



# **Tyypipiirien luonti Vijeo Citect - valvomo-ohjelmistolla**

Lauri Rajamäki

Opinnäytetyö  
Marraskuu 2014  
Sähkötekniikka  
Automaatiotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkötekniikka  
Automaatiotekniikka

RAJAMÄKI, LAURI:

Tyypipiirien luonti Vijeo Citect -valvomo-ohjelmistolla

Opinnäytetyö 58 sivua, joista liitteitä 5 sivua  
Marraskuu 2014

---

Insta Automation Oy on kokonaisvaltainen automaatioalan laite- ja ohjelmistotoimittaja, ja sen tuotteisiin kuuluu myös asiakas- ja projektikohtaisten valvomo-ohjelmistojen toimittaminen. Ohjelmisto- ja valvomosuunnittelun tehostamiseksi yritys on kehittänyt käytetyimpiin ohjelmistoihinsa tyypipiirit, joiden avulla valvomosta hallitaan toimilaittekohtaista ohjausta ja tiedonsiirtoa. Vijeo Citect on Schneider Electricin omistama valvomo-ohjelmisto, jolle lähdettiin kehittämään toiminnaltaan vastaavia tyypipiirejä.

Tässä työssä paneuduttiin valvomon ja valvomo-ohjelmiston toimintaan, kommunikointiyhteyden muodostamiseen kahden eri laitevalmistajan laitteiden välille ja erityisesti tyypipiirien luontiin Vijeo Citect -valvomo-ohjelmistolla. Haastavaa tyypipiirien teossa oli löytää ohjelmointitekniikka, joka varmistaa, että samaa tyypipiiriä voidaan käyttää samankaltaisten toimilaitteiden kanssa ja että luodut tyypipiirit vastaavat toiminnaltaan yrityksen aikaisemmin muille ohjelmistoille luomia tyypipiirejä: kone, mittaus, säädin ja venttiili.

Työn tuloksena yritys saa itselleen omalla väripaletillaan luodut toimivat tyypipiirit ja käyttöohjeet sekä tyypipiirien käyttöönottoon että kommunikointiyhteyden muodostamiseen valvomo-ohjelmiston ja logiikan välille. Edelleen kehittämällä luotujen tyypipiirien animoituja ominaisuuksia ja toiminnallisuutta voisi lisätä: vain ohjelmointi on rajana.

## **ABSTRACT**

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

**LAURI RAJAMÄKI**

Creating a Basic Circuits with Vijeo Citect SCADA Software

Bachelor's thesis 58 pages, appendices 5 pages  
November 2014

---

Insta Automation Oy is a comprehensive hardware and software supplier in the field of automation, and its products include client and project-based custom designed SCADA software. To enhance software and SCADA design the company developed basic circuits for its most used software, which enable device-specific control and information transfer through SCADA software. Vijeo Citect is SCADA software owned by Schneider Electric, for which I started to develop functionally similar type of basic circuits.

This thesis focuses on the functionality of SCADA software, on setting up communication between devices from different manufacturers, and particularly on creating the basic circuits with Vijeo Citect SCADA software. The challenge in creating the basic circuit was to find a programming technique that ensures that the created basic circuits would function with all devices of the same type in the field, and that they would be functionally equivalent to the basic circuits the company has created in the past for other software. The basic circuits created were machine, measurement, controller and valve.

As a result of this work, the company obtains working basic circuits created to use the color palette of their own brand, and the manuals for putting the basic circuits into use and setting up communication connection between SCADA software and PLC. Further development of the basic circuits could result in more efficient use of animated functions; the programming is the only limit.

---

Key words: Insta Automation, Vijeo Citect, Scada software, basic circuits

## SISÄLLYS

1	JOHDANTO.....	6
2	TYÖPROSESSI.....	7
	2.1 Insta Automation Oy.....	7
	2.2 Työn teko.....	9
	2.3 Työn tavoitteet.....	10
3	VALVOMO.....	11
	3.1 Valvomon tehtävä.....	11
	3.2 Valvomo-ohjelmisto.....	12
	3.2.1 Vijeo Citect.....	13
	3.2.2 Simatic WinCC.....	13
	3.2.3 InTouch.....	14
	3.3 Tagit eli muuttujat.....	15
	3.4 OPC.....	16
4	VIJEO CITECTIN JA SIEMENS S7:N VÄLINEN KOMMUNIKOINTI.....	17
	4.1 Tarve ja ajurivalinta.....	17
	4.2 Siemens-yhteysasettelut.....	18
	4.3 Vijeo Citect -yhteysasettelut.....	21
5	TYYPPIPIIRIT.....	26
	5.1 Tyyppipiirien ulkoasu.....	27
	5.1.1 Citect Graphics Builder -työkalut.....	28
	5.1.2 Värivalinta.....	30
	5.1.3 Omat symbolit.....	31
	5.1.4 Muut tyyppipiireissä käytetyt symbolit, arvokentät ja painonapit.....	32
	5.2 Tyyppipiirikohtaiset muuttujat ja niiden hallinta.....	34
	5.2.1 Ulkoiset muuttujat.....	36
	5.2.2 INOUT_BITIT.....	38
	5.2.3 Sisäiset muuttujat.....	40
	5.2.4 Genie-alustus.....	40
	5.2.5 Sisäisten muuttujien alustus.....	42
	5.3 Tyyppipiiriobjektien ohjelmointi.....	43
	5.3.1 Objektien yleiset ominaisuudet.....	45
	5.3.2 Numero- ja tekstikenttien ohjelmointi.....	46
	5.3.3 Nappien ja symbolien ohjelmointi.....	48
	5.3.4 Kuvioden ohjelmointi.....	49
	5.4 Tyyppipiirien käyttöönotto ja testaus.....	50
6	KÄYTTÖOHJE.....	51

7 POHDINTA.....	52
LÄHTEET .....	53
LIITTEET .....	54
Liite 1. Konepiirin Super Genie .....	54
Liite 2. Mittauspiirin Super Genie.....	55
Liite 3. Säädinpiirin Super Genie .....	56
Liite 4. Venttiilipiirin Super Genie.....	57
Liite 5. Testisivu.....	58

## 1 JOHDANTO

Ihmiskunta ja sen kehittämä teknologia on ottanut viime vuosikymmeninä jättiharppauksen. Teknologian kehitys nopeutuu eksponentiaalisesti kasvavalla asteikolla. Automaatiotekniikan kehittyessä kehittyvät ja laajenevat tuotantolaitokset ja prosessit, joita ihminen hallitsee valvomo-ohjelmiston välityksellä. Prosessien kehittyessä kasvaa niissä olevien laitteiden määrä ja näin ollen valvottavien yksiköiden määrä. Kuitenkin niissä olevien toimilaitteiden, esimerkiksi sähkömoottorin, toimintaperiaate säilyy samanalaisena. Tällöin niiden hallintaan tarvittavaa informaationvälitystä voidaan ohjata samanklaisilla ohjausyksiköillä eli tyyppipiireillä valvomosta käsin.

Insta Automation on luonut käyttöönsä tällaiset toimilaitekohtaiset tyyppipiirit, joiden käyttö nopeuttaa automaatioprojektin ohjelmistosuunnittelua ja standardoi kunkin projektikohtaisen ohjelmiston toiminnan. Tyyppipiirit on rakennettu eri ohjelmistojen välillä siten, että ne vastaavat toiminnaltaan ja sähköiseltä viestinnältään toisiaan. Niiden käyttö on siis mahdollista myös silloin, kun järjestelmää hallitaan eri automaatiotekniikka-alan yritysten tuotteilla. Vijeo Citect on Schneider Electricin omistama valvomo-ohjelmisto, jolle yritys ei ole aikaisemmin luonut tyyppipiirejä. Opinnäytetyön tavoitteena on siis luoda Vijeo Citectillä tyyppipiirit, jotka vastaavat toiminnaltaan yrityksen aikaisemmin luomia tyyppipiirejä.

Opinnäytetyössä tutkitaan kommunikointiyhteyden muodostamista eri ohjelmistotoimittajien tuotteiden välillä. Lisäksi tutkitaan valvomon ja valvomo-ohjelmistojen toimintaa. Tutkimuksen pääpaino on kuitenkin Vijeo Citect -valvomo-ohjelmiston toiminnassa ja siinä, miten sitä käytetään tyyppipiirien luomiseen.

## 2 TYÖPROSESSI

### 2.1 Insta Automation Oy

Insta Automation Oy on osana Tamperelaista teollisuus- ja teknologiakonsernia, Insta Group Oy:tä. Insta Group Oy on 1960-luvulla alkunsa saanut perheyritys, jonka omistus on säilynyt Mattsonin perheellä yrityksen perustamisesta asti. Finn Mattson perusti Automatiikka-Asentajat Mattson & Kumppanit vuonna 1960. Yritys sai vahvan jalansijan automaatioalalla Suomessa prosessiteollisuuden kehittyessä hurjasti. 1960-luvun lopussa se työllisti jo 220 henkilöä. Liiketoiminnan kehittyessä Automatiikka-Asentajien rinnalle perustettiin tytäryhtiö Instrumentointi Oy, jonka tarkoitus oli keskittyä metsäteollisuuden kunnossapitotöihin. Pian yritys kuitenkin kasvatti liiketoimintaa ja alkoi valmistaa huoltamolaitteita ja korjaamaan lentokonemittareita (Insta 50 vuotta 2014).

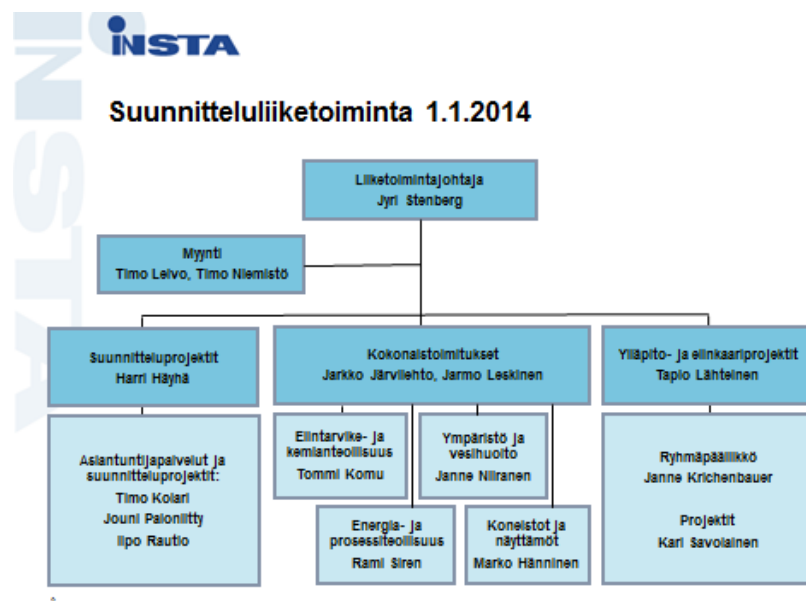
1980-luvulla laajenemista tapahtui jälleen ja uudeksi toimilinjaksi Instrumentointi Oy:llä alkoi puolustusteknologian kehitys. 1980-luvulla konsernissa tapahtui myös sukupolvenvaihdos, jolloin Markus Mattsonista tuli fuusioituneen Automatiikka-Asentajat Oy:n ja Instrumentointi Oy:n pääomistaja. Vuosituhannen alussa yhtiön rakenne ja toimisuunta vahvistuivat muutaman ison yrityskaupan jälkeen, mikä edelleen vahvisti Insta Group Oy:n osaamista kahdella toimialalla; teollisuusautomaatio ja puolustusteknologia. Vuonna 2014 konsernin liikevaihto oli 87 miljoonaa euroa ja se työllisti noin 750 henkilöä (Insta 50 vuotta 2014).

Insta DefSec Oy toimittaa palveluja ja ratkaisuja verkkokeskeisiin koulutus-, tietoliikenne- ja johtamisjärjestelmiin sekä tietoturvapalveluja sähköiseen liiketoimintaan. Lisäksi yritys tarjoaa ylläpito- ja integrointipalveluja puolustusjärjestelmiin. Heidän asiakkaisiinsa kuuluvat valtionhallinto, puolustusvoimat ja turvallisuusorientoituneita organisaatioita (Insta DefSec Oy 2014).

Insta Automation Oy tuottaa sähköautomaatiopalveluita asiakkaan projektin koko elinkaarelle, sillä yritys hallitsee prosessin kaikki toimenpiteet: suunnittelun, keskusvalmistuksen, asennuksen, kokonaistoimitukset, ylläpidon ja modernisoinnin. Yhtiön tärkeimmät asiakkaat löytyvät seuraavilta toimialoilta: prosessi- ja energiateollisuus, materiaalin käsittely, näyttämöt sekä vesilaitokset. Insta Automation Oy:llä on vahva osaa-

minen energia-alalla ja se tekee yhteistyötä johtavien järjestelmätoimittajien, kattilavalmistajien ja loppukäyttäjien kanssa. Sähköautomaatio-osaamista energia-alalla Insta Automation Oy:llä löytyy muun muassa biopolttoaine- ja soodakattiloista, hiilivoimalaitoksista ja vesilaitoksista. Prosessiteollisuuteen, eli muun muassa puunjalostusteollisuuteen, metallien valmistukseen ja jalostukseen sekä petrokemianteollisuuteen, yhtiö tarjoaa laiteriippumatonta palvelua esisuunnittelusta aina käyttöönottoon asti. Yhtiö toimittaa myös automaatio- ja sähkökeskukset sekä pneumatiikkakeskukset asiakkaalle. Insta Automation Oy:llä on Suomessa kahdeksan toimipistettä, joiden painopisteet ovat Keski- ja Etelä-Suomessa. Kaikkien toimipisteiden kautta yhtiö tarjoaa alueellista huolto- ja kunnossapitopalvelua. Tarjolla on ympärivuorokautinen huoltopalvelu, jonka kautta asiakas saa yhteyden yhtiöön ongelman ilmetessä. Insta Automation Oy on jaettu viiteen eri toimiyksikköön, joista jokainen hoitaa vastuullista alaansa, ja joiden asiakkailleen tarjoama yhteistyö tuottaa kokonaisvaltaisen automaatiopalvelun. Insta Automation Oy:n liikevaihto on 56 prosenttia konsernin liikevaihdosta eli noin 49 miljoonaa euroa (Insta Automation Oy 2014).

Insta Automation Oy:n suunnitteluosasto (IAS) tuottaa asiakkaalle suunnitelmia automaatioon, instrumentointiin ja prosessisähköistykseen. Tuotettavia suunnitelmia ovat muun muassa esi- ja hankintasuunnittelu, perus-, kenttä-, sähkö- sekä järjestelmäsuunnittelu. Se on jaettu sisäisesti tuottaen palveluita eri teollisuudenalojen haaroihin. Osastoyksiköt ja osaston organisaatiokaavio on esiteltyinä kuviossa 1. IAS:n liikevaihto vuonna 2013 oli 13 miljoonaa euroa (Insta Automation Oy 2014).



KUVIO 1. IAS:n organisaatiokaavio (Insta Automation Oy 2014)



## 2.2 Työn teko

Opinnäytetyön tutkimus tehdään IAs:lle, jonka toimipiste sijaitsee Tampereella Sarankulmalla niin sanotulla Insta-kampuksella. Sarankulmalla on Insta Group Oy:n päätuikkokohhta ja samassa osoitteessa sijaitsee myös muiden toimilinjojen päätoimipisteet. Insta DefSec Oy:n suunnitteluosasto kuitenkin muuttaa kampukselle vasta syksyllä 2014. Allekirjoittanut on toiminut harjoittelijana IAs:n palveluksessa loppuvuodesta 2013 asti, ja erilaisia työtehtäviä on kertynyt laajalti ennen tutkimuksen aloittamista. Työkoke-musta on kertynyt IAs:n pajalla, jossa rakennetaan keskuksia ja joka toimii varastona suunnitteluosaston tarvitsemille tarvikkeille huolto- ja testaustöissä. Lisäksi kokemusta on tullut reissuhommista, jotka ovat pitäneet sisällään asennuksia, käyttöönottoa sekä korjaustöitä. Tutkimuksen tekijä on myös tehnyt suunnittelutyötä IAs:n kokonaistoimi-tuksen projektille, jossa toimenkuvana oli laitesuunnittelu.

Tutkimuksen luonteesta johtuen tärkein ja oikeastaan ainoa tarvittava työkalu opinnäy-tetyön tekoon on tietokone oheislaitteineen. Perustutkimus, piirisuunnittelu ja itse kirjoi-tustyö tapahtuvat siis toimistotyönä IAs:n konttorilla. Tutkimuksenteon aikana ja erityi-sesti perustutkimuksen aikana tarvittavia ohjelmistoja on useita. Nämä ohjelmat ovat kauttaaltaan isoja, ja niitä kaikkia ei haluta asentaa saman käyttöjärjestelmäosion päälle. Käytössä on siis useita eri virtuaalikoneita, joissa puhtaan käyttöjärjestelmäasennuksen päälle on asennettu pelkästään kussakin tapauksessa tarvittavat ohjelmistot. Virtuaali-koneita käytetään päällä tarpeen mukaan, ja näin on helppo hallita tutkimustyötä eri ohjelmistojen välillä.

Insta Automation Oy:n kanssa sovittiin tutkimuksen tekoon käytettävästä ajasta, jonka sisällä oli tarkoitus saada valmiiksi perustutkimus ja piirisuunnittelu sekä loppuasiak-kaalle tehtävä käyttöopas valvomo-ohjelmistosta. Tämä aika oli 6 kuukautta ja sen ajan työsuhteen laatu oli tuntityöläinen. Tämä tarkoittaa sitä, että tutkimuksen tekoon käytet-ty työpanos voidaan siirtää tarpeen mukaan muiden projektien pariin. Tutkimusta ei voitu aloittaa heti uuden työsuhteen alussa, johtuen koulutuksista ja keskeneräisistä pro-jekteista. Opinnäytetyön tekoa IAs:n puolesta ohjaa kokonaistoimitukset-osaston alai-suudessa olevan energia- ja prosessiteollisuusosaston ryhmäpäällikkö Rami Siren. Hän toimii projektipäällikkönä projektissa, jonka tarpeesta alun perin opinnäytetyön aihe valittiin ja toimeksi annettiin.

### 2.3 Työn tavoitteet

IAs sai suunnittelutyöksi höyrykattilalaitoksen, jonka tarkoitus on tuottaa, vesilaitoksen tavoin, höyryä. Tässä tapauksessa höyry kuitenkin käytetään naapurilaitoksen prosessissa. Asiakkaan tahdosta höyrykattilalaitoksen valvomoon haluttiin käyttöön nykyisin ranskalaisen Schneider Electricin omistama Vijeo Citect -valvomo-ohjelmisto. Kyseinen valvomo-ohjelmisto ei ole yhtä yleisesti käytetty kuin Siemensin WinCC tai Wonderwaren InTouch. Schneider Electric osti alun perin australialaisen Citect Pty Ltd:n vuonna 2006 ja vuoden 2008 lopulla Citect Pty Ltd lopetti valvomo-ohjelmiston toimitamisen. Yrityskauppojen toteuduttua voidaan olettaa, että Vijeo Citect tulee ottamaan isompaa jalansijaa valvomo-ohjelmistomarkkinoilla tulevaisuudessa. Tämä on myös motiivina opinnäytetyön tutkimuksen tekemiselle.

Aivan kuten suunnittelutyössä kokonaisuuksia ajatellaan piireinä, valvomotyöskentelyssä hallitaan kokonaisuuksia piireinä. Vaikka tekniikka kehittyy hurjaa vauhtia, jokaisessa mittauksessa, moottorissa ja säätimessä perustoimintaperiaate pohjalla on aina samanlainen. Kahden eri laitteen toiminta saattaa olla aivan erilainen, mutta toimintaperiaate on samanlainen. Tällöin laitteita voidaan ja halutaan hallita samankaltaisilla piireillä, ja niiden valvomo-ohjelmistoa rakennettaessa halutaan ja voidaan käyttää samankaltaisia valvomotyypipiirejä. Tätä samankaltaisuutta voidaan hyödyntää valvomoa luotaessa silloin, kun on olemassa valmiit piirit, joita hyödynnetään kaikissa samankaltaisissa kokonaisuuksissa eli piireissä.

Opinnäytetyön päätavoite onkin luoda IAs:lle Vijeo Citect -valvomo-ohjelmistoon valmiit tyyppipiirit, joita voidaan hyödyntää tulevissa projekteissa. Valmiiden tyyppipiirien avulla voidaan varmistaa, että yhtiön kaikissa tulevissa projekteissa, joissa on käytössä Vijeo Citect -valvomo-ohjelmisto, on samanlainen ulkonäkö ja toiminnallisuus. Tutkimuksen aikana tutkittiin myös yleisimmin käytössä olevia valvomo-ohjelmistoja, WinCC:tä ja InTouchia, joissa Insta Automation Oy:llä on jo luotuna valmiit, toistensa kaltaiset tyyppipiirit. Vijeo Citect -tyyppipiirit pyritään luomaan toiminnaltaan ja ulkonäöltään samankaltaisiksi, jolloin asiakkaan vaihtaessa valvomo-ohjelmistoa käyttöympäristö säilyy samanlaisena. Tarkoitus on myös luoda loppukäyttäjälle valvomo-ohjelmiston toiminnasta ja käytöstä käyttöohje, joka tullaan antamaan asiakkaalle aina luovutuksen yhteydessä.

## 3 VALVOMO

### 3.1 Valvomon tehtävä

Nykyaikaiseen tietotekniikkaan pohjautuva automaatio on perusedellytys teollisuuden tuotantoprosessien ja koneiden toiminnalle tuotantolaitoksissa riittävällä tehokkuudella. Automaatio mahdollistaa tuotantolaitosten energiatehokkaan toiminnan, auttaa säästämään prosessissa käytettäviä raaka-aineita ja ehkäisee ympäristön saastumista. Lisäksi automaatio poistaa ihmiseltä työn, joka on fyysisesti kuluttavaa, raskasta ja vaarallista. Automaatio lisääntyy prosessiteollisuudessa ja ylipäättään kaikkialla, missä ihminen voidaan korvata koneilla, jotka hoitavat työn paremmin, turvallisemmin ja tarkemmin. Tästä huolimatta ihmistä ei voida poistaa yhtälöstä kokonaan: Ihminen on välttämätön osa kaikessa prosessitoiminnan ohjauksessa ja suunnittelussa, jotta toiminta olisi kokonaisuudessaan joustavaa (Suomen Automaatioseura Ry 2010, 7).

Ihminen hallitsee tuotantoa, siihen tarvittavia raaka-aineita, prosessin toimintaa ja välttämätöntä kehitystä, joka johtaa parempaan automaatioon. Ihmisellä on myös kyky muuttua muuttuneeseen tilanteeseen ja näin ollen ennakoida tuotantolaitoksen ohjaukseen tarvittavin toimenpitein. Ihmisen ja tekniikan välinen toiminta täytyy olla saumatonta, koska se on tärkeä osa monimutkaisten järjestelmien hallinnassa. Tuotantolaitoksessa tai missä tahansa muussa laitoksessa, jonka toimintaa ohjaa automaatio yhteistyössä ihmisen kanssa, tapahtuu laitoksen hallinta ihmisen osalta valvomossa (Suomen Automaatioseura Ry 2010, 7).

Valvonta toteutetaan joko laitoksen yhteydessä olevassa tilassa tai lisääntyvässä määrin olevista etävalvomoista, joista monitoroidaan, ohjataan ja hallitaan prosessia kokonaisuutena. Yleensä valvomo on ihmiselle mieluisa ja siisti työympäristö, jossa on lukuisia monitoreja, joiden välityksellä tuodaan kentältä tietoa prosessin laitteistolta automaation avulla. Samalla ihminen ohjaa prosessin laitteistoa automaation välityksellä monitoreiden kautta. Valvomon yhteyteen kuuluu usein kokoustila, jossa voidaan suunnitella prosessin ajotapaa. Valvomon sydän on kuitenkin SCADA (Supervisory Control And Data Acquisition) eli ohjelmisto, jonka välityksellä ihminen ja automaatiokoneisto kommunikoivat.

### 3.2 Valvomo-ohjelmisto

Valvomo-ohjelmisto on tietokoneella toimiva graafinen käyttöliittymä, jonka välityksellä ihminen hallitsee prosessin toimintaa. Sen kautta voidaan kerätä historiatietoa laitteiden käynnistä ja rajoista sekä piirtää näytölle trendiä kaikesta kerätystä tiedosta. Sen kautta hallitaan laitekohtaisia asetteluita, ohjauksia ja ajoja. Valvomo-ohjelmistolla prosessi saadaan käyntiin ja ajettua turvallisesti alas, tai vaikka laukaista hätä-seis -piiri, jolla voidaan pysäyttää prosessin ja koko laitoksen toiminta. Valvomon monitoreilla kuvataan siis yksityiskohtaisesti tuotantolaitoksen laitteisto ja sen toiminta. Valvomo-ohjelmistolla voidaan rakentaa laitokohtainen graafinen käyttöliittymä, jolla voidaan esittää laitoksen laitteisto halutulla tarkkuudella. Periaatteessa voidaan rakentaa oma monitorointinäyttö jokaiselle omalle piirille. Voidaan rakentaa myös näyttö, jossa esitetään koko laitoksen toiminta ja laitteisto pienemmällä tarkkuudella.

Suuri osa valvomo-ohjelmistolla saavutettavista hyödyistä perustuu siihen, että ohjelmistot keräävät lakkaamatta tietoa prosessin toiminnasta. Tämä tieto tallennetaan suoraan ohjelmiston toiminnasta tietokantaan, josta sitä voidaan hyödyntää luotaessa trendikuvia ja toimintalokeja. Tietokantoihin voidaan tallentaa esimerkiksi laitetilän muutoksesta tieto, jossa näkyy muutoksen tekijän kirjautumistunnus, aikaleima sekä se, mitä on muutettu. Opinnäytetyössä käytetyssä valvomo-ohjelmassa tehdyistä muutoksista, käynnistyksistä ja ohjauksista tallentuu merkintä tietokantaan eli tiedostoon, jonka tiedostopäätte on CSV. Tämä tiedostomuoto on avattavissa Exceltaulukkolaskentaohjelmalla. Excelin käyttö on muutenkin osana Vijeo Citectillä työskennellessä, sillä lähes kaikkea ohjelmassa käsiteltävää dataa voidaan muokata yksinkertaisesti Excelillä.

Valvomo-ohjelmistoja on monia eri versioita monilta eri valmistajilta. Perustutkimuksen aikana tutustuttiin Schneider Electronicsin Vijeo Citect:n, Siemensin WinCC:n ja Klinkmannin omistamaan InTouch -valvomo-ohjelmistoon. Nämä ohjelmat saadaan näyttämään samankaltaisilta valvomo-operaattorin näkökulmasta, mutta todellisuudessa ohjelmat ovat hyvin erilaisia keskenään ohjelmoinnin ja käyttöliittymän rakentamisen kannalta. Moni valvomo-ohjelmistoa kehittävä yritys liittyy automaatioalaan muullakin tavalla. Esimerkiksi Schneider ja Siemens valmistavat myös logiikoita ja muuta automaatioon liittyvää rautaa ja tekniikkaa.

### 3.2.1 Vijeo Citect

Vijeo Citectin kehitti alun perin vuonna 1973 perustettu australialainen ohjelmistokehitysyritys Citect, joka erikoistui automaatioteollisuuteen. Vuonna 2006 Citectin hankki itselleen ranskalainen sähköalan jätti Schneider Electric. Vuoden 2008 lopulla Citect lakkasi itsenäisen kaupan teon, koska sen jäljellä oleva toiminta siirtyi kaikkineen Schneider Electricin haltuun. Vijeo Citectissä, niin kuin muissakin Citectin ohjelmissa, käytetty ohjelmointikieli on Cicode. Kieli muistuttaa rakenteeltaan Pascal-ohjelmointikieltä. Vijeo Citect tarjoaa laajan ohjelmointirajapinnan, joka sisältää hienostuneita ohjelmointirakenteita, kuten samanaikaiset tehtävät ja viitoitukset (Citect yritysmyynti 2014).

Vijeo Citect sisältää kolme eri työkalua valvomokäyttöliittymän suunnitteluun: Explorerin, Project Editorin ja Graphics Builderin. Explorer-työkalulla pystyy hallitsemaan projektikonaisuuksia ja tarkastelemaan kunkin projektin sisäisiä elementtejä ja asetuksia. Project Editorilla nimen mukaisesti hallitaan yksittäisen projektin ominaisuuksia, kuten esimerkiksi tageja tai ohjelmiston ja logiikan välistä kommunikointia. Graphics Builder on pyhitetty kokonaan käyttöliittymän ulkoasun luontiin ja sillä suunnitellaan kaikki näkymät, joita valvomoikkunassa halutaan näyttää.

Valvomo-ohjelmistoon on asennettavissa lisäosa, joka mahdollistaa projektikohtaisten asetusten hallinnan Excel-taulukkolaskentaohjelmalla. Tämä ominaisuus on osoittautunut korvaamattomaksi lisäominaisuudeksi valvomo-ohjelmistolle projektia luotaessa. Lisäosan asennettua käyttäjä pystyy hallitsemaan muun muassa projektin hälytyksiä, tageja ja paikallisia muuttujia suoraan Excelin kautta. Näin ollen suuria tietomääriä käsiteltäessä työ helpottuu huomattavasti verrattuna siihen, että käyttäjä muokkaa projektin ominaisuuksia pelkästään Project Editorin kautta.

### 3.2.2 Simatic WinCC

Simatic WinCC on Siemensin vuonna 1996 julkaisema SCADA -ohjelmisto, joka on käytössä kaikkialla teollisuuden aloilla. Se on yleisesti käytetty myös Suomessa ja kuuluu ehdottomasti IAS:n työkalujen joukkoon. Niinpä se oli osana myös tätä opinnäyte-

työtä, jossa sitä käytettiin vertailukohtana perustutkimuksen aikana luotaessa tyyppipiirejä Vijeo Citectillä (WinCC 2014).

Toisin kuin Vijeo Citectillä, WinCC:llä suunnittelutyö tapahtuu kokonaan yhdellä työkalulla, WinCC Explorerilla. Sen kautta ohjelmoija pystyy hallitsemaan valvomo-ohjelmiston tagikantaa, graafista ulkonäköä ja toiminnallisuutta. Simatic WinCC:tä käytettäessä alustana toimii Microsoftin Windows käyttöjärjestelmä ja samalla se käyttää Microsoftin SQL-tietokantaa kaiken lokitiedon keräämiseen ja tallentamiseen. Ohjelmointirajapintana toimii valinnaisesti joko VBScript (Visual Basic Scripting Edition) tai ANSI C eli C-ohjelmointikieli.

### **3.2.3 InTouch**

InTouch -valvomo-ohjelmiston on kehittänyt Wonderware, joka esitteli HMI (Human Machine Interface) ohjelmistonsa vuonna 1987. Ohjelmistolla oli vuonna 2011 alansa suurin markkinaosuus, mikä tekee siitä todennäköisesti maailman käytetyin valvomo-ohjelmisto. Huomioitavaa on myös se, että Wonderwaren tuotteita myi yritysostojen kautta vuoden 2014 tammikuusta asti Schneider Electric, joka omistaa myös Vijeo Citectin. InTouchin suuren suosion takana on se, että se oli ensimmäinen ohjelmisto, joka toimi Microsoftin Windows-käyttöjärjestelmän päällä. Ohjelma lukeutuu myös yhteen käytetyimmistä valvomo-ohjelmistoista IAs:lla (InTouch 2014).

InTouchin pääasiallinen suunnittelutyökalu on nimeltään ArcestrA IDE, jolla käyttäjä pystyy luomaan projektikohtaiset mallit tyyppipiireistä. Näille voidaan määritellä muutujat ja graafinen ulkonäkö ja ne ovat helposti monistettavissa ohjelmistossa projektin sisällä. Samoin kuin WinCC, InTouch käyttää SQL-tietokantaa historia- ja tietopalveluiden tallentamiseen ja käyttämiseen. Ohjelmassa on luotuna myös erittäin käytännöllinen simulointityökalu, jolla rakennusvaiheessa pystytään testaamaan piirien toimivuutta niin sanotusti online-tilassa.

### 3.3 Tagit eli muuttujat

Automaatiossa tagi on käsite, jolla tarkoitetaan muuttujaa. Valvomo-ohjelmistojen ominaisuus on kommunikoida kentälle sijoitettujen laitteiden, kuten mittareiden ja venttiilien, kanssa. On mahdollista käyttää avoimia tai suljettuja kommunikointiprotokollia, jotta ohjelmisto saa yhteyden laitteisiin. Laitteilla taas täytyy olla mekanismi, jolla tuoda parametrit valvomo-ohjelmistoon, jotta niitä voidaan kirjoittaa, lukea tai kirjoittaa ja lukea (Tagit 2014).

Protokollasta riippuen laitteelta annettava tai lähetettävä tieto on yleensä osoitteen tai rekisterin muodossa, mutta se voidaan antaa myös muuttujajonona. Näin ollen valvomo-ohjelmiston tulee antaa tiedolle muuttuja tai muistiosoite, jotka voidaan yhdistää suoraan mihin tahansa yksittäiseen kentälaitteen parametriin. Juuri tätä muuttujaa tai muistiosoitetta kutsutaan tagiksi, I/O-tagiksi tai *Variable Tags*, kuten Vijeo Citectissä niitä kutsutaan. Nimitykset tulevat siitä, että tagit ovat valvomo-ohjelmiston tulo- ja lähtökanavia, joiden kautta kentän ja valvomo-ohjelmiston välinen tieto liikkuu (Tagit 2014).

Valvomo-ohjelmisto siis kommunikoi kaikkien ulkoisten laitteiden kanssa niille osoitetun muuttujan eli tagin kautta. Tägeja on myös valvomo-ohjelmiston sisäiseen käyttöön ja niitä kutsutaan paikallisiksi muuttujiksi. Vijeo Citectissä niitä kutsutaan *Local Variables* -nimellä. Sisäisiä muuttujia käytetään ohjelmiston sisällä tapahtuvan tiedon käsitteilyyn ja niiden avulla voidaan ohjelmaan rakentaa sisäisiä toiminnallisuuksia.

Valvomo-ohjelmistot ovat arvokkaita, eikä niitä yleensä hankita kovin pieniä projekteja varten. Ne ovat suuria ja niiden kehitystyön määrä nostaa tuotteen hintaa. Myös markkina-ala nostaa hintaa, koska valvomo-ohjelmistoja ei hankita pieniin prosesseihin. Pääosin valvomo on käytössä voimalaitoksissa tai isoissa tuotantolaitoksissa, joiden budjetti on kooltaan muutenkin niin suuri, että valvomo-ohjelmiston hinta on vain pieni osa laitoksen kokonaisbudjettia. Valvomo-ohjelmiston tagien määrä vaihtelee luonnollisesti siis projektin koon mukaan ja tämä on myös ratkaiseva tekijä valvomo-ohjelmiston hinnassa. Mitä enemmän projektissa on laitteita, ja näin ollen tageja, sitä enemmän lisenssi ohjelmistoon maksaa. Jotkin valvomo-ohjelmistovalmistajat veloittavat lisähintaa jopa sisäisten muuttujien määrän lisääntyessä.

### 3.4 OPC

Object Linking and Embedding (OLE) for Process Control (OPC) on avoin tiedonsiirto-standardi, joka on kehitetty OPC Foundationin toimesta vuonna 1996. Se yksilöi reaaliaikaisen kommunikoinnin eri valmistajien laitteiden välillä. Standardia kehitettiin kourallisen teollisuusautomaatioalan vaikuttajan toimesta. Tekniikka perustuu Microsoftin kehittämiin OLE, Component Object Model (COM) ja Distributed Component Object Model (DCOM) tiedonsiirto- ja tiedonosoitustekniikoihin. OPC-tiedonsiirto-standardi määrittää ohjelmoinnissa standardoidun joukon olioita, käyttöliittymän ja kommunikointitavat automaatioalan prosessinohjaukseen helpottamaan eri valmistajien laitteiden yhteentoimivuutta (OPC 2014).

OPC:stä on olemassa muutama eri luokka, joista yleisimmin käytetty oli Data Access (OPC-DA), jota on käytetty tuomaan tietoa kentältä niin sanotusti suorana valvomoon. Historical data Access:ia (OPC-HDA) käytetään keräämään ja tallentamaan tietoa laitoksen toiminnasta. Minkä tahansa laitoksen ajoon haetaan tietoa historiasta ja historia-tietoja käytetään pohjana, kun tehdään päätöksiä laitoksen toiminnasta. Lisäksi on olemassa Alarms and Events (OPC-A&E) -standardi hälytys- ja tapahtumatietojen kommunikoinnista. OPC-serveri, johon on asennettu OPC-A&E, osaa poimia nämä hälytykset väylästä ja levittää tiedon ohjelmistoille, jotka tukevat OPC:tä. Tämän lisäksi on muutama muu OPC-määrittely. Kaikki edellä mainitut luokat ovat osana OPC Classic-sia (OPC 2014).

Nämä tekniikat pohjautuvat Microsoftin kehittämiin ratkaisuihin. Vuonna 2006 OPC Foundation julkaisi Unified Architecture:n (OPC-UA), jonka tavoitteena oli parantaa ja lisätä OPC-standardin toimivuutta muiden käyttöjärjestelmien kanssa, sekä lisätä standardin turvallisuutta ja käytettävyyttä. OPC-UA vastaa toiminnaltaan OPC Classic-sia ja pystyy suorittamaan paljon enemmän kuin vanhat standardit. Se toimii yleisimpien käyttöjärjestelmien, kuten Microsoft Windowsin, Apple OSX:n, Androidin tai minkä tahansa Linux-version, kanssa ja sitä voidaan ajaa pilvipohjaisen serverin päällä, logiikoilla ja mikrokontrollereilla. Turvallisuuskin on kohentunut OPC-UA:ssa, sillä jokainen viesti lähetetään 128 tai 256 bitin salauksella ja ne myös vastaanotetaan juuri samanlaisena, kuin ne on lähetetty. Lisäksi jokainen OPC-UA toimilaite ja serveri tunnustetaan käyttäen apuna OpenSSL-sertifikaattia, joka määrittää keskenään samassa verkossa kommunikoivat laitteet (OPC 2014).



## 4 VIJEO CITECTIN JA SIEMENS S7:N VÄLINEN KOMMUNIKOINTI

### 4.1 Tarve ja ajurivalinta

Ennen peruspiirien luontia haluttiin tutkia, miten Vijeo Citect -valvomo-ohjelmisto ja Siemensin S7 -logiikka saadaan keskustelemaan keskenään. Höyrykattilalaitokseensa automaation IAs:lta tilanneen asiakkaan kanssa tehdyssä kaupassa oli sovittu, että laitokseen toimitetaan Schneider Electricin valvomo-ohjelmisto ja Siemensin S7 -logiikat. IAs on aikaisemmin toimittanut asiakkaille Vijeo Citect -valvomo-ohjelmiston, kuitenkin sillä poikkeuksella, että samalla on toimitettu saman valmistajan logiikat. Eli aikaisemmissa projekteissa, jossa on ollut käytössä Schneider Electricin valvomo-ohjelmisto, on myös käytetty Schneider Electricin logiikoita.

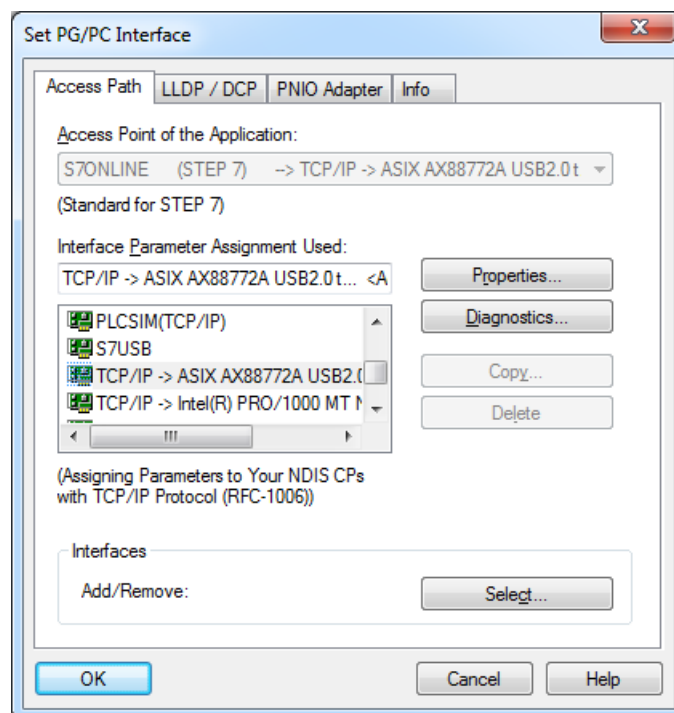
Heti tutkimuksen alkuvaiheessa kävi ilmi, että kun käytössä on kahden eri valmistajan laitteisto ja ohjelmisto, tarvitaan yhteyden muodostamiseksi ohjelmiston lisäksi myös palvelimelle asennettava OPC-serveri, joka mahdollistaa standardoidulla tekniikalla valvomo-ohjelmiston ja logiikan välisen kommunikoinnin. Alustavaa tutkimusta tehtäessä selvisi, että OPC-serverin asennukseen oli molemmilla valmistajilla oma ohjelmisto. PSDirect-ajuri on Schneider Electricin luoma CitectSCADA-ajuri, joka kommunikoi Siemensin S5 - ja S7 -järjestelmien kanssa ilman tarvetta lisätyille Siemensin ohjelmistolle tai laitteistolle. Toinen toimiva vaihtoehto OPC-serverin asennukseen on Siemensin SimaticNet Softnet.

Molemmat kyseiset ohjelmistot ovat erikseen kaupattavia ja lisenssioikeuden vaativia ohjelmistoja, joiden arvo on huomattava. IAs on toimittanut historiansa aikana paljon Siemensin tuotteita ja yritykseltä löytyi jo lisenssi SimaticNet SoftNetin asennukseen, minkä vuoksi oli helppo päättää sen käyttämisestä tutkimuksessa, eikä lähteä hankkimaan lisenssiä Schneider Electriciltä PSDirectin käyttöön. Tutkimuksen alkuvaiheessa koottiin virtuaalikoneet perustutkimuksen tekoa ja tyyppiin luontia varten. Virtuaalikoneen käyttöjärjestelmänä toimi Microsoft Windows 7 Professional 32 bit ja sille asennettiin seuraavat sovellukset: Vijeo Citect 7.4 SP1, Simatic Manager (STEP 7 ver. 5.5) ja SimaticNet SoftNet 8.2.

## 4.2 Siemens-yhteysasettelut

Höyrykattilalaitoksen rakentaja tilasi laitokseen Siemensin S7-400 -sarjan logiikan. Työn tekoon ja testaukseen saatiin käyttöön Siemensin S7-300 -sarjan logiikka, joka vastaa kommunikointiprotokolliltaan asiakkaan logiikkaa ja näin ollen se on pätevä testaamaan yhteysasetuksia. Testauksen ajaksi valittiin kommunikointiväyläksi käyttöön Ethernet-väylä, koska käytössä olleessa S7-300 -logiikan CPU-yksikössä oli Ethernet-väylä. Tietokoneessa oli kiinni USP/Ethernet-adapteri, jonka kautta logiikka saatiin kiinni Ethernet-väylään.

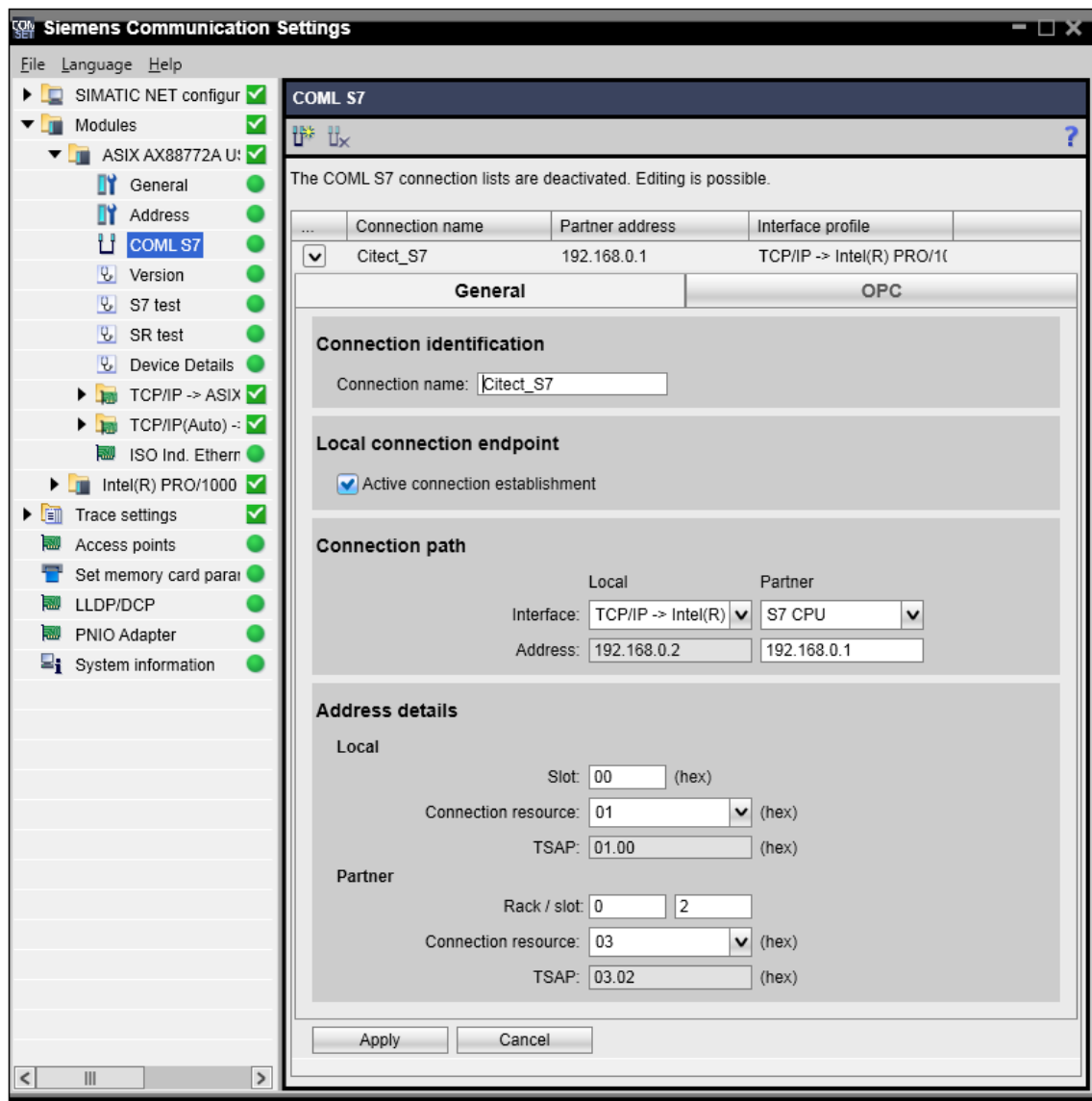
Aluksi oli määriteltävä Simatic Managerin kautta logiikan käyttämä ohjelmointilaitteen ja tietokoneen välinen liityntä, PG/PC Interface, joka määrittää, mitä väylää pitkin tietokone yrittää muodostaa yhteyden logiikkaan. Kun käytössä oli Ethernet-väylä, valittiin liityntäksi listalta TCP/IP. Liityntää valittaessa on huomioitava oikean verkkokortin valinta.



KUVIO 2. Ohjelmointilaitteen ja tietokoneen välisen liityntän valinta

Tämän jälkeen luotiin yhteysasettelut logiikan ja OPC-serverin välille Siemens Communications Settings -työkalulla, joka on osana SimaticNet-asennuspakettia. Modules-valikon alta valitaan käytettävä verkkokortti, ja itse yhteyden asetukset määritellään COM1 S7 -yhteyslistalle. Sinne lisättiin uusi yhteys, jolle luotiin nimi (Citect\_S7), ja

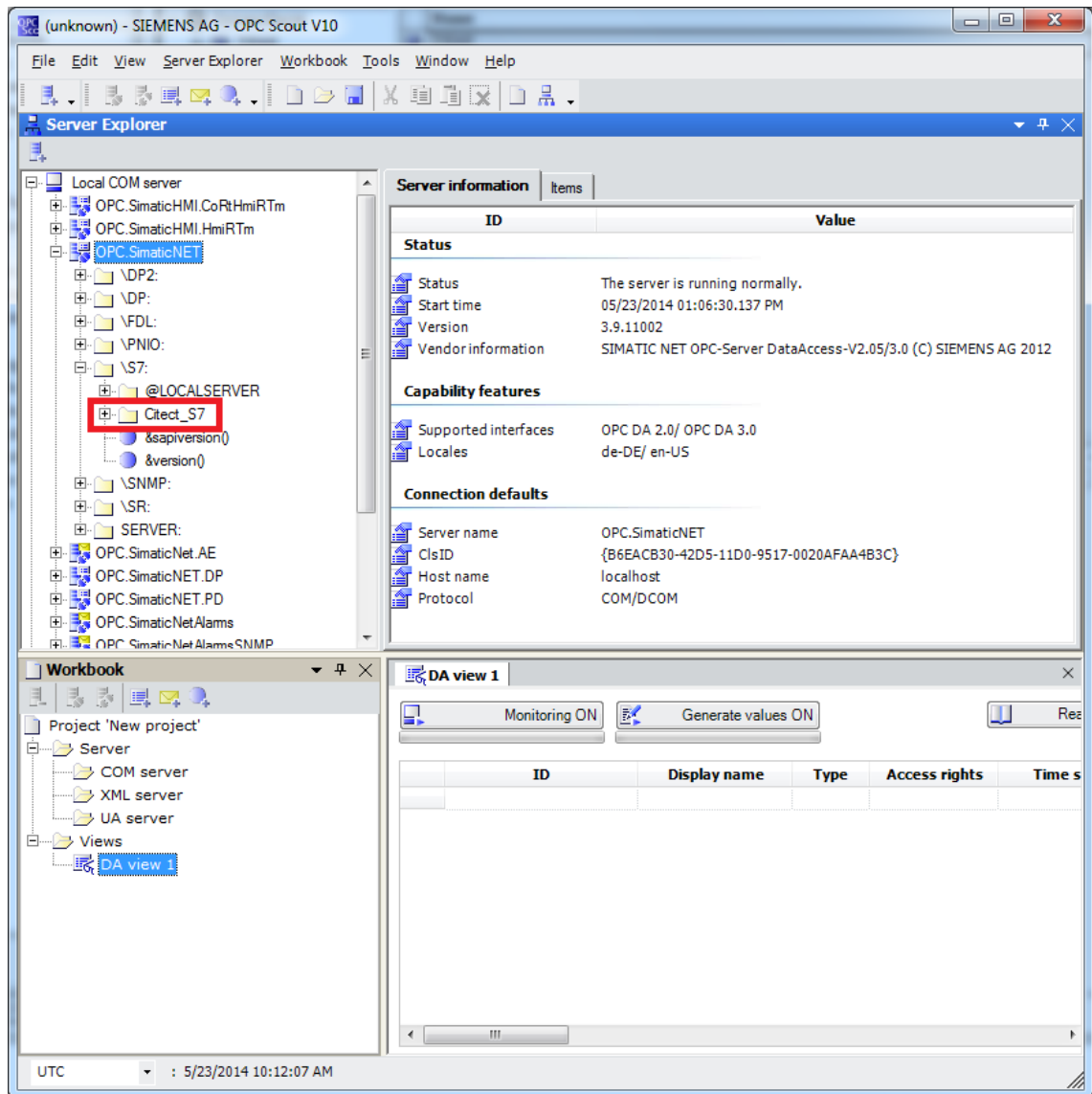
valittiin yhteyspolku, jossa Local toimi valvomokoneen verkkokortin asetteluna ja Partner taas käytettävän logiikan määrittelynä. Address detailssiin määriteltiin laitteiden asemointi, tässä tapauksessa ainut tehtävä muutos oli muuttaa logiikan slot-numero oikeaksi. Siemensin järjestelmissä CPU on yleisesti aina korttipaikassa kaksi ja virtalähde korttipaikassa yksi. Jälleen tulee huomioida oikean verkkokortin valinta, ja että logiikalle asetetaan sama IP-osoite kuin Simatic Managerin kautta.



KUVIO 3. Siemens Communications Settings asetelut

Seuraavaksi lista otettiin käyttöön, mikä tapahtui painamalla hiiren oikealla napilla modules-valikon päältä, ja valitsemalla ponnahdusvalikosta Enable COML S7 connection lists.... Tämän jälkeen logiikan ja OPC-serverin välinen yhteys oli toiminnassa. Yhteyden testaamiseen Siemens on kehittänyt työkalun nimeltään OPC Scout, joka myös kuuluu SimaticNet asennuspakettiin. Tämä työkalu tunnistaa olemassa olevat OPC-serverit

ja sen avulla voi testata luotua yhteyttä. SoftNet-asennuspaketin mukana asennetun OPC-serverin nimi on OPC.SimaticNET ja se löytyy OPC Scoutin avulla. Selattaessa tiedostopuuta OPC Scoutilla alaspäin löydetään juuri luotu yhteys, Citect\_S7.



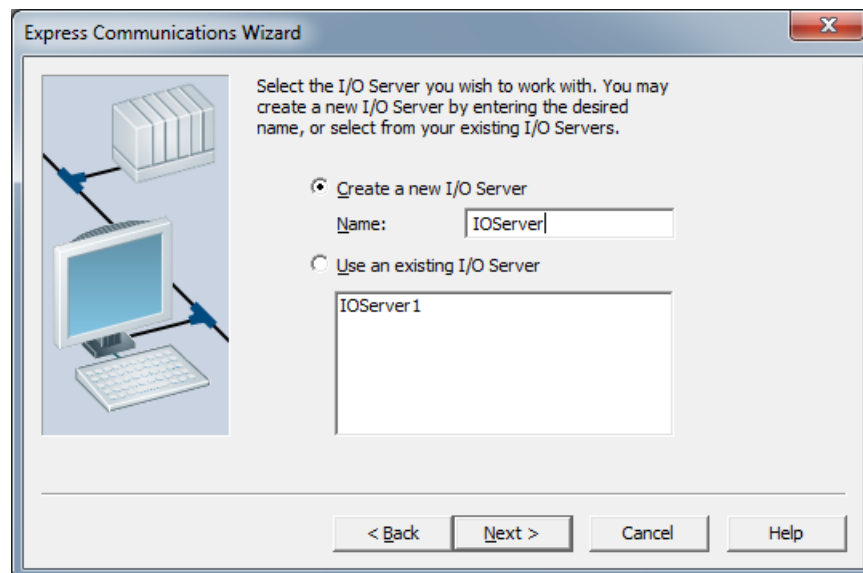
KUVIO 4. OPC Scout

OPC Scout tunnistaa logiikan ja sinne ladatun ohjelman. Yhteyden toimintaa testattiin rakentamalla Simatic Managerilla testiohjelma, jonne luotiin muutama testibitti ja ladattiin ohjelma logiikkaan. OPC Scout löysi luodun Data Blockin (DB), jonka päälle voitiin luoda OPC Scoutissa vastaavat testibittit. Niiden tilaa voi muuttaa OPC Scoutissa ja Simatic Managerin ollessa online-tilassa voi seurata, vastaako bittien arvo Simatic Managerissa, OPC Scoutissa annettua tilaa. Tällä tavalla tutkimuksessa varmistuttiin siitä, että logiikan ja OPC-serverin välinen yhteys todella toimii.

### 4.3 Vijeo Citect -yhteysasettelut

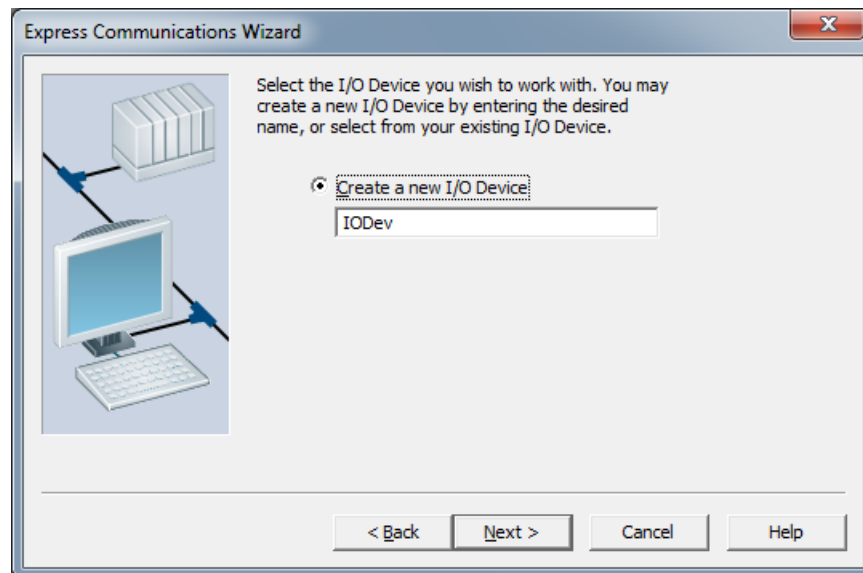
Vijeo Citectissä on kommunikointiyhteyden muodostamiseen tarkoitettu Express Wizard -työkalu, jonka avulla kaikki tarpeelliset asetukset saadaan aseteltua valvomo-ohjelmiston, OPC-serverin ja logiikan välille. Työkalun käyttö helpottaa yhteysasetuksien luontia, mutta se ei ole välttämätöntä, sillä asetukset voi syöttää ohjelmaan niin sanotusti manuaalisesti. Yhteyden testaamista varten luotiin oma projekti Vijeo Citectiin Vijeo Citect Explorerin avulla, sillä yhteysasetukset ovat aina projektikohtaiset ja niitä pääsee muokkaamaan Project Editorin kautta. Project Editorin työkalupalkissa on Communications-valikko, josta löytyy Express Wizard. Se puolestaan on velhotyökalu, jossa määritellään kohta kohdalta yhteysasettelut.

Kun velho käynnistetään, ohjelman ensimmäisessä ikkunassa kerrotaan, että se on tarkoitettu kommunikointiasetusten tekoon. Siitä päästään eteenpäin painamalla Next-painiketta. Seuraavassa ikkunassa luodaan projektin käyttämä I/O-serveri. Valitaan joko jo olemassa oleva aiemmin luotu palvelin, tai vaihtoehtoisesti luodaan kokonaan uusi palvelin.

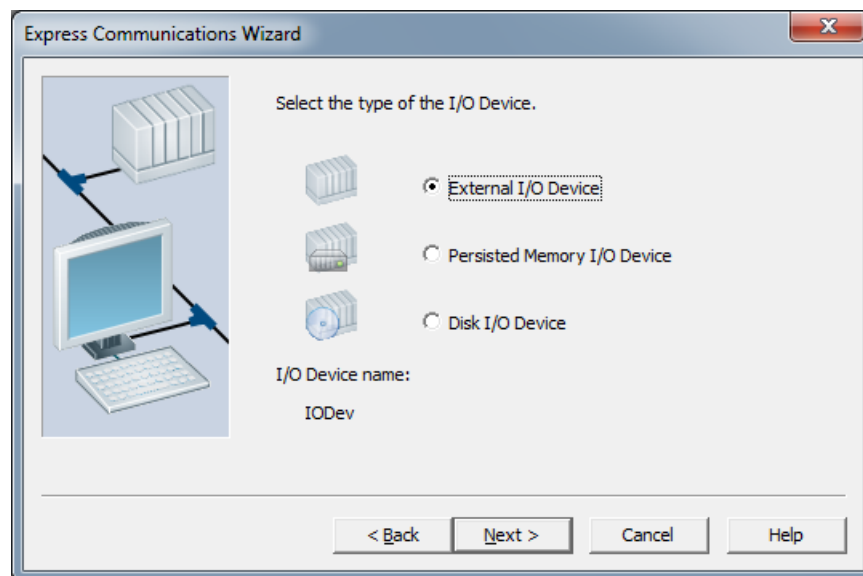


KUVIO 5. I/O-serverin luonti

Seuraavassa ikkunassa annetaan käytettävälle I/O-laitteelle nimi ja tämän jälkeen valitaan sen tyyppi. I/O-laitteella tarkoitetaan käytössä olevaa logiikkaa ja sen tyyppiä valitaan ulkoinen.

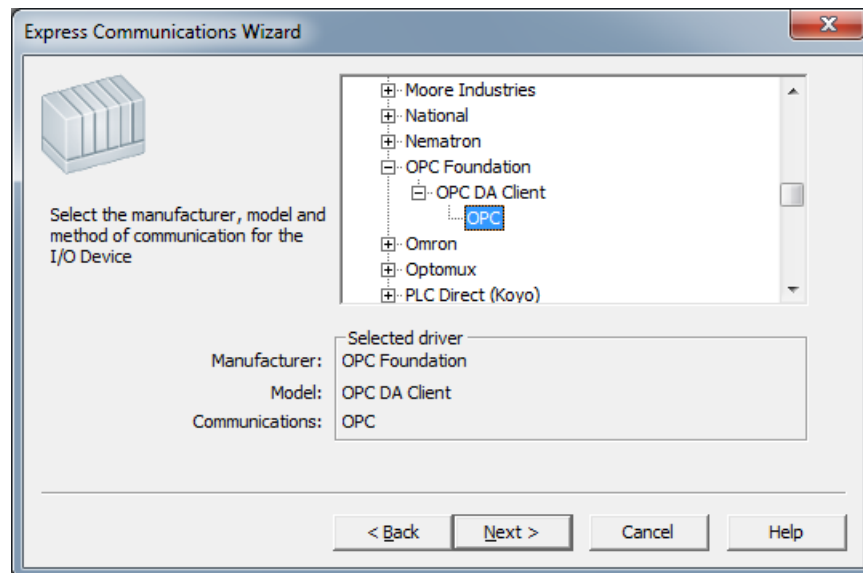


KUVIO 6. I/O-laitteen nimeäminen

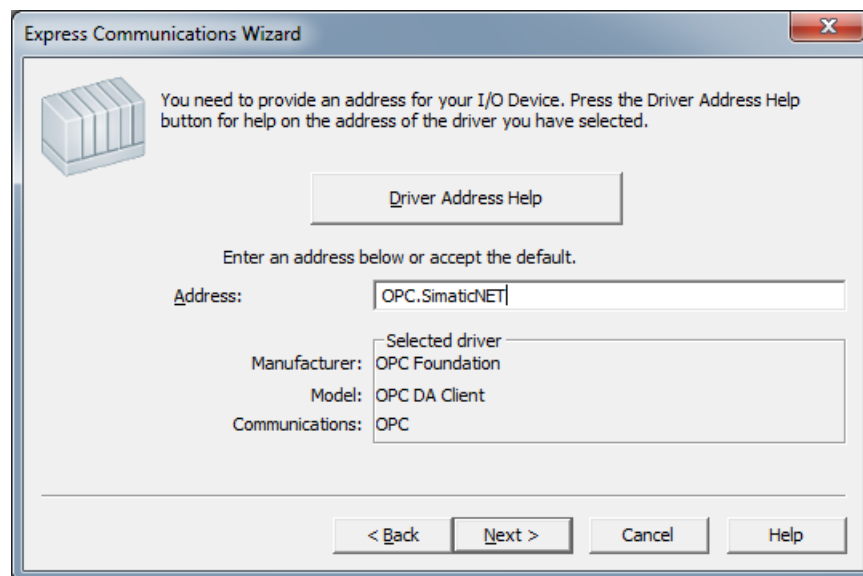


KUVIO 7. I/O-laitteen tyypin valinta

Seuraavaksi valitaan luettelosta käytettävä kommunikointimenetelmä, ja koska käytössä on OPC palvelin, halutaan valvomo-ohjelmiston yhdistävän siihen. Listalta valitaan ensin OPC Foundation, sen jälkeen OPC DA Client ja sitten OPC. Tämän jälkeen annetaan logiikan osoite. Koska logiikka ja valvomo-ohjelmisto kommunikoivat OPC-serverin kautta, osoitteeksi laitetaan OPC -serverin nimi eli OPC.SimaticNET.



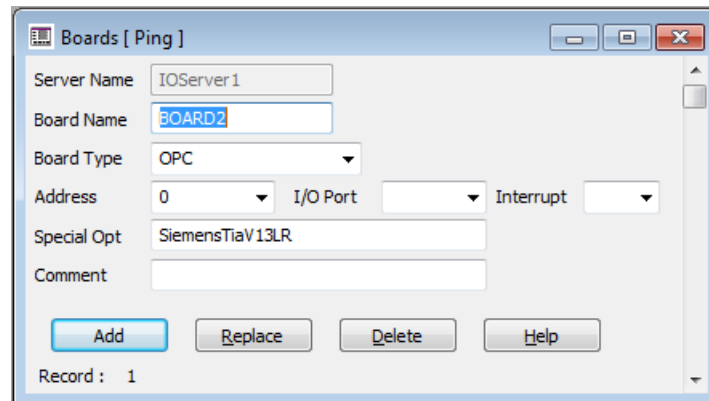
KUVIO 8. Viestintämenetelmän valinta



KUVIO 9. Laitteiston osoite

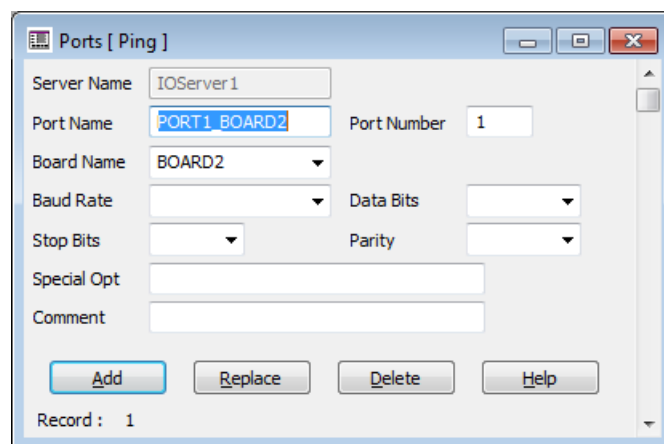
Tämän jälkeen työkalu kysyy käyttäjältä, että halutaanko laitteelle ottaa käyttöön ulkoinen tagikanta. Koska projekti on tarkoitettu yhteystestaukseen, sille ei ole käytettävissä omaa tagikantaa. Seuraavaksi siirrytään uuteen ikkunaan painamalla Next-painiketta. Viimeinen ikkuna on yhteenveto projektiin tehtävistä asetuksista ja työkalu päätetään painamalla Finish-painiketta. Nyt yhteysasetukset ovat valmiit ja yhteys valvomo-ohjelmiston ja logiikan välillä toimii. Tarkastellaan vielä tehtyjä asetuksia valvomo-ohjelmiston kautta. Tehdyt asetukset näkyvät kolmessa eri paikassa, joista niitä voi muokata jälkepäin.

Kaikki tarkasteltavat osiot, joiden kautta yhteysasetuksia päästään muokkaamaan, löytyvät valvomo-ohjelmistosta työkalupalkista Communications-painikkeen alta. Ensimmäisenä tutkitaan valikkoa Boards, jossa määritellään yhteyden muodostamiseen käytettävä alusta, joka voi olla esimerkiksi verkkokortti, sarjaportti tai logiikan viestintäkortti. Tässä tapauksessa se on kuitenkin OPC-serveri, joka on valittuna kohdassa Board Type. Alustalle annetaan myös nimi (BOARD2) ja osoite, joka on nolla, kun käytössä on OPC-serveri. Kohtaan Special Opt on kirjoitettu valvomotietokoneen nimi.



KUVIO 10. Boards -valikko

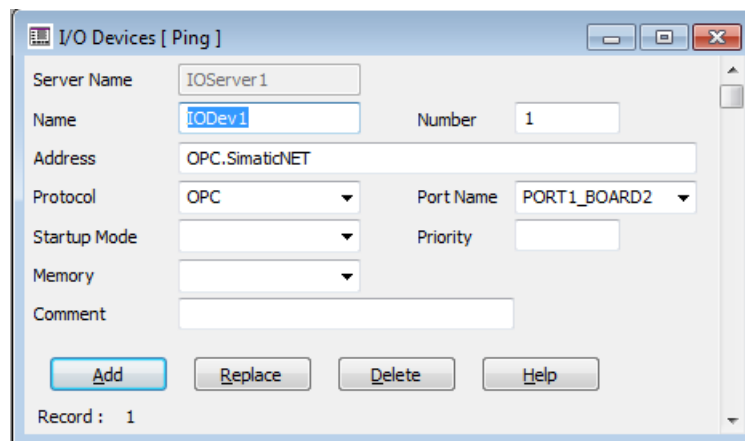
Seuraavaksi tarkastellaan Communications-valikosta löytyvää Ports-valikkoa, jossa määritellään alustan käyttämä portti. Usein käytettävällä alustalla voi olla monta eri porttia ja valikossa määritellään mikä niistä on käytössä. Portille annetaan nimi (PORT1\_BOARD2) ja numero, joka on tässä tapauksessa yksi. Lisäksi määritellään, mitä alustaa eli Boardia se käyttää.



KUVIO 11. Ports -valikko



Viimeinen tarkasteltava valikko, jonne Communications Settings -työkalulla tehdyt asetukset näkyvät, on I/O Devices. Tässä valikossa määritellään laite, jonka kanssa valvomo-ohjelmisto kommunikoi. Laitteelle on määriteltävä nimi ja numero sekä portti, jota se käyttää (PORT1\_BOARD2). Aivan kuten alustaa luotaessa valittiin alustan tyyppi, valitaan nyt laitteen käyttämä protokolla eli OPC. Sitten on vielä osoitekenttä, johon syötetään käytettävän laitteen osoite. Valvomo-ohjelmisto ja logiikka kommunikoivat tässä työssä Siemensin SimaticNetin työkaluasennuspaketin OPC-serverin kautta. Serveri on nimeltään OPC.SimaticNET.



KUVIO 12. I/O Devices -valikko

## 5 TYYPPIPIIRIT

Tyypipiirejä, joita alettiin tutkimusta varten rakentaa, oli yhteensä neljä. Konepiiri on tarkoitettu koneen eli sähkömoottorin ohjaukseen. Teollisuusautomaatiossa sähkömoottori on yleisimmin käytetty voimanlähde. Moottorit antavat energian muun muassa pumpuille, puhaltimille ja kuljetinhihnoille. Mittauspiirin tarkoitus on esittää mitattavaa suuretta ja sen kautta on mahdollista antaa mittaukselle varoitus- ja hälytysrajat. Kolmas tyypipiiri on säädinpiiri, jonka kautta operaattori voi muun muassa asettaa PID -säädön parametrit. Venttiilipiirillä taas ohjataan venttiilin asentoa ja toimintaa automaatiojärjestelmässä.

Rakennettavien piirien tulee vastata valvomo-ohjelmistolliselta toiminnaltaan IAs:n muiden käytössä olevien ohjelmistojen tyypipiirejä, jolloin on mahdollista tarjota asiakkaalle käyttöön kahden eri valmistajan valvomo-ohjelmistoa ja logiikkaa. Insta Automation Oy on laiteriippumaton automaatiojärjestelmien toimittaja, mikä edelleen ajaa tavoitetta tyypipiirien toiminnasta eri ohjelmisto- ja laitevalmistajien tuotteiden välillä.

Toimivan valvomon valvomo-ohjelmistossa kaikkien tyypipiirien tulee toimia siten, että laitoksen prosessia kuvaavissa ikkunoissa näytetään jokainen yksittäinen laite, joka on yhteydessä automaatiojärjestelmään. Tähän ei lasketa käsiventtiilejä ja muita paikalliskäyttöisiä toimilaitteita, joilla ei ole esimerkiksi sähköisiä ohjauksia tai mittatietoja. Pääikkunoissa olevien laiteikoneiden tulee indikoida toimilaitteen pääasiallista toimintaa. Konepiirin ikoni näyttää väreillä, käykö moottori vai onko se seis-tilassa. Venttiilipiirin ikoni osoittaa väreillä venttiilin tilaa eli sitä, onko venttiili auki vai kiinni. Mittauspiirin ikoni osoittaa mitattua arvoa. Samalla se osoittaa väreillä, jos mitattu arvo ylittää asetellut varoitusrajat. Näistä toimilaittekohtaisista ikoneista valvomo-operaattori kykenee hiirenpainalluksella avaamaan laitekohtaisen ponnahdusikkunan, josta hän voi hallita laitteen toimintaa.

Opinnäytetyön pääpaino ja eniten työtä vaativa osuus on siis juuri tyypipiirien ponnahdusikkunoiden suunnittelu ja ohjelmointi niin, että ne vastaavat toiminnaltaan yrityksen muita tyypipiirejä. Vijeo Citectiin tutustumisen jälkeen ensimmäisiä tavoitteita olikin selvittää se toimintatapa, jolla ohjelmassa on toteutettu laitekohtaisten ponnahdusikkunoiden toiminta. Perustutkimusta tehtäessä eli tutustuessa WinCC:llä ja InTouc-hilla tehtyihin piireihin tutkittiin myös sitä, miten tyypipiirien käyttö on toteutettu

edeltä mainituissa ohjelmissa. WinCC:llä ponnahdusikkunoiden avaaminen tapahtuu koodilla. Laitekuvaketta valvomonäytöltä painettaessa ohjelma siis suorittaa scriptin, jossa määritellään C-script- tai Visual Basic -kielellä kirjoitettu koodi oikean toimilaittekohtaisen ponnahdusikkunan avaamiseen. InTouch -valvomo-ohjelmistossa toimilaitteiden ohjauksen toiminta perustuu templateiden käyttöön. Näillä templateilla määritellään tyyppipiirien käyttämät muuttujat, ulkonäkö ja se, miten ne avataan varsinaisesta valvomonäytöstä.

Vijeo Citect -valvomo-ohjelmistoon tutustuminen aloitettiin tekemällä kaksiosainen opetusohjelma, jossa käyttäjä tutustutettiin ohjelmiston yleisimpiin ominaisuuksiin ja toimintaperiaatteisiin. Sen avulla opittiin myös se, miten Vijeo Citectissä on toteutettu tyyppipiirien toiminta, muuttujien luonti ja grafiikkaobjektien luonti. Opetusohjelmassa siis harjoiteltiin Genien ja Super Genien käyttöä. Geniellä tarkoitetaan yleisesti valvomonäytölle sijoitettua laitekohtaista ikonia tai kuvaketta, josta saa avattua varsinaisen tyyppipiirin eli Vijeo Citectin Super Genien.

Vaikka opetusohjelma tarjosi hyvät perusedellytykset valvomo-ohjelmiston käyttöön, oltiin työn alkuvaiheessa yhteydessä Schneider Electricillä työskentelevään Marko Latvasaloon, jonka kautta alun perin saatiin valvomo-ohjelmiston asennuspaketti ja opetusmateriaali. Hän kehotti asentamaan valvomo-ohjelmiston rinnalle lisätyökalun, jonka avulla lähes kaikkia ohjelman ominaisuuksia pystyy hallitsemaan Microsoftin Excel-taulukkolaskentatyökalulla. Työn edetessä pidemmälle huomattiin tämän lisätyökalun käyttö erittäin hyödylliseksi, sillä se teki ulkoisten ja sisäisten muuttujien hallinnasta erittäin käytännöllistä verrattuna Citect Project Editorista löytyviin Variable Tags - ja Local Variables -listaustyökaluihin. Ulkoisia muuttujia käsiteltäessä käyttäjä joutuu usein tekemisiin suurten massojen kanssa. Tällöin samaa tietoa käytetään monessa eri muuttujassa. Excelillä tällaisten suurten massojen hallinta on erittäin yksinkertaista verrattuna siihen, että käyttäjä lisäisi saman tiedon erikseen valvomo-ohjelmiston vakio-työkaluja käyttäen.

## **5.1 Tyyppipiirien ulkoasu**

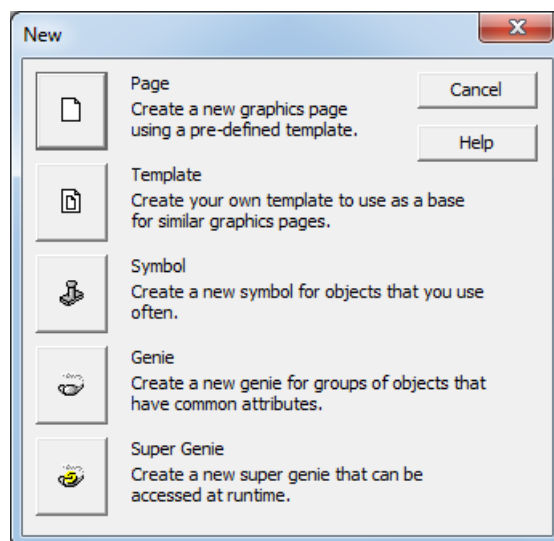
Valvomo-ohjelmistojen toiminta tehdään grafiikkaobjektien ohjelmoinnilla. Tällä määritellään se, mitä tapahtuu, jos nappia painetaan tai se, milloin objektien väreillä ilmais-

taan jotain tapahtunutta. Tyypipiirien teko oli siis helppo aloittaa graafisella suunnittelulla, koska ennen kuin mitään ohjelmointia voi tehdä, täytyy olla ohjelmoitavat objektit. Tyypipiirien toiminnan tapaan myös ulkonäön tulee vastata muiden valvomo-ohjelmistojen tyypipiirejä, mikä helpottaa yrityksen työntekijöiden sekä heidän asiakkaiden lähestymistä Vijeo Citect -valvomo-ohjelmistoon ja sen käyttöön.

Ulkoasun luontiin otettiin mallia yrityksen aikaisemmin tekemistä tyypipiirien ponnahdusikkunoista, joita käytettiin perustutkimuksen tekoon. Näiden mallien pohjalta lähdettiin luomaan Citect Graphics Builderilla Super Genieitä tyypipiireille. Nämä tyypipiirit ovat kone, mittaus, säädin ja venttiili. Kaikkien valmiiden tyypipiirien kuvakaappaukset löytyvät liitteistä 1-4.

### 5.1.1 Citect Graphics Builder -työkalut

Citect Graphics Builderilla käyttäjä voi luoda kaikki graafiset elementit ja niiden ulkonäön. Luodut graafiset elementit päätyvät varsinaisen valvomon näytöille. Opinnäytetyössä Citect Graphics Builder oli käytetyin työkalu. Sillä luodaan kaikki uudet valvomonäytöt eli sivut. Sillä luodaan myös Geniet ja Super Geniet, joita opinnäytetyön tyypipiirit ovat. Graphics Builderilla käyttäjä voi myös tehdä itselleen projektiinsa valmiita pohjia eli templateja sekä symboleja, joita voi käyttää grafiikkaobjektien sisällä. Kaikki varsinaisen valvomon näytölle päätyvät näkyvät objektit ovat jotain näistä tyypeistä.



KUVIO 13. Uuden grafiikkaobjektin luonti

Piirtämiseen on suunnittelijalla käytössä hyvin yksinkertaiset piirtotyökalut Graphics Builderin sisällä. Se sisältää kuitenkin kaikki tarpeelliset piirtotoiminnot kuten viivan, suorakulmion ja ympyrän. Periaate piirtämisessä on se, että piirrettyä objektia muokataan varsinaisen piirtämisen jälkeen, jolloin käyttäjällä on mahdollisuus muuttaa kappaleen kokoa, sen viivan paksuutta ja väriä sekä yhdistää piirrettyjä objekteja toisiinsa. Työkalupalkista löytyy myös toiminnot seuraavien objektien lisäykseen:

- Tekstikenttä
- Numerokenttä
- Nappi
- Animoitu symboli
- Trendikehys
- CiCode -objekti
- Putki
- Symboli
- Genie
- Prosessianalyysi
- Tietokannan hallinta
- Ajastin
- ActiveX komponentti
- Web -selain



KUVIO 14. Citect Graphics Builder -työkalupalkki

### 5.1.2 Värivalinta

Opinnäytetyön ohjaajan kanssa sovittiin tyyppipiirejä koskevista asioista, joita ei ole lukittu tarkasti vanhoihin eli muissa valvomo-ohjelmistoissa käytettyihin tyyppipiirien malleihin. Yksi näistä asioista oli ponnahdusikkunoiden väritys, jonka tekoon ja valintaan saatiin tässä tutkimuksessa vapaat kädet. Tyyppipiireissä käytettävien värien tuli kuitenkin noudattaa yrityksen laatimaa väripalettia.

Yritykselle omat värit ovat iso osa brändiä ja identiteettiä. Myös Insta on vakiinnuttanut värivalikoiman, jota tulee noudattaa heidän tuotteissaan. Tämä paletti pitää sisällään muutaman tummansinisen sävyn, muutaman turkoosin ja yhden sävyn vaaleanharmaata. Johtoajatuksena ponnahdusikkunoiden väritystä laadittaessa oli asetettava valvomo-operaattorin saappaisiin. Tavoitteena oli luoda väriyhdistelmä, josta laitteen toimintaa vastaavat objektit pysyvät tyyppipiireissä selkeinä ja helposti erottuvina. Erityisesti keskityttiin myös siihen, että tyyppipiirejä jaksaa tuijottaa, eikä sävytys käy silmälle rasakaaksi.

Tyyppipiirien graafinen ulkoasu on jokaisessa tapauksessa sellainen, että pohjalla on kaksi eri väriä. Parhaan väriyhdistelmän löytämiseksi luotiin kopioita tehdystä kone-tyyppipiiristä. Näihin kopioihin vaihdettiin eri taustavärit ja väriyhdistelmä Instan väripaletista. Eri väri vaihtoehtojen asettamisen jälkeen asetettiin tyyppipiirikopiot vierekkäin, jolloin niitä oli helppo verrata keskenään, ja löytää paras väriyhdistelmä käytettäviiin tyyppipiireihin. Väriyhdistelmä, johon päädyttiin, piti sisällään pohjavärinä vaaleanharmaan ja laatikoissa vaalean turkoosin. Molempien värien päältä on helposti erottavissa kaikki teksti- ja numerokentät sekä muut tyyppipiireissä esiintyvät symbolit.



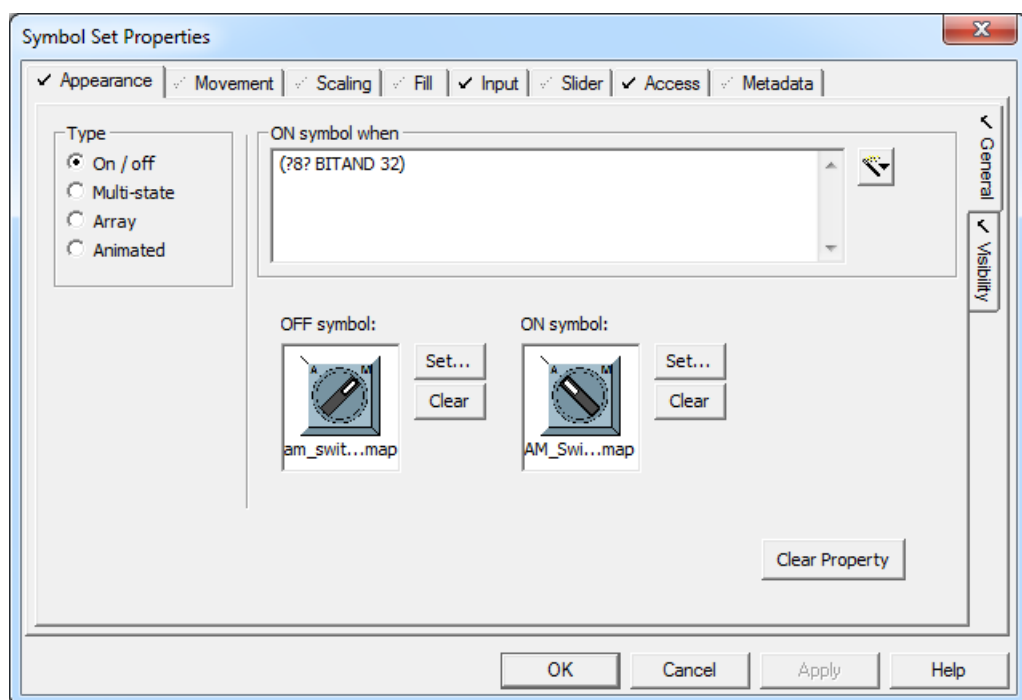
KUVIO 15. Instan väripaletti

### 5.1.3 Omat symbolit

Citect Graphics Builder pitää sisällään kattavat symbolikirjastot, josta suunnittelija voi lisätä projektiinsa merkkivaloja, linjastoja ja esimerkiksi venttiilejä. Aivan kaikkia tyyppiin tarvittavia symboleja ei kirjastosta kuitenkaan löydy, joten muutama niistä jouduttiin luomaan itse. Kone-, venttiili- ja säädintyyppiin on käytössä A/M-kytkin, jolla valitaan se, toimiiko laitteen ohjaus automaattisesti vai ohjataan sitä käsin valvomosta. Lisäksi säädinpiirissä on käytössä niin sanotut radionapit, joilla valitaan käytettävä asetusarvo; sisäinen tai ulkoinen.

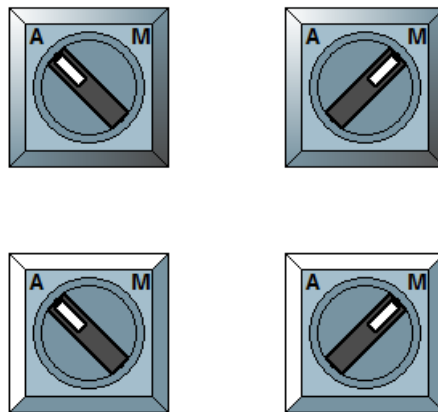
Molemmat painonapit on toteutettu siten, että niissä on kaksi eri kuvaa kuvaamaan molempia mahdollisia asentoja. Kytkimessä on siis kaksi erillistä kuvaa: Toisessa kytkin on suunnattuna A-kirjainta ja toisessa M-kirjainta kohti. Kirjaimet kuvaavat automaattia ja manuaalia. Radionapilla on myös kaksi mahdollista tilaa, joista toisessa tapauksessa nappi on valittuna ja toisessa taas ei. Kun nappi on aktiivisena, ympyrän keskellä valintaa indikoi musta piste. Kun nappi ei ole aktiivisena, ympyrä on tyhjä.

Tyyppiin symbolit on otettu käyttöön animoituina symboleina, joilla on kaksi eri tilaa ja ne vaihtavat tilaansa ehdon mukaan. Kuvio 16 on animoitujen symbolien yleisistä asetuksista konetyyppiin.



KUVIO 16. Animoidun symbolin asetukset

Käytettäessä animoituja symboleja, joilla on muuttuvat tilat, Citect Graphics Builder vaatii, että symbolit ovat bitmap-muodossa. Tämä tarkoittaa sitä, ettei symboleissa voida käyttää niin sanottua liukuvaa väritystä. Alkuperäisessä A/M-kytkin -symbolissa käytettiin liukuvaa väritystä napin reunoilla, sillä tämä efekti korosti napin asemaa. Ohjelma ei kuitenkaan hyväksynyt näitä alkuperäisiä symboleja käytettäväksi animoidussa symbolissa ja ne jouduttiin muuttamaan bitmap-muotoon. Tämä onnistuu helposti Graphics Builderin Tools valikosta löytyvällä *Convert to Bitmap* -toiminnolla. Kuviossa 17 on kuvat alkuperäisistä ja bitmap-muotoon muunnetuista kytkinsymboleista. Kuviossa 18 on kuvat suunnitelluista radionappisymboleista.



KUVIO 17. A/M-kytkinsymbolit



KUVIO 18. Radionapit

#### 5.1.4 Muut tyyppiäireissä käytetyt symbolit, arvokentät ja painonapit

Tyyppiäirejä luotaessa hyödynnettiin myös ohjelmiston valmiita symboleja. Citect Graphics Builderin symbolikirjastosta löytyy oikein hyvä valikoima merkkilamppuja, joista valittiin käytettäväksi neliskanttiset merkkivalot. Näitä merkkivaloja käytetään näyttämään laitekohtaista toimintaa tyyppiäireissä. Valot osoittavat esimerkiksi tietoa vioista, turvakytkimen tilasta, ohjausristiriidasta, huollon tarpeesta ja laitteen käyntitiedosta. Jokaisella merkkivalolla on aktiivisena oma, tilaa kuvaava väritys. Käyntitietoa osoittavat valot ovat aktiivisena vihreät ja vikaa indikoivat valot aktiivisena taas punaiset. Kaikki merkkivalot ovat harmaita silloin, kun ne eivät ole aktiivisena.

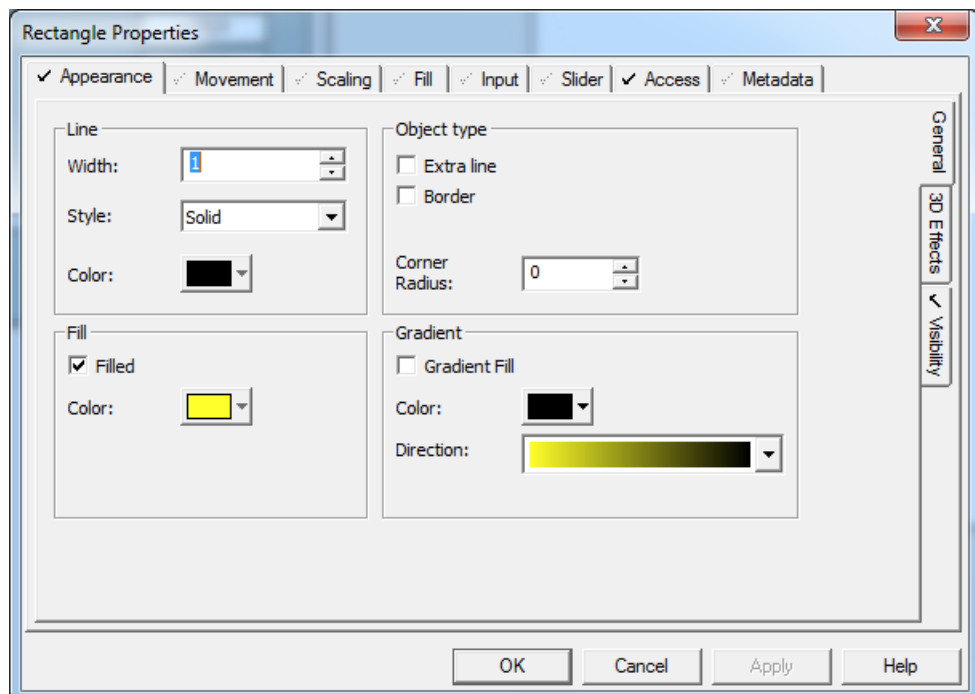




KUVIO 19. Merkkivalojen värit

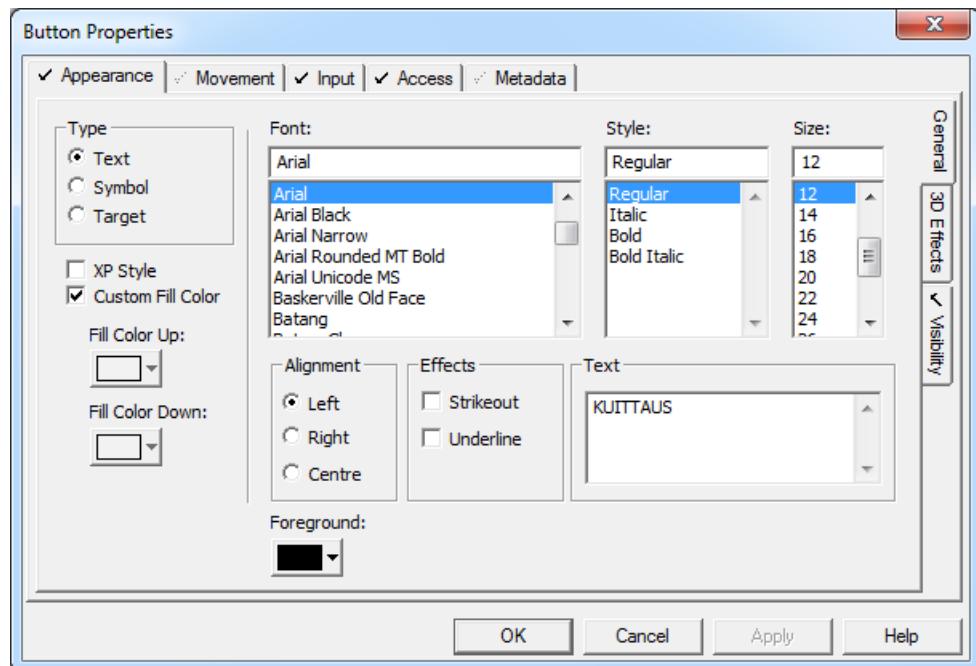
Kone- ja venttiilityyppiin Super Genieissä on molemmissa vasemmassa ylälaudassa tyyppiä kuvaava ikoni. Konepiirin ylälaudassa on sähkömoottorin kuva ja venttiilityyppiin ylälaudassa on kuva käsiventtiilistä. Myös nämä kuvakkeet on otettu käyttöön piirto-ohjelman symbolikirjastosta.

Jokaisessa tyyppiin on keltaisella, valkoisella tai sinisellä pohjalla värityttyä arvokenttiä, jotka osoittavat kussakin tapauksessa tekstillä kerrottua arvoa. Valkoisella pohjalla oleva arvokenttä osoittaa, että lukuarvo luetaan toimilaitteelta. Keltaisella pohjalla olevaan kenttään käyttäjä voi syöttää ohjausarvon, esimerkiksi taajuusmuuttajalla varustetun koneen pyörimisnopeuden. Sininen väri arvokentän pohjalla taas indikoi sitä, että arvo tulee logiikalta. Nämä arvokentät on tehty siten, että pohja on piirretty käyttämällä Graphics Builderin suorakulmiopiirtotyökalua, joka on valittu täytettäväksi kussakin tapauksessa vaadittavalla värillä. Näiden suorakulmioiden päälle on sen jälkeen lisätty numerokenttä.



KUVIO 20. Suorakulmion yleiset asetukset

Suurin osa napeista on toteutettu käyttämällä ohjelman napinpiirtotyökalua, jolla käyttäjä voi määrittää napin koon, sen päällä näkyvän tekstin, kulmamuodon ja napin ikonissa näkyvät efektit, esimerkiksi napin varjostukset. Venttiilipiiriä varten jouduttiin tekemään napit venttiilin ohjaukseen, koska näiden tulee indikoida logiikalta tulevaa auki- ja kiinniohjaus-signaalia väreillä. Napit jouduttiin piirtämään, koska napinpiirtotyökallulla tehdyillä napeilla ei ole ominaisuutta, jolla napin väriä voisi muuttaa toiminnan mukaan.



KUVIO 21. Painonapin ominaisuudet

## 5.2 Tyypipiirikohtaiset muuttujat ja niiden hallinta

Super Genieiden ulkoasun luonnin jälkeen seuraava iso opinnäytetyön osakokonaisuus, johon keskityttiin, oli tyypipiireissä käytettyjen muuttujien tutkinta. WinCC:llä ja InTouchilla IAs:n luomia tyypipiirejä tutkittiin ja kerättiin näistä käytetyt muuttujat. Eri-tyisesti kiinnosti muuttujien nimeäminen, koska se kertoo paljon muuttujan käyttötarkoituksesta. Tutkinnan ja työn edistyessä pääpaino sille, kumman valvomo-ohjelmiston tyypipiireistä enemmän esimerkkiä otettiin, kallistuttiin koko ajan WinCC:n suuntaan. Vijeo Citect on tarkastelujen jälkeen toiminnaltaan lähempänä WinCC:tä, kuin InTouch-valvomo-ohjelmistoa. Esimerkkipainon kallistumiseen WinCC:n suuntaan vaikutti varmasti myös se, että opinnäytetyötä tehtäessä käytössä oli Siemensin S7-300 -sarjan logiikka, jonka avulla testattiin tehtyjen tyypipiirien toimintaa.

Ulkoisia muuttujia hallitaan Citect Project Editorista löytyvällä Variable Tags -luettelolla. Sen avulla annetaan muuttujalle nimi, muuttujan koon mukaan määräytyvä tyyppi, osoite, jolla kerrontaan muuttujan käyttämä osoite logiikassa sekä käytettävä I/O-laite eli logiikka. Muuttujien tyyppikohtaiset ominaisuudet on esitelty tarkemmin kuviossa 24. Muuttujalle voidaan antaa sen toimintaa kuvaava kommentti. Luettelossa muuttuja voidaan osoittaa suoraan tietylle laitteelle, joka on luotu käyttämällä Vijeo Citectin Equipment Editoria. Tätä kenttää ei ole kuitenkaan missään käytetyissä muuttujissa täytetty, sillä tyyppiin luonti ja opinnäytetyön teko ei edellytä laitekohtaisia linkityksiä. Nämä voidaan ottaa käyttöön tulevilla projekteilla, joissa on käytössä luodut tyyppiin.

KUVIO 22. Ulkoiset muuttujat luettelo

Sisäisiä muuttujia käyttäjä hallitsee myös Project Editorilla, josta löytyy Local Variables -luettelo. Sen avulla käyttäjä voi antaa sisäiselle muuttujalle nimen, tyyppin, käytettävän yksikön ja sen muodon sekä kommentin. Tämän lisäksi käyttäjä voi antaa muuttujan arvolle käytettävän skaalan.

KUVIO 23. Sisäiset muuttujat luettelo

Muuttujia, niin sisäisiä kuin ulkoisiakin, pystyy siis hallitsemaan myös Excelillä. Tämä on huomattavasti tehokkaampi tapa hallita kuin edellä esitettyjen listojen avulla hallitseminen. Lisäksi Exceliä käytettäessä ulkoisten muuttujien muokkaamiseen käyttäjä voi asettaa muuttujalle skaalauksen, jonka avulla voidaan valvomossa esittää logiikalta tullut arvo eri skaalalla. Eli jos logiikalta tuleva arvo on välillä 0—4000, se voidaan muuttaa valvomoon muotoon 0—100.

Data Type	Variable	Size	Allowed Values
BCD	Binary- Coded Decimal	2 bytes	0 to 9,999
BYTE	Byte	1 byte	0 to 255
DIGITAL	Digital	1 bit or 1 byte	0 or 1
INT	Integer	2 bytes	-32,768 to 32,767
UINT	Unsigned Integer	2 bytes	0 to 65,535
LONG	Long Integer	4 bytes	-2,147,483,648 to 2,147,483,647
ULONG	Unsigned Long Integer (Only for display on a screen.Arithmetic operations are not supported. )	4 bytes	0 to 4,294,967,295
LONGBCD	Long Binary- Coded Decimal	4 bytes	0 to 99,999,999
REAL	Floating Point	4 bytes	-3.4E38 to 3.4E38
STRING	String	256 bytes (maximum)	ASCII (null terminated)

KUVIO 24. Muuttujien tyypikohtaiset ominaisuudet

### 5.2.1 Ulkoiset muuttujat

Ulkoisten muuttujien nimeämiskäytännön malli otettiin IAs:n WinCC:llä aikaisemmin luotujen tyyppipiirien mukaan. Aivan kuten Vijeo Citectissä, WinCC:ssä on käytössä ulkoiset ja sisäiset muuttujat. Nämä löytyvät selkeästi toisistaan eroteltuina projektikohtaisesta ohjelmasta, joka oli mallina tyyppipiirejä varten. Vijeo Citectin ja WinCC -valvomo-ohjelmistojen välillä on toki toiminnallisia eroja sekä eroavaisuuksia siinä, miten muuttujia voidaan käsitellä ohjelmoinnilla. Tämä ero osoittautuu Vijeo Citectin hyväksi sillä tavalla, että sillä voidaan kirjoittaa ja lukea tiettyä bittiä sanan tai tuplasanan mittaisesta muuttujasta. Vijeo Citectissä voidaan siis yhdistää kaikki muuttujat, jotka ovat tyyppiltään DIGITAL eli 1 bitti, yhdeksi sanan mittaiseksi muuttujaksi. Näin tehtäessä voidaan säästää runsaasti muuttujien lukumäärässä. Esimerkiksi tehdyissä tyyppipiireissä tätä on käytetty hyväksi siten, että kaikki DIGITAL-tyyppiset muuttujat

on yhdistetty yhdeksi sanan mittaiseksi muuttujaksi, jota kutsutaan nimellä INOUT\_BITIT. Näin voidaan säästää jopa viidentoista muuttujan käyttö yhdistämällä ne yhdeksi erityyppiseksi muuttujaksi.

Vijeo Citectin ulkoisten muuttujien käytöstä löytyy myös muita huomattavia etuja, sillä käyttäjä voi liittää Excelin avulla muuttujaan sen käyttämän yksikön ja asteikon, jonka sisällä esimerkiksi liukuluku esitetään REAL-tyyppisissä muuttujissa. Näitä kahta ominaisuutta voidaan pitää merkittävänä siksi, että niiden käyttö vähentää tarvittavien sisäisten muuttujien määrää luoduissa tyyppiireissä. Lisäksi ne vähentävät käytettävien graafisten objektien määrää tyyppiirien Super Genieissä vähentäen samalla objektien tarvitsemaa tilaa. Eli koko yksikön kulkiessa niin sanotusti muuttujan mukana, sille ei tarvitse luoda omaa, muuttujasta erillistä tekstiobjektia. Taulukossa 1 on esitettyä tyyppiireissä käytettyjen muuttujien nimet, tyypit ja lopulliset kommentit. Siinä on esitettyä jokaisen eri yksittäisen tyyppiirin tarvitsemat ulkoiset muuttujat eli yhden toimilaitteen tarvitsemat muuttujat jokaiselle eri tyyppiirille.

Laitekohtaisesti muuttujat erotellaan muuttujan nimen alkuosalla. Taulukossa 1 kone-tyyppiiriin muuttujilla on nimen alkuosana K01, mittaustyyppiiriin muuttujilla M01, säädintyyppiiriin muuttujilla S01 ja venttiilityyppiiriin muuttujilla V01. Luodussa testiympäristössä oli käytössä useampi tyyppiiri jokaista eri tyyppiiriä ja siinä käytetään toisen konetyyppiiriin muuttujien nimen alkuosassa tunnusta K02. Kun tyyppiirejä käytetään oikeassa laitosympäristössä, korvataan luodut tunnukset laitteiden laitepositioilla. Kommenttikenttään on myös merkattu vakioyksiköt niille muuttujille, joiden yksikköä ei käyttäjän tule muuttaa.

TAULUKKO 1. Ulkoiset muuttujat

NAME	TYPE	COMMENT
K01_IN_VALVONTAVIIVE	INT	Kone: Valvontaviive (vakioyksikkö s)
K01_OUT_VRK_O	INT	Kone: Käyntiaika vrk. (vakioyksikkö min)
K01_OUT_VRK_K	INT	Kone: Käynnistyskerrat vrk. (vakioyksikkö kpl)
K01_OUT_OLO	REAL	Kone: Oloarvo
K01_OUT_MITT2	REAL	Kone: Mittaus 2 (vakioyksikkö A)
K01_OUT_MITT3	REAL	Kone: Mittaus 3 (vakioyksikkö %)
K01_INOUT_KASIOHJE	REAL	Kone: Käsihje
K01_INOUT_BITIT	UINT	Kone: Valvomobitit
K01_LUKITUS_BITIT1	UINT	Kone: Lukitusbitit 1
K01_LUKITUS_BITIT2	UINT	Kone: Lukitusbitit 2
K01_INOUT_BITIT2	UINT	Kone: Valvomobitit 2
M01_IN_ARH_RAJA	REAL	Mittaus: Alaraja hälytys

M01_IN_YRH_RAJA	REAL	Mittaus: Yläraja hälytys
M01_IN_ARV_RAJA	REAL	Mittaus: Alaraja varoitus
M01_IN_YRV_RAJA	REAL	Mittaus: Yläraja varoitus
M01_IN_HALYHYST	REAL	Mittaus: Hälytyshystereesi
M01_IN_HALYVIIVE	INT	Mittaus: Hälytysviive (vakioyksikkö s)
M01_IN_SIM_PV	REAL	Mittaus: Syötettävä simulointiarvo
M01_IN_SIM_TO	INT	Mittaus: Syötettävä simulointiaika (vakioyksikkö min)
M01_OUT_SIM_OLO	INT	Mittaus: Kulunut simulointiaika (vakioyksikkö min)
M01_OUT_PV_TOD	REAL	Mittaus: Todellinen mittausarvo (simulointi ei vaikuta)
M01_OUT_PV_OUT	REAL	Mittaus: Mittausarvo
M01_INOUT_BITIT	UINT	Mittaus: Valvomobitit
S01_IN_MIN_OHJE	REAL	Säädin: Minimiohjaus (vakioyksikkö %)
S01_IN_MAX_OHJE	REAL	Säädin: Maksimiohjaus (vakioyksikkö %)
S01_IN_P	REAL	Säädin: Vahvistus P
S01_IN_I	REAL	Säädin: Integrointiaika (vakioyksikkö s)
S01_IN_D	REAL	Säädin: Derivointiaika (vakioyksikkö s)
S01_IN_HALYVIIVE	INT	Säädin: Hälytysviive (vakioyksikkö s)
S01_IN_TOLERANSSI	REAL	Säädin: Säädetoleranssi
S01_OUT_PV_OUT	REAL	Säädin: Mittausarvo
S01_OUT_OHJE	REAL	Säädin: Ohjausarvo (vakioyksikkö %)
S01_INOUT_SP	REAL	Säädin: Sisäinen asetusarvo
S01_INOUT_SP_RAMP_VAL	REAL	Säädin: Rampitusnopeus (vakioyksikkö u/s)
S01_OUT_SP_ULK	REAL	Säädin: Ulkoinen asetusarvo
S01_OUT_SP_KAYTOSSA	REAL	Säädin: Käytössä oleva asetusarvo
S01_OUT_MYOTAKYTKENTA	REAL	Säädin: Myötäkytkennän vaikutus (vakioyksikkö %)
S01_INOUT_BITIT	UINT	Säädin: Valvomobitit
V01_IN_VALVONTAVIIVE	INT	Venttiili: Valvontaviive (vakioyksikkö s)
V01_IN_KASIOHJE	REAL	Venttiili: Käsihje (vakioyksikkö %)
V01_OUT_VRK_A	INT	Venttiili: Aukiohujauskerrat vrk (vakioyksikkö kpl)
V01_OUT_VRK_K	INT	Venttiili: Kiinniohjauskerrat vrk (vakioyksikkö kpl)
V01_OUT_ASENTO	REAL	Venttiili: Asento (vakioyksikkö %)
V01_INOUT_BITIT	UINT	Venttiili: Valvomobitit
V01_LUKITUS_BITIT1	UINT	Venttiili: Lukitusbitit 1
V01_LUKITUS_BITIT2	UINT	Venttiili: Lukitusbitit 2

### 5.2.2 INOUT\_BITIT

Jokaiselle tyyppiirityypille löytyy ulkoinen muuttuja, joka on nimetty INOUT\_BITIT-nimellä. Muuttuja on tyypiltään *Unsigned Integer* eli se on kahden tavun mittainen, etumerkitön muuttuja. Se on tarkoitettu käytettäväksi siirtämään eli luettavaksi ja kirjoitettavaksi kooltaan yhden bitin arvoisia tietoja valvomo-ohjelmiston ja logiikan välillä. Tämän muuttujan jokaista kuuttatoista bittiä voidaan siis kirjoittaa ja lukea erikseen.

Vijeo Citectissä ulkoisen muuttujan eli INOUT\_BITIT:n lukeminen ja kirjoittaminen tapahtuvat osoittamalla muuttujan tiettyyn bittiin. Tätä muuttujaa käytettäessä ja tyyppi-

piiriin oikean toiminnan takaamiseksi täytyi selvittää, mitä ulkoista muuttujaa mikäkin INOUT\_BITIT-muuttujan bitti vastaa. Tyypipiirien toiminta pohjautuu logiikkaohjelmassa luotuun ohjelmakoodiin. Siemensillä, jonka ohjelmaa käytettiin tyypipiirien testaamiseen, se puolestaan pohjautuu SCL -kielellä kirjoitettuun Source-koodiin. Jokaiselle eri tyypipiirille on siis luotu oma Source-koodi, jossa määritellään tyypipiirin toiminta.

Jotta valvomo-ohjelmistolla pystyttäisiin käsittelemään oikeaa bittiä ulkoisesta muuttujasta INOUT\_BITIT, täytyi siis selvittää tyypipiirikohtaisista Source-koodeista, mitä muuttujaa mikäkin bitti vastaa. Source-koodeissa INOUT\_BITIT on nimetty uudestaan nimellä VALVOMO\_BITIT ja sille on koodin sisällä annettu tyypiksi STRUCT. Muuttujien järjestys, eli mitä VALVOMO\_BITIT-sanan bittiä mikäkin muuttuja vastaa, määräytyy sen mukaan, miten ne on syötetty koodiin STRUCT-komennon jälkeen. SCL -kielellä on STRUCT:ia käytettäessä sellainen ominaisuus, että se kääntää tavujen paikat. Seuraavassa taulukossa on listattuna konetyypipiiriin INOUT\_BITIT-muuttujan bittien muuttujat, osoite-esimerkit ja desimaaliarvot. Muiden tyypipiirien käyttämät INOUT\_BITIT-muuttujat vastaavat rakenteeltaan tätä esimerkkiä.

TAULUKKO 2. INOUT\_BITIT

MUUTTUJA	OSOITE	DEC
KASI_AUTO_VAL	0.0	256
KUITTAUS	0.1	512
KASI_KAYSEIS_ETEEN	0.2	1024
KASI_KAYSEIS_TAAKSE	0.3	2048
OOS_VAL	0.4	4096
OOS	0.5	8192
KAY_TAAKSE	0.6	16384
OHJAUS_TAAKSE	0.7	32768
KAY_ETEEN	1.0	1
TURVAKYTKIN	1.1	2
RISTIRIITA	1.2	4
VIKA	1.3	8
AUTO_KENTALTA	1.4	16
AUTO_JARJESTELMASTA	1.5	32
LUKITTU	1.6	64
OHJAUS_ETEEN	1.7	128

### 5.2.3 Sisäiset muuttujat

Ulkoisista muuttujista poiketen sisäiset muuttujat eivät saa arvoaan logiikalta, vaan niille syötetään arvot valvomo-ohjelmiston sisältä. Malli sisäisille muuttujille otettiin WinCC:llä luodusta projektista, jota käytettiin muutenkin pohjana tyyppiipiirejä luotaessa. Sisäiset muuttujat on nimetty siten, että nimi kuvaa muuttujan toimintaa todella hyvin. Jokaisen sisäisen muuttujan tyyppiä asetettiin STRING, joka mahdollistaa mahdollisimman suuren koon muuttujalle. Tätä tarvitaan koska suurin osa sisäisistä muuttujista on tekstipohjaisia. Tällöin niissä voidaan käyttää tarvittavaa määrää merkkejä. Sisäisten muuttujien nimeäminen noudattaa samoja sääntöjä kuin ulkoisten muuttujien nimeäminen. Eli kaikki K01-alkuiset sisäiset muuttujat ovat konetyyppiin sisäisiä muuttujia ja tuo alkuosa tullaan myöhemmissä projekteissa korvaamaan laitepositiolla.

TAULUKKO 3. Sisäiset muuttujat

NAME	TYPE
K01_NIMI	STRING
K01_POSITIO	STRING
K01_TOOLTIP	STRING
K01_TYYPPI	STRING
M01_POSITIO	STRING
M01_NIMI	STRING
M01_SKAALA_MIN	STRING
M01_SKAALA_MAX	STRING
M01_TOOLTIP	STRING
S01_POSITIO	STRING
S01_NIMI	STRING
S01_TOOLTIP	STRING
S01_VAL_SALLI_AS	STRING
V01_NIMI	STRING
V01_POSITIO	STRING
V01_TOOLTIP	STRING
V01_TYYPPI	STRING

### 5.2.4 Genie-alustus

Vijeo Citectin Super Genieen osoitetaan siinä käytettävät muuttujat sen käyttämän Genien kautta. Jokaisella tyyppiipiirikohtaisella Geniellä siis määritellään ne muuttujat, joita siihen liitetty Super Genie käyttää. Tämä tarkoittaa sitä, että kaikkia muuttujia, joita Super Geniellä käytetään, täytyy kutsua Geniessä. CiCodessa on käytössä kouralli-



nen *Assign*-alkuisia komentoja tähän tarkoitukseen. Näillä komennoilla linkitetään Super Genieen siinä käytettävät muuttujat. Komennot eroavat hieman toiminnoiltaan.

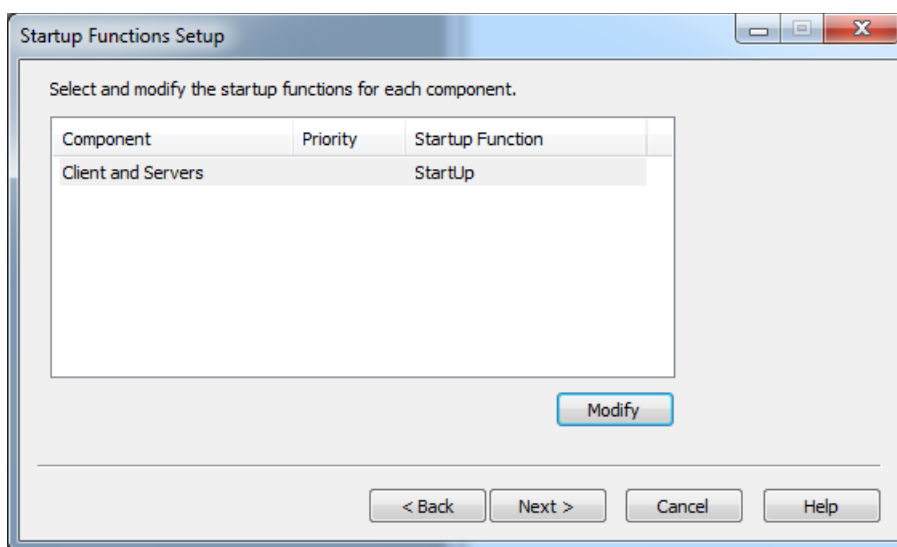
Suurin osa näistä komennoista ei tue maksimissaan kuin kahdeksaa muuttujaa. Koska IAs:n tyyppiireissä on kuitenkin käytössä runsaasti enemmän muuttujia kuin CiCode:ssa olevat komennot antavat maksimissaan käsitellä, täytyi muuttujien linkittämiseksi Super Genieen käyttää seuraavaa CiCode komentoa; *AssVarTags*. Tämä komento mahdollistaa kahdeksan muuttujan linkittämisen ja kyseisen komennon ketjuttamisen. Linkitys tehtiin siis niin, että ensimmäisellä *AssVarTags*-komennolla määritettiin kahdeksan ensimmäistä Super Genieessä käytettävää muuttujaa, minkä jälkeen lisättiin toinen *AssVarTags*-komento, jossa linkitettiin kahdeksan seuraavaa muuttujaa ja niin edelleen. Komentoa käytettäessä ketjutettuna määritellään siinä kokonaisluvulla ensimmäisen muuttujan lukuarvo. Tämä tarkoittaa sitä, että kun sarjassa toisena olevan *AssVarTags*-komentoon syöttää luvun kahdeksan, osaa ohjelma antaa sen sisällä ensimmäisenä olevalle muuttujalle järjestysnumeron yhdeksän.

Muuttujien linkitys, eli edellä esitettyjen komentojen syöttö, Super Genieen tehdään siihen liittyvän Genien sisällä sen input-komentokentässä. Tätä komentokenttää käytettäessä törmättiin kuitenkin sellaiseen ongelmaan, että siinä käytettävien merkkien maksimimäärä on 255. Kenttään voi tästä huolimatta syöttää enemmän merkkejä, joita kuitenkaan ei lueta tietyn määrän jälkeen. Muuttujien pitkien nimien ja niiden runsaan lukumäärän johdosta Super Geniellä ei saatu yhteyttä jokaiseen muuttujaan, koska ohjelma ei pystynyt lukemaan kirjoitettuja komentoja loppuun saakka, koska ne ylittivät maksimimerkkimäärän.

Ongelman ratkaisuksi päätettiin luoda valvomo-ohjelmiston sisältä löytyvällä CiCode Editorilla jokaiselle eri tyyppiirille funktio, jossa määritellään jokaisen eri Super Genien tarvitsemat muuttujat, alustetaan nämä muuttujat ja kutsutaan tarvittavia CiCode -komentoja muuttujien linkittämiseksi Super Genieen ja itse Super Genien avaamiseen. Näin voidaan kutsua yhtä funktiota Genien sisällä sen sijaan, että siellä kutsuisi *Assign*-komennolla suoraa muuttujaa. Ohjelma toimii siis siten, että kun runtimessä käyttäjä painaa Genietä, se kutsuu tyyppiirille kirjoitettua funktiota, jossa ohjelma osaa liittää oikeat muuttujat oikeaan Genieen toimilaitekohtaisen position avulla. Nämä kaikki funktiot löytyvät ohjelman sisältä tiedostosta *Genie\_alustus.ci*.

### 5.2.5 Sisäisten muuttujien alustus

Kuten aikaisemmin on esitelty, sisäisille muuttujille täytyy antaa arvo ohjelman sisältä. Kuten Genieidenkin alustus, myös tämä hoituu Cicode Editorilla. Sisäisten muuttujien alustusta varten tehtiin toinen Cicode-tiedosto, jossa määritellään jokaisen luodun sisäisen muuttujan arvo koodilla. Selkeyden ja ymmärrettävyyden säilyttämiseksi tehtiin jokaiselle eri tyyppiin omalle kutsuttava funktio. Esimerkiksi kaikkia kone- ja venttiilipiiriin paikallisten muuttujien alustuksia kutsutaan funktiossa IA\_Kone(). Kaikki sisäisten muuttujien alustamiseen kutsuttavat funktiot kirjoitettiin samaan Cicode-tiedostoon, jonka nimi on Paikallisten\_muuttujien\_alustus.ci. Tiedoston alkuun tehtiin StartUp-niminen funktio, jossa kutsutaan näitä sisäisten muuttujien alustamiseen käytettäviä funktioita. Käyttäjällä on mahdollisuus määrittää, minkä nimistä funktioita valvomo-ohjelmisto kutsuu valvomo-ohjelmiston runtime-tilan käynnistyessä. Tämä määrittäminen tehdään Computer Setup Wizard -työkalulla, jonka avulla vaihdettiin ohjelman käyttämä vakiofunktio. Näin ollen sisäiset muuttujat luetaan aina valvomon käynnistyessä. Toisin sanoen tyyppiin saavat staattiset arvonsa aina valvomon käynnistyessä.



KUVIO 25. Startup-funktion määrittäminen

Kone- ja venttiilityyppiin eroavat mittauksesta ja säätimestä siten, että niiden tyyppiin piirit ja niissä käytettävät toiminnot vaihtelevat niillä hallittavan toimilaitetyypin mukaan. Säädin ja mittaus pysyvät toiminnaltaan yleisesti ottaen aina samanlaisina, mutta moottori voi pyöriä kahteen suuntaan. Sitä voidaan ohjata taajuusmuuttajalla ja venttiili voi olla tyyppiltään säätö- tai on/off -venttiili. Opinnäytetyön alussa yksi tutkittavista asioista oli se, miten tyyppiin lopullisissa tyyppiin ratkaistaisiin. Mahdollis-

suutena oli siis myös se, että jokaiselle eri tapaukselle tehdään oma Genie ja Super Genie. Tämä olisi tietysti lisännyt runsaasti tehtävien piirien määrää, mutta tavoitteena oli, että tyyppivalintaan löytyy tehokkaampi ratkaisu.

Tyyppin valintaan päätettiin luoda kone- ja venttiilityyppiireille sisäinen muuttuja nimeltä TYYPPI. Se on toimilaittekohtainen muuttuja, jolle annetaan lukuarvo, jonka mukaan määräytyy jokaisen eri tyyppiiriin asetukset. Muuttujan lukuarvo määräytyy summattavista kokonaisluvuista, jotka seuraavat kahden potenssiin korotettujen kokonaislukujen arvoja. Jokaiselle valittavalle ominaisuudelle määritettiin koodiin sitä vastaava kokonaisluku, jotka käyttäjän tulee summata yhteen toimilaitteen varsinaisten ominaisuuksien mukaan. Toisin sanoen ohjelmassa tutkitaan, mitkä TYYPPI -muuttujan bitit eli ominaisuudet ovat päällä. Muuttujan arvon mukaan tyyppiiriikohtaisessa Super Geniessä piilotetaan objekteja, joita vastaavia kokonaislukuja ei ole summattu muuttujan arvoon.

### 5.3 Tyyppiiriobjektien ohjelmointi

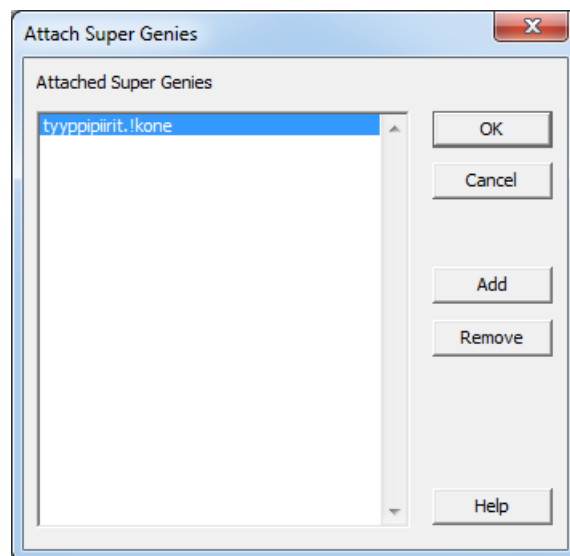
Genieiden ja Super Genieiden ohjelmointi tehdään Citect Graphics Builderilla, jossa jokaisella siinä käytettävällä objektilla on oma properties-valikko, jossa objektin kaikki ominaisuudet määritellään. Nämä valikot ja niiden toiminta ovat pääsääntöisesti samanlaiset kaikilla eri objekteilla eli niiden käyttö on myös samanlaista. Objektien toimintoa ohjataan Cicode-ohjelmointikielellä tehdyillä komennoilla, joilla luetaan tai kirjoitetaan muuttujien tiloja.

WinCC-ohjelmistossa ohjelmointi on hyvin samankaltaista kuin Vijeo Citectissä. Objekteille tai objektiryhmille on siis käytössä valikko, jossa tarvittavat ominaisuudet ohjelmoidaan tai otetaan käyttöön. Super Genieiden ulkoasun valmistuttua ja kaikkien käytettävien muuttujien luomisen jälkeen opinnäytetyö eteni seuraavasti: WinCC -valvomo-ohjelmistolla tutkittiin objekti objektilta kaikki niiden käyttämät asetukset, rajoitukset ja ominaisuudet, minkä jälkeen luotiin objektikohtaiset asetukset toiminnaltaan vastaaviksi Vijeo Citect -valvomo-ohjelmiston tyyppiireissä.

Super Genieiden toiminta riippuu muuttujien tiloista, jotka myös samalla ohjaavat Super Genieiden toimintaa täysin. Näin ollen kaikkien ohjelmoitavien objektien toimintaa oh-

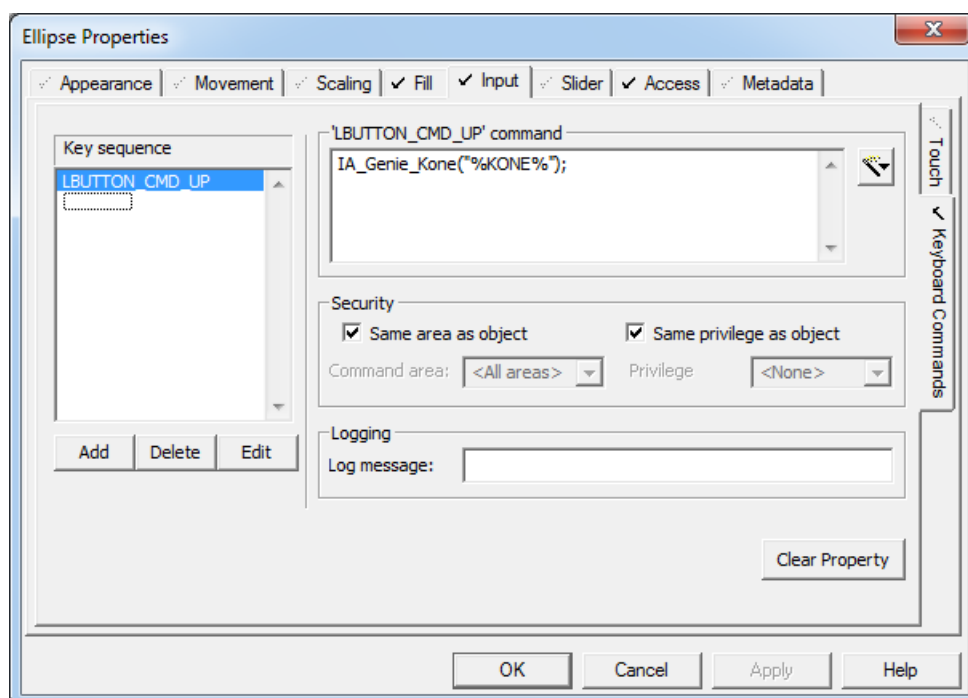
jaa kussakin tapauksessa määräävä muuttuja. Objekteja ohjelmoimissa täytyy siis aina viitata johonkin muuttujaan. Tämä viittaus tapahtuu Vijeo Citectissä numeroilla, joiden alkuun ja loppuun on sijoitettu kysymysmerkit; esimerkiksi ?12?. Tämä muoto kertoo ohjelmalle, mihin muuttujaan halutaan viitata aiemmin esitellyn AssVarTags-komennon sisällä.

Että ohjelma pystyy muodostamaan yhteyden Super Genien sisällä kutsuttavan muuttujan ja muuttujaluettelon välille, täytyy oikeaa muuttujaa kutsua Super Genietä ohjaavassa Geniessä. Genietä luodessa siihen liitetään tieto siitä, mikä Super Genie avautuu sitä painettaessa valvomonäytöllä. Tämä tehdään Graphics Builderin toiminnolla Attach Super Genie, jossa luetteloon valitaan Genien käyttämä Super Genie.



KUVIO 26. Super Genien liittäminen Genieen

Lisäksi Genien asetuksissa täytyy määrittää, että se sitä painettaessa kutsuu tehtyä funktiota, jonka kautta se osaa linkittää oikean muuttujan sitä vastaavaan numeroon. Funktiota kutsu toteutetaan Genie-objektin Input-asetuksista, joissa määritellään, että hiiren vasenta nappia painettaessa kutsutaan tyyppiiriikohtaista funktiota. Kyseisen funktion sisään kirjoitetaan myös prosentimerkkien sisään funktiokohtainen muuttuja; esimerkiksi %KONE%. Funktiokohtaisella muuttujalla määritetään Genietä käytettäessä toimilaitekohtainen positio. Tämän positiotiedon avulla ohjelma osaa myös liittää kirjoitettussa Genie\_alustus.ci-koodissa toimilaiteposition kaikkiin tyyppiiriin käyttämiin muuttujiin. Näin ollen se osaa linkittää oikeat muuttujat muuttujaluetteloon luotaviin toimilaitekohtaisiin muuttujiin.

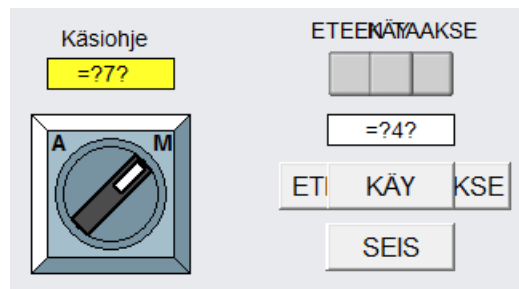


KUVIO 27. Funktion kutsu Geniessä

### 5.3.1 Objektien yleiset ominaisuudet

Kaikilla tyyppiireissä käytetyillä objekteilla on tietyt toiminnot, jotka ovat toiminnaltaan samanlaisia. Jokaisella ohjelmoitavalla objektilla on Input-välilehti, jossa määritellään komento ja tarvittava toimenpide komennon suorittamiseen. Toimenpiteen voi määrittellä kahdella eri tavalla: Toinen tapa on kosketuksesta (Touch), mikä tarkoittaa hiiren painallusta. Toinen tapa on näppäinyhdistelmästä (Keyboard Commands). Ensimmäisessä toimenpidetyypissä valitaan aktivoivaksi signaaliksi joko hiiren napin alastai ylöspainallus tai toistuva painaminen. Jälkimmäisessä toimenpidetyypissä eli näppäinyhdistelmässä käyttäjä voi määrittää, mitä nappia painamalla suoritetaan inputkentän komento. Tämä voi olla joko hiiren nappi tai näppäimistöltä näppäinyhdistelmä.

Toinen yleisesti käytetty objektien yleinen ominaisuus on objektin piilottaminen. Tässä käyttäjä syöttää Appearance → Visibility -välilehdelle ehdon (Hidden when), jolloin objekti halutaan piilottaa. Tätä ominaisuutta käytetään hyväksi erityisesti kone- ja venttiilityypipiirien TYYPPI-muuttujaa luettaessa. Samalla ominaisuudella estetään myös päällekkäisten objektien, esimerkiksi nappien ja merkkivalojen, näyttäminen. Kuvio 28 on valmiin konetyypipiiriin Super Geniestä, kun sitä Graphics Builderilla muokataan. Kyseisestä kuvioista käy ilmi nappien ja valojen päällekkäisyys.



KUVIO 28. Objektien päällekkäisyys

Kolmas yleinen ja hyödyllinen toiminto objektien ominaisuuksissa on objektin käyttökyvyn poisto. Access → Disable -välilehdellä annetaan objektille ehto (Disable when), jolloin se ei ole käytössä. Tätä toimintoa hyödynnetään muun muassa siinä tilanteessa, kun toimilaitteen ohjaus on paikallisohjauksella. Kun toimilaitetta ohjataan kentältä, kaikki valvomosta tapahtuva hallinta on estettynä. Tässä tilanteessa kaikki napit ja symbolit on estetty valittavaan tyyliin. Esimerkkinä tästä on Embossed-tyyli, jolloin napin fontti upotetaan merkiksi siitä, että sen käyttö on estetty.

Neljäs usein objekteissa käytetty ominaisuus on tooltipin näyttö, jolla määritellään objektin päällä näytettävä teksti silloin, kun valvomo-ohjelmisto on käynnissä ja operattori vie hiiren kursorin objektin päälle. Tämän saa asetettua Access → General -välilehdeltä. Tämä kenttä täytettiin jokaiseen tyyppiin objektiin, jolla on jonkinlaista toiminnallisuutta.



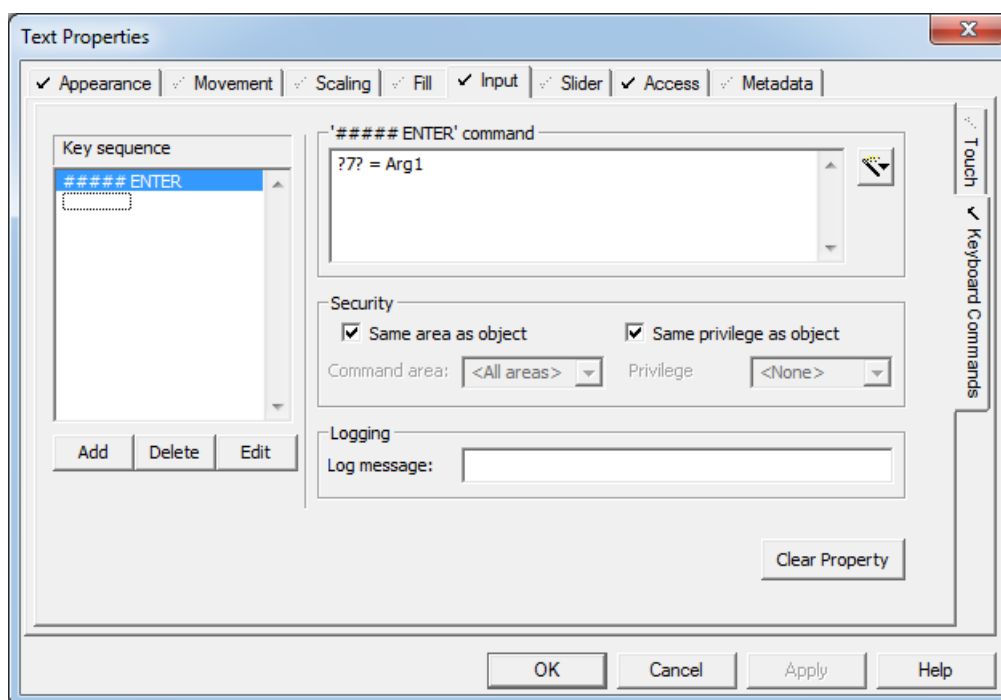
KUVIO 29. Tooltipin käyttö

### 5.3.2 Numero- ja tekstikenttien ohjelmointi

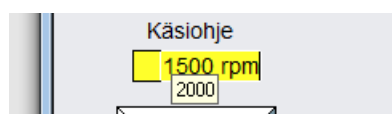
Numero- ja tekstikentät ovat toiminnaltaan lähes samanlaisia ja ne on tarkoitettu muuttujan arvon esittämiseen. Osa numerokentistä on ohjelmoitu niin, että niiden kautta voidaan muuttujalle syöttää myös arvo. Objektin ominaisuuksista löytyy välilehti Appearance → Display Value, jossa valitaan näytettävän arvon tyyppi ja kenttä, johon syötetään viittaus käytettävään muuttujaan. Ohjelmoinnin alkuvaiheessa käytettiin nume-

roiden esittämiseen Numeric-tyyppiä. Tämän käyttö kuitenkin korvattiin myöhemmin käyttämällä String-tyyppistä tietokenttää, joka mahdollistaa Excelissä syötettävän yksikön näyttämisen. Tällöin muuttujasta esitetään mitattu tai syötetty arvo ja muuttujan yksikkö halutussa muodossa, eikä Super Genieen tällöin tarvitse lisätä kiinteitä yksiköitä kullekin muuttujalle.

Kun numerokenttää käytetään niin, että sillä voidaan asettaa muuttujalle arvo, täytyy siihen ohjelmoida Input-välilehdelle tarvittava toiminto. Keyboard Commands -välilehdelle asetetaan Key Sequence (##### ENTER) ja tälle komennoksi muokattava muuttuja, joka asetetaan muokattavaksi komennolla Arg1 (esimerkiksi ?? = Arg1). Näin tehtäessä käyttäjä voi asettaa arvokentän muuttujalle uuden arvon viemällä hiiren kentän päälle, syöttämällä halutun arvon näppäimistöllä ja painamalla tämän jälkeen Enter-painiketta.



KUVIO 30. Syötettävän arvokentän ominaisuudet

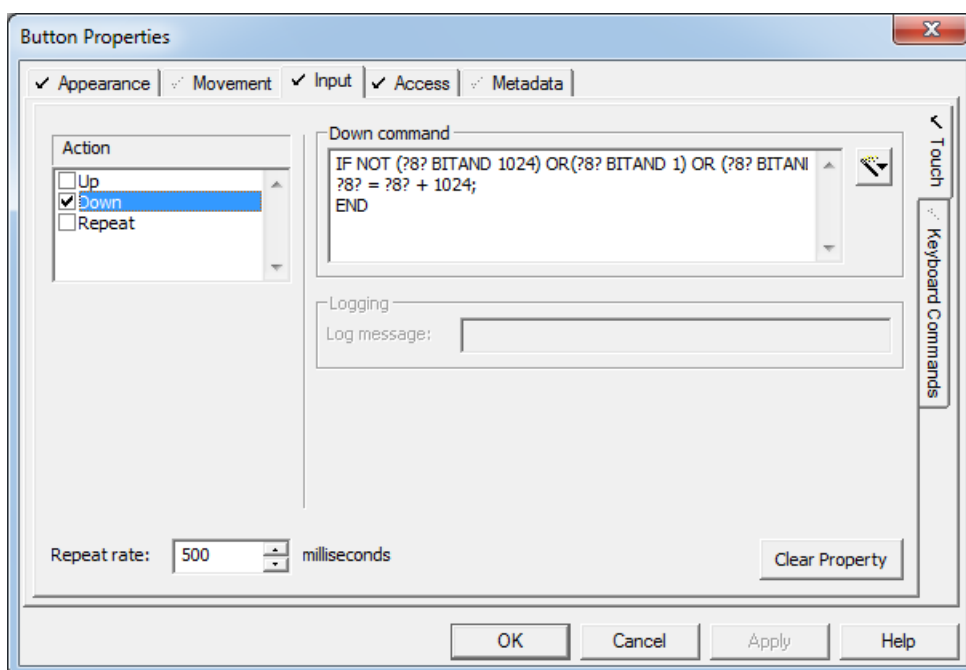


KUVIO 31. Arvokenttään syöttö

### 5.3.3 Nappien ja symbolien ohjelmointi

Nappien ja symbolien toiminta valvomo-ohjelmistossa tehdään myös syöttämällä ehto ja komento objektin Input-välilehdelle. Yleisesti jokaisen napin painallus asettaa tarvittavan bitin päälle muuttujasta, mutta ei palauta sitä. Tämä tapahtuu siksi, että bitin palautus tehdään logiikkaohjelmistossa. Esimerkiksi tässä tapauksessa, kun testikäytössä on Siemensin logiikka, IAs:n luomissa tyyppiirikohtaisissa Source-koodeissa. Käytössä olevassa komentomuodossa aluksi tutkitaan, onko asetettava bitti päällä, tai tarkastellaan muita käynnistämistä rajoittavien bittien tiloja. Jos mikään muuttujan arvo ei rajoita suoritettavaa IF-lauseetta, kirjoitetaan käskyssä tarvittava muuttujan arvo.

Ainoastaan A/M-kytkinsymboleiden käytössä pitää palauttaa muuttujaa ohjaavan bitin tila. Eli IF-lausekkeen lisäksi täytyy komennossa käyttää ehtoa IFELSE. Näillä komennoilla saadaan rakennettua animoidulle symbolille komento, jossa tutkitaan aluksi muuttujan bitin tilaa ja sen mukaan bitti joko kytketään päälle tai pois päältä.



KUVIO 32. Painonapin komento (esimerkkikuvassa konetyyppiiriin käynnistys-painike)

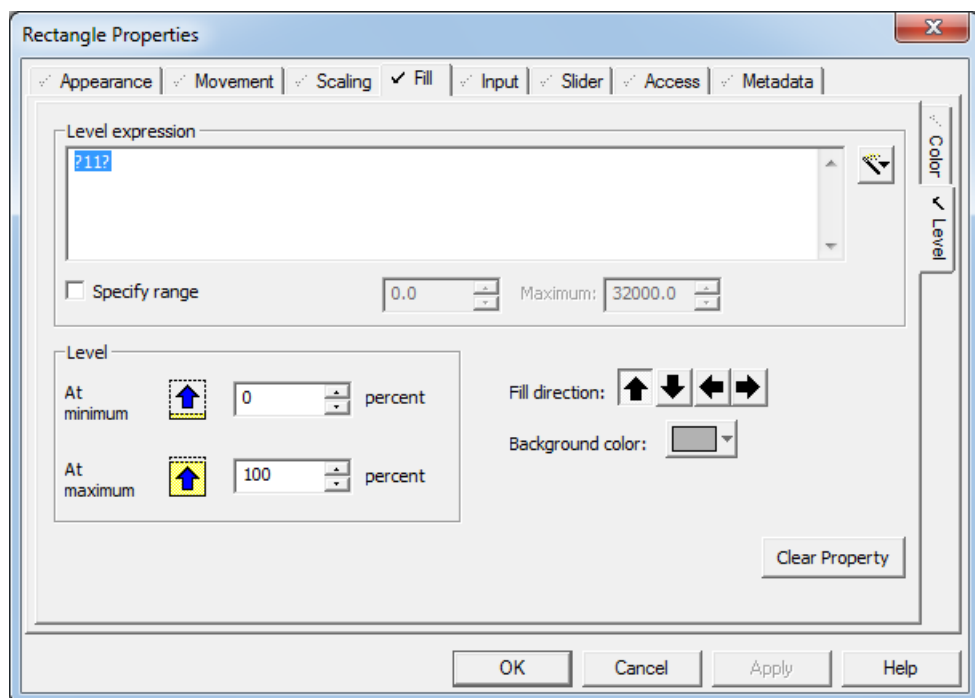
Mittaustyyppiiriistä löytyy TRENDI-nappi, jolla avataan uusi Super Genie -ikkuna, jossa on kyseisen mittauksen trendinäyttö. Nappia käytettäessä ohjelma suorittaa Cicode-funktion AssChainWinFree(), joka avaa uuden funktiossa määritellyn Super Genien vanhan Super Genien arvoilla. Toisin sanoen Genie-funktiota kutsutaan samalla laitepo-



sitiolla, jolla alkuperäinen Super Genie avattiin. Komento on siis tarkoitettu ponnahdusikkunan avaamiseen ponnahdusikkunasta. Tätä painiketta ohjelmoitaessa törmättiin sellaiseen ongelmaan, että sitä käytettäessä runtime-tilassa ikkuna sulki vanhan ikkunan, kuten kuuluu. Tämän lisäksi se sulki myös koko valvomon ja jätti vain uuden avatun ikkunan aktiiviseksi valvomosta. Ongelman tiimoilta oltiin yhteydessä muun muassa Schneider Electric tukeen. Lopulta asia korjaantui sillä, että käytettävä komento siirrettiin Keyboard Commands -välilehdeltä, jossa Key sequence -kenttään oli määritelty vasemman hiiren-napin painallus (LBUTTON\_CMD\_DN) Touch-välilehteen. Ongelman korjaannuttua tehtiin vastaava komennonsiirto kaikille tyyppiireissä esiintyville Input-komennoille, joille siirto oli mahdollista tehdä. Tällä tavalla ehkäistiin ongelman muodostumisen uudestaan missään yhteydessä.

### 5.3.4 Kuvioiden ohjelmointi

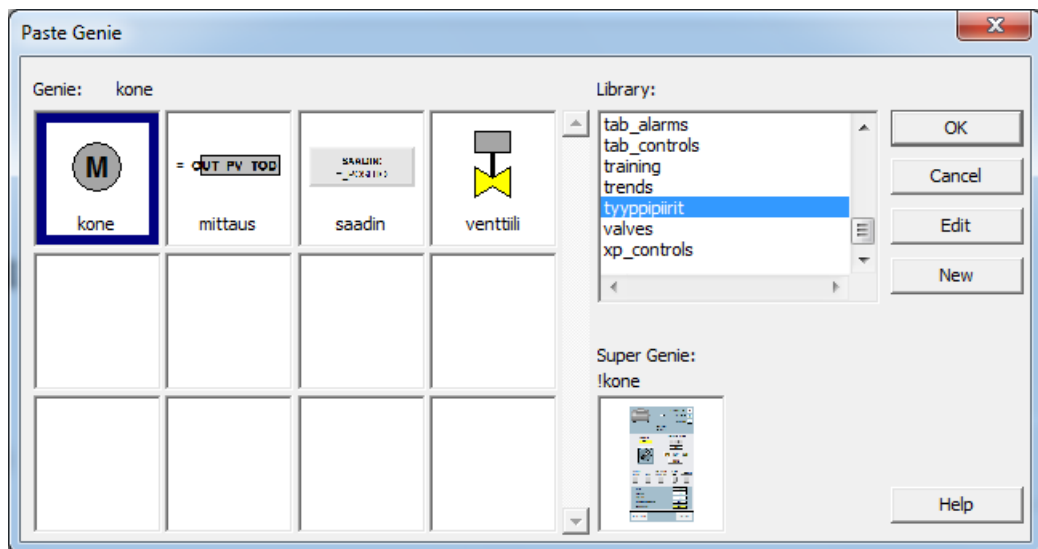
Osa arvokentistä toimii siten, että kentän taustaväri muuttuu sen ilmaiseman arvon lähteen muuttuessa. Mittauspiirissä sen mitta- ja raja-arvopalkit vaihtavat väriään mitattujen ja asetettujen arvojen mukaan. Tällainen toiminto tehdään kuviokohtaisista asetuksista löytyvältä Fill → Color tai Level-välilehdillä. Ehtokenttään syötetään osoitus käytettävään muuttujaan ja sivulta valitaan täytön haluttu suunta tai tilojen osoittamat värit.



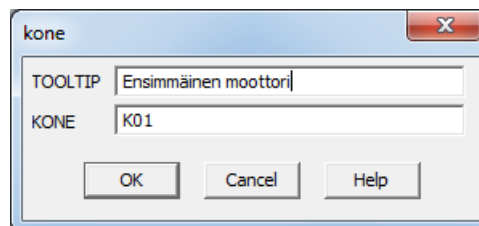
KUVIO 33. Täytön asettelu

## 5.4 Tyypipiirien käyttöönotto ja testaus

Toimilaittekohtainen Genie lisätään valvomonäytön sivulle Graphics Builderin Paste Genie -toiminnolla, josta valitaan oikea kirjasto (tyypipiirit) ja sen jälkeen käytettävä Genie. Ohjelma myös näyttää käyttäjälle Genieen liitetyn Super Genien. Genietä lisättäessä ohjelma pyytää käyttäjää antamaan toimilaitteelle tooltip- ja positio -tiedot. Positio-tiedon avulla ohjelma osaa liittää toimilaitteeseen sen käyttämät muuttujat, kuten Genie\_alustus.ci -tiedostossa on määritelty.



KUVIO 34. Genien liittäminen



KUVIO 35. Genien tiedot

Tyypipiirien toiminnan testaamista varten tehtiin projektiin testisivu, jonne liitettiin tarvittava määrä kutakin luotua tyypipiiriä. Testausta varten ja tyypipiirien oikean toiminnan todentamiseksi täytyi ohjelmoida myös logiikkaohjelmisto vastaamaan valvomo-ohjelmistossa käytettyä testikokoonpanoa. Siemensin Simatic Managerilla luotiin siis ohjelma, jossa käytettiin yrityksen sille luomia tyypipiirejä ja niiden Sourcekoodia. Sen avulla puukotettiin tarvittavaa osoitetta sen toiminnan todentamiseksi. Testisivun kuvakaappaus on liitteessä 5.

## 6 KÄYTTÖOHJE

Opinnäytetyöstä sovittaessa työnantajan kanssa sovittiin myös siitä, että osana suoritettavaa työtä olisi aiheeseen liittyen käyttöohjeen teko asiakkaalle. IAs:lla on siis olemassa asiakkaalle välitettävä käyttöohje, joka opastaa valvomo-ohjelmiston käyttöä kun sillä ohjataan varsinaista tuotantolaitosta. Siinä siis opastetaan muun muassa vastaavien tyyppipiirien käyttö, väritysten merkitykset ja hälytysten hallinta. Alkuperäinen tarkoitus oli, että vastaava käyttöohje luotaisiin Vijeo Citect ympäristöön, jossa esiteltäisiin laadittujen tyyppipiirien toiminta. Ulkopuolisista tekijöistä johtuen ja opinnäytetyöhön käytettävän ajan lähestyessä loppuaan oli kuitenkin selvää, ettei tällaiselle käyttöohjeelle ollut tarvetta.

Opinnäytetyön ohjaajan ja osastopäällikön kanssa sovittiinkin, että opinnäytetyöhön kuuluvan käyttöohjeosuuden tehtäisiin alkuperäisestä suunnitelmasta poiketen IAs:n omaan käyttöön. Tämä käyttöohje opastaa IAs:n työntekijöitä siinä, miten luodut tyyppiipiirit otetaan käyttöön uudessa Vijeo Citect -projektissa. Toisin sanoen käyttöohjeessa kerrotaan se, miten valmiit tyyppiipiirit saa siirrettyä projektista toiseen, miten käyttäjä hallitsee tyyppiipiirin tarvitsemia muuttujia ja miten Graphics Builderilla otetaan tyyppiipiirikohtainen Genie käyttöön. Aiheesta keskusteltaessa todettiin, että tämä aihe käyttöohjeelle tukee myös paljon paremmin varsinaista opinnäytetyötä kuin alkuperäinen käyttöohjeen aihe.

Opinnäytetyön alkupuolella tutkittiin sitä, miten Vijeo Citect -valvomo-ohjelmiston ja sen kanssa käytettävän eri valmistajan tekemän logiikan välille saadaan muodostettua yhteys eli miten ne saadaan keskustelemaan keskenään. Asiaan perehdyttyä kollega esitti toiveen, että tästä aiheesta tehtäisiin käyttöohje myös yrityksen sisäiseen käyttöön. Opinnäytetyön aikana tehtiin siis kaksi eri käyttöohjetta aiheeseen liittyen; toinen yhteyden muodostamiseen ja toinen tyyppiipiirien käyttöönottoon.

## 7 POHDINTA

Opinnäytetyöni tavoitteena oli tutkia, miten Vijeo Citect -valvomo-ohjelmistolla luodaan tyyppiipiirit sillä tavalla, että ne vastaavat toiminnaltaan Insta Automation Oy:n luomia ja käyttämiä tyyppiipiirejä. Tavoitteena oli myös luoda käyttöohje IAs:n asiakaskäyttöön liittyen työssä käytettävään ohjelmistoon. Minulla ei ollut aikaisempaa työkokemusta valvomosuunnittelusta koulun ulkopuolelta, joten uuden asian opettelu lisäsi mielenkiintoa ja haastetta tutkimustyöhön. Tutkimuksen tekoon varattiin aikaa puoli vuotta, jonka aikana varsinaisen työn ja itse opinnäytetyön täytyy tulla valmiiksi. Näiden asioiden vuoksi tai niistä huolimatta tavoitteeseen päästiin molempien töiden osalta.

Neljä suunniteltua tyyppiipiiriä vastaavat toiminnaltaan yrityksen tyyppiipiirimallia, jonka pohjalta ne on luotu käyttäen tutkimustyössä kahden muun valvomo-ohjelmiston tyyppiipiirejä. Suurin ero vanhoilla ja minun luomilla tyyppiipiireillä on niiden ulkoasussa, sillä käytin niiden tekoon Instan väripalettia ja uusimmasta ohjelmistoversiosta löytyviä symboleja. Lisäksi niistä poistui kiinteät yksikkömerkinnät, minkä mahdollisti Vijeo Citect -valvomo-ohjelmiston ominaisuus, jolla yksikköä voi kuljettaa muuttujan mukana. Suunnittelutyön aikana löytyi myös ratkaisu sille, miten uuden projektin aikana voi tyyppiipiirejä luodessa toteuttaa niin sanotun tyyppiivalinnan, jolla määritellään valvomo-ohjelmistossa näytettävän ponnahdusikkunan ominaisuudet toimilaitekohtaisesti. Opinnäytetyönäni tehdyt tyyppiipiirit ovat siis täysin toimintavalmiit ja ne voidaan ottaa käyttöön tulevilla projekteilla niitä varten luotujen käyttöohjeiden avulla.

Työn teon yhteydessä, ja varsinkin tyyppiipiirien testauksen tekoon ja järjestämiseen, käytin paljon aikaa myös Siemensin ohjelmistojen parissa. Työn aikana siis kertyi käyttö- ja ohjelmointikokemusta eri ohjelmien tiimoilta. Näitä muita ohjelmia ei opinnäytetyössäni sen tarkemmin käsitellä. Koen työn aikana oppineeni valvomo-ohjelmiston toiminnasta ja ohjelmoinnista erittäin kattavasti ja uskon, että työ on antanut minulle vahvat edellytykset jatkaa työskentelyä vastaavissa työtehtävissä.

Tyyppiipiirin visuaalista toiminnallisuutta voisi kehittää edelleen. Lisäksi voitaisiin hyödyntää valvomo-ohjelmiston objektikohtaisia animointiominaisuuksia. Tutkimusta voisi jatkaa selvittämällä, miten muiden yrityksen käyttämien valvomo-ohjelmistojen tyyppiipiirien ulkoasu voitaisiin päivittää.

## LÄHTEET

Citect yritysmyynti. Luettu 3.7.2014. <http://en.wikipedia.org/wiki/Citect>

Insta 50 vuotta. Historiikki. Luettu 30.6.2014. <http://www.insta.fi/group/insta-50-vuotta-23-6-2010-historiakatsaus/>

Insta Automation Oy. Luettu 2.7.2014. <http://www.insta.fi/automation/>

Insta DefSec Oy. Luettu 2.7.2014. <http://www.insta.fi/defsec/>

InTouch. Luettu 7.7.2014. <http://en.wikipedia.org/wiki/Wonderware>

OPC. Luettu 13.8.2014. <https://opcfoundation.org/>

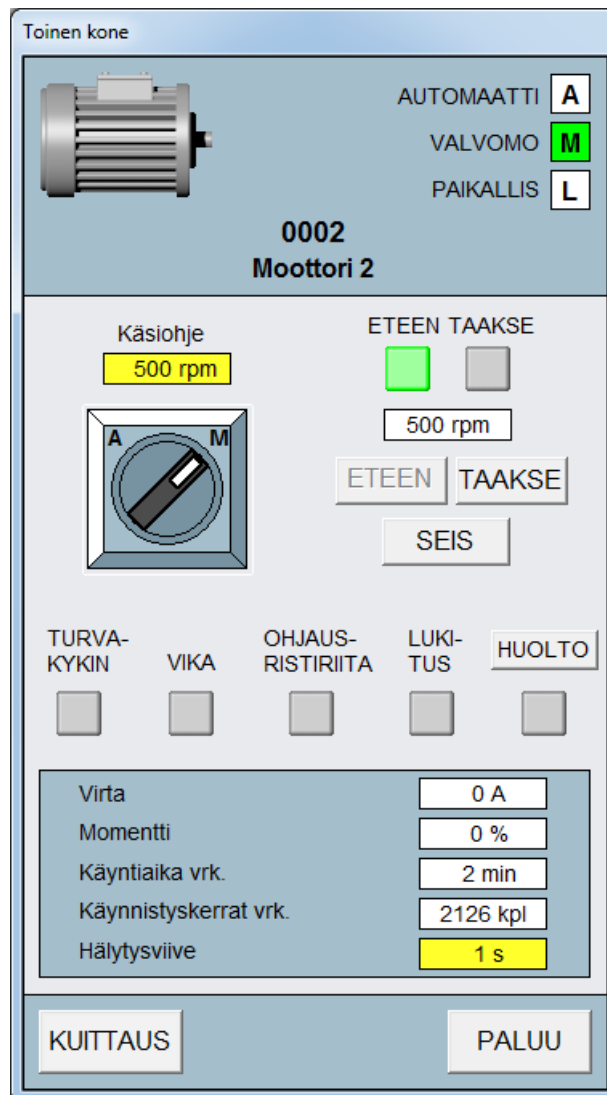
Suomen Automaatioseura Ry, 2010. Valvomo – Suunnittelun periaatteet ja käytännöt. Copy-Set Oy, Helsinki 2010.

Tagit. Luettu 12.8.2014. <http://www.integraxor.com/blog/what-is-tag-or-point-in-scadahmides>

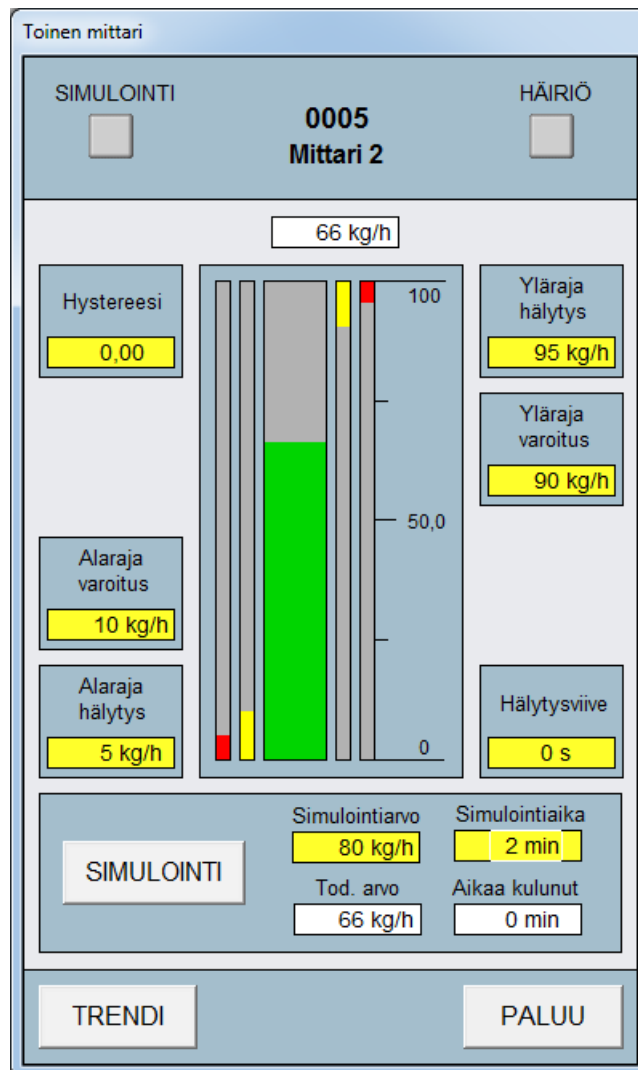
WinCC. Luettu 4.7.2014. <http://en.wikipedia.org/wiki/WinCC>

# LIITTEET

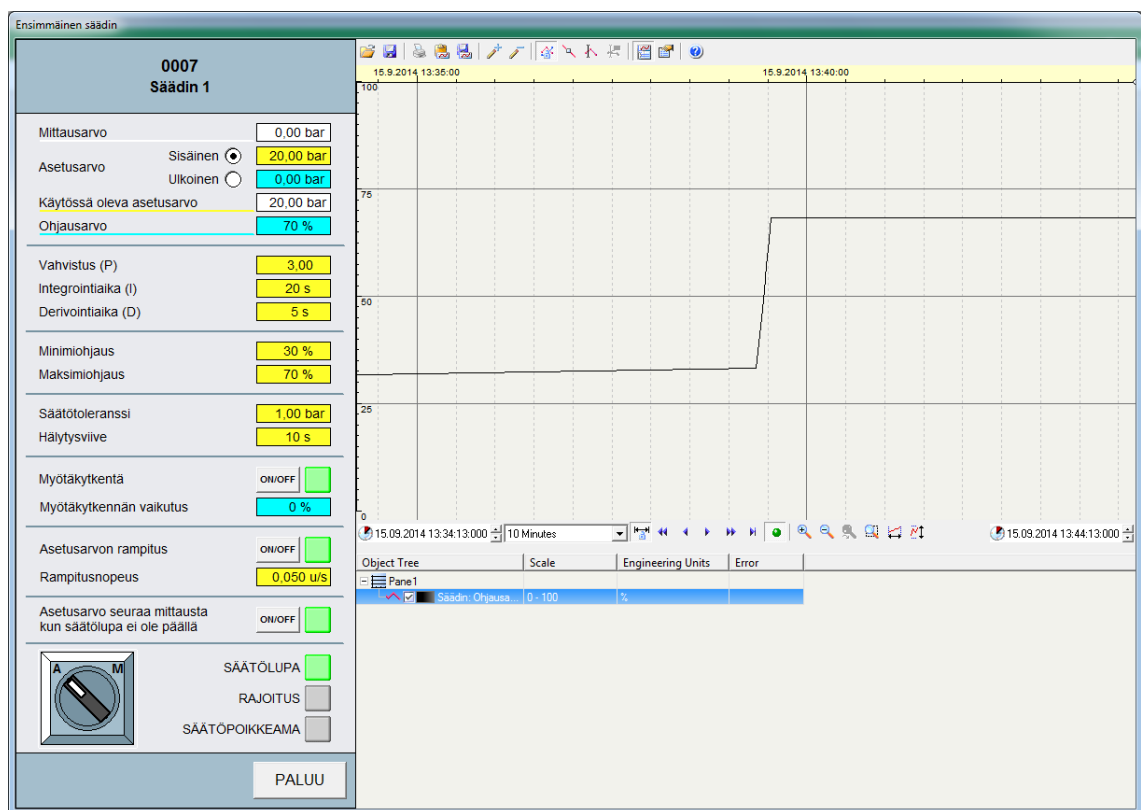
## Liite 1. Konepiirin Super Genie



## Liite 2. Mittauspiirin Super Genie




## Liite 3. Säädinpiirin Super Genie





## Liite 4. Venttiilipiirin Super Genie

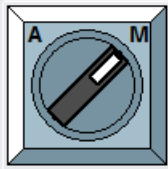
Toinen venttiili



AUTOMAATTI  A  
VALVOMO  M  
PAIKALLIS  L

0011  
Venttiili 2

AUKI  KIINNI









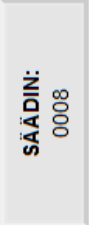






AUKI  
KIINNI

VIKA  OHJAUS-  
RISTIRIITA  LUKI-  
TUS  HUOLTO

Aukiohjaus kerrat vrk	<input type="text" value="10 kpl"/>
Kiinniohjaus kerrat vrk	<input type="text" value="9 kpl"/>
Hälytysviive	<input type="text" value="1 s"/>

KUITTAUS  PALUU

## Liite 5. Testisivu

KONE	MITTAUS	SÄÄDIN	VENTTIILI
1. 	4. 	7. 	10.  MG
2. 	5. 	8. 	11.  ONOFF
3. 	6. 	9. 	12.  ONOFFSÄÄTÖ
			13.  SÄÄTÖ