Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

PETRI RANTAVALLI

PLC-datan keskitetty tiedonkeruu ja kyberturvallisuus

Opinnäytetyö

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN TUTKINTO-OHJELMA
2022

Tekijä(t) Rantavalli, Petri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 4 2022
	Sivumäärä 50	Julkaisun kieli Suomi

Julkaisun nimi
PLC-datan keskitetty tiedonkeruu ja kyberturvallisuus

Tutkinto-ohjelma
Sähkö- ja automaatiotekniikka, tekniikan ammattikorkeakoulututkinto

Tiivistelmä

Tässä työssä toteutettiin prosessidatan tallennus ja vertailu automaatiojärjestelmän ja relaatiotietokannan välillä käyttäen hyväksi Siemens PLC:n omia kirjastoja. Samalla asiaa tarkasteltiin myös käytännön ja kyberturvallisuuden näkökulmasta. Työn toisena tavoitteena oli tuottaa samalla toiminnallinen kokonaisuus, jota voitaisiin hyödyntää myöhemmin myös opetuskäytössä.

Työ toteutettiin laatimalla aluksi dokumentaatio, jossa kuvaillaan järjestelmän toiminta sekä kerrotaan kyberturvallisuudesta ja siitä millaisia riskejä laitteiston käyttöönottoon liittyy ja miten niitä voidaan minimoida tai poistaa. Lopuksi laitteisto koottiin, ohjelmoitiin ja testattiin SAMK.n automaatiolaboratoriossa. SAMK myös vastasi tarvittavan SQL-palvelimen toimittamisesta.

Avainsanat
teollisuusautomaatio, tietokannat, ohjelmoitavat logiikat, SQL, Siemens, PLC, relaatiotietokannat, kyberturvallisuus.

Author(s) Rantavalli, Petri	Type of Publication Bachelor's thesis	Date 4 2022
	Number of pages 50	Language of publication: Finnish

Title of publication
Centralized PLC data collection and cyber security

Degree programme
Bachelor of Engineering, electrical and automation engineering

Abstract

In this work collecting and comparing of a process data, between the PLC and a relational database by using existing libraries of a Siemens PLC, was accomplished. While doing so the system was also examined from a perspective of a practical application and that of a cyber security. A secondary goal was to produce a functional automation system that could be later used for educational purposes.

Work was done by first creating a documentation that describes the system functions, explains about cyber security and risks involved when commisioning the equipment and how to minimize those risks. Finally the system was assembled, programmed and tested at the SAMK automation laboratory. SAMK also provided the necessary IT-infrastructure for this project.

Keywords
Industrial automation, data base, programmable logic controller, SQL, Siemens, PLC, relational databases, cyber security.

ALKUSANAT

Kiitokset Lehtori Hannu Asmalalle työn ohjaamisesta sekä lehtori Timo Suvelalle opetuksesta ja korvaamattomista vinkeistä työn kuluessa, sekä Satakunnan ammatti-korkeakoululle, joka tarjosi tarvittavat ohjelmistot ja laitteet.

SISÄLLYS

SYMBOLI- JA LYHENNELUETTELO

- PLC - Programmable Logic Controller eli ohjelmoitava logiikka on laite, jolla ohjataan teollisuuden automaatioprosesseja.
- SQL - Structured Query Language (*SQL*) on standardoitu kyselykieli, jolla relaatiotietokantaan voidaan tehdä erilaisia hakuja, muutoksia ja lisäyksiä.
- HMI - Human Machine Interface tarkoittaa käyttöliittymää, joka on tavallisesti ohjelmoitava kosketuspaneeli tai – tietokone, termillä voidaan viitata kokonaisuutena myös muihin hallintalaitteisiin.
- Tagi eli tag on PLC ohjelmoinnissa käytetty nimitys varatulle muistialueelle. Valmistajasta riippuen kutsutaan myös ”muuttujaksi (variable)” tai ”symboliksi (symbol)”.
- RFID - Radio-Frequency IDentification eli radiotaajuinen etätunnistus on tekniikka, joka välittää viestejä käyttämällä sähkömagneettista säteilyä. Järjestelmä koostuu lähetin/vastaanottimesta ja tunnisteesta eli ns. saattomuistista, joka voi olla vaikkapa vaatteissa oleva tarralappu tai bussikortti. Kuluttajatavaroissa olevat tarralapputunnisteet ovat useimmiten vain luettavissa eikä niissä olevaa tietoa voida ylikirjoittaa. Kutsutaan toisinaan myös englanninkielisellä nimellä transponder.
- Bitti eli bit ilmaisee tilaa 1 tai 0
- Byte eli tavu koostuu 8 bitistä, jotka voivat olla 1 tai 0 eli yhteensä 2^8 = 256 eri tilaa.
- Tulo- ja lähtöporttien kautta ohjataan PLC-järjestelmiä. Digitaaliset signaalit käyttäytyvät kuten kytkimet, ne ilmaisevat päällä - tai poissa tilan (1 tai 0, tosi tai epätosi). Analoginen signaali ilmaistaan jännite- tai virtatasoina, 4–20 mA ja +-10 V ovat yleisesti käytetyimmät viestialueet.
- IEC 61131-3 on standardi, jossa määritellään PLC käyttämät ohjelmointikielet. Kieliä on virallisesti 5 mutta eri valmistajilla on käytössä myös suljettuja ohjelmointikieliä. Viralliset kielet ovat FBD (*Function Block Diagram*), LD (*Ladder Diagram*), ST (*Structured Text*), IL (*Instruction List*) ja SFC (*Sequential Function Chart*).
- IP-osoite eli Internet Protocol on yksilöivä osoite, jonka perusteella laite tunnistetaan Internetissä tai paikallisessa verkossa.
- SCADA eli Supervisory control and data acquisition eli valvomo-ohjelmisto, johon erilaiset kenttälaitteet kuten PLC ja HMI yhdistetään. Windows PC pohjaisena ohjelmistona myös SCADA on altis kyberhyökkäyksille.

1 JOHDANTO

PLC eli ohjelmoitava logiikka on jo 70-luvulla suunniteltu laite kömpelöiden releaulujen ja logiikkapiirikorttien korvaajaksi ja siten useimmiten ohjannut lähinnä yksittäisiä laitteistoja tai prosesseja ja sen eri toiminnot ja liitynnät ovatkin kehittyneet erityisesti tukemaan yksittäisessä laitteistossa ja suljetussa verkossa toimimista. Viime vuosikymmenten aikana tietoverkkojen, ja etenkin ethernet-pohjaisen Internetin kasvun myötä, on PLC-laitteistojen verkottuminen osaksi laajempaa Internetiä arkipäiväistynyt, samalla monet teollisuuden tiedonsiirto- ja verkkoprotokollat, jotka aiemmin olivat sarjaliikenteeseen perustuvia, ovat muuttuneet ethernet-pohjaisiksi ja siten ”Internet kelpoisiksi”. Tämä kehitys on, paitsi tuonut teollisuusautomaatioon aivan uusia mahdollisuuksia (etähallinta ja -valvonta, -ohjelmistopäivitykset yms.), myös luonut uusia uhkia kyberturvallisuudelle kun kuka tahansa ympärimaailman voi nyt yrittää kirjautua sisään teollisuuden ohjausjärjestelmiin. Tässä dokumentissa kuvataan järjestelmä, jossa tietoa voidaan siirtää verkon yli PLC:n ja tietokantapalvelimen välillä ja samalla tuodaan myös konkreettisesti esille tällaisen salaamattoman Internet-liikenteen haavoittuvuus kyberhyökkäyksille. Kaikki verkkoliikenne tässä työssä on salaamatonta mikä aiheuttaa palvelimelle selkeän kyberturvallisuus uhan, jonka torjumiseen on useita keinoja. Tässä dokumentissa on käytössä se yksinkertaisin eli rajoitetaan pääsyä avoimesta verkosta tai eristetään laitteisto siitä kokonaan. Verkkoliikenne on mahdollista salata käyttämällä eri protokollia (esim. Modbus ja OPCUA) ja/tai salausavaimia mutta niiden soveltuvuus PLC - PC kommunikointiin on usein monimutkaista ja toisinaan hidastakin, joten useimmiten PLC itsessään ei keskustele suoraan palvelimelle vaan se hoidetaan muilla tavoin. Yleisimmät tavat ovat käyttää PLC:n sijasta SCADA-ohjelmistoa ja kosketusnäytöllistä paneeli PC:tä tai lähettää palvelinliikenne HMI:n kautta koska molemmat kykenevät käyttämään salausavaimia ja -sertifikaatteja palvelinliikenteessä. Eri PLC malleille on toki saatavilla salattua liikennettä käyttäviä kommunikaatiokirjastoja, joiden hinnat vaihtelevat toimittajan mukaan. Tässä dokumentissa on kuitenkin käytössä Siemensin vakio kommunikaatiokirjasto, joka ei siis salaa verkkoliikennettä mitenkään. PLC:n heik-

koon kybersuojaukseen ovat vaikuttaneet paitsi tekniset asiat myös vahvistusharha, että PLC ohjelmisto olisi erityisen suojassa, ollessaan kokonaan suljettu ohjelmisto ja laitteisto.

Viimeistään Stuxnet verkkomato, joka levisi lopulta myös Internetiin hyökättyään aluksi Iranin ydinkoelaitokseen, on todistanut väitteen vääräksi. Laboratorioiden PLC-laitteisto ei ollut yhteydessä avoimeen Internetiin, mikä omalta osaltaan todisti myös sen, ettei pelkästään laitteiston eristäminen Internetistä tee siitä yksinään kyberturvallista vaan poistaa vain helpoimman hyökkäysreitin. Kyberturvallisuus on siis kaikkien työntekijöiden vastuulla, ei vain teknisen henkilökunnan. Tutkinnan johtopäätöksenä 2012 oli, että joku valtiollinen toimija oli antanut työntekijälle salaa saastutetun USB-muistitikun, joka sitten vastoin ohjeita kytkettiin laboratorion tietokoneisiin. (Stuxnet worm hit industrial systems, 2010; Israel testasi Stuxnet-matoa ennen verkkohyökkäystä Iraniin, 2012).

PLC valmistajat ovat tapauksen jälkeen alkaneet kiinnittää erityistä huomiota kyberturvallisuuteen ja esimerkiksi tässä dokumentissa käytetty TIA-Portal v17 ohjelmisto sisältää huomattavan määrän lisäyksiä kyberturvallisuuteen verrattuna edelliseen v16 versioon.

2 JÄRJESTELMÄN TOIMINTA

2.1 Laitteisto

Laitteisto koostuu ohjelmoitavasta logiikasta (PLC), käyttöpaneelista (HMI) ja tietokantapalvelimesta, sijoitettuna SAMK - Automaation laboratorioon, jonne pääsy ulkopuolisilta on estetty. Laitteisto ei myöskään keskustele tai ole suoraan yhteydessä Internetiin vaan se on liitetty kytkimen kautta suoraan palvelimeen. Teknisesti järjestelmä voidaan kuitenkin skaalata lähes mihin tahansa käyttötarkoitukseen, kuten robottisoluissa tarttujan vaihtamiseksi työkappaleen mukaan, varastokirjanpitoon tai eri työvaiheiden automaattiseen kirjanpitoon.

Työssä käytettiin seuraavia laitteita ja ohjelmistoja:

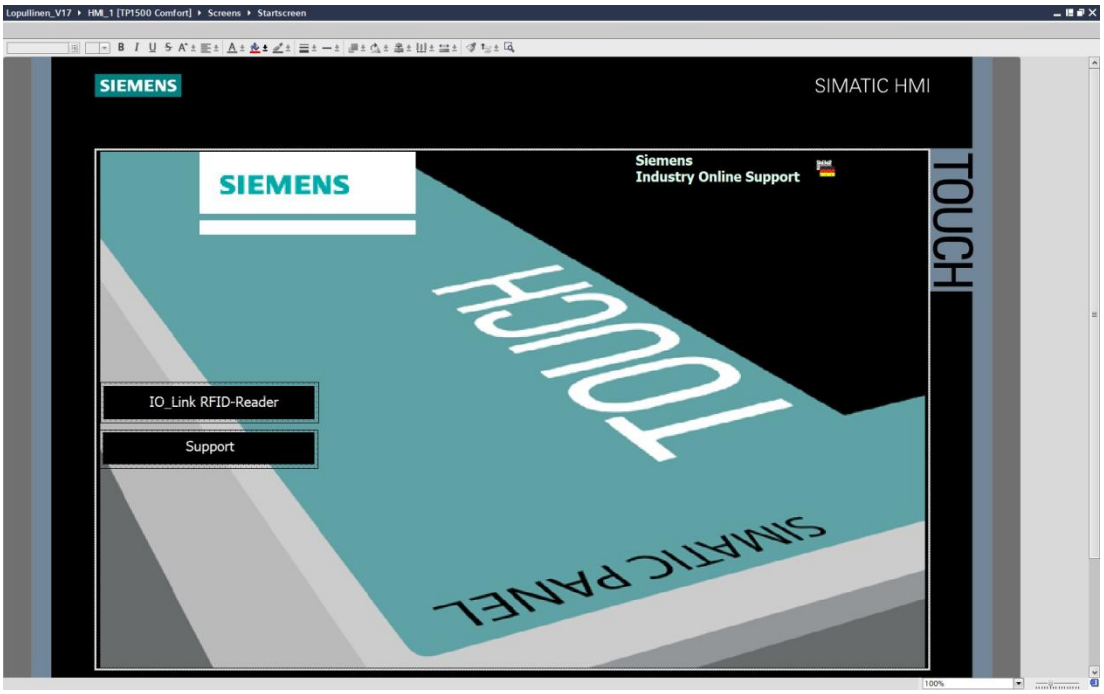
Palvelin: Microsoft SQL-server 2019
https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019
PLC: Siemens ET200SP CPU 1512SP F-1 PN
https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10255636#
, jossa CM 4xIO-Link moduuli
https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7137-6BD00-0BA0
Lukija: Siemens RF220R IO-Link v1.1
https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6GT2821-2BC32
HMI: Siemens TP1500 Comfort 15”
https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6AV2124-0QC02-0AX1

Ohjelmiston pohjana käytetään soveltuvilta osin Siemensin ohjelmointi esimerkkejä:

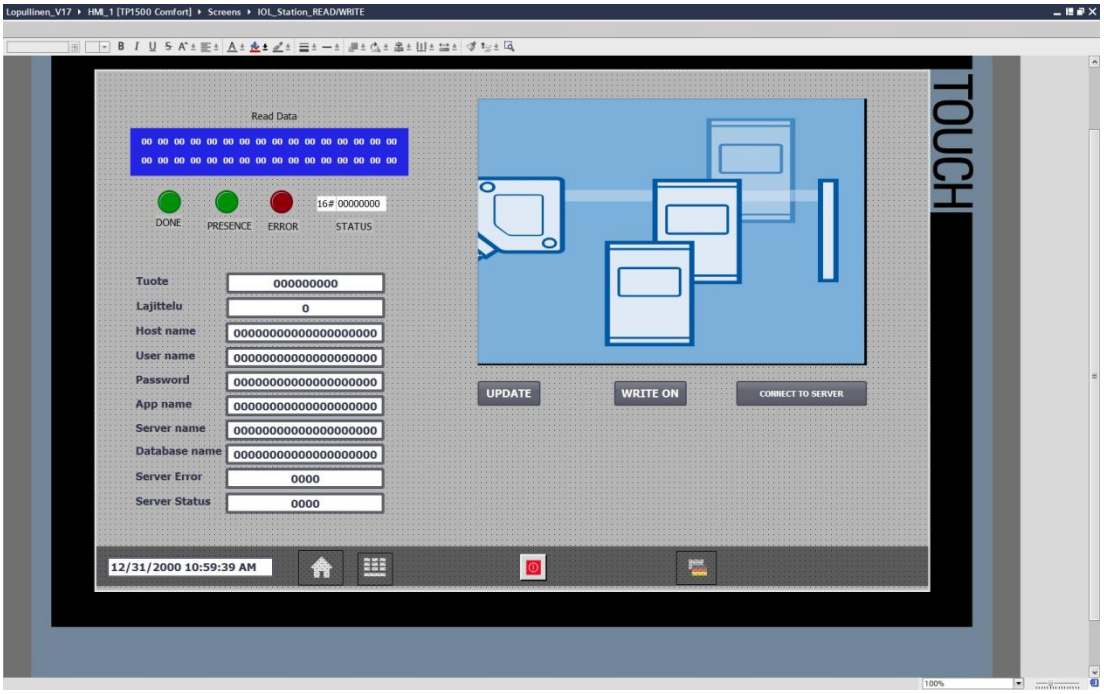
Connecting a S7-1500 to a SQL Database
https://support.industry.siemens.com/cs/ww/en/view/109779336

Reading and Writing RFID Data with SIMATIC S7-1500 via IO-Link
https://support.industry.siemens.com/cs/ww/en/view/73565887

MERKITTÄVÄ KYBERTURVALLISUUS RISKI, TÄTÄ OHJELMAA EI TULE KÄYTTÄÄ TUOTANTOYMPÄRISTÖSSÄ ENNEN RIITTÄVIÄ VAROTOIMIA! (Katso kappale 5).



Kuva 1 HMI aloitusnäkymä



Kuva 2 käyttöliittymä

Laitteistossa on kuljettimia, joille RFID - lukijan ohitse kulkevat kappaleet ohjataan, yhdelle radalle menevät kappaleet joita ei tunnisteta, muut jatkavat matkaansa oikeille radoille. Kappaleen tunnistus tapahtuu vertaamalla saatua RFID - koodia SQL-tietokannassa oleviin koodeihin.

2.2.1 Käyttöliittymän toiminta

Laitteistoa käytetään HMI - paneelin kautta, joka ilmoittaa käyttäjälle luetun RFID - tunnisteen, tuotteen nimen, ja radan jolle tuote halutaan ohjata. PRESENCE valo ilmaisee, että tunniste on lukijalla. DONE ilmaisee, että luku/kirjoitus prosessi on valmis ja kuljetin voi siirtyä eteenpäin. ERROR ilmaisee, jos lukijassa on häiriö.

Järjestelmässä tulee huomioida seuraavat seikat:

- RFID - tunnisteet voi poistaa palvelimelta vain pääkäyttäjän oikeuksilla.
- RFID- tunnisteet ovat yksilöiviä.
- Tunnisteisiin ei talleteta mitään vaan kaikki tieto on SQL-tietokannassa.
- Lukeminen tapahtuu automaattisesti kun tunniste on lukijalla.

Jotta järjestelmä voidaan ottaa käyttöön pitää HMI - kenttiin antaa seuraavat tiedot:

- Host name: Isännän nimi (ei pakollinen)
- User name: Käyttäjä nimi (pakollinen)
- Password: Palvelimen salasana (pakollinen)
- App name: Sovelluksen nimi (ei pakollinen)
- Server name: Palvelimen nimi (pakollinen)
- Database name: Tietokannan nimi (pakollinen)

Näiden lisäksi voidaan tarvittaessa täyttää myös kentät Tuote ja Lajittelu.

Kun kentät on täytetty paina ”connect to server” – painiketta, virhetilanteissa palvelimen tila ja virhekoodi voidaan lukea kentistä ”server error” ja ”server status” joista:

- 7000 Odottaa käskyä
- 7002 Palvelinyhteys muodostettu
- 8602 yhteyttä ei voida muodostaa tai yhteysvirhe

Muut TCON liittyvät virhekoodit löydät Siemensin Online oppaasta https://support.industry.siemens.com/cs/mdm/109773506?c=105138078347

2.2.2 Uuden tunnisteen lisääminen tietokantaan

Paina näppäintä WRITE ON, täytä kentät Tuote ja Lajittelu ja tuo tunniste lukijan eteen. Kun, olet valmis paina näppäintä uudelleen. Huomaa, että jo olemassa olevia tunnisteita ei voi lisätä järjestelmään.

2.2.3 Lisätyn tunnisteen muuttaminen

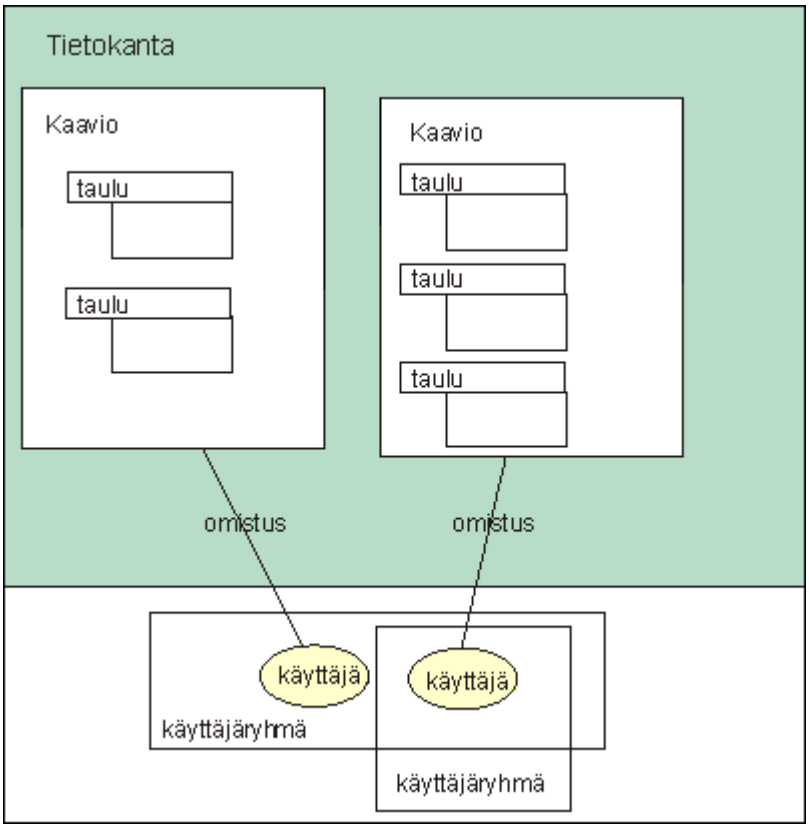
Täytä ensin kentät Tuote ja Lajittelu, vie tunniste lukijalle ja paina sen jälkeen UPDATE näppäintä. Tunnisteita ei voi HMI:n kautta poistaa järjestelmästä, mutta siihen liitettyjä tietoja voidaan vapaasti muokata.

3 SQL-TIETOKANNAN RAKENNE

SQL-tietokanta muodostuu yhden tai useamman kaavion (*schema*) määrittelemistä tauluista (*table*). Taulu vastaa relaatiomallin relaatiota, mutta

- Sallii etenkin kyselyiden tuloksissa samanlaisen rivin toistumisen useaan kertaan.
- Kyseessä ei ole siis matemaattinen relaatio vaan ns. monijoukko (multiset).

Taulu muodostuu riveistä (*row*). Rivit vastaavat relaatiomallin monikkoja. Taulussa on vähintään yksi sarake (column). Sarakkeille on annettava nimet. Jokaiseen sarakkeeseen liittyy arvojoukko (domain). Arvojoukot perustuvat SQL:n perustietotyyppeihin. SQL-92 standardin mukaisesti arvot ovat atomisia, kuten relaatiomallissa. Sen sijaan SQL-99 sallii myös rakenteiset arvot eli sellaiset arvot, jotka voivat jakautua nimettyihin osiin. SQL-99 mahdollistaa myös omien tietotyyppien määrittelyn.
Tauluja on kahden tyyppisiä: perustauluja (base table) ja johdettuja tauluja (derived tables, views). Perustaulut ovat aidosti olemassa apumuistille tallennettuina. Johdetut taulut määritellään kyselyjen avulla. Niitä voidaan tietokantaoperaatioissa käyttää miltei perustaulujen tapaan. (Helsingin Yliopisto.)



Kuva 3 Tyypillisen SQL-tietokannan rakenne (Helsingin Yliopisto)

Kullakin kaaviolla on omistaja (owner), joka omistaa myös kaavion määrittelemät taulut. Omistaja identifioidaan käyttäjätunnuksella (user account). Käyttäjätunnus voidaan liittää useaan käyttäjäryhmään. Sekä käyttäjälle että käyttäjäryhmälle voidaan myöntää erilaisia oikeuksia tietokantaan liittyen.

Omistajalla on kaikki oikeudet omistamiinsa tauluihin liittyen. Oletusarvoisesti muilla käyttäjillä (pääkäyttäjiä lukuun ottamatta) ei ole pääsyä muiden käyttäjien tauluihin. Omistaja voi kuitenkin antaa muille käyttäjille tai käyttäjäryhmille oikeuksia tauluihinsa.

Tauluihin voi liittyä erilaisia eheysehtoja (constraint). Nämä voivat koskea yksittäisiä taulun rivejä, koko taulua tai jopa useita tauluja. Sellaisia tietokantaan kohdistuvia muutoksia, jotka rikkoisivat eheysehtoja, ei hyväksytä. (Helsingin yliopisto.)

3.1 SQL-kielen kirjoitusasu

SQL-kielen kirjoitusasu on vapaa. Tämä tarkoittaa sitä, että kielen avainsanat, kaavioiden, taulujen ja sarakkeiden nimet voidaan kirjoittaa joko suur- tai pienaakkosilla tai sekamerkein.

Esimerkiksi 'select nimi, Osoite from henkilo ' on merkitykseltään sama kuin 'SELECT NIMI, OSOITE from HENKILO '.
Perinteisesti kaikki viittaukset taulujen tietosisältöön on kuitenkin pitänyt kirjoittaa talletusasun mukaisina. Täten

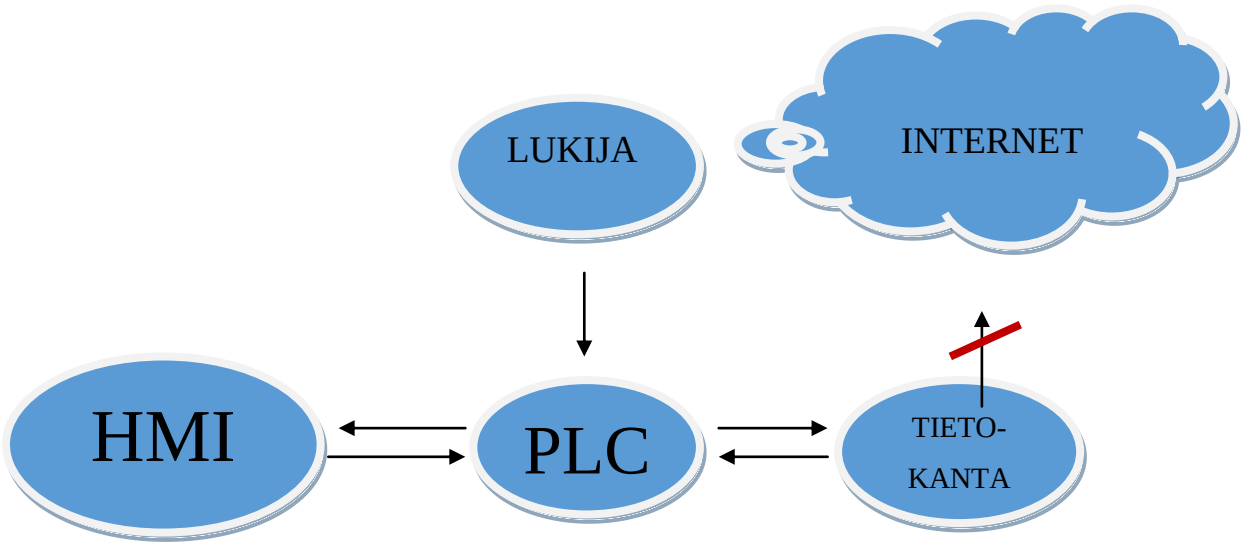
select osoite from henkilo where nimi like 'Kalle%'

tuottaa eri tuloksen kuin

select osoite from henkilo where nimi like 'KALLE%'

Joissain järjestelmissä kirjoitusasukäyttäytymistä voidaan säädellä asetusparametrein. (Helsingin Yliopisto.)

4 JÄRJESTELMÄN RAKENNE



Kuva 4 Järjestelmän rakenne

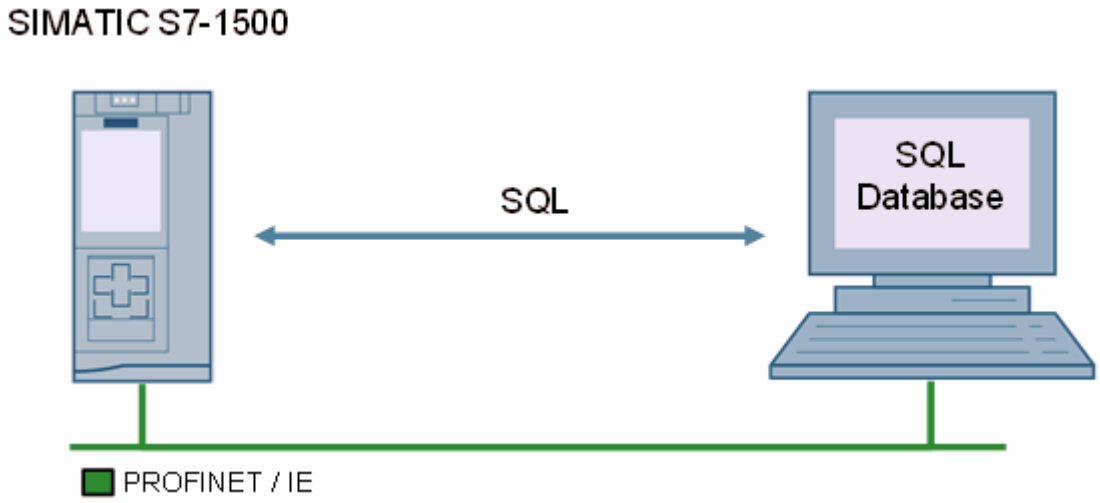
4.1 Järjestelmän osien yhteydet

Järjestelmä ja sen eri osien yhteydet esitetään kuvassa 4 ja itse laitteet on lueteltu kappaleessa 1. Nuolet kuvastavat informaation kulkua järjestelmän eri osien välillä.

Lukija lukee saapuvan RFID - tunnisteen (koodin) ja siirtää tiedon PLC:lle, joka vertaa saatua tietoa tietokantaan lähettämällä palvelimelle SQL-kyselyn ja kyselystä saatu palvelinvastaus ohjaa PLC:n lähtöjen toimintaa. HMI toimii rinnakkaisesti PLC:n kanssa ja PLC päivittää HMI - näyttöä aina sen hetkisen tilanteen mukaan. HMI voi myös erikseen käskeä PLC:tä muodostamaan halutun SQL-lausekkeen (tunnisteiden tietojen muutokset kantaan), jotka PLC lähettää edelleen palvelimelle eli tietokanta keskustelee suoraan vain PLC:n kanssa. Tietokantapalvelin olisi mahdollista yhdistää erilaisiin pilvipalveluihin mm. data-analytiikkaa varten mutta tässä järjestelmässä sitä kuitenkaan käytetä.

4.2 Tabular Data stream protokolla

Tabular Data Stream protocol (TDS) mahdollistaa suoran yhteyden Microsoft SQL palvelimelle. Käyttämällä TDS:ää voit kirjautua SQL palvelimen tietokantaan ja lähettää SQL käskyjä, tämä mahdollistaa datan kirjoittamisen ja lähettämisen tietokantanasta sekä sen käyttämisen tietovarastona.



Kuva 5 Simatic järjestelmä (Siemens)

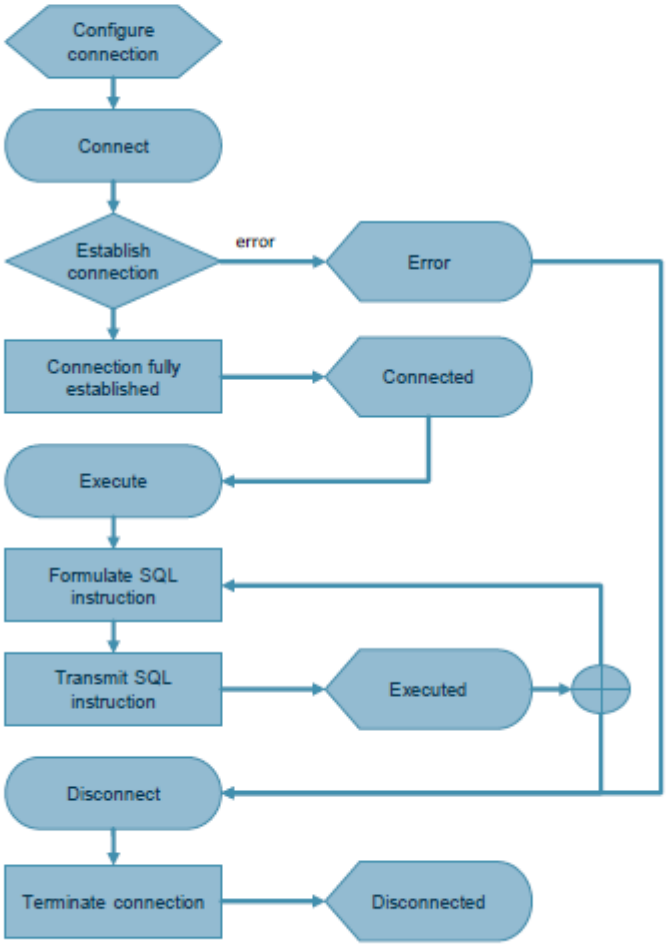
Perustuen "avoimiin käyttäjä kommunikointi blokkeihin" (TCON, TSEND, TRCV and TDISCON), S7-1500 ja S7-1200 pystyvät TDS protokollan avulla muodostamaan yhteyden Microsoft SQL palvelimelle. Käyttämällä käskyjä "insert into", "update" ja "select" voidaan tallettaa, päivittää sekä lukea palvelimella olevaa tietokantaa. (Connecting a S7-1200 / S7-1500 to a SQL Database, 2021).

4.3 Toimintalohkonkuvaus

1.2 Principle of operation

The following figure shows the principle of operation, structure and states of the function block "LSql_Microsoft".

Figure 1-2



Kuva 6 LSql_Microsoft blockin toimintakaavio (Siemens)

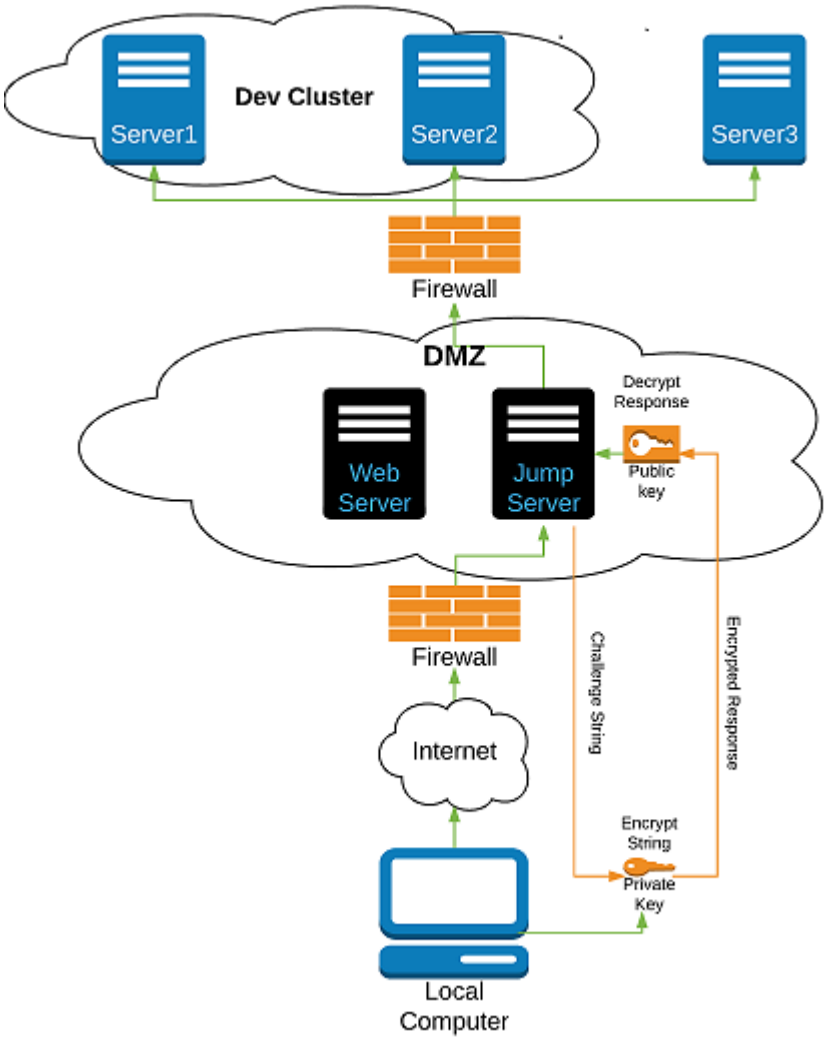
Yhteys palvelimelle tapahtuu siis vaiheittain:

- Aluksi Muodostetaan yhteys palvelimeen, asetukset määritellään yhteysblokissa.
- Yhteysvirhetilanteessa yhteys katkaistaan ja annetaan käyttäjälle ilmoitus ”ei voitu yhdistää palvelimelle”.
- Kun yhteys on saatu, muodostetaan yhteysblokissa oleva käsky.
- Käsky voidaan nyt, joko hylätä tai suorittaa.
- Lopuksi katkaistaan palvelin yhteys.

4.4 Kyberturvallisuudesta

TDS-protokollan avulla tapahtuva palvelinliikenne ei ole salattua, joten PLC:n liittämistä muutoin kuin suoraan SQL-palvelimeen (tai samaan aliverkkoon) ei voida suositella, ilman jotakin seuraavista toimenpiteistä:

- Salataan liikenne PLC:n ja palvelimen välillä, PLC ei kuitenkaan välttämättä tue tai kykene vahvaan SQL-salaukseen vaikka mm. OPC UA standardi mahdollistaa liikenteen salaamisen.
- Käytetään viestien välittäjänä PLC:n sijaan laitetta, joka kykenee liikenteen vahvaan SQL-salaukseen kuten HMI - paneelia tai paneeli - PC:tä, jossa on SCADA ohjelmisto.
- Eristetään koko järjestelmä avoimesta verkosta ja/tai sallitaan etäyhteys vain VPN-tunnelin kautta.
- Käytetään ns. hyppypalvelinta, josta liikenne ohjataan edemmäs sisäverkkoon.



Kuva 7 Hyppypalvelin

5 KYBERTURVALLISUUDEN YLEISET VAROTOIMET

Nykyaikaisessa teollisuusympäristössä tarve informaation laaja-alaiseen hyödyntämiseen kasvaa jatkuvasti, samalla kuitenkin sovelluskohtaiset standardit ja ohjeet tietoturvasta ja käytännön toteutuksista voivat olla joko puutteellisia tai niitä ei edes ole. Joissakin tapauksissa ohjaukseen liittyviä mittauksia ja prosessinohjauksia on tehty yleisten tietoverkkojen kautta, tämä ei ole kyberturvallisuuden kannalta hyvä ratkaisu.

Tietoturvauhat nykyisessä tietoyhteiskunnassa ovat laajat ja kohdistuvat kaikkiin järjestelmiin jotka ovat kytköksissä avoimeen verkkoon. Järjestelmän tietoturvan voi vaarantaa myös yrityksen oma henkilöstö omalla toiminnallaan antamalla väärinkäytöksellään ulkopuoliselle pääsyn yrityksen verkkoon, joko sisä- tai ulkopuolelta.

Hyökkääjän päästyä sisäverkkoon on heikkouksien löytäminen yleensä helppoa sillä järjestelmät suojataan usein lähinnä ulkoisilta uhilta, eikä sisäverkosta tuleviin uhkiin juurikaan varauduta. Tietoturva voi helposti vaarantua myös yksittäisten laitteiden osalta joiden sisään kirjautuminen on hoidettu automaattisen tunnistuksen avulla (selain keksit, VPN – sertifikaatit, ilman salasanoja yms.) tai käytetään ns. tehdasasetusten salasanoja.

Tietotekniikan yleiset lähestymistavat soveltuvat myös tuotannolliseen automaatioon mutta järjestelmien ominaispiirteet on otettava huomioon kuten rajalliset laskentaresurssit, reaaliaikaisuus ja tuotannon jatkuva käynnissä pito ja turvallisuusvaatimukset. Nykyään tietotekniikkaan liittyvä riskienhallinta on osa laitteiden kokonaisturvallisuuden riskiarviota. Riskienhallinnalla pyritään vähentämään, ei toivottujen tapahtumien toteutumista, ja pienentämään niiden haitallisia seurauksia.

Automaatioympäristössä on aina ollut tavoitteena painottaa ennaltaehkäisevästi tapaturmien ja muiden ei toivottujen tapahtumien syntymistä, tietoturvassa tämä on mahdollista vasta kun perusasiat ja käytännöt on luotu ja otettu käyttöön. tällaisia keinoja ovat mm.

- Suora yhteys yleisestä verkosta automaationtietojärjestelmiin estetään
- Palvelut rajataan tarkasti vain tarvittavaan käyttöön
- Kriittisten järjestelmien jatkuva seuranta
- Toimintahäiriöiden varalle laaditaan toipumissuunnitelmat
- Tietojärjestelmien ja laitteiden päivitykset ja huollot suoritetaan organisoidusti

5.1 Tietotekniset riskit

Ilmeisimmät riskit tietoverkoille ovat järjestelmään tunkeutuminen ja erilaiset haitta-ohjelmat. Riskit ja niiden toteutuminen riippuvat siitä millaista tietoa järjestelmissä käsitellään, sekä siitä millaisia ovat yrityksen käytössä olevat tietotekniset ratkaisut, näitä voivat olla mm. radio ja telemetriset laitteet ja ulkoiset palvelut, joita käytetään maantieteellisesti erillisten alueiden väliseen kommunikointiin. näihin järjestelmiin voidaan tunkeutua joko fyysisesti tai verkonvälityksellä mm.

- Valtuuttamaton henkilö pääsee käsiksi luottamuksellisiin tietoihin
- Väärentämällä tunnistautuminen
- Valtuutetun henkilön väärä tai jopa kielletty toiminta
- Tahallinen käytönesto

Erityisen haavoittuvia ovat usein mm.

- Internet- ja intranet – yhteydet
- Modeemit joissa ei ole takaisin soittoa tai salausta
- Langattomien yhteyksien solmukohdat
- Etätyöpisteiden ohjelmistot kuten VPN tunnelit ja asiantuntijaohjelmistot joilla on pääsy suoraan järjestelmiin, mukaan lukien järjestelmien etäkäyttö, kaukovalvonta ja mittaukset
- Kaikki yhteydet jotka eivät ole osa varsinaista automaatiojärjestelmää mutta ovat yhteydessä siihen (yrityksen sisäverkko)
- Verkkoyhteydet ja SCADA - järjestelmät, jotka eivät ole osana fyysisesti varmistettua tuotanto- tai ohjausjärjestelmäverkkoa

5.2 Haittaohjelmat

Tällaiset ohjelmat voidaan luokitella seuraavasti: virukset, Troijan hevoset ja roskapostitus sekä phishing. Tietokoneviruksiin kuuluvat alijoukkona erilaiset tietokonemadot ja troijalaisissa takaportit ja loogiset pommit. Myös erilaiset vakoiluohjelmat voidaan nähdä Troijalaisina. Haitallisilla ohjelmatyökaluilla toteutetaan mm. palvelunestohyökkäyksiä, käyttäjien suoraa vakoilua yms. Koodi voi olla myös tahattomasti haitallista, jolloin hyökkääjä pääsee hyödyntämään siinä olevia aukkoja.
Merkittävämpänä uhkana sisäverkoille ovat saastuneet tai sille alttiit ulkoa tuotavat kannettavat laitteistot kuten USB - tikut tai kannettavat tietokoneet jotka ovat samalla yhteydessä suojaamattomaan ulkopuoliseen verkkoon (julkinen WLAN).

5.3 Puskurin ylivuodot

Puskurin ylivuoto tarkoittaa ohjelmassa olevaa virhettä, jonka seurauksena ohjelman keskusmuistista varaamaa tilaa kirjoitetaan virheellisesti muille muistialueille. tällöin on vaarana, että ulkopuolinen pääsee käsittelemään normaalisti suojatulla muistialueella olevaa tietoa tai jopa kaatamaan koko järjestelmän, tällainen haavoittuvuus on myös hyvin yksinkertaista automatisoida mikä mahdollistaa laajatkin DDOS hyökkäykset.

5.4 toimistojärjestelmien ja teollisuuden automaatiojärjestelmien erot

Automaation tietojärjestelmiltä vaaditaan tavallisesti korkeampaa tietoturvantasoa kuin yleisissä toimistojärjestelmissä. Siksi automaatiojärjestelmiä voidaankin luonnehtia seuraavasti:

- Vakiintuneempia kuin toimistojärjestelmät; Organisaatio tunteet laitteet paremmin ja niiden käyttöikä on pidempi. Konfiguraatioita muutetaan vain erittäin harvoin
- Eivät yleensä sisällä liiketoimintojen kannalta merkittävää tietoa
- Suoraa Internet yhteyttä ei yleensä tarvita
- Laitteita ei tavallisesti käytetä muihin tarkoituksiin ja ovat useimmiten hajautettuja toiminnan mukaan
- Pääsy hallintaan on useimmiten tarkasti rajattua, henkilöstö on koulutettua ja valvonta tiukempaa

Lisäksi teollisuusautomaatiolla on usein seuraavia erityispiirteitä kuten:

- Ihmisten terveyden- ja turvallisuuden varmistaminen
- Järjestelmillä on tietty kokoonpano; tehty tiettyyn tarkoitukseen
- Tiedon ja palveluiden saatavuus; seisokit eivät ole hyväksyttäviä
- Ei-toivotut seuraukset; keskeytykset voivat olla paitsi kalliita jopa tuhoisia
- Aikakriittisyys; ohjaus ja turvatoiminnot, joissa viiveet saattavat olla kohtalokkaita, vaativat usein reaaliaikaisuutta
- Ohjelmistot saattavat vaatia erityistä asiantuntijuutta
- Salaus ei tai virhelokit eivät aina ole saatavilla tai mahdollisia toteuttaa
- Resurssit saatavat olla vaatimattomat tai niitä ei voida käyttää tietoturvaan
- Tiedon eheys on järjestelmän toiminnalle ehdottoman tärkeää
- Yhteydet eivät välttämättä ole suoraan yhteensopivia ns. kuluttajalaitteiden kanssa
- Päivittäminen on monimutkaista ja sen vaikutus järjestelmään on aina ensin arvioitava.

5.5 Käytön hallinta

Tietoturvan keskeisenä tarkoituksena on varmistaa automaatiojärjestelmän oikea ja turvallinen toiminta, sekä suojata järjestelmässä olevat tiedot. Näistä keskeisenä toimintana on estää asiattomien pääsy laitteisiin ja palveluihin sekä seurata kaikkea pääsyä ja niiden yrityksiä näihin järjestelmiin. Käyttäjien tietoturva voidaan jaotella kolmeen A:han, tunnistukseen (Authentication), valtuutukseen (Authorisation), sekä seurantaan (Accounting). Tunnistus tarkoittaa järjestelmään kytkettävän laitteen tunnistamista jollakin luotettavalla tavalla (kulun valvonnan RFID, henkilökortti, verkkoyhteyden sertifikaatti yms.). Valtuutus tarkoittaa pääsynrajoittamista vain tiettyihin tietoteknisiin palveluihin tai rakennuksen osiin. Seuranta tarkoittaa jäljitettävyyttä aina työasemalle kirjautumisesta fyysiseen kulunvalvontaan, jotta palvelun tai tilan käyttäjä on aina jäljitettävissä.

6 OHJELMOINTI

Järjestelmään kuuluvat laitteet ja ohjelmistot on listattu kappaleessa 1.1 ja ne vaativat asennuksen ennen ohjelmoinnin aloittamista. Varmista siis ennen ohjelmoinnin aloittamista, että kaikki kytkennät ovat oikein, myös TIA – portalissa, ja että kaikki tarvittavat ohjelmistot ja tuotekirjastot on asennettu.

6.1 Tarvittavat kirjastot

TIA-portalissa on oltava asennettuna laitteiston tuotekirjastot, jotta RFID-tunnisteen luku ja sen tallettaminen SQL-palvelimelle olisi mahdollista:

RFID-lukijaa varten tarvitaan IO-link kirjasto:

- LIOLink_LIB_V6_0 artikkeli nro. 82981502
  https://support.industry.siemens.com/cs/document/82981502/library-for-io-link-(liolink)?dti=0&lc=en-WW

SQL-palvelinliikennettä varten tarvitaan Tabular Data Stream protocol (TDS):

- SQL_S7_1500_CODE_V21 artikkeli nro. 109779336
  https://support.industry.siemens.com/cs/document/109779336/connecting-a-s7-1200-s7-1500-to-a-sql-database-?dti=0&lc=en-WW

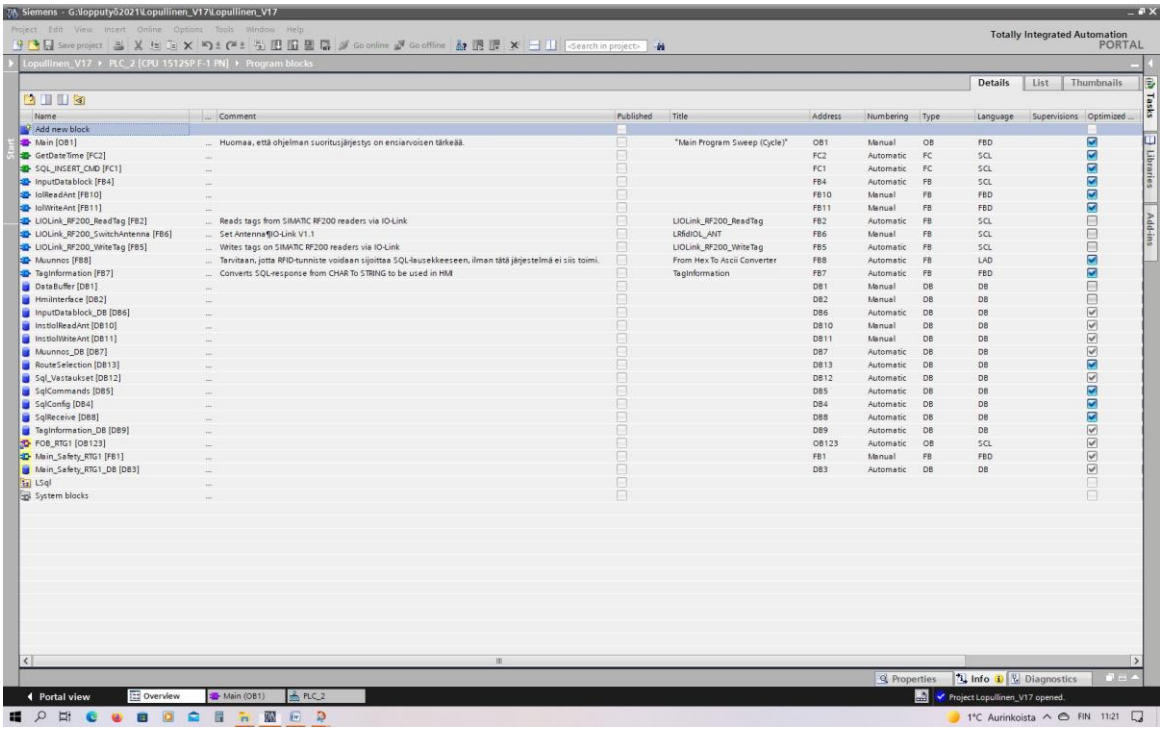
Lisäksi käytämme hyväksi Siemensin julkaisemaa valmista esimerkkiä:

- Reading and Writing RFID data with SIMATIC RF200 IO-Link readers and SIMATIC S7 controllers artikkeli nro. 73565887
  https://support.industry.siemens.com/cs/document/73565887/reading-and-writing-rfid-data-with-simatic-rf200-io-link-readers-and-simatic-s7-controllers?dti=0&lc=en-US
  Huomaa, että Esimerkissä oleva 10” Comfort paneeli on vaihdettu TP1500 malliin ja päivitetty samalla TIA v.17 yhteensopivaksi.

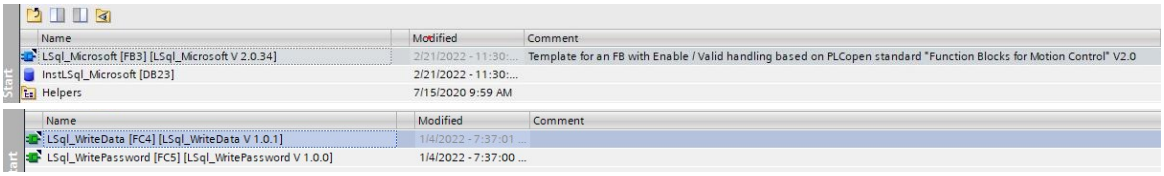
On myös syytä varmistaa, että käytetyn laitteiston firmware on päivitetty yhteensopivaksi käytetyn TIA version kanssa. Päivittämistä varten on suoritettava maksuton rekisteröinti Siemensin tukiportaaliin.

Kun kirjastot on asennettu TIA - portaliin, kopioidaan SQL- ja RFID - esimerkit projektiin niissä olevien ohjeiden mukaisesti. SQL-palvelin rakennetaan kappaleen 5 mukaisesti.

6.2 Ohjelmalohkot



Kuva 8 PLC-ohjelma koostuu funktioista



Kuva 9 sekä erilaisista apufunktioista

6.2.1 Lohkojen kuvaukset

Main eli pääohjelmassa (MAIN) kutsutaan ohjelmaan luotuja funktioita. Pääohjelmia voi olla toiminnassa samaan aikaan useita rinnakkain, funktioita voidaan siirtää pääohjelmaan hiirellä vetämällä. Funktiot joita ei kutsuta lainkaan pääohjelmissa tai aliohjelmissa eivät osallistu ohjelmakiertoon.
Funktiot(F) ja funktioblokit(FB) eroavat toiminnallisuudeltaan siten, että funktioblokissa olevat arvot voidaan säilyttää funktioblokin suorituksen jälkeen. Funktion arvot taas ovat hetkellisiä eli säilyvät vain funktion suorituksen ajan. Datablokkeihin talletetaan funktioista ja funktio blokeista tulevat muuttujat, jotka ovat joko globaaleja eli muut blokit voivat viitata niihin tai yksittäiseen funktioblokkiin sidottuja.

6.2.2 PLC Tagit

Tag tarkoittaa viittausta tiettyyn muistialueeseen eli se voi olla yleisemmin esim. Boolean tyyppinen bitti 1 tai 0 tai vaikka jokin lukuarvo. Kun tag luodaan varaa se käyttöönsä PLC:n muistista datatyyppinsä kokoisen alueen. Esim. WORD tyyppinen muuttuja varaa muistia aina 16bitin kokoisen alueen, joka alkaa sille osoitetusta muistisoitteesta. Yleisemmin käytetyt M, I ja Q ovat 8 bitin tavuja eli jakaantuvat 8 osaan (I0.0 … I0.7, Q1.0 … Q1.7, M3.2 jne.) yleisimmät datatyypit ominaisuuksineen on lueteltu liitteessä 2: Step 7 Elementary Data Types.

Lisätään [Default Tag Tableen] seuraavat rivit:

Taulukko 1 lisättävät rivit:

Name	Path	Data	Logical	Comment	Hmi	Hmi Ac-	Hmi Wri-

		Type	Address		Visible	cessible	teable
IOLink-RFID lukija 1	Default tag table	Bool	%I11.5		True	True	True
Peeker_Byte	Default tag table	Char	%MB7		True	True	True

6.2.3 Funktio SQL_INSERT_CMD (SCL)

Funktio muodostaa annetuista syötteistä palvelimelle lähetettävät komennot.

		Name	Data type
1		Input	
2		SQL_INSERT	String
3		SQL_SELECT	String
4		SQL_UPDATE	String
5		Timestamp	String
6		SignalName	String
7		SignalValue	String
8		Product	String
9		Sorting	String
10		Output	
11		SQL_INSERT_CMD	String
12		InOut	
13		HMI_Write	Bool
14		HMI_Read	Bool
15		HMI_Update	Bool
16		Temp	
17		Valinta	Word
18		Value_Str	String
19		Constant	
20		<Add new>	
21		Return	
22		Ret_Val	Void

Kuva 10 Funktion sisälle tulevat muuttujat

Funktion rakentamisessa voisi käyttää myös CASE – rakennetta mutta HMI -ohjauksen vuoksi valinta toteutetaan IF lauseella.

// merkitään kommentit ohjelmakoodiin.

REGION jakaa ohjelman helpommin hallittaviin kokonaisuuksiin, muttei vaikuta muutoin ohjelman toimintaan.

IF... CASE... OF... FOR... TO DO.. WHILE.. DO... (*...*) REGION

```
REGION kirjoitus
IF #HMI_Write = 1 THEN

    // WRITE komento valittuna
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT, IN2 := '$'');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #SignalValue);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$','');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Timestamp);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$','');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Product);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$','');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Sorting);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
    // VAL_STRG(IN:=#SignalValue,SIZE:=10,PREC:=2,FORMAT:=4,P:=10,OUT=>#Value_Str);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Value_Str);
    #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := ')');
    ;
    END_IF;
END_REGION
REGION kyselyt
    IF #HMI_Read = 1 THEN
        ;
        // READ komento valittuna
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_SELECT, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #SignalValue);
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'= ID');
        ;
    END_IF;
END_REGION
REGION tietuiden muutokset
    IF #HMI_Update = 1 THEN
        ;
        // UPDATE komento valittuna
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_UPDATE, IN2 := '');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := ' PRODUCT_NAME = ');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Product);
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := ',');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := ' SORTING = ');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #Sorting);
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := ' WHERE ID = ');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := #SignalValue);
        #SQL_INSERT_CMD := CONCAT(IN1 := #SQL_INSERT_CMD, IN2 := '$'');
        ;
    END_IF;
END_REGION

// SELECT [ID],[PRODUCT_NAME],[SORTING]
// FROM [S7PLCSQLDB].[dbo].[PLCDATA]
// IF #HMI_Read = 1 THEN
// WHERE 'E008013E20CAB0B9000000' = [ID];
//
//UPDATE Customers SET
//ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1;
//
// UPDATE PLCDATA SET PRODUCT_NAME = #product, SORTING = #Sorting WHERE ID = #SignalValue;
//
//
```

Kuva 11 Funktio SQL_INSERT_CMD (SCL)

6.2.4 Funktio GetDateTime (SCL)

Muodostaa palvelimelle lähetettävän aikaleiman

GetDateTime

	Name	Data type
1	Input	
2	<Add new>	
3	Output	
4	<Add new>	
5	InOut	
6	<Add new>	
7	Temp	
8	RET_VAL_1	Int
9	dt	DTL
10	YEAR	UInt
11	MONTH	USInt
12	DAY	USInt
13	WEEKDAY	USInt
14	HOUR	USInt
15	MINUTE	USInt
16	SECOND	USInt
17	NANOSECOND	UDInt
18	strDateTime	String
19	strYear	String[4]
20	strMonth	String[2]
21	strDay	String[2]
22	strHour	String[2]
23	strMinute	String[2]
24	strSecond	String[2]
25	Constant	
26	<Add new>	
27	Return	
28	GetDateTime	String

Kuva 12 Funktion sisälle tulevat muuttujat

Aikaleima olisi mahdollista leimata myös HMI:ssä tai SQL-palvelimella mutta käytännössä se kannattaa leimata PLC:ssä käskyä muodostettaessa.

```
IF...  CASE... OF...  FOR... TO DO..  WHILE.. DO...  (*...*)  REGION

1  // Read time of the day
2  #RET_VAL_1 := RD_SYS_T(OUT=>#dt);
3  // Convert integer to string
4  VAL_STRG(IN:=#dt.YEAR,SIZE:=4,PREC:=0,FORMAT:=W#16#0,P:=1,OUT=>#strYear);
5  VAL_STRG(IN := #dt.MONTH,SIZE := 2,PREC := 0,FORMAT := W#16#0,P := 1,OUT => #strMonth);
6  VAL_STRG(IN := #dt.DAY,SIZE := 2,PREC := 0,FORMAT := W#16#0,P := 1,OUT => #strDay);
7  VAL_STRG(IN := #dt.HOUR,SIZE := 2,PREC := 0,FORMAT := W#16#0,P := 1,OUT => #strHour);
8  VAL_STRG(IN := #dt.MINUTE,SIZE := 2,PREC := 0,FORMAT := W#16#0,P := 1,OUT => #strMinute);
9  VAL_STRG(IN := #dt.SECOND,SIZE := 2,PREC := 0,FORMAT := W#16#0,P := 1,OUT => #strSecond);
10
11 // Format date and time string
12 #strDateTime := #strYear;
13 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := '-');
14 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := #strMonth);
15 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := '-');
16 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := #strDay);
17 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := ' ');
18 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := #strHour);
19 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := ':');
20 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := #strMinute);
21 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := ':');
22 #strDateTime := CONCAT(IN1 := #strDateTime, IN2 := #strSecond);
23 // Return date and time string
24 #GetDateTime := #strDateTime;
```

Kuva 13 Funktio GetDateTime (SCL)

6.2.5 Funktioblokki InputDatablock (SCL)

FB tutkii PEEK komennolla annettua Input I/O-aluetta ja siirtää siellä olevat tavut datapuskuriin ”DataBuffer".readData

I/O-linkistä tulevat bitit alkavat osoitteesta % IB11 ja niistä luetaan ensimmäisestä alkaen seuraavat 20 tavua, joka asetetaan main funktiossa.

Huomaa, että PEEK funktio on erikoistapaus siinä, että se vaatii toimiakseen globaalin muuttujan, sisäinen muuttuja ei toimi ja saattaa jopa kaataa ohjelman.

InputDatablock

		Name	Data type	Default value
1		Input		
2		Input	Int	0
3		Output		
4		InOut		
5		Static		
6		Temp		
7		i	Int	
8		Constant		

Kuva 14 Funktion sisälle tulevat muuttujat

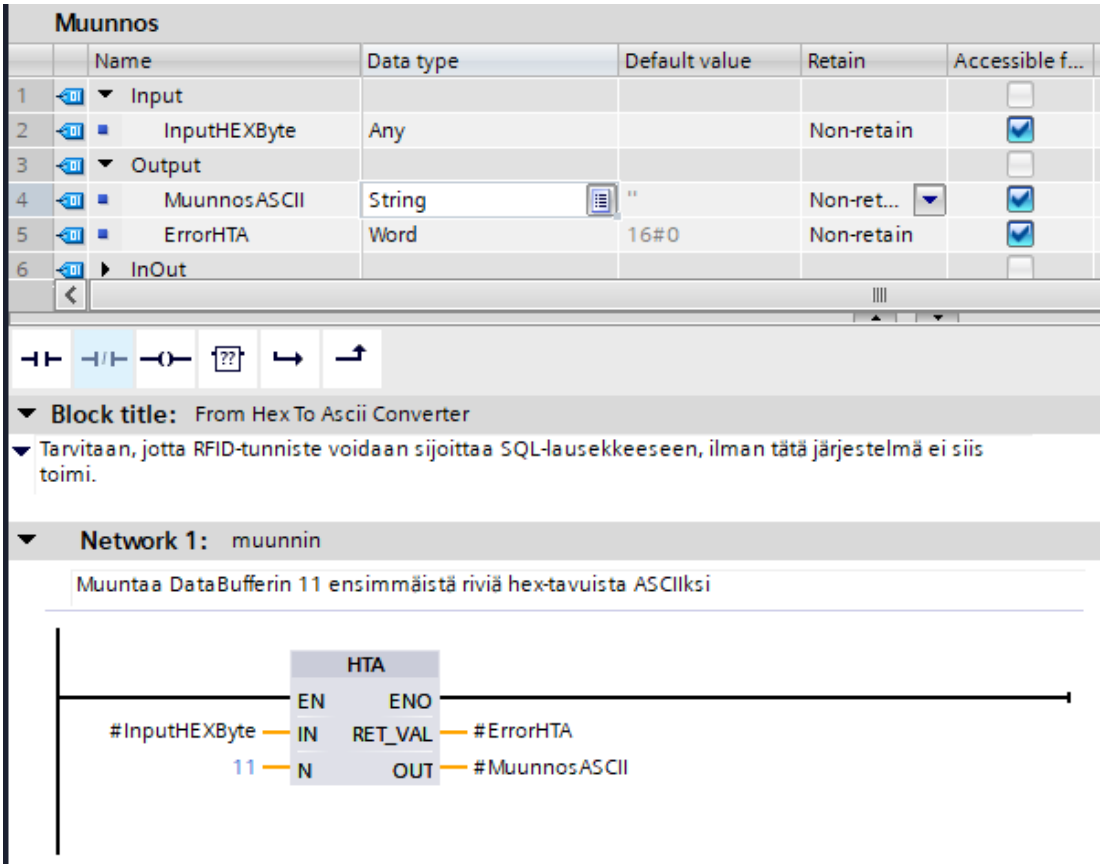
```
IF...  CASE... OF...  FOR... TO DO..  WHILE.. DO...  (*...*)  REGION

1       // FOR #i := 0 TO 11 BY 1 DO
2       // "DataBuffer".readData[#i] := #i;
3       // Esimerkki; täyttää data bufferin solut 0-11
4       // END_FOR;
5
6     //siirtää %IB ** arvot suoraan datapuskuriin
7     FOR #i := 0 TO #Input DO
8
9         "Peeker_Byte" := PEEK(area := 16#81,
10                              dbNumber := 0,
11                              byteOffset := #i + 11);
12        "DataBuffer".readData[#i] := "Peeker_Byte";
13
14    END_FOR;
15
```

Kuva 15 Funktioblokki InputDatablock (SCL)

6.2.6 Funktioblokki Muunnos (LAD)

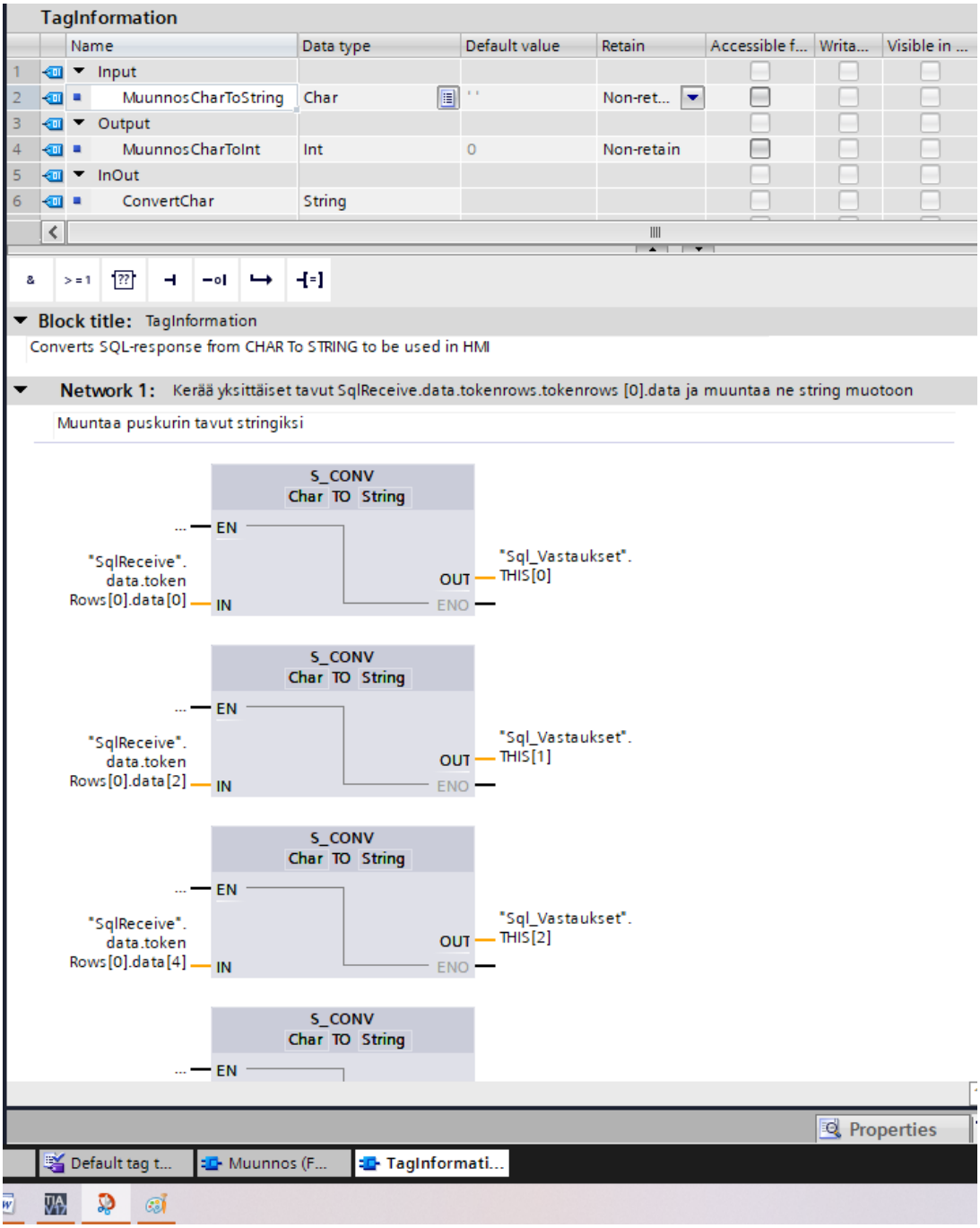
FB muuntaa DataBufferin 11 ensimmäistä riviä hex-tavuista ASCII merkeiksi.



Kuva 16 Funktioblokki Muunnos (LAD)

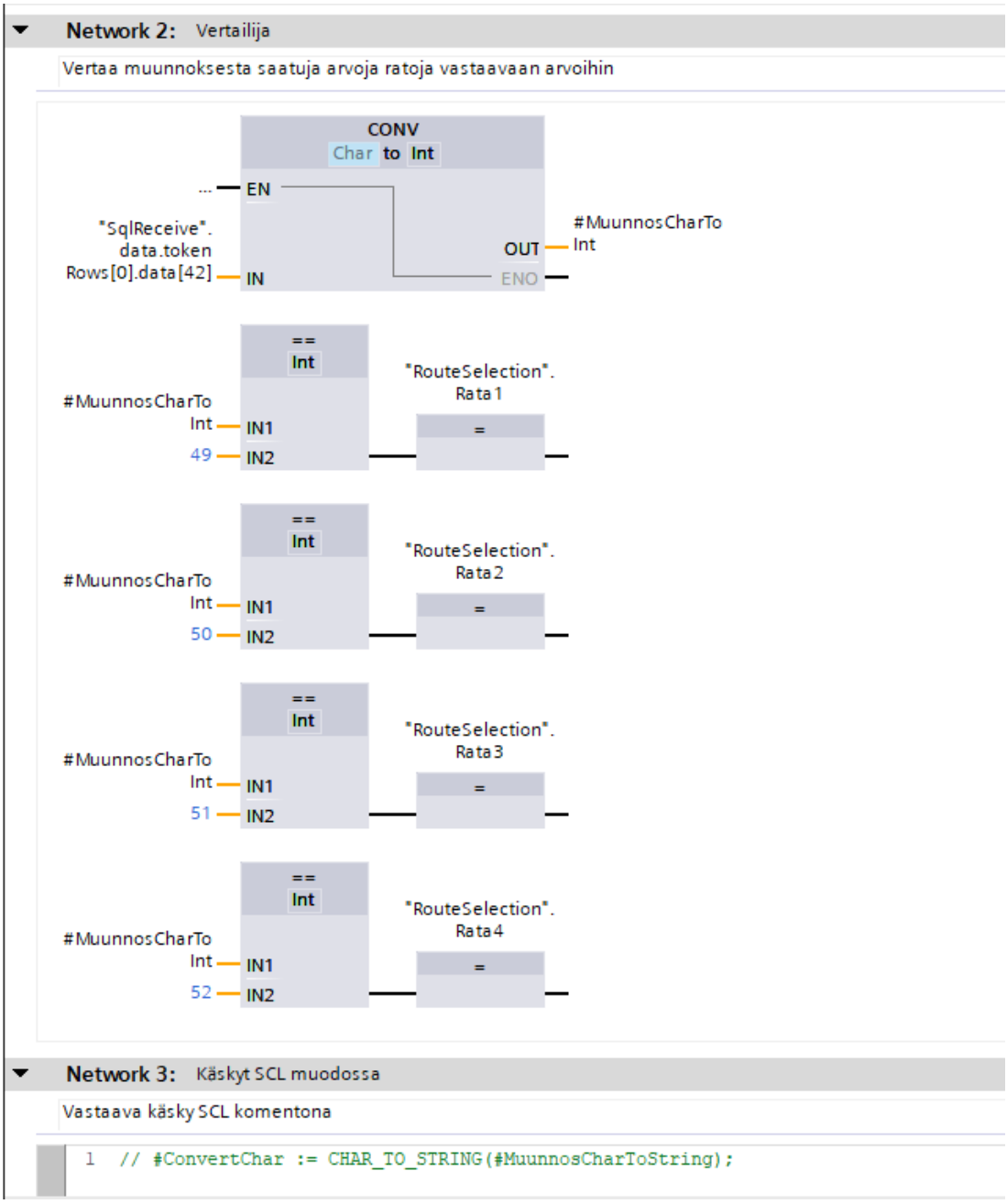
6.2.7 TagInformation (FBD)

FB Kerää yksittäiset tavut SqlReceive.data.tokenrows[*].data ja muuntaa ne string muotoon ja lopuksi vertaa muunnoksesta saatuja arvoja ratoja vastaavaan arvoihin. Tässä ohjelmassa tuotteet ohjataan radoille, mutta samoin voisi käynnistää minkä tahansa muunkin tuotantoprosessin. Muunnos suoritetaan 8 ensimmäisille merkkiriville hypäten joka toinen DB rivin yli. Sql_Vastaukset on Array tyyppinen DB, jolloin THIS on TIA - Portalin automaattisesti generoima rivinimike.



Kuva 17 TagInformation (FBD)

Loppuun tulee muunnoksen riviksi 42, jolta luetaan SQL-vastauksena saatu lajittelutunniste, tässä tapauksessa numero on char tyyppisenä, joten sille pitää suorittaa ylimääräinen muunnos CHAR TO INT, jotta vertailu toimii.



Kuva 18 vertailija

Kuvassa näkyy sama käsky myös SCL muodossa Networkissä 3. Huomaa, että muunnoksen jälkeen vertailuarvo alkaa luvusta 48 siis 48 + 1 = 49, 48 + 2 = 50 jne.

6.2.8 Siemensin toimittamat lohkot

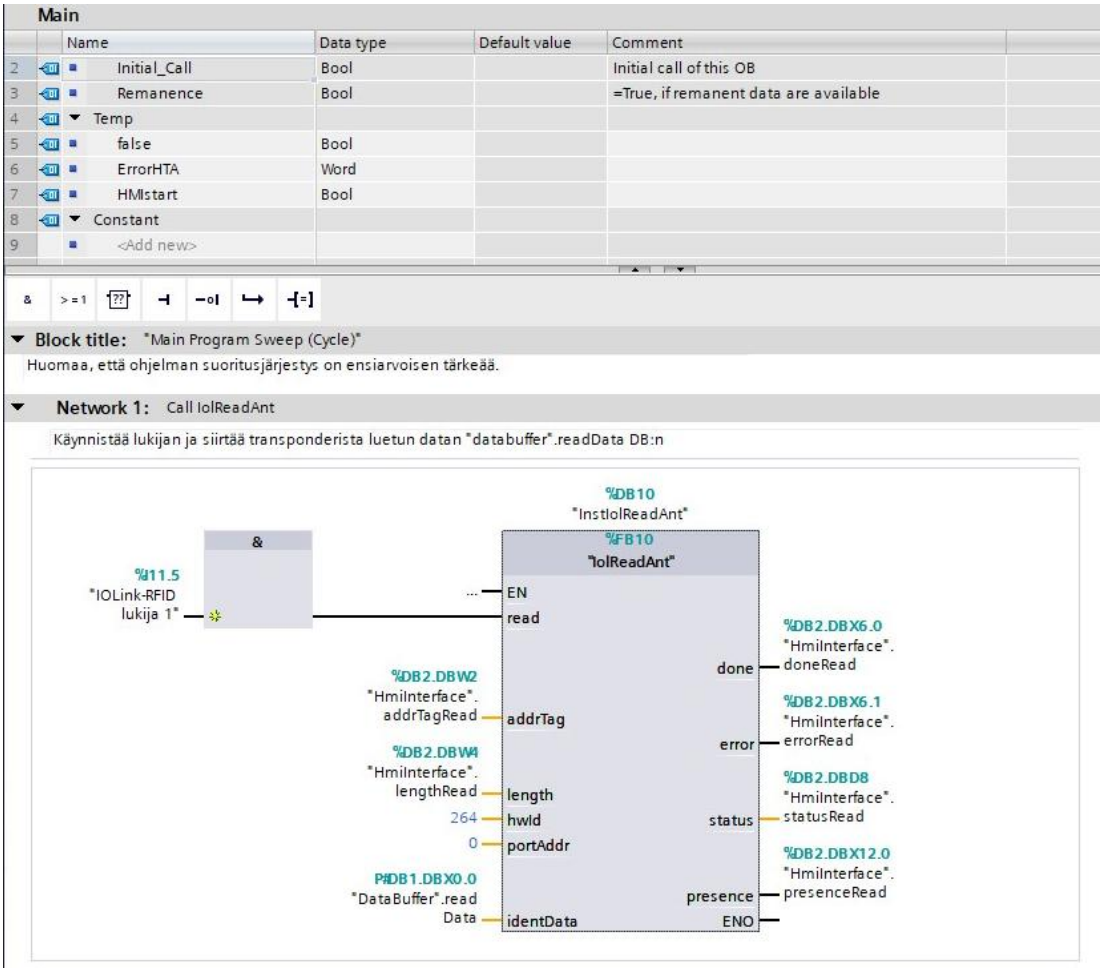
Siemensin esimerkeistä hyödynnetään seuraavat lohkot:

- IolReadAnt
- IolWriteAnt
- LIOLinkRF200_ReadTag
- LIOLinkRF200_SwitchAntenna
- LIOLinkRF200_WriteTag
- LSql – kansio

Joitain lohkoista käytetään sisäisesti toisissa lohkoissa, älä siis poista niitä projektista vaikka FB:itä ei erikseen kutsuttaisi main ohjelmassa! Voit halutessasi siirtää LSql -kansion sisällön muiden joukkoon tai ryhmitellä blokkeja eri kansioihin mutta se ei ole pakollista ohjelman toiminnalle.

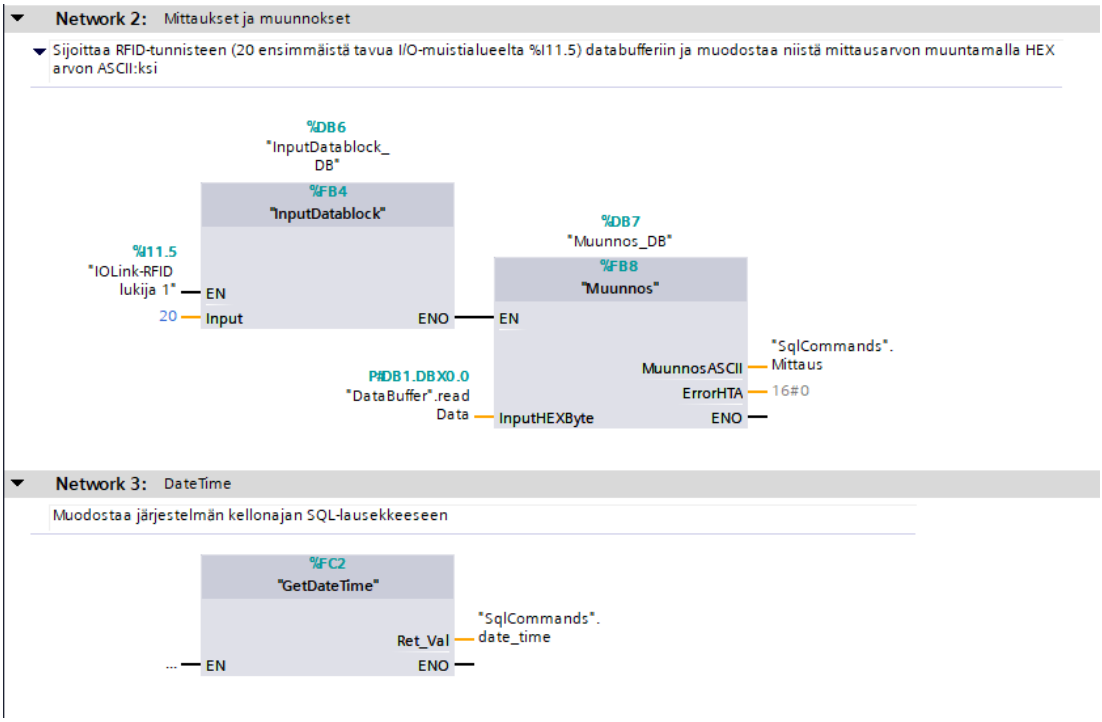
6.2.9 Main – ohjelma

Main – ohjelmassa määritetään funktioiden suoritusjärjestys, syklisen ohjelmankierron mukaisesti, ylhäältä alas ja ensimmäisenä on ”Call IolReadAnt” joka aktivoi lukijan havaitessaan tunnisteen lukualueellaan.



Kuva 19 Main - ohjelman sisälle tulevat muuttujat ja IolReadAnt

Blokkiin kytketään ”%11.5 IOLink-RFID lukija 1” ja Network 2:n ja 3:n tulevat ”Mittaukset ja muunnokset” sekä ”Datetime”.

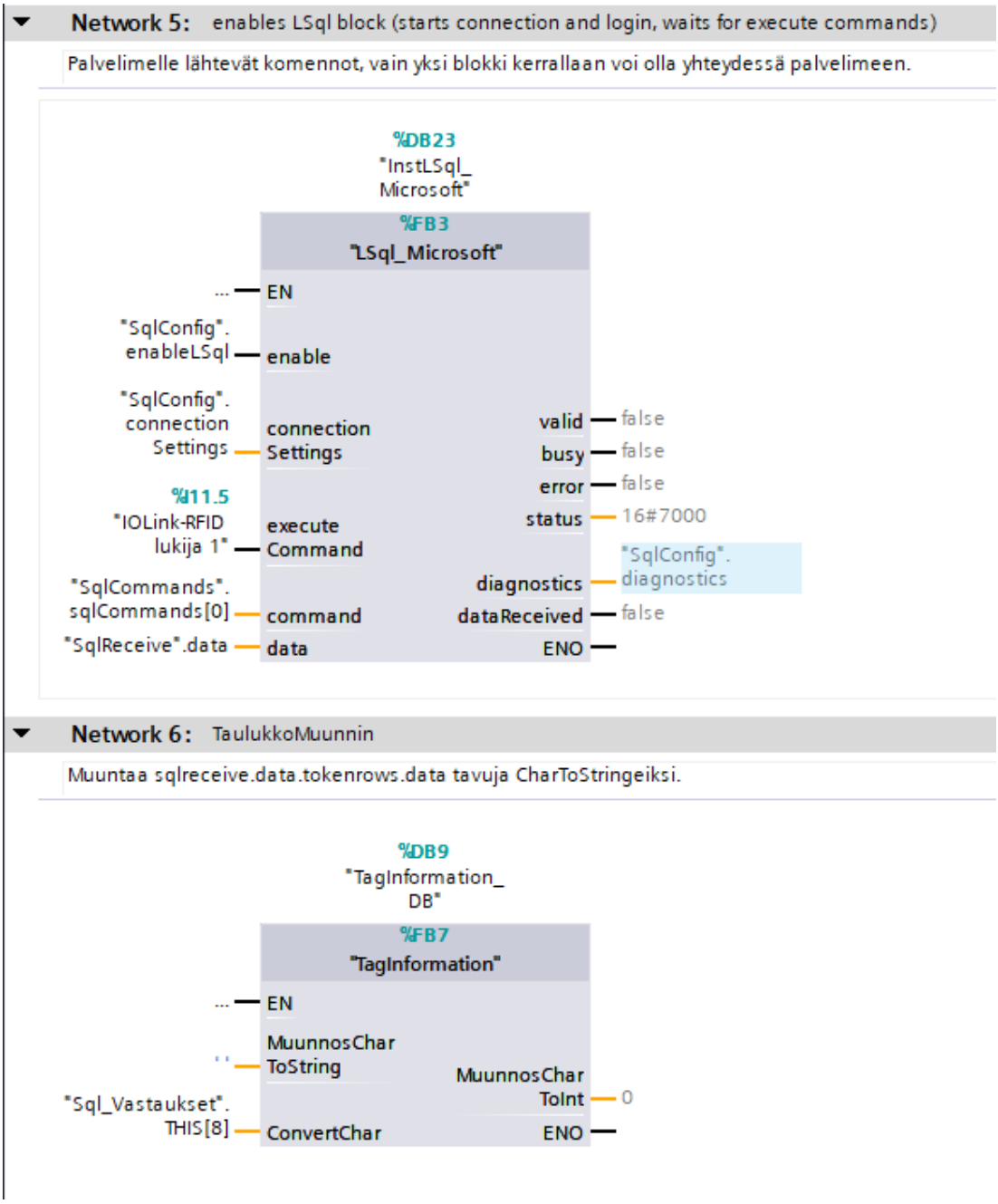


Kuva 20 Network 2 ja 3

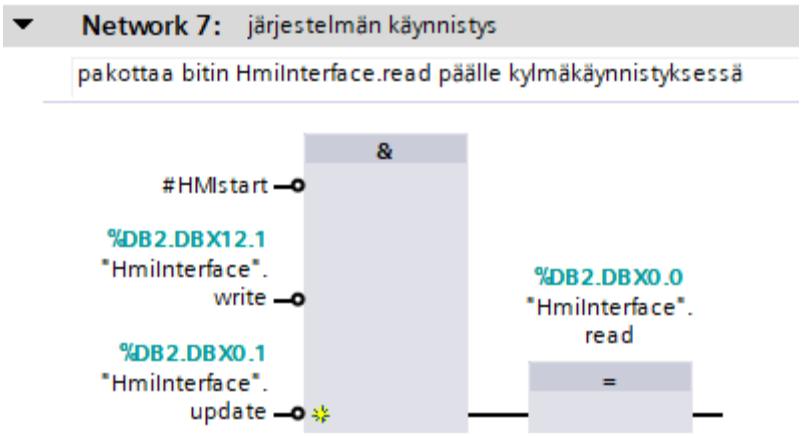
Network 4:n tulee koko jutun avain eli ”SQL INSERT CMD”, joka muodostaa SQL palvelimelle lähtevät komennot. Kuvassa 21 näkyvä input arvo SignalName ei osallistu blokin toimitaan, ja sen voi halutessaan poistaa. SignalValuen voi halutessaan korvata suoraan arvolla ”SqlCommands”.Mittaus

Kuva 21 Network 4 SQL INSERT CMD

Networkit 5 ja 6 sisältävät kaksi tärkeää lohkoa ”InstLSql_Microsoft”, joka huolehtii kirjautumisesta ja käskyjen välittämisestä SQL-palvelimelle, sekä ”TaulukkoMuunnin”, joka muuntaa palvelimelta tulevat vastaukset HMI:n ymmärtämään muotoon ja lopuksi Network 7:n tulee vielä järjestelmän käynnistys alkutilaan, eli bitti ”HmiInterface”.read on aina päällä ja valmiina kun järjestelmä käynnistyy.



Kuva 22 Networkit 5 ja 6



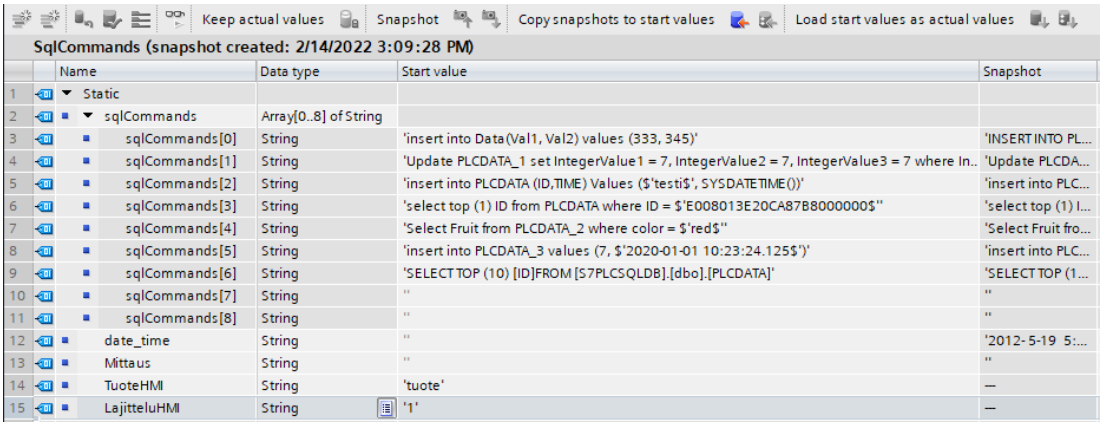
Kuva 23 Network 7

6.2.10 Datablokit, kirjastot ja datatyypit

Datatyypeistä ja kirjastoista kerrotaan tarkemmin: https://support.industry.siemens.com/cs/mdm/109742272?c=70810486923&lc=en-AZ, tämän dokumentin liitteessä 2 sekä oppaassa https://support.industry.siemens.com/cs/attachments/109747503/109747503_Library_Guideline_DOC_v10_en.pdf PLC Datatyyppejä kutsuttiin vanhemmissa STEP7 nimellä UDT (User Data Type) ja nimeen viitataan toisinaan edelleen mm. Siemensin käsikirjoissa ja ne eroavat perusdatatyypeistä mm. siinä että ne ovat useimmiten rakenteellisia ja/tai koostuvat muista datatyypeistä, myös toinen UDT on mahdollinen eikä niitä siten ladata sellaisenaan suoraan PLC:n kuten muita datatyyppejä. Pääosin hyödynnämme esimerkkien mukana tulevia datatyyppejä ja blokkeja sellaisenaan, mutta joitakin niistä pitää muokata tai lisätä itse.

- DataBuffer [DB] käsittää yhteensä 4 kilotavua jaettuna tasan kirjoitus- ja lukualueisiin, tätä DB ei tarvitse muokata.
- HmiInterface [DB] sisältää HMI:ltä tulevat komennot.
- RouteSelection [DB] on kerätty lähdöt, jotka kytketään fyysisesti ratojen ohjaukseen.
- Sql_Vastaukset [DB] on Array Of String – tyyppinen muuttuja jossa on kymmenen riviä. Tästä [DB] haetaan palvelimen vastaukset HMI:n ruudulle muunnoksen jälkeen.
- SqlCommands sisältää kaikki palvelimelle lähtevät komennot.

Muokkauksen jälkeen SqlCommands näkyy kuvassa 24:



Kuva 24 SQLCommands

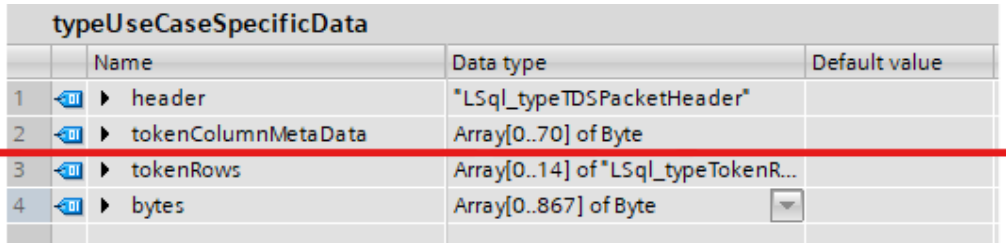
Huomaa, että komennot muodostava ”SQL INSERT CMD” ylikirjoittaa [DB] valmiina olevat rivit, samoin kuin myös HMI, Start value kenttien arvot siis ovat siis lähinnä ohjeellisia ja ne voi jättää myös tyhjiksi.

- SqlConfig [DB] sisältää palvelinliikenteen asetukset kuten IPv4 osoitteen ja portin, tässä esimerkissä ne ovat 192.168.0.10 ja 1433 Huomaa, että PLC ja Sql-palvelimen on oltava samassa aliverkossa.

  MERKITTÄVÄ KYBERTURVALLISUUS RISKI, TÄTÄ OHJELMAA Ei TULE SELLAISENAAN KÄYTTÄÄ VARSINAISESSA TUOTANTOYMPÄRISTÖSSÄ!

- Kohtaan LoginInformation tulevat HMI:stä laitetut kirjautumistiedot, jotka menevät siis palvelimelle verkossa täysin salaamatta ja selkokielisenä ASCII:na.
- Samasta syystä palvelimen kirjautumistunnuksia ei käsitellä tässä ohjeessa.
- SqlReceive [DB] tulee palvelimelta saapuva data.

Datatyypit muokataan ohjeen ”Connecting a S7-1500 to a SQL Database https://support.industry.siemens.com/cs/ww/en/view/109779336” mukaisesti niin että vastaus käsittää oikean määrän tavuja, tässä esimerkissä 70.



Kuva 25 typeUseCaseSpecificData

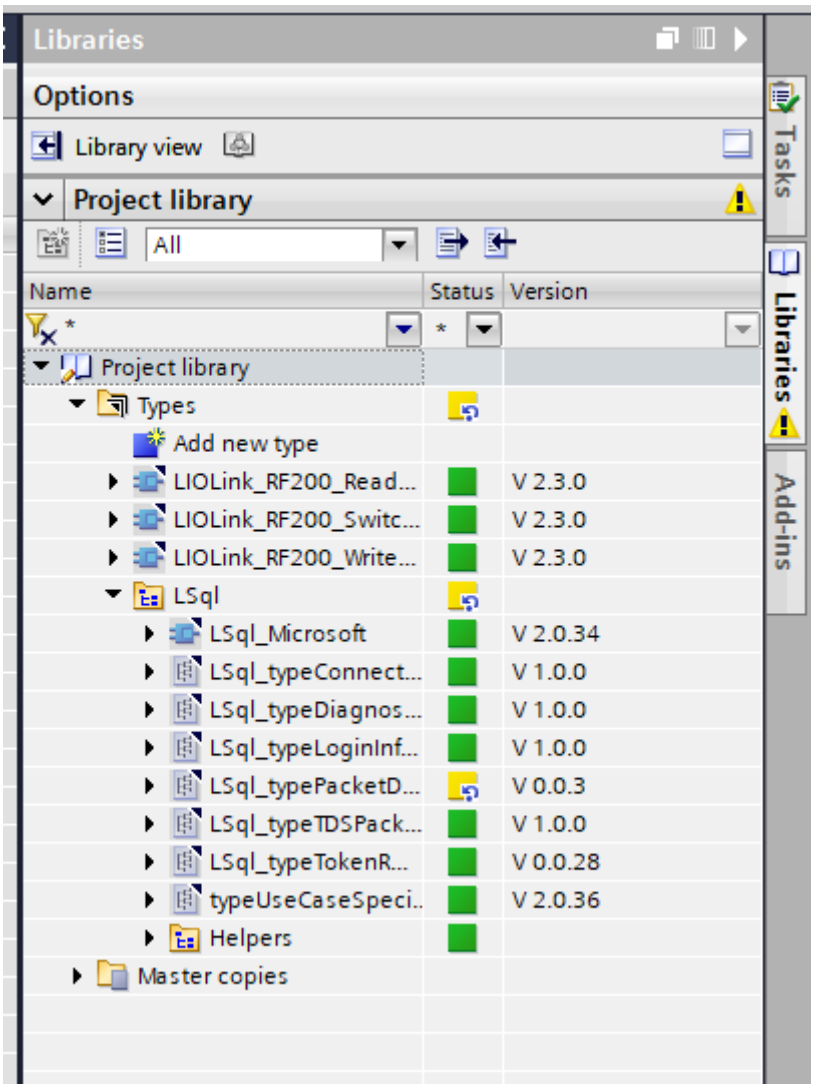
Ja riveiksi tulee LSql_typeTokenRow array 60 merkkiä.

LSql_typeTokenRow

	Name	Data type	Default value
1	Token_type	Byte	16#0
2	length	Byte	16#0
3	data	Array[0..59] of Char	

Kuva 26 LSql_typeTokenRow

Huomaa, että kirjastot tulee aina muokkaamisen jälkeen ”julkaista” oikealla olevasta Libraries-valikosta.



Kuva 27 Libraries valikko

6.3 HMI eli käyttöliittymän ohjelmointi

Projektissa Siemens 10” KTP-sarjan paneeli vaihdettiin Siemens 15” Comfort-sarjan paneeliin, tämä aiheutti joitakin muutoksia ruudun asetteluun ja skaalaukseen muttei sinänsä vaikuta paneelin toiminnallisuuteen. Tässä ohjeessa ei kuitenkaan käsitellä Comfort tai muidenkaan HMI-paneeleiden configurointia PLC-liikenteelle, vaan ainoastaan projektikäyttöliittymän asetuksia ja toimintoja.
HMI lyhenne tulee sanoista ”Human Machine Interface”, ja vaikka termi yleensä mielletään tarkoittamaan lähinnä teollisuuden kosketusnäyttöpaneeleja, kattaa termi myös muutkin käyttöliittymien osa-alueet kuten kauko-ohjaimet ja ohjausvivut.

Tässä Projektissa HMI:n kautta voidaan tehdä seuraavat asiat:

- Lukea mitä tietoja kuhunkin RFID - tunnisteeseen on sidottu (tuotteen nimi, rata, RFID - tunniste) näistä kolmesta RFID - tunniste on yksilöivä eli sitä ei voi vaihtaa eikä niitä voi esiintyä SQL-tietokannassa kuin yksi.
- Hoitaa kirjautumisen palvelimelle sekä kertoo virhetilanteissa tulostuvan virhekoodin.
- Päivittää tietokannassa olevia tietoja sekä lisätä uusia tunnisteita. Huomaa, että saman tunnisteen voi lisätä kantaan vain kerran.
- Lukijan tilaa valvotaan tilavaloilla: Presense syttyy kun lukija on havainnut tunnisteen, Done valo kertoo kun tunnisteen luku/kirjoitus prosessi on valmis ja Error ilmoittaa, jos luettaessa tai kirjoitettaessa tapahtuu virhe.
- Status kertoo lukijavirheeseen liittyvän Hex koodin.

6.3.1 Käyttöliittymän rakenne

HMI - liittymä koostetaan ns. näyttöruuduista (screen) joiden välillä liikkumista ohjataan liittymän toiminnoilla. Ruuduille voidaan luoda ns. template, joka toimii pohjana kaikille myöhemmin luotaville ruuduille. Templatea käytetään yleensä luotaessa staattisia elementtejä kuten taustagrafiikkaa ja navigointipalkkeja.
Käyttöliittymä koostuu neljästä ruudusta IOL_Station_READ/WRITE, Startscreen ja Support sekä template.

6.3.2 HMI Startscreen

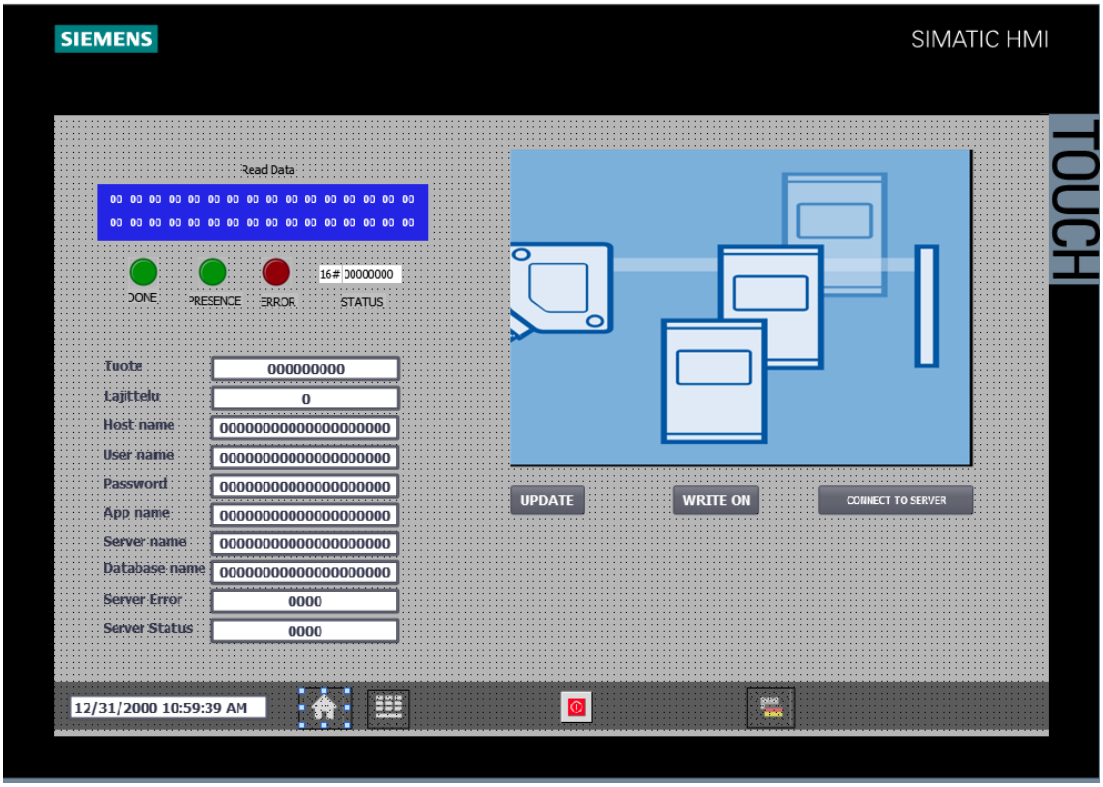
Kuva 28 HMI aloitusruutu

Tässä ruudussa on useita staattisia elementtejä, joista huomattavin ovat laatikoiden päällä olevat näkymättömät ”painikkeet” niiden avulla voidaan toteuttaa interaktiiviset toiminnallisuudet myös staattisille objekteille, oleellista on, että ”painikkeet” ovat aina päällimmäisessä kerroksessa (layer) muuhun grafiikkaan nähden. Tämän voi testata viemällä hiiren painikkeen päälle ja klikkaamalla ja liikuttamalla sitä; et voi muuttaa painikkeen läpinäkyvää tekstikenttää koska painike on ikään kuin sen päällä. Painike pitää siis siirtää ensin ”pois tieltä” mikäli alla olevaa kerrosta halutaan muokata. Tässä ruudussa on alkuperäisen mallin taustagrafiikka venytetty 15” sekä muutettu painikkeiden toimintaa seuraavasti:

- I/O-Link RFID-Reader painike viittaa nyt sivulle IOL_Station_READ/WRITE

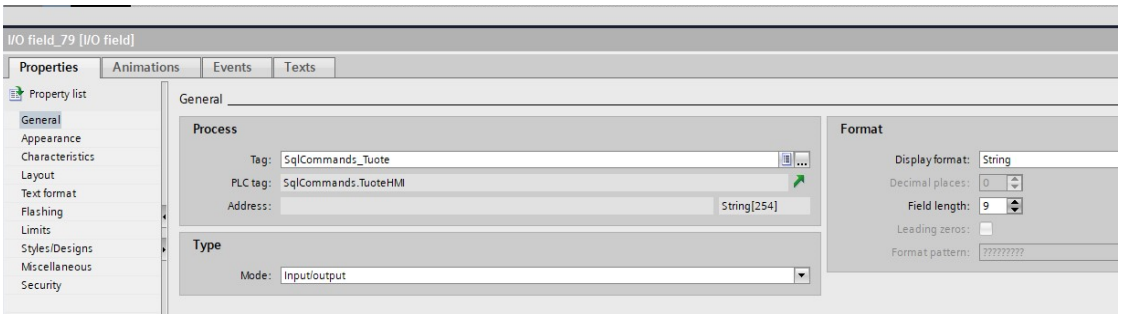
6.3.3 HMI IOL_Station_READ/WRITE

Tähän ruutuun on yhdistetty aiemmat IOL_Station_READ ja – WRITE ruudut ja lisätty järjestelmän vaatimat toiminnallisuudet. Lisäksi navigaationäppäimet on poistettu, koska toiminnallisia sivuja on vain yksi.



Kuva 29 IOL_Station_READ/WRITE ruutu

Kentät toteutetaan sitomalla HMI-tagit PLC muuttujiin ja tageihin tämä tapahtuu WINCC:ssä automaattisesti valitsemalla tagit tai arvot, jotka halutaan yhdistää.



Kuva 30 Esimerkki I/O field kentän kytkemisestä

Kaikki tekstikentät ovat string tyyppisiä paitsi ”server error” (bool) ja Server Status (word) mikäli HMI-tagia ei ole sellainen luodaan automaattisesti. Kentät voidaan myös salata jolloin kentässä näkyy vain ’*’ merkkien sijaan. Arkkitehtuuri on kaikkialla WINCC:ssä yhtenäinen, joten kaikissa elementeissä tiedot yhdistetään samalla tavalla:

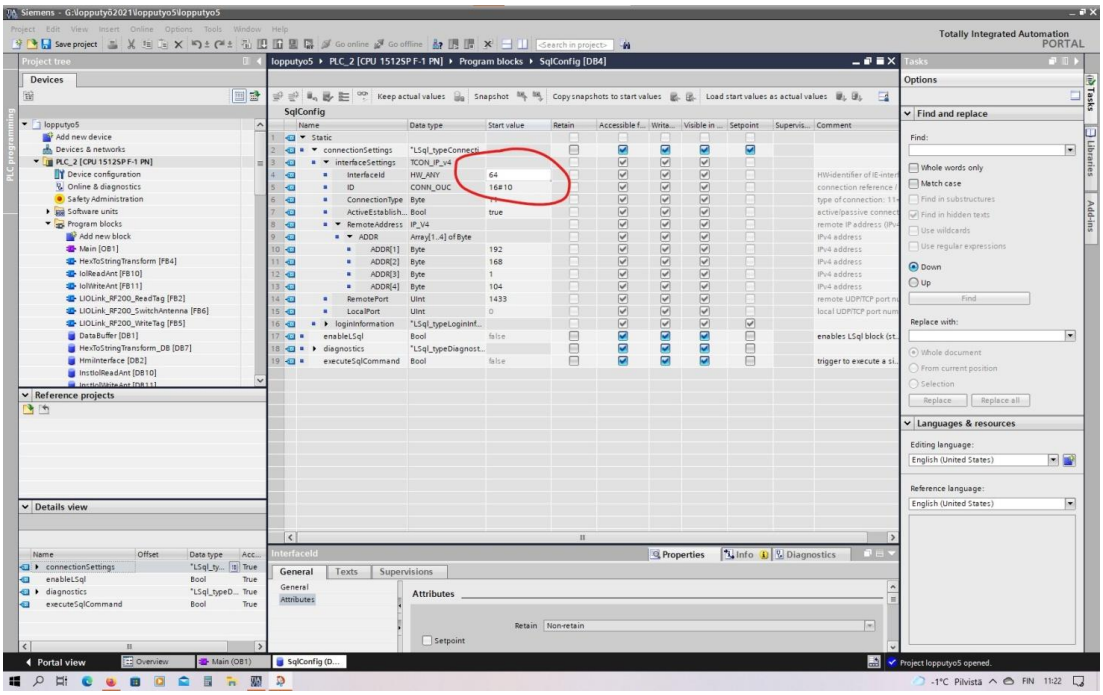
Read Data taulukko alkaa kirjoittamisen vasemmasta yläkulmasta, joten ensimmäiseksi täytetään ensimmäiset rivit DataBuffer_readData{0} - DataBuffer_readData{15} tähän tulevat siis tavut jotka kertovat tunnisteen RFID-koodin. Seuraavalle riville tulevat SqlCommands_SQLvastaus{0} - SqlCommands_SQLvastaus{9} huomaa, että SqlCommands_SQLvastaus viittaa array-tyyppiseen muuttujaan, joten viittaukset ovat siis muotoa Sql_Vastaukset.THIS[9]. Loput kentistä jätetään varalle.

Lamput Done, Presence ja error ovat ohjelmassa jo valmiiksi sidottuja samoin kuin status I/O-kenttä eikä niihin tarvitse tehdä varsinaisia muutoksia.

Vasempaan alakulmaan on lisätty kello ja päivämäärä WINCC:n kirjastosta, ”update” on painonappi, ”write on/off” ja ”connect to server” ovat switch-tyyppisiä painikkeita. Kaikki elementit yhdistetään taulukon HMI tags taulukko mukaisesti mikä löytyy tämän dokumentin liitteestä 1.

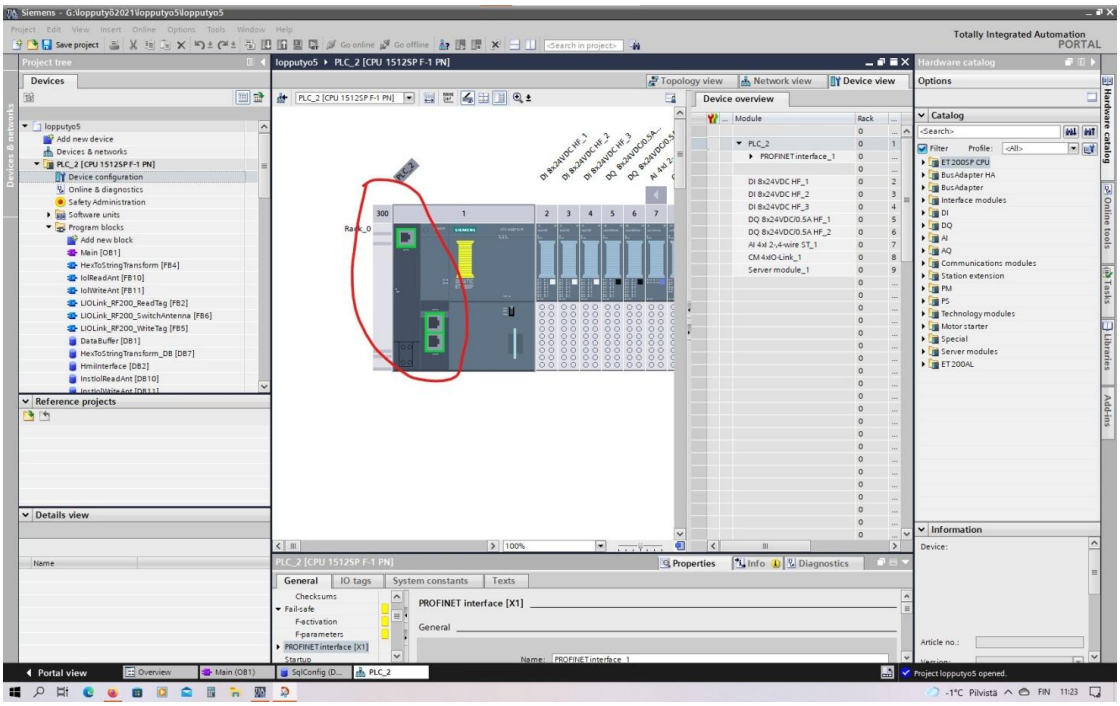
6.3.4 Ohje SQL-palvelimen asetukset PLC

Configurointi tiedostossa olevan HW numeron tulee täsmätä PLC kommunikaatioportin HW numeroon.



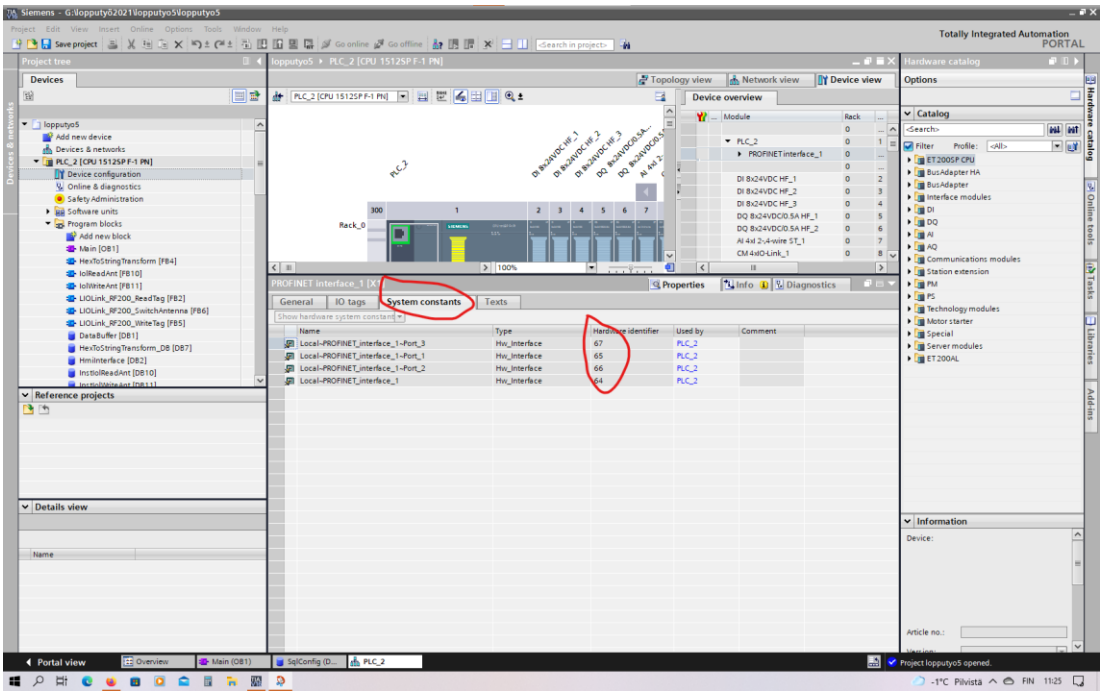
Kuva 31 HW numero asetuksissa

Tuplaklikkaa device configuration ikkunassa PLC:n ethernet portteja.



Kuva 32 HW numero löytyy klikkaamalla portteja

Klikkaa system constants välilehteä ja varmista että portin HW numero täsmää projektin configuraatioon.



Kuva 33 moduulien HW numerot laiteikkunassa

7 SQL-PALVELIMEN PERUSTAMINEN

Palvelimen luomiseen ja ylläpitoon tarvitaan kaksi erillistä ohjelmaa:

- Microsoft SQL Server 2019 Configuration Manager (MSSQL)
- Microsoft SQL Server Management Studio 18 (SSMS)

Asennusohjeet: Microsoft SQL Server 2019 Configuration Manager

1. Valitse asennustavaksi: Basic tai download media, jos haluat vain ladata asennustiedostot. Valinnan jälkeen ohjelma lataa Internetistä asennustiedostot. Hyväksy ehdot ja asennuskansio. Huomaa, että Basic asennus antaa aina palvelimelle nimeksi SQLEXPRESS mutta sen voi muuttaa myöhemmin.
2. Seuraavaksi klikkaa asennus ikkunassa INSTALL SSMS jatkaaksesi asennusta, ohjelma avaa selaimen sivustolle jolta voit asentaa Microsoft Server Management Studion. Nyt voit klikata COMPLITE ja sulkea palvelimen asennusohjelman.

Asennusohjeet: Microsoft Server Management Studio 18

1. Valitse asennuskansio ja odota asennuksen valmistumista.
2. Käynnistä kone uudelleen ja asennus on valmis.

7.1 Palvelimen configurointi PLC liikenteelle.

Etsi äsken asentamasi ohjelmat ja klikkaa hiiren oikealla pin to start / kiinnitä käynnistä valikkoon, jotta ohjelmat ovat helpommin löydettävissä.

1. Käynnistä Server Configuration Manager ja mene hiirellä kohtaan: SQL Server Network Configuration.
   Klikkaa välilehteä TCP/IP ja valitse enabled.
   Mene IP osoitteet välilehdelle ja valitse osoite numero 3 ja muuta sen osoite vastaamaan tietokoneen IP-osoitetta, valitse active ja enabled: yes.
   Laita portiksi 1433, tämä voi vaatia myös erillisen palomuurisäännön luomisen, ohjeet sääntöjen luomiseen löytyvät Microsoftin help sivustolta.
   Lopuksi laita vielä kohtaan IPALL TCP-portiksi 1433.
2. Lopuksi klikkaa OK ja uudelleen käynnistä palvelin.
3. Seuraavaksi klikkaa SQL Server Services välilehteä ja varmista, että SQL Server Browser on käynnistystilassa automatic ja käynnistä se local servicenä.
4. Uudelleen käynnistä SQL server (SQLEXPRESS) ja SQL Server Browser

Käynnistä Microsoft SQL Server Management Studio

Windows käyttäjätili on automaattisesti palvelimen pääkäyttäjä, joten klikkaa connect ja kirjaudut palvelimelle.

1. Klikkaa valikossa palvelimen nimeä hiiren oikealla ja valitse properties.
2. Mene kohtaan security klikkaa salli myös SQL-kirjautuminen. Klikkaa OK ja käynnistä palvelin uudelleen.
3. Mene palvelimen kansioon ja valitse security -> logins
4. Klikkaa hiiren oikealla ja valitse new login luodaksesi uuden käyttäjän, tämä on käyttäjä jona PLC tulee kirjautumaan sisään palvelimelle.
   Klikkaa kohtaa SQL Server Authentication ja kirjoita tilin nimeksi S71500, salasanaksi tulee S71500.
   Tällä hetkellä PLC-tilille ei ole vielä luotu tietokantaa, johon se kirjautuu oletuksena sisään, joten klikkaa OK.
5. Seuraavaksi mene kansioon Database ja valitse hiiren oikealla create New Database.
6. Anna sille nimeksi S7PLCSQLDB ja klikkaa Ok
7. Mene äsken luomaasi DB ja klikkaa hiiren oikealla Tables kansiota ja valitse New Table.
8. Column kohtaan tulee taulukon nimi ”ID” ja muuttujan tyypiksi tulee nchar(10), lopuksi klikkaa ylävalikosta refresh (nuoli pylpyrä).
9. Lopuksi klikkaa vielä ylävalikosta save ja tallenna taulu nimellä PLCDATA.
10. Jos haluat nähdä juuri luomasi taulun sisällön voit nyt klikata sitä hiiren oikealla ja valitse show (200) rows tai show (1000) rows ja palvelin tulostaa sinulle taulukon arvot, jotka ovat tietysti tällä hetkellä tyhjiä (NULL). 1000 näyttää lisäksi käskyn rakenteen, joka lähetettiin palvelimelle.
11. Klikkaa taas tallenna ja palaa aiempaan Security kansioon

12. Mene luomasi PLC-käyttäjän kohdalle ja klikkaa hiiren oikealla properties, vaihda käyttäjän oletus kirjautumiseen äsken luomasi DB ja valitse tämän jälkeen Server Roles välilehti.
13. Varmista että PLC-käyttäjällä on vain oikeudet ”public”, siirry välilehteen User Mapping ja valitse luomasi DB (S7PLCSQLDB), tämä antaa PLC-käyttäjälle ”omistajuuden” kyseiseen DB:n.
14. Varmista samalla alemmasta ikkunasta, että S7PLCSQLDB:n ovat oikeudet vain käyttäjätasolle public.
15. Kohdasta tarkista vielä Status lehdeltä, että käyttäjällä ovat oikeudet ”permission to connect to database engine” grant ja ”login” enabled
16. Lopuksi klikkaa taas OK
17. Klikkaa seuraavaksi luomaasi databasea hiiren oikealla napilla ja valitse Properties ja sirry välilehdelle Permissions.
18. klikkaa aluksi käyttäjää ja sen jälkeen ala ruksittamaan alla olevasta ikkunasta käyttäjän oikeuksia:
    - Backup database
    - Connect
    - Control
    - Create function
    - Create queue
    - create Rule
    - create service
    - create table
    - ceate type
    - create view
    - Delete
    - Execute
    - Insert
    - select
    - references
    - update
    - take ownership
    - view any column encryption key definition
    - view any column master key definition
    - view any sensitivity classification
    - view database state
    - view definition

Lopuksi klikkaa taas OK ja ylävalikosta REFRESH

Seuraavaksi testaa tilin asetukset klikkaamalla töpselin kuvaa jossa lukee disconnect ja kirjaudu sisään (SQL server authentication). Kohdassa 4 luotu PLC-tili on S71500 ja salasana on S71500. Mikäli saat virhe ilmoituksen, sulje ohjelma, käynnistä server management studio ja boottaa palvelin uudelleen.

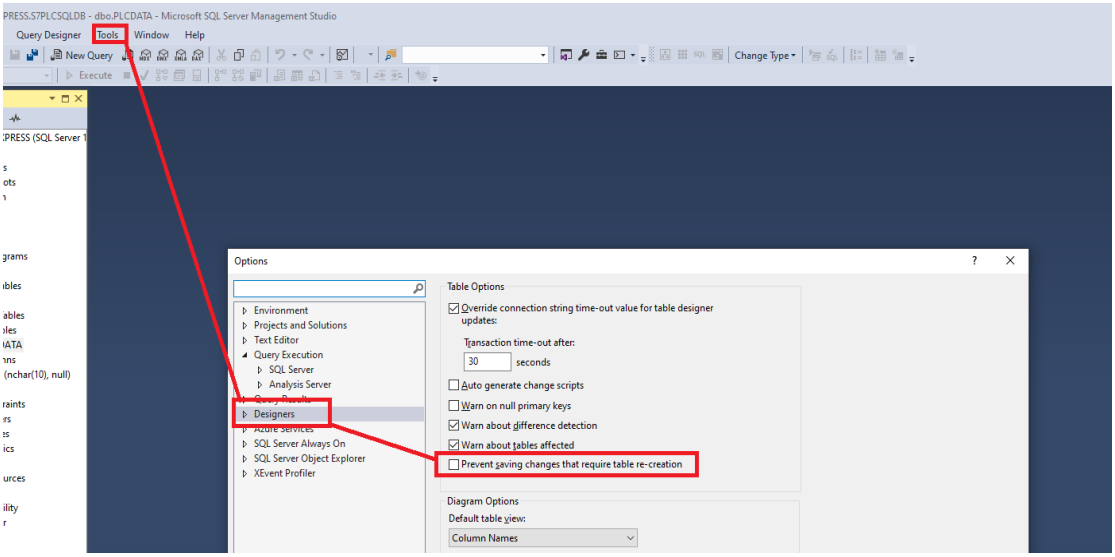
Kirjauduttuasi sisään näet vain sen näkymän johon sinulle on käyttäjänä annettu oikeudet eli tarkista, että näet DB kansiosi ja suorita taulukolle kysely klikkaamalla hiiren oikealla ja valitsemalla edit 200 tai show 1000 rows.
Lopuksi luo vielä pystyriville ”ID” ns. unique key, jolloin kyseisellä pystyrivillä ei voi esiintyä duplikaatteja (paitsi NULL) eli jokainen vaakarivi on nyt siis sidottu omaan yksilölliseen RFID-tunnisteeseensa.
Tallenna kaikki ja käynnistä Windows uudelleen, tämän jälkeen luo palomuurisääntö portille 1433.

- verkon ja internetin asetukset -> windowsin palomuuri -> lisäasetukset -> saapuvan liikenteen säännöt ja menevän liikenteen säännöt -> uusi sääntö portti 1433.

Uudelleen käynnistyksen jälkeen palvelimen asetukset ovat valmiit.

7.2 Ongelma tilanteet:

Mikäli palvelin ei anna muokata tauluja vaikka käyttäjälle on annettu kaikki oikeudet tarkista seuraava asetus:



Kuva 34 Oikeus tehdä tauluihin myös sellaisia muutoksia, jotka vaativat taulukon uudelleen kokoamista.

8 HYÖDYLLISIÄ OPPAITA JA VINKKEJÄ

- Automaation tietoturva -julkaisut | Suomen automaatioseura ry. https://www.automaatioseura.fi/julkaisut-kirjakauppa/automaation-tietoturva-julkaisut/.
- Teollisuusautomaation tietoturva | Verkottumisen riskit ja niiden hallinta | Suomen automaatioseura ry. ISBN 952-5183-24-6
- Petković, D. Microsoft SQL Server 2016 : a beginner's guide
- OPC UA - Siemens S7-1500 (OPC UA Client) and S7-1200 (OPC UA Server) TIA Portal - YouTube. https://www.youtube.com/watch?v=oChVcRaUt9M
- Connecting a S7-1200 / S7-1500 to a SQL Database - ID: 109779336 - Industry Support Siemens. https://support.industry.siemens.com/cs/document/109779336/connecting-a-s7-1200-s7-1500-to-a-sql-database-?dti=0&lc=en-WW
- Creating of OPC UA clients with .NET and helper class - ID: 109737901 - Industry Support Siemens. https://support.industry.siemens.com/cs/document/109737901/creating-of-opc-ua-clients-with-net-and-helper-class?lc=en-ww
- S7 user block for the OPC UA client of a SIMATIC S7-1500 - ID: 109762770 - Industry Support Siemens. https://support.industry.siemens.com/cs/document/109762770/s7-user-block-for-the-opc-ua-client-of-a-simatic-s7-1500?dti=0&lc=en-WW
- Palvelimen IP - osoitteen ja portin selvittämiseen löytyvät ohjeet: https://amberpos.zendesk.com/hc/en-us/articles/215978723-How-to-find-your-database-IP-address-and-SQL-port

9 YHTEENVETO JA POHDINTAA

Työ aloitettiin valitsemalla SAMK opetuskäytössä olevista Siemens ET200SP laitteista yksi johon asennettiin CM 4xIO-Link moduuli RF220R IO-Link RFID-lukijaa varten. HMI:ksi valittiin Siemens TP1500 Comfort 15". Lisäksi TIA Portal v16 ohjelmiston laitekirjastoja piti päivittää, jotta se toimisi nykyisen laitteiston kanssa. Välissä TIA Portal ohjelmisto vaihtui versioon v17, johon on lisätty huomattavia

lisäyksiä kyberturvallisuuteen uutuutena mm. mahdollisuus salata HMI:n ja PLC:n välinen liikenne.

Ohjelmien asennuksessa meni kaikkiaan noin nelisen tuntia ja järjestelmän konfigurointiin meni kokonaisuudessaan useita päiviä, jotta se saatiin toimimaan saumattomasti. Samalla huomattiin, että TIA portal ohjelmiston asennuksessa voi esiintyä ongelmia ja virheitä, joiden vuoksi ohjelmisto toimi joillakin työasemilla vain osittain, asian varmistaminen pelkästään ohjelmistosta johtuvaksi aiheutti huomattavan määrän ylimääräistä työtä. Varsinainen ohjelmointi sujui kuitenkin nykyaikaisella TIA portal ohjelmistolla varsin kivuttomasti vaikka koodi jouduttiinkin rakentamaan kolmesta eri ohjelmointikielestä, tämä ei kuitenkaan ole PLC ympäristössä mitenkään epätavallista koska IEC 61131-3 mukaisesti eri ohjelmointikielet sopivat paremmin eri sovelluksiin eikä kaikkia käskyjä, varsinkaan Siemens maailmassa, aina edes löydy kuin yhdelle tai kahdelle eri kielelle. Ohjelmoinnissa käytettiin apuna myös Wireshark – ohjelmaa, jolla voitiin tarkastella muutakin verkkoliikennettä, kuin vain ASCII koodattuina viesteinä kulkevia TDS - paketteja PLC:n ja tietokannan välillä, mikä nopeutti vianhakua. RFID - tunnisteisiin kirjoittamisen sijaan projektissa päädyttiin hyödyntämään niissä valmiina olevaa uniikkia tehdastunnistetta. Itse tunnisteisiin ei siis talleteta mitään prosessitietoa vaan se tieto sijaitsee ainoastaan palvelimella. Työhön kuului tietenkin PLC.n lisäksi myös itse käyttöliittymän ja tietokannan suunnittelu ja ohjelmointi, sekä SQL-palvelimen pystytys. Palvelin kytkettiin paitsi kyberturvallisuuden myös käytännön syistä samaan aliverkkoon PLC:n ja HMI:n kanssa, koska Siemensin ohjeiden mukaan TDS - palvelinliikenne ei soveltunut reititettäväksi ja tämä huomattiin myös käytännössä kokeilemalla. Tämä johtunee siitä, että OSI - kerrokset 1, 2 ja 7 (joita esimerkiksi Profinet RT käyttää, TDS protokolla käyttää vain kerrosta 7) eivät ole reititettävää liikennettä, kerros 7 kuitenkin mahdollistaa liikenteen salaamisen ja todentamisen. Profinet CBA kuitenkin sallii myös IP(3) ja TCP/UD(4) liikenteen, jolloin se voidaan tarvittaessa reitittää. Tämä Profinetin duaalisuus aiheutti myös sen, että toisinaan laite saattoi hyvinkin löytyä verkosta (MAC – osoite eli OSI 1 ja 2 kerrokset) mutta sen kanssa ei voinut ”keskustella” koska IP – osoite (OSI kerrokset 3 ja 4) oli määritetty väärin. Tämä siksi, että TIA Portal ohjelmisto tarvitsee kommunikointiin 3 ja 4 kerroksia (IP ja UD) mutta väylässä olevat laitteet löytävät toisensa myös pelkän 1 ja 2 kerroksen avulla (MAC). Alkujaan järjestelmä suunniteltiin sijoitettavaksi laboratoriossa olevaan kuljetinra-

taan mutta viime hetkillä päätettiin, että mikäli toimintaa opetuskäytössä tarvitsee erikseen visualisoida, voidaan se suorittaa myös simulaatiomallina.

Simulaatiomallin rakentaminen OPCUA - liityntöineen ei kuitenkaan ole enää tämän työn aihepiirissä vaan se tullaan suorittamaan myöhempänä ajankohtana. Kaiken kaikkiaan projekti on opettanut paljon asioita, joita ei voida yleensä tavanomaisen kurssimuotoisen opetuksen puitteissa testata tai kokeilla ja toivonkin, että se tuottaa samanlaisia oivalluksia ja elämyksiä vastaisuudessa myös opiskelijoille joidenka tulevien harjoitustöiden pohjaksi tämä projekti on tarkoitettu.

LÄHTEET

Step 7 Elementary Data Types. (2022). Step 7 Elementary Data Types | PLCdev. Haettu 3.4.2022 osoitteesta http://www.plcdev.com/step_7_elementary_data_types

Helsingin Yliopisto. Tietokantojen perusteet. SQL-tietokanta. Haettu 9.10.2021 osoitteesta https://www.cs.helsinki.fi/u/laine/tkpv/sql/sql_tietokanta.html

Stuxnet worm hit industrial systems. (2010). Computerworld. Haettu 03/28/2022 osoitteesta https://www.computerworld.com/article/2515570/siemens--stuxnet-worm-hit-industrial-systems.html

Israel testasi Stuxnet-matoa ennen verkkohyökkäystä Iraniin. (2012). Yle Uutiset. Haettu 03/28/2022 osoitteesta https://yle.fi/uutiset/3-5306825

Kuvat ja kaaviot Google kuvahaku ja Siemens oppaat.

LIITE 1 REFERENSSI TAULUKOT

Taulukko 1 HMI Tags taulukko:

Name	PLC tag	DataType
HmiInterface_errorWrite	HmiInterface.errorWrite	Bool
DataBuffer_readData{0}	DataBuffer.readData[0]	Byte
DataBuffer_readData[1]	DataBuffer.readData[1]	Byte
DataBuffer_readData[2]	DataBuffer.readData[2]	Byte
DataBuffer_readData[3]	DataBuffer.readData[3]	Byte
DataBuffer_readData[4]	DataBuffer.readData[4]	Byte
DataBuffer_readData[5]	DataBuffer.readData[5]	Byte
DataBuffer_readData[6]	DataBuffer.readData[6]	Byte
DataBuffer_readData[7]	DataBuffer.readData[7]	Byte
DataBuffer_readData[8]	DataBuffer.readData[8]	Byte
DataBuffer_readData[9]	DataBuffer.readData[9]	Byte
DataBuffer_readData[10]	DataBuffer.readData[10]	Byte
DataBuffer_readData[11]	DataBuffer.readData[11]	Byte
DataBuffer_readData[12]	DataBuffer.readData[12]	Byte
DataBuffer_readData[13]	DataBuffer.readData[13]	Byte
DataBuffer_readData[14]	DataBuffer.readData[14]	Byte
DataBuffer_readData[15]	DataBuffer.readData[15]	Byte
DataBuffer_readData[16]	DataBuffer.readData[16]	Byte
DataBuffer_readData[17]	DataBuffer.readData[17]	Byte
DataBuffer_readData[18]	DataBuffer.readData[18]	Byte
DataBuffer_readData[19]	DataBuffer.readData[19]	Byte
DataBuffer_readData[20]	DataBuffer.readData[20]	Byte
DataBuffer_readData[21]	DataBuffer.readData[21]	Byte
DataBuffer_readData[22]	DataBuffer.readData[22]	Byte
DataBuffer_readData[23]	DataBuffer.readData[23]	Byte
HmiInterface_read	HmiInterface.read	Bool
HmiInterface_errorRead	HmiInterface.errorRead	Bool
HmiInterface_statusRead	HmiInterface.statusRead	DWord
HmiInterface_lengthRead	HmiInterface.lengthRead	Word
HmiInterface_presenceRead	HmiInterface.presenceRead	Bool
DataBuffer_writeData[0]	DataBuffer.writeData[0]	Byte
DataBuffer_writeData[1]	DataBuffer.writeData[1]	Byte
DataBuffer_writeData[10]	DataBuffer.writeData[10]	Byte
DataBuffer_writeData[11]	DataBuffer.writeData[11]	Byte
DataBuffer_writeData[12]	DataBuffer.writeData[12]	Byte
DataBuffer_writeData[13]	DataBuffer.writeData[13]	Byte
DataBuffer_writeData[14]	DataBuffer.writeData[14]	Byte
DataBuffer_writeData[15]	DataBuffer.writeData[15]	Byte
DataBuffer_writeData[16]	DataBuffer.writeData[16]	Byte

DataBuffer_writeData[17]	DataBuffer.writeData[17]	Byte
DataBuffer_writeData[18]	DataBuffer.writeData[18]	Byte
DataBuffer_writeData[19]	DataBuffer.writeData[19]	Byte
DataBuffer_writeData[2]	DataBuffer.writeData[2]	Byte
DataBuffer_writeData[20]	DataBuffer.writeData[20]	Byte
DataBuffer_writeData[21]	DataBuffer.writeData[21]	Byte
DataBuffer_writeData[22]	DataBuffer.writeData[22]	Byte
DataBuffer_writeData[23]	DataBuffer.writeData[23]	Byte
DataBuffer_writeData[3]	DataBuffer.writeData[3]	Byte
DataBuffer_writeData[4]	DataBuffer.writeData[4]	Byte
DataBuffer_writeData[5]	DataBuffer.writeData[5]	Byte
DataBuffer_writeData[6]	DataBuffer.writeData[6]	Byte
DataBuffer_writeData[7]	DataBuffer.writeData[7]	Byte
DataBuffer_writeData[8]	DataBuffer.writeData[8]	Byte
DataBuffer_writeData[9]	DataBuffer.writeData[9]	Byte
HmiInterface_write	HmiInterface.write	Bool
HmiInterface_lengthWrite	HmiInterface.lengthWrite	Word
HmiInterface_statusWrite	HmiInterface.statusWrite	DWord
HmiInterface_doneRead	HmiInterface.doneRead	Bool
HmiInterface_doneWrite	HmiInterface.doneWrite	Bool
HmiInterface_addrTagRead	HmiInterface.addrTagRead	Word
HmiInterface_addrTagWrite	HmiInterface.addrTagWrite	Word
HmiInterface_presenceWrite	HmiInterface.presenceWrite	Bool
DataBuffer_writeData[31]	DataBuffer.writeData[31]	Byte
DataBuffer_writeData[30]	DataBuffer.writeData[30]	Byte
DataBuffer_writeData[29]	DataBuffer.writeData[29]	Byte
DataBuffer_writeData[28]	DataBuffer.writeData[28]	Byte
DataBuffer_writeData[27]	DataBuffer.writeData[27]	Byte
DataBuffer_writeData[26]	DataBuffer.writeData[26]	Byte
DataBuffer_writeData[25]	DataBuffer.writeData[25]	Byte
DataBuffer_writeData[24]	DataBuffer.writeData[24]	Byte
DataBuffer_readData[24]	DataBuffer.readData[24]	Byte
DataBuffer_readData[25]	DataBuffer.readData[25]	Byte
DataBuffer_readData[26]	DataBuffer.readData[26]	Byte
DataBuffer_readData[27]	DataBuffer.readData[27]	Byte
DataBuffer_readData[28]	DataBuffer.readData[28]	Byte
DataBuffer_readData[29]	DataBuffer.readData[29]	Byte
DataBuffer_readData[30]	DataBuffer.readData[30]	Byte
DataBuffer_readData[31]	DataBuffer.readData[31]	Byte
SqlCon-fig_connectionSettings_loginInformation_hostName	SqlCon-fig.connectionSettings.loginInformation.hostName	String
SqlCon-fig_connectionSettings_loginInformation_userName	SqlCon-fig.connectionSettings.loginInformation.userName	String

SqlCon-fig_connectionSettings_loginInformation_password	SqlCon-fig.connectionSettings.loginInformation.password	String
SqlCon-fig_connectionSettings_loginInformation_appName	SqlCon-fig.connectionSettings.loginInformation.appName	String
SqlCon-fig_connectionSettings_loginInformation_serverName	SqlCon-fig.connectionSettings.loginInformation.serverName	String
SqlCon-fig_connectionSettings_loginInformation_databaseName	SqlCon-fig.connectionSettings.loginInformation.databaseName	String
SqlConfig_diagnostics_status	SqlConfig.diagnostics.status	Word
SqlConfig_enableLSql	SqlConfig.enableLSql	Bool
InstLSql_Microsoft_status	InstLSql_Microsoft.status	Word
InstLSql_Microsoft_error	InstLSql_Microsoft.error	Bool
SqlCommands_Tuote	SqlCommands.TuoteHMI	String
SqlCommands_Lajittelu	SqlCommands.LajitteluHMI	String
HmiInterface_update	HmiInterface.update	Bool
SqlRe-ceive_data_tokenRows{0}_data{0}	SqlRe-ceive.data.tokenRows[0].data[0]	Char
SqlCommands_SQLvastaus	Sql_Vastaukset.THIS	Array [0..9] of String
SqlCommands_SQLvastaus{0}	Sql_Vastaukset.THIS[0]	String
SqlCommands_SQLvastaus{1}	Sql_Vastaukset.THIS[1]	String
SqlCommands_SQLvastaus{3}	Sql_Vastaukset.THIS[3]	String
SqlCommands_SQLvastaus{2}	Sql_Vastaukset.THIS[2]	String
SqlCommands_SQLvastaus{4}	Sql_Vastaukset.THIS[4]	String
SqlCommands_SQLvastaus{9}	Sql_Vastaukset.THIS[9]	String
SqlCommands_SQLvastaus{5}	Sql_Vastaukset.THIS[5]	String
SqlCommands_SQLvastaus{8}	Sql_Vastaukset.THIS[8]	String
SqlCommands_SQLvastaus{7}	Sql_Vastaukset.THIS[7]	String
SqlCommands_SQLvastaus{6}	Sql_Vastaukset.THIS[6]	String

LIITE 2 YLEISIMMÄT STEP 7 DATATYYPIT

Taulukko 2 Step 7 Elementary Data Types (*Step 7 Elementary Data Types*, 2022.)

Type and Description	**Size in Bits**	**Format Options**	**Range and Number Notation (lowest to highest values)**	**Example in STL**
BOOL (Bit)	1	Boolean text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadecimal number	B#16#0 to B#16#FF	L B#16#10 L byte#16#10
WORD (Word)	16	Binary number	2#0 to 2#1111_1111_1111_1111	L 2#0001_0000_0000_0000
		Hexadecimal number	W#16#0 to W#16#FFFF	L W#16#1000 L word#16#1000
		BCD	C#0 to C#999	L C#998
		Decimal number unsigned	B#(0,0) to B#(255,255)	L B#(10,20) L byte#(10,20)
DWORD (Double word)	32	Binary number	2#0 to 2#1111_1111_1111_1111_ 1111_1111_1111_1111	L 2#1000_0001_0001_1000_1011_1011_0111_1111
		Hexadecimal number	W#16#0000_0000 to W#16#FFFF_FFFF	L DW#16#00A2_1234 L dword#16#00A2_1234
		Decimal	B#(0,0,0,0) to	L B#(1, 14, 100, 120)

		number unsigned	B#(255,255,255,255)	L byte#(1,14,100,120)
INT (Integer)	16	Decimal number signed	-32768 to 32767	L 101
DINT (Double integer)	32	Decimal number signed	L#-2147483648 to L#2147483647	L L#101
REAL (Floating-point number)	32	IEEE Floating-point number	Upper limit +/- 3.402823e+38 Lower limit +/- 1.175495e-38	L 1.234567e+13
S5TIME (SIMATIC time)	16	S7 time in steps of 10ms (default)	S5T#0H_0M_0S_10MS to S5T#2H_46M_30S_0MS and S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC time)	32	IEC time in steps of 1 ms, integer signed	T#24D_20H_31M_23S_648MS to T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (IEC date)	16	IEC date in steps of 1 day	D#1990-1-1 to D#2168-12-31	L D#1996-3-15 L DATE#1996-3-15
TIME _OF_DAY (Time)	32	Time in steps of 1 ms	TOD#0:0:0.0 to TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Character)	8	ASCII characters	A', 'B' etc.	L 'E'