



Pauli Kauppinen

Siemens PCS7 WinCC-mukautetut kuvakkeet

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Koneautomaatio

Insinööriyö

10.5.2024

Tiivistelmä

Tekijä: Pauli Kauppinen
Otsikko: Siemens PCS WinCC-mukautetut kuvakkeet
Sivumäärä: 23 sivua + 5 liitettä
Aika: 10.5.2024

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Konetekniikan tutkinto-ohjelma
Ammatillinen pääaine: Koneautomaatio
Ohjaajat: Lehtori Antti Liljaniemi

Tämän opinnäytetyön tavoitteena oli luoda Siemens PCS7 WinCC-ajokuvaan kevyemmät versiot automaattisesti luoduista lohkokuvakkeista prosessivalvontakäyttöön sekä kerätä yleisimmin käytetyistä lohkokuvakkeista muokatut helpommin käytettävät versiot omaan tyyppikirjastotiedostoon.

Työssä kerrotaan aluksi vektorigrafiikasta ja WinCC:n yleisimmistä työkaluista ja kuinka ne toimivat, jonka jälkeen siirrytään passiivisista aktiivisiin objekteihin. Yksinkertaisimmat aktiiviset objektit ovat vain liitetty signaali, mutta monimutkaisemmat voivat sisältää useita signaaleja tai kirjoituskoodia. Valmiiden kuvakkeiden muokkauksesta Siemensillä on olemassa oma ohje, johon lähteissä on linkki.

Lopputuloksena on kaksi tiedostoa: APL-lisätiedosto ja projektiaputiedosto, joista lopputuloksissa on käyttöohje. Liitetiedostoissa on ohjeineen Visual Basic -kirjoituskoodi lohkokuvakkeiden sijaintien tallennukseen ja palauttamiseen ajokuvissa.

Avainsanat: Siemens PCS7 WinCC HMI Advanced Process Library mukautetut kuvakkeet

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Pauli Kauppinen
Title: Siemens PCS7 WinCC Custom Objects
Number of Pages: 23 pages + 5 appendices
Date: 30 April 2024

Degree: Bachelor of Engineering
Degree Programme: Mechanical Engineering
Professional Major: Machine Automation
Supervisor: Antti Liljaniemi, Senior Lecturer

The Goal of this Thesis was to create a helpfile for Siemens Process Control System 7 projects with lighter alternative objects for control room screens to monitor processes and to create a library for the most commonly used customized block icons for control screens.

First, the work examines vector graphics and most common WinCC tools and how they work, after those passive and active objects are presented. Simplest active object is just signal connected to the object, but more complicated objects may have several signals or scripts codes. Siemens has their own manual for customizing generated objects, which is linked in the references of this thesis.

As result there are two files, one for APL extension and other for project helpfile with instructions for use. The Appendices have Visual Basic script for block saving and retrieving block icon positions in runtime pictures.

Keywords: Siemens PCS7 WinCC HMI Advanced Process Library
Custom Objects

Sisällys

Lyhenteet

1	Johdanto	1
2	Lähtökohdat ja tavoitteet	2
3	Vektorigrafiikka	3
4	PCS7 WinCC	4
4.1	WinCC-piirustustyökalut	4
4.2	Väripaletti	5
4.3	Kerrokset	5
5	Objektit	7
5.1	Passiiviset objektit	7
5.2	Aktiiviset kuvaobjektit	8
5.2.1	VB / C-kirjoituskoodi	9
5.2.2	Dynaamiset toiminnot	11
5.2.3	Dynaamisten toimintojen taustakoodi	13
6	Kuvakkeiden muokkaus	15
7	WinCC Ajonäkymä	18
8	Lopputulos	19
9	Yhteenveto	21
	Lähteet	23

Liitteet

- Liite 1: Siemens WinCC-väripaletti
- Liite 2: Valvomokuvakevertailu
- Liite 3: Kuvakkeiden paikoitus VB-koodi
- Liite 4: Kuvaketyypitiedosto
- Liite 5: Projektiaputiedosto

Lyhenteet

PCS7	Process Control System 7 on Siemensin valmistama prosessin ohjausjärjestelmä
HMI	Human Machine Interface, eli käyttöliittymä
I/O	Input / Output, tulo ja lähtö
PLC	Programmable Logic Controller on ohjelmoitava logiikkajärjestelmä, jolla ohjataan yleensä laitetta tai pientä järjestelmää
SCADA	Supervisory Control and Data Acquisition, automaatiojärjestelmä
DCS	Distributed Control System, automaatiojärjestelmä
WinCC	Siemensin visualisointijärjestelmä, jota voidaan käyttää tietokoneen näyttöpäätteellä, HMI-paneelilta tai jopa tabletilta
WinCC RT	WinCC RunTime on järjestelmän ajonäkymä, josta voidaan ohjata ja tarkkailla laitteita.
SVG	Scalable Vector Graphics tarkoittaa grafiikkaa, jossa objektien ominaisuudet ja muodot esitetään matemaattisin funktioin koordinaatistossa
OS	Operator Station on operaattoriasema, jolla käytetään automaatiojärjestelmää
ES	Engineering Station on insinööriasema, jolla luodaan ja muokataan automaatiojärjestelmää
AS	Automation Station on automaatiojärjestelmän prosessori

CFC	Control Flow Chart on tehokas graafinen ohjelmointikieli logiikalle
SFC	Sequential Function Charts on ohjelmointikieli sekvensseille.
PI-kaavio	Putkitus- ja instrumentointikaavio on piirustustyyppi, jota käytetään kuvaamaan prosessilaitteistoja
Faceplate	Objektin tieto ja ohjausikkuna
APL	Advanced Process Library on WinCC-kuvakkeiden mallikirjasto
DOL	Tulee sanoista Direct OnLine ja on moottorin tähti-/kolmiokäynnistin
VSD	Tulee sanoista Variable Speed Drives eli taajuusmuuttaja
MonAnaL	Monitor Analog Large, analogitulonkuvakelohko
MonDiL	Monitor Digital Large, digitaalitulonkuvakelohko
MotL	Motor Large, päällä / pois päältä -moottorikuvakelohko
MotSpdCL	Motor Speed Control Large, taajuusmuuttajamoottorinkuvakelohko
MotSpdL	2-vauhtimoottorinkuvakelohko, hidas ja nopea
OpDi01	Operate Digital, digitaalilähdönkuvakelohko
PIDConL	PID Control Large, PID-säätimenkuvakelohko
VlvL	Valve Large, auki/kiinni-venttiilinkuvakelohko
VlvAnL	Valve Analog Large, analogiventtiilinkuvakelohko
VlvMotL	Valve Motor Large, moottoriventtiilinkuvakelohko
Vlv2way	Valve 2 way, 3-tieventtiilinkuvakelohko

PCS7DiIn Digital Input driver, digitaalitulonajurilohko

PCS7DiOu Digital Output driver, digitaalilähdönajurilohko

PCS7AnIn Analog Input driver, analogitulonajurilohko

PCS7AnOu Analog Output driver, analogilähdönajurilohko

CMT Control Module Type, ohjausmoduulityyppi

PDL WinCC-grafiikkatiedosto pääte

EMF Enhanced Metafiles, paranneltu metatiedosto

XML Extensible Markup Language on merkintäkielten standardi, joka määrittää tietojen merkintämuodon loogisella rakenteella

CSV Comma-Separated Values, tiedostomuoto, jota käytetään taulukko-
muotoisen datan tallentamiseen tekstimuodossa

1 Johdanto

Tämän insinööriyön idea lähti asiakastyöstä, jossa Siemens Process Control System 7 (PCS 7) -järjestelmän valvomon päätasonpäätekuvat piti piirtää ja ohjelmoida käsin, koska asiakas ei halunnut käyttää pääkuivissa automaattisesti luotuja kuvakkeita "Block Icon". Miksi järjestelmän valmiita kuvakkeita ei haluttu käyttää, johtui projektin laajuudesta ja siitä, että alkuperäiset ikonit ovat raskaita ja hitaita ladata, kun kuvista siirrytään toiseen. Monessa PCS7-projektissa on tullut vastaan, että asiakas tai loppuasiakas on halunnut jotain erilaista, mitä vaikeakirjastosta ei sellaisenaan löydy, joten on pitänyt muokata kuvakkeista perinteisempiä kuvakkeita. Tähänkin on erilaisia tapoja tehdä, kuten muokata vaikeakirjaston kuvakkeista sarja halutuilla merkeillä tai poistaa merkit kokonaan ja lisätä suuntanuoli taustalle tai hyödyntää esimerkiksi tekemiäni aputiedoston kevyempiä vaihtoehtoisia kuvakkeita.

Viimeisimmässä projektissa oli myös muutamia asioita, joihin tehtiin omia apuvälineitä muun muassa mittapalkkiobjektien mitoittamiseen näyttöruutukuvissa.

Aluksi on hyvä ymmärtää mitä ja miten vektorigrafiikka sekä kerrokset toimivat sekä aikaisempaa ohjelmointikokemusta esimerkiksi Visual Basic -ohjelmoinnista. Erilaiset käyttöliittymäohjelmointikokemukset ovat olleet eduksi myös PCS7:n WinCC-ohjelmoinnissa samankaltaisten työkalujen ja objektien vuoksi. Aikaisemmasta toimintolohko-ohjelmoinnista on myös apua, kun haetaan objekteille käytettäviä tunnuksia.

2 Lähtökohdat ja tavoitteet

Tavoitteena on tehdä kuvakkeiden lisäkirjasto yleisimmillä muokatuilla kuvakkeilla, jota voidaan käyttää eri PCS7-projektien oman APL-kirjastotiedoston rinnalla ja sisältää kopioitavat helposti käyttöön otettavat vaihtoehtoiset kevyemmät kuvakkeet.

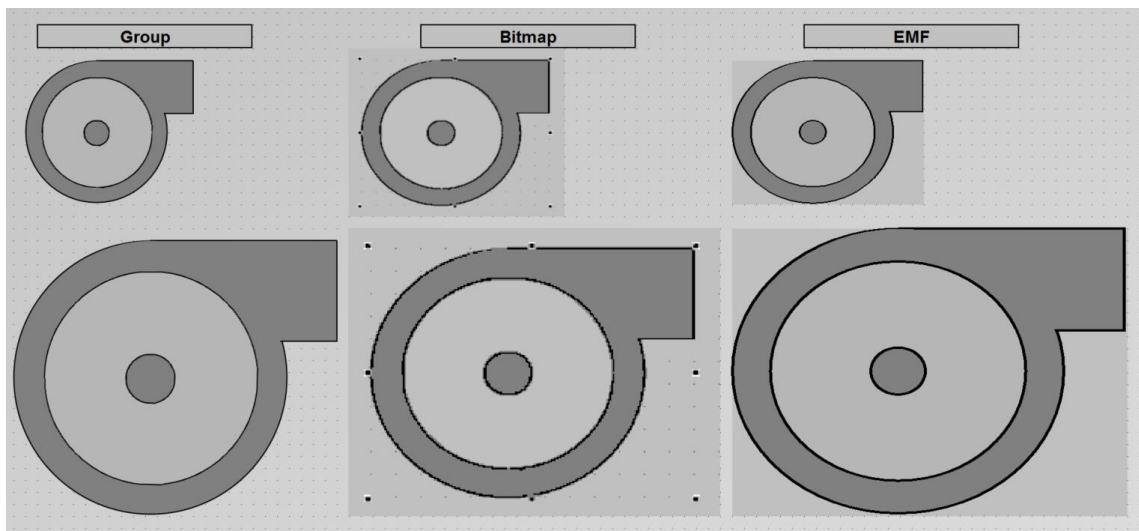
Opinnäytetyö oli tarkoitus tehdä asiakkaalle ja oikeiden laitteistojen kanssa, mutta maailman tilanteen vuoksi ei uutta tilausta tullutkaan ja päädyin virtuaali-koneeseen ja ohjelmiston demoversioon. Ohjelmisto on hieman vanhempi versio: Siemens PCS7 9.0 (julkaistu 2017) ja Siemens WinCC 7.4. Opinnäytetyöhön on käytetty Siemensin vakiomalliesimerkkejä toimilohkoille ja generoiduista kuvakkeista tehdään omat kevyemmät versiot. Näihin versioihin tulee liittää operaattoreille muitakin tärkeitä tietoja kuten käyntitieto. Esimerkiksi, onko toimilaitte automaattilla vai käsikäytöllä ja saako säädin- tai säätöventtiiliasetuksen ulkoisesti vai sisäisesti. Pinnanmittaus voidaan näyttää numeerisesti sekä palkkina.

Ohjelman ohjelmointi voidaan toteuttaa muutamalla eri tavalla. Kaavioiden ja muiden tiedostojen nimet tulee olla yksilöllisiä, samoin kaavion sisällä lohkojen on oltava yksilöllisesti nimettyjä. Lohkoilla voi siis olla samoja nimiä, mutta niiden pitää sijaita eri kaavioissa. Pienempiin kokonaisuuksiin voidaan ohjelmoida yhden kaavion alle useampia laitteita, mutta suuremmissa ohjelmistoissa on suositeltavaa ohjelmoida yksi kaavio laitetta kohden. Yhden laitteen kaavioon tulee esimerkiksi digitaalitulokuvakelohkon lisäksi digitaalitulonajurilohko (PCS7DiIn). Keskisuurissa ja suurissa projekteissa käytetään ohjausmoduulityyppejä (CMT), joissa tehdään mallikaavio ja tämän jälkeen voidaan ajaa ohjelmaan mallikaavion mukaisia kaavioita taulukosta tuotavalla XML- tai CSV-tiedoston avulla.

3 Vektorigrafiikka

Vektorigrafiikalla tarkoitetaan grafiikkaa, jossa objektien ominaisuudet ja muodot esitetään matemaattisin funktioin koordinaatistossa. Vektorigrafiikkaobjektin koko on helposti muutettavissa ilman, että laatu huononee toisin kuin bittikartta-kuvissa. Esimerkiksi ympyrälle määritetään koordinaatistossa piste ja halkaisija, kun taas neliölle tulee lähtöpisteen lisäksi leveys ja korkeus. Esimerkin pumpun kuvaan on piirretty nelikulmio ja neljä ympyrää. Lisää aiheesta Siemensin verkkoaineistossa kuinka tehdä omia lohkokuvakkeita.

Kuvasta (Kuva 1, s. 3) näkee, kuinka keskimmäisen kuvan ympyröiden reunat alkavat muuttumaan laatikkomaisiksi suurennettaessa, kun alkuperäinen vektorigrafiikalla piirretty ryhmä pysyy samanlaisena. EMF-tiedostoksi käännettynä muoto pysyy, mutta kuvaksi käännettynä tulee tausta näkyviin. Tämä on mahdollista muuttaa läpinäkyväksi WinCC-editorissa ja siksi olisi taustan värin hyvä olla projektin muista väristä poikkeava ja vakiona WinCC-kuvaobjektien läpinäkyvyysväri on vaaleanpunainen värikoodilla #FF00FF.



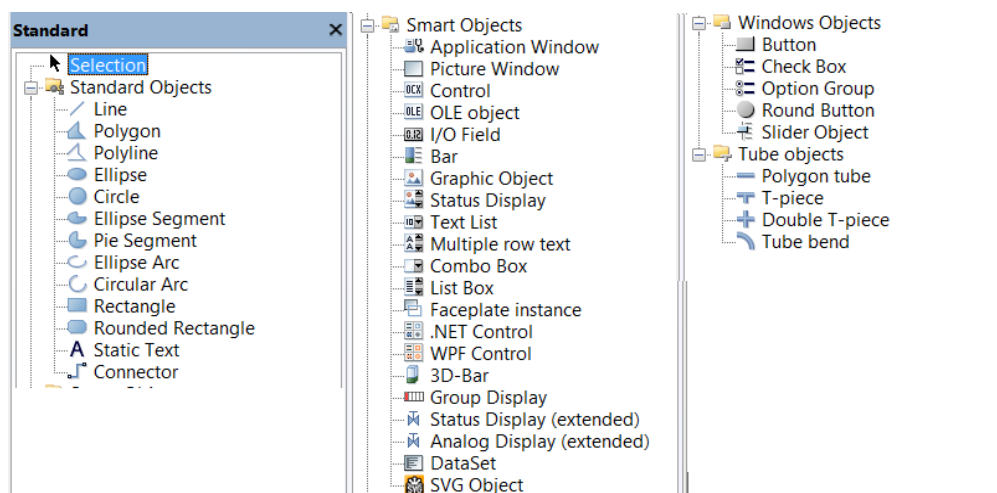
Kuva 1. Vektorigrafiikan ja bittikarttakuvan erot

4 PCS7 WinCC

Siemensin Simatic WinCC-automaation visualisointijärjestelmä tuo automaatiojärjestelmän signaalit helpommin luettavaan muotoon, vaikka kokenut operattori teoriassa pystyisi lukemaan suoraan signaalilistaltakin. Visualisointijärjestelmiin piirretään kuvia laitteista ja laitteistoista ja myös ohjelmoidaan aktiivisia objekteja, kuten painikkeita, valoja tai pintapalkkeja, joiden avulla operattorilla on helpompi valvoa ja ohjata prosessia WinCC-ajonäkymässä (WinCC Runtime).

4.1 WinCC-piirustustyökalut

WinCC-editorin piirustustyökaluvalikko "View" valikosta "Standard"-työkalut vakiopaikka WinCC-editorin oikeassa laidassa. (Kuva 2, s. 4) Ensimmäisenä tavallinen valintatyökalu (hiiren osoitin). Toisena tavalliset vektorigrafiikkatyökalut ja kolmantena "Visual Studio":sta tuttuja älykkäitä objekteja ja neljäntenä perus Windows-objektit ja viimeisenä toisenlainen putkienpiirtotyökalu.



Kuva 2. WinCC-vakiotyökalut

Muitakin työkaluja löytyy esimerkiksi viivan tyyliin, paksuuteen tai pätyihin tai muun kuvion täyttöön liittyen. Yläpalkeista löytyy vielä muita työkaluja objektien järjestelyyn tai ominaisuuksiin. Samassa kerroksessa voi olla useita päällekkäisiä objekteja, silti on suositeltavaa käyttää useampia kerroksia.

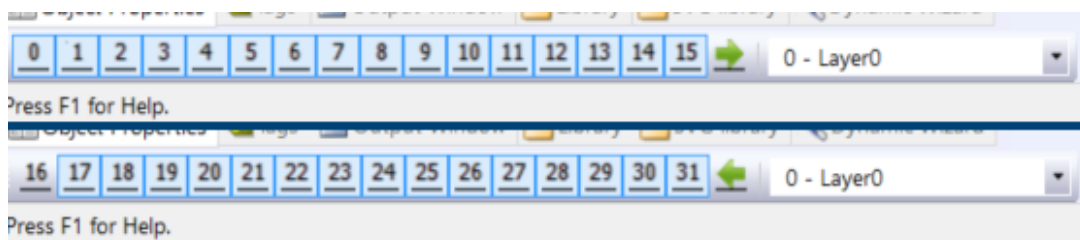
4.2 Väripaletti

Liitteissä (Liite 1) on WinCC:ssä käytettävä väripaletti. Tähän voidaan syöttää väreit joko poimien pipetillä, vetopalkeilla arvoilla 0–255 tai HTML-koodina. Väriin valinnan jälkeen väri voidaan tuoda vielä omaan palettiin, josta väri on helposti käytettävissä myöhemmin. Väriin tuonti palettiin tapahtuu painamalla palettiin painiketta ”To Palette” ja valitsemalla haluttu lokero. Väripaletin värejä kannattaa käyttää erilaisille materiaaleille ja jos myöhemmin halutaan vaihtaa materiaalin väriä, niin voidaan se nopeasti vaihtaa koko projektiin paletin lokeron väriä muuttamalla.

Esimerkkinä valitaan kirkkaanpunainen (HTML FF0000) kohtaan 33 ja nimetään tämä kohta SEIS. Tehdessä aktiivista objektia ja valittaessa laitteen käyntitiedon nolaksi (0), paletin 33-väri voidaan myöhemmin helposti vaihtaa toiseen väriin ja kaikkiin laitteisiin valitsemalla paletin kohtaan uusi väri.

4.3 Kerrokset

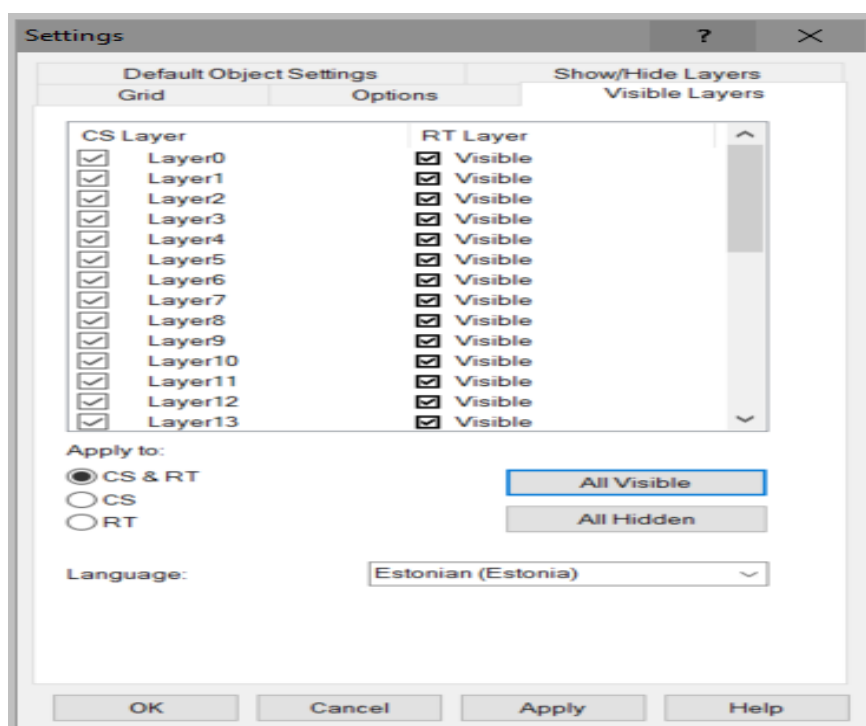
Siemens PCS7 WinCC:ssä on yhteensä 32 kerrosta, joista kerros 0 on alimmainen ja 31 on päällimmäinen (Kuva 3, s. 5). Kun objektit päätyvät päällekkäin, niin kerroksissa ylimmäinen objekti on näkyvä ja käytettävissä. Pienessä yksinkertaisessa projektissa voidaan rakentaa kaikki 0-kerrokselle, mutta suuremmissa projekteissa, joihin tulee paljon objekteja, olisi hyvä hyödyntää kerroksia.



Kuva 3. Kerrospalkki

Kerrosraskista voidaan valita aktiivinen kerros oikean laidan "Combo box"-valinnasta sekä editorissa näkyvät kerrokset 0 - 31 voidaan valita päälle tai pois päältä. Aktiivista kerrosta ei voida piilottaa.

Mikäli ajoikkunaan tahdotaan poistaa näkyviä kerroksia, on se taas valittava ylävalikon kautta "View" ja "Layers..." kautta ja poistaa RT:n alta valintamerkki (Kuva 4, s.6).



Kuva 4. Kerrokset-ikkuna

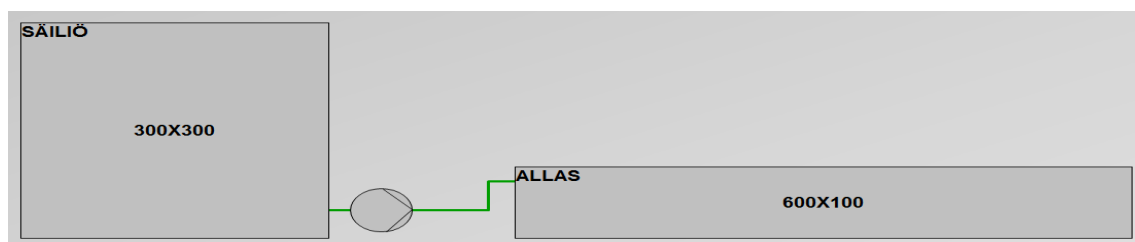
Esimerkkinä kerrosten käytöstä, WinCC-editoriin voidaan jättää viesti myöhempäälle muokkausta varten. Taustan harmaa levy laitetaan ylimpään kerrokseen ja tämän alla olevaan kerrokseen sijoitetaan tekstikenttä, johon viesti kirjoitetaan. Tämän jälkeen editorissa piilotetaan ylimmäinen kerros, jolloin viesti näkyy editorissa ja ei-ajokuvassa. Tästä löytyy malliobjekti liitteestä 5. Oikeasta reunasta "Peitelevy", josta näkyy editorissa musta teksti keltaisella pohjalla kerroksessa 30 ja tämän päällä on harmaa levy kerroksessa 31, joka on WinCC-editorissa piilotettu ja WinCC-ajonäkymässä näkyvissä.

5 Objektit

WinCC-objektit voidaan jakaa aktiivisiin ja passiivisiin. Passiiviset ovat vektorigrafiikalla toteutettuja muotoja eri kerroksissa ja aktiivisiin objekteihin on ohjelmoitu erilaisia ominaisuuksia, joista yleisin on tilanäyttö. Yksinkertaisimmassa tilanäytössä on vain päällä (vakiona vihreä) tai pois päältä (vakiona harmaa). Generoiduissa ikoneissa on myös STOP- tai vikatilassa punainen ja simulaatiossa keltainen. PCS7:ssa kaikista ohjelman laitteista esimerkiksi moottorikuvake luodaan automaattisesti ohjelmankäännössä (Compile). Lohkokuvakkeet tulevat vastaaviin kuviin "Plant View" -kuvakansiossa kuin kaaviot ovat ohjelmakansiossa. Ensin pitää kääntää kaaviot ja tämän jälkeen ajokuvat.

5.1 Passiiviset objektit

Passiiviset graafiset objektit luodaan piirustustyökalujen vakio-objektit "Standard Objects" -työkaluilla (Kuva 2, s. 4). Viivatyökalut 2 pisteen välinen viiva, monipisteviiva ja "Connector" piirtää viivan kahden pisteen väliin ja jos alku- ja loppupisteet ovat eri tasossa, niin yhdistysviivan puoleen väliin tulee tasokorjaus. Kolme erilaista laatikkotyökalua on tavallinen suorakulmio ja pyöristetyillä kulmilla sekä monikulmio. Tekstityökalu muodostaa myös nelikulmion, vaikka onkin tarkoitettu tekstiin, voidaan sitä käyttää myös tavallisen nelikulmion sijasta. Lopuksi on vielä ympyrä ja ellipsi sekä näiden osa- ja kaariversiot.



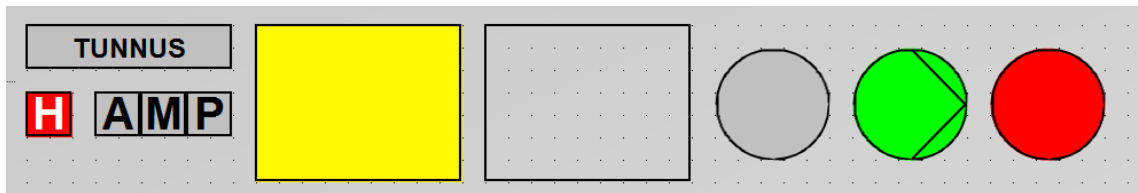
Kuva 5. Esimerkki: säiliö, pumppu ja allas

Kuvan esimerkissä (Kuva 5, s. 7) on 4 kappaletta tekstiä, 2 monipisteviivaa ja 1 ympyrä. Vihreä viiva kuvastaa vesiputkea ja toinen viiva on ympyrän sisällä pumpun tunnusmerkkinä. Säiliö ja allas ovat tekstejä ja niiden keskellä on toinen teksti, joiden reunaviivan paksuus on vähennetty nolnaan. Pumpun kuvan päälle

voisi tuoda luodun lohkokuvakkeen tai tehdä ympyrästä dynaaminen kuvake liittämällä pumpun käyntitieto ympyrän taustaväriin valintaan.

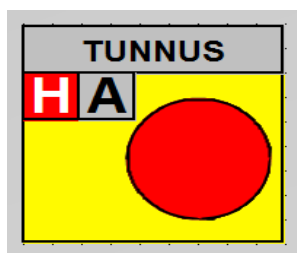
5.2 Aktiiviset kuvaobjektit

WinCC-kuvaobjektitkin voivat olla useissa kerroksissa, joita voidaan käyttää, kun suunnitellaan omia kuvaobjekteja. Esimerkiksi kehysruudut voisivat olla kerroksessa 1, laitetunnus kerroksessa 2, harmaa, vihreä ja punainen näyttö kerroksissa 3–5. Automaattikäyttö-, käsikäyttö- ja paikalliskäyttönäytöt kerroksessa 6 ja hälytysmerkki kerroksessa 7 (Kuva 6, s. 8).



Kuva 6. Oman kuvakkeen osat

Näin esimerkiksi käyntitiedon ja vikatiedon samanaikainen päällä olo näyttäisi punaisen eikä vihreää käyntistatusta. Tavallisista objekteista saadaan tehtyä aktiiviset objektit joko kirjoituskoodeilla(5.2.1 s.9) tai dynaamisilla toiminnoilla (5.2.2 s. 11)



Kuva 7. Oma kuvaobjekti

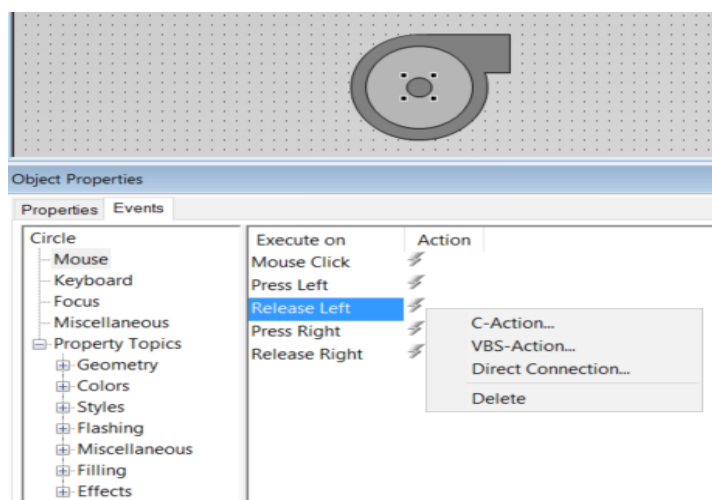
Liitteissä (Liite 2) on tuotu yleisimmät projekteissa käytetyt kuvakkeet, oikealta puolelta löytyy vakio WinCC-kuvakkeet ja vasemmalta puolelta niitä jäljittelevät omat kevyemmät versiot. Tehdyissä projekteissa suurin osa käytetyistä moottoreista on ollut joko taajuusmuuntajalla "VSD" (MotSpdCL) tai käynnistimellä

”DOL” (MotL) -moottoreita. Venttiileistä auki/kiinni (VlvL) ja analogiventtiilit (VlvAnL), sekä moottoriventtiilit (VlvMotL). Tarkkailusta digitaalituloihin (MonDiL) ja analogiarvoihin (MonAnaL) tai säädin (PIDConL) näyttää säätöarvon lisäksi myös analogituloarvon. Useimmat käskyt voidaan toteuttaa digitaalilähdöllä (OpDi01) tai lohkon liitetyllä Windows-painikkeella. Muitakin kuvakkeita on, mutta niitä ei ole ollut suurissa määrin käytössä.

5.2.1 VB / C-kirjoituskoodi

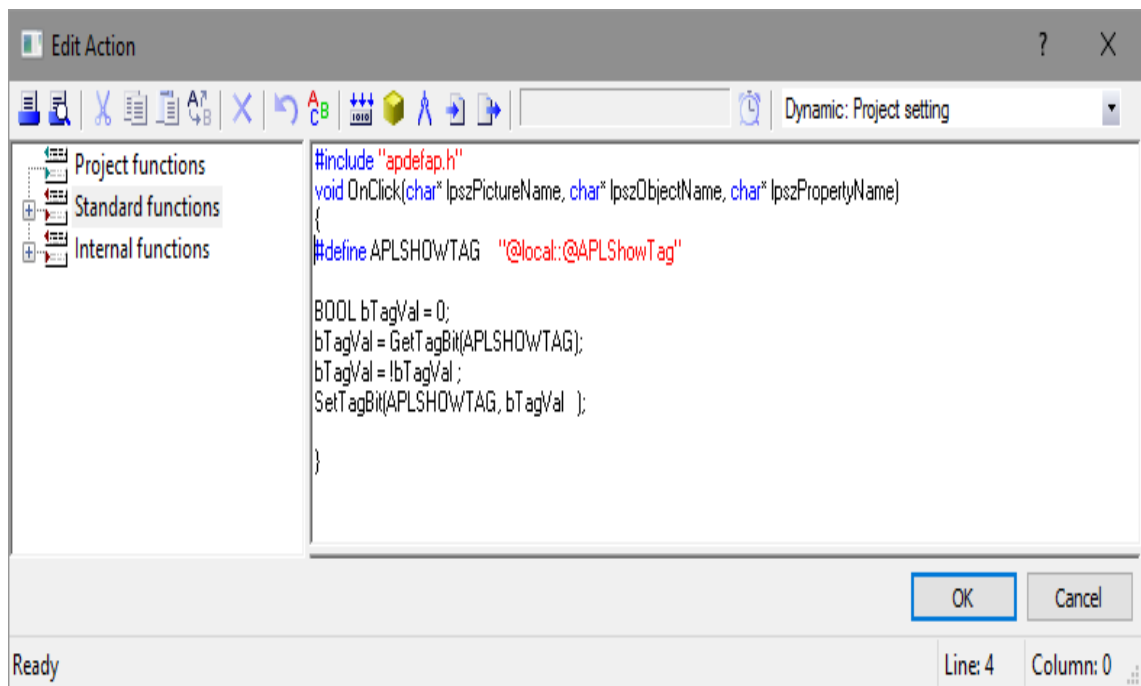
Visual Basic- tai C-kirjoituskoodilla saadaan tehtyä monimutkaisempia toimintoja kuin tavalliset dynaamiset toiminnot. Kirjoituskoodit lisätään valitsemalla objektin ominaisuudet-ikkunasta ominaisuudet sijasta ”Events”-välilehti. Hyvän ohjelmointitavan mukaan on suositeltavaa lisätä toiminnot tapahtumaan, kun painike vapautetaan eikä painettaessa, tällöin vahinkoklikkaus on vielä peruttavissa, jos painiketta ei ole päästetty irti.

Kirjoituskoodien avulla saadaan myös monimutkaisempia toimintoja aikaiseksi esimerkiksi, jos halutaan piilottaa painike, josta saadaan vaikka käyttöstatistiikka käyttöönnotossa, jota ei haluta normaalioperaattoreiden käyttöön. Ensimmäinen tällainen kirjoituskoodipainike, jonka olen nähnyt, oli kahden objektin väliin näkymätön painike, mitä pystyi käyttämään vain, jos painoi näppäimistöä ALT+VAIHTO ja hiiren oikeaa painiketta.



Kuva 8. Kirjoituskoodien ja suoratoimintojen lisäys

Muita esimerkkejä kirjoituskoodien käytöstä on useamman toiminnon samanaikainen aktivointi tai objektin ohjausikkunan "Faceplate" avaaminen. C-Action ja VBS-Action valintojen alla on oma editori-ikkuna, joissa "Script"-koodi kirjoitetaan. "Direct Connection" -avulla saadaan esimerkiksi painikkeeseen "näytä ponnahdusikkuna" -toiminta. Mallina C-kirjoituskoodi näyttä/piilota-tunnukset painikkeen koodi, jolla muutetaan järjestelmä merkin @APLShowTag-tilaa painikkeella. Tätä on käytetty myös oman kuvakearkiston vaihtoehtoisten kuvakkeiden tunnuksissa.



Kuva 9. C-kirjoituskoodi malli

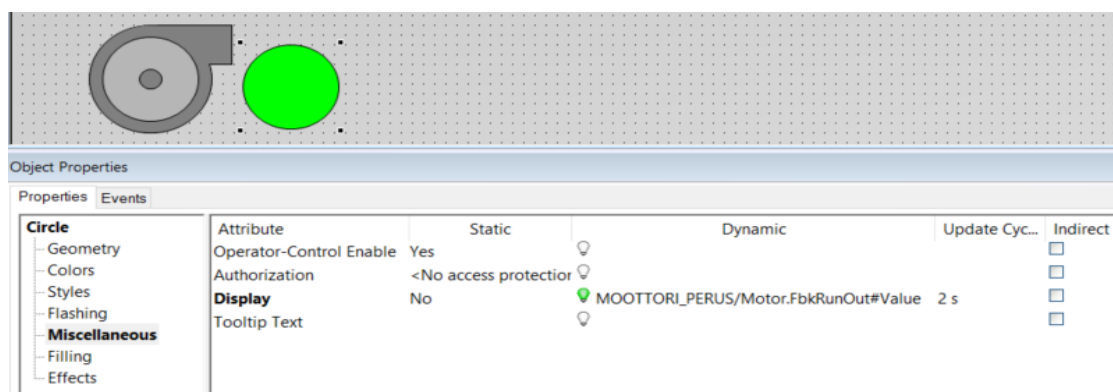
```
{
#define APLSHOWTAG "@local::@APLShowTag"
BOOL bTagVal = 0
bTagVal = GetTagBit(APLSHOWTAG);
bTagVal = !bTagVal;
SetTagBit(APLSHOWTAG, bTagVal);
}
```

Esimerkkikoodi 1 C-kirjoituskoodi näyttä tai piilota tunnisteet.

5.2.2 Dynaamiset toiminnot

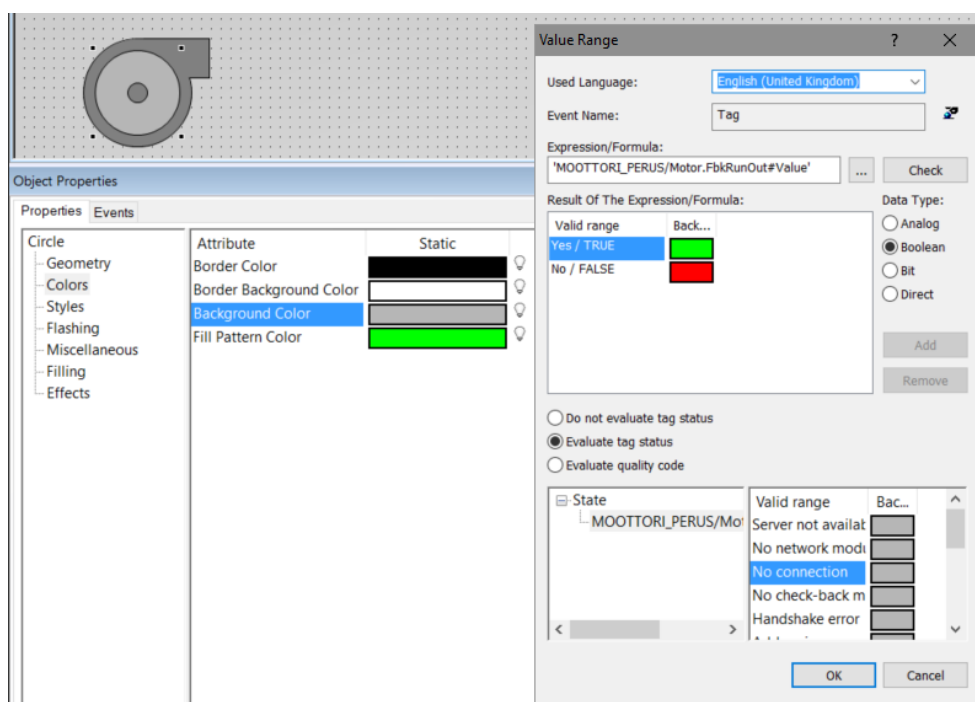
Objekteihin voidaan liittää dynaamisia toimintoja ja linkittää niitä esimerkiksi moottorin käyntitietoon, jolloin saadaan tavallisesta objektista aktiivinen objekti. Dynaamisiin toimintoihin voidaan kytkeä suoraan signaali tai signaaleja ohjelmasta ja näihinkin voidaan käyttää ”JA &”, ”TAI |” sekä ”EI !” -ehtoja, yleisimmin käytettiin ”TAI” / ”|” -ehtoa, koska päänäyttöruudulle tehtiin tilan puutteen vuoksi vain 1 indikaattori ja käytössä 2 pumppua. Bittitietoverailussa käytetään yhtä merkkiä ”|” tai ”&” ja lausekevertailussa kahta ”&&”-merkkiä.

Dynaamisille toiminnoille on tärkeää, että objektista, jonka ominaisuuksia halutaan hyödyntää esimerkiksi I/O-signaaleilla, objektin ominaisuuksista ”Object Properties” alta tehosteista ”Effects” on otettava pois päältä projektinlaajuinen värimalli ”Global Color Scheme”. Tämän jälkeen objektien ominaisuudet on muutettavissa. Kuvan (Kuva 10. s. 11) esimerkissä pumpun käynnin paluutieto on liitetty vihreään ympyrään, joka laitetaan pumpun harmaan ympyrän päälle ja muutetaan näytä ”Display”-ominaisuus pois päältä. Pumpun moottorin käynnistyessä paluutieto ”Feedback” muuttuu nolasta ykköseksi, jolloin vihreän ympyrän näytä-ominaisuus muuttuu nolasta ykköseksi ja näkyy ajokuvassa. Käyttämällä kuvassa olevaa Dynamic-kenttää voidaan käyttää vain yhtä signaalia. Jos on tarpeellista käyttää eteen- ja taakse-suuntien signaaleja, pitää käyttää ”Dynamic dialog”-valintaa (Dynaamiset toiminnot esimerkki 2), jolloin voidaan käyttää to-
tuusarvomuuttujia tai analogiarvolippua.



Kuva 10. Dynaamiset toiminnot, esimerkki 1

Toisessa esimerkissä (Kuva 11, s. 12) sama paluutieto laitetaan "Dynamic Dialog" -valintaikkunaan "Expression/Formula:"-kenttään, niin siihen pitää lisätä ' -merkit ympärille, että signaalista saadaan vertailukelpoinen. Vertailukelpoisesta I/O-tiedosta voidaan nyt "boolean" valinnalla muuttaa pumpun harmaa taustaväri käynnissä vihreäksi, pysähdyksissä punaiseksi ja lisäksi voidaan seurata signaalin laatua ja muuttaa esimerkiksi harmaaksi, jos yhteys katkeaa. "Value range" -ikkunassa pystytään myös hakemaan suoraan tunniste "..."-painikkeen alta tulevasta ikkunasta.



Kuva 11. Dynaamiset toiminnot esimerkki 2

Tässä pitää huomata, että automaatiassa 0 ei ole täysin 0, vaan esimerkiksi 0,4V toisi punaisen tilan ja 0,0V, mikä nyt toisi harmaan tilan.

WinCC:ssä on myös paljon valmiita älykkäitä objekteja, joita käytetään myös va-kiogeneroiduissa lohkokuvakkeissa. Näistä eniten näkynyt kuvaikkuna, I/O-kenttä, palkki, kuva, tilanäyttö "Group Display" sekä painike.

5.2.3 Dynaamisten toimintojen taustakoodi

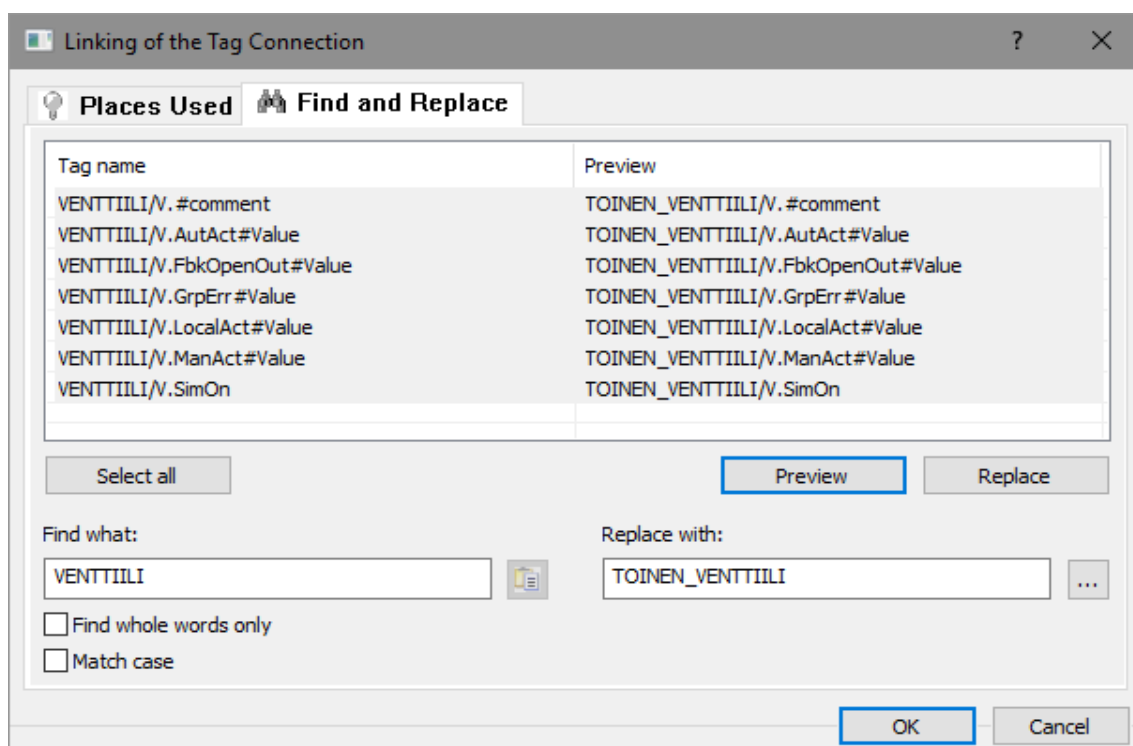
Tehdyissä lohkokuvakkeissa dynaamisten toimintojen taustalinkit saadaan kerralla näkyviin WinCC-editorissa valitsemalla kuvake ja hiiren oikean painikkeen valikosta valitaan "linking → tag connections" ja etsi- ja korvaa toiminnolla saadaan vaihdettua yhdistetty kaavio toiseen, jolloin kuvake kopio on toiminnassa.

Alla olevassa kuvassa (Kuva 12, s. 13) on alkuperäinen venttiililohkokuvake oikealla ja tehty vasemmalla sekä näiden alapuolella kopio hajautettuna. Kuvan ikkunasta näkyy valitut kuvakkeen osat ja näihin liitetyt signaalit. Listan ensimmäiset kaksi on liitetty samaan kuvakkeen osaan Tagi1. "@APLShowTag" on järjestelmätunnus, jolla näytetään ja piilotetaan kaikki kuvakkeiden tunnuskyltit ja toinen ".#comment" on normaalisti liitetty koko ryhmään ja näyttää hiirensoittimella kuvaketta osoitettaessa kaavion kommentin. ".AutAct#Value" on liitetty "A"-kirjaimen näytä "Display"-ominaisuuteen ja lohkon ollessa automaattitilassa näyttää A-kirjaimen. Kuvan alareunassa venttiililohkon signaalit lohkokuvana ja I/O-auluna. Signaali "SimOn" on tyyppiä "BOOL" jolloin tätä voidaan käyttää suoraan, kun muut ovat "STRUCT"-tyyppiä. Tyyppeihin "Struct" sisältyy päällä ja pois "BOOL", lisäksi myös signaalin tilatieto "ST". Tämän takia liitettäessä kuvaobjekteihin "Struct"-tyypin signaalia, täytyy lohkoista löytyvän lipun perään lisätä "#Value".

#	Name	I/O	Type	Comment
97	SimOn	IN	BOOL	Simulation on/off
251	GrpErr	OUT	STRUCT	1 = Group error is active
275	LocalAct	OUT	STRUCT	1=Local operation mode is active
278	AutAct	OUT	STRUCT	1=Automatic mode is active
281	ManAct	OUT	STRUCT	1: Manual mode is active
287	FbkOpenOut	OUT	STRUCT	1=Valve is Opened
290	FbkCloseOut	OUT	STRUCT	1 = Valve is Closed

Kuva 12. Lippuyhteydet

Muutosesimerkissä (Kuva 13, s. 14) näkyy, kuinka helposti etsi- ja korvaa ominaisuudella voidaan saman tyyppinen kuvake muuttaa toimimaan toisena kuvakkeena. Ensimmäiseksi kirjoitetaan etsintäkenttään korvattava kuvakkeen kaavion nimi ja korvaavan kaavion nimi oikealle, tämän jälkeen voidaan esikatsella painamalla esikatsele "Preview"-painiketta ja hyväksyä lopputulos painamalla korvaa "Replace"-painikkeella. Kun halutut muutokset on tehty, voidaan poistua ikkunasta "OK"-painikkeella. Kuvan listasta jää pois näkyvistä "@APLShowTag", koska se ei sisällä sanaa "VENTTIILI" ja tätä myöskään ei kuulu muuttaa, koska se kuuluu järjestelmän sisäisiin tunnuksiin.

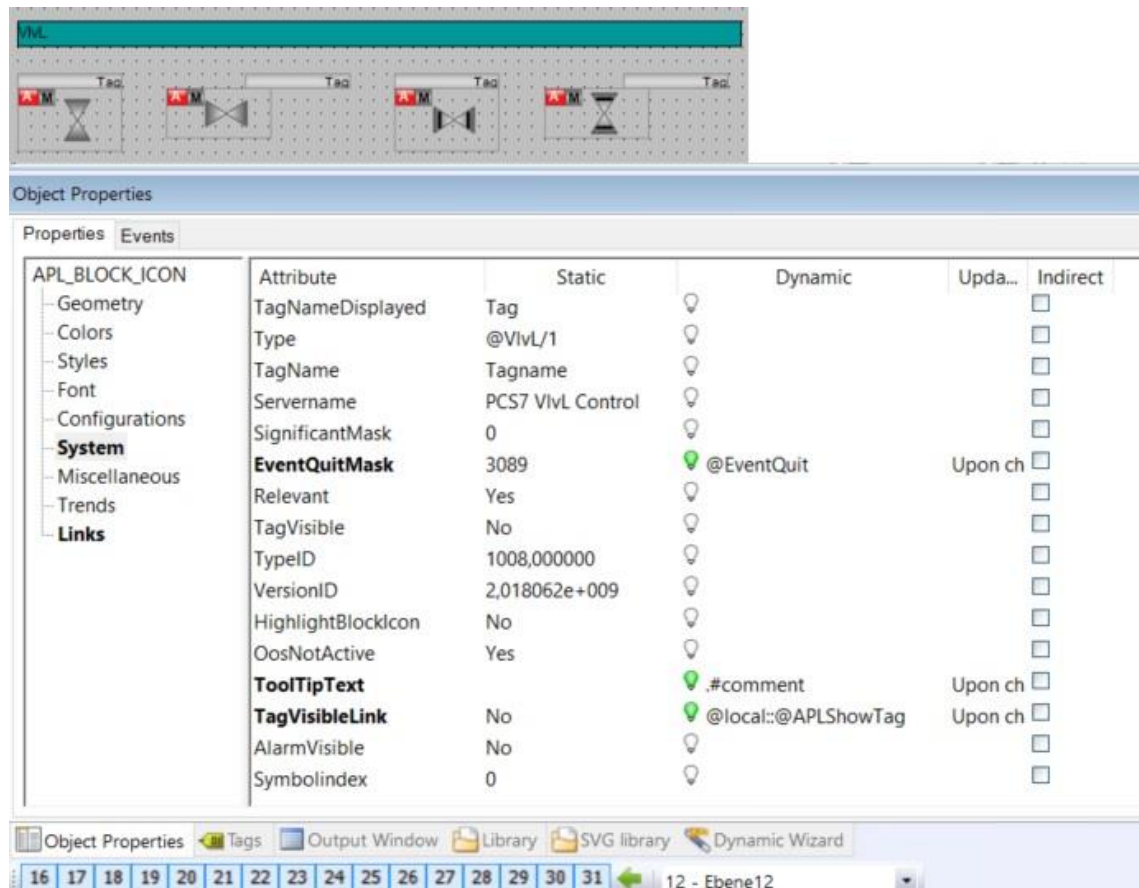


Kuva 13. Esimerkki muutos toiseksi venttiiliksi

Pienemmässä yhden OS:n projektissa koodi on "KAAVIO/LOHKO.TOIMINTO" ja suuremmassa useamman OS:n projektissa kannattaa käyttää koko osoitetta "PALVELIN::KAAVIO/LOHKO.TOIMINTO". Näin saadaan kuvakkeet toimimaan kaikilla ruuduilla samassa verkossa.

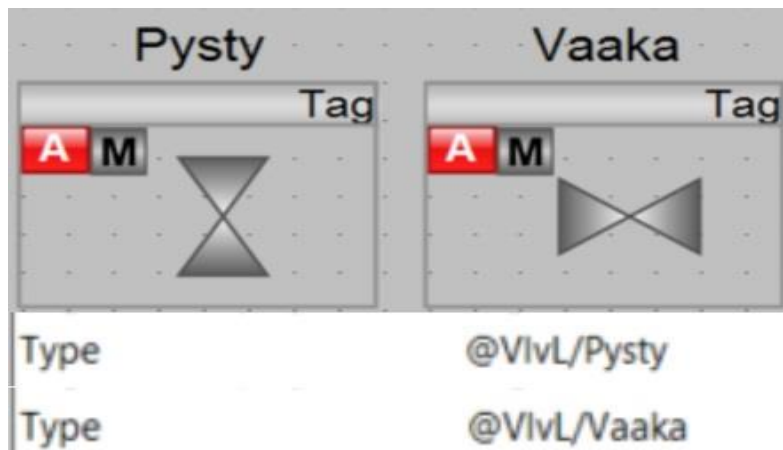
6 Kuvakkeiden muokkaus

”Block Icon” muokkausta on helppo harjoitella, kun kopioi muokattavasta kuvakkeesta kopioita ja testaa ensin näillä ja tämän jälkeen vasta tyyppitiedostoa esimerkiksi ”@PCS7TypicalsAPLV8.PDL”-tiedostoa. Tyyppitiedostosta löytyy kaikki käytössä olevat ikonien mallit, joista näyttöruutujen ikonit luodaan. Esimerkiksi venttiilin tyyppi VlvL:stä on kirjastossa 4 erilaista versiota, mutta aina tämä ei riitä. Useissa projekteissa ovat asiakkaat vaatineet venttiilit staattisiksi, eli asento ei saanut muuttua avautuessa ja sulkeutuessa, mutta värin piti vaihtua. Kuvassa (Kuva 14, s. 15) pystyssä olevat venttiilit ovat vain suljettuna pystyssä ja muuttavat asentoa 90° sekä värin vihreäksi.



Kuva 14. Siemens WinCC @PCS7TypicalsAPLV8.PDL

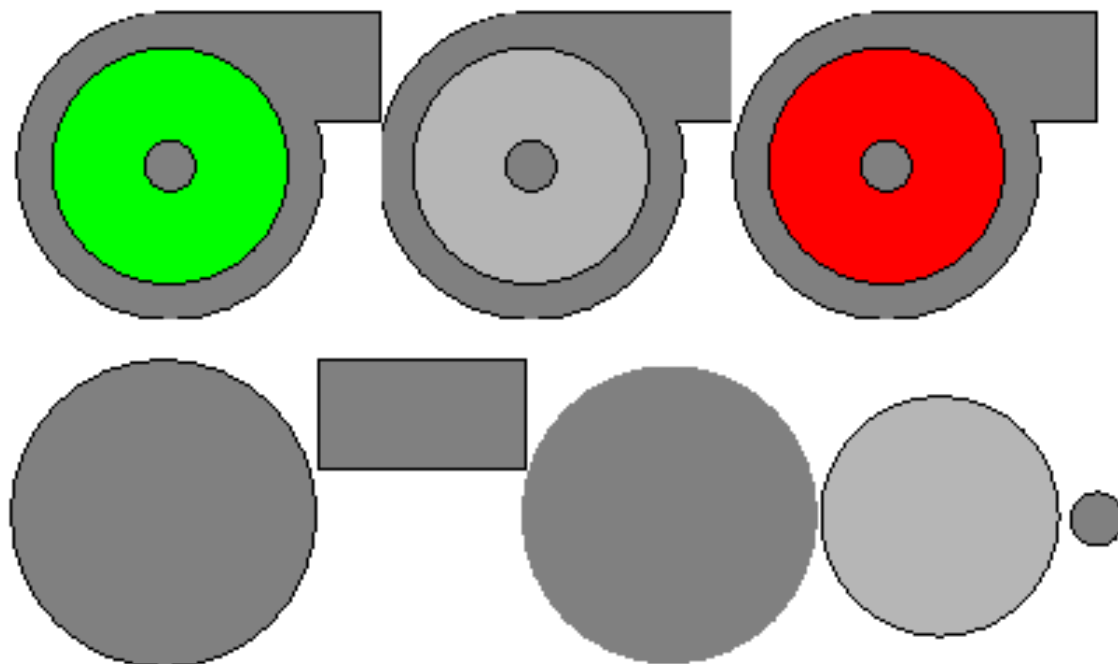
Kaikkien kuvakkeiden tyyppi pitää olla uniikki "Properties / System / Type". Ensimmäisen venttiilin tyyppi on @VlvL/1. Uudet staattiset mallit voisivat olla @VlvL/Pysty ja @VlvL/Vaaka. Näin nämä on helppo muistaa (Kuva 15, s.16).



Kuva 15. Omat venttiilit

Myös generoituja kuvakkeita voi olla vain 1 kopio projektissa per generoitu taso (voidaan generoida useammalle tasolle). Tätä ominaisuutta pystyy kiertämään poistamalla esimerkiksi @-merkin kopion tyyppikohdasta, mutta sen jälkeen ikoni ei enää päivity automaattisesti. Mikäli samaa kuvaketta on tarve käyttää valvontaan useassa paikassa, voidaan hyvin käyttää liitteenä olevia kopioitavia kuvakkeita valvontatarkoitukseen.

SIEMENSiltä löytyy oma ohje Block Iconeiden luomiseen (How do you create your own block icons with SIMATIC PCS 7. 2017), jossa pitää piirtää 4 ympyrää ja 1 nelikulmio ja kopioidaan 3 eri versiota näistä ja näistä vielä viedään ulos "Export" EMF-tiedostot, joita käytetään "Block Icon":eissa. Oppaassa on hyvät opasteet oman pumppukuvakkeen luomiseen. Linkki ohjeeseen on lähteissä.

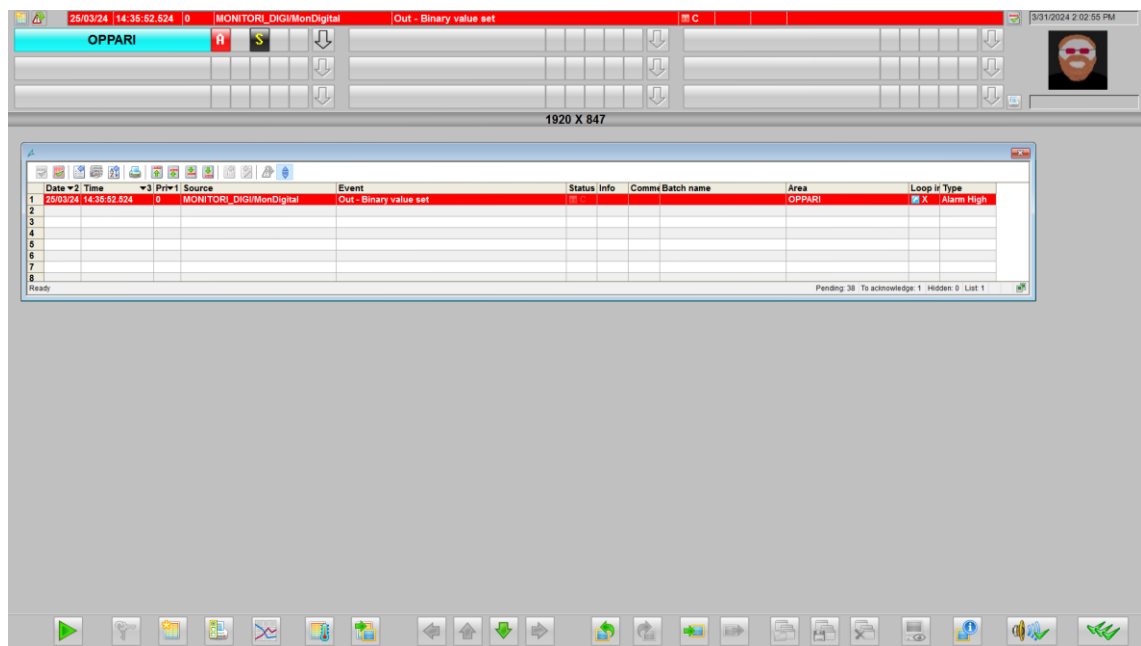


Kuva 16. Siemens lohkokuvake-esimerkki

Kun oma kuvake on luotu, on se hyvä tallentaa omaan tyyppitiedostoon esimerkiksi "@PCS7TypicalsOMAT.PDL". Eräässä foorumissa oli kirjoitettu, että PCS7 -ohjelma hakee käänösvaiheessa kuvakkeiden tyytit ensimmäisenä suurimmasta tyyppitiedostosta. Varmin tapa saada oma kuvake generoitua oikein on vaihtaa kuvakkeen tyyppi esimerkkisi MotL/1 MotL/PUMPPU, jolloin kuvakkeen lohkon "Generate Block Icon" -ikkunaan voi kirjoittaa PUMPPU.

7 WinCC Ajonäkymä

WinCC Runtime eli prosessina ”WinCC RT” on järjestelmän ajonäkymä, josta voidaan tarkastella tai ohjata laitteiden toimintaa. Navigoinnissa hälytyslistan alla löytyy projektin pääalueet, joita klikkaamalla pääsee kyseisten alueiden pääkuviin. Näiden painikkeiden vieressä on nuoli, josta saa hakemistopuun tapaan avattua linkit alakuviin.



Kuva 17. WinCC Runtime 1920 x 1080

Alapalkin navigoinnissa vasemmalle menee kuvapuun juurta kohti ja oikealle syvemmälle ja ylös- tai alas-painikkeista siirtyy puun saman tason kuviin.

PCS7-projekteissa yhdessä kansiossa voi olla vain 1 kuva ja useita kansioita, ja tämän vuoksi usein laitos jaetaan alueisiin, joista muodostuu juurikansiot, eli päätasot. Näiden alle tulevat ajokuvat ja niiden alle tarvittaessa vielä tarkenne-kuvat. Pienissä projekteissa voidaan tehdä ajokuvat suoraan juuritasolle, jolloin yläpalkit riittävät navigointiin. Ajokuvia ohjelmoitaessa on hyvä ottaa huomioon, että ilman vierityspalkkeja 1920 x 1080 resoluution ajokuvaan on käytettävissä vain 1920 x 847 pikseliä.

8 Lopputulos

Aikaisemmin PCS7-versioinnin kanssa ei ole ollut ongelmia, kun on käytetty samaa versiota, mutta siirryttäessä versiosta toiseen ei aina pystytä käyttämään yhtä ja samaa kirjastoa. Tällöin järjestelmän omat kuvakkeet pitää muokata uudelleen eri versioon. Tämä luonnistuu kokeneelta henkilöltä melko nopeasti. Omat tehdyt vaihtoehtoiset kuvakkeet toimivat eri versioissa, koska niissä on käytetty versioriippumattomia perusosia. Omien tehtyjen kuvakkeiden viimeisimmät versiot muistuttavat ulkonäöltään enemmän generoituja lohkokuvakkeita ja esimerkiksi järjestelmän näytä / piilota tagit -painike toimii myös vaihtoehtoisissa kuvakkeissa. Myös mallikuvien käyttö toisista projekteista, joihin on tehty suoria linkkejä voi tuottaa virheilmoituksia avattaessa kuvaa.

Lopputuloksena on kaksi PDL-tiedostoa @-alkuinenlisäkirjasto käytettäväksi Siemensin tyyppitiedoston rinnalle projektin "GraCS"-kansioon sellaisenaan ja toisesta PDL-tiedostosta voidaan kopioida kevyempiä kuvakkeita tai muita työkaluja käyttöön valvomokuvien ohjelmoinnissa. Lisäkirjastotiedostoon (Liite 4) on tehty aikaisemmissa projekteissa useimmiten käytetyistä kuvakkeista versiot, joita on helpompi käyttää. Kirjaston generoitavien lohkokuvakkeiden yllä on kirjainyhdistelmä, jonka voi laittaa lohkokuvakegenerointinumeron tilalle. Esimerkkinä asentoa muuttamattomat venttiilit pysty (PYS) ja vaaka (VAA) sekä moottoreiden eteenpäin suuntia ylös (YLO), alas (ALA), vasemmalle (VAS) ja oikealle (OIK). Digitaalitulon MonDiL01-lohkokuvake ilman muuta kuin päällä / pois tieto, tälle lohkolle värinlipun "Color" värikoodit 0=vihreä, 1=punainen, 2=keltainen, 3=sininen, 4=musta, 5=tumman vihreä ja 6=violetti. Kirjastossa on myös kopiot vaihtoehtoisistakuvakkeista valvontakäyttöön, joita voidaan kopioida ja saattaa käyttöön vaihtamalla kuvakkeen kaavio- ja lohkotunnukset etsi-korvaa toiminolla.

Toisessa kuvatiedostossa (Liite 5) on samat kopioitavat kevyemmät kuvakkeet niiden tyyppiotsikkopalkeilla sekä muita tehtyjä objekteja helpottamaan. Esimerkiksi tekstikentillä on eri kielillä tekstin lisäksi eri muotoilut ja fontit ja tässä on nyt muutama malliteksti kerrotulla fontilla, koolla ja tasauksilla sekä lihavoitu maininta. Ensimmäinen kirjain tarkoittaa tekstin Y- ja toinen X-tasausta, seuraavana on käytetty fontti ja koko ja viimeisenä lihavointi maininta. Projekteissa vakiokielet ovat saksa, englanti (United States), espanja (perinteinen), ranska ja italia.

Tiedoston näytä/piilota-painike on samaa kokoa kuin ajokuvan alapalkin painikkeet, joka on hyvä sijoittaa kuvaan @Button12.PDL, jolloin painiketta ei tarvitse sijoittaa ajokuvaan, vaan painike löytyy ajonäkymän alapalkista.

Oikeanalakulman ruudukko on tarkoitettu mittaustyökaluksi, joka sijoitetaan mittattavan objektin päälle ja muutetaan koko objektin kokoiseksi, jolloin viivat näytävät kohdat objektin 25 %, 50 % sekä 75 % X- ja Y-akseleilla.

Näiden lisäksi on muutama reaktoriajokuvamalli ja niihin pintapalkit eri ominaisuuksilla. Näiden ominaisuuksien kopiointi ja sijoitus toimii samalla WinCC-työkalulla kuin muissakin objekteissa.

9 Yhteenveto

Projekti on ollut pitkä ja materiaalia sekä tietoa ja taitoa on kerätty ensimmäisestä asiakasprojektista asti. Viimeisin suurempi projekti valmistui omalta osalta jo vuonna 2022, jonka jälkeen vielä jatkoin aputiedoston suunnittelua tulevia PCS7-projekteja varten. Projektin aikana tein yhteistyötä kahden muun tekijän ja asiakkaan prosessiasiantuntijoiden kanssa koko projektin ajan, näin saatiin asiakkaan mielestä hyväksyttävää jälkeä loppuasiakkaalle.

Projektin alkuvaiheessa tein kuvakkeista pienen kirjaston, jossa on englanniksi ja toisella projektin kielellä kaikkea tarpeellista usein käytettäviä objekteja. Muun muassa otsikkopalkki kolmelle tasolle, tekstikentät projektiin määritellyllä fontilla eri tasauksilla sekä siirtymislinkkejä. Totta kai mukana oli myös pari moottorikuvaketta, auki/kiinni ja analoginen venttiili, digitaalinen signaaliosoitin sekä analoginen numeronäyttö, joista saatiin helposti päätasoon tai yksityiskohtaisiin näyttöihin tarpeelliset objektit. Lisää realismia projektiin toi myös se, että saimme kerättyä asiakkaan kanssa heidän laitoksensa 3D-mallista laitteistojen mittoja, jolloin pystyimme piirtämään lähes kaikki laitteet oikeassa mittasuhteessa operaattoreiden ruuduille.

Tulevia projekteja varten on kerätty omaan tiedostoon valmiiksi muokattuja objekteja, joita voi tarjota loppuasiakkaalle näytettäväksi. Monien kielten tuen säilyttämiseksi on alkuperäinen tukitiedosto säilytettävä erillään projektikoneesta, koska projektikoneella tallennuspainiketta painaessa muut kielet kuin projektikoneessa olevat tekstit muuttuvat "??????"-jonoiksi.

Myös aikaisemmista työkokemuksista ja koulutuksista on ollut hyötyä PCS7-projektien toteuttamisissa, koska useimmissa projekteissa olen ollut mukana pystyttämässä laitteistoa, mutta viimeisimmässä ne oli jo ehditty pystyttämään. Saimme koulutuksen myöhemmin vielä uusiin VMware, vSan ja VMware vSphere -järjestelmiin, koska ohjelmoimamme järjestelmä pyöri lähes kokonaisuudessaan virtuaalipalvelimella. Kahdennettu automaatioasema ja insinööriasema olivat

omat laitteet ja loput työasemista oli toteutettu mini-PC-laitteistoilla ja etätyöpöydillä paikallisella VMware-palvelimella.

Teknologia 23 -messuilla kerrottiin, että tuki PCS7-ohjelmistoille pysyy ainakin vielä 2035 vuoteen asti, oletettavasti juuri julkaistavalle versiolle 10. Vanhempien PCS7-versioiden, kuten 8.2 (2016) ja 9.0 (2017) normaali tuki on päättynyt syksyllä 2023 (McCastlin 2023). Mikäli aikataulu pysyy samana, uusimman 9.1 (2021) version tuki olisi päätymässä noin vuonna 2027.

Jatkokehitysideana olisi lisätä vielä signaalin laadun seuranta sekä VB- tai C-kirjoituskoodi oikean ponnahdusikkunan avaamiseen omiin kuvakkeisiin. Tai omille kuvakkeille voisi tehdä versiot, joissa kaaviotunnisteiden sijasta tietolohkoista ja "PUT-GET"-kommunikointiin (S7 Communication with PUT/GET 2019.) muun järjestelmän kanssa, esimerkiksi laitoksen ulkoinen valvomolaite. Tietoturvan takia tämäkin pitäisi järjestää niin, ettei ulkoa pääsisi lähettämään ohjauksia laitokselle, vaan pelkästään ulospäin menisi valvontaa varten tietoa. Tähänkin on olemassa ratkaisu, jossa käytetään kahta johdinta kommunikointiin, toinen lähetykseen ja toinen vastaanottoon. Näistä voidaan laitoksen suuntaan vastaanotettava johto fyysisesti katkaista, jolloin valvontatieto saadaan ulos ja laitokseen ei voida lähettää mitään.

Lähteet

Biokaasulaitos valvomopäätteidenohjelmointi. 2018. Asiakasprojekti. BS.

Energiasiirtolaitoksen PCS7-ohjelmointi. 2024. Asiakasprojekti. HE.

HMI-päivitys. 2023. Asiakasprojekti. AM.

How do you create your own block icons with SIMATIC PCS 7. 2017. Verkkoaineisto. Siemens. <<https://support.industry.siemens.com/cs/document/16514590/how-do-you-create-your-own-block-icons-with-simatic-pcs-7-?dti=0&lc=en-FI>>.

Kaasulaitos ohjelma- ja valvomopäivitys. 2022. Asiakasprojekti. TC.

Kalkkilaitos luokitin valvomokuvanohjelmointi. 2019. Asiakasprojekti. NK.

Kalkkilaitos ohjelma- ja valvomopäivitys. 2021. Asiakasprojekti. NK.

McCastlin Christopher. 2023. BIG CHANGES TO PCS7 COULD AFFECT YOUR SYSTEM (BUSINESS?). Verkkoaineisto. <<https://www.ace-net.com/blog/big-changes-to-pcs7-could-affect-your-system-business>>.

Mineraaliprosessilaitos valvomopäätteidenohjelmointi. 2018. Asiakasprojekti. AD.

Mineraaliprosessilaitos valvomopäätteidenohjelmointi. 2018. Asiakasprojekti. KL.

Mineraaliprosessilaitos valvomopäätteidenohjelmointi. 2020. Asiakasprojekti. UD.

Mineraaliprosessilaitos valvomopäätteidenohjelmointi. 2022. Asiakasprojekti. RC.

Prosessilaitetestauksia. 2023. Asiakasprojekti. RN.

S7 Communication with PUT/GET. 2019. Verkkoaineisto. Siemens. <https://cache.industry.siemens.com/dl/files/115/82212115/att_108330/v2/82212115_s7_communication_s7-1500_en.pdf>.

Support forum. Verkkoaineisto. 2018. Siemens. <mall.industry.siemens.com>.

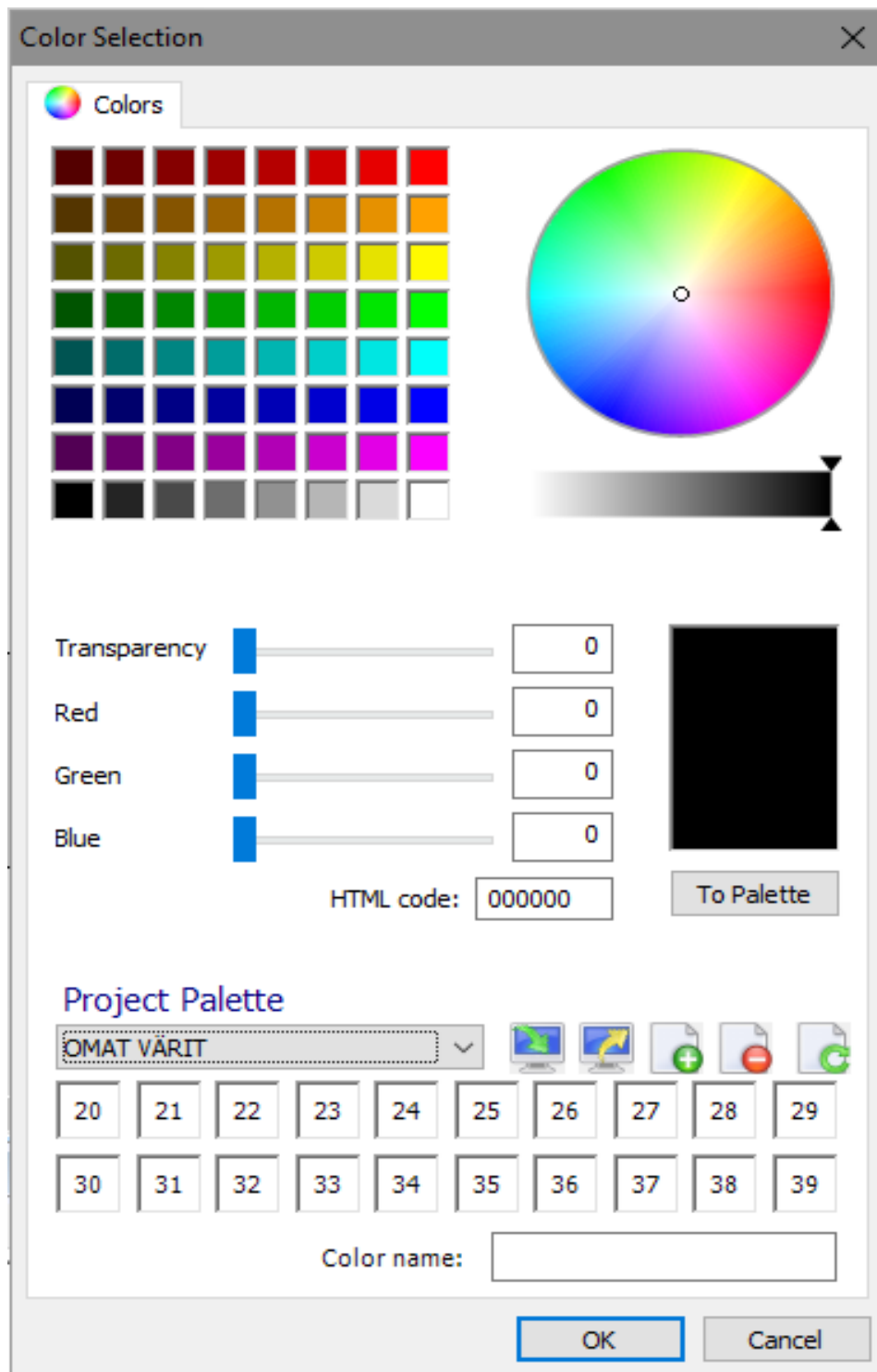
Tiivistinprosessilisäosa valvomopäätteohjelmointi. 2021. Asiakasprojekti. LG.

WinCC-projektin korjaus- ja päivitystehtäviä. 2023. Asiakasprojekti. VV.

Liitteet

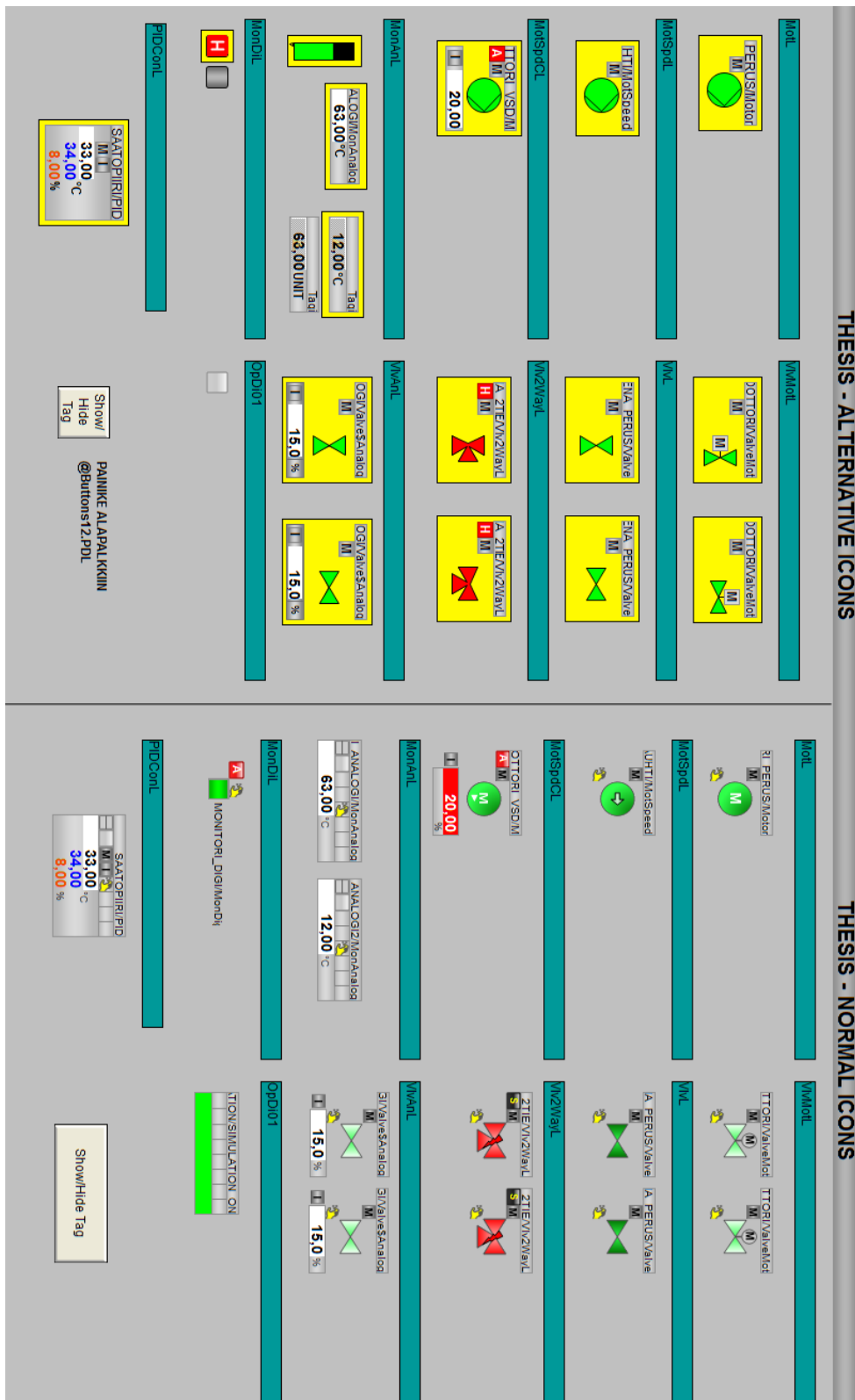
1. Siemens WinCC-väripaletti
2. Valvomokuvakevertailu
3. Kuvakkeiden paikoitus VB-kirjoituskoodi
4. Kuvaketyypitiedosto
5. Projektiaputiedosto

Siemens WinCC-väripaletti



Kuva 1 WinCC-Väripaletti

Valvomokuvakevertailu



Kuva 1 Kuvakkeiden vertailu

Kuvakkeiden paikoitus VB-koodi

Kuvakkeiden paikkojen haku ja sijoitus Visual Basic koodi.

Paikkojen tallennus

```
Sub SaveLocations()  
Dim Objekti As HMIObject  
Dim Storefile As String  
Storefile = "C:\OPPARI" & Left(ActiveDocument.Name, Len(ActiveDocu-  
ment.Name) - 3) & ".txt"  
fnum = FreeFile()  
Open Storefile For Output As #fnum  
For Each Objekti In ActiveDocument.HMIObjects  
    If Objekti.Type = "HMICustomizedObject" Then  
        Print #fnum, Objekti.ObjectName.value & ", " & Objekti.Top & ", " &  
        Objekti.Left  
    End If  
Next  
Close #fnum  
End Sub
```

Paikkojen palautus

```
Sub Rearrange()
```

```
Dim Storefile As String
```

```
Storefile = "C:\OPPARI\" & Left(ActiveDocument.Name, Len(ActiveDocu-  
ment.Name) - 3) & ".txt"
```

```
For Each Objekti In ActiveDocument.HMIObjects
```

```
    If Objekti.Type = "HMICustomizedObject" Then
```

```
        fnum = FreeFile()
```

```
        Open Storefile For Input As fnum
```

```
        Do While Not EOF(1)           " Loop until end of file.
```

```
            Line Input #fnum, FullString " Read data into two variables.
```

```
            ObjektiData = Split(FullString, ",")
```

```
            If ObjektiData(0) = Objekti.ObjectName.value Then
```

```
                Objekti.Left = ObjektiData(2)
```

```
                Objekti.Top = ObjektiData(1)
```

```
            End If
```

```
        Loop
```

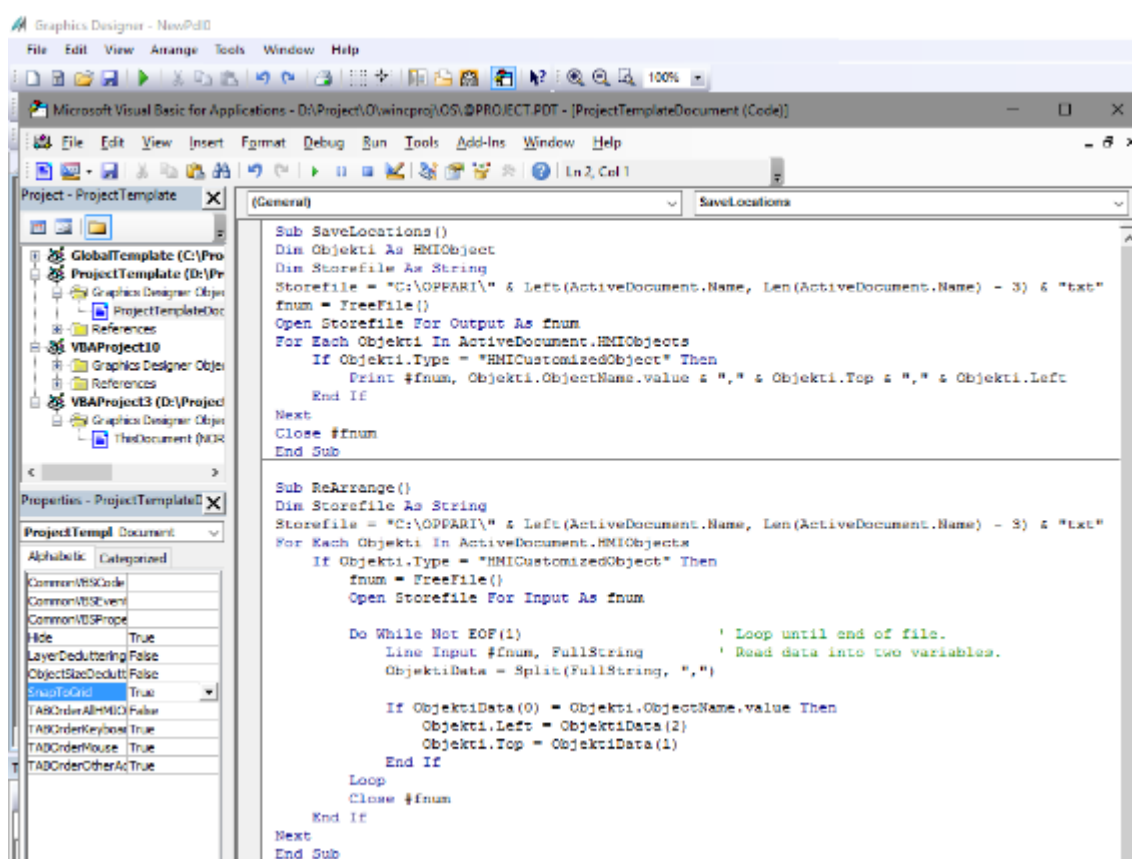
```
        Close #fnum
```

```
    End If
```

```
Next
```

```
End Sub
```

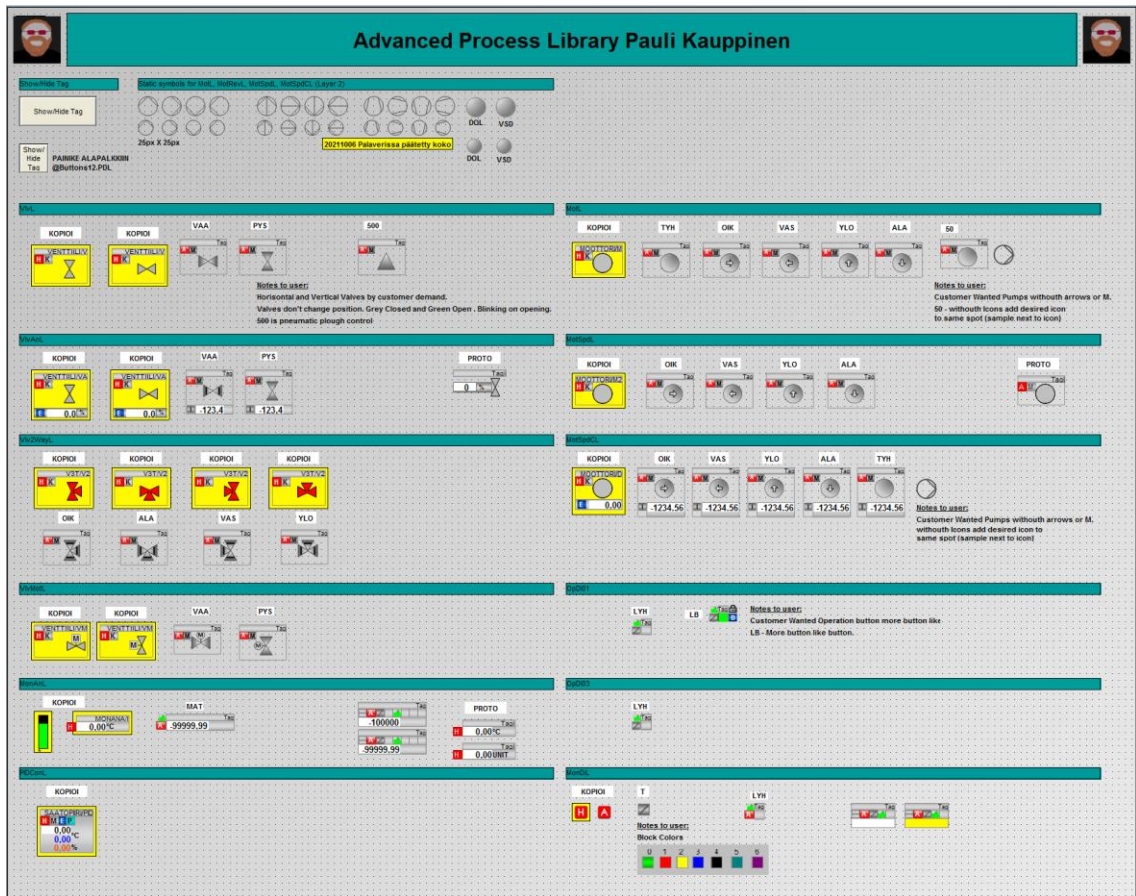
Tällä Visual Basic-koodilla voidaan tallentaa ja palauttaa kuvakäännöissä luotujen kuvakkeiden sijainnit ajokuvuihin. Koodia voidaan käyttää WinCC ikkunan kautta, sen yläpalkissa on painike ”Visual Basic Editor” (Kuva 1 s. 3), jolla ikkuna saadaan auki. Koodissa on kaksi osaa, joista ylempi tallentaa ja alempi palauttaa kuvakkeiden sijainnit. Käsiteltävä kuva pitää olla avoimena taustalla ja VB-Editorista valitaan haluttu toiminto joko yläreunan oikeasta vetopalkista tai hiirellä klikkaamalla osioon ja tämän jälkeen ”PLAY”-painikkeesta yläpalkissa tai ”F5”-painikkeella näppäimistöstä. Koodi tallentaa tai hakee kuvakkeiden sijainnit tekstitiedostoista koodissa annetusta kansioista.



```
Sub SaveLocations()  
Dim Objekti As HMIObject  
Dim Storefile As String  
Storefile = "C:\OPPARI\" & Left(ActiveDocument.Name, Len(ActiveDocument.Name) - 3) & ".txt"  
fnum = FreeFile()  
Open Storefile For Output As #fnum  
For Each Objekti In ActiveDocument.HMIObjects  
If Objekti.Type = "HMICustomizedObject" Then  
Print #fnum, Objekti.ObjectName.value & ", " & Objekti.Top & ", " & Objekti.Left  
End If  
Next  
Close #fnum  
End Sub  
  
Sub ReArrange()  
Dim Storefile As String  
Storefile = "C:\OPPARI\" & Left(ActiveDocument.Name, Len(ActiveDocument.Name) - 3) & ".txt"  
For Each Objekti In ActiveDocument.HMIObjects  
If Objekti.Type = "HMICustomizedObject" Then  
fnum = FreeFile()  
Open Storefile For Input As #fnum  
  
Do While Not EOF(1) ' Loop until end of file.  
Line Input #fnum, FullString ' Read data into two variables.  
ObjektiData = Split(FullString, ",")  
  
If ObjektiData(0) = Objekti.ObjectName.value Then  
Objekti.Left = ObjektiData(2)  
Objekti.Top = ObjektiData(1)  
End If  
Loop  
Close #fnum  
End If  
Next  
End Sub
```

Kuva 1 Esimerkki Microsoft Visual Basic kirjoituskoodista

Kuvaketyyppitiedosto



Kuva 1 @PCS7TypicalsAPVL90_Mukautetut_kuvakkeet.PDL

