



TAMPEREEN  
AMMATTIKORKEAKOULU

# Yksityisen vesivoimalan automatisointi

Kasper Laine

Opinnäytetyö  
Joulukuu 2015  
Kone- ja tuotantotekniikka  
Kone- ja laiteautomaatio



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Kone- ja tuotantotekniikka  
Kone- ja laiteautomaatio

LAINEN, KASPERI:

Yksityisen vesivoimalan automatisointi

Opinnäytetyö 46 sivua, joista liitteitä 2 sivua  
Joulukuu 2015

---

Tämän opinnäytetyön tarkoituksena oli esittää automaatoratkaisu Mikkelissä sijaitsevaan pienvesivoimalaan demolaitetta hyväksi käyttäen. Työn tekemisen hetkellä vesivoimalan käynnistys ja sammutus tapahtui täysin manuaalisesti paikanpäällä. Työn tavoitteena oli luoda automaatiojärjestelmä, jonka avulla vesivoimalan käyttö tapahtuisi pelkästään automaattisesti. Toiminnallisia vaatimuksia olivat muun muassa ylemmän vesitason pinnankorkeuden tarkastelu, voimalan käynnistys- ja sammutusprosessin automatisointi, käynnistysten ja sammuttamisen suorittaminen haluttuihin kellonaikoihin, datan kerääminen vedenpinnan korkeudesta, mahdollinen etäkäytettävyys, syötettävän sähköverkon tilan tarkastelu ja voimalassa sähköä tuottavan generaattorin taajuuden tarkastelu. Suurimmassa osassa tässä työssä oli kaiken automatiikan sydän, itse ohjelma, sekä sen luominen ja testaus. Työn yhtenä tarkoituksista olikin avata ohjelman tekemistä ja siinä huomioon otettavia asioita.

Vesivoimalan automaatiojärjestelmän vaatimuksiin onnistuttiin vastaamaan hyvin, eikä mitään kriittisiä toimintoja edellä mainituista jäänyt huomiotta. Tosin etäkäytettävyttä työhön ei ehditty sisällyttää, mutta vaihtoehtoja tutkittiin hieman. Työn alussa etäkäyttöön mietittiin GSM-ratkaisua, joka toimisi siten, että voimalaa voitaisiin ohjata tekstiviesteillä. Tämä osoittautui asiaksi, joka olisi vaatinut hieman laajempaa paneutumista, ja se päätettiin viime hetkillä jättää työn ulkopuolelle. Tehdyistä järjestelmää ohjaavasta ohjelmasta onnistuttiin saamaan aikaan selkeä, viiteen eri osaan jaettu kokonaisuus. Voimalaa simuloivalla demolaitteella saatiin testattua niitä asioita, joita oli tarkoituskin.

Työssä käytetty LOGO! 8 ja sen ohjelmoimiseen käytetty LOGO!Soft Comfort V8.0 olivat käytettävyydeltään selkeitä ja helppoja. Toiminnot olivat yksinkertaisia, mutta niillä pystyi toteuttamaan myös monimutkaisempia kokonaisuuksia. Ohjelman simulointi LOGO!Soft Comfortilla toimi erinomaisesti, ja sen avulla pystyttiin jo aikaisessa vaiheessa karsimaan pienempiä virheitä pois. Kyseinen logiikka oli myös juuri oikea valinta vesivoimalan automaatiojärjestelmän toteutuksessa. Vaikka etäkäytettävyttä ei saatukaan tähän työhön sisällytettyä, nousi esiin toinen vaihtoehto sen toteuttamiseksi. GSM-yhteyksien sijaan tässä vaihtoehdossa käytettäisiin WLAN-yhteyttä. Tosin siinä missä ohjauksien käyttöliittymä olisi WLAN-ratkaisussa helpompi toteuttaa, jouduttaisiin siinä kiinnittämään paljon enemmän huomiota tietoturvallisuuteen vaikuttaviin seikkoihin. Kaiken kaikkiaan lopputulokseksi saatiin aikaan toimiva järjestelmä, jonka voisi asentaa työhön liittyvään vesivoimalaan.

---

Asiasanat: automaatiojärjestelmä, vesivoimala, ohjelmointi, LOGO!

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Mechanical Engineering  
Machine Automation

LAINEN, KASPERI:  
Automation of Small Hydroelectric Power Station

Bachelor's thesis 46 pages, appendices 2 pages  
December 2015

---

Purpose of this thesis was to present an automation solution to a small hydroelectric power plant located in Mikkeli. At the time of this thesis, the stations startup and shutdown processes were controlled manually. The aim was to create an automation system that allows the use of the hydroelectric power station to be almost entirely automatic. Functional requirements included the examination of the upper level water height, automation of the plant startup and shutdown processes, the execution of startup and shutdown processes daily at the desired times, data collecting from the water levels, the potential remote access, the power grids status checking and examination of the power plant generator frequency. The majority of this work was concentrated on the creation and testing of the automation processes. One of the targets on this thesis was to show how the program is done and the things to consider in the process.

Requirements of the plant automation system were successfully met and no critical functions were left out. There was not enough time to include remote access to this thesis, but the alternatives were studied slightly. At first the plan was to use GSM-module, so that the power plant could be controlled by SMS messages. This turned out to be a matter that would have required a slightly wider dedication, and it was decided to leave it out from this thesis. The program was made out of five different sections and it came out good. Simulations of the automation system was done with a demo device, and the results were as expected.

LOGO! 8 logic and LOGO!Soft Comfort V8.0 software were simple and easy to use. The functions were simple, but they could also carry out more complex entities. Simulation of the program in LOGO! Soft Comfort was excellent, and it was a good tool to cut down some early stage errors. LOGO! 8 logic was also the right choice for this automation system. Although remote access was not included in the automation system, it turned out that there is also another solution for it alongside the GSM-module. It could also be done with WLAN. WLAN would be easier to use and provides better solution for the monitoring of the plant, but there would be a lot of information security issues to consider.

---

Key words: automation, hydroelectric, programming, LOGO!

## SISÄLLYS

1	JOHDANTO.....	7
2	VESIVOIMA.....	8
2.1	Vesivoiman toimintaperiaate .....	8
2.2	Vesivoima Suomessa .....	9
3	Automatiikan lähtötiedot ja vaatimukset.....	10
3.1	Laitoksen tekniset tiedot .....	10
3.2	Vaaditut toiminnot .....	11
3.2.1	Turvallisuuteen liittyvät vaatimukset.....	11
3.2.2	Asiakkaan vaatimukset.....	12
3.2.3	Halutut lisätoiminnot.....	13
4	Komponenttien valinta .....	15
4.1	Logiikka .....	15
4.2	Anturit.....	16
4.3	Muut komponentit.....	17
4.4	I/O-lista .....	17
5	Ohjelman teko .....	18
5.1	Perus toiminnot ja blokit.....	18
5.1.1	Input .....	19
5.1.2	Analoginen input.....	19
5.1.3	Output.....	19
5.1.4	AND .....	20
5.1.5	OR .....	21
5.1.6	AND nousevan reunan tarkastelu .....	21
5.1.7	NAND laskevan reunan tarkastelu.....	22
5.1.8	RS-kiikku .....	23
5.1.9	On-delay.....	23
5.1.10	Off-delay .....	24
5.1.11	Viikoittainen ajastin .....	24
5.1.12	Analogiavahvistin .....	25
5.1.13	Analogiavertailija.....	25
5.1.14	Rajakytkin .....	26
5.1.15	Data log .....	27
5.1.16	Cut/Join .....	27
5.2	Toteutus .....	27
5.2.1	Aloittaminen.....	28
5.2.2	Ohjelman rakennus.....	28

5.2.3	Komentointi .....	34
5.3	Ohjelman simulointi .....	35
5.4	Riskitilanteet .....	36
6	DEMOLAITE.....	38
6.1	Rakennus.....	38
6.2	Automaatiikan testaus demolaitteella.....	39
7	POHDINTA.....	42
	LÄHTEET.....	44
	LIITTEET .....	45
	Liite 1. Ohjelma kokonaisuudessaan.....	45
	Liite 2. I/O-lista .....	46

**LYHENTEET JA TERMIT**

MW	megawatti
kW	kilowatti
m <sup>3</sup> /s	kuutiometriä sekunnissa
V	voltti, jännitteen yksikkö
PNP-anturi	anturi kytkee kuorman virtalähteen positiiviseen napaan

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on esittää pienen vesivoimalan automatisointiratkaisu, sekä mahdolliset jatko- ja kehitystoimenpiteet ja riskit. Työssä käydään lyhyesti läpi vesivoimassa käytettävää tekniikkaa, jotta lukija ymmärtää miten vesivoimala yleisellä tasolla toimii. Sitä kautta saadaan selvyyttä siihen, mitä uusia ulottuvuuksia pienen vesivoimalan automatisointi voi tuoda sen käytölle. Työ tehtiin voimalan omistavalle Jaakko Mattilalle.

Tarkemmin työssä käydään läpi kuinka automatisointi toteutettiin juuri kyseiselle vesivoimalalle, ja mitkä olivat asioita, jotka piti ottaa huomioon automatiikkaa tehdessä. Mihin logiikkaan tai ohjelmoitavaan releeseen päädyttiin? Mitkä olivat ominaisuudet joita järjestelmältä haluttiin? Näiden kysymysten pohjalta perustellaan ja esitellään työssä käytetyt komponentit, sekä koko automatiikan sydän, itse voimalaa ohjaava ohjelma.

Ohjelman rakenne, toiminnot, toteutus ja testaus esitellään niin, että lukijalle jää hyvä peruskäsitys pientä vesivoimalaa ohjaavasta kokonaisuudesta. Työssä käydään läpi myös sen tekemisen aikana esiintyneet ongelmat ja niiden ratkaisut. Mitä käy, jos tulee sähkökatkos? Mitä tapahtuu, jos jokin johto katkeaa? Mitkä ovat ohjelmaa tehdessä tärkeitä asioita, jotka pitää ottaa huomioon? Muun muassa näihin kysymyksiin vastataan.

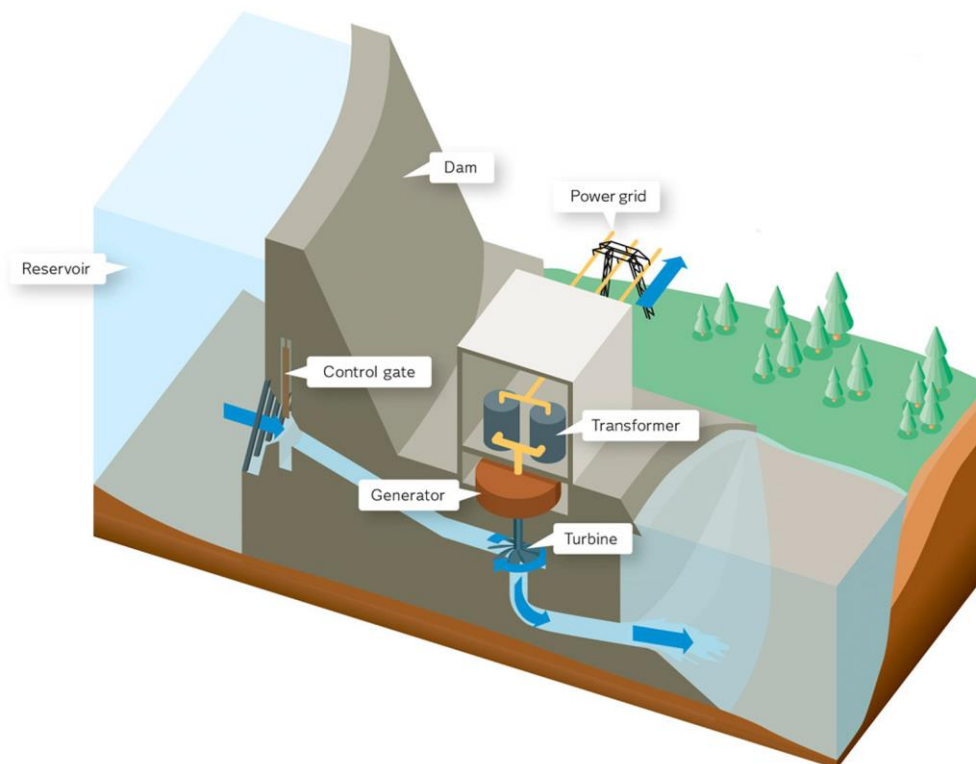
Ohjelman rakennetta avataan ja perustellaan. Myös ohjelman testaamista, niin tietokoneella kuin konkreettisella demolaitteella, käydään läpi. Eli kuinka simulointi toimii, mitkä ovat simulointia tehdessä huomioon otettavia asioita, sekä mitä saavutetaan ohjelman testaamiseen tarkoitettulla demolaitteella.

## 2 VESIVOIMA

### 2.1 Vesivoiman toimintaperiaate

Vesivoimassa käytettävä tekniikka on yleisellä tasolla erittäin yksinkertaista. Tarkoituksena on muodostaa veden liike-energiasta sähköä. Vesivoimalan voi siis rakentaa vaikka puron varteen. Tietenkin mitä isompi virtaus ja korkeusero saadaan aikaiseksi, sen tehokkaampi voimala voidaan rakentaa. Monia isoja vesivoimaloita varten rakennetaan patoja, joiden avulla voidaan muodostaa iso tekojärvi, josta sitten voidaan säännöstellä sähkön tuottoa melko helposti. Isot kosket ovat myös oiva paikka vesivoimalalle, esimerkkinä vaikka Tammerkosken vesivoimala. (Vattenfall 2014)

Jokaisessa vesivoimalassa toimintaperiaate on kaikesta huolimatta sama (kuva 1). Virtaus ohjataan ylemmältä vesitasolta alemmalle turbiinin läpi, joka pyörittää turbiiniakselia. Turbiiniakseli on yhteydessä generaattoriin, jossa pyörimisliike muunnetaan sähköenergiaksi. Tämä sähköenergia ohjataan muuntajiin, jotka muuntavat jännitteen sähköverkkoon sopivaksi. (Vattenfall 2014)



KUVA 1. Vesivoimalan toimintaperiaate. (Vattenfall 2014)



## 2.2 Vesivoima Suomessa

Suomessa sähköntuotannosta noin 10–20 % saadaan vesivoimasta. Tämä lukema on ollut joskus isompikin, nimittäin 1950- ja 1960-luvuilla vesivoimasta saatu sähköntuotannon osuus oli jopa 90 %. Vesivoiman kapasiteetti nykyään on Suomessa noin 3100 MW, joka jakautuu yli 220 vesivoimalaitoksen kesken. (Energiateollisuus 2015)

Suomessa vesivoima on luonnollinen valinta uusiutuvalle energialle maan vesistöjen vuoksi. Niin kuin monilla tiedossa on, Suomessa on järviä, koskia, puroja ja vesistöjä erittäin paljon. Aivan kuten Energiateollisuuden artikkelissa kerrotaan, vielä tänäkin päivänä osa mahdollisesta potentiaalista, varsinkin pienvesivoimalle, on käyttämättä. Isoja vesivoiman rajoittavia tekijöitä ovat ympäristövaatimukset ja kiristynyt EU:n vesipolitiikka. Toisaalta taas ilmastoasioiden kiristyessä vesivoima on vahvoilla, voimaloista kun ei hiilidioksidipäästöjä tai muitakaan ilmanpäästöjä juurikaan synny. (Energiateollisuus 2015)

### 3 Automatiikan lähtötiedot ja vaatimukset

Vesivoimasta puhuttaessa yleensä puhutaan säätövoimasta. Säätövoima tarkoittaa sähköntuotantoa, jolla voidaan vastata nopeasti kysynnän muutoksiin. Tämä kyseinen asia otetaan huomioon myös voimalan automatiikkaa tehdessä. Automatiikkaa luodessa on myös otettava huomioon paljon muita asioita, kuten vaadittavat toiminnot sekä asiakkaan toiveet. Selkeiden lähtötietojen ja vaatimusten laatiminen auttaa paljon työn myöhemmissä vaiheissa.

#### 3.1 Laitoksen tekniset tiedot

Vesivoimalaitos, johon työ tehtiin, on kokoluokaltaan pieni ja sijaitsee Mikkelissä (kuva 2). Se on teholtaan 27 kW. Tehovertauksena voidaan ottaa Suomen isoista vesivoimalaitoksista Imatran laitos, jonka teho on 185 MW. Putoamiskorkeus, joka syntyy Mikkelin laitoksen ylemmän ja alemman vesitason välille, on 3 metriä. Keskivirtaus voimalan kohdalla on 1,45 m<sup>3</sup>/s.



Kuva 2. Korpikosken vesivoimalaitos Mikkelissä.

Laitoksesta saataisiin enemmänkin tehoa irti, mutta se on aikanaan rakennettu keskivirtauksen mukaan. Yleisimmät turbiinityypit, joita käytetään vesivoimaloissa, ovat Francis

ja Kaplan (Vattenfall 2014). Näistä kahdesta Mikkelin laitos käyttää Francis –turbiinia. Generaattorina voimalaitoksessa toimii oikosulkumoottori.

Tällä hetkellä voimalaitos toimii täysin käsinohjauksella. Paikanpäällä on painikkeet, joilla pystytään avaamaan ja sulkemaan hydraulikalla toimivaa porttia, joka päästää veden virtaamaan turbiinin lävitse. Voimalassa on myös taajuusmittari, jota joudutaan seuraamaan, kun voimala kytketään sähköverkkoon. Voimalan sammutus ja käynnistys tapahtuu tällä hetkellä siis aina paikanpäällä.

Voimalaitoksen tekniset tiedot eivät vaikuta yhtä paljon automaatiojärjestelmän suunnitteluun kuin voimalaitokselta haluttavat toiminnot ja ominaisuudet. Tosin on hyvä tietää, minkä kokoisesta laitoksesta on kyse työtä tehdessä. Suurimmat muutokset pienen ja ison voimalaitoksen automaatiojärjestelmien välillä ovat enemmän komponenteissa kuin ohjauksessa. Eroja on muun muassa ohjattavien releiden koossa, käytettävässä logiikassa, käyttöjännitteissä jne. Itse laitosta ohjaava ohjelma saattaa kummassakin tapauksessa olla hyvin samankaltainen.

## **3.2 Vaaditut toiminnot**

On olemassa vaadittuja toimintoja, jotka jokaisen automaatiojärjestelmän tulisi täyttää. Näistä toiminnoista voidaan eritellä asiakkaan puolelta tulleet vaatimukset, jotka voivat vaihdella paljonkin tapauskohtaisesti. Myös hyvälle ohjelmalle on olemassa erikseen määrittelyitä, ja niitä käydään läpi ohjelman teko – luvussa.

### **3.2.1 Turvallisuuden liittyvät vaatimukset**

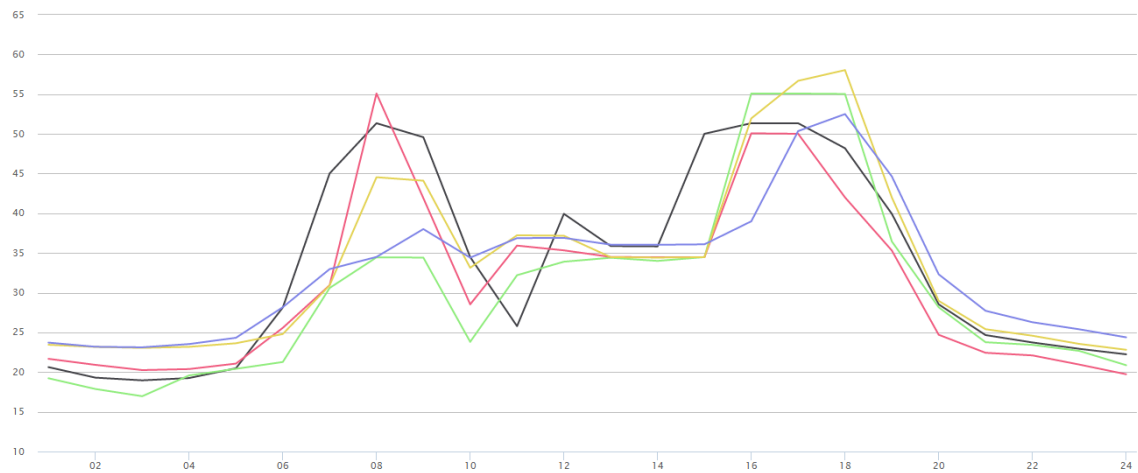
Jokaisen automaatiojärjestelmän vaadittuihin toimintoihin kuuluvat ehdottomasti turvallisuuden liittyvät toiminnot (Sundquist. Työturvallisuus automaation varassa.). Näitä toimintoja toteutetaan niin järjestelmässä kokonaisuudessaan, sekä sitä ohjaavassa ohjelmassa. Mikkelin pienvesivoimalaan liittyen huomioon otettavia turvallisuusasioita ovat sähkökatkokset, sähköverkon tila sekä järjestelmän poiskytkentä. Sähkökatkoksen sattuessa järjestelmän tulee vähintään ajaa itsensä alas hallitusti. Tämä tarkoittaa voimalassa sitä, että automaatiikka suorittaa katkoksen sattuessa sammutustoiminnon. Tämän jälkeen voimalaa ei pidä saada käyntiin, ennen kuin vika on korjattu.

Sähköverkon tilalla viitataan sen verkon tarkasteluun, johon voimalalla tuotetaan sähköä. Tämä sähköverkko voi olla pois toiminnasta, vaikka voimala tuottaa vielä sähköä. Tilanteen ollessa näin, automaatiojärjestelmän pitää toimia samalla tavalla kuin sähkökatkoksen sattuessa. Tämä siksi, että jos sähköverkkoa tullaan vaikka huoltamaan, on hengen- vaarallista tuottaa samanaikaisesti verkkoon sähköä. Lisäksi vaatimuksena oli, että kun verkko palautuu normaalitilaan, saisi voimalan päälle vasta jonkin ajan kuluttua. Automaatiojärjestelmä on myös saatava pois päältä, esimerkiksi voimalan eri osien huolto- töitä, puhdistamista tai muuta vastaavaa varten.

Turvallisuustoimintoihin on hyvä kiinnittää huomiota työtä tehdessä. Mahdolliset vaara- tilanteet voivat vaihdella työstä riippuen, joka tarkoittaa että aina samanlainen toteutus jossakin työssä ei välttämättä päde enää toisessa. Turvallisuutta kannattaa siis ajatella aina tapauskohtaisesti, jotta kaikki vaaratilanteet tulisi huomioitua.

### **3.2.2 Asiakkaan vaatimukset**

Kun turvallisuusvaatimukset on otettu huomioon, voidaan siirtyä työn tilaajan vaatimuk- siin. Mikkeliin toteutettavassa automatiikassa yhtenä vaatimuksista oli voimalan käyn- nistäminen ja sammuttaminen haluttuina aikoina päivässä. Tämä perustuu sähkön hinnan vaihteluun tunneittain päivän sisällä. Vesivoimalaa halutaan pitää päällä sähkön kalliina tunteina. Kuvaajassa 1. esitetään marraskuun loppupuolelta viiden eri päivän sähkön hin- nanvaihtelu tunneittain. Y-akselilla on hinta ja x-akselilla kellonaika. Yksittäinen erivä- rinen käyrä edustaa yhtä päivää. Niin kuin kuvaajasta nähdään, sähkön hinta on korkealla noin kello 6:00–10:00 sekä n. klo 15:00–20:00. Nämä ovat ne kellonajat, jolloin vesivoi- malaa halutaan pitää päällä. Aiemmin mainittu säätövoima siis tarkoittaa juurikin tätä toimintoa, eli sähköntuotannolla vastataan kuluttajien tarpeisiin nopeasti.



KUVAAJA 1. Sähkön hinnanvaihtelu tunneittain. (Nordpoolspot 2015)

Vaatimukseen sisältyi myös vedenpinnan tarkastelu. Mikkelin laitoksen ylemmälle vesitasolle on asetettu pinnan ylä- ja alarajat, joissa sen pitää pysyä. Automaatiikan tulee huolehtia siitä, että voimala ei käy, jos veden pinta on asetetun rajan alapuolella, sekä vastavasti voimalan käynnistyksestä ylärajan ylittyessä. Vedenpinnan rajojen maksimi vaihtelu on 60 senttimetriä.

Automaatiikalta vaadittiin myös kykyä tarkastaa, että generaattorin tuottama jännitteen taajuus on oikea, ennen kuin voimala kytketään sähköverkkoon. Tämä tarkoittaa sitä, että generaattorin tulisi pyöriä kytkettäessä samalla taajuudella, kuin mikä sähköverkon taajuus on. Näin sähköverkkoon ei synny mitään isompia häiriöitä, jotka voisivat aiheuttaa vahinkoa esimerkiksi muihin verkkoon liitettyihin laitteisiin. Vaatimuksena oli, että sama tarkastelu suoritetaan myös vesivoimalan sammutustoiminnossa.

### 3.2.3 Halutut lisätoiminnot

Automaatiikalle saattaa olla vähimmäisvaatimusten lisäksi lisätoimintopyyntöjä. Lisätoimintojen ja vaadittujen toimintojen rajapinta voi olla häilyvä. Erottelua voikin miettiä siten, että vaatimukset ovat sellaisia toimintoja, joita ei voi jättää ohjelmasta pois missään nimessä, ja lisätoimintoja voidaan toteuttaa/lisätä mahdollisuuksien mukaan.

Lisätoimintoja voidaankin tehdä, hieman automaatiojärjestelmästä riippuen, hyvinkin helposti. Tietynlaiset lisätoiminnot voidaan toteuttaa ohjelman sisällä, kun taas jotkut

vaativat ylimääräisiä komponentteja tai osia, kuten antureita tai lisämoduuleita logiikkaan. On kuitenkin hyvä muistaa, että lisätoiminnoista tulisi olla järjestelmän kannalta jotakin hyötyä, eikä työskentelyaikaa käytetä tarpeettomien lisätoimintojen luomiseen.

Yhtenä toiveena vesivoimalan ohjaukseen liittyen oli etäkäyttö. Etäkäytöllä toivottiin, että voimala voitaisiin laittaa päälle ja sammuttaa esimerkiksi puhelimen avulla. Myös vedenpinnan arvojen lukeminen etäältä oli ominaisuus josta oltiin kiinnostuneita.

Lisäksi toivomuksena oli vedenpinnan korkeuden tiedonkeräämiseen liittyvä toiminto. Käytännössä tarkoittaen sitä, että kerran päivässä tieto vedenpinnan korkeudesta tallentuisi johonkin, josta sitten päivittäiset lukemat saataisiin haluttaessa käyttöön.

## 4 Komponenttien valinta

Nyt kun tiedetään laitoksen koko ja tekniset tiedot, sekä mitä automatiikalta vaaditaan ja toivotaan, voidaan alkaa valitsemaan tarvittavia komponentteja. Tässä vaiheessa pyritään kiinnittämään huomiota eri vaihtoehtojen ominaisuuksiin suhteessa hintaan. Joskus budjetti saattaa olla vähäinen, ja joistakin toivotuista lisätoiminnoista joudutaan luopumaan. Jos taas budjettia on laittaa reilusti hankkeeseen, voi komponenttivalinnoilla saavuttaa ylimääräisiä toimintoja ja ominaisuuksia. Mielestäni kultainen sääntö on pyrkiä toteuttamaan kaikki vaaditut ja halutut ominaisuudet mahdollisimman halvalla, mutta laadukkaasti. Komponenttien valinnassa on myös hyvä ottaa huomioon niiden elinkaarikustannukset. Laadukkaat tuotteet maksavat itsensä monesti takaisin käyttöiässä, kun taas halvemmalla hankittu komponentti joudutaan ehkä korjaamaan tai vaihtamaan useammin.

### 4.1 Logiikka

Lähes heti, kun järjestelmän koko ja halutut ominaisuudet alkavat muodostumaan, voidaan alkaa miettimään millaisella logiikalla ohjaus olisi hyvä toteuttaa. Isot järjestelmät toteutetaan yleensä ohjelmoitavalla logiikalla, joita esimerkiksi suuressa tehtaassa saattaa olla useampiakin. Pienimuotoisten automaatiojärjestelmien toteuttamiseen tarkoitettua logiikkaa nimitetään pienlogiikaksi, tai ohjelmoitavaksi releeksi. Pienlogiikan ja ohjelmoitavan releen välinen ero on tosin hyvin tulkinnanvaraista.

Logiikka on automaatiossa kaiken toiminnan sydän. Yksinkertaisuudessaan siihen syötetään haluttuja tietoja ja ulos saadaan toimintakäskyjä. Tällä välillä logiikkaa laskee, vertailee, käsittelee, suodattaa ja tekee ratkaisuja sille annettujen ohjeiden mukaisesti.

Ohjelmoitavan logiikan ja pienlogiikan välinen hintaero voi olla erittäin merkittävä. On siis hyvä miettiä, voidaanko automaatiojärjestelmä toteuttaa pienlogiikalla. Tämän työn tapauksessa vastaus on kyllä. Työssä päädyttiin käyttämään Siemensin valmistamaa Logo! 8 –pienlogiikkaa (kuva 3). Logo oli ominaisuuksiltaan juuri sopiva Mikkelin vesivoimalan automatiikan toteutukseen.



KUVA 3. LOGO! 8 (Siemens 2014)

Asiat, joita listattiin vaatimuksiin, vaikuttivat suuresti Logon valitsemiseen. Logolla onnistuu helposti päivittäiset käynnistykset haluttuina kellonaikoina, vedenpinnan tarkastelua varten vaadittavan analogiaviestin lukeminen, datan kerääminen ja yleiset automaattien toiminnot. Näin ollen siirtymistä järeämpään logiikkaan ei jouduttu oikeastaan edes harkitsemaan.

LOGO! 8:ssa on kahdeksan sisääntuloa (input) ja neljä lähtöä (output). Tarvittaessa tuloja ja lähtöjä voidaan lisätä Logoon kiinnitettävillä lisämoduuleilla. Logoon on myös olemassa GSM-lisämoduuli, jota voitaisiin hyödyntää etäohjaukseen. Työhön tarvitsimme myös Power-moduulin, sillä Logon syöttöjännite on 24 V. Power-moduuli muuntaa verkkojännitteen Logolle ja muille 24 V syöttöjännitettä käyttäville laitteille sopivaksi. Logossa itsessään on neljä analogista inputtia, mutta nämä toimivat vain jänniteviestiä antavilla antureilla. Koska ultraäänianturi tulisi antamaan virtaviestiä, lisättiin Logoon yksi lisämoduuli, jolla tällaista tietoa pystyttäisiin käsittelemään.

## 4.2 Anturit

Automaatiojärjestelmät sisältävät lähes aina jonkinlaisia antureita riippuen järjestelmän koosta ja halutuista ominaisuuksista. Työn järjestelmään antureita tarvittiin kaksi. Esimerkiksi lämpötilojen ja etäisyyksien mittaamiseen tarvitaan analogiatietoa antava anturi. Veden pinnan rajojen mittaus on juurikin etäisyyden mittausta. Tämä toteutetaan ultraäänianturilla. Ultraäänianturilla on tietty mitta-alue, eli mikä on kyseisen anturin minimi ja



maksimi etäisyydet. Tämä mitta-alue tulee ottaa huomioon anturia valittaessa. Muita huomioon otettavia asioita ovat anturin antama tieto, eli antaako anturi jännite- vai virtaviestiiä, sekä paikka mihin anturi asennetaan. Demolaitteessa käytettävä anturi ei luultavasti tule olemaan sama, joka asennetaan itse vesivoimalaan, sillä sen mittausalue on alle 60 cm ja kyseinen anturi ei itsessään kestä vettä.

Toinen anturi, jota käytetään pulssin mittaamiseen generaattorilta, on jo asennettuna paikankäällä voimalassa. Näin ollen tässä työssä tarvitsi valita vain anturi, jota pystyttäisiin käyttämään demolaitteessa. Tarkoitukseen sopiva anturi oli aivan tavallinen induktiivinen PNP-anturi.

### **4.3 Muut komponentit**

Demolaitteeseen tarvittiin lisäksi hieman eri komponentteja kuin itse vesivoimalaan. Sähkömoottorilla ja sitä ohjaavalla releellä kuvattaisiin sähköverkkoon liittymistä. Kahdella valolla ja niitä ohjaavilla releillä kuvattaisiin hydrauliiikan avaamista ja sulkemista. Käynnistykseen, sammutukseen ja häiriön kuittaukseen tarvittavat painonapit saattavat taas demolaitteessa olla hyvin samanlaiset kuin itse voimalassakin. Demolaitteeseen lisättiin vielä kaksiasentoinen kytkin, jolla pystyttäisiin kuvaamaan verkon tilaa.

### **4.4 I/O-lista**

Valituiden komponenttien perusteella tein I/O-listan. Lista on lähtökohta ohjelman tekemiselle. Siitä saadaan selville kunkin logiikkaan kytkettävän laitteen tunnus, tulo- ja lähtöosoitteet. Lista pitää sisällään myös pienen kuvauksen laitteen toiminnosta, sekä huomion itse laitteesta. Lista on tehty demolaitteen käyttämien komponenttien perusteella ja se löytyy tämän raportin liitteenä.

## 5 Ohjelman teko

Kun automatiikan vaatimukset ovat selvät ja komponentit valittu, on seuraavana tehtävänä luoda automatiikan ydin, eli ohjelma, joka ohjaa laitosta. Ohjelmakieli saattaa vaihdella riippuen logiikasta tai ohjelmoitavasta releestä, mutta perusteet ovat usein melko samanlaisia. Logossa käytettävä ohjelmointikieli on FBD, joka on lyhenne sanoista Function Block Diagram. Tämä tarkoittaa sitä, että erilaisia toiminnallisia lohkoja (blokkeja) yhdistellään isommaksi toimintakokonaisuudeksi, joka määrittää ohjelman sisällä kulkevien digitaalisten signaalien kulkua. Ohjelmassa signaalitieto on 1 = tosi tai 0 = epätosi. Yksinkertaisesti selitettynä siis signaalin ollessa ”päällä” sen arvo on 1, ja signaalin ollessa ”pois päältä” sen arvo on 0.

Logon ohjelmat toimivat sykleissä. Yhden syklin, eli ohjelmakierron aikana ohjelma suorittaa siinä olevat toiminnot sekä lukee tulot ja asettaa lähdöt (Valkeejärvi 2012). Syklin aika voi vaihdella ohjelmasta riippuen, mutta yleensä se on hyvin pieni, esimerkiksi n. 0,01 sekuntia.

Jos mieleen on hankala muodostaa kuvaa ohjelmasta, voi sitä ajatella vaikka putkistona, jolla ohjataan veden kulkua. Eri vedenohjauspisteet yhdistellään putkilla, joissa vesi pääsee kulkemaan. Esimerkiksi kahdessa putkessa virtaava vesi yhdistyy vedenohjauspisteessä (eli toimintalohkossa tai blokissa), ja jatkaa matkaansa yhdessä putkessa seuraavaa vedenohjauspistettä kohti. Signaalitieto on helppo mieltää seuraavalla esimerkillä. Jos käyttämäsi vesihana olisi toimintalohko eli blokki, avatessasi hanan, sen ulostulo olisi 1, eli tosi. Kun hana suljettaisiin, ulostulo olisi 0, eli se olisi vastaavasti epätosi.

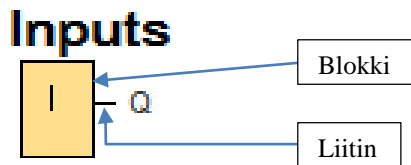
Ohjelmoinnissa saman lopputuloksen saavuttamiseen voi olla monta erilaista toteutustapaa. Yleensä ohjelmoijalle syntyykin jonkinlainen oma rutiini ja tapa ohjelman rakentamiseen. Ohjelmaa tehdessä on tarpeellista pitää mieli avoinna, sillä toimivat ratkaisut saattavat olla yllättävänkin yksinkertaisia.

### 5.1 Perus toiminnot ja blokit

Tässä kappaleessa esitellään vesivoimalan ratkaisuun käytetyt toiminnot ja blokit. Logon ohjelmointia varten on olemassa runsaasti muitakin erilaisia toimintoja ja blokkeja, mutta jos ne käytäisiin kaikki läpi, tulisi raportista turhan laaja.

### 5.1.1 Input

Inputit (kuva 4) ovat niitä blokkeja, jotka vastaavat ohjelman sisällä Logoon fyysisesti kytkettyjä tuloja. Liitin Q tarkoittaa blokin lähtöä. Eli jos Logon inputtiin tulee signaali, sitä vastaava input-blokki antaa ohjelmassa signaalitiedon 1 liittimestä Q. (Siemens 2014)

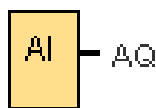


KUVA 4. Input-blokki.

### 5.1.2 Analoginen input

Analoginen input (kuva 5) vastaa ohjelmassa Logoon fyysisesti syötettävää analogia tietoa. Input blokkiin määritellään, onko kyseessä jännitetieto vai virtatieto, jonka jälkeen se skaalataan tapaukseen sopivalla tavalla. Blokin liitin AQ antaa ulos jännite- tai virtaviestistä riippuvaisen tai suoraan verrannollisen lukeman, joka vesivoimalan ohjelman tapauksessa vastaa veden pinnan korkeutta. Analoginen input tarkastelee siis aiemmin mainitun ultraäänianturin antamaa tietoa. (Siemens 2014)

### Analog inputs

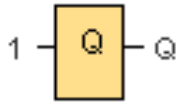


KUVA 5. Analoginen input-blokki.

### 5.1.3 Output

Outputit (kuva 6) ovat blokkeja, joilla ohjataan Logon fyysisiä ulostuloja. Eli kun output-blokkiin Q tulee signaalitieto 1, antaa sitä vastaava Logon output signaalin eteenpäin. On otettava huomioon, että vain output-blokit ohjaavat Logon fyysisiä lähtöjä, ei minkä tahansa blokin output-liittimet. Blokkien liittimet ovat vain ohjelman sisäisiä, ja sen kyseisen blokin lähtöjä, joilla signaalitietoa voidaan jakaa ohjelmassa eteenpäin. (Siemens 2014)

## Outputs



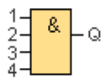
KUVA 6. Output-blokki.

### 5.1.4 AND

AND-blokki on mitä tahansa ohjelmaa luodessa tärkeä perustoiminto. Se toimii siten, että kaikkiin siihen kytkettyjen inputtien signaalitietojen tulee olla 1, jotta blokin liittimen Q signaalitieto on 1. Eli niin kuin alla olevasta kuvasta 7. nähdään, taulukossa output on 1 silloin kun kaikki inputit ovat myös 1. (Siemens 2014)

Vielä kerran vesiesimerkillä selitettynä: Kun kaikkiin input-liittämiin tulisi vettä, vasta sitten vesi pääsisi jatkamaan ulostulosta Q. Jos mihin tahansa yhteen input-liittimeen ei tulisi vettä enää, sulkisi se myös output-liittimen veden virtauksen. Tästä eteenpäin siis puhuttaessa signaalitiedoista ja blokkien toiminnoista, voi asian hahmottamisen helpottamiseksi miettiä toimintoja veden ohjaamisen avulla.

### AND



The output of an AND function is only 1 if **all** inputs are 1, that is, when they are closed.

A block input that is not used (x) is assigned: x = 1.

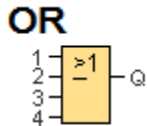
#### AND function logic table

Input 1	Input 2	Input 3	Input 4	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

KUVA 7. AND-blokki ja sen toiminta.

### 5.1.5 OR

Myös OR-blokki (kuva 8) on ohjelmoinnissa tärkeä perustoiminto. Blokin ulostulo Q on 1 silloin, kun mihin tahansa input-liittimeen tulee signaalitieto 1. (Siemens 2014)



The output of an OR function is 1 if **at least one** input is 1 (closed).

A block input that is not used (x) is assigned:  $x = 0$ .

**OR function logic table**

Input 1	Input 2	Input 3	Input 4	Output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

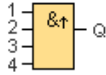
KUVA 8. OR-blokki ja sen toiminta. (Siemens 2014)

### 5.1.6 AND nousevan reunan tarkastelu

Nousevalla reunalla viitataan siihen hetkeen, kun signaalitieto vaihtuu  $0 \rightarrow 1$  ja laskevalla reunalla siihen hetkeen, kun signaalitieto vaihtuu  $1 \rightarrow 0$ . Alla oleva blokki (kuva 9) toimii siten, että output Q on 1 eli tosi, jos kaikki inputit ovat 1, ja niistä vähintään yksi oli 0 edellisessä syklissä. Output Q on 1 yhden syklin ajan, kunnes se taas resetoituu. (Siemens 2014)

Jos siis blokkiin tulee vain yksi input, Q on 1 nousevaa reunaa seuraavalla syklillä. Alla olevasta kuvaajasta voi toiminnan hahmottamiseksi seurata mitä tapahtuu sykleittäin inputtien ollessa 1 tai 0. Inputtien numero ja output Q y-akselilla ja syklit x-akselilla. (Siemens 2014)

### AND with edge evaluation

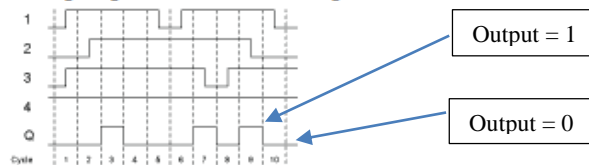


The output of an AND function with edge evaluation is only 1 if **all** inputs are 1 and **at least one** input was 0 during the last cycle.

The output is set to 1 for the duration of one cycle and must be reset to 0 for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned:  $x = 1$ .

Timing diagram of an AND with edge evaluation:

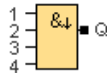


KUVA 9. AND nousevan reunan tarkastelulla ja sen toiminta. (Siemens 2014)

### 5.1.7 NAND laskevan reunan tarkastelu

Tämä kyseinen NAND-blokki (kuva 10) toimii päinvastoin kun edellinen AND-blokki, eli seurataankin laskevaa reunaa. Ainoana erona on, että kaikkien NAND-blokkiin tulevien inputtien ei pidä olla 0 tai 1, vaan riittää, että edellisessä syklissä on jossakin inputissa tapahtunut muutos  $1 \rightarrow 0$ . Alla olevasta kaaviosta näkee myös tämän blokin toiminnan. (Siemens 2014)

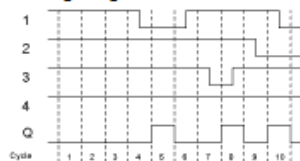
### NAND with edge evaluation



The output is set to 1 for the duration of one cycle and must be reset to 0 at least for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned:  $x = 1$ .

Timing diagram of a NAND with edge evaluation



KUVA 10. NAND laskevan reunan tarkastelulla ja sen toiminta. (Siemens 2014)

### 5.1.8 RS-kiikku

RS-kiikku (kuva 11) voidaan luokitella ohjelmoinnissa käytettäviin perustoimintoihin. Kun liittimeen S (set) tulee signaali, asettaa se liittimen Q arvoksi 1.  $Q = 1$  niin kauan kunnes liittimeen R (reset) tulee signaali, jolloin  $Q = 0$ . Jos on tilanne, jossa kumpaankin input liittimeen tulee signaali,  $Q = 0$ . Resetillä on siis etusija. (Siemens 2014)

#### Latching relay

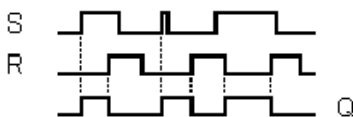


#### Short description

A signal at input S sets output Q. A signal at input R resets output Q.

Connection	Description
Input S	Set output Q with a signal at input S (Set).
Input R	Reset output Q with a signal at input R (Reset). Output Q is reset if S and R are both set (reset has priority over set).
Parameter	<b>Retentivity</b> set (on) = the status is retentive in memory.
Output Q	Q is set with a signal at input S and remains set until it is reset with signal at input R.

#### Timing diagram

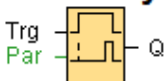


KUVA 11. RS-kiikku ja sen toiminta. (Siemens 2014)

### 5.1.9 On-delay

On-delay (kuva 12) toimii nimensä mukaisesti. Kun Trg, eli trigger-liittimeen tulee signaali, alkaa blokkiin määritetty aika käymään. Ajan käytyä loppuun, output liitin Q kytkeytyy päälle. Jos triggeriin tuleva signaali katkeaa ennen ajan kulumista, toiminto keskeytyy. Seuraavan signaalin tullessa aika alkaa taas alusta. (Siemens 2014)

#### On-delay



#### Short description

The output does not switch on until a configured delay time has expired.

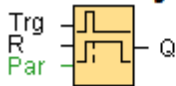
Connection	Description
Trg input	The Trg (Trigger) input triggers the on-delay time.
Parameter	<b>T</b> : represents the on-delay time after which the output is switched on (output signal transition 0 to 1). <b>Retentivity</b> on = the status is retentive in memory.
Output Q	Q switches on after a specified time T has expired, provided Trg is still set.

KUVA 12. On-delay ja sen toiminta. (Siemens 2014)

### 5.1.10 Off-delay

Off-delayn (kuva 13) toimintaperiaate on lähes sama kuin On-delayn, mutta päinvastainen. Kun trigger-liittimeen tulee signaali, output Q asettuu. Kun trigger-liittimeen tuleva signaalitieto muuttuu 1 → 0 (laskeva reuna), blokki alkaa laskemaan sille määriteltyä aikaa. Ajan kuluttua loppuun, kytkeytyy output Q pois päältä. (Siemens 2014)

#### Off-delay



#### Short description

The output with off-delay resets after a defined time has expired.

Connection	Description
Input <b>Trg</b>	Start the off-delay time with a negative edge (1 to 0 transition) at input <b>Trg</b> (Trigger).
Input <b>R</b>	Reset the off-delay time and set the output to 0 via the <b>R</b> (Reset) input. Reset has priority over <b>Trg</b> .
Parameter	<b>T</b> : The output is switched off on expiration of the delay time <b>T</b> (output signal transition 1 to 0). <b>Retentivity on</b> = the status is retentive in memory.
Output <b>Q</b>	<b>Q</b> switches on for the duration of the time <b>T</b> after a trigger at input <b>Trg</b> .

KUVA 13. Off-delay ja sen toiminta. (Siemens 2014)

### 5.1.11 Viikoittainen ajastin

Viikoittaisen ajastimen (kuva 14) toimintaperiaate on hyvinkin yksinkertainen. Blokkiin voidaan määrittellä halutuille päiville ajat, jolloin output Q on päällä. Tällöin asetuksiin laitetaan siis aika jolloin output halutaan päälle, sekä aika jolloin se halutaan sammuttaa. Blokissa on myös vaihtoehtona antaa pulssitietoa lähdöstä. Tällöin parametreihin määritellään vain yksi kellonaika, eli aika, jolloin pulssi lähdöstä halutaan. (Siemens 2014)

#### Weekly timer



#### Timing diagram (three practical examples)



Cam 1:	Daily:	06:30 h to 8:00 h
Cam 2:	Tuesday:	03:10 h to 04:15 h
Cam 3:	Saturday and Sunday:	18:30 h to 23:10 h

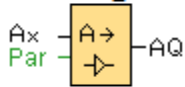
KUVA 14. Viikoittainen ajastin ja sen toiminta. (Siemens 2014)



### 5.1.12 Analogiavahvistin

Analogiavahvistimella (kuva 15) voidaan nimensä mukaan vahvistaa sekä skaalata anturilta tulevaa analogiatiedon lukemaa. Esimerkiksi ultraäänianturilta tuleva virtaviesti voidaan käsitellä vahvistimella siten, että sen pienin antama virta vastaa lukemaa 0 ja isoin antama virta vastaa lukemaa 1000. (Siemens 2014)

#### Analog amplifier



#### Short description

This SFB amplifies an analog input value and returns it at the analog output.

Connections	Description
Input <b>Ax</b>	Input Ax is one of the following analog signals: <ul style="list-style-type: none"> <li>● AI1 to AI8 (↺)</li> <li>● AM1 to AM6 (if 0BA6), AM1 to AM16 (if 0BA7), or AM1 to AM64 (if 0BA8)</li> <li>● NAI1 to NAI32 (if 0BA7 or 0BA8)</li> <li>● AQ1 and AQ2 (if 0BA7), or AQ1 to AQ8 (if 0BA8)</li> <li>● NAQ1 to NAQ16 (if 0BA7 or 0BA8)</li> <li>● The block number of a function with analog output</li> </ul>
Parameter	<b>Gain</b> Range of values: -10.00 to 10.00 <b>Offset</b> Range of values: -10000 to 10000 <b>p</b> : Number of decimals Possible settings: 0, 1, 2, 3
Output <b>AQ</b>	Value range for AQ: -32768 to +32767
* AI1 to AI8: 0 to 10 V corresponds with 0 to 1000 (internal value).	

KUVA 15. Analogiavahvistin ja sen toiminta. (Siemens 2014)

### 5.1.13 Analogiavertailija

Analogiavertailijaan (kuva 16) voidaan määrittellä lukemat, jonka mukaan output Q määrittyy joko todeksi tai epätodeksi, riippuen vertailijaan tulevasta analogiatiedosta. Tämä on se blokki, jolla tullaan tarkastelemaan vedenpinnan rajoja tässä työssä. (Siemens 2014)

## Analog comparator



### Short description

The output is set and reset depending on the difference  $A_x - A_y$  and on two configurable thresholds.

Connection	Description
Inputs $A_x$ , $A_y$	Inputs $A_x$ , $A_y$ are two analog signals from the followings: <ul style="list-style-type: none"> <li>● AI1 to AI8 <sup>(*)</sup></li> <li>● AM1 to AM6 (if 0BA6), AM1 to AM16 (if 0BA7), or AM1 to AM64 (if 0BA8)</li> <li>● NAI1 to NAI32 (if 0BA7 or 0BA8)</li> <li>● AQ1 and AQ2 (if 0BA7), or AQ1 to AQ8 (if 0BA8)</li> <li>● NAQ1 to NAQ16 (if 0BA7 or 0BA8)</li> <li>● The block number of a function with analog output</li> </ul>
Parameter	<b>Gain</b> Range of values: -10.00 to 10.00 <b>Offset</b> Range of values: -10000 to 10000 <b>On:</b> On threshold Range of values: -20000 to 20000 <b>Off:</b> Off threshold Range of values: -20000 to 20000 <b>p:</b> Number of decimals Range of values: 0, 1, 2, 3
Output $Q$	$Q$ is set or reset depending on the set thresholds.
* AI1 to AI8: 0 to 10 V corresponds with 0 to 1000 (internal value).	

KUVA 16. Analogiavertailija ja sen toiminta. (Siemens 2014)

## 5.1.14 Rajakytkin

Rajakytkin (kuva 17) laskee Fre-inputtiin (frequency) tulevan signaalin muutoksia  $0 \rightarrow 1$ . Muutokset  $1 \rightarrow 0$  jätetään laskematta. Blokkiin määritellään haluttu aika, ja pulssien määrä. Kun pulsseja tulee asetettu määrä asetetun ajan sisällä, antaa blokki ulostulosta pulssin. (Siemens 2014)

### Threshold trigger



#### Short description

The output is switched on and off depending on two configurable frequencies.

Connection	Description
Input <b>Fre</b>	The function counts 0 to 1 transitions at input <b>Fre</b> . Transitions from 1 to 0 are not counted. <ul style="list-style-type: none"> <li>● Use the inputs I3, I4, I5, and I6 for high-frequency counts (LOGO! 12/24RC/RCo, LOGO! 12/24RCE, LOGO! 24/24o and LOGO! 24C/24Co): max 5kHz, if the fast input is directly connected to the threshold trigger function block</li> <li>● Use any other input or circuit element for low frequencies (typical 4 Hz).</li> </ul>
Parameter	<b>On:</b> On threshold Range of values: 0000 to 9999 <b>Off:</b> Off threshold Range of values: 0000 to 9999 <b>G_T:</b> Time interval or gate time during which the input pulses are measured. Range of values: 00:00 s to 99:99 s
Output $Q$	$Q$ is set or reset according to the threshold values.

KUVA 17. Rajakytkin ja sen toiminta. (Siemens 2014)

### 5.1.15 Data log

Data log (kuva 18) tarkoittaa toimintoa, jolla voidaan esimerkiksi muistikortille kerätä eri blokeilta niiden arvoja. Muistikortin voi ottaa Logosta irti ja laittaa tietokoneeseen, josta tallennetut arvot saa näkyviin vaikkapa Exceliin. (Siemens 2014)

Kun blokin liittimeen En tuleva signaali vaihtuu  $0 \rightarrow 1$ , niin tiedonkeruu käynnistyy. Ne blokit, joista tietoa halutaan kerätä, määritellään erikseen data log -blokin sisäisillä parametreilla. (Siemens 2014)

#### What is a data log?



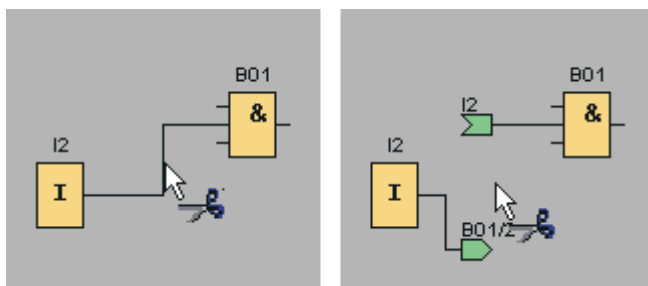
You can configure a data log to record the actual values of the function blocks and memory areas in a circuit program.

Connection	Description
En	The Data Log begins logging data with a positive edge (0 to 1 transition) at input En (Enable)

KUVA 18. Data log ja sen toiminta. (Siemens 2014)

### 5.1.16 Cut/Join

Tämä työkalu on sitä varten, että liittimien välille vedettyjä viivoja voidaan katkoa, että kuva pysyisi selkeänä. Katkotun viivan kummassakin päässä on tieto siitä, mistä toisen pään löytää (kuva 19). (Siemens 2014)



KUVA 19. Cut/Join-työkalu ja sen toiminta. (Siemens 2014)

## 5.2 Toteutus

Nyt kun on käyty läpi tässä työssä käytetyt blokit ja toiminnot, voidaan alkaa purkamaan itse ohjelman tekoa. Niin kuin aiemmin jo todettiin, samaan lopputulokseen voidaan

päästä monellakin eri ratkaisulla. Ohjenuoria ohjelman tekoon ja rakentamiseen on kuitenkin olemassa ja niitä esitellään seuraavaksi.

### **5.2.1 Aloittaminen**

Ohjelman tekoa aloittaessa on muutamia asioita, joihin jokaisen pitäisi kiinnittää huomiota. Järjestelmän turvallinen käyttö edellyttää sitä, että kohtuullisesti ennakoitavat inhimilliset käyttäjän erehdytykset eivät aiheuta vaaratilannetta (Siirilä 2009, 211). Käytännössä esimerkiksi, että ohjelma ei saa suorittaa virhetoimintaa, jos käyttäjä vaikka painaa käynnistyksen aikana vahingossa sammutuspainiketta.

Kun ohjelmaa tehdään, on pyrittävä yksinkertaisuuteen ja hallittavuuteen. Ohjelmisto tulisi järjestää modulaarisiin rakenteisiin, jotka ovat selvästi tunnistettavissa. Tämä helpottaa mm. ohjelman eri toimintojen seuraamista, vian etsintää ja mahdollista kehitystyötä. Viitaten myös käyttäjän erheisiin, ohjelmisto tulisi suojata virheellisiltä tietojen syötöiltä. (Siirilä 2009, 201–202.)

Jos mahdollista, on hyvä käyttää jo ennalta hyvin toimiviksi todettuja ohjelmistoja mahdollisimman pitkälle. Näin voi säästää ohjelmiston suunnittelussa kallisarvoista aikaa, eikä tarvitse ns. keksiä pyörää uudestaan. Rakenteiden tulisi olla myös helposti muutettavissa. Tämä mm. siksi, että joskus järjestelmään saattaa tulla muutoksia, jotka vaativat ohjelman muokkaamista. (Siirilä 2009, 202.)

Ohjelmiston dokumentoinnin pitää olla laadukasta. Tämä sisältää mm. valmistuksen, kehityksen ja ylläpidon dokumentoinnin (esim. muutosten päivittäminen kuviin) sekä käyttöön liittyvän dokumentoinnin (esim. käyttöohjeet). Kaiken tämän lisäksi ohjelman teon alussa on vielä hyvä kerrata aiemmin laaditut järjestelmän vaatimukset sekä komponenttivalintojen vaikutukset itse ohjelmaan. (Siirilä 2009, 202.)

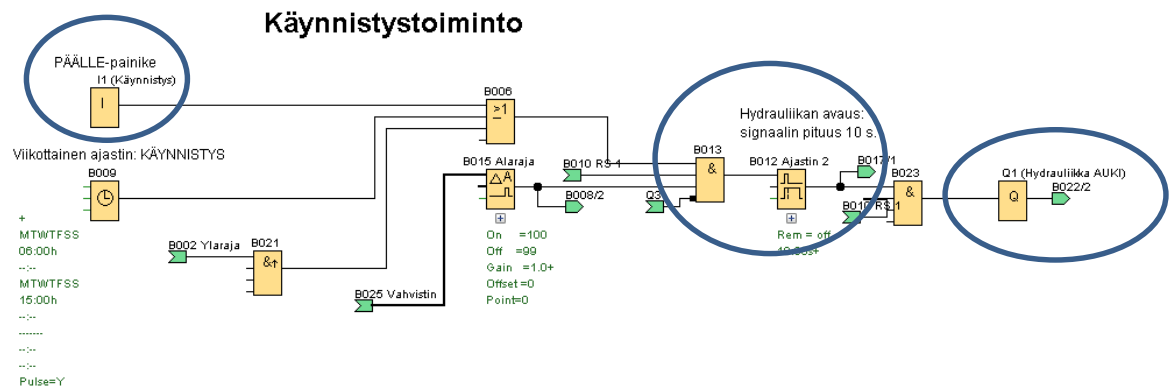
### **5.2.2 Ohjelman rakennus**

Jotta ohjelman rakenne pysyisi mahdollisimman selkeänä, jaoin ohjelman toiminnot viiteen eri pääosaan: Käynnistystoiminto, sammutustoiminto, verkon tilan tarkastelu, vedenpinnan tarkastelu sekä pulssin tarkastelu. Näin pystyin jo alusta asti rakentamaan ohjelmaa siten, että kukin yksittäinen blokki menisi suoraan jonkin ns. päätoiminnon alle. Se

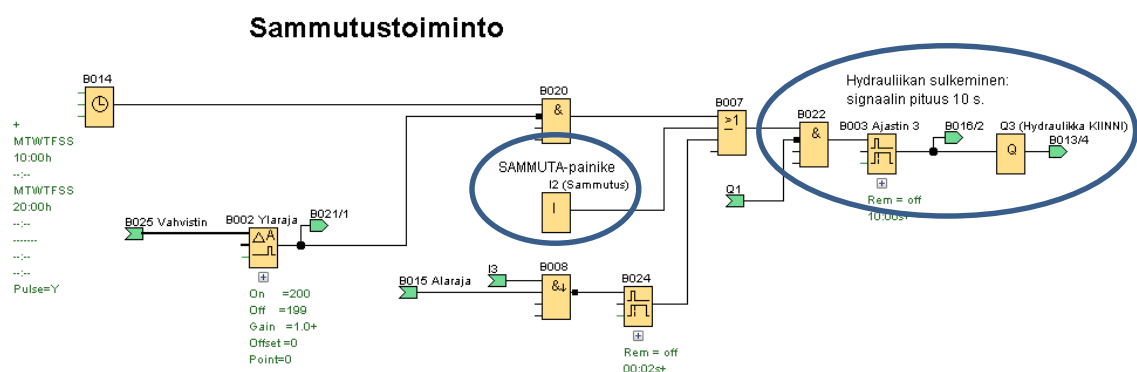
auttoi suuresti ohjelman tekemisen aikaisessa testaamisessa sekä ohjelman vikojen ja toimintojen tutkimisessa.

Lähdin muodostamaan ohjelmaa kriittisten toimintojen kautta. Tarkoittaen sitä, että aluksi tein ohjelmaan vain ja ainoastaan keskeisimmät toiminnot, ilman mitään lisäyksiä ja hienouksia. Sitten pikkuhiljaa eri toimintoja lisäämällä muodostin voimalaa ohjaavan kokonaisuuden. Kuva koko ohjelmasta on tämän raportin liitteenä.

Ensimmäisenä muodostin käynnistys- ja sammutustoimintoihin yksinkertaiset ohjaukset hydraulikkaa varten. Inputeilla I1 ja I2 sai annettua hydraulikkaa ohjaaville outputeille signaalin 10 sekunnin ajan Off-delayta käyttäen. Kuvista 20. ja 21. olen ympäröinyt ne blokit, joidenka ympärille lähdin toimintoja jalostamaan.



KUVA 20. Käynnistystoiminnon ohjauksen pohjana käytetyt blokit ympäröitynä.

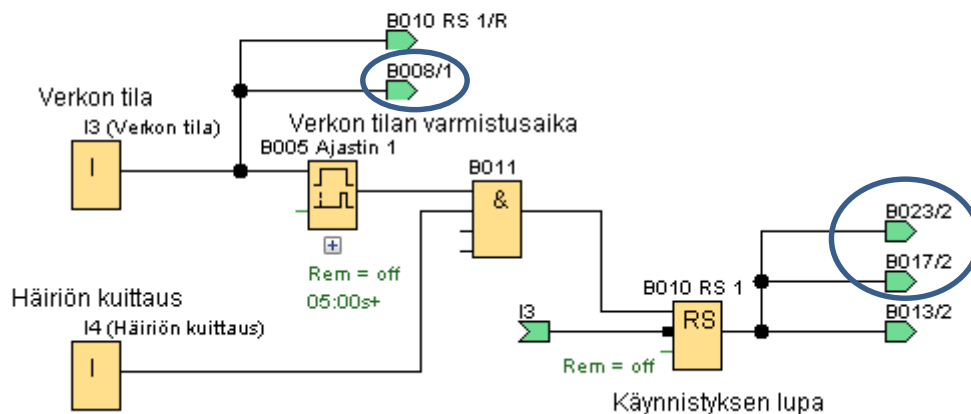


KUVA 21. Sammutustoiminnon ohjauksen pohjana käytetyt blokit ympäröitynä.

Tein alkuun myös yksinkertaisen verkon tilan tarkastelun häiriön kuittauksella (kuva 22). Kun I3 antaa signaalin, eli verkko on kunnossa ja päällä, alkaa ajastin laskea siihen ase-

tettua turva-aikaa. Kun asetettu aika on kulunut, pystyy häiriön vasta kuittaamaan painikkeella, joka on kytketty inputtiin I4. Vasta sen jälkeen RS-kiikku saa S-liittimeen signaalin, ja sen tulo muuttuu todeksi. RS-kiikun outputista signaali jatkaa eteenpäin käynnistystoimintoon tietona siitä, että voimalaa saa alkaa käynnisteleään. Verkon tilaa tarkasteleva tulo I3 on kytketty käännetyksi myös RS-kiikun reset-liittimeen. Näin ollen RS-kiikku on aina resetoituna, jos verkko on alhaalla.

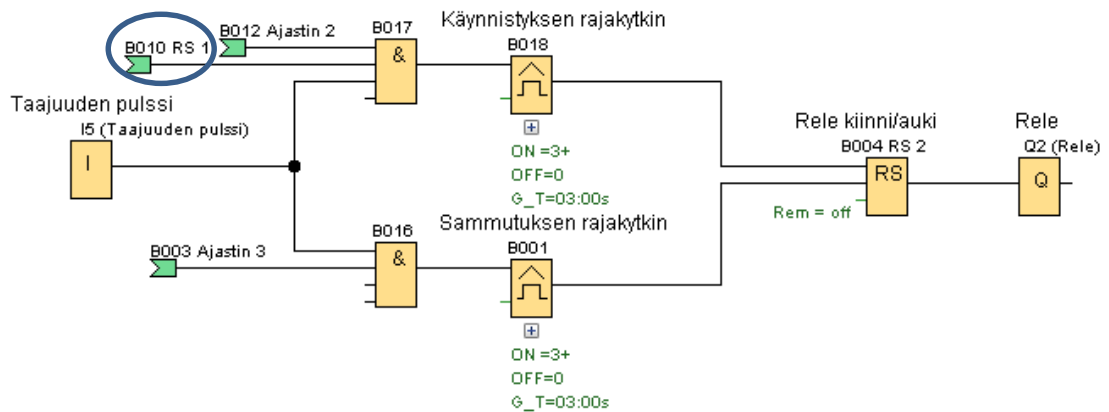
## Verkon tilan tarkastelu



KUVA 22. Verkon tilan tarkastelu, ympyrällä merkityt toiminnot, jotka tehtiin ohjelman myöhemmässä vaiheessa.

Seuraava päätoiminto, jonka tein, oli pulssin tarkastelu. Tällä toiminnolla tarkastellaan generaattorin taajuuden sopivuutta verkon taajuuteen. Tähän kokonaisuuteen ei ohjelman edetessä tullut juurikaan mitään lisää (kuva 23). Input I5 antaa signaalin aina kun pulssi-anturi antaa signaalin. Tätä signaalia tarkastellaan tilanteesta riippuen, joko käynnistys-rajakytkimellä tai sammutuksen rajakytkimellä. Anturin pulssisignaalit pääsevät rajakytkimille ainoastaan silloin kuin käynnistys- tai sammutustoiminnossa annetaan hydraulikan ohjaussignaalia. Tämä estää sen, että ohjelmassa ei suoriteta turhaa pulssin laskentaa. Pulssi-anturi on kytketty tuloon I5 siksi, että se on ns. nopea input, eli tarkoitettu isommille taajuuksille. Vesivoimalassa pulsseja pitää tulla 16 per sekunti, jotta generaattori on sopivassa taajuudessa syötettävän verkon kanssa.

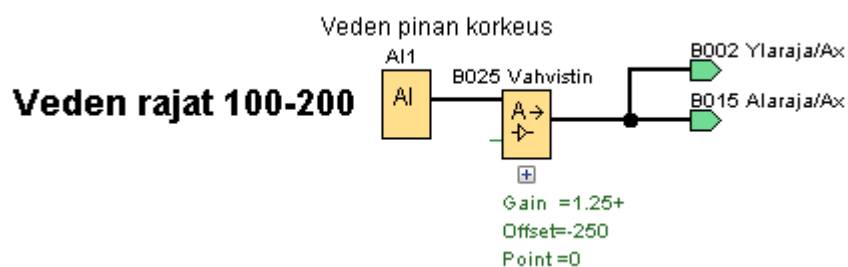
## Pulssin tarkastelu



KUVA 23. Pulssin tarkastelu, ympyrällä merkitty ohjelman myöhemmässä vaiheessa tulut toiminto.

Edellä mainituin tavoin sain yksinkertaisen version ohjelmasta tekemään perustoiminnot, eli käynnistämään hydraulikkaa ohjaavat outputit, tarkastamaan verkon tilan ja tarkastelemaan pulssia. Seuraava askel oli alkaa jalostamaan ohjelmaa.

Viimeinen ns. päätoiminto, jonka lisäksi, oli vedenpinnan rajojen tarkastelu (kuva 24). Niin kuin kuvasta näkee, tässä toiminnossa ei ole muuta kuin analoginen input sekä analogivahvistin. Tämä johtuu siitä, että analogiavertailijat ovat sijoitettuna käynnistys- ja sammutustoiminnon alle.



KUVA 24. Vedenpinnan tarkastelu.

Samalla lisätessäni vedenpinnan tarkastelua varten tarvittavat blokit, lisäksi käynnistys- sekä sammutustoimintoihin muita pieniä lisäominaisuuksia (kuva 25; kuva 26). Käynnistykseen yhdistin OR-blokilla painikkeen inputin rinnalle viikoittaisen ajastimen. Sammutustoiminnossa järjestyksestä tuli hieman erilainen. Viikoittainen ajastin yhdistyy ylärajaa

tarkkailevan vertailijan kanssa AND-blokkiin. Tällä AND-blokilla varmistetaan, että voimala ei sammuta itseään viikoittaisella ajastimella, jos vedenpinnan raja on yli sallitun. Näin vesivoimala pysyy käynnissä kunnes veden taso on taas asetettujen arvojen sisäpuolella.

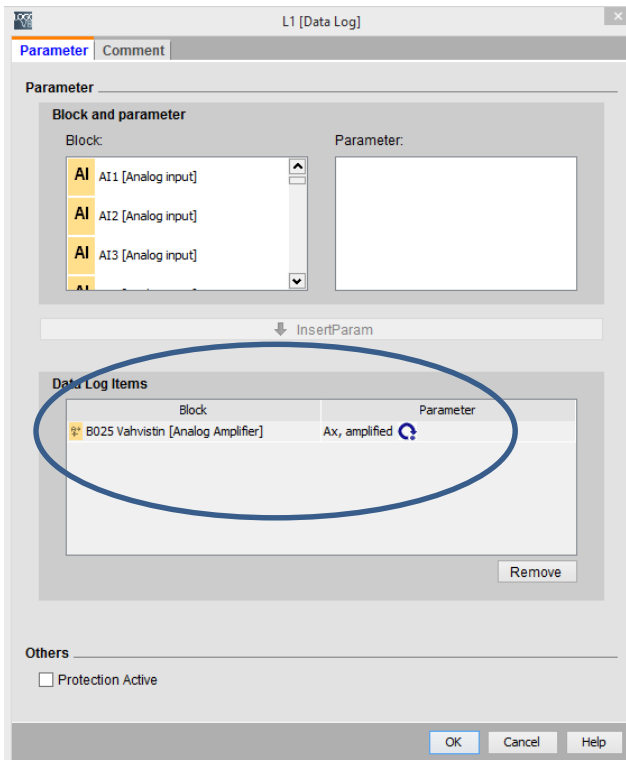
Samankaltainen tarkastelu tapahtuu myös käynnistystoiminnossa. Jos vedenpinnan raja alittaa sallitut arvot, ei voimalaa pystytä käynnistämään. Jos rajan alitus tapahtuu voimalan ollessa käynnissä, suorittaa ohjelma automaattisesti voimalan sammutustoiminnot. Vastaavasti jos vedenpinnan raja ylittyy voimalan ollessa sammuksissa, suorittaa ohjelma käynnistystoiminnot. Lisäsin ohjelmaan myös ominaisuuden, jolla estetään hydrauliiikan ohjaussignaalien samanaikainen päälle meno. Tämä oli yksinkertaista toteuttaa AND-blokeilla. Jos toinen hydrauliiikan ohjaussignaali on päällä, se estää toista ohjausta saamasta signaalia.

Ohjelmaan on sisällytetty myös toiminto, joka sammuttaa voimalan, jos verkon tila vaihtuu epätodeksi. Ohjelmaa tehdessä huomasin yhden ison suunnitteluvirheen. Verkon tilan tippuessa käynnistyksen aikana, tarkemmin hydrauliiikan avaamisen aikana, sammutustoiminto ei toteutunut. Näin ollen voimala olisi voinut jatkaa sähköntuotantoa verkkoon, vaikka verkko olisi alhaalla. Tämä voisi aiheuttaa suuria riskejä mm. verkkoa huoltaville työntekijöille.

Kehitin ratkaisun, jossa verkon tilan laskevaa reunaa käytettiin sammutustoimenpiteen suorittamiseksi tilanteessa kuin tilanteessa. Sammutustoiminnossa NAND-blokkiin tuleva input I3 on verkon tila. Jos verkko putoaa, eli se menee pois päältä, antaa NAND-blokki signaalin Off-delay -blokkiin. Siitä signaali jatkaa matkaansa aina hydrauliiikan ohjaukseen asti, ja sammutustoiminto suoriutuu hyvin. Off-delay -blokkia jouduin käyttämään sen takia, ettei meneillään oleva hydrauliiikan avaus signaali estä sammutustoimintoa. Off-delay on säädetty niin, että verkon tilan tarkastelussa ohjattava RS-kiikku ehtii resetoitumaan ja estämään käynnistystoiminnon jatkumisen ja käynnistyksen pulssin tarkastelun. Sammutustoiminto lähtee käyntiin heti resetoinnin jälkeen.



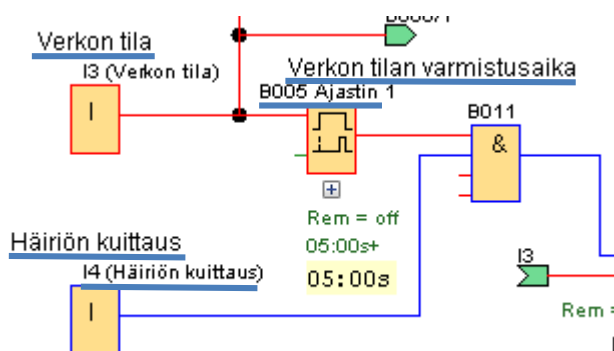




KUVA 28. Data log –blokin sisään on määritelty, että se kerää tietonsa vahvistimesta.

### 5.2.3 Kommentointi

Kommentointi on iso osa selkeää ohjelmaa ja sen avulla pyritään mahdollistamaan, että ohjelmaa ja sen toimintoja olisi helppo seurata. Hyvän kommentoinnin merkkejä ovat mm. selkeys, johdonmukaisuus, informatiivisuus ja sopivan tiivis sisältö. Kommentoissa siis ohjelman blokkeja tai toimintoja, tulisi pyrkiä yksinkertaiseen, jämäkkään tekstiin, joka kertoo tarpeellisen asian (kuva 29).

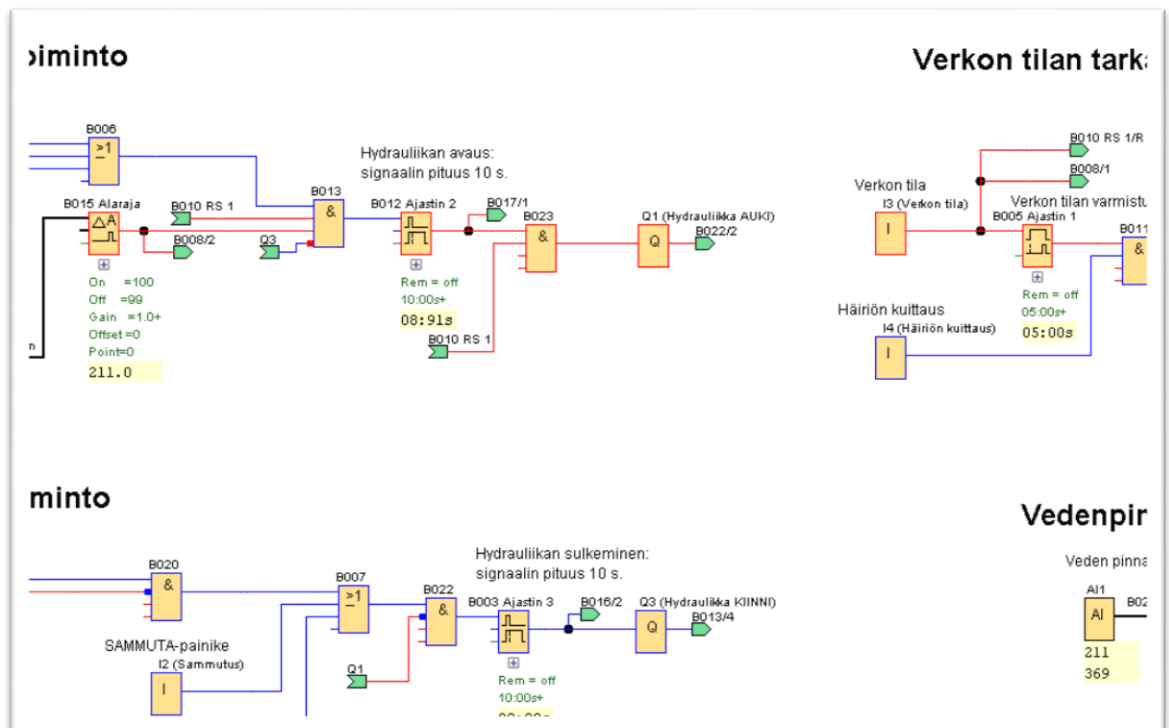


KUVA 29. Kommentointi ja nimeäminen ovat tärkeä osa ohjelman tekoa ja sen jakamista eri osiin.

Hyvä kommentointi auttaa myös itseä pysymään perillä kaikesta ohjelmoinnin aikana, sekä muita sen seuraamiseen myöhemmin. Mitä isompi ohjelmasta tulee, sen tärkeämpään rooliin kommentointikin nousee.

### 5.3 Ohjelman simulointi

Simulointi Logolla on erinomainen työkalu tarkastaa ja tutkia ohjelman toimintaa (kuva 30). Rakentaessani ohjelmaa, käytin simulointia erilaisten sovellusten testaamiseen, sekä vianetsintään. Jälkimmäiseen siinä mielessä, että kaikki toiminnot eivät mene aina juuri niin kuin on ajatellut. Simuloinnilla pystyy helposti paikantamaan ongelmakohdan.



KUVA 30. Simuloidessa LOGO!Soft Comfortilla, on signaalin seuraaminen helppoa, sillä se näkyy punaisena.

Logon simuloinnissa erittäin hyvää on vielä se, että signaalin kulun näkee jatkuvasti. Tämä helpottaa suunnittelussa, sillä ohjelmaa voi hioa todella pitkälle, ennen kuin sitä tarvitsee testata käytännössä.

Simuloinnin avulla pystytään helposti koittamaan, mitä ohjelmassa tapahtuu, jos jotakin nappia esimerkiksi painetaan ns. väärään aikaan. Jos vikapainalluksista syntyy ongelmia, on ne helppo ratkoa simuloinnista saadun tiedon avulla. Logon simulaattorissa inputit

voidaan määrittää painonapeiksi tai kaksi asentoiseksi kytkimiksi, riippuen millaista laitetta halutaan jäljitellä. Muun muassa näitä edellä mainittuja asioita testailin ohjelmaa tehdessäni jatkuvasti.

Ohjelmaa voidaan simuloida myös sykleittäin. Eli voidaan määritellä vaikka, että ohjelma menee 10 sykliä eteenpäin, jonka jälkeen nähdään mihin tilanteeseen päädytään. Sama asia voidaan toteuttaa myös halutulla ajanjaksolla (kuva 31). Etuuksia ovat myös se, että analogiatietoakin voidaan simuloida eräänlaisella liukusäätimellä, sekä ajastimien ja analogiatietojen arvot nähdään reaaliajassa simuloinnin ollessa käynnissä.



KUVA 31. Simuloida voidaan sykleissä, ajanjaksoina tai ihan vaan play ja stop tyyliin.

Ohjelman valmiiksi saatuani kävin läpi mahdollisia vikatilanteita ja virheellisiä painalluksia. Niin kuin ohjelman rakentamisen yhteydessä mainitsin, sain mielestäni eliminoidua kaikki kriittiset tai mahdollista vahinkoa aiheuttavat virhetilanteet. Voimalaa ei saa päälle ennen kuin häiriö on kuitattu. Häiriötila tulee aina kun verkon tila käy alhaalla. Eri nappien painallukset eivät aiheuta ohjelmassa vikatiloja tai samanaikaisia haitallisia output -ohjauksia. Voimalan saa painonapilla joka tilanteessa pois päältä. Automatiikan poiskytkentä tulisi vielä toteuttaa paikanpäällä siten, että Logon ohjauksen saa kytkettyä pois päältä.

#### 5.4 Riskitilanteet

Sähkökatkoksen sattuessa on vesivoimalassa akku, joka tuottaa tarvittavan jännitteen automatiikalle. Sähkökatkoksen seuranta voidaan tehdä samalla tavalla kuin verkon tilan seuranta, jolloin katkoksen sattuessa ohjelma sammuttaa akun avulla hallitusti voimalan, ja menee häiriötilaan. Jos akku ehtii loppumaan ennen sähköjen palaamista, on voimala silti häiriötilassa sähköjen tullessa takaisin päälle. Näin ollen voimalaa ei saada käyntiin ennen kuin häiriö on kuitattu

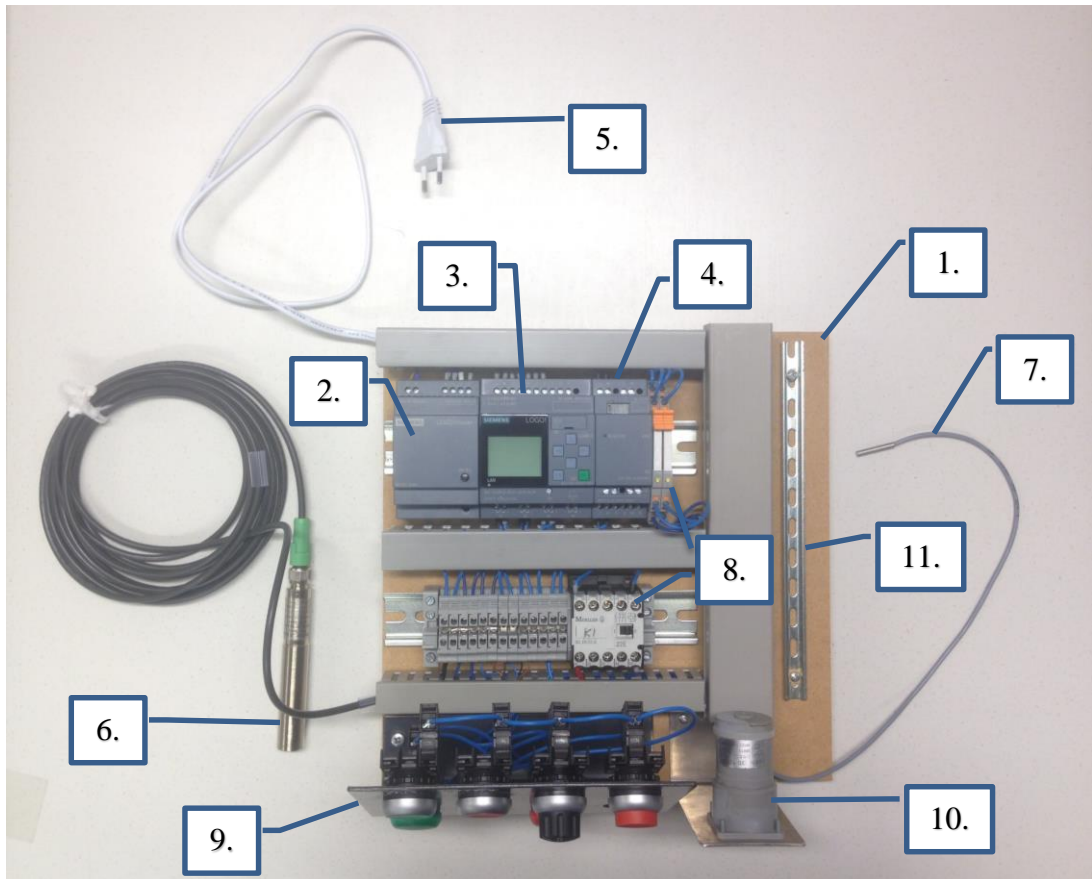
Mahdollisissa johdon katkeamistilanteissa on riski, että ohjelma ei pysty toteuttamaan tarvittavia toimintoja tai saamaan tietoa antureilta. Tämä tulisi huomioida järjestelmää

asennettaessa oikeilla johdinvalinnoilla, koteloinnilla, automatiikan fyysisellä poiskytkemisellä ja huolellisella asennustyöllä.

## 6 DEMOLAITE

### 6.1 Rakennus

Demolaitteen (kuva 32) rakentaminen tapahtui koululla koneautomaatiolaboratoriossa. Tarkoitus oli luoda vastaavanlaisia toimintoja, mitä oikeassa sovelluksessa tulisi olemaan. Demolaitteen rakennuksen tarkoitus on myös testata Logoa ja sen ohjelmaa käytännön komponenteilla. Demolaite on myös loistava tapa saada alustavat kytkentäkaaviot selville, tosin niitä saatetaan joutua muuttamaan hieman järjestelmää kytkettäessä itse voimalaan. Lisäksi kokonaiskuvasta näkee hyvin johtokourujen ja asennuskiskojen asettelun.



KUVA 32. Demolaite kokonaisuudessaan.

Demolaite rakennettiin seuraavista osista:

1. Puusta tehty asennuslevy
2. Logon Power-moduuli
3. LOGO! 8

4. Analogiainput lisämoduuli
5. Laitteen sähkönsyöttökaapeli
6. Ultraäänianturi
7. Induktiivinen anturi
8. Moottorin ja lamppujen releet
9. Käyttöpaneeli, jossa kiinni painonapit, kaksiasentoinen kytkin ja lamput
10. Moottori
11. Kisko, jolla saadaan aikaiseksi pulssia liikuttamalla induktiivista anturia reikien yli

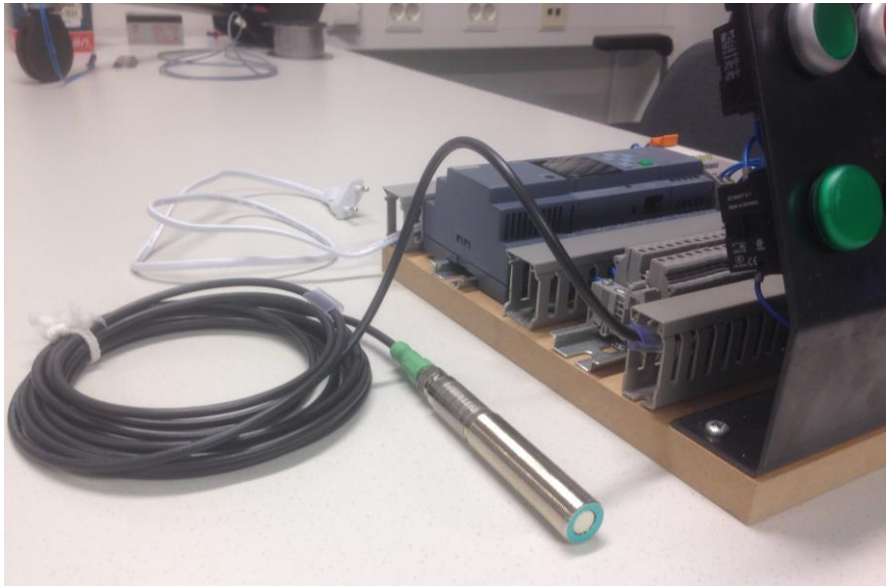


Kuva 33. LOGO! 8

## 6.2 Automatiikan testaus demolaitteella

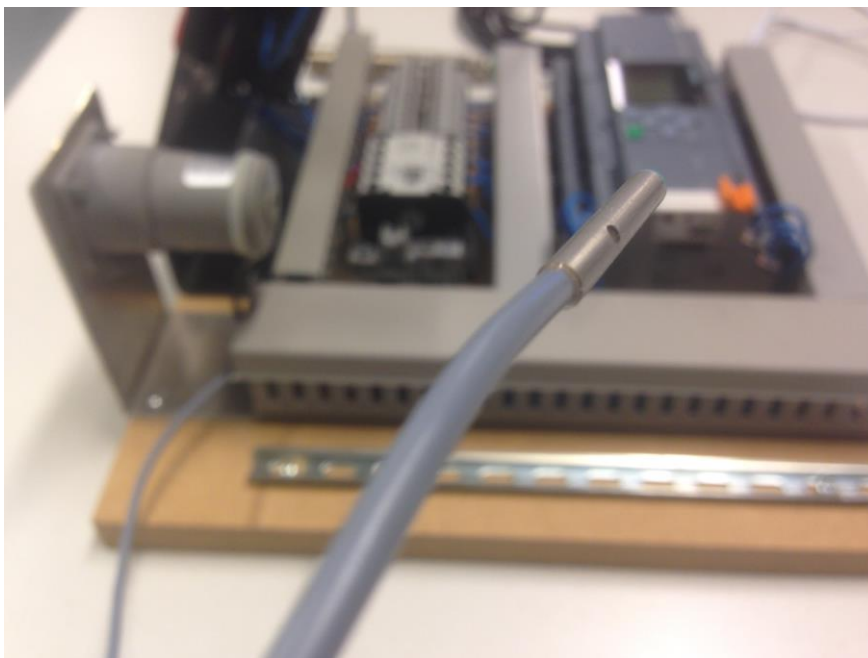
Ohjelman testaus tietokoneella on yksi iso osa automatiikan testausta, mutta sillä ei pystytä käymään laitteistojen toimintaa fyysisillä komponenteilla läpi. Sen vuoksi on hyvä tehdä demolaite, jolla eri komponenttien soveltuvuutta ja ominaisuuksia voidaan testata,

jotta aikanaan itse voimalan asennuksen ja käyttöönoton yhteydessä ei suuria yllätyksiä tulisi.



KUVA 34. Ultraäänianturi.

Demolaitteella testatessa voidaan myös kokoajan tietokoneella monitoroida, samaan tyyliin kuin simuloinnissa, mitä ohjelmassa tapahtuu. Näin saadaan selville mahdollisia komponenttien yhteensopimattomuuksia sekä järjestelmän toimintavirheitä.



KUVA 35. Induktiivinen anturi.



Koska LOGO!Soft Comfortissa oleva simulointiominaisuus on monipuolinen, pystyttiin ohjelman testaus käymään erittäin perusteellisesti läpi. Kaikki ohjelmassa havaitut virheet oli saatu kitkettyä aikaisemmissa testauksissa pois. Komponenttien toiminta oli sellaista, niin kuin ajateltiin. Demolaitteella taas pystyttiin testaamaan data log -toimintoa, jonka testaaminen pelkässä ohjelmointisovelluksessa ei onnistu.

## 7 POHDINTA

Tämä opinnäytetyö oli kaiken kaikkiaan erittäin mielenkiintoinen projekti. Emme olleet käsitelleet Logoa koulutuksen aikana paljoa. Oli siis mukavaa päästä tutkimaan jotain laitetta, joka ei entuudestaan ollut itselle täysin tuttu.

Työn lähtöasetelmat olivat selkeät. Tavoitteet olivat tiedossa, vaatimukset käsitelty ja työn rakenne hyvin suunniteltuna mielessä. Ihan työn alkuvaiheissa tosin lähdin tekemään ohjelmaa hieman liian aikaisin, vielä kun kaikki vaatimukset eivät olleet selvillä. Tämä johti siihen että kyseisestä ohjelmasta tuli epäselvä ja toiminnaltaan melko vajaa. Aloitin ohjelman teon täysin uudestaan vaatimusten varmistuttua.

Käytin opinnäytetyön tekemisen ajastani noin kolmanneksen ohjelman tekemiseen ja testaamiseen LOGO!Soft Comfortilla. Tämä oli hyvä päätös, sillä sain testattua ohjelman lähestulkoon läpikotaisesti. Ainut ominaisuus, jota en pystynyt Soft Comfortissa testaamaan, olikin data log. Näin ollen demolaitteen rakentamisen vaiheessa minun ei tarvinnut enää keskittyä ohjelman toimivuuteen, vaan komponenttien kytkentään ja toimintaan. Ohjelmasta tulikin omasta mielestäni erittäin toimiva kokonaisuus, joka varmasti soveltuu Mikkelin vesivoimalan ohjaukseen.

Työstä jätettiin pois Logon etäohjaukseen tarkoitettu GSM-moduuli. Ratkaisu perustui siihen, että minulla ei olisi ollut tarpeeksi aikaa perehtyä kyseisen moduulin toimintaan. Etäkäytettävyyttä mietitään Mikkelin voimalaan edelleen ja GSM-moduulin parissa työskennelläänkin jo, että siitä saataisiin hyvät käyttöohjeet. Vaikka etäkäytettävyyys ei toteutunutkaan, nousi GSM-moduulin rinnalle toiseksi vaihtoehdoksi WLAN-yhteyksillä toimiva ratkaisu. Näiden kahden välillä tuleekin tehdä tarkka vertailu, jos jompikumpi aikaan vesivoimalaan asennetaan.

Demolaitteella saatiin kuvattua hyvin komponenttien toimintaa Logon kanssa. Laitteesta saa hyvin myös hahmoteltua mahdollista ohjauskaapin kokoa voimalaan, sillä siitä nähdään millainen kokonaisuus kyseisestä automaatiojärjestelmästä oikein tulee. Voimalaan asennettavaan järjestelmään tulee joitakin komponenttimuutoksia, esimerkiksi ultraäänianturi tulee olemaan erilainen verrattuna demolaitteen vastaavaan. Voimalaan tulevalle ultraäänianturille pitää rakentaa myös teline, joka pitää anturin paikoillaan ja tasaa aallot.

Tähän sopisi esimerkiksi jonkinlainen putkimainen ratkaisu. Voi olla että telineeseen joudutaan tekemään talven varalle myös lämmitys, että anturi pystyy veden pintaa mittaamaan myös silloin.

Tekemäni järjestelmä olisi varmasti soveltuva moneen muuhunkin pienvesivoimalaan Suomessa. Uskonkin, että jos Mikkeliin asennettava järjestelmä toimii hyvin, saatetaan sitä tarvittavilla muutoksilla käyttää jossain muissakin, joko uusissa tai vanhoissa pienvesivoimaloissa. Vesivoimalan automatisointi tuo niin suuret käytön hyödyt, että toimivalle järjestelmälle löytyisi varmasti kysyntää.

Vaikka oma opinnäytetyöni loppuu, on Mikkelin vesivoimala askeleen lähempänä automatisointia. Lopullisten komponenttien määrityksen, kotelointien ja testausten jälkeen järjestelmä varmasti voimalaa vielä pyörittää. Olen myös itse luvannut olla avuksi tarpeen vaatiessa, sillä hienoahan se olisi nähdä omatekemän ohjelman pyörittävän kyseistä voimalaa.

## LÄHTEET

Energiateollisuus. 2015. Vesivoima. Luettu 3.11.2015. <http://energia.fi/energia-ja-ym-paristo/energialahteet/vesivoima>

Siemens. 2014. LOGO!Soft Comfort Online Help. Luettu 22.11.2015.

Nordpoolspot. 2015. Market data. Elspot prices. Luettu 23.10.2015. <http://www.nordpoolspot.com/Market-data/Elspot/Area-Prices/ALL1/Hourly/?view=chart>

Siirilä, T. 2009. Koneturvallisuus. Ohjausjärjestelmät ja turvalaitteet. Keuruu: Otavan Kirjapaino Oy.

Sundquist, M. Työturvallisuus automaation varassa. Luettu 4.12.2015. [https://www.automatioseura.fi/index/tiedostot/Tyoturvallisuus\\_automaaation\\_varassa.pdf](https://www.automatioseura.fi/index/tiedostot/Tyoturvallisuus_automaaation_varassa.pdf)

Valkeejärvi, P. 2012. Öljynsiirtojärjestelmän automatisointi. Automaatiotekniikan koulutusohjelma. Seinäjoen ammattikorkeakoulu. Opinnäytetyö.

Vattenfall. 2014. Tietoa energiasta. Vesivoima – toimintaperiaate. Luettu 3.11.2015. <http://corporate.vattenfall.fi/tietoa-energiasta/sahkon-jalammontuotanto/tietoa-vesivoimasta/vesivoima-toimintaperiaate/>



## Liite 2. I/O-lista

I/O-lista			
Osoite	Toiminto	Tunnus	Huom.
I1	Käynnistys	S1	Painonappi
I2	Sammutus	S2	Painonappi
I3	Verkon tila	S3	Valintakytkin
I4	Häiriön kuittaus	S4	Painonappi
I5	Taajuuden pulssi	C1	Induktiivinen anturi
AI1	Veden pinnan korkeus	C2	Ultraäänianturi
Q1	Hydrauliikan ohjaus, auki	K1	Rele
Q2	Verkkoon kytkeytyminen	K2	Rele
Q3	Hydrauliikan ohjaus, kiinni	K3	Rele