

RIKU LINDFORS

VALVOMOSOVELLUKSEN TOTEUTTAMINEN  
SEKOITUSPROSESSILLE

Automaatiotekniikan koulutusohjelma

2014

## VALVOMOSOVELLUKSEN TOTEUTTAMINEN SEKOITUSPROSESSILLE

Lindfors, Riku  
Satakunnan ammattikorkeakoulu  
Automaatiotekniikan koulutusohjelma  
Toukokuu 2014  
Ohjaaja: Asmala, Hannu  
Sivumäärä: 32  
Liitteitä:

Asiasanat: WinCC, Sekoitusprosessi, Käyttöliittymä, simulointi

---

Tämän opinnäytetyön tavoitteena oli toteuttaa käyttöliittymä, joka mahdollistaisi jo olemassa olevan sekoitusprosessin simulointimallin hallinnan. Työ toteutettiin Satakunnan ammattikorkeakoulun automaatiolaboratorion tiloissa.

Käyttöliittymään mallinnettiin olemassa oleva fyysinen ohjauspulpetti, jolla simuloitua sekoitusprosessia oli tähän asti operoitu. Koko järjestelmä muodostui Siemensin S7-300-sarjan logiikasta, fyysisestä ohjauspulpetista sekä kahdesta Pc yksiköstä, joista toisessa ajettiin sekoitusprosessin simulointimallia ja toiseen toteutettiin käyttöliittymä prosessin hallintaa varten. Käyttöliittymän toteutuksessa käytettiin Siemensin S7-Logiikkaohjelmaa ja WinCC flexible-valvomo-ohjelmistoa.

Opinnäytetyössä tutustutaan sekoitusprosessin periaatteisiin, käyttöliittymään, siihen mistä hyvä käyttöliittymä muodostuu sekä toteutuksessa tarvittaviin laitteisiin ja työkaluihin. Lopuksi annetaan ohjeita valvomosovelluksen luontiin ja arvioidaan käyttöliittymän toteutusta.

# CREATING A CONTROL ROOM INTERFACE FOR A MIXING PROCESS

Lindfors, Riku

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in automation

May 2014

Supervisor: Asmala, Hannu

Number of pages: 32

Appendices:

Keywords: WinCC, mixing process, User interface, Simulation

---

The purpose of this thesis was to create a user interface for an existing simulation model of mixing process. The work was carried out at the facilities of Satakunta University of Applied Sciences.

The user interface was based on an existing physical operating panel which had been the only way to control the simulated mixing process until today. The system contained a Siemens S7-300 series logic, a physical operating panel and two pc units from which the other was meant for the mixing process and the other where the interface for controlling was going to be built. User interface was carried out by using Siemens S7 logic programming software and WinCC flexible control room software.

In this thesis we get to know the basics of the mixing process, what tools and components are needed for creating the user interface and also what does it need to create a good user interface. At the end instructions for creating a user interface and evaluation of the user interfaces execution.

# SISÄLLYS

1	JOHDANTO .....	6
2	SEKOITUSPROSESSIN KUVAUS .....	7
2.1	Prosessi .....	7
2.1.1	Simatic Manager STEP 7 .....	8
2.2	Simulointimalli .....	9
2.2.1	3DCreate .....	9
2.3	Valvomosovellus.....	10
2.3.1	Automaatti/reseptiohjaus.....	10
2.3.2	Manuaaliohjaus .....	11
2.3.3	WinCC Flexible .....	11
3	KÄYTTÖLIITTYMÄ .....	12
3.1	Käyttöliittymän luokittelu.....	13
3.1.1	komentopohjainen käyttöliittymä .....	13
3.1.2	Graafinen käyttöliittymä.....	14
3.2	Käyttöliittymän ominaisuudet.....	14
3.2.1	Selkeys .....	15
3.2.2	Yhtenäisyys .....	15
3.2.3	Tehokkuus .....	15
3.2.4	Tarkoituksenmukaisuus.....	16
4	VALVOMOSOVELLUKSEN TOUTEUTUS.....	16
4.1	Yhteydet.....	16
4.1.1	MPI-väylä .....	17
4.1.2	Profibus .....	17
4.1.3	STEP7:n JA WinCC:n yhteys.....	17
4.1.4	Tagit .....	19
4.2	Valvomokäyttöliittymän luominen .....	21
4.2.1	Painonappi .....	23
4.2.2	Toimintovallo .....	24
4.2.3	Palkkinäyttö .....	25
4.2.4	Hätä-seis-painike.....	26
4.2.5	ON/OFF-Kytkin.....	27
4.3	Välilehdet.....	27
4.3.1	Info-sivu .....	28
4.3.2	Reseptiohjaus .....	29
5	KÄYTTÖLIITTYMÄN TESTAUS .....	29
6	YHTEENVETO .....	30

LÄHTEET .....	32
LITTEET	

## 1 JOHDANTO

Erilaisten prosessien ohjauksessa käytetään valvomoista löytyviä fyysisiä ohjauspaneeleja tai PC:ltä löytyvää virtuaalista käyttöliittymää. Virtuaalisen käyttöliittymän avulla on prosessin tiedot ja ohjaukset helppo jakaa verkon yli useammallekin päätelaitteelle, oli kyseessä sitten PC tai kosketusnäyttölinen ohjauspaneeli. Satakunnan ammattikorkeakoululla on opetuskäyttöön ja esittelykäyttöön tarkoitettu sekoitusprosessin simulointimalli, jota on tähän asti ollut mahdollista ohjata vain fyysisen ohjauspulpetin avulla ja jolle nyt haluttiin toteuttaa valvomosovellus ohjausta varten.

Työn aloittamista varten tuli ensin selvittää, mitä vaatimuksia valvomosovellukselle oli asetettu, mitä asioita haluttiin ohjata ja mistä prosessin vaiheista haluttiin saada tarkempaa tietoa näytölle. Selvitettiin myös, mikä valvomosovellustyökalu olisi paras kyseiseen käyttötarkoitukseen sekä helpoin myös lopulliselle käyttäjälle. Selvityksen pohjana käytettiin kyseiseen sekoitusprosessiin jo aikaisemmin käyttöliittymien opintojaksolla suunniteltua valvomosovelluspohjaa.

Työ toteutettiin Siemensin STEP 7 -ohjelmalla, jolla oli jo aiemmin toteutettu sekoitusprosessin logiikkaohjelma. Prosessin ohjaukseen liitettiin WinCC Flexible-valvomo-ohjelmistolla toteutettu käyttöliittymäympäristö.

Opinnäytetyön kirjallisessa osuudessa perehdytään simuloitun sekoitusprosessin toimintaan, käytettyihin ohjelmistoihin ja käyttöliittymäsuunnitteluun. Pääpaino on kuitenkin WinCC-ohjelmistolla ja sen eri ominaisuuksiin tutustumisella. Lisäksi käsitellään tarpeellisia rajapintoja ja liityntöjä, jotka ovat projektin toteutuksen kannalta tärkeitä.

## 2 SEKOITUSPROSESSIN KUVAUS

Opinnäytetyössä toteutettiin valvomosovellus sekoitusprosessin simulointimallille. Visual Components 3DCreate-ohjelmalla mallinnettu sekoitusprosessi pyörii omalla tietokoneellaan Satakunnan ammattikorkeakoulun automaatiolaboratoriotiloissa. Opinnäytetyön laitteistoon kuului lisäksi toinen tietokone, johon valvomosovellus toteutettiin, fyysinen ohjauspulpetti, sekä Siemensin ohjelmoitava S7-300-sarjan lo-  
giikka. Valvomosovelluksen toteuttamiseen käytettiin Siemensin Step7 Simatic Manageria ja saman valmistajan WinCC Flexible-ohjelmistoa. Tässä osiossa tutustaan projektin eri osa-alueisiin ja käydään läpi niissä käytettyihin ohjelmistoihin.



Kuva 1. Opinnäytetyön laitteistoa

### 2.1 Prosessi

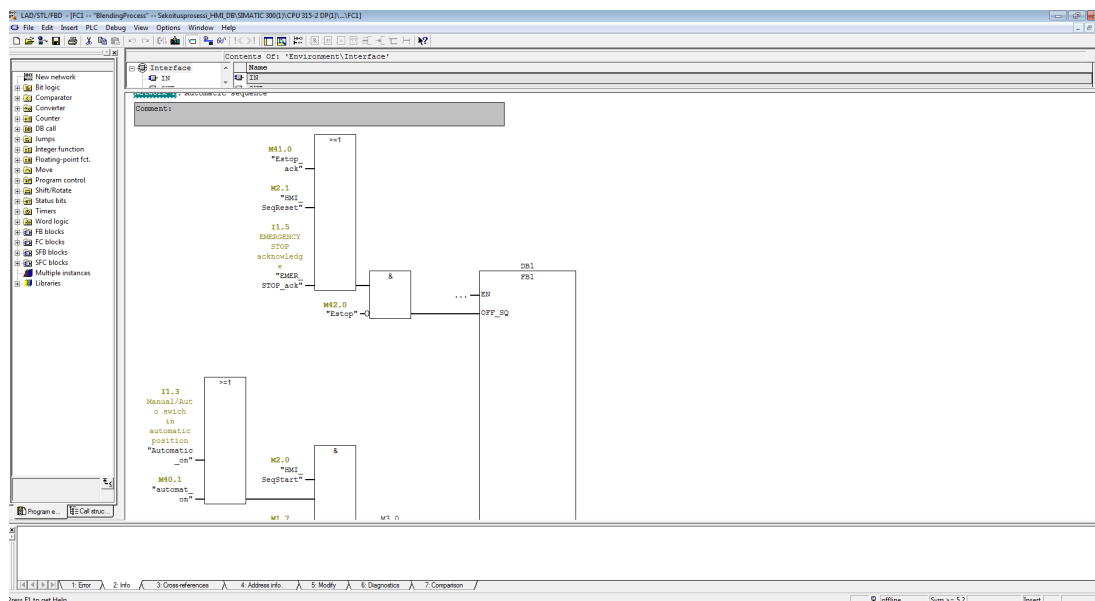
Sekoitusprosessin simulaatiomallissa prosessiin sisältyy kolme tankkia, sekoitin ja kaksi pumppua. Sekoitustankkiin nesteet johdetaan pumppujen avulla kahdesta raa-

ka-ainesäiliöstä. Sekoitustankista valmis nesteseos ohjataan tyhjennysputken kautta ulos järjestelmästä. Putkistoihin on asennettu venttiilejä, joiden avulla voidaan nesteiden virtausta hallita. Sekä pumput, sekoittaja ja lämmitys ovat käyttäjän hallittavissa simulointimallissa sijaitsevan ohjauspulpetin tai fyysisen ohjauspulpetin kautta. Pumput ja venttiilit ovat ON/OFF-tyyppisiä, joten niiden avulla ei pystytä määrittelemään virtausta tarkasti. Prosessilla on kaksi ohjausmuotoa, manuaali- ja automaattireseptiohjaus. Simulointimallissa sijaitsevan ohjauspulpetin sisältä löytyvät moottorien releet sekä moottorinsuojakytkin. Sekoitusprosessin logiikkaohjelma on toteutettu Siemensin Simatic Manager-ohjelmistolla.

### 2.1.1 Simatic Manager STEP 7

STEP 7 on Siemensin valmistama logiikkaohjelmointiin tarkoitettu ohjelmisto, joka mahdollistaa käyttäjälleen erilaisten logiikkaohjelmien toteuttamisen. Ohjelmistoon määritellään projektissa tarvittavat laitteistot, I/O-tiedot, yhteydet sekä rakennetaan tarvittava ohjelma. Ohjelmien luontiin käytetään pääasiassa seuraavia ohjelmointikieliä Ladder (LAD, Instruction (IL) sekä Function Block Diagram (FBD). Simatic Managerin näkymässä vasempaan laitaan on sijoitettu työkaluvalikot ohjelmien erilaisten toimilohkojen luontia varten. Käyttäjän valittavissa on paljon erilaisia toiminnallisuuksia, yksinkertaisten OR-piirien muodostamisesta aina monimutkaisempiin laskureihin. Yksinkertaisen piirin muodostaminen onnistuu valitsemalla esimerkiksi OR-piiri (tunnus:  $\geq 1$ ) ja määrittelemällä siihen tarvittavat tulot (esim. I2.4 ja I1.6), joista toisen asettuessa päällä menee määritelty lähtö päälle (esim. Q8.0).





Kuva 2. Simatic manager perusnäkyä

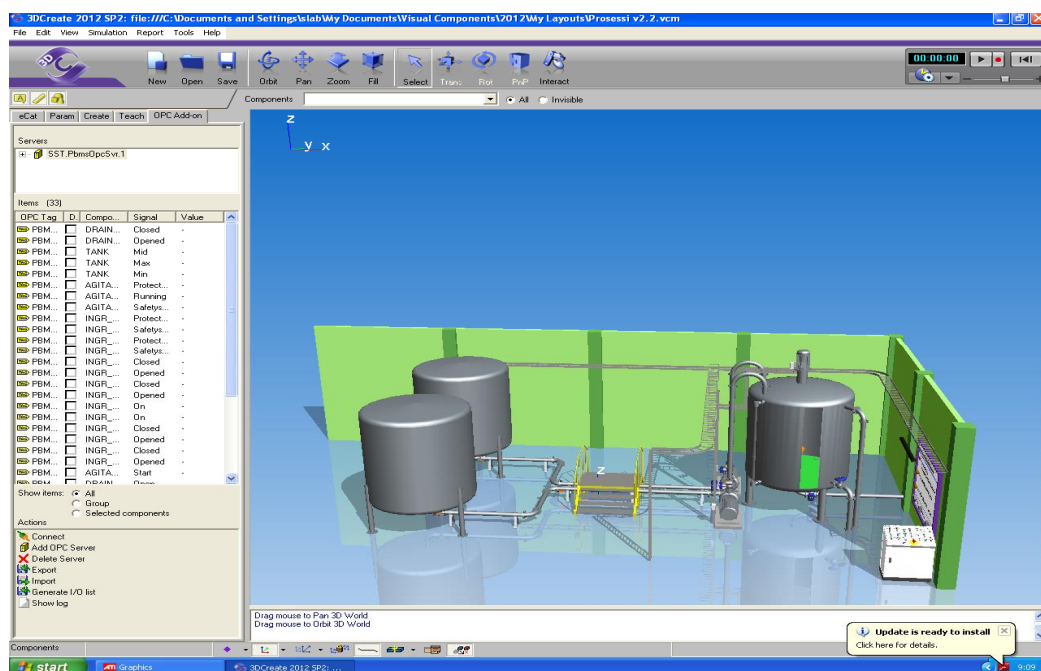
## 2.2 Simulointimalli

Simulointimallin avulla pystytään jäljittelemään käyttäjälle todellinen tai kuviteltu järjestelmä kaikkine laitteineen ja toimintoineen. Simulointimallin avulla käyttäjä pääsee tutustumaan järjestelmään lähemmin eri kuvakulmista, ja hän pystyy näin helpommin hahmottamaan prosessin toimintaa. Simulointimallin kautta käyttäjällä on mahdollisuus koeajaa järjestelmää ja testata sen eri osa-alueita. Simulointia käytetään tuotanto- ja logistiikkaprosessien tutkimiseen ja suunnitteluun. Tämä tehdään tietokoneella tuotetun virtuaalisen mallin kautta (Fimatic www-sivut). Opinnäytetyössä käytetty simulointimalli on toteutettu käyttämällä Visual Components yhtiön 3DCreate-ohjelmistoa.

### 2.2.1 3DCreate

3DCreate on suomessa pääkonttoriaan pitävän Visual Components Oy:n tuottama ohjelmisto, joka mahdollistaa käyttäjän tarpeiden mukaisen mallinnuksen pienistä roboteista, aina suuriin tehdaskokonaisuuksiin ja prosesseihin. 3DCreate-ohjelma mahdollistaa käyttäjälleen jo valmiin projektin mallintamisen tai auttaa uuden vasta

suunnitteilla olevan järjestelmän suunnittelun ja hahmottamisen ennen kalliita rakennusvaiheita.



Kuva 3. Visual Components 3DCreate-ympäristö

## 2.3 Valvomosovellus

Valvomosovellus luotiin Siemens WinCC Flexible-ohjelmistolla, jonka avulla rakennettiin prosessin ohjaukseen graafinen käyttöliittymä. Ohjelmalla rakennettuun valvomosovellukseen toteutettiin ohjauspulpetti, joka ulkonäöltään ja toiminnoiltaan vastaisi mahdollisimman tarkasti olemassa olevaa fyysistä ohjauspulpettia. Valvomosovellukseen toteutettiin myös välilehdet reseptiohjaukselle (käyttäjä voi valita ja muokata reseptejä, joita prosessissa ajetaan), hälytyksille (sisältää tiedon prosessin virhetiloista) ja käyttöohjeille sekä kuvaus järjestelmän toiminnoista (venttiilien ja moottorien tilatiedot). Käyttöliittymässä sijaitsevan ohjauspulpetin kautta käyttäjällä on mahdollisuus valita joko järjestelmän manuaali- tai automaatti/reseptiohjaus.

### 2.3.1 Automaatti/reseptiohjaus

Automaatiohjauksen avulla käyttäjä ohjaa erikseen valitut sekoitusreseptit päälle valitsemalla automaattitila päälle ohjauspulpetista ja valitsemalla haluttu resepti re-

septiohjausvälilehdeltä. Reseptiohjauksessa on käyttäjän mahdollista vaikuttaa raaka-aineiden A ja B määrään, prosessin sekoitusaikaan ja haluttuun lämpötilaan. Autoaattiohjauksen aikana käyttäjä ei pysty vaikuttamaan prosessiin muutoin kuin hätäseis-painikkeen kautta, jonka seurauksena prosessi pysähtyy, kunnes hätäseis kuitataan.

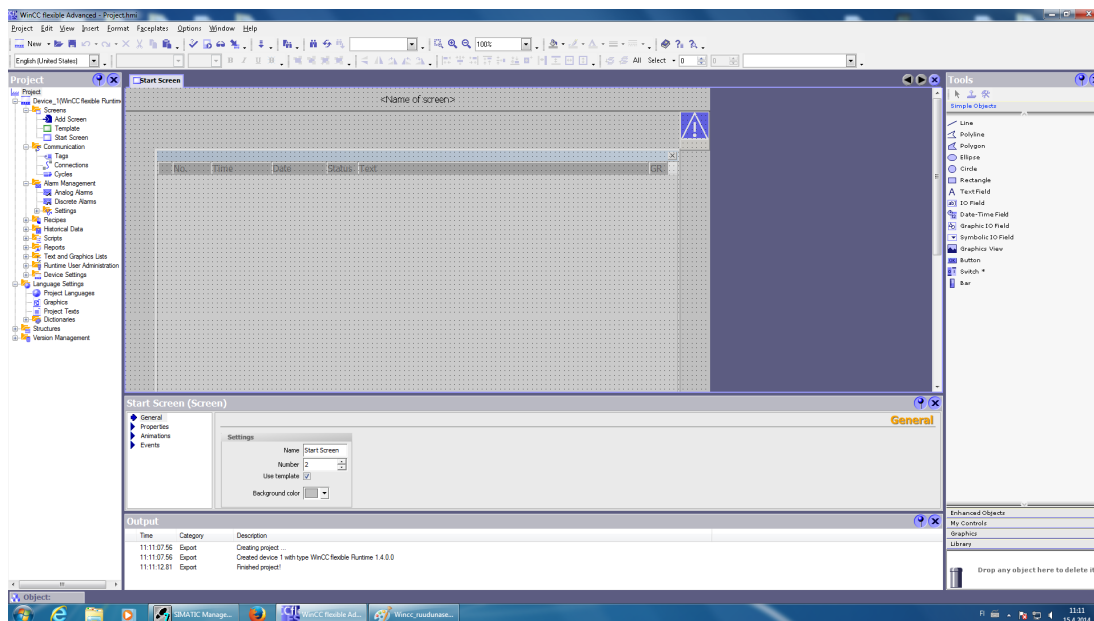
### 2.3.2 Manuaaliohjaus

Manuaalillassa käyttäjän on mahdollista ohjata sekä pumppujen, että sekoittajan moottoreita tai lämmitystä päälle/pois. Ohjauspulpetin kautta on myös mahdollista kuitata moottorien huoltoilmoitukset sekä tyhjentää sekoitussäiliö. Logiikkaohjelmaan on toteutettu estoja, niin jottei käyttäjä pysty ohjaamaan sekoittajaa päälle, kun tankin pinnankorkeus on liian alhainen.

### 2.3.3 WinCC Flexible

WinCC Flexible on Siemensin tuottama Simatic Manageriin liitettävä ohjelmisto, joka antaa käyttäjälleen monipuoliset työkalut erilaisten valvomokäyttöliittymien luontiin. Ohjelman avulla käytävä voi luoda lähes kaikkia valvomossa tarvittavia toimintoja aina yksinkertaisista merkkivaloista ja painikkeista suurempiinkin sovelluskokonaisuuksiin. Käyttöliittymien tekoa helpottaa ohjelmistossa jo valmiina olevat kirjastot, joiden avulla käyttäjä voi helposti poimia ja pudota menetelmällä luoda erilaisia käyttöliittymä näkymiä. Ohjelman avulla voidaan ohjelmoida operointipaneelit pienistä Micro-paneeleista PC-sovelluksiin saakka (Siemens www-sivut).

WinCC-ohjelmiston perusnäkyvä koostuu vasemmassa reunasta löytyvästä projektipuusta jonka kautta hallitaan projektin asetuksia, luodaan sivuja ja hallinnoidaan yhteyksiä. Oikeaan reunaan on sijoitettuna ohjelman työkalu-näkyvä joka sisältää työkalut yksinkertaisten kuviodien luomiseen sekä kirjastot erilaisille valmiille kuvakkeille. Ruudun keskelle on sijoitettu alue, jolle valvomosovelluksen ikkunat rakennetaan. Ruudun yläreunasta löytyvät normaalit valikot joiden kautta voidaan hallita ohjelman toimintoja.



Kuva 4. WinCC flexible perusnäkökulma

### 3 KÄYTTÖLIITTYMÄ

Käyttöliittymä on asia, johon jokainen ihminen törmää nykypäivänä yhä etenevissä määrin. Käyttöliittymän on keino, jonka avulla pyritään toteuttamaan ihmisen ja koneen välinen kommunikaatio. Oli kyseessä sitten tietokone, puhelin, televisio tai vaikkapa jokin nykyaikaisista kodinkoneista, niin lähes aina kyseistä laitetta hallitaan jonkinlaisen käyttöliittymän avulla. Yleisesti tämän päivän kuluttajille suunnatut käyttöliittymät ovat tyyliltään hyvin värikkäitä ja sisältävät paljon liikkuvaa kuvaa ja ääntä, kun taas valvomokäytössä suositaan edelleen yksinkertaisia ja selkeitä malleja.

Käyttöliittymän suunnittelua aloittaessa tulee ottaa huomioon käyttökohde, käyttäjäkunta sekä minkälaiselle näytölle käyttöliittymä tulee, käytetäänkö perinteistä tietokoneen näyttöä vai kosketusnäyttöistä näyttöpäätettä. Puhelimen näytölle tarkoitettu käyttöliittymä ei välttämättä ole paras mahdollinen suurelle tietokoneen näytölle edes skaalattuna. Käyttöliittymän tulee olla selkeä ja käytännöllinen. Visuaalisuutta suunniteltaessa tulee selvittää loppukäyttäjän vaatimuksia, toiveita ja mieltymyksiä. Jos loppukäyttäjien ryhmä koostuu laajemmasta käyttäjäkunnasta, joudutaan usein tekemään kompromisseja ihmisten erilaisten mieltymysten ja kielellisten asioiden vuoksi.

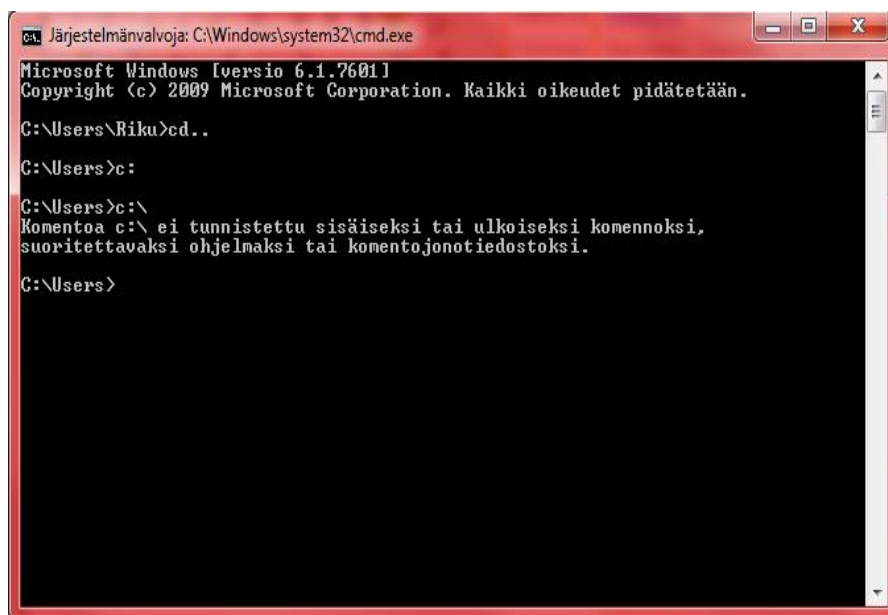
Suunnittelijan tulee myös välttää tilanteita, joissa hänen omat henkilökohtaiset mielityksensä pääsisivät liiaksi vaikuttamaan lopputulokseen. Käyttöliittymän selkeys ja helppo omaksuttavuus on myös tärkeää, kun otetaan huomioon käyttäjäkunnan ikäjakauma ja kokemustaso erilaisten laitteiden käytössä. Tässä osiossa tutustutaan erityylyisiin käyttöliittymiin ja siihen mitä hyvän käyttöliittymän suunnittelussa tulee huomioida.

### 3.1 Käyttöliittymän luokittelu

Käyttöliittymät voidaan jakaa pääpiirteittäin kahteen ryhmään niiden ulkonäön ja toiminnallisuuden kannalta

#### 3.1.1 komentopohjainen käyttöliittymä

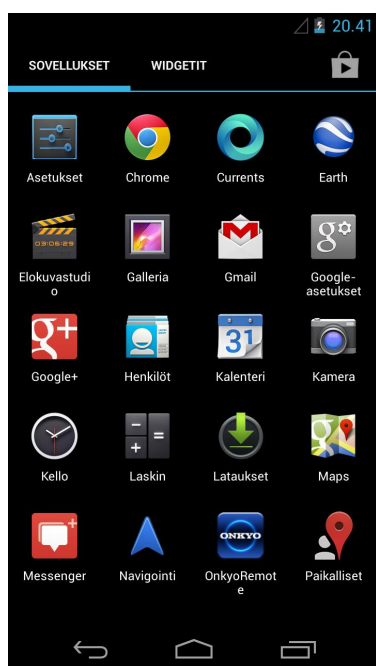
Komentopohjaisessa käyttöliittymässä käyttäjä hallitsee laitteita syöttämällä merkkipohjaisia komentoja komentokehoteeseen ja saamalla vastauksen merkkeinä. Merkkipohjainen käyttöliittymä vaatii käyttäjältään huomattavasti enemmän kuin graafinen. Käyttäjän tulee hallita useita eri komentoja joita vaaditaan toimintojen toteutuksessa. Mahdollisten virheiden etsintä saattaa tuottaa hankaluuksia kokemattomammalle käyttäjälle ja näin ollen työn edistyminen on hitaampaa.



Kuva 5. Komentopohjainen käyttöliittymä

### 3.1.2 Graafinen käyttöliittymä

Graafinen käyttöliittymä avautuu käyttäjän eteen erilaisina kuvakkeina ja elementteinä, joiden avulla käyttäjä pystyy hallinnoimaan haluamiaan toimintoja. Graafisessa käyttöliittymässä käyttäjä antaa komennot yleensä liikuttamalla hiirtä tietokoneen näytöllä ja hiiren nappia painamalla valitsee haluamansa toiminnon. Toinen nykyään jo hyvin yleinen tapa on käyttää kosketusnäyttöä, jossa käyttäjä sormella tai tarkoitukseen soveltuvalla kynällä painelee näytön kuvakkeita ja toimintapainikkeita. Graafiseen käyttöliittymään toiminnoille voidaan usein määritellä myös erilaisia äänitehosteita, joiden avulla tehostetaan käyttöliittymän käytettävyyttä. Graafinen käyttöliittymä on huomattavasti informatiivisempi ja helposti omaksuttavampi kuin komentopohjainen käyttöliittymä. Graafisen käyttöliittymään voi olla mahdollistettu käyttäjän pääsy myös taustalla pyörivään komentopohjaiseen käyttöliittymään, mutta sen käyttö on yleensä hyvin rajoitettua.



Kuva 6. Graafinen käyttöliittymä

### 3.2 Käyttöliittymän ominaisuudet

Käyttöliittymän ominaisuuksiin voidaan luetella selkeys, yhtenäisyys, tehokkuus ja tarkoituksenmukaisuus. Kaikkia edellä mainitut osa-alueet tulee ottaa huomioon

käyttöliittymää rakennettaessa, mikäli lopputuloksesta halutaan käyttäjäkuntaan vetoava.

### 3.2.1 Selkeys

Osana selkeän käyttöliittymän muodostamiseen, kuuluu esitettävien tietojen ja toimintojen asettaminen ruudulla loogiseen järjestykseen. Käyttöliittymän värimaailmaa luodessa, tulee miettiä tarkkaan luottaako selkeisiin sävyihin, vai onko tarpeellista tuoda asioita esille hyvin kirkkaalla värimaailmalla. Kirkkailla väreillä toteutettu käyttöliittymä voi olla raikkaamman näköinen, mutta kääntöpuolella on mahdollinen sekavuus ja joidenkin ihmisten kohdalla kyky havaita tiettyjä värejä. Kielelliset asetukset tulee olla käyttäjäkunnan ymmärrettävissä, niin tekstin kuin symboliikankin puolesta.

### 3.2.2 Yhtenäisyys

Yhtenäisyydellä pyritään osaltaan selkeyttämään käyttöliittymä. Järjestelmässä sijaitsevien samantyyppisten laitteiden ilmentämisessä tulisi aina käyttää samaa tyyllisiä kuvakkeita, joiden toisistaan erottaminen voidaan toteuttaa lisäämällä kuvakkeiden viereen merkintä (esim. M1=moottori1 ja M2=moottori2). Jos käyttöliittymään on toteutettu useampia välilehtiä, tulisi niiden toteutuksessa käyttää yhtenäistä pohjaa, jossa valikkorakenteet säilyvät samankaltaisina ja helposti tunnistettavina.

### 3.2.3 Tehokkuus

Tehokkaassa käyttöliittymässä käyttäjällä on koko ajan mahdollisimman hyvä kuva järjestelmässä tapahtuvista asioista. Tehokkaassa käyttöliittymässä käyttäjältä ei vaadita aikaa tai liiallista teknillistä osaamista tietyn toiminnon hallitsemiseen, vaan kaikki tieto on helposti omaksuttavaa.

### 3.2.4 Tarkoituksenmukaisuus

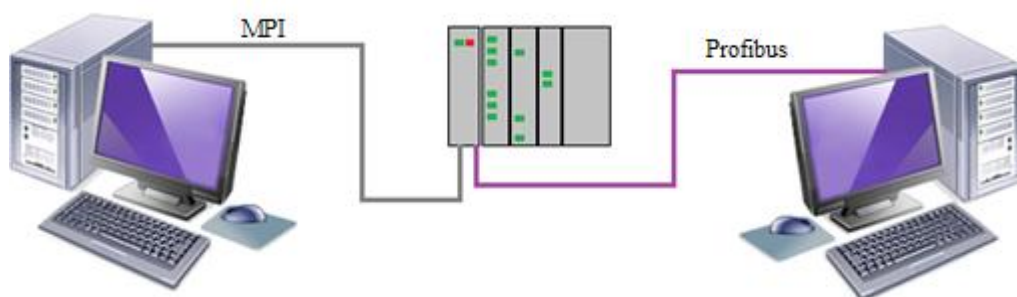
Tarkoituksenmukaisuudella pyritään entisestään selkeyttämään käyttöliittymää karsimalla ruudulta ne tiedot pois joilla ei ole loppukäyttäjän kannalta merkitystä. Kaikki tarpeettomat toiminnot ja ilmoitukset karsitaan pois, jotteivät ne vaikuta käyttäjän keskittymiseen häiritsevästi.

## 4 VALVOMOSOVELLUKSEN TOTEUTUS

Tässä osiossa perehdytään opinnäytetyössä luodun valvomosovelluksen toteutukseen ja WinCC Flexible-ohjelmiston käytön perusteisiin. Aluksi tutustutaan hieman yhteyksiin joiden avulla työn eri osa-alueet liitetään toisiinsa. Lopuksi tutustutaan valvomosovelluksen toimintojen luomiseen. Valvomosovellus toteutettiin itse jo aiemmin käyttöliittymien kurssilla toteuttamani pohjan päälle.

### 4.1 Yhteydet

Erilaisia yhteysmuotoja laitteistossa on kaksi, Profibus (simulointimallin ja logiikan välillä) ja MPI-väylä (valvomosovelluksen ja logiikan välillä).



Kuva 7. Järjestelmän yhteydet



### 4.1.1 MPI-väylä

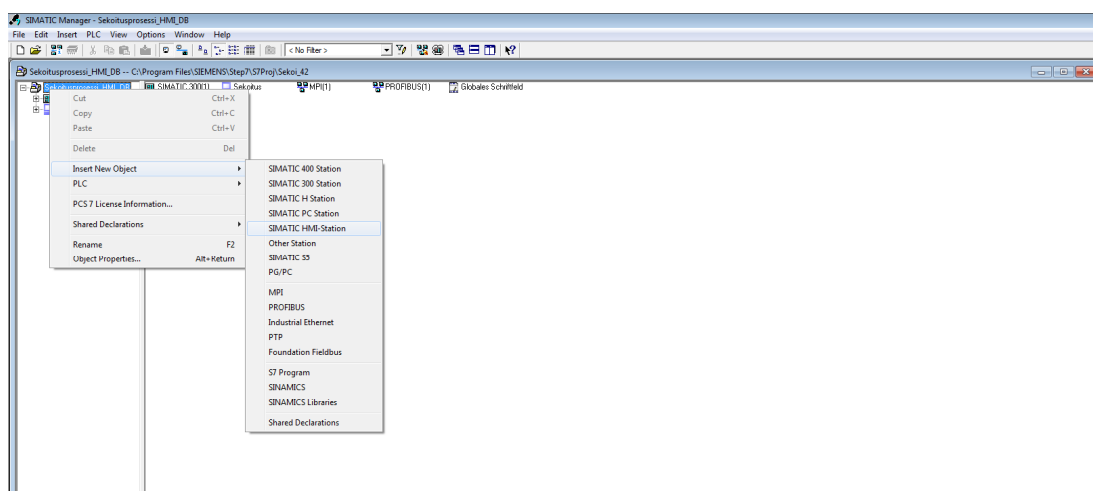
Tämän väylän kautta toteutettiin tietokoneen jossa valvomosovellus ja logiikan välinen kommunikointi. Yhteys toteutettiin suoraan tietokoneen sarjaportin kautta.

### 4.1.2 Profibus

Profibus-väylää käytettiin logiikan ja simulointi-PC:n välisen kommunikoinnin toteuttamiseen. Profibusia käytetään liittämään kentälaitteita, kuten hajautettua I/O:ta tai taajuusmuuttajia, automaatiojärjestelmään. Profibus on avoin kenttäväyläjärjestelmä, jolla on lyhyet vasteajat ja yhteensopivuus IEC 61158:n kanssa (Siemensin www-sivut.)

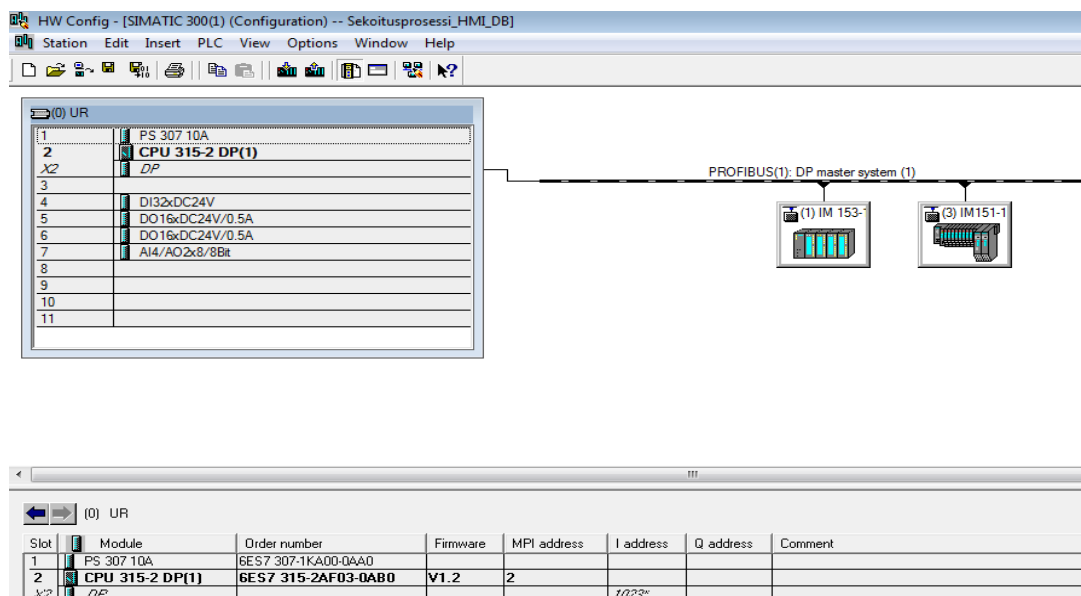
### 4.1.3 STEP7:n JA WinCC:n yhteys

STEP 7:n ja WinCC:n välisen yhteyden muodostaminen aloitettiin lisäämällä sekoitusprosessin logiikkaohjelmaan valvomoasema, johon valvomosovelluksen käyttöliittymä myöhemmin toteutettaisiin. Valvomoaseman lisäys toteutettiin Simatic Managerin kautta valitsemalla sekoitusprosessi ja oikealla hiirenpainikkeelle avautuvasta valikosta insert new object ja Simatic HMI station. Tämä toiminto lisää uuden valvomoaseman prosessiin.



Kuva 8. Valvomoaseman lisäys STEP7

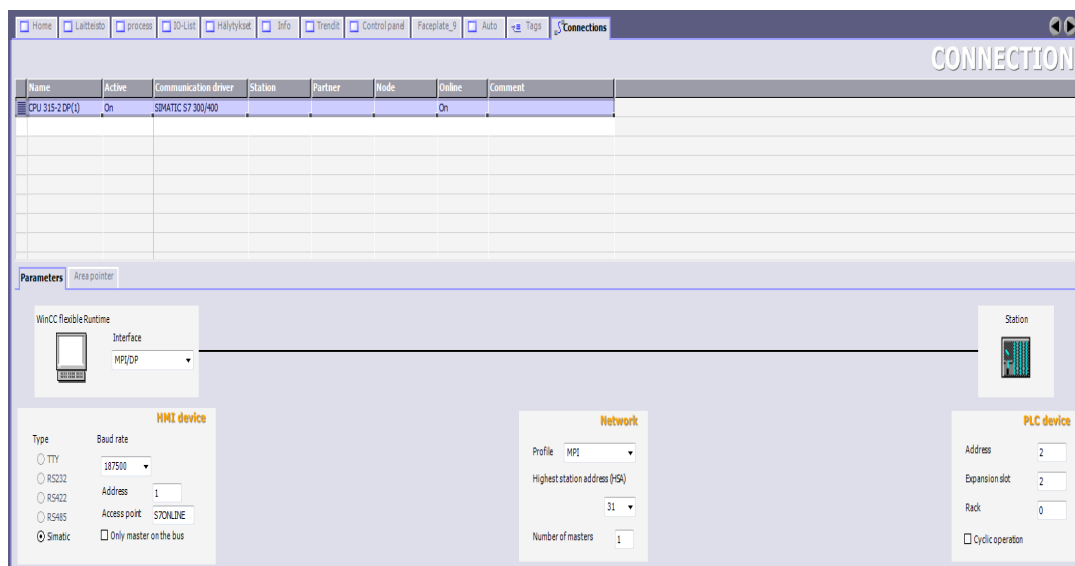
Valvomoaseman lisäyksen jälkeen Simatic Managerin valikkoon ilmestyy valvomoaseman kuvake, josta oikean hiiren painikkeen kautta valitaan open object, joka käynnistää WinCC-ohjelmiston. Seuraavana toimenpiteenä avattiin Simatic Managerin hardware-konfigurointi, jonka kautta tarkastettiin S7-300-logiikka-aseman osoite (MPI address), joka tulisi myöhemmin asettaa WinCC:n puolella vastaavaksi.



Kuva 9. Simatic Manager Hardware-konfigurointi

WinCC-ohjelman puolella yhteyden muodostus suoritettiin projekti hakemiston communication-kohdasta ja valitsemalla sieltä connections. Connections-välilehdelle aukeaa kuva, jossa oikeassa laidassa sijaitsee logiikkaa ilmentävä kuvake (station), vasemmassa laidassa valvomoaseman kuvake (WinCC Flexible) ja niiden välissä valikko, josta valitaan yhteyden profiili (MPI).

Kuvakkeiden yläpuolelta löytyvästä valikosta valitaan halutunlainen yhteystyyppi, joka on tässä tapauksessa SIMATIC S7 300/400. Logiikkakuvakkeen alta vaihdettiin aseman osoitteeksi (address) 2, joka vastaa tässä tapauksessa Simatic Managerista aiemmin tarkastettua asemanumeroa ja vastaavasti valvomoasemalle osoitteeksi valitaan 1.



Kuva 10. Yhteyden määrittäminen WinCC:ssä

#### 4.1.4 Tagit

Tagit ovat keino, joiden avulla voidaan toteuttaa WinCC Flexiblen ja logiikkaohjelman välinen kommunikointi. Koska WinCC flexiblen kautta ei ole mahdollista ohjata logiikan I/O-kortteja suoraan (esim. lähtö I2.1), vaan ainoastaan monitoroida, täytyy käyttäjän luoda tageja, jotka viittaavat logiikkaohjelmaan luotuihin muistipaikkoihin. Tagit tulee määrittää WinCC:n puolella tags välilehdelle ja vastaavat muistipaikat kirjoitetaan Simatic Managerissa symboli-listaan. Tags-välilehdelle tulee kirjata myös jokainen logiikkaohjelman I/O-tieto, josta halutaan saada tietoa käyttöliittymän puolella. On tärkeää varmistaa, että molempien ohjelmien tiedot ovat arvoiltaan vastaavat.

esim.

Tagin nimi: automat\_on

Muistipaikka: M40.1

Datatyyppi: Boolean

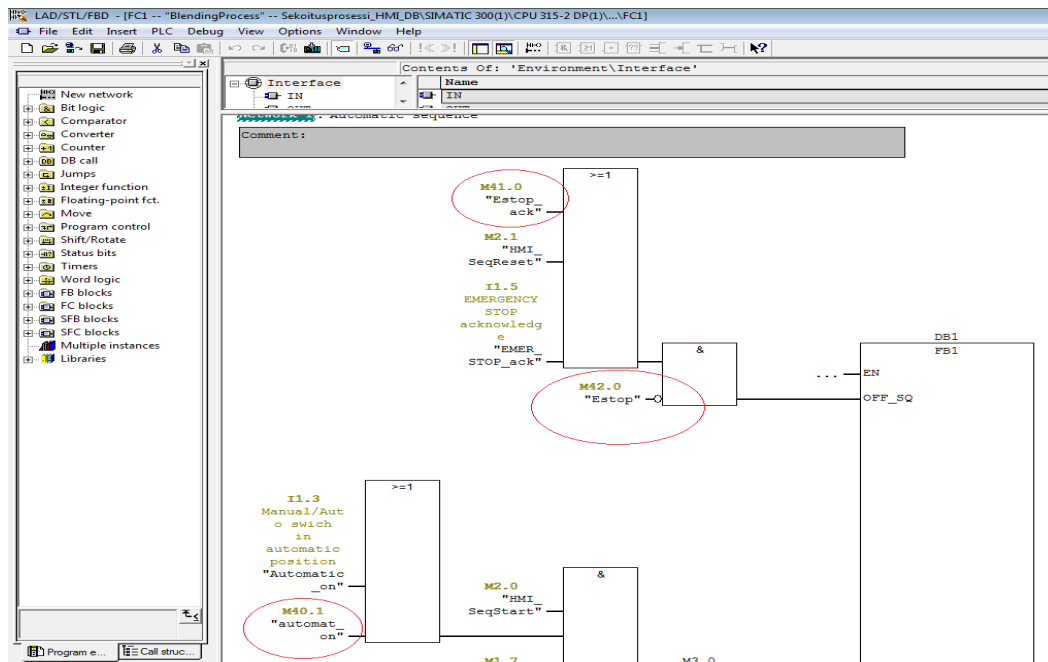
Name	Display na...	Connection	Data type	Symbol	Address
Feed_valve_B		CPU 315-2 DP(1)	Bool	<Undefined>	Q 16.4
Feed_pump_A		CPU 315-2 DP(1)	Bool	<Undefined>	Q 16.2
Feed_valve_A		CPU 315-2 DP(1)	Bool	<Undefined>	Q 16.1
Estop		CPU 315-2 DP(1)	Bool	<Undefined>	M 42.0
Estop_ack		CPU 315-2 DP(1)	Bool	<Undefined>	M 41.0
automat_off		CPU 315-2 DP(1)	Bool	<Undefined>	M 40.2
automat_on		CPU 315-2 DP(1)	Bool	<Undefined>	M 40.1
Enable_DB_Trigger		CPU 315-2 DP(1)	Bool	<Undefined>	M 4.3
DB_Trigger		CPU 315-2 DP(1)	Bool	<Undefined>	M 4.1
maintenance_reset		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.7
stop_agitator		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.6
start_agitator		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.5
stop_feedpump_B		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.4
start_feedpump_B		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.3
start_feedpump_A		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.2
stop_feedpump_A		CPU 315-2 DP(1)	Bool	<Undefined>	M 31.1
drain_valve_open		CPU 315-2 DP(1)	Bool	<Undefined>	M 30.6
drain_valve_close		CPU 315-2 DP(1)	Bool	<Undefined>	M 30.5
tank_heating_off		CPU 315-2 DP(1)	Bool	<Undefined>	M 30.1
tank_heating		CPU 315-2 DP(1)	Bool	<Undefined>	M 30.0
HMI_SeqReset		CPU 315-2 DP(1)	Bool	<Undefined>	M 2.1
HMI_SeqStart		CPU 315-2 DP(1)	Bool	<Undefined>	M 2.0
Seq_Running		CPU 315-2 DP(1)	Bool	<Undefined>	M 1.7
Drain_Step		CPU 315-2 DP(1)	Bool	<Undefined>	M 1.5
Heating_Step		CPU 315-2 DP(1)	Bool	<Undefined>	M 1.4

Kuva 11. Tagit WinnCC flexible:ssä

Statu	Symbol	Address	Data type
	Heating_on	I 12.5	BOOL
	Tank_below_max	I 12.6	BOOL
	Tank_above_min	I 12.7	BOOL
	Tank_not_empty	I 13.0	BOOL
	1Hz_clock	M 0.5	BOOL
	Init_Step	M 1.0	BOOL
	IngrA_Fill_Step	M 1.1	BOOL
	IngrB_Fill_Step	M 1.2	BOOL
	Blending_Step	M 1.3	BOOL
	Heating_Step	M 1.4	BOOL
	Drain_Step	M 1.5	BOOL
	Seq_Running	M 1.7	BOOL
	HMI_SeqStart	M 2.0	BOOL
	HMI_SeqReset	M 2.1	BOOL
	DB_Trigger	M 4.1	BOOL
	Enable_DB_Trigger	M 4.3	BOOL
	M1ManRun	M 10.0	BOOL
	M1Automatic	M 10.1	BOOL
	IngrA_ValvesOpen	M 10.2	BOOL
	DrainOpen	M 10.4	BOOL
	M2ManRun	M 10.5	BOOL
	M2Automatic	M 10.6	BOOL
	IngrB_ValvesOpen	M 10.7	BOOL
	M3ManRun	M 11.0	BOOL
	M3Automatic	M 11.1	BOOL
	M1Fault	M 11.2	BOOL
	IngrA_Fault	M 11.3	BOOL
	M2Fault	M 11.4	BOOL
	IngrB_Fault	M 11.5	BOOL
	M3Fault	M 11.6	BOOL
	M1AutoRun	M 12.0	BOOL
	M2AutoRun	M 12.1	BOOL
	M3AutoRun	M 12.2	BOOL
	VSAutoOpen	M 12.3	BOOL
	tank_heating	M 30.0	BOOL
	tank_heating_off	M 30.1	BOOL
	drain_valve_close	M 30.5	BOOL
	drain_valve_open	M 30.6	BOOL
	stop_feedpump_A	M 31.1	BOOL
	start_feedpump_A	M 31.2	BOOL
	start_feedpump_B	M 31.3	BOOL
	stop_feedpump_B	M 31.4	BOOL
	start_agitator	M 31.5	BOOL
	stop_agitator	M 31.6	BOOL
	maintenance_reset	M 31.7	BOOL
	automat_on	M 40.1	BOOL
	Estop_ack	M 41.0	BOOL
	Estop	M 42.0	BOOL

Kuva 12. Simatic Managerin symbolilista

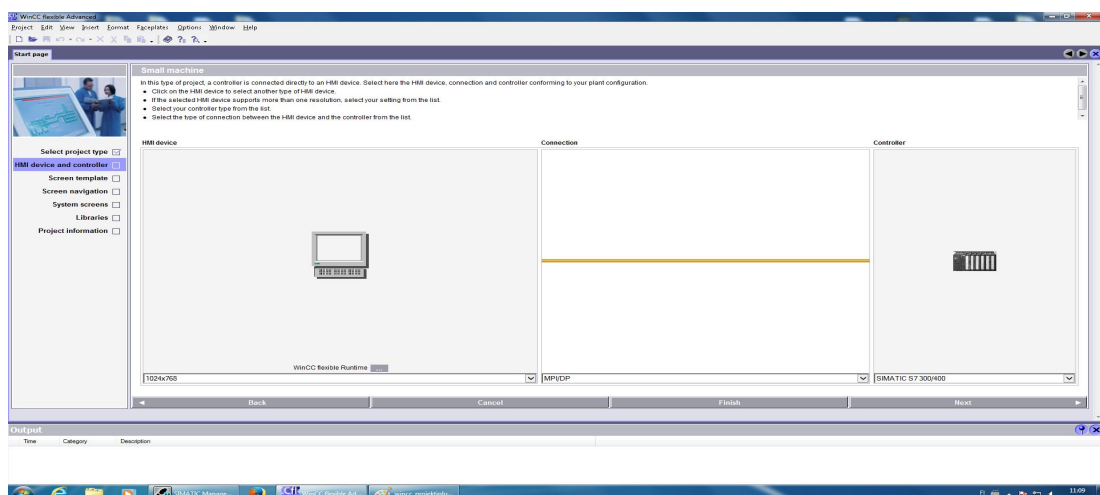
Kun tarvittavat tagit oli luotu ja listattu myös Simatic Managerin symbolilistaan, tuli ne myös liittää logiikkaohjelmaan. Liittäminen tapahtui yksinkertaisesti lisäämällä haluttuun toimilohkoon uusi haara ja asettamalla tarvittava ohjaus. Toimilohkoihin tuli lisätä fyysistä ohjauspulpetin toimintoa vastaava toiminto, jotta ohjaus käyttöliittymästä olisi mahdollista.



Kuva 13. Toimilohkoihin lisättyjä ohjauksia

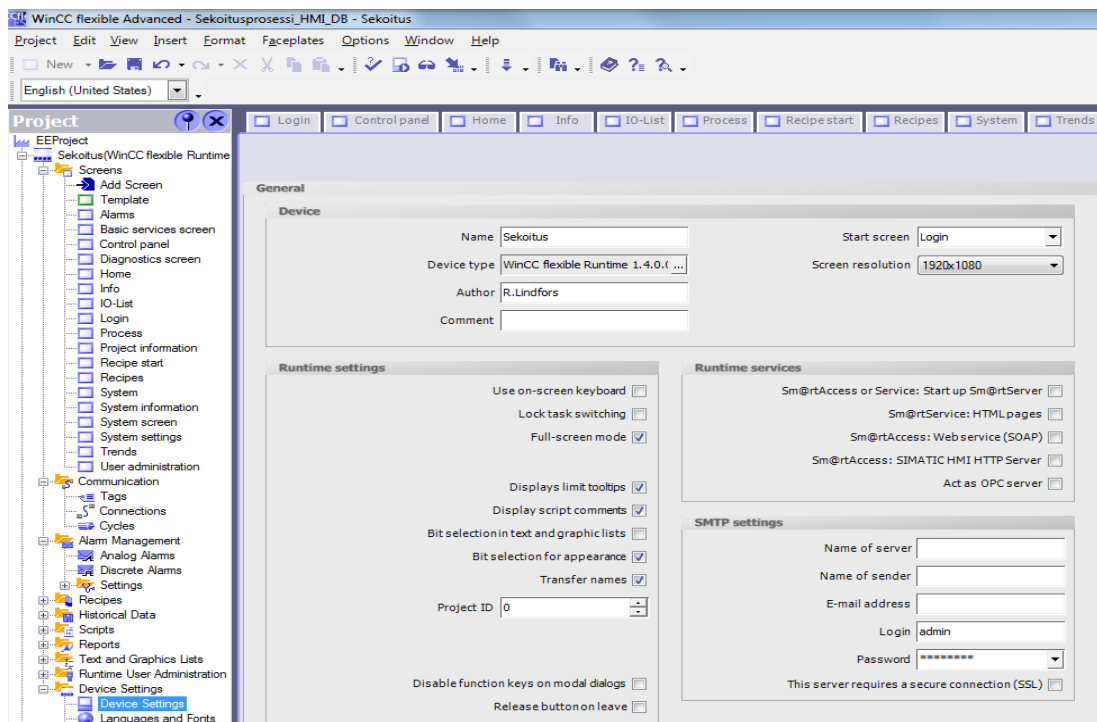
## 4.2 Valvomokäyttöliittymän luominen

Kun WinCC Flexible käynnistetään, avautuu eteen valinta ruutu, jonka kautta on mahdollisuus luoda uusi tai avata jo olemassa oleva projekti. Uuden projektin luoja opastetaan vaihe vaiheelta. Käyttäjä aloittaa valitsemalla haluamansa ohjauspaneeli-tyypin, näytön asetukset, yhteysmuodon ja kirjastot. Halutessaan käyttäjä voi hypätä näiden vaiheiden yli ja muokata asetuksia myöhemmin projekti valikoista.



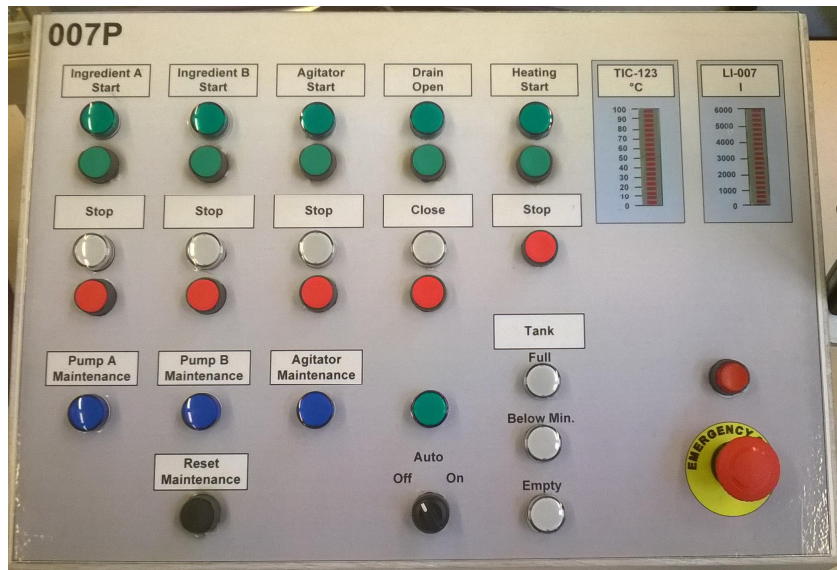
Kuva 14. WinCC Flexible projektin luonti

Käyttöliittymän luominen WinCC Flexible-ohjelmassa aloitettiin määrittelemällä haluttu näytön tyyppi ja resoluutio. Tässä työssä resoluutioksi tarkentui 1920x1080, koska käyttöliittymä toteutettiin 21 tuumaiselle PC:n näytölle. Resoluutio päästiin asettamaan projektivalikon kohdasta Device settings. Samasta valikosta voidaan myös määrittellä käyttöliittymän etusivu (kohdassa Start Screen).



Kuva 15. Näytön resoluution valinta

Tämän toimenpiteen jälkeen luotiin uusi tyhjä ruutu project valikon add screen kohdasta, joka toimi fyysisen ohjauspaneeli mallin pohjana. Ruudulle annettiin nimeksi ohjauspaneeli. Fyysinen ohjauspaneeli sisälsi yhteensä 12 painonappia, 16 toimintovaloa, 2 palkkinäyttöä, hätä-seis painikkeen ja kaksi-asentoisen kytkimen, jotka kaikki jäljennettiin valvomosovellukseen.

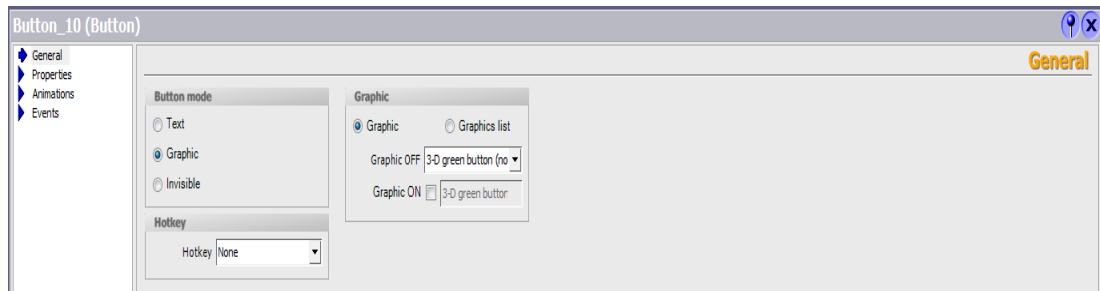


Kuva 16. Fyysinen ohjauspulpetti

#### 4.2.1 Painonappi

Käyttöliittymän painonappien avulla käyttäjän on mahdollista ohjata prosessin eri osa-alueita. WinCC Flexible tarjoaa kirjastoissaan käyttäjälle monia erilaisia painikemalleja esim. pyöreä painonappi tai keinukytkin, joiden avulla käyttöliittymän ulkonäköä voidaan muokata halutunlaiseksi. Tämän opinnäytetyön tapauksessa halusin selkeyden kannalta luoda napeista riittävän kokoiset.

Painonapit luotiin kaikki samalla periaatteella. Kirjastosta valittiin perusmallinen nappi ja se asetettiin tyhjälle ruudulle. Hiiren oikealla napilla tupla klikattiin kuvaketta, jolloin ruudun alareunaan ilmestyy valikko, josta kuvaketta päästään muotoilemaan. General kohdasta asetetaan napille graafinen toiminto. Graphic OFF ruutuun valittiin kirjastosta halutun värinen kuva napista, jota ei ole painettu (button not pressed). Asetettiin Graphic ON kohtaan ruksi ja kirjastosta kuva painetusta napista (button pressed). Events kohdasta valittiin painikkeella varsinainen toiminto. Tässä tapauksessa haluttiin, että kun painiketta klikataan hiirellä, asettuu aiemmin määritelty muistipaikkaan viittaava tagi päälle (esim. M30.0). Vihreät painikkeet laitettiin asettamaan bitti päälle painettaessa (esim. tank\_heating) ja resetoitumaan kun nappi vapautetaan. Punaisille napeille asetettiin stop-toiminto vastaavalla tavalla, mutta asetettava bitti oli käänteinen start-toimintoon verrattuna (esim. tank\_heating\_off).



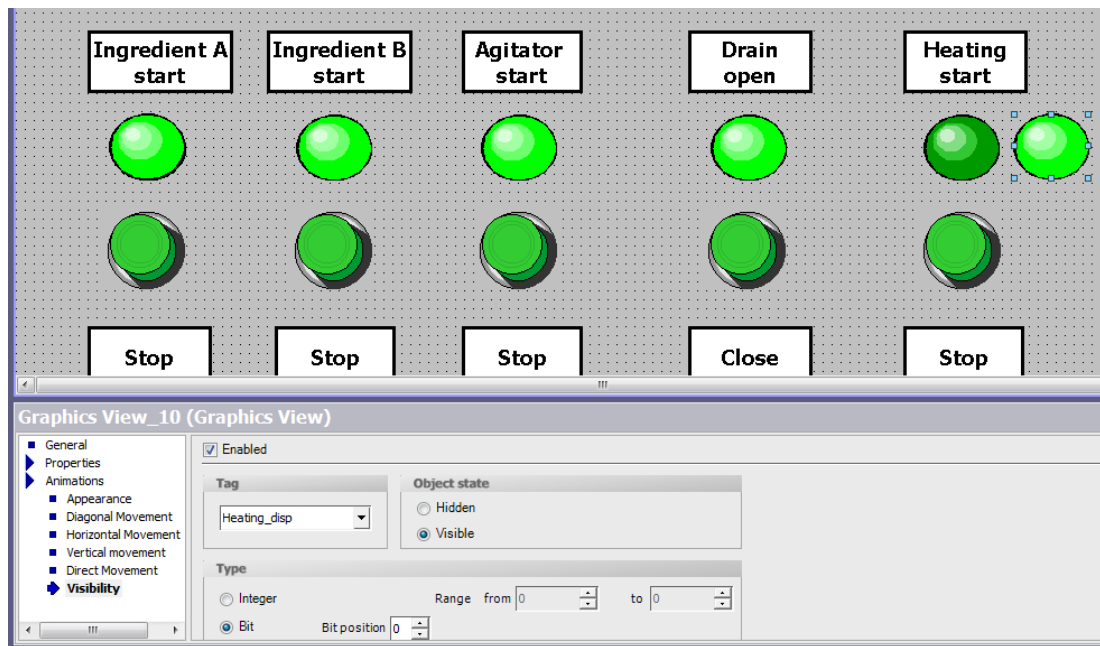
Kuva 17. Kuvan liittäminen painonappiin

#### 4.2.2 Toimintovalo

Käyttöliittymään toteutettavien valojen avulla pyritään antamaan käyttäjälle tietoa prosessissa käynnissä olevista asioista. Valojen avulla voidaan ilmaista prosessissa päällä olevaa toimilaitetta tai vaikka mahdollista virhetilaa. Valoa toteutettaessa tulee miettiä miten valon tulee ilmentyä sen aktivoituessa. Toteutetaanko vain staattinen valo vai halutaanko huomiota herättävä vilkkuva valo.

Valojen luonti toteutettiin lähes samoilla periaatteilla kuin nappienkin. Aluksi valitaan kaksi valokuvaketta (esim. vihreät), joista toinen on tummemman värinen ja toinen kirkas. Valokuvakkeille määriteltiin näkyvyys (visibility) ominaisuudet, joiden avulla voitiin määrittää kulloinkin näkyvissä oleva kuvake (tumma pois päältä ja kirkas päällä). Koska valot ilmaisevat tässä työssä ainoastaan logiikka ohjelman lähdöissä tapahtuvia muutoksia, voitiin niiden näkyvyys toteuttaa niin että kirkas valokuvake on näkyvissä vain kun logiikkaohjelmassa valoa ohjaava lähtö menee päälle (esim. Q8.0). Tupla klikkaamalla valon kuvaketta, avattiin valikko josta päästiin muokkaamaan toimintoja. Valokuvakkeen näkyvyys asetettiin päälle kohdasta visibility (Kuva 18). Enabled ruutuun asetettiin rasti, valittiin tag kohtaan haluttu tagi ja tyypiksi bit. Object state kohdasta valittiin milloin kuvake on näkyvillä ja milloin piilossa. Kirkkaalle valolle valittiin kohta visible, jolloin kuvake tulee näkyviin kun logiikkaohjelmassa kyseinen lähtö asettuu päälle.



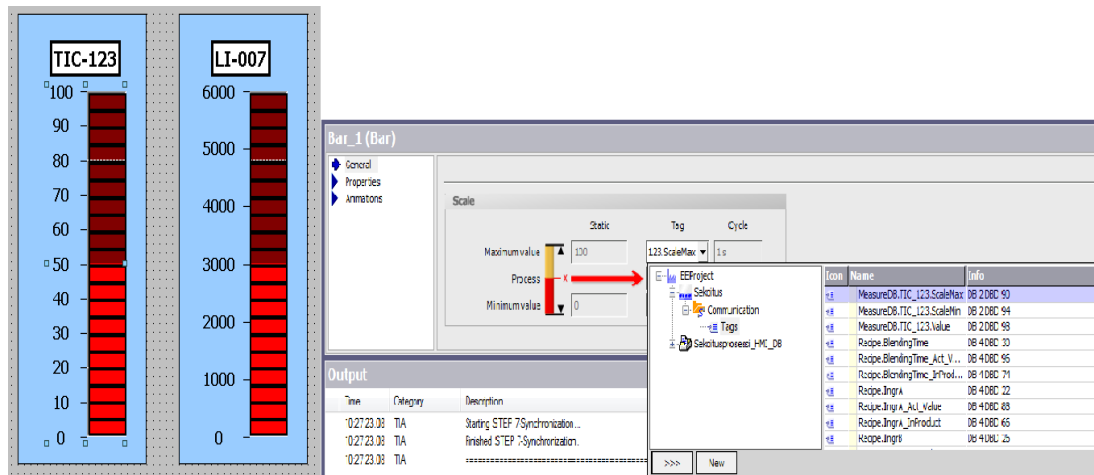


Kuva 18. Valojen asettaminen

#### 4.2.3 Palkkinäyttö

Käyttöliittymässä olevien erilaisten mittarien ja näyttöjen avulla pyritään antamaan käyttäjällä tarkempia tietoja prosessissa tapahtuvista asioista ja järjestelmän tilasta. Näyttöjen avulla pystytään ilmaisemaan prosessin asiat huomattavasti tarkemmin, kuin pelkkien valojen avulla.

Käyttöliittymään toteutettiin kaksi palkkinäyttöä, joiden avulla voitiin tarkkailla sekoitussäiliössä olevan tuotteen määrää litroina ja lämpötilaa celsiusina. Säiliön tilavuus skaalattiin tuhannen litran välein (0-6000l) ja lämpötila kymmenen asteen välein (0-100°C). Palkkinäytöt siirrettiin ohjauspaneeli pohjalle suoraan WinCC:n kirjastosta. Tämän jälkeen tehtäväksi jäi näytön muokkaaminen tarkoitukseen sopivaksi. Tupla klikkaamalla kuvaketta aukeavat valikot, joiden kautta päästiin näyttöä muokkaamaan. Properties kohdassa muokattiin palkin asentoa, värejä ja asteikkoja. General valikon kohdasta palkkinäyttö liitettiin logiikkaohjelman DB-tietoihin.



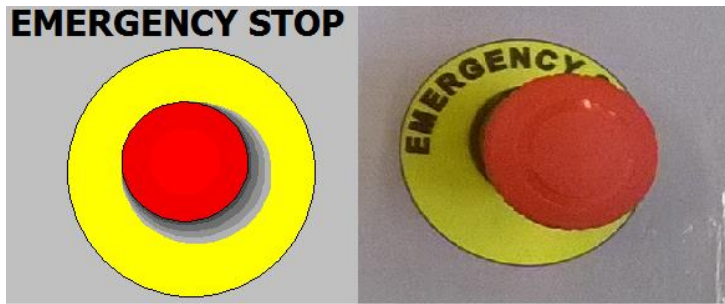
Kuva 19. Palkkinäyttö ja liittymät

#### 4.2.4 Häätä-seis-painike

Hätä-seis-painikkeen avulla voidaan nopeasti pysäyttää, joko koko prosessi tai jokin prosessin osa-alue mahdollisen tapaturman tai vaaratilanteen uhatessa. Painikkeen tulee olla käyttäjien hyvin erotettavissa ja sen luokse on oltava esteetön pääsy. Häätä-seis-painike on yleensä lukittuvaa mallia, jolloin se voidaan vapauttaa vetämällä tai kiertämällä. Yleensä tapauksissa hätä-seis-nappi on vielä kuitattava ennen kuin prosessi on mahdollista käynnistää uudelleen. Useimmin painike on punainen, suurikokoinen ja sen keltaisen alustan yhteydessä lukee "HÄTÄ-SEIS" ö. (Wikipedia www-sivut)

Käyttöliittymään hätä-seis-painike toteutettiin luomalla ensiksi iso keltainen ympyrä, joka toimi napin pohjana. Pohjan päälle asetettiin kaksi neliön muotoista nappia, joiden toiminnot toteutettiin samaan tapaan kuin muidenkin nappien (toinen nappi asettaa bitin ja toinen resetoit bitin). Häätä-seis-napin toteutuksessa toinen nappi jätettiin kuitenkin pienemmäksi, jotta voitiin toteuttaa mielikuva alas painetusta painikkeesta. Molempiin nappeihin vaihdettiin tekstin tilalle kirjastosta pyöreän punaisen napin kuva. Muuttamalla vielä napin tausta läpinäkyväksi, saatiin illuusio pyöreästä napista. Toiminnallisesti Häätä-seis-bitti on päällä aina kun nappi on yläasennossa ja painettaessa se asettaa bitin pois päältä, jolloin prosessi pysähtyy, joten painettaessa isompaa nappia resetoituu bitti ja se asettuu takaisin päälle vasta kun pienempään nappiin vaikutetaan.

Hätä-seis-painikkeen yläpuolelle lisättiin vielä kuittauspainike.



Kuva 20. Hätä-seis-painikkeet (oikealla fyysinen ja vasemmalla virtuaalinen nappi)

#### 4.2.5 ON/OFF-Kytkin

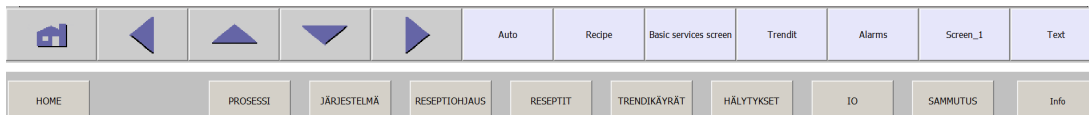
Käyttöliittymässä olevan kytkimen avulla käyttäjä valitsee haluamansa ohjausmuodon. Off-asennossa käyttäjällä on mahdollisuus ohjata prosessin eri tiloja päälle oman valintansa mukaan, ainoastaan reseptien ajo ei ole mahdollista. On-asennossa asettuu automaattitila päälle, joka mahdollistaa reseptien ajon.

Käyttöliittymän kytkin toteutettiin Hätä-seis-painikkeen tavoin lisäämällä kahteen painonappiin kuvat eri asennoissa olevista kytkimistä ja liittämällä kumpaankin nappiin tarvittava tag-tieto.

#### 4.3 Välilehdet

Käyttöliittymä sisälsi valmiiksi erinäisiä välilehtiä, Koska käytin projektin luonnissa aikaisemmin toteuttamaani käyttöliittymä pohjaa. Erilaisia välilehtiä on mm. hälytys- (järjestelmän virhetila ilmoitukset), reseptinohjaus- (käyttäjä voi luoda ja tallentaa erilaisia sekoitusreseptejä) ja järjestelmäikkuna (käyttäjä näkee järjestelmän toimilaitteiden tilatiedot). Jo valmiiden Välilehtien lisäksi toteutettiin muutamia uusia ikkunoita, joihin sijoitettiin käyttäjälle tietoja järjestelmästä ja ohjeita sen käyttöön. Välilehtiin siirrytään joko alitruudun kautta tai jokaisen ruudun alareunasta löytyvien painikkeiden kautta.

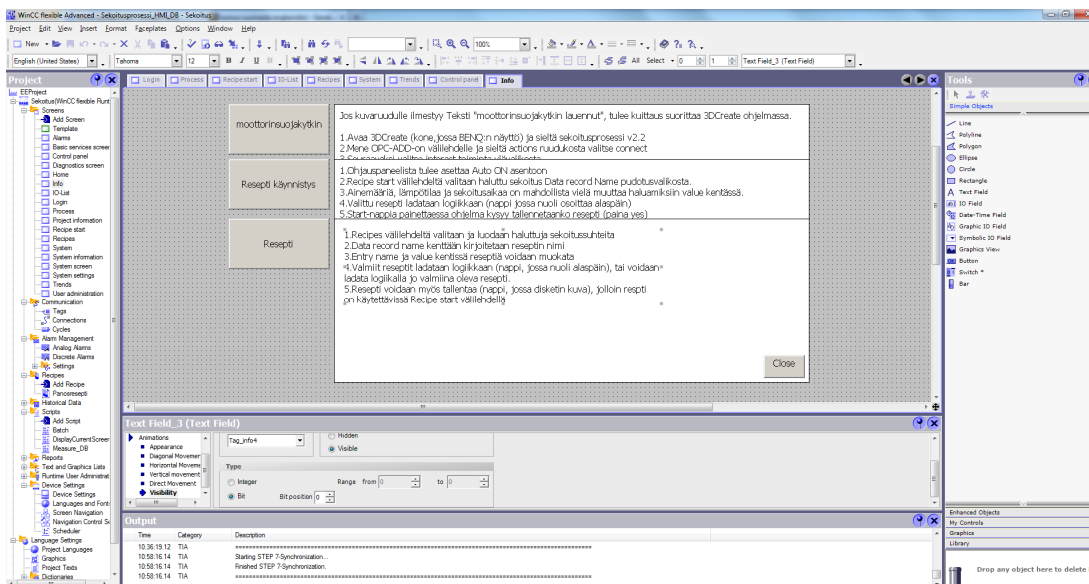
Siirryttäessä tiettyyn välilehteen katoaa ruudun alareunasta kyseisen välilehden painike. Painikkeet luotiin, koska ne olivat mielestäni selkeämpi vaihtoehto ohjelman vakiona tarjoamille nuolipainikkeille.



Kuva 21. Liikkuminen toimintaikkunoiden välillä (ylhällä alkuperäinen)

#### 4.3.1 Info-sivu

Info-sivu toteutettiin helpottamaan käyttäjää operoimaan järjestelmää ja antamaan vinkkejä ongelma tilanteissa. Info ruudut toteutettiin asentamalla painikkeen kuvia pöydälle (isot harmaat ruudut). Jokainen painike asettaa tietyn ohjelman sisäisen bitin päälle. Bitin asettuessa päälle ilmestyy ruudulle valkoinen laatikko, joka sisältää tekstiä ja sulje painikkeen. Valkoinen pohja, siinä oleva teksti ja sulje nappi tulevat näkyviksi kun niihin liitetty bitti (esim. tag\_info4) on tilassa 1. Vastaavasti sulje nappi asetettiin resetoimaan kyseinen bitti, jolloin nappia painamalla vaalea pohja teksteineen katoaa.

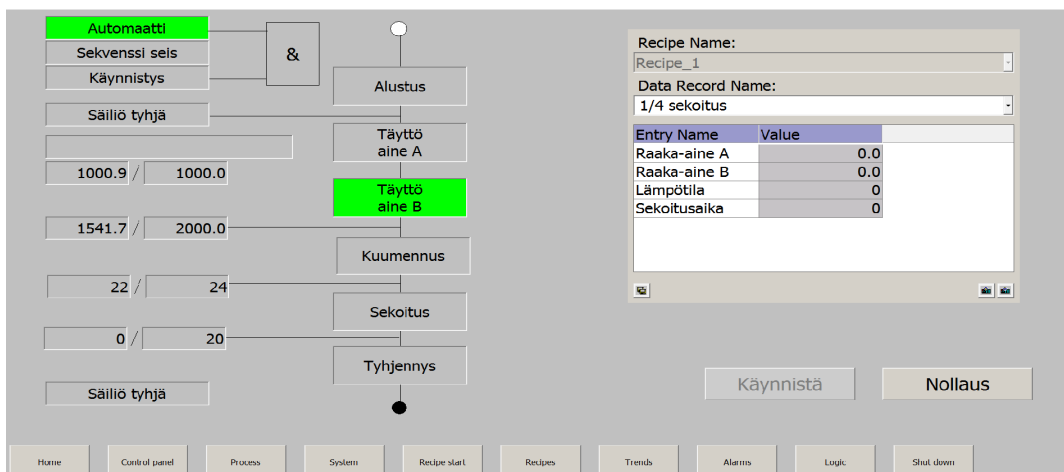


Kuva 22. Info välilehti

### 4.3.2 Reseptinohjaus

Reseptinohjaus välilehden vasempaan laitaan toteutettiin yksinkertainen kaavio prosessin vaiheista, joka sisältää tietoja käynnissä olevasta prosessin vaiheesta. Ruudun oikeaan reunaan asetettiin valinta-ikkuna reseptien muokkaamista, tallentamista ja lataamista varten. Valinta-ikkunasta käyttäjä voi muokata valitsemansa reseptin raaka-aine määriä, lämpötilaa ja sekoitusaikaa, muuttamalla value kentän arvoja. Käyttäjä lataa haluamansa reseptin logiikalle (pieni kuvake, jossa nuoli alaspäin) ja painaa käynnistä nappia (automaattitila tulee olla asetettuna ohjauspaneelistä).

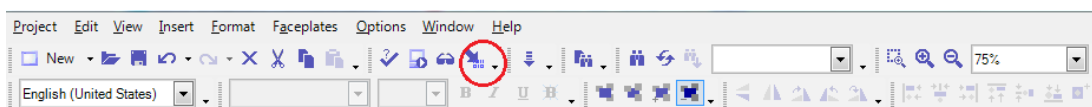
Tämän jälkeen prosessi alkaa pyöriä ja käyttäjälle havainnoidaan kulloinkin prosessin vaihe vihreäksi vaihtuvilla ruuduilla.



Kuva 23. Reseptinohjaus

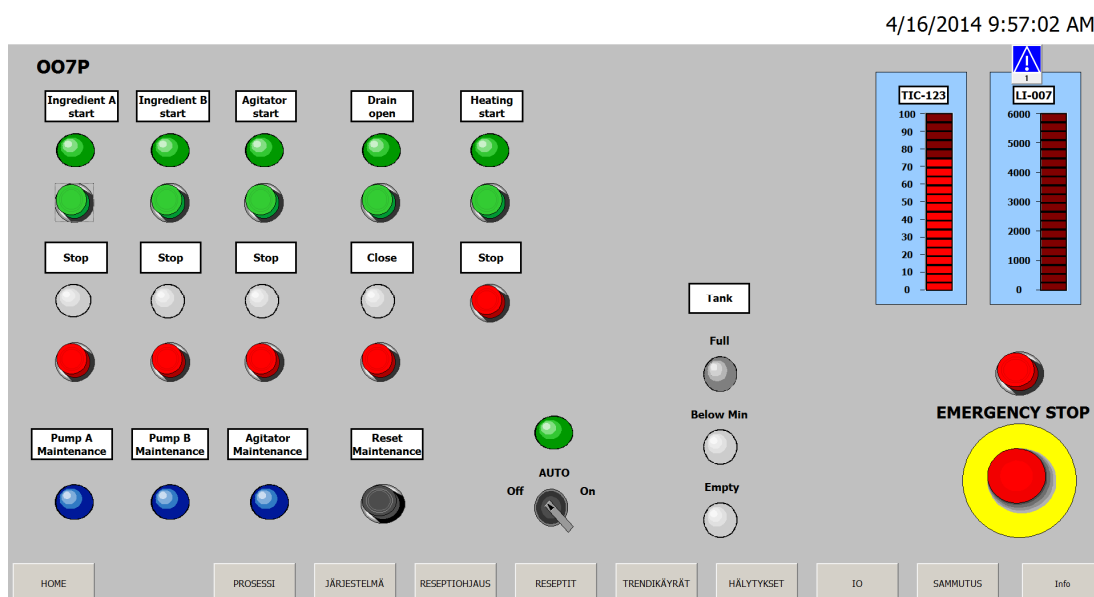
## 5 KÄYTTÖLIITTYMÄN TESTAUS

Käyttöliittymän testaus aloitettiin jo rakennusvaiheessa testaamalla jokaisen napin toiminto erikseen. WinnCC Flexible:n yläreunasta löytyy painike, jonka avulla voidaan luotu sovellus käynnistää simulaatio tilaan, jossa on mahdollisuus mm. asettaa haluamiaan bittejä päälle. Jokainen nappi ja valo testattiin erikseen, jotta voitiin varmistaa, että jokainen niistä oli liitetty oikeaan tagiin.



Kuva 24. Simuloinnin käynnistys

Varsinainen toiminnallisuuden testaus aloitettiin kun käyttöliittymään oli mallinnettu kaikki tarvittavat toiminnot ja ominaisuudet. Jokainen toiminto testattiin WinnCC flexible:n Runtime-tilassa. Kaikkien ominaisuuksien toiminnot varmistettiin sekä logiikkaohjelman puolelle, että fyysisen ohjauspaneelin kautta. Käyttöliittymästä testattiin sekä manuaali- että automaati/reseptiohjaus tilat. Automaattitilassa luotiin useita sekoitusreseptejä, jotka ajettiin läpi, samalla tarkkaillen käyttöliittymää ja sekoitusprosessin simulointimallia



Kuva 25. Valvomosovellukseen toteutettu ohjauspulpetti

## 6 YHTEENVETO

Opinnäytetyön aiheena oli toteuttaa valvomosovellus, joka mahdollistaisi simuloidun sekoitusprosessin ohjauksen. Opinnäytetyön tekeminen oli hyvin opettavainen prosessi, koska sen aikana oli mahdollisuus tutustua tarkemmin moniin eri osa-alueisiin. Työssä pääsin tutustumaan prosessin toimintaan, niin simulaation, kuin myös logiikkaohjelman tasolla ja sain paljon uutta tietoa käyttöliittymän suunnittelusta.

Mikään työn osa-alueista ei aiheuttanut ylitse pääsemätöntä vaikeutta toteutuksen kannalta. Suurin ongelma muodostui sopivan valvomo-ohjelmiston valinnassa. Työn eri vaiheet oli suhteellisen helppo toteuttaa ja yhdistää toimivaksi kokonaisuudeksi, vaikka minulla ei ollut aikaisempaa työkokemusta automaatioalalta ennen ammatti-

korkeakoulun aloitusta. Koulun aikana käymäni kurssit tarjosivat riittävän tiedon tämän kyseisen projektin toteutukseen.

Käyttöliittymän mahdollisina tulevaisuuden kehittämiskohteina voisi pitää graafisen ilmeen hienosäätöä, sekä verkon kautta tapahtuvan valvonnan ja ohjauksen toteuttamista. Myös useamman välilehden näyttäminen samaan aikaan ruudulla voisi olla yksi tulevaisuuden päivitys kohteista. Tällöin toisin jouduttaisiin muuttamaan radikaalista jokaisen välilehden tarjoamaa visuaalista ilmettä, jotta ne mahtuisivat yhdelle sivulle.

## LÄHTEET

Heimbürger, H. Markkanen, P. Paunonen, H. Sundqvist, M. Norros, L. & Savioja, P. & Tommila, T. 2011. Valvomo-Suunnittelun periaatteet ja käytäntö. Suomen automaatioseura

Joensuu yliopiston www-sivut. Viitattu 14.4.2014

[http://cs.joensuu.fi/~jimmonen/gkl\\_moniste/gkl\\_v202.html](http://cs.joensuu.fi/~jimmonen/gkl_moniste/gkl_v202.html)

Fimatic www-sivut. Viitattu 21.4.2014

<http://www.fimatic.fi/simulointi.aspx>

Siemens www-sivut. Viitattu 23.4.2014

[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automatiotekniikka/kayttoliittymat/ohjelmistot/paneelien\\_ohjelmointi\\_wincc\\_flexible.php](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automatiotekniikka/kayttoliittymat/ohjelmistot/paneelien_ohjelmointi_wincc_flexible.php)

Siemens www-sivut. Viitattu 25.4.2014

[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automatiotekniikka/teollinen\\_tiedonsiirto\\_esim\\_profinet/profibus.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automatiotekniikka/teollinen_tiedonsiirto_esim_profinet/profibus.htm)

Wikipedia www-sivut. Viitattu 24.4.2014

<http://fi.wikipedia.org/wiki/H%C3%A4t%C3%A4-seis-painike>