Time and event triggered communication in OPC UA

The objective of this exercise is to familiarize with two communication modes available in OPC UA:

(i)   time triggered communication based on clients' subscription and pooling information source by server

(ii)   event and alarm communication based on clients' notifications in the case of events/alarms detected by server

The OPC UA project is performed under Microsoft Visual Studio 2015 environment including OPC UA SDK C++ library and prepared lesson templates. Created OPC servers and clients should be verified using OPC UA Expert software.

Assumptions: An automatic assembly station performs assembly operations. First it retrieves an element from the input of the station, then performs the assembly operations, and finally passes the assembled element to the output of the station.

The current state of the station is indicated by its status word. STATUS word consists of 7 bits that represent following situations that can appear during production:

- *b0 - RUN* - station performs its function (production running)
- *b1 - ALARM*- work stopped by failure
- *b2 - INTERVENTION* - work stopped by an operator
- *b3 - EMPTY* - station don't work for lack of parts (no input)
- *b4 - BLOCKED* – operation does not run through a closed output (station can't get rid of the machined work piece)
- *b5 - EXCLUDED*– intentionally exclusion (do not work by decision) ☐ *b6 - TIME OUT*- assumed cycle time exceeded

There are binary signals that indicate state of input and output buffer:

- ST_INPUT indicates that there is an element on the input to be assembled by the station
- ST_OUTPUT indicates that there is an element on the output of the station. The element should be removed before the end of next production cycle or station will go to BLOCKED state

The duration of ongoing production cycle is represented by a variable:

- CYCLE_TIME (in seconds). The average value for production cycle is 30s maximum 60 s.

Tasks to be done during the exercise:

1/ Familiarize with "Connecting the Nodes to Real Time Data" as presented in Lesson 3 of Server Getting Started example.

2/ Prepare an OPC UA address space that contains three instances of an automatic assembly station:

- AS1_ gr_sec (gr is your group nb sec section)
- AS21_ gr_sec (gr is your group nb sec section)
- AS2_ gr_sec (gr is your group nb sec section)

Base on AssemblyStationType_prepared in UaModeler. The software part for implementation of station's behavior should be as follows:

a/Simulation has to be prepared on OPC UA server side (as in example given in Lesson 3)
b/An automatic assembly station should perform following behavior:
- the station starts its production cycle when previous cycle is finished and element is available on its input and station is not excluded
- the corresponding status bits should reflect the current state of the station
- the time of production cycle should be simulated by random function with value between 30s and 60s (random value)
- after the end of production cycle the ST_OUTPUT signal should be set to true
- if ST_OUTPUT is equal true and the next production cycle is finished - the station status should be changed to Blocked (until output element is removed)
- the TIME OUT status is generated when a production cycle exceeds 45 s

c/verify your simulation by using OPC UA Expert software

3/ Familiarize with "Adding Support for Events" as presented in Lesson 5 of Server Getting Started example.

4/ Base on the automatic assembly station instance prepared in step 2 implement the following events that should be generated by OPC UA server:
- start of production cycle nr: xxx – where xxx is serial number of part produced (in order to receive serial number, implement simple counter related with each automatic assembly station)
- end of production cycle nr: xxx

5/ Familiarize with "Adding Support for Alarms & Conditionsts" as presented in Lesson 6 of Server Getting Started example.

6/Base on the automatic assembly station instance prepared in step 2 and 4 implement: "Watchdog Alarm" that should be generated by OPC UA server:
- use the UaModeler project to add a new objects called "WatchdogAlarm" and "StationBlocked" as children of AssemblyStationType and choose "OffNormalAlarmType"
- Manage the state of the alarm WatchdogAlarm in the BlockStateChanged EventHandlerend. The alarm should be active when production time exceeds 45 s.
- Manage the state of the alarm StationBlocked in the BlockStateChanged EventHandlerend. The alarm should be active when the station state is "Blocked ".
- Add handling for Acknowledge calls by clients by overriding the Acknowledge method defined at the BaseNodeManager

Report:
1. Archived source configuration files for UAModeler.
2. Archivised server project.
4. Preent screens from OPC UA Expert tests related to steps: 2, 4 & 6
5. Short report with highlighted main steps performed during the exercise