

# Käyttöliittymän suunnittelu ja toteutus HTL-prosessille



Ammattikorkeakoulututkinnon opinnäytetyö  
Sähkö- ja automaatiotekniikka, insinööri (AMK)

Kevät 2022

Konsta Lallo

## TIIVISTELMÄ

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa käyttöliittymä Beckhoff HMI -ohjelmistolla Tampereen yliopiston hankkeessa rakennetulle biopolttoainetta valmistavalle koelaitteelle. Käyttöliittymä haluttiin pitää käyttäjän kannalta mahdollisimman selkeänä, mutta kuitenkin toiminnallisesti suhteellisen monipuolisena ja turvallisena operoida.

Opinnäytetyössä esiteltiin koelaitteen perustoiminta sekä selitettiin sen hyödyntämää HTL-prosessia biopolttoaineen valmistuksessa. Pääasiallisesti opinnäytetyössä perehdyttiin kuitenkin HMI-suunnitteluun sekä Beckhoff HMI -sovelluksen käyttöön ja sen ominaisuuksiin.

Kokonaisuudessaan työ oli onnistunut, sillä käyttöliittymä saatiin kehitettyä täysin tyhjästä ja se saatiin lopulta myös toimimaan halutun mukaisesti. Entuudestaan tuntemattoman HMI-ohjelmiston opetteluun kului aikaa, mutta opetteluun jälkeen, osoittautui se hyvin tehokkaaksi ja monipuoliseksi työkaluksi. Näin ollen opinnäytetyö oli opettavainen projekti. Työn edetessä alkuperäistä ajatusta sen laajuudesta karsittiin, että opinnäytetyön aihepiiri pysyisi selkeänä eikä työn laajuus kasvaisi liialliseksi.

ABSTRACT

The aim of the thesis was to design and implement human machine interface with Beckhoff's HMI software for a biofuel producing pilot plant built in a project at the University of Tampere. The aim was to keep the user interface simple for the user. However, functionally relatively versatile, and safe to operate.

The basic operation of the pilot plant was presented in the thesis and the HTL process utilized in the production of biofuel was explained. However, the thesis mainly focused on HMI design and the use of the Beckhoff HMI software and its features.

Overall, the work can be considered successful, as the HMI was developed from scratch and eventually made to work as desired. It took time to learn the HMI software because I had no previous experience with it, but after learning to use it, it proved to be a very effective and versatile tool. Thus, the thesis was an instructive project. As the work progressed, the initial idea of its scope was curtailed so that the topic of the thesis would remain clear, and the scope of the work would not become excessive.

## Sisälllys

1	Johdanto .....	1
2	EHTA-hanke .....	2
2.1	Koelaite .....	2
2.2	HTL-prosessi .....	4
3	HMI .....	6
4	Ohjelmointiympäristö.....	9
4.1	TwinCAT 3 Engineering .....	9
4.2	TwinCAT 3 HMI.....	10
4.3	TwinCAT 3 HMI Server .....	11
5	Ohjelmointi.....	12
5.1	HMI-ohjelmiston käyttö .....	12
5.1.1	Konfigurointi ja HMI:n julkaiseminen .....	12
5.1.2	Perusrakenteen ja sivujen luominen.....	15
5.1.3	Toiminnot ja ehdot.....	17
5.1.4	Arvojen syöttäminen käyttöliittymään .....	18
5.1.5	Sisäiset ja ulkoiset muuttujat .....	19
5.1.6	Kuvatiedosto käyttöliittymässä.....	21
5.1.7	Kielivalinta .....	22
5.1.8	Kuvaajat.....	24
5.1.9	Skriptikielet .....	25
5.1.10	Omien elementtien luominen ja käyttö .....	28
5.2	Hälytykset.....	31
6	Lopullinen käyttöliittymä.....	34
6.1	Pääajo.....	34
6.2	Testi- ja huuhteluajo .....	36
7	Yhteenveto .....	37
	Lähteet.....	38

## Kuvat, taulukot ja kaavat

Kuva 1. Koelaitteen 3D-malli .....	3
Kuva 2. HMI Serverin konfigurointisivu .....	13
Kuva 3. Hakemistosijainti HMI Serverin automaattista käynnistystä varten .....	13
Kuva 4. HMI:n konfigurointi projektin sisältä .....	14
Kuva 5. HMI:n julkaiseminen .....	15
Kuva 6. Region-elementin asettelu käyttöliittymän pääsivulle.....	15
Kuva 7. Sivunvaihdon määrittäminen painikkeeseen .....	16
Kuva 8. Live View -painike .....	17
Kuva 9. Toiminnot ja ehdot -ikkuna.....	17
Kuva 10. Input-elementin konfigurointi .....	19
Kuva 11. Sisäisen muuttujan luominen .....	20
Kuva 12. ADS:n välityksellä vastaanotetut PLC-muuttujat.....	20
Kuva 13. HMI Gallery Explorer.....	22
Kuva 14. Painonappeja, joissa käytetty HMI Gallery Explorer -kuvia.....	22
Kuva 15. Uuden kielipaketin tuominen projektiin.....	23
Kuva 16. Localization-muuttujat.....	23
Kuva 17. Localization Select -elementti.....	24
Kuva 18. Esimerkissä käytetty tilanne, jolloin tapahtuma aktivoituu .....	27
Kuva 19. JavaScript-koodi .....	27
Kuva 20. JavaScript-ponnahdusikkuna todellisessa näkymässä.....	28
Kuva 21. Esimerkki UserControl-elementistä.....	29
Kuva 22. UserControl-parametri-ikkuna .....	30
Kuva 23. Parametrien kiinnittäminen elementteihin kyseisessä esimerkkitilanteessa ..	30
Kuva 24. Properties-ikkunaan ilmaantuneet UserControl-parametrit.....	31
Kuva 25. Esimerkkitalanteen UserControl-elementti todellisessa näkymässä .....	31
Kuva 26. References-kansion sijainti, josta lisäkirjastot ladataan.....	32
Kuva 27. Hälytysten luominen.....	32
Kuva 28. Esimerkki ST-kielisestä PLC-koodista, jolla hälytyksiä käytetään .....	33
Kuva 29. EventGrid-elementti .....	33
Kuva 30. Käyttöliittymän Desktop-sivu .....	34
Kuva 31. Pääajotavan ensimmäinen sivu .....	35

Kuva 32. Huuhteluajon käyttöliittymäsivu .....36

Taulukko 1. Beckhoff-muuttujien käyttäminen JavaScriptillä.....26

## **Liitteet**

Liite 1      Pääajotavan käyttöliittymäsivut

## 1 Johdanto

Energiankulutus on ollut nopeassa kasvussa maailmanlaajuisesti. Lisäksi tiukentuneet ympäristövaatimukset ohjaavat ihmisiä kehittämään uusia ympäristöystävällisempiä energiaratkaisuja. Tämä opinnäyte kertoo koelaitteesta, joka valmistaa biomassasta biopolttoainetta HTL eli hydrotermisellä nesteytysprosessilla lukuisien monimutkaisien reaktioiden seurauksena. Biopolttoaineilla on paljon sovelluskohteita fossiilisten polttoaineiden korvikkeina mm. tie- ja lentoliikenteessä, mutta myös sähköntuotannossa. Uusiutuvista luonnonvaroista valmistettuja biopolttoaineita voidaan pitää hiilineutraaleina, sillä ne absorboivat hiilidioksidia fotosynteesillä saman verran kuin mitä ne vapauttavat, kun polttoaine palaa.

Opinnäytetyössä esitellään lyhyesti koelaitteen toimintaa sekä sen käyttämän HTL-prosessin toimintamekanismia. Lisäksi selvitetään tehokkaan käyttöliittymän suunnitteluun liittyviä toimintaohjeita käyttöliittymäsuunnitteluun tarkoitettun standardin pohjalta. Pää tavoitteena opinnäytetyöllä oli kuitenkin suunnitella ja luoda tälle Tampereen yliopistolla rakennetulle koelaitteelle käyttöliittymä prosessin ohjausta varten. Prosessia ohjaa Beckhoffin ohjelmoitava logiikka, joten myös käyttöliittymä tehtiin heidän TwinCAT 3 - ohjelmointiympäristöön ladatulla lisäosalla.

Opinnäytetyössä esitellään Beckhoff HMI -ohjelmiston toimintoja suhteellisen kattavasti ja koska materiaaleja ohjelmiston käytöstä on tällä hetkellä niukasti olemassa ja erityisesti suomen kielellä materiaaleja on erittäin vähän, voi opinnäytetyötä käyttää myös eräänlaisena opetusmateriaalina ohjelmiston käytössä. Lisäksi opinnäytetyössä on muutamia pienimuotoisia esimerkkejä HMI-ohjelmiston toiminnoista, joita seuraamalla ja muokkaamalla pääsee opinnäytetyön lukija konkreettisesti kiinni Beckhoff HMI -lisäosan toimintoihin sekä oppii niitä käyttämään.

## 2 EHTA-hanke

EHTA eli Efficient HydroThermal Application on Tampereen yliopiston hanke. Projektissa rakennetaan koelaitte, jolla pyritään valmistamaan fossiilisille polttoaineille korvaavaa biopolttoainetta tehokkaasti ja toteutuskelpoisesti. Hankkeen vetäjä professori Tero Joronen listasi hankkeen tavoitteita PowerPoint-esityksessään. Projektilla tavoitellaan mm. uusien keinojen löytämistä ilmastonmuutoksen ehkäisemiseksi. Lisäksi pyritään omalta osaltaan vastaamaan mm. Pariisin ilmastopöytäkirjan tavoitteisiin sekä Euroopan unionin Puhdasta energiaa kaikille eurooppalaisille -paketin tavoitteita. Hankkeella on myös useita tutkimuksellisia tavoitteita erityisesti HTL-prosessiin liittyen. (T. Joronen, julkaisematon lähde, 2021)

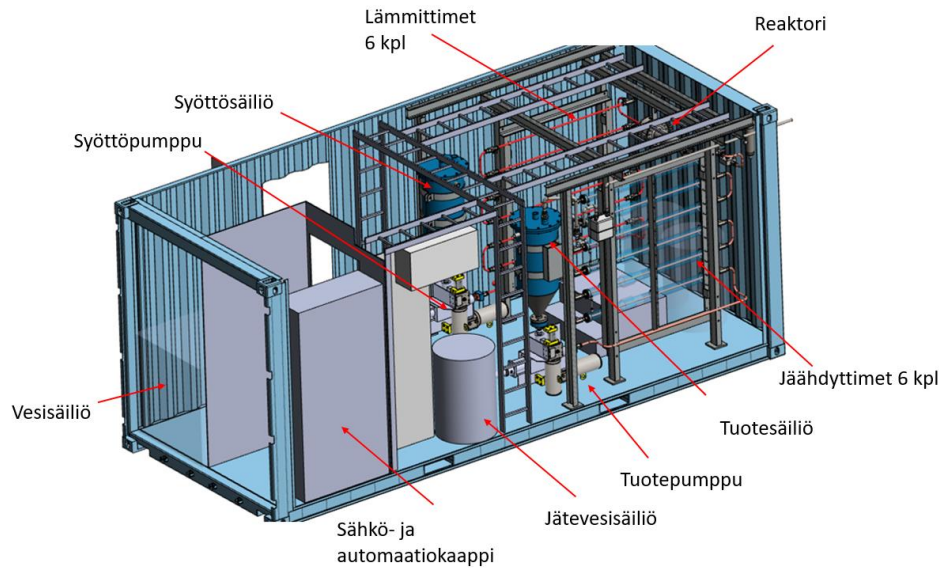
### 2.1 Koelaitte

Koelaitte rakennetaan Tampereen yliopistolla Hervannan kampuksella. Koelaitte käyttää HTL-prosessia, joka on lyhenne englannin kielen sanoista Hydrothermal Liquefaction, ja jota suomeksi kutsutaan hydrotermiseksi nesteytykseksi. Tämä prosessi mahdollistaa kiinteän biomassan muuntamisen nestemäiseksi biopolttoaineeksi vaativissa olosuhteissa (Kumar ym., 2018, ss. 1745–1746).

Koelaitteisto rakennetaan merikonttiin (Kuva 1), jolloin laitteiston kuljettaminen on helppoa sekä Suomen sisällä että sen ulkopuolella esim. sellaisille paikoille, jossa raaka-ainetta syntyy. Koelaitetta tullaan käyttämään useassa hankkeessa mm. ILPO-ilmastopositiiviset yritysalueet ja arvoketjut -hankkeessa (6Aika, 2020). Sekä Euroopan unionin BL2F-hankkeessa, jonka nimi on lyhenne sanoista Black Liquor to Fuel. Kyseisessä hankkeessa nimensä mukaisesti raakamateriaalina käytetään mustalipeää. (BL2F, 2020)



Kuva 1. Koelaitteen 3D-malli



Yllä olevassa 3D-mallissa (Kuva 1) on koelaitteen tärkeimmät komponentit nimettyinä. Kokonaisuudessaan koelaitteessa on noin 100 I/O:ta ja Beckhoff CX5130 -sarjan ohjelmoitava logiikka vastaa niiden ohjauksesta. Prosessin toiminta etenee pääpiirteittäin seuraavasti: Prosessi alkaa syöttösäiliöstä, joka nimensä mukaisesti sisältää prosessissa käytetyn syötteen. Syöte pumpataan syöttöpumpulla tarvittavaan paineeseen, joka tässä tapauksessa on 200–300 baaria. Syöte laimennetaan n. 10–20 % kiintoainepitoisuuteen vesisäiliöstä tulleen veden kanssa. Tämän jälkeen liete kulkee sähköisten vastuslämmittimien kautta, joilla lietteen lämpötila nostetaan 300–400 °C:seen. Liete ohjautuu reaktoriin, jonka jälkeen liete on nesteytynyt sisältäen bioöljyä, vesifaasin, kaasuja sekä mahdollisia kiintoaineita. Saatu tuote jäähdytetään 80 °C:seen vedellä toimivilla jäähdyttimillä. Tämän jälkeen alennetaan paine takaisin normaaliin ilmakehän paineeseen. Viimeisenä lopputuote pumpataan tuotesäiliöön. Tieto perustuu projektin vetäjän professori Tero Jorosen suulliseen, mutta myös sähköpostiviestien väliseen tiedonantoon sekä muihin julkaisemattomiin lähteisiin opinnäytetyöprosessini aikavälillä 5/2021–12/2021. (T. Joronen, henkilökohtainen tiedonanto, 5/2021–12/2021)

Riskienarviointidokumentista käy ilmi, että prosessin vaativien olosuhteiden vuoksi laitteistoa suunniteltaessa on täytynyt ottaa huomioon monta turvallisuuteen liittyvää seikkaa (TTY, julkaisematon lähde, 2017). Kuitenkin suurin osa mahdollisista ihmisille kohdistuvista riskeistä on saatu poistettua sillä, että prosessi on erillisessä kontissa sekä prosessin ohjaus käyttöliittymästä tapahtuu omassa toimistokontissa tietokoneelta käsin. Näin ollen prosessikontti on täysin miehittämätön laitteiston ollessa käynnissä.

Prosessilaitteet voivat kuitenkin edelleen vahingoittaa, joten varo- ja turvatoimenpiteet on määritelty jokaiselle mahdolliselle virhetilalle tarvittavilla riskin arviointimenetelmillä. Näillä turvallisuustoimenpiteillä prosessi on riittävän turvallinen käyttää ja vakavat vahingot pystytään välttämään. (Saarinen, 2020, s. 20)

Koelaitetta on pyritty suunnittelemaan siten, että se olisi mahdollista skaalata suurempaan kokoluokkaan. Kuitenkin koelaitteen suurentaminen pilottikoosta teolliseen kokoluokkaan on haastavaa. HTL-prosessit ovat vielä kehittyvä teknologia ja suurin osa HTL-prosessilaitteista on laboratorionkokoluokkaa, eikä niitä juurikaan ole käytetty kaupallisesti suuressa kokoluokassa. Suuria haasteita laitteiston kokoluokan kasvattamisessa ovat prosessin vaativat olosuhteet, jotka aiheuttavat erityisesti korroosiota. Jotta laitteistoa voidaan suojata siltä, on välttämätöntä käyttää kalliita metalliseoksia ja tämä lisää laitteiston kokonaiskustannusta merkittävästi. Prosessin korkea operointipaine aiheuttaa myös instrumenteille tiukat vaatimukset, kuten syöttöpumpuille. (Toor ym., 2011, s. 2328) Muita haasteita ovat mm. reaktorin suunnittelu, lämmönsiirto, paineen laskeminen, kiintoaineiden erottaminen sekä bioöljyn ja vesifaasin erotus toisistaan. Nämä haasteet tulisi ratkaista, jotta HTL olisi kilpailukykyinen teknologia. (Saarinen, 2020, s. 20)

## 2.2 HTL-prosessi

Hydrotermien nesteytys on prosessi, jossa voidaan käyttää syötteenä lähes mitä tahansa biomassaa. Biomassa muuntuu kuumassa ja korkeapaineisessa vedessä bioöljyksi lukuisien monimutkaisten reaktioiden seurauksena (Kumar ym., 2018, s. 1745). Biomassat voidaan karkeasti jakaa joko nestemäisiksi tai kiinteiksi biomassoiksi. Biomassan muuntaminen bioöljyksi tapahtuu yleensä olosuhteissa, jossa lämpötila on 280–370 °C sekä paine 100–250 baaria, mutta prosessiolosuhteet voivat myös vaihdella näistä, kuten tämän koelaitteen kanssa, joka on suunniteltu lievästi veden ylikriittiselle puolelle eli se ylittää veden ylikriittisen rajan, joka on 374 °C ja 221 baaria. (Gollagota ym., 2018, s. 1387)

HTL-prosessissa syntyvä nestemäinen bioöljy on prosessin päätuote, tämän lisäksi prosessissa syntyy kaasu-, vesi- ja kiintoainefaaseja sivutuotteina. Osaa sivutuotteista

voidaan hyödyntää, kuten vesifaasi voidaan uudelleen kierrättää prosessiin ja kiinteitä aineita voidaan käyttää lannoitteena tai katalyytteinä. (Gollagota ym., 2018, ss. 1387–1388)

Prosessin perusmekanismi koostuu kolmesta päävaiheesta, joita ovat biomassan depolymerointi, hajottaminen ja rekombinaatio. Depolymeroinnissa liuotetaan biomassan makromolekyylejä hyödyntämällä niiden fyysisiä sekä kemiallisia ominaisuuksia, tämä tapahtuu korkean lämpötilan sekä paineen avulla. Nämä olosuhteet muuntavat vedystä, hapesta ja hiilestä koostuvia pitkän ketjun polymeerejä lyhyemmän ketjun hiilivedyksi. HTL jäljittelee luonnollista miljoonien vuosien mittaista geologista fossiilisten polttoaineiden prosessia, mutta nopeammassa ja pienemmässä mittakaavassa. Depolymeroinnin jälkeen tapahtuu biomassan hajottaminen, josta seuraa sekä vesimolekyylien (dehydraatio) että CO<sub>2</sub>-molekyylien häviämistä (dekarboksylaatio) ja lisäksi aminohappopitoisuus laskee (deaminaatio). Dehydraatio ja dekarboksylaatio helpottavat hapen poistamista biomassasta veden ja hiilidioksidin muodossa. Viimeisessä eli rekombinaatiovaiheessa päinvastaisesti ensimmäiseen askeleeseen verrattuna reaktiiviset osat uudelleenpolymeroituvat vedyn puutteesta johtuen. Tämän seurauksena syntyy korkeamolekyyliä sisältäviä hiiliyhdisteitä. (Gollagota ym., 2018, ss. 1386–1387)

### 3 HMI

HMI tarkoittaa ihmisen ja koneen välistä käyttöliittymää ja se tulee englannin kielen sanoista Human Machine Interface. Se toimii nimensä mukaisesti ihmisen ja koneen välisenä rajapintana, joka mahdollistaa koneen, systeemin tai laitteen ohjaamisen. HMI-sanaa käytetään yleisesti teollisuudessa olevien laitteiden ja prosessien yhteydessä. HMI koostuu painikkeista, mittareista, graafeista, indikaattoreista ja monista muista elementistä, jotka on jaettu yhdelle tai useammalle näytölle eli sivulle. Edellä mainitut muodostavat yhdessä tarvittavat ohjaimet prosessin ohjausta ja seuranta varten. (Inductive Automation, 2018)

Käyttöliittymään tehdyt painikkeet sekä muut elementit yhdistetään PLC-ohjelmassa olevien I/O-muuttujien eli lähtö- ja tulomuuttujien kanssa. Tämän takia PLC- ja HMI-ohjelmistot tulee olla yhteensopivia. Tunnetuimmilla laitevalmistajilla on omat PLC- sekä HMI-ohjelmistot, jotka ovat luonnollisesti yhteensopivia keskenään, kuten myös käyttämälläni Beckhoff-valmistajalla. (Realpars, 2018)

HMI voi olla laitteessa kiinteästi rakennetulla näytöllä tai se voi olla esim. etänä puhelimesta, tabletilla tai tietokoneessa. Käytännössä HMI voi olla lähes millä tahansa näytöllä. Tässä projektissa näyttönä tulee olemaan perinteinen tietokoneen näyttö. (Inductive Automation, 2018)

Ohjelmoijalla on lähes rajattomat mahdollisuudet HMI-suunnittelun kanssa. Kuitenkin koska käyttöliittymällä on suuri merkitys prosessin turvallisen ja tehokkaan operoinnin kannalta, on usein hyvä seurata jo ennestään tehokkaiksi toimintatavoiksi todettuja ohjeita, suosituksia ja standardeja, vaikka niiden seuraaminen ei välttämättä olisikaan.

HMI-suunnitteluun julkaistiin heinäkuussa 2015 ensimmäinen versio ANSI/ISA-101.01-2015 standardista. Sitä voidaan pitää tämän hetken merkittävimpänä standardina HMI-suunnittelussa. Vuonna 2020 kyseinen standardi hyväksyttiin kehitettäväksi IEC-standardiksi, joka tulee vahvistamaan sen asemaa myös globaalisti. Sen nimi tulee olemaan IEC 63303. (ISA, 2020)

Standardi pitää sisällään ohjeita, periaatteita, filosofiaa ja terminologiaa tehokkaan ja monipuolisen HMI:n kehittämiseksi. Se myös ohjaa käyttöliittymäsuunnittelua kohti älykkäämpiä ja monipuolisempia käyttöliittymiä, vanhempien ASM- ja EEMUA 201 -suositusten tilalle. (ISA, 2020) Standardi tukee käyttöliittymäsuunnittelua käyttäjää ajatellen, jolloin käyttöliittymästä pyritään tekemään monipuolinen ja informatiivinen, mutta samalla kuitenkin mahdollisimman käyttäjäystävällinen. (Realpars, 2020)

Standardin ohjeistusten mukaan värimaailma olisi hyvä pitää maltillisena ja jättää räikeät ja kirkkaat värit kokonaan pois, ellei niiden käyttö ole aivan välttämätöntä. Erityisesti harmaita sävyjä suositetaan. (ISA, n.d.) Värien käyttö on yleinen ja tehokas tapa erotella objekteja sekä niiden tiloja toisistaan. Kuitenkin värien lisäksi on syytä olla myös muita erottelumenetelmiä. (ISA-101.01/2015, s. 23) Suunnitteluvaiheessa on syytä ottaa huomioon myös loppukäyttäjän mahdolliset rajoitteet, kuten värisokeus, jolloin tiettyjä väriyhdistelmiä on syytä välttää (ISA-101.01/2015, s. 36).

Käyttöliittymän tulisi olla johdonmukainen, jolloin käyttäjän on helpompi ymmärtää sitä ja tottua sen käyttöön (ISA-101.01/2015, s. 35). Kuvia voidaan käyttää käyttöliittymässä ja toisinaan ne ovat hyvä vaihtoehto, sillä niillä voidaan havainnollistaa asioita usein huomattavasti tehokkaammin kuin pelkällä tekstillä (ISA, n.d.). Myös animaatioita voidaan hyödyntää, kuten vilkutusta tai objektin liikettä. Nämä tulisi kuitenkin jättää vain siihen tarkoitukseen luoduille objekteille. Sillä esim. tekstin ja numeroiden lukeminen hankaloituu, mikäli ne vilkkuvat tai liikkuvat. (ISA-101.01/2015, s. 38) Näin ollen animaatioita tulee käyttää maltillisesti, sillä ne voivat olla myös häiritseviä käyttäjän kannalta, kuten tilanne, jossa moottorin ollessa päällä esitetään se pyörivänä liikkeenä käyttöliittymässä. Tämä voitaisiin yksinkertaisesti korvata merkkivalolla tai muulla käyttäjäystävällisellä indikaattorilla. Hyvä esimerkki animaatioista käyttöliittymässä on tilanne, jossa animaatiolla saadaan opastettua käyttäjää esim. prosessin ohjauksen kanssa. (ISA, n.d.)

Käyttöliittymässä olevien painikkeiden ja muiden elementtien sijoittelua kannattaa miettiä sen mukaan, kuinka usein niitä käytetään. Elementtien on hyvä olla myös aseteltu muiden elementtien kanssa kohdistetusti. (ISA, n.d.) HMI:ssä on myös suotavaa pystyä navigoimaan jokaiselle oleelliselle sivulle mahdollisimman vähillä klikkauksella pääsivulta. (ISA-101.01/2015, s. 35)

Painonappien, tekstilaatikoiden ja indikaattorien lisäksi käyttöliittymässä on hyvä esittää prosessia myös visuaalisesti, lisäämällä sinne prosessiin kuuluvia rakenteita, kuten tankkeja, toimilaitteita, putkituksia sekä muita prosessiin kuuluvia asioita. Tämä tulee kuitenkin toteuttaa käyttäjän kannalta helposti ymmärrettävästi. (Realpars, 2021) Sen toteutus voi olla esim. PI-kaaviotyylinen, mutta suotavaa on, että ei suorana kopiona siitä (ISA-101.01/2015, s. 40). Edellä mainitut asiat on syytä esittää käyttöliittymässä johdonmukaisesti. Esimerkiksi prosessiputket ovat suotavaa tehdä tumman harmaalla tai mustalla värillä, lisäksi niiden paksuus tulisi olla johdonmukainen niiden merkittävyyteen prosessissa. Toisin sanoen pääputket tehdään paksummalla viivalla kuin muut putket. Jokainen keskenään vastaava instrumentti tulee tehdä samankokoisiksi. Myös graafit ovat tärkeä osa tehokasta käyttöliittymää. Niiden avulla saadaan esitettyä käyttäjälle dataa helposti omaksuttavassa muodossa. (Realpars, 2021)

Käyttöliittymässä olisi hyvä olla myös älykkäitä toiminnallisuuksia, kuten hälytysten ja tapahtumien seuranta, josta nähdään myös aikaleima, milloin hälytys tai tapahtuma aktivoitui (ISA, n.d.). Tässä työssä käytetyssä Beckhoff-järjestelmässä tämä voidaan toteuttaa Event Logger -kirjastolla, joka on juuri hälytyksien ja tapahtumien kirjausta varten tarkoitettu. Tästä kerrotaan tarkemmin luvussa 5.2.

## 4 Ohjelmointiympäristö

Koelaite automatisoidaan käyttäen Beckhoffin PLC:tä, joten käytössä on luonnollisesti myös heidän kehittämä TwinCAT 3 -ohjelmointiympäristö lisäosineen ohjelmointia ja käyttöliittymää varten. TwinCAT on lyhenne sanoista The Windows Control and Automation Technology (Beckhoff, n.d. -b).

Eräänä Beckhoffin tavoitteista on ollut yksinkertaistaa ohjelmistojen käyttöä ja tästä syystä he eivät ole kehittäneet omaa erillistä ohjelmointiympäristöä, vaan he käyttävät Microsoft Visual Studiota pohjana, johon tämä Beckhoff TwinCAT 3 integroidaan. Tämä on ollut järkevä tapa, sillä useilla ohjelmoijilla on ohjelmointikokemusta Visual Studio -ympäristöstä, joten säästetään aikaa, joka kuluisi täysin uuden ohjelmiston opetteluun. (Beckhoff, n.d. -e, s. 8)

Tiedonsiirto ja TwinCAT-ohjelmistojen välinen kommunikointi tapahtuu Beckhoffin järjestelmissä ADS-protokollan välityksellä, joka tulee sanoista Automation Device Specification. ADS on Beckhoffin TwinCAT-ympäristöön kehittämä tiedonsiirtoprotokolla. (Beckhoff, n.d. -f)

Toisin kuin useilla muilla valmistajilla Beckhoffin TwinCAT-ohjelmistoa voidaan käyttää 7 päivän ilmaislisenssillä, jonka voi uusia rajattoman monta kertaa. Tämä mahdollistaa PLC-ohjelman luomisen ja testaamisen ilmaiseksi ennen lopullisen lisenssin hankkimista. Ilmaislisenssi sopii hyvin myös opetuskäyttöön. Lisenssiasiat hoidetaan TwinCAT-ohjelmiston sisällä projektipuussa löytyvältä License-sivulta. (Beckhoff, n.d. -a)

### 4.1 TwinCAT 3 Engineering

TwinCAT 3 Engineering tai TE1000 on TwinCAT-ympäristön pohja. TE1000 toimii ohjelmointiympäristönä, jossa ohjelmointi tapahtuu IEC 61131-3 -standardin mukaisilla PLC-ohjelmointikielillä, joita ovat IL (Instruction list), FBD (Function block diagram), ST (Structured Text), LD (Ladder diagram) ja SFC (Sequential function chart). Edellämainituista kielistä ST ja IL ovat tekstipohjaisia kieliä ja loput graafisia kieliä. Lisäksi ohjelmisto sisältää graafisen CFC-kielen (Continuous function chart) kääntäjän. TE1000 toimii perustana, johon kaikki muut tarvittavat lisäosat integroidaan myöhemmässä vaiheessa. (Beckhoff, n.d. -g)

Ohjelmointia voi tehdä myös C++-kielellä, mutta tällöin TwinCAT tulee integroida Microsoft Visual Studioon täysversion kanssa. Kuitenkin tässä tapauksessa myös Visual Studioon on ladattava C++-kielipaketti. Integroimisen voi tehdä ensimmäisellä kerralla TwinCATtia asentaessa tai myöhemmin ajamalla TwinCATin asennusohjelman. Mikäli TwinCAT on asennettu ennen Visual Studiota, ei integrointia voida suorittaa. Tällöin se on uudelleenasetettava Visual Studioon asentamisen jälkeen. (Beckhoff, n.d. -e, s. 8)

## 4.2 TwinCAT 3 HMI

TwinCAT 3 HMI tai TE2000 on TwinCAT-ympäristöön ladattava lisäosa, jolla tehdään ihmisen ja koneen välisiä käyttöliittymiä PLC-projekteille. TE2000 integroidaan Microsoft Visual Studioon kanssa asentamisen yhteydessä. Ohjelmisto on verkkopohjainen ja se käyttää viimeisimpiä verkkoteknologioita HTML5 ja JavaScriptiä, jotka mahdollistavat mm. saman käyttöliittymänäkymän käyttöä eri kuvasuhteissa erikokoisilla näytöillä. Ohjelmisto käyttää ensisijaisesti graafista WYSIWYG-editoria, jonka nimi tulee englannin kielen sanoista What You See Is What You Get eli mitä näet, sitä saat. Tarkemmin sanottuna muokatessa käyttöliittymää, näkymä vastaa hyvin pitkälti lopullista näkymää. Näin ollen ohjelmointitaidot eivät ole välttämättömät käyttöliittymää tehdessä. (Beckhoff, n.d. -c, s. 11)

Graafisen editorin pohjalla pyörii kuitenkin myös HTML-editori, joka on tekstipohjainen. Edellä mainitut WYSIWYG- sekä HTML-editori toimivat synkronoidusti, joka tarkoittaa sitä, että graafista editoria muokatessa, koodi generoituu automaattisesti myös HTML-editorin puolelle. Tästä koodieditorista voidaan seurata generoitua koodia ja sitä voidaan myös muokata sieltä. (Beckhoff, n.d. -c, ss. 43–46)

Käyttöliittymään luodut elementit, kuten painikkeet ja tekstipalkit voidaan liittää reaaliaikaisesti muuttujiin esim. PLC-muuttujiin ADS-protokollan avulla. Kaikkia käyttöliittymään tehtyjä toimintoja voidaan testata reaaliaikaisella datalla Live View -tilassa. (Beckhoff, n.d. -c, s. 11)



### 4.3 TwinCAT 3 HMI Server

TwinCAT 3 HMI Server eli TF2000 on verkkopalvelin, jonka kautta käyttäjä liittyy eli HMI:tä käytetään. Käyttäjä pääsee HMI:hin selaimessa IP-osoitteella tai vaihtoehtoisesti Hostname-tunnuksella. Käyttäjältä vaaditaan vain HTML5-yhteensopiva selain, joka on saatavilla kaikilla yleisimmillä käyttöjärjestelmillä. HMI Server -lisäosa asennetaan itse laitteelle eli PLC:lle, eikä sitä tarvitse asentaa kehitystietokoneelle. Luonnollisesti myös HMI Server käyttää ADS protokollaa ja näin ollen se pystyy kommunikoimaan kaikkien TwinCAT-laitteiden kanssa. Kolmannen osapuolen systeemien kanssa kommunikointi tapahtuu OPC UA -protokollalla. (Beckhoff, n.d. -d, s. 7)

## 5 Ohjelmointi

### 5.1 HMI-ohjelmiston käyttö

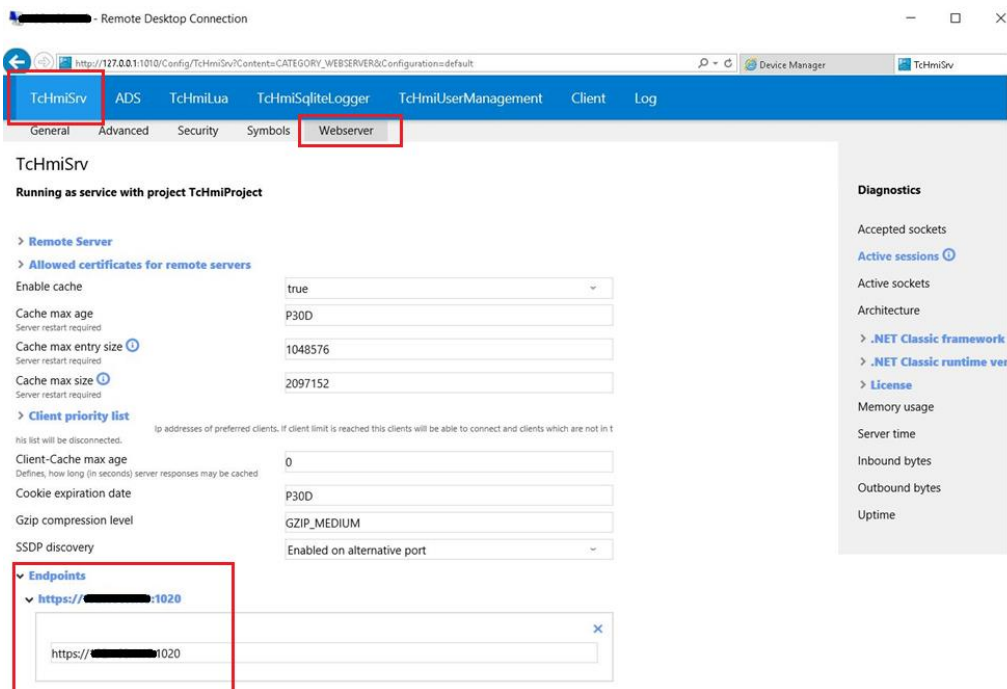
Tässä luvussa käydään läpi käyttöliittymän rakentamisen kannalta tärkeitä perustietoja TwinCAT HMI -ympäristössä. Ensimmäisenä käydään läpi HMI-projektin julkaisemiseen ja konfigurointiin liittyviä asioita. Tämän jälkeen aihealueina ovat mm. käyttöliittymäsivujen luominen ja niissä navigointi sekä tapahtumien, muuttujien ja parametrien luonti ja käyttö sekä paljon muuta.

#### 5.1.1 Konfigurointi ja HMI:n julkaiseminen

HMI:hin liittyviä konfigurointeja tarvitsee tehdä sekä kehitystietokoneen puolella että kohdelaitteessa, jossa HMI Server pyörii. Käydään ensiksi läpi kohdelaitteen konfigurointia. Jotta HMI saadaan toimimaan, on kohdelaitteelle eli itse serveriä pyörittävälle laitteelle ladattava HMI Server -lisäosa eli TF2000. Tässä tulee huomioida, että HMI Server on samaa versiota kuin kehityskoneella tehty HMI projekti. TF2000 ladataan Beckhoffin verkkosivuilta, kuten heidän muutkin lisäosat.

Latauksen jälkeen ja kun HMI Server on käynnistetty, täytyy sille luoda salasana, jotta HMI Serveriä päästään konfiguroimaan. HMI Serverin konfigurointisivu saadaan auki tehtäväpalkin oikeassa reunassa olevasta HMI Server pikkukuvakkeesta oikealla klikkauksella ja valitsemalla Config. Salasana luodaan tältä sivulta ensimmäisellä avauskerralla. Tätä salasanaa tarvitaan myöhemmin HMI:n julkaisuvaiheessa. Salasanan luomisen jälkeen kohdelaitteelle tulee määrittää Endpoints, kuvassa 2. olevasta paikasta. Sinne määritetään laitteen IP-osoite sekä käytössä oleva portti.

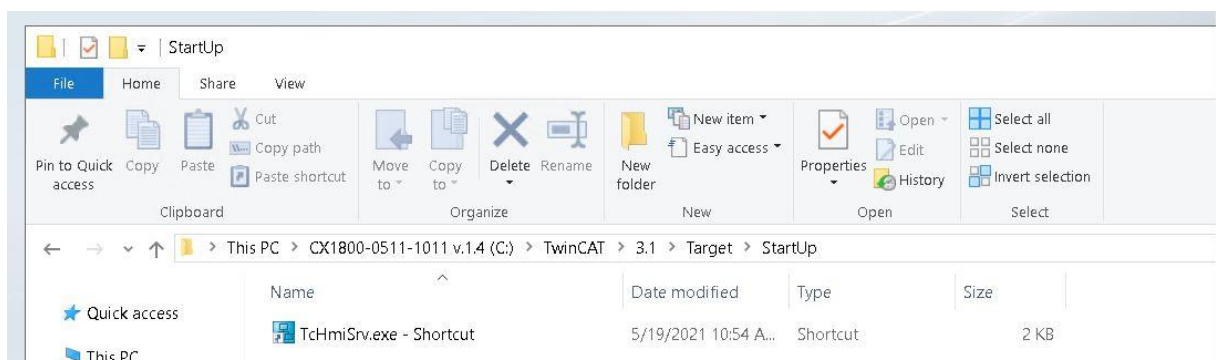
Kuva 2. HMI Serverin konfigurointisivu



Joissain tapauksissa kohdelaitteen palomuuuri saattaa estää yhteyden muodostamisen. Tässä tapauksessa tulee kohdelaitteen palomuurista sallia portti, jota HMI Server käyttää, kuten 1010 http:lle ja 1020 https:lle.

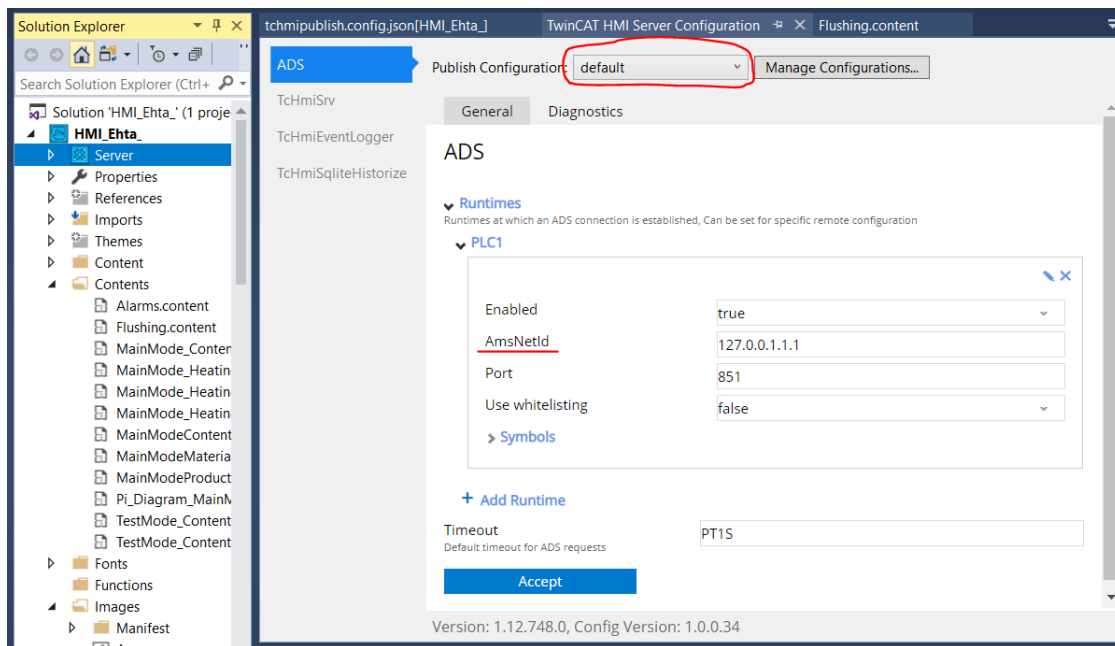
Vakioasetuksilla HMI Serveri ei käynnisty automaattisesti kohdelaitteen käynnistyessä. Tämä voidaan kuitenkin muuttaa lisäämällä HMI Serverin pikakuvake kuvan 3. mukaiseen kansioon.

Kuva 3. Hakemistosijainti HMI Serverin automaattista käynnistystä varten



Kehityskoneelta tulee konfiguroida siten, että HMI Server osoittaa siihen laitteeseen, jossa PLC-ohjelma pyörii. Tämä tapahtuu valitsemalla kuvassa 4. ympyröidystä paikasta default tai remote konfiguraation. Sekä määrittämällä ArmsNetId-kohtaan kohdelaitteeseen osoittava osoite.

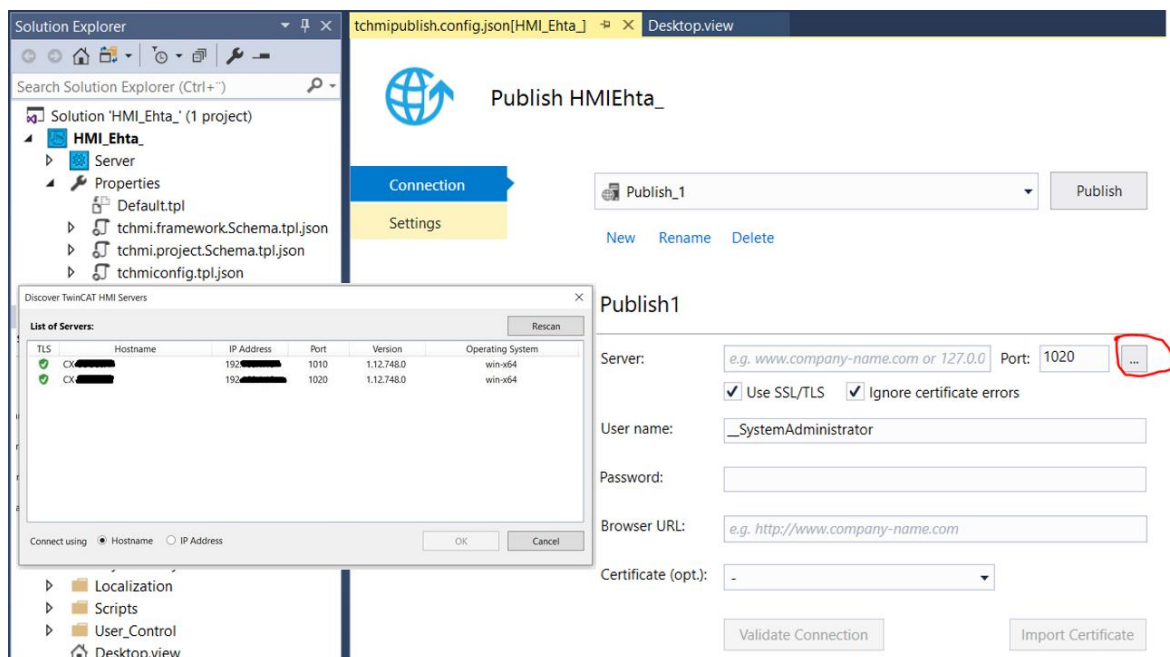
Kuva 4. HMI:n konfigurointi projektin sisältä



Mikäli HMI Server ja PLC pyörivät samassa laitteessa, kuten minun tapauksessani voi AmsNetId olla 127.0.0.1.1.1. Tämä viittaa paikalliseen koneeseen.

HMI:n julkaiseminen HMI Serverille on kohtuullisen suoraviivaista. Olennaisena asiana on määrittää oikea serveri ja portti. Nämä voidaan suoraan valita ympyröidystä paikasta kuvan 5. mukaisesti tai vaihtoehtoisesti kirjoittaa manuaalisesti kuvassa näkyviin Server ja Port kohtiin. Lisäksi aiemmin HMI Server -puolella luotu salasana tarvitsee kirjoittaa kuvassa 5. näkyvään Password-kohtaan.

Kuva 5. HMI:n julkaiseminen

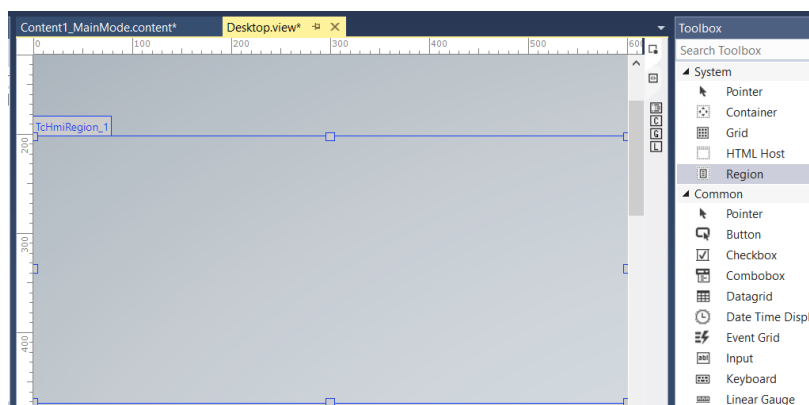


### 5.1.2 Perusrakenteen ja sivujen luominen

Käyttöliittymän aloitusnäky eli pääsivu on Desktop.view-niminen sivu, joka halutessaan voidaan vaihtaa toiselle sivulle. Vaihto tapahtuu lisäämällä projektiin uusi View-sivu ja asettamalla se aloitusnäkyksi.

Mikäli käyttöliittymään halutaan päänäkymän lisäksi muita sivuja, on päänäkymään välttämätöntä lisätä Region-alue Toolbox-välilehdeltä (Kuva 6.). Region on järkevää sijoittaa esimerkiksi kuvassa 6. näkyvällä tavalla, jossa jätetään yläosaan tyhjää tilaa.

Kuva 6. Region-elementin asettelu käyttöliittymän pääsivulle

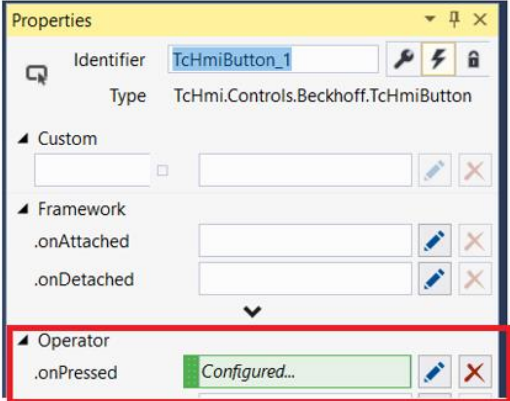
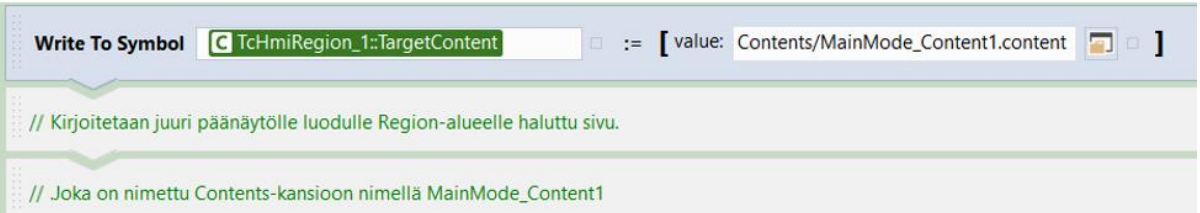


Mikäli sivuja on päänäkymään lisäksi muita, on järkevää, mutta ei välttämätöntä lisätä yksi tai useampi Container- tai Grid-alue. Ne löytyvät myös Toolbox-välilehdeltä. Alue sijoitetaan ylempänä näkyneen Region-alueen jättämille tyhjiille alueille. Tämä alue jää näkyviin, vaikka pääsivulta siirryttäisiin seuraavalle sivulle. Alueelle voidaan sijoittaa mitä tahansa haluttuja elementtejä, esimerkiksi painikkeita sivulta toiselle liikkumiseen, kielivalintapainikkeen sekä mahdollisen yrityksen logon tai muun halutun tekstin.

Uusien sivujen luominen kannattaa aloittaa lisäämällä projektipuuhun selkeyden kannalta uusi kansio ja nimetä se esimerkiksi nimellä Contents. Tämän jälkeen uusi sivu saadaan luotua kansioon klikkaamalla oikealla hiirenpainikkeella Contents-kansiota ja valitsemalla ADD-> New Item -> Content. Tämän jälkeen voidaan tämä juuri luotu Content-sivu muokata halutun näköiseksi.

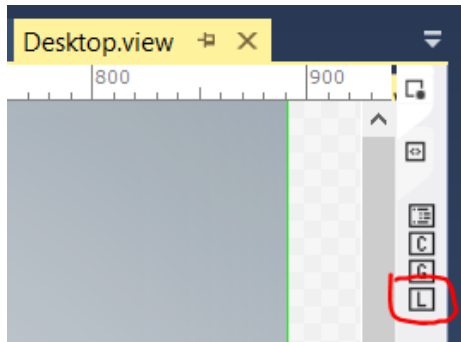
Content-sivu tulee näkymään ylempänä luodulla Region-alueella. Ensin luodaan painike sivujen vaihtoa varten päänäytölle tehdylle Container- tai Grid-alueelle. Tämän jälkeen painikkeeseen määritetään sivunvaihtotoiminto. Kuva 7. havainnollistaa, kuinka painikkeeseen määritetään sivunvaihto.

Kuva 7. Sivunvaihdon määrittäminen painikkeeseen

1. 
2. 

Painonappi on nyt valmis käyttöä varten ja sivunvaihtoa voidaan testata todellisessa käyttöliittymänäkymässä, jonka saa auki Live View -painikkeella (Kuva 8).

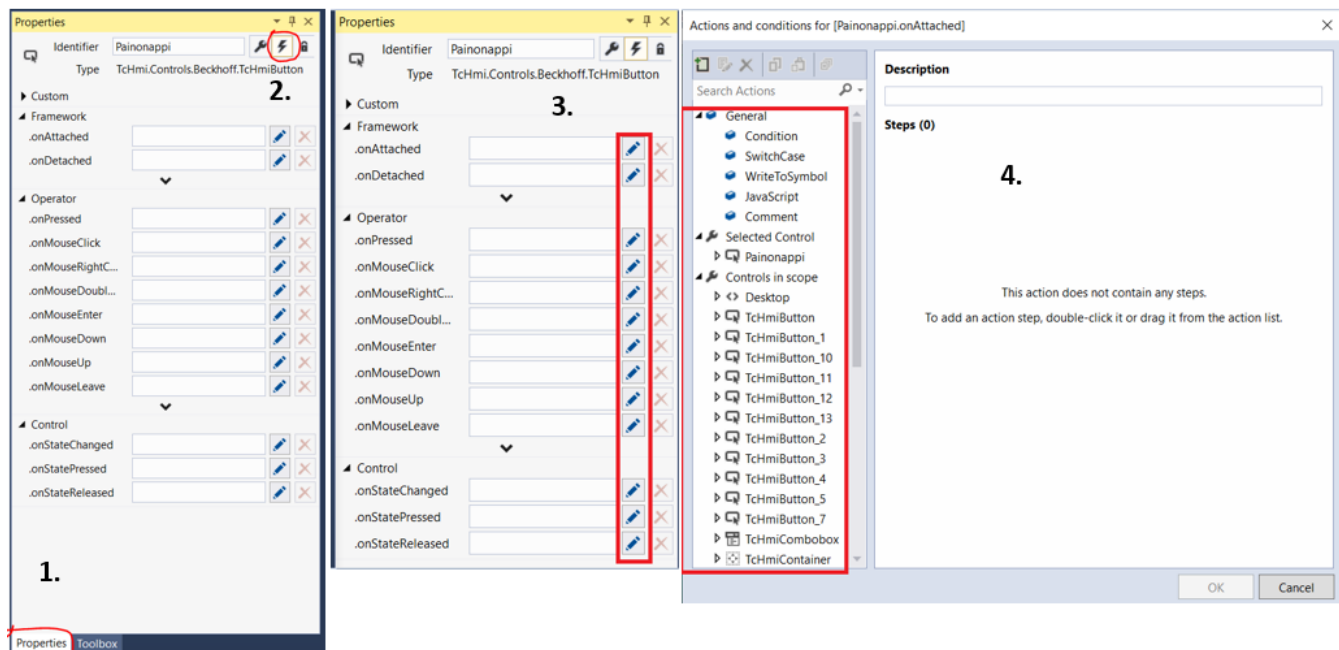
Kuva 8. Live View -painike



### 5.1.3 Toiminnot ja ehdot

Toiminnot ja ehdot luodaan halutun elementin Properties-sivun Events-välilehdeltä, kuvan 9. mukaisesti. Ensinnä valitaan mikä tekijä käynnistää tapahtuman. Esimerkiksi tapahtuma käynnistyy "onPressed" eli kun elementtiä painetaan. Tämän jälkeen luodaan tapahtuma.

Kuva 9. Toiminnot ja ehdot -ikkuna



Esittelen edellisen kuvan 9. Actions and conditions -kohdassa näkyvät General-toiminnot.

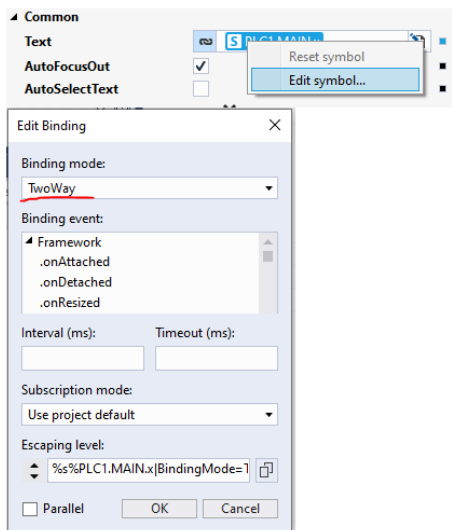
- Condition eli ehto on perinteinen ohjelmoinnista tuttu IF-lause. Pudotusvalikosta valitaan tarvittava vertailuoperaattori ja tarvittaessa lisätään AND- tai OR- operaattorit. Lisäksi ELSE IF -lauseita voidaan lisätä haluttu määrä.
- SwitchCase on vastaavasti perinteisestä ohjelmoinnista tuttu rakenne, jossa toimintoja toteutetaan askeleittain. Tätä varten on usein hyvä luoda sisäinen muuttuja askeleiden numerointia varten.
- WriteToSymbol on hyvin tärkeä osa tapahtumien luomista. Sillä tämä nimensä mukaisesti kirjoittaa haluttuun muuttujaan, elementtiin tms. halutun arvon.
- JavaScript on skriptikieli, jolla voidaan tehdä monipuolisia ominaisuuksia käyttöliittymään. Kerron tästä tarkemmin luvussa 5.1.9.
- Comment on perinteinen ohjelmoinnista tuttu toiminto. Se on tarkoitettu ohjelmoijan omaksi selitykseksi tai merkinnäksi ohjelmakoodista.

#### **5.1.4 Arvojen syöttäminen käyttöliittymään**

Arvojen syöttäminen on oleellinen asia käyttöliittymässä, sillä niillä voidaan säätää prosessin ohjausta PLC-puolelle. Arvojen syöttäminen käyttöliittymästä on helppoa. Sillä Toolboxista löytyy Input-niminen elementti, johon tarvitsee vain kiinnittää haluttu muuttuja, jonka arvoa käyttäjä haluaa muuttaa. Lisäksi Input-elementtiä tulee konfiguroida siten, että se on kaksisuuntainen eli se kykenee sekä lukemaan muuttujaa että kirjoittamaan siihen. Konfigurointi tapahtuu elementin Properties-sivulta kuvan 10 mukaisesti.



Kuva 10. Input-elementin konfigurointi



Käyttäjä voi kirjoittaa käyttöliittymänäkymässä halutun numeroarvon tai tekstin omalta näppäimistöltä. Mikäli omaa fyysistä näppäimistöä ei ole käytössä, voidaan käyttöliittymänäkymässä käyttää Keyboard-elementtiä, jolloin näytölle saadaan näppäimistö. Sillä tämä vakionäppäimistö on hyvin suuri ja vie tilaa paljon näytöltä, on mahdollista ladata projektipuusta erilaisia näppäimistöasetelmiä. Esimerkiksi jos käyttäjän on tarve syöttää vain numeroarvoja, on järkevää ladata näppäimistö, jossa on vain numeronäppäimet. Mikäli näin haluaa tehdä, suunnataan projektipuusta ennestään löytyvän KeyboardLayouts-kansion kohdalle, painetaan hiiren oikeaa ja valitaan ADD → New Item → Keyboard Layout. Avautuu valikko erilaisia näppäimistöasetelmiä. Niistä valitaan Numpad-niminen näppäimistö ja lisätään se projektiin.

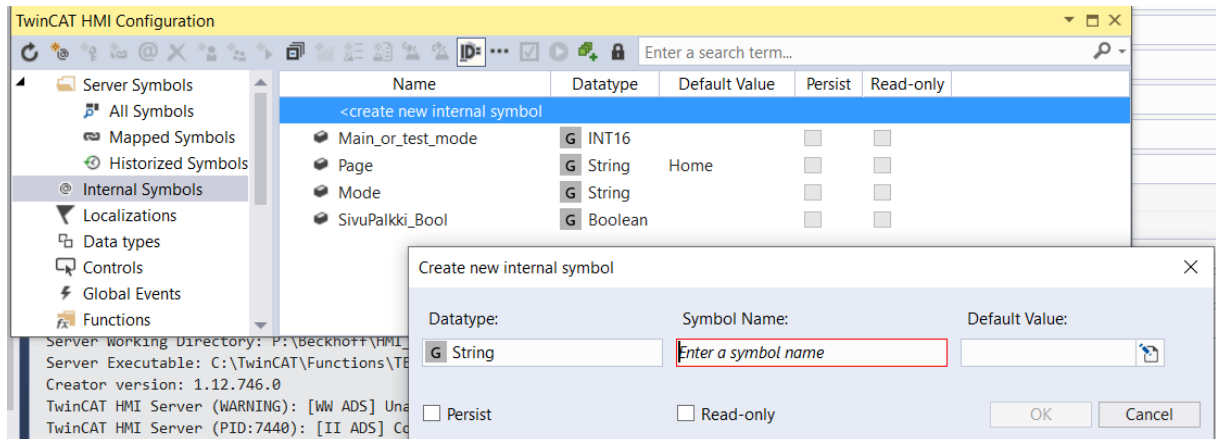
Joissakin tapauksissa arvojen syöttämistä varten on järkevä tehdä JavaScript-kielinen ohjelmakoodi, jolla estetään käyttäjää syöttämästä arvoja halutun numeroalueen ulkopuolelta. Käyttäjää voidaan myös esim. estää syöttämästä merkkejä, jotka eivät ole numeroita. Luvussa 5.1.9 esittelen tarkemmin tähän tarkoitukseen olevaa JavaScript-koodia.

### 5.1.5 Sisäiset ja ulkoiset muuttujat

HMI:ssä voidaan käyttää useita erityyppisiä muuttujia, joista aiemmissa luvuissa kävimme jo muutaman läpi. Tässä luvussa perehdymme sisäisiin Internal Symbols -muuttujiin sekä ulkoisiin PLC-muuttujiin. Muuttujat löytyvät Configuration-sivulta. Itse suosin sisäisten

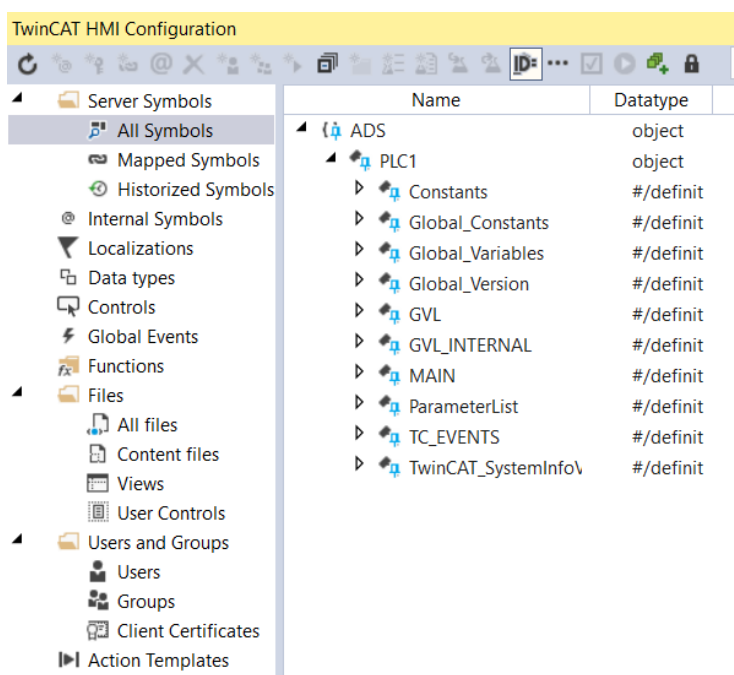
muuttujien käyttöä silloin, kun muuttujaa ei tarvita PLC-puolella ollenkaan. Vaikka ohjelman toiminnan kannalta ei ole suurta merkitystä, onko muuttuja tehty HMI- vai PLC-puolelle. Sisäinen muuttuja voidaan luoda Configuration-sivulla Internal Symbols -kohdassa (Kuva 11), jossa sille määritetään haluttu tietotyyppi, nimi ja alkuarvo.

Kuva 11. Sisäisen muuttujan luominen



Ulkoiset muuttujat, kuten PLC-muuttujat vaativat HMI:n ja PLC:n välisen yhteyden ADS-protokollan avulla. Mikäli tätä yhteyttä ei ole, ei PLC-muuttujia saada näkyviin HMI:ssä. Kuvassa 12. nähdään Configuration-sivu sekä ADS:n kautta löytyneet ulkoiset muuttujat.

Kuva 12. ADS:n välityksellä vastaanotetut PLC-muuttujat



Toinen vaihtoehto saada ulkoiset muuttujat käyttöön on TMC-tiedosto. TMC-tiedostoa käytetään luokkatietojen määrittämiseen, se sisältää mm. muuttujien tietotyypit, data-alueet ja pointerit. (Beckhoff, n.d. -h)

TMC-tiedosto saadaan PLC-projektin puolelta. Se tuodaan tämän jälkeen HMI-projektiin HMI Configuration -sivulta Attach TMC File -kohdasta, joka nähdään aiemman kuvan 12. oikeassa yläreunassa. TMC-tiedosto on hyvä tapa käyttää PLC-projektin muuttujia, mikäli projekteja tehdään erillisinä toisistaan esim. kahdella eri tietokoneella. Koska TMC:t ovat vain muuttujien määrittäjiä, täytyy ne kartoittaa oikeiksi muuttujiksi viimeistään siinä vaiheessa, kun sekä PLC- että HMI-projekti halutaan saada ohjelmoitavan logiikan sisälle. (Beckhoff, n.d. -i)

#### **5.1.6 Kuvatiedosto käyttöliittymässä**

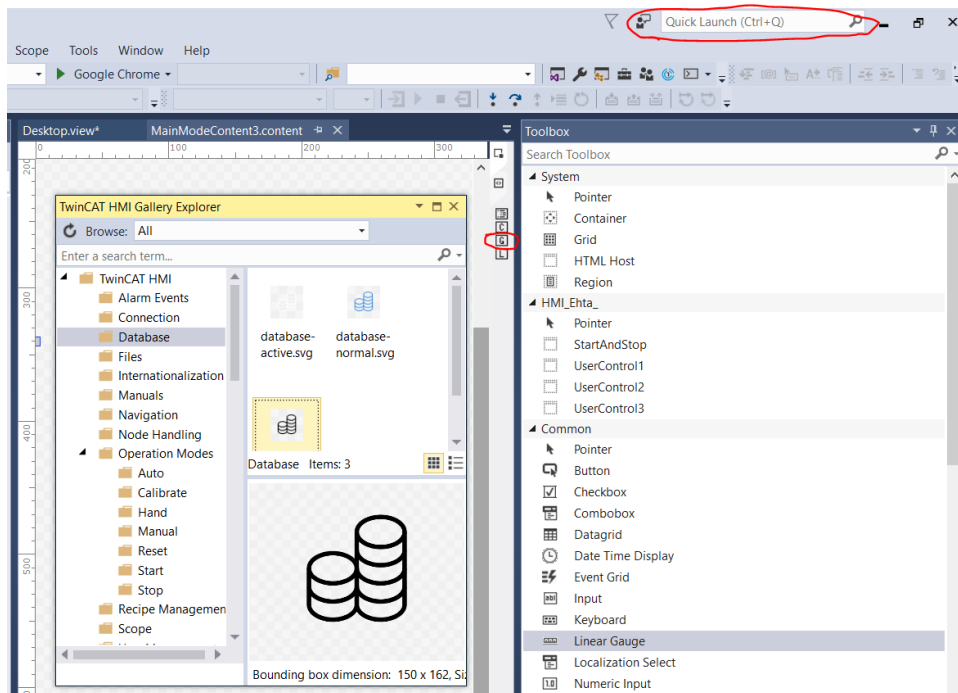
Käyttöliittymään voidaan lisätä myös kuvia projektipuuhun tallennetuista kuvista.

Projektipuussa on valmiina Images-kansio, johon kuvat ovat järjestyksessä tallentaa. Ensiksi kuva tuodaan projektipuuhun napauttamalla oikeaa hiirenpainiketta kansion päällä ja valitsemalla ADD → Existing Item. Kuva voidaan tuoda projektipuuhun myös Drag and Drop -menetelmällä.

Haluttu kuva kiinnitetään pohjakuvaksi haluttuun elementtiin käyttöliittymässä, elementin Properties-näkymästä. Kuva voidaan myös Drag and Drop -menetelmällä siirtää projektipuusta käyttöliittymään, jolloin kuva kiinnittyy automaattisesti Toolboxista löytyvään Image-elementtiin.

Beckhoffilla on myös kuvagalleria. Näitä valmiita kuvatiedostoja voi tuoda projektiin HMI Gallery Explorer -sivulta, jonka saa auki kuvan 13. paikoista.

Kuva 13. HMI Gallery Explorer



Galleriassa on suuri määrä erilaisia kuvia, joiden avulla voidaan tehdä esim. visuaalisesti siistin näköisiä painonappeja (Kuva 14). Näin ollen, kun halutaan lisätä jokin kuvatiedosto käyttöliittymään, on järkevää tarkastaa, löytyykö sellaista entuudestaan jo kyseisestä kuvagalleriasta.

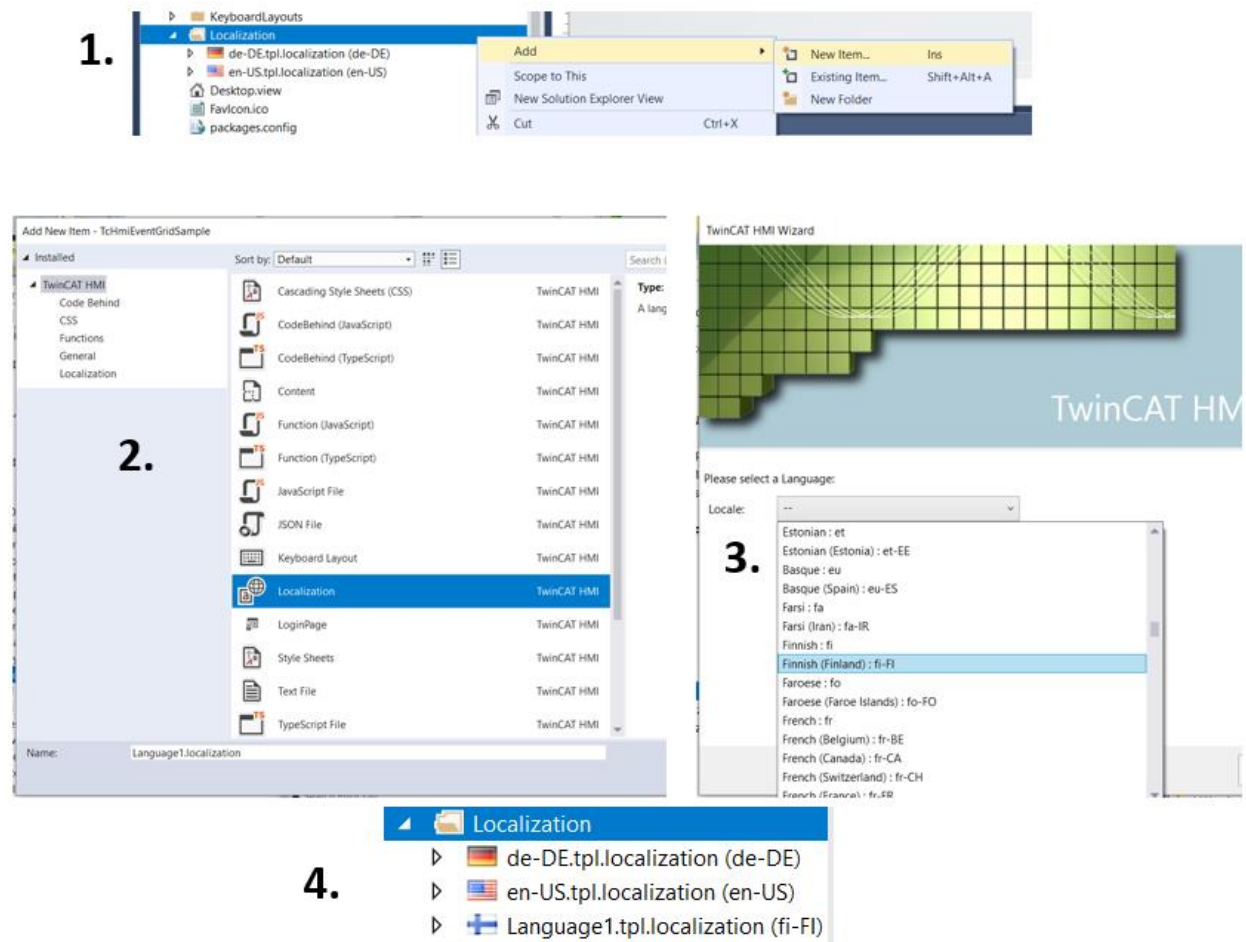
Kuva 14. Painonappeja, joissa käytetty HMI Gallery Explorer -kuvia



### 5.1.7 Kielivalinta

Kielipaketit löytyvät projektipuusta Localization-kansiosta, mikäli kansion nimeä ei ole muokattu. Jos kansioista puuttuu haluttu kieli esim. Suomi, voidaan se sinne lisätä painamalla hiiren oikeaa painiketta ja valitsemalla ADD → New Item. Aukeaa valikko, mistä valitaan Localization, tämän jälkeen kielivalikosta valitaan Suomi, joka esiintyy siellä nimellä Finnish (Finland): fi-Fi. Lopuksi tarkistetaan ja todetaan suomen kielen ilmestyneen Localization-kansioon. Kuvassa 15. havainnollistetaan kielipaketin asentaminen.

Kuva 15. Uuden kielipaketin tuominen projektiin



Muuttujat luodaan kaksoisklikkaamalla Localization-kansiossa olevia kieliä. Se avaa kuvassa 16. olevan näkymän, jossa voidaan halutulle kielellä määrittää haluttu teksti. Muuttujia ei tarvitse tehdä samalla nimellä usean eri kielen alle, vaan ne tulevat näkyviin jokaisen kielen alle heti ensimmäisen luontikerran jälkeen. Kyseisiä muuttujia voi tehdä myös Configuration-lehdellä.

Kuva 16. Localization-muuttujat

Muuttujan nimi	Haluttu viesti englannin kielellä	Muuttujan nimi	Haluttu viesti suomen kielellä
Key	en-US	Key	fi-FI
Variable_x	Hello World!	Variable_x	Moi Maailma!
Variable_y	I'm a localized text.	Variable_y	Olen lokalisoitu muuttuja
Start	Start	Start	Käynnistys
Stop	Stop	Stop	Pysäytys
Language	Language	Language	Kieli
Text1	Text1	Text1	Teksti 1
Text2	Text 2	Text2	Teksti 2
<create new localized symbol>	<create new localized symbol>	<create new localized symbol>	<create new localized symbol>

Tämän jälkeen voidaan luotuja muuttujia kiinnittää haluttuihin elementteihin, kuten tekstilaatikoihin tai painonappeihin, jolloin niissä oleva teksti on näkyvissä käyttäjän valitsemalla kielellä. Yksinkertaisin tapa tehdä kielen valinta HMI:ssä on käyttää valmista Localization Select -pudotusvalikkoa, joka löytyy Toolboxista. Siinä kielipaketit on valmiiksi kiinnitetty elementtiin. Kuvassa 17. esitellään käyttöliittymänäkymä pudotusvalikosta, jossa käyttäjälle on mahdollisuus valita haluttu kieli projektin tallennetuista kielipaketeista.

Kuva 17. Localization Select -elementti



Mikäli kielivalinta halutaan tehdä HMI:hin käyttämällä jotain muuta elementtiä, on elementtiin tuleva kielipakettien data luettava `GetRegisteredLocales()`-funktioilla. Lisäksi elementtiin on tehtävä toiminto, kuten ehtolause, jolla haluttu kieli voidaan valita. (Beckhoff, n.d. -c, s. 551)

### 5.1.8 Kuvaajat

Kuten muutkin elementit myös kuvaajat löytyvät Toolbox-sivulta. Kuvaajatyyppejä on kahdenlaisia – Line Chart sekä Trend Line Chart. Näistä ensimmäinen on tarkoitettu reaaliaikaisen datan esittämiseen ja toinen aiempaa dataa eli historoitua dataa esittämään.

Historointitoiminnot saadaan käyttöön lataamalla HMI-projektiin `HMI.SqliteHistorize`-niminen laajennus. Laajennus mahdollistaa muuttujien datan tallentamisen SQLite-tietokantaan, josta halutut muuttujat saadaan näkymään kuvaajassa halutulla aikavälillä. Dataa tallennetaan tietokantaan vain, kun palvelin on päällä. Historoitua dataa voidaan käyttää vain Trend Line Chartin kanssa. Graafissa voidaan käyttää useita muuttujia samanaikaisesti, jolloin jokaisesta piirtyy oma kuvaajansa. (Beckhoff, n.d. -c, s. 637)

### 5.1.9 Skriptikielet

TwinCAT HMI -ympäristössä voidaan tehdä toimintoja myös JavaScript- ja TypeScript-kielillä. JavaScript on tekstipohjainen skriptikieli. Sitä käytetään pääasiassa web-pohjaisiin sovelluksiin ja verkkoselaimiin, mutta sitä voidaan käyttää myös verkon ulkopuolisissa ohjelmistoissa.

TypeScript on Microsoftin kehittämä ja ylläpitämä avoimeen lähdekoodiin perustuva tekstipohjainen olio-ohjelmointikieli. TypeScript on suunniteltu suurien ohjelmistojen kehittämiseen ja se tarjoaa monia etuja JavaScriptiin verrattuna, esim. ominaisuuksiltaan se on helpommin luettavaa ja ymmärrettävää. Yksinkertaistettuna TypeScript on kuin JavaScript, mutta lisäominaisuuksilla. Valmis TypeScript-koodi käännetään JavaScript-koodiksi, koska kääntäjä ei suoraan ymmärrä TypeScript-koodia. (DZone, 2018)

TwinCAT HMI -ympäristössä skriptikoodia voidaan käyttää kolmella eri tavalla:

- Action-tapa, jolloin koodi kirjoitetaan suoraan Actions and Conditions -editoriin halutun elementin Properties-sivulta. Tätä tapaa käytetään yksinkertaisien JavaScript-toimintojen kanssa. Myöhemmin näkyvässä JavaScript-koodiesimerkissä on koodi tehty juuri tällä tavalla. Tämä tapa ei tue TypeScript-kieltä.
- Function-tapa on tapa, jossa koodi koostuu JavaScript- tai TypeScript-koodista, ja joka lisätään erillisenä osana projektiin. Tämän jälkeen skriptiä käytetään kutsumalla kyseistä funktiota Actions and Conditions -editorista.
- Code-Behind koostuu JavaScript- tai TypeScript-koodista ja sitä käytetään monimutkaisten skriptien suorittamiseen. Se nimensä mukaisesti pyörii taustalla globaalisti ja jonka toimintoja voidaan käyttää kaikkialla projektissa.

Skriptikielet mahdollistavat monimutkaisten toimintojen toteutuksen, myös sellaisien mitä ilman niitä ei käyttöliittymään pystyisi tekemään. TwinCAT HMI on tukenut JavaScript-kieltä 1.8 versiosta alkaen ja TypeScript-kieltä 1.12 versiosta alkaen. (Beckhoff, n.d. -c, s. 3659)

Tässä luvussa perehdymme JavaScriptin käyttöön Actions and Conditions -editorissa. Sillä voidaan tehdä kaikki 5.1.3 luvussa mainitut ehtolauseet, symboleihin kirjoittamiset jne. Lisäksi sillä voi tehdä paljon muutakin, kuten ponnahdusikkunoita sekä muita interaktiivisia toimintoja. Alla näkyvässä taulukossa 1. nähdään JavaScriptin syntaksi käyttäessä erilaisia muuttujia TwinCAT-ympäristössä.

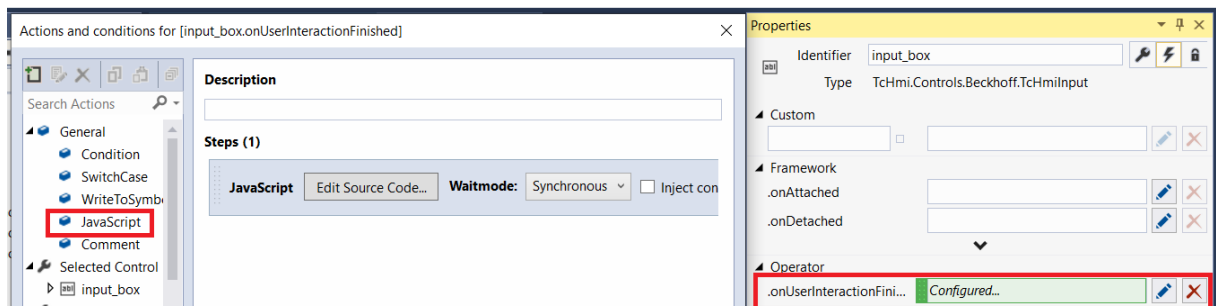
Taulukko 1. JavaScript-syntaksi erilaisille muuttujille TwinCAT-ympäristössä (Mueller, 2020)

<b>Tyyppi</b> (Muuttujat, parametrit, elementit jne.)	<b>JavaScript-syntaksi</b> "... " korvataan muuttujan nimellä
<b>Server symbols</b> (esim. PLC-muuttujat)	<code>%s%...%/s%</code>
<b>Internal symbols</b> (Sisäiset muuttujat)	<code>%i%...%/i%</code>
<b>Localized symbols</b> (Kielimuuttujat)	<code>%l%...%/l%</code>
<b>User Control parameters</b> (UC-parametrit)	<code>%pp%...%/pp%</code>
<b>Action template parameters</b> (AT-parametrit)	<code>%tp%...%/tp%</code>
<b>Functions</b> (Aiemmin mainitut JavaScript-funktiot)	<code>%f%...%/f%</code>
<b>Controls</b> (Elementit, esim. tekstilaatikko tai painonappi)	<code>%ctrl%...%/ctrl%</code>



Teen esimerkkitilanteen JavaScriptin käytöstä. Tässä tapauksessa elementtinä on Input-laatikko. Luvussa 5.1.4 kävimme läpi arvojen syöttämistä käyttöliittymästä. Jotta arvojen syöttäminen olisi toimivaa voidaan JavaScript-koodin avulla tehdä siihen hyödyllisiä lisäominaisuuksia. Tässä esimerkissä tarvittavia lisätoimintoja varten on ensiksi tehtävä kolme sisäistä muuttujaa, jotka minulla on nimetty Value, MAX ja MIN nimillä. Lisäksi Input-laatikkoon tehdään JavaScript-toiminto Actions and Conditions -editorista (Kuva 18).

Kuva 18. Esimerkissä käytetty tilanne, jolloin tapahtuma aktivoituu



Ohjelmakoodi (Kuva 19) tiedottaa käyttäjää Input-laatikkoon syötettyjen arvojen perusteella. Se vertailee ennestään määriteltyjä rajoja, jotka ohjelmakoodissa ovat Max- ja Min-muuttujat ja antaa käyttäjän syöttää luvun siihen ainoastaan, jos syötetty luku osuu niiden väliin. Kuvassa 19. nähdään myös, kuinka aiemmin taulukossa 1. nähtyjä muuttujia voidaan lukea *readEx*-toiminnolla ja kuinka niihin voidaan kirjoittaa *writeEx*-toiminnolla.

Kuva 19. JavaScript-koodi

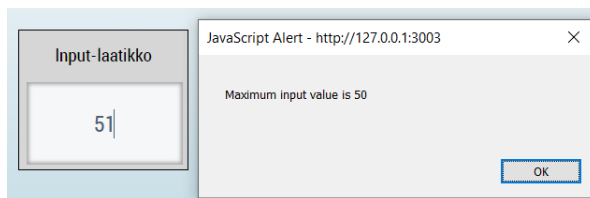
```

JavaScript
1 //Luodaan muuttujat
2 var oldValue = TcHmi.Symbol.readEx('%i%Value%/i%'); //Saa arvonsa sisäisestä Value-muuttujasta
3 var newValue = TcHmi.Symbol.readEx('%ctrl%input_box::Text%/ctrl%'); //Saa arvonsa Input-boxiin syötetyistä arvoista
4 var Max = TcHmi.Symbol.readEx('%i%MAX%/i%'); //Saa arvonsa MAX-nimisestä sisäisestä muuttujasta
5 var Min = TcHmi.Symbol.readEx('%i%MIN%/i%'); //Saa arvonsa MIN-nimisestä sisäisestä muuttujasta
6
7 //Vertailu
8 if (newValue > Max) { //jos uusi arvo on suurempi kuin maksimiarvo
9     TcHmi.Symbol.writeEx('%ctrl%input_box::Text%/ctrl%', oldValue); //Palautetaan aiempi arvo Input-laatikkoon
10    alert('Maximum input value is ' + Max); //Nousee ponnahdusikkuna sulkujen sisällä olevalla tekstillä + MAX-arvolla
11 }
12 } else if (newValue < Min) { //Jos uusi arvo on pienempi kuin minimiarvo
13     TcHmi.Symbol.writeEx('%ctrl%input_box::Text%/ctrl%', oldValue); //Palautetaan aiempi arvo Input-laatikkoon
14     alert('Minimum input value is ' + Min); //Nousee ponnahdusikkuna sulkujen sisällä olevalla tekstillä + MIN-arvolla
15 }
16 } else if (isNaN(newValue)) { //Mikäli syötetty arvo ei ole numero
17     TcHmi.Symbol.writeEx('%ctrl%input_box::Text%/ctrl%', oldValue);
18     alert('Input must be a number!'); //Nousee ponnahdusikkuna sulkujen sisällä olealla tekstillä
19 } else {
20     //Kirjoittaa input-elementtiin käyttäjän kirjoittaman luvun
21     TcHmi.Symbol.writeEx('%ctrl%input_box::Text%/ctrl%', newValue);
22 };

```

Mikäli käyttäjä syöttää liian ison tai pienen arvon nousee ponnahdusikkuna, jossa huomautetaan asiasta ja kerrotaan, mikä on maksimiarvo tai minimiarvo riippuen siitä, mitä syötetty arvo oli (Kuva 20). Vastaavasti mikäli käyttäjä syöttää arvon, joka ei ole numeroarvo, nousee ponnahdusikkuna, joka huomauttaa asiasta.

Kuva 20. JavaScript-ponnahdusikkuna todellisessa näkymässä



JavaScriptille voidaan ladata myös lisäkirjastoja, jotka tuovat vielä entisestään enemmän toimintoja. jQuery on eräs erittäin suosittu monipuolinen avoimen lähdekoodin JavaScript-kirjasto, joka on sekä yritysten että itsenäisten ohjelmistokehittäjien suosiossa. Kirjaston päätavoitteena on tuoda JavaScriptiin monipuolisia ominaisuuksia syntaksiltaan helposti ymmärrettävinä kirjastofunktiona. (w3schools, n.d.)

jQuery voidaan ottaa käyttöön myös Beckhoff-ympäristössä tuoden projektille valtavasti lisäominaisuuksia, kuten elementtien raahaamis- ja skaalaus-, sijoitus-, himmennys-, korostus- sekä lukuisia muita ominaisuuksia. (jQuery user interface, n.d.)

jQuery ladataan HMI-projektiin samalla tavalla, kuin muutkin kirjastot eli painamalla Manage NuGetPackages oikealla hiiren painikkeella projektipuusta. Sieltä tulee ladata sekä jQuery-että jQuery.UI-kirjastot.

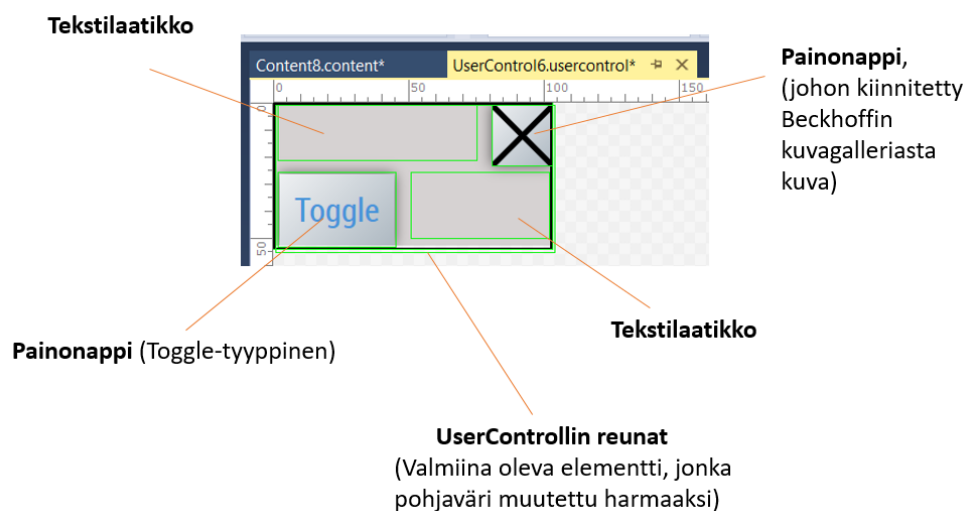
#### 5.1.10 Omien elementtien luominen ja käyttö

On mahdollista yhdistää useita elementtejä ja tehdä niistä yksittäinen UserControl-elementti. Näitä varten on suotavaa tehdä uusi kansio projektipuuhun esimerkiksi nimellä UserControls. UserControl luodaan kansioon ADD -> New Item -> User Control. Kun tämä juuri luotu UserControl avataan, on ensimmäisenä syytä selvittää minkä kokoinen elementti halutaan tehdä, joten kannattaa ensimmäiseksi säätää skaalaus kuntoon. UserControllin skaalausta muutetaan vastaavasti, kuten mitä tahansa muutakin elementtiä. Tämän jälkeen

näkymään lisätään halutut elementit Toolbox-välilehdeltä samaan tapaan, kuin muille sivuille.

Seuraavaksi esimerkki UserControlin käytöstä. Tämä elementti koostuu kahdesta painonapista ja kahdesta tekstikentästä, jotka ovat tuotu valmiina olevalle UserControl-pohjalle (Kuva 21). Kyseisellä UserControl-elementillä halutaan ohjata ON/OFF-venttiilin toimintaa sekä nähdä venttiilin tila reaaliaikaisesti.

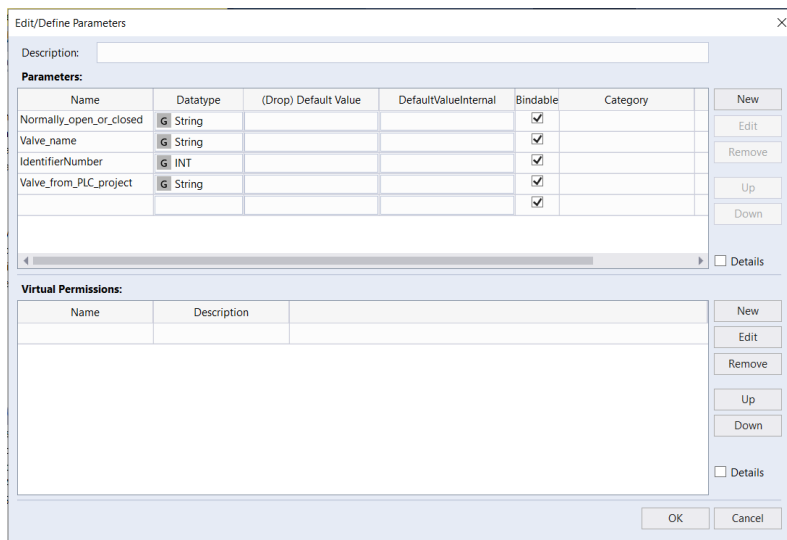
Kuva 21. Esimerkki UserControl-elementistä



UserControllille halutaan usein luoda omia parametreja, jotka vastaavat tavallisia muuttujia. Niille määritetään nimi, tietotyyppi ja lähtöarvo, mutta ne toimivat vain kyseisen UserControlin sisällä. Näitä parametreja käytetään erityisesti luomaan UserControlille monipuolisuutta, jolloin usea samanlainen UserControl-elementti pystyvät kaikki näyttämään erilaisia arvoja, eivätkä ole täysiä kopioita toisistaan. Luon neljä parametria näitä ylemmässä kuvassa olevia elementtejä varten. Parametrit luodaan UserControlille Edit/Define Parameters -välilehdellä.

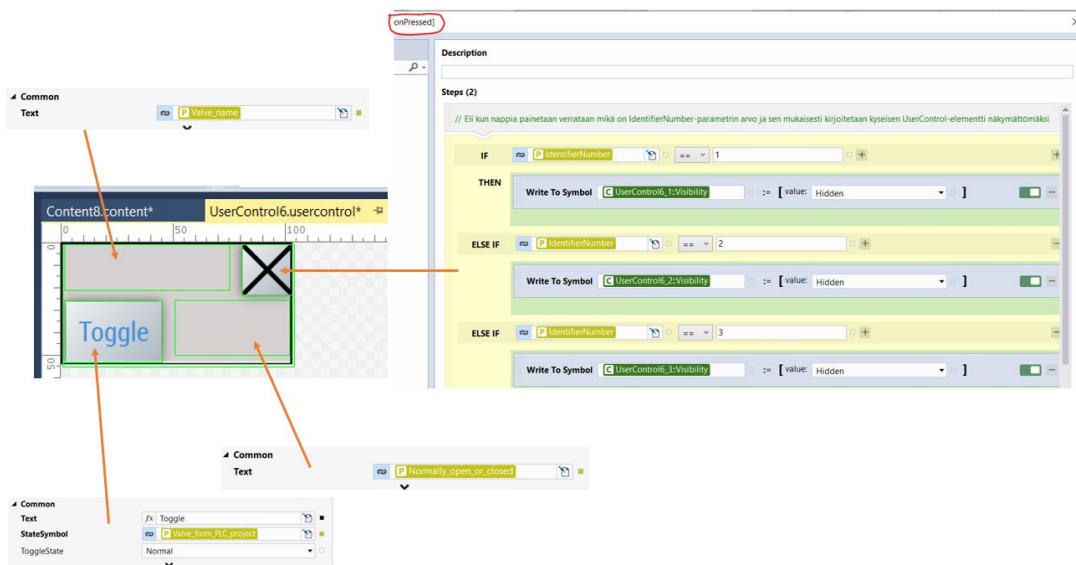
Kuvassa 22. nähdään Edit/Define Parameters -sivun ulkoasu sekä sinne luodut parametrit. Tässä olevat parametrit on luotu, että venttiililtä pystytään määrittämään nimi, tila ja onko venttiili normaalitilanteessa auki vai kiinni. Tein lisäksi IdentifierNumber-parametrin jonka avulla eritellään nämä UserControl-elementit toisistaan, mikäli samaa elementtiä on useampi kappale käyttöliittymässä.

Kuva 22. UserControl-parametri-ikkuna



Tämän jälkeen parametrit täytyy kiinnittää luotuihin elementteihin, elementtien Properties-näkymästä. Tässä esimerkkitapauksessa yhdistin parametrit elementteihin näin (Kuva 23).

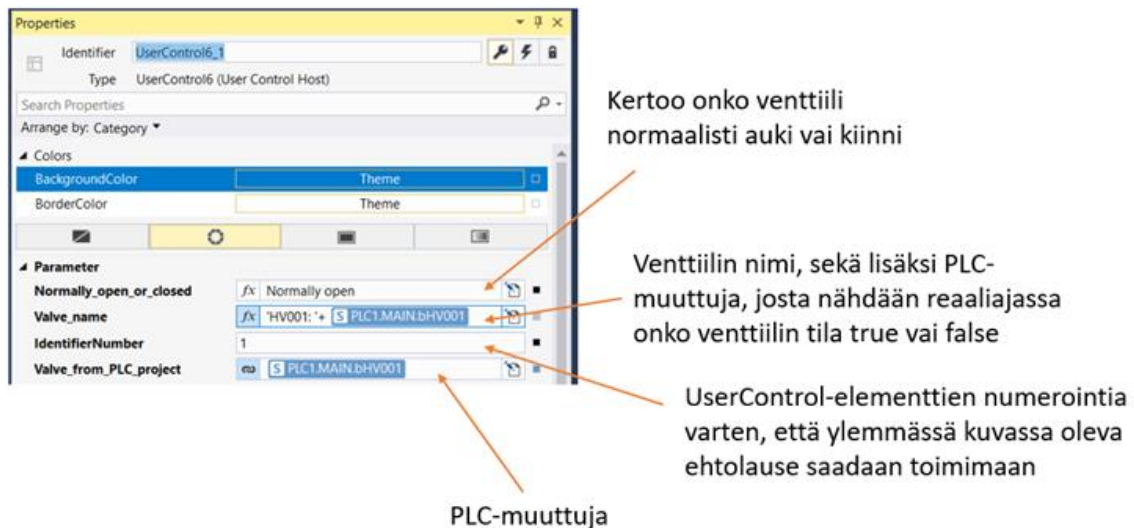
Kuva 23. Parametrien kiinnittäminen elementteihin kyseisessä esimerkkitilanteessa



Luotu UserControl-elementti tulee näkyviin muiden elementtien joukkoon Toolboxiin. Huomataan myös, että kun tämä elementti tuodaan HMI:hin ja siirrytään sen Properties-näytölle, on sinne ilmestynyt uusi Parameter-osio (Kuva 24). Sieltä pystytään muuttamaan näiden aiemmin luotujen parametrien lähtöarvoja. Jokaiselle yksittäiselle käyttöliittymään tuodulle UserControl-elementille määritetään nämä erikseen. Kuvassa 24. olen jo

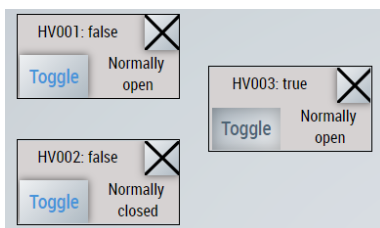
määrittänyt parametrien lähtöarvot. Tulee kuitenkin huomioida, että lähtöarvot eivät kuitenkaan tule näkyviin, mikäli parametreja ei ole kiinnitetty elementteihin, kuten aiemmassa kuvassa 23.

Kuva 24. Properties-ikkunaan ilmaantuneet UserControl-parametrit



Kuvassa 25. nähdään kolme kappaletta näitä itseluotuja elementtejä todellisessa käyttöliittymänäkymässä. Luotujen parametrien avulla nämä UserControl-elementit eivät ole täysiä kopioita keskenään, vaan jokainen pystyy avaamaan ja sulkemaan oman venttiilinsä ja kertomaan venttiilin normaalitilan sekä reaaliaikaisen true- tai false-tilan.

Kuva 25. Esimerkkitalanteen UserControl-elementti todellisessa näkymässä

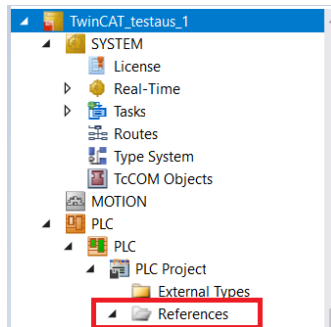


## 5.2 Hälytykset

Hälytyksiä voidaan tehdä usealla tavalla. Kuitenkin suotavaa on käyttää Beckhoffin Tc3\_EventLogger -kirjastoa, joka on tehty juuri hälytyksiä varten. Kirjasto ladataan sekä PLC-projektiin että HMI-projektiin erikseen.

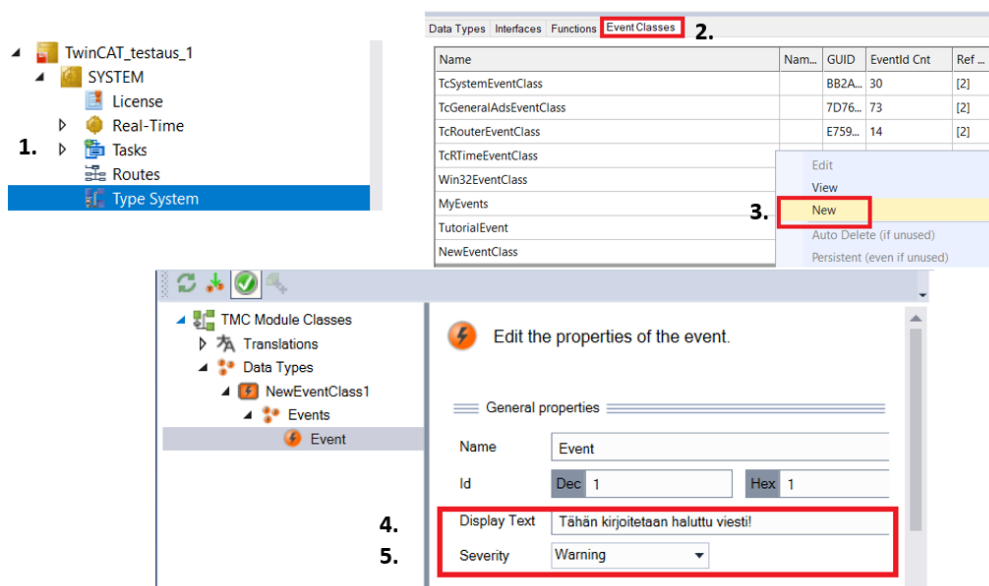
PLC-puolella kirjasto ladataan navigoimalla projektipuussa PLC-projektin alla olevaan References-kansioon kuvan 26. mukaisesti. Avataan kansio ja kirjoitetaan hakukenttään Tc3\_EventLogger.

Kuva 26. References-kansion sijainti, josta lisäkirjastot ladataan



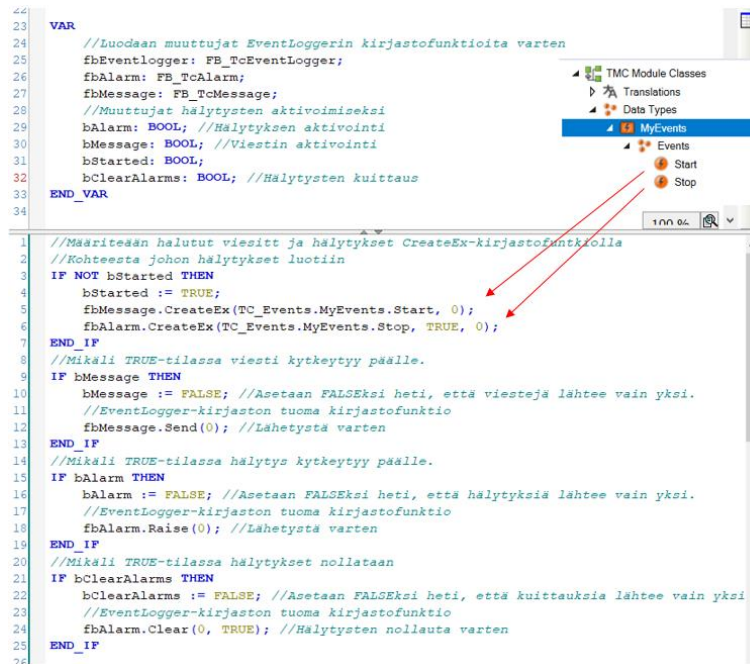
Hälytysten luominen tapahtuu PLC-puolella avaamalla Type System TwinCAT-projektin alta projektipuusta. Sieltä valitaan Event Classes. Tämän jälkeen napautetaan näkymää hiiren oikealla painikkeella ja valitaan New, jolloin aukeaa ikkuna, jossa hälytys tehdään. Hälytykselle voidaan määrittää haluttu viesti sekä vakavuusaste. Kuvassa 27. opastetaan hälytysten luominen.

Kuva 27. Hälytysten luominen



Hälytysten luomisen jälkeen niitä täytyy myös päästä käyttämään, joten tätä varten on tehtävä ohjelmakoodi, joka käyttää näitä luotuja hälytyksiä. Kuvassa 28. esitetään hälytysten demonstrointia varten tehty ohjelmakoodi PLC-projektin puolella.

Kuva 28. Esimerkki ST-kielisestä PLC-koodista, jolla hälytyksiä käytetään



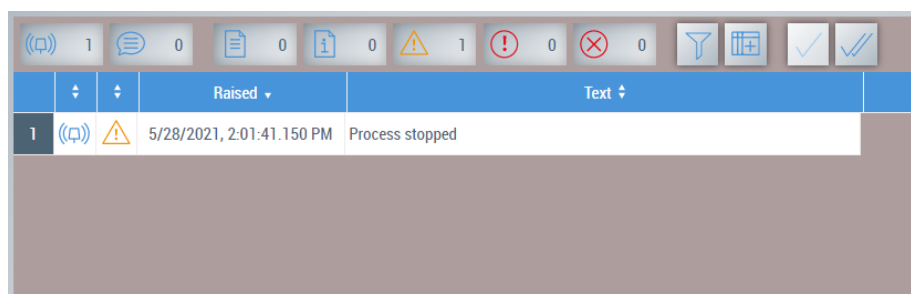
```

22 VAR
23 //Luodaan muuttujat EventLoggerin kirjastofunktioita varten
24 fbEventlogger: FB_TcEventLogger;
25 fbAlarm: FB_TcAlarm;
26 fbMessage: FB_TcMessage;
27 //Muuttujat hälytysten aktivoimiseksi
28 bAlarm: BOOL; //Hälytyksen aktivointi
29 bMessage: BOOL; //Viestin aktivointi
30 bStarted: BOOL;
31 bClearAlarms: BOOL; //Hälytysten kuittaus
32 END_VAR
33
34
35 //Määritetään halutut viestit ja hälytykset CreateEx-kirjastofunktiolla
36 //Kohteesta johon hälytykset luotiin
37 IF NOT bStarted THEN
38   bStarted := TRUE;
39   fbMessage.CreateEx(TC_Events.MyEvents.Start, 0);
40   fbAlarm.CreateEx(TC_Events.MyEvents.Stop, TRUE, 0);
41 END_IF
42 //Mikäli TRUE-tilassa viesti kytkeytyy päälle.
43 IF bMessage THEN
44   bMessage := FALSE; //Asetaan FALSEksi heti, että viestejä lähtee vain yksi.
45   //EventLogger-kirjaston tuoma kirjastofunktio
46   fbMessage.Send(0); //Lähetystä varten
47 END_IF
48 //Mikäli TRUE-tilassa hälytys kytkeytyy päälle.
49 IF bAlarm THEN
50   bAlarm := FALSE; //Asetaan FALSEksi heti, että hälytyksiä lähtee vain yksi.
51   //EventLogger-kirjaston tuoma kirjastofunktio
52   fbAlarm.Raise(0); //Lähetystä varten
53 END_IF
54 //Mikäli TRUE-tilassa hälytykset nollataan
55 IF bClearAlarms THEN
56   bClearAlarms := FALSE; //Asetaan FALSEksi heti, että kuittauksia lähtee vain yksi
57   //EventLogger-kirjaston tuoma kirjastofunktio
58   fbAlarm.Clear(0, TRUE); //Hälytysten nollauts varten
59 END_IF
60

```

HMI-puolella kirjasto ladataan projektipuusta oikealla hiirenpainikkeella HMI-Projektin kohdalta. Tämän jälkeen valitaan Manage NuGetPackages ja ladataan Beckhoff.TwinCAT.HMI.EventLogger. Kun paketti on ladattu, voidaan Toolboxista ottaa käyttöön EventGrid-elementti. Hälytykset sekä viestit tulevat näkyviin EventGridiin automaattisesti, kun hälytykset sekä ohjelmakoodi on luotu PLC-puolella (Kuva 29).

Kuva 29. EventGrid-elementti

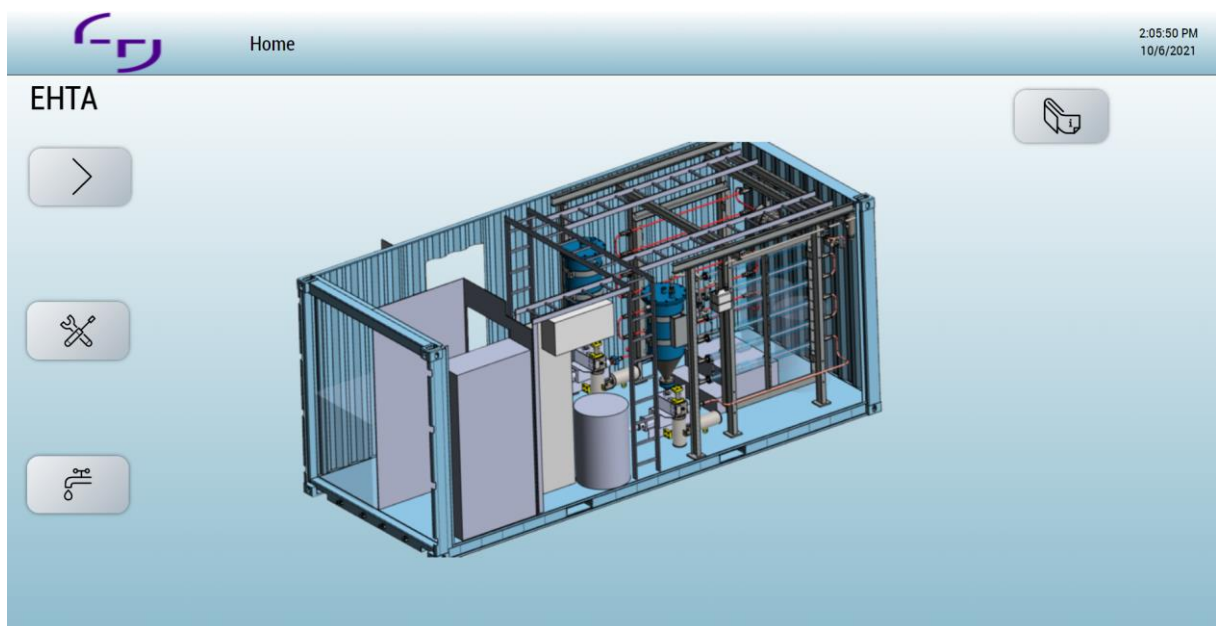


		Raised	Text
1	(□) ⚠	5/28/2021, 2:01:41.150 PM	Process stopped

## 6 Lopullinen käyttöliittymä

Koska kyseinen käyttöliittymä tulee olemaan vain koelaitteen käytössä eikä se ole kaupallinen tuote, oli minulla suhteellisen vapaat kädet sen suunnittelussa. Tärkeimpinä huomioitavina asioina oli kehittää käyttöliittymä, joka on selkeä ja jota on helppo operoida. Tähän pyrin pitämällä jokaisen käyttöliittymäsivun ulkoasultaan toisiaan vastaavina, pitämällä värimaailman hillittynä ja käyttämällä prosessin eri vaiheita esittäviä kuvia, joilla opastetaan käyttäjää. Käyttöliittymä suunniteltiin siis soveltaen ISA-101-standardin ohjeistuksia. Käyttöliittymä jakaantuu kolmeen ajotapaan, joita ovat pääajo, testiajo sekä huuhteluajo. Ajotapa valitaan HMI:n Desktop-sivulta eli koko käyttöliittymän ensimmäiseltä sivulta (Kuva 30). Kuvan vasemmassa reunassa näkyvät kolme painiketta ohjaavat eri ajotapoihin. Oikealla oleva painike on infopainike.

Kuva 30. Käyttöliittymän Desktop-sivu



### 6.1 Pääajo

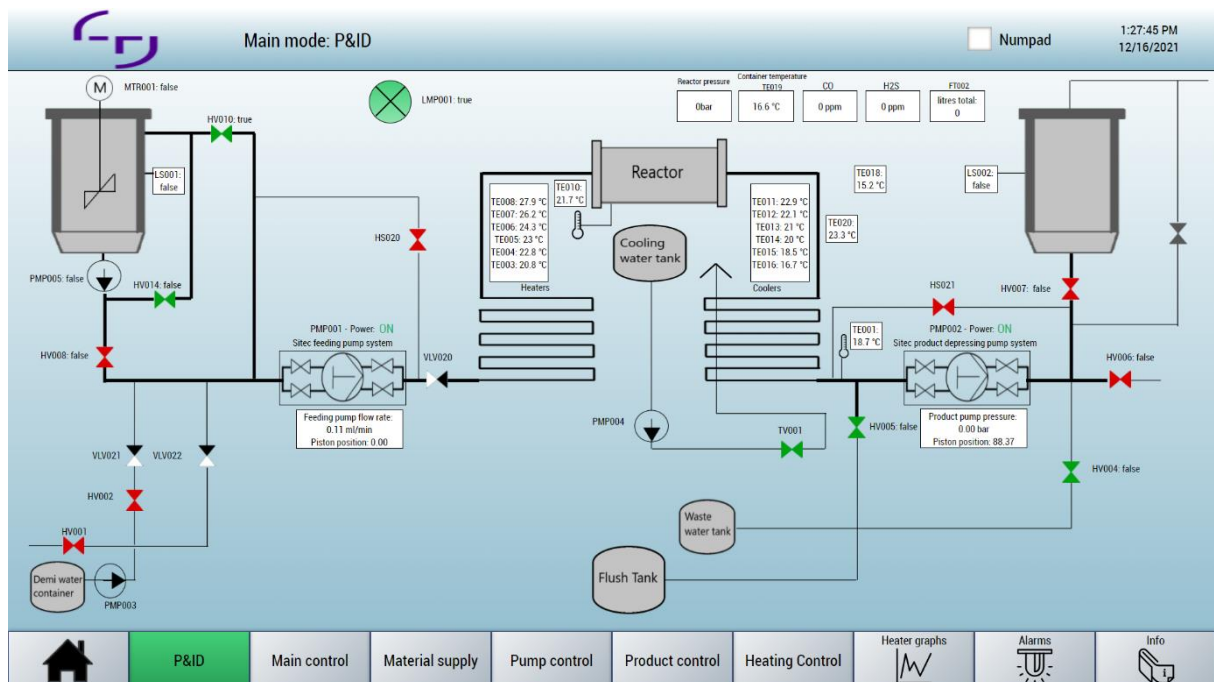
Pääajotapa valitaan kuvassa 30. olevasta nuolen näköisestä painikkeesta. Pääajotapa on nimensä mukaisesti se ajotapa, jota tullaan käyttämään, kun koelaitetta käytetään. Se pitää sisällään suurimman osan koko käyttöliittymästä, kuten yksinkertaistetun PI-kaaviosivun, jossa olevat instrumentit ovat piirretty erillisellä ohjelmalla ja tuotu PNG-kuvamuodossa projektiin. Putkistot ovat tehty Line-elementtiä käyttäen. PI-kaavio-sivu on tarkoitettu



helpottamaan käyttäjää seuraamaan koko prosessin tilaa helposti ymmärrettävästi yhdeltä sivulta. PI-kaavio sivulta nähdään reaaliajassa toimilaitteiden, kuten venttiilien, pumppujen ja moottoreiden tiloja. Lisäksi nähdään mm. erilaisia anturiarvoja. Ohjauksia kyseiseltä sivulta ei voida kuitenkaan tehdä.

Ohjauksia varten pääajossa on viisi erillistä sivua, joista ensimmäinen on prosessin käynnistystä varten ja loput neljä prosessin tärkeimpien osien ohjaukseen, joita ovat materiaalin syöttö, pumppujen ohjaus, tuotteen ohjaus ja lämmittimien ohjaus. Pääajo pitää vielä sisällään lämmittimien trendisivun, josta nähdään eri lämmittimien muuttuja-arvoista luotuja trendejä käyttäjän säätämältä aikaväliltä. Myös hälytyssivuun eli Alarms-sivuun pääsee vain pääajotavan kautta. Kaikki hälytykset ja virheilmoitukset ilmaantuvat tälle sivulle. Kuvassa 31. näemme pääajon ensimmäisen sivun eli PI-kaaviosivun. Muut pääajon sivut ovat liitteenä.

Kuva 31. Pääajotavan ensimmäinen sivu



## 6.2 Testi- ja huuhteluajo

Testiajo, johon päästään painamalla kuvan 30. työkaluja esittävää painiketta on ajotapa, jolla haluttiin vain ja ainoastaan testata koelaitteen I/O-yhteyksien ja toimilaitteiden toimintaa yksinkertaisesti yhdeltä käyttöliittymäsivulta. Kyseistä sivua ei siis normaalitilanteessa käytetä lainkaan, joten sen ulkoasuun ei ole panostettu.

Huuhteluajo, johon päästään painamalla kuvan 30. vesihanaa esittävää painiketta on ajotapa, jota käytetään koelaitteen putkistojen huuhteluun. Putkistot huuhdellaan joko vedellä tai typellä. Huuhteluajo pitää sisällään vain yhden käyttöliittymäsivun (Kuva 32). Kyseinen sivu pitää sisällään painonappeja tarvittavien venttiilien sekä pumppujen ohjaukseen, että prosessin putkistot saadaan huuhdeltua.

Kuva 32. Huuhteluajon käyttöliittymäsivu

The screenshot displays the 'Flushing' control interface. At the top left is a logo, and at the top right, the time '3:28:22 PM' and date '11/16/2021' are shown. The interface is divided into three main sections by dashed lines: 'N2 Flush', 'Water Flush', and 'Sitec pumps'. On the far left, under 'Bypass Valves', there are two buttons: 'Closed' (red) and 'ByPass1' (grey). In the 'N2 Flush' section, there is a 'StartN2Flush' button, an 'N2 Flushing OFF' indicator (red), and a status box for 'HV001: false' with 'Currently Closed' and 'Normally Closed' indicators. The 'Water Flush' section contains a 'StartWaterFlush' button, a 'Water flush OFF' indicator (red), a status box for 'HV002: false' with 'Currently Closed' and 'Normally Closed' indicators, and a status box for 'PMP003: false' with an 'Offline' indicator. The 'Sitec pumps' section features a 'Start PMP001 and PMP002' button and a data table for pump parameters.

	Volumetric flow SP (ml/min)	Feeding pump flow rate (ml/min)
PMP01 Power OFF	0	0
PMP02 Power OFF	0	0
	Pressure SP (bar)	Product pump pressure (bar)
	0	0

## 7 Yhteenveto

Voi todeta, että opinnäytetyö oli kokonaisuudessaan onnistunut työ, sillä käyttöliittymä saatiin luotua täysin tyhjästä, ja se saatiin lopulta myös toimimaan halutulla tavalla. Näin ollen opinnäytetyön päätavoitteeseen päästiin. Lisäksi muita tavoitteita, kuten tietoa, ohjeita ja vinkkejä Beckhoff HMI -ohjelmiston käytöstä saatiin sisällytettyä opinnäytetyöhön kohtuullisen monipuolisesti. Opinnäytetyössä saatiin myös selvitettyä tehokkaan käyttöliittymän suunnitteluun liittyviä toimintaohjeita ISA101-standardin pohjalta.

Työn laajuutta karsittiin vähän. Alkuperäinen ajatus kattoi käyttöliittymän lisäksi myös SQL-tietokannan, joka toteutettiin vastaavasti Beckhoffin tuotteilla. Tämä kuitenkin jätettiin opinnäytetyöstä pois, että opinnäytetyön aihepiiri pysyisi selkeänä eikä työn laajuus kasvaisi liialliseksi.

Opinnäytetyön aihe oli mielenkiintoinen ja opettavainen. Tämä Beckhoffin minulle entuudestaan tuntematon HMI-ohjelmisto osoittautui hyvin monipuoliseksi ja tehokkaaksi työkaluksi. Olen myös tyytyväinen, että pääsin käyttämään JavaScript-ohjelmointikieltä, sillä aiempaa ohjelmointikokemusta minulla ei kyseisellä kielellä ollut. JavaScript oli hyödyllinen työkalu interaktiivisten toimintojen luomiseksi käyttöliittymään.

Opinnäytetyössä olevat kuvankaappaukset eivät ole välttämättä käyttöliittymän aivan lopullisesta ulkoasusta, sillä käyttöliittymään tehtiin pieniä muutoksia koelaitteen käyttöönoton yhteydessä esiin tulleiden kehitysehdotusten mukaisesti. Näitä kaikkia muutoksia ei ehditty päivittämään opinnäytetyöhön aikataulullisista syistä johtuen.

## Lähteet

6Aika. (2020). ILPO – Ilmastoposiitiiviset yritysalueet ja arvoketjut. Haettu 24.11.2021 osoitteesta <https://6aika.fi/project/ilpo-ilmastoposiitiiviset-yritysalueet-ja-arvoketjut/>

ANSI/ISA-101.01-2015. (2015). Human Machine Interfaces for Process Automation Systems. ISA.

Beckhoff. (n.d. -a) Licensing. Haettu 28.9.2021 osoitteesta [https://infosys.beckhoff.com/english.php?content=../content/1033/tf6340\\_tc3\\_serial\\_communication/63050395045796619.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tf6340_tc3_serial_communication/63050395045796619.html&id=)

Beckhoff. (n.d. -b) List of Abbreviations. Haettu 20.9.2021 osoitteesta <https://infosys.beckhoff.com/english.php?content=../content/1033/ek9300/2609008523.html&id=3352755806519760039>

Beckhoff. (n.d. -c) Manual TE2000 TC3 HMI Engineering. [Manual TE2000 TC3 HMI Engineering \(filfar.by\)](#)

Beckhoff. (n.d. -d) Manual TF2000 TC3 HMI Server [https://download.beckhoff.com/download/document/automation/twincat3/TF2000\\_TC3\\_HMI\\_Server\\_EN.pdf](https://download.beckhoff.com/download/document/automation/twincat3/TF2000_TC3_HMI_Server_EN.pdf)

Beckhoff. (n.d. -e) Manual TwinCAT 3 Product Overview. [https://download.beckhoff.com/download/document/automation/twincat3/Product\\_overview\\_EN.pdf](https://download.beckhoff.com/download/document/automation/twincat3/Product_overview_EN.pdf)

Beckhoff. (n.d. -f) TwinCAT 3 ADS. Haettu 28.9.2021 osoitteesta <https://www.beckhoff.com/fi-fi/products/automation/twincat/tc1xxx-twincat-3-base/tc1000.html>

Beckhoff. (n.d. -g) TwinCAT PLC Control. Haettu 28.9.2021 osoitteesta [https://infosys.beckhoff.com/english.php?content=../content/1033/tcquickstart/html/tcquickstart\\_plccontrol1.htm&id=593292738783223705](https://infosys.beckhoff.com/english.php?content=../content/1033/tcquickstart/html/tcquickstart_plccontrol1.htm&id=593292738783223705)

Beckhoff. (n.d. -h) TwinCAT Module Class Editor (TMC). Haettu 19.12.2021 osoitteesta [https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_c/721428747.html&id=8165600958249830669](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_c/721428747.html&id=8165600958249830669)

Beckhoff. (n.d. -i) Offline symbols (TMC). Haettu 19.12.2021 osoitteesta [https://infosys.beckhoff.com/english.php?content=../content/1033/te2000\\_tc3\\_hmi\\_engine\\_ering/8775310603.html&id=4414587988328531597](https://infosys.beckhoff.com/english.php?content=../content/1033/te2000_tc3_hmi_engine_ering/8775310603.html&id=4414587988328531597)

BL2F. (2020) Black Liquor to fuel. Haettu 24.11.2021 osoitteesta <https://www.bl2f.eu/>

DZone. (2018) Why you should use TypeScript for developing web applications. Haettu 14.9.2021 osoitteesta <https://dzone.com/articles/what-is-typescript-and-why-use-it>

Inductive Automation. (2018). What is HMI? Haettu 31.8.2021 osoitteesta <https://www.inductiveautomation.com/resources/article/what-is-hmi>

Gollakota, A.R.K., Kishore, N., Gu, S. A review on hydrothermal liquefaction of biomass, *Renewable & sustainable energy reviews*, (81), 2018, ss. 1378-1392. <https://www.sciencedirect.com/science/article/pii/S1364032117308146>

ISA. (2020). ISA-101.01, Human Machine Interfaces for Process Automation Systems. Haettu 3.9.2021 osoitteesta <https://www.isa.org/intech/2020/september-october/isa-101-01-human-machine-interfaces-for-process-au>

ISA. (n.d.). Design Tips to Create a More Effective HMI. Haettu 3.9.2021 osoitteesta <https://blog.isa.org/design-tips-effective-industrial-machine-process-automation-hmi>

jQuery user interface. (n.d.) jQuery UI Demos. Haettu 14.9.2021 osoitteesta <https://jqueryui.com/demos/>

Kumar, M., Oyedun, A., Kumar, A. A review on the current status of various hydrothermal technologies on biomass feedstock, *Renewable & sustainable energy reviews*, (81), 2018, ss. 1748-1770. <https://www.sciencedirect.com/science/article/pii/S1364032117309164>

Mueller, O. (3.4.2020). TwinCAT 3 HMI Tutorial JavaScript in UserControl [Video]. YouTube.  
[https://www.youtube.com/watch?v=IPGPiX86L0c&ab\\_channel=OlegMueller](https://www.youtube.com/watch?v=IPGPiX86L0c&ab_channel=OlegMueller)

RealPars. (2018). What is an HMI? Learn Some of the Basics of an HMI or Human Machine Interface. Haettu 31.8.2021 osoitteesta <https://realpars.com/what-is-hmi/>

Realpars. (2020). What is High-Performance HMI? Let's move away from poorly designed HMIs and learn about the High-Performance HMI. Haettu 3.9.2021 osoitteesta <https://realpars.com/high-performance-hmi/>

Realpars. (2021). Detailed Design Principles of High-Performance HMI Displays. Let's talk about the detailed design of a High-Performance HMI. Haettu 3.9.2021 osoitteesta <https://realpars.com/HMI-Display/>


Realpars. (2021). Development of High-Performance HMI Philosophy. The basic principles of High-Performance HMI and the Development of High-Performance HMI. Haettu 3.9.2021 osoitteesta <https://realpars.com/hmi-philosophy/>

Saarinen, E. (2020) *Risk Assessment of Hydrothermal Liquefaction Test Equipment Using Failure Mode and Effects Analysis* [Kandidaatin tutkielma, Tampereen yliopisto].  
<https://trepo.tuni.fi/bitstream/handle/10024/124594/SaarinenElias.pdf?sequence=2&isAllowed=y>

Toor, S., Rosendahl, L., Rudolf, A. Hydrothermal liquefaction of biomass: A review of subcritical water technologies, *Energy*, 36(5), 2011, ss. 2328-2342.  
<https://www.sciencedirect.com/science/article/pii/S0360544211001691>

w3schools. (n.d.) jQuery Introduction. Haettu 14.9.2021 osoitteesta [https://www.w3schools.com/jquery/jquery\\_intro.asp](https://www.w3schools.com/jquery/jquery_intro.asp)

### Liite 1: Pääajotavan käyttöliittymäsivut

 Main mode: Main Control 3:23:40 PM 11/16/2021

Numpad

---

Main Control

Controls

Indicators

Start process Online

Enable Test Offline

Start Flushing Offline

Enable

Main Power

Lamp 1 : On

UPSrunEnable  
UPS Time:  
PTOS

Fault current protection alarm

Motor protection fault

STOP

Material supply

Pump control

Product control

Heating Control


Heater graphs


Alarms

Info

P&ID

Main control



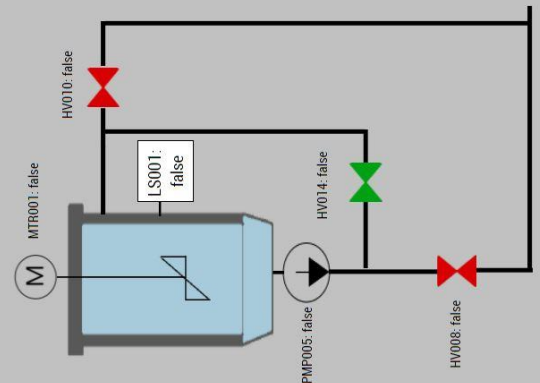
Main mode: Material Supply

3:24:10 PM11/16/2021

Numpad

---

### Material Supply



MTR001: false  
LS001: false  
PMP005: false  
HV010: false  
HV008: false  
HV014: false

### Controls

#### Valves

HV008: false  
Currently Closed  
Normally Closed

HV010: false  
Currently Closed  
Normally Closed

HV014: false  
Currently Open  
Normally Open

#### Pumps and motors

PMP005: false  
Offline

MTR001: false  
Offline


Frequency Converter  
Input (Hz)  

0

Max: 50  
Min: 0  
Previous:

#### Sensors

Level Indicator  
LS001: false



P&ID

Main control

Material supply

Pump control

Product control

Heating Control

Heater graphs

Alarms

Info



Main mode: Material Supply

3:24:10 PM  
11/16/2021

Numpad

Material Supply

Controls

Valves

Pumps and motors

Sensors



P&ID

Main control

Material supply

Pump control

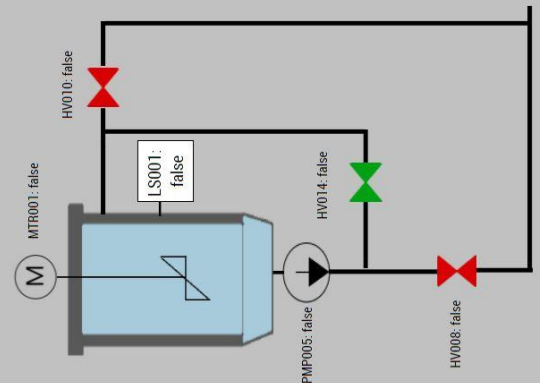
Product control

Heating Control

Heater graphs

Alarms

Info



HV010: false

LS001: false

MTR001: false

PMP005: false

HV014: false

HV008: false

HV008: false

HV010: false

HV014: false

Currently Closed  
Normally Closed

Currently Closed  
Normally Closed

Currently Open  
Normally Open

PMP005: false

Offline

MTR001: false

Offline

Frequency Converter  
Input (Hz)

0

Max: 50  
Min: 0  
Previous:

Level Indicator

LS001: false



P&ID

Main control

Material supply

Pump control

Product control

Heating Control

Heater graphs

Alarms

Info





# Main mode: Pump Control

3:25:22 PM  
11/16/2021

Numpad

## Pump Control

### Startup

Start PMP001 and PMP002

**PMP01 Power off**

**PMP02 Power off**

HML\_baUnitsAutoMan

### Sitec Material pump

Volumetric Flow (ml/min)	200
Max: 220	Min: 0
Previous: 200	

Material pump flow rate (ml/min)	0
Piston position	0

### Sitec Product pump

Pressure (bar)	125
Max: 300	Min: 0
Previous: 125	

Product pump pressure (bar)	0
Piston position	0



P&ID

Main control

Material supply

**Pump control**

Product control

Heating Control

Heater graphs

Alarms

Info

**Main mode: Product Control**

1:29:59 PM  
12/16/2021

Numpad

---

**Product control**

**Controls**

**Valves**

HV004: false

Currently: Open  
Normally: Open

HV005: false

Currently: Active  
Normally: Open

HV006: false

Currently: Closed  
Normally: Closed

HV007: false

Currently: Closed  
Normally: Closed

Pressure high limit: 315

Pressure low limit: 10

Manual pulse time (ms): P70.5S

Auto pulse On time (ms): P70.5S

Auto pulse Off time (ms): P70.5S

**Sensors**

CO: 0.00

H2S: 0.00

Level Indicator: LS002: false

Airflow: litres total: 0

Start

Reset

litres per minute: 0

litres per hour: 0

Material supply: ON

P&ID

Main control

Material supply

Pump control

Product control

Heating Control

Heater graphs

Alarms

Info



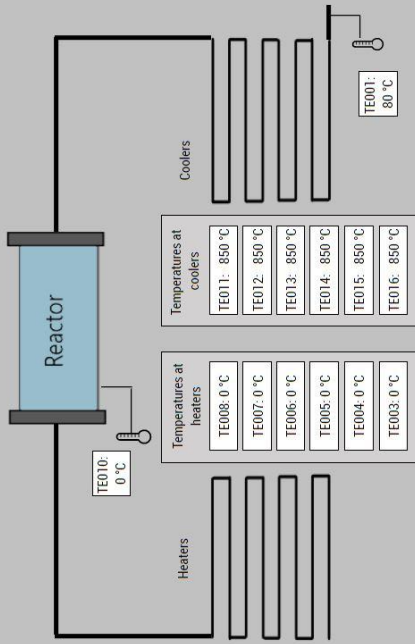
Main mode: Heating Control

3:27:07 PM  
11/16/2021

Numpad

Heaters

<p>TC001</p> <p>SP: 70, PV: 80, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC003</p> <p>SP: 60, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC004</p> <p>SP: 70, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC005</p> <p>SP: 80, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>
<p>TC006</p> <p>SP: 90, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC007</p> <p>SP: 100, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC008</p> <p>SP: 110, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>	<p>TC010 - Reactor</p> <p>SP: 130, PV: 0, CV: 0</p> <p>Auto mode / Manual mode</p>



<p>TC001 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC003 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC004 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC005 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC006 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC007 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC008 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>	<p>TC010 - Manual SP</p> <p>Min: 0, Max: 100</p> <p>0</p>
---	---	---	---	---	---	---	---

TC001 - Parameters

- HmiafTC01gHeat: 0
- HmiafTC01nHeat: PTOS
- HmiafTC01wHeat: PTOS
- HmiafTC01dHeat: PTOS
- HmiafErrorsTC: false
- HmiafErrorsTC: 0
- HmiafAlarmsTC: 0



# Main mode: Heater graphs

1:31:28 PM  
12/16/2021

Numpad

Heating graphs 1/2

Next Page

To zoom specific line graph, select it by clicking on it



- 1
- 2
- 3
- 4

Hide/Reveal MenuBar

Bring selected graph front

Lock graph position

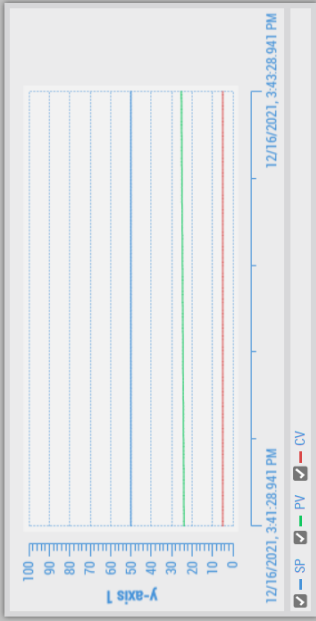
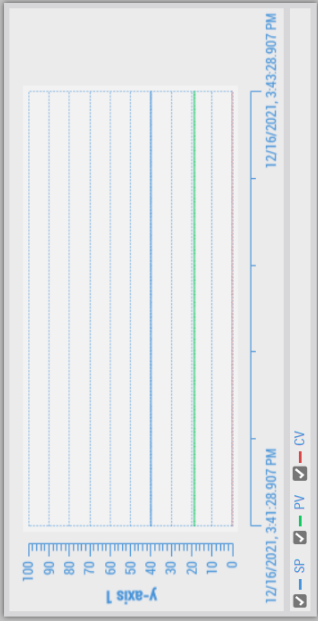
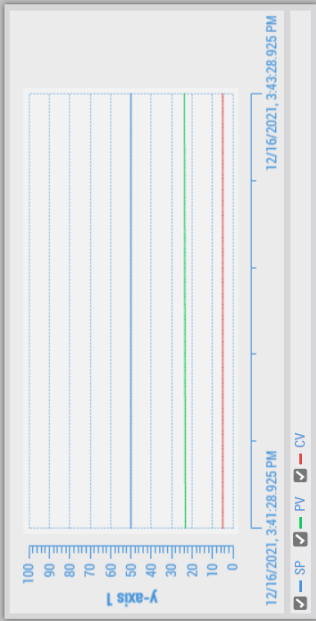
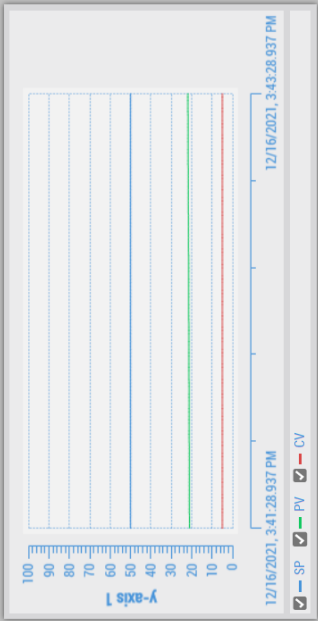
Heater graph order

TC001

TC004

TC003

TC005



P&ID

Main control

Material supply

Pump control

Product control

Heating Control

Heater graphs

Alarms

Info



# Main mode: Alarms

9:40:45 AM  
10/28/2021

Numpad

All Events

Active Events

Clear

(ID)	210	790	0	0	0	0	1000	0	Text
		Raised							
1	(C)	10/11/2021, 1:58:45.758 PM	TC002 Cooling water Lo alarm						
2	(C)	10/11/2021, 1:58:45.758 PM	Pump unit 2 power off alarm						
3	(C)	10/11/2021, 1:58:45.758 PM	Pump unit 1 power off alarm						
4	(C)	10/11/2021, 1:58:45.758 PM	Emergency stop alarm						
5	(C)	10/11/2021, 1:58:45.758 PM	K21 400vac power on (Off or Blackout: '0')						
6	(C)	10/11/2021, 1:58:45.758 PM	2F1-4 Motor protection switch alarm						
7	(C)	10/11/2021, 1:58:45.758 PM	1F1-13 Fault current protection alarm						
8	(C)	10/11/2021, 1:51:07.398 PM	TC002 Cooling water Lo alarm						
9	(C)	10/11/2021, 1:51:07.398 PM	Pump unit 2 power off alarm						
10	(C)	10/11/2021, 1:51:07.398 PM	Pump unit 1 power off alarm						
11	(C)	10/11/2021, 1:51:07.398 PM	Emergency stop alarm						
12	(C)	10/11/2021, 1:51:07.398 PM	K21 400vac power on (Off or Blackout: '0')						
13	(C)	10/11/2021, 1:51:07.398 PM	2F1-4 Motor protection switch alarm						
14	(C)	10/11/2021, 1:51:07.398 PM	1F1-13 Fault current protection alarm						
15	(C)	10/11/2021, 1:51:06.438 PM	TC002 Cooling water Lo alarm						
16	(C)	10/11/2021, 1:51:06.438 PM	Pump unit 2 power off alarm						



P&ID

Main control

Material supply

Pump control

Product control

Heating Control

Heater graphs

Alarms

Info