



# Logiikkaohjaus automaattiselle työstökoneelle

Pentti Suurhasko

OPINNÄYTETYÖ  
Maaliskuu 2021

Sähkö- ja automaatiotekniikka  
Automaatiotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkö- ja automaatiotekniikka  
Automaatiotekniikka

SUURHASKO, PENTTI  
Logiikkaohjaus automaattiselle työstökoneelle

Opinnäytetyö 48 sivua, joista liitteitä 0 sivua  
Maaliskuu 2021

---

Nokialainen yritys Par-Met Team on vuonna 1997 perustettu alihankintakonepaja. He olivat erikoistuneet suursarjojen automaattikoneistukseen. Nykyään Par-Met Team toimii pääasiassa konsulttina ja antaa CNC-koneiden käyttöopastusta. Opinnäytetyö sai alkunsa Par-Met Teamin saadessa tilauksen logiikkaohjauksesta, joka ohjaisi automaattista porayksikköä. Asiakas tarvitsi sitä suursarjatuotantoon sähkökaappien lukitustappien tekoon. Työ oli alun perin tehty käsin ja sen automatisointi nostaisi tuotantokapasiteettia noin kolminkertaiseksi.

Asiakkaan toivomuksina olivat laitteiston helppo muunneltavuus muihin projekteihin sekä ohjauksen toteutus jollakin tunnetulla logiikalla. Myös valmiin laitteiston pieni koko oli toivomuksena. Tämä antoi työlle paljon valinnanvaraa. Osien valinta oli haastavaa, sillä vaihtoehtoja askelmoottoreiksi sekä niiden ohjaimiksi oli monia. Logiikan osalta valinta oli helppo, sillä tunnettuja valmistajia ei ole montaa. Vaihtoehtoina olivat laajasti käytetyt Siemensin ja Omronin valmistamat mallit. Näistä valinta osui Siemensin S7-1200-sarjaan, sillä sen käytöstä oli aiempaa kokemusta.

Laitteiston ohjaus toteutettiin käyttäen Siemensin TIA Portalia. Tällä luotiin askelmoottoreille yhtenä työstökiertona toimiva ohjelma. Ohjelma siis suoritti kappaleen porauksen ja uuden aseman vaihdon nappia painamalla. Opinnäytetyön tuloksena syntyi helposti käyttäjän muokattavissa oleva ohjelma kompaktissa koossa. Ohjelmaa on helppo muokata käyttämään useampia poria mikäli osaa hieman PLC-ohjelmointia. Laitteiston muuttaminen muihin prosesseihin kuin vain poraukseen on myös helppoa. Näitä voisi olla muun muassa jyrsintä ja kokoonpano.

Työn onnistumisessa ei ollut suuria ongelmia. Kuten aina projekteissa, suunnitelmat muuttuivat kesken matkan, mutta ne eivät vaikuttaneet työn toteutukseen. Koko laitteistosta tuli kompakti ja se on helposti liikuteltavissa. Asiakas oli tyytyväinen toteutukseen.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Electrical and Automation Engineering  
Automation Engineering

SUURHASKO, PENTTI:  
PLC Program for Automatic Drill Department

Bachelor's thesis, pages, 48 appendices 0 pages  
March 2021

---

The purpose of this thesis was to create a PLC program for an automatic drill department. This study was carried out as a project for Par-Met Team Oy. The project was supposed to be easily modifiable by its user for different kinds of purposes. The theoretical section explores the working mechanics of the stepper motors, CNC-machines and programming languages. The empirical part consists of how the control was created and how the wiring of electrical components was made.

As a request by the client the control was built using a well known logic manufacturer. In this case the choice was to use Siemens S7-1200 programmable logic. Using TIA Portal, the control was built for two stepper motors to control the drill and the drum. The drum would spin into a right position and the drill would drill the part. The whole sequence would be started again by a press of a button.

The results of this thesis were successful. The control is easily moveable and modifiable. The customer's wishes were taken into consideration and they were pleased with the result. If any modifications are required in the future, it will not require a significant amount of work.

---

Key words: engineer, automation, control, logic

## SISÄLLYS

1	JOHDANTO.....	6
2	TEORIA .....	7
	2.1.1 CNC-koneen teoria.....	7
	2.1.2 Siemens Simatic S7-1200 1212 DC/DC/DC-logiikka .....	7
	2.1.3 Simatic HMI KTP400 Basic-operointipaneeli.....	9
	2.1.3 Nema 23 askelmoottori ja CL57Y askelmoottoriohjain .....	10
	2.1.4 Siemensin suunnitteluohjelma TIA Portal ja WinCC.....	12
	2.1.5 Projektin luonti.....	18
	2.1.6 Työn toimintaperiaate ja kytkentäkuva.....	32
3	TYÖN SUORITUS .....	34
	3.1 Alustus.....	34
	3.1.1 Ohjauksen toteutus.....	34
	3.1.2 HMI-näyttöjen luonti.....	43
	3.1.3 Kasaus ja sähkökaappi.....	45
4	POHDINTA.....	47
	LÄHTEET .....	48

**LYHENTEET JA TERMIT**

CNC	Computerized Numerical Control
CPU	Central Processing Unit (Prosessori)
HMI	Human-Machine Interface (yhdistää käyttäjän laitteeseen)
I/O	Input ja Output
NEMA	National Electrical Manufacturers Association määrittelemä standardi
Nm	Newtonmetri eli vääntömomentti
Pinoli	Kara, joka liikuttaa poran terää
PLC	Programmable logic controller (Ohjelmoitava logiikka)
PN/IE	Profinet/Industrial Ethernet (verkkoyhteys)
V	Voltti

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena oli suunnitella ja rakentaa logiikkaohjaus automaattiselle työstökoneelle, tässä tapauksessa porakoneelle. Työ toteutettiin Par-Met Team Oy:lle, joka myy sen eteenpäin.

Par-Met Team on vuonna 1997 perustettu alihankintakonepaja, joka on erikoistunut suursarjojen automaattikoneistukseen. Parhaimmillaan se työllisti 4 henkilöä. Konekantana oli yhteensä 14 mekaanista- ja CNC-ohjattuja tankoautomaattisorveja. Nykyään Par-Met Team toimii konsulttoijana ja alihankkijana sekä antaa CNC-koneistuksen käyttöopastusta.

Työ sai alkunsa Par-Met Teamin saadessa tilauksen helposti muunneltavalle porausyksikölle suursarjaporauksiin. Asiakas käyttäisi sitä alkujaan sähkökaappien lukitustappien valmistukseen. Alun perin nämä oli tehty käsin ja työ oli hidasta. Yhden kappaleen tekeminen kesti noin 15 sekuntia. Automatisoidun porausyksikön avulla työ nopeutui viiteen sekuntiin kappale, mikä nostaa tuotannon kolminkertaiseksi.

Asiakas ei asettanut työlle suuria vaatimuksia: he halusivat vain ohjauksen, joka olisi helposti muunneltavissa ja tekisivät itse mekaanisen toteutuksen porakoneelle. Niinpä minulle jäi vapaat kädet valita sopiva logiikka itse. Tämä myös rajasi työn alaani hyvin sopivaksi. Minulle jäi automaatio- ja sähköpuoli, asiakkaalle mekaaninen toteutus.

Logiikaksi valikoitui Siemensin S7-1200-sarjalainen, koska sen käytöstä oli kokemusta Parmetilla ja halutessani pystyin kysymään neuvoa. Tavoitteena oli siis suunnitella ohjelma, jota ohjattaisiin Siemensin kosketusnäytöllä. Nappia painamalla se suorittaisi työkierron ja palautuisi takaisin alkuasetuksiinsa.

## 2 TEORIA

Tässä luvussa käydään läpi työn ja sen laitteiston teoria. Lisäksi perehdytään TIA Portalin käyttöön ja projektinluontiin. Lopuksi selvitetään laitteiston toimintaperiaate ja miten se on kytketty.

### 2.1 Työssä käytettävät laitteet

Työn laitteisto koostui Siemensin logiikasta, Siemensin kosketusnäytöstä, kahdesta askelmoottorista, kahdesta askelmoottorin ohjaimesta sekä 24V ja 36V tasavirtalähteestä. Seuraavissa kappaleissa esitellään laitteiden toimintaperiaatteet ja teoria.

#### 2.1.1 CNC-koneen teoria

Numeerinen ohjaus (CNC) on työstö- tai muissa koneissa käytetty yksiselitteinen symboliohjaus, jossa koneen ohjauselektroniikkaa muuttaa ne servomoottorien liikkeiksi. CNC-kone on siis tietokoneavustettu yleiskone, joka suorittaa sille ohjelmoitua työtä. CNC-koneiden hyödyt ovat suuret verrattuna manuaaliseen työhön. Koneen työstökierron alettua se ei tarvitse muuta kuin satunnaista materiaalin uudelleen lisäystä.

Numeerinen ohjaus käyttää M- ja G-koodia. G-koodi kertoo, kuinka koneen tulee liikkua. Sitä voisi siis kutsua geometriseksi koodiksi. M-koodi kertoo esimerkiksi, milloin kone pysähtyy ja vaihtaa kappaletta (Andreas Velling 2020).

#### 2.1.2 Siemens Simatic S7-1200 1212 DC/DC/DC-logiikka

Työn logiikaksi valikoitui Siemensin valmistama S7-1200 1212 DC/DC/DC-logiikka. Logiikkaan päädyin sen hinnan ja kompaktiuden vuoksi. Lisäksi työssä ei tarvittu paljon tuloja ja lähtöjä eikä ollenkaan analogiaportteja.

Ohjelmoitavalla logiikalla tarkoitetaan automaatiossa pientä tietokonetta, jota käytetään ohjaamaan erilaisia prosesseja. Se kehitettiin 1960-luvulla korvaamaan releohjaukset. Monimutkaisten kytkentöjen muuttamisen sijaan käyttäjä voi vain muuttaa logiikan ohjelmaa (Carlos Gonzalez 2015; Vidya Muthukrishnan 2021)

Ohjelmoitava logiikka sisältää pienen CPU:n, joka toimii PLC:n aivoina. Se ohjaa koko laitteen toimintaa. PLC:n tapauksessa puhutaan usein I/O-osoitteista. Näillä tarkoitetaan tuloja (input) ja lähtöjä (output). Tuloporttien kautta logiikka saa tietoa prosessin tilasta ja ohjaa sitä lähtöjen kautta (Carlos Gonzalez 2015; Vidya Muthukrishnan 2021).

I/O-portit voidaan jakaa digitaalisiin ja analogisiin. Digitaaliset saavat joko arvon 1 tai 0 eli päälle tai pois. Niillä voidaan ohjata esimerkiksi venttiilien toimintaa. Analogisia käytetään, kun halutaan esimerkiksi säätää moottorin kierrosnopeutta. Nämä arvot näyttyvät logiikassa usein kokonaislukuina.

Siemensin 1200-sarja on suunniteltu pienten ja keskikokoisten automaatioprojektien keskuksiksi. Se sisältää integroidut I/O:t. Eri 1200-sarjan logiikkamalleja on monia, mutta suurimmat erot niiden kesken ovat muistin määrä, I/O-porttien määrä sekä laajennusmahdollisuudet. Jokaisesta on lisäksi tarjolla kolme vaihtoehtoa: DC/DC/DC, DC/DC/Relais ja AC/DC/Relais. Nämä eroavat toisistaan sillä, että kaksi jälkimmäistä sisältää releohjauksen ja viimeinen ottaa virransa vaihtojännitteestä (S7 1200-sarjan käyttöohje).

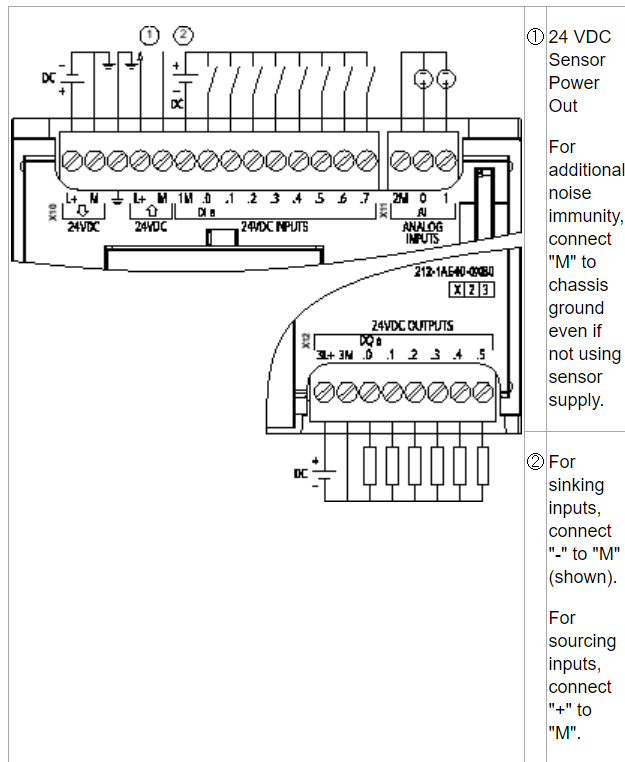


KUVIO 1. Simatic s7-1200 1212 DC/DC/DC (Siemens)



1212 DC/DC/DC sisältää kahdeksan digitaalituloa ja kuusi digitaalilähtöä (kuvio 1). Analogiatuloportteja siinä on kaksi. Se on laajennettavissa erilaisilla moduuleilla, jotka lisäävät tuloja ja lähtöjä. Virtansa se ottaa 24V tasavirrasta. Logiikka ohjataan Siemensin Tia Portal-ohjelmalla, josta lisää myöhemmin. Kuviossa 2 on esitetty logiikan kytkentäkuva.

Table: CPU 1212C DC/DC/DC (6ES7-212-1AE40-0XB0)



KUVIO 2. Logiikan kytkentäkuva (Siemens)

### 2.1.3 Simatic HMI KTP400 Basic-operointipaneeli

Näytöksi, jolla projektia ohjataan, valikoitua Siemensin KTP400 Basic-paneeli (kuvio 3). Se on kosketusnäyttölinen ja siihen voi vapaasti konfiguroida eri painikkeita ja toimintoja. Se sisältää kaikki HMI-perustoiminnot eli kielen vaihdon, signaalijärjestelmän, vastaanoton hallinnan ja kaaviotoiminnot. Sen kytkentä tapahtuu Profinet- tai Ethernet-yhteyden avulla. Paneelin ohjelmointi tapahtuu Siemensin TIA Portalilla, josta lisää myöhemmin.



KUVIO 3. KTP400 näyttö (Siemens)

### 2.1.3 Nema 23 askelmoottori ja CL57Y askelmoottorihjain

Moottoreiksi työhön valittiin kaksi 1,2 Nm Nema 23 askelmoottoria takaisinkytkennällä. Molemmat moottorit vaativat 36V virtalähteen. Askelmoottorin toiminta perustuu elektromagnetismiin. Sähköpulssi luo magneettikentän, joka saa roottorin pyörimään. Pulssien ansiosta roottorin liike on helppo ennakoida, mikä tekee askelmoottoreista hyvin soveltuvia erilaisiin tarkkuutta vaativiin sovelluksiin (What is a Stepper Motor: Types & Its Working).

Nykyisin askelmoottorit voidaan jakaa kolmeen eri kategoriaan. Näitä ovat Permanent Magnet (PM), Variable Reluctance (VR) ja Hybrid (HY). PM-moottorit omaavat suuren väännön, mutta heikon tarkkuuden. Ne ovat tavallisesti kaksivaiheisia. Moottorissa on ympyrässä käämejä, joista joka toinen saa positiivisen virran, seuraava ei saa virtaa ollenkaan ja seuraava saa negatiivisen virran. Käämien keskellä on levynä magneetti, joka on sähköistetty samalla tavalla kuin käämit, mutta eri järjestyksessä. Muuttamalla virran suuntaa saadaan levy kääntymään suhteessa käämeihin, sillä vastakkaisesti magnetisoidut navat vetävät toisiaan puoleensa (Matthew Scarpino 2016).

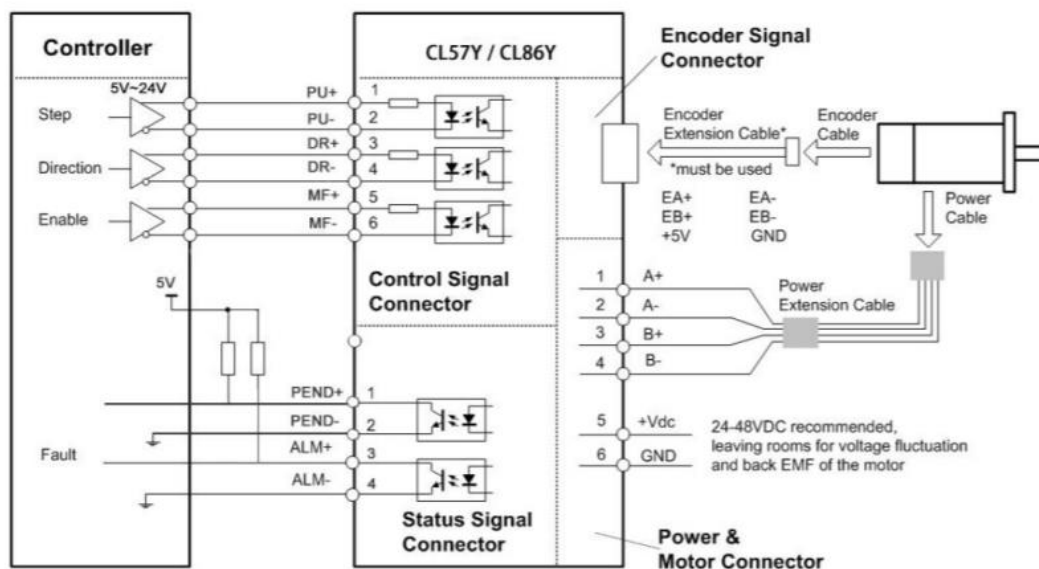
VR-moottorien toiminta perustuu samaan toimintaperiaatteeseen kuin yllä esitellyn PM-moottorin. Ne eroavat toisistaan rakenteen puolesta. VR-moottorissa ei ole magneetteja vaan pyörivässä levyssä on pienet hampaat. Levy on tehty helposti magnetisoituvasta materiaalista ja näin ollen se pyrkii kääntymään kää-

miä vasten magnetisoitaessa. Roottorin rakenteesta johtuen sen vääntömomentti on huomattavasti pienempi kuin PM-moottorilla. Tästä syystä VR-moottoreja ei usein käytetä teollisuudessa (Jori Turja 2017).

Hybridimoottorit yhdistävät molempien PM- ja VR-moottorien hyvät puolet. Siinä on PM-moottorin suuri vääntömomentti ja VR-moottorin korkea resoluutio ja nopeus. Tämä saadaan aikaan samanlaisella hampaistuksella kuten VR-moottorissa, mutta siihen on PM-moottorin tavoin lisätty myös kestopagneetti. Hybridimoottorin huonoja puolia ovat kalliimmat kustannukset ja suurempi koko.

Työssä käytettävä Nema 23 askelmoottori oli tyypiltään bipolar. Tämä tarkoittaa, että toimiakseen virran suuntaa on muutettava navan polarisuuden vaihtamiseksi. Tähän tarvitaan askelmoottoriohjainta. Askelmoottori sisälsi sisäänrakennetun pulssiohjaimen, jonka ansiosta takaisinkytkentä oli mahdollinen. Ohjain laskee pulssien suhdetta akselin liikkeeseen ja välittää sen askelmoottoriohjaimelle (What is a Stepper Motor: Types & Its Working).

Takaisinkytkennällä tarkoitetaan prosessin lähtösignaalin ohjaamista takaisin sen tulosignaaliiksi. Prosessin aiemmin tuottama informaatio vaikuttaa siis sen myöhempään informaatioon. Tämän ansiosta prosessi voi käyttää lähtötietoa säädön parantamiseen ja häiriöiden korjaamiseen.



KUVIO 4. Askelmoottorin kytkentä (OMC-Stepperonline)

CL57Y askelmoottoriohjain vastaanottaa asematiedon ja vertaa sitä haluttuun asemaan. Haluttu asema saadaan logiikalta. Ohjaimesta pystytään säätämään haluttu pulssimäärä kierrokselle eli montako pulssia tarvitaan yhteen kokonaiseen akselin kierrokseen. Haluttu arvo pystyttiin säätämään dippikytkimillä (kuvio 5). Tässä työssä käytettiin arvoa 1000.



KUVIO 5. Askelmoottoriohjain (OMC-Stepperonline)

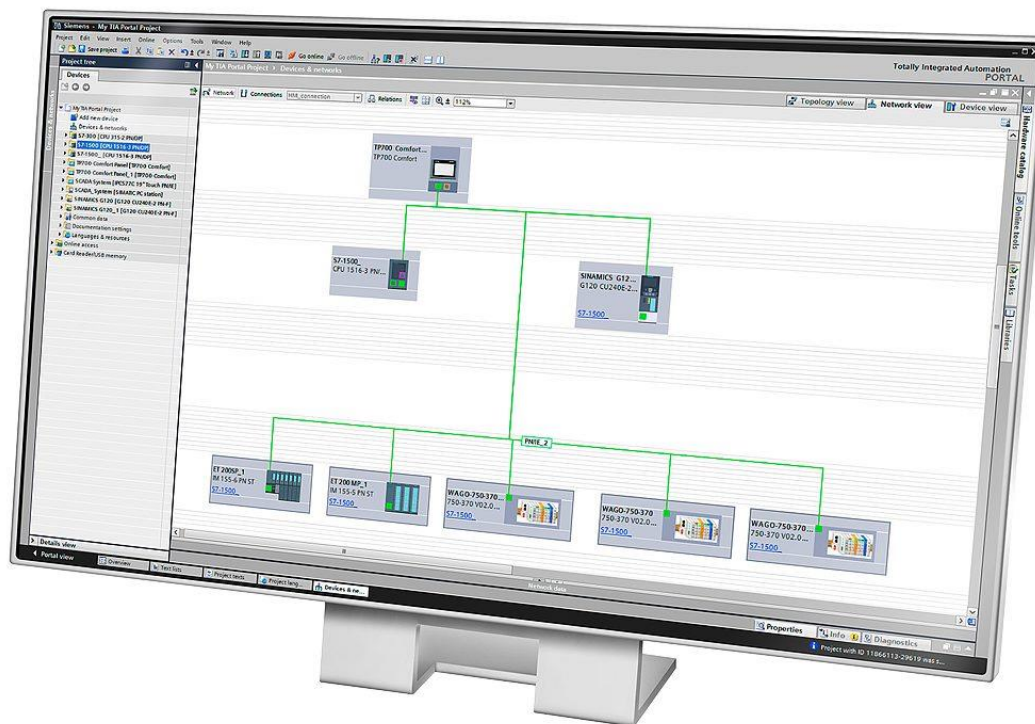
## 2.1.4 Siemensin suunnitteluohjelma TIA Portal ja WinCC

TIA Portal eli Totally Integrated Automation Portal on Siemensin kehittämä suunnitteluohjelmisto, joka yhdistää kaikki sen ohjelmistot yhteen. Tämän työn kannalta oleelliset ovat logiikkaohjelmointiin käytettävät Step7 ja HMI-näyttöjen ohjelmointiin käytettävä WinCC (Siemens 1200-sarjan logiikat).

Simatic Step7 on levinnein suunnitteluohjelmisto maailmalla. Sitä käytetään ohjelmoimaan Siemensin PLC-sarjoja. Sen avulla voidaan suunnitella esimerkiksi kokonainen tehdas. Ohjelma voidaan jakaa kolmeen eri osaan, jotka esitellään seuraavaksi.

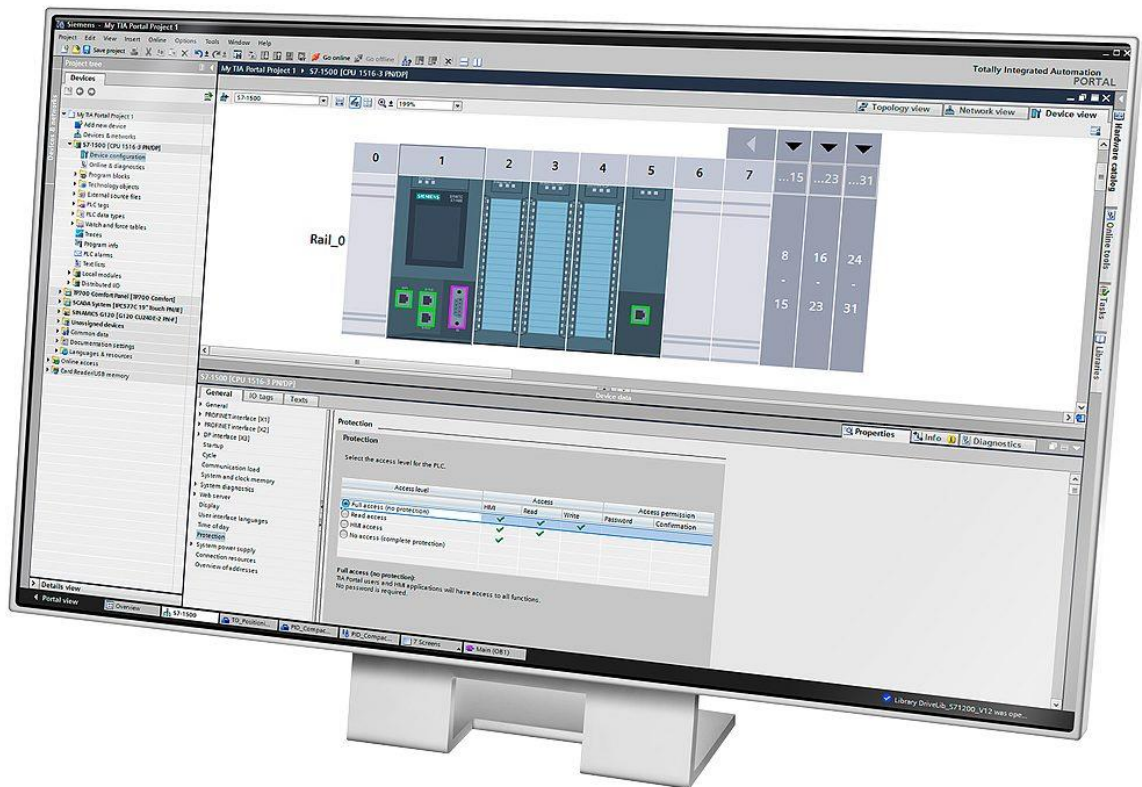
Network-näkymässä kommunikaatiolinkit laitteiden välillä yhdistetään. Tässä näkymässä siis otetaan käyttöön tehtaan kommunikointiohjelma. Laitteiden valinnan jälkeen ohjelma yhdistää ne automaattisesti tai käyttäjä voi yhdistää ne itse (kuvio 6). Näkymä selkeyttää erityisesti suuria automaatioprojekteja monilla

eri logiikoilla, näytöillä ja toimilaitteilla. Näkymässä vihreät viivat tarkoittavat PN/IE-liityntää ja violetit profinettiä.



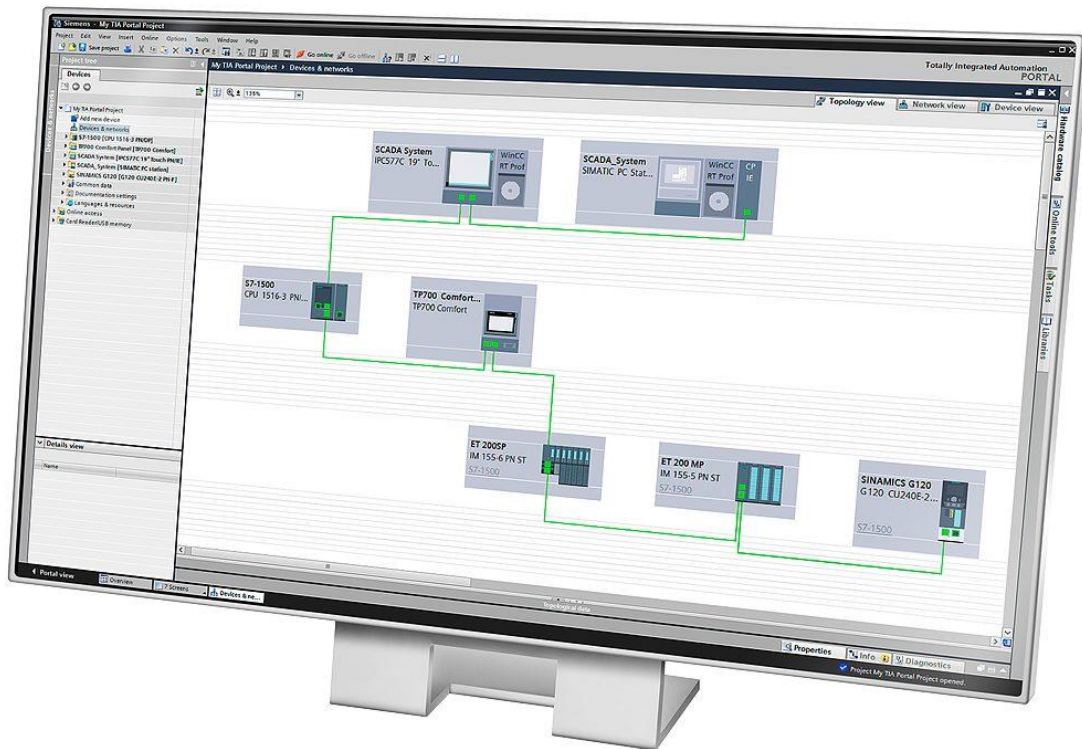
KUVIO 6. Network-näkymä (Siemens)

Laitenäkymässä (kuvio 7) käyttäjä voi konfiguroida I/O-räkkejä sekä asettaa niiden I/O-osoitteita. Näkymä vastaa oikeaa logiikkaa täsmällisesti, mikä tekee konfiguroinnista helppoa. Käyttäjällä voi raahata osoitteita ohjelmasta suoraan I/O-portteihin. Tässä näkymässä voi logiikkaan lisätä uusia tulo- ja lähtöportteja. Lisäksi käyttäjä voi asettaa logiikan eri asetuksia kuten IP-osoitteen.



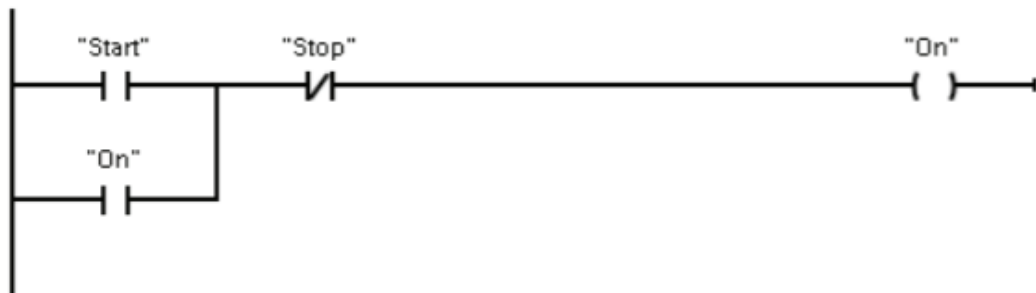
KUVIO 7. Laitenäkö (Siemens)

Topology-näkymä on suunniteltu Profinet-verkkoa varten. Se muistuttaa ulko­näöllisesti ylempänä mainittua laitenäkymää, mutta siitä näkee mitkä laitteet kommunikoivat keskenään ja mitkä ovat ulkona prosessista. Topology-näkymää käytetään myös yksittäisten porttien yhdistämiseen.



KUVIO 8. Topologynäkymä (Siemens)

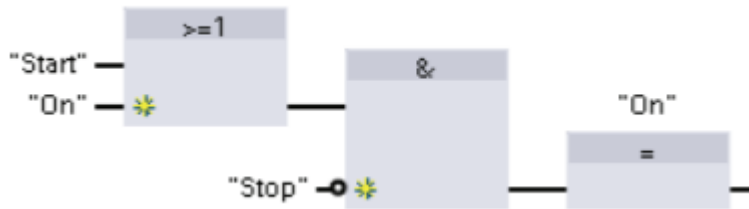
Tia Portal sisältää neljä eri ohjelmointikieltä: LAD, FBD, SCL ja STL (S7-1200-sarjan käyttöohje). LAD eli ladder on niin kutsuttu tikapuukaavio. Se on graafinen ohjelmointimuoto, jossa koskettimet saavat arvoja 1 tai 0.



KUVIO 9. Tikapuukaavio esimerkki (Siemens)

FBD eli Function Block Diagram on myös graafinen ohjelmointitapa. Tikapuukaavion sijaan siinä luodaan Boolean algebraan perustuvia logiikkasymboleja.





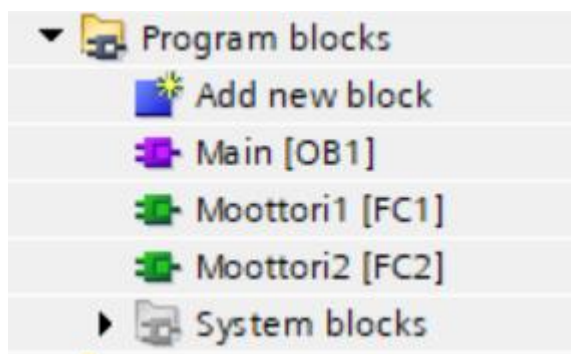
KUVIO 10. FBD esimerkki (Siemens)

SCL eli Structured Control Language on PASCAL-tyyppinen ohjelmointikieli. Se käyttää normaaleja matemaattisia merkkejä, kuten + ja =. Se käyttää myös sanoja kuten IF-THEN-ELSE.

Ohjelmointikielen valinnan jälkeen perehdytään Tia Portalin ”Blockeihin” eli lohkoihin, mihin ohjelma tehdään. OB eli organisointilohkot ovat päälohkoja, jotka määrittelevät ohjelman rakenteen. Niiden tukemiseksi käytetään funktioita (FC), funktiolohkoja (FB) sekä datalohkoja (DB).

FC ja FB-lohkoihin voi sijoittaa sen osan ohjelmaa, joka suorittaa erityistä ohjelmaa, esimerkiksi moottorinohjausta. FB-lohko voi myös käyttää Data blockia varastoimaan arvoja muiden lohkojen käytettäväksi. Molempia FC- ja FB-lohkoja voi kutsua päälohkosta eli suorittaa ne sieltä. Myös muita FC-ja FB-lohkoja voi kutsua muista FC-ja FB-lohkoista. Datalohkoja käytetään varastoimaan dataa. Niitä käyttävät ohjelman muut lohkot.

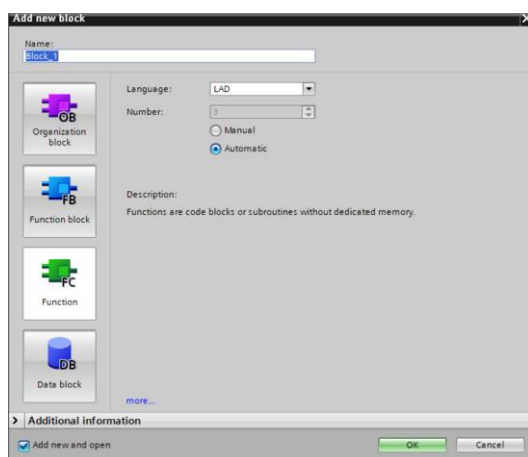
Lohkon valintaan päästään valitsemalla Add new block (kuvio 11).



KUVIO 11. Uuden lohkon lisäys sekä valmiita ohjelmalohkoja



Avautuu kuvion 12 mukainen valikko, josta päästään valitsemaan haluttu lohko ja ohjelmointikieli.



KUVIO 12. Lohkotyyppin sekä ohjelmointikielen valinta

Työn kannalta oleellisia olivat datan sijoituspaikat. Näistä tärkeimmät olivat Glo-baalin muistin osoitteet I (Input), Q (Output) ja M (Bittimuisti.) Näistä M-alueen muisti on kaikkien toimintalohkojen käytettävissä ilman rajoituksia.

I ja Q-muistit ovat todellisen prosessin kuvia. Ne määritellään käytettävän PLC:n mukaan. Kun prosessi suoritetaan, ohjelma kopioi muistipaikan arvon lo-giikan fyysisiin osoitteisiin. M-muistia käytetään varastoimaan säätö- ja data-muistia.

Osoitteita, joita alueille on käytettävä ovat esimerkiksi %I1.0 ; %Q0.2 ja %M0.1. Nämä osoitteet kirjoittavat vain tietyn bitin. Alueille voidaan kirjoittaa myös Byte (kahdeksanosainen bitti), Word (16-bittinen arvo) tai DWord (32-bittinen tupla-sana arvo.)

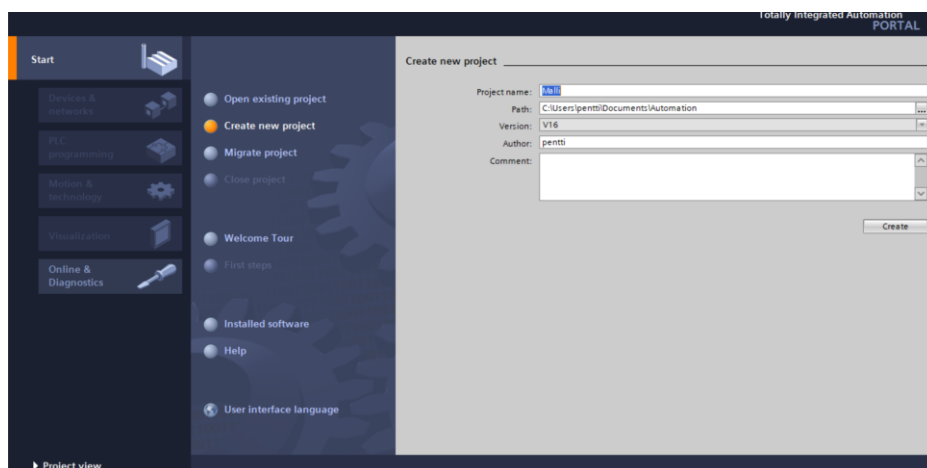
Kaikkia luotuja arvoja kutsutaan tageiksi. Ne löytää logiikan Tag tablesta (kuvio 13). Täältä on helppo muokata niitä ja luoda niitä lisää. Ohjelman tekoon hyvä aloittaa luomalla sille ensiksi halutut tagit.

PLC tags									
	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	C
1	Axis_1_Pulse	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Axis_1_Direction	Default tag table	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	start	Default tag table	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Axis_2_Pulse	Default tag table	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Axis_2_Direction	Default tag table	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Moottori1 Kierto	Default tag table	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Moottori1 Kierto takaisin	Default tag table	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Tag_1	Default tag table	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Tag_2	Default tag table	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Tag_3	Default tag table	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Tag_4	Default tag table	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Moottori2 kierto	Default tag table	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Tag_5	Default tag table	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Tag_6	Default tag table	Bool	%M2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Tag_7	Default tag table	Bool	%M2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Hätäseis	Default tag table	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Tag_8	Default tag table	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Syliinteri ON	Default tag table	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	Jog	Default tag table	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

KUVIO 13. Työn Tag table

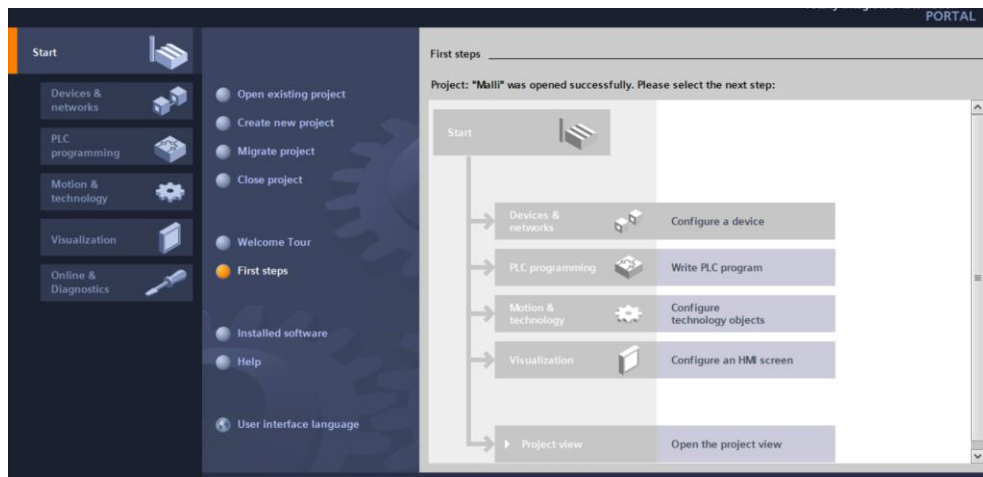
### 2.1.5 Projektin luonti

Tässä luvussa käydään läpi TIA Portalin yleistä toimintaa ja miten siinä luodaan uusi projekti. Esimerkkinä toimii opinnäytetyössä luotu PLC-ohjelma. Ohjelman luonti Tia Portalissa aloitetaan painikkeella ”Create new project.”



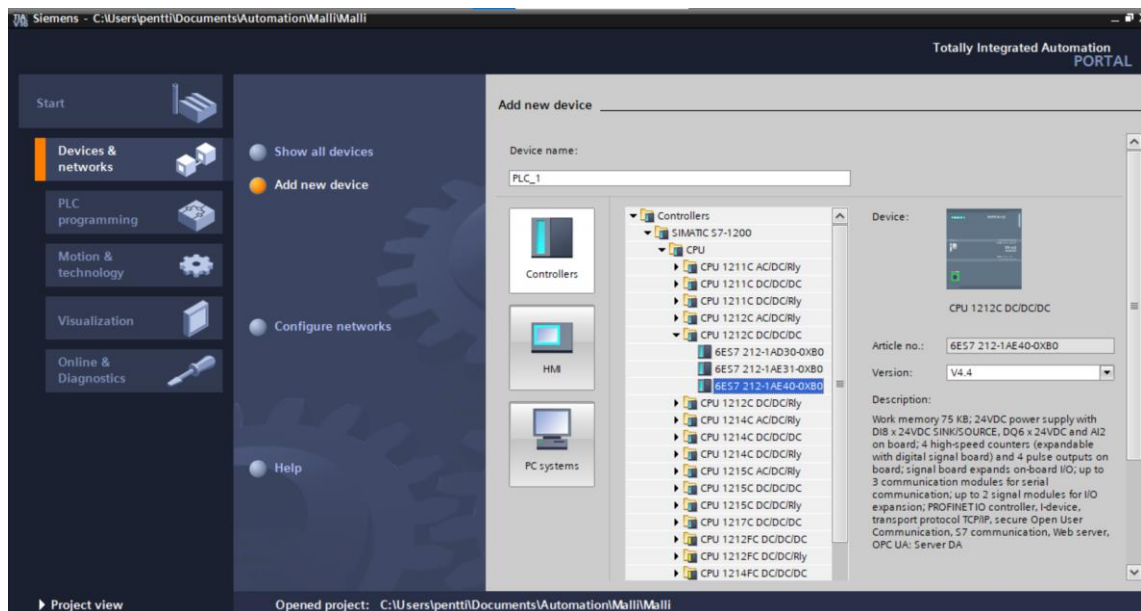
KUVIO 14. Aloitusnäkömä

Seuraavaksi valitaan ”Device & Network”, josta päästään valitsemaan haluttu laite (kuvio 16).



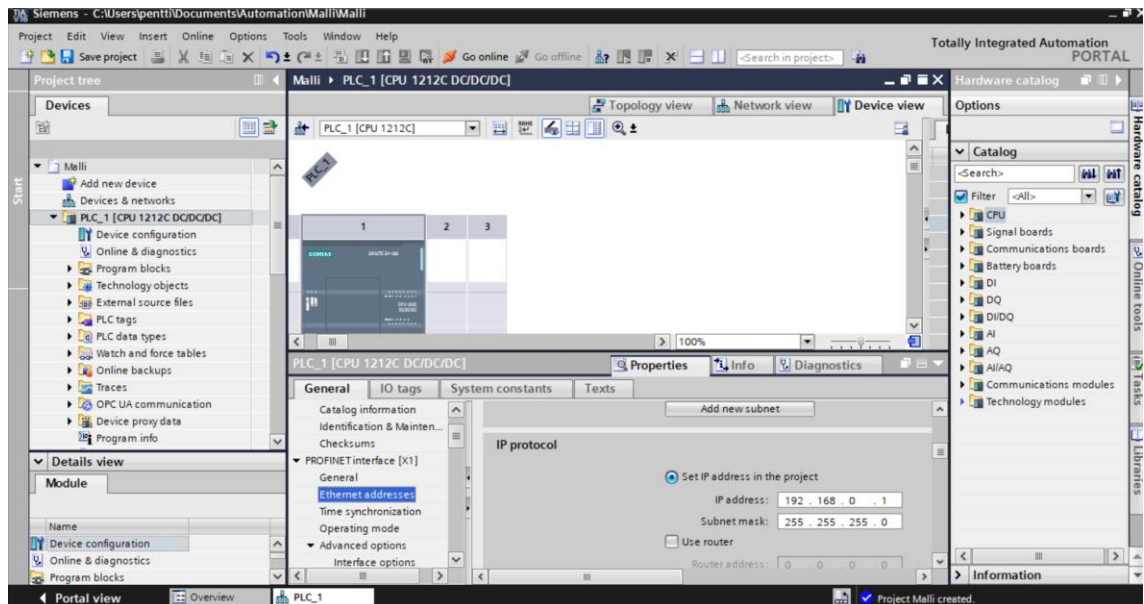
KUVIO 15. Projektin luonti

Valikosta löytyy kaikki Siemensin PLC-laitteet sekä HMI-näytöt. Voidaan myös valita ”Unspecified PLC”, jolloin myöhemmin voidaan hakea oikea PLC-malli kytkemällä se tietokoneeseen.



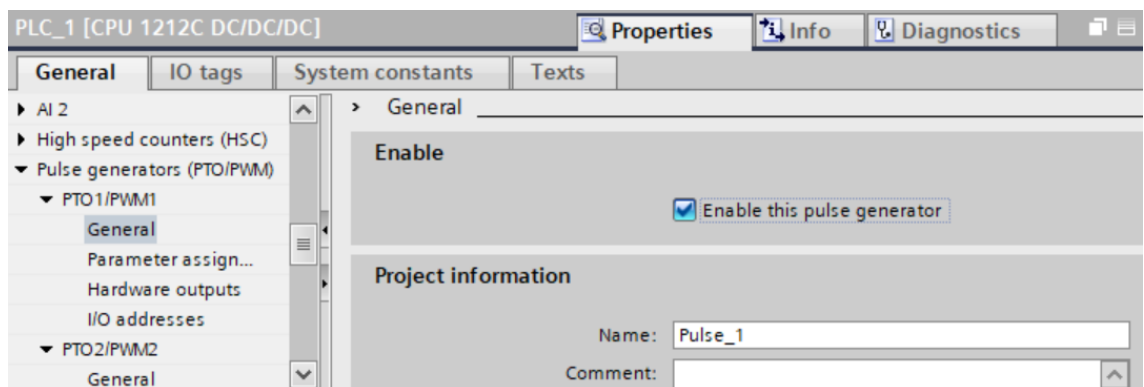
KUVIO 16. Laitteen valinta

Oikean PLC:n löydyttyä valitaan se ja lisätään se ohjelmaan ”Add new device” painikkeella. Tämän jälkeen päästään kuvion 17 esittämään tilaan. Klikataan loogikka aktiiviseksi, jolloin alareunaan aukeaa sen asetukset. Täältä valitaan ”Properties” ja sieltä Profinet Interface ja Ethernet addresses. Täältä päästään asettamaan loogikalle oma IP-osoite. Huomioitavaa on, että kytkettäessä loogikka tietokoneeseen ja HMI-näyttöön, tulee kaikilla laitteilla olla eri IP-osoite.



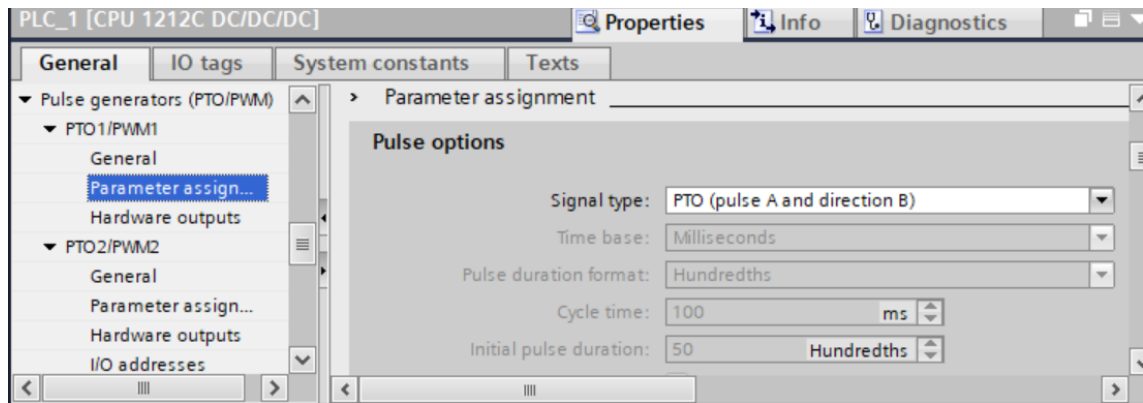
KUVIO 17. IP-osoitteen asetus

Jatketaan Properties-valikkoa alaspäin ja valitaan Pulse Generator (PTO/PWM). Tämä valinta ottaa TIA Portalissa käyttöön Technology Objectin, jonka avulla päästään ohjaamaan askelmoottoria (kuvio 21). Täältä päästään luomaan askelmoottorille tarvittava pulssigeneraattori. Valitaan "Enable", jolloin Tia Portaliin ilmestyy "Techonology object" (Kuvio 21).



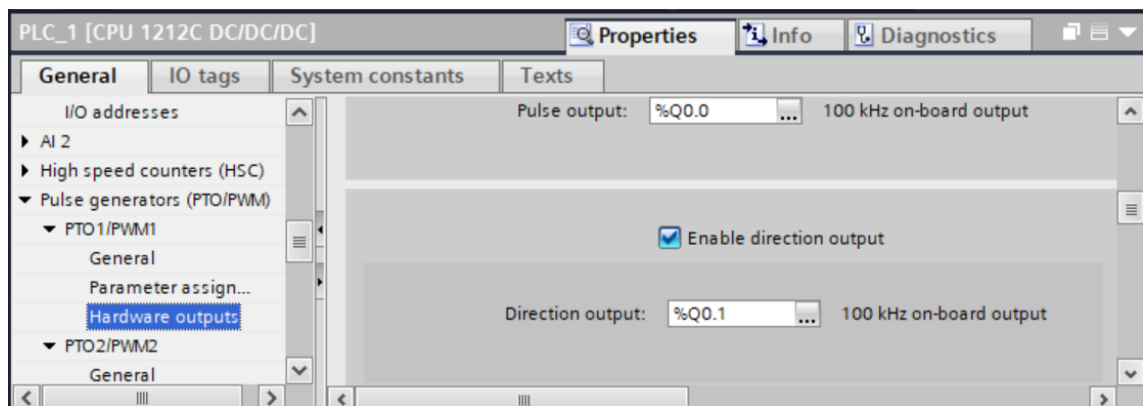
KUVIO 18. Pulssigeneraattorin käyttöönotto

Parameter assignment-valikosta signaalityypiksi tulee valita PTO eli Pulse Train Output. Tämä siis lähettää viestin moottorille pulssina.



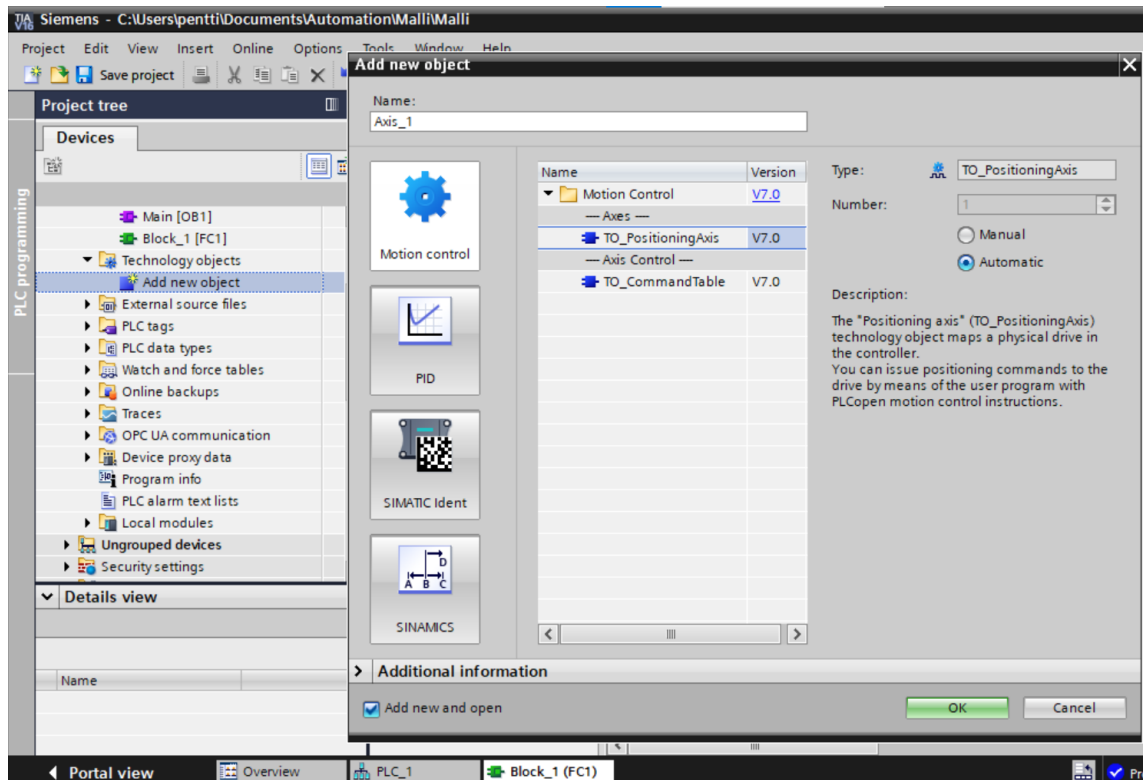
KUVIO 19. PTO-pulssin valinta

Hardware outputs päästään valitsemaan pulssille ja sen suunnalle lähtöosoitteet. Pulssin askel tarvitsee 100 kHz taajuuden, joten lähtöosoitteet tulee valita oikein. Koska työssä käytettiin vain kahta moottoria ei ongelmia ole, sillä logiikan neljä ensimmäistä lähtöä (Q0.0-Q0.3) ovat kaikki 100 kHz. Jos moottoreita olisi enemmän tulisi lähdöt valita siten, että jokaisen moottorin askel sijoittuisi neljälle ensimmäiselle lähdölle ja suunnat loppuille.



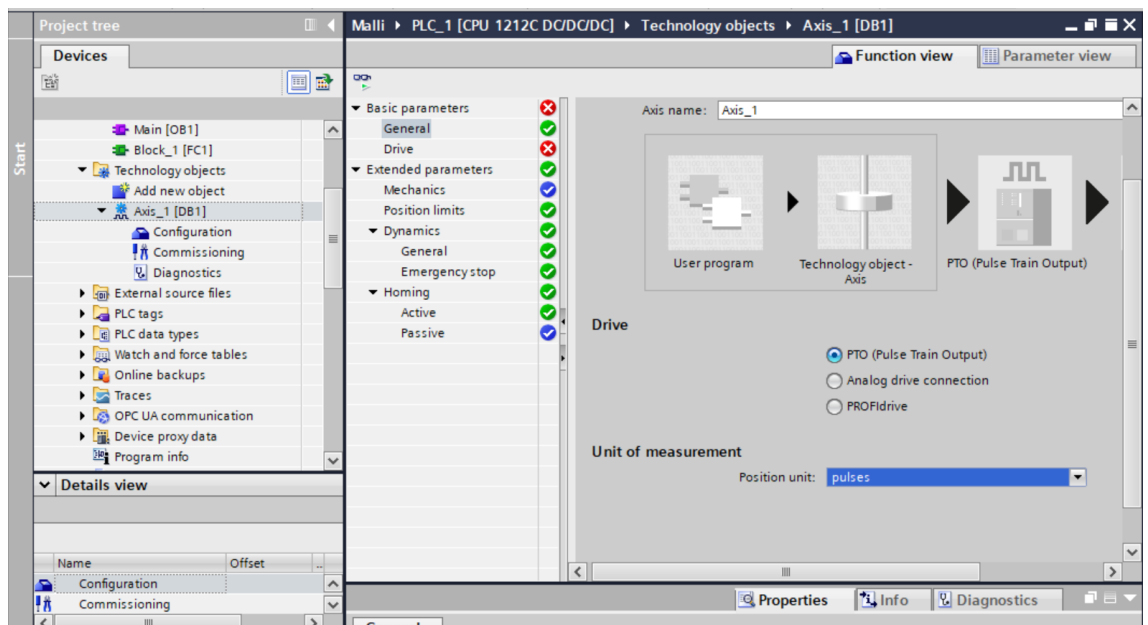
KUVIO 20. Lähtöosoitteiden määrittäminen

Nyt on saatu luotua sopivat asetukset, jotta päästään luomaan teknologiaobjekteja (Kuvio 21). Luodaan uusi objekti ja valitaan sen tyyppi ”motion control.” Valitaan ”To\_PositioningAxis”, koska tämä luo fyysisen ohjauksen säädölle ja sille voidaan antaa käskyjä käyttäjän tekemästä ohjelmasta.



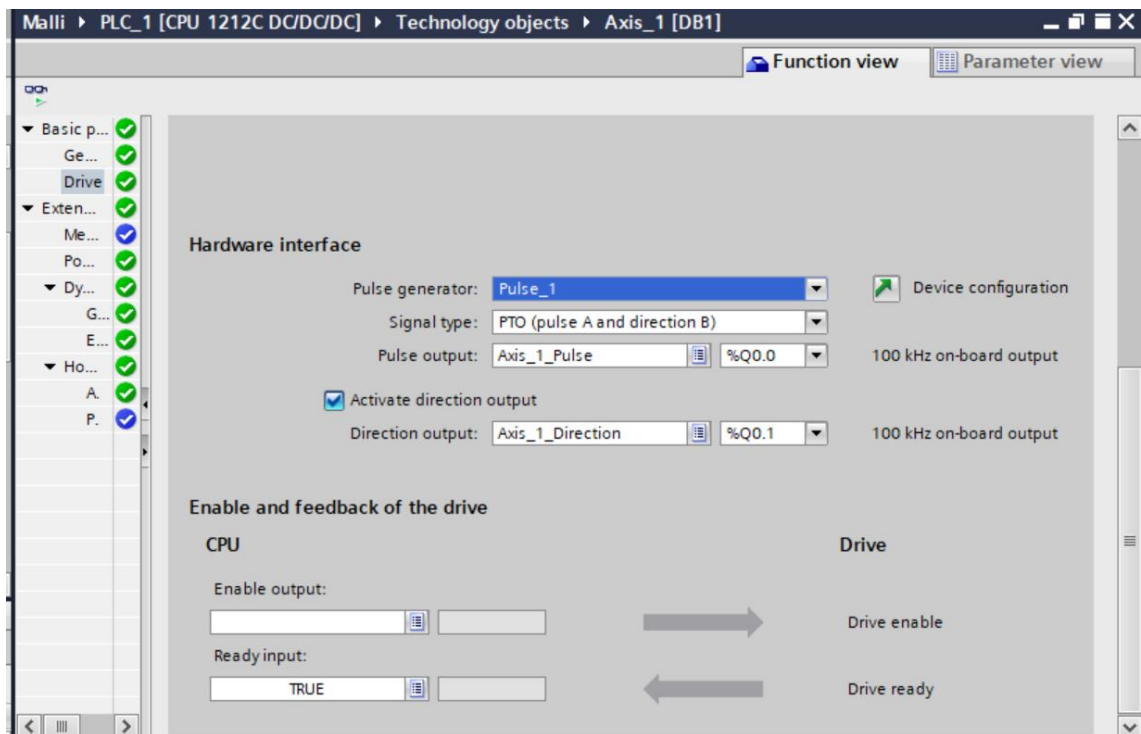
KUVIO 21. Axis-lohkon luonti

Akselin luonnin jälkeen siirryttiin sen asetuksiin. Configuration valikosta mennään General ja valitaan tyypiksi jälleen PTO. Mittayksiköksi valitaan pulssit, sillä moottoreita on helppo ohjata, kun tiedetään, kuinka monta pulssia yksi kierros on.



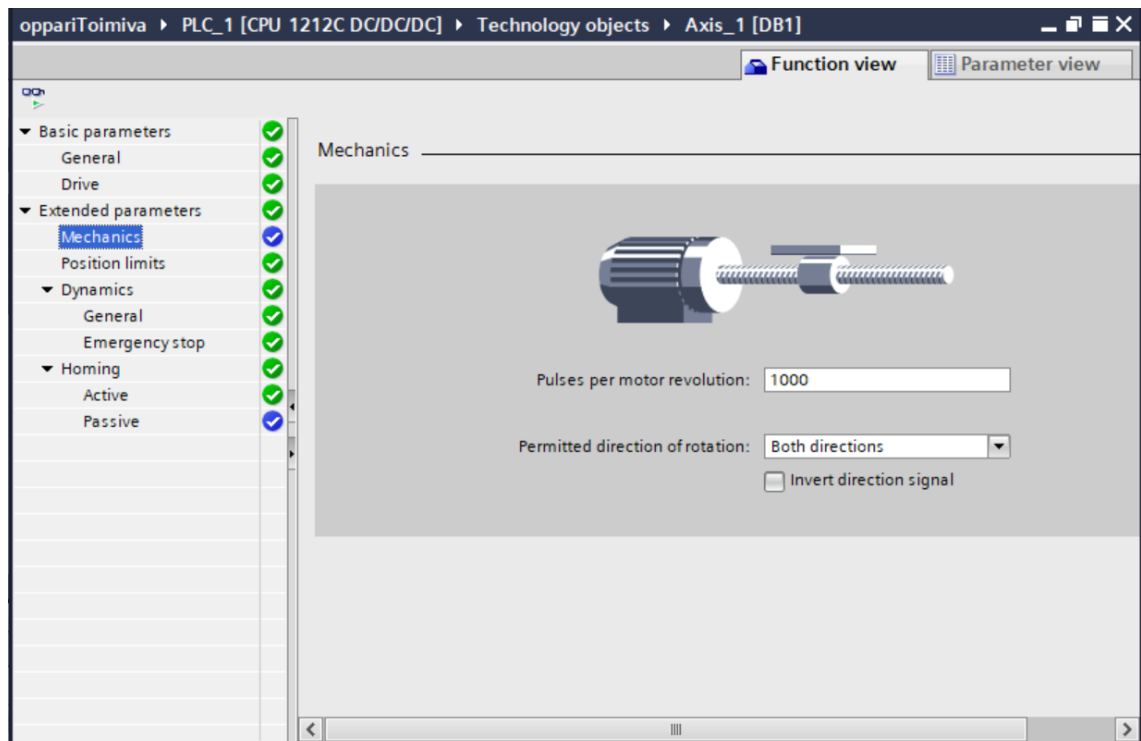
KUVIO 22. Yleisasetukset

Drive-valikosta valitaan, mitä pulssia säädetään. Tässä tapauksessa se on Pulse\_1, mutta moottoreiden lisääntyessä valikosta löytyy Pulse\_2 jne. Tässä voi myös nimetä edellä jo asetetut lähtöosoitteet pulssin askeleelle ja suunnalle. Enable- ja feedbacklähtöjä ei otettu käyttöön.



KUVIO 23. Pulssin valinta

Mechanics-valikosta (Kuvio 24) määritellään, kuinka monta pulssia yksi kierros on. Tässä tapauksessa käytettiin arvoa 1000, sillä askelmoottorin ohjain tuki sitä. Permitted direction of rotation määrittelee mihin suuntaan moottori pyörii. Tässä tapauksessa valittiin, että molempiin suuntiin, sillä toisen akseleista pitää pystyä palaamaan takaisin. Valinnalla ei lisäksi ole suurta väliä, sillä suunta voidaan myöhemmin asettaa Motion Control-lohkoista.

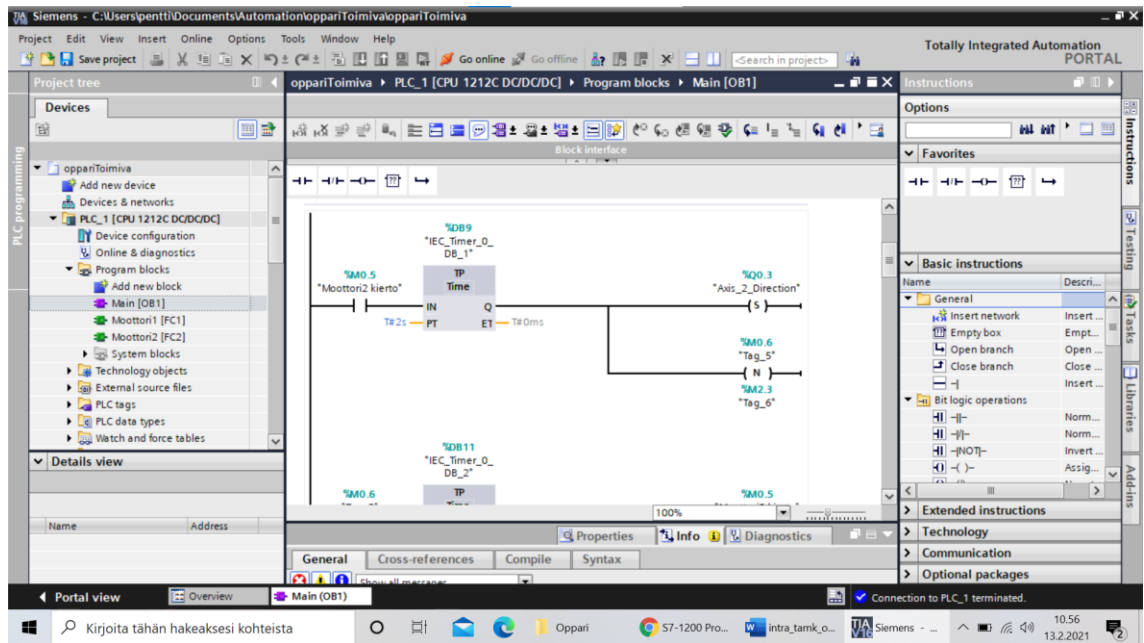


KUVIO 24. Pulssin mekaniikat

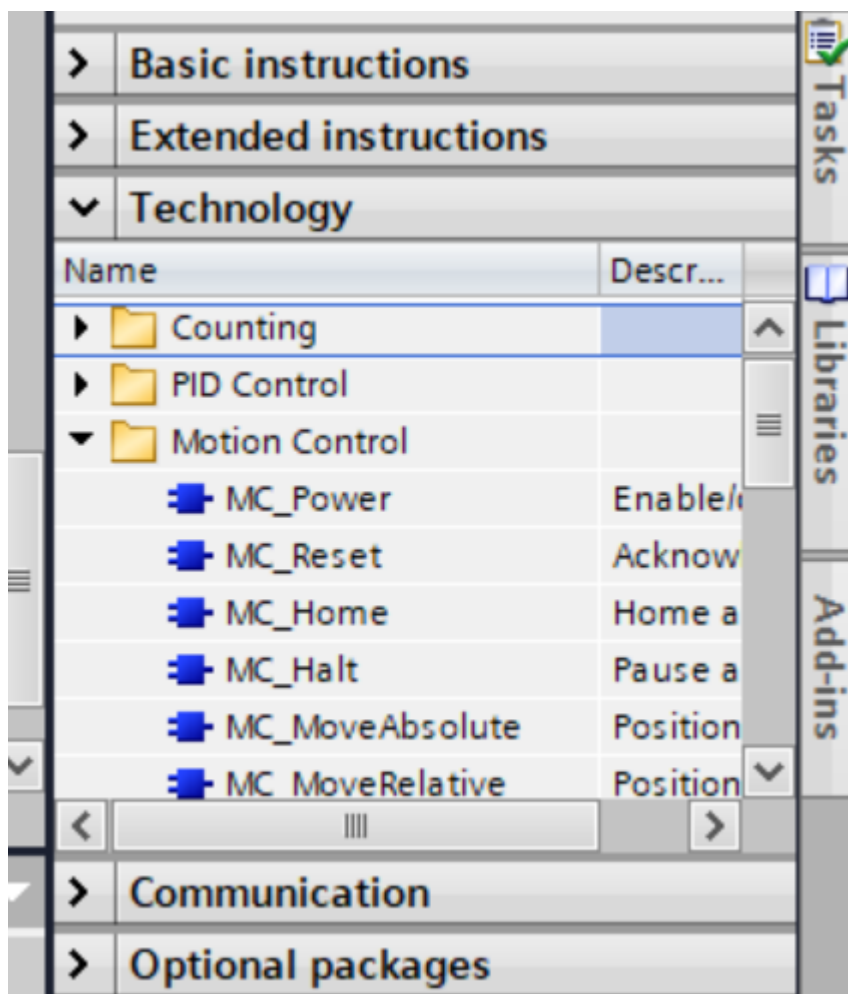
Muihin asetuksiin ei tarvitse koskea. Niistä voi muun muassa säätää pulssin leveyttä ja tiheyttä.

Kun akseli on luotu, voidaan siirtyä itse ohjelmointiin. Main Blockin alle luodaan uusi funktio (FC), jonne rakennetaan moottorin ohjaus. Sitä voi sitten kutsua Main Blockista tarpeen tullen. Moottorilohkojen lisääminen tapahtuu TIA Portalissa oikealta Technology ja Motion Control-valikkojen alta (Kuvio 26). Yleisimmin käytettyjä lohkoja löytyy Basic Instructions-valikosta (kuvio 25). Kaikki lohkot yksinkertaisesti raahataan paikoilleen.





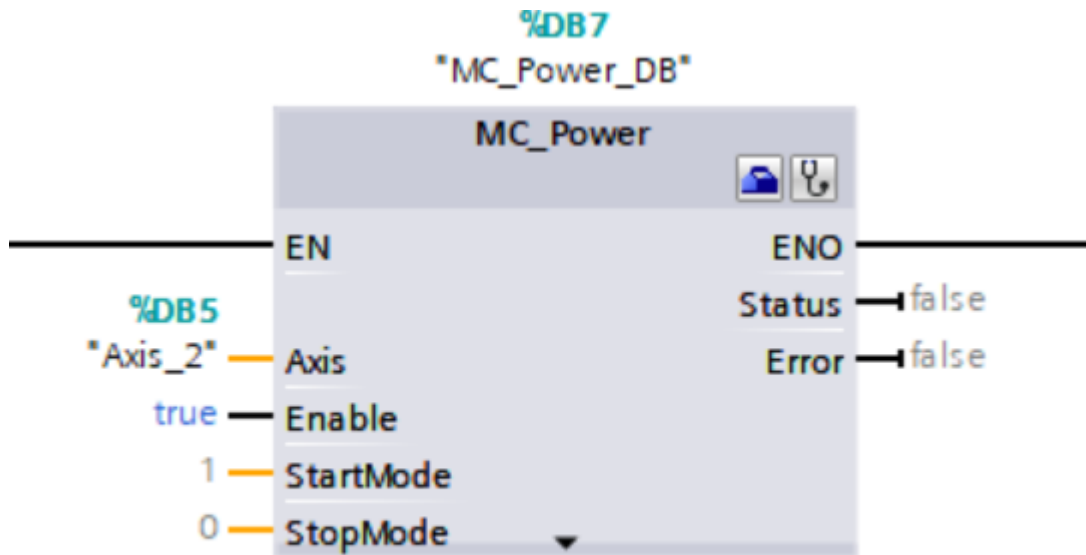
KUVIO 25. Tia Portal yleisnäkymä ja Basic Instructions



KUVIO 26. Moottorinohjauslohkot

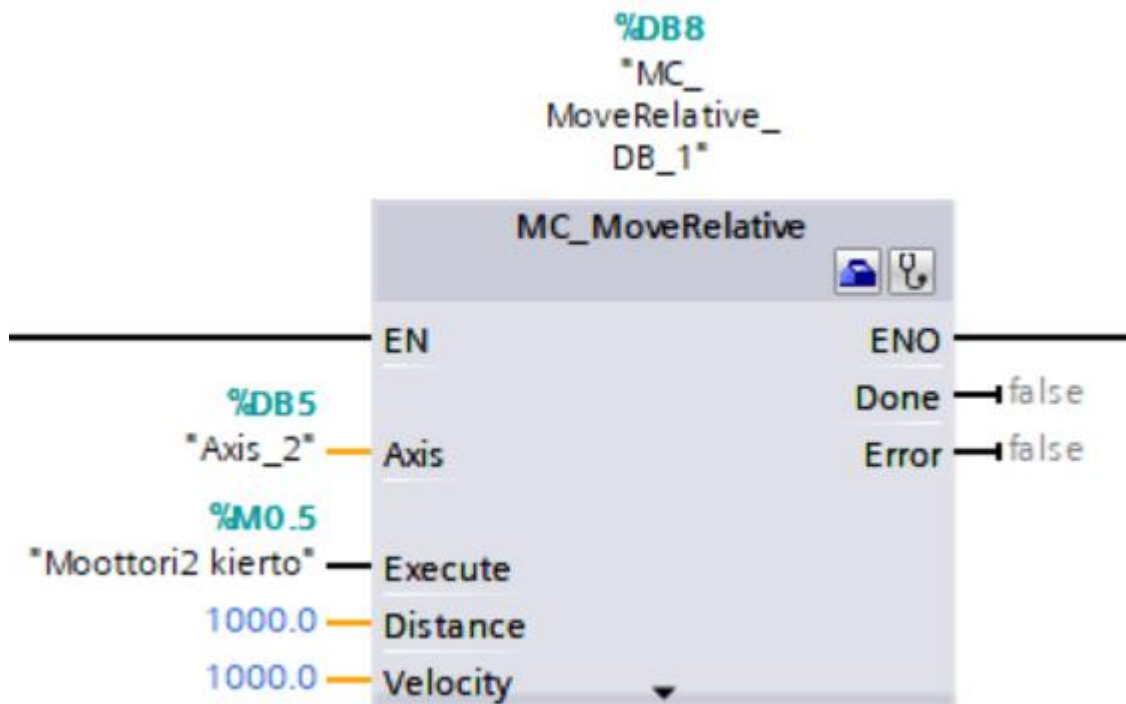
Lohkoja akselin ohjaukseen on monia, mutta tässä työssä käytettiin lohkoja: MC\_Power, MC\_MoveRelative ja MC\_MoveJog.

MC\_Power lohkoa tarvittiin yksinkertaisesti ottamaan moottori käyttöön. Ilman sitä moottori ei saisi virtaa (kuvio 27). Axis-kohtaan tulee valita edellä luotu akseli, mitä halutaan ohjata. Enable-bittiä ei työssä ollut käytössä, joten lohkon kohtaan kirjoitettiin "true" eli tosi. Näin se oli koko ajan päällä.



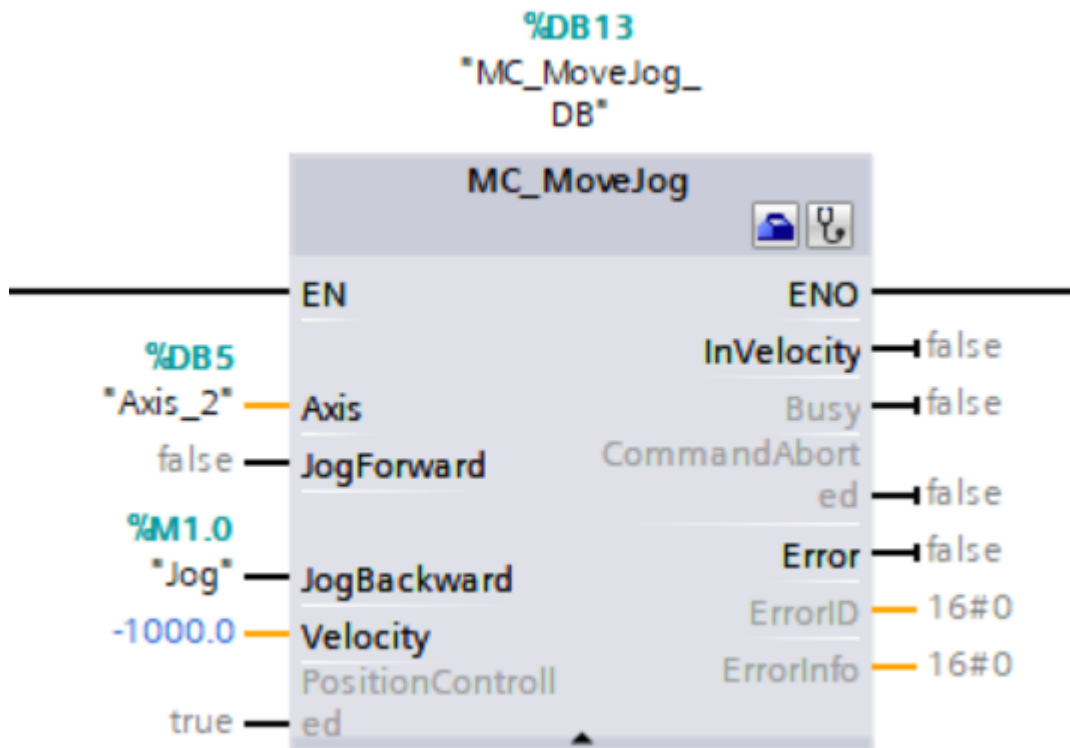
KUVIO 27. MC\_Power moottorinohjauslohko

Relative -lohkolla moottorin akselia pystytään ajamaan suhteellinen matka. Lohko tarvitsee toimiakseen nousevan positiivisen reunan. Siksi "Execute", kohtaan on laitettu osoite, josta voidaan lähettää pulssi lohkolle. Distance-kohtaan asetetaan arvo, kuinka paljon halutaan akselin pyörähtävän. Esimerkin tapauksessa akseli pyörähtää kokonaisen kierroksen, koska matka on 1000 pulssia. Velocity-kohdassa asetetaan haluttu nopeus.



KUVIO 28. Relative-lohko

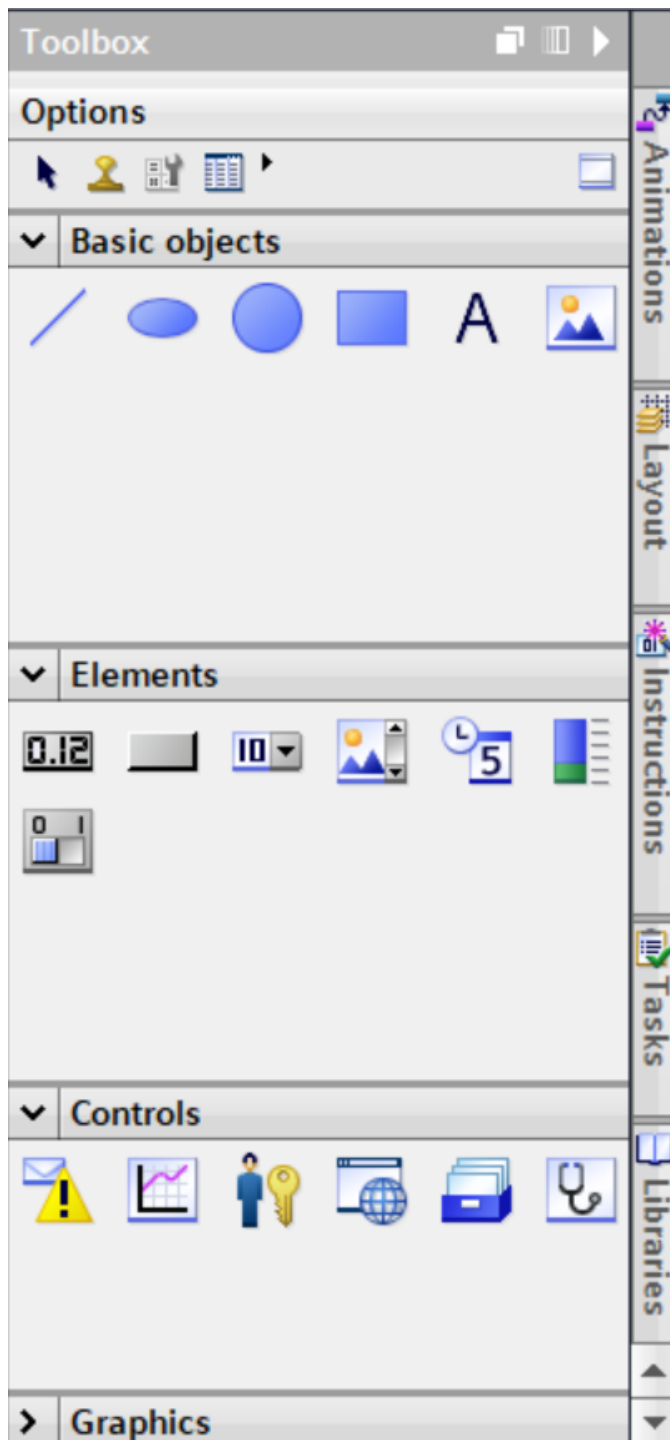
MC\_MoveJogilla akselia voidaan ajaa nappia painamalla. Eli kun nappia pidetään pohjassa, akseli pyörii. Napin vapauttaminen lopettaa liikkeen. JogForward- ja Backward-kohtiin pitää antaa nousevan reunan pulssi, jotta lohko toimi. Tässä tapauksessa akselia haluttiin liikuttaa takaisinpäin, joten backward-kohtaan annettiin pulssi. Nopeus -1000, kertoo myös akselin suunnasta. Negatiivinen luku liikuttaa sitä vastapäivään ja positiivinen myötäpäivään.



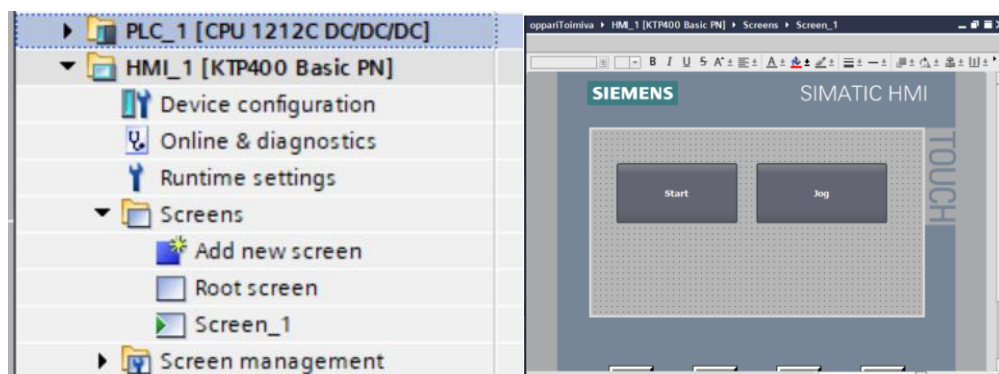
KUVIO 29. Jog-lohko

Seuraavaksi käsitellään HMI-paneelin eli näytön käyttöönottoa. HMI-paneelin etsiminen ja yhdistäminen tietokoneeseen tapahtuu samalla tavalla kuten logiikallakin. Erona on, että näytölle ei ole toimilohkoja (block), mitä tehdään vaan valvomokuvia (screen). Näille kuville voidaan luoda erilaisia painikkeita, säiliöitä tai jopa tehtaan moottorikaavioita (kuvio 30). Lisääminen tapahtuu raahaamalla objektit näytölle.

Näytöstä pystytään ohjaamaan logiikkaa ja näkemään prosessin arvoja. Sitä siis käytetään operointipaneelina. Objekteille annetaan "eventteja", joiden kautta voidaan antaa niille erilaisia toimintoja. Tästä lisää "Työn suoritus- kohdassa."

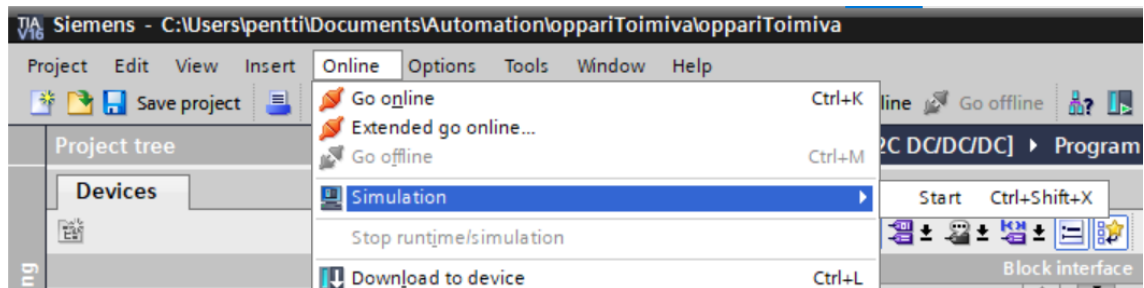


KUVIO 30. HMI-objekteja

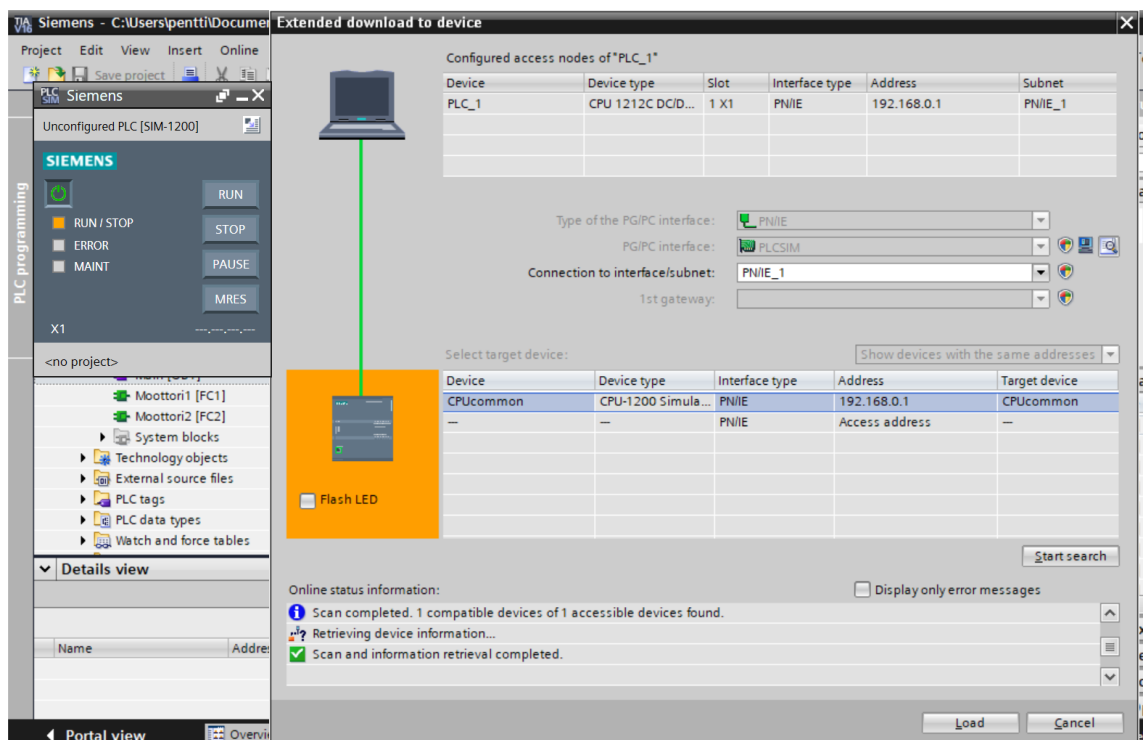


KUVIO 31. Yleisnäkymä

Tehtyjä ohjelmia voi testata Siemensin PLCSIM-ohjelmalla. Se pitää asentaa erikseen kuin TIA Portal. Simuloinnin voi käynnistää Online-valikosta (Kuvio 32), joko Simulation tai Download to device -painikkeilla. Kumpikin vaihtoehto vie käyttäjän Kuvion 32 esittämään tilanteeseen. Tässä muodostetaan yhteys kuviteltuun logiikkaan. Näkymä ja käytäntö olisi sama, jos käytössä olisi oikea logiikka.

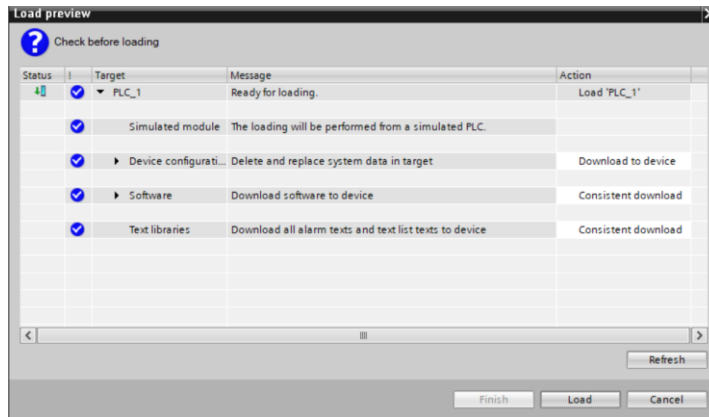


KUVIO 32. Simuloinnin käynnistys



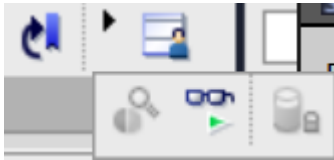
KUVIO 33. Logiikan etsintä ja yhdistäminen

Logiikan löytämisen jälkeen ladataan tiedot sille (Kuvio 34).



KUVIO 34. Ohjelman lataus logiikalle

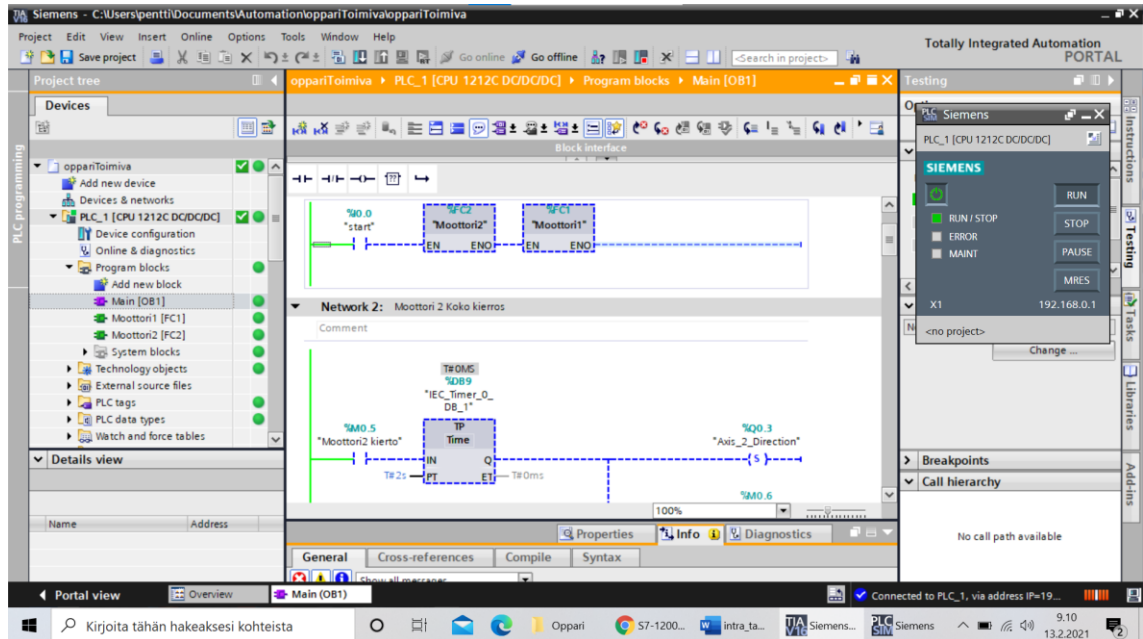
Tietojen latauksen jälkeen näyttöön ilmestyy CPU, josta voi käynnistää logiikan. Jos tietokone olisi kytketty oikeaan logiikkaan, kuvitteellista CPU:ta ei ilmesty näytölle. Ohjelmaa päästään testaamaan monitor-painikkeella (kuvio 35), joka myös vie ohjelman automaattisesti online-tilaan.



KUVIO 35. Monitor-painike

Online-tilassa ohjelman tuloja voidaan laittaa päälle ja pois. Huomioitavaa on, että tulo- ja lähtöportteja ei voi pakottaa päälle. Pakotus onnistuu ainoastaan muistipaikoille, joissa voidaan bittiä muuttamalla laittaa se päälle tai pois.

Monitor-tilassa ohjelman vasempaan laitaan ilmestyy vihreitä palloja (kuvio 36), merkinä siitä, että ohjelmassa ja logiikassa kaikki täsmäävät toisiaan. Mikäli jokin olisi pielessä, pallot olisivat sinioransseja.



KUVIO 36. Yleisnäkymä. Oikeassa laidassa näkyy kuvitteellinen CPU.

### 2.1.6 Työn toimintaperiaate ja kytkentäkuva

Tavoitteena oli siis rakentaa ohjaus automaattiselle työstökoneelle. Työstökoneessa toinen askelmoottori pyörittää rumpua ja toinen liikuttaa poraa. Rumpu lukitaan paikoilleen paineilmasylinterillä.

Toinen askelmoottoreista pyörittää kuusi asemaista työkappalerumpua ja toinen moottori siirtää porakoneen pinolia, joka tekee porausliikkeen.

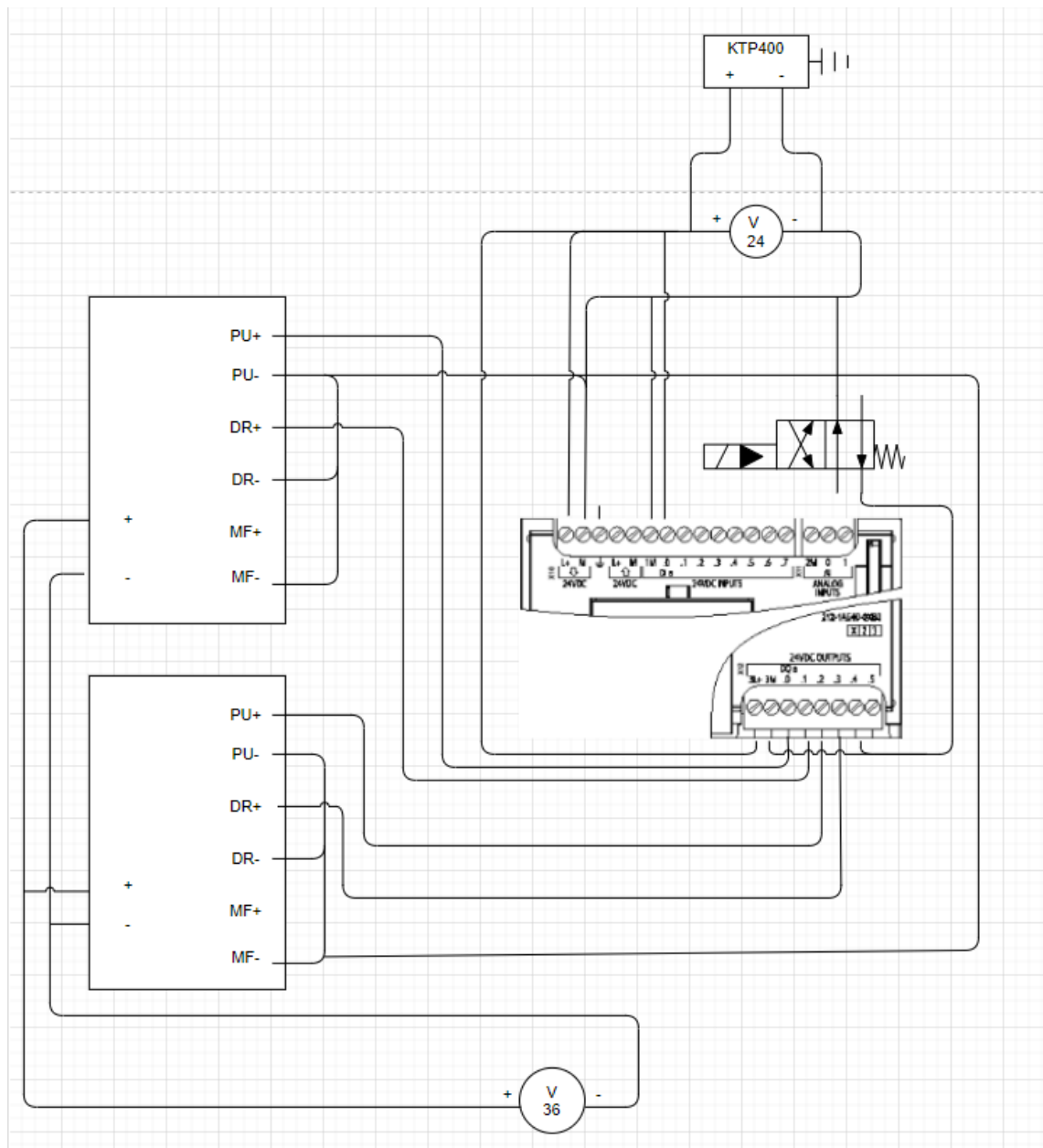
Koneen käyttäjä kiinnittää työstettävän kappaleen työkappalerumpuun, painaa käynnistysnappia, jolloin rumpu kääntyy yhden aseman eli 60 astetta. Rummun kääntyttyä paineilmasylinteri lukitsee sen paikoilleen ja porakoneen pinoli tekee porausliikkeen ja palaa takaisin. Porausliikkeen aikana käyttäjä vaihtaa uuden työkappaleen rumpuun tai odottaa kierron päättymistä. Tämän jälkeen hän käynnistää työkierron uudestaan napin painalluksella.

Koneeseen on helppo lisätä uusia porausyksiköitä jokaisen rummun vapaaseen vaiheeseen. Valitulla logiikalla voidaan ohjata yhteensä neljää askelmoottoria.



Työssä 24V-virtalähde kytkettiin antamaan virta logiikalle sekä näytölle ja 36V-virtalähde antoi virran askelmoottoreille. Logiikka ohjasi askelmoottorin-ajureita. Virtalähteet kytketään porakoneen moottorin ohjausreleeseen, mistä ne saavat jännitteen. Niiden kytkennän suoritti asennusoikeudet omaava sähköasentaja.

Koneeseen rakennettiin myös hätäseis-toiminto, mutta se kytkettiin porakoneessa valmiina olevaan piiriin, joka ohjaa moottorin käynnistysrelettä. Eli hätäseispainiketta painettaessa se katkaisee verkkovirran porakoneelta ja sitten sähkökaapilta. Kytkennän suoritti myös sähköasentaja.



KUVIO 37. KytKentäkuva

### **3 TYÖN SUORITUS**

Tämän luvun tarkoitus on käydä läpi, miten projekti toteutettiin. Käydään läpi PLC-ohjelman teko sekä HMI-näytön ohjelmointi. Lopuksi suoritetaan sähkökaapin kasaus.

#### **3.1 Alustus**

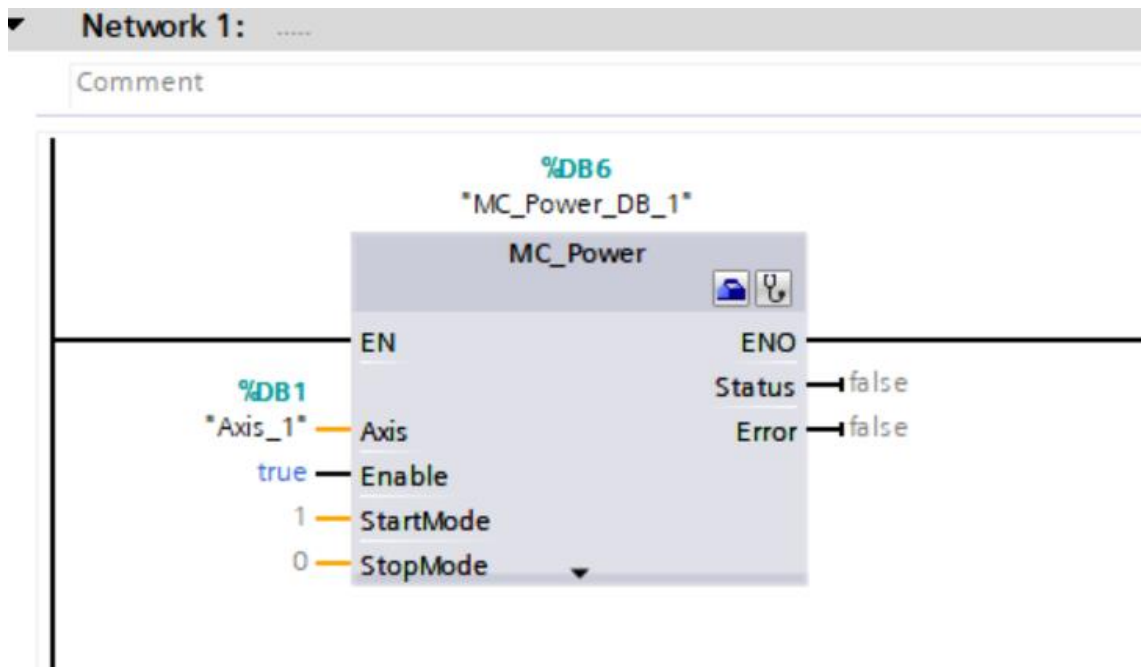
Työ aloitettiin suunnittelemalla asiakkaan kanssa, millainen ohjauksesta tehtäisiin. Tämän jälkeen etsittiin tietoa netistä erilaisista askelmootoreista ja logikoista. Saatujen tietojen pohjalta päädyttiin edellä esiteltyihin vaihtoehtoihin.

Osin saavuttua suoritettiin ohjelman teko ja sähkökaapin kasaus. Lopuksi testattiin kaiken toiminta ennen asiakkaalle lähetystä.

##### **3.1.1 Ohjauksen toteutus**

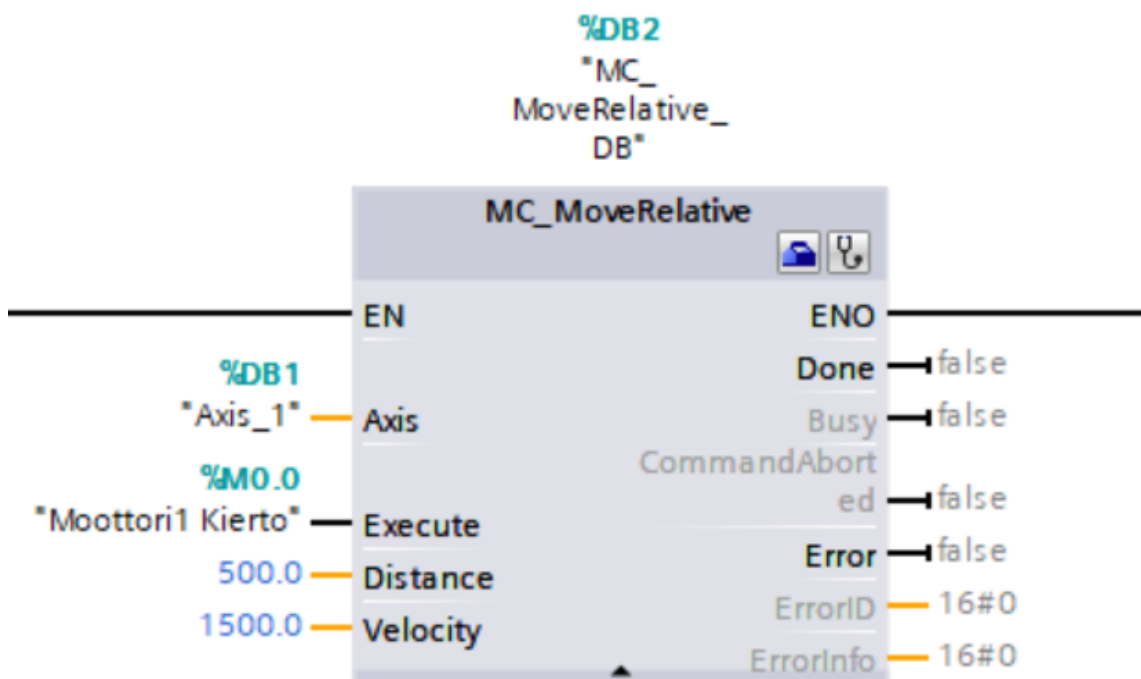
Ohjaus toteutettiin ladderilla. Ensimmäiseksi luotiin tarvittavat lohkot. Moottori 1 ohjaisi poran vipuvartta ja moottori 2 pyörittäisi rumpua. Luotiin ensimmäiseksi funktiot moottori 1 ja 2, sillä niiden jälkeen oli helppo lähteä rakentamaan pääohjelmaa.

Teoria osuudessa on selitetty lohkojen käyttö, joten sitä ei enää tässä käydä läpi. Power-lohkolla otettiin moottori käyttöön:



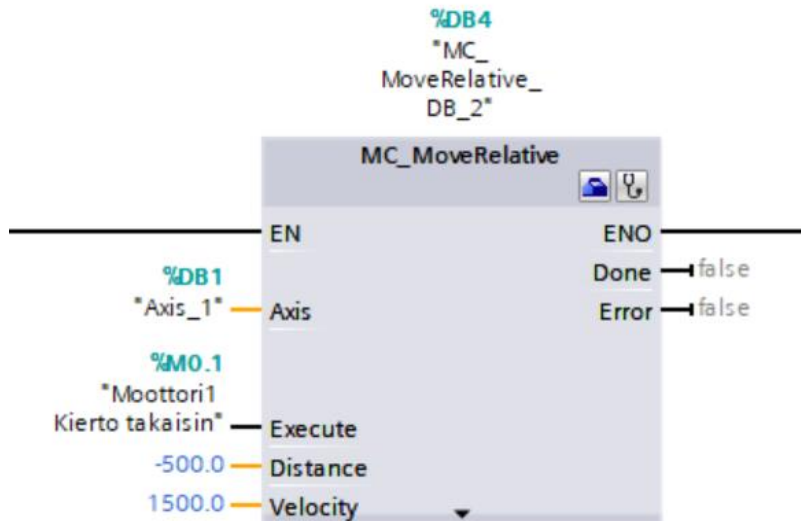
KUVIO 38. Power-lohko

Ensimmäisellä Relative-lohkokalla ohjataan moottorin akseli kääntymään puoli kierrosta eteenpäin. Tämä toteutetaan muuttamalla bitti M0.0 (Moottori1 Kierto) arvosta 0 arvoon 1. Distance asetetaan 500, koska kokonainen kierros on 1000 pulssia ja akselin halutaan vain laskevan poran alas.



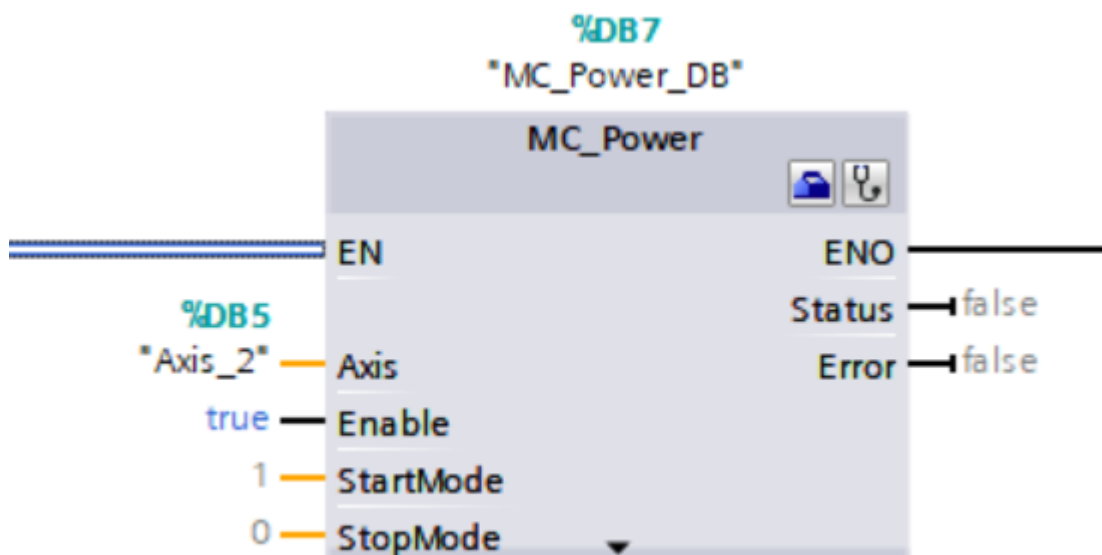
KUVIO 39. Relative-lohko poran laskemiseen

Toinen Relative-lohko suoritti moottorin akselin palautuksen takaisin alkuperäiseen arvoonsa. Tässä tapauksessa se toteutettiin antamalla positiivinen nousureuna Execute-kohtaan muuttamalla bitti M0.1 arvosta 0 arvoon 1. Kuljettu matka eli Distance asetettiin arvoon -500. Negatiivinen merkki kääntää suunnan.



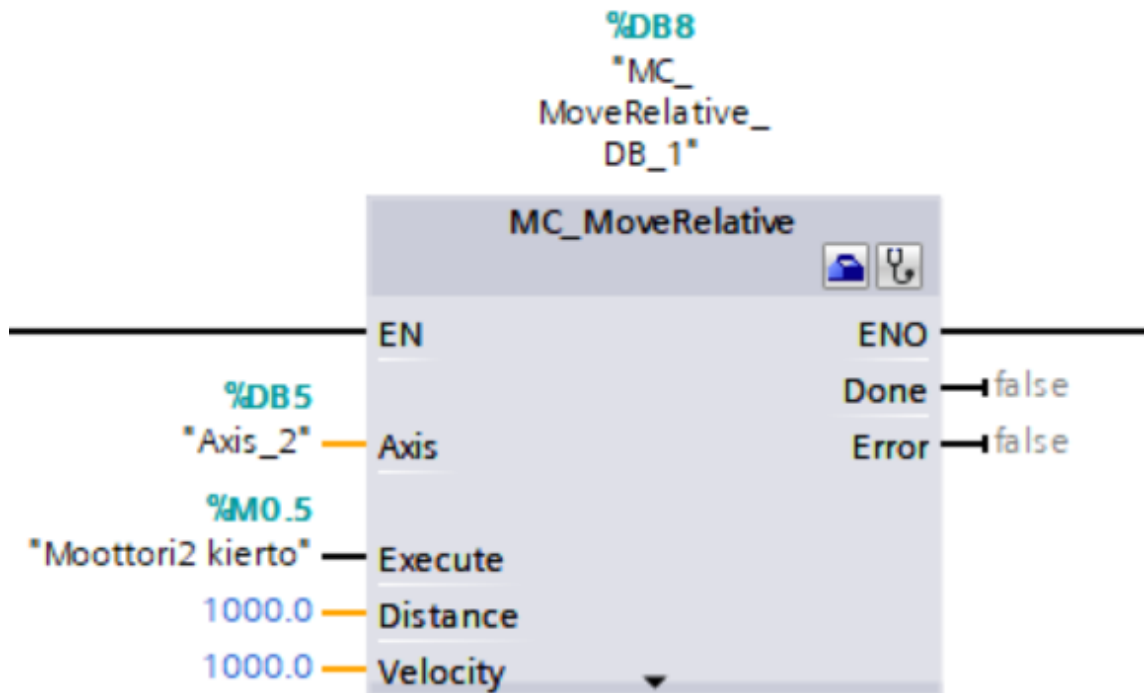
KUVIO 40. Relative-lohko poran palauttamiseen

Moottori 2 pyöritti rumpua ympäri. Se toteutettiin samalla periaatteella:



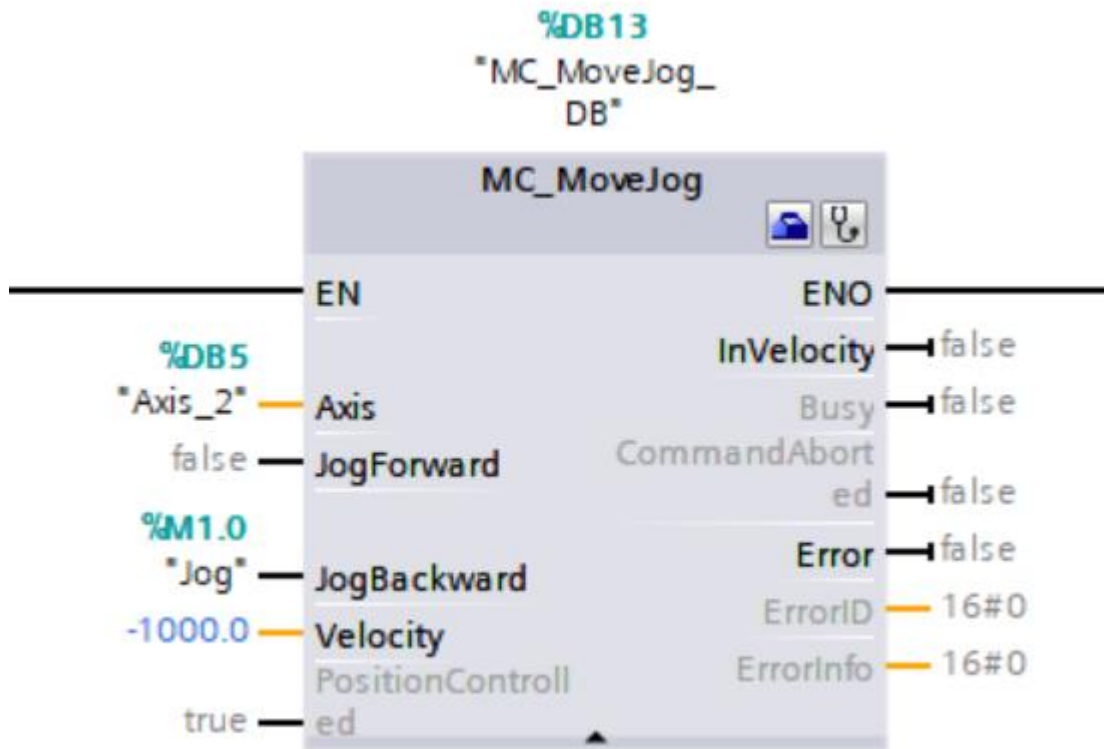
KUVIO 41. Toisen moottorin Power-lohko

Bitistä M0.2 annettiin nousureuna Relative-lohkolle. Tässä tapauksessa akselin haluttiin pyörähtävän kokonaisen kierroksen, joten etäisyydeksi asetettiin 1000.



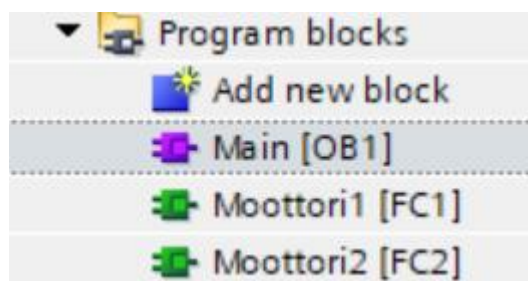
KUVIO 42. Relative-lohko moottorille 2

Moottorille luotiin myös jog-lohko. Tämän avulla akselia pystyttiin ajamaan haluttuun asemaan nappia painamalla. Jälleen lohko tarvitsi toimiakseen nousureunan, joka saatiin osoitteesta M1.0.



KUVIO 43. Jog-lohko

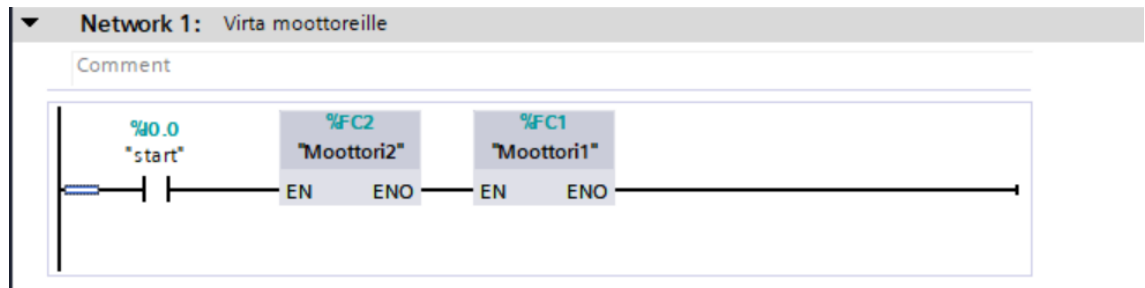
Moottorilohkojen luonnin jälkeen Program blocks -osio näytti kuvion 44 mukaiselta. Tämän jälkeen voitiin alkaa luomaan ohjelmaa OB1-lohkoon.



KUVIO 44. Ohjelman lohkot

Prosessi aloitetaan "Start", painikkeella. Painike on kytketty logiikan tuloon 0 ja sitä painettaessa se kytkee virran päälle. Startin ollessa päällä se päästää virran myös moottoreille. Moottorilohkot voi yksinkertaisesti raahata Program blocks-valikosta ohjelmaan. Huomioitavaa on, että oikeassa elämässä kyseinen

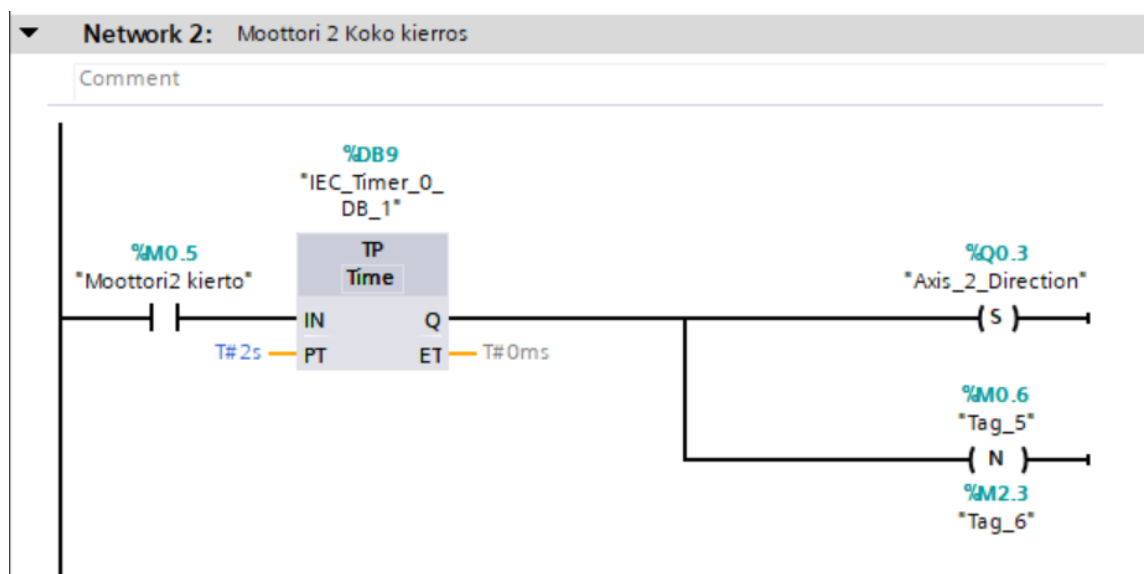
”Start”, on vain nappi, jolla koko prosessi saa virran. Sillä ei siis käynnistetä itse ohjelmaa.



KUVIO 45. Virran kytkeä päälle

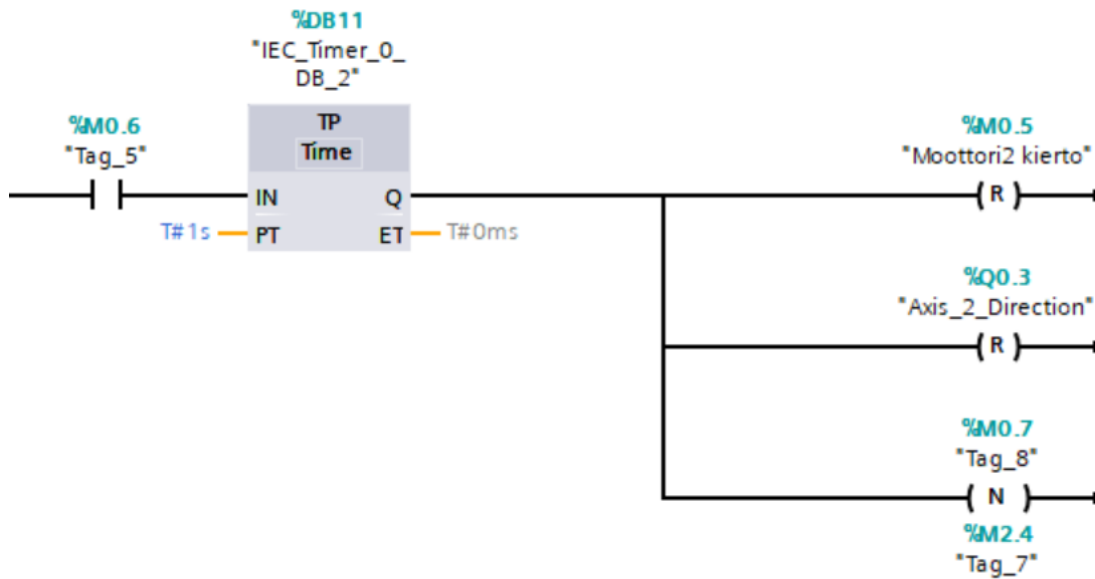
HMI-näyttöön tullaan myöhemmin luomaan Start-painike, joka käynnistää po-rausprosessin. HMI-paneelin Start-painike aktivoi bitin M0.5 (Moottori2 kierto), joka päästää virran pulssigeneraattorille. Generaattori lähettää pulssia kaksi sekuntia. Tämä pulssi asettaa lähdön Q0.3 päälle, joka ohjaa moottorin akselin pyörähtämään.

Sen alapuolella on osoitepaikassa M0.6 sijaitseva Tag\_5, joka kytkeytyy päälle sen havaitessa laskureunan eli pulssin loppuvan.



KUVIO 46. Moottori2 osaohjelma

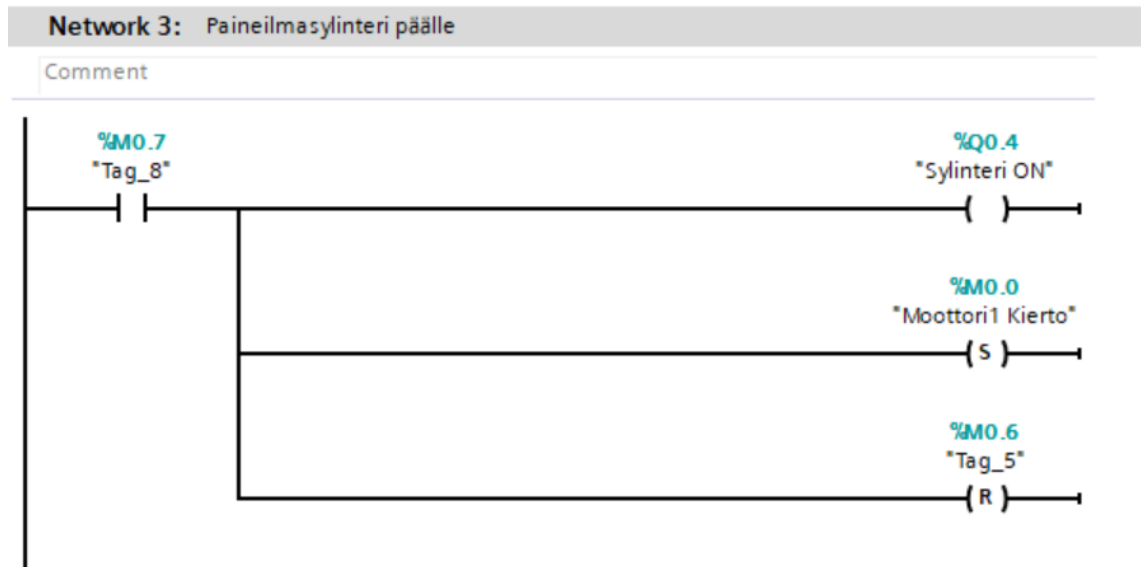
Tag\_5 mennessä päälle se asettaa seuraavassa vaiheessa pulssigeneraattorin päälle. Pulssi on päällä sekunnin, jonka aikana se resetoit aikaisemmassa kierrossa käytetyt osoitteet M0.5 ja Q0.3. Näin näitä voidaan käyttää uudessa kierrossa taas.



KUVIO 47. Loput Moottori2 ohjelmasta

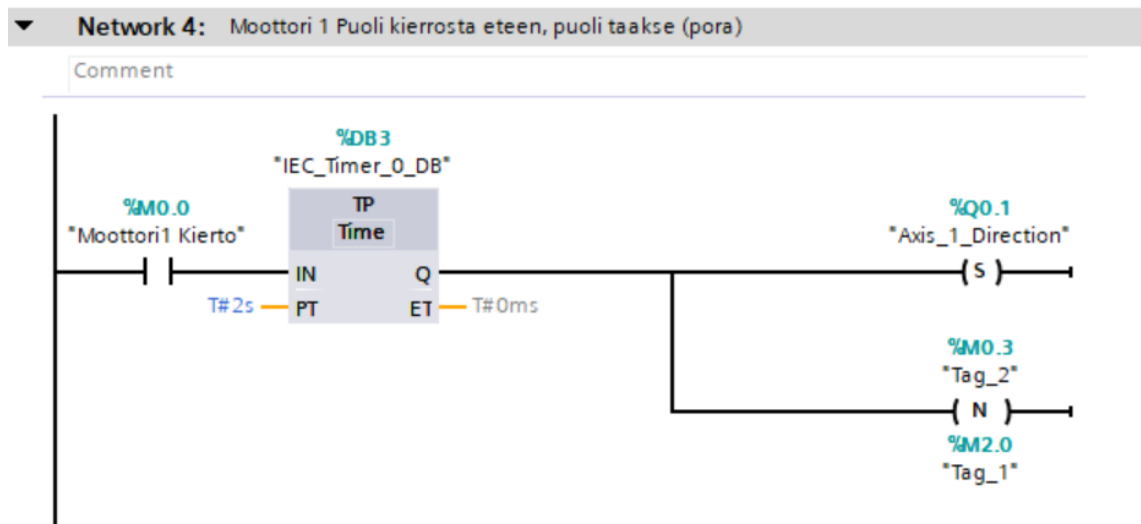
Tag\_7 havaitsee pulssin loppumisen ja asettaa Tag\_8 päälle (Kuvio 48). Tämä aloittaa paineilmasylinterin toiminnan, joka lukitsee rummun kiinni porauksen ajaksi. Tag\_8 asettaa lähdön Q0.4 päälle, joka aktivoi sylinterin toiminnan. Se myös asettaa muistipaikan M0.0 päälle, joka aloittaa porausyksikön moottorin toiminnan. Se myös resetoit edellisen kohdan Tag\_5.





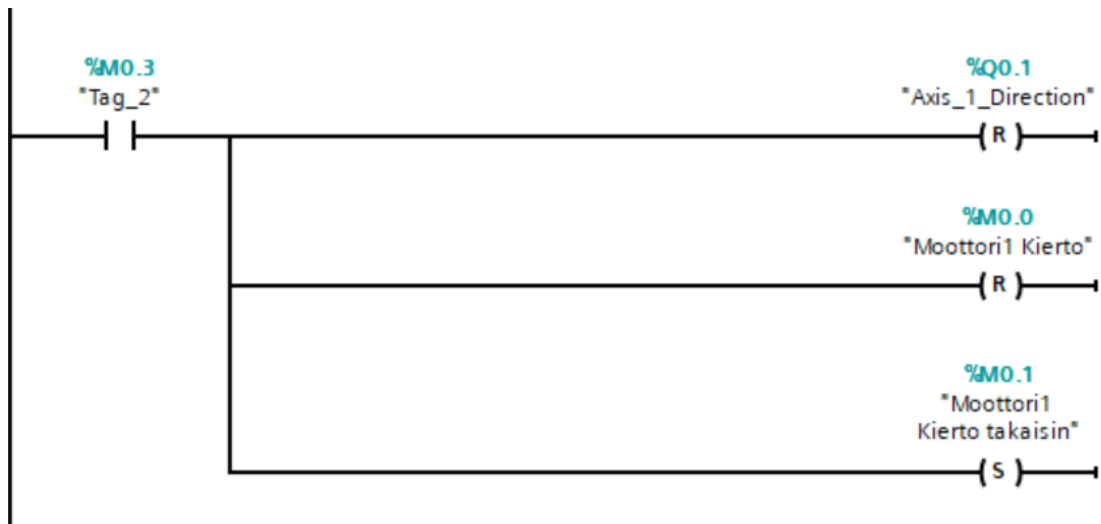
KUVIO 48. Paineilmasylinterin ohjaus

Moottori 1 kierro (M0.0) bitti käynnistää akselin, joka laskee poran alas. Tämä tapahtuu syöttämällä positiivinen nousureuna lähdölle Q0.1, kuten edelläkin. Tag\_1 havaitsee jälleen pulssin sammumista ja sen tapahtuessa asettaa muistipaikkaan M0.3 (Tag\_2) arvon 1.



KUVIO 49. Poran ohjaus

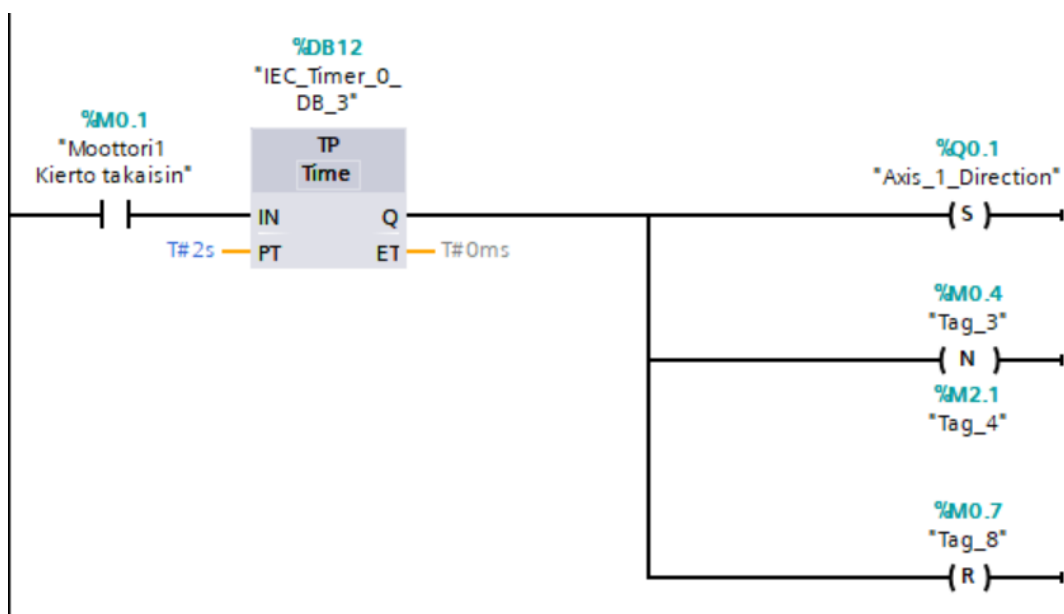
Tag\_2 mennessä päälle se resatoi akselin suuntapulssin eli lähdön Q0.1 sekä muistipaikan M0.0, jota käytettiin suorittamaan akselin ensimmäinen liike. Tag\_2 asettaa myös päälle muistipaikan M0.1, joka käynnistää akselin takaisin paluun.



KUVIO 50. Poran ohjaus jatkuu

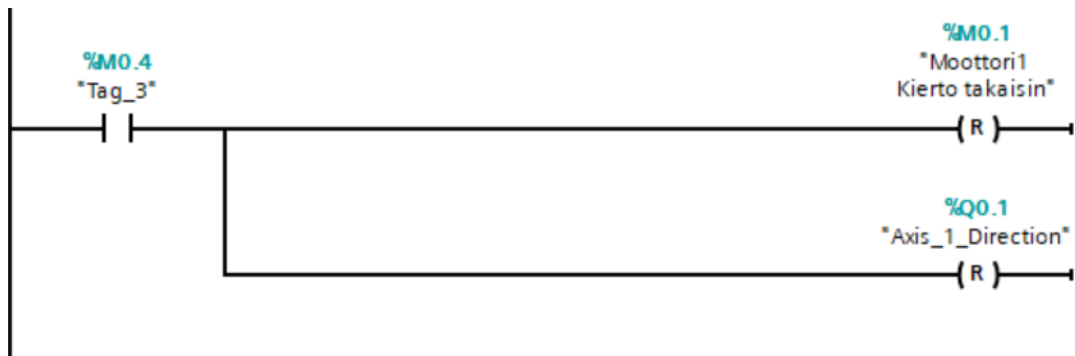
Moottori1 kierto takaisin eli muistipaikka M0.1 käynnistää akselin takaisinpaaluun. Tämä osa ohjelmaa siis nostaa poran takaisin ylös.

Kuten edellä pulssiajastin antaa positiivisen nousureunan lähdölle Q0.1 ja resetoit Tag\_8. Tag\_4 odottaa jälleen pulssin negatiivista reunaa, jolloin se asettaa Tag\_3 päälle.



KUVIO 51. Poraohjaus jatkuu

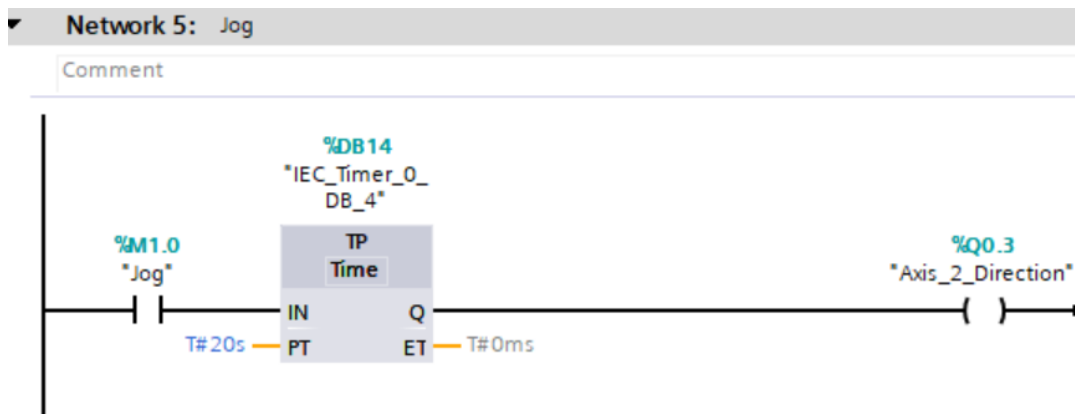
Tag\_3 resetoit bitin, joka aloittaa moottori1 kierron eli bitin M0.1 sekä akselin suuntapulssin Q0.1.



KUVIO 52. Ohjelmakierron loppu

Viimeisenä ohjelmaan lisättiin Jog-lohkon käyttö. Jogille luotiin oma käynnistysensä sekä erillinen painike HMI-paneeliin. Tämä siksi, että sitä voitaisiin ohjata muusta ohjelmasta riippumatta ja akseli saataisiin asetettua haluttuun paikkaan.

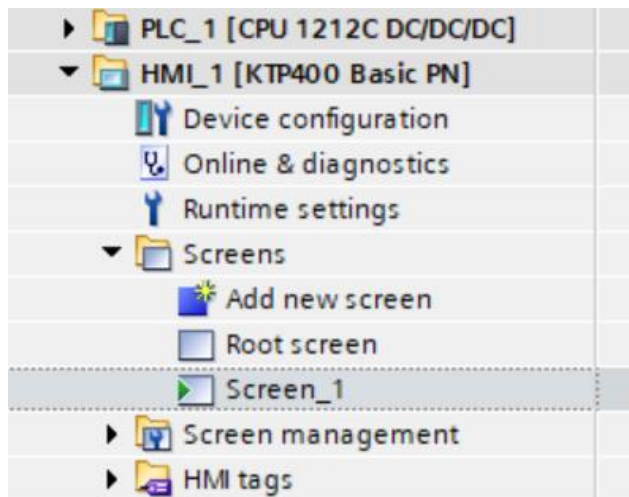
Muistipaikka M1.0 arvon vaihtuessa arvosta 0 arvoon 1 pulssiajastin lähettää 20 sekunnin ajan (tai vähemmän aikaa, mikäli napin vapauttaa) pulssia rummun moottorin akselille.



KUVIO 53. Jogin hallinta

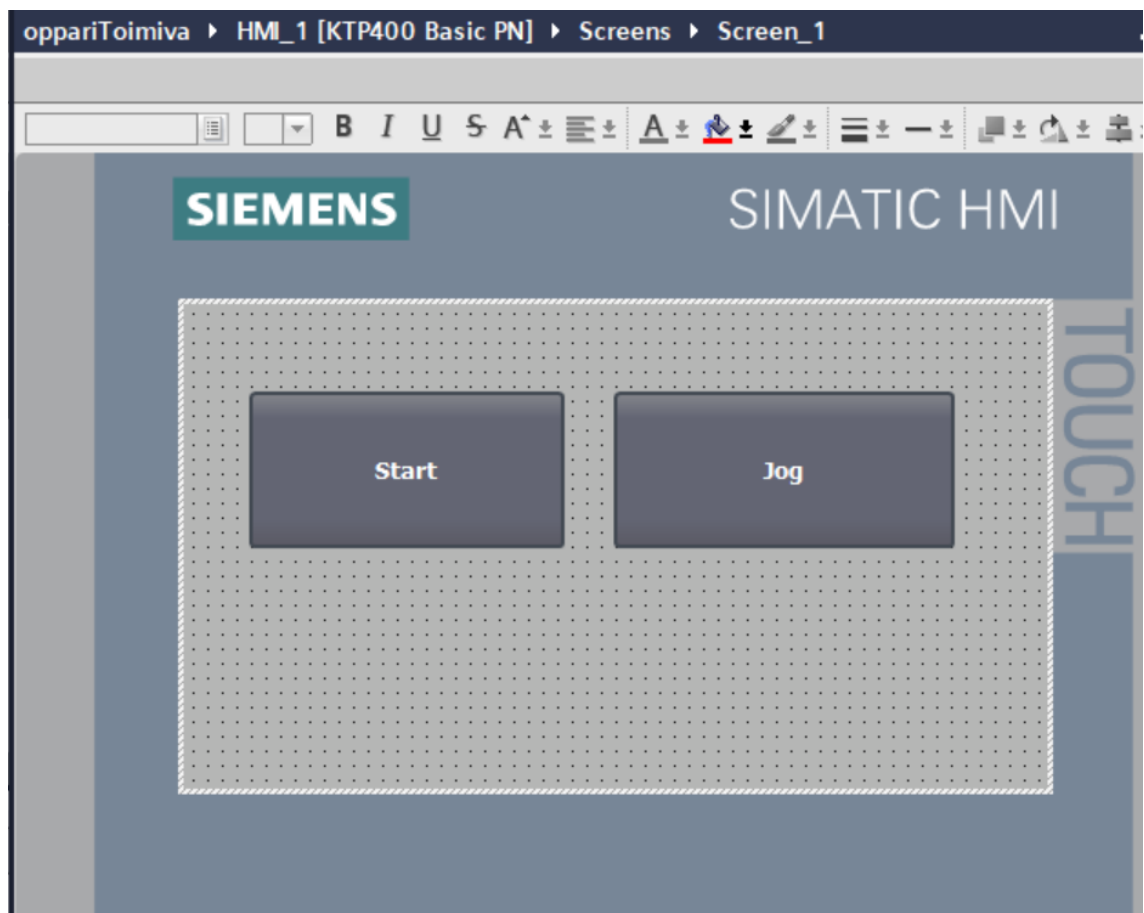
### 3.1.2 HMI-näyttöjen luonti

HMI-näyttöjä tarvitsee luoda vain yksi ja se tapahtuu Add new screen-painikkeella (kuvio 54). HMI-näytölle luodaan kaksi painiketta: Start ja Jog.



KUVIO 54. HMI-näyttö valikko

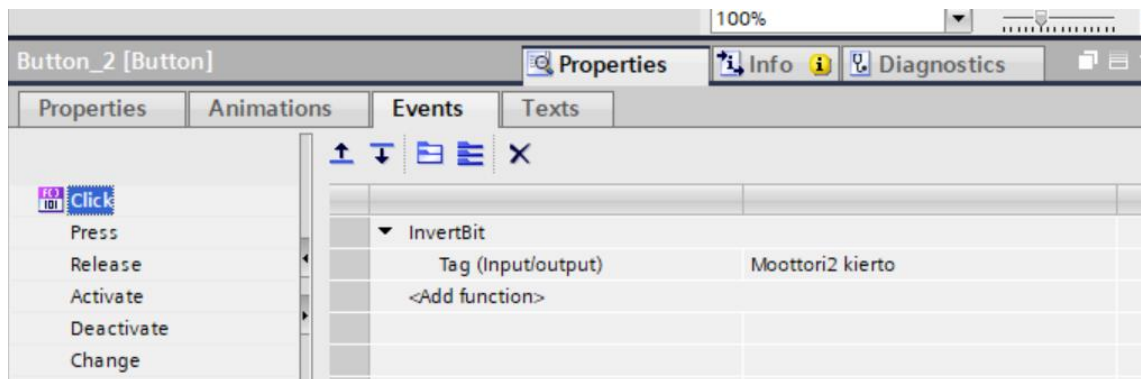
Näppäimet raahataan kuvion 30 valikosta ja ne voi nimetä tuplaklikkaamalla niitä.



KUVIO 55. Valmis näyttö

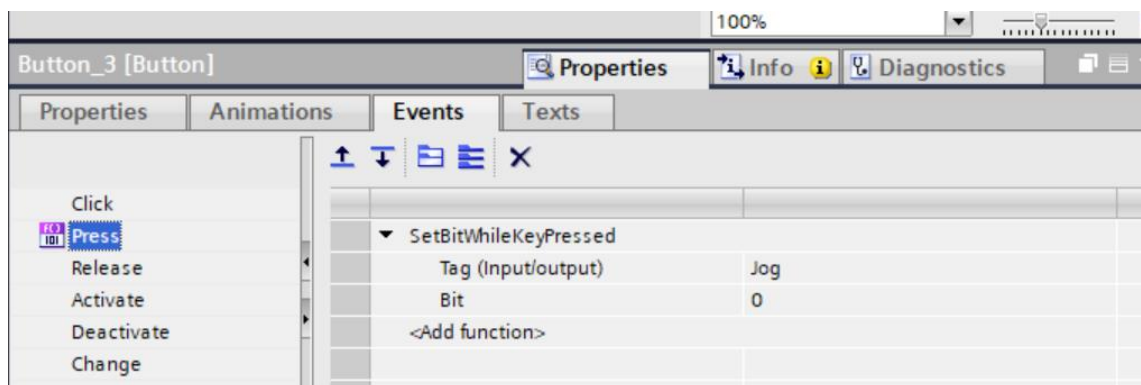
Valittaessa näppäin alas aukeaa sen asetukset valikko. Events-kohdasta päästään määrittelemään miten painike liittyy ohjelmaan. Start-painikkeella valitaan click eli kerran painettaessa sitä, se aktivoi halutun asian.

Tämän jälkeen napille valitaan toiminto. Tässä tapauksessa ”InvertBit”, joka kääntää tietyn bitin arvon. Toiminnolle pitää valita tagi, joka tässä tapauksessa oli bitti, joka käynnistää rumpua pyörittävän moottorin toiminnan. Näin napilla voitiin käynnistää koko prosessi.



KUVIO 56. Napin toiminnan määrittäminen

Jog-toiminnolle valitaan ”Press” käsky sekä ”SetBitWhileKeyPressed” komento. Näillä kun nappia painaa pohjassa, se pitää bitin päällä niin kauan, kunnes se vapautetaan.



KUVIO 57. Jog-napin määrittäminen

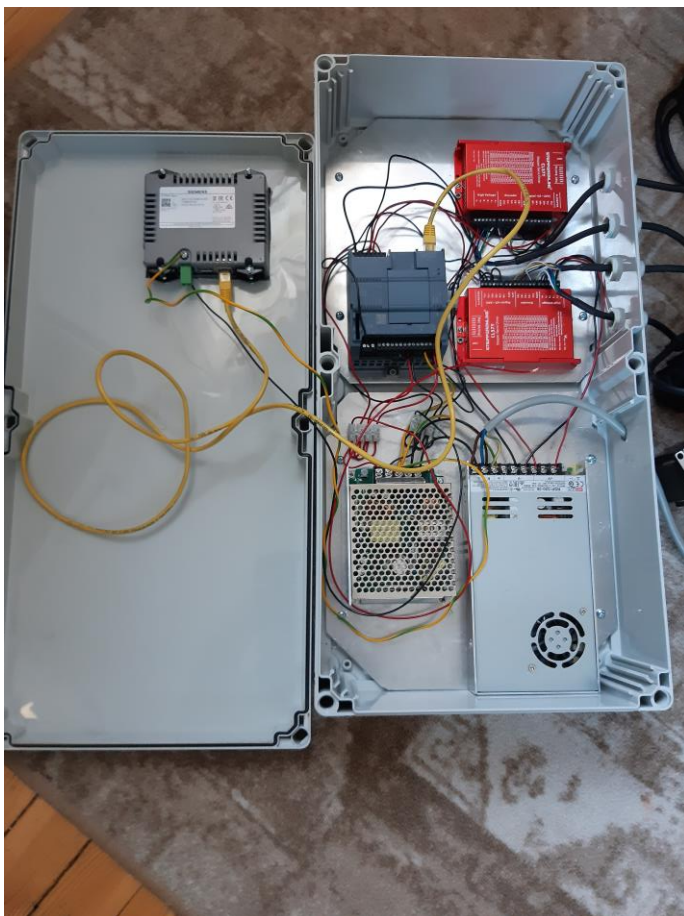
### 3.1.3 Kasa ja sähkökaappi

Kasa suoritettiin sähkökaappiin. Kaapille porattiin reiät johtoja varten sekä leikattiin näytölle aukko. Läpivienneillä estettiin johtojen liikkuminen. Virtalähteet, logiikka ja pulssiohjaimet kiinnitettiin kaappiin asennettuun peltiin ruuveilla.

Verkkovirran kytkennän suoritti sähköasentaja, jolla oli luvat kunnossa. Alla kuvat valmiista työstä.



KUVIO 58. Kaappi valmiina



KUVIO 59. Kaappi sisältä

## 4 POHDINTA

Työn tuloksena saatiin siis logiikkaohjaus ja valmis kaappi työstökoneelle. Projekti onnistui hyvin. Työn tavoitteena oleva muokattavuus toteutui sangen hyvin. Käyttäjän on helppo muokata ohjausta soveltumaan muuhunkin kuin vain porausyksikköön. Logiikkaan on saatavilla signaalimoduuleja, joilla pystytään lisäämään tuloja ja lähtöjä. Täten esimerkiksi askelmoottoreita voitaisiin ohjata enemmän. Kasvattamalla askelmoottoreiden määrää pystyttäisiin luomaan monimutkaisempia prosesseja. Lisäksi kasvattamalla porausyksiköiden määrää saataisiin tuotantokapasiteettia nostettua, sillä se nopeuttaisi prosessia.

Poraukseen käytettävän yksikön lisäksi laitteistoa voisi muokata tekemään kierteiden jyrshintää (esimerkiksi ruuvin kantaan ura ruuvimeisselille) tai vaikkapa kokoonpanoa. Samanlaisia yksiköitä käytetään Abloyn tehtaalla kokoonpanoroboteissa. Par-Met Team voisi tehdä samankaltaisia yksiköitä myyntiin muillekin asiakkaille, mutta tällä hetkellä se keskittyy vain tilaustöihin.

Työtä olisi voinut parantaa tekemällä selkeämmän Tag tablen, jolloin siitä näkisi suoraan mikä tagi viittaa mihinkin ohjelmassa. Lisäksi työssä on käytetty M-muistialuetta. Siemens ei suosittele tätä aluetta käytettäväksi projekteissa, sillä ne eivät ole niin tehokkaita kuin käytettäessä globaaleja datalohkoja. Globaaleja datalohkoja, jonne varastoidaan tieto, suositellaan erityisesti käytettäväksi funktiolohkoissa. Lisäksi jos ohjelma haluttaisiin siirtää toiselle logiikalle voisi M-muistia käytettäessä ilmentyä ongelmia, sillä muistialueet voivat erota eri logiikkamallien kesken. Tässä työssä ei kuitenkaan ole tarkoitus vaihtaa käytettävää logiikkaa eri malliin, joten muistialueella ei ole väliä. Lisäksi tämän kokoluokan laitteistossa ei bittialueen teholla ole väliä. Mikäli laitteistoon haluttaisiin lisätä useampia askelmoottoreita, olisi funktioiden muistipaikat hyvä päivittää globaaleihin datalohkoihin.

## LÄHTEET

Fibox. Kotelointiluokitus. Luettu 23.2.2021. [https://www.fibox.fi/9/NEMA/UL%20-luokitukset\\_FIN1.html](https://www.fibox.fi/9/NEMA/UL%20-luokitukset_FIN1.html)

Gonzalez, C. 2015. Engineering Essentials: What is a programmable logic controller? Julkaistu 1.6.2015. Luettu 15.2.2021. <https://www.machinedesign.com/learning-resources/engineering-essentials/article/21834250/engineering-essentials-what-is-a-programmable-logic-controller>

Scarpino, M. 2016. Motors for Makers A Guide To Steppers, Servos And Other Electronical Machines. Julkaistu 2015. Luettu 1.4.2021. <http://ptgmedia.pearsoncmg.com/images/9780134032832/samplepages/9780134032832.pdf>

Muthukrishnan, V. 2021. Programmable Logic Controllers (PLCs): Basics, Types & Applications. Päivitetty 14.1.2021. Luettu 15.2.2021. <https://www.electrical4u.com/programmable-logic-controllers/>

Siemens. S7 1200-sarjan PLC käyttöohje. Luettu 1.1.2021. [https://cache.industry.siemens.com/dl/files/465/36932465/att\\_106119/v1/s71200\\_system\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf)

Siemens 1200-sarjan logiikat. Luettu 2.1.2021 <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html>

Stepperonline. Askelmoottori. Käyttöohje. Luettu 1.1.2021. <https://www.omc-stepperonline.com/download/Y-serie-closed-loop-drivers.pdf>

Turja, J. Askelmoottorin testaaminen ja soveltuvuus xy-nosturiin. Julkaistu 29.09.2017. Luettu 1.4.2021. [https://www.theseus.fi/bitstream/handle/10024/133283/Jori\\_Turja.pdf?sequence=1&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/133283/Jori_Turja.pdf?sequence=1&isAllowed=y)

Velling, A. What is CNC Machining? Working Princiciples, Capabilities & More. Julkaistu 18.2.2020. Luettu 3.4.2021. <https://fractory.com/what-is-cnc-machining/>

What is a Stepper Motor: Types & Its Working. Luettu 17.2.2021 <https://www.elprocus.com/stepper-motor-types-advantages-applications/>