
ASE-6030 Automaation reaaliaikajärjestelmät
Suunnitteludokumentti
Minipanosprosessi
Versio 3.0

Konsta Pirttilahti, 253080
Pyry Vehmas, 242160
Dokumentin tila: palautus versio Muokattu: 3.12.2019

SISÄLLYSLUETTELO

Sisällysluettelo.....	2
1 Johdanto	3
2 Vaatimusmäärittely	4
2.1 Vaatimukset	4
2.2 Käyttötapaukset	4
2.2.1 KT 1: Yhteyden muodostaminen	5
2.2.2 KT 2: Sovelluksen sulkeminen.....	6
2.2.3 KT 3: Prosessiparametrien asettaminen	6
2.2.4 KT 4: Prosessin käynnistys	6
2.2.5 KT 5: Prosessin keskeytys.....	6
2.3 Käyttöliittymähahmotelmat	7
3 Suunnittelu	8
3.1 Sovellusarkkitehtuuri	8
3.2 Rakenne	8
3.3 Toiminta ja tilat.....	9
3.3.1 Tilakaavio	10
3.3.2 Sekvenssikaaviot	11
4 Toteutus.....	14
4.1 Kehitysympäristö	14

1**JOHDANTO**

Tämä dokumentti on tehty kurssin ASE-6030 Automaation reaaliaikajärjestelmät kurssille harjoitustyönä. Dokumentissa suunnittelemme kurssin henkilökunnan luomalle minipanosprosessin simulaatiolle käyttöliittymän, jonka avulla prosessia pystytään ohjaamaan käyttäjän haluamilla ohjausarvoilla.

2**VAATIMUSMÄÄRITTELY**

Tässä osiossa käydään läpi mitä vaatimuksia sovellukselle on asetettu, että se soveltuu käytettäväksi, sekä mitä käyttötapauksia sovellusta operoidessa tulee vastaan ja miten sovelluksen tulisi kunkin tapauksen kohdalla suoriutua.

2.1**Vaatimukset**

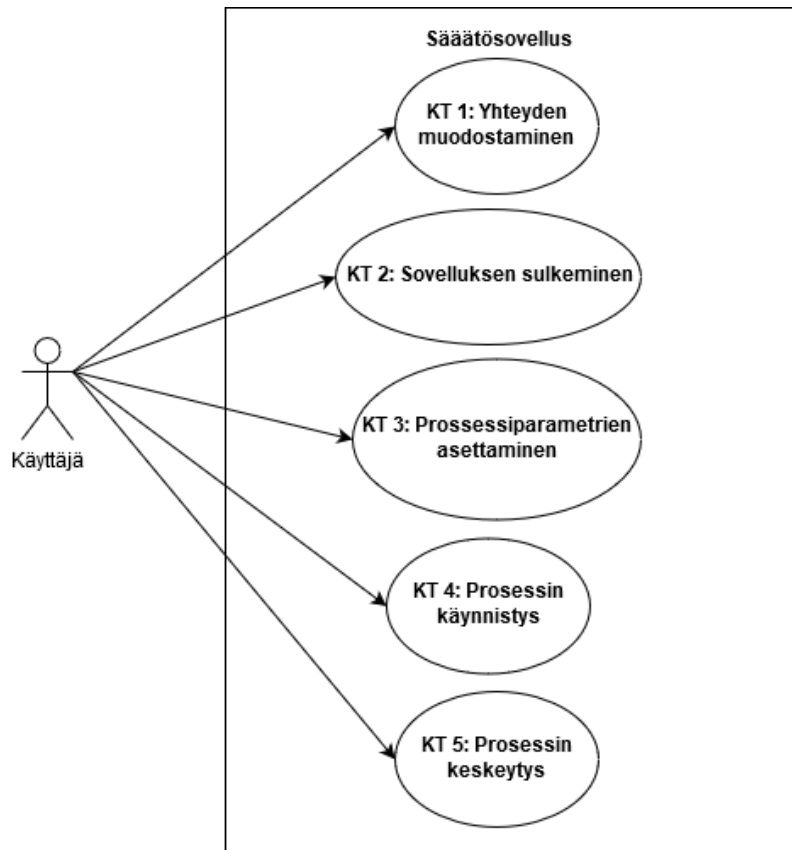
Vaatimusmäärittelyyn on käytetty seuraavia kurssihenkilöiden antamia dokumentteja: asiakasvaatimukset, mpp_asiakasvaatimukset_1_1.pdf, prosessin sekvenssikaaviota, mpp_pfc.pdf, sekä simulaation ja sovelluksen välisen rajapinnan, MppOpcUaClientLib_dll, Doxygen html dokumentti.

Sovelluksen tulee muodostaa yhteys simulaattorin kanssa onnistuneesti, tallentaa käyttäjän määrittämät prosessiparametrit ja käynnistää sekvenssi kyseisillä parametreilla. Haluttaessa sekvenssi tulee pysähtyä ja asettaa simulaattorin prosessin toimilaitteet alkutiloihinsa.

Sovellus ei saa kaatua, vaikkei yhteyttä saataisikaan muodostettua tai jos jokin muu poikkeustilanne ilmenee kesken ajon.

2.2**Käyttötapaukset**

Sovelluksen käyttöön liittyy muutama alapuolella olevan kaavion mukainen käyttötapaus.



Kuva 1. Käyttötapauskaavio.

Alta löytyy käyttötapauskaaviossa kuvatut käyttötapaukset. Nämä avaavat enemmän, mitä käyttötapauksella tulee olla esiehtona, kyseisen tapauksen aktorin, tarkemman kuvauksen, mahdolliset poikkeustilanteet, sekä käyttötapauksen suorituksen jälkeisen lopputuleman.

2.2.1

KT 1: Yhteyden muodostaminen

Suorittajat: Käyttäjä

Esiehdot: Käyttäjä on avannut sovelluksen, sekä simulaattori on käynnistetty.

Kuvaus: Sovelluksen käyttöliittymässä on painike ”Connect”, käyttäjän painettua painiketta sovellus yrittää yhteyden muodostamista simulaattorin kanssa. Sovellus ilmoittaa käyttäjälle, missä tilassa yhteyden muodostus on menossa: ”Connecting”, jos yhteyttä vielä muodostetaan. ”Connected”, jos yhteys saadaan muodostettua. ”Disconnected”, jos yhteyden muodostaminen epäonnistuu tai yhteys katkeaa.

Poikkeukset: Erilaiset bugit sekä sovellus ohjelmassa tai simulaattori ohjelmistossa, voivat aiheuttaa häiriötä yhteyden muodostamisessa. Tällöin käyttöliittymä ilmoittaa, että yhteys on katkennut tai sitä yritetään muodostaa. Käyttäjä voi yrittää yhteyden ottamista uudelleen.

Lopputulos: Sovelluksen ja simulaattorin välillä on toimiva yhteys. Simulaattori pystyy ottamaan käskyjä vastaan käyttöliittymän kautta.

2.2.2

KT 2: Sovelluksen sulkeminen

Suorittajat: Käyttäjä

Esiehdot: Käyttäjä on avannut sovelluksen.

Kuvaus: Käyttäjä voi sulkea sovelluksen joko painamalla käyttöliittymässä olevaa raksia tai "Close" painiketta. Mikäli sekvenssin ajo on käynnissä, tämän suoritus keskeytetään ja yhteys simulaattoriin katkaistaan.

Lopputulos: Sovellus on suljettu ja prosessi on pysähtänyt.

2.2.3

KT 3: Prosessiparametrien asettaminen

Suorittajat: Käyttäjä

Esiehdot: Sovellus on käynnissä.

Kuvaus: Käyttöliittymässä on textbox tyyppinen käyttöliittymäkomponentti jokaiselle prosessiparametrille. Näihin käyttäjä voi syöttää haluamansa numero arvot. Syötettyään prosessi parametrit käyttäjä painaa käyttöliittymässä painiketta "Set Values", jolloin arvot tallentuvat ohjelmaan.

Lopputulos: Ohjelmaan tallentuvat sille kelpaavat prosessiparametrien arvot.

2.2.4

KT 4: Prosessin käynnistys

Suorittajat: Käyttäjä

Esiehdot: Sovellus on käynnissä, sovelluksella ja simulaattorilla on yhteys, sekä käyttäjä on syöttänyt prosessiparametrit onnistuneesti.

Kuvaus: Käyttäjä painaa käyttöliittymässä painiketta "Start", joka käynnistää prosessin suorituksen valituilla prosessiparametreilla.

Lopputulos: Sovellus suorittaa sekvenssin ajon.

2.2.5

KT 5: Prosessin keskeytys

Suorittajat: Käyttäjä

Esiehdot: Sovellus suorittaa prosessin ajoa.

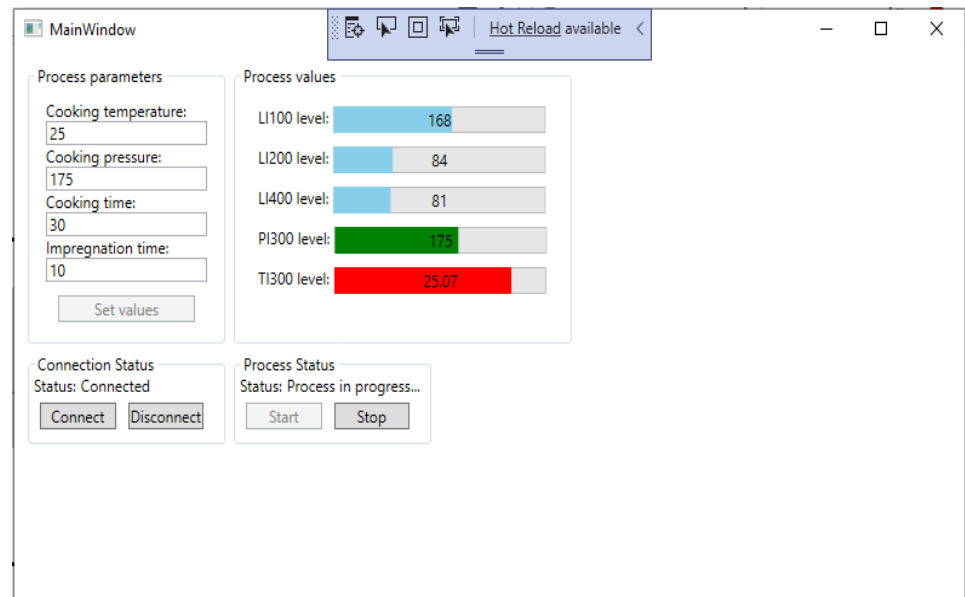
Kuvaus: Käyttäjä painaa käyttöliittymässä painiketta "Stop".

Lopputulos: Prosessin suoritus keskeytyy, sekä tällöin sovellus ja simulaattorissa olevat toimilaitteet palaavat alkutilaansa säiliöiden nestemääriä lukuun ottamatta.

2.3

Käyttöliittymähahmotelmat

Alla olevassa kuvassa on hahmotelma suunnitellusta käyttöliittymästä.



3

SUUNNITTELU

Tässä luvussa dokumentoidaan varsinaista sovellussuunnittelua.

3.1

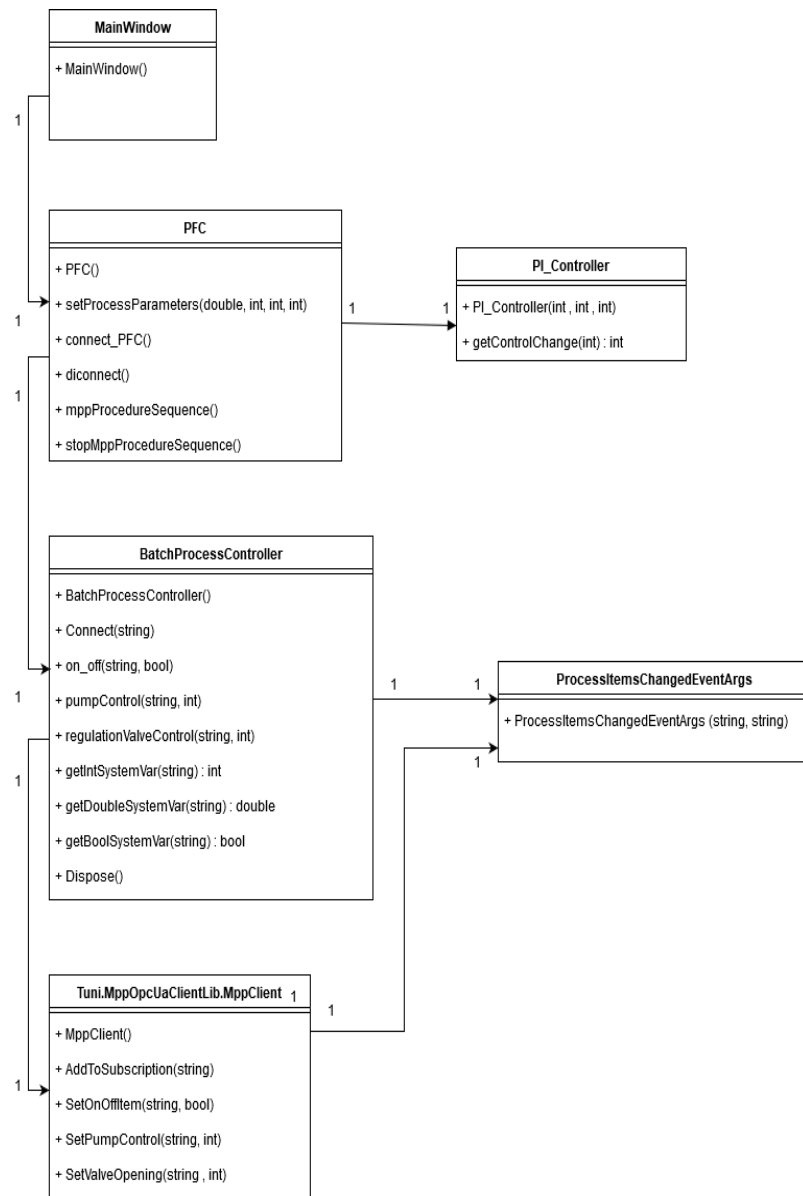
Sovellusarkkitehtuuri

Sovelluksessa on käytetty Model-Control-View -mallia. View muodostuu säiliöiden arvojen sekä numeraalisesta, että graafisesta näyttämisestä. Controllerina toimii MainWindow.cs luokka ja modelina toimii käyttöliittymän takana oleva muu koodi, joka suorittaa prosessia.

3.2

Rakenne

Sovelluksen rakennetta kuvaava luokkakaavio. Luokat sekä niiden tehtävät ovat selitetty tässä aliluvussa.



Kuva 3. Luokkakaavio.

MainWindow

MainWindow luokka toteuttaa käyttöliittymän toiminnot käyttäen PFC luokasta luotua oliota.

PFC

PFC luokassa toteutetaan prosessiparametrien tallennus, prosessin sekvenssi, sekä tähän tarvittavat alisekvenssit. Muita tämän luokan tehtäviä ovat connect_PFC ja disconnect metodien toteutus.

BatchProcessController

BatchProcessController luokassa toteutetaan yhteyden ottaminen prosessilaitteistoon TunimppOpcUaClientLib rajapinna avulla, asetetaan tarvittavat asetusarvot, sekä toteutetaan metodit yksittäisten toimilaitteiden ohjaamiselle.

PI_Controller

PI_Controller luokka suorittaa säädön määritetyillä säätöparametrin arvoilla. Luokalle annetaan säädön maksimi ja minimi arvot, sekä haluttu arvo prosessisuurelle ja kutsuttaessa luokka palauttaa säädön, jolla päästään lähemmäksi haluttua prosessisuureen arvoa.

ProcessItemsChangedEventArgs

ProcessItemsChangedEventArgs luokka toimii Event tilanteessa räätälöitynä EventArgs luokkana, joka lähettää muuttuneen prosessi laitteiston arvon, sekä tunnisteen.

TuniMppOpcUaClientLib.MppClient

MppClient on TunimppOpcUaClientLib kirjaston valmis luokka, joka on saatu käyttöön kurssi henkilökunnalta.

3.3

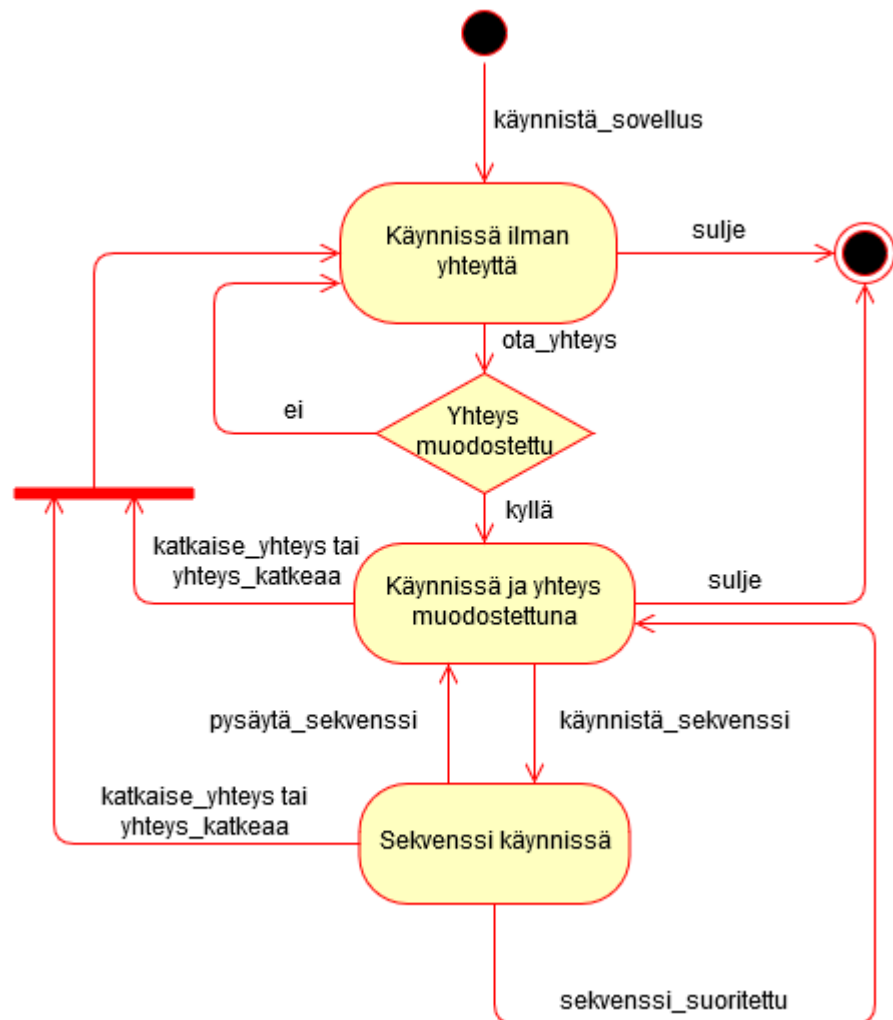
Toiminta ja tilat

Tässä aliluvussa on kuvattu järjestelmän mahdolliset toiminnot, tilat, sekvenssi-, sekä tilakaaviot. Kyseiset kaaviot on avattu enemmän tekstissä alapuolella.

3.3.1

Tilakaavio

Kaaviossa on määritelty mahdolliset sovelluksen tilat, joita voi tulla vastaan. Sekä mihin tiloihin näistä tiloista pystytään siirtymään, joko hallitusti tai virhetilanteen kautta.



Kuva 3. Tilakaavio

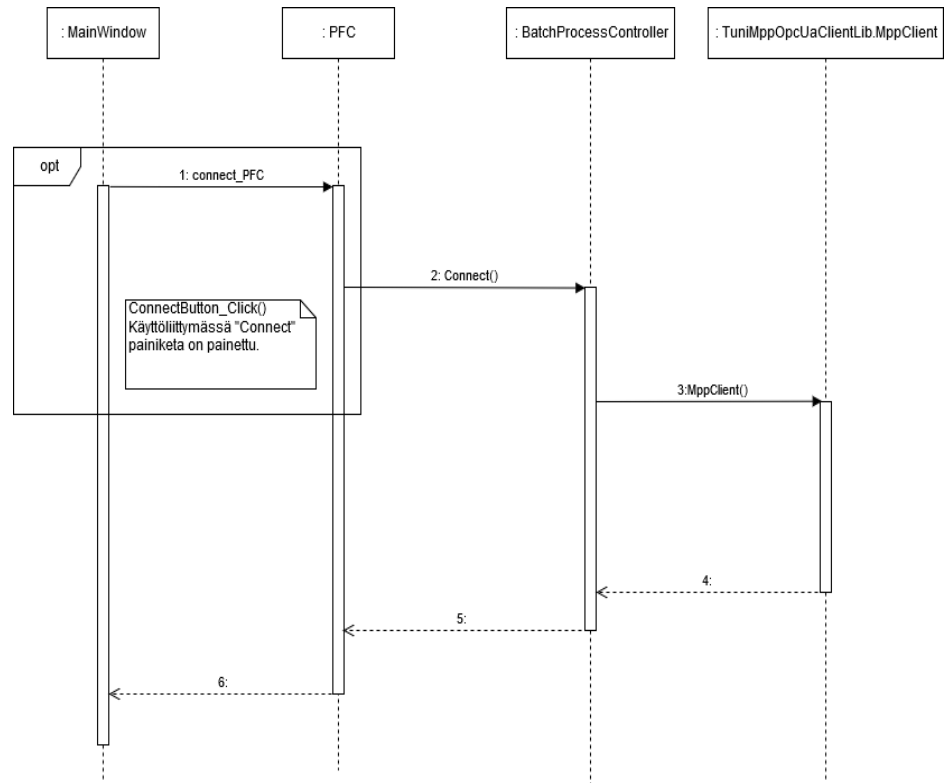
Sovelluksella on kolme tilaa. Ensimmäinen tila on sovellus käynnissä, mutta yhteyttä simulaattoriin ei ole vielä muodostettu. Yhteyden ottamisen yhteydessä oleva ”salmiakki” kuvio kuvaa ehtoa sille, miten yhteyden muodostaminen onnistui.

Seuraava ja toinen tila on sovellus, jolla on yhteys simulaattoriin. Tästä tilasta onnistuu siirtyminen sovelluksen kolmanteen tilaan eli, sekvenssin ajotilaan. Huomattavaa kaaviosta on se, että sovelluksen voi suoraan sulkea ”Close” painikkeella ainoastaan kahdesta aiemmasta tilasta. Mikäli sovellus suljetaan ajotilassa, sekvenssin ajo suljetaan hallitusti ennen itse sovelluksen sulkemista.

3.3.2

Sekvenssikaaviot

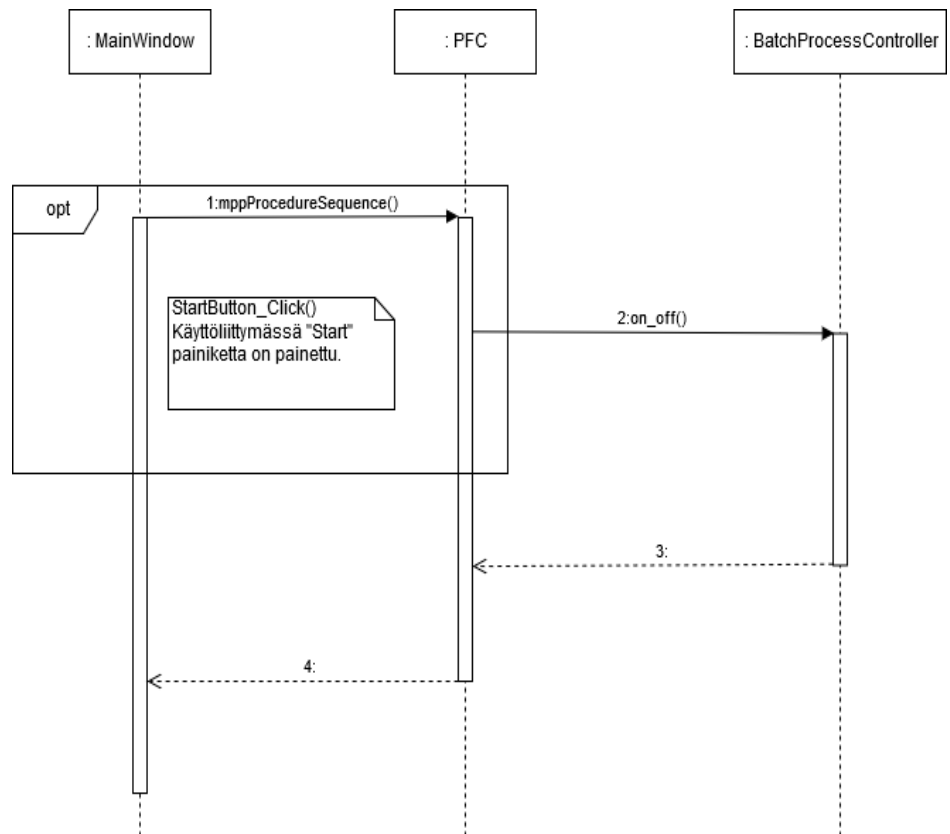
Tämä aliluku sisältää sekvenssikaaviot seuraavista tapahtumista: yhteyden avaaminen simulaattoriin, sekvenssin käynnistäminen, sekä mittaus arvojen päivittyminen simulaattorista käyttöliittymään.



Kuva 5. Yhteyden avaaminen simulaattoriin

Yhteyden avaaminen simulaattoriin

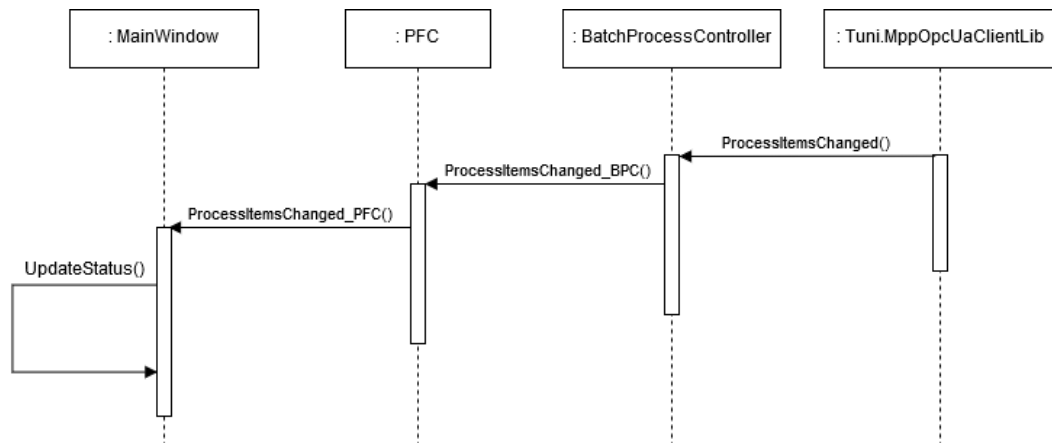
MainWindow luokassa oleva metodi "connectButton_Click" aloittaa yhteyden avaamiseen liittyvän käsittelyn. Metodissa käytetään PFC luokan olion metodia "connect_PFC", ja näin ohjelman suoritus siirtyy PFC luokkaan. PFC luokan metodissa käytetään BatchProcessController luokan oliolle "Connect" metodia, ja ohjelman suoritus siirtyy BatchProcessController luokkaan. Connect Metodissa luodaan TuniMppOpcUaClientLib kirjaston MppClient luokan olio. Tämän luokan rakentajassa on toteutettu varsinainen yhteyden muodostus. Yhteyden muodostamisen jälkeen ohjelman suoritus palaa MainWindow luokkaan.



Kuva 6. Sekvenssin käynnistyminen

Sekvenssin käynnistyminen

MainWindow luokan metodi "StartButton_Click" aloittaa sekvenssin käynnistymiseen liittyvän käsittelyn. Metodissa kutsutaan PFC luokan "mppProcedureSequence" metodia. Tässä metodissa on toteutettuna sekvenssit dokumentin mpp_pfc.pdf mukaan BatchProcessController luokan metodeja käyttäen (esim "on_off"), joista PFC luokan sisällä on aluksi toteutettu alisekvenssit private metodeina, joista varsinaiset pääsekvenssit muodostuvat. Sekvenssi metodien suorituksen jälkeen ohjelman suoritus palaa takaisin MainWindow luokaan.



Kuva 6. Mittausarvojen päivittyminen.

Mittausarvojen päivittäminen

Kuvassa 6 on kuvattu sekvenssikaaviolla mittausarvon päivittäminen. Kun prosessin edetessä simulaattorin arvot muuttuvat lähettää UaLib kirjaston MppClient luokka tästä signaalin. Signaalin vastaanottanut BatchProcessController luokka tallentaa signaalin mukana saadut arvot ja lähettää nämä arvot uudella signaalilla metodissa "ProcessItemChanged". Signaalin vastaanottanut PFC luokka lähettää signaalin mukana tulleen datan vain eteenpäin metodissa "ProcessItemChanged_BPC". Viimein MainWindow luokka ottaa signaalin vastaan metodissa "ProcessItemChanged_PFC" ja tämän jälkeen päivittää käyttöliittymän uusilla arvoilla metodissa "UpdateStatus", mikäli päivitetty mittausarvon tunniste oikea.

4**TOTEUTUS**

4.1**Kehitysympäristö**

Sovelluksen toteutukseen on käytetty Visual Studio 2019 community .NET Framework 4.7.2 kirjasto versiolla, Microsoft Windows 10 Home käyttöjärjestelmällä ja kaavioiden mallintamiseen draw.io websovellusta.