

TAMPEREEN AMMATTIKORKEAKOULU
Kone- ja tuotantotekniikan koulutusohjelma
Kone- ja laiteautomaatio

Opinnäytetyö

Jouni Lahtinen

GX Works2 -ohjelman käyttöohje ohjelmoitavan logiikan ohjaamiseen

Työn ohjaaja
Työn teettäjä

DI, lehtori Yrjö Viitanen
Beijer Electronics Oy, Support Engineer Sauli Kotiranta

Tampere 2010

Tekijä	Lahtinen, Jouni
Työn nimi	GX Works2 -ohjelman käyttöohje ohjelmoitavan logiikan ohjaamiseen
Sivumäärä	37 sivua + 33 liitesivua
Valmistumisaika	Toukokuu 2010
Työn ohjaaja	DI, lehtori Yrjö Viitanen
Työn teettäjä	Beijer Electronics Oy, Support Engineer Sauli Kotiranta

Tiivistelmä

Opinnäytetyön tarkoituksena oli tehdä suomenkielinen käyttöohje MELSOFT GX Works2 -ohjelmasta Beijer Electronics Oy:lle ja Tampereen ammattikorkeakoulun kone- ja laiteautomaation suuntautumislinjalle. Ohjeet tehtiin, jotta uudet käyttäjät pystyvät käyttämään ohjelmaa alusta asti tehokkaasti, eikä heidän tarvitse suomentaa vieraskielistä tekstiä.

Beijer Electronics Oy on osa Beijer Electronics Automationia, joka toimii teollisuusautomaatioalalla Pohjoismaissa ja Baltian maissa. Beijer Electronics Oy keskittyy pääasiassa tuotteiden markkinointiin, myyntiin ja asiakkaiden kouluttamiseen.

MELSOFT GX Works2 on uusien FX- ja Q-logiikoiden ohjelmointiohjelmisto, joka korvaa GX Developer ja GX IEC Developer -ohjelmat. Ohjelmoitava logiikka on ohjauslaite, jolla ohjataan lähtöjensä avulla toimilaitteita kuten sähkömoottoreita, releitä ja merkkilamppuja. Ohjaus tapahtuu ohjelman avulla, joka ladataan logiikkaan.

Käyttöohjeen avulla käyttäjä oppii tekemään logiikkapiirejä, jotka ladataan ohjelmoitavalle logiikalle, ja sen jälkeen osaa simuloida ja monitoroida ohjelmaa. Käyttöohjeessa kerrotaan myös teoriaa esimerkiksi binäärijärjestelmästä, BCD-koodista, syklisestä ohjelmankäsittelystä ja operaattoreista.

Avainsanat Ohjelmoitava logiikka, PLC, GX Works2, FX2n

Writer	Lahtinen, Jouni
Thesis	GX Works2 manual for controlling a PLC
Pages	37 pages + 33 appendices
Graduation time	May 2010
Thesis Supervisor	M. Sc., Lecturer Yrjö Viitanen
Co-operating company	Beijer Electronics Oy, Support Engineer Sauli Kotiranta

Abstract

The purpose of the thesis work was to make a Finnish manual to the MELSOFT GX WORKS2 software for Beijer Electronics Oy and TAMK University of applied sciences. The manual was made so that new users could use the software efficiently from the beginning and they shouldn't have to translate the text to Finnish.

MELSOFT GX WORKS2 is the latest programming software for the FX and Q programmable logic controllers. A programmable logic controller is a controlling device that controls for example electric motors, relays and signal lamps. The controlling works through a program that is downloaded to the logic controller.

With the help from the manual the user learns to do logic circuits which are downloaded to the programmable logic controller and after that knows how to simulate and monitor the program. The manual also includes theory about the binary number system, BCD code, cyclic program handling and operators.

Keywords Programmable Logic Controller, PLC, GX Works2, FX2n

Alkusanat

Haluaisin ensinnäkin kiittää Beijer Electronics Oy:n Sauli Kotirantaa ja Tampereen ammattikorkeakoulun laboratorioinsinööri Seppo Mäkelää, kun sain tehdä kyseisestä aiheesta opinnäytetyöni. Sauli Kotirannalle myös kiitokset uuteen ohjelmaan perehdyttämisestä, jonka avulla pääsin nopeammin alkuun. Seppo Mäkelän apu ja neuvot työn kirjoittamisessa olivat myös korvaamattomat. Lisäksi haluaisin kiittää lehtori Yrjö Viitasta opastuksesta työn teossa ja myös ajankäytössä. Näiden lisäksi vielä kiitokset muille opiskelijoille, joilta pystyin kysymään neuvoa aina kun oli vaikea paikka.

Tampereella toukokuussa 2010

Jouni Lahtinen

Sisällysluettelo

Tiivistelmä	2
Abstract	3
Alkusanat	4
Sisällysluettelo	5
Lyhenteitä.....	6
1 Johdanto	7
2 Käyttöohjeen kirjoittaminen	8
3 IEC 61131-3 -standardi	9
4 GX Works2 -käyttöohje FX2n -logiikan ohjelmointiin	11
4.1 Ohjelmoitava logiikka.....	11
4.2 Syklinen ohjelmankäsittely	12
4.3 POU ja Task	13
4.4 Execution program	15
4.5 GX Works2 -ohjelma	15
4.5.1 Yleiskuva	15
4.5.2 Ohjelman käynnistys	16
4.5.3 Uuden projektin luominen	16
4.5.4 Ohjelman alustaminen.....	17
4.5.5 Yleismuuttujien listaaminen.....	19
4.5.6 Ohjelman luominen.....	20
4.5.7 Aputoiminta	22
4.6 Osoitteet	23
4.6.1 Bitti, tavu, puolitavu, sana ja kaksoissana.....	23
4.6.2 I/O-osoitteet	24
4.6.3 Apumuistit.....	24
4.6.4 Virhekoodit	24
4.7 Luvut	25
4.7.1 BCD-luvut	25
4.7.2 Datatyypit.....	25
4.8 Operaattorit	26
4.8.1 Perusoperaattorit	26
4.8.2 Vertailijat.....	28
4.8.3 Ajastimet	29
4.8.4 Laskurit	30
4.8.5 Aritmetiikkakäskyt.....	32
4.8.6 Pääohjaustoiminto	33
4.9 Simulointi.....	33
4.9.1 Projektin kääntäminen.....	33
4.9.2 Offline-simulaattori.....	34
4.10 Ohjelman lataaminen logiikalle	34
4.11 Ohjelman monitorointi	35
5 Loppusanat	36
Lähdeluettelo.....	37
Liitteet	38
Liite 1: OPETUSLAITESALKKU - YLEISKUVA	38
Liite 2: E-DESIGNER.....	40
Liite 3: E-Designer-ohjelman graafisia symboleja	49
Liite 4: GX Works2 -käyttöohjeen lyhyempi versio.....	50

Lyhenteitä

b	Bit, suom. bitti
B	Byte, suom. tavu
BCD	Binary Coded Decimal, suom. binäärikoodattu desimaalilukuesitys
CPU	Central Processing Unit, suom. keskusyksikkö
DW	Double-Word, suom. kaksoissana
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
IB	Input Byte, suom. tulotavu
IL	Instruction List, suom. käskylista
IW	Input Word, suom. tulosana
LD	Ladder Diagram, suom. relekaavio
MC	Master Control, suom. pääkoskettimen asetus
MCR	Master Control Reset, suom. pääkoskettimen nollaus
PLC	Programmable Logic Controller, suom. ohjelmoitava logiikka
POU	Program Organization Unit, suom. rakenneyksikkö
QB	Quit Byte, suom. lähtötavu
QW	Quit Word, suom. lähtösana
R	Reset
REAL	Reaaliluku
SFC	Sequential Function Chart, suom. askelkaavio
S	Set
ST	Structured Text, suom. lausekieli
W	Word, suom. sana

1 Johdanto

Ohjelmoitavien logiikoiden ohjausta ohjelmointiohjelmien avulla käytetään teollisuudessa esimerkiksi koneistamisessa, pakkauksessa, materiaalien käsittelyissä ja automatisoiduissa asennuksissa. Ohjaus tapahtuu ohjelman avulla, joka ladataan logiikkaan.

Työn taustalla on idea, että Tampereen ammattikorkeakoulun pitää pysyä opetuksessaan ajan tasalla teollisuuden kanssa. Tämän johdosta ohjelmia pitää päivittää ja ostaa uudempia versioita. Tässäkin tapauksessa koululle tulee uusi versio Mitsubishin logiikkaohjelmista, jonka nimi on GX Works2. Uusi ohjelma vaatii uuden käyttöohjeen ja tästä johtuen tarve tälle opinnäytetyölle.

Tavoitteena oli tehdä uusille käyttäjille selkeä ja johdonmukainen käyttöohje, jotta he pääsevät nopeasti tekemään haluamiaan töitä. Koska saatavilla on vain vieraskielisiä ohjeita, niin tarve on suomenkieliselle ohjeelle.

Opinnäytetyö on jaoteltu niin, että aluksi kerrotaan vähän teoriataustaa ohjelmoitavista logiikoista ja logiikkaohjelmista ja sen jälkeen käytännössä koko loppuosa työstä on itse käyttöohjetta ohjelman tekemiseen. Teoriaosuudet on pyritty jättämään lyhyiksi, mutta selventäviksi, ja keskitytty pääasiaan eli käyttöohjeeseen.

2 Käyttöohjeen kirjoittaminen

Käyttöohjeen kirjoittaminen on aina haastava työ, koska pitää pystyä miettimään, miten vieras henkilö ymmärtää asioita. Usein saman lauseen voi ymmärtää monella eri tavalla, jos sitä ei ole kirjoitettu tarpeeksi tarkasti, ja tästä voi syntyä tarpeetonta sekaannusta ohjeessa.

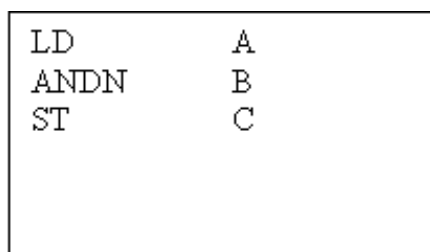
Tässä ohjeessa on pyritty käymään asiat tarkasti kohta kohdalta ja itse ohjelman tekemisen joka vaiheeseen on pyritty laittamaan kuvia tai taulukoita helpottamaan ymmärrystä. Tavoitteena käyttöohjeessa on myös ollut johdonmukainen eteneminen, ettei lukijan tarvitse lukiessaan selata sivuja eteen- tai taaksepäin.

3 IEC 61131-3 -standardi

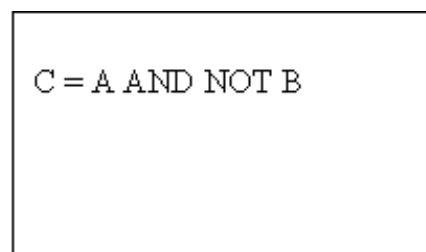
IEC 61131-3 on kansainvälinen ohjelmoitavissa logiikoissa (Programmable Logic Controller, PLC) käytetty standardi, joka määrittelee ohjelmointikielten käyttöehdot. Standardin ensimmäinen versio kehitettiin vuonna 1993 ja toinen versio vuonna 2003. Standardin yksi tärkeä hyöty on, että se sallii useiden ohjelmointikielten käyttämien samassa ohjelmoitavassa logiikassa. Täten ohjelman suunnittelija pystyy valitsemaan oikean kielen eri tehtäviin. (IEC 1131 or 61131: status of the Standard & What is IEC 1131?)

Alalla käytetään seuraavia ohjelmointikieliä, joista kaksi on tekstipohjaista ja kaksi on graafista (Kuva 1):

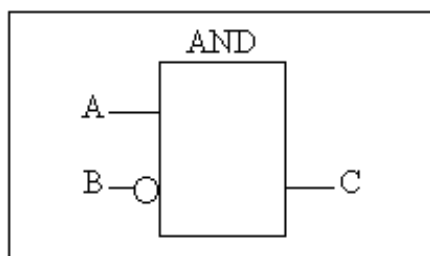
- Käskylista (Instruction List, IL)
- Lausekieli (Structured Text, ST)
- Logiikkakaavio (Function Block Diagram, FBD)
- Relekaavio (Ladder Diagram, LD)



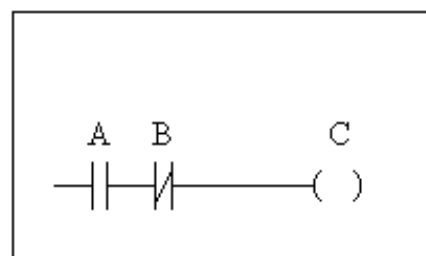
Käskylista



Lausekieli



Logiikkakaavio



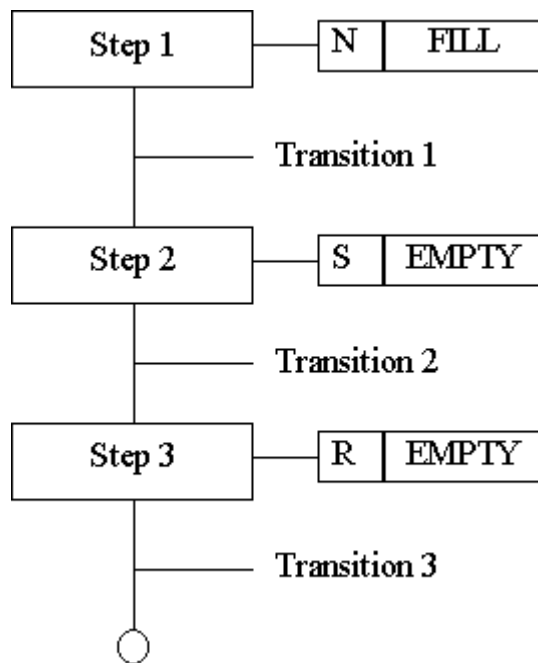
Relekaavio

Kuva 1. Esimerkkitapauksia IEC 61131-3 -standardin ohjelmointikielten käytöstä (Asmala, Hannu 1999)

Ohjelmointikielten yhteydessä voidaan usein mainita myös askelkaaviot (Sequential Function Chart, SFC) (Kuva 2), vaikkei tämä varsinaisesti kuulu standardiin.

Askelkaavio kuvaa graafisesti ohjausohjelman käyttäytymistä sekvensseissä. Sen avulla ohjelma voidaan pilkkoa pienempiin, paremmin hallittaviin osakokonaisuuksiin.

Askelkaaviosta näkee järjestelmän toiminnan. Sekvensseistä voi kutsua toisia sekvenssejä ja niitä voi suorittaa rinnakkain toisistaan riippumatta. (Asmala, Hannu 1999)



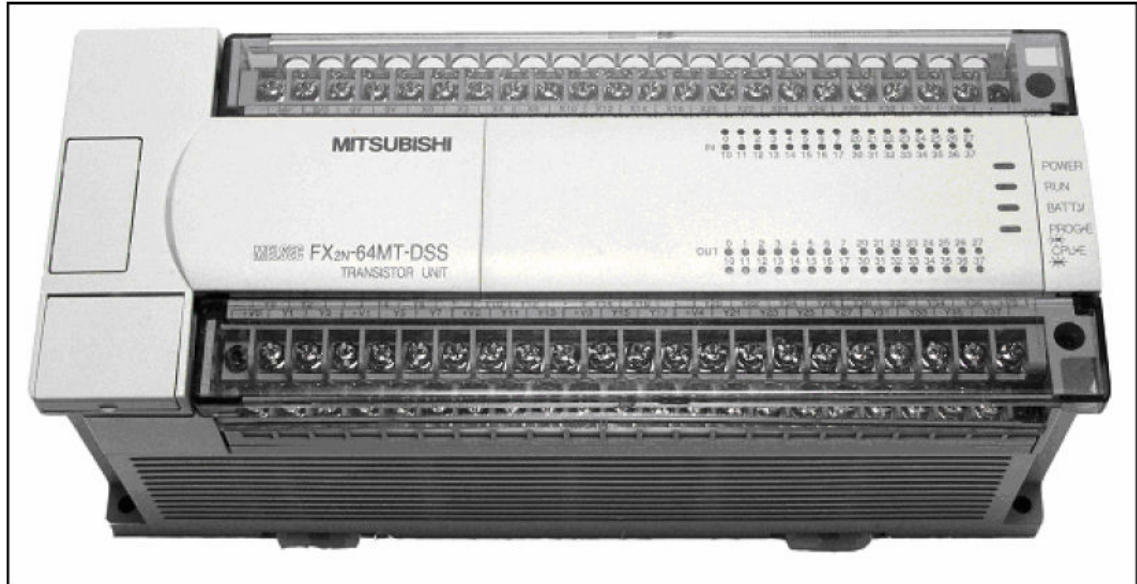
Kuva 2. Esimerkki askelkaavion toimintaperiaatteesta (Asmala, Hannu 1999)

4 GX Works2 -käyttöohje FX2n -logiikan ohjelmointiin

4.1 Ohjelmoitava logiikka

Ohjelmoitava logiikka (Kuva 3) on ohjauslaite, jolla ohjataan lähtöjensä avulla toimilaitteita, kuten sähkömoottoreita, releitä ja merkkilamppuja. Ohjaus tapahtuu ohjelman avulla, joka ladataan logiikkaan. Useimmiten ohjelma tehdään tietokoneessa olevalla ohjelmalla. Ohjelmoitava logiikka toimii seuraamalla logiikan tuloja, ja niiden tilojen perusteella kytkee lähdöt päälle tai pois. PLC:tä hyödynnetään nykyään monella teollisuuden alalla kuten koneistamisessa, pakkauksessa, materiaalien käsittelyissä ja automatisoiduissa asennuksissa. (Automaatiotekniikan perusteet, 2008, 54 & PLCS.net)

Alun perin logiikka kehitettiin releohjauksen korvaajaksi, koska releohjaus ei ollut joustava tuotemuutosten yhteydessä. (Koneautomaatio, Automaatiolaitteet, 1996, 102)



Kuva 3 FX2n-logiikka. (Heinonen, Anu 2006, 12)

PLC-laitteita on olemassa useita erilaisia, ja periaatteellinen ero on niiden valmiudessa suorittaa ohjaustoimintoja. Tikapuuperiaatteella toimivia laitteita kutsutaan askeltaviksi logiikoiksi ja muita vapaasti ohjelmoitaviksi logiikoiksi.

Askeltavat logiikat ovat sekvenssityyppisiin ohjauksiin valmistettuja tietokoneita. Näitä laitteita käytettiin koneautomaatiossa aiemmin korvaamaan pneumaattisia ja releohjattuja logiikkajärjestelmiä. Nykyisin ohjelmoitavat logiikat ovat korvanneet askeltavat logiikat täysin.

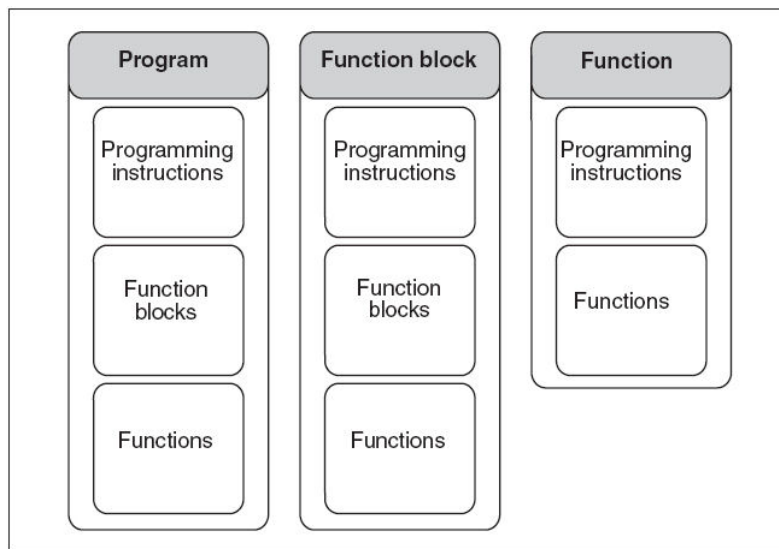
Ohjelmoitava logiikka toimii oleellisena osana ohjelmoitavaa ohjausjärjestelmää. Sen tuloihin kytketään järjestelmän tilaa havainnoivat anturit ja lähestymiskytkimet eli aistit. Sen muistiin voidaan kirjoittaa ohjelma, joka valvoo järjestelmän tilaa tosiaikaisesti. Tämä on merkittävä ero askeltavaan logiikkaan verrattuna. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2000, 243)

4.2 Syklinen ohjelmankäsittely

Automaatiolaitteen ohjausyksikkö käsittelee muistiin ohjelmoidut ohjauskäskyt yksitellen peräkkäin. Käsiteltyään viimeisen muistissa olevan käskyn ohjausyksikkö aloittaa jälleen alusta ja käsittelee ensimmäisen käskyn. Nimitys syklinen ohjelmankäsittely johtuu juuri siitä, että käskyjen käsittely toistuu jatkuvasti, toisinaan laite toimii pyyhkäisyperiaatteella. Aikaa, joka kuluu ohjelman kaikkien käskyjen käsittelyyn kertaalleen, kutsutaan kierrosajaksi tai sykliajaksi. Automaatiolaitteessa on kierrosajan valvonta, joka keskeyttää ohjelman ja automaatiolaite pysähtyy, mikäli sykli aika ylittää asetellun valvonta-ajan. (Koneautomaatio, Sähköiset automaatiolaitteet, 1990, 188)

4.3 POU ja Task

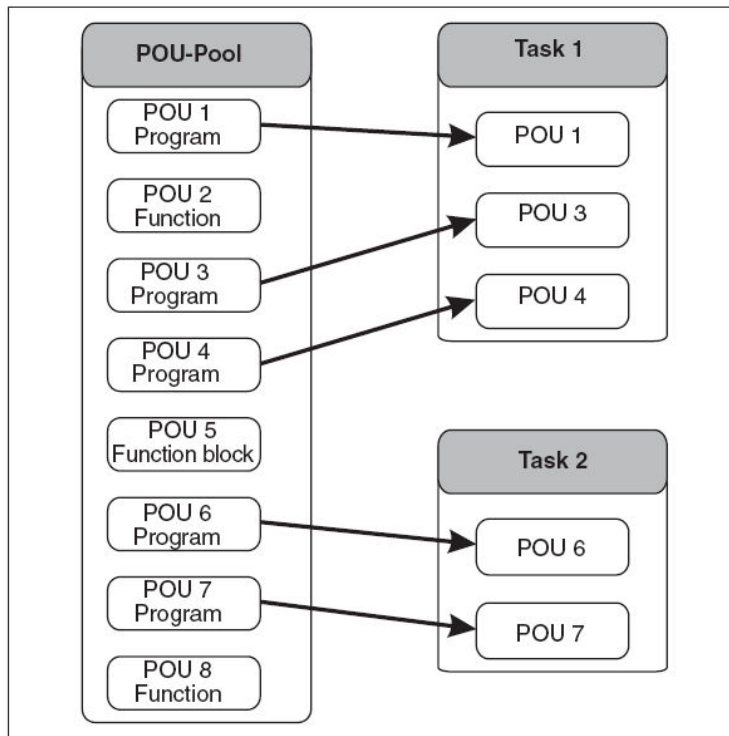
Nykyaikaisessa rakenteellisessa ohjelmoinnissa ohjelmat on järjestetty erillisiin ohjelmamoduuleihin. Näitä moduuleja kutsutaan yleisemmin ohjelman rakenneyksiköiksi (Program Organization Unit, POU). Rakenneyksiköt on jaettu kolmeen eri luokkaan toimintojensa perusteella; ohjelmiin (Programs, PRG), funktioihin (Functions, FUN) ja toimilohkoihin (Function Blocks, FB) (Kuva 4). Jokainen POU sisältää muuttujalistan (Header) ja ohjelman rakenteen (Body).



Kuva 4. POU:n eri luokat (GX IEC Developer 7.01, Beginner's Manual, 2006, 3 - 3)

Lopullinen ohjelma koostuu rakennyksiköistä, jotka on määritelty ohjelmiksi (Kuva 5).
Samaan ryhmään koottuja rakennyksikköjä kutsutaan nimellä Task.

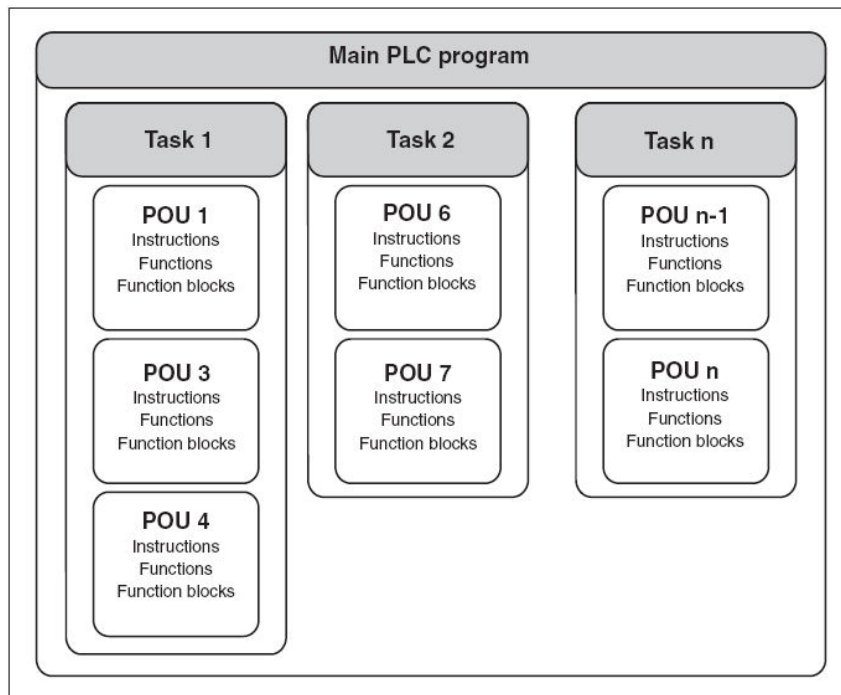
Yksi Task sisältää yhden tai useamman POU-rakennyksikön ohjelman (PRG). Task ohjaa POU-ohjelmien suorittamista. (GX IEC Developer 7.01, Beginner's Manual, 2006, 2 - 2)



Kuva 5. POU-ohjelmat jaoteltuina Task-ryhmiin (GX IEC Developer 7.01, Beginner's Manual, 2006, 2 - 3)

4.4 Execution program

Execution program, eli TASK vanhalta nimeltään, sisältää ohjelman kaikki Taskit, rakenneyksiköt ja niiden suoritusjärjestyksen (Kuva 6).



Kuva 6. Kokonaisuus ohjelmarakenteesta (GX IEC Developer 7.01, Beginner's Manual, 2006, 2 - 3)

4.5 GX Works2 -ohjelma

4.5.1 Yleiskuva

GX Works2 on uusien FX- ja Q-logiikoiden ohjelmointiohjelmisto, joka korvaa GX Developer ja GX IEC Developer -ohjelmat. Se on 32-bittinen Windows-pohjainen ohjelmisto sovellusten ohjelmointiin, simulointiin, monitorointiin, vianmääritykseen ja dokumentointiin. Ohjelmistoa on parannettu tuottavuuden kasvattamiseksi ja suunnittelukustannusten alentamiseksi. Ohjelmistossa on paljon uusia hyödyllisiä toimintoja, kuten esimerkiksi logiikkasovellusten versionhallinta. (Beijer Electronics Automation - iQ Works & Beijer Electronics Automation - GX IEC Developer)

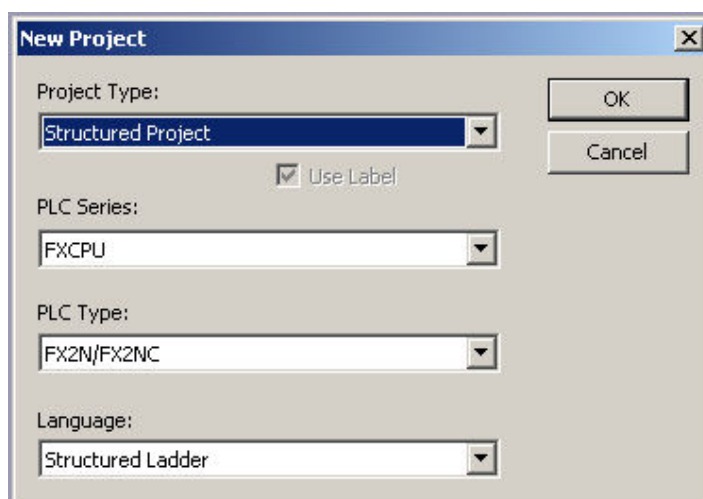
GX Works2 -ohjelma mahdollistaa viiden eri ohjelmointikielen valinnan, jolloin suunnittelija pystyy valitsemaan sopivimman kielen tarvitsemaansa työhön. Valittavissa olevat ohjelmointikielet ovat käskylistä (IL), lausekieli (ST), logiikkakaavio (FBD), relekaavio (LD) ja askelkaavio (SFC). Ohjelmointikielet perustuvat IEC 61131-3 -standardiin.

4.5.2 Ohjelman käynnistys

Ohjelman saa käyntiin Windowsin työpöydältä löytyvältä GX Works2 -ikonilla tai start-valikon alta. Eli tarkemmin sanottuna Start -> All Programs -> MELSOFT Application -> GX Works2 -> GX Works2.

4.5.3 Uuden projektin luominen

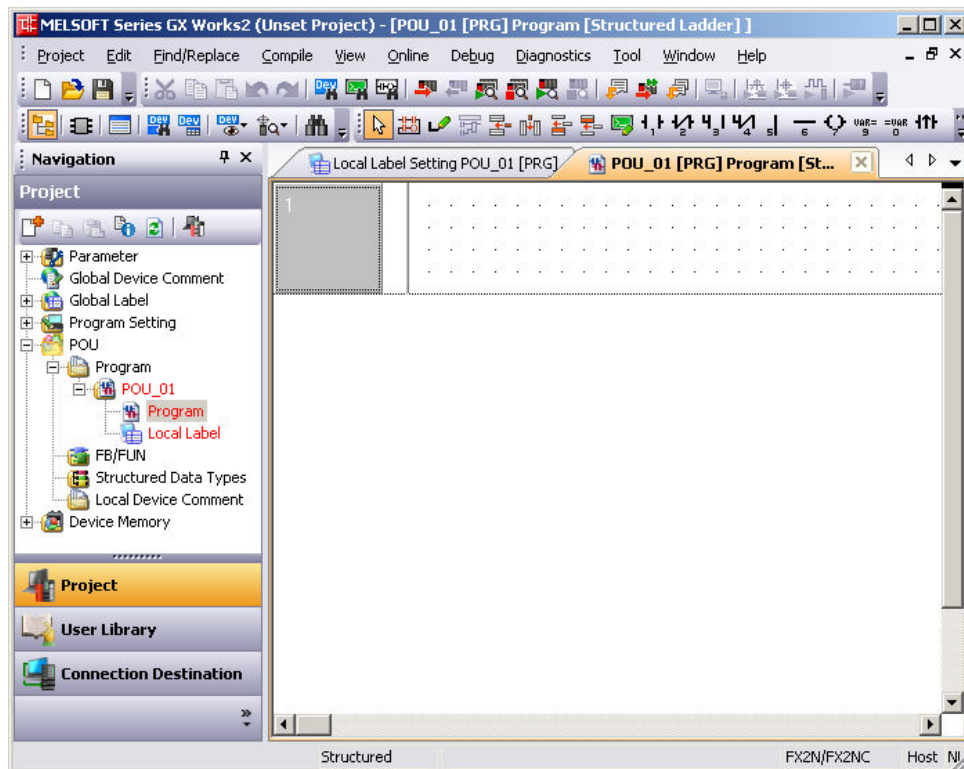
Uusi projekti tehdään valitsemalla Project-valikosta New. Ikkuna aukeaa (Kuva 7), jossa kysytään projektityyppiä, logiikkasarjaa, logiikkatyyppiä ja kieltä. Tehdään valinnat kuvan mukaisesti ja painetaan OK.



Kuva 7. Uuden projektin luonti-ikkuna

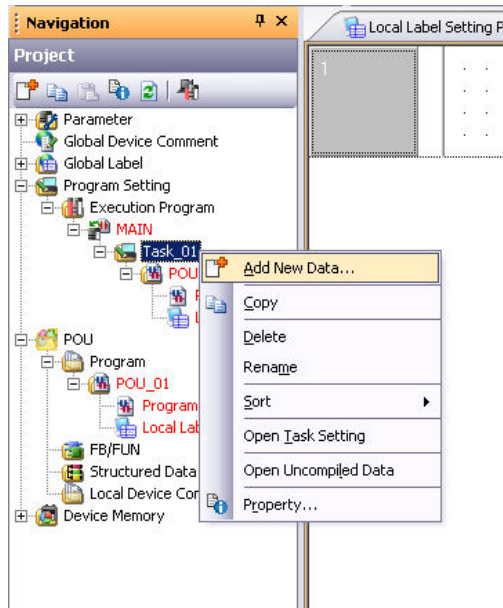
4.5.4 Ohjelman alustaminen

Kun uusi projekti on luotuna (Kuva 8), aloitetaan ohjelman alustaminen. Kuvan vasemmalla näkyy projektipuu, ylhäällä työkalurivi ja keskeltä löytyvät valitun rakenneyksikön (Programmable Organization Unit, POU) ohjelman rakenne ja muuttujalista.



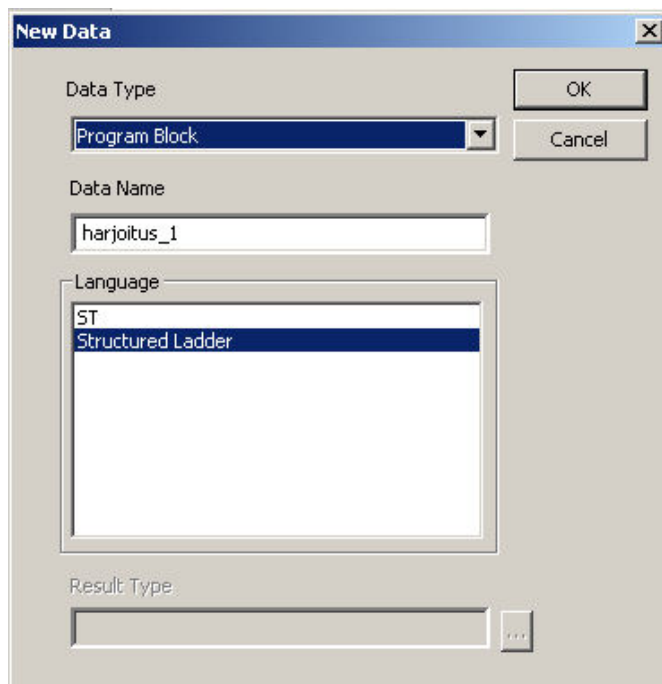
Kuva 8. Uusi projekti on luotuna

Ensimmäiseksi luodaan uusi rakenneyksikkö. Avataan vasemmalta projektipuusta Program Setting -puu kokonaan. Painetaan oikeaa hiiren nappia Task_01:ssä ja valitaan Add New Data (Kuva 9).



Kuva 9. Uuden datan lisääminen

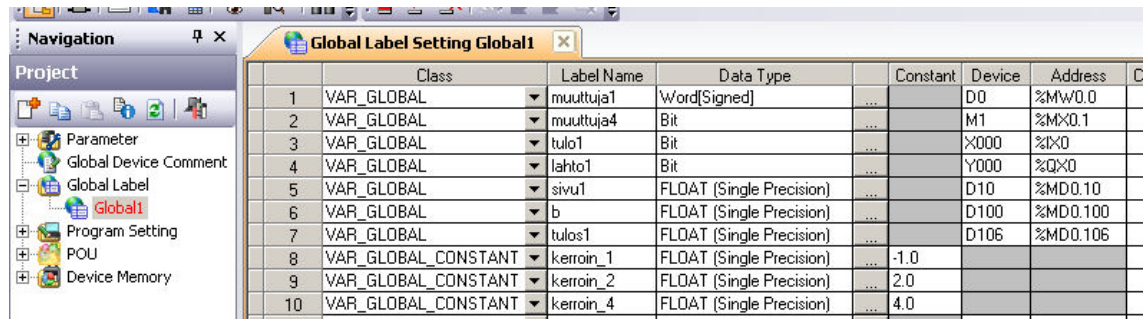
Aukeaa valintaikkuna, jossa kysytään datatyyppiä, työn nimeä ja kieltä. Valitaan kuvan 10 mukaisesti datatyyppiksi Program Block ja kieleksi Structured Ladder. Työn nimen voi valita oman mielen mukaan, ja sitten painetaan OK.



Kuva 10. Uuden datatyyppin valintaikkuna

4.5.5 Yleismuuttujien listaaminen

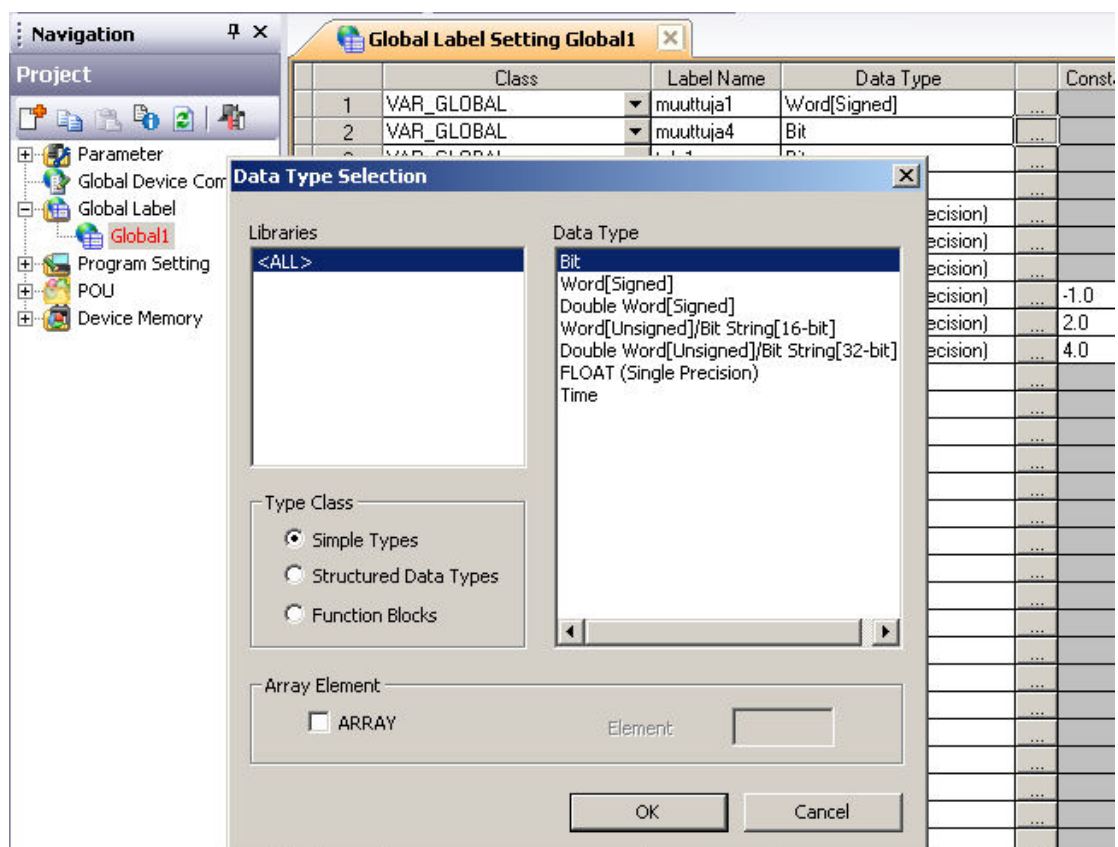
Yleismuuttujien taulukko avataan vasemmalta rakennepuusta Global Label -> Global1 -kohdasta (Kuva 11).



	Class	Label Name	Data Type	Constant	Device	Address	C
1	VAR_GLOBAL	muuttuja1	Word(Signed)	...	D0	%Mw0.0	
2	VAR_GLOBAL	muuttuja4	Bit	...	M1	%Mx0.1	
3	VAR_GLOBAL	tulo1	Bit	...	X000	%Ix0	
4	VAR_GLOBAL	lahto1	Bit	...	Y000	%Qx0	
5	VAR_GLOBAL	sivu1	FLOAT (Single Precision)	...	D10	%MD0.10	
6	VAR_GLOBAL	b	FLOAT (Single Precision)	...	D100	%MD0.100	
7	VAR_GLOBAL	tulos1	FLOAT (Single Precision)	...	D106	%MD0.106	
8	VAR_GLOBAL_CONSTANT	kerroin_1	FLOAT (Single Precision)	... -1.0			
9	VAR_GLOBAL_CONSTANT	kerroin_2	FLOAT (Single Precision)	... 2.0			
10	VAR_GLOBAL_CONSTANT	kerroin_4	FLOAT (Single Precision)	... 4.0			

Kuva 11. Globaalilista avattuna

Taulukosta valitaan ensimmäiseksi yleismuuttujat (Class) pudotusvalikosta. Seuraavaksi listataan Label Name -kohtaan tarvittavien muuttujien nimet, kuten esimerkiksi tulot ja lähdöt. Datatyyppin valintaikkuna saadaan auki kuvan xx mukaisesti ...-painikkeesta. Yksinkertaiset datatypit löytyvät Simple Types -valikon alta, ja laskurit ynnä muut löytyvät Function Blocks -valikosta (Kuva 12).



Kuva 12. Simple Types -datatyyppien valinta

Jos muuttuja tarvitsee vakioluvun, se lisätään Constant-sarakkeelle. Logiikan muistipaikka laitetaan Device-sarakkeelle. Osoitteen pitäisi täyttyä tämän jälkeen automaattisesti.

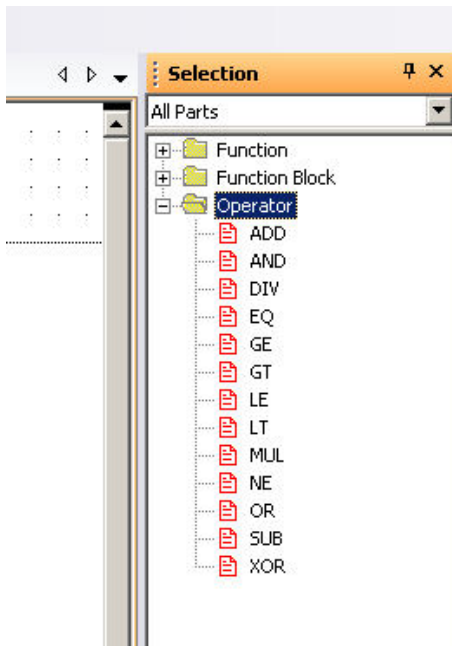
4.5.6 Ohjelman luominen

Tämän jälkeen voidaan aloittaa itse ohjelman tekeminen. Tarvittavat operaattorit löytyvät Function Block Selection Window -painikkeella (Kuva 13).



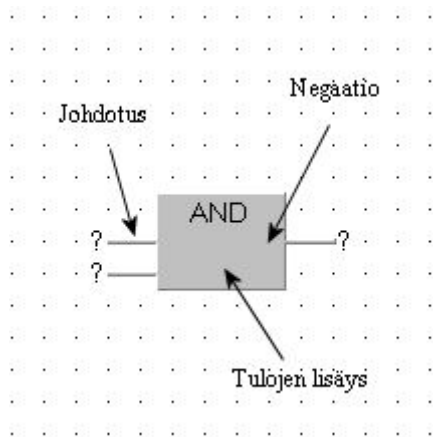
Kuva 13. Function Block Selection Window -painikkeen sijainti

Ruudun oikeaan reunaan avautuu uusi valikko (kuva 14), josta löytyvät kaikki tarvittavat operaattorit virtapiirien tekoon, kunhan valikosta on valittuna All Parts. Yleisimmin tarvittut perusoperaattorit löytyvät Operator-kansiosta.



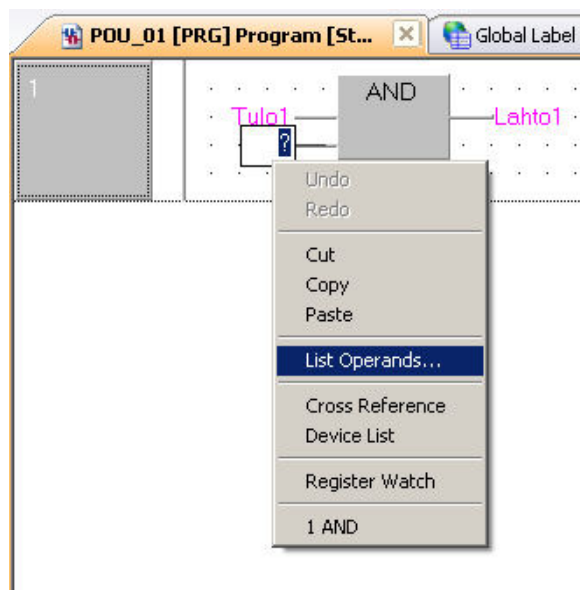
Kuva 14. Perusoperaattorit Function Block Selection Windows -valikossa

Operaattorit saadaan virtapiirille joko tuplaklikkaamalla tai raahaamalla. Johdotukset onnistuvat työkaluriviltä löytyvällä Interconnect Modella (CTRL+T). Painetaan kerran vasemmalla napilla kohtaan, josta johto vedetään, ja toisen kerran vasenta nappia kohdassa, johon johto päättyy. Negaation saa tuplaklikkaamalla kuvan 15 mukaisesti. Operaattoriin saa lisättyä tuloja painamalla ensin vasemmalla napilla operaattori aktiiviseksi ja sitten vetämällä sen alareunasta.



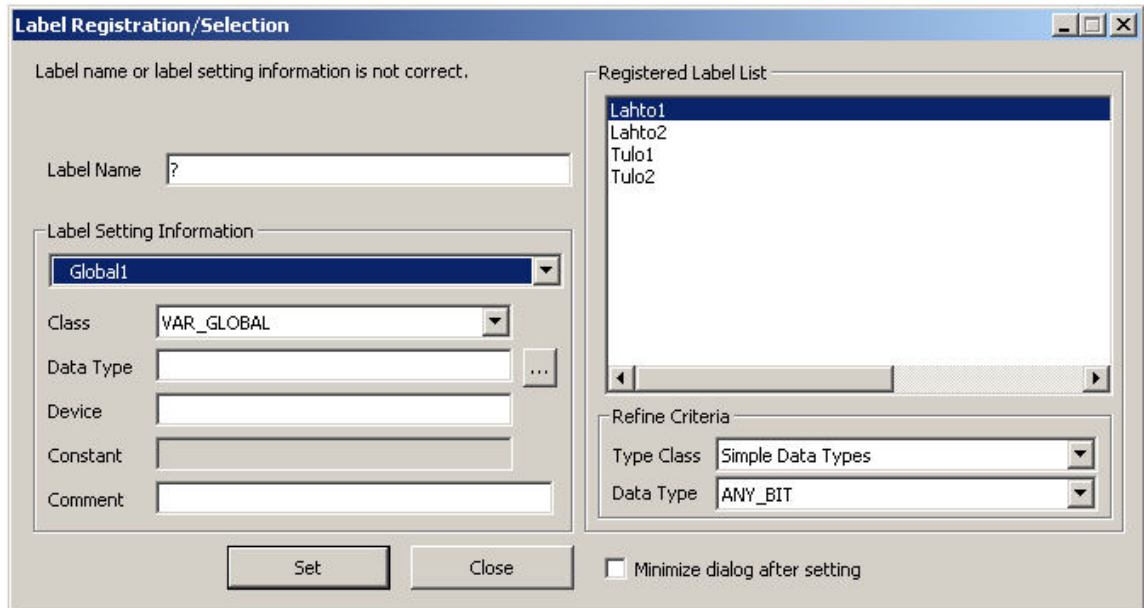
Kuva 15. Operaattorin eri toiminnot

Muuttujien lisääminen tehdään painamalla vasemmalla napilla kysymysmerkin kohdalta ja tämän jälkeen oikealla napilla laatikon sisältä. Tästä aukeaa valintaikkuna, josta otetaan List Operands (Kuva 16).



Kuva 16. Opastus muuttujien löytämiseen

Muuttujat valitaan listasta, jonka saa näkyviin, kun valitsee Label Setting Information -kohtaan Global1 (Kuva 17). Listasta voidaan tämän jälkeen etsiä haluttu muuttuja, ja se otetaan käyttöön painamalla Set ja Close.



Kuva 17. Muuttujien valinta listasta

4.5.7 Aputoiminta

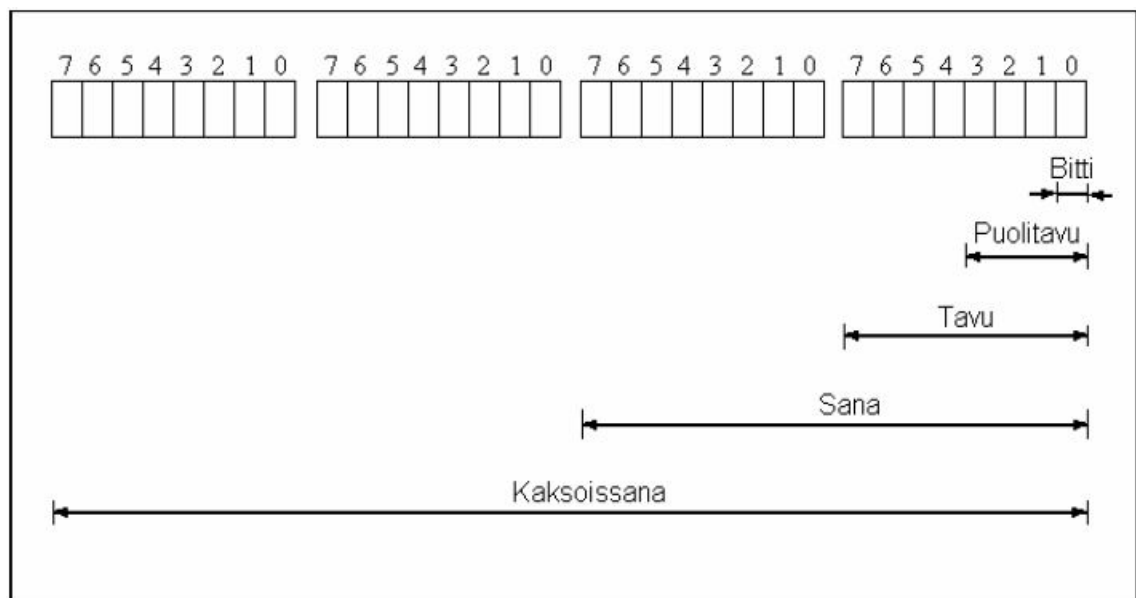
GX Works2 -ohjelma sisältää kattavan Help-valikon, jossa on kerrottuna virhekoodeja ja tarkempia neuvoja ohjelmien tekoon.

4.6 Osoitteet

4.6.1 Bitti, tavu, puolitavu, sana ja kaksoissana

Bitti (Bit, b) voi saada tilat 0 tai 1, ja se on pienin tietoalkio, jota automaatiojärjestelmät käsittelevät (Kuva 18). Kahdeksan peräkkäisen bitin kokonaisuutta sanotaan tavuksi (Byte, B). Logiikassa yhdistetään esimerkiksi kahdeksan tulon tai lähdön signaalit yhdeksi tulotavuksi (IB) tai lähtötavuksi (QB). Jokainen yksittäinen tavun bitti (binääripaikka) voi saada arvon 0 tai 1. Logiikassa käsitellään usein koko tavu tai sana, siis kaikki 8 tai 16 bittiä yhdessä.

Puolitavulla (Nibble), tarkoitetaan neljän bitin ryhmää eli puolikasta tavua. Kahden tavun kokonaisuutta kutsutaan sanaksi (Word, W) ja kahden sanan kokonaisuutta kaksoissanaksi (Double-Word, DW). Logiikassa voidaan yhdistää esimerkiksi 16 tulon tai 16 lähdön signaalit tulosanaksi (IW) tai lähtösanaksi (QW). (Automaatiolaitteet, Koneautomaatio, 1996, 149)



Kuva 18. Binäärijärjestelmä (Heinonen, Anu 2006, 26)

4.6.2 I/O-osoitteet

GX Works2 -ohjelmassa muuttujille pitää antaa logiikan muistipaikkoja vastaavat osoitteet, että toimilaitteita pystytään ohjaamaan. Ohjelma määrittää osoitteet itse ellei niitä määritetä. Muuttujat määritellään POU-rakenneyksikössä, jotta logiikan keskusyksikön (CPU) muistipaikat saadaan käyttöön. Osoitteet voidaan myös kirjoittaa viittaamaan johonkin fyysiseen tulo- ja lähtöpaikkaan. (Heinonen, Anu 2006, 26)

4.6.3 Apumuistit

Apumuistit ovat mekaanisen releen kaltaisia, logiikan sisäisiä muistipaikkoja. Niitä voidaan myös kutsua sisäisiksi releiksi, lipuiksi tai merkkereiksi. Tässä tapauksessa niitä kutsutaan apumuisteiksi. Apumuisteilla voi olla vain kaksi tilaa, varattuna (1) ja ei käytössä (0).

Apumuisteja käytetään tiedontallennuspaikkana, kun jokin asia halutaan pistää muistiin. Näitä käytetään esimerkiksi kappaleiden lajittelun ohjauksessa ja laskentatehtävissä. Erikoismuistipaikat ovat apumuistipaikkoja, jotka on valmiiksi tallennettu käyttäjää varten nopeampaa käyttöä varten. GX Works2 -ohjelmasta löytyy Help-> Special Relay/Special Register -> FX series PLC -kohta, jossa on lueteltuna kaikki erikoismuistipaikat. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 251-252)

4.6.4 Virhekoodit

GX Works2 -ohjelman Help-valikon CPU Error -kohdasta pystyy etsimään selvennyksen ja korjausehdotuksia eri virhekoodeille.

4.7 Luvut

4.7.1 BCD-luvut

Binäärikoodattuja desimaalilukuja (Binary Coded Decimal, BCD) käytetään esimerkiksi taskulaskimien numeronäytöissä. BCD-luvut perustuvat lukujen 0-9 esittämiseen neljällä bitillä. Esittämistapa toimii siis samalla tavalla kuin binääriluvuilla. Esimerkiksi BCD-luku 0010 vastaa desimaalilukua 2 ja BCD-luku 0101 vastaa desimaalilukua 5. Useamman luvun ilmoittaminen tapahtuu vain laittamalla kaksi neljän bitin lukua peräkkäin. Esimerkiksi desimaaliluku 25 on BCD-lukuna 0010 0101 eli 2 ja 5. Taulukossa 1 luvut on listattuna tarkemmin. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 65-66)

Taulukko 1. BCD-lukujen muunnokset

Desimaaliluku	BCD-luku
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010
953	1001 0101 0011
8274	1000 0010 0111 0100

4.7.2 Datatyypit

Muuttujan datatyyppi määrittelee, miten sitä käsitellään, muuttujan lukualueen ja montako bittiä datatyyppi sisältää. GX Works2 -ohjelmassa on mahdollisuus valita datatyyppiä taulukon 2 mukaisesti.

Taulukko 2. Valittavissa olevat datatyypit (GX IEC Developer 7.01, Beginner's Manual, 3 - 8)

Datatyypit		Koko (bittejä)
BOOL	Boolean	1
INT	Integer	16
DINT	Double integer	32
WORD	Bit string 16	16
DWORD	Bit string 32	32
REAL	Floating-point value	32
TIME	Time value	32
STRING (vain Q-sarjassa)	Character string	Maksimi 50 merkkiä.

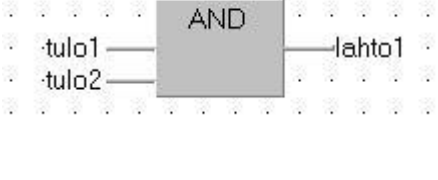
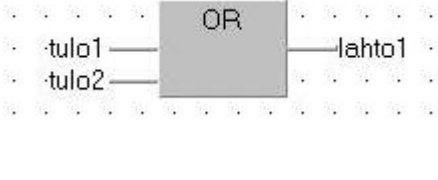
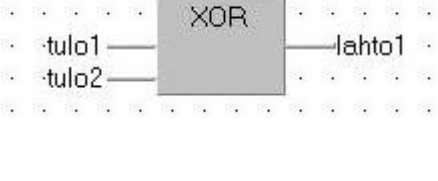

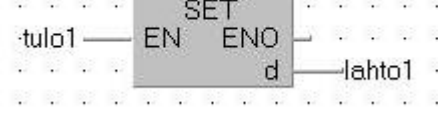
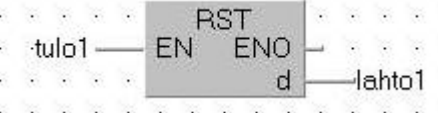
4.8 Operaattorit

Ohjelmien luomiseen tarvitaan erilaisia operaattoreita, ja seuraavassa on koottuna GX Works2:n yleisimmin käytetyt operaattorit ja selitykset näiden toimintaperiaatteista.

4.8.1 Perusoperaattorit

Perusoperaattorit (Taulukko 3) löytyvät Function Block Selection Window -napin takaa. AND-, OR- ja XOR-operaattorit löytyvät Operator-kansiosta ja loput kolme Function-kansiosta.


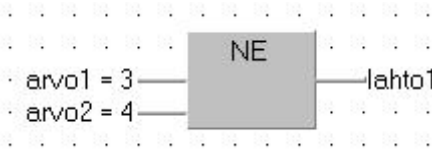

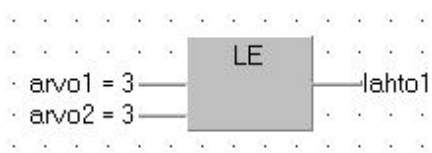
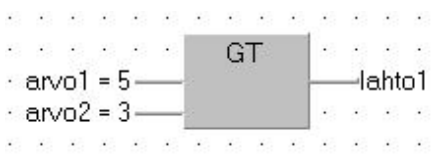

Taulukko 3. Perusoperaattorit ja niiden toimintaperiaate

Käsky	Kuvaus															
<p>JA-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</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>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	0	0	1	0	1	1	1
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	0														
0	1	0														
1	1	1														
<p>TAI-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</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>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	1	0	1	1	1	1	1
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	1														
0	1	1														
1	1	1														
<p>EHDOTON TAI -käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</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>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	1	0	1	1	1	1	0
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	1														
0	1	1														
1	1	0														
<p>EI-käsky eli invertteri</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	Tulo1	Lahto1	0	1	1	0									
Tulo1	Lahto1															
0	1															
1	0															
<p>Set- eli asetuskäsky</p> 	<p>Kun tulo1 saa arvon 1, menee lahto1 päälle eli saa myös arvon 1.</p>															
<p>Reset- eli nollauskäsky</p> 	<p>Kun tulo1 saa arvon 1, menee lahto1 pois päältä eli saa arvon 0.</p>															

4.8.2 Vertailijat

Vertailijat (Taulukko 4) löytyvät perusoperaattoreiden tapaan Function Block Selection Window -napin avulla. Kaikki vertailijaoperaattorit löytyvät Operator-kansiosta.

Taulukko 4. Vertailijaoperaattorit ja niiden toimintaperiaatteet (Heinonen, Anu 2006, 32)


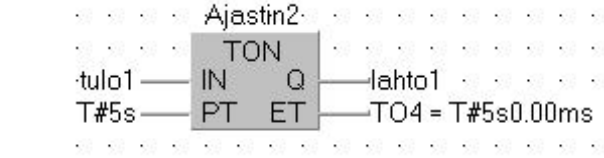
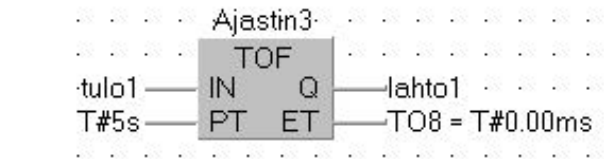
Käsky	Kuvaus
	<p>Yhtäsuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa yhtä suuret lähtö1 on päällä.</p>
	<p>Erisuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa eri suuret lähtö1 on päällä.</p>
	<p>Pienempi kuin -käsky</p> <p>Kun arvo1 on pienempi kuin arvo 2, lähtö1 on päällä.</p>
	<p>Pienempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo1 on pienempi tai yhtä suuri kuin arvo2, lähtö1 on päällä.</p>
	<p>Suurempi kuin -käsky</p> <p>Kun arvo1 on suurempi kuin arvo2, lähtö1 on päällä.</p>
	<p>Suurempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo1 on suurempi tai yhtä suuri kuin arvo2, lähtö1 on päällä.</p>

4.8.3 Ajastimet

Ohjelmoitavien logiikoiden ajastimet ovat yleensä vetohidasteisia, eli kello lähtee käyntiin jollakin tuloehdolla. Ajastimilla voidaan prosessiin lisätä viiveitä, jotta tietty työsuoritus ehtii tapahtua. Eräänä esimerkkinä on tavaratalojen automaattiovien automatisointi. Ajastimien käskyrakenne vaihtelee eri logiikkamerkkien välillä. Jotkin käyttävät niiden ohjelmointiin apumuistipaikkoja, kun taas joissakin ohjelmointiin tarvitaan vain parin rivin ohjelmarutiini. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 252-253)

Ajastimet (Taulukko 5) löytyvät Function Block -kansioista Function Block Selection Window -napin takaa.

Taulukko 5. Muutama esimerkki ajastimien toiminnasta (Heinonen, Anu 2006, 33)

Ajastin	Kuvaus
	<p>Coil = ajastimen käynnistys Preset = ajastettava arvo (> ValueIn) ValueIn = lähtöarvo (yleensä 0) ValueOut = näyttää ajan Status = ilmoittaa kun aika on kulunut</p>
	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO4) Q = ilmoittaa kun aika on kulunut</p>
	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO8) Q = ilmoittaa kun aika on kulunut</p>

4.8.4 Laskurit

Laskureilla voidaan esimerkiksi hallita kappaletavaran määrää. Myös virvoitusjuomapullojen laskenta täyttölinjalla on tästä tyyppiesimerkki. Pysäköintitalon ajoneuvojen laskenta voidaan automatisoida myös käyttämällä logiikan laskuriominaisuutta. Tavallisissa laskureissa on kaksi tuloa: nollaustulo ja laskuritulo. Laskuritulon etureuna lasketaan ja verrataan sitä nimettyyn esiasetettuun arvoon. Lähtö pysyy nollana niin kauan, kunnes laskettu arvo on suurempi kuin esiasetettu arvo. Tämän jälkeen lähtö asettuu ykköseksi. Laskuri resetoidaan eli palautetaan alkuun jälleen nollaustulolla. Laskurit ovat yleensä ylhäältä alaspäin laskevia, ja laskurin väliarvoja voidaan tallentaa apumuisteihin aritmeettisten laskusuoritusten ja vertailujen suorittamiseksi. Tähän käytetään yleensä FUN-komentoja. Logiikoissa on yleensä myös kolmitulaisia ylös-alas-laskureita. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 253)

Myös laskurit (Taulukko 6) löytyvät Function Block Selection Window -napilla ja sen alta Function Block -kansiosta.

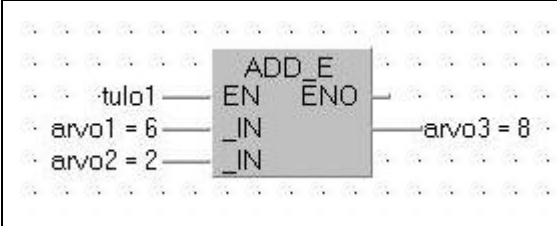
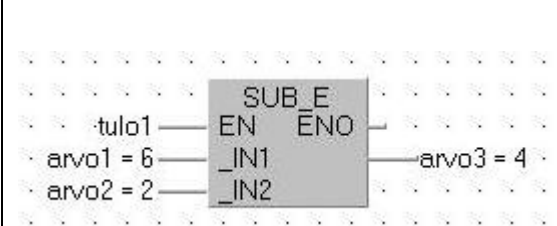
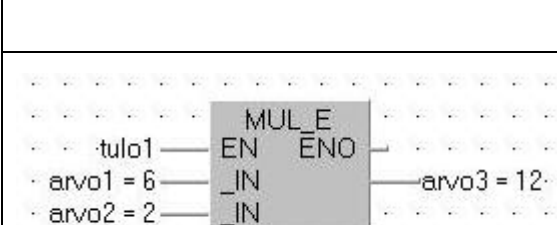
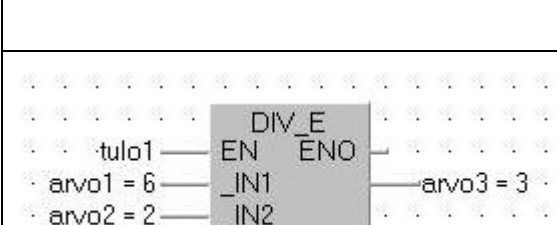
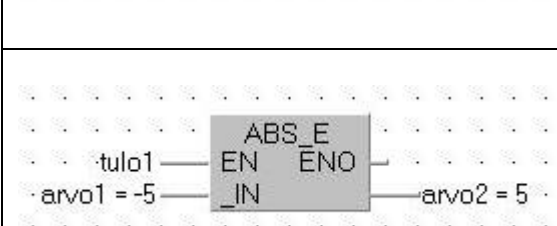
Taulukko 6. Laskurien toimintaperiaatteet (Heinonen, Anu 2006, 35)

Laskuri	Kuvaus
<pre> Laskuri1 CTU_E tulo1 — EN ENO — tulo2 — CU Q — lahto1 tulo3 — RESET CV — arvo2 = 5 arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) kasvaa tulolla 2, kun tulo1 on päällä.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtä suuri kuin esiasetusarvo (PV).</p> <p>Laskuriarvo resetoidaan tulolla 3.</p>
<pre> Laskuri2 CTD_E tulo1 — EN ENO — tulo2 — CD Q — lahto1 tulo3 — LOAD CV — arvo2 = 0 arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) pienenee tulolla 2.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtä suuri kuin nolla.</p> <p>Esiasetusarvo (PV) saadaan ladattua tulolla, kun tulot 1 ja 2 on kytketty päälle.</p>
<pre> Laskuri3 CTUD_E tulo1 — EN ENO — tulo2 — CU QU — lahto1 tulo3 — CD QD — lahto2 tulo4 — RESET CV — arvo2 = 3 tulo5 — LOAD arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) kasvaa tulolla 2 ja pienenee tulolla 3.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtä suuri kuin esiasetusarvo (PV).</p> <p>Lähtö2 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtä suuri kuin nolla.</p> <p>Esiasetusarvon lataamisessa tulon 5 avulla tulee tulojen 1, 2 ja 3 olla päällä.</p>

4.8.5 Aritmetiikkakäskyt

Taulukon 7 aritmetiikkakäskyt eli laskentakäskyt löytyvät Function-kansiosta. Laskentakäskyjen käytössä pitää muistaa, että molempien puolien arvojen pitää olla samaa datatyyppiä eli esimerkiksi $REAL + REAL = REAL$, jolloin reaaliluku on datatyyppinä. (Heinonen, Anu 2006, 36)

Taulukko 7 Aritmetiikkakäskyt selityksineen

Käsky	Kuvaus
	<p>Yhteenlasku</p> <p>Kun tulo1 saa arvon 1, lasketaan arvot 1 ja 2 yhteen.</p>
	<p>Vähennyslasku</p> <p>Kun tulo1 saa arvon 1, vähennetään _IN1 arvo _IN2 arvosta.</p>
	<p>Kertolasku</p> <p>Kun tulo1 saa arvon 1, kerrotaan arvot 1 ja 2 keskenään.</p>
	<p>Jakolasku</p> <p>Kun tulo1 saa arvon 1, jaetaan _IN1 arvo _IN2 arvolla.</p>
	<p>Vastaluku</p> <p>Kun tulo1 saa arvon 1, muutetaan arvo1 sen vastaluvuksi.</p>

4.8.6 Pääohjaustoiminto

Kun pääohjaustoiminto (Master Control Set/Reset, MC/MCR) asetetaan päälle, se katkaisee ohjelman luvun. Tämä vastaa fyysisen pääkytkimen käyttöä, paitsi että sähkököt säilyvät päällä. Resetointi palauttaa ohjelman luvun ja toiminta jatkuu siitä, mihin se MC-komennolla jäi. Komentoa voidaan käyttää ohjelmallisen hätä-seis-kytkimen ohjauksessa. Paikallinen lukitusvirtapiiri eli hätäpysäytyspiiri tulee tehdä tämän lisäksi aina ilman ohjelmoitavaa logiikkaa. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 255)

MC- ja MCR-operaattori löytyy Function-kansiosta, jonka saa näkyviin Function Block Selection Window -nappia painamalla.

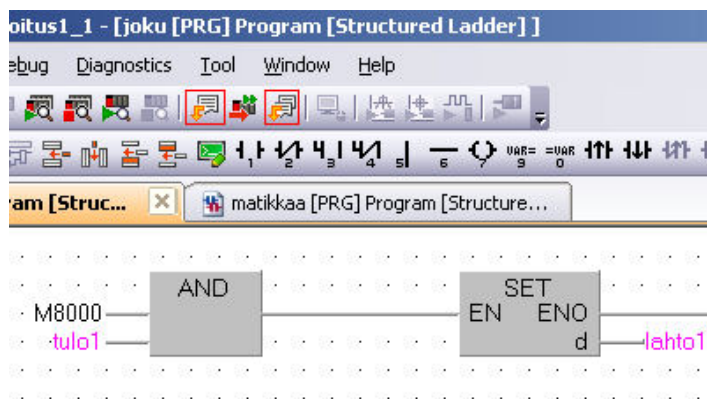
4.9 Simulointi

Valmista ohjelmaa pystytään simuloimaan offline-tilassa, jolloin ei tarvitse fyysisesti olla logiikan vierellä. Ennen simulointia ohjelma pitää kuitenkin aina kääntää.

4.9.1 Projektin kääntäminen

Projektin kääntäminen tehdään Build ja Rebuild all -nappien avulla (Kuva 19). Build-komento kääntää vain kääntämättömät ohjelmaosiot Execution Program -puusta.

Rebuild all kääntää kaiken eli myös jo ennestään käännetyt.



Kuva 19. Build ja Build All -nappien sijainti

Ohjelma kysyy tämän jälkeen kääntämiseen varmistusta, johon vastataan Yes.

Tulokset ja mahdolliset virheet ohjelma ilmoittaa alhaalla löytyvässä Output-ikkunassa. Virheiden sattuessa tarkistetaan ja korjataan vika.

4.9.2 Offline-simulaattori

Ohjelman simulointi saadaan päälle yläpalkin Start/Stop Simulation -napista tai ylävalikosta Debug -> Start/Stop Simulation (Kuva 20).



Kuva 20. Start/Stop Simulation -napin sijainti

4.10 Ohjelman lataaminen logiikalle

Ohjelma ladataan logiikalle Write to PLC -napilla, joka löytyy yläpalkista tai valikosta Online -> Write to PLC (Kuva 21).



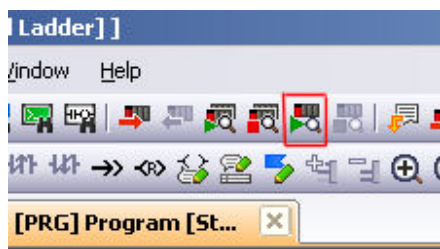
Kuva 21. Write to PLC -napin sijainti

Online Data Operation -ikkuna aukeaa, ja siihen valitaan oikea logiikka ja haluttu projekti. Ladattavat ohjelmat valitaan ruksaamalla Target-kohta halutun ohjelman riviltä. Kun valinnat ovat tehtyinä, painetaan Execute-nappia ja ohjelma latautuu logiikalle. Jos logiikassa oli ladattuna jo jokin ohjelma, aukeaa varmistusikkuna, johon kuitataan Yes to all, jolloin olemassa olevan ohjelman päälle kirjoitetaan. Ikkunat voi sulkea, kun ohjelma on ladattuna logiikalle.

4.11 Ohjelman monitorointi

Ohjelman monitorointi tapahtuu valitsemalla haluttu ohjelma tuplaklikkaamalla POU -> Program -> POU_01 -> Program, jolloin ohjelman ikkuna aukeaa.

Tämän jälkeen valitaan yläpalkista Start Monitoring tai valikosta Online -> Monitor -> Start Monitoring (Kuva 22).



Kuva 22. Start Monitoring -napin sijainti

Asetetaan ohjelmasta ohjattava CPU RUN-tilaan. Myös PLC:n RUN/STOP-kytkin asetetaan RUN-tilaan.

Ohjelman monitoroinnin lopetus tapahtuu päinvastaisesti käynnistykseen eli asetetaan logiikka ja ohjelma STOP-tilaan. Yläpalkista löytyy Stop Monitoring -nappi tai vaihtoehtoisesti valikosta Online -> Monitor -> Stop Monitoring.

5 Loppusanat

Tästä opinnäytetyöstä on tuloksena luvun 4 käyttöohje. Itse käyttöohje onnistui hyvin ja näyttää selkeältä, koska tekstin seassa on paljon kuvia ja taulukoita. Vaikka välillä työtä tehdessä tuntui olevan kiire, että ehdin saada työn valmiiksi ajallaan, niin silti lopputulos on tyydyttävä. Sivuja odotin ehkä 5 - 10 enemmän, mutta kun näen lopputuloksen, niin sivumäärä tuntuu ihan sopivalta. Mahdollisesti kuukauden lisääjalla työhön olisin voinut lisätä tekstiä teoriaosuuteen ja laajentaa käyttöohjetta esimerkkitapauksilla. Myös joitakin ohjeen kohtia olisin voinut käydä vieläkin tarkemmin vaihe vaiheelta läpi.

Tiedon löytäminen Internetin kautta osoittautui yllättävän vaikeaksi, jos ei tiedä suoria osoitteita hyvillä sivuilla. Hakukoneella etsiminen tuntui usein vievän monta tuntia, ellei ensimmäinen hakusana ja linkki osunut oikeaan. Hakukoneeseen tottuneena ihmisenä aloin ymmärtää enemmän kirjastojen hyviä puolia ja sitä miten nopeasti sieltä joko löytää tietoa tai huomaa, että ei löydä.

Lopun liitteistä löytyy tietoa opetuslaitesalkusta, E-Designer-ohjelman käyttöohje ja Beijer Electronics Oy:lle menevä versio GX Works2 -käyttöohjeesta. Opetuslaitesalkun liitteessä kerrotaan koululla käytetystä opetusapuvälineestä ja sen sisällöstä. Salkusta löytyvää E-300-operointipäätettä ohjataan E-Designer-ohjelmalla. E-Designer-käyttöohjeessa kerrotaan perusasiat ohjelman käytöstä.

Lähdeluettelo

1. van der Wal, Eelco: IEC 1131 or 61131: status of the Standard. [www-sivu]. [viitattu 27.04.2010]. Saatavissa: http://www.plcopen.org/pages/tc1_standards/iec_1131_or_61131/
2. Rockwell Automation. What is IEC 1131? [www-sivu]. [viitattu 27.04.2010]. Saatavissa: <http://www.rockwellsoftware.com/corporate/reference/Iec1131/>
3. Asmala, Hannu 1999, muutettu 2003. Automaatiojärjestelmiin liittyviä standardeja. [www-sivu]. [viitattu 27.04.2010]. Saatavissa: http://www.tp.spt.fi/~salabra/ha/Automaatiotekniikka/iec_1131.html
4. Kippo, Asko K. & Tikka, Aimo 2008. Automaatiotekniikan perusteet. Edita Publishing Oy, Helsinki.
5. PLCS.net - Your personal PLC tutor. [www-sivu]. [viitattu 27.04.2010]. Saatavissa: <http://www.plcs.net/chapters/whatis1.htm>
6. Fonselius, Jaakko; Pekkola, Kari; Selosmaa, Seppo; Ström, Markku & Välimaa, Taisto 1996. Koneautomaatio, Automaatiolaitteet. Oy Edita Ab, Helsinki.
7. Heinonen, Anu 2006. GX IEC Developer FX -ohjekirja ohjelmointiin ja ohjelman testaamiseen. Opinnäytetyö. Tampereen ammattikorkeakoulu. Kone- ja laiteautomaatio. Tampere.
8. Keinänen, Toimi; Kärkkäinen, Pentti; Metso, Tommi & Putkonen, Kari 2000. Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2. painos. WSOY.
9. Fonselius, Jaakko; Pekkola, Kari; Selosmaa, Seppo & Välimaa, Taisto 1990. Koneautomaatio, Sähköiset automaatiolaitteet, 3. painos. Valtion painatuskeskus, Helsinki.
10. Beijer Electronics Automation - iQ Works. [www-sivu]. [viitattu 27.04.2010]. Saatavilla: http://www.beijer.fi/web/web_aut_fi.nsf/AllDocuments/ED4AD75925B73090C125755500337B84
11. Beijer Electronics Automation - GX IEC Developer. [www-sivu]. [viitattu 27.04.2010]. Saatavilla: http://www.beijer.fi/web/web_aut_fi.nsf/AllDocuments/C125701A003AA919C1256F87004A951B
12. Mitsubishi Electric, Industrial Automation 2006. GX IEC Developer 7.01, IEC Programming and Documentation System, Beginner's Manual. Berliini.

Liitteet

Liite 1: OPETUSLAITESALKKU - YLEISKUVA

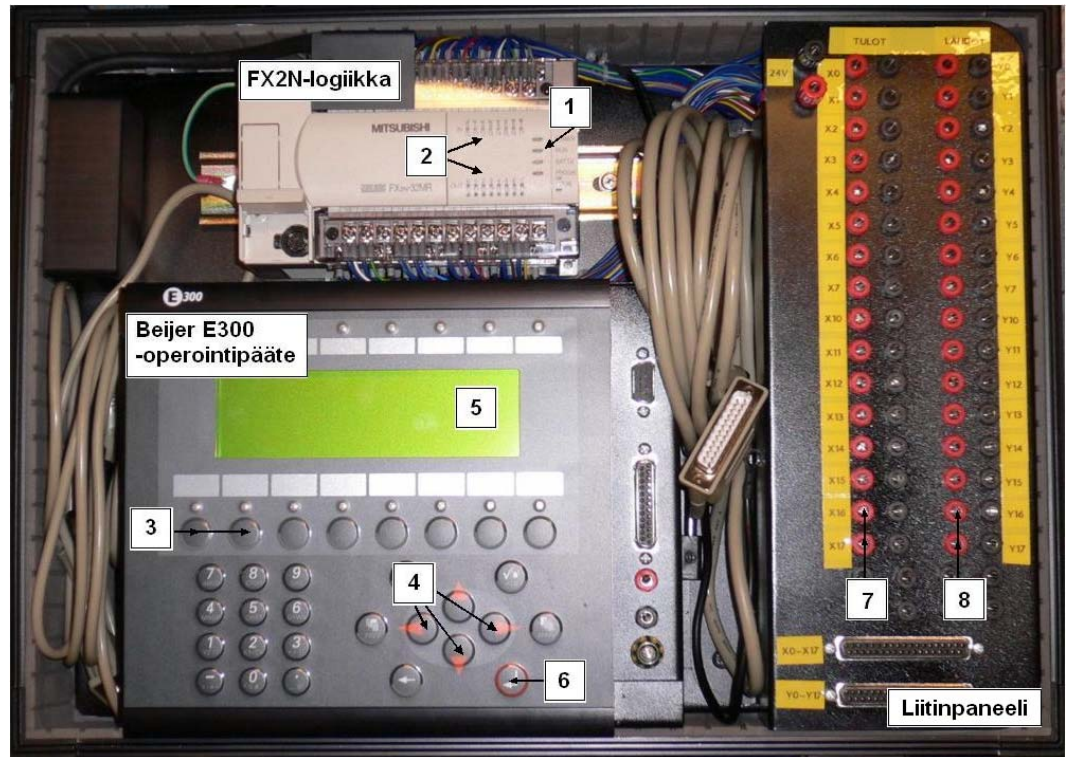
(Heinonen, Anu 2006, 13-15)

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

Opetuslaitesalkun sisältö on seuraava:

1. FX2N-logiikka
2. Beijer E300 -operointipääte
3. liitinpaneeli.

Opetuslaitesalkussa oleva logiikka (kuva 3) kuuluu Mitsubishin FX2N-sarjaan. Logiikan tyyppimerkinnässä oleva luku, kuten FX2N 64-MR, ilmaisee logiikassa 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 tarkoittaa relelähtötyyppiä ja T-kirjain tarkoittaa transistorilähtötyyppiä. Perusyksikkö ja kompakti lisälaitte on kiinnitetty toisiinsa DIN-kiskokiinnikkeen ja nauhakaapeliliittimen 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.



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).

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

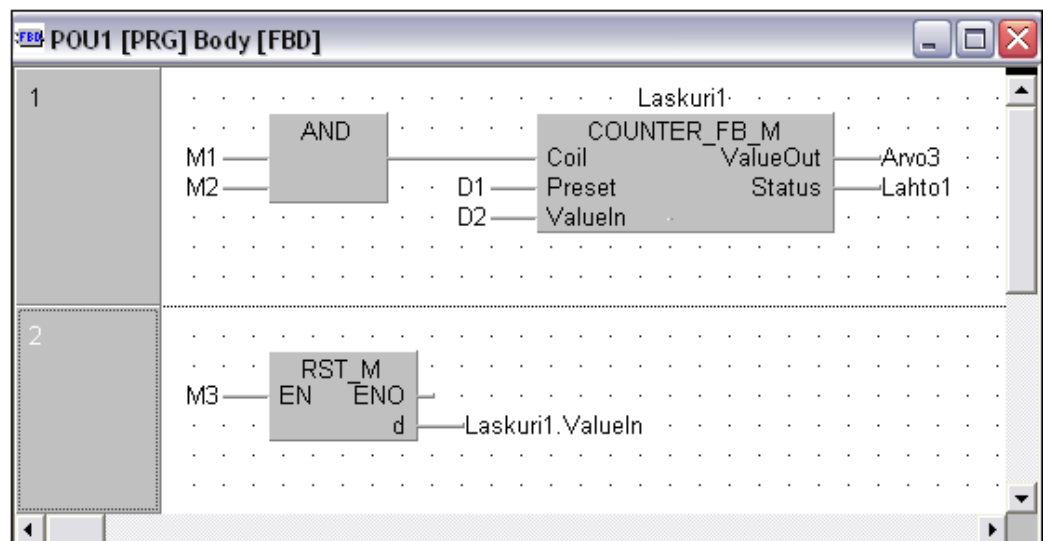
Liite 2: E-DESIGNER

(Heinonen, Anu 2006, 48-57)

E-Designer-ohjelma

Kun automaatio-ohjelma on luotu GX Works2 -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.

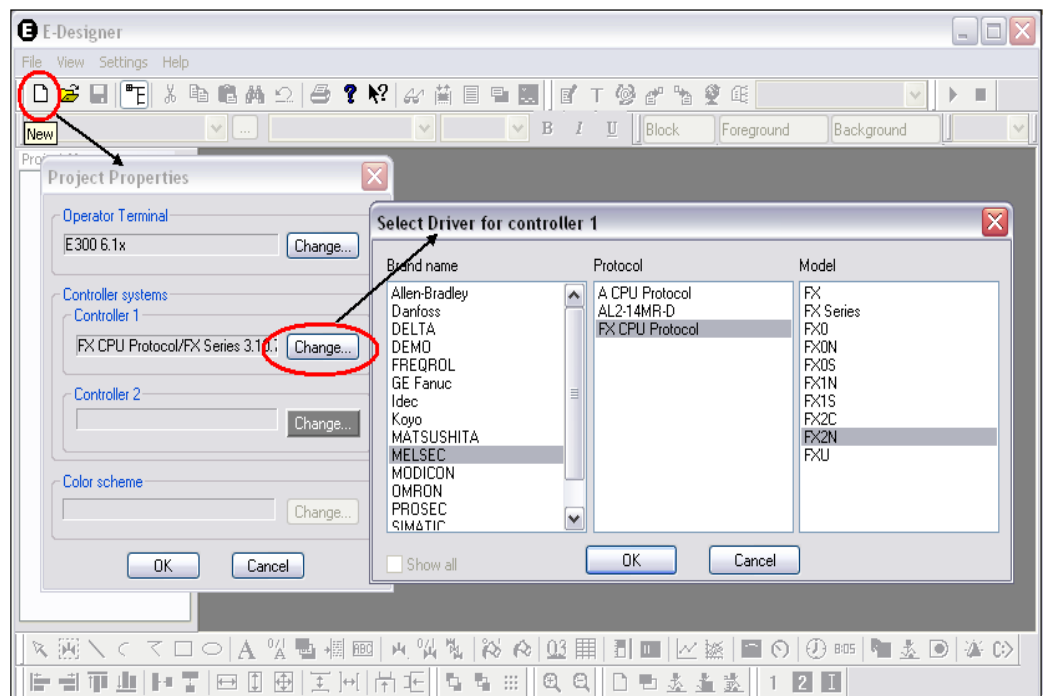
Koska opetuslaitesalkun operointipäätteeltä logiikalle kulkeva virta on heikko, logiikka ei välttämättä reagoi päätteellä tehtyihin toimintoihin. Tämän vuoksi GX Works2 -ohjelmassa tehtyyn projektiin kannattaa muuttaa kaikkien tulojen tilalle merkkerit. Esimerkiksi tulo1 voi olla M1, tulo 2 voi olla 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.



Kuva 37 Merkkereitä käytetty tulosten tilalla

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).



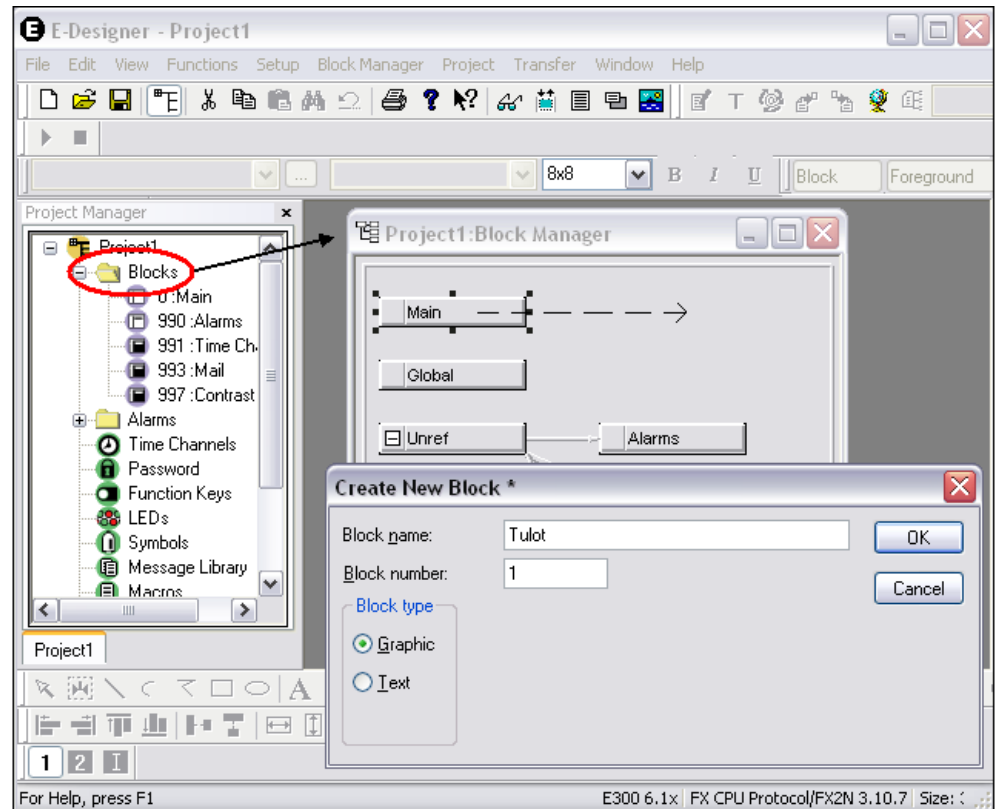
Kuva 38 Uuden projektin luominen sekä logiikan valitseminen

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

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

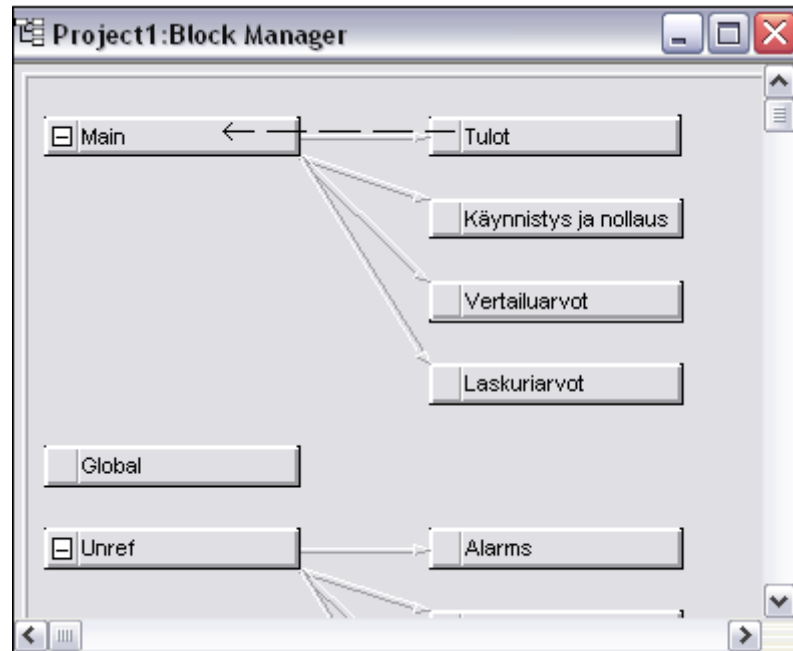
voidaan vapauttaa, kun ohjelma on piirtänyt riittävän pituisen nuolen oikealle. Tämän jälkeen näytölle ilmestyy välittömästi Create New Block -ikkuna, jossa Block Name -kenttään annetaan sivulle nimi, esimerkiksi Tulot. Nimeäminen hyväksytään OK-painikkeella.



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.

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.



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.

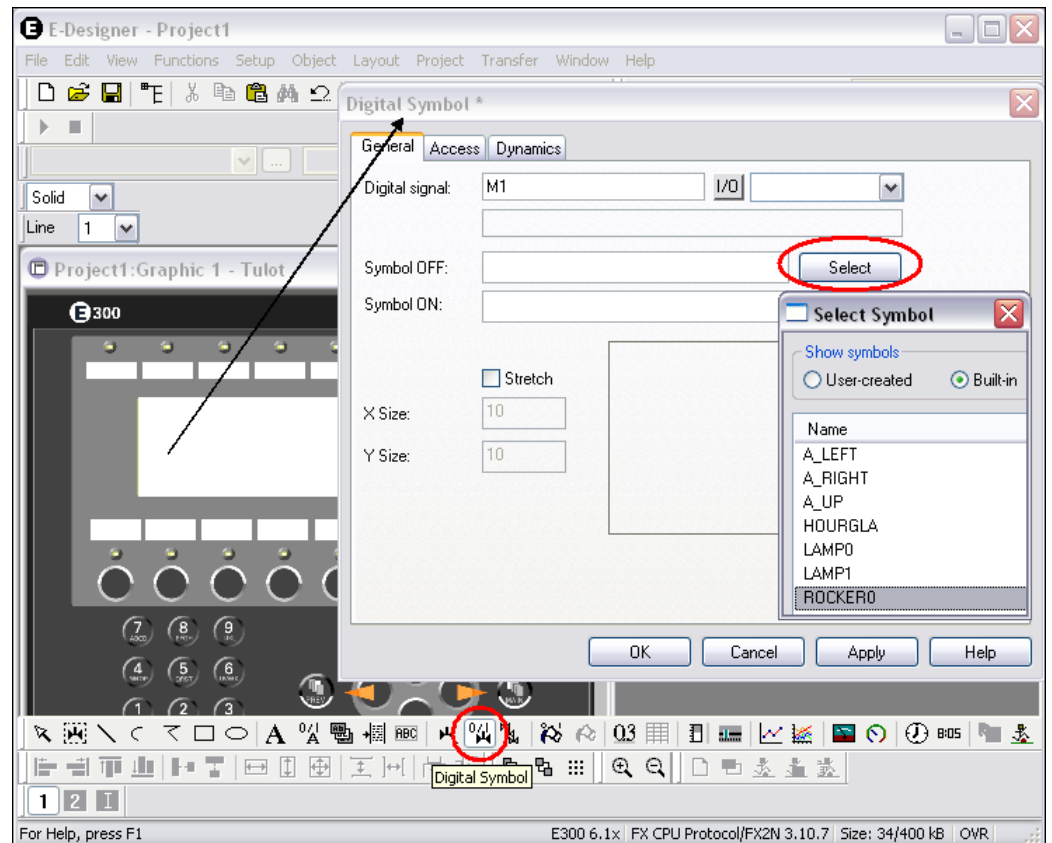
Digitaalisymbolit

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 Works2 -ohjelmassa esiintyvälle päälle-kytkettävälle 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 operointipääteen valkoiselle keskialueelle. Paina kerran hiiren vasenta

painiketta alueella, jolloin näyttöön avautuu Digital Symbol -ikkuna. Kirjoita Digital signal -kenttään GX Works2 -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.

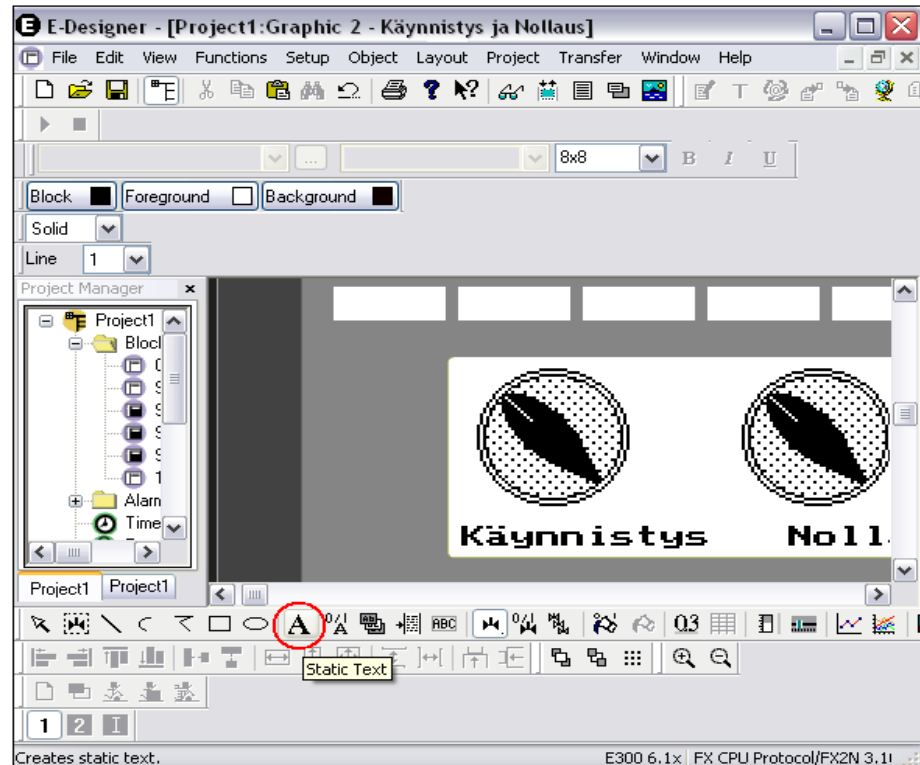


Kuva 41 Ohjauspainikkeiden valinta

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.

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.

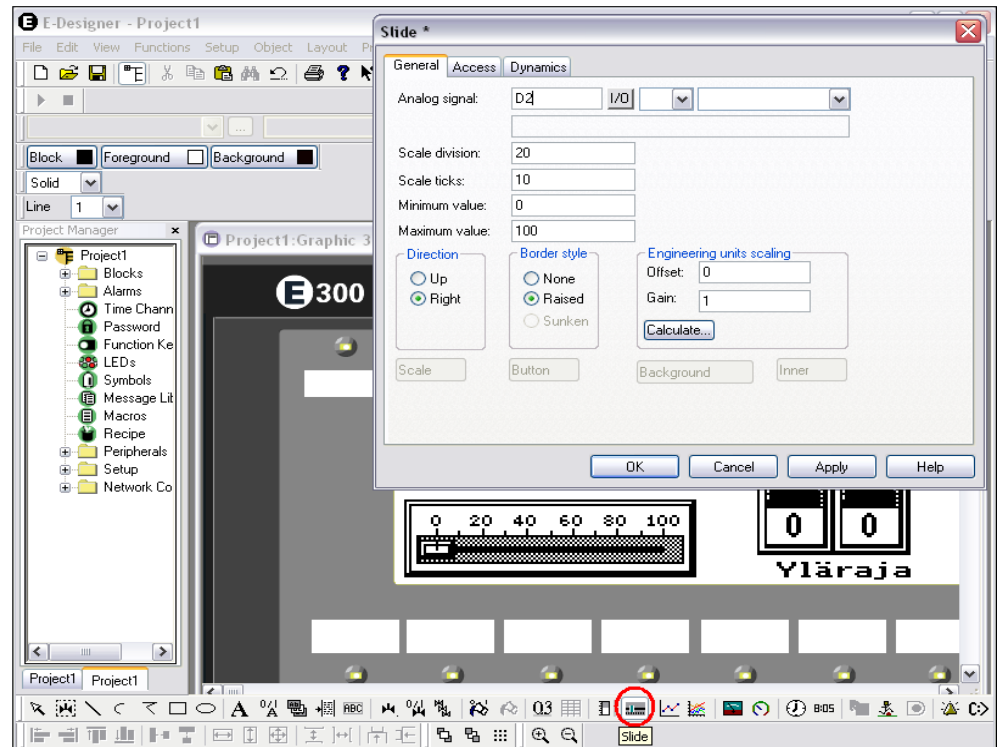


Kuva 42 ROT-kytkimien nimeäminen tekstityökalun avulla

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 Works2 -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). Liukukytкин vahvistetaan toimivaksi laittamalla rasti Enable Operator Input -kohtaan Access-välilehdellä. Kytкин hyväksytään painamalla ensin Apply ja tämän jälkeen OK.



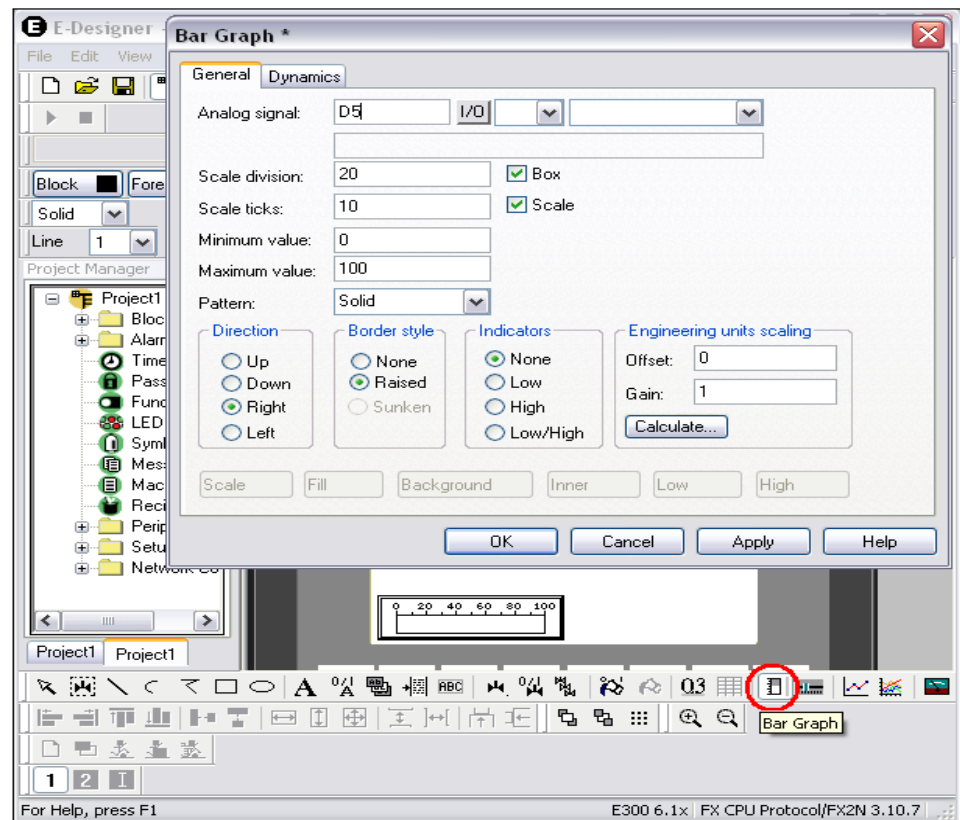
Kuva 43 Liukukytकिन lisääminen

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 Works2 -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

asteikkotyypiksi (Border style) nouseva (Raised). Valinnat vahvistetaan painamalla ensin Apply ja tämän jälkeen OK.



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 toiminnoista löytyy E-Designer-ohjelmamanuaalista.

Ohjelman lataaminen operointipäätteelle










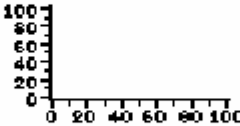

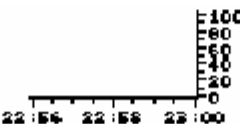
Ennen E-Designer-ohjelman siirtämistä Beijer E300 -operointipäätteelle tarkista, että GX IEC Developer FX -ohjelma muutoksineen on ladattu logiikalle. E-Designer-ohjelman siirto operointipäätteelle tapahtuu valitsemalla ylävalikosta Transfer → Project, jonka jälkeen ohjelma avaa Project Transfer -ikkunan. Ohjelma siirretään Send-painikkeen avulla.

Ohjelman latauduttua Beijer E300 -operointipäätteellä voidaan tarkastella E-Designer-ohjelmalla luotuja sivuja eri toimintonäppäinten avulla.

Sivuilta paluu tapahtuu toimintonäppäimen F8 avulla. Painikkeiden päälle- ja poispainaminen tapahtuu nuolinäppäinten ja enter-painikkeen avulla. Samoilla painikkeilla asetetaan myös liukukytkimien arvot. Tuloksia voidaan katsella joko suoraan toimilaitteilla tai GX Works2 -ohjelmaan luodusta projektista, monitoritilan ollessa päällä.

Liite 3: E-Designer-ohjelman graafisia symboleja

(Heinonen, Anu 2006, liite 3)

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.

Liite 4: GX Works2 -käyttöohjeen lyhyempi versio

4.5 GX Works2 -ohjelma

4.5.1 Yleiskuva

GX Works2 on uusien FX- ja Q-logiikoiden ohjelmointiohjelmisto, joka korvaa GX Developer ja GX IEC Developer -ohjelmat. Se on 32-bittinen Windows-pohjainen ohjelmisto sovellusten ohjelmointiin, simulointiin, monitorointiin, vianmääritykseen ja dokumentointiin. Ohjelmistoa on parannettu tuottavuuden kasvattamiseksi ja suunnittelukustannusten alentamiseksi. Ohjelmistossa on paljon uusia hyödyllisiä toimintoja, kuten esimerkiksi logiikkasovellusten versionhallinta. (Beijer Electronics Automation - iQ Works & Beijer Electronics Automation - GX IEC Developer)

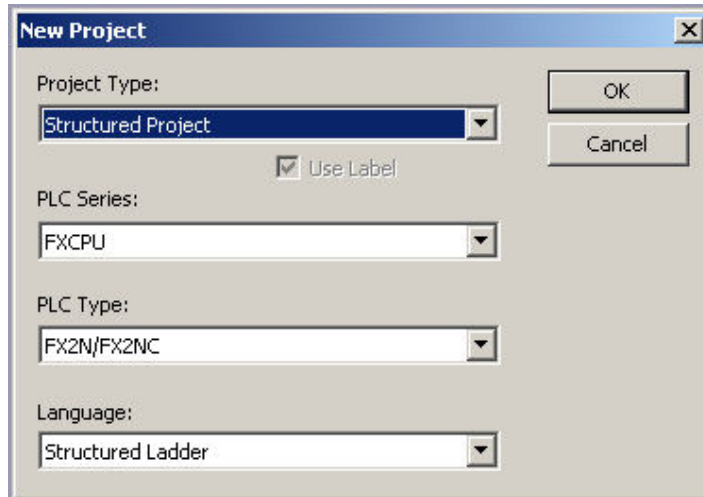
GX Works2 -ohjelma mahdollistaa viiden eri ohjelmointikielen valinnan, jolloin suunnittelija pystyy valitsemaan sopivimman kielen tarvitsemaansa työhön. Valittavissa olevat ohjelmointikielet ovat käskylistä (IL), lausekieli (ST), logiikkakaavio (FBD), relekaavio (LD) ja askelkaavio (SFC). Ohjelmointikielet perustuvat IEC 61131-3 -standardiin.

4.5.2 Ohjelman käynnistys

Ohjelman saa käyntiin Windowsin työpöydältä löytyvältä GX Works2 -ikonilla tai start-valikon alta. Eli tarkemmin sanottuna Start -> All Programs -> MELSOFT Application -> GX Works2 -> GX Works2.

4.5.3 Uuden projektin luominen

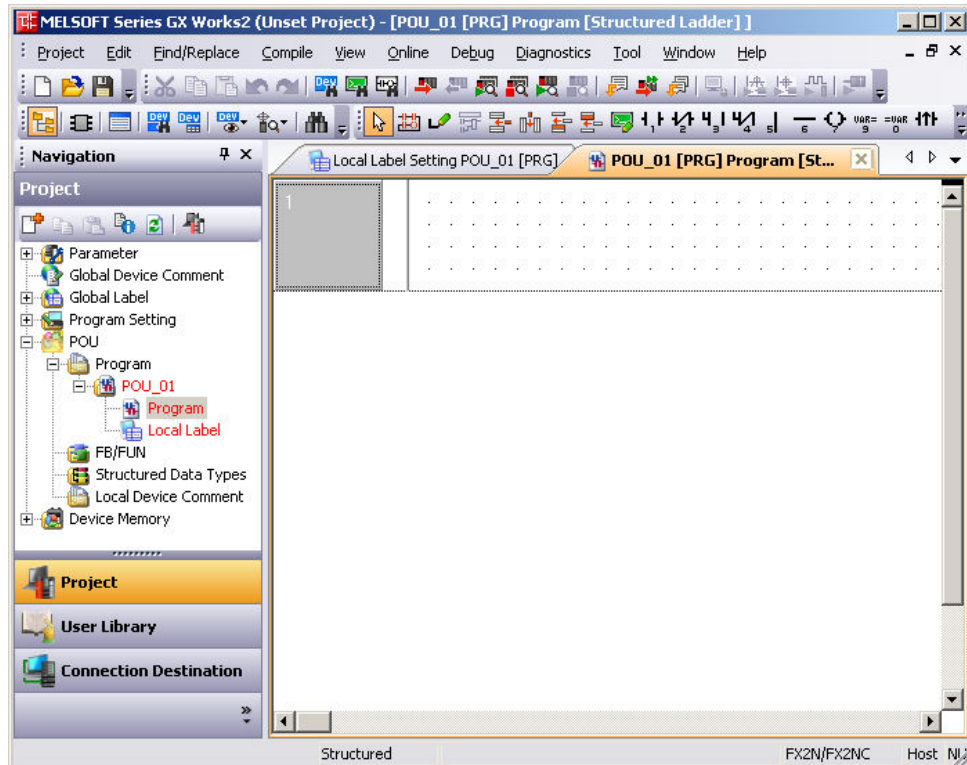
Uusi projekti tehdään valitsemalla Project-valikosta New. Aukeaa ikkuna (Kuva 7), jossa kysytään projektityyppiä, logiikkasarjaa, logiikkatyyppiä ja kieltä. Tehdään valinnat kuvan mukaisesti ja painetaan OK.



Kuva 7. Uuden projektin luonti-ikkuna

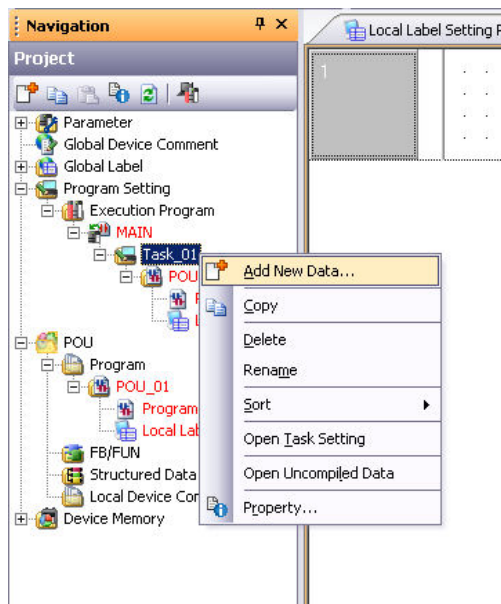
4.5.4 Ohjelman alustaminen

Kun uusi projekti on luotuna (Kuva 8), aloitetaan ohjelman alustaminen. Kuvan vasemmalla näkyy projektipuu, ylhäällä työkalurivi ja keskeltä löytyvät valitun rakenneyksikön (Programmable Organization Unit, POU) ohjelman rakenne ja muuttujalista.



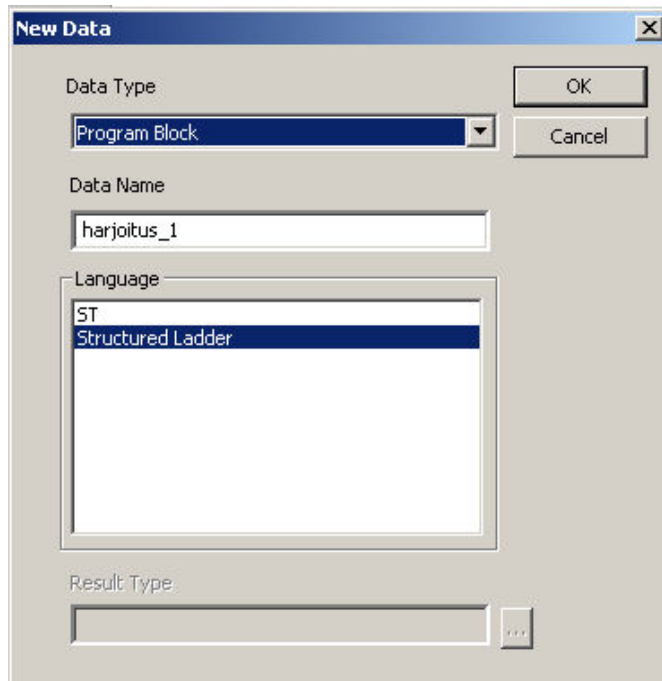
Kuva 8. Uusi projekti on luotuna

Ensimmäiseksi luodaan uusi rakenneyksikkö. Avataan vasemmalta projektipuusta Program Setting -puu kokonaan. Painetaan oikeaa hiiren nappia Task_01:ssä ja valitaan Add New Data (Kuva 9).



Kuva 9. Uuden datan lisääminen

Aukeaa valintaikkuna, jossa kysytään datatyyppiä, työn nimeä ja kieltä. Valitaan kuvan 10 mukaisesti datatyyppiä Program Block ja kielesi Structured Ladder. Työn nimen voi valita oman mielen mukaan ja sitten painetaan OK.



Kuva 10. Uuden datatyyppin valintaikkuna

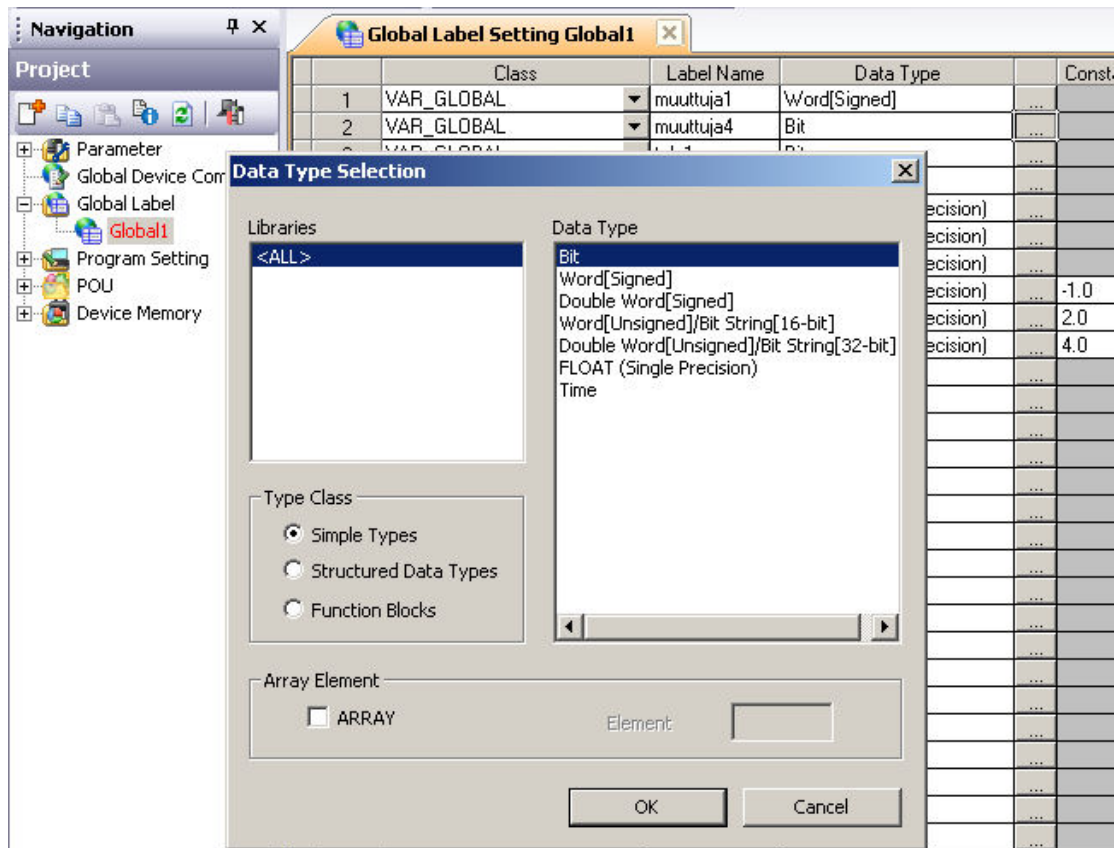
4.5.5 Yleismuuttujien listaaminen

Yleismuuttujien taulukko avataan vasemmalta rakennepuusta Global Label -> Global1 -kohdasta (Kuva 11).

	Class	Label Name	Data Type	Constant	Device	Address	C
1	VAR_GLOBAL	muuttuja1	Word(Signed)	...	D0	%MWO.0	
2	VAR_GLOBAL	muuttuja4	Bit	...	M1	%MX0.1	
3	VAR_GLOBAL	tulo1	Bit	...	X000	%IX0	
4	VAR_GLOBAL	lahto1	Bit	...	Y000	%QX0	
5	VAR_GLOBAL	sivu1	FLOAT (Single Precision)	...	D10	%MD0.10	
6	VAR_GLOBAL	b	FLOAT (Single Precision)	...	D100	%MD0.100	
7	VAR_GLOBAL	tulos1	FLOAT (Single Precision)	...	D106	%MD0.106	
8	VAR_GLOBAL_CONSTANT	kerroin_1	FLOAT (Single Precision)	-1.0			
9	VAR_GLOBAL_CONSTANT	kerroin_2	FLOAT (Single Precision)	2.0			
10	VAR_GLOBAL_CONSTANT	kerroin_4	FLOAT (Single Precision)	4.0			

Kuva 11. Globaalilista avattuna

Taulukosta valitaan ensimmäiseksi yleismuuttujat (Class) pudotusvalikosta. Seuraavaksi listataan Label Name -kohtaan tarvittavien muuttujien nimet, kuten esimerkiksi tulot ja lähdöt. Datatyyppin valintaikkuna saadaan auki kuvan xx mukaisesti ...-painikkeesta. Yksinkertaiset datatypit löytyvät Simple Types -valikon alta, ja laskurit ynnä muut löytyvät Function Blocks -valikosta (Kuva 12).



Kuva 12. Simple Types -datatyyppien valinta

Jos muuttuja tarvitsee vakioluvun, se lisätään Constant-sarakkeelle. Logiikan muistipaikka laitetaan Device-sarakkeelle. Osoitteen pitäisi täyttyä tämän jälkeen automaattisesti.

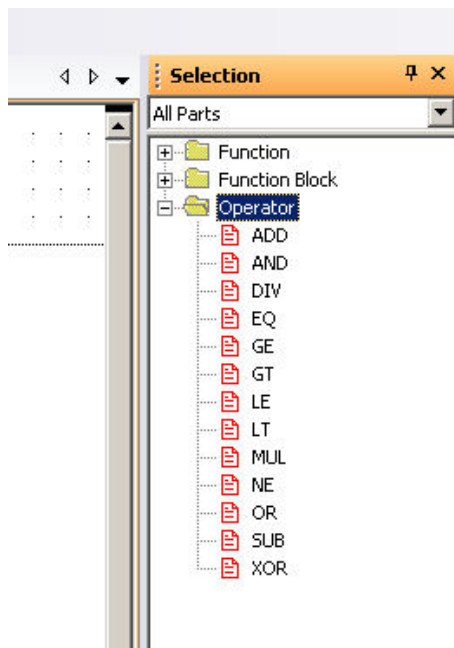
4.5.6 Ohjelman luominen

Tämän jälkeen voidaan aloittaa itse ohjelman tekeminen. Tarvittavat operaattorit löytyvät Function Block Selection Window -painikkeella (Kuva 13).



Kuva 13. Function Block Selection Window -painikkeen sijainti

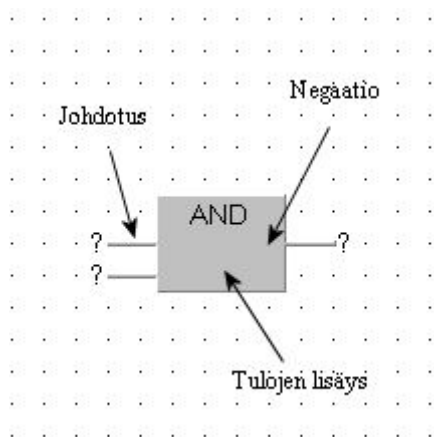
Ruudun oikeaan reunaan avautuu uusi valikko (kuva 14), josta löytyvät kaikki tarvittavat operaattorit virtapiirien tekoon, kunhan valikosta on valittuna All Parts. Yleisimmin tarvittut perusoperaattorit löytyvät Operator-kansiosta.



Kuva 14. Perusoperaattorit Function Block Selection Windows -valikossa

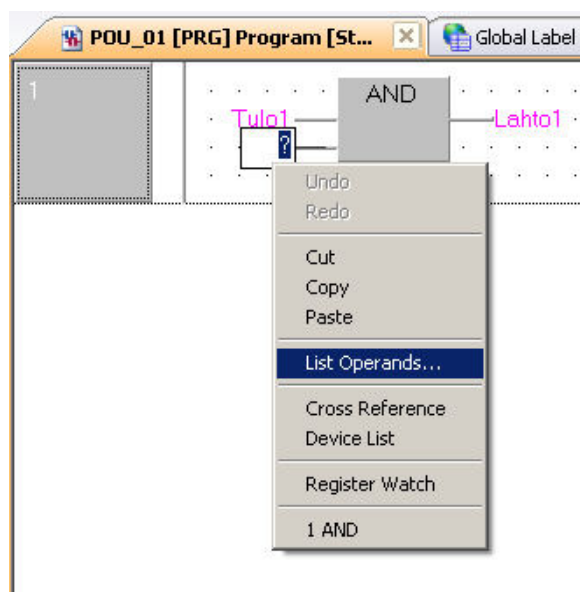
Operaattorit saadaan virtapiirille joko tuplaklikkaamalla tai raahaamalla. Johdotukset onnistuvat työkaluriviltä löytyvällä Interconnect Modella (CTRL+T). Painetaan kerran vasemmalla napilla kohtaan, josta johto vedetään, ja toisen kerran vasenta nappia kohdassa, johon johto päättyy. Negaation saa tuplaklikkaamalla kuvan 15 mukaisesti.

Operaattoriin saa lisättyä tuloja painamalla ensin vasemmalla napilla operaattori aktiiviseksi ja sitten vetämällä sen alareunasta.



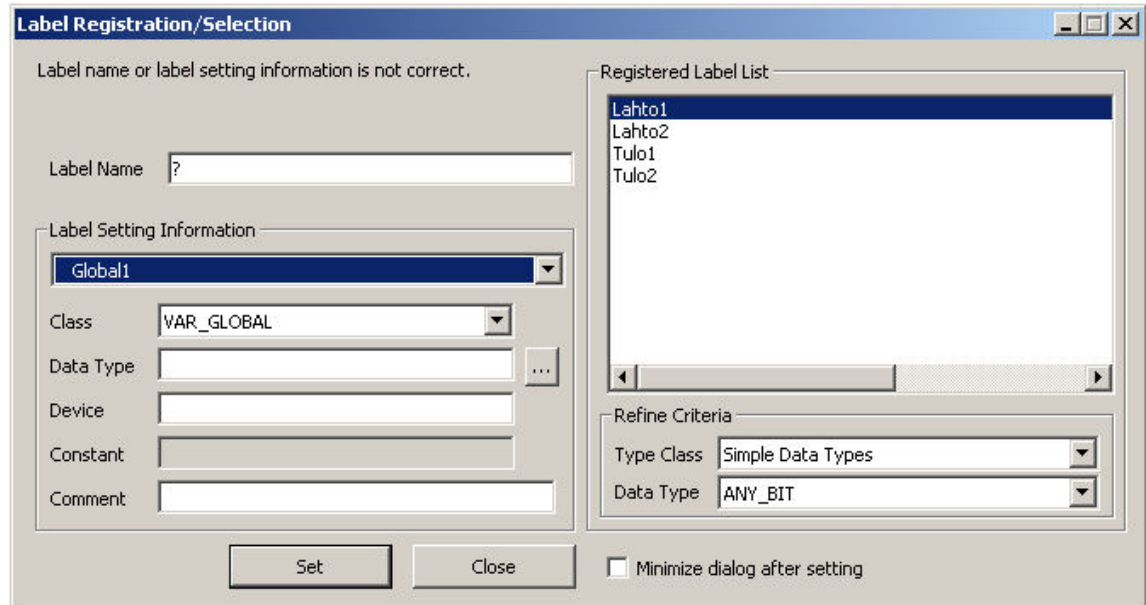
Kuva 15. Operaattorin eri toiminnot

Muuttujien lisääminen tehdään painamalla vasemmalla napilla kysymysmerkin kohdalta ja tämän jälkeen oikealla napilla laatikon sisältä. Tästä aukeaa valintaikkuna, josta otetaan List Operands (Kuva 16).



Kuva 16. Opastus muuttujien löytämiseen

Muuttujat valitaan listasta, jonka saa näkyviin, kun valitsee Label Setting Information -kohtaan Global1 (Kuva 17). Listasta voidaan tämän jälkeen etsiä haluttu muuttuja, ja se otetaan käyttöön painamalla Set ja Close.



Kuva 17. Muuttujien valinta listasta

4.5.7 Aputoiminta

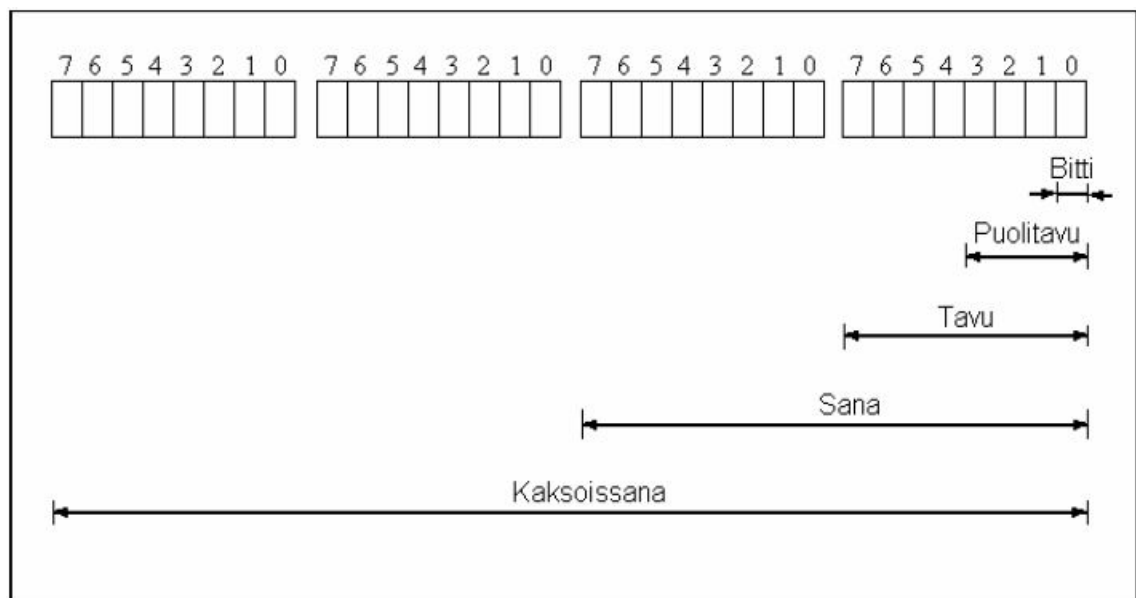
GX Works2 -ohjelma sisältää kattavan Help-valikon, jossa on kerrottuna virhekoodeja ja tarkempia neuvoja ohjelmien tekoon.

4.6 Osoitteet

4.6.1 Bitti, tavu, puolitavu, sana ja kaksoissana

Bitti (Bit, b) voi saada tilat 0 tai 1, ja se on pienin tietoalkio, jota automaatiojärjestelmät käsittelevät (Kuva 18). Kahdeksan peräkkäisen bitin kokonaisuutta sanotaan tavuksi (Byte, B). Logiikassa yhdistetään esimerkiksi kahdeksan tulon tai lähdön signaalit yhdeksi tulotavuksi (IB) tai lähtötavuksi (QB). Jokainen yksittäinen tavun bitti (binääripaikka) voi saada arvon 0 tai 1. Logiikassa käsitellään usein koko tavu tai sana, siis kaikki 8 tai 16 bittiä yhdessä.

Puolitavulla (Nibble), tarkoitetaan neljän bitin ryhmää eli puolikasta tavua. Kahden tavun kokonaisuutta kutsutaan sanaksi (Word, W) ja kahden sanan kokonaisuutta kaksoissanaksi (Double-Word, DW). Logiikassa voidaan yhdistää esimerkiksi 16 tulon tai 16 lähdön signaalit tulosanaksi (IW) tai lähtösanaksi (QW). (Automaatiolaitteet, Koneautomaatio, 1996, 149)



Kuva 18. Binäärijärjestelmä (Heinonen, Anu 2006, 26)

4.6.2 I/O-osoitteet

GX Works2 -ohjelmassa muuttujille pitää antaa logiikan muistipaikkoja vastaavat osoitteet, että toimilaitteita pystytään ohjaamaan. Ohjelma määrittää osoitteet itse, ellei niitä määritetä. Muuttujat määritellään POU-rakenneyksikössä, jotta logiikan keskusyksikön (CPU) muistipaikat saadaan käyttöön. Osoitteet voidaan myös kirjoittaa viittaamaan johonkin fyysiseen tulo- ja lähtöpaikkaan. (Heinonen, Anu 2006, 26)

4.6.3 Apumuistit

Apumuistit ovat mekaanisen releen kaltaisia, logiikan sisäisiä muistipaikkoja. Niitä voidaan myös kutsua sisäisiksi releiksi, lipuiksi tai merkkereiksi. Tässä tapauksessa kutsutaan niitä apumuisteiksi. Apumuisteilla voi olla vain kaksi tilaa, varattuna (1) ja ei käytössä (0).

Apumuisteja käytetään tiedontallennuspaikkana, kun jokin asia halutaan panna muistiin. Näitä käytetään esimerkiksi kappaleiden lajittelun ohjauksessa ja laskentatehtävissä. Erikoismuistipaikat ovat apumuistipaikkoja, jotka on valmiiksi tallennettu käyttäjää varten nopeampaa käyttöä varten. GX Works2 -ohjelmasta löytyy Help-> Special Relay/Special Register -> FX series PLC -kohta, jossa on lueteltu kaikki erikoismuistipaikat. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 251-252)

4.6.4 Virhekoodit

GX Works2 -ohjelman Help-valikon CPU Error -kohdasta pystyy etsimään selvennyksen ja korjausehdotuksia eri virhekoodeille.

4.7 Luvut

4.7.1 BCD-luvut

Binäärikoodattuja desimaalilukuja (Binary Coded Decimal, BCD) käytetään esimerkiksi taskulaskimien numeronäytöissä. BCD-luvut perustuvat lukujen 0-9 esittämiseen neljällä bitillä. Esittämistapa toimii siis samalla tavalla kuin binääriluvuilla. Esimerkiksi BCD-luku 0010 vastaa desimaalilukua 2 ja BCD-luku 0101 vastaa desimaalilukua 5. Useamman luvun ilmoittaminen tapahtuu vain laittamalla kaksi neljän bitin lukua peräkkäin. Esimerkiksi desimaaliluku 25 on BCD-lukuna 0010 0101 eli 2 ja 5. Taulukossa 1 luvut ovat listattuina tarkemmin. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 65-66)

Taulukko 1. BCD-lukujen muunnokset

Desimaaliluku	BCD-luku
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010
953	1001 0101 0011
8274	1000 0010 0111 0100

4.7.2 Datatyypit

Muuttujan datatyyppi määrittelee, miten sitä käsitellään, muuttujan lukualueen ja montako bittiä datatyyppi sisältää. GX Works2 -ohjelmassa on mahdollisuus valita datatyyppiä taulukon 2 mukaisesti.

Taulukko 2. Valittavissa olevat datatyypit (GX IEC Developer 7.01, Beginner's Manual, 3 - 8)

Datatyyppi		Koko (bittejä)
BOOL	Boolean	1
INT	Integer	16
DINT	Double integer	32
WORD	Bit string 16	16
DWORD	Bit string 32	32
REAL	Floating-point value	32
TIME	Time value	32
STRING (vain Q-sarjassa)	Character string	Maksimi 50 merkkiä.

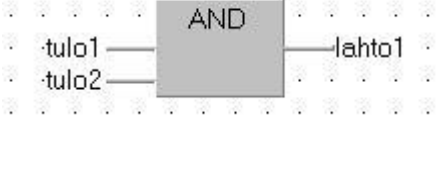
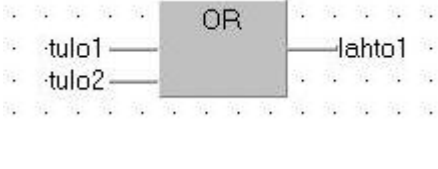

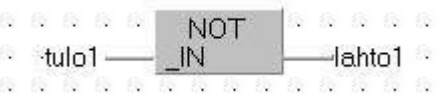
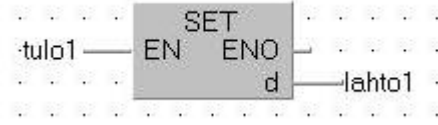

4.8 Operaattorit

Ohjelmien luomiseen tarvitaan erilaisia operaattoreita, ja seuraavassa on koottuna GX Works2:n yleisimmin käytetyt operaattorit ja selitykset näiden toimintaperiaatteista.

4.8.1 Perusoperaattorit

Perusoperaattorit (Taulukko 3) löytyvät Function Block Selection Window -napin takaa. AND-, OR- ja XOR-operaattorit löytyvät Operator-kansiosta ja loput kolme Function-kansiosta.


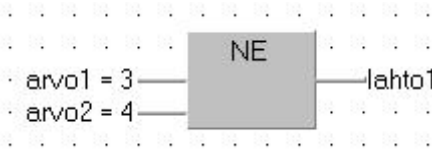

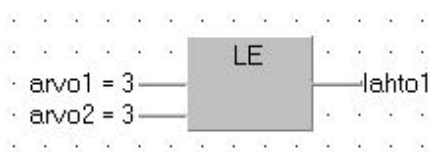
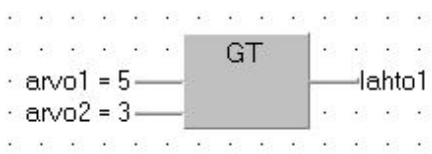

Taulukko 3. Perusoperaattorit ja niiden toimintaperiaate

Käsky	Kuvaus															
<p>JA-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</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>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	0	0	1	0	1	1	1
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	0														
0	1	0														
1	1	1														
<p>TAI-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</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>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	1	0	1	1	1	1	1
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	1														
0	1	1														
1	1	1														
<p>EHDOTON TAI -käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</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>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Tulo1	Tulo2	Lahto1	0	0	0	1	0	1	0	1	1	1	1	0
Tulo1	Tulo2	Lahto1														
0	0	0														
1	0	1														
0	1	1														
1	1	0														
<p>EI-käsky eli invertteri</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Lahto1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	Tulo1	Lahto1	0	1	1	0									
Tulo1	Lahto1															
0	1															
1	0															
<p>Set- eli asetuskäsky</p> 	<p>Kun tulo1 saa arvon 1, menee lahto1 päälle eli saa myös arvon 1.</p>															
<p>Reset- eli nollauskäsky</p> 	<p>Kun tulo1 saa arvon 1, menee lahto1 pois päältä eli saa arvon 0.</p>															

4.8.2 Vertailijat

Vertailijat (Taulukko 4) löytyvät perusoperaattoreiden tapaan Function Block Selection Window -napin avulla. Kaikki vertailijaoperaattorit löytyvät Operator-kansiosta.

Taulukko 4. Vertailijaoperaattorit ja niiden toimintaperiaatteet (Heinonen, Anu 2006, 32)


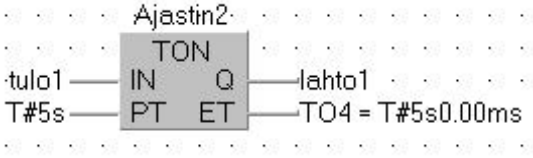
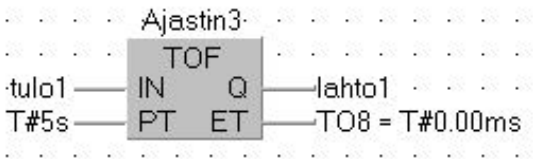
Käsky	Kuvaus
	<p>Yhtäsuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa yhtä suuret lähtö1 on päällä.</p>
	<p>Erisuuri-käsky</p> <p>Arvojen 1 ja 2 ollessa eri suuret lähtö1 on päällä.</p>
	<p>Pienempi kuin -käsky</p> <p>Kun arvo1 on pienempi kuin arvo 2, lähtö1 on päällä.</p>
	<p>Pienempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo1 on pienempi tai yhtä suuri kuin arvo2, lähtö1 on päällä.</p>
	<p>Suurempi kuin -käsky</p> <p>Kun arvo1 on suurempi kuin arvo2, lähtö1 on päällä.</p>
	<p>Suurempi tai yhtäsuuri kuin -käsky</p> <p>Kun arvo1 on suurempi tai yhtä suuri kuin arvo2, lähtö1 on päällä.</p>

4.8.3 Ajastimet

Ohjelmoitavien logiikoiden ajastimet ovat yleensä vetohidasteisia, eli kello lähtee käyntiin jollakin tuloehdolla. Ajastimilla voidaan prosessiin lisätä viiveitä, jotta tietty työsuoritus ehtii tapahtua. Eräänä esimerkkinä on tavaratalojen automaattiovien automatisointi. Ajastimien käskyrakenne vaihtelee eri logiikkamerkkien välillä. Jotkin käyttävät niiden ohjelmointiin apumuistipaikkoja, kun taas joissakin ohjelmointiin tarvitaan vain parin rivin ohjelmarutiini. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 252-253)

Ajastimet (Taulukko 5) löytyvät Function Block -kansioista Function Block Selection Window -napin takaa.

Taulukko 5. Muutama esimerkki ajastimien toiminnasta (Heinonen, Anu 2006, 33)

Ajastin	Kuvaus
	<p>Coil = ajastimen käynnistys Preset = ajastettava arvo (> ValueIn) ValueIn = lähtöarvo (yleensä 0) ValueOut = näyttää ajan Status = ilmoittaa kun aika on kulunut</p>
	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO4) Q = ilmoittaa kun aika on kulunut</p>
	<p>IN = ajastimen käynnistys PT = ajastettava arvo (esim. T#5s300ms) ET = näyttää ajan (esim. TO8) Q = ilmoittaa kun aika on kulunut</p>

4.8.4 Laskurit

Laskureilla voidaan esimerkiksi hallita kappaletavaran määrää. Myös virvoitusjuomapullojen laskenta täyttölinjalla on tästä tyyppiesimerkki. Pysäköintitalon ajoneuvojen laskenta voidaan automatisoida myös käyttämällä logiikan laskuriominaisuutta. Tavallisissa laskureissa on kaksi tuloa: nollaustulo ja laskuritulo. Laskuritulon etureuna lasketaan ja verrataan sitä nimettyyn esiasetettuun arvoon. Lähtö pysyy nollana, kunnes laskettu arvo on suurempi kuin esiasetettu arvo. Tämän jälkeen lähtö asettuu ykköseksi. Laskuri resetoidaan eli palautetaan alkuun jälleen nollaustulolla. Laskurit ovat yleensä ylhäältä alaspäin laskevia, ja laskurin väliarvoja voidaan tallentaa apumuisteihin aritmeettisten laskusuoritusten ja vertailujen suorittamiseksi. Tähän käytetään yleensä FUN-komentoja. Logiikoissa on yleensä myös kolmitulaisia ylös-alas-laskureita. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 253)

Myös laskurit (Taulukko 6) löytyvät Function Block Selection Window -napilla ja sen alta Function Block -kansiosta.

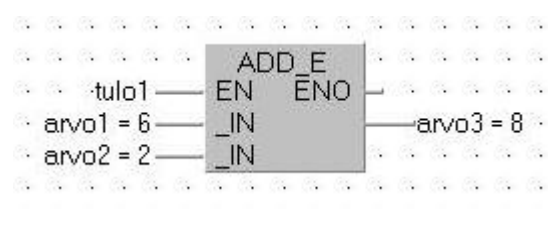
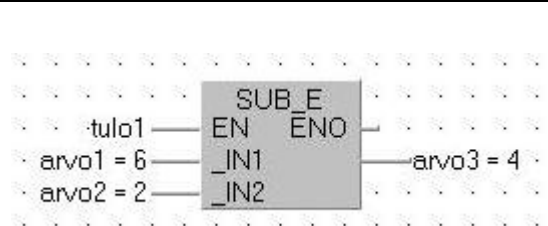
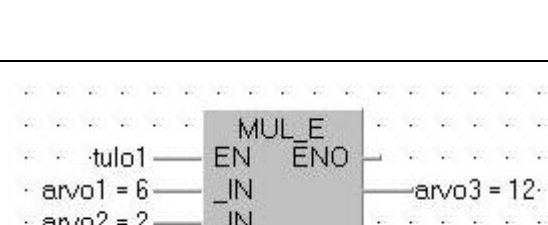

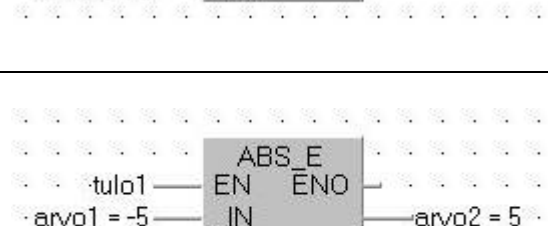
Taulukko 6. Laskurien toimintaperiaatteet (Heinonen, Anu 2006, 35)

Laskuri	Kuvaus
<pre> Laskuri1 CTU_E tulo1 — EN ENO — tulo2 — CU Q — lahto1 tulo3 — RESET CV — arvo2 = 5 arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) kasvaa tulolla 2, kun tulo1 on päällä.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtä suuri kuin esiasetusarvo (PV).</p> <p>Laskuriarvo resetoidaan tulolla 3.</p>
<pre> Laskuri2 CTD_E tulo1 — EN ENO — tulo2 — CD Q — lahto1 tulo3 — LOAD CV — arvo2 = 0 arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) pienenee tulolla 2.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtä suuri kuin nolla.</p> <p>Esiasetusarvo (PV) saadaan ladattua tulolla, kun tulot 1 ja 2 on kytketty päälle.</p>
<pre> Laskuri3 CTUD_E tulo1 — EN ENO — tulo2 — CU QU — lahto1 tulo3 — CD QD — lahto2 tulo4 — RESET CV — arvo2 = 3 tulo5 — LOAD arvo1 = 5 — PV </pre>	<p>Laskuriarvo (CV) kasvaa tulolla 2 ja pienenee tulolla 3.</p> <p>Lähtö1 on päällä, kun tulo1 on päällä ja laskuriarvo on suurempi tai yhtä suuri kuin esiasetusarvo (PV).</p> <p>Lähtö2 on päällä, kun tulo1 on päällä ja laskuriarvo on pienempi tai yhtä suuri kuin nolla.</p> <p>Esiasetusarvon lataamisessa tulon 5 avulla tulee tulojen 1, 2 ja 3 olla päällä.</p>

4.8.5 Aritmetiikkakäskyt

Taulukon 7 aritmetiikkakäskyt eli laskentakäskyt löytyvät Function-kansiosta. Laskentakäskyjen käytössä pitää muistaa, että molempien puolien arvojen pitää olla samaa datatyyppiä eli esimerkiksi $REAL + REAL = REAL$, jolloin reaaliluku on datatyyppinä. (Heinonen, Anu 2006, 36)

Taulukko 7 Aritmetiikkakäskyt selityksineen

Käsky	Kuvaus
	<p>Yhteenlasku</p> <p>Kun tulo1 saa arvon 1, lasketaan arvot 1 ja 2 yhteen.</p>
	<p>Vähennyslasku</p> <p>Kun tulo1 saa arvon 1, vähennetään _IN1 arvo _IN2 arvosta.</p>
	<p>Kertolasku</p> <p>Kun tulo1 saa arvon 1, kerrotaan arvot 1 ja 2 keskenään.</p>
	<p>Jakolasku</p> <p>Kun tulo1 saa arvon 1, jaetaan _IN1 arvo _IN2 arvolla.</p>
	<p>Vastaluku</p> <p>Kun tulo1 saa arvon 1, muutetaan arvo1 sen vastaluvuksi.</p>

4.8.6 Pääohjaustoiminto

Kun pääohjaustoiminto (Master Control Set/Reset, MC/MCR) asetetaan päälle, se katkaisee ohjelman luvun. Tämä vastaa fyysisen pääkytkimen käyttöä, paitsi että sähköit säilyvät päällä. Resetoiminta palauttaa ohjelman luvun, ja toiminta jatkuu siitä, mihin se MC-komennolla jäi. Kommentoita voidaan käyttää ohjelmallisen hätä-seis-kytkimen ohjauksessa. Paikallinen lukitusvirtapiiri eli hätäpysäytyspiiri tulee tehdä tämän lisäksi aina ilman ohjelmoitavaa logiikkaa. (Koneautomaatio 2, Logiikat ja ohjausjärjestelmät, 2002, 255)

MC- ja MCR-operaattori löytyvät Function-kansiosta, jonka saa näkyviin Function Block Selection Window -nappia painamalla.

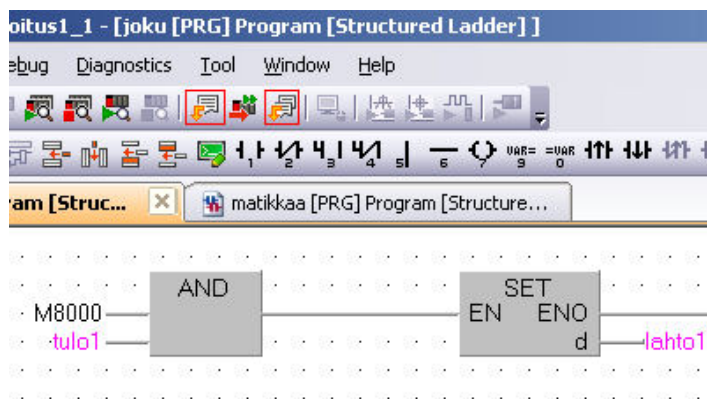
4.9 Simulointi

Valmista ohjelmaa pystytään simuloimaan offline-tilassa, jolloin ei tarvitse fyysisesti olla logiikan vierellä. Ennen simulointia ohjelma pitää kuitenkin aina kääntää.

4.9.1 Projektin kääntäminen

Projektin kääntäminen tehdään Build ja Rebuild all -nappien avulla (Kuva 19). Build-komento kääntää vain kääntämättömät ohjelmaosiot Execution Program -puusta.

Rebuild all kääntää kaiken eli myös jo ennestään käännettyt.



Kuva 19. Build ja Build All -nappien sijainti

Ohjelma kysyy tämän jälkeen varmistusta kääntämiseen, johon vastataan Yes.

Tulokset ja mahdolliset virheet ohjelma ilmoittaa alhaalla löytyvässä Output-ikkunassa. Virheiden sattuessa tarkistetaan ja korjataan vika.

4.9.2 Offline-simulaattori

Ohjelman simulointi saadaan päälle yläpalkin Start/Stop Simulation -napista tai ylävalikosta Debug -> Start/Stop Simulation (Kuva 20).



Kuva 20. Start/Stop Simulation -napin sijainti

4.10 Ohjelman lataaminen logiikalle

Ohjelma ladataan logiikalle Write to PLC -napilla, joka löytyy yläpalkista tai valikosta Online -> Write to PLC (Kuva 21).



Kuva 21. Write to PLC -napin sijainti

Online Data Operation -ikkuna aukeaa, johon valitaan oikea logiikka ja haluttu projekti. Ladattavat ohjelmat valitaan ruksaamalla Target-kohta halutun ohjelman riviltä. Kun valinnat on tehty, painetaan Execute-nappia, ja ohjelma latautuu logiikalle. Jos logiikassa oli ladattuna jo jokin ohjelma, aukeaa varmistusikkuna, johon kuitataan Yes to all, jolloin olemassa olevan ohjelman päälle kirjoitetaan. Ikkunat voi sulkea, kun ohjelma on ladattuna logiikalle.

4.11 Ohjelman monitorointi

Ohjelman monitorointi tapahtuu valitsemalla haluttu ohjelma tuplaklikkaamalla POU -> Program -> POU_01 -> Program, jolloin ohjelman ikkuna aukeaa.

Tämän jälkeen valitaan yläpalkista Start Monitoring tai valikosta Online -> Monitor -> Start Monitoring (Kuva 22).



Kuva 22. Start Monitoring -napin sijainti

Asetetaan ohjelmasta ohjattava CPU RUN-tilaan. Myös PLC:n RUN/STOP-kytkin asetetaan RUN-tilaan.

Ohjelman monitoroinnin lopetus tapahtuu päinvastaisesti käynnistykseen eli asetetaan logiikka ja ohjelma STOP-tilaan. Yläpalkista löytyy Stop Monitoring -nappi tai vaihtoehtoisesti valikosta Online -> Monitor -> Stop Monitoring.