



Simulaatioympäristön liityntäraja- pintojen hyödyntäminen koulutuksessa

Jouko Murtonen

Opinnäytetyö, AMK

Huhtikuu 2024

Insinööri (AMK), sähkö- ja automaatiotekniikan tutkinto-ohjelma

Automaatiotekniikka

Murtonen Jouko

Simulaatioympäristön liityntärajapintojen hyödyntäminen koulutuksessa

Jyväskylä: Jyväskylän ammattikorkeakoulu. Huhtikuu 2024, 97 sivua.

Sähkö- ja automaatiotekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

OPC UA on vahvasti yleistynyt ja tärkeä rajapinta, joten myös sen osaamiselle on tarvetta työelämässä. OPC UA -tekniikkaa ei ole paljon opetettu automaatioinsinöörin koulutuksessa. Sen vuoksi tavoitteena oli tutkia OPC UA -rajapintaa ja miten toteuttaa OPC UA:n yhteys Visual Components -ohjelmaan. Tavoitteena oli myös luoda koulutusmateriaalia OPC UA -rajapinnan käytöstä Visual Components -simulaation ja PLC:n välillä. Tekemääni koulutusmateriaalia voidaan hyödyntää insinöörikoulutuksen kehittämisessä sekä kehittämis- ja tutkimustoiminnassa.

Työ toteutettiin toimintatutkimuksena. Aluksi tutustuttiin OPC UA -rajapintaan ja Visual Components -ohjelmaan. Tämän jälkeen tutkittiin, kuinka toteuttaa OPC UA -palvelin, jonka avulla saadaan luotua yhteys Visual Componentsin ja PLC:n välille. Tutkinnan tuloksena päädyttiin valitsemaan PLC:ksi Beckoffin TwinCAT ohjelman. Onnistuneen testauksen jälkeen luotiin koulutusmateriaali, jota voidaan hyödyntää koulutuksessa. Koulutusmateriaali sisältää ohjeet, kuinka Visual Components -ohjelmaan lisätään OPC UA -yhteys, jonka avulla tiedonsiirto onnistuu TwinCAT:n ja Visual Componentsin välillä.

Työssä keskityttiin OPC UA:n asiakas-palvelin-malliin ja sen perusteisiin ja kehitettiin koulutusmateriaalia. Koulutusmateriaali sisältää ohjeet ja esimerkkiohjelmat Visual Componentsille ja TwinCAT:lle. Työ oli pintaraapaisu OPC UA:sta ja tarjoaa pohjan jatkotutkimukselle. Jatkotutkimusaiheita olisivat mm. julkaisija-tilaaja-malliin tutustuminen ja fyysisen robottisolun mallinnus Visual Componentsiin, joka toimisi synkronoidusti fyysisen robottisolun kanssa.

Avainsanat (asiasanat)

OPC UA, PLC, Visual Components, IoT

Muut tiedot (salassa pidettävät liitteet)

-

Murtonen Jouko

Utilization of the interfaces of the simulation environment in education

Jyväskylä: JAMK University of Applied Sciences, April 2024, 97 pages.

Degree Programme in Electrical and Automation Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

OPC UA is a very common and important interface, so there is also a need for its knowledge in working life. OPC UA technology has not been taught much in automation engineer training. Therefore, the goal was to study the OPC UA interface and how to implement the connection of OPC UA to the Visual Components program. The goal was also to create training material on the use of the OPC UA interface between the Visual Components simulation and the PLC. The educational material I have made can be used in the development of engineering education and in development and research activities.

The work was carried out as an operational study. At first, we got acquainted with the OPC UA interface and the Visual Components program. After that, it was investigated how to implement an OPC UA server, which can be used to create a connection between Visual Components and PLC. As a result of the investigation, it was decided to choose Beckhoff's TwinCAT program as the PLC. After successful testing, training material was created that can be used in training. The training material contains instructions on how to add an OPC UA connection to the Visual Components program, which enables data transfer between TwinCAT and Visual Components.

The work focused on the OPC UA client-server model and its basics and developed training material. The training material contains instructions and example programs for Visual Components and TwinCAT. The work scratched the surface of OPC UA and provides a basis for further research. Further research topics would be e.g. Getting to know the publisher-subscriber model and modeling the physical robot cell to Visual Components, which would work in sync with the physical robot cell.

Keywords/tags (subjects)

OPC UA, PLC, Visual Components, IoT

Miscellaneous (Confidential information)

-

Sisältö

Lyhenteet	4
1 Johdanto	5
1.1 Tavoitteet ja rajaus.....	5
1.2 Käytetyt tutkimus- ja kehitysmenetelmät.....	6
2 OPC UA	6
2.1 Asiakas-palvelin-tiedonsiirtomalli	8
2.2 Osoiteavaruus	10
2.2.1 Solmuluokat	12
2.3 Palvelut.....	13
2.4 Tiedonsiirto	16
2.4.1 Koodaus	17
2.4.2 Turvallisuus	18
2.4.3 Sovelluskerros.....	19
2.4.4 Kommunikointikerros	21
2.4.5 Kuljetuskerros	22
2.5 OPC UA tietomalli.....	23
2.6 Profiilit	24
3 Toteutus	26
3.1 Visual Components	26
3.2 UaExpert.....	26
3.3 Palvelimen valinta	27
3.4 Yhteyden kokeilu	28
3.5 Pick and place -robottisolu.....	30
3.6 OPC UA:n Ohjeistaminen	33
4 Johtopäätökset	34
Lähteet	36
Liitteet	38
Liite 1. OPC UA -palvelimen alustus TwinCAT:lla ja UaExpertillä – Palvelimen yhdistäminen Visual Componentsiin	38

1	OPC UA -palvelimen alustaminen	40
2	TwinCAT-projekti	43
3	Palvelimen alustus TwinCAT:llä	62
4	Palvelimen alustus UaExpertillä	67
5	Data Accessin lisäys	77
6	Testaus TwinCAT:n ja UaExpertin välillä	83
7	Visual Components -mallinnuksen teko ja yhdistäminen OPC UA -palvelimeen	87
8	OPC UA -yhteyden testaus	93

Kuviot

Kuvio 1	OPC UA käyttökohteet (Overview: OPC Unified Architecture, 2010, 4).....	7
Kuvio 2	OPC UA määrittelyt (OPC UA Specifications, n.d.).....	8
Kuvio 3	OPC UA järjestelmäarkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 13).....	9
Kuvio 4	OPC UA asiakas arkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 14).	9
Kuvio 5	OPC UA -palvelimen arkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 15).	10
Kuvio 6	OPC UA -objektimalli (UA Part 3: Address Space Model, 2022, 5).	11
Kuvio 7	Perussolmuluokan ominaisuudet (UA Part 3: Address Space Model, 2022, 21).....	11
Kuvio 8	Esimerkki solmun ominaisuuksista (UaExpert, n.d.).....	12
Kuvio 9	OPC UA objekti malli (UA Part 3: Address Space Model, 2022, 5).....	13
Kuvio 10	Etsintäpalvelun käyttö (Mahnke, Leitner. 2009, 132).....	14
Kuvio 11	Turvakanava palvelusarja (UA Part 4: Services, 2021, 5).....	14
Kuvio 12	Solmujen hallinta palvelusarja (UA Part 4: Services, 2021,6).	15
Kuvio 13	Näkymä- ja kyselypalvelusarjat (UA Part 4: Services, 2021,6).....	15
Kuvio 14	Ominaisuuspalvelusarja (UA Part 4: Services, 2021,7).	16
Kuvio 15	Tietoliikennekanavien eri tasoja (Mahnke, Leitner. 2009, 134).....	17
Kuvio 16	OPC UA kartoitus (Mahnke, Leitner. 2009, 192).....	17
Kuvio 17	Dekoodattu merkkijono UA Binaryn avulla (Mahnke, Leitner. 2009, 193).....	18
Kuvio 18	OPC UA:n turvallisuusarkkitehtuuri (UA Part 2: Security Model, 2022, 13).....	18
Kuvio 19	Käyttäjät TwinCAT:issä.	19
Kuvio 20	OPC UA:n päätepiisteet (An In-Depth Look at OPC-UA Cyber Threats – Part 3: OPC Server Authentication Traffic. 2023).....	20
Kuvio 21	Sertifikaatin vaihto TwinCAT:ssä (Certificate exchange, n.d.).....	21
Kuvio 22	Asiakas käynnistää OPC UA -yhteyden (UA Part 6: Mappings,2022, 68).....	22
Kuvio 23	UA TCP -rakenne (Mahnke, Leitner. 2009, 198).	23

Kuvio 24 OPC UA -arkkitehtuuri (UA Companion Specifications. n.d.).....	23
Kuvio 25 Profiili, yhdenmukaisuusyksiköt ja testitapaukset (Mahnke, Leitner, 2009, 300).....	25
Kuvio 26 UA-TCP UA-SC UA-Binary (Profile Reporting, n.d.).	26
Kuvio 27 OPC UA -osoiteavaruus UaExpertissä.	28
Kuvio 28 Palvelimen onnistunut yhdistäminen Visual Componentsissa.	28
Kuvio 29 Tekemäni TwinCAT-ohjelma.	29
Kuvio 30 Onnistunut testi Visual Componentsissa.	30
Kuvio 31 Tekemäni Pick and place -robottisolun.....	31
Kuvio 32 Tekemäni Pick and place -robottisolun muuttajat.	31
Kuvio 33 Pick and place -robottisolun TwinCAT-ohjelma.	32
Kuvio 34 HMI-käyttöliittymä.	32
Kuvio 35 Valmis Pick and place -robottisolun OPC UA yhteydellä.....	33

Lyhenteet

AES = Advanced Encryption Standard

API = Application Programming Interface

ERP = Enterprise Resource Planning

GDS = GlobalDiscoveryServer

HMI = Human Machine Interface

IP = Internet Protocol

OPC UA = Open Platform Communications Unified Architecture

SHA = Secure Hash Algorithm

TCP = Transmission Control Protocol

1 Johdanto

Työn tarkoituksena oli testata ja ohjeistaa OPC UA -rajapinnan käyttämistä Visual Components -simulaation ja PLC:n välillä. Työn tavoitteena oli myös luoda koulutusmateriaalia OPC UA -rajapinnan koulutukseen.

Opinnäytetyö on merkittävä, koska tähän mennessä OPC UA -tekniikkaa ei ole paljon opetettu automaatioinsinöörin koulutuksessa. OPC UA on vahvasti yleistynyt ja tärkeä rajapinta, joten myös sen osaamiselle on tarvetta työelämässä. OPC UA:n tarkoituksena on korvata irralliset määrittelyt ja rajapinnalla on monia hyötyjä. Nykypäivän trendinä on myös ollut viime aikoina etätyön tekeminen. Tällä opinnäytetyöllä voitaisiin mahdollistaa oppilaan työskentely kotona, mikä mahdollistaisi lisää tapoja oppia tavallisen koululla käynnin ohella. Opinnäytetyöstäni saatuja tuloksia voidaan hyödyntää insinöörikoulutuksen kehittämisessä, sekä kehittämis- ja tutkimustoiminnassa.

1.1 Tavoitteet ja rajaus

Tarkastelin aihetta koulutuksen näkökulmasta. Olennaista on, kuinka voin tuoda paremmin ymmärrystä OPC UA -rajapinnasta ja kuinka sitä voidaan hyödyntää koulutuksessa. Tarkastelin myös, minkälaisia ohjelmoitavia logiikoita voidaan käyttää Visual Componentsin kanssa. Näiden avulla sain kattavan kuvan OPC UA -rajapinnan käytöstä simulaatio-ohjelmassa, jonka avulla opiskelijat voisivat ymmärtää paremmin OPC UA -rajapintaa.

Opinnäytetyötä oli rajattu Visual componetsin ja PLC:n välille, jotta aihe ei paisu turhan isoksi ja pystyin syventymään OPC UA -rajapinnan tutkimiseen ja soveltamiseen. Rajauksella pystyin keskittymään ohjeistamiseen ja testaukseen, jotka olivat opinnäytetyöni tavoitteita. Rajaus on myös kehittämisen näkökulmasta hyvä, koska tästä näkökulmasta ei ole aikaisemmin tehty opinnäytetöitä.

Tällä kehittämistyöllä pyrittiin kehittämään insinöörikoulutusta OPC UA:sta ja tuomaan koulutusmateriaalia koulutukseen. Tavoitteena oli myös selvittää erilaisia vaihtoehtoja toteuttaa OPC UA -palvelin, joka toimisi Visual Componetsin kanssa.

Tärkeimpinä kysymyksinä opinnäytetyössä oli:

- Mikä on OPC UA?
- Miksi OPC UA on tärkeä?
- Miten OPC UA palvelin toteutetaan Visual Componentssin kanssa?
- Miten ohjeistaa OPC UA -rajapinnan käyttämistä?

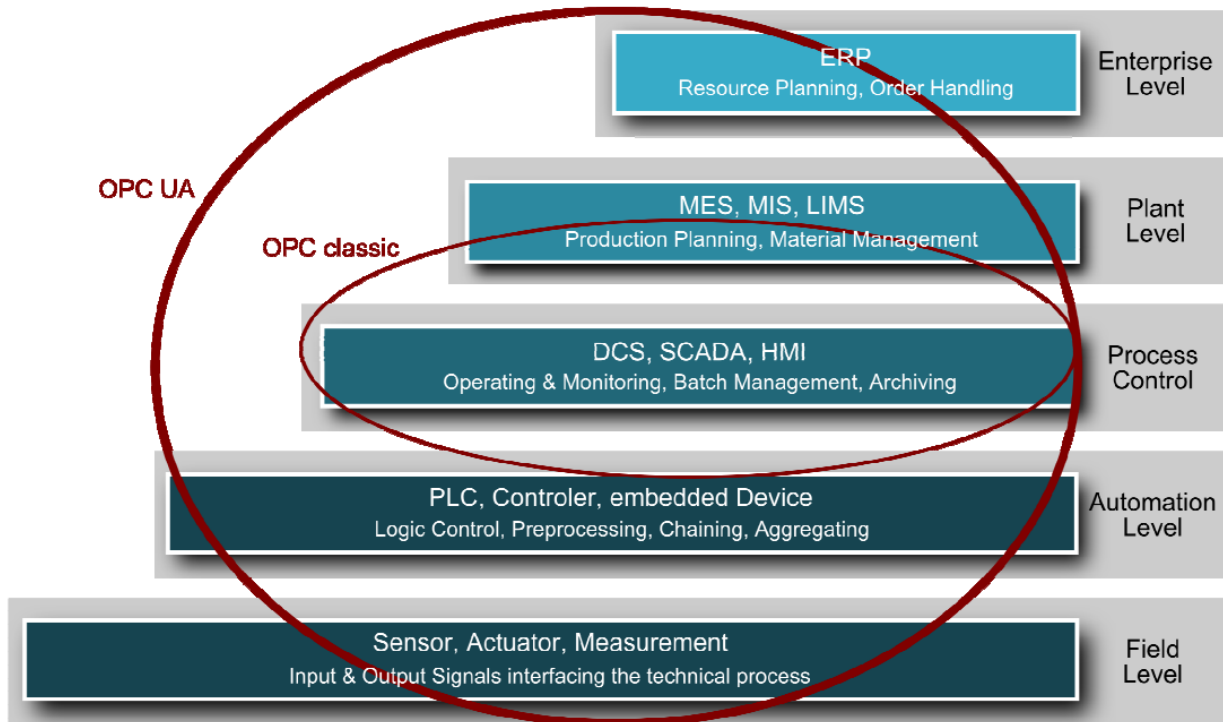
1.2 Käytetyt tutkimus- ja kehitysmenetelmät

Opinnäytetyö oli toimintatutkimusta. Toimintatutkimus on samankaltainen kehittämistutkimuksen kanssa. Niistä löytyy pieniä eroja. Kanasen (2014, 11) mukaan ”toimintatutkimuksessa toteutuvat tutkimus ja toiminta samanaikaisesti”. Toimintatutkimus on käytännönläheistä, tutkija pääsee toimimaan aktiivisena toimijana. Toimintatutkimuksella pyritään saamaan aikaan muutosta kartoittamalla ensin nykytila, minkä jälkeen mietitään ongelmaa. Sitten tehdään parannusehdotus, kokeillaan, arvioidaan ja seurataan tilannetta. Sen jälkeen tehdään sama kaava uudestaan niin paljon, kuin on tarvetta. Tämä malli toistui useasti opinnäytetyötä tehdessäni.

2 OPC UA

OPC UA on lyhenne sanasta Open Platform Communications Unified Architecture. OPC UA -rajapinta on alustariippumaton, tietoturvallinen, laajennettevissa oleva palvelukeskeinen arkkitehtuuri. (Unified Architecture, n.d.)

OPC UA -rajapinta soveltuu monenlaiseen käyttöön. Sitä voidaan käyttää mm. yhdestä anturin mittauksesta suurempiin kokonaisuuksiin ja ohjelmoitavista logiikoista hajautettuihin prosessinohjauksiin (UA Part 1: Overview and Concepts, 2022, 10). OPC UA skaalautuu antureista ERP järjestelmiin ja siltä väliltä (Introduction to OPC Unified Architecture). Kuviosta 1 näemme, että OPC UA -rajapintaa voidaan hyödyntää monella eri tasolla. OPC UA käyttökohteet (Overview: OPC Unified Architecture, 2010, 4).



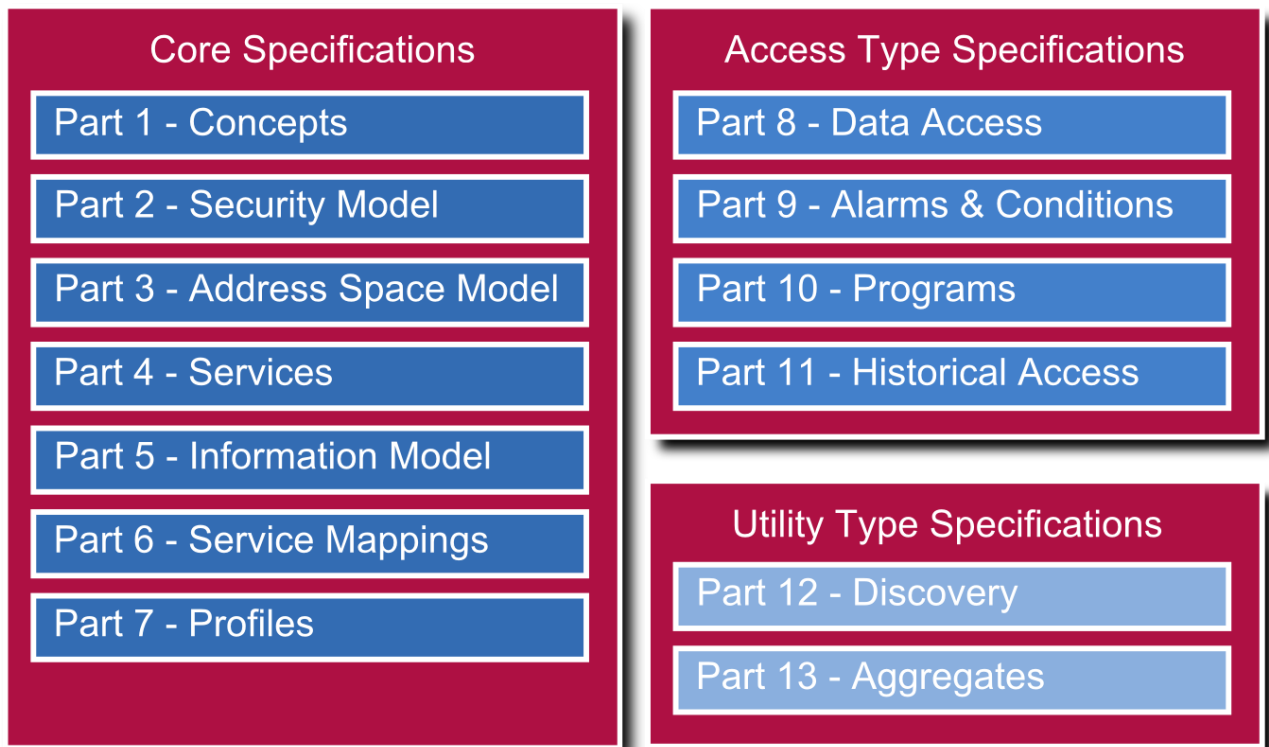
Kuvio 1 OPC UA käyttökohteet (Overview: OPC Unified Architecture, 2010, 4).

Ascolab:in verkkosivuilla olevan artikkelin OPC UA Overview (n.d.) mukaan OPC UA -rajapinnan hyötyjä ovat:

- Skaalautuvuus
- Dynaaminen osoiteavaruus
- Tietoturvallisuus
- Nopeus
- Laajennettavuus
- Palvelukeskeisyys
- Alustariippumattomuus

OPC UA koostuu monista eri osista. OPC UA:n määrittelyt antavat kattavan kuvan OPC UA-rajapinnasta. Määrittelyt on jaettu 13 eri osaan. Kuviossa 2 havainnollistetaan määrittelyt, jotka on jaettu kolmeen eri ryhmään, jotka ovat ydinmäärittelyt (Core Specifications), sisällepääsyn määrittelyt (Access Type Specifications) ja apuohjelmamäärittelyt (Utility Type Specifications). Ydinmäärittelyt on tärkeä osa, josta saa tietoa OPC UA:n rungosta ja peruskonsepteista. Sisällepääsyn määrittelyt kertovat, kuinka dataa voidaan siirtää ja käyttää eri toiminnoilla.

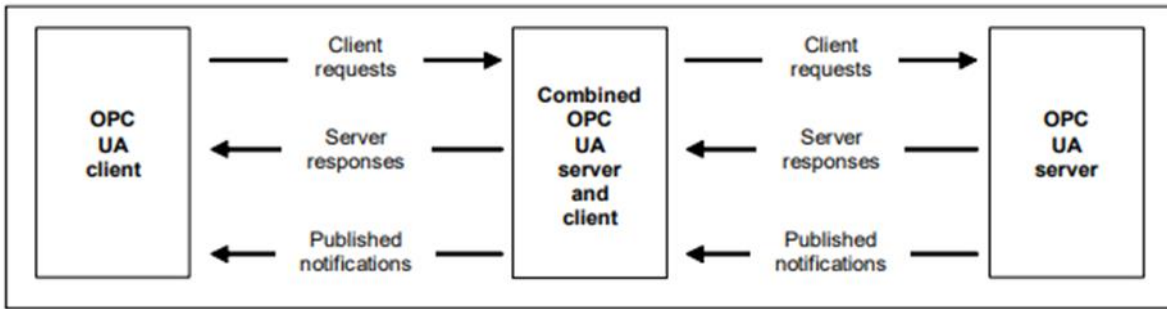
Apuohjelmamäärittelyt puolestaan kertovat yhteyksien muodostamisesta ja kuinka seurata dataa (Mahnke, Leitner. 2009, 13).



Kuvio 2 OPC UA määrittelyt (OPC UA Specifications, n.d.).

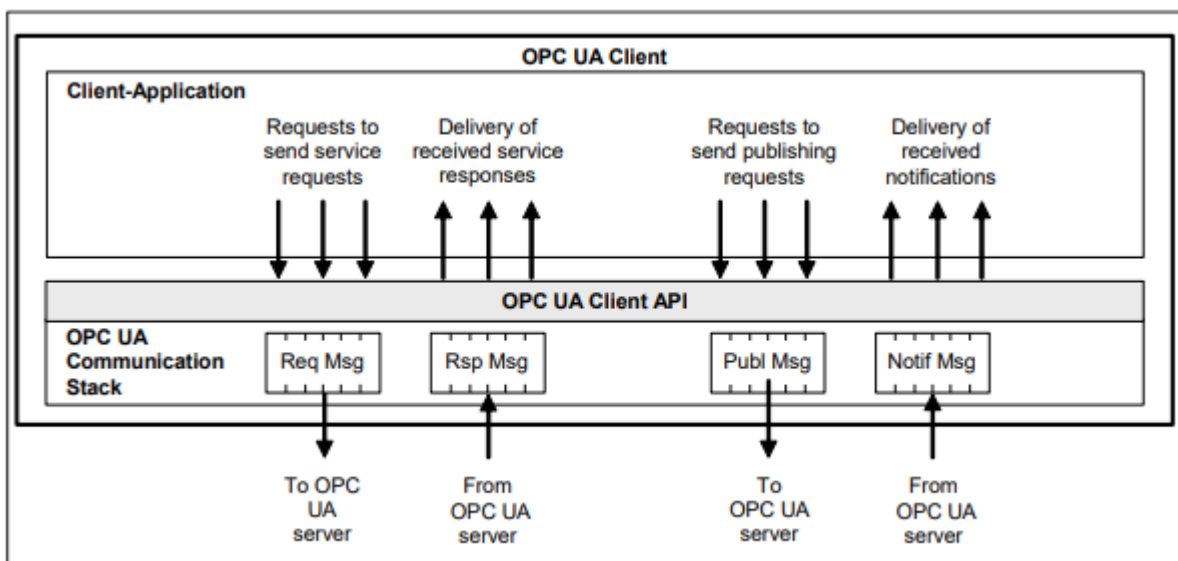
2.1 Asiakas-palvelin-tiedonsiirtomalli

OPC UA toimii asiakas-palvelin-tiedonsiirtomallilla. Kuvio 3 näyttää OPC UA järjestelmäarkkitehtuurin, jonka mukaan tiedonsiirtomalli toimii. Asiakas pyytää palvelimelta dataa, minkä jälkeen palvelin vastaa viestiin ja antaa pääsyn dataan. Asiakas pystyy lukemaan ja kirjoittamaan dataa, eli palvelin ja asiakas toimivat samanaikaisesti. Yhdessä OPC UA -systeemissä voi olla monia palvelimia ja asiakkaita, jotka voivat keskustella toistensa kanssa samanaikaisesti (UA Part 1: Overview and Concepts, 2022, 13).



Kuvio 3 OPC UA järjestelmäarkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 13).

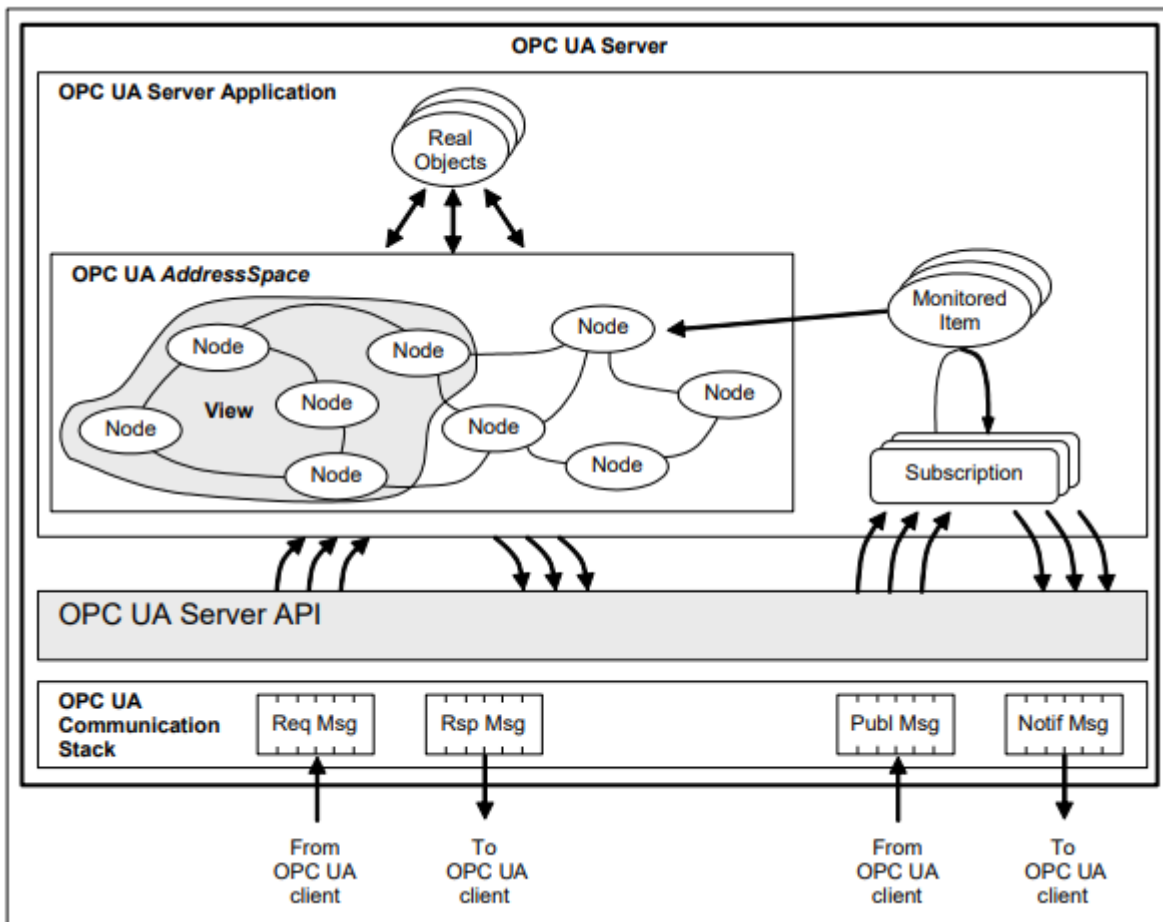
Kuvio 4 näyttää asiakkaan arkkitehtuurin ja havainnollistaa, kuinka asiakas toimii. Asiakas pyyntö (Client Application) toteuttaa asiakkaan toimintoja sen sisältämän koodin avulla ja asiakas API eli ohjelmointirajapinnan avulla se lähettää ja vastaanottaa OPC UA palvelu pyyntöjä ja vastauksia. (UA Part 1: Overview and Concepts, 2022, 14)



Kuvio 4 OPC UA asiakas arkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 14).

Kuviossa 5 havainnollistetaan palvelimen toimintaa. Aidot objektit (Real Objects) edustavat fyysisiä tai ohjelmallisia objekteja, jotka ovat palvelimen saatavilla. Esim. lämpötila-anturi tai simulaatiosovelluksen anturi. Osoiteavaruuden solmut edustavat aitoja objekteja, joihin asiakas pääsee käsiksi palveluitten avulla. Palvelinpyyntö (Server Application) käyttää palvelimen

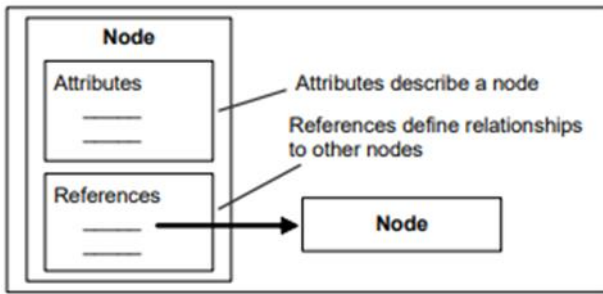
ohjelmointirajapintaa lähettämään ja vastaanottamaan OPC UA -asiakkaiden viestejä (UA Part 1: Overview and Concepts, 2022, 15).



Kuvio 5 OPC UA -palvelimen arkkitehtuuri (UA Part 1: Overview and Concepts, 2022, 15).

2.2 Osoitevaruus

OPC UA -osoitevaruus (AddressSpace) toimii palvelimen tapana esittää objektit asiakkaille. Osoitevaruudessa on objekteja, joita palvelin vie asiakkaalle, kuten kuvio 6 esittää. OPC UA -objektimalli koostuu solmuista (node), joka pitää sisällään kuvauksen solmun ominaisuuksista ja referensseistä, joista saa selville solmun suhteen muihin solmuihin (UA Part 3: Address Space Model, 2022, 5. Mahnke, Leitner, 2009, 24).



Kuvio 6 OPC UA -objektimalli (UA Part 3: Address Space Model, 2022, 5).

Solmu pitää sisällään eri solmuluokkia jotka näkyvät kuviossa 7. Näistä ominaisuuksista solmuid (NodeId), solmuluokka (NodeClass), selainnimi (BrowseName) ja näyttönimi (DisplayName) ovat pakollisia ominaisuuksia (UA Part 3: Address Space Model, 2022, 21-22).

Name	Use	Data Type
Attributes		
NodeId	M	NodeId
NodeClass	M	NodeClass
BrowseName	M	QualifiedName
DisplayName	M	LocalizedText
Description	O	LocalizedText
WriteMask	O	AttributeWriteMask
UserWriteMask	O	AttributeWriteMask
RolePermissions	O	RolePermissionType[]
UserRolePermissions	O	RolePermissionType[]
AccessRestrictions	O	AccessRestrictionType

Kuvio 7 Perussolmuluokan ominaisuudet (UA Part 3: Address Space Model, 2022, 21).

- Solmuid toimii yksilöllisenä osoitteena serverissä. Solmuid sisältää nimiavaruusindeksin (namespaceindex), tunnistetyypin (identifiertype) ja tunnisteen (identifier).
- Solmuluokka joka kertoo onko kyseessä objekti, muuttuja (Variable) vaiko metodi(Method).
- Selainnimi (BrowseName) on lokalisoinaton nimi eli käytännössä miten kone löytää tämän solmun jota käytetään polkujen luomiseen.
- Näyttönimi (DisplayName) on lokasiloitu nimi eli nimi joka näkyy asiakkaan näytöllä.
- Kuvaus (Description) on lokalisoitu kuvaus.

Kuviossa 8 havainnollistetaan, kuinka solmun ominaisuudet näkyvät UaExpert-ohjelmassa.

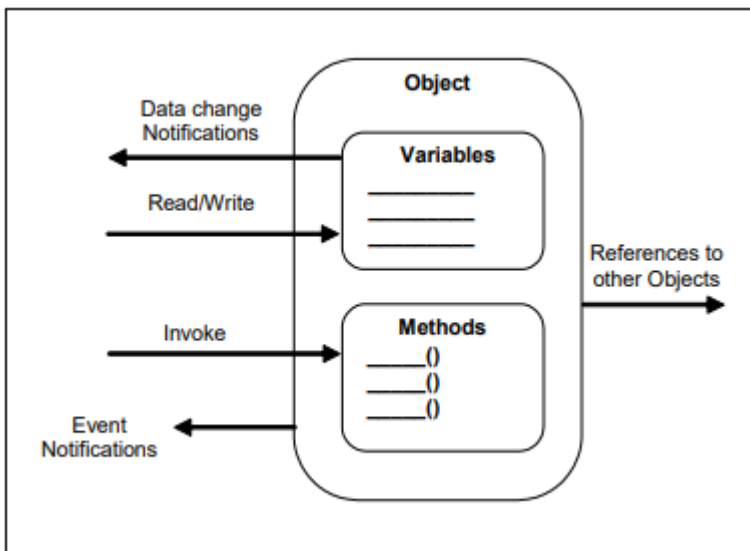
Attribute	Value
▼ Nodeld	ns=4;s=GVL.Motor_ON_OFF
NamespaceIndex	4
IdentifierType	String
Identifier	GVL.Motor_ON_OFF
NodeClass	Variable
BrowseName	4, "Motor_ON_OFF"
DisplayName	"", "Motor_ON_OFF"
Description	"", ""

Kuvio 8 Esimerkki solmun ominaisuuksista (UaExpert, n.d.).

2.2.1 Solmuluokat

OPC UA:n tärkeimmät solmuluokat ovat objekti (Object), muuttuja (Variable) ja metodi (Method). Nämä konseptit ovat tuttuja olio-ohjelmoinnista.

Objektit sisältävät metodeita ja muuttujia. Metodi määrittelee objektien käyttäytymisen. Muuttujat ovat arvoja. Muuttujan arvo riippuu siitä, mitä data tyyppiä se on. Asiakkaat voivat lukea, muuttaa ja kirjoittaa muuttujien arvoja (Mahnke, Leitner, 2009, 30). Muuttujien avulla voidaan esimerkiksi mitata lämpötilaa tai säätää lämpötilan asetusarvoja. Metodit sisältävät toimintoja, jotka toimivat tulosten perusteella. Asiakas kutsuu metodeita, minkä jälkeen metodi suorittaa tehtävänsä ja palauttaa tuloksen asiakkaalle (UA Part 3: Address Space Model, 2022, 12). Kuvio 9 havainnollistaa objektimallin.

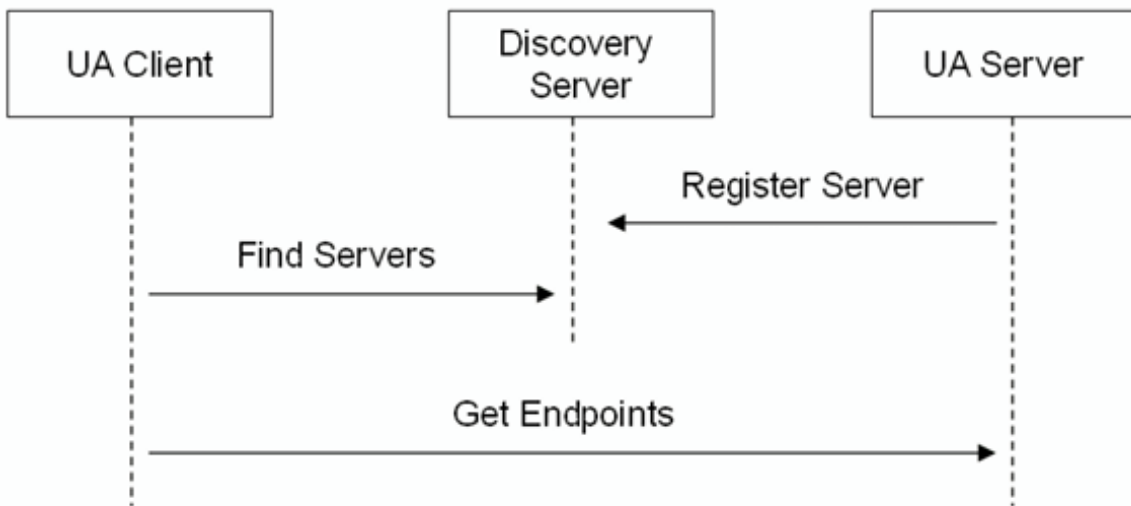


Kuvio 9 OPC UA objekti malli (UA Part 3: Address Space Model, 2022, 5).

2.3 Palvelut

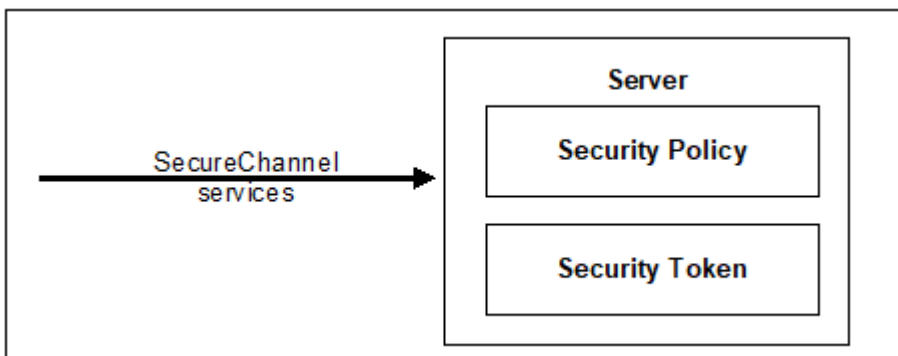
OPC UA on palvelukeskeinen rajapinta joka sisältää erilaisia palveluita (Service set). Palvelut antavat pääsyn palvelimen eri osiin (UA Part 1: Overview and Concepts, 2022, 20). Palveluitten määritelmät ovat abstrakteja eivätkä esitä määrittelyä toteutukselle (UA Part 4: Services, 2021,4). Palvelut toimivat pyyntö ja vastaus -periaatteella. Asiakas lähettää pyynnön palvelimelle, joka käsittelee pyynnön ja lähettää vastauksen takaisin asiakkaalle (Mahnke, Leitner. 2009, 126). OPC UA määrittelee yhteensä 37 palvelua, joista 21:tä käytetään perusrakenteen tietoliikenteen hallintaan ja 16:ta käytetään eri tietojen vaihtamiseen (Mahnke, Leitner. 2009, 130).

Etsintäpalvelu (Discovery service set) nimensä mukaisesti auttaa asiakasta löytämään palvelimen ja päätepisteitten kuvaukset, joita asiakas voi hyödyntää esim. turvalliseen yhdistämiseen. Kuviossa 10 havainnollistetaan etsintäpalvelun toimintaa etsintäpalvelimen, asiakkaan ja palvelimen välillä. Etsintäpalvelin ylläpitää listaa vapaista palvelimista. Jokainen palvelin sisältää vapaat päätepisteet. (Mahnke, Leitner. 2009, 131.)



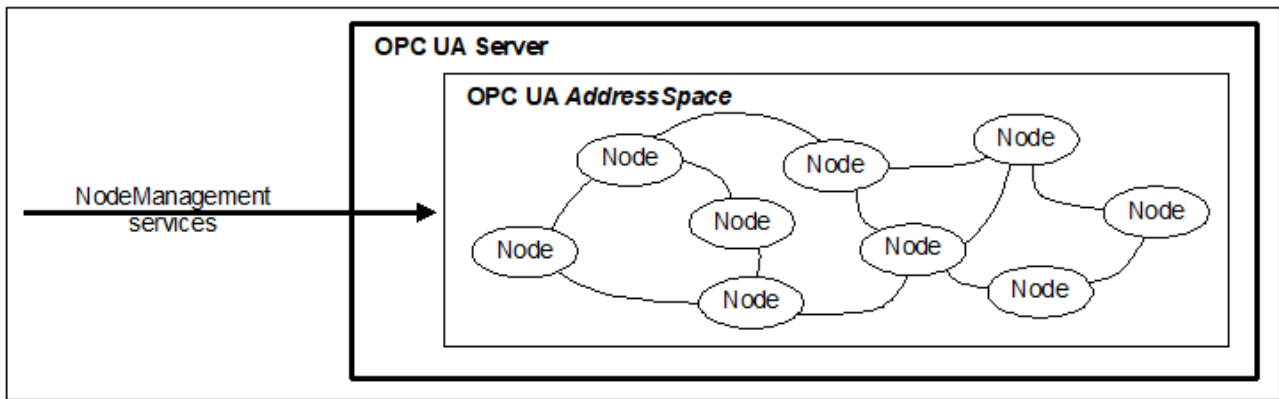
Kuvio 10 Etsintäpalvelun käyttö (Mahnke, Leitner. 2009, 132).

Turvakanavapalvelut (SecureChannel Service Set) auttavat asiakasta luomaan viestien luottamuksellisen ja eheän eli turvallisen yhteyden palvelimen ja asiakkaan välillä. Turvakanava todentaa ja salaa yhteyden, jotta ulkopuolinen yhteys ei voi päästä dataan käsiksi. (UA Part 1: Overview and Concepts, 2022, 21.) Kuvio 11 havainnollistaa turvakanavan toimintaa.



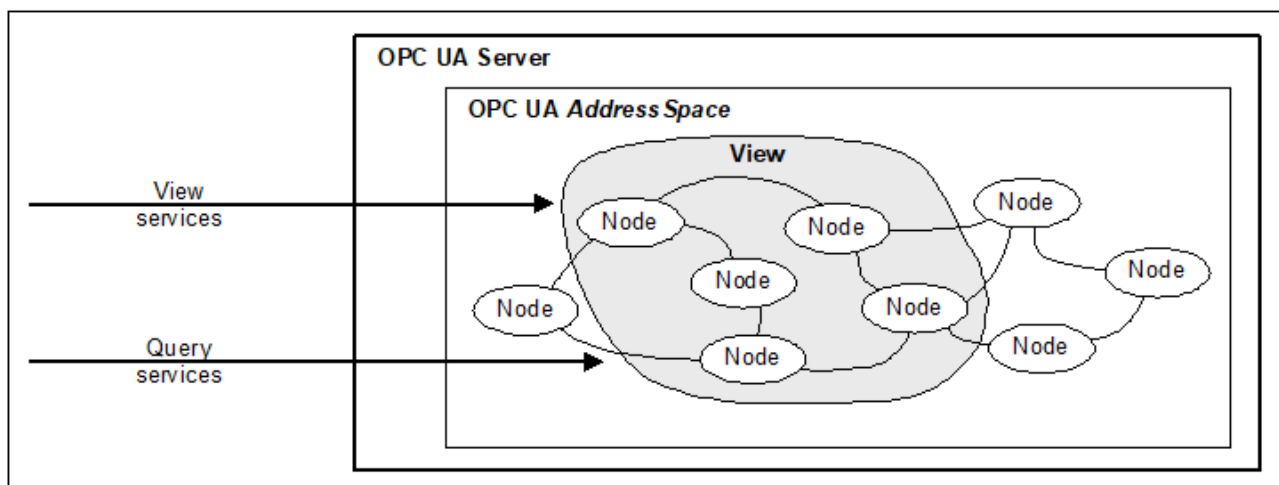
Kuvio 11 Turvakanava palvelusarja (UA Part 4: Services, 2021, 5).

Solmunhallintapalvelun (NodeManagement Service Set) avulla (ks. kuvio 12) palvelimen osoitevaruuteen pystytään lisäämään solmuja ja niitä pystyy myös muokkaamaan ja poistamaan. Palvelulla voidaan myös lisätä ja poistaa referenssejä solmujen välillä. (UA Part 4: Services, 2021,31.)



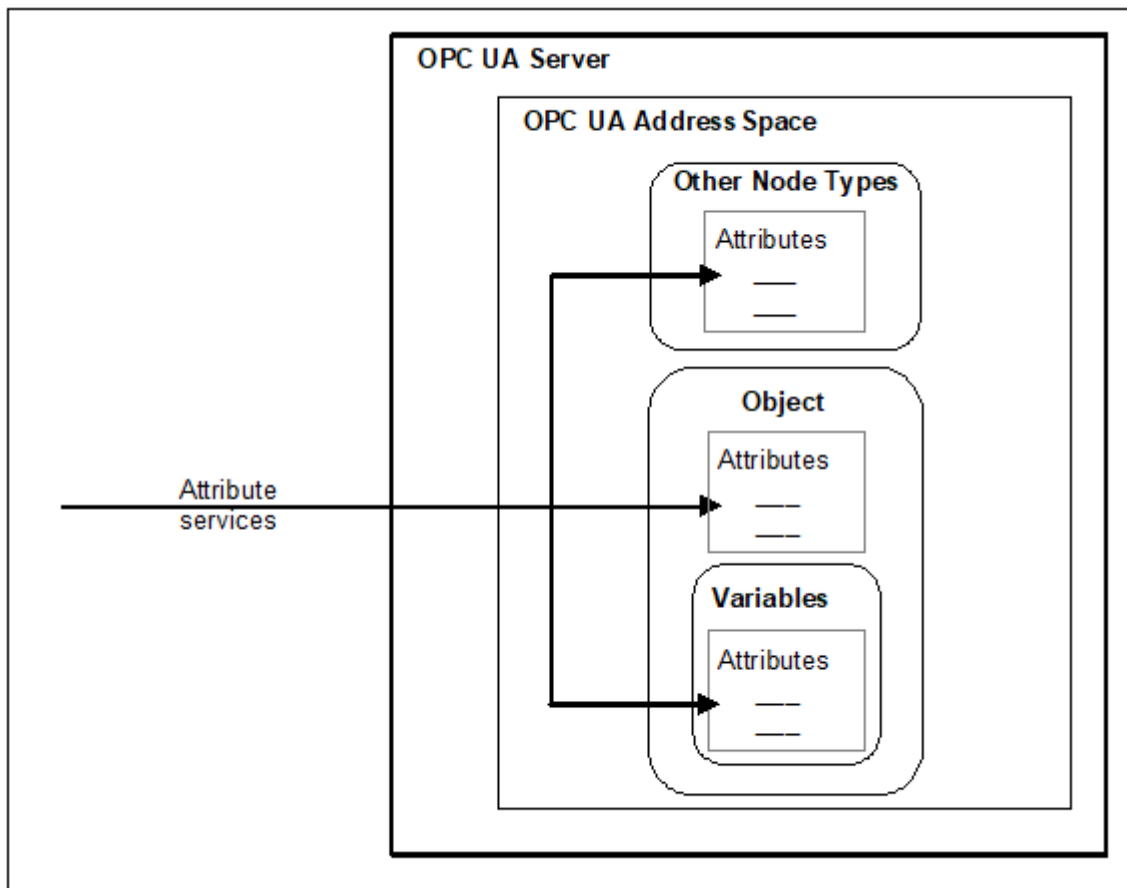
Kuvio 12 Solmujen hallinta palvelusarja (UA Part 4: Services, 2021,6).

Näkymäpalvelusarja (View Service Set) antaa palvelimen osoiteavaruuden asiakkaan nähtäväksi. Sen avulla voidaan etsiä tietoja solmuista. (UA Part 4: Services, 2021,37.) Kyselypalvelusarjalla (Query Service Set) voidaan suorittaa suodatettu haku osoiteavaruudessa (UA Part 4: Services, 2021,44). Kuviossa 13 näkyy molempien palvelusarjojen yhteys osoiteavaruuteen.



Kuvio 13 Näkymä- ja kyselypalvelusarjat (UA Part 4: Services, 2021,6).

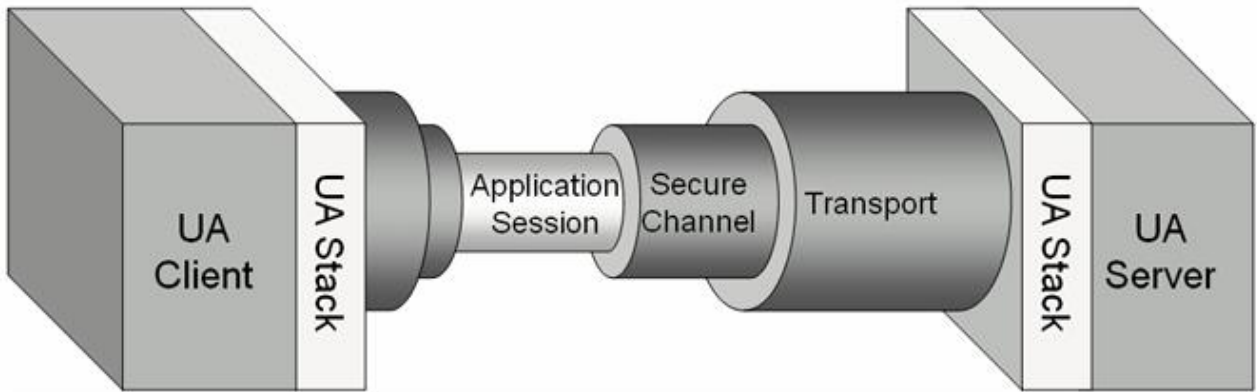
Ominaisuuspalvelusarja (Attribute Service Set) antaa asiakkaalle pääsyn solmujen arvoihin. Kuvio 14 esittää ominaisuuspalveluitten toimintaa. Tämän palvelun avulla asiakkaat voivat lukea tai kirjoittaa arvoja.



Kuvio 14 Ominaisuuspalvelusarja (UA Part 4: Services, 2021,7).

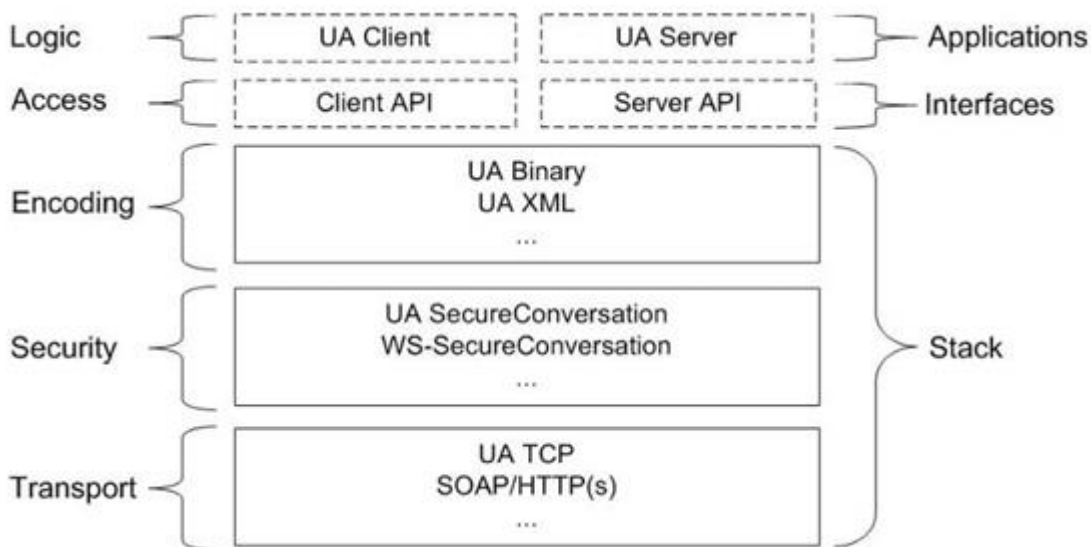
2.4 Tiedonsiirto

OPC UA vaatii tietoliikenne kanavien eri tasoja jotta saadaan luotua turvallinen, joustava ja luotettava tietoliikenne. Kuviossa 15 näytetään tietoliikenne kanavien eri tasoja palvelimen ja asiakkaan välillä.



Kuvio 15 Tietoliikennekanavien eri tasoja (Mahnke, Leitner. 2009, 134)

OPC UA kartoitus (Mapping) on määrittely, kuinka toteuttaa OPC UA ominaisuus tietyllä teknologialla. (UA Part 6: Mappings, 2022, 3) Kuvio 16 esittää yksityiskohtaisen kartoituksen josta näemme että OPC UA pino (Stack) muodostuu dekodauksesta, turvallisuudesta ja kuljetuksesta.



Kuvio 16 OPC UA kartoitus (Mahnke, Leitner. 2009, 192)

2.4.1 Koodaus

OPC UA -rajapinnan koodaus tapahtuu UA Binaryn avulla, joka tarjoaa nopean koodauksen ja dekodauksen. UA Binaryssa on monia etuja. Se mm. käyttää vähän resursseja ja kommunikointiin vain yhtä TCP porttia 4840. (OPC UA Protocols, n.d.; UA Part 6: Mappings, 2022, 10.)

Kuvio 17 esittää esimerkin ”OPCUA”-merkkijonosta, joka on dekodattu UA Binaryn mukaisesti. Numero 5 kertoo merkkijonon pituuden, ja merkkijonossa käytetään utf-8-kirjainformaattia.

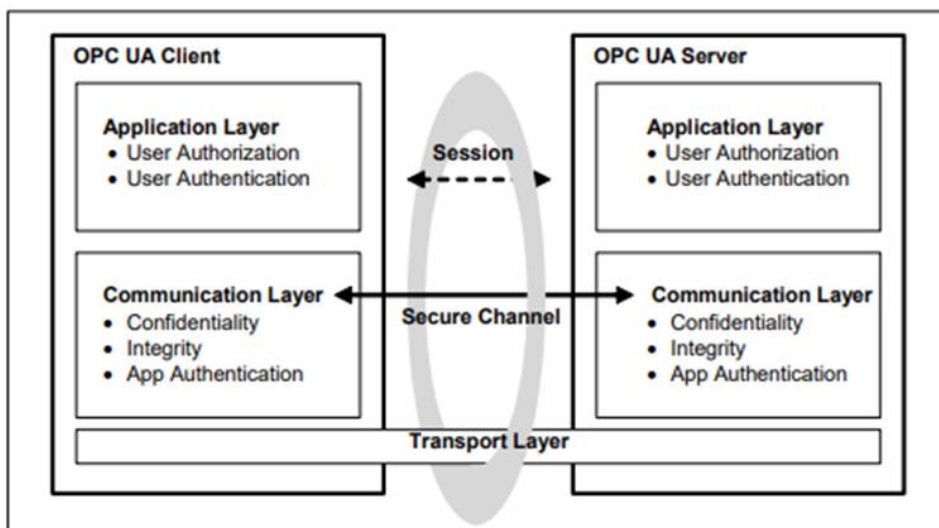
5				O	P	C	U	A
05	00	00	00	4F	50	43	55	41

Kuvio 17 Dekoodattu merkkijono UA Binaryn avulla (Mahnke, Leitner. 2009, 193).

2.4.2 Turvallisuus

Turvallisuus on oleellinen osa OPC UA -rajapinnassa, koska se on niin laajasti käytetty.

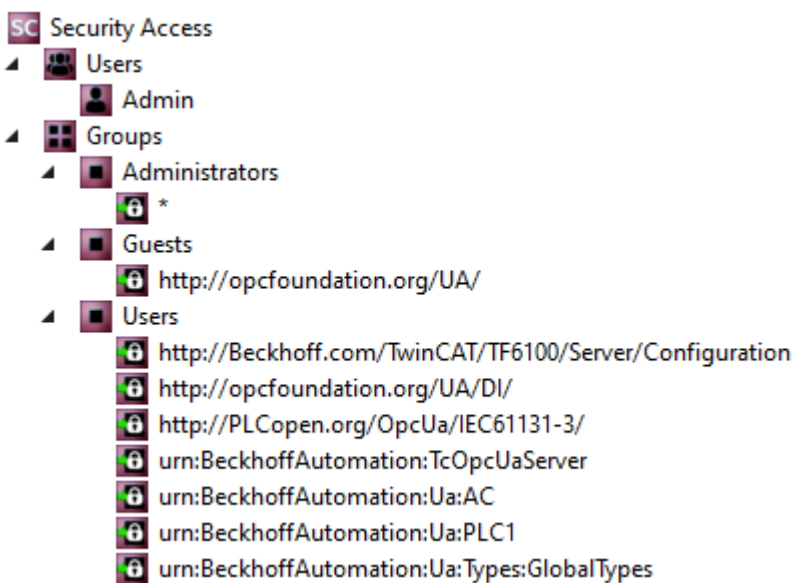
Turvallisuudesta on tärkeää pitää kiinni, ettei data joudu väärin käsiin. Turvallisuudella on kova kysyntä nykyaikana. OPC UA -rajapinnan turvallisuus on otettu huomioon monella tavalla. OPC UA -rajapinnan turvallisuus on monikerroksinen konsepti. Kuvio 18 esittää OPC UA:n turvallisuusarkkitehtuurin, jossa on eri tasoja: sovelluskerros (Application Layer), kommunikointikerros (Communication Layer) ja kuljetuskerros (Transport Layer).



Kuvio 18 OPC UA:n turvallisuusarkkitehtuuri (UA Part 2: Security Model, 2022, 13).

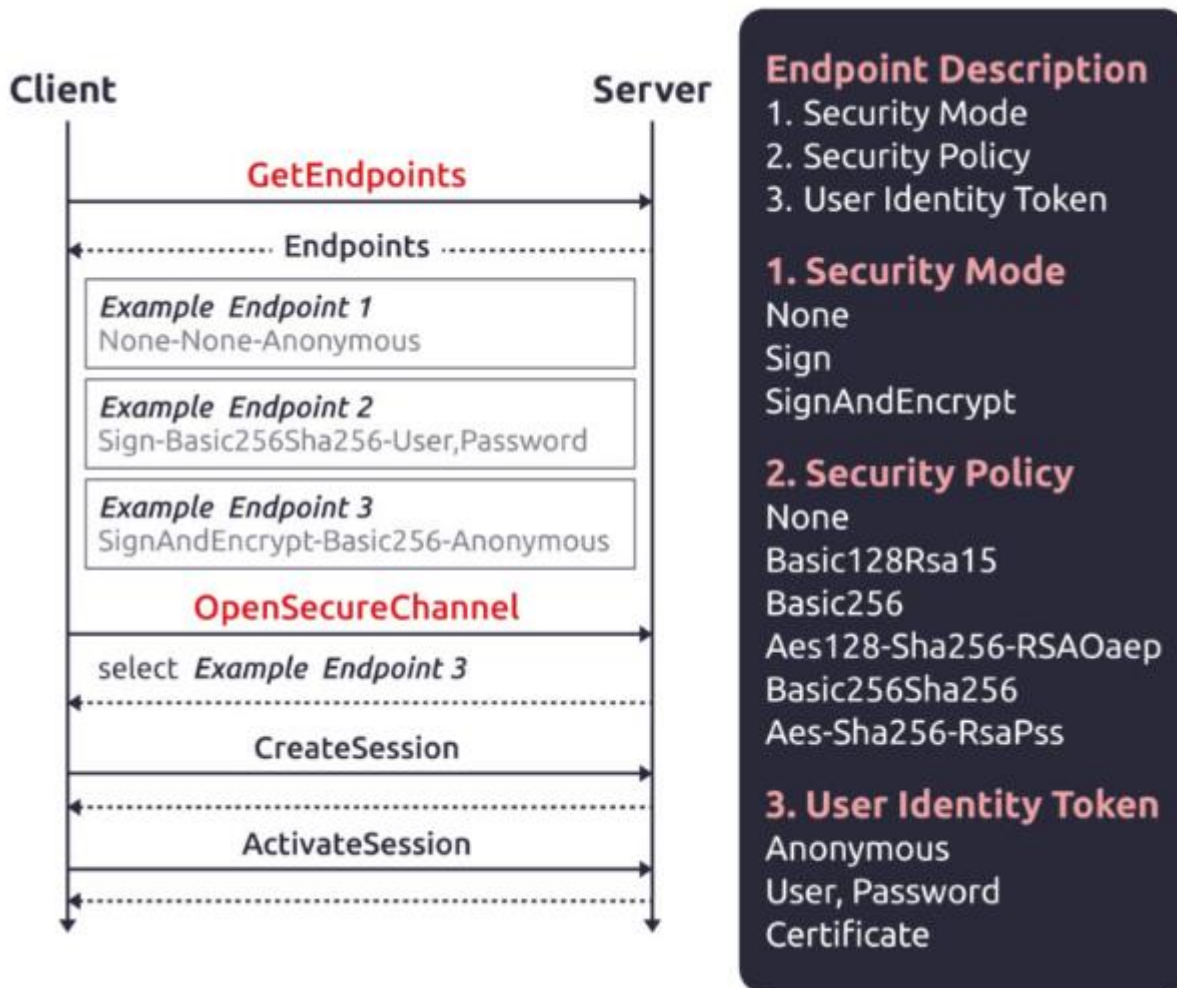
2.4.3 Sovelluskerros

Sovelluskerroksessa käyttäjän tunnistaminen (User authentication) voi tapahtua, joko käyttäjänimellä ja salasanalla tai käyttämällä X.509-todistusta (UA Part 2: Security Model, 2022, 26). OPC UA käyttää X.509-todistusta (Uwe Pohlmann, Axel Sikora. 2018). Sovelluskerroksessa voidaan myös määrittää, mitä eri käyttäjät voivat tehdä. Esimerkiksi järjestelmän valvojalla on kaikki oikeudet ja käyttäjä voi vain lukea solmuja, muttei kirjoittaa uusia arvoja. (OPC UA Security, n.d.) Kuviossa 19 on esimerkki käyttäjistä, joista Admin on järjestelmän valvoja, jolla on kaikki oikeudet. Muilla käyttäjillä on rajoitetut oikeudet.



Kuvio 19 Käyttäjät TwinCAT:issä.

OPC UA sisältää päätepisteitä (Endpoints), joita portti 4840 käyttää. Päätepiste on fyysinen osoite, joka antaa pääsyn palvelimien tarjoamiin palveluihin (UA Part 4: Services, 2021,2). Päätepisteitten avulla määritetään, minkälainen turvallisuus palvelimen ja asiakkaan välillä on. Kuvioista 20 näkyvät päätepisteet ja niiden määrittelyt.



Kuvio 20 OPC UA:n päätepiestet (An In-Depth Look at OPC-UA Cyber Threats – Part 3: OPC Server Authentication Traffic. 2023).

Päätepiesteen määrittelyssä valitaan ensiksi turvallisuustila kolmesta vaihtoehdosta, jotka ovat ”ei salausta”, ”allekirjoitus” tai ”allekirjoitus ja salausta”.

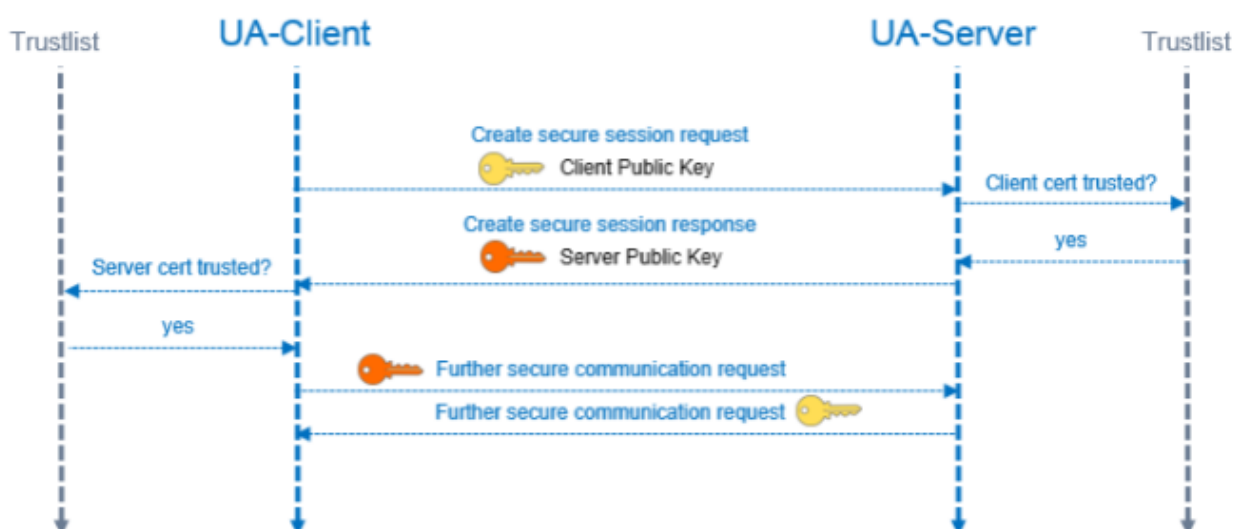
Seuraavaksi valitaan turvallisuuskäytäntö, eli mitä profiilia ja salausalgoritmiä käytetään. ”None”-profiilissa yhteyttä ei ole turvattu. Sitä voidaan käyttää testauksessa, mutta muuten sitä ei suositella käytettäväksi. Basic128Rsa15 ja Basic256 ovat vanhentuneita profiileja (Certificate exchange). Aes128-Sha256-RsaOaep, Basic256Sha256 ja Aes256-Sha256-RsaPss ovat puolestaan hyviä vaihtoehtoja, joissa kaikissa turvallisuuden taso on korkea.

Kolmanneksi valitaan vielä, miten käyttäjä tunnistetaan. Tunnistustavaksi valitaan joko anonyymi, käyttäjä ja salasana tai sertifikaatti. Kun pääte piste on määritelty, on mahdollista luoda turvakanava.

2.4.4 Kommunikointikerros

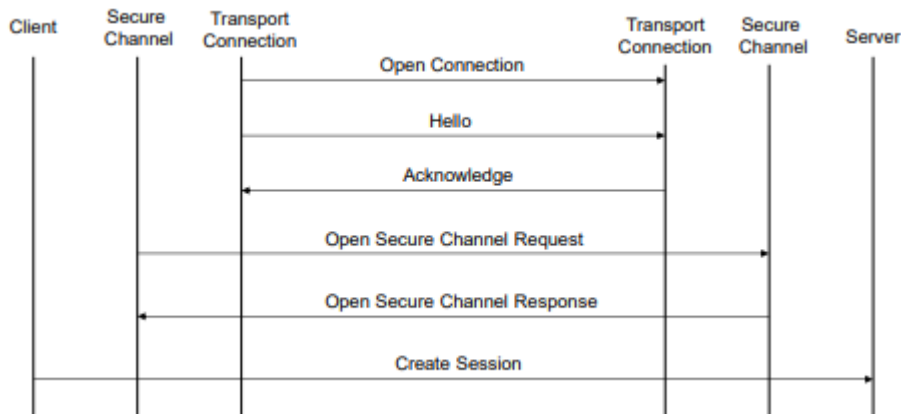
Kommunikointikerros hoitaa luottamuksellisuuden, eheyden ja sovelluksen todennuksen turvakanavan avulla. Turvakanavaa käytetään turvaamaan viestit palvelimen ja asiakkaan välillä. (UA Part 2: Security Model, 2022, 15.) OPC UA käyttää turvallisuusprotokollaa nimeltä UA Secure Conversation, jonka avulla turvataan OPC UA -yhteys käyttämällä binäärikoodattuja viestejä. (Mahnke, Leitner. 2009, 197; UA Part 6: Mappings, 2022, 51.)

Kommunikointikerros sisältää myös sovellustodennuksen. Se tarkoittaa, että palvelimeen lisätään asiakas, niin tehdään sertifikaatin vaihto, eli asiakkaalta tai palvelimelta kysytään, voivatko palvelin ja asiakas luottaa toisiinsa, jotta yhteys saadaan luotua. Tämän avulla saadaan luotua turvallinen yhteys. OPC UA käyttää X.509-sertifikaattia. (Mahnke, Leitner. 2009, 224.) Kuviossa 17 on esimerkki sertifikaatin vaihdosta TwinCAT:ssä jossa käytetään X.509-todistusstandardia. Hyväksytyt sertifikaatin jälkeen asiakkaan sertifikaatti pääsee luottamuslistalle. Palvelimen ja asiakkaan väliseen yhteyteen voidaan nyt luottaa ja muodostaa yhteys niiden välille.



Kuvio 21 Sertifikaatin vaihto TwinCAT:ssä (Certificate exchange, n.d.).

Yhteyden OPC UA- rajapintaan voi muodostaa joko palvelin tai asiakas. Kuviossa 22 näytetään, kuinka asiakas muodostaa turvakanavan. Eli asiakas avaa yhteyden ja lähettää kerran "Hello" -viestin palvelimelle. Jos palvelin saa viestin, palvelin ilmoittaa takaisin asiakkaalle. Jos yhteyttä ei saada, tulee vikailmoitus ja yhteys katkeaa. Kun yhteys on saatu, asiakas lähettää Open Secure Channel -pyynnön, palvelin vastaa ja Secure channel on luotu. Tämän jälkeen luodaan istunto (Session) ja yhteys on käynnistetty. (UA Part 6: Mappings, 2022, 68.)



Kuvio 22 Asiakas käynnistää OPC UA -yhteyden (UA Part 6: Mappings,2022, 68).

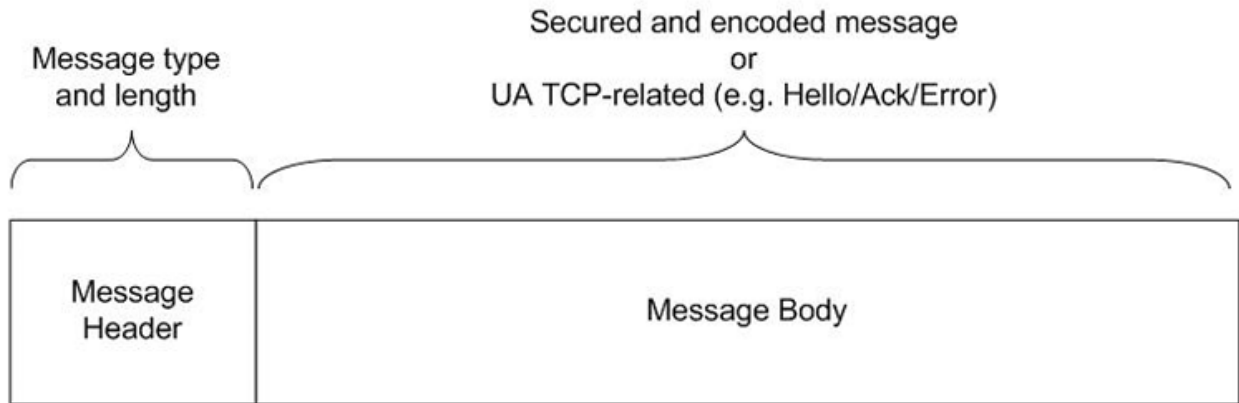
2.4.5 Kuljetuskerros

Kuljetuskerros hoitaa kommunikointikerroksen datan kuljetuksen, lähetyksen ja vastaanoton. Kuljetuskerros saa datan kommunikointikerrokselta ja vie sen asiakkaalta palvelimelle tai palvelimelta asiakkaalle. Jos yhteys katoaa, niin kuljetuskerros hoitaa uudelleen yhdistämisen ilman, että turvakanava häiriintyy. (UA Part 2: Security Model, 2022, 14-15.)

Kuljetuskerros käyttää tiedonsiirrossa UA TCP -protokollaa, joka perustuu TCP ja IP -tiedonsiirto-protokolleihin. TCP tulee sanoista Transmission Control Protocol ja IP sanoista Internet Protocol. Protokollan avulla muodostetaan yhteys verkkotasolla palvelimen ja asiakkaan välillä. (Mahnke, Leitner. 2009, 198.) Kokonaisuudessaan OPC UA -yhteys muodostaa osoitteen, esim. `opc.tcp//192.168.10.10:4840`, joka koostuu TCP-osoitteesta, IP-osoitteesta ja portista 4840.

Kuviossa 23 esitetään UA TCP -viestin rakenne. Viestin otsikko sisältää viestin tyyppin ja pituuden. Viestin runko sen sijaan sisältää suojatun ja koodatun viestin. Viesti kulkee palvelimen ja asiakkaan

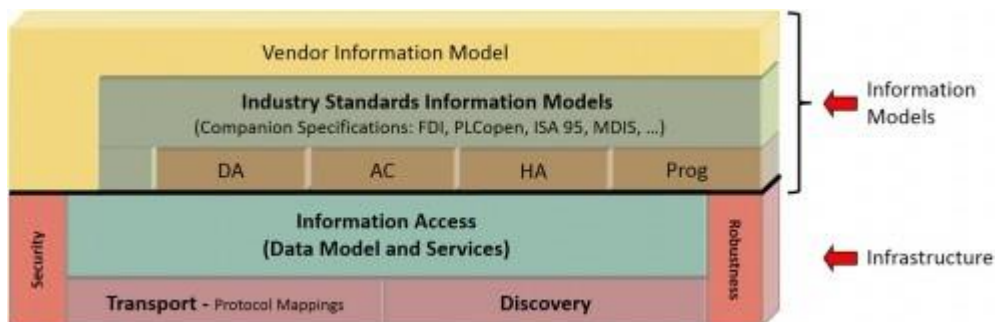
välillä. Viestin runko voi myös lähettää UA TCP:hen liittyviä viestejä, joita on mm. hello-viesti, jolla saadaan yhdistettyä turvakanava, vastaus hello-viestiin ja vikaviesti, jos yhteydessä on jotain vikaa. (Mahnke, Leitner. 2009, 198-199.)



Kuvio 23 UA TCP -rakenne (Mahnke, Leitner. 2009, 198).

2.5 OPC UA tietomalli

OPC UA -tietomalli (Information Model, n.d.) on tärkeä osa, johon kuuluu erilaisia laajennuksia, jonka avulla voidaan esittää osoiteavaruuden tietoa. Kuviossa 24 näkyy, kuinka tietomallit tulevat OPC UA -perusrakenteen päälle. Laajennukset näyttävät, kuinka nykyinen data, historiallinen data, ohjelmat, hälytykset ja ehdot on mallinnettu. (Mahnke, Leitner, 2009, 107.)



Kuvio 24 OPC UA -arkkitehtuuri (UA Companion Specifications. n.d.).

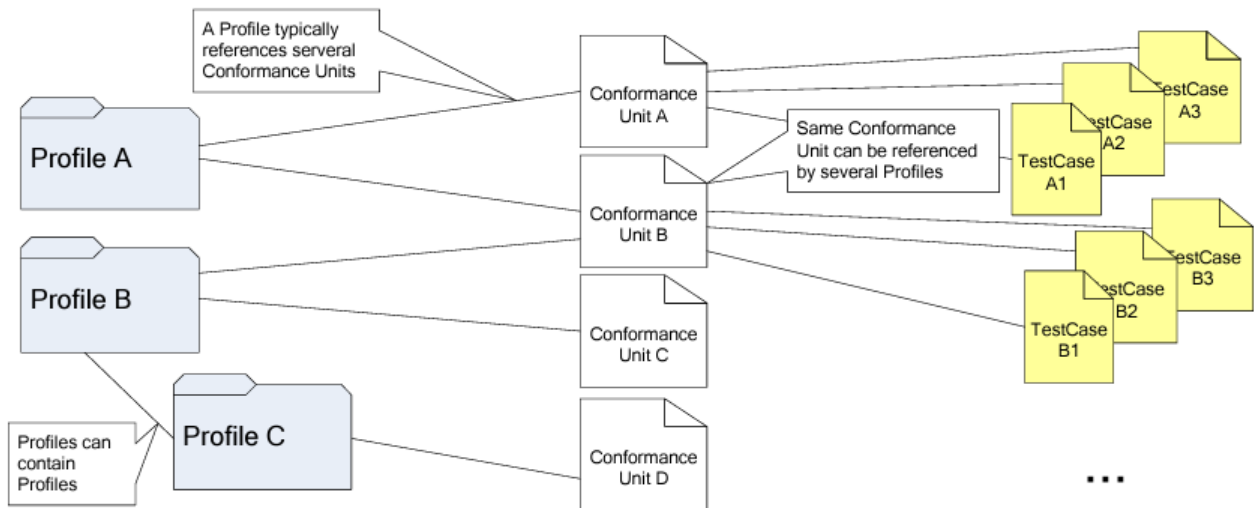
DA-tietomalli eli Data Access tallentaa tiedot palvelimelle, jonka kautta asiakkaat pääsevät muuttujaan käsiksi ja voivat lukea, kirjoittaa tai monitoroida arvon muutoksia reaaliajassa (UA Part

8: Data Access, 2021, 2). Data Accessiä käytetään PLC:ssä liikuttamaan valittua dataa eteenpäin. Muuten data jää piiloon, eikä sitä voida hyödyntää asiakkailta. Data Accessiä hyödynnetään 99 %:ssa OPC -tuotteista, mikä tekee siitä todella tärkeän osan OPC -teknologiaa (Mahnke, Leitner, 2009, 4).

HA-tietomallin eli Historical accessin avulla voidaan katsoa historiallisia arvoja ja historiallisia tapahtumia. AC:n eli Alarms and Conditions -tietomallin avulla voidaan tehdä erilaisia ehtoja ja hälytyksiä. (Mahnke, Leitner, 2009, 119.) Programs-tietomalli määrittelee mekanismin ohjelmien käynnistämiseksi, käsittelemeksi ja suorittamiseksi (Mahnke, Leitner, 2009, 118). Eri organisaatiot voivat rakentaa omia tietomalleja jo valmiina olevien tietomallien päälle (Mahnke, Leitner, 2009, 11).

2.6 Profiilit

OPC UA -sovellukset sisältävät profiileja, jotka määrittävät OPC UA -sovelluksen toiminnallisuuden. Kuvio 25 näyttää, kuinka sovellukset voivat tukea useita profiileja. Profiili voi sisältää muita profiileja. Profiili koostuu yhdenmukaisuusyksiköistä (Conformance units), jotka voivat olla yhteydessä muihin profiileihin. Yhdenmukaisuusyksiköt edustavat tiettyä ominaisuutta ja voivat sisältää testitapauksia. (Mahnke, Leitner, 2009, 299.)



Kuvio 25 Profiili, yhdenmukaisuusyksiköt ja testitapaukset (Mahnke, Leitner, 2009, 300).

OPC UA -profiilit on jaettu 8 kategoriaan (Profile Reporting, n.d.):

- Sovellus
- Palvelin
- Asiakas
- Kuljetus
- Julkaisija
- Tilaaja
- Turvallisuus
- GDS

Esimerkiksi kuljetuskategorian alta löytyy erilaisia profiileja, joista yksi on UA-TCP UA-SC UA-Binary, joka sisältää seuraavat yhdenmukaisuusyksiköt: Protocol UA TCP, UA Binary Encoding ja UA Secure Conversation. Nämä yhdenmukaisuusyksiköt eivät sisällä testitapauksia. Kuvio 26 näemme esimerkin UA-TCP UA-SC UA-Binary -profiilista ja sen yhdenmukaisuusyksiköistä.

Name	UA-TCP UA-SC UA-Binary
Profile URI	http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary
Release Status	Released
Profile Group	UACore 1.05
<p>This transport Facet defines a combination of network protocol, security protocol and message encoding that is optimized for low resource consumption and high performance. It combines the simple TCP based network protocol UA-TCP 1.0 with the binary security protocol UA-SecureConversation 1.0 and the binary message encoding UA-Binary 1.0.</p>	
Included Conformance Units	
Name	Is Optional Description
Protocol and Encoding	
Protocol UA TCP	<input type="checkbox"/> Support the UA TCP transport protocol defined in UA Part 6.
UA Binary Encoding	<input type="checkbox"/> Support UA Binary Encoding. Values of these data types are encoded in compact binary formats, contiguously and without tagging. I.e. the receiver is assumed to understand the structure it is decoding.
UA Secure Conversation	<input type="checkbox"/> Support UA Secure Conversation specified in UA Part 6.

Kuvio 26 UA-TCP UA-SC UA-Binary (Profile Reporting, n.d.).

3 Toteutus

Opinnäytetyö aloitettiin ensin tutustumalla OPC UA -rajapintaan ja Visual Components -ohjelmaan. Tutustumisen jälkeen toimeksiantajan kanssa käyty keskustelu antoi hyvän vision, mihin suuntaan lähteä. Toimintatutkimuksessa kuuluu aloittaa nykytilan kartoituksella, kuten nyt tein. Alkutilanne oli, että opiskelumateriaalia ei juuri ole OPC UA -rajapinnasta suomeksi ja minun piti ottaa selvää eri vaihtoehdoista, joilla yhteys voidaan toteuttaa.

3.1 Visual Components

Opinnäytetyössä käytin Visual Components Premium -ohjelmistoa. Visual Components on simulaatio-ohjelma, jolla voidaan suunnitella, rakentaa ja simuloida tehdasympäristöjä. Visual components on yli 2400 organisaation käyttämä ohjelma, jota käytetään ympäri maailmaa, joten se on erittäin merkittävä ohjelma. (Essentials, n.d.)

3.2 UaExpert

Käytin myös UaExpert-ohjelmistoa, joka on Unified Automationin tekemä OPC UA -asiakas. UaExpert on suunniteltu OPC UA testaukseen. Se on kattava testaustyökalu, joka sisältää erilaisia ominaisuuksia, mm. OPC UA Data Access, OPC UA Alarms & Conditions, Historical Trend ja metodien kutsumisen. Eri ominaisuuksien avulla voidaan tarkastella osoiteavaruutta, jonka avulla voidaan myös kirjoittaa solmuihin arvoja. (UaExpert, n.d.)

3.3 Palvelimen valinta

Toimintatutkimuksessa analysoidaan ongelmatilannetta. Ongelmatilanteeksi koitui yksi opinnäytetyöni kysymyksistä, joka oli: millä tavalla toteuttaa OPC UA -rajapinta Visual Componentsin kanssa. Tätä kysymystä varten otin selvää eri vaihtoehdoista. Aluksi päätin kokeilla tehdä oman palvelimen python-kielellä. Internetin avustuksella sain tehtyä yksinkertaisen palvelimen, jonka sain yhdistettyä Visual Componentsiin. Palvelinta tehdessä opin paljon OPC UA:n peruskäsitteistä ja kuinka solmuja liitetään palvelimeen testauksesta. Testauksessa käytin UAExpert -ohjelmaa, jolla voi tarkastella hyvin itse tehtyä palvelinta. Huomasin myös monia vajavaisuuksia. Koska OPC UA:ssa on monia komplekseja asioita, yksinkertainen python-palvelin ei toiminut yhtä hyvin verrattuna muihin palvelimiin. Vajavaisuuksia olivat muun muassa yhteyden luotettavuus, toimintojen lisääminen ja turvallisuuden puute.

Kokeilin myös tehdä oman palvelimen C#:lla. Vähän aikaa testailtuani totesin, että opinnäytetyöni aihe paisuisi liian laajaksi, jos tekisin oman palvelimen C#:lla.

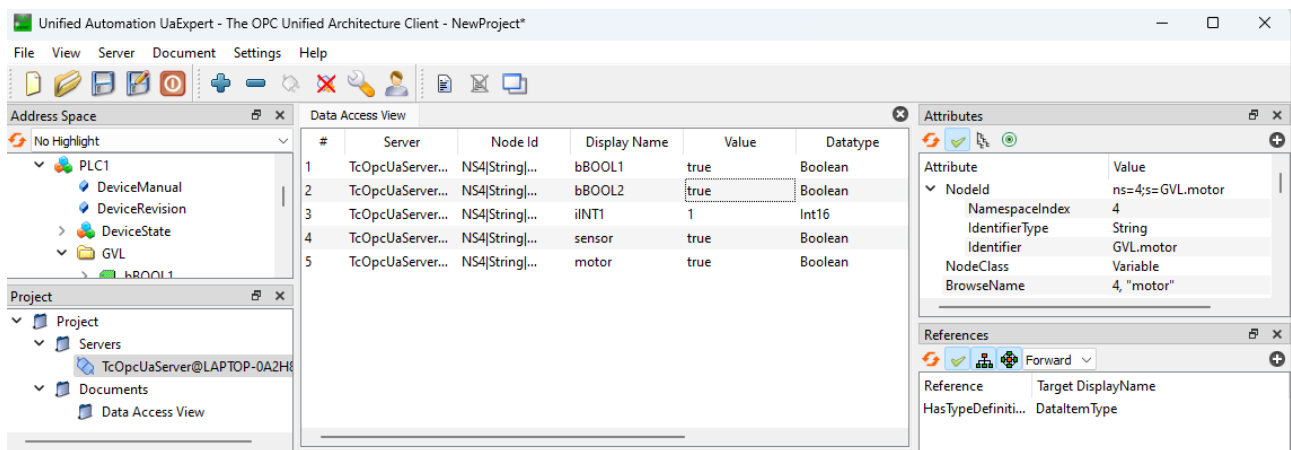
Otin selvää, kuinka OPC UA -palvelimen voisi tehdä Siemensin omalle palvelimelle. Siemensiltä löytyi paljon materiaalia ja tietoa, kuinka tehdä palvelin. Suurimmaksi miinukseksi osoittautui lisenssien ja ohjelmien maksullisuus.

Sain myös toimeksiantajalta kattavan materiaalin B&R Automationilta ja siellä oli paljon hyvää tietoa. Miinukseksi muodostui että B&R Automation kuvaili palvelimen tekoa monimutkaisesti verrattuna Beckoffin materiaaleihin, joita käsittelen seuraavaksi.

Synteesikseni eli parannusehdotukseksi osoittautui Beckoffin TwinCAT 3 -palvelimen käyttäminen. Heidän palvelimessaan on monia hyviä puolia. Ensinnäkin heidän tuottensa ovat ilmaisia eli kuka tahansa voi ladata heidän sovelluksensa ja ympäristönsä. Lisenssit saa aina uusittua ilmaiseksi, joten se on erittäin kätevä alusta opetella palvelimen tekoa ja tutustua OPC UA -rajapintaan. Päädyin myös TwinCAT:iin, koska Beckoffin sivuilta löytyy kattavat käyttöohjeet ja paljon tietoa palvelimen tekemisestä ja käytöstä, joka on vapaasti saatavana kaikille.

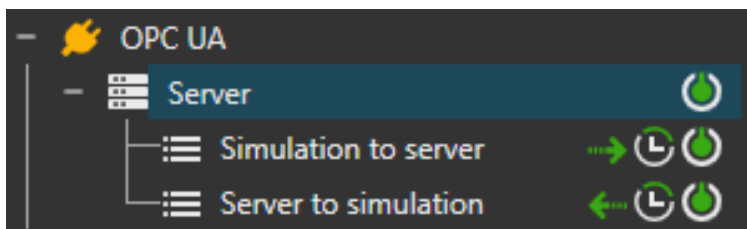
3.4 Yhteyden kokeilu

Palvelimen valittuani pääsin seuraavaan vaiheeseen eli kokeiluun. Aloin tekemään TwinCAT - palvelinta. Palvelimen tekeminen TwinCAT:lla onnistui muutamassa tunnissa, jonka jälkeen pääsin kokeilemaan yhteyttä. Yhteyden kokeilemiseen käytin UaExpert-ohjelmistoa. Yhteyden testaus onnistui ja pääsin käsiksi tekemäni palvelimen osoiteavaruuteen, kuten kuviossa 27 näkyy. Vasemmalla kuviossa voi selata osoiteavaruutta, ja keskellä voi tarkastella solmuja ja muokata niiden arvoja. Oikealla voi katsoa solmujen arvoja ja viittauksia.



Kuvio 27 OPC UA -osoiteavaruus UaExpertissä.

Testasin myös TwinCAT:n ja Visual componentsin yhdistämistä (Kuvio 28).



Kuvio 28 Palvelimen onnistunut yhdistäminen Visual Componentsissa.

Yhdistämisen jälkeen tein TwinCAT-ohjelman (kuvio 29), jolla pystyin testamaan OPC UA - ominaisuuksia. Tekemässäni ohjelmassa laskuri laskee palikoita, jotka menevät sensorin ohi,

kunnes 5 palikkaa on mennyt. Silloin laskuri pysäyttää moottorin ja odottaa 5 sekuntia, minkä jälkeen laskuri nollaantuu ja aloittaa alusta.

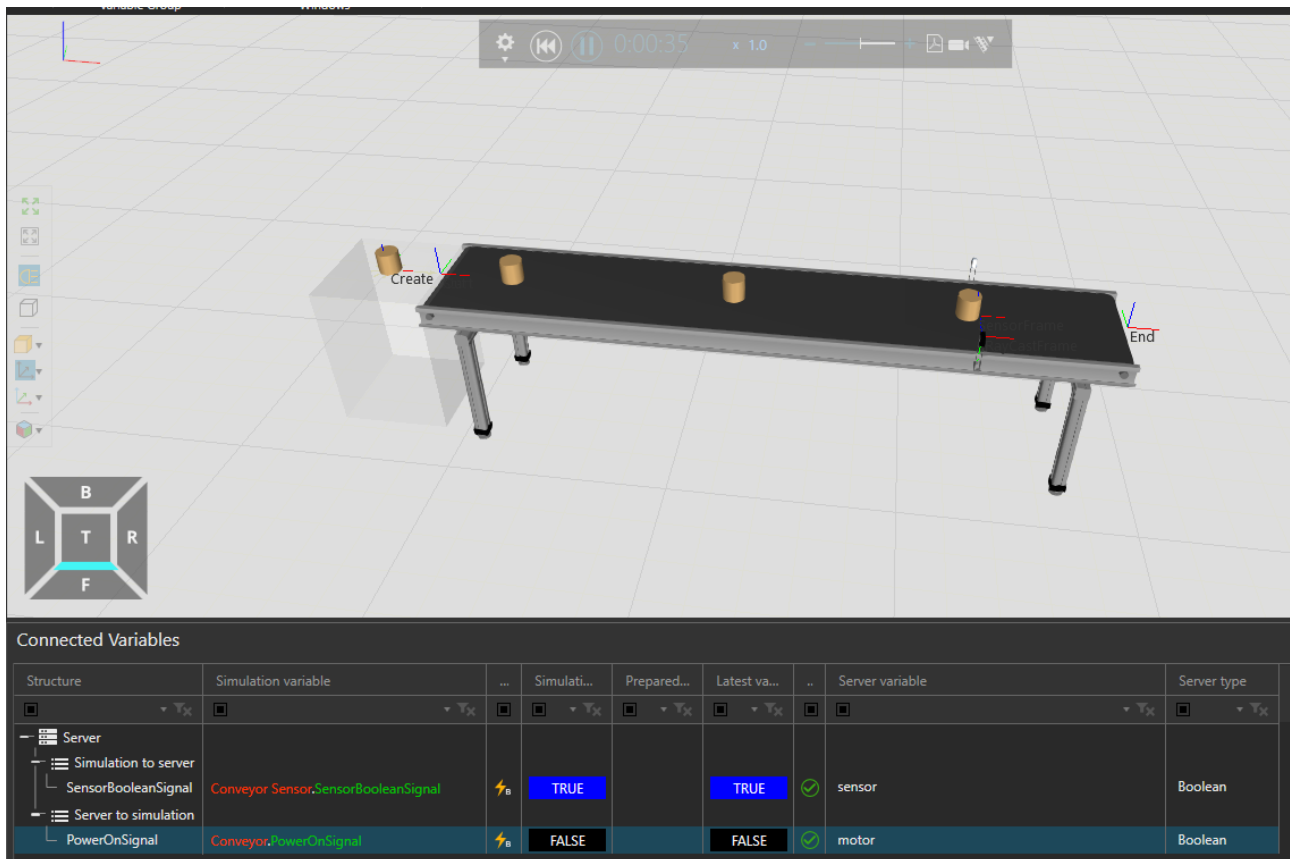
```

1  GVL.bBOOL1 TRUE := GVL.bBOOL2 TRUE ;
2
3  counter(CV TRUE := GVL.sensor TRUE, PV 5 := 5, RESET FALSE := btimer.Q FALSE );
4
5  wword 1 := counter.CV 1 ;
6  GVL.iINT1 1 := WORD_TO_INT(wword 1) ;
7
8
9
10 IF counter.Q FALSE THEN
11     GVL.motor TRUE := FALSE;
12
13
14 ELSE
15     GVL.motor TRUE := TRUE;
16
17 END_IF
18
19 btimer(IN FALSE := counter.Q FALSE, PT T#5s := T#5S);
20
21
22
23
24
25 RETURN

```

Kuvio 29 Tekemäni TwinCAT-ohjelma.

Toimintatutkimuksessa arvioidaan kokeilua. Kokeilu onnistui hyvin. Kuvio 30 näyttää onnistuneen testin, jossa palikka pysähtyi sensorin kohdalle.

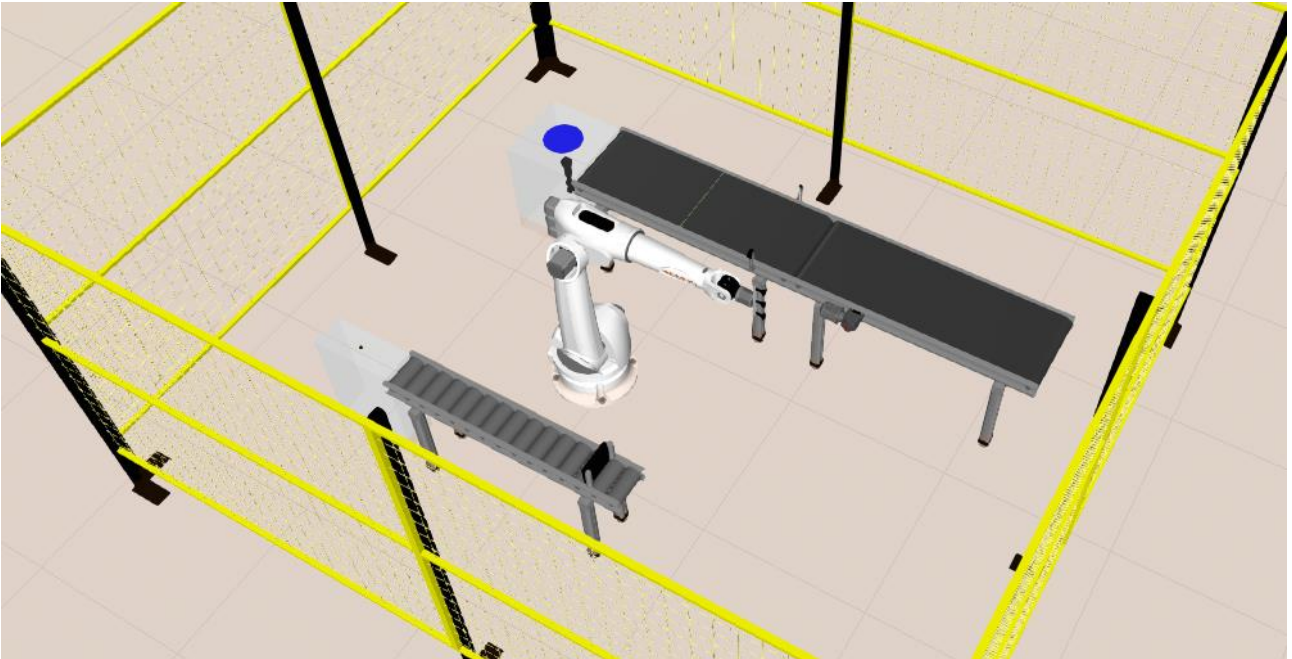


Kuvio 30 Onnistunut testi Visual Componentsissa.

Esitin tulokset toimeksiantajalle. Keskustelun pohjalta sain uuden tavoitteen, johon pyrkiä. Tavoitteena oli seuraavaksi tehdä Pick and place -robottisolu Visual Componentsilla. Sovellusta ohjattaisiin HMI:llä ja se voitaisiin pysäyttää HMI:n kautta. HMI:n kautta voitaisiin myös seurata kuinka monta kertaa robotti nostaa tavaraa.

3.5 Pick and place -robottisolu

Aloitin toisen kokeilun tekemällä Pick and place -sovelluksen Visual Componentsilla. Sen tekemiseen oli hyvät ohjeet Visual Components Academyllä (Grasp and Release Multiple Objects, n.d.), joita noudattamalla sai hyvän käsityksen Pick and place -robottisolu (Kuvio 31).



Kuvio 31 Tekemäni Pick and place -robottisolu.

Seuraavaksi piti miettiä, minkälaisia muuttujia ohjelmaan pitäisi saada. Päädyin kuvion 32 mukaiseen muuttujalistaan.

```

1  {attribute 'qualified_only'}
2  VAR GLOBAL
3
4      {attribute 'OPC.UA.DA' := '1'}
5      Product_Ready : BOOL;
6
7      {attribute 'OPC.UA.DA' := '1'}
8      Reset_counter : BOOL;
9
10     {attribute 'OPC.UA.DA' := '1'}
11     Start_Stop : BOOL;
12
13     {attribute 'OPC.UA.DA' := '1'}
14     Finished_Products : WORD;
15
16     {attribute 'OPC.UA.DA' := '1'}
17     Batchsize : INT;
18
19  END_VAR

```

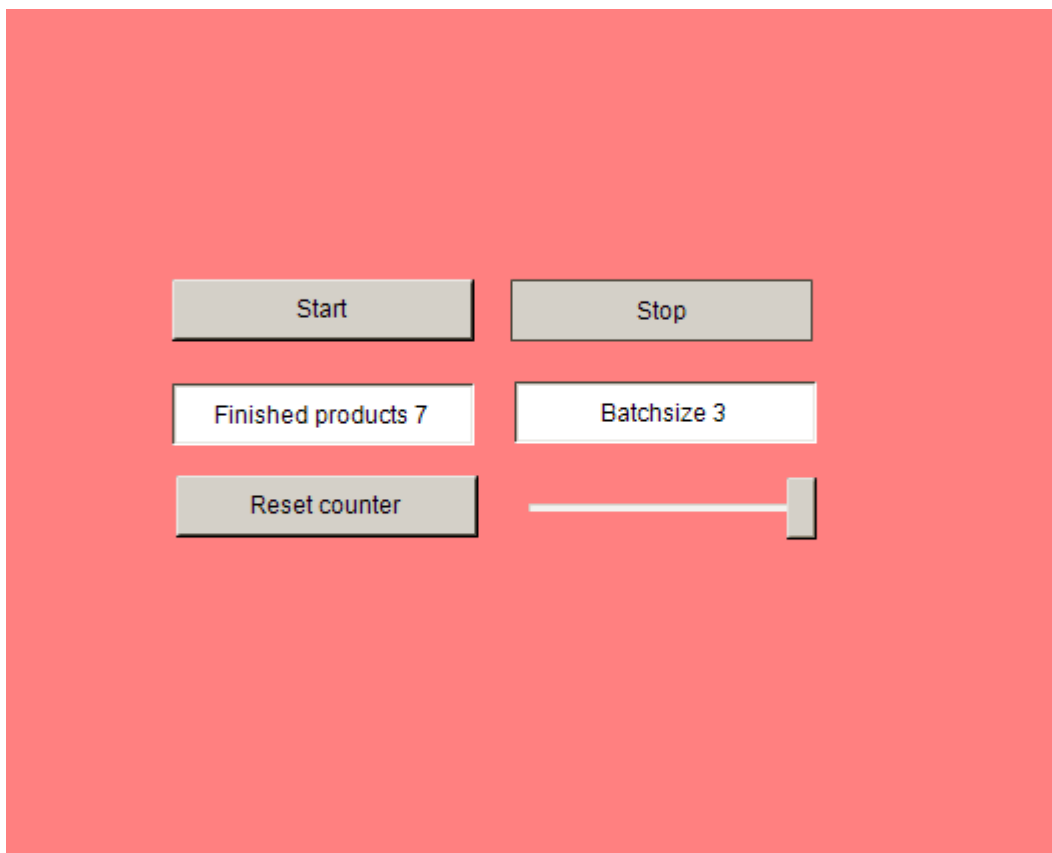
Kuvio 32 Tekemäni Pick and place -robottisolun muuttujat.

Tuote valmis -muuttuja (Product_Ready) kertoo, kun tuote on lisätty laatikkoon. Kuviossa 33 on TwinCAT-ohjelma, joka lisää arvon valmiit tuotteet -muuttujaan (Finished_Products), kun tuote valmis -muuttuja saa arvon "true".

```
1 counter(CV TRUE := OPCUA.Product_Ready TRUE, RESET FALSE := OPCUA.Reset_counter FALSE);  
2  
3 OPCUA.Finished_Products 7 := counter.CV 7;  
4  
5  
6  
7
```

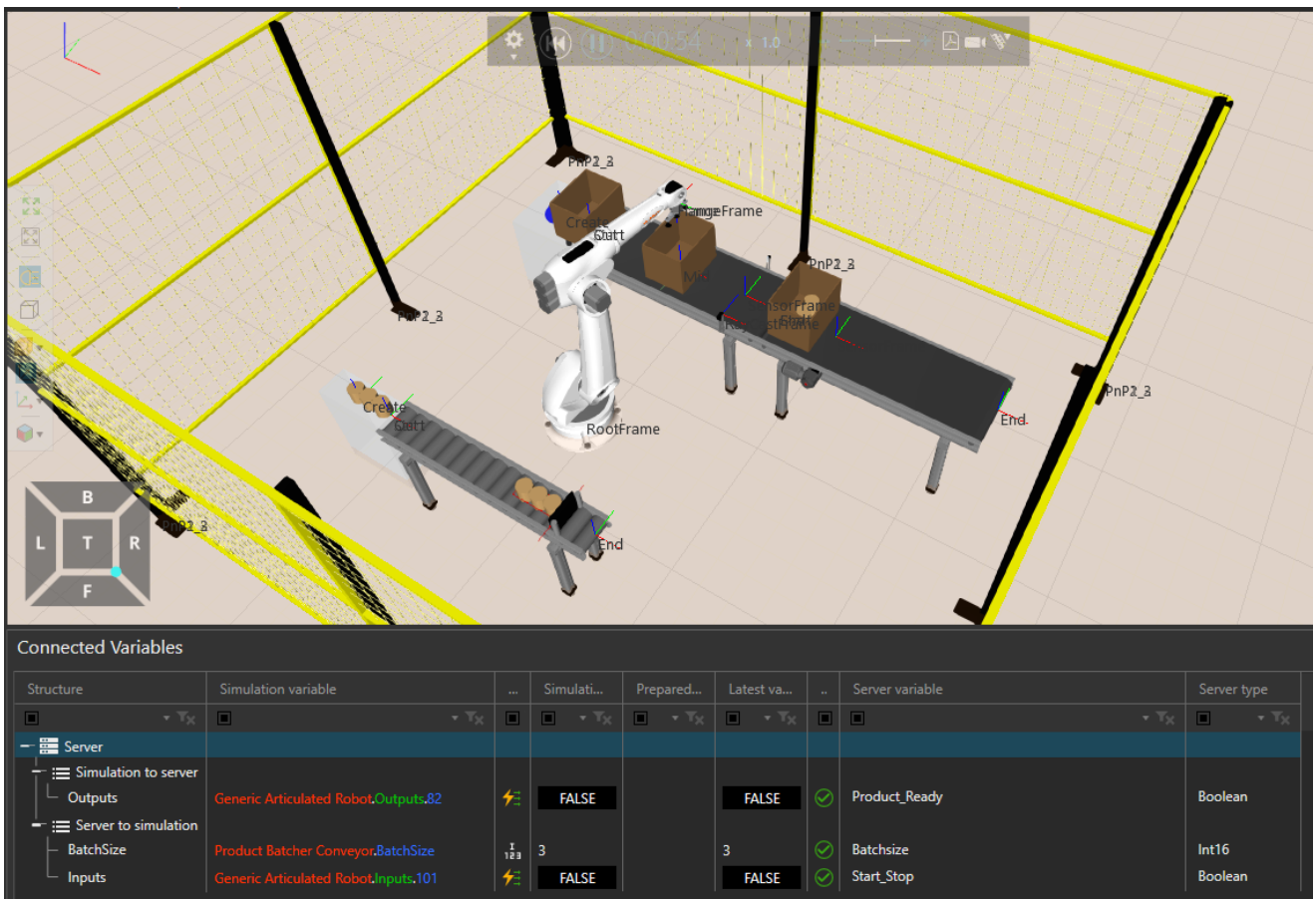
Kuvio 33 Pick and place -robottisolun TwinCAT-ohjelma.

Viimeisenä tein ohjelmalle käyttöliittymän (Kuvio 34), jolla voidaan ohjata ohjelmaa. Lisäsin Start- ja Stop-nappien lisäksi mahdollisuuden muokata tuotteen eräkokoja 1-3.



Kuvio 34 HMI-käyttöliittymä.

Seuraavaksi testasin, että kaikki toimii. Kuviossa 35 on valmis ohjelma, jota voidaan ohjata HMI-käyttöliittymällä.



Kuvio 35 Valmis Pick and place -robottisolu OPC UA yhteydellä.

3.6 OPC UA:n Ohjeistaminen

Työn toteutuksessa päädyttiin käyttämään Visual Componentsin kanssa TwinCAT:ä, jolla sai tehtyä OPC UA -palvelimen. Yksi tavoitteistani oli ohjeistaa OPC UA -rajapinnan käyttämistä. Sitä varten tein ohjeet, kuinka alustaa OPC UA -palvelin TwinCAT:ssä ja UaExpertillä ja kuinka yhdistää Visual Components palvelimeen. Ohje löytyy liitteestä 1.

Toimeksiantaja kokeili alustaa palvelinta ohjeideni avulla, mutta hänelle tuli ongelma. Hän ei voinut valita OPC UA -palvelinta. Ongelma oli itselleni uusi, joten päätin kokeilla itsekin ja huomasin, että oma palvelimeni oli mennyt jotenkin jumiin enkä pystynyt enää yhdistämään palvelimeen. Ongelmaa pähkäillessäni kokeilin poistaa palvelinsovelluksen koneelta ja ladata sen

uudestaan. Kokeilin myös tehdä kokonaan uuden TwinCAT-projektin. Kumpikaan ei onnistunut. Lopulta kokeilin poistaa palvelinsovelluksen ja kaikki siihen liittyvät tiedostot tiedostopolusta C: -> Twincat -> Functions -> TF6100 OPC UA, jotka eivät poistu automaattisesti. Latasin palvelimen ja konfiguraation uudestaan. Tämän jälkeen palvelimen alustus voidaan tehdä joko suoraan TwinCAT:lla tai UaExpertillä. Itse suosin UaExpertiä. Sitä käytetään myös Beckoffin OPC UA -manuaalissa, mikä sai minut vakuttuneeksi sen kannattavuudesta (Manual TwinCAT 3 | OPC UA Server, 2024, 22). Näin ongelmani ratkesi ja sen jälkeen pystyin käyttämään palvelinta normaaliin tapaan. Lähetin myös toimeksiantajalleni ohjeet, kuinka palvelimen voi alustaa UaExpertillä, mutta sekään ei poistanut toimeksiantajan ongelmaa.

Myöhemmin toimeksiantaja sai uuden tietokoneen, jonka kautta hän kokeili uudestaan alustaa palvelinta, mutta nyt ilmeni uusi ongelma alustuksen kanssa. Alustus pyysi normalisti käyttäjätunnusta ja salasanaa, mutta nyt ne eivät menneet läpi, joten alustus jumittui aina siihen. Etsin tietoa tähän ongelmaan ja sain selville, että kun alustuksen tekee ensimmäistä kertaa, niin palvelin luo tiedostopolkuun C: -> TwinCAT -> Functions -> TF6100-OPC-UA -> Win32 -> Server tiedoston TcUaSecurityConfig.xml, joka määrittelee profiilin, jonka avulla voidaan kirjautua palvelimelle. Esittelin tämän toimeksiantajalle ja huomasimme, että tiedosto luodaan vasta, kun alustus on tehty. Tämä ei auttanut löytämään ratkaisua ongelmaan, joten päädyimme siihen tulokseen, että ongelma voi myös johtua koulun tietokoneesta syystä, jota emme tiedä.

4 Johtopäätökset

Opinnäytetyössäni tavoitteenani oli tutkia OPC UA -rajapintaa ja kehittää materiaalia koulutukseen. Työssäni pääsin tutustumaan OPC UA -rajapinnan toimintaan ja eri yritysten omiin tuotteisiin. Pääsin myös kokeilemaan itse yhteyden tekemistä. Nämä yhdessä auttoivat minua ymmärtämään OPC UA -rajapintaa, mikä auttoi minua koulutusmateriaalin teossa.

Työtä tehdessä opin, että OPC UA -kommunikointi tapahtuu kahdella tavalla: joko asiakas-palvelin- tai julkaisija-tilaaja-mallilla. Päädyin tutkimaan OPC UA:n asiakas-palvelin-mallia, joka oli työn kannalta oleellisempi. Tämän takia keskityin OPC UA:n asiakas-palvelin-mallin perusteisiin ja kuinka sen voisi yksinkertaisesti ymmärtää ja opettaa muille. OPC UA on kompleksi rajapinta, johon voisi syventyä paljon enemmänkin erilaisista näkökulmista.

Työssäni sain tehtyä hyvin koulutusmateriaalia, jota varten tein step to step -tyyppisen ohjeen, jota on helppo seurata. Selitän, mitä kussakin vaiheessa tapahtuu. Esittelen myös esimerkkiohjelmat Visual Componentsille ja TwinCAT:lle. Saavutin tavoitteeni opinnäytetyössä, vaikka harmillisesti toimeksiantaja ei onnistunut palvelimen alustuksessa, eikä päässyt testamaan sitä ongelman vuoksi, jota ei saatu selvitettyä.

OPC UA:lla on todella paljon käyttökohteita ja mahdollisuuksia simulaatioympäristöissä. Simulaatioympäristöä voitaisiin paremmin hyödyntää etäohjelmoinnissa ja myös paikan päällä, jos halutaan tehdä muutoksia prosessiin. Tämä voi säästää rahaa, koska koneita ei tarvitsisi pysäyttää muutoksen ajaksi.

Tämä opinnäytetyö on vain pintaraapaisu OPC UA -rajapinnasta ja tästä voisi muun muassa jatkaa simulaation kehittämistä ja tehdä kompleksimpia ohjelmia, jossa voisi hyödyntää rajapinnan ominaisuuksia. Toinen mielenkiintoinen sovellus, jota tästä voisi jatkaa, olisi tehdä mallinnus fyysisestä robottisolusta Visual Componentsiin. Samalla kun fyysinen robotti liikkuisi, liikkuisi se myös Visual Componentsissä ja päinvastoin. Tämä olisi hyödyllistä muun muassa ohjelman testauksessa. Tällaisen sovelluksen avulla voitaisiin myös seurata robotin liikkeitä etänä. Kolmas suunta olisi tehdä etäpalvelin tai palvelin fyysiselle komponentille, jota käytettäisiin fyysisessä ympäristössä, johon asiakas voisi yhdistää internetin kautta. Näin OPC UA:ta voitaisiin hyödyntää etäohjelmoinnissa. Mielenkiintoinen jatkotutkimuksen aihe voisi olla myös julkaisija-tilaaja-malliin tutustuminen ja sen käyttökohteet.

Lähteet

An In-Depth Look at OPC-UA Cyber Threats – Part 3: OPC Server Authentication Traffic. 2023.

Viitattu 13.12.2023 <https://www.txone.com/blog/an-in-depth-look-at-opc-ua-cyber-threats-part-3-opc-server-authentication-traffic/>

Certificate exchange. N.d. Viitattu 6.11.2023

[https://infosys.beckhoff.com/english.php?content=../content/1033/tf6100_tc3_opcua/1466709259.html&id=\)](https://infosys.beckhoff.com/english.php?content=../content/1033/tf6100_tc3_opcua/1466709259.html&id=))

Essentials. N.d. Viitattu 25.9.2023 <https://www.visualcomponents.com/products/essentials/>

Grasp and Release Multiple Objects. N.d. Viitattu 13.12.2023

<https://academy.visualcomponents.com/lessons/grasp-and-release-multiple-objects/>

Introduction to OPC Unified Architecture. Viitattu 26.10.2023

<https://reference.opcfoundation.org/Tightening/v100/docs/4.3>

Mahnke, W. a., Leitner, S. a., & Damm, M. a. 2009. OPC Unified Architecture 1st ed. 2009. Viitattu 5.10.2023. <https://janet.finna.fi>, Ellislibrary.

Manual TwinCAT 3 | OPC UA Server. 2024. Viitattu 10.4.2024

https://download.beckhoff.com/download/document/automation/twincat3/TF6100_TC3_OPc_UA_Server_EN.pdf

OPC UA Overview. N.d. Artikkelel ascolab:in sivuilla. Viitattu 3.11.2023

<https://www.ascolab.com/en/technology-unified-architecture/technology-overview.html>

OPC UA Protocols. N.d. Artikkelel ascolab:in sivuilla. Viitattu 3.11.2023

<https://www.ascolab.com/en/technology-unified-architecture/technology-protocols.html>

OPC UA Security. N.d. Artikkelel ascolab:in sivuilla. Viitattu 3.11.2023

<https://www.ascolab.com/en/technology-unified-architecture/technology-security.html>

OPC UA Specifications. N.d. Artikkelel ascolab:in sivuilla. Viitattu 3.11.2023

<https://www.ascolab.com/en/technology-unified-architecture/technology-specifications.html>

OPC Unified Architecture Part 1: Overview and Concepts. 2022. Viitattu 11.10.2023

<https://opcfoundation.org/developer-tools/documents/view/158>

OPC Unified Architecture Part 2: Security Model. 2022. Viitattu 25.10.2023

<https://opcfoundation.org/developer-tools/documents/view/159>

OPC Unified Architecture Part 3: Address Space Model. 2022. Viitattu 27.10.2023

<https://opcfoundation.org/developer-tools/documents/view/160>

OPC Unified Architecture Part 4: Services. 2021. Viitattu 3.11.2023
<https://opcfoundation.org/developer-tools/documents/view/161>

OPC Unified Architecture Part 6: Mappings. 2022. Viitattu 2.11.2023
<https://opcfoundation.org/developer-tools/documents/view/163>

OPC Unified Architecture Part 8: Data Access. 2021. Viitattu 26.10.2023
<https://opcfoundation.org/developer-tools/documents/view/165>

Profile Reporting. N.d. Viitattu 13.12.2023 <https://profiles.opcfoundation.org/>

Prosys OPC. Viitattu 4.10.2023 <https://www.prosysopc.com/opc-ua/>

UA Companion Specifications. N.d. Viitattu 13.12.2023 <https://opcfoundation.org/about/opc-technologies/opc-ua/ua-companion-specifications/>

UaExpert—A Full-Featured OPC UA Client. N.d. Viitattu 3.11.2023 <https://www.unified-automation.com/products/development-tools/uaexpert.html>

Unified Architecture. N.d. Artikkele OPC foundation sivustolla. Viitattu 25.9.2023
<https://opcfoundation.org/about/opc-technologies/opc-ua/>

Uwe Pohlmann, Axel Sikora. Practical Security Guidelines for Building OPC UA Applications. 2018. Viitattu 4.11.2023 <https://opconnect.opcfoundation.org/2018/06/practical-security-guidelines-for-building-opc-ua-applications/>

Uwe Steinkrauss, Overview: OPC Unified Architecture, 2010. Viitattu 26.10.2023
https://www.ascolab.com/images/stories/ascolab/doc/ua_whitepaper_technicaloverview_e.pdf



Liitteet

Liite 1. OPC UA -palvelimen alustus TwinCAT:lla ja UaExpertillä – Palvelimen yhdistäminen Visual Componentsiin

OPC UA -palvelimen alustus TwinCAT:lla ja UaExpertillä – Palvelimen yhdistäminen Visual Componentsiin

Jouko Murtonen

Opinnäytetyön ohjeistus

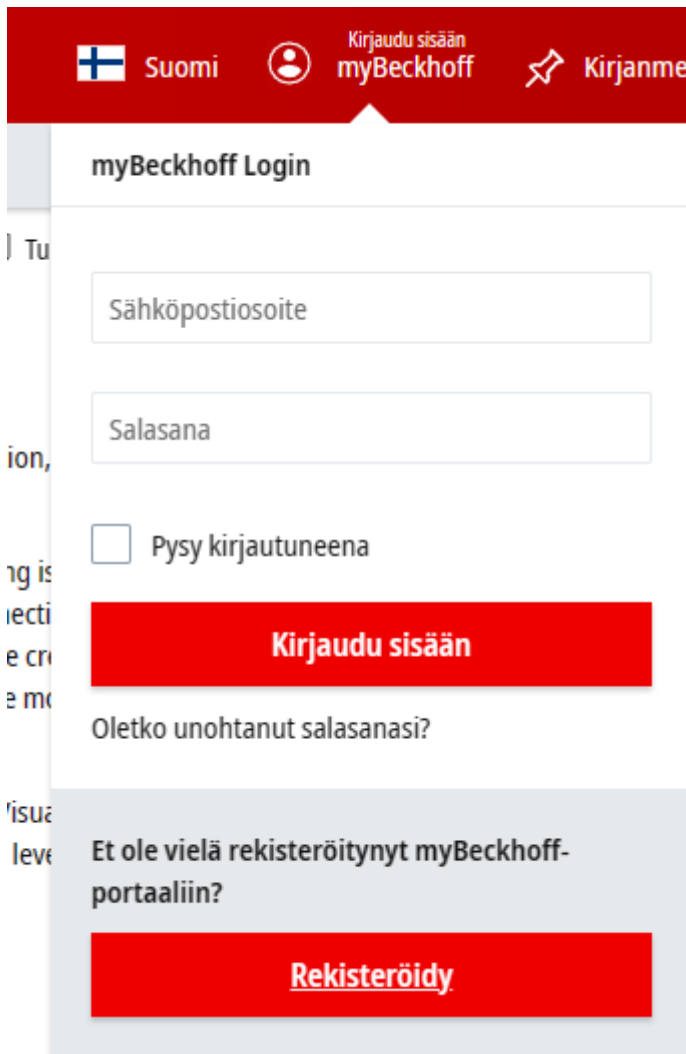
Sähkö- ja automaatiotekniikan tutkinto-ohjelma (AMK)

Sisältö

1	OPC UA -palvelimen alustaminen.....	40
2	TwinCAT-projekti.....	43
3	Palvelimen alustus TwinCAT:llä.....	62
4	Palvelimen alustus UaExpertillä.....	67
5	Data Accessin lisäys.....	77
6	Testaus TwinCAT:n ja UaExpertin välillä.....	83
7	Visual Components -mallinnuksen teko ja yhdistäminen OPC UA -palvelimeen.....	87
8	OPC UA -yhteyden testaus.....	93

1 OPC UA -palvelimen alustaminen

Ensiksi ladataan TwinCat 3. Tästä linkistä pääsee Beckhoff:in sivuille <https://www.beckhoff.com/fi-fi/products/automation/twincat/texxxx-twincat-3-engineering/te1000.html?>. Ensiksi pitää tehdä tunnukset beckhoff:in sivuille, jotta voi ladata TwinCAT:in.



The screenshot shows the myBeckhoff login and registration interface. At the top, there is a red navigation bar with the Finnish flag, the word "Suomi", a user icon, the text "Kirjaudu sisään myBeckhoff", and a star icon with the text "Kirjanmerkit". Below this is the "myBeckhoff Login" section. It contains two input fields: "Sähköpostiosoite" (Email address) and "Salasana" (Password). There is a checkbox labeled "Pysy kirjautuneena" (Stay logged in). A prominent red button labeled "Kirjaudu sisään" (Log in) is positioned below the fields. Below the button, the text "Oletko unohtanut salasanasi?" (Have you forgotten your password?) is visible. At the bottom, a grey box contains the text "Et ole vielä rekisteröitynyt myBeckhoff-portaaliin?" (You have not yet registered to the myBeckhoff portal?) and a red button labeled "Rekisteröidy" (Register).

Yläkulmassa on kirjaudu sisään painike, josta voi rekisteröityä.

Product information

Technical data Ordering information

Documentation and downloads Beckhoff Information System Product news

Additional products

Language File type



+ Information media


× Software and tools

Software and tools

TwinCAT 3 download | eXtended Automation Engineering (XAE)

Login required

Build	Version	
4024.53	3.1	EXE (1.4 GB)  

More information 

Rekisteröitymisen jälkeen samasta linkistä valitaan Documentation and downloads ja sieltä Software tools. Tämän jälkeen voidaan ladata TwinCAT 3 painamalla oikealla olevaa punaista EXE tiedostoa.


× Software and tools

Software and tools

TF6100 | TwinCAT 3 OPC UA Server

Valid for TF6100 Login required

Version	
4.4.70.0	EXE (61 MB)  

More information 

Software and tools

TF6100 | TwinCAT 3 OPC UA Configurator

Valid for TF6100

Version

4.4.138.0

EXE (33 MB)

More information

Login required

Ladataan vielä TF6100 Server, Configurator samalla tavalla tästä linkistä.

<https://www.beckhoff.com/en-en/products/automation/twincat/tfxxx-twincat-3-functions/tfxxx-connectivity/tf6100.html?>

Unified Automation Downloads

On this page you will find latest downloads and documentation. All content is offered free of charge, however if you download or install software from this page you automatically accept the Unified Automation Software License Agreement (SLA). The use of Software and Information is granted as described under "[Evaluation License](#)". All rights reserved by Unified Automation GmbH.

The Software and Information available on this page is for evaluation purpose only. It is provided "AS IS" with no warranty of any kind, including the warranty of design, merchantability and fitness for a particular purpose.

To access the download links you **must register** at the Unified Automation web site and you must activate your account.

By download and/or installation you accept the conditions in the [Software License Agreement](#).

Unified Automation Release PGP Signatures

All releases are cryptographically signed using OpenPGP-compliant signatures. Everyone is strongly encouraged to verify the integrity of downloaded files by verifying the corresponding signatures. This way you can ensure that the file downloaded files have not been corrupted or manipulated.

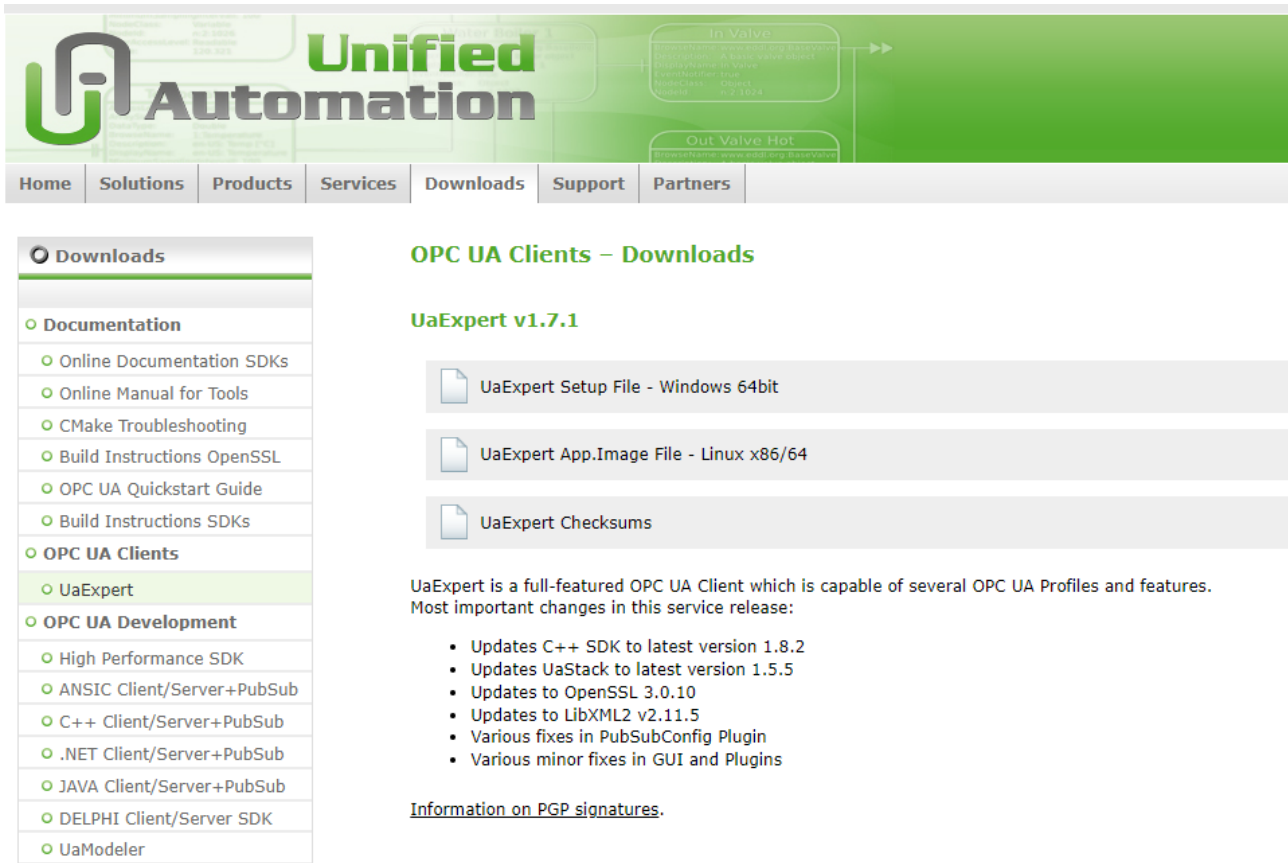
[Information about verifying the PGP signatures of our Website downloads.](#)

Download Categories

- Documentation**
In this section you can find different documentation and technical information.
- OPC UA Clients**
In this section you can find Client products that can be installed on a standard PC (e.g. UaExpert)
- OPC UA Development**
In this section you can find tools and products for developers. Only certain products are available in binary format for evaluation purpose only, versions for other platforms and compilers are available on request.
- OPC UA Servers**
In this section you can find evaluation versions of Server products that can be installed on a standard PC (x86).
- Webinars**
In this section you can find webinar recordings.

Seuraavaksi ladaan UaExpert työkalu. Ensiksi pitää tehdä käyttäjätunnukset jonka jälkeen valitaan Downloads ja vasemmalla UaExpert tai alla olevasta linkistä. UaExpert:tiä voidaan hyödyntää palvelimen alustamisessa ja OPC UA testauksessa.

<https://www.unified-automation.com/downloads/opc-ua-clients.html>



Unified Automation

Home Solutions Products Services Downloads Support Partners

Downloads

- Documentation
 - Online Documentation SDKs
 - Online Manual for Tools
 - CMake Troubleshooting
 - Build Instructions OpenSSL
 - OPC UA Quickstart Guide
 - Build Instructions SDKs
- OPC UA Clients**
 - UaExpert**
- OPC UA Development
 - High Performance SDK
 - ANSIC Client/Server+PubSub
 - C++ Client/Server+PubSub
 - .NET Client/Server+PubSub
 - JAVA Client/Server+PubSub
 - DELPHI Client/Server SDK
 - UaModeler

OPC UA Clients – Downloads

UaExpert v1.7.1

- UaExpert Setup File - Windows 64bit
- UaExpert App.Image File - Linux x86/64
- UaExpert Checksums

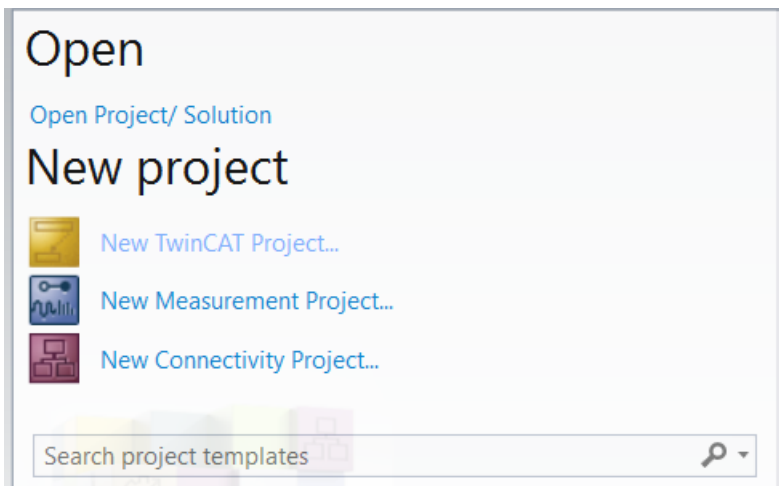
UaExpert is a full-featured OPC UA Client which is capable of several OPC UA Profiles and features. Most important changes in this service release:

- Updates C++ SDK to latest version 1.8.2
- Updates UaStack to latest version 1.5.5
- Updates to OpenSSL 3.0.10
- Updates to LibXML2 v2.11.5
- Various fixes in PubSubConfig Plugin
- Various minor fixes in GUI and Plugins

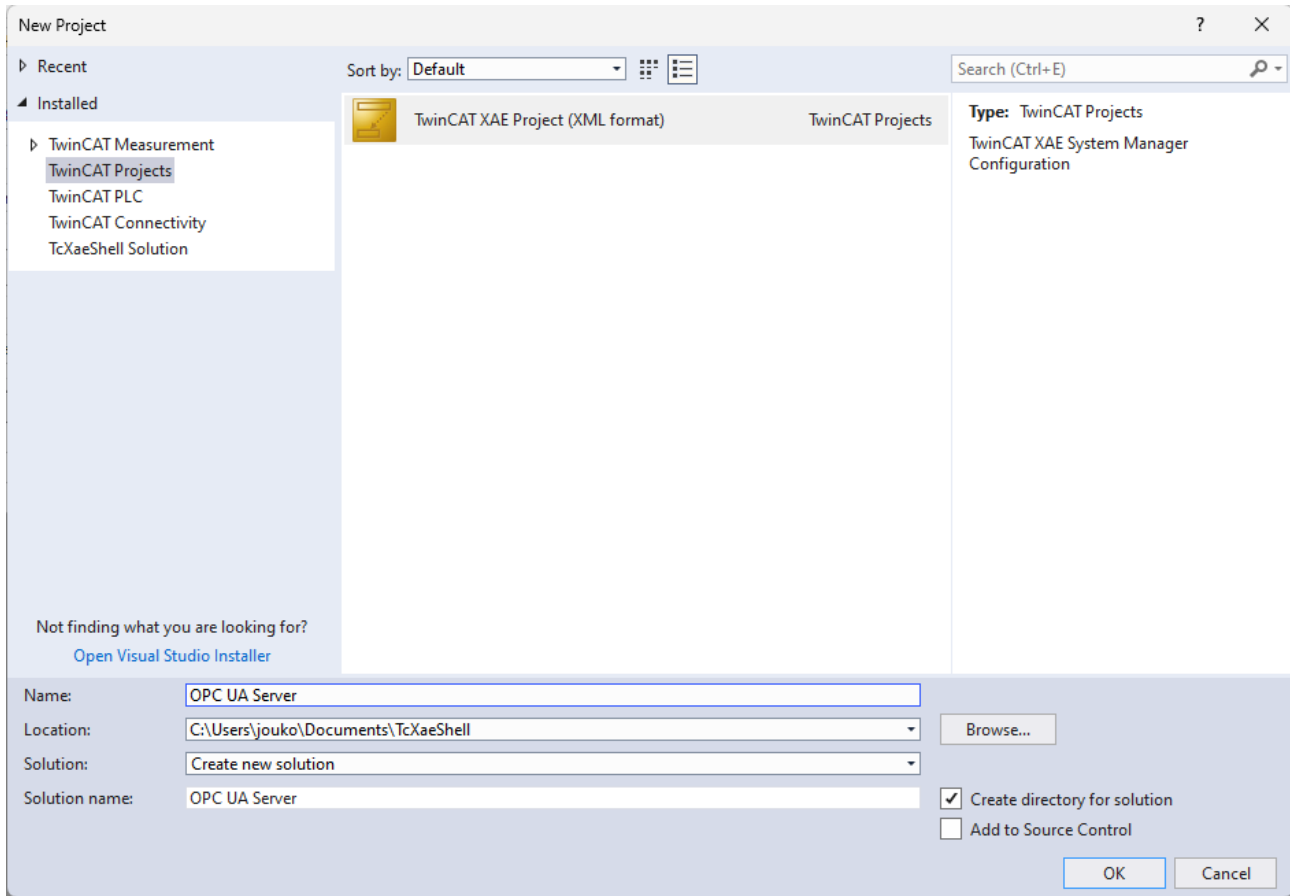
[Information on PGP signatures.](#)

Ladetaan UaExpert Setup File.

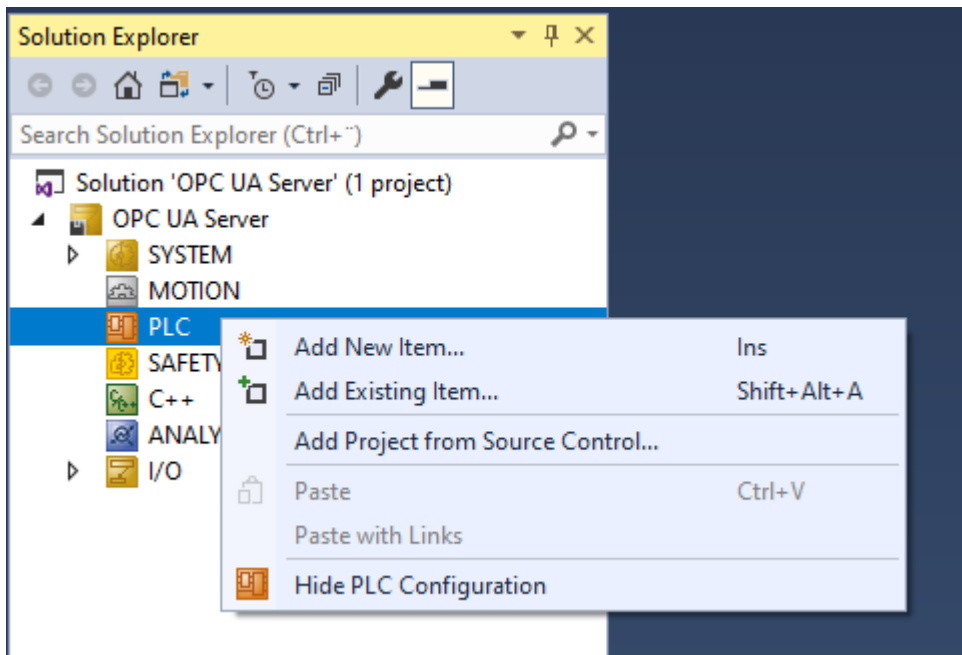
2 TwinCAT-projekti



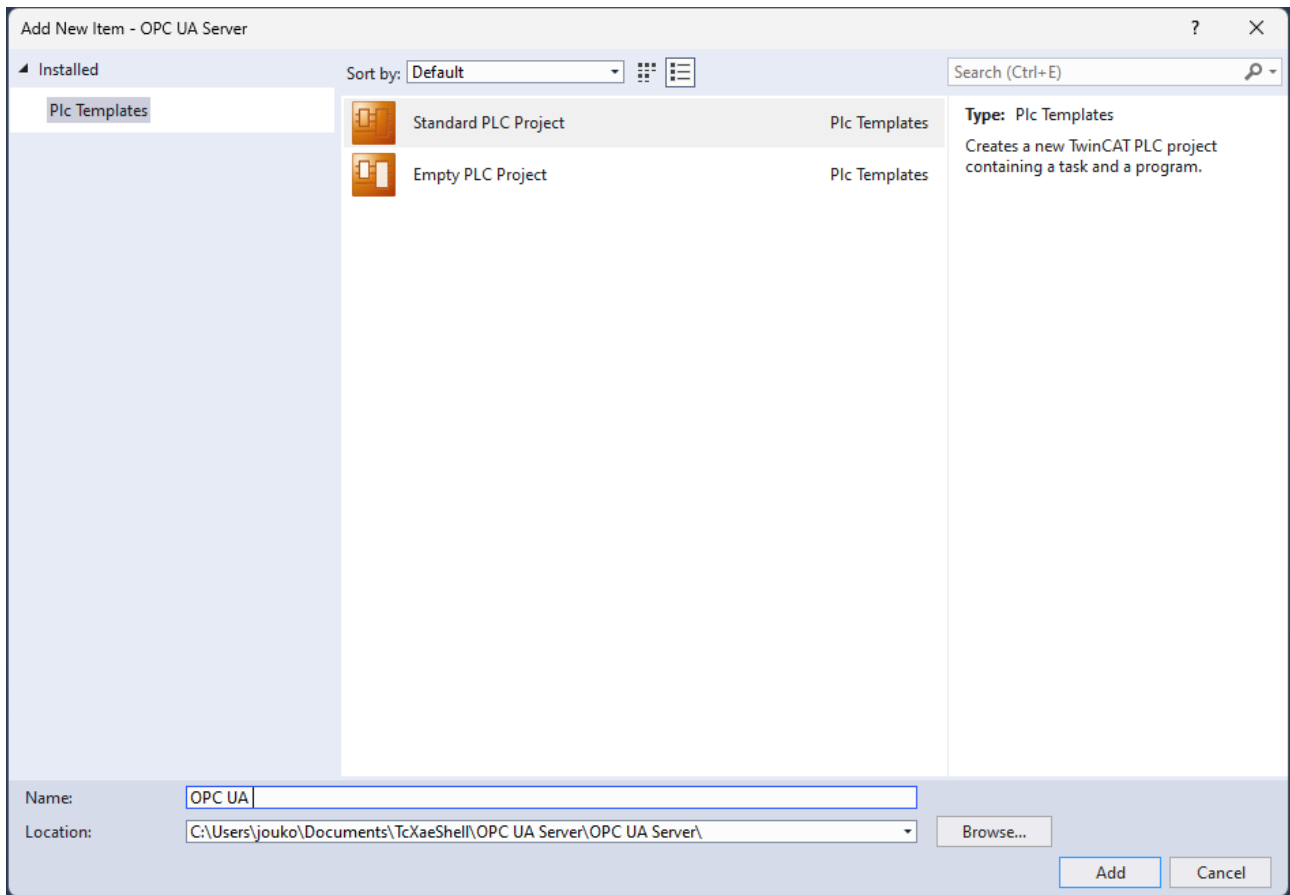
Aloitetaan palvelimen teko avaamalla Twincat Xae shell. Tehdään ensin uusi projekti.



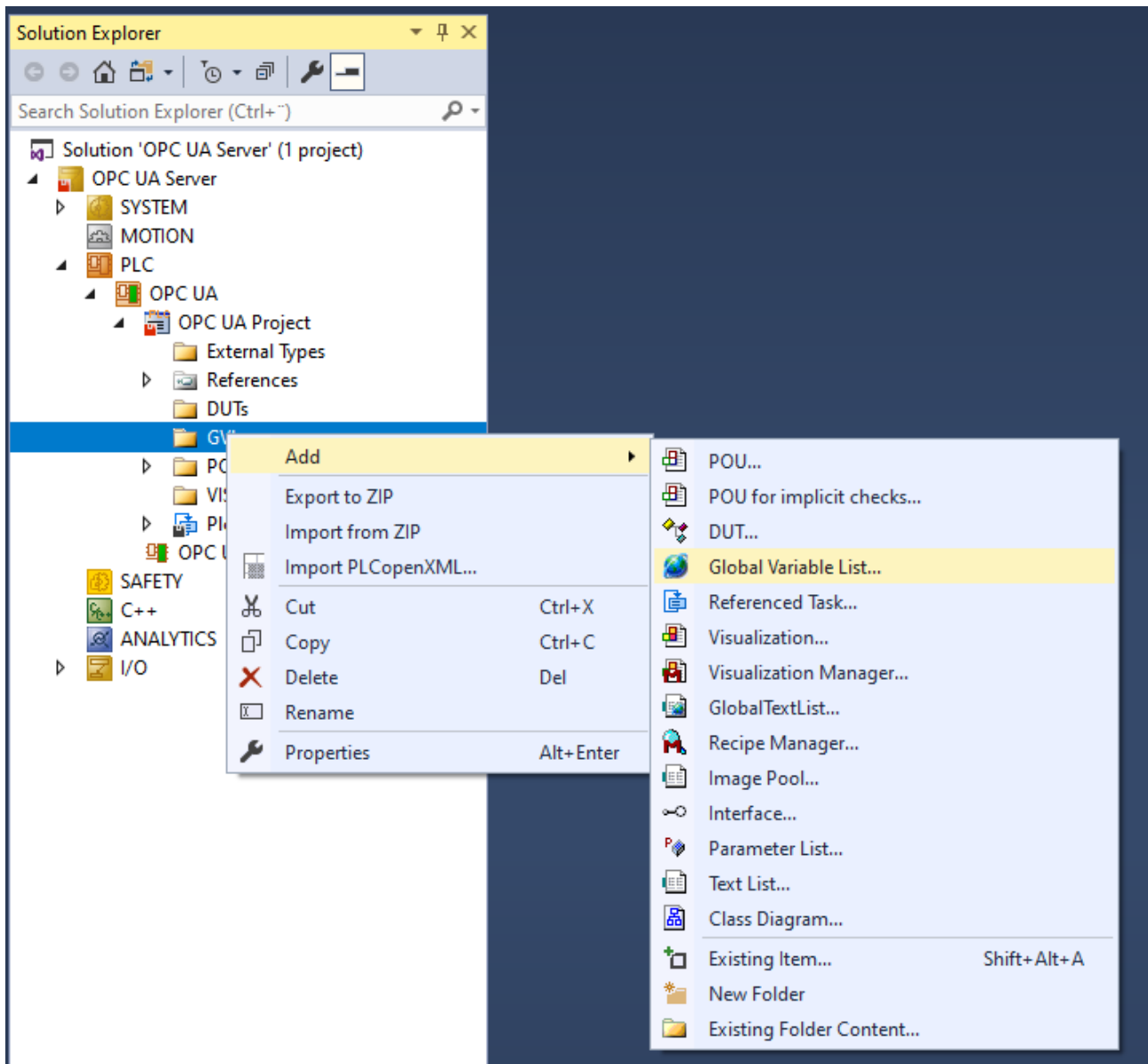
Valitaan TwinCAT XAE projekti ja nimetään projekti, jonka jälkeen painetaan OK.



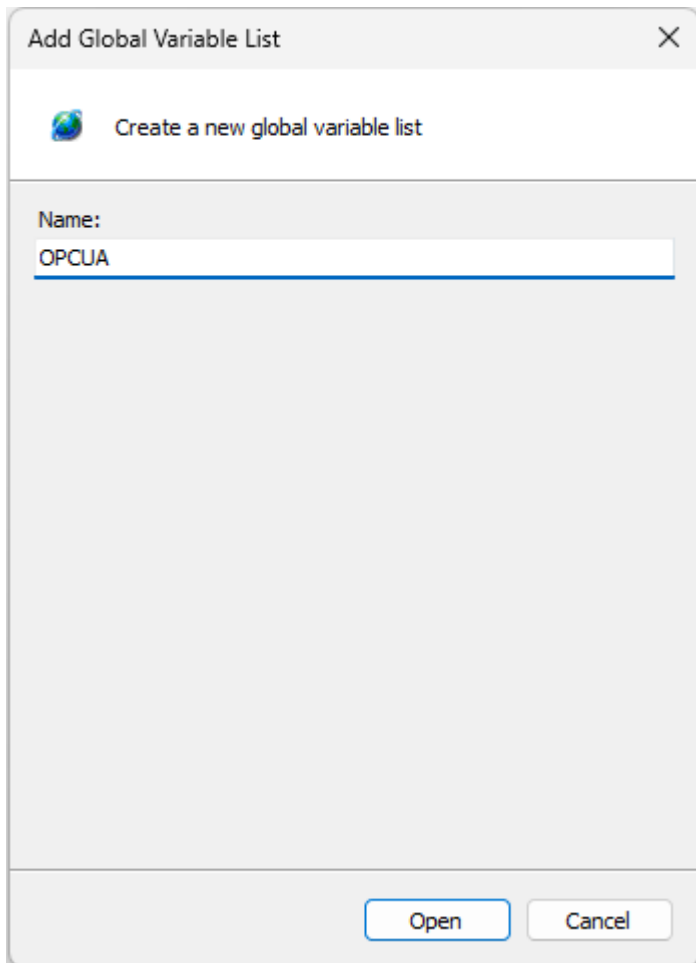
Lisätään uusi PLC painamalla oikealla hiirella PLC ja sieltä Add New Item...



Valitaan Standard PLC Project. Nimetään projekti ja painetaan Add.



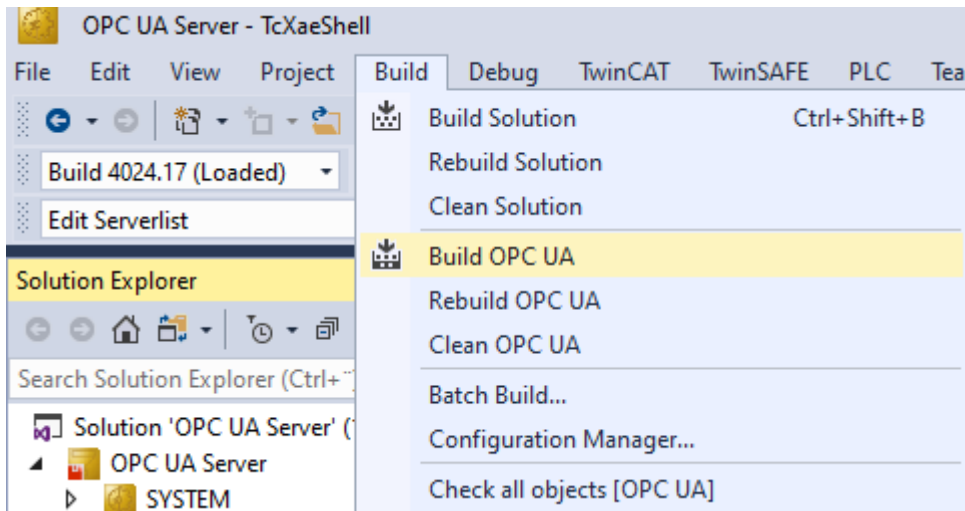
Valitaan GVL oikealla hiirellä ja painetaan Add -> Global Variable List...



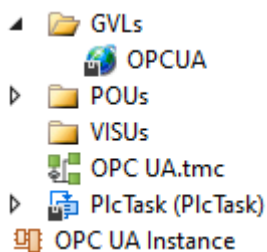
Nimeä global variable list ja paina Open.

```
1 {attribute 'qualified_only'}
2 VAR GLOBAL
3
4 {attribute 'OPC.UA.DA' := '1'}
5 Sensor_Signal: BOOL;
6
7 {attribute 'OPC.UA.DA' := '1'}
8 Motor_ON_OFF: BOOL;
9
10 {attribute 'OPC.UA.DA' := '1'}
11 Counter_up : WORD;
12
13
14 END_VAR
```

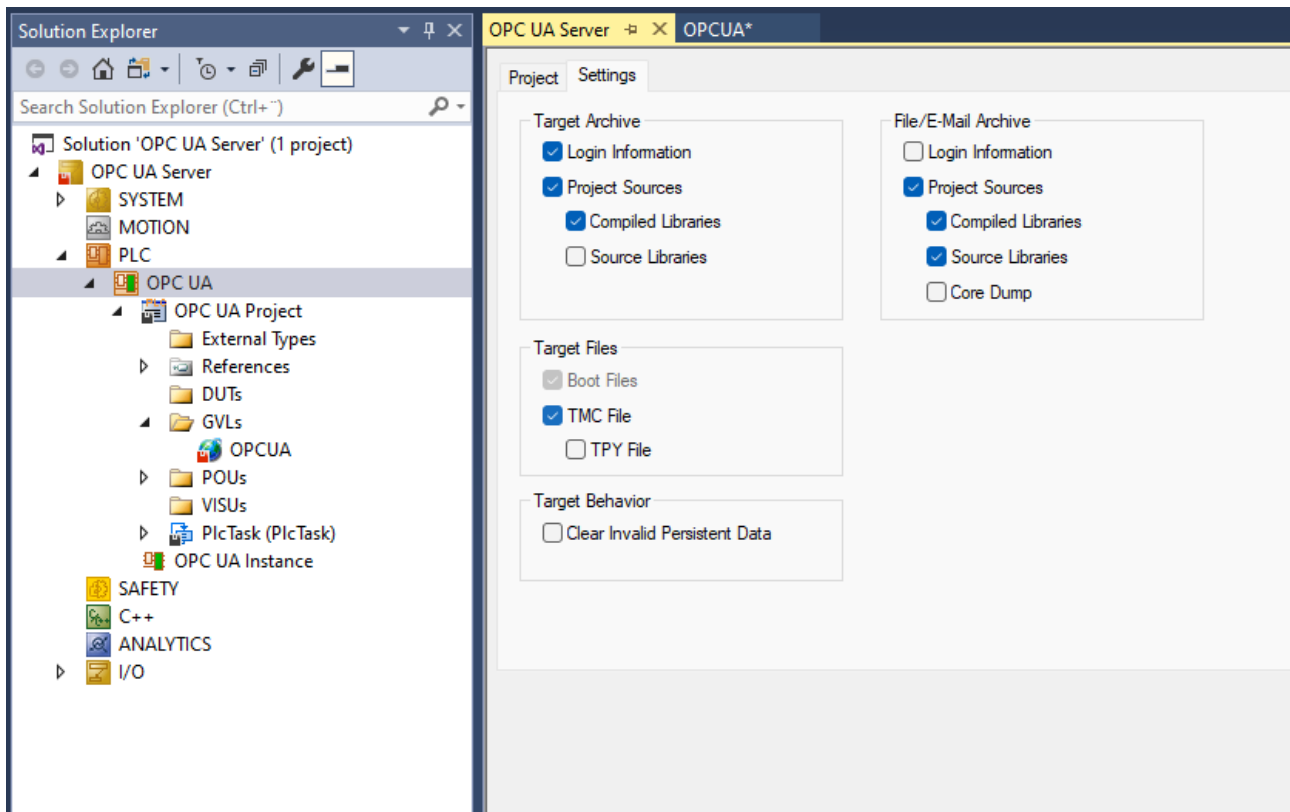
Tehdään GVL muuttujat esim kuvan mukaan. Tärkeintä on muistaa {attribute 'OPC.UA.DA' := '1'} joka mahdollistaa muuttujan pääsyn OPC UA Data Access:siin.



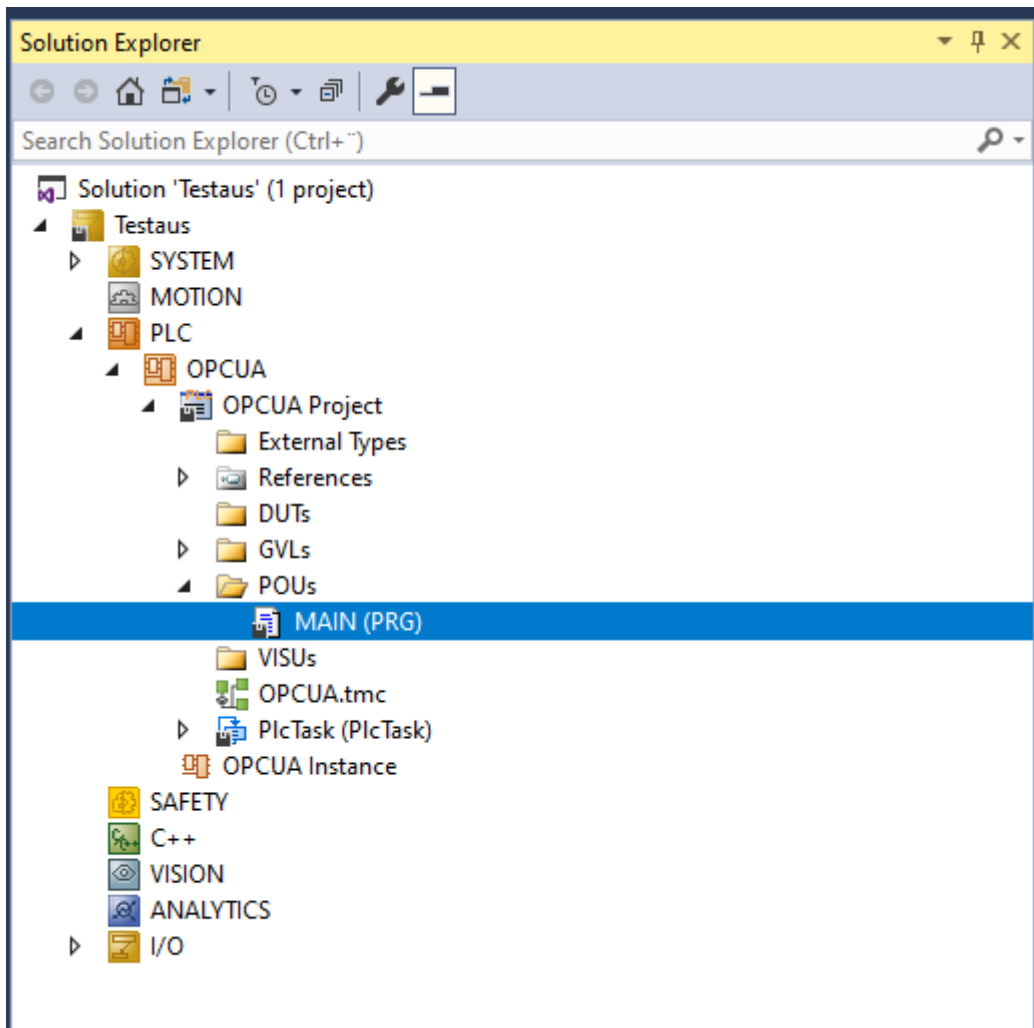
Painetaan yläkulmasta Build. Build toiminto rakentaa tehdyt muutokset ohjelmassa.



Rakentamisen jälkeen pitäisi ilmestyä tmc tiedosto. TMC tiedosto on lyhenne sanasta The trace message. TMC on teksti tiedosto joka sisältää OPC UA- konfiguraation, joka juuri tehtiin GLV.



Tupla klikataan lisättyä PLC, joka löytyy PLC kohdan alta tässä tapauksessa se on OPC UA. Sen jälkeen painetaan Settings ja klikataan TMC File päälle, jotta saamme TMC tiedoston käyttöön.



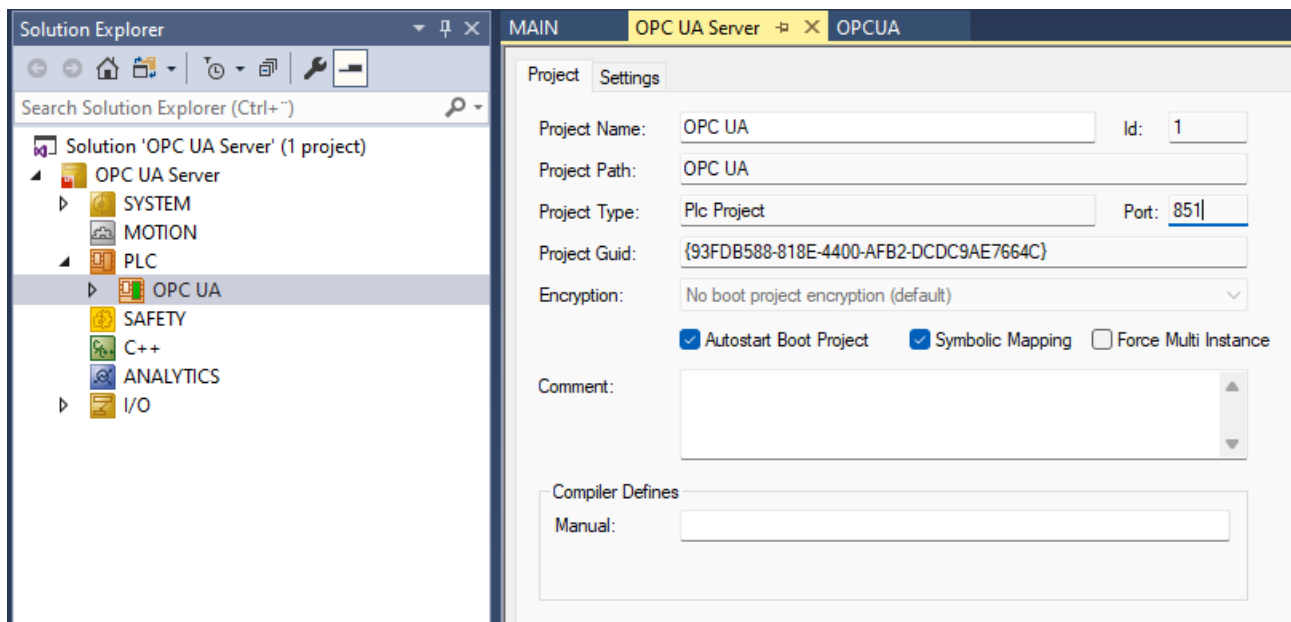
Avataan POUs kansio ja avataan Main(PRG). Main on pääohjelma, jonne voimme tehdä pienen ohjelman.

```

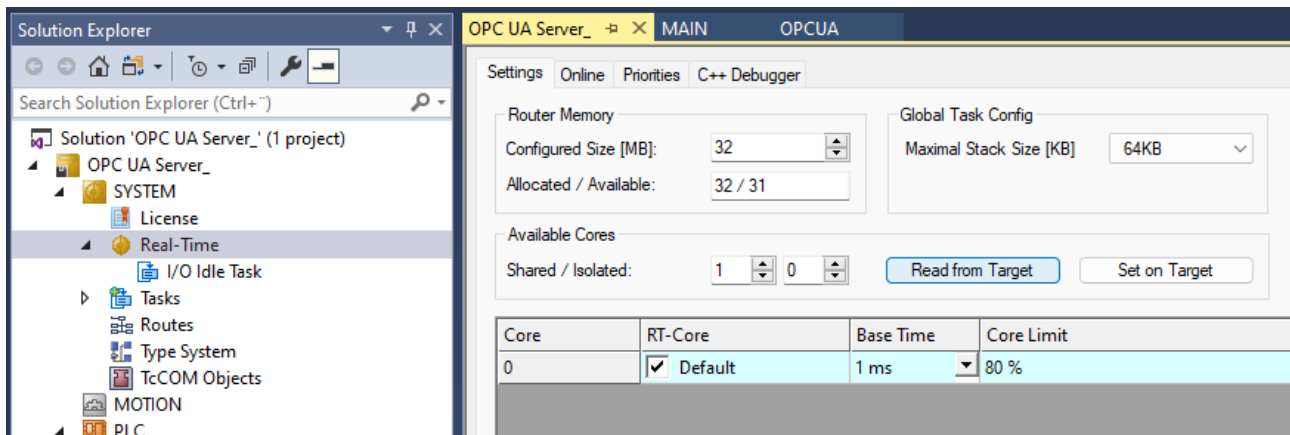
1 PROGRAM MAIN
2 VAR
3     btimer : TON;
4     counter : CTU;
5 END_VAR
6
7
8
9
10
11
12
13
14
15 // Kun 5 kappaletta on mennyt sensorin kohdalle moottori pysähtyy 5 sekunniksi ja aloittaa laskemisen alusta.
counter(CU:= GVL.Sensor_Signal,PV := 5,RESET := btimer.Q );
GVL.Counter_up := counter.CV;
IF counter.Q THEN
    GVL.Motor_ON_OFF := FALSE;
ELSE
    GVL.Motor_ON_OFF := TRUE;
END_IF
btimer(IN:= counter.Q,PT:= T#5S);

```

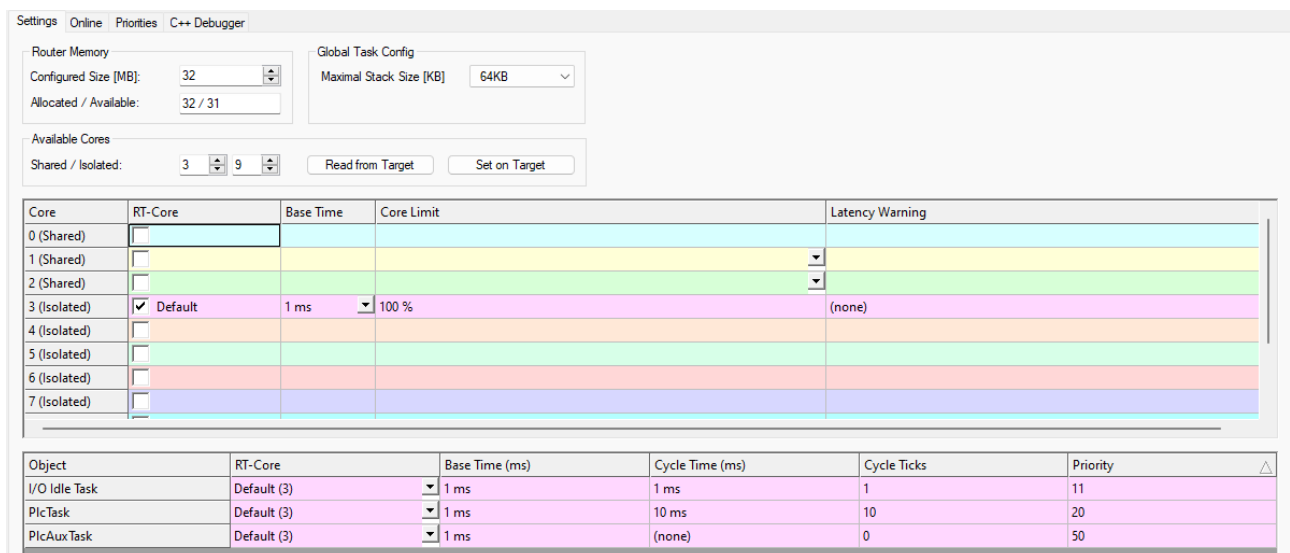
Tehdään MAIN (PRG) yllä on esimerkki ohjelma, jossa on lisätty laskuri.



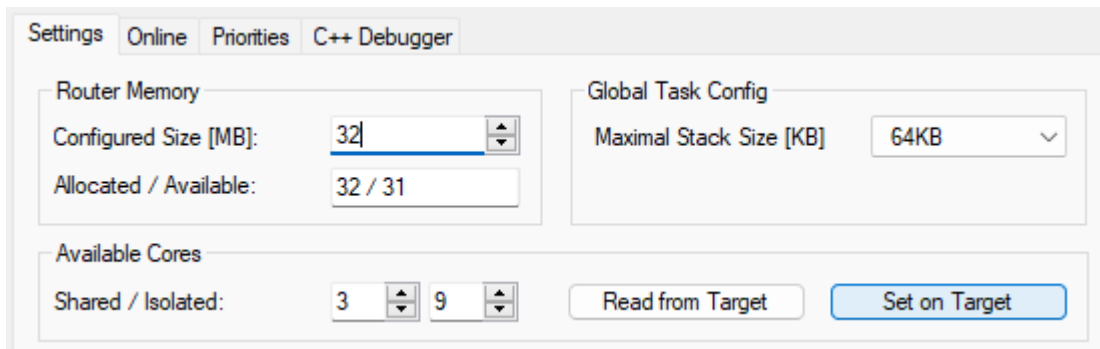
PLC käyttää porttia 851 joten varmistetaan, että portti numero on 851.



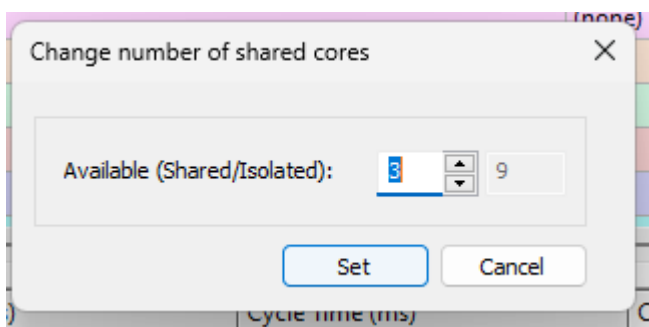
Valitetaan Real-Time ja siellä painetaan nappia Read from Target.



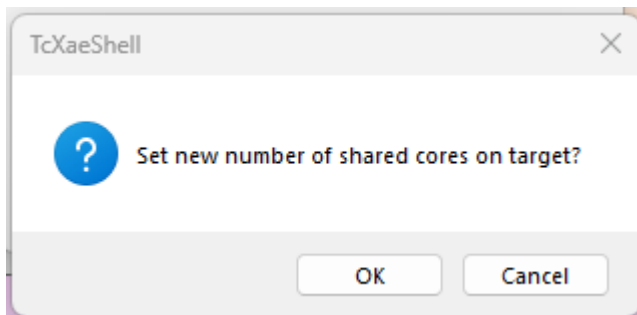
Sen jälkeen pitäisi näkyä tämän kaltainen näkymä ja valitaan Core 1 (isolated) kohta ja valitaan Core 0 pois. Tämä tehdään koska isolated core on avoinna 100% TwinCAT käyttöön ja käyttää 100% tietokoneen tehosta (Core Limit). Shared core on rajoitettu 80%.



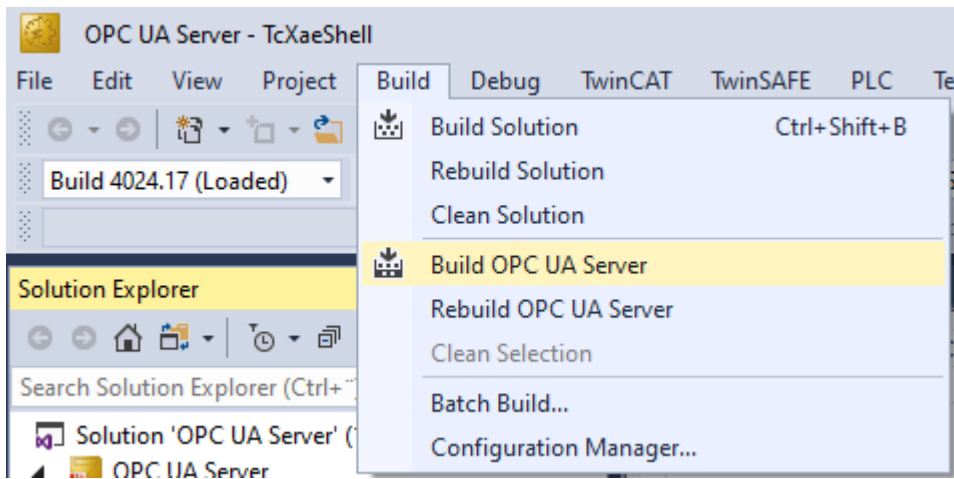
Painetaan Set on Target.



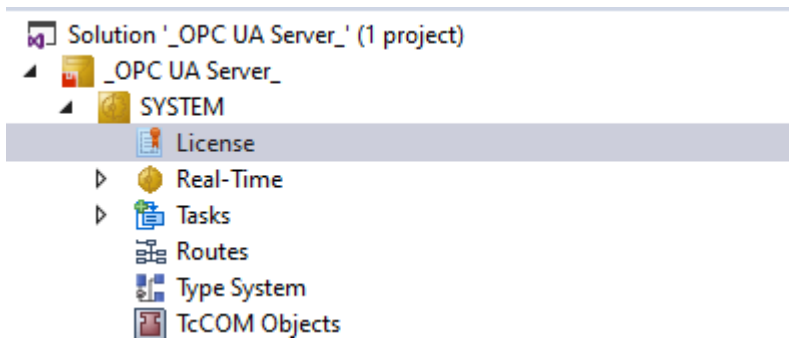
Painetaan Set.



Painetaan OK.



Rakennetaan projekti uudestaan.



Valitaan License.

Order Information (Runtime) Manage Licenses Project Licenses Online Licenses

Disable automatic detection of required licenses for project

Order No	License	Add License
TF5420	TC3 Motion Pick-and-Place	<input type="checkbox"/> cpu license
TF5430	TC3 Planar Motion Base	<input type="checkbox"/> cpu license
TF5440	TC3 Physics Base	<input type="checkbox"/> cpu license
TF5800	TC3 Digital Cam Server	<input type="checkbox"/> cpu license
TF5810	TC3 Hydraulic Positioning	<input type="checkbox"/> cpu license
TF5820	TC3 Line Control	<input type="checkbox"/> cpu license
TF5850	TC3 XTS Technology	<input type="checkbox"/> cpu license
TF5890	TC3 XPlanar Technology	<input type="checkbox"/> cpu license
TF6000	TC3 ADS-Communication-Library	<input type="checkbox"/> cpu license
TF6020	TC3 JSON Data Interface	<input type="checkbox"/> cpu license
TF6100	TC3 OPC-UA	<input checked="" type="checkbox"/> cpu license
TF6105	TC3 OPC UA Pub/Sub	<input type="checkbox"/> cpu license
TF6120	TC3 OPC-DA	<input checked="" type="checkbox"/> cpu license
TF6220	TC3 EtherCAT Redundancy 250	0
TF6221	TC3 EtherCAT Redundancy unlimited	<input type="checkbox"/> cpu license
TF6225	TC3 EtherCAT External Sync	<input type="checkbox"/> cpu license
TF6230	TC3 Parallel Redundancy Protocol	<input type="checkbox"/> cpu license
TF6250	TC3 Modbus-TCP	<input type="checkbox"/> cpu license
TF6255	TC3 Modbus-RTU	<input type="checkbox"/> cpu license
TF6270	TC3 Profinet IO-Device	<input type="checkbox"/> cpu license
TF6271	TC3 Profinet Controller	0
TF6280	TC3 EtherNet/IP Adapter (Slave)	0
TF6281	TC3 EtherNet/IP Scanner (Master)	0

Valitaan ylhäältä Manage Licenses kohta ja lisätään TF6100 ja TF6120.

Order Information (Runtime) Manage Licenses Project Licenses Online Licenses

License Device: Target (Hardware Id) Add...

System Id: 4C5A19DD-71BB-CCBC-5C9B-29ADBAD1F487 Platform: other (91)

License Request

Provider: Beckhoff Automation Generate File...

License Id: Customer Id:

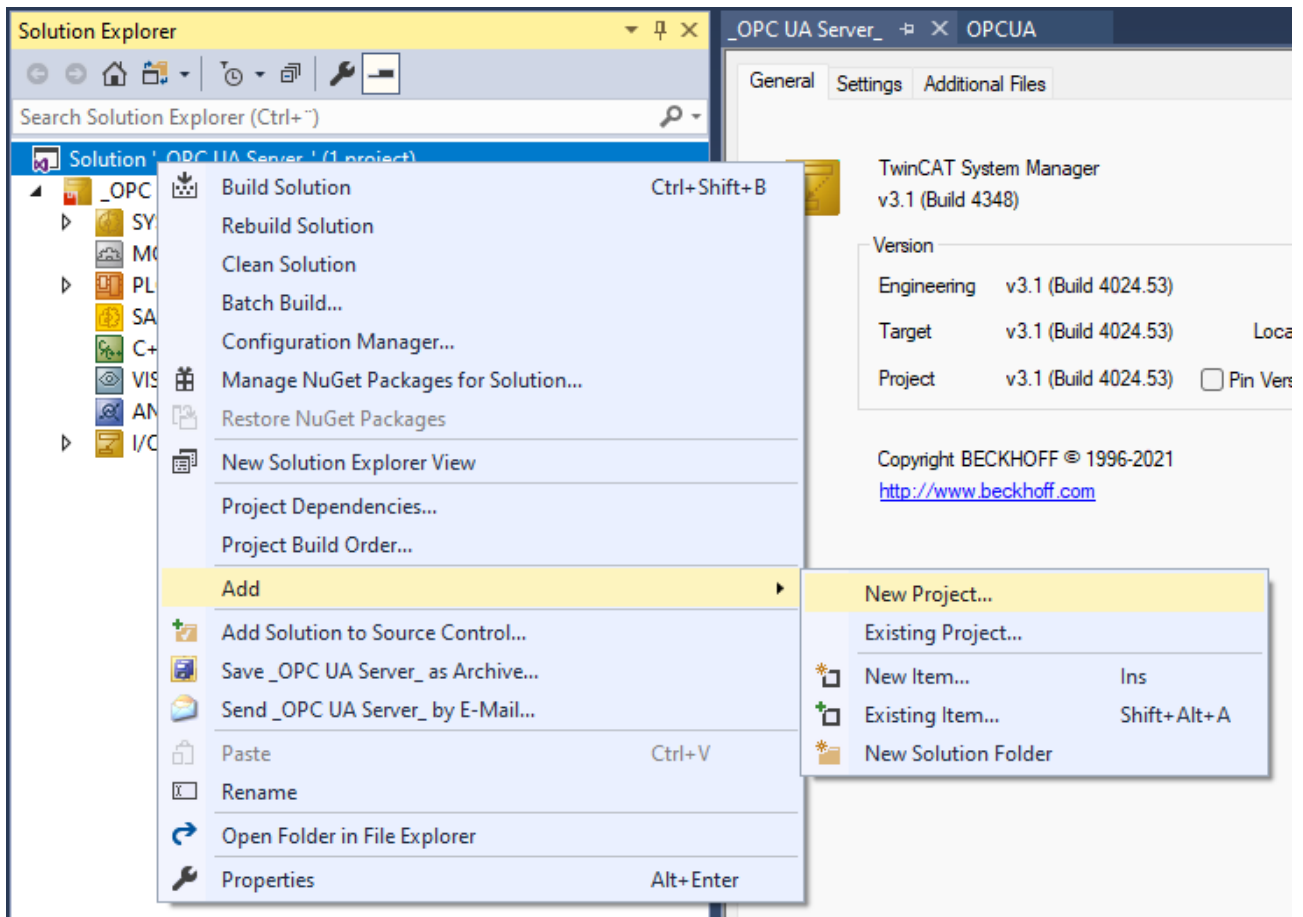
Comment:

License Activation

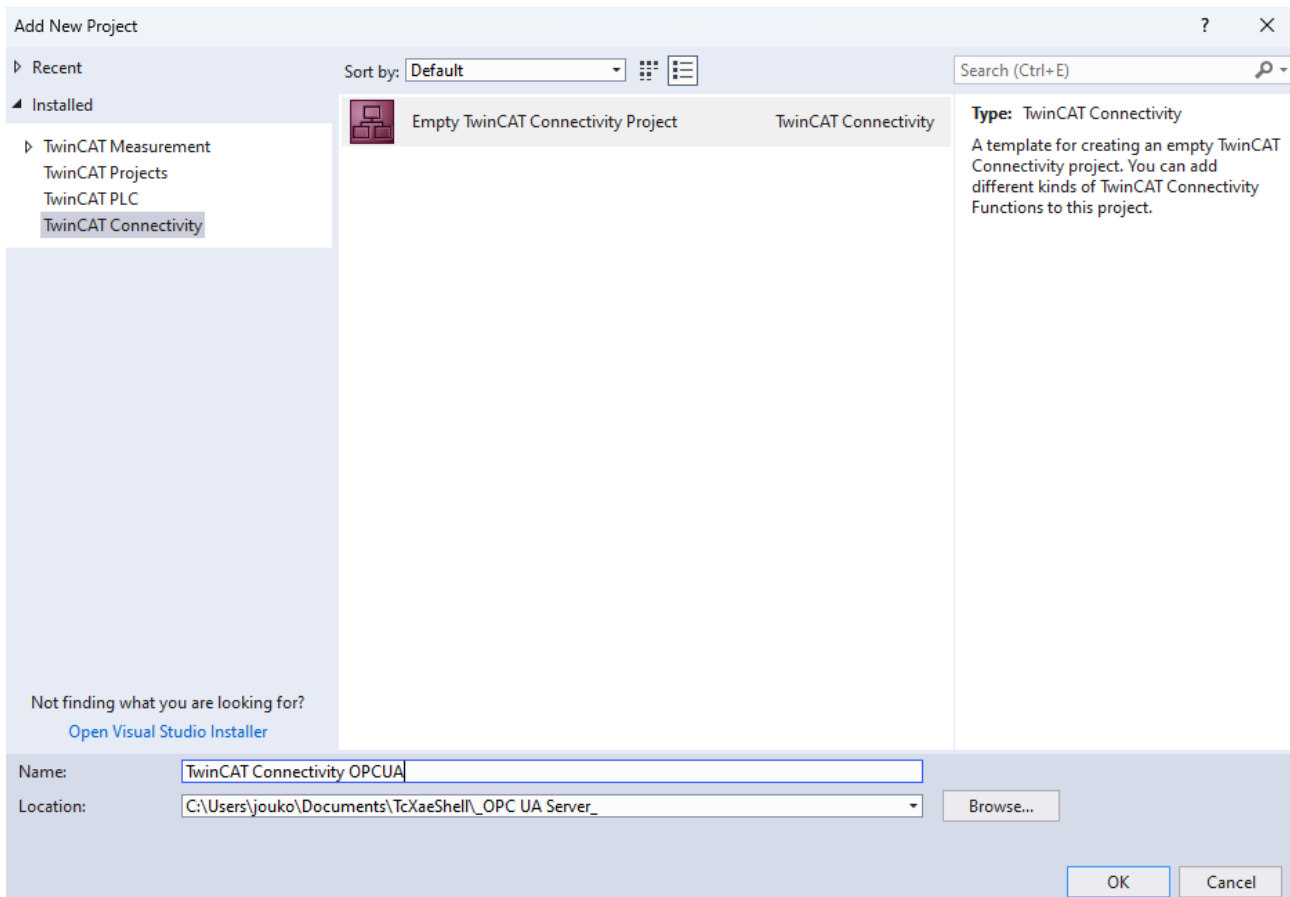
7 Days Trial License... License Response File...

Order No	License	Instances	License TAN	Current Status
TC1200	TC3 PLC	cpu license		expires on Nov 2, 2023 (tria...
TF6100	TC3 OPC-UA	cpu license		expires on Nov 2, 2023 (tria...
TF6120	TC3 OPC-DA	cpu license		expires on Nov 2, 2023 (tria...

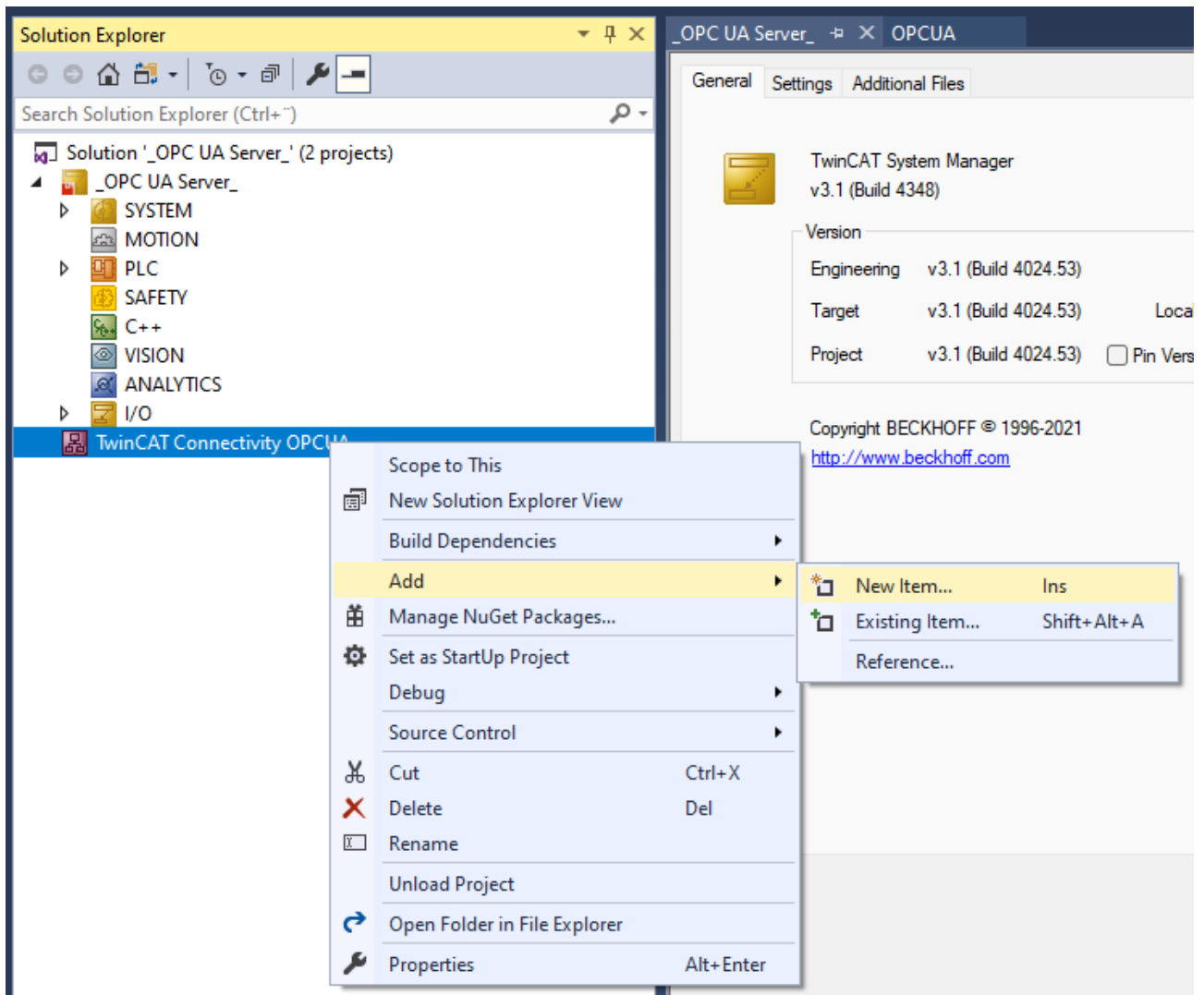
Aktivoidaan 7 päivän lisenssi jonka voi uusia loputtomasti. Tämän jälkeen kannattaa tallentaa ohjelma ja käynnistää tietokone uudestaan, jotta varmistetaan että palvelin menee päälle tietokoneen taustalle.



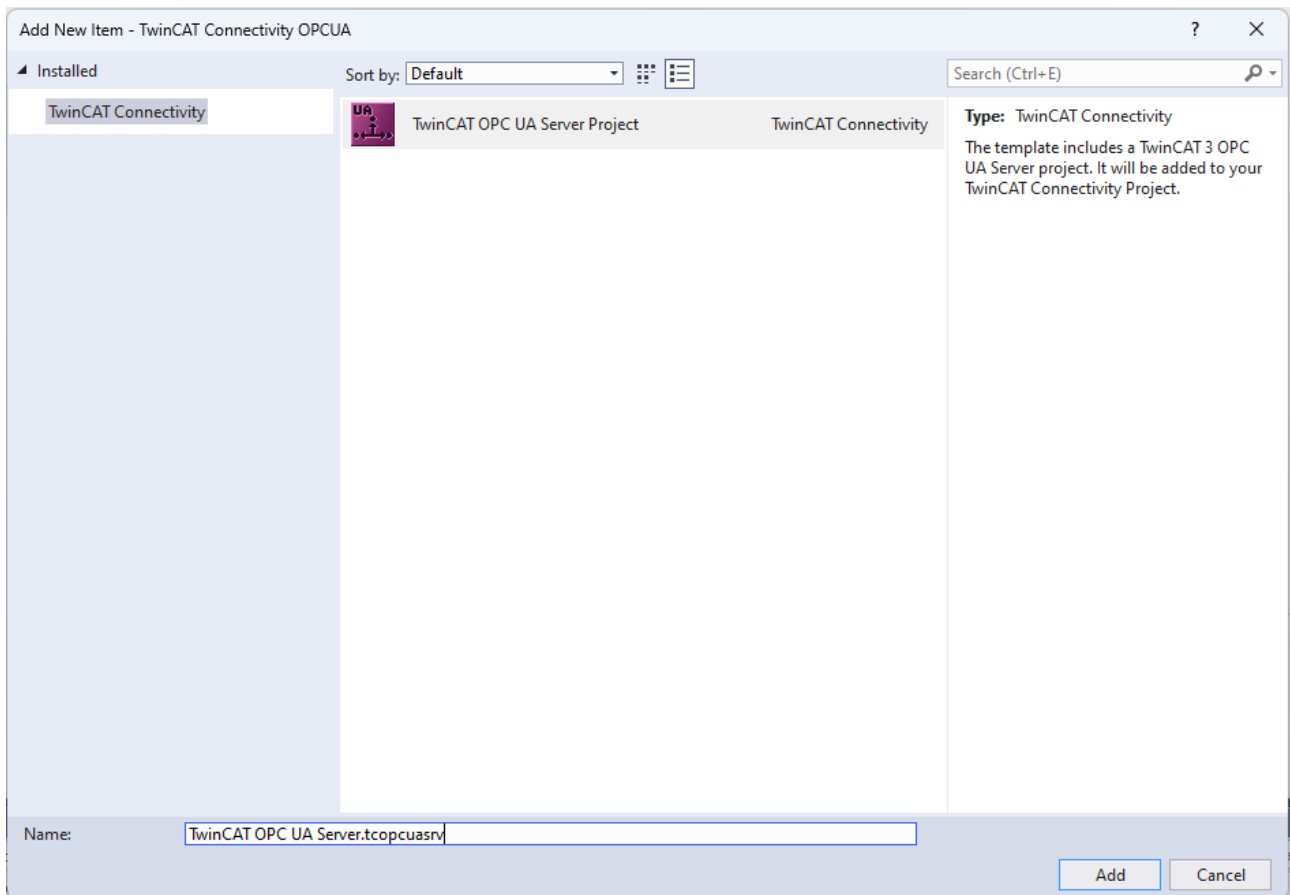
Lisätään OPC UA palvelin valitsemalla Solution ja painetaan Add -> New Project...



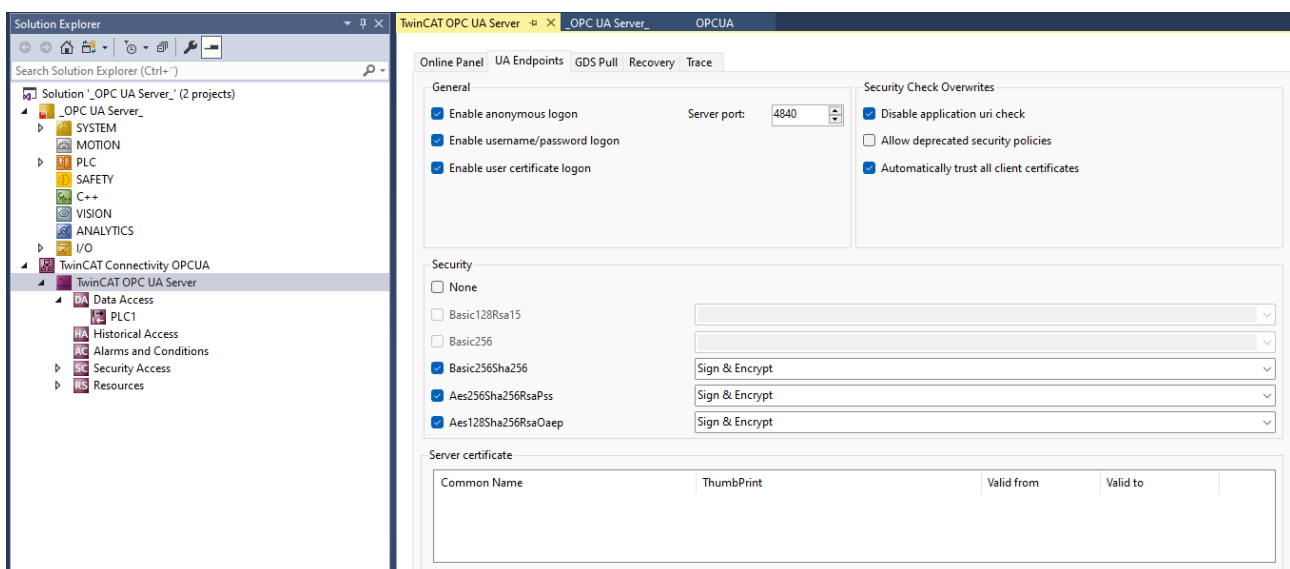
Valitaan TwinCAT Connectivity ja sieltä Empty TwinCAT Connectivity Project.



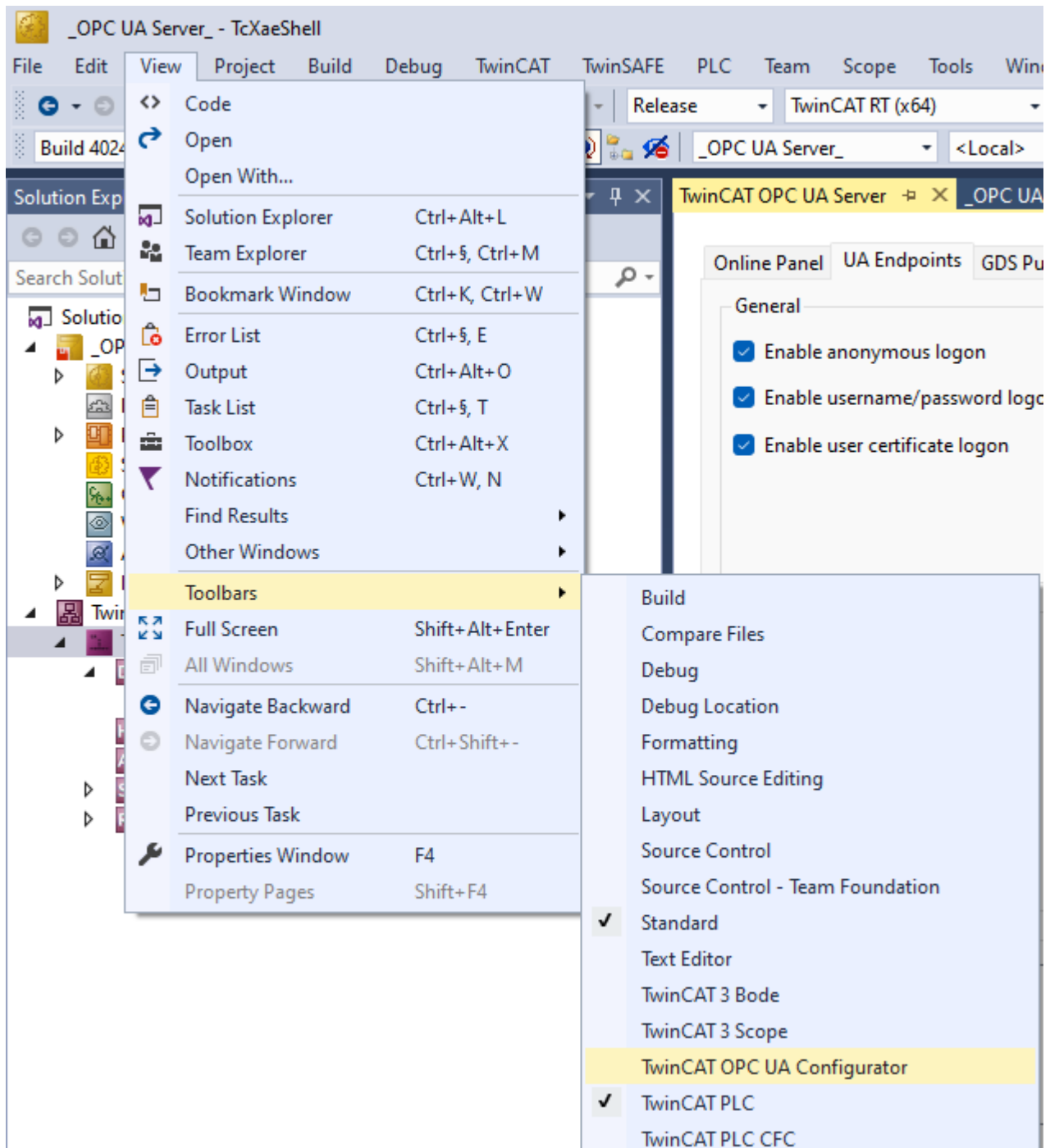
Lisätään New item...



Lisätään TwinCAT OPC UA Server Project.



OPC UA käyttää yhteydessään porttia 4840, joten tuplaklikataan juuri tehtyä palvelinta ja valitaan asetuksista UA Endpoints ja tarkastetaan että portti on 4840. Päätepiste on osoite jonka avulla saadaan pääsy palvelimeen. Security kohdassa voidaan muokata päätepisteiden (Endpoints) turvallisuutta tarvittaessa mutta nyt niitä ei tarvi muuttaa.



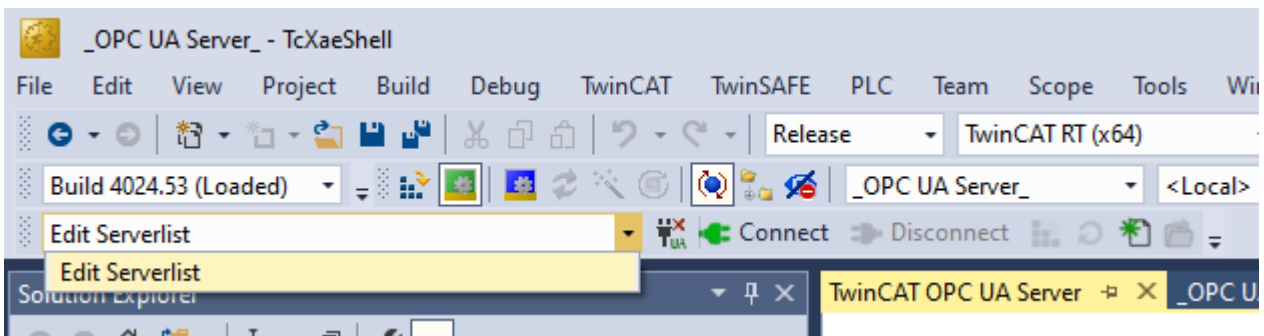
Lisätään TwinCAT OPC UA Configurator jonka avulla voimme hallita palvelinta.



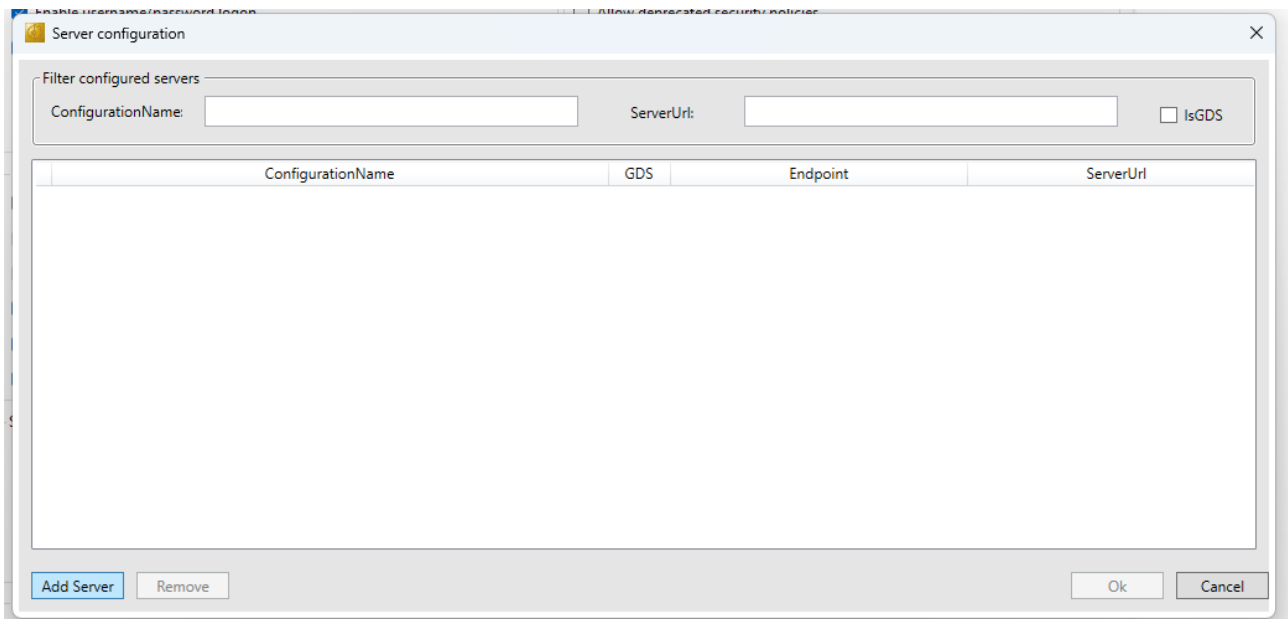
Sen pitäisi näkyä yläkulmassa itse olen siirtänyt sen kuvan mukaisesti.

Nyt ollaan saatu tehtyä TwinCAT projekti siihen pisteeseen että voimme alustaa palvelimen tämän kohdan voi tehdä monella tapaa näytän seuraavaksi 2 esimerkkiä ensiksi TwinCAT:tillä ja sen jälkeen UaExpert:illä.

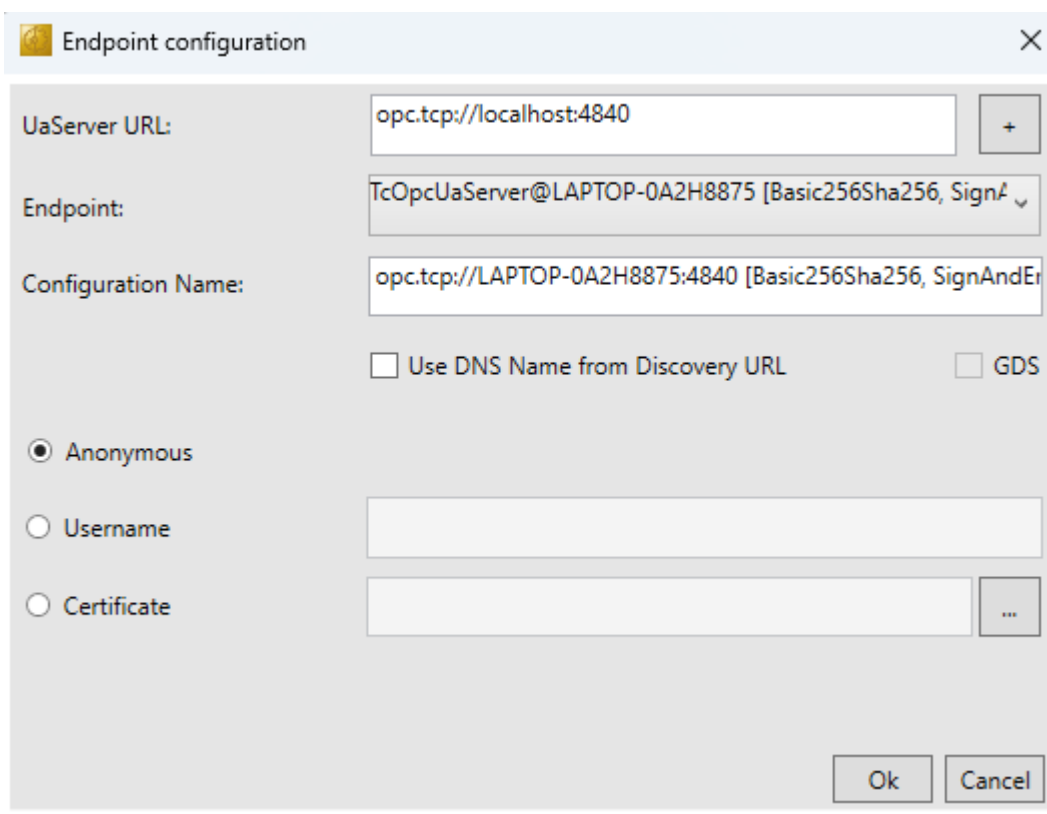
3 Palvelimen alustus TwinCAT:llä



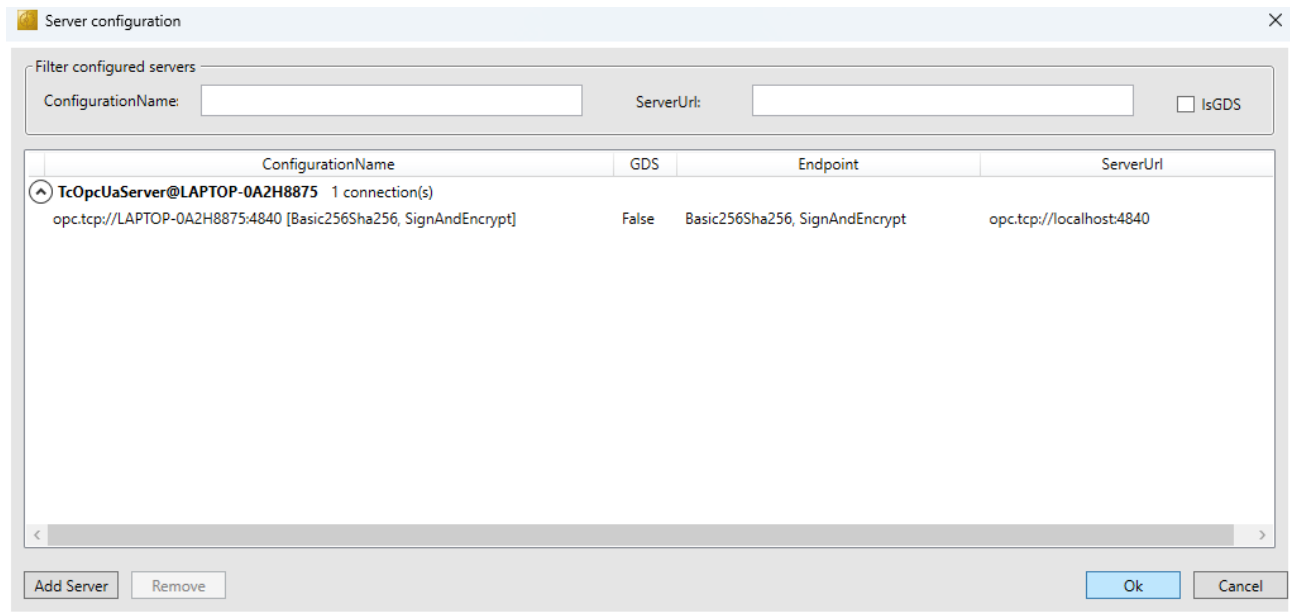
Sen pitäisi ilmestyä oikealla, kuvassa configurator on lisätty 3 riville. Lisätään palvelin valitsemalla Edit Serverlist.



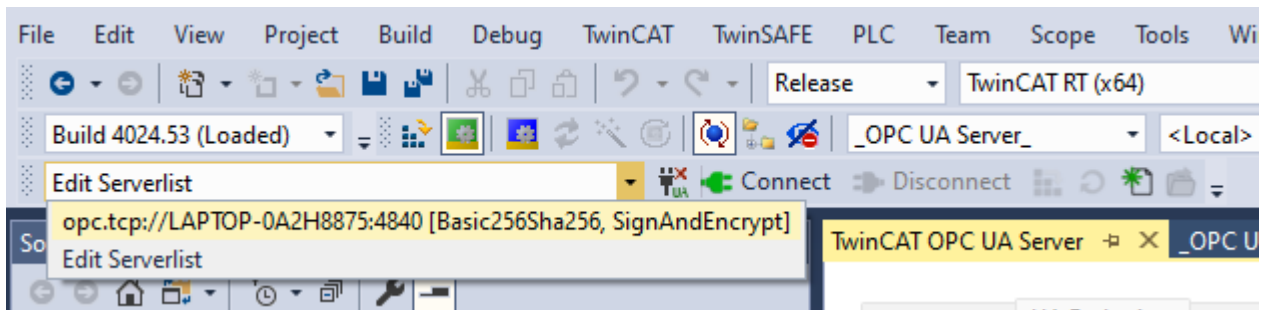
Painetaan Add Server.



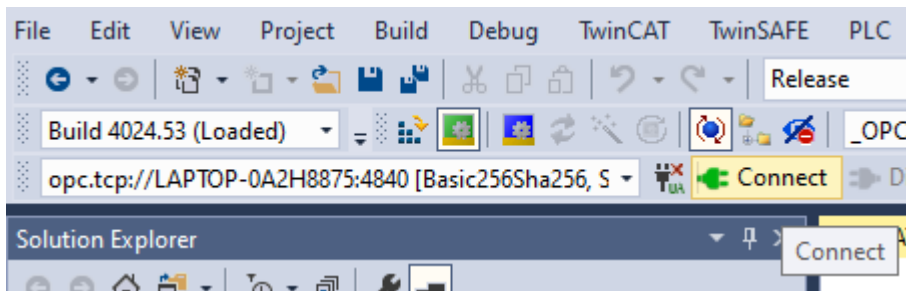
Valitaan Endpoint eli pääte piste joka määrittelee minkälaiset turvallisuus asetukset palvelimella on. Basic256Sha256 on hyvä valinta. Valitaan vielä Anonymous ja painetaan OK.



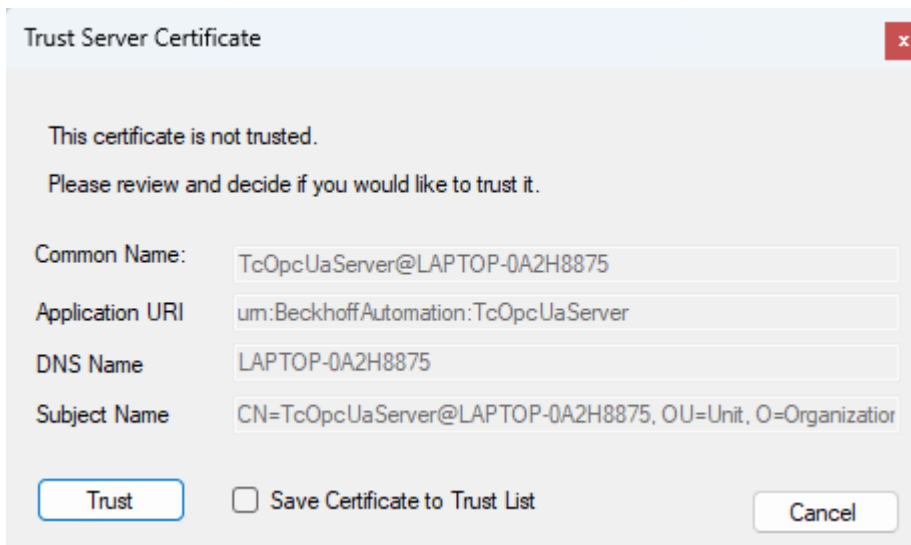
Tarkastetaan vielä että palvelin on lisätty ja painetaan OK.



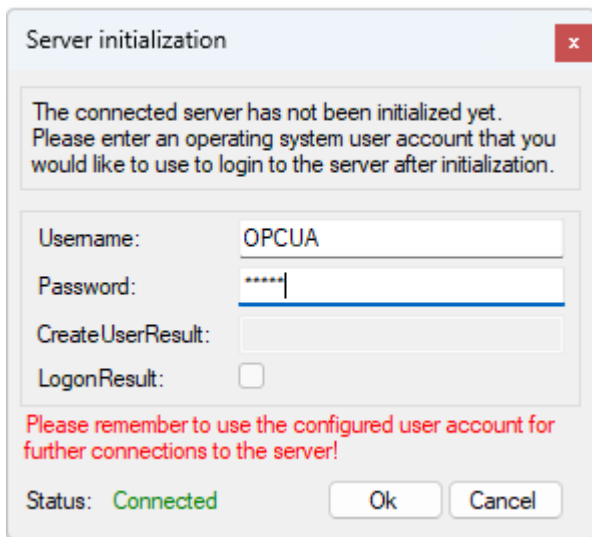
Valitaan juuri tehty palvelin.



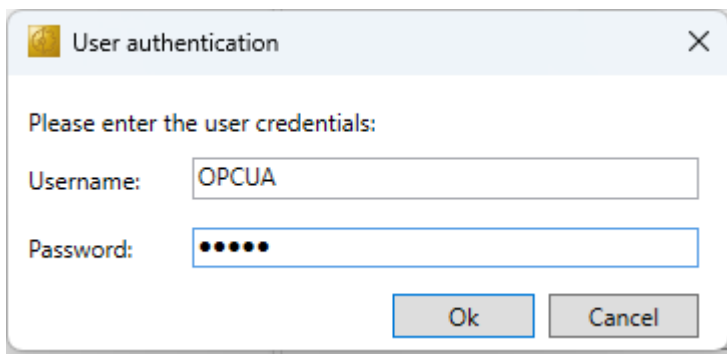
Painetaan Connect.



Painetaan Trust jotta palvelimen sertifiikaatti tallentuu trust listalle jota kautta turvallinen yhteys palvelimen ja PLC välillä on syntynyt.

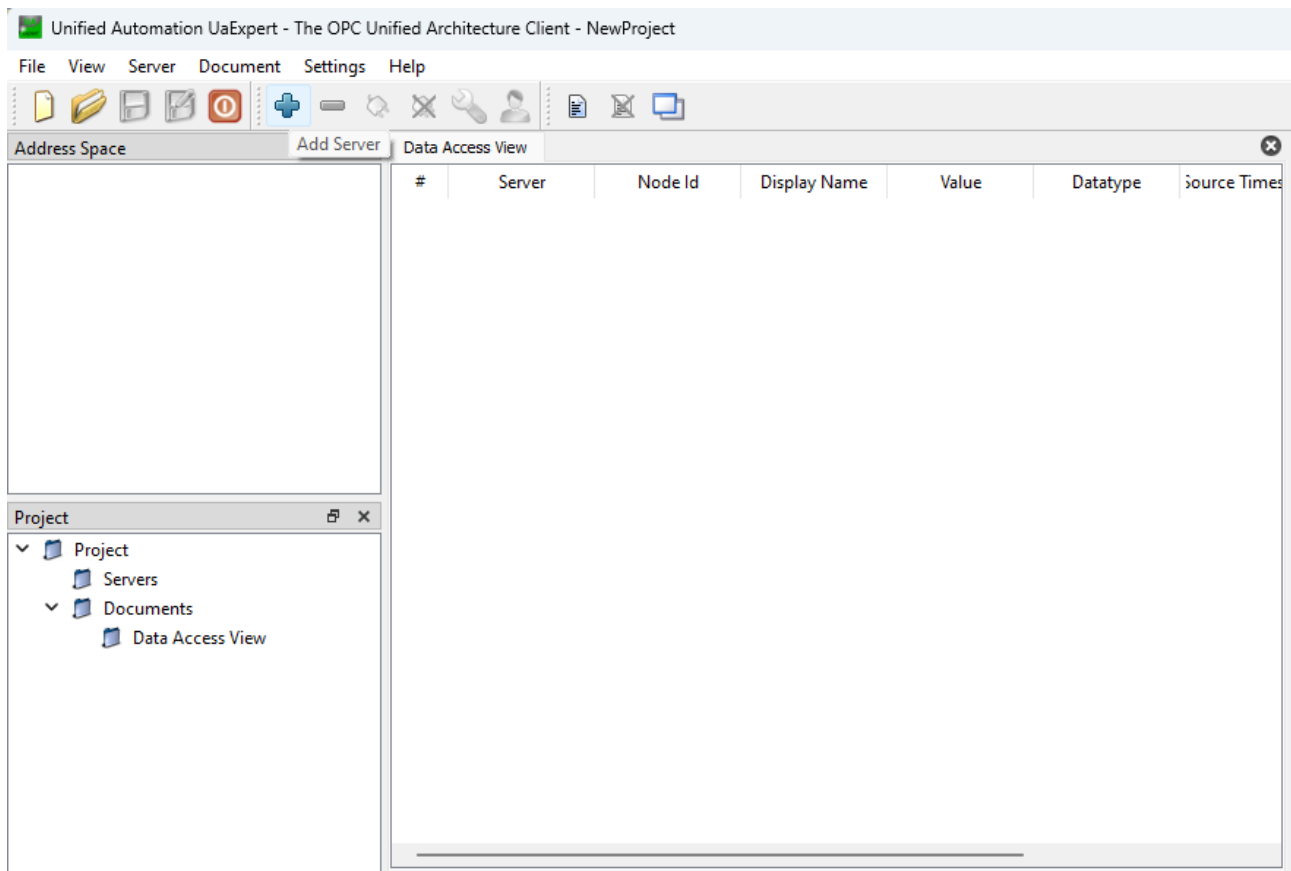


Kun palvelimeen yhdistetään ensimmäistä kertaa niin palvelin alustetaan ja valitaan palvelimelle käyttäjänimi ja salasana jonka jälkeen painetaan Ok.

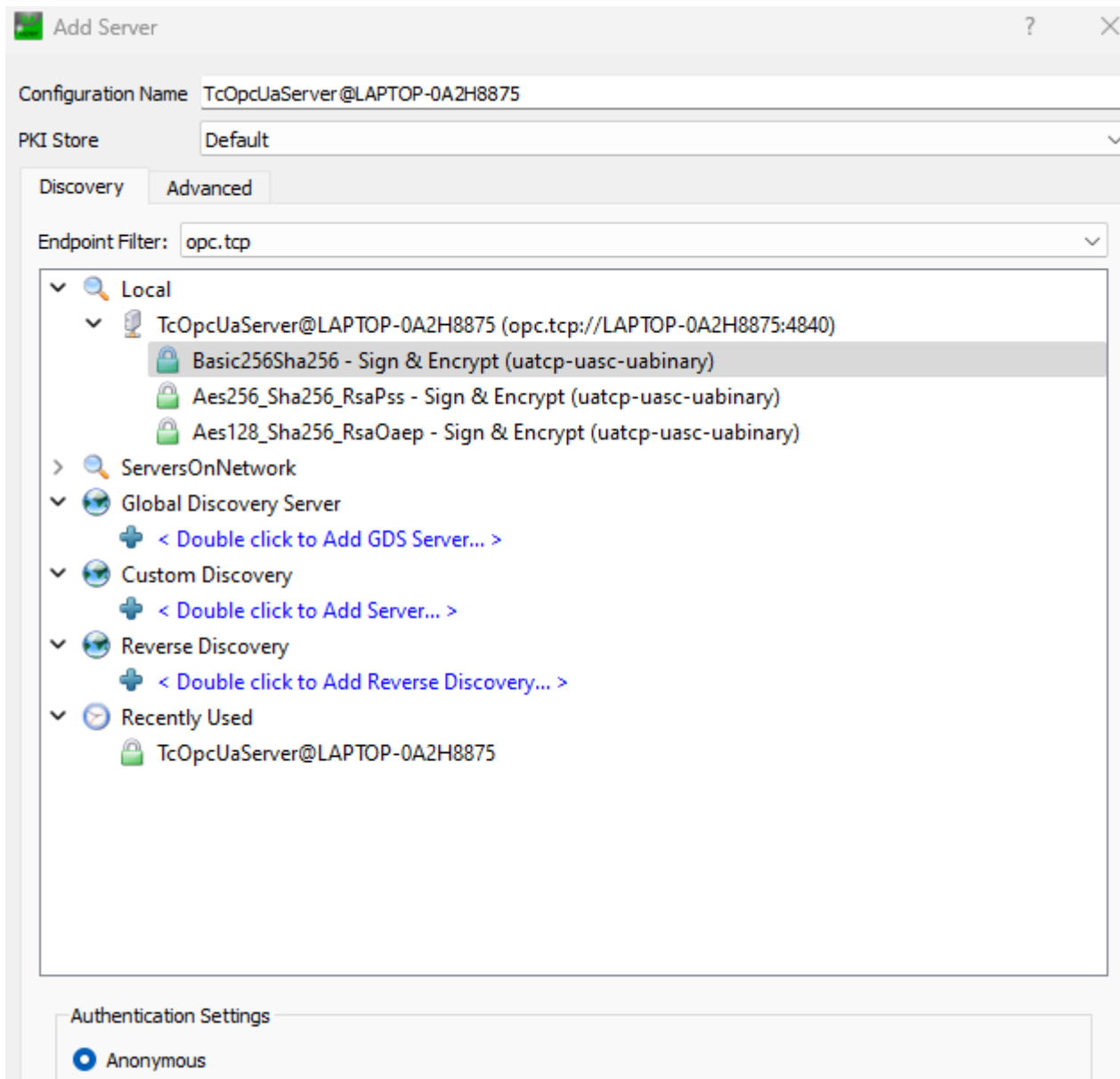


Jos yhteys palvelimeen ei tule automaattisesti, voidaan yhdistää uudestaan laittamalla juuri tehdyt käyttäjätunnukset.

4 Palvelimen alustus UaExpertillä



Tehdään seuraavaksi OPC UA palvelin UaExpertillä jos palvelin on tehty TwinCAT:illä niin voidaan vain yhdistää jo tehtyyn palvelimeen ei vain tehdä alustusta. Avataan UaExpert ja painetaan + näppäintä (Add Server)



Tämän jälkeen valitaan local josta pitäisi löytyä TcOpcUaServer jonka alla on kolme vaihtoehtoa. Nämä kolme vaihtoehtoa ovat eri päätepiteitä joissa on eri turvallisuustaso tässä tapauksessa kaikki ovat hyviä vaihtoehtoja. Valitaan vaikka Vasic256Sha256 ja painetaan OK.

Authentication Settings

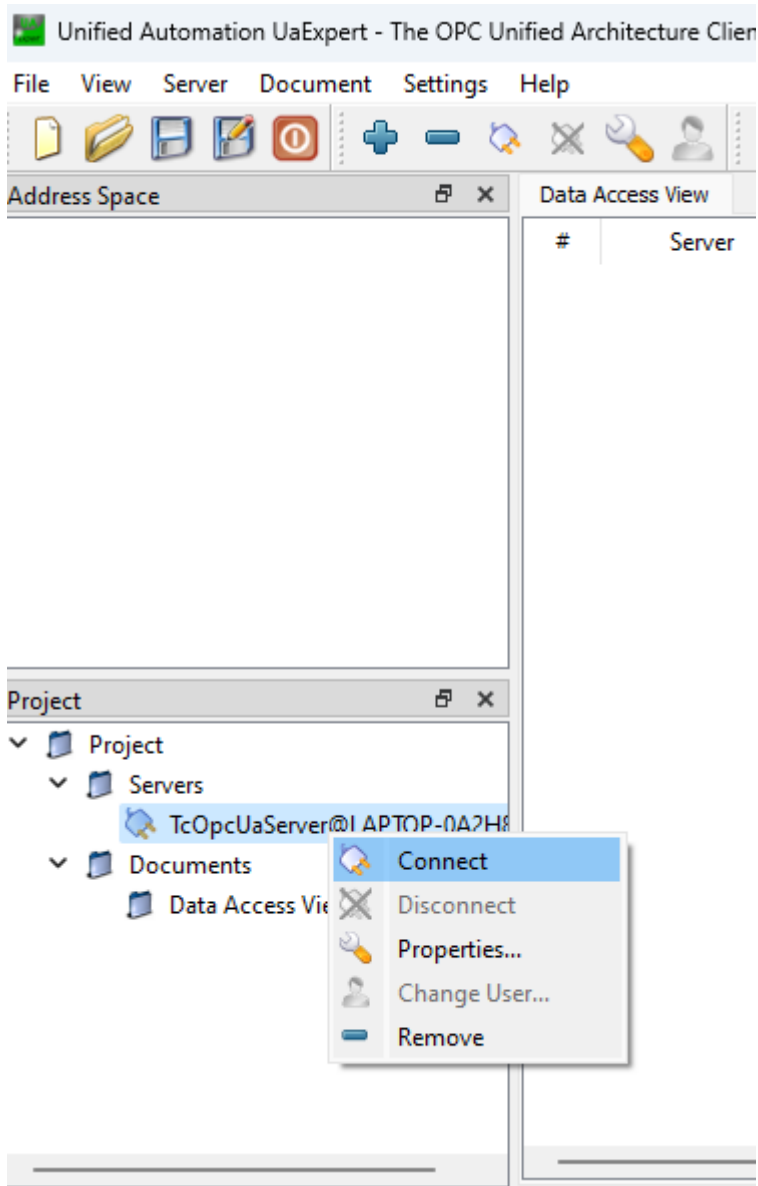
Anonymous

Username Store

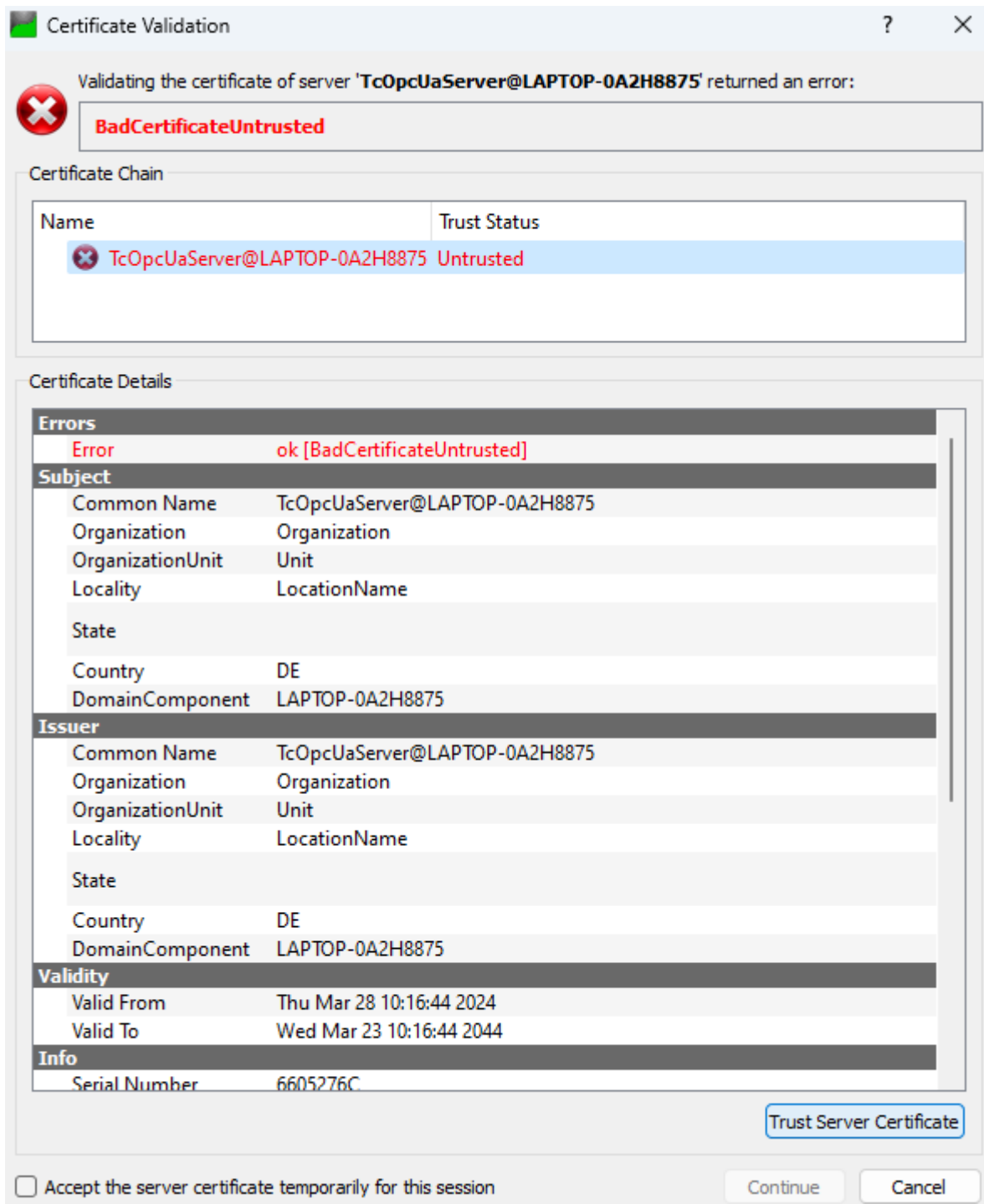
Certificate ...

Private Key ...

Jos palvelin on tehty, niin lisätään palvelimen tunnukset. Jos tehdään uusi valitaan Anonymous.

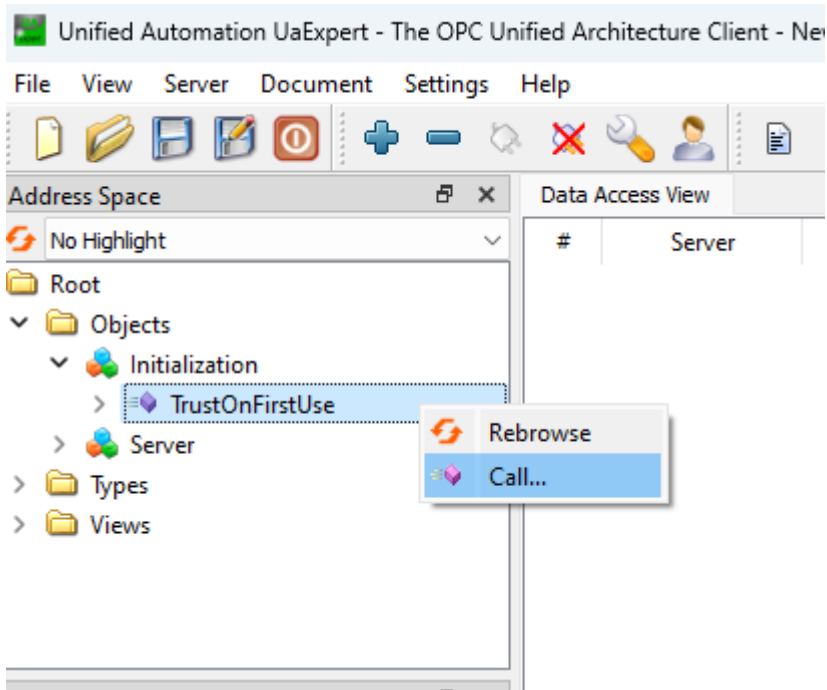


Yhdistetään palvelimeen painamalla connect.

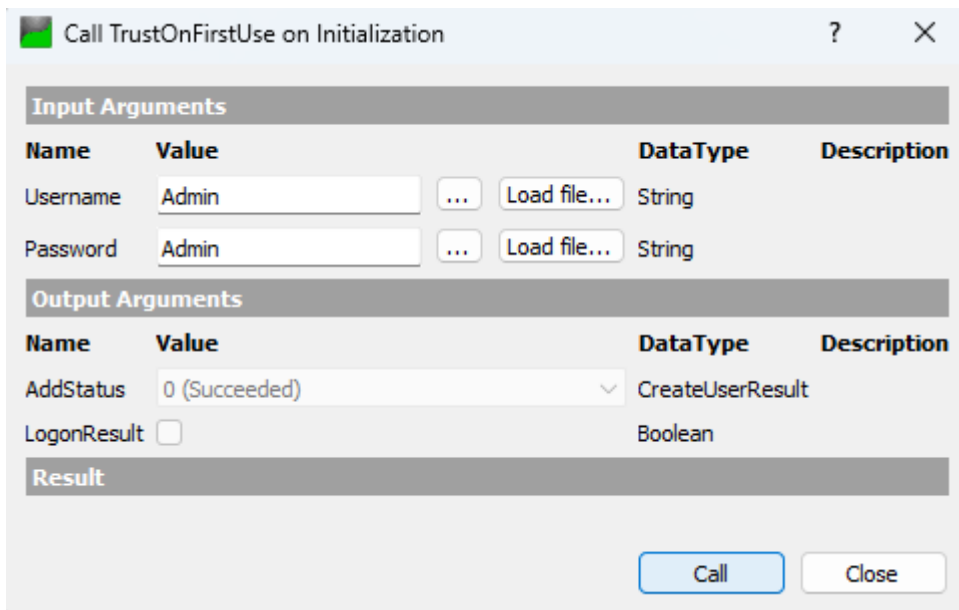


Valitaan Trust Server Certificate eli luodaan turvallinen yhteys palvelimen ja tietokoneen välille.

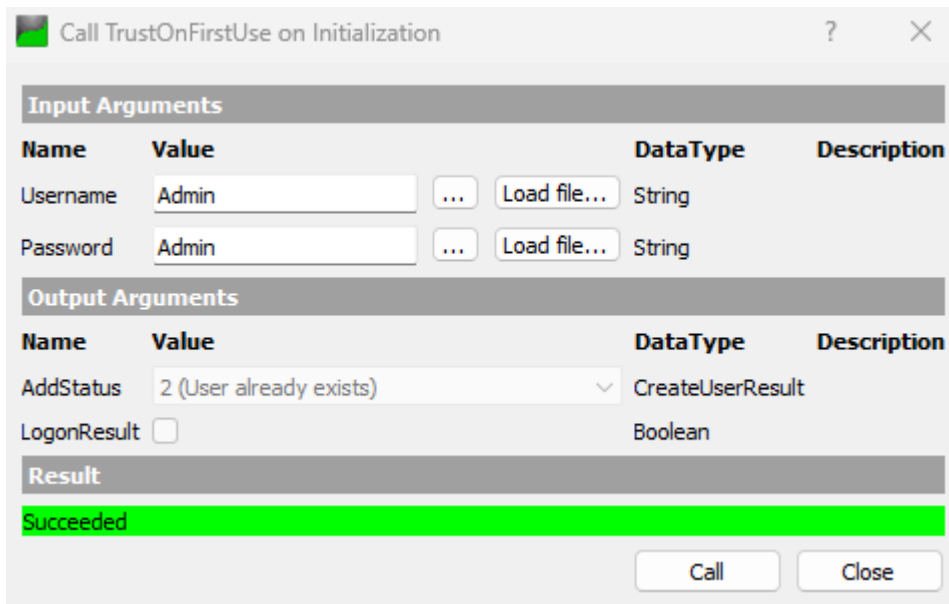
Tämän jälkeen painetaan Continue.



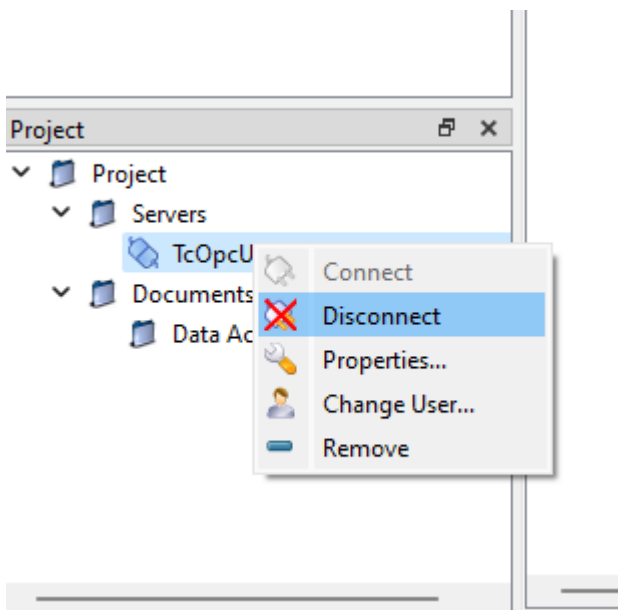
Osoiteavaruuden ilmestyi Initalization painike, josta löytyy TrustOnFirstUse metodi jonka avulla voimme luoda palvelimelle tunnukset. Valitaan Call..



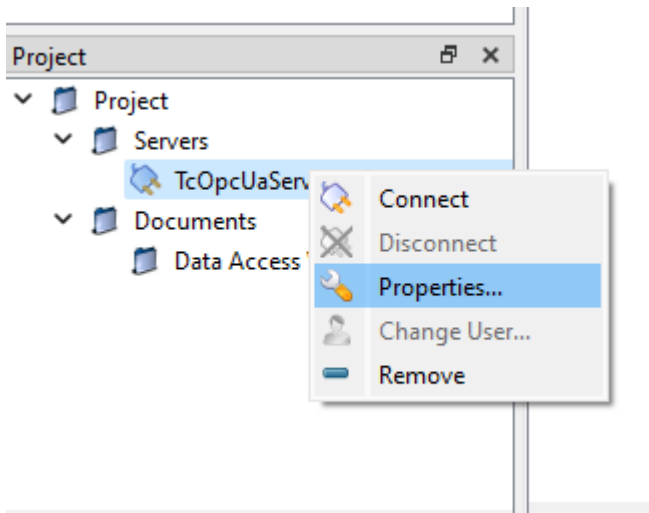
Lisätään käyttäjätunnus ja salasana, jonka jälkeen painetaan Call.



Onnistuneen alustuksen merkinä tuloksiin ilmestyy Succeeded, jonka jälkeen ikkunan voi sulkea.



Tämän jälkeen katkaistaan yhteys Disconnect napista.



Mennään sen jälkeen palvelimen ominaisuuksiin.

Server Settings - TcOpcUaServer@LAPTOP-0A2H8875

Configuration

Configuration Name: TcOpcUaServer@LAPTOP-0A2H8875

PKI Store: Default

Server Information

Endpoint Url: opc.tcp://LAPTOP-0A2H8875:4840

Reverse Connect:

Security Settings

Security Policy: Basic256Sha256

Message Security Mode: Sign & Encrypt

Authentication Settings

Anonymous

Username: Admin Store

Password: ●●●●●

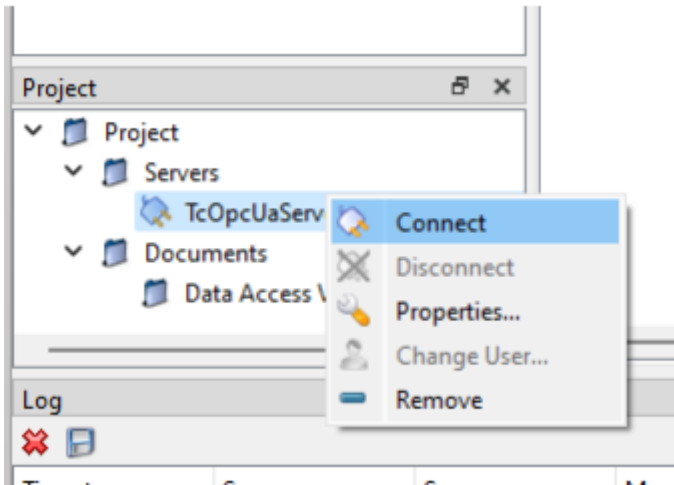
Certificate: ...

Private Key: ...

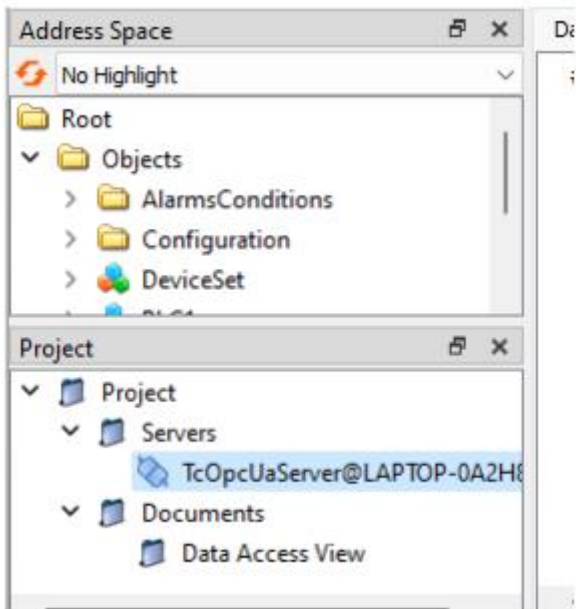
Session Settings

Session Name: TOP-0A2H8875:UnifiedAutomation:UaExpert

Mennään Authenciation Settings kohtaan ja lisätään juuri tekemät tunnukset ja painetaan OK.

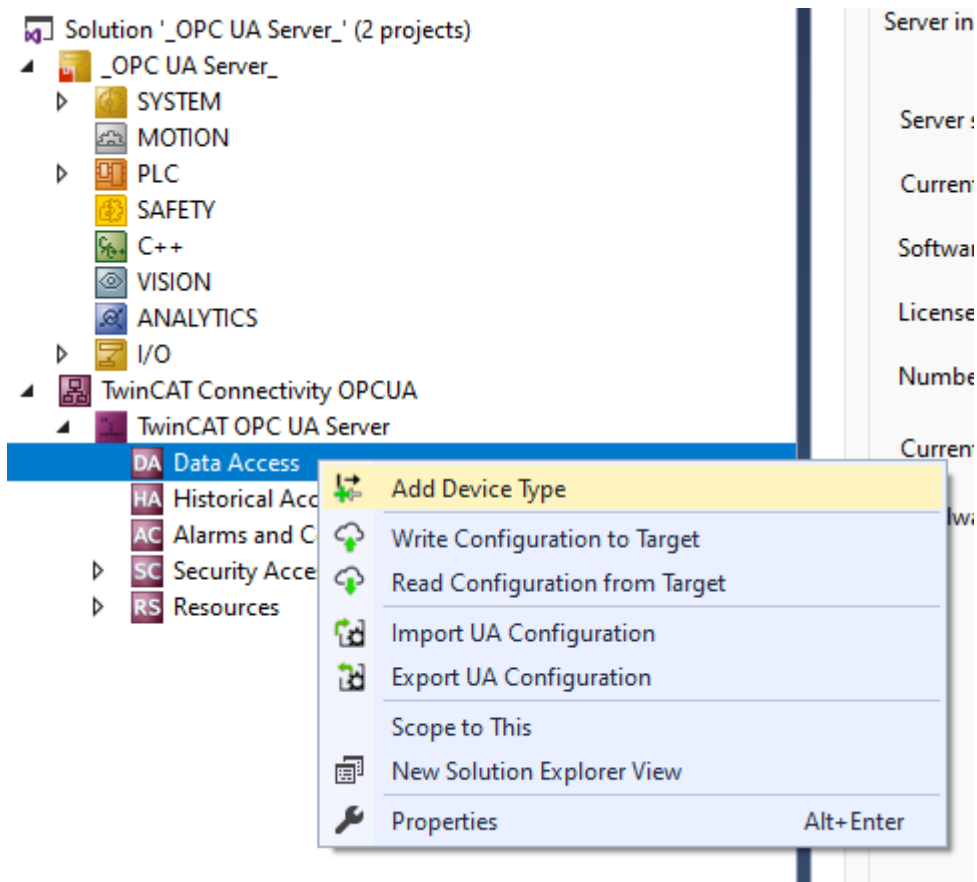


Yhdistetään uudestaan palvelimeen.

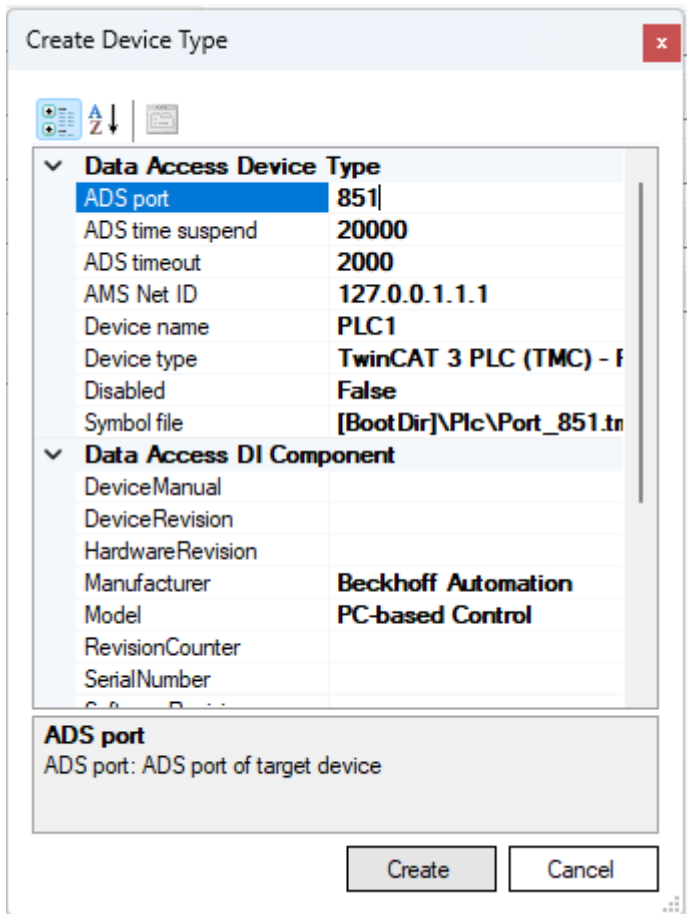


Osoitevaruuteen pitäisi ilmestyä tiedostoja yhdistymisen onnistumisen merkiksi.

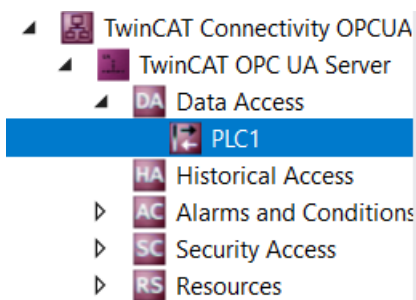
5 Data Accessin lisäys



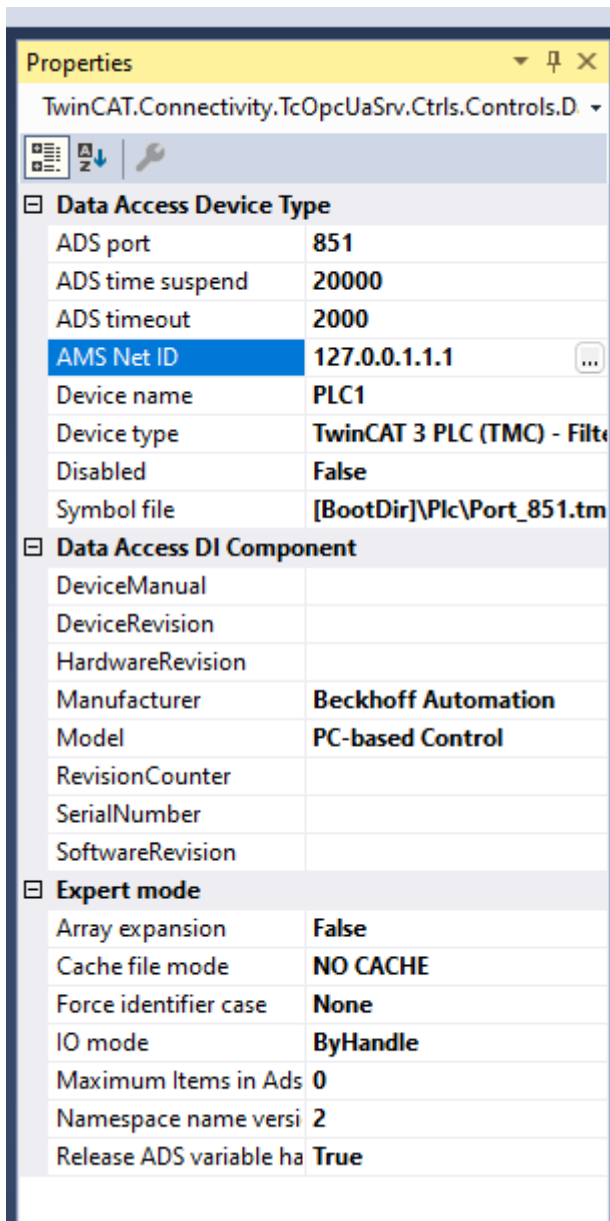
Lisätään Add Device Type joka lisää uuden Data Access. Data Access tallentaa GVL lisätyt muuttujat tmc tiedoston avulla. Nämä muuttujat voidaan lukea, kirjoittaa ja monitoroida OPC UA- rajapinnan kautta.



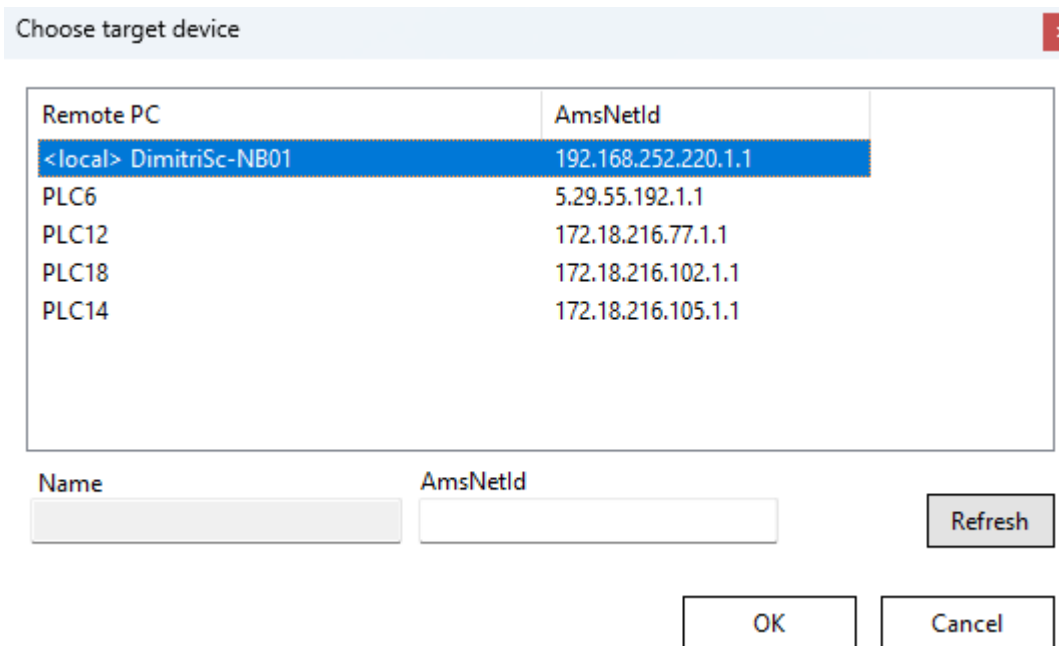
Lisätään ADS port kohtaan 851, tämä on se portti jota PLC käyttää ja painetaan Create.



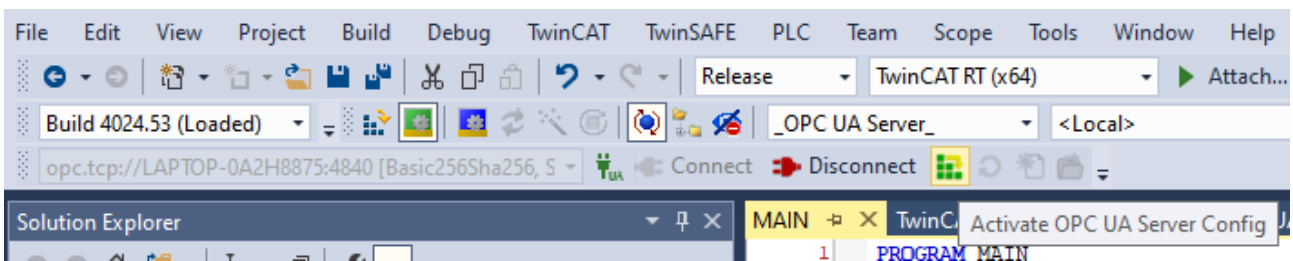
Valitaan juuri tehty Data Access.



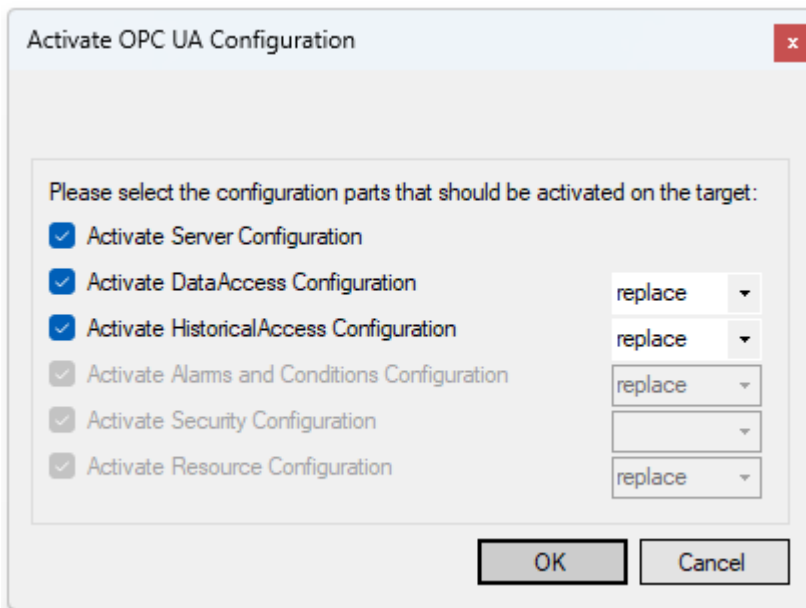
Oikealla pitäisi näkyä Properties. Sieltä painetaan AMS Net ID kolmea pistettä.



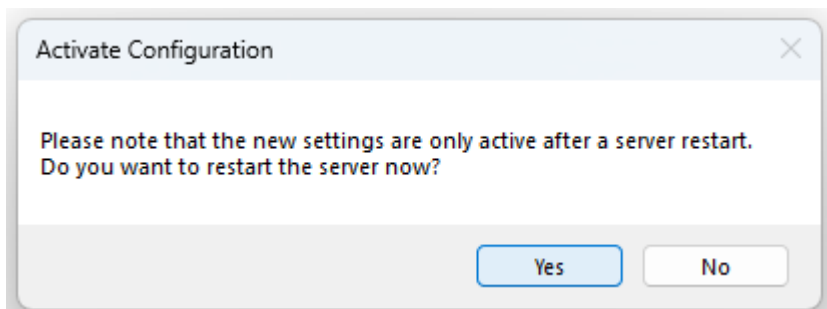
Valitaan Local joka on paikallinen AmsNet id TwinCAT:in verkossa.



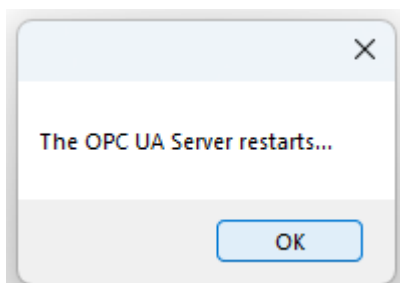
Painetaan Activate OPC UA Server Config nappia.



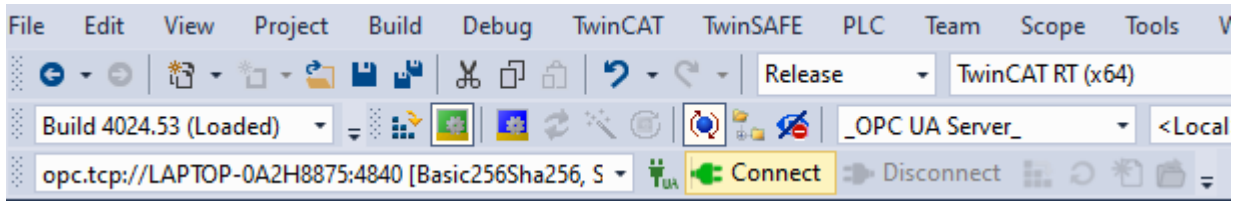
Painetaan OK joka korvaa aikaisemman konfiguraation.



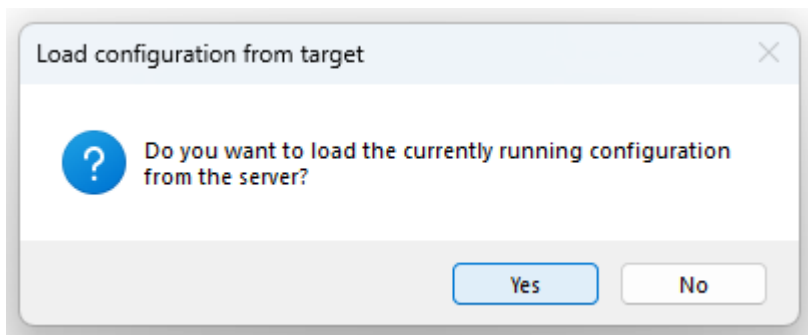
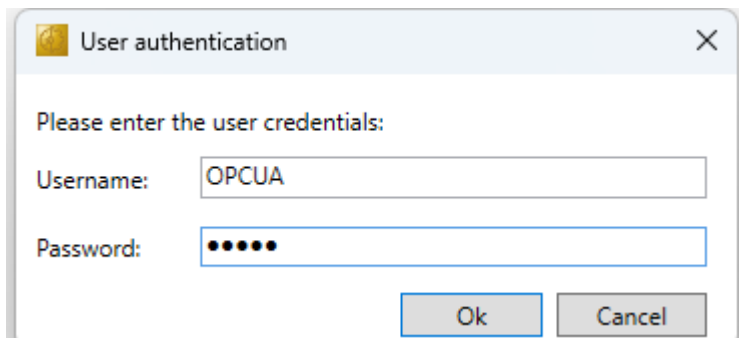
Painetaan Yes palvelimen uudelleenkäynnistämistä varten.



Painetaan OK.

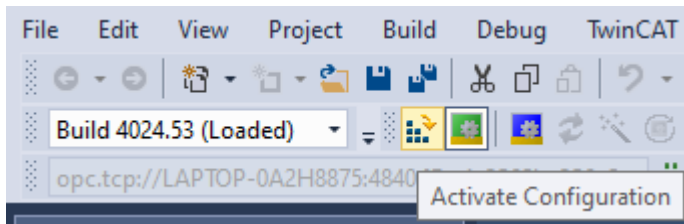


Yhdistetään uudestaan palvelimeen.

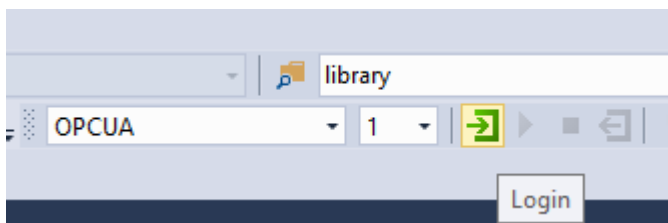
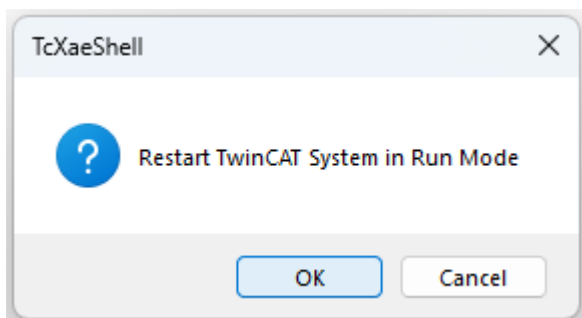
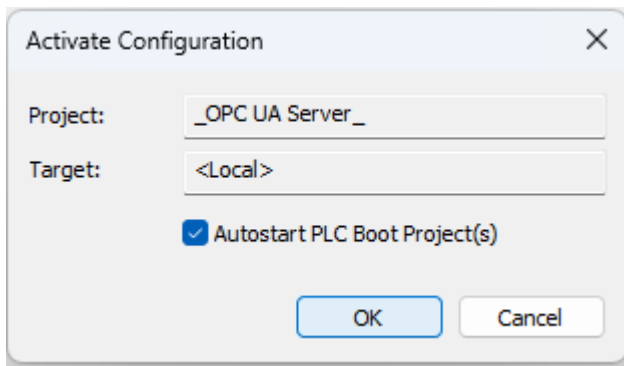


Tämän jälkeen ohjelma kysyy halutaanko ladata tämän hetkiset konfiguraatiot eli se konfiguraatio joka on tällä hetekllä päällä. Painetaan Yes tämän jälkeen palvelin on valmis.

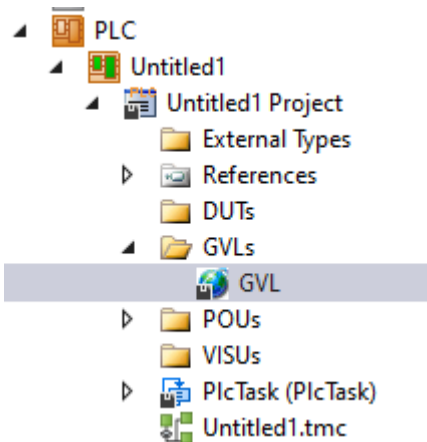
6 Testaus TwinCAT:n ja UaExpertin välillä



Ensiksi tehdään TwinCAT aktiiviseksi painamalla Activate Configuration. Palvelin käyttää tämän konfiguraation TMC tiedostoa juuri tehdyssä Data Access:issa.



Painetaan Login ohjelman jonka jälkeen PLC on päällä ja tehtyä ohjelmaa voidaan testata.

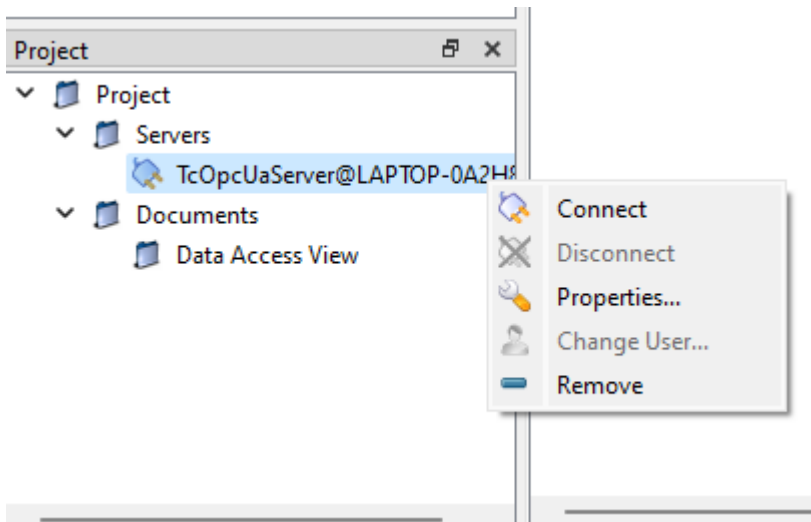


Valitaan vielä GVL.

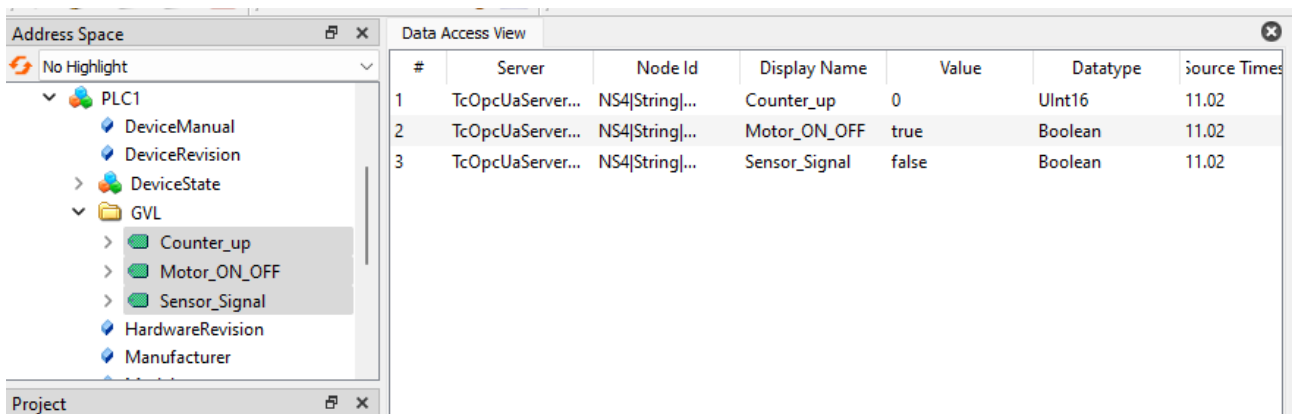
The image shows a screenshot of the GVL variable declaration table in TwinCAT. The window title is 'GVL [Online] - x MAIN [Online]'. The table is titled 'TwinCAT_Project10.Untitled1.GVL' and has six columns: 'Expression', 'Type', 'Value', 'Prepared value', 'Address', and 'Comment'. There are three rows of data:

Expression	Type	Value	Prepared value	Address	Comment
Sensor_Signal	BOOL	FALSE			
Motor_ON_OFF	BOOL	TRUE			
Counter_up	WORD	0			

Pitäisi ilmestyä kuvan mukainen näkymä jossa muuttujat ovat aktiivisia.



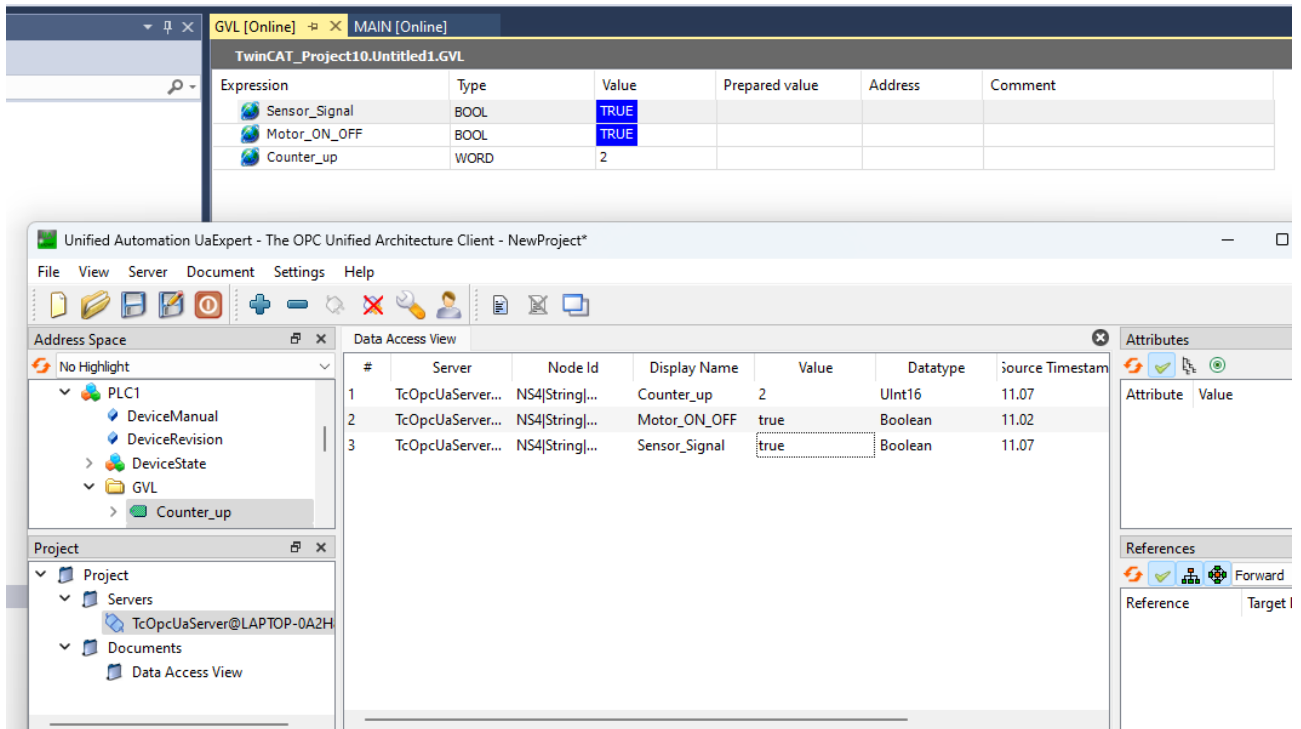
Yhdistetään vielä palvelimeen UaExpert:illä.



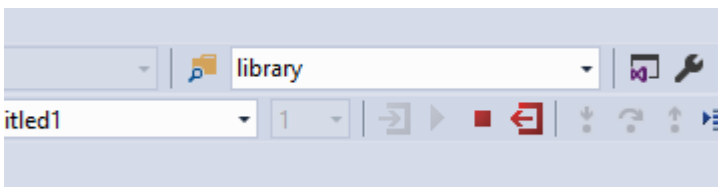
Valitaan osoitevaruudesta PLC -> GVL -> valitaan muuttujat ja raahataan ne Data Access näkymään.

#	Server	Node Id	Display Name	Value	Datatype	Source T
1	TcOpcUaServer...	NS4 String ...	Counter_up	1	UInt16	11.06
2	TcOpcUaServer...	NS4 String ...	Motor_ON_OFF	true	Boolean	11.02
3	TcOpcUaServer...	NS4 String ...	Sensor_Signal	<input checked="" type="checkbox"/>	Boolean	11.07

Testaus tapahtuu valitsemalla muuttujan esim itse valitsin Sensor_Signal muuttujan joka on boolean joten tuplaklikkaa Value kohdasta jolloin voidaan painaa sitä jolloin muuttuja vaihtaa arvoa False -> True.

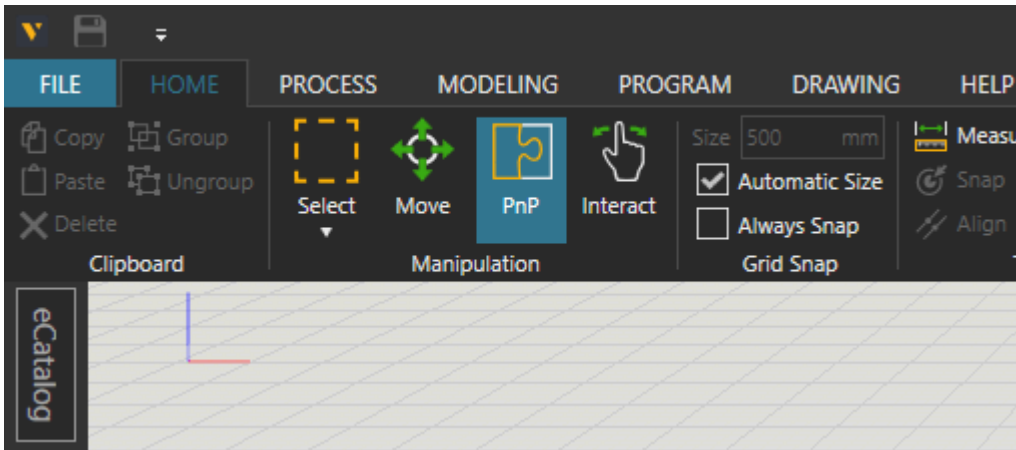


Testaus on valmis jos arvot muuttuvat. Kannattaa muistaa että kaikkien muuttujien arvoa ei voi muokata esim tässä tapauksessa ohjelma estää esim Counter_up arvon muuttamisen.

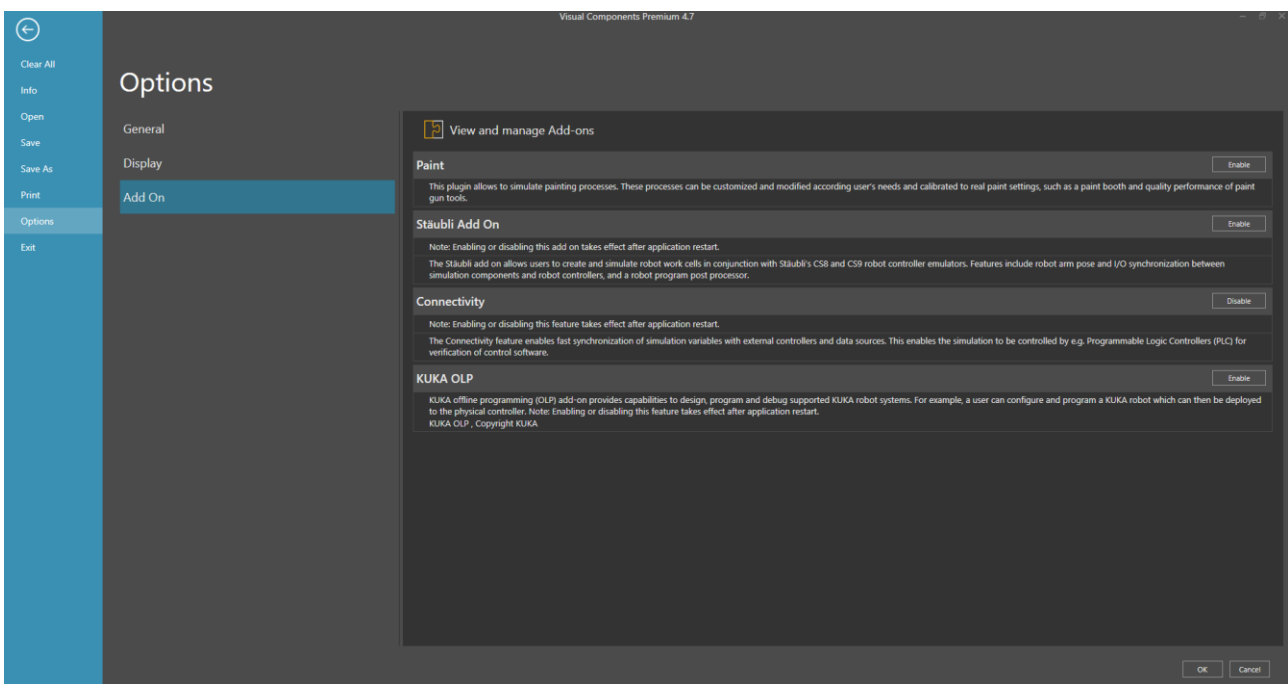


Ohjelman voi pysäyttää punaisesta neliöstä tai voi myös kirjautua ulos.

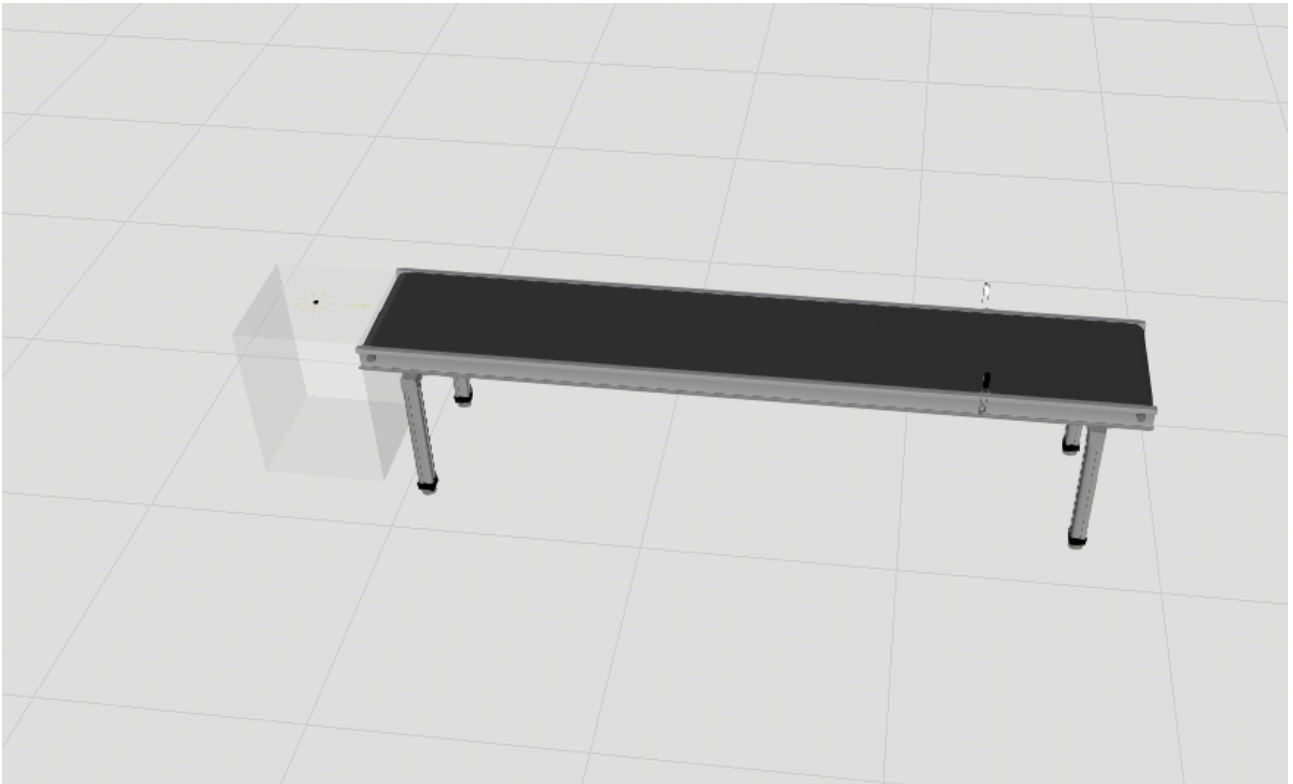
7 Visual Components -mallinnuksen teko ja yhdistäminen OPC UA -palvelimeen



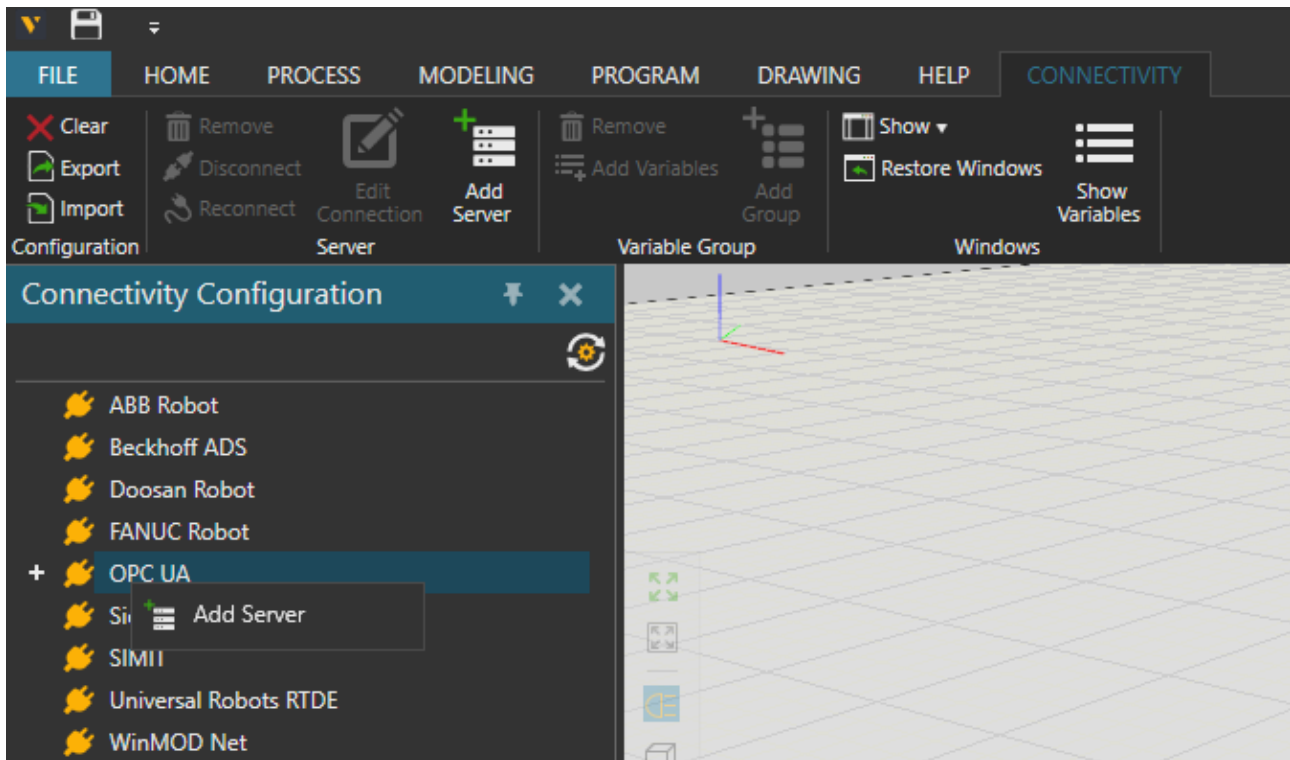
Ensin valitaan FILE.



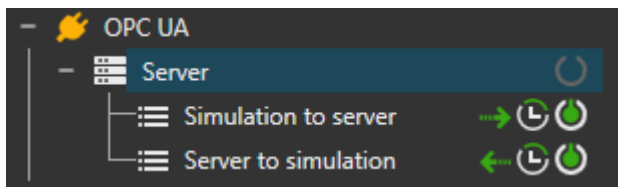
Valitaan Options -> Add On -> Connectivity Enable. Tämän avulla saadaan Connectivity valikko käyttöön jonka avulla voidaan yhdistää OPC UA palvelimeen.



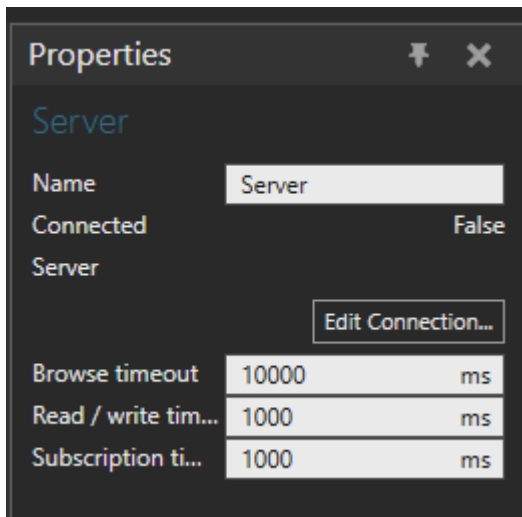
Tehdään yksinkertainen Visual Components ohjelma jossa on feeder, conveyor ja conveyor sensor.



Mennään Connectivity sivulle. Lisätään palvelin painamalla Add Server.



Valitaan Server.



Oikealla ilmestyy Properties valikko josta painetaan Edit Connection.

Edit Connection [X]

Connect to a new server or edit connection parameters.

Find Servers

Discovery server:

Discovered Servers

TcOpcUaServer@LAPTOP-... opc.tcp://LAPTOP-0A2H8875:4840

Connection

Server address:

Use secure endpoint:

Authentication

None

Username

Username:

Password:

Save password:

Certificate

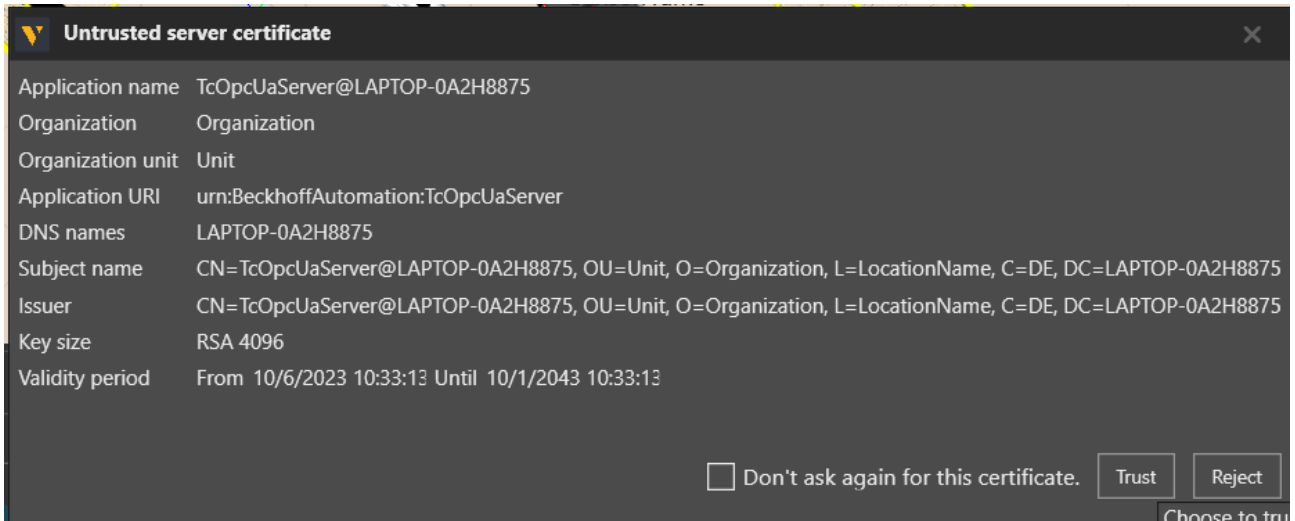
User private key file: ...

Password:

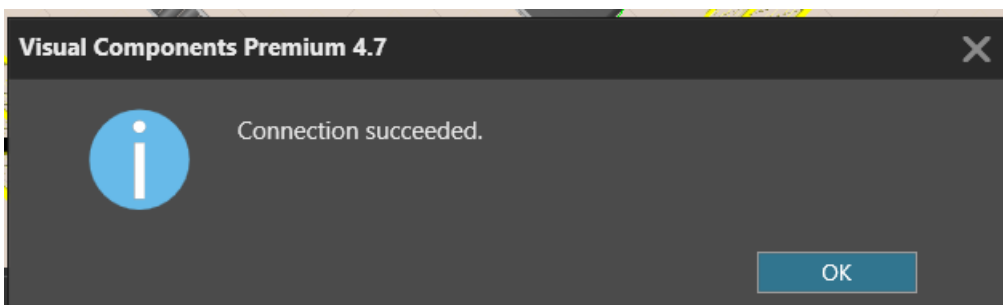
Save password:

Begin a connection attempt using the currently visible settings.

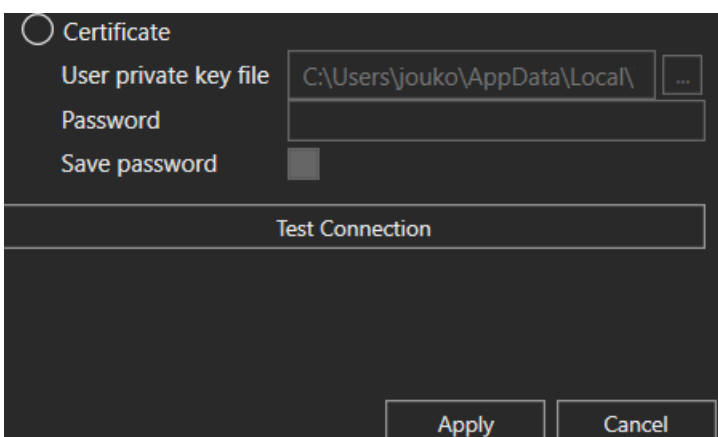
Valitaan Discovered Servers alta palvelin tai jos ei löydy palvelinta, niin Server address kohtaan voidaan kirjoittaa oman palvelimen nimi. Valitaan Use secure endpoint, joka muodostaa turvallisen yhteyden palvelimen ja asiakkaan välille. Lisätään vielä oma käyttäjänimi ja salasana (jos palvelin ei vaadi käyttäjätunnusta ja salasanaa voidaan valita myös None) jonka jälkeen painetaan Test Connection.



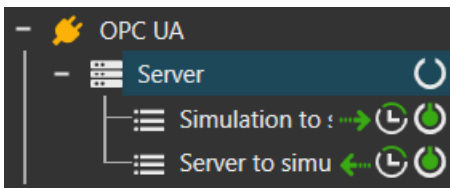
Valitaan Trust joka lisää tämän asiakkaan palvelimen trust listalle.



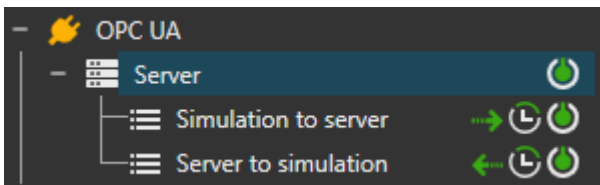
Onnistuneen testin jälkeen ilmestyy Connection succeeded.



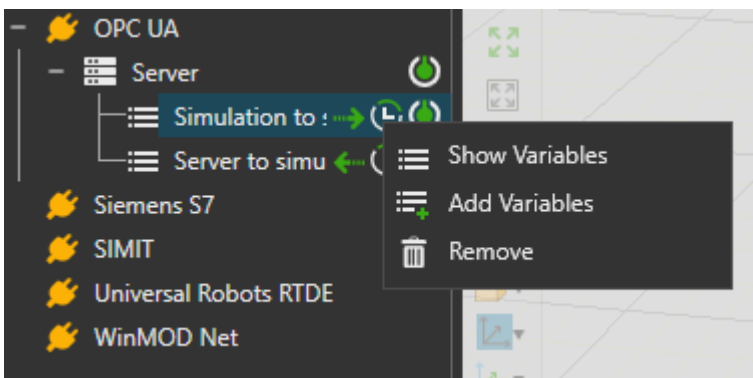
Painetaan Apply.



Painetaan Server nappia.

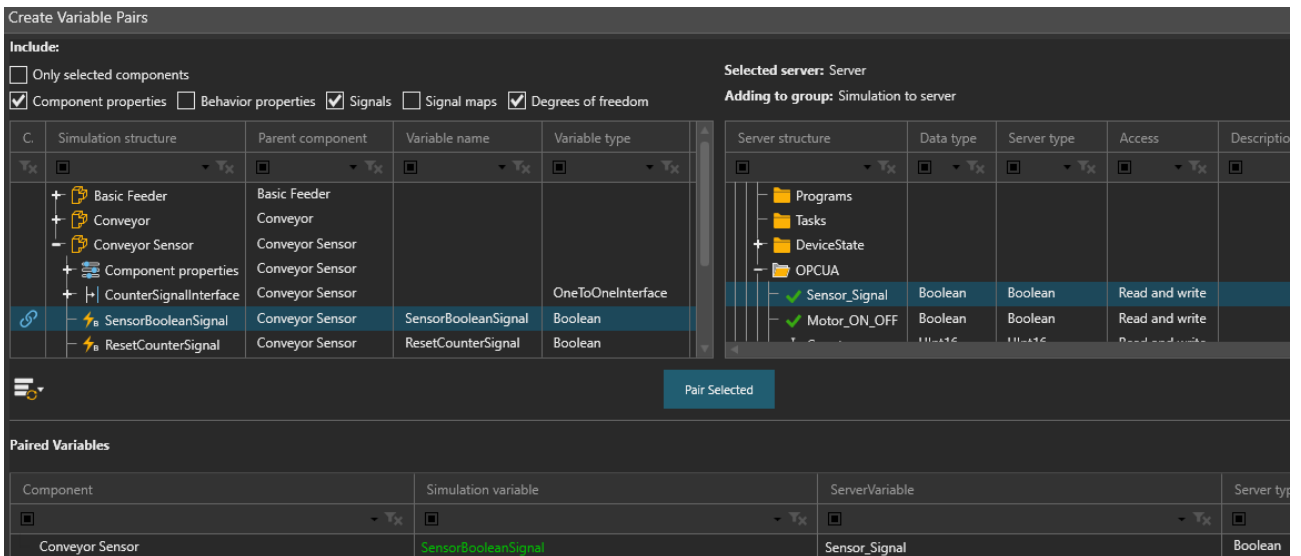


Napin pitäisi muuttu vihreäksi onnistuneen yhteyden merkiksi.

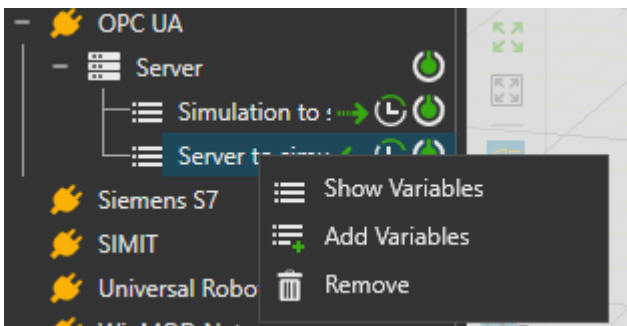


8 OPC UA -yhteyden testaus

Tehdään seuraavaksi tarvittavat yhteydet mitkä muuttujat kommunikoivat keskenään. Lisätään muuttuja Simulation to server, painamalla Add Variables.



Valitaan vasemmalta puolelta Conveyor Sensor -> SensorBooleanSignal ja oikealta Objects -> PLC1 -> OPCUA (GVL) -> Sensor_Signal jonka jälkeen pitäisi näkyä alla kohdassa Paired Variables.



Lisätään Server to simulation.

Create Variable Pairs

Include:

Only selected components

Component properties Behavior properties Signals Signal maps Degrees of freedom

Selected server: Server

Adding to group: Server to simulation

C.	Simulation structure	Parent component	Variable name	Variable type	Server structure	Data type	Server type	Access	Description
	Basic Feeder	Basic Feeder			Programs				
	Conveyor	Conveyor			Tasks				
	Component properties	Conveyor			DeviceState				
	PowerOnSignal	Conveyor	PowerOnSignal	Boolean	OPCUA				
	TransitionSignal	Conveyor	TransitionSignal	Boolean	Sensor_Signal	Boolean	Boolean	Read and write	
	Conveyor Sensor	Conveyor Sensor			Motor_ON_OFF	Boolean	Boolean	Read and write	

Pair Selected

Paired Variables

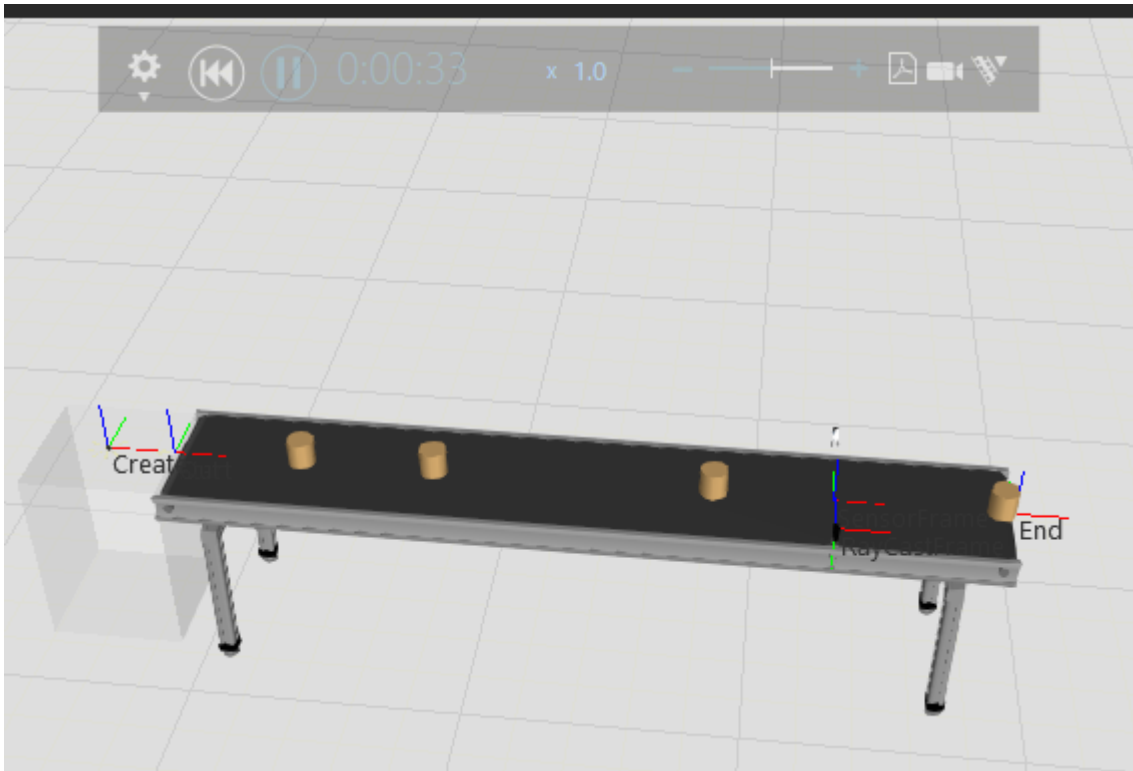
Component	Simulation variable	ServerVariable	Server type
Conveyor	PowerOnSignal	Motor_ON_OFF	Boolean

Tehdään sama toiselle muuttujalle vasemmalta Conveyor -> PowerOnSignal ja oikealta Objects -> PLC1 -> OPCUA (GVL) -> Motor_ON_OFF.

Connected Variables

Structure	Simulation variable	...	Simulati...	Prepared...	Latest va...	..	Server variable	Server type
Server								
Simulation to server								
SensorBooleanSignal	Conveyor.Sensor.SensorBooleanSignal	⚡	FALSE		FALSE	✓	Sensor_Signal	Boolean
Server to simulation								
PowerOnSignal	Conveyor.PowerOnSignal	⚡	FALSE		TRUE	✓	Motor_ON_OFF	Boolean

Nyt pitäisi näkyä alhaalla kohdassa Connected Variables tehdyt muuttujat.



Ohjelmaa voidaan nyt testata painamalla Play näppäintä.

Structure	Simulation variable	...	Simulati...	Prepared...	Latest va...	..	Server variable	Server type
Server								
Simulation to server								
SensorBooleanSignal	Conveyor.Sensor.BooleanSignal	⚡	TRUE		TRUE	✓	Sensor_Signal	Boolean
Server to simulation								
PowerOnSignal	Conveyor.PowerOnSignal	⚡	TRUE		TRUE	✓	Motor_ON_OFF	Boolean

```
1 counter(CU TRUE := GVL.Sensor_Signal TRUE, PV 5 := 5, RESET FALSE := btimer.Q FALSE);
2
3 GVL.Counter_up 3 := counter.CV 3;
4
5 IF counter.Q FALSE THEN
6     GVL.Motor_ON_OFF TRUE := FALSE;
7
8     ELSE
9         GVL.Motor_ON_OFF TRUE := TRUE;
10
11 END_IF
12
13 btimer(IN FALSE := counter.Q FALSE, PT T#5s := T#5S);RETURN
```

Onnistuneen yhteyden jälkeen voidaan seurata muuttuja Visual Component:sin ja PLC kautta.