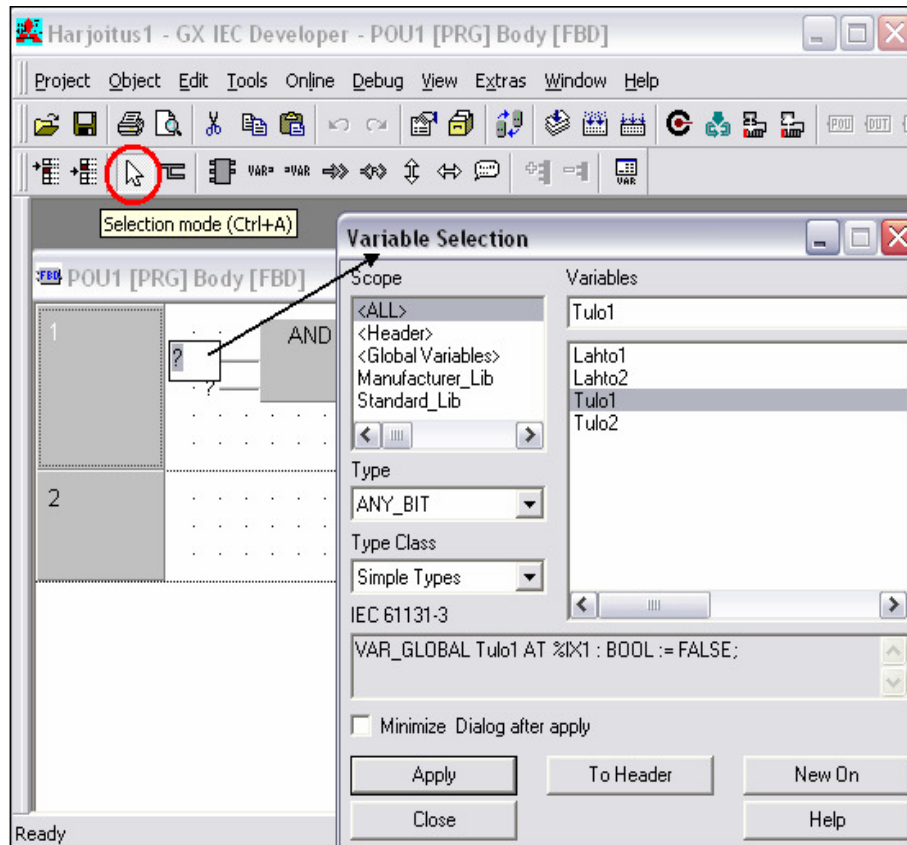
Painikkeella *New Network After* saadaan luotua uusi virtapiiri edellisen alle (kuva 17). Operaatioiden väliset yhteydet saadaan painikkeella *Line mode* (*Ctrl+T*). Tämä tapahtuu viemällä kursori lähtöön ja napauttamalla hiiren vasen painike pohjaan. Vie näppäin pohjassa kursori toisen operaation sisäänmenoon. Odota, että mustat neliöt tulevat näkyviin, ja napauta kaksi kertaa hiiren vasenta painiketta. *Add Pin* -painikkeella voit lisätä sisäänmenojen lukumäärää operaation ollessa aktiivisena. Vastaavasti miinusmerkkisellä painikkeella voit vähentää sisäänmenoja. Tulojen ja lähtöjen muuttaminen negaatioksi tapahtuu kaksoisnäpäyttämällä kuvassa 17 olevien nuolien osoittamista kohdista. Ohjelmaan aukeavasta *Signal Configuration* -ikkunasta rastitetaan *Negation*-kohta ja painetaan OK. /2;7/



Kuva 17 Muita työkalurivin toimintoja ja negaation muodostaminen

Muuttujat operaatioille saadaan *Selection mode (Ctrl+A)* -painikkeen avulla (kuva 18). Vie kursori tulon/lähdön päälle, ja napauta se aktiiviseksi. Tällöin tulo/lähtö muuttuu laatikoksi, jonka sisällä on kysymysmerkki. Paina tämän jälkeen hiiren oikea painike alas, jolloin ruudulle ilmestyy Variable Selection -ikkuna. Valitse haluttu muuttuja listalta. Huomaa, että saat enemmän vaihtoehtoja vaihtamalla Type Class -kohtaan toisen tyyppivaihtoehdon. Haluttu muuttuja hyväksytään Apply-painikkeella. Esimerkiksi kiikkujen, laskureiden ja ajastimien yläpuolella lukee Instance. Tämän tilalle haetaan globaalilistaan kirjoitettu nimitys, esimerkiksi Kiikku1 tai Laskuri1, käytettävän operaation mukaisesti. Nimitys haetaan Variable Selection -ikkunasta samalla tavalla kuin muuttujat.

/2;7/

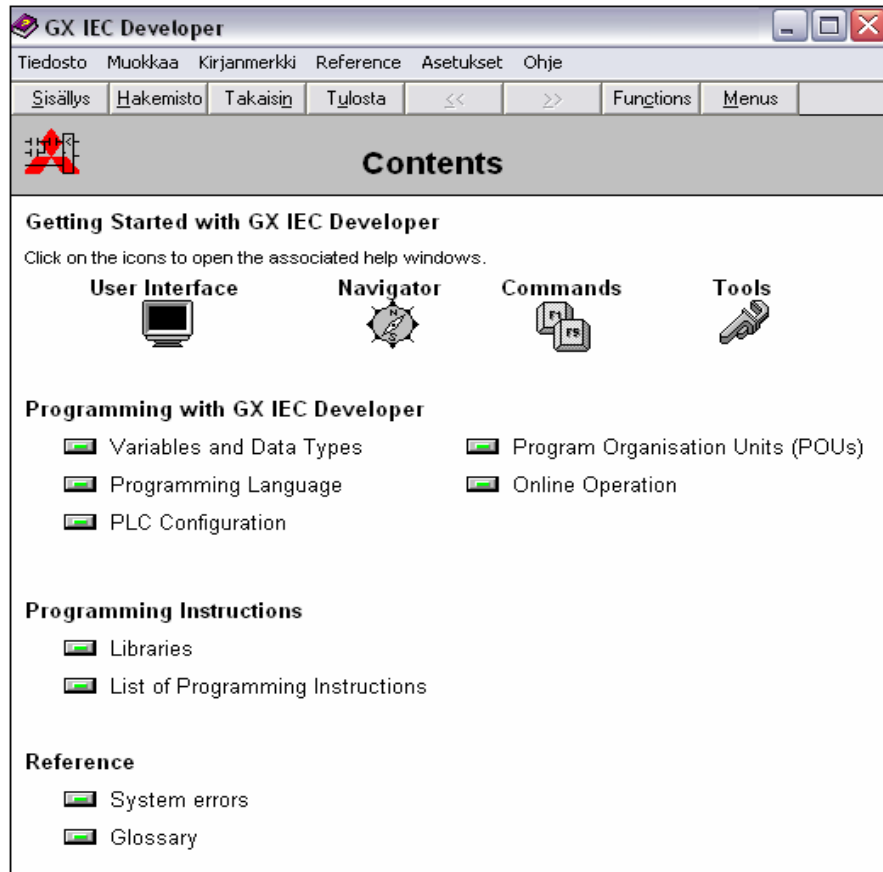


Kuva 18 Muuttujien hakeminen

4.5.7 Aputoiminta

Ongelmatilanteita varten GX IEC Developer -ohjelmassa on kattava Online-Help -aputoiminta, joka avautuu F1-painikkeella. Apujärjestelmä sisältää tietoa käyttöliittymästä, projektinavigaattorista, käskyistä ja työkaluista (kuva 19) sekä tarjoaa vastauksen myös ohjelmoinnissa esiintyviin ongelmiin. Tietoa löytyy muuttujista, datatyypeistä, konfiguroinnista, ohjelmakielistä sekä rakenneyksiköistä. Myös kaikki käytettävät funktiot löytyvät kuvattuina IEC standardin mukaisesta kirjastosta, joka aukeaa Functions-painikkeen avulla. /2;31/

Apujärjestelmässä voi lisäksi tutkia virheilmoitusten tarkoitusta sekä tarkistaa ohjelmassa ilmenevien sanojen merkitykset sanaston avulla. Järjestelmä sisältää myös etsintätoiminnon, jonka avulla tarvittava tieto on helposti ja nopeasti löydettävissä. /2;31/

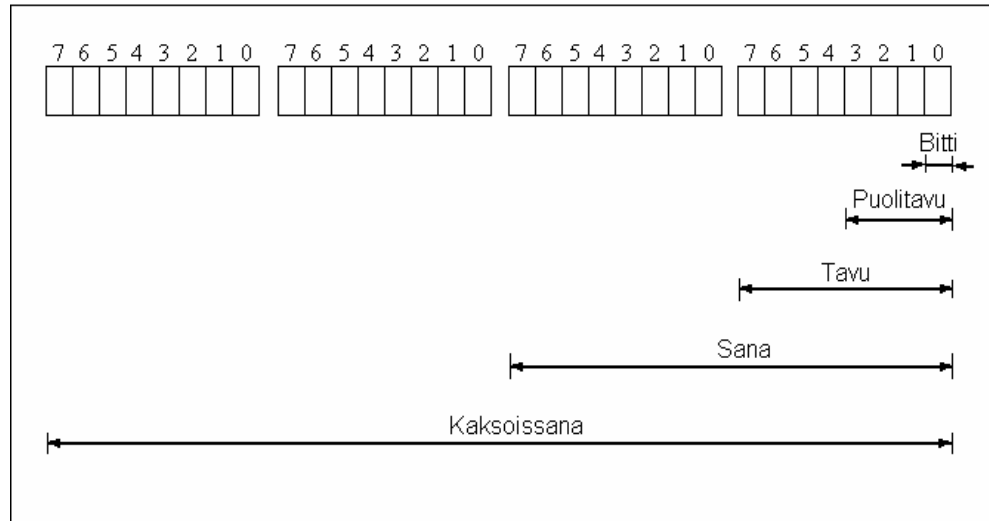


Kuva 19 Apujärjestelmän sisältöikkuna

4.6 Osoitteet

4.6.1 Bitti, tavu, puolitavu, sana ja kaksoissana

Bitti (b) eli binäärinumero on pienin mahdollinen binäärisen signaalin tietoyksikkö, jonka tilana voi olla joko 0 ("ei ole") tai 1 ("on"). Bittiä verrataan usein kytkimeen, joka antaa signaalitilaa "kytkin päällä" tai "kytkin pois päältä". Usein bittejä kuitenkin yhdistellään, jotta voitaisiin laskea suurempia kokonaisuuksia. Esimerkkinä tällaisesta yhdistelmästä on *tavu* (B) (kuva 20), joka on kahdeksan bitin kokonaisuus. Tavun avulla voidaan esittää yhteensä 256 erilaista arvoa nollasta 255:een asti. Käytettävissä on myös neljästä bitistä koostuva *puolitavu*, sekä muita bittiyhdistelmiä, kuten kahdesta tavusta eli 16 bitistä koostuva *sana* (W) sekä kahdesta sanasta eli 32 bitistä muodostuva *kaksoissana* (D). /6;9;22;23;24/



Kuva 20 Bittiyhdistelmät ja niiden suuruudet

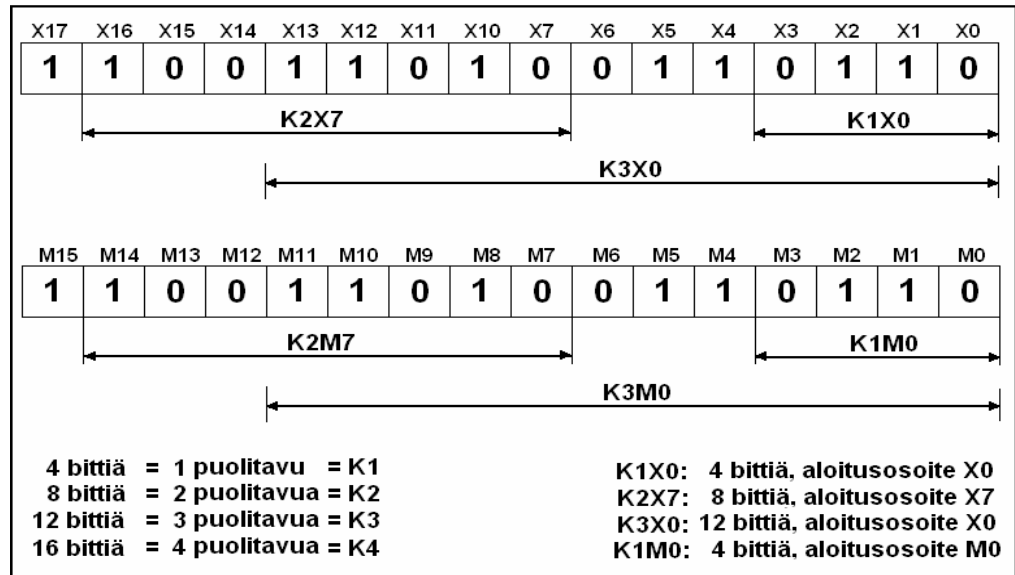
4.6.2 I/O-osoitteet

Jotta GX IEC Developer FX -ohjelmaa voidaan hyödyntää toimilaitteilla, tulee muuttujille antaa logiikan muistipaikkaa vastaavat osoitteet. Jos muuttujille ei anna osoitteita, ohjelma määrää ne itse. Logiikan keskusyksikön (CPU) muistipaikkojen käyttöönotto tapahtuu määrittelemällä muuttujat POU-rakenneyksikössä. Osoitteet voidaan kirjoittaa viittaamaan myös johonkin fyysiseen tulo- ja lähtöpaikkaan. /2/

I/O-osoitteen antaminen FX-sarjan laitteissa suoritetaan oktaalimuodossa eli 8-kantaisena lukujärjestelmänä. Kaikkiaan käytettävissä on 128 osoitetta. FX2N-32MR-logiikan bool-muuttujia ovat tulot (X0-X7, X10-X17) sekä lähdöt (Y0-Y7, Y10-Y17). Muistipaikkoja (M) logiikassa on yhteensä 3072 ja paikat ovat desimaalinumeroituja. Tulot (X) ja lähdöt (Y) ovat puolestaan oktaalinumeroituja ja tulevat Boolean algebran yhtälöistä. Muistipaikkojen lisäksi muita sisäisiä muuttujia ovat askelmuistipaikka (S), virhemuistipaikka (F), ajastin (T), laskuri (C), datarekisteri (D), indeksirekisteri (V, Z) sekä osoitin (P). /3;12;15;16;17;26/

Osoitteet voidaan määritellä puolitavuina eli 4 bitin ryhminä 32 bittiin saakka (kuva 21). Tällöin osoite ilmaistaan tavutiedon tunnuksen, ja tavutiedon aloitusosoitteen avulla. Tavutiedon tunnus ilmaisee, kuinka monta bittiä kyseinen koodi sisältää. Esimerkiksi tunnus K1 sisältää 1 puolitavun eli 4 bittiä. Vastavasti K2 sisältää 2 puolitavua eli 8 bittiä. Tavutiedon aloitusosoite on kohta,

josta tavutiedon tunnuksen mukainen bittijono alkaa. Tämä aloituskohta voidaan määrätä mihin tahansa kohtaan bittijonoa. /17/



Kuva 21 Tavutiedon muodostaminen /12;17/

4.6.3 Erikoismuistipaikat

Erikoisapumuistit (M) toimivat muistipaikkoina erikoistoimintoja varten. Niiden suuren lukumäärän vuoksi liitteeseen 1 on koottu vain pieni joukko erikoisapumuisteja. Lisää erikoismuistipaikkoja ja niiden selityksiä löytyy FX-sarjan käyttöohjeesta (lähde 3). /3;17/

4.6.4 Virhekoodit

Kun ohjelmassa ilmaantuu ongelmia, viat ilmoitetaan virhekoodien avulla. Liitteessä 2 on esiteltyä yleisimpiä ohjelman käytössä olevia virhekoodeja sekä virheiden korjausehdotuksia.

4.6.5 Nopeat laskurit /3/

Nopeat laskurit, toiselta nimeltään pikalaskurit, ovat ulkoisia signaaleja käsitteleviä 32-bittisiä laskureita. Näille keskeytysperiaatteella toimiville laskureille on varattuna yhteensä 21 osoitetta (C235-C255). Nopeaa laskuria ohjelmoitaessa on pidettävä huolta, ettei käytetä tuloja, jotka toinen laskuri on jo varannut. Las-

kentatuloina toimivat kuusi tuloa (X0-X5), ja niiden kaksinkertainen varaus ei ole sallittua.

Nopeita laskureita löytyy neljää eri tyyppiä, jotka ovat:

- 1-vaiheinen laskuri ilman käynnistys- ja nollaustoimintoja (start/reset)
- 1-vaiheinen laskuri, jossa käynnistys- ja nollaustoiminnot
- 2-vaiheinen laskuri, kaksisuuntainen
- A/B-vaihe-laskuri.

1-vaiheiset laskurit (taulukko 1) ovat jaettuina kahteen eri ryhmään: laskureihin, jotka ovat ilman käynnistys- ja nollaustoimintoja (C235-C240), sekä laskureihin, jotka sisältävät kyseiset toiminnot (C241-C245). Näistä ensimmäinen laskurityyppi sisältää vain yhden laskentatulon. Jälkimmäinen tyyppi sisältää laskentatulon sekä nollaustulon, ja osassa laskureista (C244-C245) on lisäksi käynnistystulo. Nämä laskurit toimivat kuten muut 32-bittiset laskurit. Kun asetusarvo on saavutettu, laskurin lähtö asetetaan. Ylöspäin laskettaessa asetettu lähtö pysyy päällekytkettynä, kun taas alaspäin laskettaessa se nollautuu. Laskureiden (C235-C245) laskentasuunta voidaan määrätä erikoisapumuistien (M8235-M8245) avulla. Erikoisapumuistin numerointi vastaa kyseisellä numerolla varustetun laskurin osoitteeseen. Esimerkiksi erikoisapumuistin ollessa päällekytkettynä laskuri laskee alaspäin. Vastaavasti, erikoisapumuistin ollessa kytkettynä pois päältä, laskuri laskee ylöspäin.

Taulukko 1 Laskurityypit /3/

X/M	1-vaiheinen laskuri ilman käynnistys- ja nollaustoimintoja (start/reset)						1-vaiheinen laskuri, jossa käynnistys- ja nollaustoiminnot					2-vaiheinen laskuri, kaksisuuntainen					A/B-vaihe-laskuri				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0	U/D						U/D			U/D		U	U		U		A	A		A	
X1		U/D					R			R		D	D		D		B	B		B	
X2			U/D					U/D		U/D			R		R			R		R	
X3				U/D				R		R			U		U			A		A	
X4					U/D				U/D				D		D			B		B	
X5						U/D		R					R		R			R		R	
X6*										S					S					S	
X7*										S					S					S	
M8...	Erikoisapumuistit M8235-M8245 Apumuistin ollessa asetettuna, laskuri laskee alaspäin. Apumuistin ollessa asettamatta, laskuri laskee ylöspäin.										Erikoisapumuistit M8246-M8255 Laskurin laskeessa alaspäin, apumuisti asetetaan. Laskurin laskeessa alaspäin, apumuisti nollautuu.										
U: ylöspäin laskeva tulo D: alaspäin laskeva tulo A: A-vaihe-tulo B: B-vaihe-tulo R: nollaustulo S: käynnistystulo																					
* Tulot X6 ja X7 toimivat käynnistysignaaleina. Niitä ei voida käyttää laskuritoiminnoissa.																					

2-vaiheiset laskurit sisältävät yhden laskentatulon ylöspäin sekä yhden laskentatulon alaspäin. Osassa laskureista on myös käynnistys- ja nollaustulo. Taulukossa 1 on esitettyä nopeiden laskureiden ja eri tulojen väliset toiminnot. Esimerkiksi laskuri C246 laskee ylöspäin tulolla X0 ja alaspäin tulolla X1. Tämän sarjan laskureista poikkeuksena on laskuri C250, joka sisältää sekä nollausta että käynnistystulon. Laskurin suunta asetetaan erikoisapumuistin M8250 avulla samoin kuin 1-vaiheisten laskureiden tapauksessa.

A/B-vaihelaskurit (C252-C255) sisältävät kukin A- ja B-vaiheen laskentatulot. Osassa on myös nollausta ja käynnistystulot. Laskurin laskentasuunta määrätään edellä mainittujen A- ja B-vaiheen tulojen signaalien avulla. Ylöspäin laskevan laskurin A-vaihe on ON-tilassa ja B-vaiheessa on nouseva pulssinreuna. Vastavasti alaspäin laskevan laskurin B-vaiheessa on laskeva pulssinreuna A-vaiheen ollessa myös ON-tilassa.

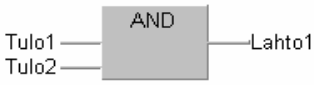
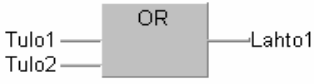
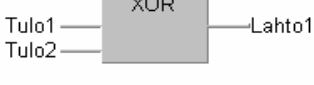
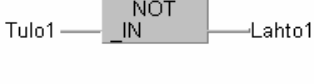
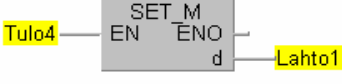
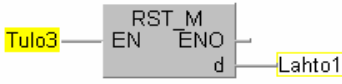
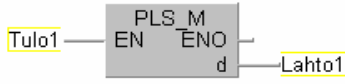
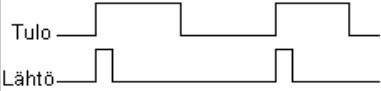
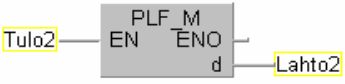
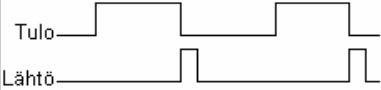
4.7 Operaatiot

GX IEC Developer FX -ohjelma luodaan operaatioiden avulla. Koska ohjelma sisältää suuren määrän käskyjä eri tarkoituksiin, on tähän lukuun koottu vain käytetyimpiä operaatioita. Lisätietoa erilaisista käskyistä löytyy ohjelmamanuaalista sekä ohjelmasta löytyvästä Online Help -aputoiminnasta.

4.7.1 Peruskäskyt

Perusoperaatiot (taulukko 2) löytyvät Function Block -painikkeella (kuva 16). Perusoperaatioita voidaan käyttää suoraan projektin luomisessa, joten niitä ei tarvitse kirjoittaa globaalilistaan.

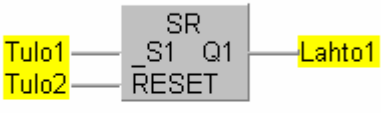
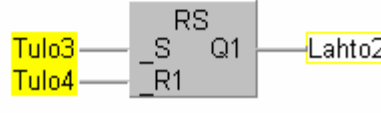
Taulukko 2 Perusoperaatiot ja niiden kuvaus /6;16;31/

Käsäy	Kuvaus															
<p>JA-käsäy</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>=1</td></tr> <tr><td>1</td><td>0</td><td>=0</td></tr> <tr><td>0</td><td>1</td><td>=0</td></tr> <tr><td>0</td><td>0</td><td>=0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	=1	1	0	=0	0	1	=0	0	0	=0
Tulo1	Tulo2	Lähtö1														
1	1	=1														
1	0	=0														
0	1	=0														
0	0	=0														
<p>TAI-käsäy</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>=1</td></tr> <tr><td>1</td><td>0</td><td>=1</td></tr> <tr><td>0</td><td>1</td><td>=1</td></tr> <tr><td>0</td><td>0</td><td>=0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	=1	1	0	=1	0	1	=1	0	0	=0
Tulo1	Tulo2	Lähtö1														
1	1	=1														
1	0	=1														
0	1	=1														
0	0	=0														
<p>EHDOTON TAI -käsäy</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>=0</td></tr> <tr><td>1</td><td>0</td><td>=1</td></tr> <tr><td>0</td><td>1</td><td>=1</td></tr> <tr><td>0</td><td>0</td><td>=0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	=0	1	0	=1	0	1	=1	0	0	=0
Tulo1	Tulo2	Lähtö1														
1	1	=0														
1	0	=1														
0	1	=1														
0	0	=0														
<p>EI-käsäy</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>=0</td></tr> <tr><td>0</td><td>=1</td></tr> </tbody> </table>	Tulo1	Lähtö1	1	=0	0	=1									
Tulo1	Lähtö1															
1	=0															
0	=1															
	<p>Set- eli asetuskäsäy</p> <p>Tulon ollessa päällä, asettuu lähtö tilaan 1.</p>															
	<p>Reset- eli nollauskäsäy</p> <p>Tulon ollessa päällä, asettuu lähtö tilaan 0.</p>															
	<p>Nousevan reunan liipaisu</p> <p>Kun tulo muuttuu 0->1, antaa lähtö yhden syklin ajan tilaa 1.</p> 															
	<p>Laskevan reunan liipaisu</p> <p>Kun tulo muuttuu 1->0, antaa lähtö yhden syklin ajan tilaa 1.</p> 															

Muistitoiminta GX IEC Developer -ohjelmassa toteutetaan muistipiirien (taulukko 3) avulla. Muistipiirin muisti asetetaan päälle asetuskäsäyllä S ja nollataan nollauskäsäyllä R. Jos sekä asetusta että nollaus ovat päällä, muistipiirin toiminta määräytyy sen tyyppin mukaan. SR-muistipiirillä eli -kiikulla on dominoiva asetusta, ja vastaavasti RS-kiikulla on dominoiva nollaus. Muistikäsäyt löytyvät

Function Block Selection -valikosta (kuva 16), ja ne kirjoitetaan globaalilistaan esimerkiksi nimellä Kiikku1. Globaalilistaan ei kirjoiteta osoitetta, mutta datatyyppi pitää muistaa muuttaa käytettävän kiikun mukaiseksi. /6/

Taulukko 3 Muistipiirit ja niiden toiminnan kuvaus /6;31/

Käsky	Kuvaus
<p>SR-kiikku</p> <p>Kiikku1</p> 	<p>Muistipiiri "dominoiva asetus"</p> <p>Kun molemmat tulot ovat kytkettyinä päälle, muisti on signaalitilassa 1. Kun pelkkä RESET on päällä, muisti on signaalitilassa 0.</p>
<p>RS-kiikku</p> <p>Kiikku2</p> 	<p>Muistipiiri "dominoiva nollaus"</p> <p>Kun molemmat tulot ovat kytkettyinä päälle, muisti on signaalitilassa 0. Kun pelkkä S on päällä, muisti saa signaalitilakseen 1.</p>

4.7.2 Vertailijat

Vertailijat (taulukko 4) löytyvät kuvan 16 Function Block Selection -ikkunasta. Arvojen syöttäminen voidaan toteuttaa online-tilan ollessa päällä siten, että kaksoisnäpäytetään virtapiiristä operaation arvo-kohtaa. Tällöin ohjelma avaa Modify variable value -ikkunan, johon haluttu arvo syötetään. Hexadecimal-rasti poistetaan, jos arvoa ei haluta antaa heksadesimaalilukuna. /15;31/

Taulukko 4 Vertailukäskyt /15;31/

Käsky	Kuvaus
	<p>Yhtäsuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa yhtäsuuret, lähtö 1 on päällä.</p>
	<p>Erisuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa erisuuret, lähtö 1 on päällä.</p>
	<p>Pienempi kuin -käsky</p> <p>Kun arvo 1 on pienempi kuin arvo 2, lähtö1 on päällä.</p>
	<p>Pienempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo 1 on pienempi tai yhtäsuuri kuin arvo 2, lähtö 1 on päällä.</p>
	<p>Suurempi kuin -käsky</p> <p>Kun arvo 1 on suurempi kuin arvo 2, lähtö 1 on päällä.</p>
	<p>Suurempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo 1 on suurempi tai yhtäsuuri kuin arvo 2, lähtö 1 on päällä.</p>

4.7.3 Ajastimet

Ajastimet (taulukko 5) löytyvät Function Block Selection -ikkunasta (kuva 16) muiden operaatioiden tavoin. Ajastimet kirjoitetaan globaalilistaan, kuten tulot ja lähdöt. Ajastimet voidaan nimetä globaalilistan Identifier-sarakkeeseen esimerkiksi nimillä Ajastin 1, Ajastin 2 ja niin edelleen. Ajastimille ei tarvitse kirjoittaa osoitteita. Riittää kun ajastimelle valitaan oikea datatyyppi (esim. TI-

MER_100_FB_M, TOF, TON) Type Selection -ikkunasta (kuva 14). TON- ja TOF-ajastimissa olevat ajannäyttäjät (ET) merkitään myös erikseen globaalilistaan, kuitenkin ilman Auto-sarakkeeseen tulevaa rastia. Identifier-sarakkeeseen voidaan kirjoittaa ajannäyttäjille nimet (TO4, TO8). Osoitteiksi (MIT-Addr.) ensimmäiselle on valittu D8 ja jälkimmäiselle D10. Lisäksi datatyypi tulee vaihtaa aikamuotoiseksi (TIME). /12;15;16;27/

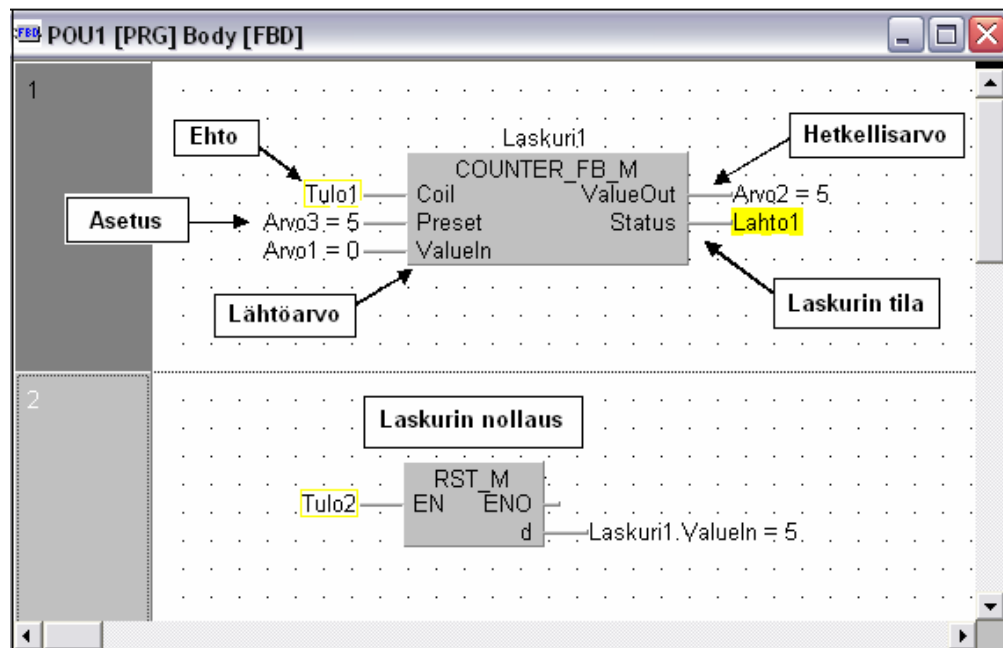
Taulukko 5 Ajastintyypit /27/

Ajastin	Kuvaus
<p>Ajastin1 TIMER_100_FB_M Tulo1 — Coil Arvo1 = 20 — Preset Arvo2 = 0 — ValueIn ValueOut — Arvo3 = 20 Status — Lahto1</p>	<p>Coil = ajastimen käynnistys Preset = ajastettava arvo (> ValueIn) ValueIn = lähtöarvo (yleensä 0) ValueOut = näyttää ajan Status = ilmoittaa kun aika on kulunut</p> <p>Tulo1 — Aika Lahto1 —</p>
<p>Ajastin2 TON Tulo1 — IN T#5s — PT Q — Lahto1 ET — TO4 = T#5s0.00ms</p>	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO4) Q = ilmoittaa kun aika on kulunut</p> <p>Tulo1 — Aika Lahto1 —</p>
<p>Ajastin3 TOF Tulo1 — IN T#5s — PT Q — Lahto1 ET — TO8 = T#0.00ms</p>	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO8) Q = ilmoittaa kun aika on kulunut</p> <p>Tulo1 — Aika Lahto1 —</p>

4.7.4 Laskurit

Laskurit löytyvät Function Block -painikkeella avautuvasta ikkunasta (kuva 16). Ennen käyttöä ne määritellään globaalilistaan samoin kuten tulot ja lähdöt, mutta ilman osoitetta. Käytettävän laskurin datatyypi pitää muuttaa laskurin mukaiseksi. Esimerkiksi ylöspäin laskeva peruslaskuri COUNTER_FB_M -laskuri (kuva 22) määritellään globaalilistaan Counter_FB_M-tyyppiseksi. /15;19/

Edellä mainitun peruslaskurin hetkellisarvo kasvaa, kun ehto kytketään vuoroin päälle ja pois päältä. Ennen tätä laskurille tulee antaa asetuasarvo, joka kuvan 22 esimerkissä on 5. Kun hetkellisarvo on yhtäsuuri kuin asetuasarvo, lähtö kytkeytyy päälle. Tällöin laskurin hetkellisarvo ei enää kasva. Tämän tyyppisen laskurin nollaamiseen tarvitaan erillinen virtapiiri ja siihen sijoitettava resetoitikäsky (RST_M). Esimerkissä laskurin 1 hetkellisarvo nollautuu tulon 2 ollessa päällä. Huomaa, että resetoitikäskyn d-lähtöhaaraan on kirjoitettu itse ”Laskuri1.ValueIn”. /31/



Kuva 22 Esimerkki ylöspäin laskevasta peruslaskurista /15;19;31/

Taulukkoon 6 on koottuna eri tyyppisiä laskureita kuvauksineen. Kullekin laskurille löytyy oma datatyyppinsä, joka pitää määritellä globaalistaan. Nämä laskurit eivät tarvitse erillistä resetoitikäskyä, kuten kuvan 22 laskuri.

Taulukko 6 Laskurityypit /31/

Ylöspäin laskevat laskurit	Kuvaus
<p style="text-align: center;">Laskuri5</p>	<p>Laskuriarvo (CV) kasvaa tulolla 1. Lähtö1 on päällä, kun laskuriarvo on saavuttanut esiasetusarvon (PV). Laskuriarvo resetoidaan tulolla 2.</p>
<p style="text-align: center;">Laskuri7</p>	<p>Laskuriarvo (CV) suurenee tulolla 2, tulo 1 ollessa päällä. Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtäsuuri kuin esiasetusarvon (PV). Laskuriarvo resetoidaan tulolla 3.</p>
Alaspäin laskevat laskurit	Kuvaus
<p style="text-align: center;">Laskuri3</p>	<p>Laskuriarvo (CV) pienenee tulolla 1. Lähtö1 on päällä, kun laskuriarvo on pienempi tai yhtäsuuri kuin nolla. Esiasetusarvolla (PV) saadaan laskuri laskemaan halutusta arvosta alaspäin. Esiasetusarvo saadaan ladattua tulolla 2, tulo 1 ollessa päällä.</p>
<p style="text-align: center;">Laskuri6</p>	<p>Laskuriarvo (CV) pienenee tulolla 2. Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtäsuuri kuin nolla. Esiasetusarvo (PV) saadaan ladattua tulolla 3, tulojen 1 ja 2 ollessa päällekytkettyinä.</p>
Ylös- ja alaspäin laskevat laskurit	Kuvaus
<p style="text-align: center;">Laskuri4</p>	<p>Laskuriarvo (CV) kasvaa tulolla 1, ja pienenee tulolla 2. Lähtö1 on päällä, kun laskuriarvo on suurempi tai yhtäsuuri kuin esiasetusarvo (PV). Lähtö2 on päällä, kun laskuriarvo on pienempi tai yhtäsuuri kuin nolla. Esiasetusarvo ladataan tulolla 4, tulojen 1 ja 2 ollessa päällä. Laskuriarvo resetoidaan tulolla 3.</p>
<p style="text-align: center;">Laskuri8</p>	<p>Laskuriarvo (CV) kasvaa tulolla 2 ja pienenee tulolla 3. Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtäsuuri kuin esiasetusarvo (PV). Lähtö2 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtäsuuri kuin nolla. Esiasetusarvon lataamisessa tulo 5 avulla, tulee tulot 1, 2 ja 3 olla päällä.</p>

4.7.5 Aritmetiikkakäskyt

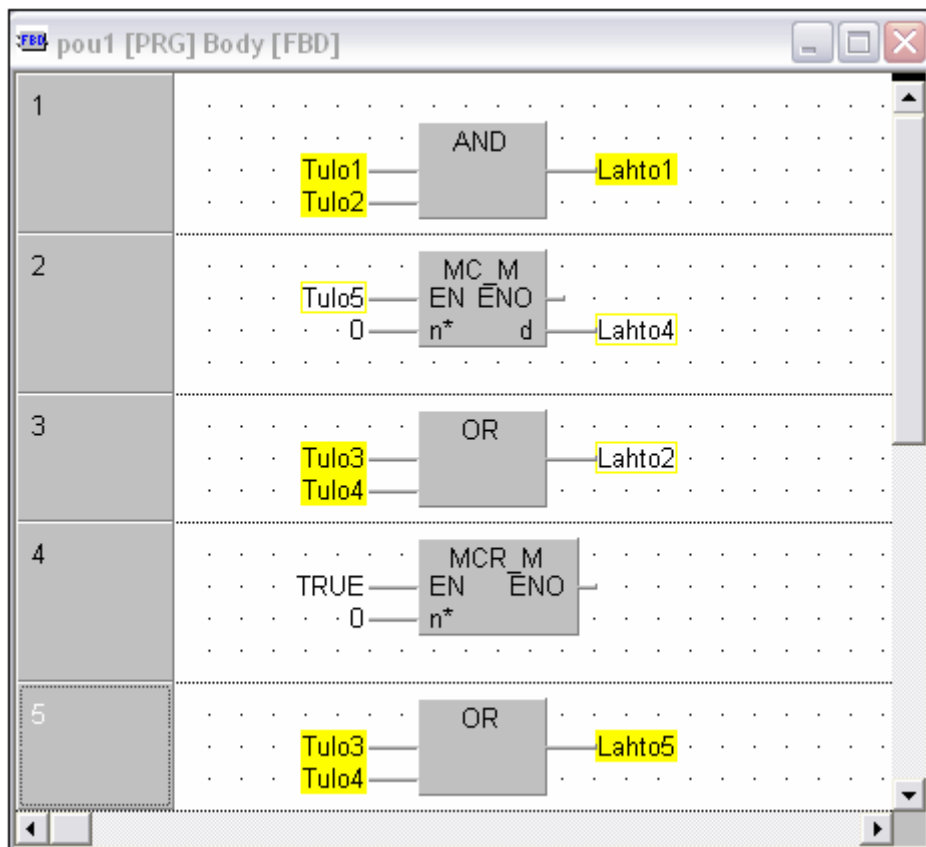
Aritmetiikkakäskyt eli laskentakäskyt (taulukko 7) löytyvät myös Function Block Selection -ikkunasta (kuva 16). Kun käytetään näitä käskyjä, on otettava huomioon, että laskentalohkon molempien puolien arvot ovat samaa datatyyppiä. Esimerkiksi $INT + INT = INT$, jolloin datatyyppinä on kokonaisluku (INT). Jos kokonaislukujen sijaan halutaan käyttää reaali-lukuja, globaalilistan datatyyppiksi on muutettava REAL (kuva 14). /15;16;17/

Taulukko 7 Laskentakäskyt /15;31/

Käsky	Kuvaus
	<p>Tulon 1 ollessa kytkettynä päälle, ADD_E -operaatio suorittaa arvojen 1 ja 2 yhteenlaskun. Tulos näkyy arvona 3.</p>
	<p>SUB_E-operaatio suorittaa arvojen 1 ja 2 vähennyslaskun.</p>
	<p>MUL_E -operaatio suorittaa arvojen 1 ja 2 kertolaskun.</p>
	<p>DIV_E -operaatio suorittaa arvojen 1 ja 2 jakolaskun.</p>
	<p>ABS-operaatio muuttaa arvon 1 positiiviseksi arvoksi.</p>

4.7.6 Master Control / Master Control Reset

Master Control (MC) eli pääkoskettimen asetus ja Master Control Reset (MCR) eli pääkoskettimen nollaus ovat yhdessä toimivia operaatioita. MC- ja MCR- operaatiot toimivat siten, että ohjelma ohittaa niiden väliin jäävät virtapiirit eikä suorita niissä olevia ehtoja, jos MC- operaatio ei ole päällä. Esimerkin (Kuva 23) virtapiirissä 3 olevan TAI-portin (OR) tulot ovat kyllä kytkettyinä päälle, mutta lähtö 2 on edelleen poissa päältä. Vasta kun MC- operaatio laitetaan päälle tulolla 5, kytkeytyy lähtö 2 päälle. Nämä operaatiot löytyvät Function Block Selection -ikkunasta (kuva 16). /28/

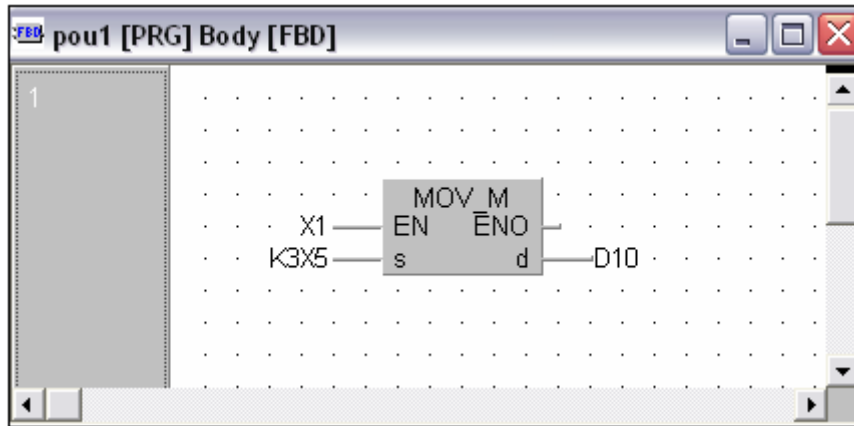


Kuva 23 Esimerkki MC- ja MCR- operaatioiden toiminnasta /28/

4.7.7 Tavutiedon siirto (MOV) I/O-avaruuteen

Datan siirto voidaan suorittaa GX IEC Developer FX -ohjelmassa MOV_M- operaation avulla, joka löytyy Function Block Selection -ikkunasta (kuva 16). Kuvassa 24 olevan tavutiedon (K3X5) siirtäminen tapahtuu, kun tulo X1 kytketään päälle. Tällöin operaatio siirtää tulosta X5 lähtien 12 bitin tiedon (K3 = 3 x

4 bittiä = 12 bittiä) datarekisteriin D10. Tavutiedon tunnuksen määräytyminen on esitettyä tarkemmin luvussa 6.2 (I/O-osoitteet). /12;29/



Kuva 24 Tavutiedon siirtäminen datarekisterille /29/

4.8 POU

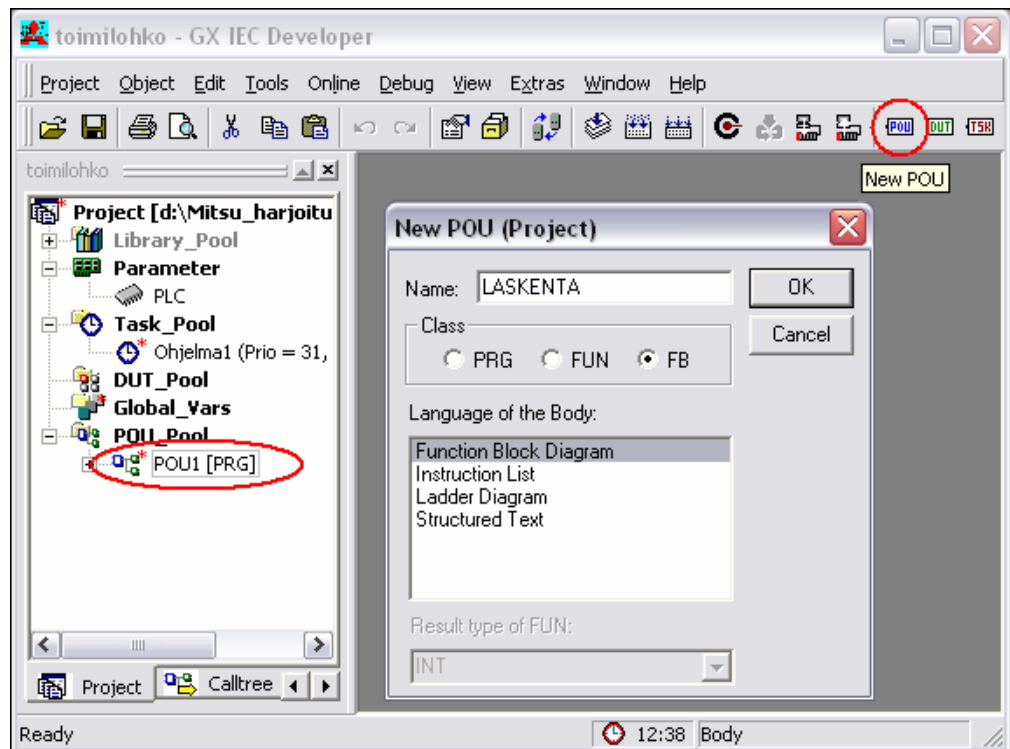
Program Organization Unit (POU) on GX IEC Developer -ohjelman rakenneyksikkö, joka koostuu muuttujalistasta (header) sekä ohjelman rakenteesta (body). POU-rakenneyksikköä on kolmea tyyppiä. Ohjelma (PRG) on logiikkaohjelma, joka sisältää käskyt, funktiot (FUN) ja toimilohkot (FB). Funktio (FUN) on aliohjelma, jossa on käytettävissä 28 tuloa ja yksi lähtö. Toimilohko (FB) on aliohjelma kuten funktio, mutta käytettävissä on sekä 28 tuloa että 28 lähtöä. Tällaisia aliohjelmiä tarvitaan esimerkiksi projekteissa, joissa on useita samanlaisia automaatiolaitteita käytössä. Käytettäviä koodeja ei tällöin tarvitse toistaa, vaan riittää kun voidaan kutsua tiettyä toimilohkoa. /7;12;15;19/

4.9 TASK

Ohjelmassa oleva tehtävä eli TASK koostuu yhdestä tai useammasta POU-rakenneyksiköstä, jotka ovat tyypiltään ohjelmamuotoisia. Kussakin projektissa voi olla yksi tai useampi tehtävä. TASK ohjaa POU-rakenneyksiköiden suoritusajankohtaa ja suoritusjärjestystä. Toisin sanoen tehtävä huolehtii siitä, milloin ja mitkä rakenneyksiköt suoritetaan. Usean tehtävän sisältävässä projektissa tehtävien suoritus ehdot voidaan asettaa erillisessä Information-ikkunassa. /7;14;15;19/

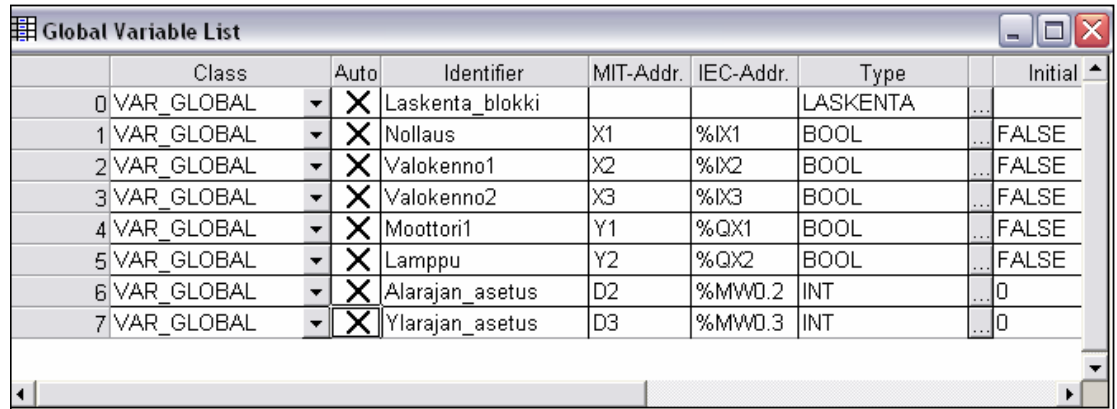
4.10 Toimilohkon luominen

GX IEC Developer FX -ohjelman sisältämiä valmiita toimilohkoja ovat esimerkiksi laskurit, ajastimet ja kiikut. Ohjelmassa voidaan kuitenkin luoda omia toimilohkoja, joita voidaan käsitellä samoin kuin edellä mainittuja valmiita toimilohkoja. Ohjeistuksessa aikaisemmin suoritettujen toimien jälkeen aktivoidaan projektinavigaattorista POU1 [PRG] ja luodaan sille uusi POU *New POU* -painikkeen avulla (kuva 25). Annetaan muodostettavalle toimilohkolle nimi, esimerkiksi LASKENTA, ja luokitellaan POU toimilohkoksi rastittamalla kohta FB Class-kentästä. Hyväksytään tämä uusi rakenneyksikkö OK-painikkeella, minkä jälkeen muodostettu toimilohko ilmestyy projektinavigaattoriin (LASKENTA [FB]). /12;30/



Kuva 25 LASKENTA-toimilohkon nimeäminen

Seuraavaksi listataan muuttujat globaalilistaan (kuva 26). Globaalilistaan merkitään muiden muuttujien lisäksi aiemmin luotu toimilohko, esimerkiksi nimellä Laskenta_blokki. Tälle ei tarvitse kirjoittaa osoitetta, mutta datatyyppi pitää vaihtaa oikeaksi. Type-kohdan painikkeesta avautuvasta Type Selection -ikkunasta valitaan toimilohkojen puolelta itse luotu tyyppi (LASKENTA). Toimilohkojen tyypit tulevat näkyviin laittamalla rasti Function Blocks -kohtaan. Valmis globaalilista tallennetaan lopuksi. /30/

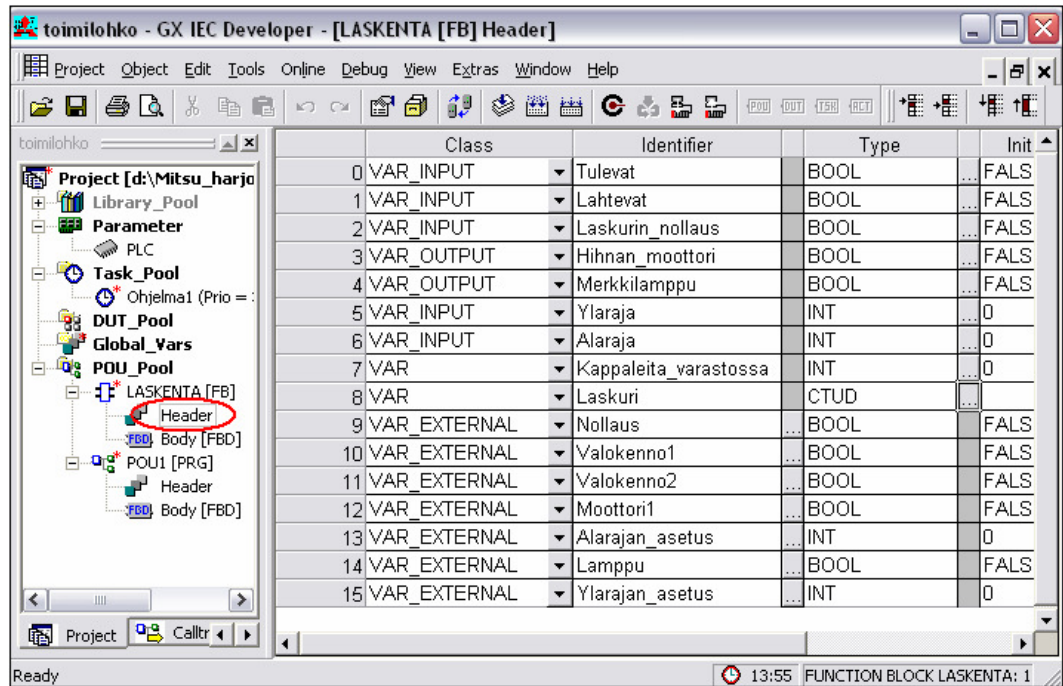


	Class	Auto	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	<input checked="" type="checkbox"/>	Laskenta_blokki			LASKENTA	...
1	VAR_GLOBAL	<input checked="" type="checkbox"/>	Nollaus	X1	%IX1	BOOL	FALSE
2	VAR_GLOBAL	<input checked="" type="checkbox"/>	Valokenno1	X2	%IX2	BOOL	FALSE
3	VAR_GLOBAL	<input checked="" type="checkbox"/>	Valokenno2	X3	%IX3	BOOL	FALSE
4	VAR_GLOBAL	<input checked="" type="checkbox"/>	Moottori1	Y1	%QX1	BOOL	FALSE
5	VAR_GLOBAL	<input checked="" type="checkbox"/>	Lamppu	Y2	%QX2	BOOL	FALSE
6	VAR_GLOBAL	<input checked="" type="checkbox"/>	Alarajan_asetus	D2	%MWD.2	INT	0
7	VAR_GLOBAL	<input checked="" type="checkbox"/>	Ylarajan_asetus	D3	%MWD.3	INT	0

Kuva 26 Muuttujien merkitseminen globaalilistaan

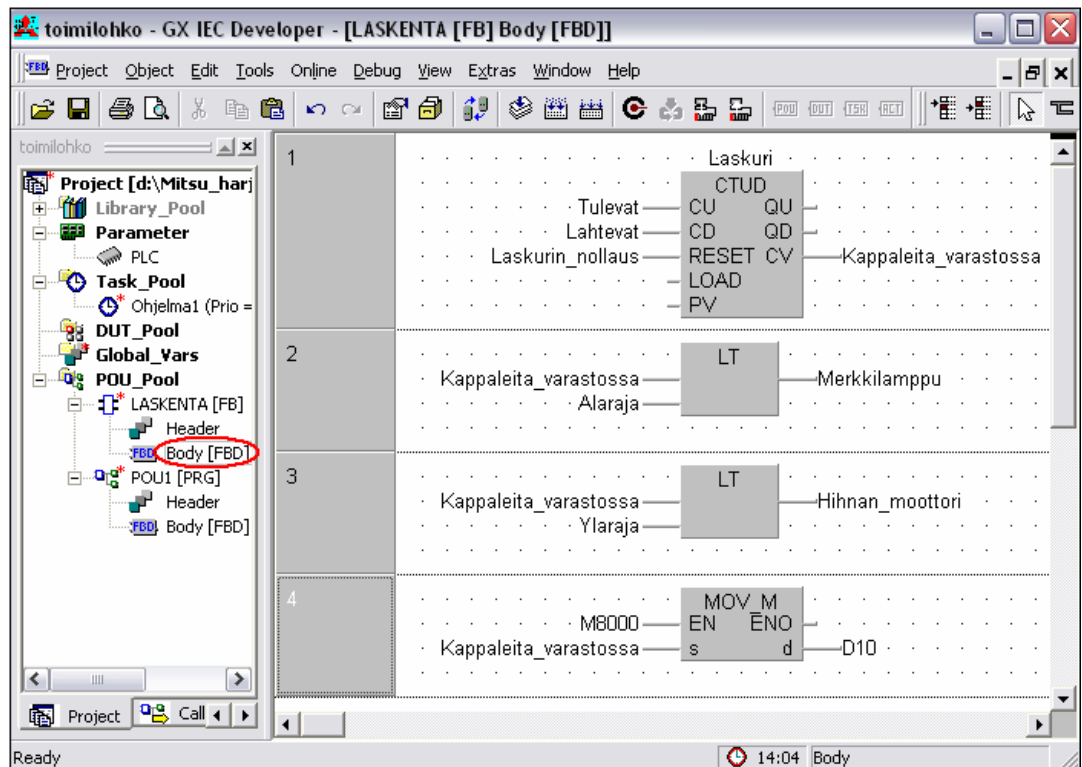
Projektinavigaattoriin ilmestynyt uusi toimilohko (LASKENTA [FB]) voidaan avata samoin kuin POU1 (kuva 27). Sen alta löytyvät myös paikalliset muuttujat sisältävä Header sekä ohjelman rakenne eli Body [FBD]. Avataan seuraavaksi tämä toimilohkon Header, jossa on jo entuudestaan globaalilistaan merkityt muuttujat. Header-listaan tarvitsee kuitenkin lisätä toimilohkon ohjelmaan tulevat muuttujat. Kuvan 27 esimerkissä tällaisia paikallisia muuttujia löytyy riveiltä 0-8. Kyseisille muuttujille on annettu nimitys sekä oikeanlainen datatyyppi. /15;30/

Header-listaan tarvitsee lisäksi määritellä kullekin muuttujalle oikeanlainen luokitus (Class). Jos kyseessä on tulo, luokituksiksi valitaan VAR_INPUT. Lähtöjen luokitus on vastaavasti VAR_OUTPUT. Valittavana on myös VAR, joka on tarkoitettu paikallisille muuttujille lohkon sisällä sekä VAR_IN_OUT, jota tarvitaan, kun muuttujana on sekä tulo että lähtö yhtä aikaa. Ohjelma tallennetaan valmiin Header-listan jälkeen. /15;30/



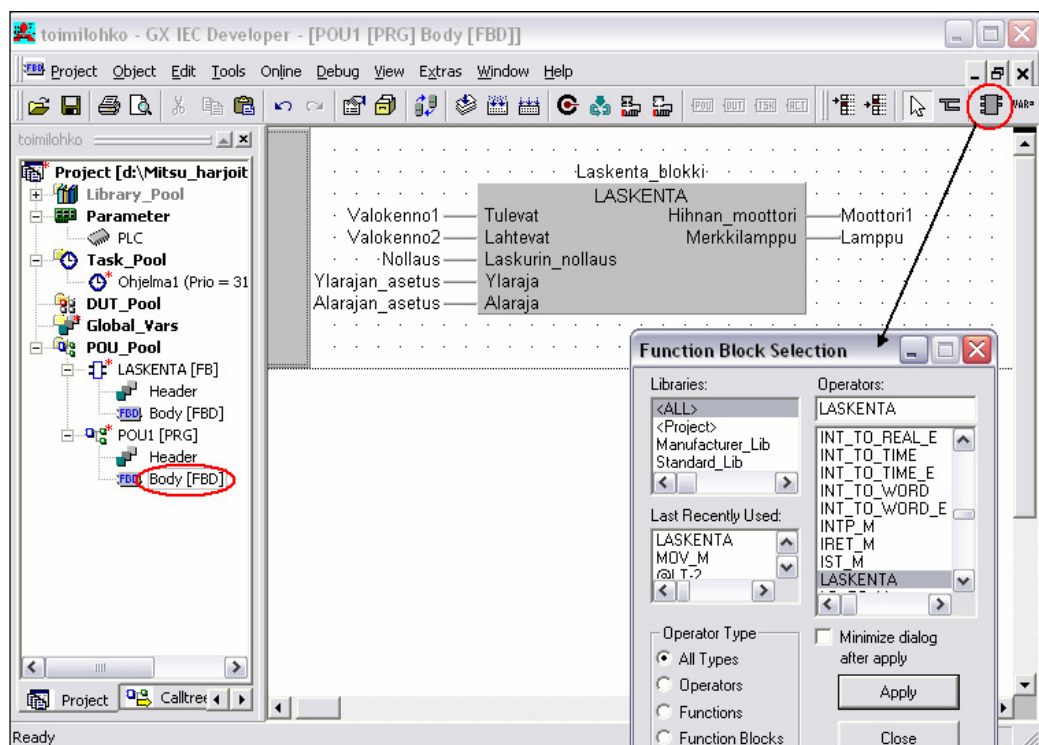
Kuva 27 Toimilohkon paikalliset muuttujat

Tämän jälkeen voidaan laatia ohjelma toimilohkolle (kuva 28). Kaksoisnäpätystä projektinavigaattorista toimilohkon Body [FBD] -kohtaa, johon haluttu aliohjelma voidaan luoda. /30/



Kuva 28 LASKENTA-toimilohkon ohjelmarakenne

Haetaan seuraavaksi POU1-rakenneyksikön virtapiiriin aiemmin luotu toimilohko (kuva 29). Avataan ensin POU1-ohjelman rakenne (Body [FBD]). Itse luotu toimilohko (LASKENTA) saadaan haettua muiden toimilohkojen ja funktioiden tavoin Function Block Selection -ikkunasta Apply-painikkeella. Jotta toimilohko ja tarvittavat muuttujat löytyisivät valintaikkunoista, tarvittavien tallennusten tulee olla tehtyinä. Kuvan 29 esimerkissä virtapiiriin 1 on haettu LASKENTA-toimilohko. Siihen käytetyt muuttujat kirjattiin aiemmin globaaliin listaan. /30/



Kuva 29 LASKENTA-toimilohkon hakeminen POU1-rakenneyksikköön

4.11 Luvut

4.11.1 BCD-luvut

Binary Coded Decimal (BCD) eli binäärikoodattu desimaalilukuesitys on automaatiojärjestelmän kannalta tärkeä lukuesitystapa. Se ei ole varsinainen lukujärjestelmä vaan systemi, jossa desimaaliluku esitetään neljän bitin koodina. Vaikka BCD-lukujen käyttö on automaatiojärjestelmissä hyvin suosittua, on sillä käyttäjän kannalta haittapuoli. Monesti BCD-luvuista tulee liian pitkiä, sillä kymmenjärjestelmän moninumeroinen luku esitetään kunkin numeron BCD-

koodina muodostuneena jonona. Esimerkiksi desimaaliluku 25 on BCD-lukuna 0010 0101, taulukon 8 mukaisesti. /6;20;21/

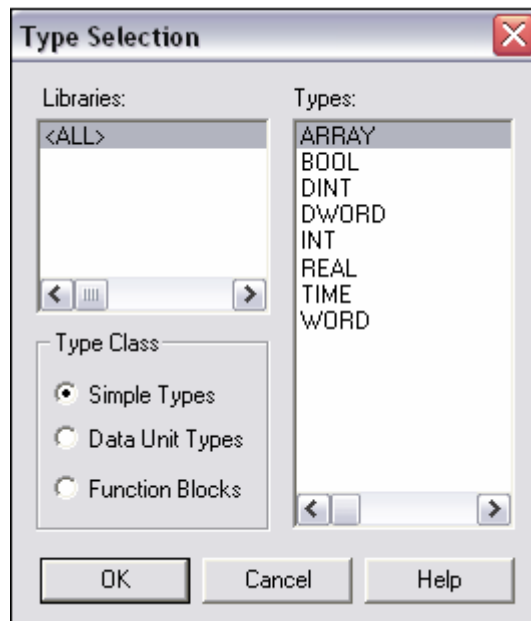
Taulukko 8 BCD-luvun muodostaminen /6/

Desimaaliluku	BCD-luku
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

4.11.2 Datatyypit

GX IEC Developer -ohjelmassa käytettävä data on jaoteltu erilaisiin datatyyppeihin suuruutensa ja bittirakenteensa mukaisesti. Nämä datatyypit perustuvat IEC 1131-3 -standardiin, jossa ne on määritelty bitti-, kokonaisluku-, reaalityyppi- tai aikatyypeiksi. Globaalilistasta aukeavasta Type Selection -ikkunasta (kuva 30) on valittavissa 8 eri datatyypiluokkaa. Boolista arvoa (BOOL) käytetään tuloilla (X), lähdöillä (Y) sekä muistipaikoilla (M ja B). BOOL-datatyypin muodostuu vain yhdestä bitistä, joka voi olla joko tosi (1) tai epätosi (0). Kokonaisluvulla on käytössään etumerkillinen 16-bitin datarekisteri INT tai 32 bitin datarekisteri DINT. Koska INT on suuruudeltaan 16 bittiä, kokonaisluku tallentuu muistissa aina yhteen sanaan. REAL eli reaalityyppi on 32 bitin suuruisen desimaaliluvun datarekisteri. Reaalityyppi varaa siis muistista kaksoissanan, jossa korkea-arvoisin bitti määrää luvun etumerkin. /6;12;15/

Datatyypiksi voidaan valita edellisten lisäksi WORD, eli 16 bitin datatasana ilman merkkibittiä, sekä DWORD, joka on 32 bitin tuplasana. Datatyypin ARRAY on tarkoitettu käytettäväksi taulukoihin, aina kolmeen dimensioon asti. Lisää datatyyppejä, esimerkiksi ajastimille ja laskureille, löytyy rastittamalla Function Blocks -kohta (kuva 30). /12;15/



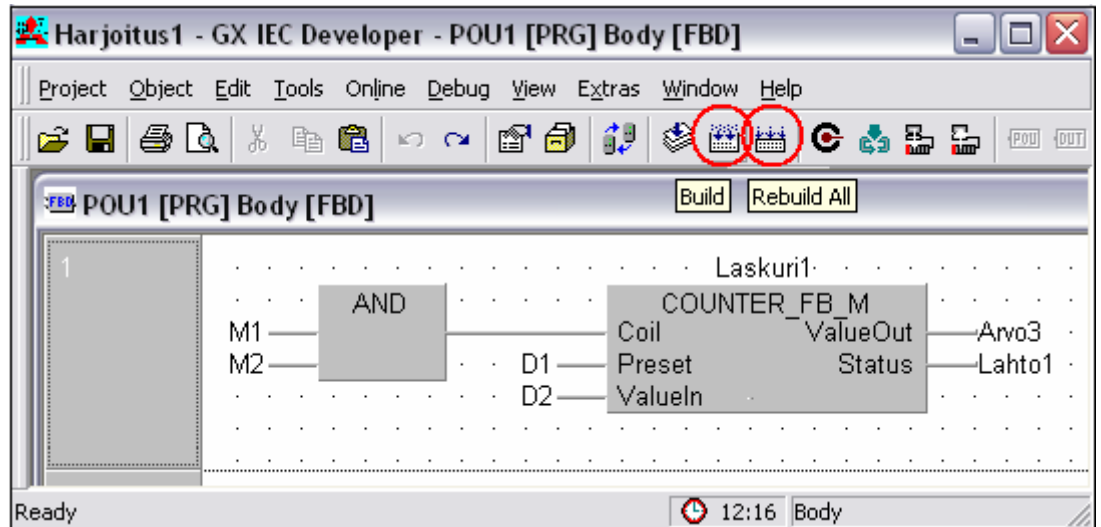
Kuva 30 Datatyyppiluokat

4.12 Simulointi

Ohjelman testaus ilman logiikkaa voidaan suorittaa offline-simulaattorin avulla. Ensiksi projekti kuitenkin tallennetaan, käännetään ja tarkistetaan mahdolliset virheet.

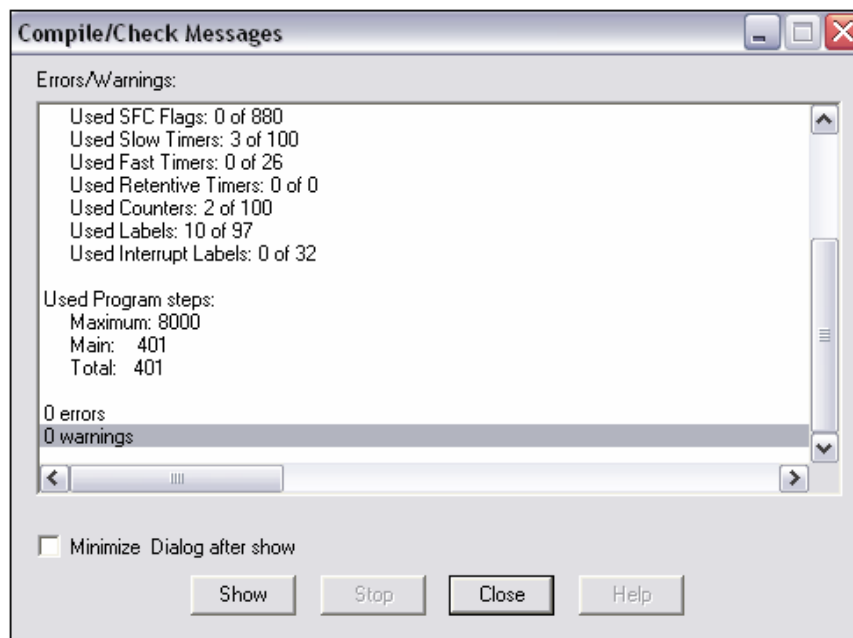
4.12.1 Projektin kääntäminen

Projektin kääntäminen suoritetaan Build- ja Rebuild All -painikkeiden avulla (kuva 31). *Build*-painikkeen avulla voidaan kääntää vain muuttuneet projektinosat, kun koko projektin kääntäminen voidaan suorittaa *Rebuild All* -painikkeen avulla. Jotta kääntäminen olisi mahdollista, projektin pitää sisältää vähintään yhden tehtävän (TASK), ja yhden rakenneyksikön (POU). Kääntämisen jälkeen ohjelmakoodi voidaan ladata logiikalle. /7;14;15;19/



Kuva 31 Projektin kääntäminen

Kuvan 31 painikkeet avaavat Compile/Check Messages -ikkunan (kuva 32), joka näyttää projektissa ilmenneet virheet ja varoitukset. Jos virheitä ei ole, ikkunan voi sulkea Close-painikkeella. Mahdollisia virheitä voi tarkastella näpäyttämällä virhe aktiiviseksi ikkunasta ja painamalla Show-painiketta. Tällöin ohjelma näyttää virheen sijainnin. /7/

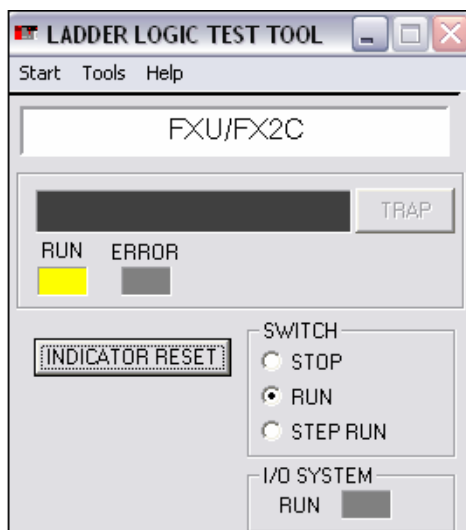


Kuva 32 Virheiden ja varoitusten ilmoitusikkuna

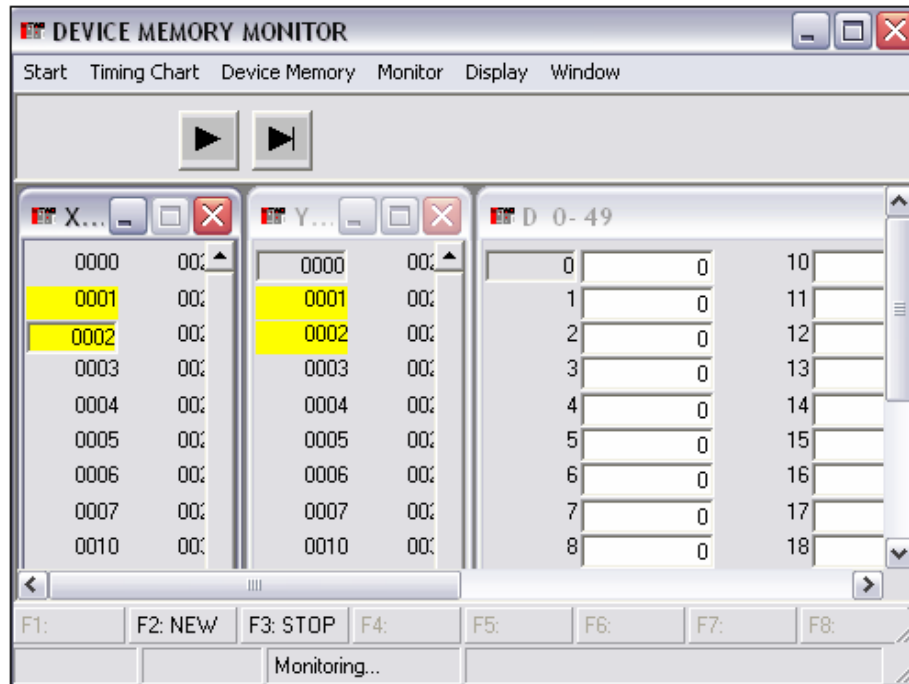
4.12.2 Offline-simulaattori

Offline-simulaattorin avulla voidaan ohjata projektissa esiintyviä muuttujia sekä testata niiden toimintaa. Muuttujien lisäksi offline-simulaattorin avulla voidaan tutkia todellisuutta vastaavia askelvasteita, kerätä signaaleja, tallentaa saatuja tuloksia sekä simuloida samanaikaisesti usean eri signaalin tilamuutoksia.

Simulaattori käynnistyy valitsemalla ylävalikosta *Online* → *GX Simulator*, minkä jälkeen ohjelma avaa Ladder Logic Test Tool -ikkunan (LTT) (kuva 33). Valitse LTT-käyttöliittymän ylävalikosta *Start* → *Device Memory Monitor*, jonka avulla voidaan esimerkiksi ohjata tuloja ja lähtöjä sekä asettaa rekisterien arvoja. Device Memory Monitor -ikkunasta (kuva 34) valitaan *Device Memory* → *Bit Device* → *X*, jolloin käyttöliittymään avautuu painikkeet tuloille. Painettaessa painikkeita ohjelmaan kirjoitettujen tulo-osoitteiden mukaisista kohdista kyseiset tulot kytkeytyvät päälle. Nämä päällekytketyt tulot ilmenevät simulointitilassa keltaisina laatikkoina. Painikkeet lähdöille saadaan samoin, mutta tulon (X) tilalle valitaan lähtö (Y). Rekisteriarvojen asettaminen tapahtuu valitsemalla *Device Memory* → *Word Device* → *D*, jolloin halutut arvot voidaan syöttää avautuvaan ikkunaan. Simulointitila saadaan suljettua painamalla uudelleen *Online* → *GX Simulator*. /14/



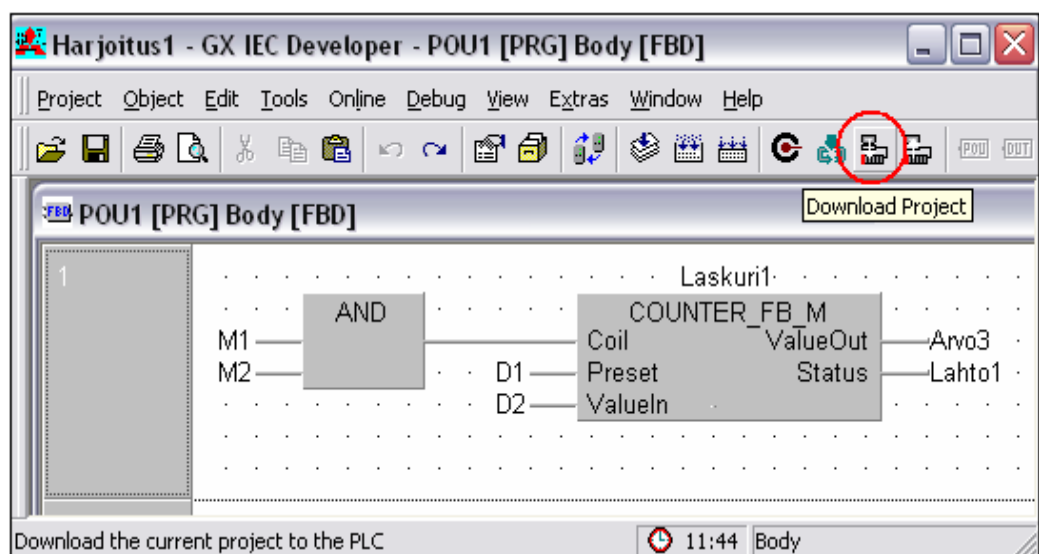
Kuva 33 LTT-käyttöliittymä



Kuva 34 Device Memory Monitor -ikkuna ja haetut painikkeet

4.13 Ohjelman lataaminen logiikalle

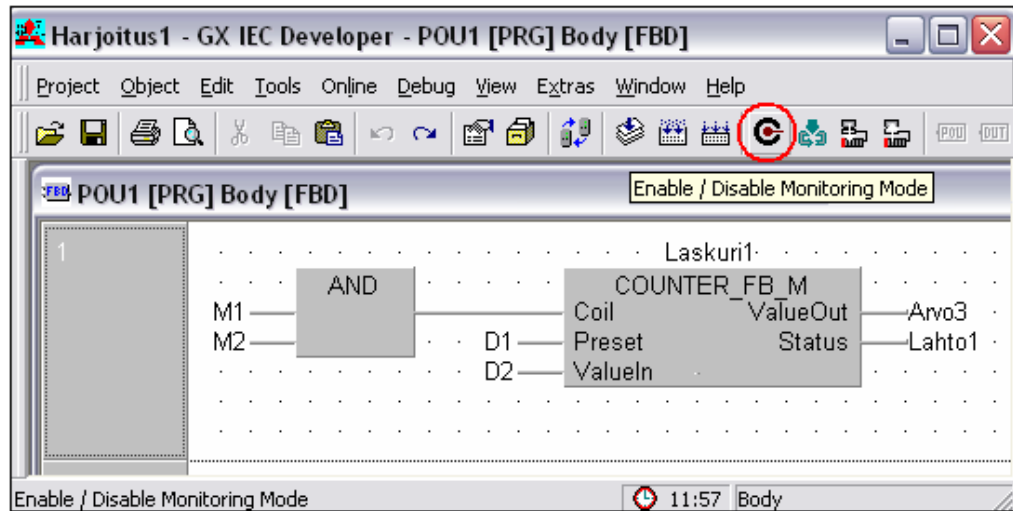
Ohjelman lataaminen logiikalle tapahtuu *Download Project* -painikkeen avulla (kuva 35) tai valitsemalla *Project* → *Transfer* → *Download to PLC*. Tällöin logiikalle siirtyvät sekä ohjelma että parametrit standardiasetusten mukaisesti. /15/



Kuva 35 Ohjelman siirto logiikalle

4.14 Ohjelman monitorointi

Ohjelma voidaan asettaa monitoritilaan *Enable/Disable Monitoring Mode* -painikkeen avulla (kuva 36). Tilan voi asettaa myös valitsemalla ylävalikosta *Online* → *Monitoring Mode*. Jos halutaan useampia ikkunoita samanaikaisesti monitoritilaan, valitse *Online* → *Start Monitoring* → *Online*. Huomaa, että monitoritilan ollessa päällä ohjelmamuutokset eivät ole mahdollisia. /15/



Kuva 36 Ohjelman monitorointi

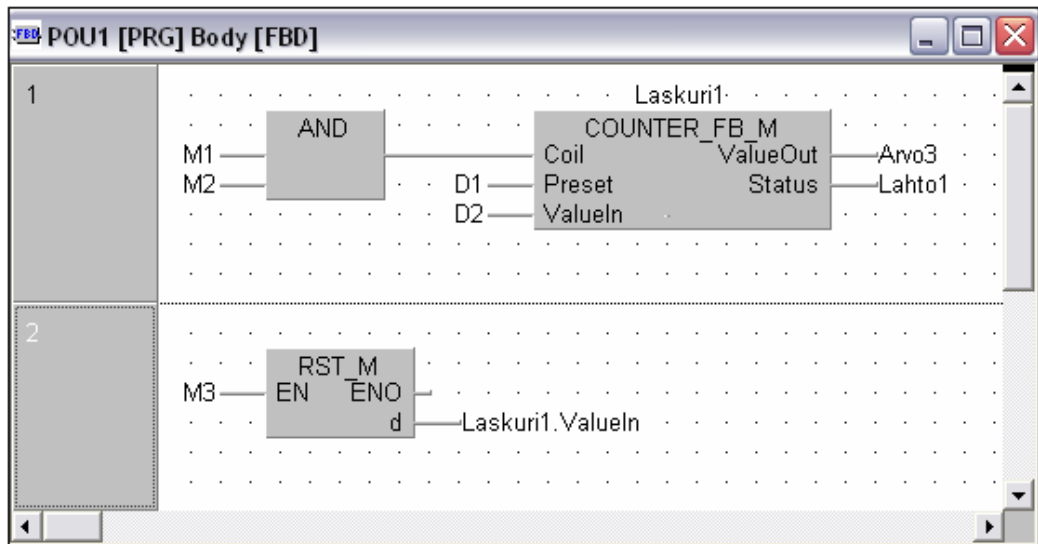
4.15 E-Designer

14.15.1 E-Designer -ohjelma

Kun automaatio-ohjelma on luotu GX IEC Developer FX -ohjelman avulla, sitä voidaan käyttää ja testata opetuslaitesalkun avulla. Opetuslaitesalkun Beijer E300 -operointipäätte tarvitsee kuitenkin E-Designer-ohjelman, jonka avulla päätettä voidaan ohjelmoida. Ohjelmalla voidaan luoda esimerkiksi painikkeita, vipuja ja kytkimiä eri toiminnoille sekä luoda numeronäyttöjä ilmaisemaan ohjelmassa esiintyviä arvoja. Ohjelmaa voidaan käyttää myös tulojen ja lähtöjen tarkastelussa sekä LED-valojen ohjauksessa. Nämä E-Designer-ohjelmassa luodut toiminnot siirretään Beijer E300 -operointipäätteelle, jossa niitä voidaan ohjata toiminto- ja nuolinäppäinten sekä Enter-painikkeen avulla. /8;10;11/

Koska opetuslaitesalkun operointipäätteeltä logiikalle kulkeva virta on heikko, logiikka ei välttämättä reagoi päätteellä tehtyihin toimintoihin. Tämän vuoksi GX IEC Developer FX -ohjelmassa tehtyyn projektiin kannattaa muuttaa kaik-

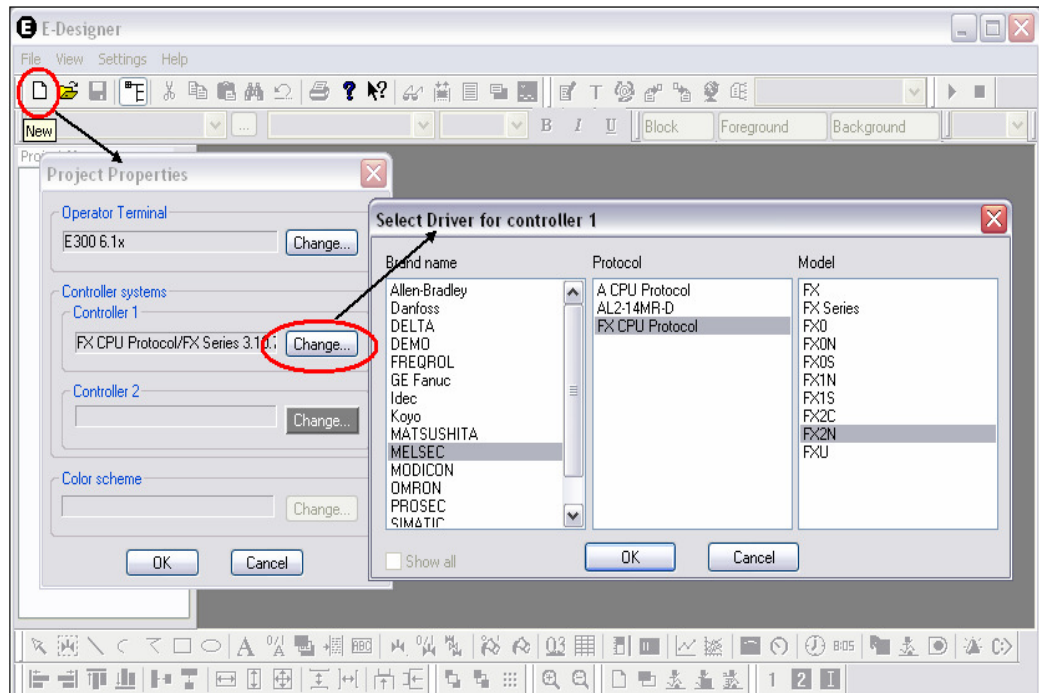
kien tulojen tilalle merkkereit. Esimerkiksi tulo 1 voi olla M1, tulo 2 voi olla vastaavasti M2 (kuva 37). Myös ohjelmassa esiintyvät arvot kannattaa muuttaa D-rekisteriarvoiksi, joita voidaan E-designer-ohjelmassa käyttää. Merkkereitä ja D-rekisteriarvoja ei tarvitse kirjoittaa globaalilistaan, vaan ne voidaan kirjoittaa suoraan ohjelmaan halutulle paikalle. /10/



Kuva 37 Merkkereitä käytetty tulojen tilalla

4.15.2 Ohjelman käynnistys ja esiasetukset

E-Designer-ohjelma avataan painamalla *Käynnistä* → *Ohjelmat* → *E-Designer 7.01* → *E-Designer7* tai painamalla suoraan *E-Designer 7* -pikakuvaketta. Ohjelman auettua aloitetaan uusi projekti valitsemalla ylävalikosta *File* → *New project* tai painamalla suoraan *New*-pikapainikkeesta (kuva 38). /5;11/

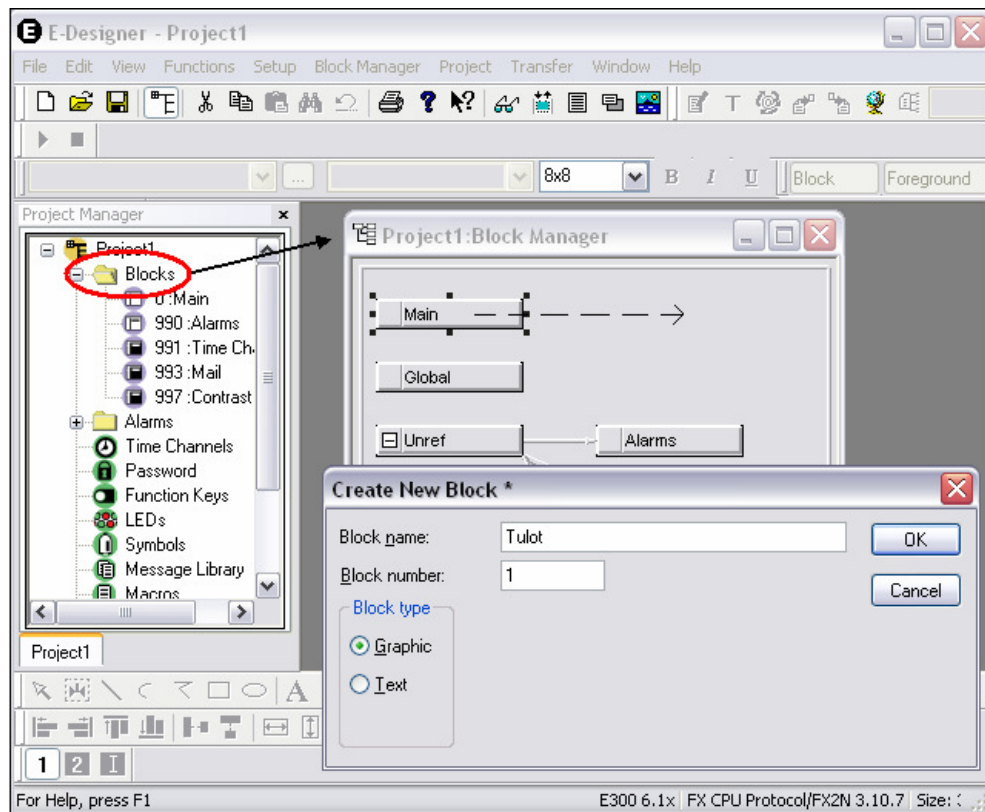


Kuva 38 Uuden projektin luominen sekä logiikan valitseminen

Ohjelmaan aukeaa Project Properties -ikkuna, johon syötetään halutut asetusarvot. Controller 1 vaihdetaan Change-painikkeen avulla, esimerkiksi *Melsec* → *FX CPU Protocol* → *FX2N*. Tarkista logiikkatietojen lisäksi, että Operator Terminal -kentässä lukee E300 6.1x. Kun esiasetukset on tehty, ne hyväksytään OK-painikkeella. /5;11/

4.15.3 Sivujen luominen

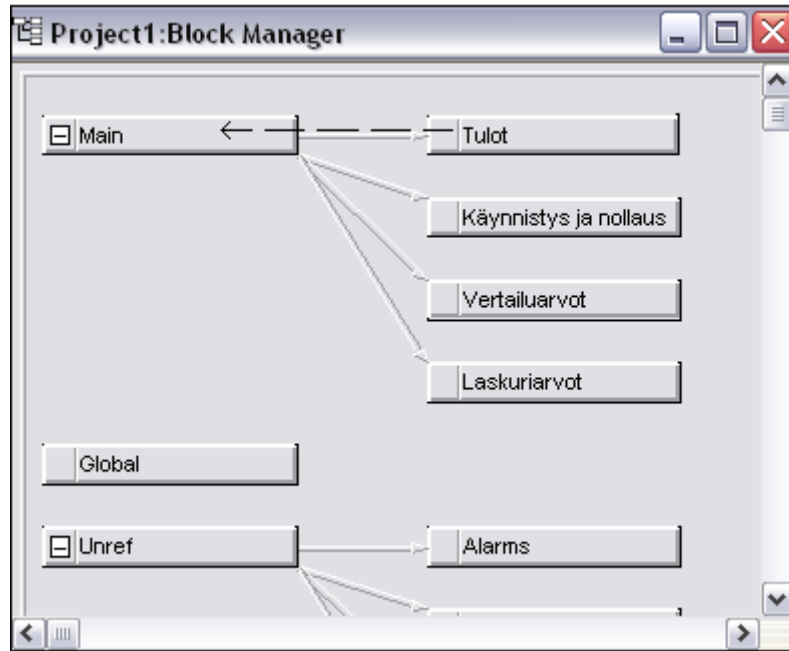
Projektiin lisätään ohjaussivuja avaamalla Blocks-kansio ohjelman projektinavigaattorista. Tällöin ohjelma avaa Project1:Block Manager -ikkunan (kuva 39). Luodaan uusi sivu vetämällä Main-palkista hiirellä oikealle, painamalla koko ajan hiiren vasenta painiketta. Hiiren painike voidaan vapauttaa, kun ohjelma on piirtänyt riittävän pituisen nuolen oikealle. Tämän jälkeen näytölle ilmestyy välittömästi Create New Block -ikkuna, jossa Block Name -kenttään annetaan sivulle nimi, esimerkiksi Tulot. Nimeäminen hyväksytään OK-painikkeella. /5;11/



Kuva 39 Tulot-sivun luominen ja nimeäminen

Tämän jälkeen näytölle ilmestyy Select Local Function Key -ikkuna, josta Tulot-sivulle valitaan toimintopainikkeeksi F2. Toiminto hyväksytään OK-painikkeella. Samalla projekti avaa graafisen operointipäätteen tuloille (Project1:Graphic 1-Tulot), jonka voi pienentää tai sulkea hetkeksi. Uudelleen sen saa näkyviin kaksoisklikkaamalla muodostunutta Tulot-palkkia. /5;11/

Seuraavaksi Tulot-palkista vedetään nuoli takaisin Main-palkille (kuva 40) samalla tavalla hiiren vasenta painiketta painaen. Select Local Function Key -ikkunaan valitaan toimintopainikkeeksi puolestaan F8, joka toimii paluupainikkeena Tulot-sivulta Main-sivulle. Hyväksytään tämä OK-painikkeella. /5;11/



Kuva 40 Paluupainikkeen muodostaminen ja Main-sivun alasisivut

Luodaan samalla tavalla Main-sivulle esimerkiksi seuraavanlaiset alasisivut: käynnistys ja nollaus, vertailuarvot ja laskuriarvot (kuva 40). Kunkin sivun toimintopainikkeeksi on valittu eri painike (F3-F5). Kaikilta sivuilta paluu Main-sivulle tapahtuu kuitenkin F8-painikkeella. /5;11/

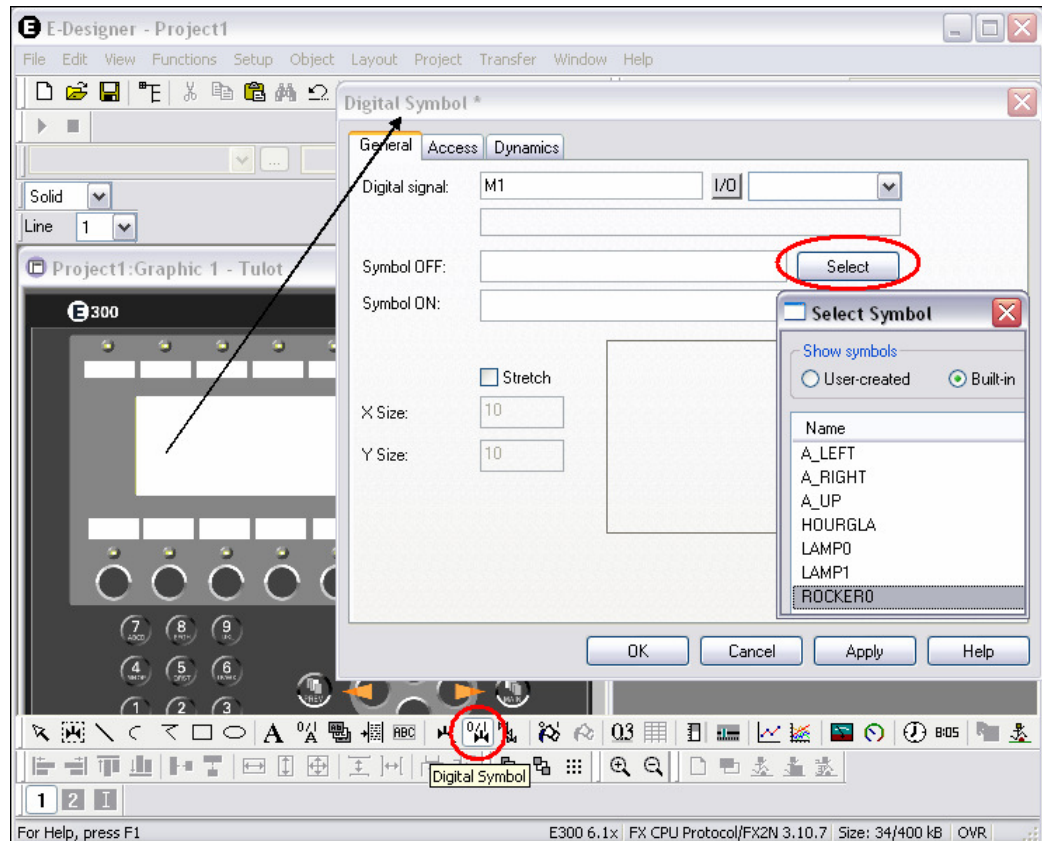
4.15.4 Digitaalisymbolit

Painikkeet ja kytkimet

Painikkeita, kytkimiä ja vipuja tarvitaan muuttamaan ohjelmassa olevan muuttujan tila päälle tai pois päältä. Jotta ohjelmaa voidaan käyttää opetuslaitesalkun avulla, täytyy jokaiselle GX IEC Developer FX -ohjelmassa esiintyvälle päällekytkettävälle tulolle luoda oma painikkeensa. Kuvan 37 esimerkissä tällaisia tuloja ovat M1, M2 ja M3.

Avataan siis Tulot-sivun aiemmin luotu graafinen operointipääte, johon lisätään painikesymbolit *Digital Symbol* -pikapainikkeella (kuva 41). Paina *Digital Symbol* -painiketta kerran ja vie hiiren kursori graafisen operointipäätteen valkoiselle keskialueelle. Paina kerran hiiren vasenta painiketta alueella, jolloin näyttöön avautuu *Digital Symbol* -ikkuna. Kirjoita *Digital signal* -kenttään GX IEC Developer -ohjelmassa käytetty tulo. Kuvan 37 tapauksessa kenttään voidaan

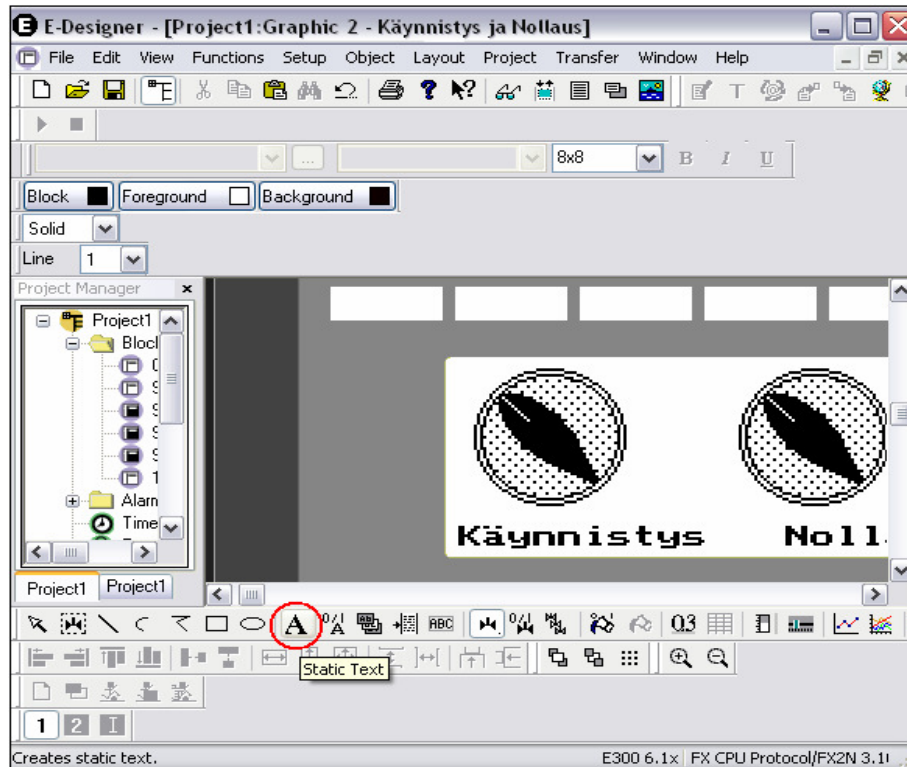
kirjoittaa M1, M2 tai M3 sen mukaan, mitä tuloa kyseisellä painikkeella halutaan ohjata. Valitaan seuraavaksi tulolle painiketyyppi ikkunassa olevan Select-painikkeen avulla. Symbol OFF -painikkeeksi voidaan valita esimerkiksi ROCKER 0 ja Symbol ON -painikkeeksi vastaavasti ROCKER 1. /4;5;10;11/



Kuva 41 Ohjauspainikkeiden valinta

Ennen hyväksymistä painikesymboli vahvistetaan toimivaksi laittamalla rasti Enable Operator Input -kohtaan Access-välilehdellä. Vasta tämän jälkeen painike hyväksytään painamalla ensin Apply ja tämän jälkeen OK. Aluksi painike näkyy graafisella ohjauspaneelilla pisteinä, mutta tulee näkyviin kun sitä liikuttaa. /5;11/

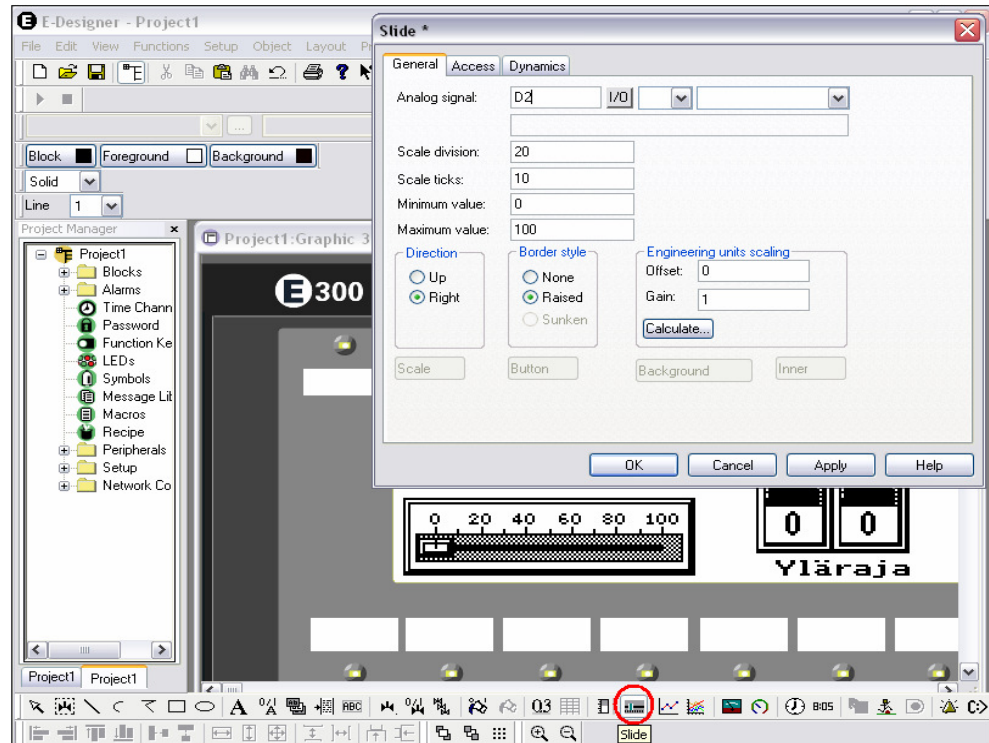
Digital Symbol -ikkunassa on valittavana monenlaisia symboleja, joista jokaisesta on kuva valintaa helpottamassa. Esimerkiksi ROT-kytkin on erinomainen kuvaamaan ohjelman käynnistys- ja nollaustoimintoja. Yleensä ohjelma sisältääkin useita digitaalisymboleja. Jotta jokaisen symbolin toiminnan muistaisi, voidaan graafiselle operointipäätteelle kirjoittaa tekstiä *Static text* -kirjoitustyökalulla (kuva 42). Kuvan esimerkissä vasemmanpuoleinen kytkin käynnistää ohjelman ja oikeanpuoleinen kytkin nolaa ohjelman. /4;5;10;11/



Kuva 42 ROT-kytkimien nimeäminen tekstityökalun avulla

Liukukytkimet

Liukukytkintä käytetään hyvin paljon esimerkiksi vertailuarvoilla, sillä siihen voidaan asettaa arvoja Beijer E300 -operointipäätteen avulla. Kyseinen symboli voidaan lisätä ohjelmaan *Slide*-painikkeen avulla (kuva 43). Painiketta painetaan kerran ja hiiri tuodaan graafisen operointipäätteen valkoiselle keskialueelle painaen kerran hiiren vasenta painiketta. Tällöin näytölle avautuu *Slide*-ikkuna, jossa Analog signal -kenttään kirjoitetaan GX IEC Developer -ohjelmassa käytettävä D-rekisteriarvo. Minimum value -kenttään kirjoitetaan liukukytkimen minimiarvo ja maksimiarvo kirjoitetaan Maximum value -kenttään. Nämä arvot kuvaavat kytkimen ala- ja yläarvoja. Scale division-kenttä kuvaa asteikkoväliä, joka voi olla esimerkiksi 20. Asteikon suunta (Direction) voidaan valita oikealle kasvavaksi (Right) ja asteikko voi olla tyypiltään (Border Style) nouseva (Raised). Liukukytkin vahvistetaan toimivaksi laittamalla rasti Enable Operator Input -kohtaan Access-välilehdellä. Kytkin hyväksytään painamalla ensin Apply ja tämän jälkeen OK. /4;5;10;11/

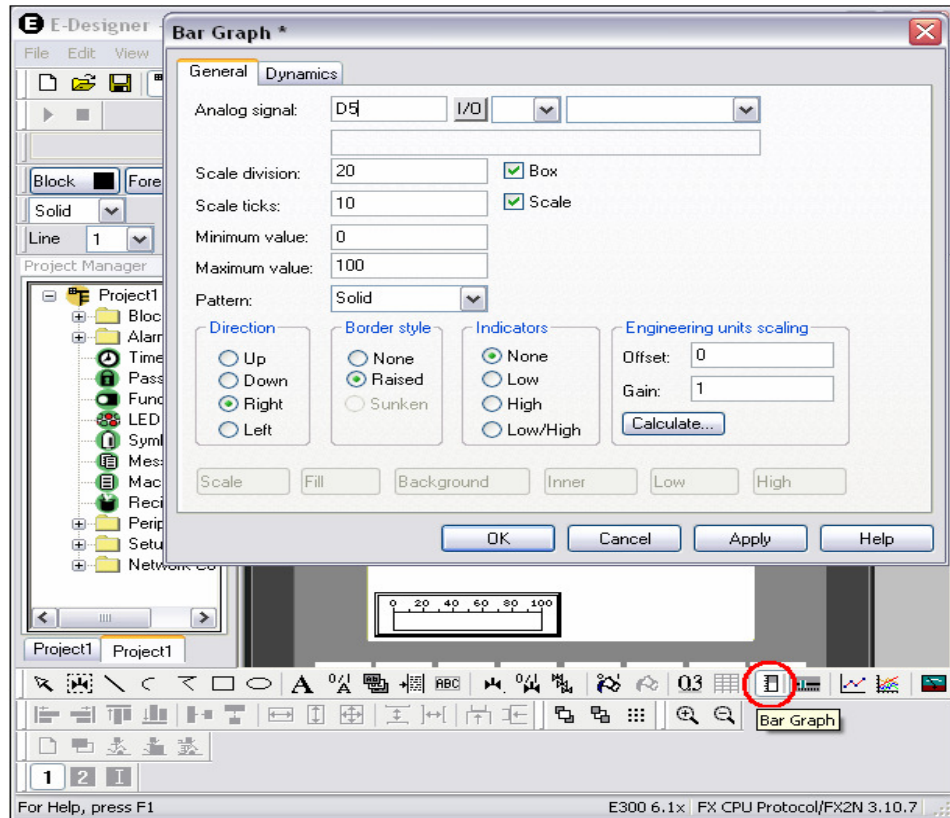


Kuva 43 Liukukytkimen lisääminen

Numeronäyttö

Kuvan 40 laskuriarvot-sivulle voidaan luoda numeronäyttöjä, jotka näyttävät ohjelmassa käytettävien laskureiden arvoja. Numeronäyttö siis näyttää kunkin laskurin laskeman arvon Beijer E300 -operointipäätteellä.

Numeronäyttö luodaan *Bar Graph* -painikkeella (kuva 44). Painiketta painetaan kerran, ja hiiri tuodaan graafisen operointipäätteen valkoiselle keskialueelle painaen kerran hiiren vasenta painiketta. Tällöin näytölle avautuu Bar Graph -ikkuna, jossa Analog signal -kenttään kirjoitetaan GX IEC Developer -ohjelmassa käytettävä D-rekisteriarvo. Kuvan 37 esimerkissä tämä voisi olla D1 tai D2. Numeroasteikon skaalaus voidaan muodostaa esimerkiksi välille 0-100 (Minimum ja Maximum value) ja asteikkoväliksi (Scale division) valitaan esimerkiksi 20. Asteikon suunnaksi (Direction) voidaan valita oikealle kasvava (Right) ja asteikkotyyppi (Border style) nouseva (Raised). Valinnat vahvistetaan painamalla ensin Apply ja tämän jälkeen OK. /5;10;11/



Kuva 44 Numeroasteikon hakeminen

Muita toimintoja

E-Designer-ohjelma sisältää lukuisia graafisia, digitaalisia ja analogisia työkaluja, joista osa on koottuna liitteeseen 3. Lisää symboleista ja niiden toiminnosta löytyy E-Designer-ohjelmamanuaalista. /4;5/

4.15.5 Ohjelman lataaminen operointipäätteelle

Tarkista ennen E-Designer-ohjelman siirtämistä Beijer E300 -operointipäätteelle, että GX IEC Developer FX -ohjelma muutoksineen on ladattu logiikalle. E-Designer-ohjelman siirto operointipäätteelle tapahtuu valitsemalla ylävalikosta *Transfer* → *Project*, minkä jälkeen ohjelma avaa Project Transfer -ikkunan. Ohjelma siirretään Send-painikkeen avulla. /4;5;10/

Ohjelman latauduttua Beijer E300 -operointipäätteellä voidaan tarkastella E-Designer-ohjelmalla luotuja sivuja eri toimintonäppäinten avulla. Sivuilta paluu tapahtuu toimintonäppäimen F8 avulla. Painikkeiden päälle- ja pois-painaminen

tapahtuu nuolinäppäinten ja Enter-painikkeen avulla. Samoilla painikkeilla asetetaan myös liukukytkimien arvot. Tuloksia voidaan katsella joko suoraan toimilaitteilla tai GX IEC Developer -ohjelmaan luodusta projektista monitoritilan ollessa päällä.

5 TULOKSET

Tämän tutkintotyön tuloksena on kappaleessa 4 esitelty ohjekirja, jonka nimi on FX2N-logiikan ohjelmointi GX IEC Developer FX -ohjelmalla. Ohjekirja on 51-sivuinen ja sisältää lisäksi 27 sivua liitteitä. Ohjekirja sisältää pääasiassa käyttöohjeet kahdelle automaatio-ohjelmalle sekä antaa tietoa niiden soveltamisesta käytännössä. Se antaa lukijalleen myös jonkin verran perustietoa ohjelmoitavista logiikoista sekä muista automaatioon liittyvistä aiheista.

Käytettyjä liitteitä ovat erikoisapumuistitaulukko, virhekooditaulukko sekä taulukko E-Designer-ohjelman graafisista symboleista. Erikoisapumuistitaulukkoon on koottu joukko tärkeimpiä erikoisapumuisteja sekä niiden kuvaukset. Virhekooditaulukko sisältää puolestaan GX IEC Developer FX -ohjelmassa ilmaantuvia virheenilmoituskoodeja sekä virheiden korjausehdotuksia. Ohjekirja sisältää liitteenä myös harjoitustehtäviä (liite 4), jotka on tarkoitettu suoritettavaksi opetuksen yhteydessä. Näitä harjoitustehtäviä on yhteensä 10 kappaletta, ja ne ovat kukin vaikeusasteeltaan erilaisia. Koska ohjekirja sisältää paljon lyhenteitä ja IEC-standardin mukaisia kirjainyhdistelmiä, niistä osa on selitetty ohjekirjan viimeisessä liitteessä.

6 TULOSTEN TARKASTELU

Käyttökelpoisuutensa kannalta tarkasteltuna ohjekirja sopii tehtäväänsä. Se on selkeä, tiivis ja helposti ymmärrettävä kokonaisuus aiheestaan. Vaikeita asioita on kuvattu esimerkkien ja kuvien avulla, mikä edesauttaa uuden asian oppimista. Mahdolliset virheet ja puutteet on pyritty minimoimaan kokoamalla käyttöohjeet vaihe vaiheelta ohjelmankäytön etenemisen aikana. Ohjekirjan käyttöohjeiden toimivuus on myös testattu opiskelijoiden koekäytössä.

Ohjekirjan kirjoittaminen onnistui suurelta osin ilman ongelmia. Tekstimäärän kasvaessa ohjeistuksen kokonaisuuden hahmottaminen ja ratkaisujen tekeminen helpottuivat. Muutamaa poikkeusta lukuunottamatta ohjekirjan asiakohdat valmistuivat yhdellä kertaa. Eniten vaikeuksia aiheutti operaatiot-osio, jonka laatiminen keskeytyi tiedonpuutteen vuoksi. Kyseinen kappale olikin viimeisiä valmistuvia asiakokonaisuuksia.

Ohjekirja on täyttänyt sille alussa asetetut tavoitteet. Se sisältää perustietoa vasta-alkajille sekä täydentää aiemmin opittuja tietoja. Koska ohjeistus on tiivis tietopaketti aiheestaan, se edellyttää käyttäjältään ohjelmoitavien logiikoiden perustuntemusta. Alun perin tarkoituksena oli tehdä ohjekirjasta 40-sivuinen, mutta tässä tapauksessa se olisi jäänyt puutteelliseksi. Valmistunut ohjekirja sisältää 10 sivua tavoiteltua sivumäärää enemmän, mutta on aiheen rajauksen kannalta johdonmukaisempi kuin 40-sivuinen versio. Johdonmukainen kulku aiheesta toiseen lisää työn selkeyttä. Käsitellyt aiheet ovat siinä järjestyksessä kuin niiden oletetaan olevan hyödyllisiä ohjelman käytön kannalta. Samojen aihepiirien toistoja on ohjekirjassa pyritty välttämään. Jos sama aihe on tullut esille myöhemmissä osioissa, tekstiin on lisätty viittaukset kohtiin, joissa aihe on ensimmäistä kertaa selitty. Nämä kaikki asiat ovat vaikuttaneet osaltaan siihen, että ohjekirjaa on tarkoitus käyttää mallina tulevien oppimateriaalien laatisemisessa kone- ja laiteautomaation koulutuksessa.

Sivumäärän rajallisuuden vuoksi aivan kaikkia aihepiirejä ei ole voitu esitellä kaiken kattavasti. Esimerkiksi muuttujien listaaminen globaalilistaan on aiheena hyvin monimutkainen, ja sen perusteellinen selittäminen veisi liian suuren sivumäärän ohjekirjasta. Tampereen ammattikorkeakoululla onkin suunnitteilla

tälle oppimateriaalille jatko-osa, jossa luultavasti syvennyttään juuri tähän aiheeseen.

LÄHTEET

Painetut lähteet

- 1 Riihelä, Sakari, Julkaisuopas. Julkaisu N:O 103. Oulun yliopistopaino. Oulu 1997. 21 s.
- 2 GX IEC Developer 6.10. IEC Programming and Documentation System. Beginner's Manual. MELSOFT. Mitsubishi Electric, Industrial Automation. Berliini 2004. 79 s.
- 3 Mitsubishi Electric Europe GMBH. Mitsubishi Electric. MELSEC FX-sarja. Käyttöohje. FX/911101/SF. 04/1991. 317 s.
- 4 E-Terminals and E_Designer. Manual. Beijer Electronics AB. Mitsubishi Electric. MA00552D. 2004-05. 370 s.
- 5 E-Designer. User's Guide. Mitsubishi Electric. Mitsubishi Electric. MA00760. 2005-02. 35 s.
- 6 Mäkelä, Seppo, Siemens Automaatiokoulutus. Simatic S7 oppilaitoksille. Tampereen ammattikorkeakoulu. Koneosasto. Siemens Osakeyhtiö 2004. 142 s. + 17 liites.

Painamattomat lähteet

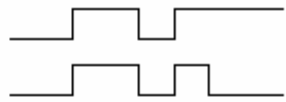



- 7 Hietanen, Hannu, Uuden sukupolven kehitysohjelma Mitsubishi-logiikkaan. Tutkintotyö. Tampereen ammattikorkeakoulu. Kone- ja laiteautomaatio. Tampere 2005. 34 s.
- 8 Martin, Mikko, Mitsubishi Melsec FX2N-64MR-logiikan ja Beijer E300-operointipäätteen opetuslaitteiston toteutus. Tutkintotyö. Tampereen ammattikorkeakoulu. Kone- ja laiteautomaatio. Tampere 2005. 24 s.
- 9 Yksiköiden ohjelmointi toimintakaaviomuodossa FBD/FUP S7-300/400. Referenssikäsikirja. Siemens Simatic. Versio 11/2002 FIN. 212 s.
- 10 Evikoski, Tero – Kotiranta, Sauli, Beijer E300 operointipäätteen ohjelmointi E-Designer 7.01 ohjelmalla. Ohjelmoitavat logiikat. Tampereen ammattikorkeakoulu. Kurssin päättötyö. 2005. 9s.
- 11 Mäkinen, Pauliina – Heinonen, Anu, E-Designer -ohjeistus. Ohjelmoitavat logiikat. Tampereen ammattikorkeakoulu. Kurssin päättötyö. 2005. 13 s.

Sähköiset lähteet

- 12 Schneider, André, GX IEC Developer-Programming. Mitsubishi Electric. Industrial Automation. Luentosarja. 135s.
- 13 Pirttioja, Teppo, IEC 61131-3 PLC ohjelmointistandardi. [www-sivu]. [viitattu 18.11.2005.] Saatavissa: <http://www.automation.hut.fi/edu/as84168/htmat/IEC61131/index.html>
- 14 Celano, Lars, MELSOFT Logiikan ohjelmointiohjelmat. Logiikoiden ohjelmointi, ylläpito ja vianetsintä. Beijer Electronics. Luentosarja. 2004. 72 s.
- 15 Carlsson, Malin – Ilomäki, Jari, GX IEC Developer kurssi. Beijer Electronics. Luentosarja. 2005. 72 s.

- 16 FX-peruskurssi13. Tervetuloa FX-peruskurssille. Beijers. Beijer Electronics. Luentosarja. 36 s.
- 17 FX-peruskurssi33. MELSEC FX-sarja. Beijers. Beijer Electronics. Luentosarja. 37 s.
- 18 FX-peruskurssi 23. MELSEC FX-sarja. Beijers. Beijer Electronics. Luentosarja. 10 s.
- 19 Carlsson, Malin – Ilomäki, Jari, GX IEC Developer. Beijer Electronics. Luentosarja. 49s.
- 20 Peltonen, Juhani, Mikroprosessorit/Tietokoneorganisaatio. [www-sivu]. Turun Yliopisto. IT-laitos. Elektroniikka ja tietoliikennetekniikka. 2003. [viitattu 18.1.2006.] Saatavissa: http://users.utu.fi/~jykapa/kurssi/ett_1007/moniste.pdf
- 21 TKT-1100 Digitaalitekniikan perusteet - syksy 2005. [www-sivu]. [viitattu 18.1.2006.] Saatavissa: <http://www.tkt.cs.tut.fi/kurssit/1100/luennot/>
- 22 Maniacen Assembly-opas. [www-sivu]. [viitattu 3.2.2006.] Saatavissa: http://wiki.mureakuha.com/wiki/Maniacen_Assembly-opas
- 23 Ensimmäinen ATK-sanasto. [www-sivu]. [viitattu 3.2.2006.] Saatavissa: <http://sano.se/suomeksi/?li=bitti>
- 24 Rinne, Olli, Pikseli. Mitä ovat bitti, tavu, kilotavu, megatavu ja megapikseli? [www-sivu]. [viitattu 3.2.3006.] Saatavissa: http://www.pikseli.fi/digifaq/2_bitti.html
- 25 Wikipedia. Vapaa tietosanakirja. Ohjelmoitava logiikka. [www-sivu]. [viitattu 6.2.2006]. Saatavissa: http://fi.wikipedia.org/wiki/Ohjelmoitava_logiikka
- 26 Wikipedia. Vapaa tietosanakirja. Oktaalijärjestelmä. [www-sivu]. [viitattu 8.2.2006]. Saatavissa: <http://fi.wikipedia.org/wiki/oktaali>
- 27 Mäkelä, Seppo, Ajastinmateriaali. [sähköpostiviesti.] 29.01.2006.
- 28 Mäkelä, Seppo, MCR-materiaali. [sähköpostiviesti.] 08.02.2006.
- 29 Mäkelä Seppo, Tavutieto-materiaali. [sähköpostiviesti.] 08.02.2006.
- 30 Mäkelä Seppo, Toimilohkon luominen. [sähköpostiviesti.] 08.02.2006
- 31 GX IEC Developer-ohjelma. Versio 6.10. IEC Programming and Documentation System for IBM-PC and Compatible. Mitsubishi Electric. MELSOFT.
- 32 E-Designer –ohjelma. Versio 7.01. Mitsubishi Electric. Build 204.
- 33 Mäkelä, Seppo, Harjoitustehtävät. Tampereen ammattikorkeakoulu. Kone- ja laiteautomaatio. Ohjausjärjestelmät.


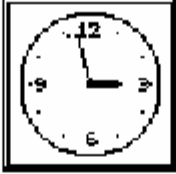









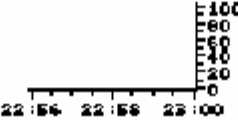
Erikoismuistipaikat /3;17/

Osoite	Toiminto	Kuvaus
M8000	RUN-tila	RUN-tulo M8000 
M8001	RUN-tila	M8001 
M8002	Alustuspulssi	M8002 
M8003	Alustuspulssi	M8003 
M8004	Virheen ilmaisu	M8004 virhe Rele asetetaan, kun M8060 ja/tai M8067 on jo asetettu.
M8005	Alhainen paristojännite	Rele asetetaan, kun paristojännite on epätavallisen alhainen
M8012	Pulssi, 100 ms	Tuottaa 0,1 sekunnin pulssin
M8013	Pulssi, 1 s	Tuottaa 1 sekunnin pulssin
M8030	Paristo-LED OFF	Kun M8030 on asetettu, ei PLC:n etulevyllä oleva LED pala, vaikka paristojännite on alhainen
M8033	Datan ennallaanpito	Kuvarekisterin ja datamuistin sisältö säilyvät ennallaan vaihdettaessa PLC:n toimintatila RUN->STOP
M8039	Vakio-ohjelmankiertoaika	Kun M9038 on asetettu, PLC aloittaa uuden ohjelmakierron rekisterissä D8039 olevan arvon mukaan.
M8060	I/O-osoitus virheellinen	LED OFF, RUN-tila
M8067	Suoritusvirhe	LED OFF, RUN-tila

Virhekoodit /3/

Virhe	Virhekoodi	Virheen kuvaus	Korjaus
		Ei virhettä	
D8061	6101-6103	Laitevika ohjelmoitavassa logiikassa	On tarkistettava, onko jatkokaapelit liitetty oikein.
D8062	6201-6205	Tiedonsiirtovirhe PLC:n ja oheislaitteen välillä	On tarkistettava, onko oheislaitte ja PLC liitetty toisiinsa oikein.
D8063	6301-6306	Linkkivirhe	On tarkistettava onko molempien logiikkojen virta kytketty päälle ja onko toisaalta linkkiadapterin ja PLC:n väliset ja toisaalta linkkiadapterien keskinäiset liitännät tehty oikein.
D8064	6401-6409	Parametriverhe	Väärät tiedot korjattava parametrintoimintatilassa.
D8085	6501-6509	Syntaksivirhe	Jokainen käsky tarkistetaan ohjelmoinnin aikana. Syntaksivirheen esiintyessä virhe on poistettava ohjelmointitilassa.
D8066	6601-6609	Ohjelmointivirhe	Ohjelmointivirhe esiintyy, kun kyseessä on virheellinen käsky-yhdistelmä tai parittain toisiinsa liittyvien käskyjen virheellinen riippuvuussuhde. Tunnistetut virheet korjattava ohjelmointitoimintatilassa.
D8067	6701-6709	Suoritusvirhe	Nämä virheet esiintyvät käskyä käsiteltäessä. Virheen esiintyessä PLC on pysäytettävä heti ja virheet on poistettava ohjelmointitoimintatilassa. Suoritusvirhe esiintyy, kun syntaksi- tai ohjelmointivirheestä tulee virheilmoitus.

E-Designer-ohjelman graafisia symboleja /4;5;10;11;32/

Painike	Symboli	Kuvaus
 Analog Clock		Analoginen kello ilmoittaa kellonajan.
 Digital Clock		Digitaalinen kello voidaan asettaa ilmoittamaan sekä ajan että päivämäärän.
 VU Meter		Graafinen VU-mittari
 Speedometer		Graafinen nopeusmittari
 Diagram		Esittää X/Y-diagrammina ohjelman rekisteriarvoja reaaliajassa.
 Trend		Näyttää graafisesti analogisia signaaliarvoja.

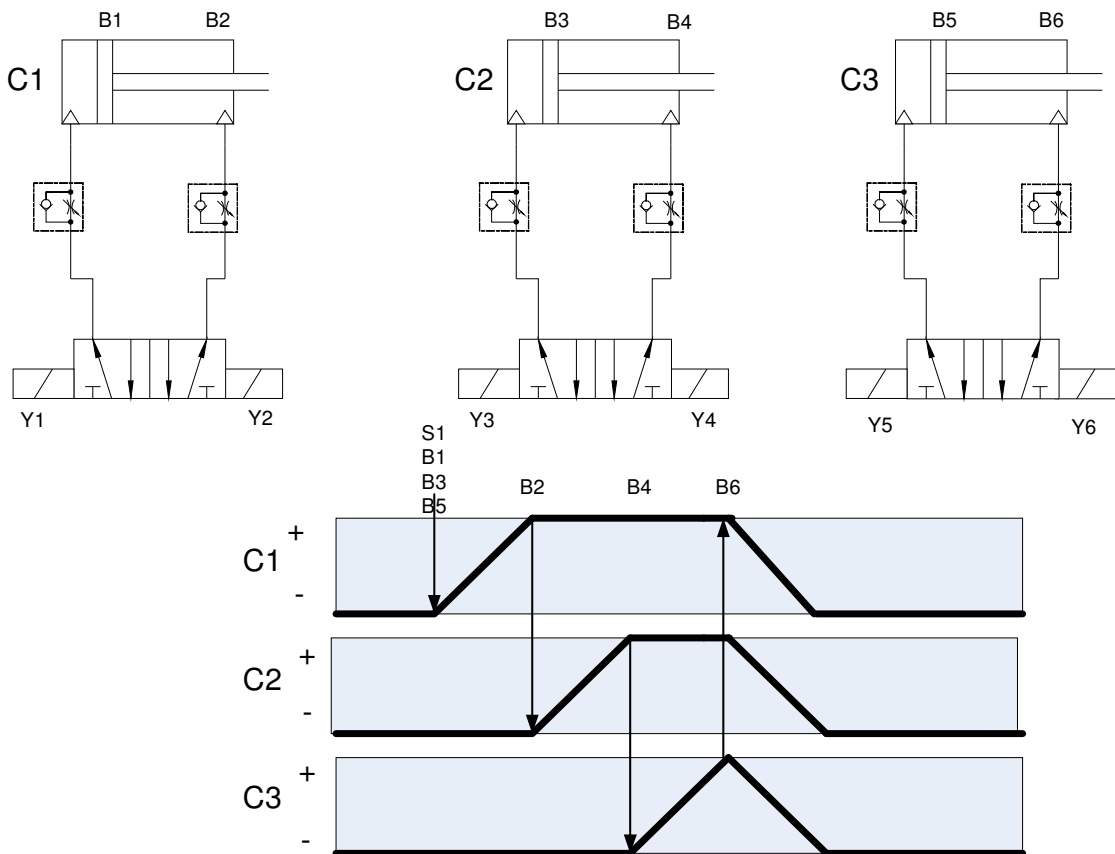
Harjoitustehtävät /33/

Harjoitus 1

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Alapuolella olevassa kaaviossa on kuvattu sylintereiden liike matka-askelkuvaajalla. Laadi ohjelma, joka toteuttaa kyseisen liikeradan sylintereillä.

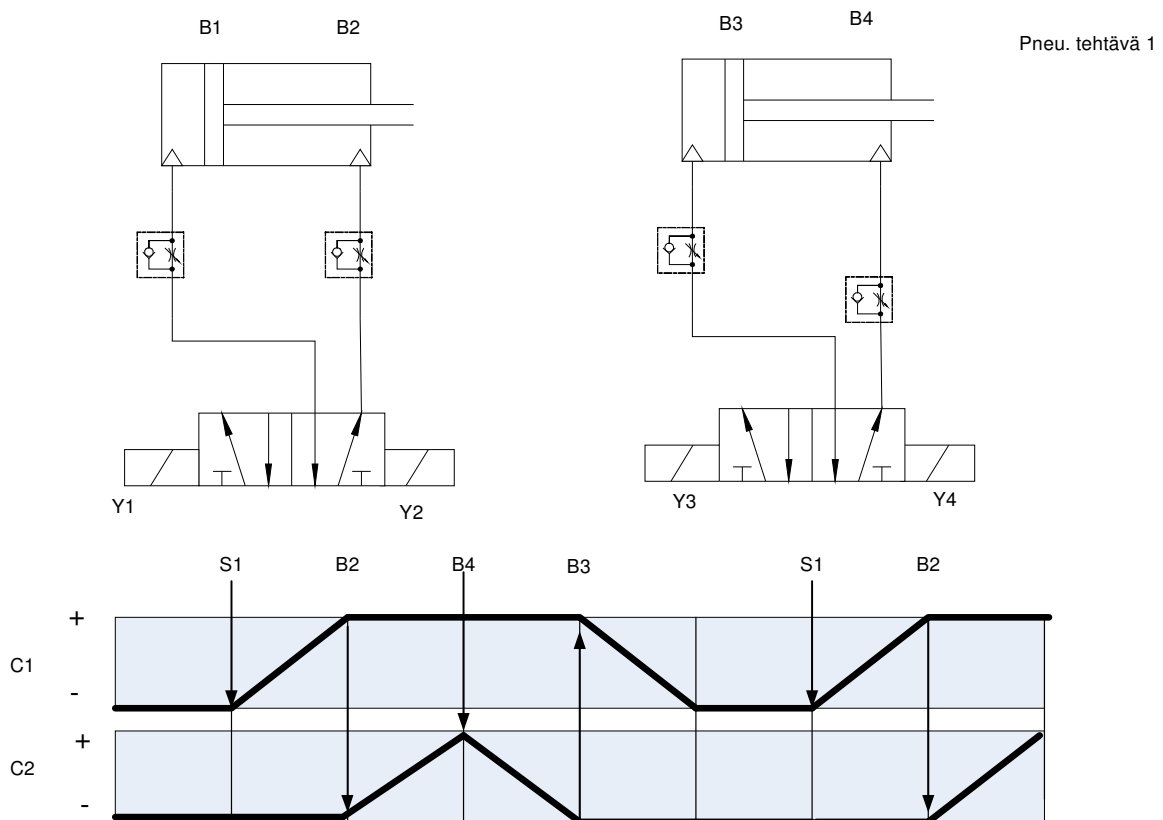


Harjoitus 2

Tehtävän asetetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla käyttäen FBD (FUP) -esitysmuotoa.

Alapuolella olevassa kaaviossa on kuvattu sylintereiden liike matka-askelkuvaajalla. Laadi ohjelma, joka toteuttaa kyseisen liikeradan sylintereillä.

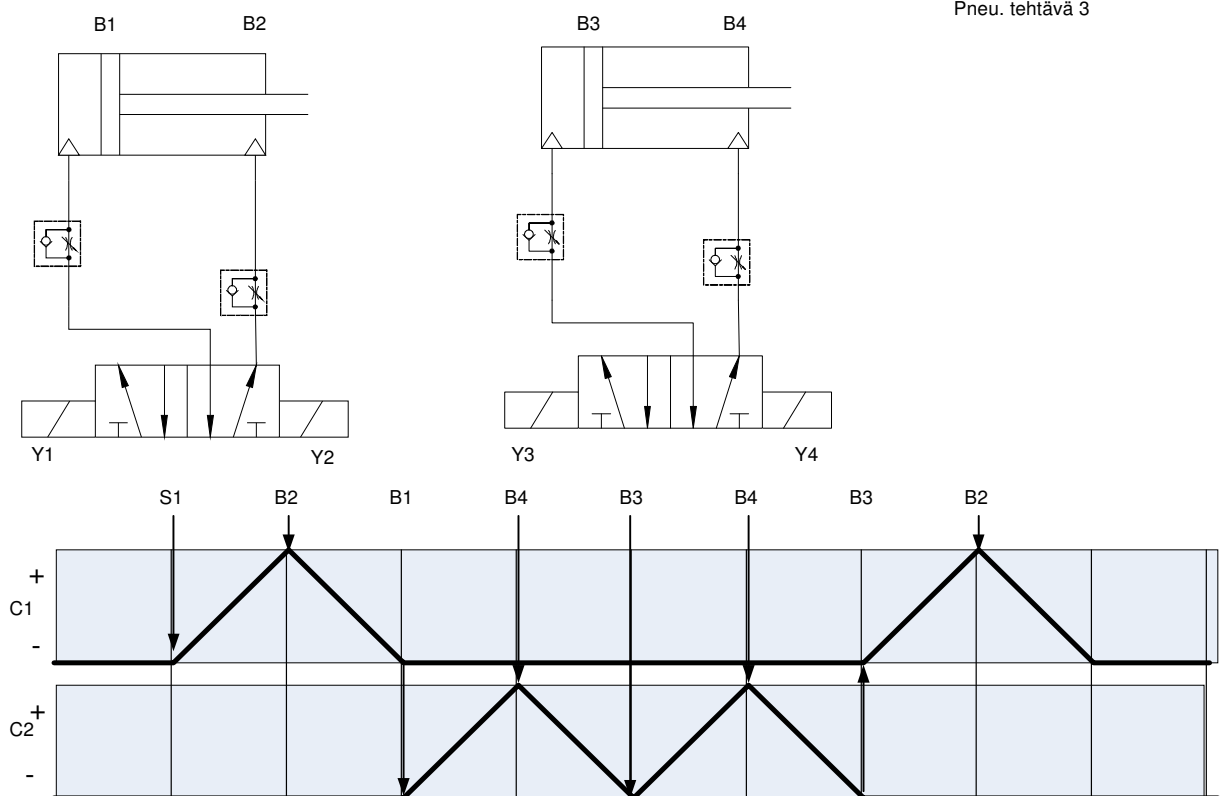


Harjoitus 3

Tehtävän asetetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla käyttäen FBD (FUP) -esitysmuotoa.

Alapuolella olevassa kaaviossa on kuvattu sylintereiden liike matka-askelkuvaajalla. Laadi ohjelma, joka toteuttaa kyseisen liikeradan sylintereillä.

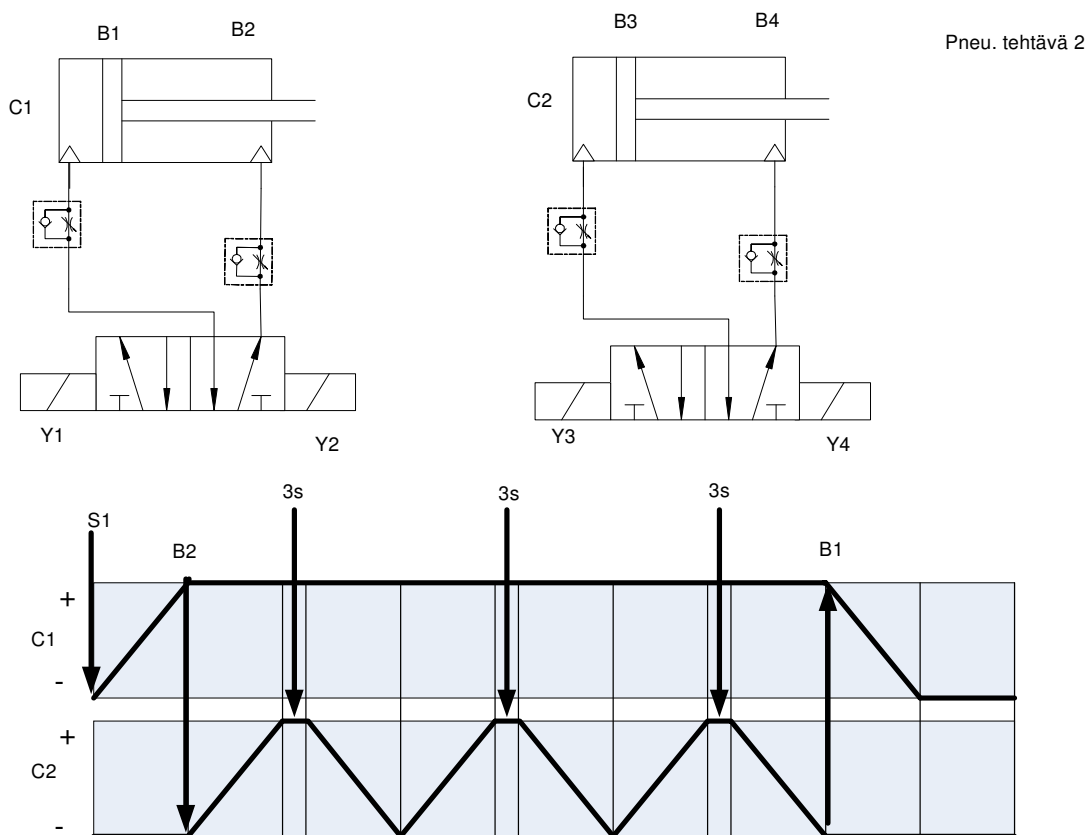


Harjoitus 4

Tehtävän asetetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla käyttäen FBD (FUP) -esitysmuotoa.

Alapuoella olevassa kaaviossa on kuvattu sylintereiden liike matka-askelkuvaajalla. Laadi ohjelma, joka toteuttaa kyseisen liikeradan sylintereillä.



Harjoitus 5 - Puristin

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla käyttäen FBD (FUP) -esitysmuotoa.

Puristimen työvaihe sujuu seuraavasti:

- Pistimen lasku

Puristimen pistin laskeutuu, kun

- suojaverkko **oli** ennen työvaihetta auki ja on **nyt** kiinni (rajakytkimet S3 ja S4 toimineena),
- puristin **on** perusasennossa, eli on ylhäällä (rajakytkin S6 toimineena) ja
- kahdenkädenkytkimet S1 ja S2 **eivät** ennen työvaihetta olleet toimineet ja **nyt niitä on painettu**.

- Pistimen pysäyttäminen laskun aikana

Pistin pysähtyy heti, kun

- jompikumpi kahdenkädenkytkimistä **päästetään vapaaksi** (pätkittäin ajo mahdollinen),
- tai suojaverkko **avataan**,
- tai pistin on saavuttanut ala-asennon (rajakytkin S5 toimineena).

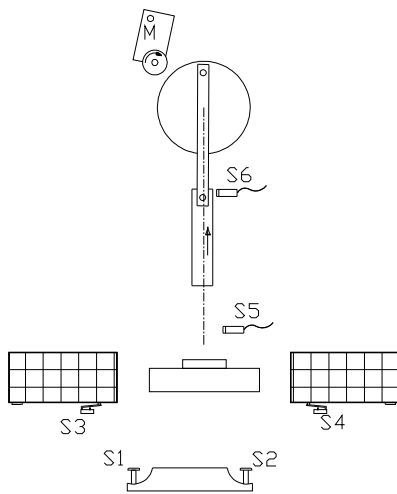
- Pistimen nosto

Kun pistin on alhaalla (rajakytkin S5 toimineena), niin se liikkuu epäkeskon kautta jälleen ylöspäin. Tällöin voidaan suojaverkko avata ja kahdenkädenkytkimet päästää irti.

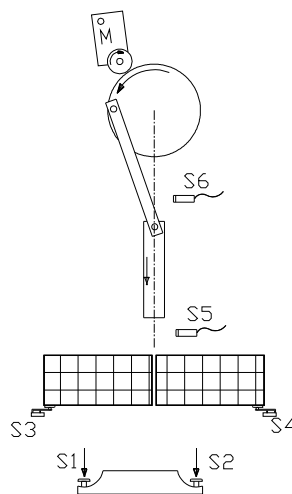
Kun pistin on jälleen perusasennossa (rajakytkin S6 toimineena), on työvaihe päättynyt. Huom. Epäkeskopuristimen moottori pyörii koko ajan. Puristimen toimintaa ohjataan kitkakytkimen kautta.

Symb.osote	Abs.osote	Nimike	Toiminta NO/NC	
S1	X1	Kahdenkädenkytkin	NO	
S2	X2	Kahdenkädenkytkin	NO	
S3	X2	Suojaverkon kytkin	NO	
S4	X3	Suojaverkon kytkin	NO	
S5	X4	Ala-asento	NO	
S6	X5	Yläasento	NO	
S7	X6			
S8	X7			
K1	Y1	Kytkimen kontaktori	NO	
M1	M1			
M2				
M3				
M4				

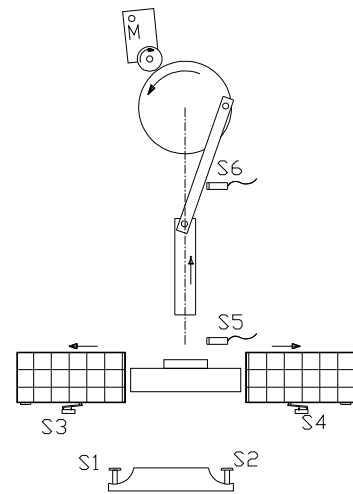
Pistin yhäällä



Pistin laskee



Pistin nousee ylös



Harjoitus 6 - Puskurivarasto

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintietoja jne.) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu ja jne.). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, ajastimet ja laskurit jotta nähtäisiin mitkä näistä on vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Asennuslinjalla on moottoreita varten puskurivarasto, johon mahtuu enintään 30 moottoria. Järjestelmän käynnistyksen merkiksi sytytetään merkkilamppu L1. Järjestelmä käynnistetään kytkimellä S1 ja laskuri nollataan painikkeella S2.

- Lukumäärän valvonta

Moottoreiden tuloa ja lähtöä valvotaan kahdella valokennolla (VK1 ja VK2). Valokennojen ”ON” signaalit menevät laskuriin. Jos puskurivarasto on työvuoron alussa tyhjä, voidaan laskuri nollata painikkeella.

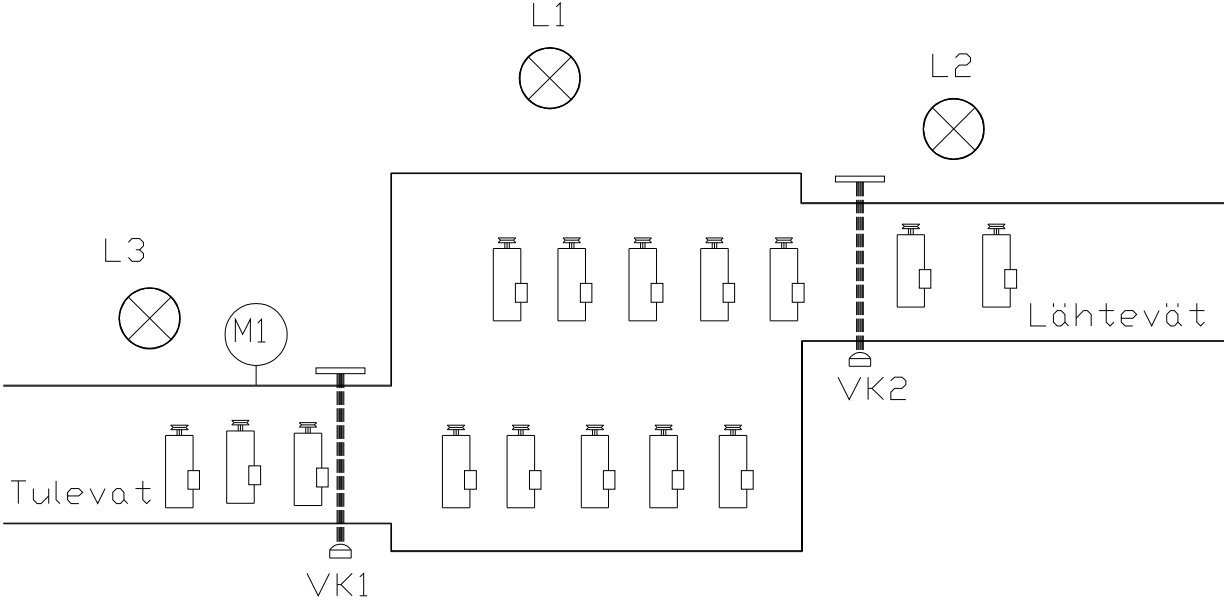
- Yläraja-arvo

Kun puskurivarastossa on 20 moottoria, se on täysi. Kuljettimen moottori M1 on kytkettävä heti pois päältä, ja sytytettävä merkkilamppu L2 vilkkumaan.

- Alaraja-arvo

Kun puskurivarastossa on vähemmän kuin 6 moottoria, siitä on saatava viesti merkkilampulle L3

Symb.osote	Abs.osote	Nimike	Toiminta NO/NC	
S1	X1	Järj. Päälle/Pois		
S2	X2	Laskurin nollaus		
VK1	X3	Vaalokenno1		
VK2	X4	Vaalokenno2		
L1	Y1	Lamppu, järjestelmä käynnissä		
L2	Y2	Lamppu, varasto täynnä		
L3	Y3	Lamppu, varasto vajaa		
M1	Y4	Hihnakuljetin		
Laskuriarvo	D10	Varastossa olevat kappaleet		

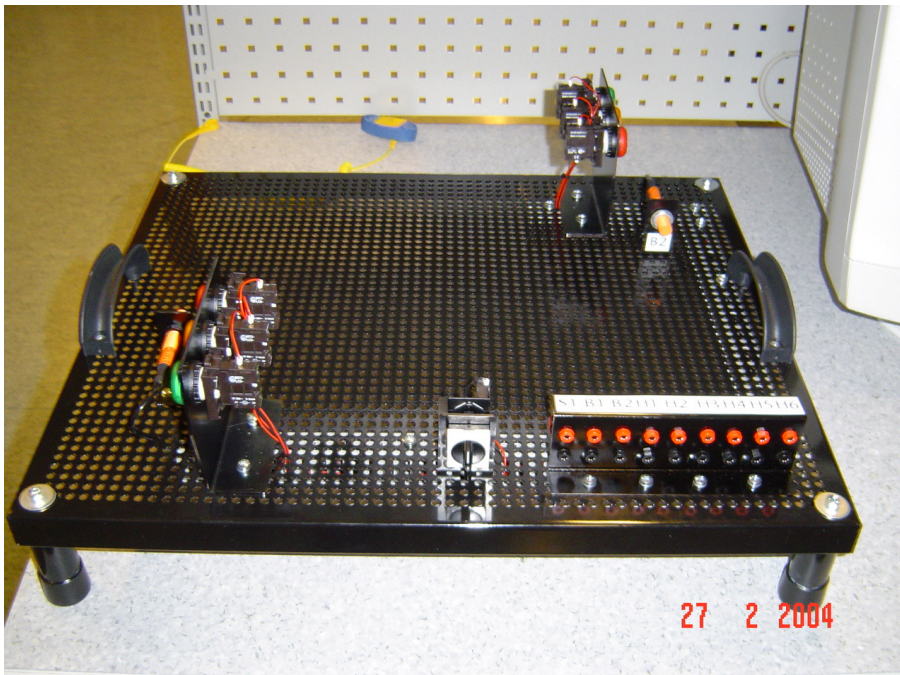


Harjoitus 7 – Työmaan liikennevalot

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto ja jne.) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu ja jne.). Tähän vertailulistaan tulee ottaa mukaan myös merkkeit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Tietöiden vuoksi on kaksisuuntaisen kadun toinen ajokaista pois käytöstä. Liikenne kadulla on hyvin vilkasta, jonka takia kadun varteen asennetaan väliaikaiset liikennevalot. Tehtävänä on laatia valojen ohjaamiseksi logiikkaohjelma ja todeta ohjelman toimivuus kuvassa 1 esitetyn harjoituslaitteiston avulla.



Kuva 1 Harjoituslaitteisto

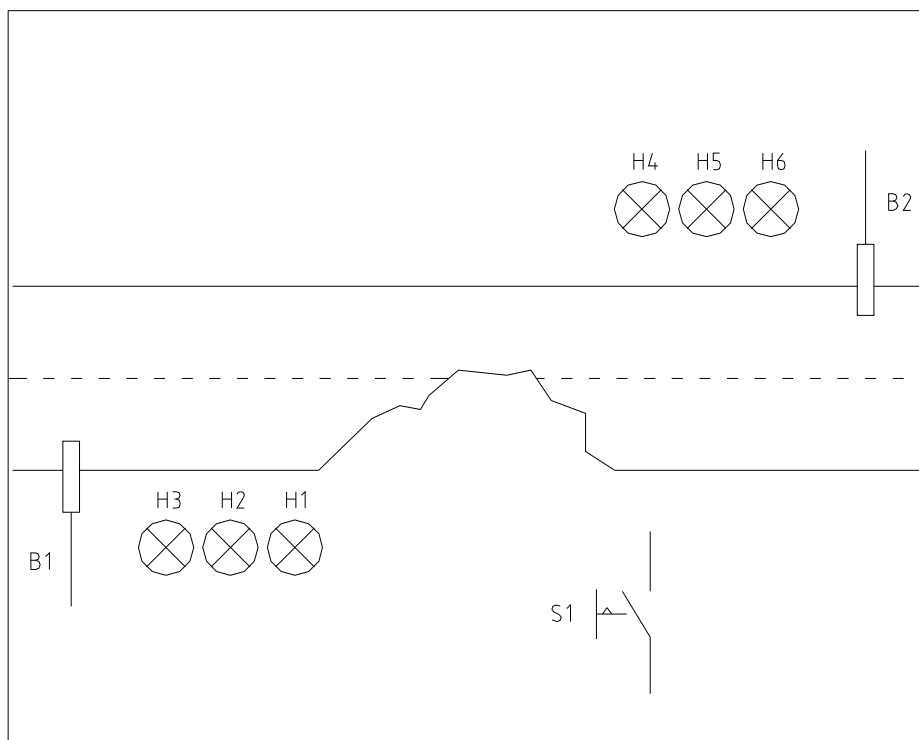
Valojen toiminta

Kun laitteisto kytketään päälle vipukytkimellä S1, palaa punainen valo molempiin suuntiin. Kun toinen induktiivisista lähestymiskytkimistä tunnistaa auton, tulee vastaavan suunnan valon vaihtua 10 sekunnin kuluttua keltaisen kautta vihreäksi. Keltainen valo palaa kolme sekuntia, jonka jälkeen vihreä valo palaa 20 sekuntia. Tämän jälkeen valo vaihtuu kolmeksi sekunniksi keltaiseksi, jonka jälkeen takaisin punaiseksi. Toiminta alkaa alusta. Nyt tietenkin vastakkaisen ajosuunnan valot vaihtuvat, jos molemmat lähestymiskytkimet tunnistavat auton.

Laitteiston tulee kytkeytyä pois päältä vain jommankumman suunnan valokierron jälkeen, vaikka kytkin S1 käännetään nolla asentoon missä valokierron vaiheessa tahansa. Ohjausta päälle kytkettäessä tulee perustilan, molemmat valot punaisena, tulla voimaan ilman ehtoja.

Prosessikaavio ja johdotuskaavio

Kuvassa 2 on harjoitustyön prosessikaavio. Taulukossa 1 on symbolien selitykset. Samat symbolit löytyvät myös harjoituslaitteistosta. Taulukkoa voidaan käyttää avuksi laadittaessa ohjelman operandilistaa. Sivulla 3 on harjoituslaitteiston johdotuskaavio. Sitä voidaan käyttää apuna laitteiston ja logiikan välisiä johtimia liitettäessä.



Kuva 2 Prosessikaavio

Kaavio 1 Symbolien selitykset

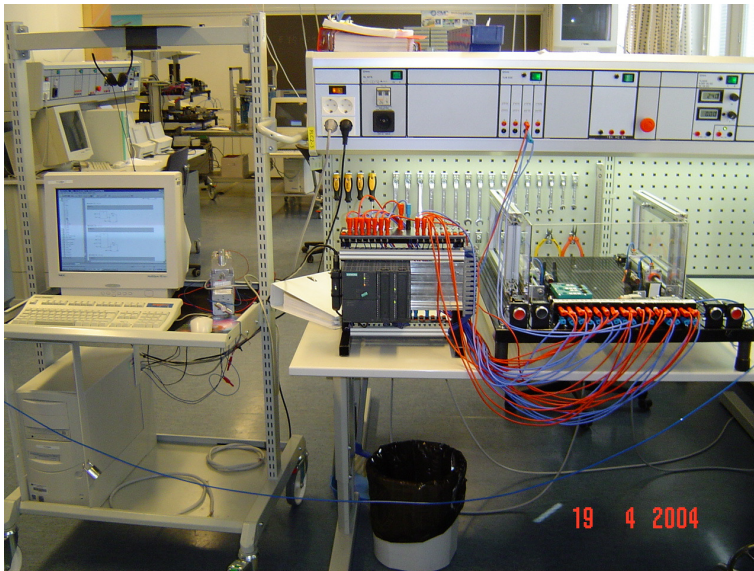
Symboli	Kommentti
S1	Laitteisto päälle (NO)
B1	Vasen lähestymiskytkin (NO)
B2	Oikea lähestymiskytkin (NO)
H1	Vasen punainen valo
H2	Vasen keltainen valo
H3	Vasen vihreä valo
H4	Oikea punainen valo
H5	Oikea keltainen valo
H6	Oikea vihreä valo

Harjoitus 8 – Puhdastilan ovet

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto ja jne.) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu ja jne.). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Laboratorion pitää pysyä mahdollisimman pölyttömänä. Sen eteen on rakennettu välitila, jonka kautta laboratorioon kuljetaan. Välitilan molemmissa päissä on liukuovet. Tehtävänäsi on laatia logiikkaohjelma ovien ohjaamiseksi ja todeta ohjelman toimivuus kuvan 1 mukaisella laitteistolla.



Kuva 1 Harjoituslaitteisto

Ovien toiminta

Päästäkseen kulkemaan välitilan läpi on painettava painiketta S1 tai S2 välitilan ulkopuolella. Painikkeissa on myös merkkilamput H1 ja H2, jotka kertovat käyttäjälle laitteiston havainneen painikkeen painamisen.

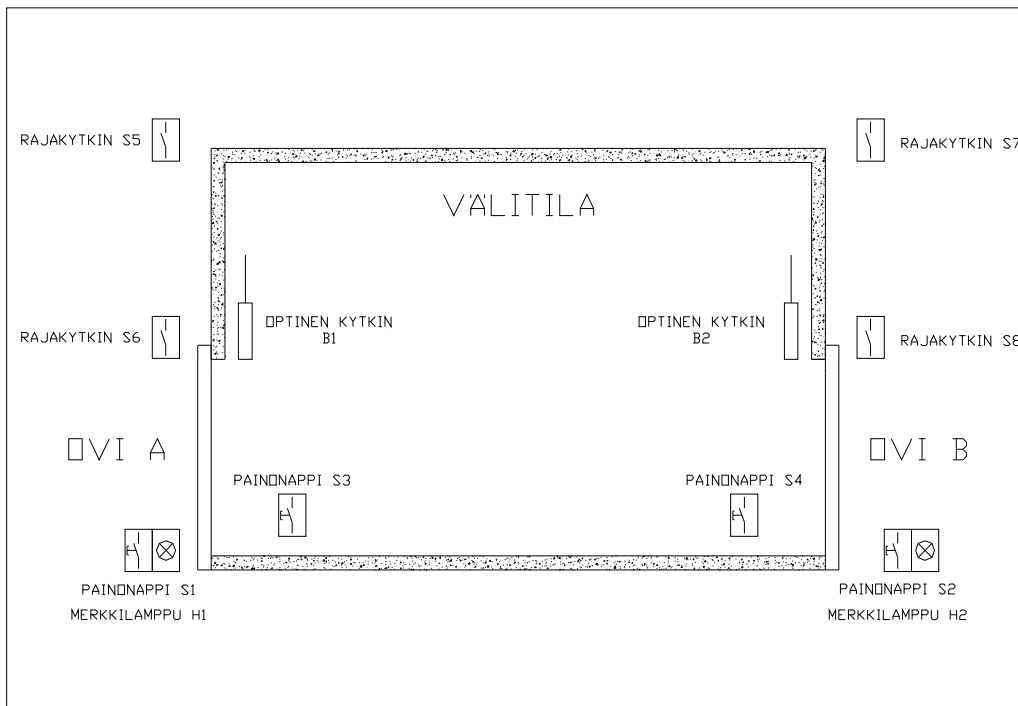
Kun halutaan mennä laboratorioon, painetaan painiketta S1. Ovi A avautuu. Kun ovi on ollut auki kolme sekuntia, se sulkeutuu. Oven A sulkeuduttua, aukeaa ovi B automaattisesti ja on auki kolme sekuntia. Toiminta toistuu päinvastaisessa järjestyksessä, kun painetaan painiketta S2, laboratoriosta ulos tultaessa.

Kumpaankin oveen on asennettu mekaaniset rajakytkimet S5, S6, S7 ja S8, jotka ilmoittavat ovien olevan täysin auki tai kiinni. Lisäksi välitilaan on asennettu kaksi optista lähestymiskytkintä B1 ja B2. Ovi ei saa mennä kiinni, jos optinen kytkin

havaitsee esteen oven välissä. Välitilassa sisällä on myös ovien avauspainikkeet S3 ja S4. Jos joku jää välitilaan, hän saa haluamansa oven auki näistä painikkeista.

Prosessikaavio ja johdotuskaavio

Kuvassa 2 on laitteiston prosessikaavio. Prosessikaaviossa käytettyjen symbolien selitykset löytyvät taulukosta 1. Sivulla 3 on laitteiston johdotuskaavio, jota voi käyttää apuna laitteiston ja logiikan välisiä johtimia kytkettäessä.



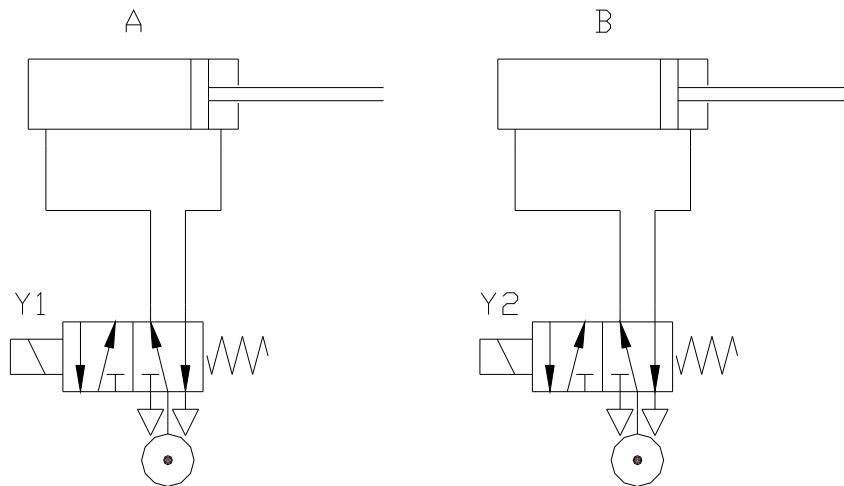
Kuva 2 Prosessikaavio

Taulukko 1 Symbolien selitykset

Symboli	Kommentti
S1	Oven A ulkopuolinen painonappi (ovien avaus)
S2	Oven B ulkopuolinen painonappi (ovien avaus)
S3	Oven A sisäpuolinen painonappi (oven A avaus)
S4	Oven B sisäpuolinen painonappi (oven B avaus)
S5	Mekaaninen rajakytkin (ovi A auki)
S6	Mekaaninen rajakytkin (ovi A kiinni)
S7	Mekaaninen rajakytkin (ovi B auki)
S8	Mekaaninen rajakytkin (ovi B kiinni)
H1	Merkkilamppu (ilmaisee S1:n painamisen)
H2	Merkkilamppu (ilmaisee S2:n painamisen)
B1	Optinen lähestymiskytkin (oven A välin valvonta)
B2	Optinen lähestymiskytkin (oven B välin valvonta)
Y1	3/2 suuntaventtiilin kela (ohjaa ovea A)
Y2	3/2 suuntaventtiilin kela (ohjaa ovea B)

Pneumatiikkakaavio

Ovia liikuttavat kaksitoimiset pneumatiikkasyylinterit. Sylinterien ohjaus on toteutettu 3/2 suuntaventtiileillä. Venttiilit ovat sähköohjattuja ja jousipalautteisia. Ovea A ohjaavan venttiilin kelan tunnus on Y1. Ovea B ohjaavan venttiilin kela on tunnukseltaan Y2. Alla olevasta kuvasta nähdään laitteiston pneumatiikkakaavio (Kuva 3).



Kuva 3 Pneumatiikkakaavio

Harjoitus 9 - Hihnakuuljetin

Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto ja jne.) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu ja jne.). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Paineilmasyylinteri 1 siirtää kappaleita N ja M kuljettimelle 2, jossa ne tarkistetaan konenäöllä. Tarkistuksesta saadaan joku seuraavista tuloksista:

- kappale N hyväksytty
- kappale M hyväksytty
- kappale N hylätty
- kappale M hylätty

Tulokset on luettavissa kameran lähtötietoina (OUT) neljän bitin koodeina:

- 0000 kappale (N hyväksytty)
- 0110 kappale (M hyväksytty)
- 1001 kappale (N hylätty)
- 1111 kappale (M hylätty)

Kappaleet N kuljetetaan hihnalla 2 suuntaan A jossa ne erotellaan hyväksytyihin ja hylättyihin. Hyväksytyt siirretään paineilmasyylinterillä 3 hihnalle 4 ja hylätyt putoavat hihnalta 2 suunnassa A hylättyjen laatikkoon.

Kappaleet M kuljetetaan hihnalla 2 suuntaan B jossa ne erotellaan hyväksytyihin ja hylättyihin. Hyväksytyt siirretään paineilmasyylinterillä 2 hihnalle 3 ja hylätyt putoavat hihnalta 2 suunnassa B hylättyjen laatikkoon.

Sylinteri 1:n toimittua on 15 s tauko, jolloin suoritetaan konenäkö tarkastus kappaleille N tai M. Tarkastuksen jälkeen hihna 2 käynnistyy tarkastuksen tuloksen mukaiseen suuntaan. Hihna 2 on toiminnassa kunnes induktiivinen tunnistin B1 tai B2 antaa signaalin, mikäli kyseessä oli hyväksytty kappale. Jos kyseessä oli hylätty kappale, hihnan 2 käyntiaika on 25 sekuntia, jonka aikana kappaleet putoavat hylättyjen laatikkoon. Hyväksytyjen kappaleiden yhteismäärä pitää laskea. Uuden kappaleen siirtäminen sylinterillä 1 tapahtuu 35 s kuluttua hihnan 2 edellisestä käynnistymisestä.

HUOM! Kuljetinjärjestelmä käsittelee vain yhtä kappaletta kerrallaan. Häätä-seis toiminnan jälkeen tulee uudelleenkäynnistys tapahtua hallitusti.

Opettaja määrittelee kullekin ryhmälle ohjausventtiilien tyypit.

Sylinteriä 1 ohjataan venttiilillä 1	5/2 sähkö/sähkö tai sähkö/jousi
Sylinteriä 2 ohjataan venttiilillä 2	5/2 sähkö/sähkö tai sähkö/jousi
Sylinteriä 3 ohjataan venttiilillä 3	5/2 sähkö/sähkö tai sähkö/jousi

Opetuslaitteen laitetunnukset

S1	kappaleen syöttö
S2	Järjestelmän käynnistys
S3	Hätä-seis
S4	Järjestelmän pysäytys
S5	Hätä-seis toiminnan kuittaus
S6	Koodikytkin 1
S7	Koodikytkin 2
S8	Koodikytkin 3
S9	Koodikytkin 4
S10	Järjestelmän pääkytkin
H1	Järjestelmä käynnissä
H2	Hätä-seis painettu
B1	Vasen tunnistin
B2	Oikea tunnistin
M1	Vasen moottori
M2	Oikea moottori
M3	Pääkuljettimen moottori
K1	M1:n kontaktori
K2	M2:n kontaktori
K3	M3:n kontaktori
K4	M3:n suuntakontaktori
K5	Turvarele
K6	5 VDC jännitesyöttö
K7	
Y1-Y8	Vaihtoehtoiset paineilmahjaukset
Asi os11	Asi-moduuli 1
Asi os9	Asi-moduuli 2
ASI os23	Asi-moduuli 3

Symb.osote	Abs.osote	Nimike	Toiminta NO/NC	
S1	X1	kappaleen syöttö		
S2	X2	Järjestelmän käynnistys		
S3	X3	Hätä-seis		
S4	X4	Järjestelmän pysäytys		
S5	X5	Hätä-seis toiminnan kuittaus		
S6	X6	Koodikytkin 1		
S7	X7	Koodikytkin 2		
S8	X10	Koodikytkin 3		
S9	X11	Koodikytkin 4		
K5:14	X12	Tieto turvareleelta		
B1	X13	Vasen tunnistin		
B2	X14	Oikea tunnistin		
K1	Y1	M1:n kontaktori		
K2	Y2	M2:n kontaktori		
K3	Y3	M3:n kontaktori		
K4	Y4	M3:n suuntakontaktori		
K5		Turvarele		
K6		5 VDC jännitesyöttö		
Y1...Y8	Y10...Y17	Vaihtoehtoiset paineilmaohjaukset		

Harjoitus 10 - Porausautomaatti

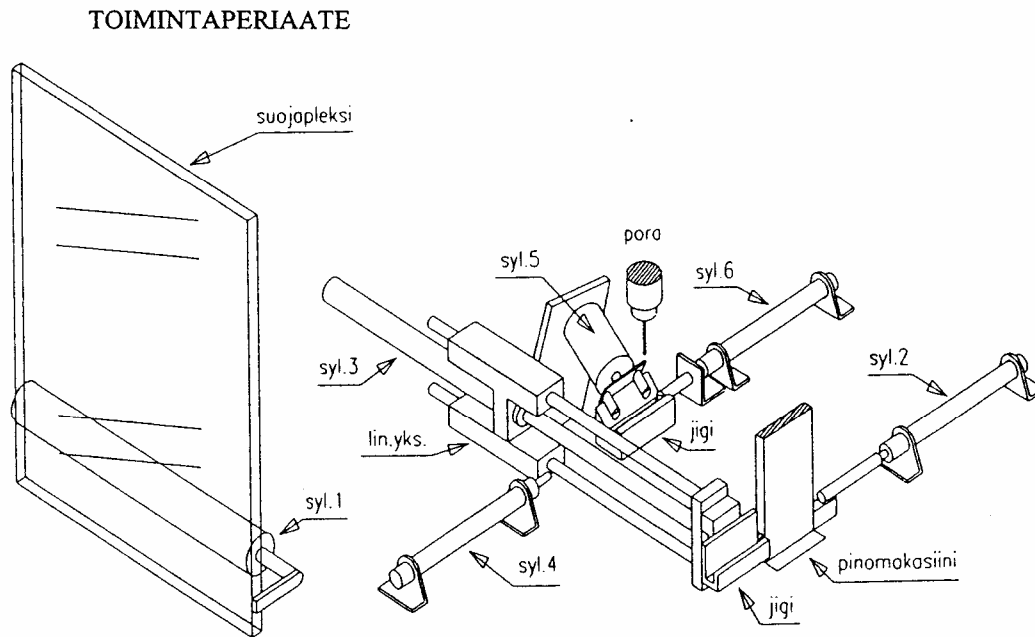
Tehtävän asetus

Ennen varsinaisen ohjelman kirjoitusta on laadittava parametri- eli vertailulista. Tässä listassa määritellään, mihin laitetuloihin on liitettävä laitteen tulotieto (anturi- ja kytkintieto ja jne.) ja mihin lähtöihin lähtevä tieto (kontaktori, merkkilamppu ja jne.). Tähän vertailulistaan tulee ottaa mukaan myös merkkerit, jotta nähtäisiin mitkä niistä ovat vielä käytettävissä. Tee harjoitus GX IEC Developer FX -ohjelmalla, käyttäen FBD (FUP) -esitysmuotoa.

Ohjelman vaatimukset ovat:

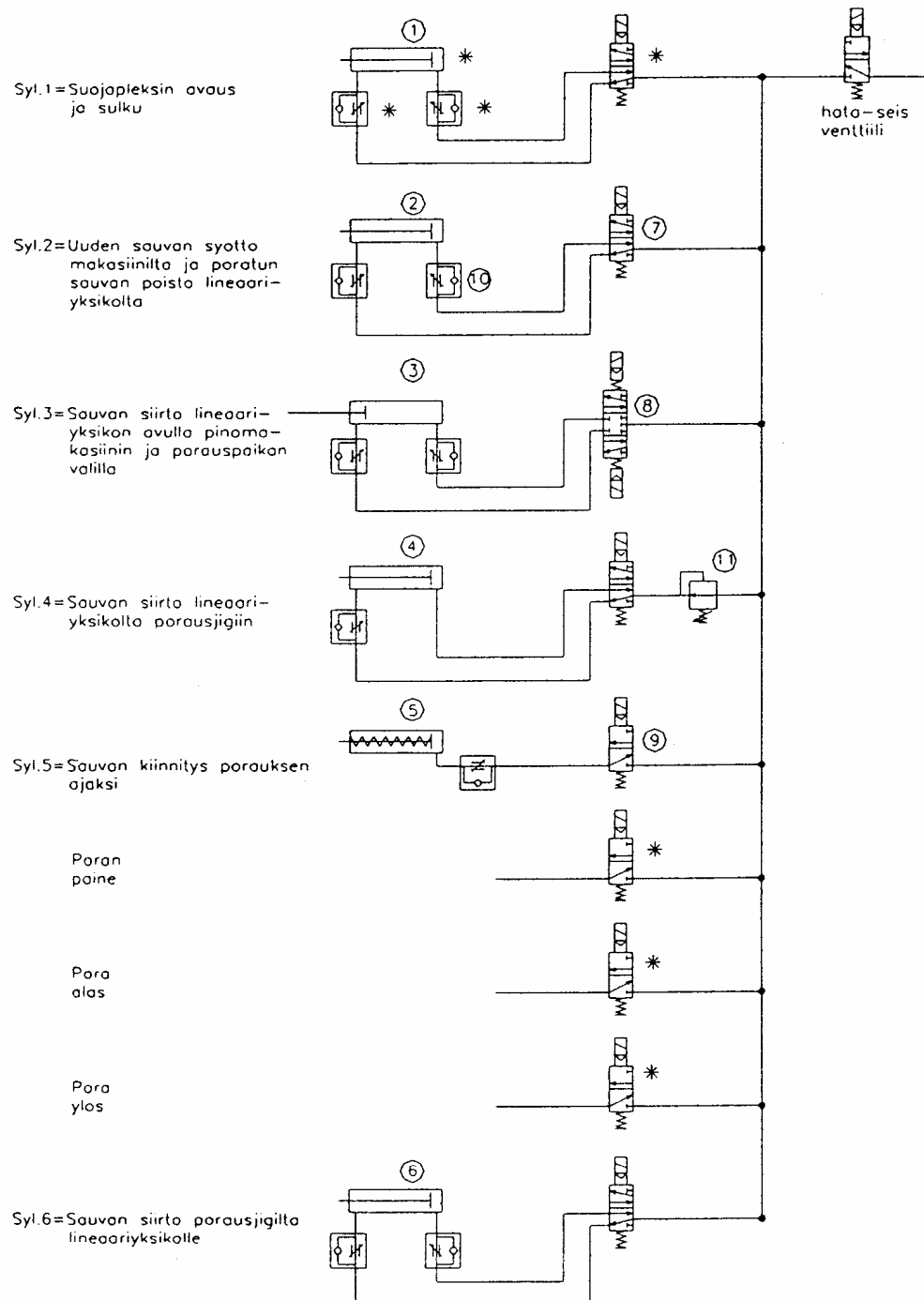
- laitteen toiminta on automaattinen
- laite käynnistyy START painikkeella
- laite pysähtyy STOP kytkimestä, suorittaen ohjelmakierron loppuun
- laite pysähtyy välittömästi HÄTÄ-SEIS painikkeesta
- laite ei saa käynnistyä uudestaan ilman kuittausta kun HÄTÄ-SEIS painike vapautetaan
- ohjelma laaditaan jäljempänä olevan parametrilistan mukaan

Symb.osote	Abs.osote	Nimike	Toiminta NO/NC	
S1	X0	Käynnistyspainike		
S2	X1	Pysäytyskytkin		
S3	X2	Hätä-Seiskytkin		
B1	X3	Syl. 1.n – asema		
B2	X4	Syl. 1.n + asema		
B3	X5	Syl. 2.n – asema		
B4	X6	Syl. 2.n + asema		
B5	X7	Syl. 3.n + asema		
B6	X10	Syl.3.n – asema		
B7	X11	Syl. 4.n – asema		
B8	X12	Syl. 4.n + asema		
B9	X13	Syl. 5.n + asema		
B10	X14	Syl. 6.n – asema		
B11	X15	Syl. 6.n + asema		
B12	X16	Pora ylhäällä		
B13	X17	Pora alhaalla		
H1	Y0	Merkkivalo		
K1	Y1	Hätä-seis venttiili		
K2	Y2	Syl. 1.n venttiili		
K3	Y3	Syl. 2.n venttiili		
K4	Y4	Syl. 3.n – liike		
K5	Y5	Syl. 3.n + liike		
K6	Y6	Syl. 4.n venttiili		
K7	Y7	Syl. 5.n venttiili		
K8	Y10	Syl. 6.n venttiili		
K9	Y11	Pora ylös venttiili		
K10	Y12	Pora alas venttiili		
K11	Y13	Poran paine		

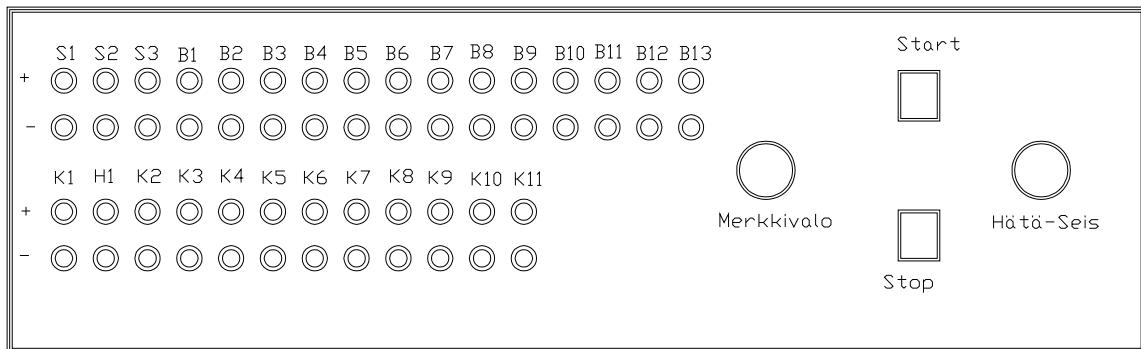


Kuva 1

Sylinteri 1:n sulkiessa suojapleksin sylinteri 2 työntää koesauvan pinomakasiinista lineaariyksikön jigille. Sylinteri 3, joka liikuttelee lineaariyksikköä, tekee miinusliikkeen, jolloin sauva siirtyy porauspaikan viereen. Sylinteri 4 työntää sauvan lineaariyksikön jigiltä porausjigille. Kiinnitys sylinteri 5 kiinnittää sauvan porauksen ajaksi. Sylinteri 6 siirtää sauvan porauksen jälkeen takaisin lineaariyksikön jigille. Lineaariyksikkö tekee plusliikkeen, jonka jälkeen pinomakasiinin sylinteri työntää uuden sauvan makasiinista jigille työntäen poratun sauvan edelleen jigiltä pois jne, kunnes makasiini on tyhjä.



Kuva 2 Laitteiston pneumatiikkakaavio



Kuva 3