**Markus Hosionaho**

# TEOLLISUUSLOGIIKAN TIETOTURVA

## Siemens Simatic S7-1200 verkkohyökkäyksen kohteena

**Opinnäytetyö**
**CENTRIA-AMMATTIKORKEAKOULU**
**Sähkö- ja automaatiotekniikan koulutusohjelma**
**Kesäkuu 2016**

**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

| **Yksikkö**<br>Ylivieska | **Aika**<br>Kesäkuu 2016 | **Tekijä/tekijät**<br>Markus Hosionaho |
|---|---|---|
| **Koulutusohjelma**<br>Sähkö- ja automaatiotekniikka | | |
| **Työn nimi**<br>TEOLLISUUSLOGIIKAN TIETOTURVA. Siemens Simatic S7-1200 verkkohyökkäyksen kohteena | | |
| **Työn ohjaaja**<br>Joni Jämsä | | **Sivumäärä**<br>37 + 8 |
| **Työelämäohjaaja**<br>Anita Rättyä | | |


Tässä opinnäytetyössä selvitettiin millaisia asioita tulee ottaa huomioon teollisuuslogiikkoja asennettaessa. Käytännön osana työssä rakennettiin 16:sta Siemens Simatic S7-1200 -ohjelmoitavasta logiikasta koostuva kokeilualusta Centrian ammattikorkeakoulun tietoliikennelaboratorioon, jossa logiikkoihin yritettiin hyökätä oululaisen Rugged Tooling Oy:n työkaluilla.

Kokeilualusta rakennettiin tietokonepöydän ylähyllylle, johon asennettiin DIN-kiskot ja 6-osaiset pistorasiat. Lisäksi rakennettiin logiikkojen toiminnan seuraamista helpottava LED-rima, jolla indikoidaan logiikkojen ulostuloja. Logiikoille ohjelmoitiin Siemensin TIA-Portal -ohjelmistolla ohjelma, joka kytkee kytkintä painamalla ensimmäisen ulostulon päälle kolmeksi sekunniksi, jonka jälkeen viesti kulkee logiikalta toiselle niin, että yhden logiikan ulostulo on kerrallaan päällä. Viimeisen logiikan jälkeen sekvenssi lähtee taas uudelleen ensimmäisestä logiikasta.

Denial of Service -tyyppisellä hyökkäyksellä saatiin logiikat hidastumaan, jos ne oli kytketty pienempään lähiverkkokytkimeen, mutta Cisco:n lähiverkkokytkin osasi suodattaa liikennettä hyvin, eikä viiveitä saatu aikaan DoS-hyökkäyksellä. Joitain tietoturvahaavoittuvuuksia löydettiin.



**Asiasanat**
teollisuuslogiikka, tietoturva, ohjelmoitava logiikka, siemens, simatic, s7-1200

## ABSTRACT


| CENTRIA UNIVERSITY OF APPLIED SCIENCES<br>Ylivieska | Date<br>June 2016 | Author<br>Markus Hosionaho |
|---|---|---|
| Degree programme<br>Electric and automation technology | | |
| Name of thesis<br>INDUSTRIAL LOGIC'S INFORMATION SECURITY. Siemens Simatic S7-1200 targeted by a network attack | | |
| Instructor<br>Joni Jämsä | | Pages<br>37 + 8 |
| Supervisor<br>Anita Rättyä | | |



In this thesis we pieced together information about what should we take in account when installing industrial logic's concerning information security. Practical phase of the work consisted of building a demonstration setup with 16 Siemens Simatic S7-1200 programmable logic controllers in a laboratory at Centria University of Applied Sciences. This demonstration setup was used to test the logic system's security with Rugged Tooling security testing tools.

Demonstration setup was built on the top shelf of a computer desk, where we installed DIN-rails for the logic controllers and 6-slot power sockets for the needed power cables. We also built a 16 LED case to indicate the outputs for easier readability of the logics outputs. Logic program was made with Siemens' TIA-Portal software package. The program turns first output on for three seconds when switch is turned and when message goes through the system one output is on from each logic at a time for three seconds. After last logic controller, the sequence will start again from the beginning unless stopped with another switch.

We were able to slow the logics and then halt them completely when they were installed on a smaller Ethernet switch, but when connected to Cisco's Ethernet switch, it was able to filter the data stream well enough and we couldn't create any delays on the logic system with DoS attacks. Some vulnerabilities were found when scanning the system.



**Key words**
industrial logic, information security, programmable logic, siemens, simatic, s7-1200

**KÄSITTEIDEN MÄÄRITTELY**

| | |
|---|---|
| ABB | ASEA Brown Boveri |
| CAT-6 | Category 6 cable (kierretty parikaapeli) |
| DB | Data Block (tietokanta) |
| DIN | Deutsches Institut für Normung (standardi) |
| DoS | Denial of Service (palvelunestohyökkäys) |
| DDoS | Distributed Denial of Service (hajautettu palvelunestohyökkäys) |
| FBD | Function block diagram (ohjelmointikieli) |
| FC | Function |
| GMT | Greenwich Mean Time |
| I/O | Input/Output (Sisääntulo/Ulostulo) |
| IL | Instruction List |
| IP | Internet Protocol address |
| IT | Information technology |
| LAD | Ladder logic (ohjelmointikieli) |
| LED | Light-emitting diode |
| MAC | Media Access Control address |
| NTP | Network Time Protocol |
| OB | Organization Block |
| PG/PC | Programmierung Gerät / Personal Computer |
| PLC | Programmable Logic Controller (ohjelmoitava logiikka) |
| PLD | Programmable Logic Devices (ohjelmoitavat laitteet) |
| SCL | Structured control language (ohjelmointikieli) |
| SFC | Sequential Function Chart |
| SIL | Safety Integrity Level |
| ST | Structured Text |
| STL | Statement list (ohjelmointikieli) |
| TCP | Transport Control Protocol (tietoliikenneprotokolla) |
| TIA Portal | Totally Integrated Automation -ohjelmisto |
| UDP | User Datagram Protocol (tietoliikenneprotokolla) |
| UTC | Universal Time Coordinated |
| VPN | Virtual Private Network |

**TIIVISTELMÄ**
**ABSTRACT**
**KÄSITTEIDEN MÄÄRITTELY**
**SISÄLLYS**



**KUVAT**

# 1 JOHDANTO

Opinnäytetyönä rakennettiin Centria ammattikorkeakoulun tietoliikennetekniikan laboratorioon tietoturvan kokeilualusta 16:sta ohjelmoitavasta logiikasta. Logiikkana toimi Siemens Simatic S7-1200, johon ohjelmoitiin ohjelma, joka kytkee ensimmäisen ulostulon päälle kytkintä painamalla. Tämän jälkeen viesti kulkee kolmen sekunnin välein seuraavalle logiikalle jne. aloittaen taas alusta, kun viimeisen logiikan ulostulo on ollut 3 sekuntia päällä.

Opinnäytetyössä käytiin myös läpi tietoturvaan liittyviä asioita, kuitenkin menemättä liian syvälle teknisiin yksityiskohtiin. Tutkittavana aiheena oli ohjelmoitavien logiikkojen historia ja kuinka tietoturva on kehittynyt verrattuna muihin tietotekniikan sovelluksiin. Tavoitteena oli tehdä tutkimus, jonka pohjalta niin automaation kuin tietoturvan ammattilaiset voivat lähteä pohtimaan ratkaisuja tietoturvakysymyksiin, joita tekniikan nopea kehittyminen ja lisääntyvät tietoturvariskit tuovat tullessaan. Millaisia ratkaisuja ohjelmoitavien logiikkojen ja muun automaation tietoturvan suhteen tehdään nyt ja tulevaisuudessa? Riittävätkö nykyiset tietoturvaratkaisut, vai olisiko jokin uusi tapa turvata järjestelmät hyökkääjien varalta? Kuinka kauan voidaan paikata vanhemmista laitteista löydettyjä tietoturva-aukkoja tai lisätä tietoturvaa lisääviä laitteita ja ominaisuuksia? Kuinka saadaan eri ammattialojen asiantuntijat toimimaan yhteistyössä laajemmin, jotta saadaan yksinkertainen ja mahdollisimman tietoturvallinen ratkaisu myös ohjelmoitaviin logiikkoihin?

Täysin suojattua järjestelmää ei voida rakentaa, koska on aina olemassa se riski, että joku tai jokin pääsee suojauksista läpi, jolloin on hyvä kiinnittää huomiota riskien hallintaan. Näihin asioihin löytyikin runsaasti materiaalia, joista ehkä huomiota herättävin ajatus oli, että miten olemme voineet sallia automaatiojärjestelmien olevan näin pitkään haavoittuvina. Pahimmassa tilanteessa järjestelmän haavoittuvuus voi johtaa ihmishenkien menetykseen. Tällä hetkellä tietoturvaratkaisut on hajautettu moneen kerrokseen, jolloin järjestelmään tunkeutuminen on hankalampaa, mutta usein järjestelmän suojaus on vain yhtä vahva kuin sen heikoin lenkki.

## 2 OHJELMOITAVAN LOGIIKAN TOIMINTAPERIAATE

Ohjelmoitavista logiikoista on tullut yleisimpiä ohjauslaitteita niiden toimintojen määrän ja suorituskyvyn kasvaessa. 2000-luvun alussa maailman logiikkamarkkinoita hallitsevat merkittävimmin monikansalliset yritykset kuten Siemens, Mitsubishi, Omron, Allen Bradley ja GE Fanuc. Ohjelmoitavia logiikkoja käytetään mm. kokoonpanolinjojen, pakkaus- ja lajittelukoneiden toistuvissa tapahtumissa ja niillä voidaan myös hallita yksittäisten laitteiden lisäksi koko tehtaan laajuisia järjestelmiä. Alun perin logiikat kehitettiin korvaamaan releohjauksen, koska releohjaus ei ollut tarpeeksi muutoskykyinen tuotannon muuttuessa. (Automaatiolaitteet, 1999, 102).

Control -lehti listaa maailman ohjelmoitavien logiikkojen markkinajohtajia vuosittain. Siemens hallitsee yhä maailman logiikkamarkkinoita selvästi, mutta markkinoille on tullut myös uusia haastajia, kuten ABB ja Fluke. Investointeja vaaditaan automaation lisääntyessä ja erityisesti tietoturvaan tulee panostaa, koska se voi olla ratkaiseva tekijä monelle yritykselle automaatiojärjestelmiä valittaessa.

Logiikat voivat olla rakenteeltaan pieniä, joilla ohjataan yksittäisiä laitteita ja sisältävät yleensä 10 - 30 tuloa/lähtöä eli input/output:ia, lyhyemmin I/O. Suurilla logiikoilla voidaan ohjata jopa kymmeniä tuhansia tuloja/lähtöjä ja näitä käytetäänkin kokonaisten tehtaiden ohjaukseen. (Automaatiolaitteet, 1999, 106).

Ohjelmoitavat logiikat ovat siirtymässä enenevissä määrin teolliseen internetiin, jolloin logiikkojen tulo/lähtö -määrällä ei ole enää niin suurta merkitystä, koska pienetkin logiikat voivat nyt keskustella keskenään ja toimittaa yhdessä suuren logiikan virkaa koko tehtaan automatisoinnissa. Tästä seuraa tietenkin tietoturvariskejä, koska aiemmin suuri logiikka on tehnyt laskentatyötä ja päätöksiä itsenäisesti, jolloin järjestelmän häirintä oli vaikeampaa. Palomuurien, käyttäjätunnusten ja salasanojen tehtävänä on huolehtia, etteivät ulkopuoliset henkilöt pääse dataan käsiksi.

Logiikka sisältää keskusyksikön eli CPU:n, jonka sisällä on muistialueita erikoismuistille, datalle ja ohjelmalle sekä kolme sisäistä prosessoria kommunikoinnille ja laskentatoimenpiteille. Logiikassa voi olla myös useita keskusyksiköitä. Siemens S7-1200 -logiikan teholähde tuottaa 24 V tasajännitettä, joka voidaan syöttää logiikan ulostuloille. Sisääntulojen tehtävänä on välittää tosi/epätosi eli true/false -tietoa, erottaa logiikka galvaanisesti, sovittaa anturijännitteet logiikan jännitteeseen sekä suojata sitä häiriöiltä. Lähdöt välittävät tietoa toimilaitteille ja sovittavat jännitteet logiikan ja toimilaitteiden käyttöön sopiviksi. (Automaatiolaitteet, 1999, 108).

Ohjelmoitavia logiikkoja on saatavilla useilta valmistajilta. Tämän opinnäytetyön logiikoiksi valittiin saksalaisen Siemens AG:n valmistamat Simatic S7-1200 -logiikat, joiden tarkempi malli on S7-1214C AC/DC/Rly. Mallimerkinnässä AC/DC/Rly, tarkoittavat syöttö-, sisääntulo- ja ulostulosjännitteitä sekä relettä. Syöttöjännite otetaan pienjänniteverkosta ja valmistaja lupaa logiikan kestävän korkeintaan 264 V syöttöjännitettä.

Maailmalla käytetyissä digitaalijärjestelmissä on kolmen tyypin laitteita: muistit, mikroprosessorit ja logiikat. Muistit tallentavat satunnaista tietoa, kuten taulukoita tai muistipankkeja. Mikroprosessorit suorittavat ohjelmallisia komentoja laajamittaisissa prosesseissa, kuten esimerkiksi tekstinkäsittelyohjelmassa tai videopelissä. Logiikkalaitteet tarjoavat erikoistoimintoja, kuten laitteelta laitteelle tapahtuvaa yhteistyötä, datakommunikointia, signaaliprosessointia, datan indikoimista, ajastimia ja ohjaustoimintoja ja melkein kaikkia muitakin toimintoja, joita järjestelmä tarvitsee toimiakseen. Logiikkalaitteet voidaan jaotella kahteen laajaan ryhmään, ennalta määrätyt ja ohjelmoitavat logiikat. Kuten nimikin vihjaa, piirit ennalta määrätyssä logiikassa ovat pysyviä, jotka toteuttavat yhden tai useamman toiminnon, eikä niitä voi valmistuksen jälkeen muuttaa. Ohjelmoitavat logiikkalaitteet (engl. PLD, programmable logic devices) ovat vakiokokoonpanolaitteita, jotka voidaan tilata sellaisenaan suoraan hyllystä ja niitä voidaan muokata haluttaessa suorittamaan erilaisia toimintoja. (Xilinx Inc., 2016). Ohjelmoitavien logiikkojen hyöty tulee parhaiten esiin käyttökohteissa, joissa tehdään muutoksia järjestelmään sekä monimutkaisissa automaatiojärjestelmissä, jolloin johtojen kytkeminen on helpompaa ja mahdollisten vikojen korjaaminen on nopeampaa.

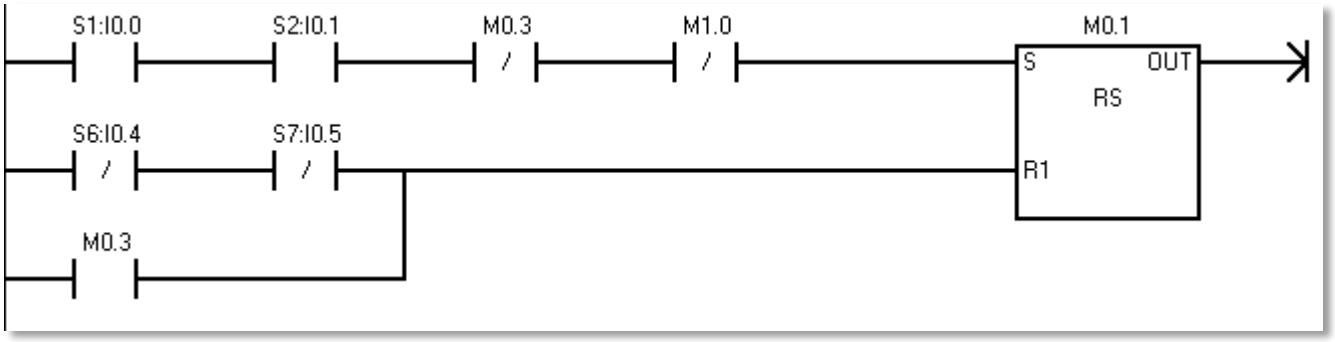
Verrattuna aiempiin S7-300/400 -sarjan ohjelmoitaviin logiikkoihin, S7-1200/1500 -sarjan logiikat tarkistavat ohjelman compile-vaiheessa, jotta ohjelma toimii odotetusti. Tästä syystä S7-1200/1500 -logiikat sietävät virheitä aikaisempia versioita paremmin. Jos halutaan muuttaa jo käytössä olevia data blokkeja, ei S7-1200/1500 -sarjan logiikkoja tarvitse pysäyttää lataamisen ajaksi. Data blokkeihin voi lisätä rivejä ja ladata ne logiikkaan ilman, että siellä jo olevat tiedot muuttuvat. (Programming Guideline for S7-1200/S7-1500, 35, 46)

Logiikka suorittaa ohjelmassa olevat network-koodipätkät eli piirit useita kertoa sekunnissa. S7-1200 -logiikan suoritusnopeus bittioperaatiolle on nopeimmillaan 0,085 mikrosekuntia, eli sekunnissa se voi suorittaa noin 11,7 miljoonaa bittioperaatiota. Tällaisia operaatioita ovat esimerkiksi sisääntulojen tilojen tarkistaminen ja muistiin kirjaaminen, ohjelman lukeminen ja ulostulojen tilojen vaihtaminen, mikäli ohjelmassa asetellut ehdot täyttyvät.

Logiikkaan tallennettu ohjelma sijaitsee logiikan ohjelmamuistissa. Toiminta perustuu pääsääntöisesti kolmeen tapahtumaan. Ensimmäiseksi logiikka lukee sisääntulojen tilat muistiin. Seuraavaksi luetaan ohjelmakoodi, jossa ulostuloille on määritelty ehtoja, jolloin niiden tulee olla päällä ja tilat tallennetaan ulostulojen muistialueelle. Viimeisessä vaiheessa luetaan ulostulojen tilat ja muutetaan niiden tilaa riippuen ohjelmassa esiintyvistä määräyksistä. Laskelmat ja päätöksentekotoimenpiteet suoritetaan logiikan prosessorissa.

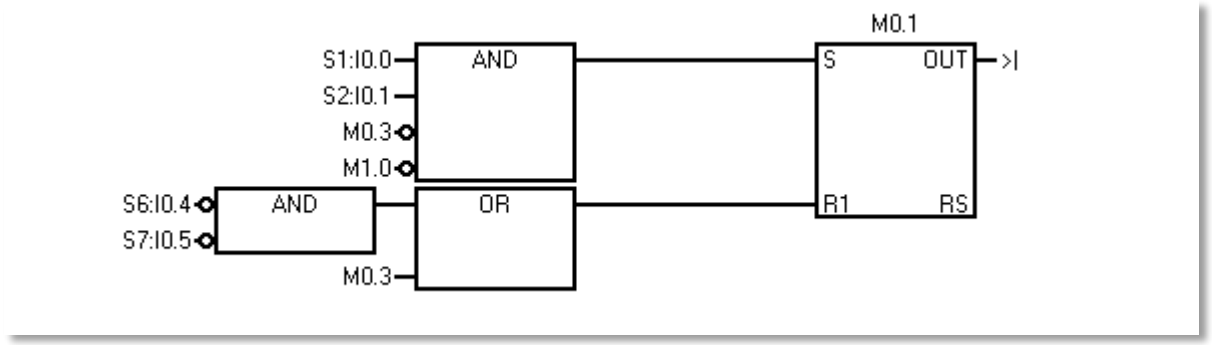
Logiikan sisääntuloja merkitään Siemensin logiikoissa kirjaimella I, jonka jälkeen seuraa numerointi 0.0, 0.1 jne. riippuen sisääntulojen määrästä. Ulostulojen merkintä on samanlainen, mutta etumerkkinä käytetään kirjainta Q, esim. Q0.0. Lisäksi ohjelmakoodissa käytetään usein logiikan bittimuistialuetta, johon viitattaessa käytetään merkintää M0.0 jne. aivan kuten sisään- ja ulostulojen tapauksessa. Näissä merkinnöissä käytettävät numerot ovat ”byte.bit” eli tavu ja bitti. Logiikka siis tallentaa sisääntulojen tilat muistialueelle I, ulostulojen tilat muistialueelle Q ja bittimuistien tilat muistialueelle M. Logiikan muistissa on 8 bitin mittaisia tavuja, joihin tallennetaan bittiarvoja 1 tai 0. Ohjelmoitaessa voidaan kuitenkin nimetä nämä muistialueet symbolitaulukkoon sisään-/ulostuloon liittyvää anturia tai moottoria paremmin kuvaavalla nimellä. Tällöin ohjelmakoodista tulee ymmärrettävämpää. Ohjelmassa olevia piirejä voidaan myös kommentoida sen yläpuolella olevaan tekstikenttään, jolloin vikatilanteessa on nopeampaa nähdä missä piirissä vika mahdollisesti on.

Mikäli logiikka havaitsee ensimmäisen sisääntulon I0.0 olevan tosi ja ohjelmakoodista luetaan ehto, jonka mukaan ulostulo Q0.0 pitää olla tosi, kun sisääntulo I0.0 on tosi, logiikka tallentaa muistipaikkaan Q0.0 bittiarvon yksi. Kolmannessa vaiheessa logiikka lukee ulostulojen tämänhetkisen tilan ja tekee muutoksen ulostulon tilaan, koska havaitsee muistissa Q0.0 bittiarvon yksi. Tämän jälkeen logiikka toteuttaa muutokset eli kytkee ensimmäisen ulostulon päälle. Tämä kaikki tapahtuu niin nopeasti, että toiminta on käytännössä viiveetön laajoillakin ohjelmilla. Koska logiikka toimii tällä tavalla, koodia ei voida kirjoittaa perustoiminnoilla samalla tavalla kuin esimerkiksi perinteisillä tietokoneiden ohjelmointikielillä, jossa ohjelmassa olevien käskyjen väliin on lisätty esimerkiksi viiveitä ja muita toimintoja. Mikäli esimerkiksi asetamme ohjelman käynnistämään ulostulon ensimmäisellä rivillä sisääntulon ollessa tosi, mutta myös käynnistämään sen viimeisellä rivillä, jos sisääntulo on epätosi, ulostulo ei käy päällä ollenkaan. Tämä johtuu siitä, että logiikan ohjelmassa viimeinen ulostulolle aseteltu ehto on määräävä ja kumoaa aiemmin ohjelmakoodissa määrätyt komennot.




KUVA 1. Ladder (LD) -esitystapa

Logiikan prosessorissa oleva ohjelma kirjoitetaan yleisimmin graafisella Ladder-esitystavalla eli relekaaviolla, jota käytettiin myös tässä opinnäytetyössä. Ohjelmointityökalu on Siemensin ohjelmoitavilla logiikoilla MicroWin STEP 7, josta tämän hetkinen versio on 13. Ohjelmakoodia tehtäessä piireihin lisätään bittilogiikkakuvakkeita, jotka indikoivat, onko ehto tosi vai epätosi. Normaalisti auki oleva kytkin on kiinni ollessaan tosi ja auki ollessaan epätosi. Normaalisti kiinni oleva kytkin on kiinni ollessaan epätosi ja auki ollessaan tosi jne. Kuvassa 1 sisääntulot I0.0 ja I0.1 on nimetty S1- ja S2-painikkeiksi ohjelman symbolitaulukkoon ja ovat normaalisti auki olevia kytkimiä. Sisääntulot I0.4 ja I0.5 on nimetty S6- ja S7-painikkeiksi ja ovat normaalisti kiinni olevia kytkimiä. Tämän lisäksi voidaan käyttää erilaisia laskureita, ajastimia, matemaattisia operaatioita yms. Kuvan 1 esimerkkiohjelmassa on käytetty myös muistibittejä eli memorybittejä M0.1, M0.3 ja M1.0.




KUVA 2. Function Block Diagram (FBD) -esitystapa

```
LD      S1:I0.0
A       S2:I0.1
AN      M0.3
AN      M1.0
LDN     S6:I0.4
AN      S7:I0.5
O       M0.3
NOT
LPS
A       M0.1
=       M0.1
LPP
ALD
O       M0.1
=       M0.1
```

KUVA 3. Statement List (STL) -esitystapa

Ohjelman esitystapoja on Ladder:n lisäksi muitakin ja usein ohjelmassa yhdellä tavalla tehty ohjelma voidaan näyttää myös muilla esitystavoilla.

Kuvien 1-3 esimerkeissä Reset-Set -kiikkua, jonka arvo tallennetaan memorybit M0.1:een, ohjataan painamalla yhtäaikaisesti painonappeja S1 ja S2. M0.3 ja M1.0 arvoja muutetaan ohjelman myöhemmässä vaiheessa, mutta tässä networkissa molempien arvon tulee olla epätosi, jotta M0.1 arvoksi tallennetaan tosi. M0.1 resetoidaan takaisin arvoon epätosi joko S6- ja S7-rajakytkimillä tai kun M0.3 on tosi.

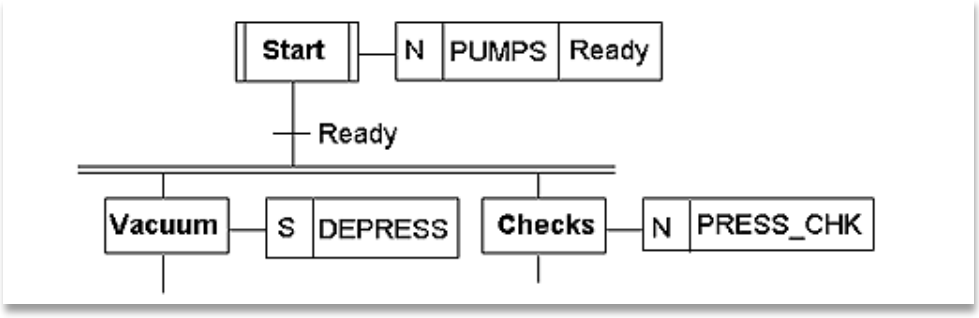
Structured Text, ST, on korkean tason tekstipohjainen kieli. Se muistuttaa suuresti PASCAL-ohjelmointikieltä ja tukee laajasti vakiofunktioita ja operaatioita. Kuvan 4 instruction list -esitystapaa kutsutaan Siemensin logiikoissa statement list -esitystavaksi.

```
      LD   R1
     MPC   RESET
      LD   PRESS_1
      ST   MAX_PRESS
RESET:  LD   0
      ST   A_X43
```

KUVA 4. Instruction List (IL) -esitystapa

```
If Speed1 > 100.0 then
   Flow_Rate: = 50.0 + Offset_A1;
Else
   Flow_Rate: = 100.0; Steam: = ON
End_If;
```

KUVA 5. Structured Text (ST) -esitystapa




KUVA 6. Sequential Function Chart (SFC) -esitystapa

Logiikkaan liitettäviä sisääntuloja ovat mm. erilaiset kytkimet, painonapit, rajakytkimet, lähestymiskytkimet, paine- ja alipainekytkimet, lämpökytkimet yms. anturit. Ulostuloihin liitetään mm. venttiilit, käynnistysmoottorit, solenoidit, summerit, hälyttimet, valot, ohjausreleet, laskurit, pumput, printterit ja tuulettimet. Usein logiikan ulostulot ovat potentiaalivapaita, joten niille täytyy syöttää jännite esimerkiksi logiikan omasta 24 V ulostulosta. Mikäli halutaan kytkeä logiikalla 3-vaihemoottori päälle, voidaan moottorin suurempi käyttöjännite kuljettaa ulkoisen releen pääkoskettimien kautta ja logiikan ulostulolla kytketään 24 V rele päälle. Pitopiiri voidaan ohjelmoida logiikkaan.

# 3 TIETOTURVA

Nykyaikainen automaatiosuunnittelu pohjautuu tiedonsiirtoon ja yksittäisten tuotantosolujen verkottamiseen suuremmiksi kokonaisuuksiksi. Etäyhteydet huollolle ja lisääntyvä IT-palvelujen, kuten Web-palveluiden ja sähköpostin hyödyntäminen ohjelmoitavissa logiikoissa puoltavat tuotantolaitteiden yhdistämistä ja liittämistä toimistoverkkoon. Tämän yhdistämisen myötä myös tietoturvariskit kasvavat. (Siemens)

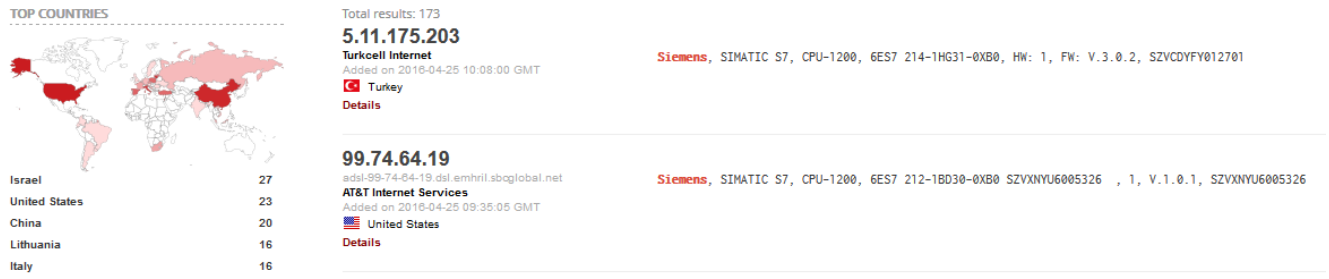
Ohjelmoitavat logiikat ovat olleet käytössä jo 1960-luvun alusta asti. Näitä alettiin käyttämään pian sen jälkeen, kun mikroprosessorit keksittiin, koska se antoi yrityksille mahdollisuuden korvata suuren määrän tuotantoautomaatiota ohjaavia releitä. Näitä relepaneeleja oli vaikea muokata, ylläpitää ja ne olivat haastavia korjata vikatilanteissa. 1970-luvulla huomattiin, että tuotannon tehokkuuden monitoroimiseksi halutaan lukea logiikoilta dataa. Vielä 20 vuotta sitten tietoturva oli käsitteenä eri asia kuin nykyisin, eikä sen ajan ohjelmoitavissa logiikoissa huomioitu tietoturvaa mitenkään. (Belden, 2016, luettu 4.4.2016)

Ensimmäinen ohjelmoitava logiikka kehitettiin Bradley Associates -yrityksen toimesta autoteollisuuteen GM Hydromatic -automaattivaihteistoja valmistavalle tehtaalle vuonna 1968. Logiikan nimeksi tuli Modicon 084, joka tulee sanoista Modular Digital Controller ja se oli yhtiön 84. projekti. Modicon käytti ladder- eli relekaavio-ohjelmointikieltä, joka on edelleen käytetyin ohjelmoitavissa logiikoissa. (plcmentor.com, luettu 25.4.2016)

Suomessa ollaan havahduttu tietoturvariskeihin Stuxnetin kaltaisten hyökkäysten seurauksena. Asiaa on tutkittu mm. Aalto-yliopiston toimesta kartoittamalla yleisen internetin kautta löydettäviä Suomessa käytössä olevia tehdasautomaatiojärjestelmiä. Tätä kartoitusta voidaan tehdä työkaluilla kuten Shodan, joka analysoi löytämiään internetissä kiinni olevia laitteita ja tallentaa ne tietokantaansa. Shodanin löytämät laitteet ovat sellaisia, joihin kuka tahansa voi ottaa yhteyden internetin yli ja siihen murtautumisen helppous riippuu laitteen ja järjestelmän tietoturvaratkaisuista. Tästä kerätystä tietokannasta voidaan etsiä ne järjestelmät, joihin rajoittamatonta pääsyä ei pitäisi olla, kuten esimerkiksi automaatiojärjestelmät. Löydetyn laitteen nimen ja IP-osoitteen avulla voidaan selvittää osoitteen omistava yritys ja arvata näiden tietojen perusteella laitteen käyttötarkoitus. Hälyttävää oli, että tutkijat löysivät useita laitteita, joiden käyttäjätunnukset ja salasanat oli tallennettuna web-käyttöliittymään, joten kuka tahansa olisi voinut kirjautua järjestelmään sisälle. (Suomen automaatioverkkojen haavoittuvuus, 2013, luettu 12.4.2016)

Tammikuussa 2013 Aalto-yliopiston tutkijat löysivät Shodanilla Suomesta yhteensä 2915 laitetta, jotka kuuluivat erilaisiin teollisuuden automaatiojärjestelmiin, rakennusautomaatioon, sähkönhallintaan ja järjestelmien etäkäyttöön. Yhteensä löydettiin 185 000 http-vastausta antavaa laitetta ja määrä on jatkuvassa kasvussa. Shodan ei kuitenkaan ollut tuossa vaiheessa vielä skannannut kaikkia suomalaisia IP-osoitteita, vaan arviolta vasta muutaman kymmenen prosentin kokonaismäärästä. Suomessa havaittiin olevan Shodanin löytämiä haavoittuvia automaatiolaitteita suhteutettuna väkilukuun enemmän kuin muissa maissa. Täytyy kuitenkin ottaa huomioon, että Shodanin tietokantaa ei tyhjennetä koskaan, joten kerran sinne päässyt laite pysyy tietokannassa, vaikka haavoittuvuus olisikin jo korjattu. Silti näitä tuloksia voidaan pitää suuntaa antavina. (Suomen automaatioverkkojen haavoittuvuus, 2013, luettu 12.4.2016)

Shodan-sivustolta voi tehdä hakuja mm. logiikan nimen perusteella, jolloin tietokantaan tallennetut IP-osoitteet ja laitteiston nimi sekä versiotiedot tulevat näkyville. Lisätietojen takaa löytyy jopa karttakuva, joka näyttää missä laite sijaitsee.




KUVA 7. Shodan

Usein ajatellaan, että teollisuusverkon irrottaminen internetistä tekisi järjestelmästä suojatun, mutta tämä ei pidä paikkaansa. Tietoturvariskejä ei voida poistaa, mutta ne voidaan pyrkiä minimoimaan, jolloin uhkana on vain muutama tarpeeksi pätevä ja motivoitunut hyökkääjä. Hyökkääjä voi käyttää hyväkseen työntekijöiden tekemiä virheitä. Tunkeutuja voi esimerkiksi kirjoittaa hyökkäysohjelman tietokonematoon, jonka välittää kohteena olevaan teollisuuslähiverkkoon USB-tikun tai muun työntekijän käyttämän laitteen kautta. Näin tapahtui Kesäkuussa 2010, kun 500 kb kokoinen Stuxnet-mato saastutti ainakin 14 teollisuusaluetta Iranissa (Spectrum, 2013). Ensin mato otti kohteekseen Microsoft Windows -järjestelmät ja verkot jatkuvasti tehden itsestään kopioita. Sen jälkeen se otti kohteekseen Siemens STEP7 -ohjelmiston, joka on myös Windows-pohjainen ohjelma. Lopulta Stuxnet-mato saastutti ohjelmoitavat logiikat. Tässä tapauksessa suurin tietoturvariski oli siis käyttäjä itse, joka tietämättään avasi

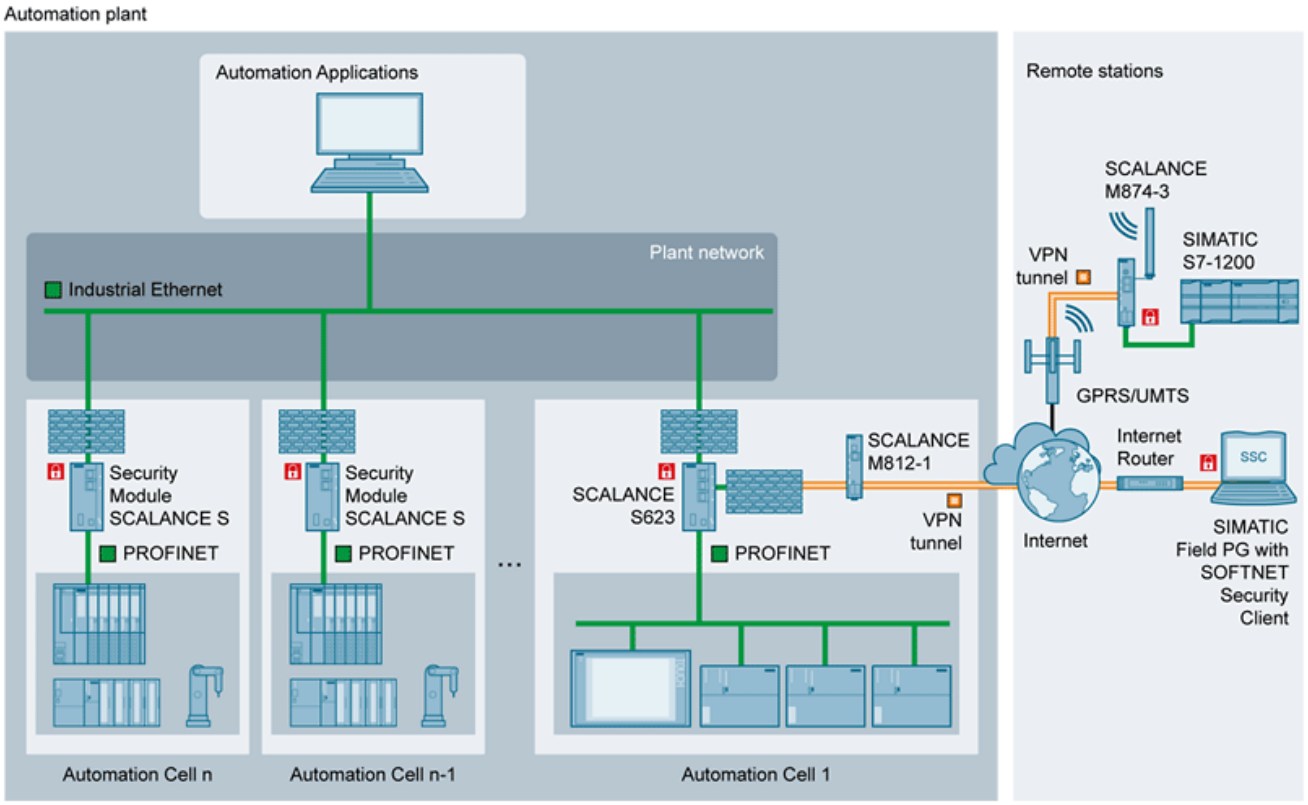
madolle reitin järjestelmään. Stuxnet aiheutti tuhoa voimalan sentrifugeissa (Suomen automaatioverkkojen haavoittuvuus, 2013, luettu 12.4.2016).

1990-luvulla haittaohjelmia pyrittiin havaitsemaan manuaalisesti, mutta 2000-luvulle siirryttäessä alettiin käyttämään automatisoitua havaintomenetelmää, jolloin alettiin löytämään jopa 250 000 uutta haittatiedostoa päivittäin. Vuonna 2010 valkovenäläinen haittaohjelmia etsivä yritys sai työkseen tutkia asiakkaan järjestelmää, joka uudelleen käynnistyi itsestään jatkuvasti. Haittaohjelman havaittiin sisältävän digitaalisen sertifikaatin, jotta se näytti tulevan luotettavalta yritykseltä. Tätä ei automaattiset suojausohjelmat kyenneet havaitsemaan. (Spectrum, 2013)

Lokakuussa 2012 Yhdysvaltain puolustusministeri Leon Panetta varoitti, että maa on haavoittuvainen ”cyber Pearl Harbor”:iin, joka voisi johtaa junien radalta suistumisiin, vesilähteiden saastumiseen ja rampauttaa sähköverkon. Kuukautta myöhemmin Chevron vahvisti spekulaatiot ja oli ensimmäinen yhdysvaltalainen yritys, joka myönsi Stuxnetin levinneen heidän järjestelmäänsä. (Spectrum, 2013)

Ohjelmoitaviin logiikkoihin saa myös usein asennettua valmistajan erillisen tietoturvamoduulin, joilla luvataan parempaa suojausta kuin perinteiset toimistoverkoissa käytössä olevat suojaukset tarjoavat. ”Olemassa olevat tietoturvajärjestelmät on kehitetty toimistomaailman tarpeisiin ja vaativat laajaa alan erityisosaamista sekä jatkuvaa ylläpitoa. Ne eivät myöskään pysty käsittelemään teollisuuden tiedonsiirron erityisprotokollia saati teollisuusympäristöä. Teollisuuden tietoturvajärjestelmän SCALANCE S -tietoturvamoduuleilla voidaan vastata teollisuuden asettamiin erityisvaateisiin.” (Siemens)

Siemens tarjoaa asiakkailleen SCALANCE S -tietoturvamoduuleja, jotka suojaavat automaatioverkkoa luvattomilta verkkoon liittymisiltä ja tarpeettomalta tiedonsiirtokuormalta. Ulkoiseen verkkoon ilmaantuneet häiriöt eivät aiheuta SCALANCE S -moduulilla suojattuun automaation aliverkkoon vikaantumista. SCALANCE S -moduuleista on kolme eri variaatiota: palomuuri S602 ja palomuurit erilaisilla määrillä VPN-yhteyksiä S612 ja S613. (Siemens)




KUVA 8. Esimerkkikaaviokuva Scalance S -modulin käytöstä (Siemens)

Teollisuusautomaatiojärjestelmän tietoturva perustuu usean osa-alueen yhdessä luovasta tietoturvasta. Näitä osa-alueita ovat mm. palomuuri, virustorjunta, käyttöjärjestelmä, lähiverkkokytkin ja ohjelmoitava logiikka. Jokainen osa-alue huolehtii omalta osaltaan tietoturvasta ja niiden toimintavarmuus riippuu valmistajasta. Palomuuri pyrkii estämään hyökkäysten pääsyn järjestelmään, virustorjunta havaitsee ja poistaa virukset, käyttöjärjestelmän valmistaja huolehtii oman ohjelmistonsa tietoturva-aukkojen paikkaamisesta, lähiverkkokytkin suodattaa ylimääräistä tietoliikennettä ja ohjelmoitavan logiikan valmistaja huolehtii logiikan sisäisen tietoturvan toiminnasta. Automaatiojärjestelmän suunnittelijan tai laitteiston käyttäjän vastuulle jää huolehtia siitä, että ohjelmistot on päivitetty ja että logiikan ohjelmistossa ei ole sellaista koodia, jota hyökkääjä voisi hyväksikäyttää hyökätessään järjestelmään. Logiikan voi esimerkiksi käskeä hakemaan jostain ulkopuolisesta tietueesta tietoja, mutta jos hyökkääjä pääsee muuttamaan tuota tietuetta, voi logiikka saada vääriä lukemia, josta voi aiheutua laitteiston rikkoutumista tai vaaratilanteita.

Verkkohyökkäystä kokeiltaessa huomattiin, että Cisco:n lähiverkkokytkin osaa jo itse suodattaa ylimääräisen tietoliikenteen, vaikka se onkin perusasetuksissa, eikä logiikkoja näin ollen voitu hidastaa. Tässä tapauksessa DoS-hyökkäys osoittautui mahdottomaksi käytettävissä olevalla laitteistolla, mutta kytkin

ei pystyisi suojaamaan teollisuuden verkkoa kaikissa muissa tapauksissa. Esimerkiksi hyökkääjän suorittaessa DDoS-hyökkäyksen, jossa dataa ohjataan verkkoon yhtäaikaisesti useista kaapatuista tietokoneista ympäri maailman.

VTT:n määritelmän mukaan teollinen internet viittaa ilmiöön, jossa sekä yritysten sisäiset liiketoimintaprosessit että myytävät tuotteet ja palvelut kytketään verkkoon. Tämän edellytyksenä on, että kaikki tuotanto- ja palveluprosessiin liittyvät asiat tai esineet on varustettu digitaalisella tunnisteella, joka välittää dataa toimitus- ja arvoketjujen eri toimijoille. Jatkuva yhteys internetiin mahdollistaa liiketoiminnan ennustettavan toiminnan. (VTT)

Kokeiluympäristön, eli demonstraatioalustan tarjoamia hyötyjä ovat suotuisat ja turvalliset mahdollisuudet esitellä järjestelmän toimintaa vikatilanteissa tai uudella kokoonpanolla sekä etsiä siitä tietoturva-aukkoja. Tällä voidaan myös esitellä järjestelmän toimivuutta potentiaalisille asiakkaille sekä lisätä järjestelmän tunnettavuutta. (VTT)

Jotta voidaan luoda tietoturvallinen järjestelmä, tulee meidän osata myös hyödyntää erilaisia verkkohyökkäystapoja. Järjestelmän tietoturvan turvalliseen testaukseen vaaditaan demonstraatioalusta, joka ei ole yhteydessä ulkomaailmaan, eikä siten voida huomaamatta aiheuttaa muulle verkolle häiriöitä. Ohjelmoitavan logiikan tietoturvaa testattaessa tulisi kaikki muut suojaukset asettaa kokonaan pois, jotta nähdään millaisia vaikutuksia onnistuneella hyökkäyksellä saadaan aikaan. Tämän jälkeen tavoitteena on asentaa sellaiset suojaukset, ettei käytetty verkkohyökkäystyyppi enää vaikuta järjestelmään. Logiikan valmistaja pyrkii huolehtimaan tästä suojauksesta ja käyttäjän tehtävänä on valita käytettävät suojaukset logiikan asetuksista, kuten esim. salasanat. Lisäksi on tärkeää, että järjestelmä on palomuurin takana.

Yleisin tapa hyökätä järjestelmiin on palvelunestohyökkäys, englanniksi Denial of Service (DoS). Siinä pyritään estämään verkkosivuston, palvelimen tai muun järjestelmän toiminta kohdistamalla sinne niin paljon liikennettä, että tämä ei käytännössä kykene enää toimimaan. Vielä tehokkaampi tapa suorittaa palvelunestohyökkäys on suorittaa Distributed Denial of Service, eli hajautettu palvelunestohyökkäys. Tässä hyökkäyksessä kohteeseen lähetetään liikennettä, eli dataa useista kaapatuista tietokoneista ympäri maailman. Yksittäiset tietokoneet lähettävät niin vähän dataa, ettei itse käyttäjä huomaa välttämättä mitään, mutta koska tietokoneita on useita tässä ns. botnetissä, kohteena oleva palvelin tai muu vastaanottaa niin suuren määrän liikennettä, ettei se enää kykene palvelemaan muita asiakkaita. Usein kytkimet

osaavat suodattaa ainakin samasta IP-osoitteesta tulevaa liikennettä, mikäli tunnistavat, että se on turhaa ja toistuvaa.

Tietoturvariskien täydellinen poistaminen on mahdotonta, mutta ennaltaehkäisevillä toimenpiteillä voidaan pyrkiä minimoimaan riskit. Teollisuuden automaatiojärjestelmän tietoturvaan vaikuttavat monet asiat, kuten onko järjestelmä kytkettynä internetiin, käyttöjärjestelmän tietoturva ja virustorjunta sekä palomuuri päivitetty ajan tasalle. Näistäkin huolimatta mahdollinen hyökkäys voi tulla työntekijöiden omasta virheestä, kuten tietokonemadon kulkeutuminen oman kannettavan laitteen kautta teollisuuden tietojärjestelmään.

Tietokoneohjelmien ohjelmoinnissa on mahdollista tehdä pieni, mutta kriittinen virhe. Esimerkiksi Windows-käyttöjärjestelmän ohjelmakoodi toimii tällöin normaalisti, mutta luo tietoturva-aukon, jota taitava hyökkääjä voi hyödyntää. Näiden aukkojen löytäminen on usein mahdotonta, koska tietoturvan testaajan tulisi olla pätevämpi kuin niiden, jotka pyrkivät hyökkäämään järjestelmään. Usein tietoturva-aukot löytyvätkin jälkikäteen ja niitä etsitään ja korjataan jatkuvasti. Käyttöjärjestelmät ovat myös niin monimutkaisia ja sisältävät paljon koodia, joten näiden pienten virheiden löytäminen on koodia lukemallakin haastavaa. Useat tietoturvayhtiöt tekevätkin yhteistyötä keskenään selvittäessään mahdollisia haittaohjelmia ja tietoturva-aukkoja. Yksittäisen yrityksen resurssit eivät riittäisi mitenkään löytämään kaikkia haavoittuvuuksia, koska uutta ohjelmakoodia kirjoitetaan niin paljon ja uusia haittaohjelmia ilmestyy päivittäin lukemattomia määriä.

Usein teollisuuden käyttöjärjestelmät ovat Windows-pohjaisia, joka on käytetyin käyttöjärjestelmä ja siksi myös useimmat hakkerit opettelevat hyökkäämään juuri näihin järjestelmiin. Automaatiojärjestelmää rakennettaessa tulisi siis huolehtia mm. käyttöjärjestelmän päivitettävyydestä. Mikäli yrityksessä vaihdetaan käyttöjärjestelmä uudempaan, ei ohjelmoitavia logiikkoja saada enää välttämättä ohjelmoitua vanhalla ohjelmistolla uudessa käyttöjärjestelmässä. Ohjelmoitavan logiikan ohjelmakoodissa virheen mahdollisuus on myös mahdollista, mutta koska logiikan ohjelma sisältää huomattavasti käyttöjärjestelmää vähemmän koodia, niin sen korjaaminen on helpompaa. Käytännössä kuitenkin logiikkaohjelmoinnissa tehdyt virheet aiheuttavat laittestoon useammin vääriä tai vaarallisia toimintoja eivätkä tietoturvariskejä.

IEC:ssä eli International Electrotechnical Commission:ssa on aloitettu valmistelutyö uusien automaatiota koskevien tietoturvastandardien laatimiseksi. Tällä hetkellä sovelletaan standardia IEC 61508, joka

on ns. kattostandardi, jonka pohjalta on valmisteltu eri aloille omat sovellusstandardit. Automaatiojärjestelmän turvallisuutta arvostellaan tässä standardissa SIL-asteikolla. Yritykset vastaavat omien tehdasverkkojensa tietoturvasta ja kehittävät omat tietoturvapolitiikkansa, joiden mukaan toimitaan. Automaatioalalla ei ole vielä yhtä paljon tietoturva-asiantuntijoita kuin muilla tietoteknisillä aloilla. Suojaus painottuu pääosin verkkojen erottamiseen palomuurien avulla sekä liikenteen tarkka rajaus. Virustorjuntaohjelmistot pidetään ajan tasalla ja tietoliikennettä tarkkaillaan. Tietoturva on yhtä vahva kuin sen heikoin lenkki. (Teollisuusautomaation tietoturva, 2010)

Verkkohyökkäys on rikos, vaikka se jäisi pelkäksi yritykseksi. Britanniassa verkkohyökkäyksestä voi saada jopa 10 vuoden tuomion. Vuonna 2005 eräs David Lennon haastettiin oikeuteen lähettäessään entiselle työnantajalleen 5 miljoonaa sähköpostia, joka johti sähköpostipalvelimen kaatumiseen. Häntä ei kuitenkaan tuomittu, koska oikeus päätyi tulokseen, ettei sähköpostin lähettäminen sähköpostipalvelimelle ole rikollista toimintaa. Tässä tapauksessa sähköpostin määrällä ei ollut väliä. (CNET)

Tietotekniikan lisääntyessä ja monipuolistuessa, tulee automaatiojärjestelmiä sisältävälle teollisuudelle lisääntyviä haasteita turvata oma järjestelmä. 90-luvulla tietotekniikan ammattilaisten määrä oli vain murto-osa nykyisestä, joten tietoturvariskejä oli vähemmän. Paras vaihtoehto tietoturvan parantamiselle on sopia eri ammattilaisille omat osa-alueensa, koska tietoturva-ammattilainen ei välttämättä hallitse automaatiojärjestelmiä, eikä automaation ammattilainen hallitse tietoturvaa. Tietoturvayritysten ja työntekijöiden yhteistyö automaatiolaitteiden valmistajien kanssa edesauttaa ohjelmoitavien logiikkojen tietoturvan kehitystä. Eletään aikaa, jolloin logiikkavalmistajien tulee vastata haasteeseen luoda turvallinen ja luotettava automaatiolaitteisto, koska panoksena on varsinkin alalle keskittyvien logiikkavalmistajien tulevaisuus.

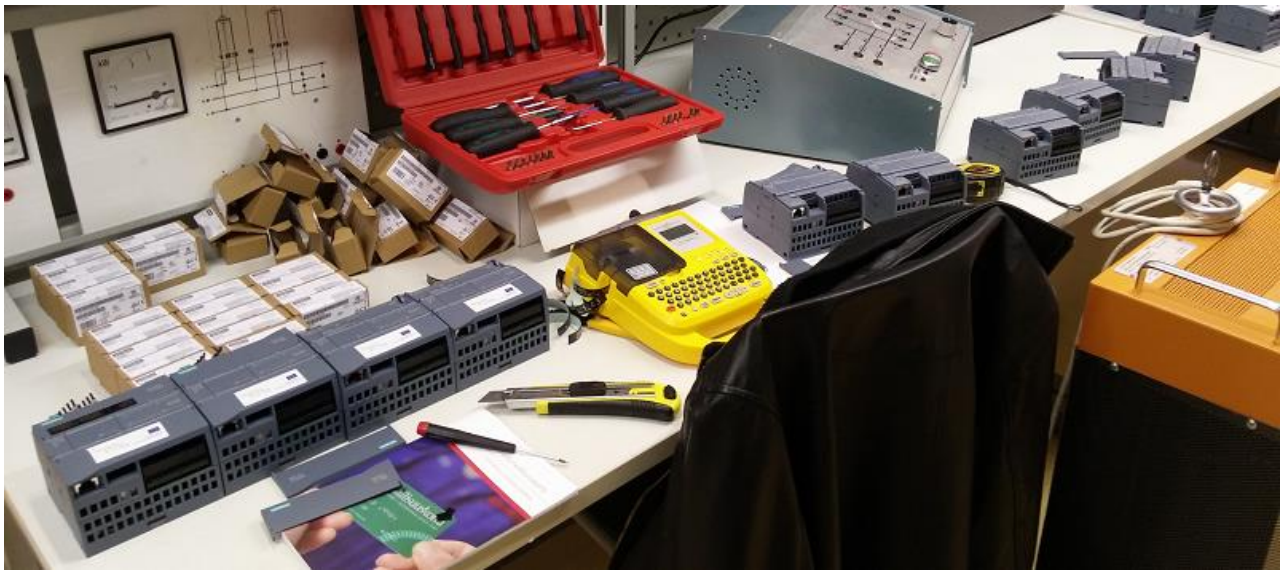
# 4 KYTKENNÄT

Siemensin S7-1200 ohjelmoitavia logiikoita varten rakennettiin tietoliikennelaboratorion pöydässä olevalle tasolle DIN-kiskosto ja neljä 6-osaista pistorasiaa. Logiikoita DIN-kiskolle mahtui 16 kpl ja jokaiselle logiikalle asennettiin oma virtajohto kosketussuojalla. Kosketussuoja on muovista 3D-mallinnettu kappale, joka suojaa jännitteisten johtimien tahattoman koskettamisen. Logiikoille päätettiin asentaa oma virtajohto, jotta niiden siirto opetustilaan on mahdollisimman helppo, eikä operaatiossa tarvita monenlaisia työkaluja. 20:sta pistorasiasta 16 menee logiikoiden ja yksi lähiverkkokytkimen käyttöön.

KUVA 9. Neljä 6-osaista pistorasiaa logiikoille ja kytkimelle

Virtojen päälle kytkemisen jälkeen huomattiin, että logiikoiden suuren 20 A käynnistysvirran vuoksi pöydän pistorasiassa oleva 10 A automaattisulake laukeaa. Ensin päädyttiin liittämään logiikat yksitellen pistorasiaan, jotta sulake kestää. Tämä ei muodostunut ongelmaksi, koska logiikat eivät vie paljoa virtaa käynnistyksen jälkeen. Mahdollisen sähkökatkoksen vuoksi lopulliseen asennukseen vaihdoimme kuitenkin pöydässä olevaan neljän 6-osaisen pistorasiaryhmään pidemmän kaapelin, jotta saamme siihen virran toisesta kauempana sijaitsevasta rasiasta, jolloin käynnistysvirta ei missään tilanteessa muodostu ongelmaksi 16 A tulppasulakkeiden ansiosta.

Ennen logiikkojen asentamista DIN-kiskoon, tulostettiin etuosaan Martten Label Shop BEE3 -tarratulostimella yksilöidyt IP-osoitteet, jotta logiikka on helppo tunnistaa.



KUVA 10. Logiikoiden IP-osoitteiden tulostus tarratulostimella

KUVA 11. Virtakaapeleiden asennusta

Lähiverkkokytkimelle ei asennettu omaa kiinteää telinettä, vaan se on DIN-kiskon taakse jääneellä vapaalla pöytätasolla. Sähköjohdot vedettiin asennuskanavaa pitkin mahdollisimman siististi toiseen päähän pöytää. Näin avoimessa asennuksessa johtojen täydellinen piilottaminen olisi kuitenkin vaatinut suuremmat asennuskanavat. Asennuksen tavoitteena oli kuitenkin olla myös helposti purettavissa, joten nippusiteitä käytettiin mahdollisimman vähän. Seuraavaksi asennettiin Siemensin toimittamat CAT6-parikaapelit logiikoilta kytkimelle portteihin 1-16 ja tietokoneelta porttiin 17. Kytkin ei ole yhteydessä koulun verkkoon, vaan toimii itsenäisesti omana verkkonaan.

KUVA 12. DIN-kiskollisen pöytätason ensiasennus logiikoineen

Seuraavaksi rakennettiin logiikoiden ulostuloja indikoiva LED-rima, jota varten tilattiin 16 kpl 10 mm vihreitä valodiodeja, eli LED:jä, etuvastukset ja kaulukset. Kyseessä olleen vihreän LED:n nimellisjännite on 2,7 V ja virraksi päätettiin valita 30 mA, jotta LED loistaisi kirkkaana.
LED:n etuvastus laskettiin kaavalla:

$$R = \frac{U - Uled}{Iled} = \frac{24\ V - 2{,}7\ V}{0{,}030\ A} = 710\ \Omega.$$

Kotelo rakennettiin itse asennuskanavasta muotoilemalla. Virtajohdoiksi valikoitui aiemmin seinästä irrotetut ylimääräiseksi jääneet kaksi kpl verkkokaapeleita liittimineen, joissa on 8 kpl johtimia per liitin. Koteloon tarvittiin vielä maajohdin, joka vedettiin harmaalla johdolla rimaliittimeen. Rimaliittimestä vedettiin sinisellä johdolla maajohto jokaisen logiikan M-liittimeen. Virtajohdot LED-valoille otettiin logiikoiden ulostulosta Q0.0. Siemens S7-1200 -ohjelmoitavan logiikan ulostulo on potentiaalivapaa, joten siihen täytyi asentaa hyppykaapeli logiikan omasta 24 V ulostulosta 1 L porttiin, joka syöttää jännitteen ulostuloihin Q0.0 – Q0.3. Sisääntulot ovat myös potentiaalivapaita, mutta logiikkoihin asennettiin kuvassa 9 nähtävä simulaatiomoduuli, joka syöttää sisääntuloille jännitteen logiikan 24 V ulostulosta.

KUVA 13. LED-riman askartelua

KUVA 14. Viimeistelyä vaille valmis ja toimiva LED-rima, jossa virtajohdot tulevat CAT5e-liittimien ja maajohto banaaniliittimen kautta

KUVA 15. Ensimmäinen 16 logiikan versio valmiina

Ensimmäisten verkkohyökkäystestien jälkeen logiikoita haluttiin kytkeä myös pienempään hubiin, jotta saadaan eliminoitua kytkimen mahdollinen verkkoliikenteen suodatus helposti pois. Tässä onnistuttiin ja logiikat hidastuivat ja jumittivat. Ciscon kytkin suodattaa verkkohyökkäyksessä käytettävää suurta datamäärää, kun hubi vuorostaan välittää kaiken datan sellaisenaan.

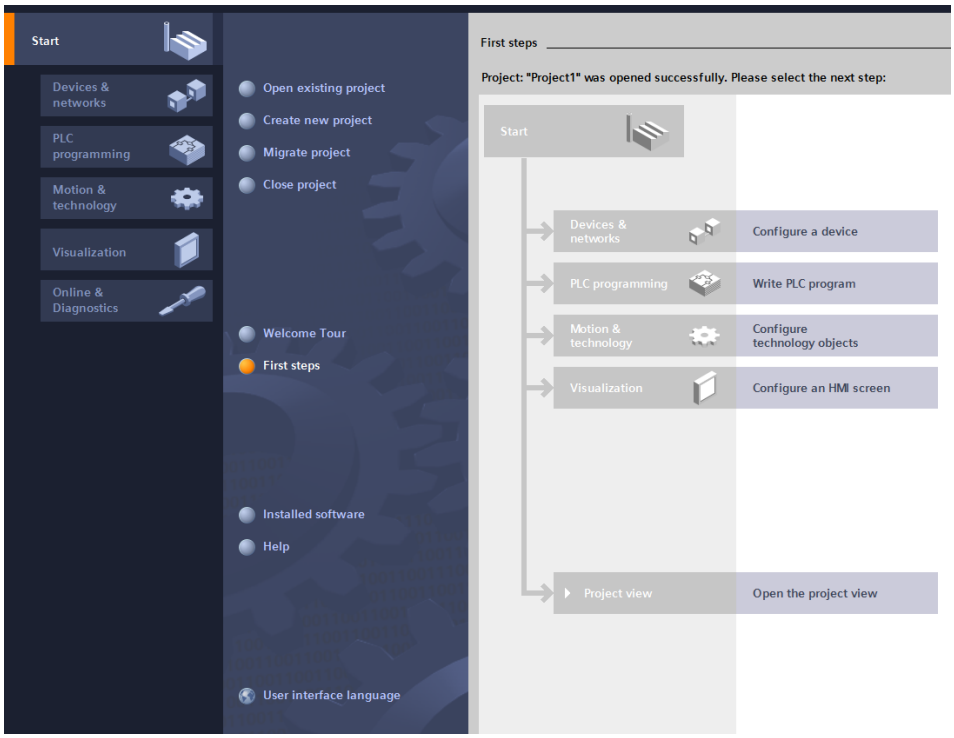
Hubiin kytkettyihin kolmeen logiikkaan hyökättäessä lopputulemana kaikkien logiikoiden ulostulot olivat päällä tai kaikki ulostulot sammuivat. Seuraavaksi haluttiin kytkeä 6 ensimmäistä logiikkaa, eli PLC 1-6 -logiikat yhteyteen hubin kautta ja loput 7-16 kytkimen kautta. Ohjelmaa ei tarvittu muuttaa, mutta function blokissa olevan PUT-komennon asetuksista piti valita logiikka 6:n partneriksi logiikka 1. Lisäksi piti asettaa uusi S7-connection näiden logiikoiden välille. Sama operaatio täytyi tehdä kytkimeen jääneille logiikoille, jossa logiikka 16:sta partneriksi asetettiin logiikka 7 ja edelleen näiden välille uusi S7-connection network connections -asetuksista. Jotta muutokset tulivat voimaan, ohjelmat ja asetukset ladattiin molempiin logiikkoihin, 1 ja 6 sekä 7 ja 16. Tällöin logiikat molemmat logiikkapartnerit tietävät, että S7-connection on luotu ja käytettävissä.

Jossain vaiheessa logiikka 8 oli jostain syystä mennyt epäkuntoon, eikä ottanut yhteyttä lähiverkkoon ja logiikan LINK sekä RX/TX -merkkivalot olivat pimeänä. Ei saatu kuitenkaan selville johtuiko tämä verkkohyökkäyksestä vai jostain muusta tuntemattomasta syystä. Ongelma kuitenkin ratkaistiin käyttämällä logiikan virtoja pois päältä.

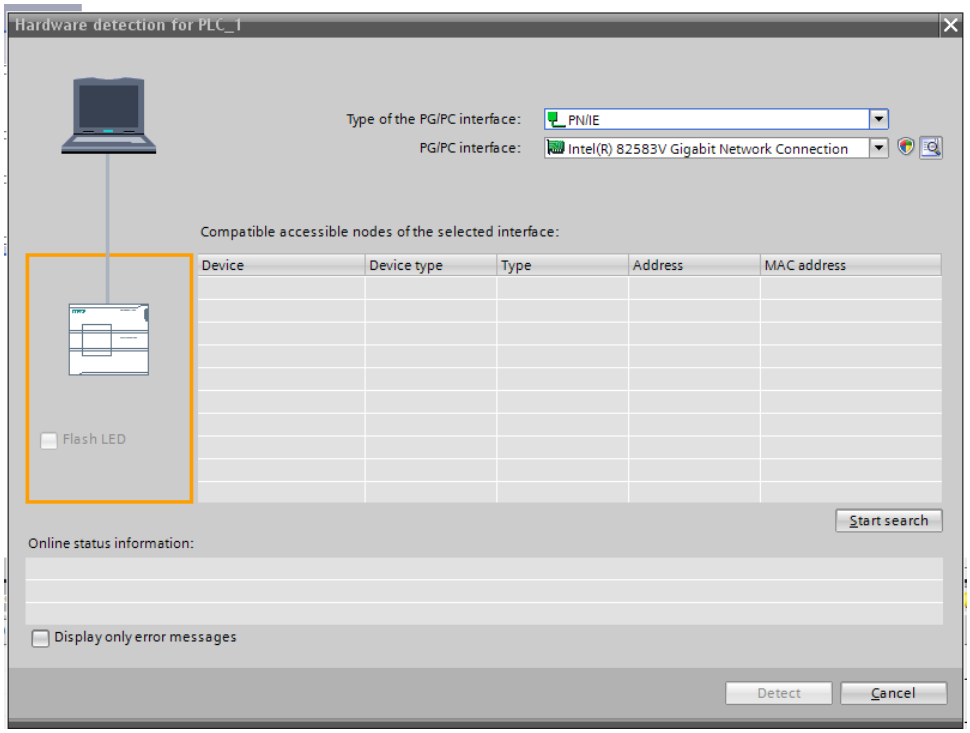
Mikäli halutaan hyökätä kytkimessä oleviin logiikkoihin, pitää olla myös tietämystä, miten päästä käsiksi kytkimen asetuksiin ja ohittaa suodatus. Käyttämämme kytkin oli kuitenkin perusasetuksissa, mutta silti tarpeeksi hyvä suodattamaan ylimääräistä dataa. Toinen mahdollisuus häiritä logiikkoja olisi löytää

bittiviesti, jolla logiikan CPU voidaan pysäyttää. S7-1200 -logiikassa ei ole itsessään nappia, jolla sen voisi käynnistää tai pysäyttää, joten sen käynnistämiseen vaadittaisiin pääsy tietokoneelta, jolle on asennettu TIA Portal MicroWin STEP 7 -ohjelmisto. Mikäli onnistuttaisiin sulkemaan tehtaan kaikki logiikat tällä tavoin, siitä voisi aiheutua pitkä tuotannon pysähdys, koska kaikki logiikat pitäisi käydä erikseen käynnistämässä. Tämän lisäksi aikaa menisi tutkimiseen, mikä pysähdyksen aiheutti.

# 5 LOGIIKAN KÄYTTÖÖNOTTO




KUVA 16. TIA Portal STEP 7 V13 asennettuna ja aloitusnäkymässä




KUVA 17. Etsitään ja lisätään projektiin lähiverkkoon kytketty PLC

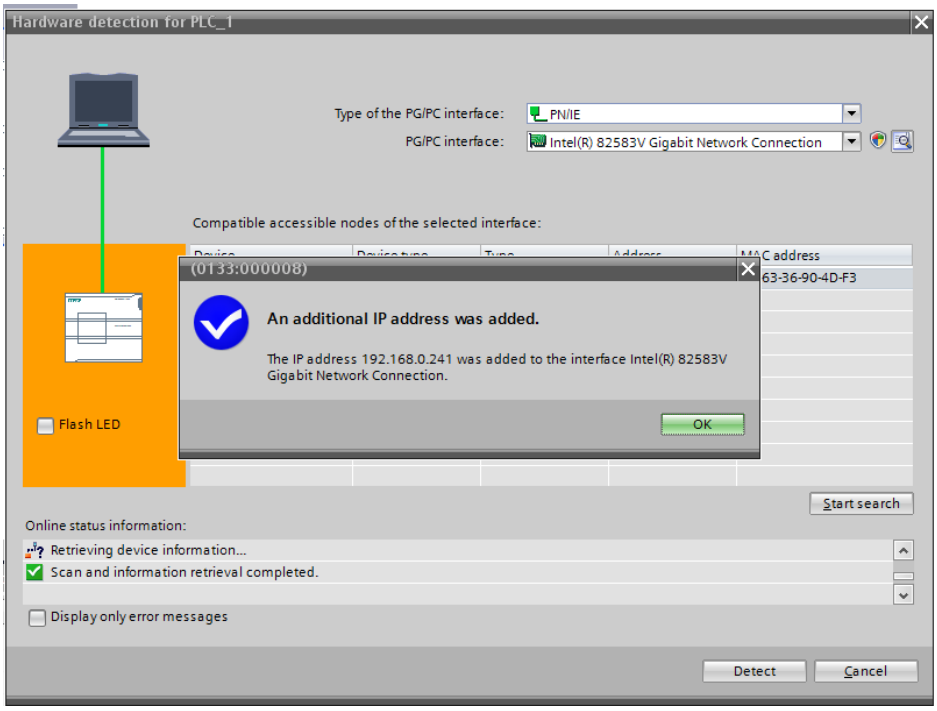
TIA Portal –ohjelmiston asennuttua luodaan uusi projekti ja konfiguroidaan ohjelmoitava logiikka valitsemalla ”Configure a device”.

Ohjelmoitavaa logiikkaa lisättäessä projektiin, valitaan ”start search”, jonka jälkeen ruudulle ilmestyy kaikki lähiverkosta löytyvät logiikat. Valikosta valitaan haluttu lisättävä logiikka esimerkiksi yksilöllisen MAC-osoitteen perusteella, joka on painettuna logiikan etuosaan. Voidaan myös varmistua oikeasta logiikasta ennen valintaa valitsemalla ”Flash LED”, jolloin logiikka alkaa välkyttämään merkkivaloa. Tämä on käytännöllistä, mikäli halutaan varmistua, että asentaja on kytkemässä johtimia oikeaan logiikkaan. Näin asentaja näkee merkkivalojaan välkyttävän logiikan ja voi suorittaa kytkennät, eikä suunnittelijan tarvitse opastaa radiopuhelimella tai muulla tavoin monesko logiikka kiskossa olevista laitteista on kyseessä. Tämän jälkeen tunnistetaan logiikka valitsemalla ”Detect” (KUVA 21).

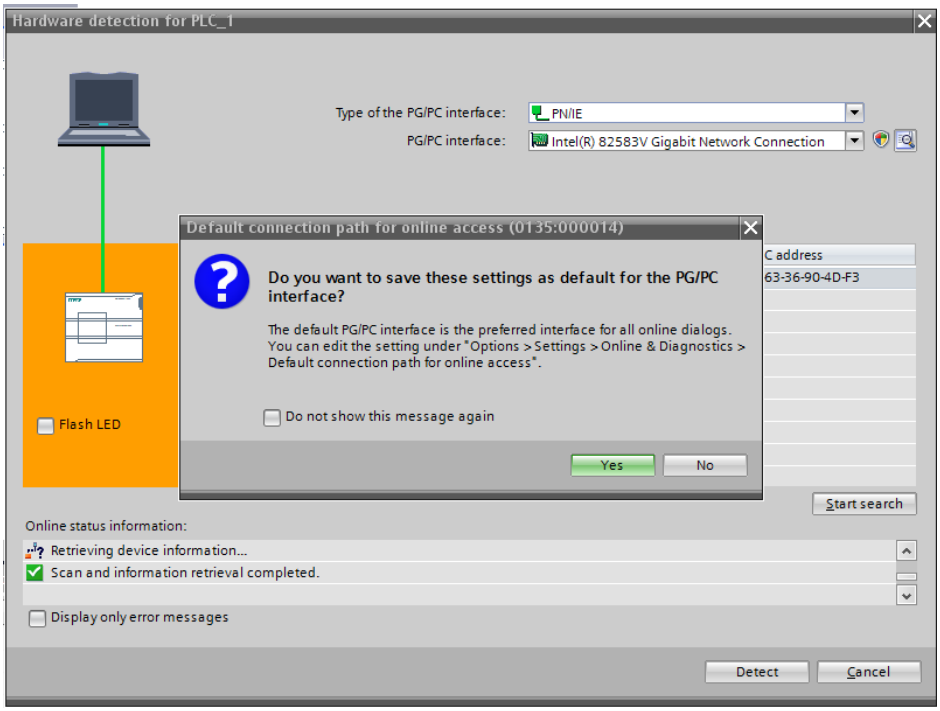
Ensimmäisellä kerralla logiikoille asetettiin IP-osoitteet 192.168.0.1 – 192.168.0.16. Nämä osoitteet haluttiin kuitenkin myöhemmin vaihtaa osoitteisiin 192.168.1.40 – 192.168.1.54. IP-osoitteiden vaihto edellyttää, että logiikat pysäytetään ja vaihdetaan jokaisen logiikan IP-osoite yhtäaikaisesti. Jos vaihdetaan yhden logiikan IP-osoite kerrallaan, tulee ohjelmaa ja IP-osoitemuutoksia ladattaessa compile-error, koska partner-logiikan IP-osoite on väärässä IP-aliverkossa.

Ohjelmiston lataaminen logiikkaan tapahtuu valitsemalla projektinäkymässä vasemmalla puolella olevista logiikoista enintään 10 kappaletta, koska ohjelmisto ei voi olla online-yhteydessä yli kymmeneen logiikkaan kerrallaan. Tämän jälkeen valitaan ”download to device”.

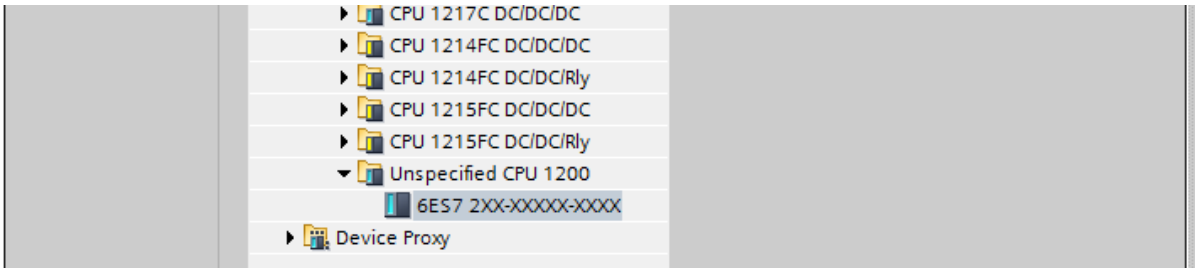
Ohjelma haluaa varmistuksen siitä, minkä logiikan IP vaihdetaan, joten täytyy taas valita ”start search” ja valita haluttu logiikka. Tässä tapauksessa IP-osoitetta vaihdettaessa valitaan uudeksi 192.168.1.40 logiikaksi tämänhetkinen 192.168.0.1 -IP-osoitteellinen logiikka jne., jonka jälkeen valitaan ”load”. Kuvassa 27 nähdään IP-osoitteiden vaihtoprosessia. Ensimmäisessä demoversiossa käytössä oli 3 logiikkaa, jotka kommunikoivat keskenään ja lopulliseen versioon asennettiin 13 logiikkaa lisää.



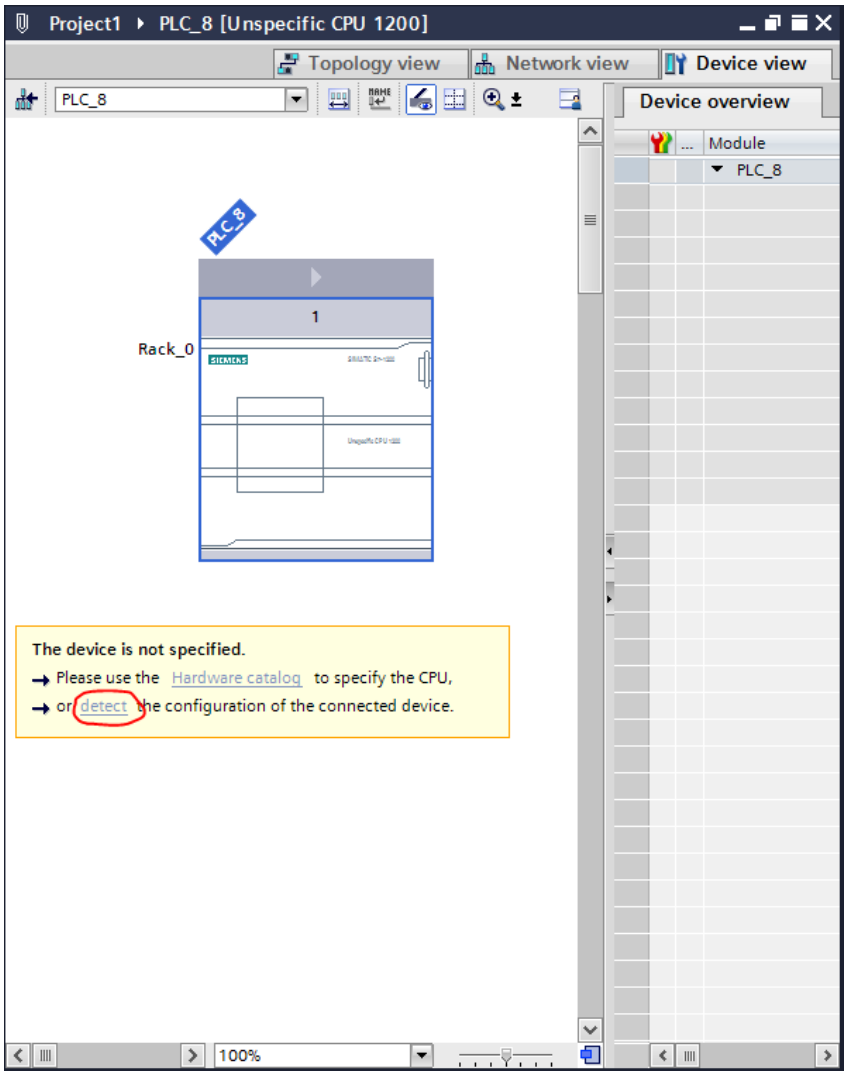

KUVA 18. Ohjelmisto määrittää automaattisesti tietokoneelle IP-osoitteen samasta IP-aliverkosta, kun projektiin lisätään ensimmäinen PLC




KUVA 19. Tallennetaan asetukset

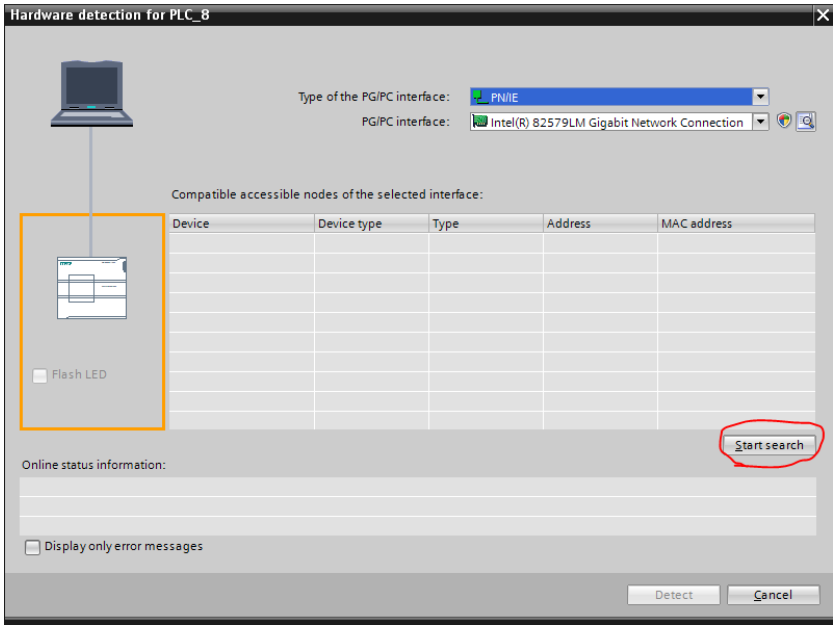



KUVA 20. Lisättäessä uusi logiikka, voidaan valita Unspecified CPU 1200

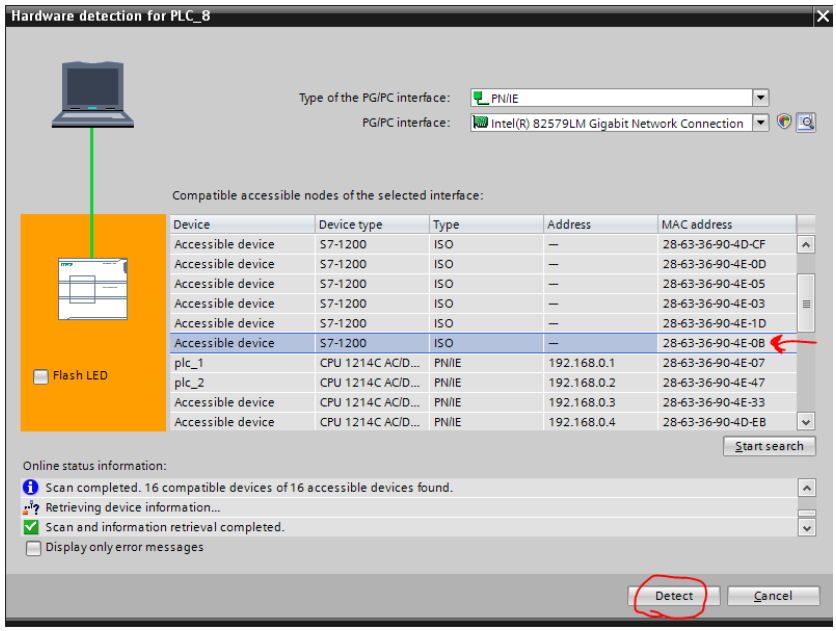



KUVA 21. Viimeistellään tunnistus valitsemalla ”Device view” -ikkunassa ”detect”

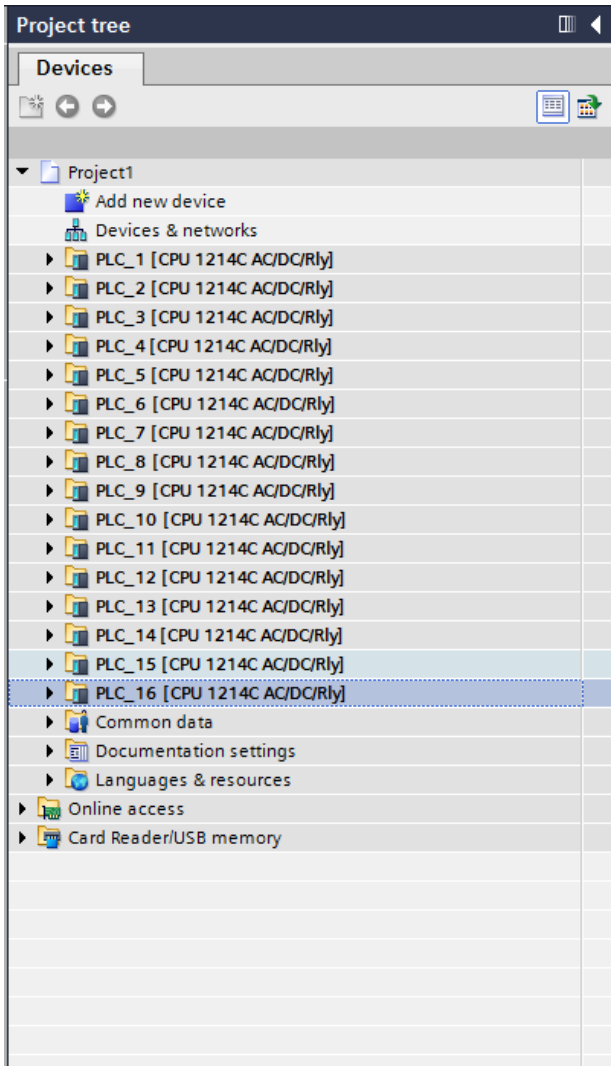
Nyt projektiin on lisätty määrittelemätön S7 1200-sarjan logiikka, jolle täytyy vielä määritellä tarkemmat tiedot. Tiedot saadaan logiikalta itseltään. Katsotaan DIN-kiskossa kahdeksantena olevan logiikan MAC-osoite sen etupaneelista, jonka jälkeen valitaan ohjelmasta ”detect” ja etsitään löydetyistä laitteista haluttu logiikka kuvien 22 ja 23 mukaan.




KUVA 22. Etsitään lähiverkosta löytyvät logiikat valitsemalla ”Start search”

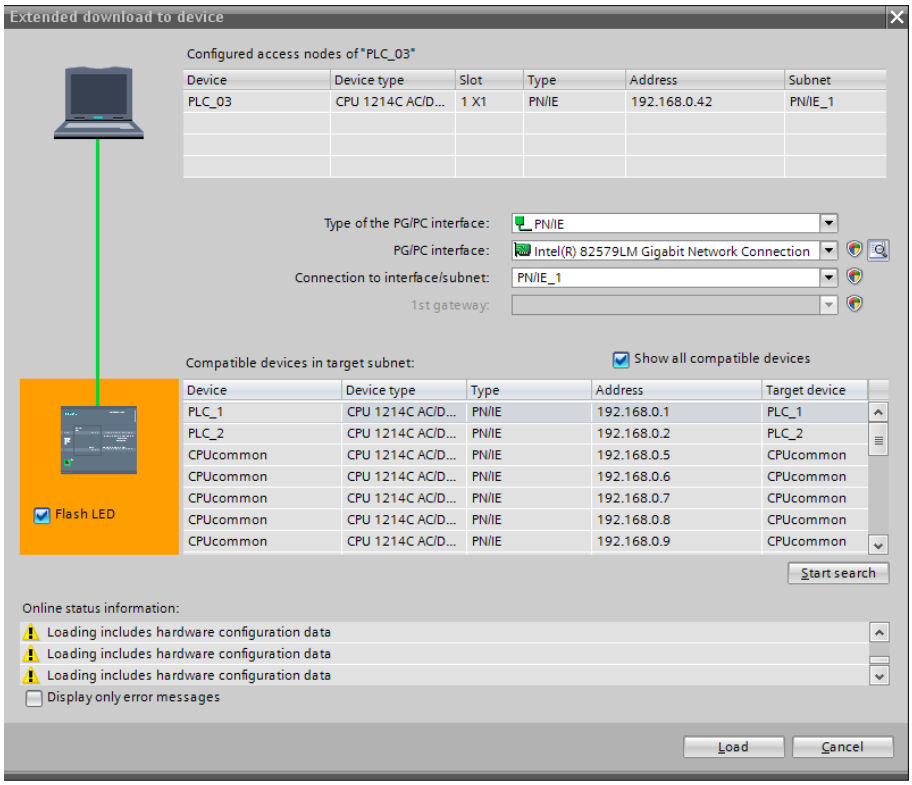



KUVA 23. Katsotaan halutun PLC_8 logiikan MAC-osoite logiikan etupaneelista ja valitaan se listasta, jonka jälkeen valitaan ”Detect”



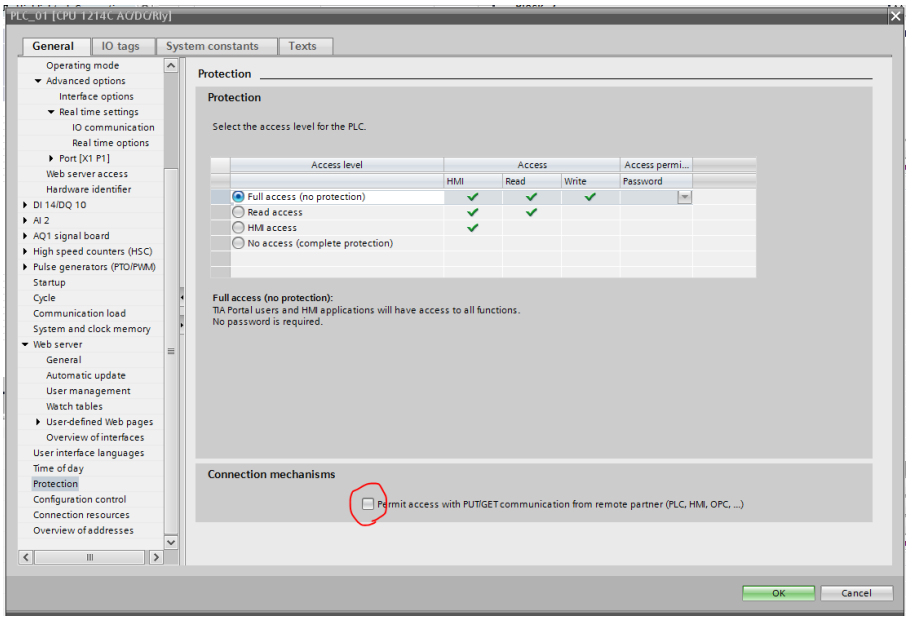

KUVA 24. Kaikki 16 logiikkaa lisättynä projektiin

Projektiin lisättyjen logiikkojen sisältö löytyy tästä valikosta kansioiden sisältä.




KUVA 25. IP-osoitteiden vaihtaminen

Kuvassa 25 vaihdetaan IP-osoitteita. Aiemmassa kokeilualustassa oli vain 3 logiikkaa pöydällä ja lopulliseen versioon DIN-kiskolle asennettiin 16 logiikkaa ja haluttiin myös vaihtaa IP-osoitteet. Tässä vaiheessa halutaan järjestää DIN-kiskoon asennetut logiikat oikeaan järjestykseen. Tarratulostimella tulostetut IP-osoitteet on asetettu irrotettavaan etupaneeliin, joten niiden paikat voi vaihtaa ensimmäiseksi. Seuraavaksi lähdetään nimeämään PLC:t uudelleen PLC_01, PLC_02 jne. käyttäen ”Flash LED”-toimintoa, jolla varmistutaan, että oikea logiikka on valittuna. Lopputuloksena saadaan DIN-kiskossa oleville ohjelmoitaville logiikoille oikeat IP-osoitteet ja nimet oikeassa järjestyksessä.




KUVA 26. Suojausasetuksista täytyy valita ”Permit access with PUT/GET…”, jotta PUT-komentoa voidaan käyttää datan lähetyksessä

Logiikan voi käskeä lähettämään dataa monella tavalla, joista valittiin PUT-komento S7-1200 logiikan tuotepäällikön kanssa käydyn keskustelun jälkeen. Muita keinoja ovat mm. TCON, TSEND, TRCV, TDISCON tai näiden yhdistelmät TSEND_C ja TRCV_C, jotka suorittavat yhteyden muodostamisen ja katkaisemisen sekä datan lähettämisen.

PUT-komento on tässä tapauksessa hyvä ja helppo tapa lähettää nopeasti tietoa eteenpäin. Alustavien testien perusteella voidaan todeta, että useimmiten viestin viive oli logiikalta toiselle noin 20-70 ms.

# 6 OHJELMOINTI

Yleisesti ottaen SIMATIC-ohjainten ohjelmointi on pysynyt samanlaisena S7-300/400 sarjasta S7-1500 sarjaan asti. On tuttuja ohjelmointikieliä, kuten LAD, FBD, STL, SCL tai kuvia/blokkeja, kuten organisointiblokki OB, funktioblokki FB, funktioita FC tai datablokkeja DB. Tästä seuraa se, että valmiit ohjelmat, jotka on luotu aiemmin S7-300/400-logiikoille, voidaan laittaa toimimaan myös S7-1200 ja -1500 -sarjan logiikoissa ilman ongelmia. (Programming Guideline for S7-1200/S7-1500, 7)

- Käynnistetään TIA Portal, Totally Integrated Automation Program -ohjelma.
- Avataan projekti
- Open the project view
- PLC → program blocks → add new block → function → data block
- Devices and networks

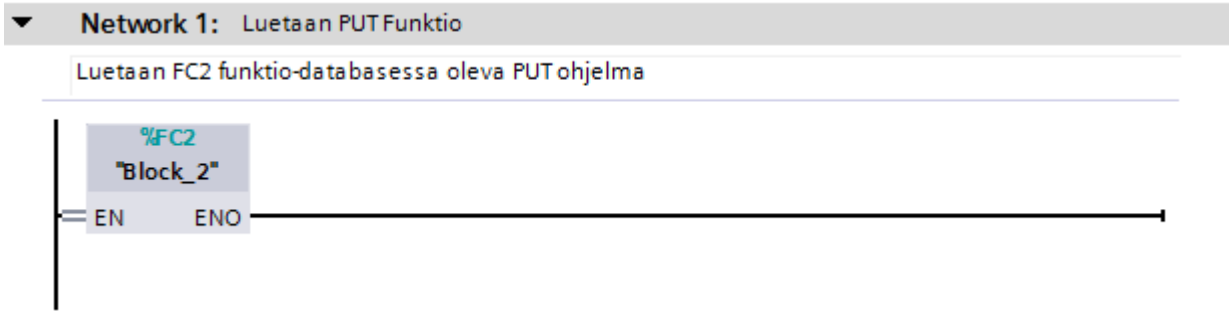
Ohjelmoitavan logiikan pääohjelma on Main[OB1] -blokissa. Logiikka pyörittää pääohjelmaa lukemattomia kertoja sekunnissa. Lisäksi voidaan tehdä lisäohjelmia FC-tiedostoihin, joita voidaan kutsua pääohjelmassa. Datablokit sisältävät erilaisia tietoja, joita ohjelman eri objektit tarvitsevat. Esimerkiksi ohjelmassa oleva laskuri tarvitsee omat tietonsa omassa datablokissaan ja PUT-komento, joka lähettää dataa toiselle logiikalle tarvitsee oman datablokkinsa, jossa on eri muuttujia. Nämä muuttujat voivat olla mm. ajastimella aloitusaika, kulunut aika, tavoiteaika ja kaksi boolean-arvoa, joilla osoitetaan onko ajastin käynnissä ja onko se saavuttanut tavoiteaikansa. Käytännössä voidaan kuitenkin käyttää vähempää määrää datablokkeja, antamalla usean toiminnon käyttää samaa yhteistä datablokkia, mutta koska ainakin Step 7 -ohjelmistossa eri operaatiot tekevät automaattisesti omat datablokkinsa, ei tätä ole järkevää tehdä.

Datablokkien tietueita voidaan käyttää ohjelmakoodissa. Esimerkiksi haluttaessa jonkin toiminnon käynnistyvän, kun PUT-komento on suorittanut tiedonsiirtonsa loppuun, on tarvittava datablokin tietue esim. muotoa Data_Block_2[DB5].Done. Data_Block_2 viittaa käytettävän datablokin nimeen, joka voidaan muuttaa halutuksi. [DB5] viittaa datablokin järjestysnumeroon. Tässä tapauksessa siis kyseessä olisi koko projektin viides datablokki. Kun datablokin 2 sisällä oleva boolean-arvoinen rivi nimeltään Start saa arvon tosi, käynnistetään tiedon lähetys. Kun PUT-komento on suorittanut tiedon lähettämisen ja saanut kuittauksen vastaanottavalta logiikalta, se muuttaa datablokin ”Done”-arvon todeksi yhden kierron ajaksi. Nyt tällä ”Done”-rivin arvolla voidaan käynnistää jokin toinen toiminto. Lisäksi datablokin sisällä oleville tiedoille voidaan määritellä muoto, jota ne ovat ohjelman käynnistyessä, esim.

boolean, jos halutaan arvon olevan joko 1 tai 0, eli tosi tai epätosi. Usein TIA Portal tekee datablokit automaattisesti kun ohjelmaan lisätään eri komentoja ja toimintoja, jotka sellaisen vaativat. Ajastin tekee data blokin IEC_Timer_0_DB_0[DB1] ja PUT-komento Put_DB[DB2].

Jos halutaan, että logiikka tekee jonkin asian ensimmäisellä kerralla, voidaan se toteuttaa mm. lisäämällä datablokkiin rivi nimeltään ”FirstStart” ja antaa sille alkuarvo tosi. Tämän jälkeen ohjelmakoodissa suoritetaan haluttu toiminto, mikäli Data_Block_1[DB1].FirstStart = tosi, jonka jälkeen se asetetaan arvoon epätosi esimerkiksi RS-kiikulla.

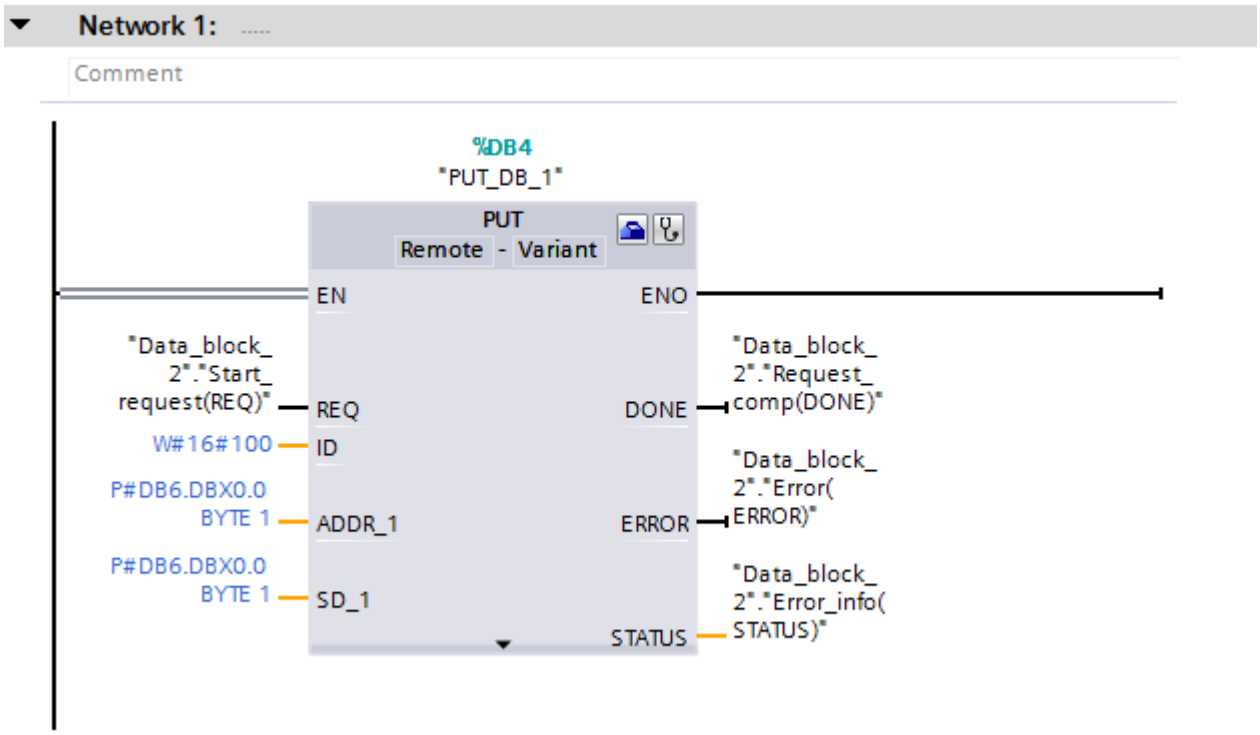
Demonstraatioalustan ohjelmaa kirjoitettaessa tavoitteena oli tehdä ohjelma, joka olisi mahdollisimman yksinkertainen ja toimintavarma, jotta nähdään helposti verkkohyökkäyksen vaikutukset. Tietoturva-asetuksista ei asetettu mitään käyttäjätunnuksia tai salasanoja, eli käytetty Siemens S7-1200 -logiikka oli täysin vakioasetuksissa.



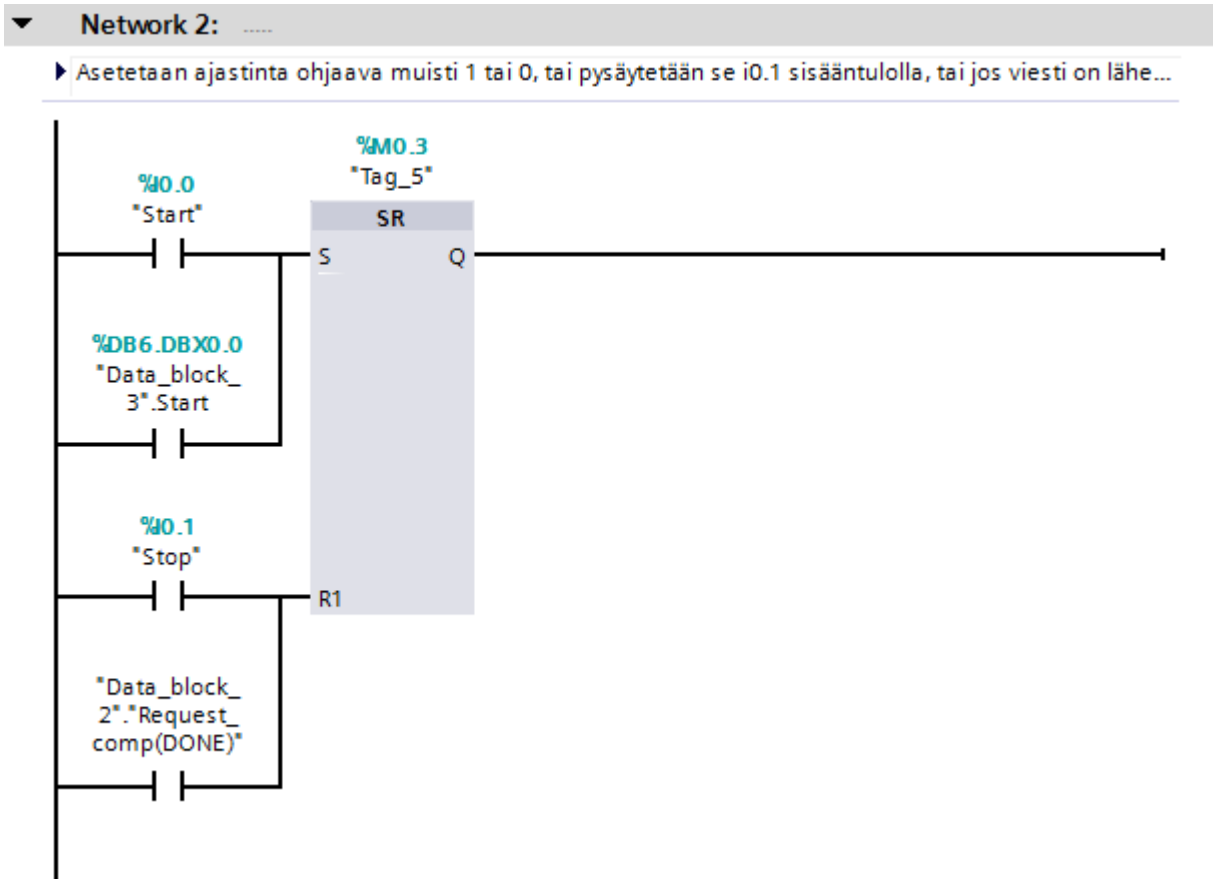

Kuva 27. Pääohjelman Network 1/9

Ohjelma on jaettu yhdeksään networkkiin eli piiriin. Piirien sisällöt on esitetty kuvissa 27 – 36. Ensimmäisessä piirissä käydään lukemassa funktio FC2, joka sisältää PUT-komennon, jolla lähetetään dataa partneriksi valitulle logiikalle.

Kuvassa 28 näkyvä PUT-komento aloittaa tiedonsiirron, kun Start_request(REQ) on tosi. ADDR_1 kohdassa on osoitettu minkä tietokannan tietoa lähetetään ja kuinka paljon. Tämä vaatii sen, että vastaanottavassa logiikassa on myös samalla nimellä löytyvä tietokanta. Tiedonsiirron suoritettuaan Request_comp(DONE) muuttuu todeksi yhden ohjelmakierron ajaksi.

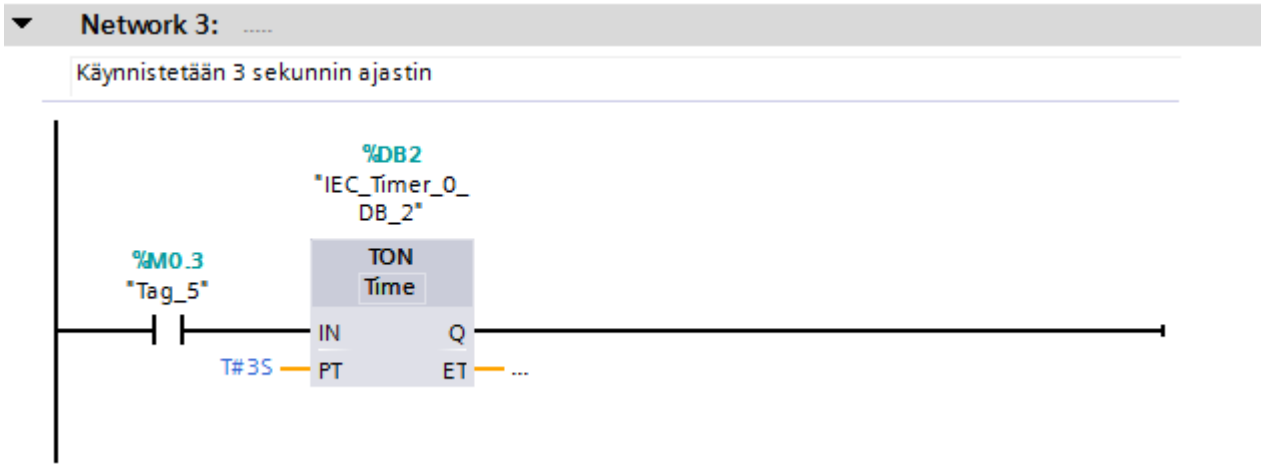



Kuva 28. Pääohjelman Network 1:ssä kutsutun funktion sisältö



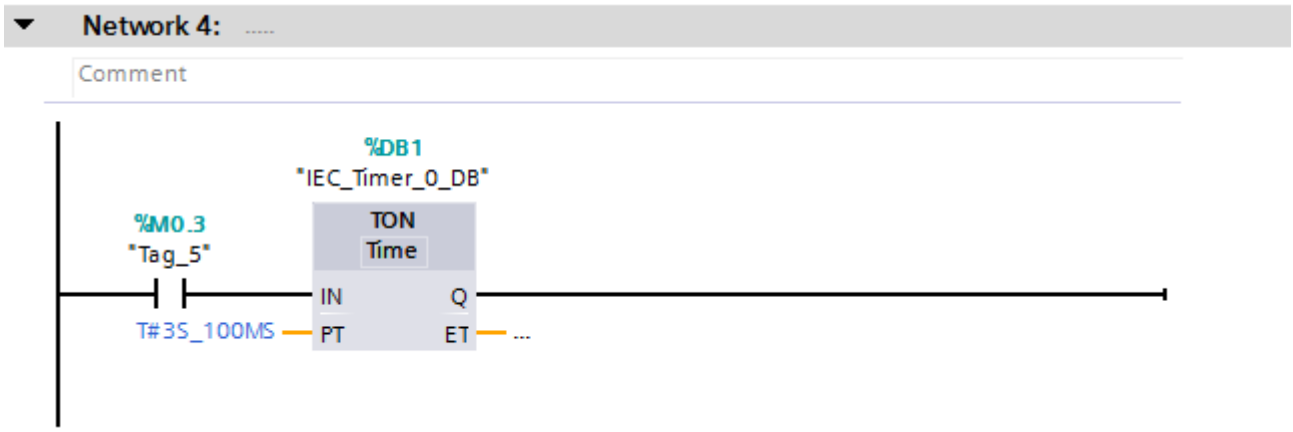

Kuva 29. Pääohjelman Network 2/9

Toisessa piirissä asetetaan muistipaikan M0.3 tila todeksi, jos sisääntulo I0.0 tai DB6-datablokin Start-arvo on tosi. Muistipaikka M0.3 asetetaan takaisin epätodeksi, mikäli sisääntulo I0.1 on tosi tai PUT-komento on suorittanut viestinlähettämisen loppuun ja Request_comp(DONE) on tosi.



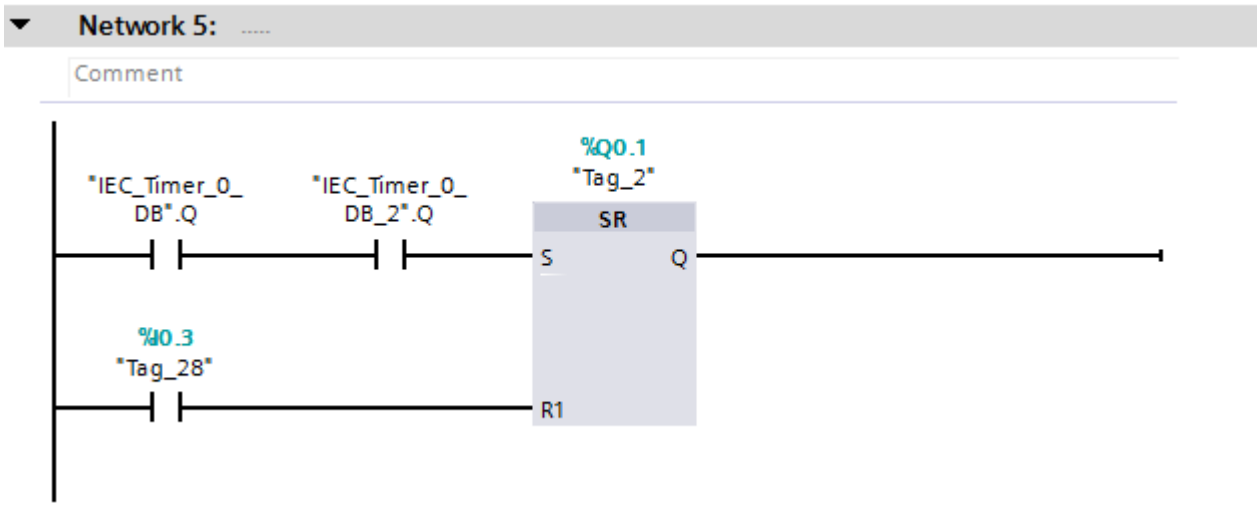

Kuva 30. Pääohjelman Network 3/9

Kolmannessa piirissä muistipaikan M0.3 ollessa tosi, käynnistyy ajastin IEC_Timer_0, jolle on asetettu kolmen sekunnin aika, kunnes sen ulostulo vaihtuu todeksi.



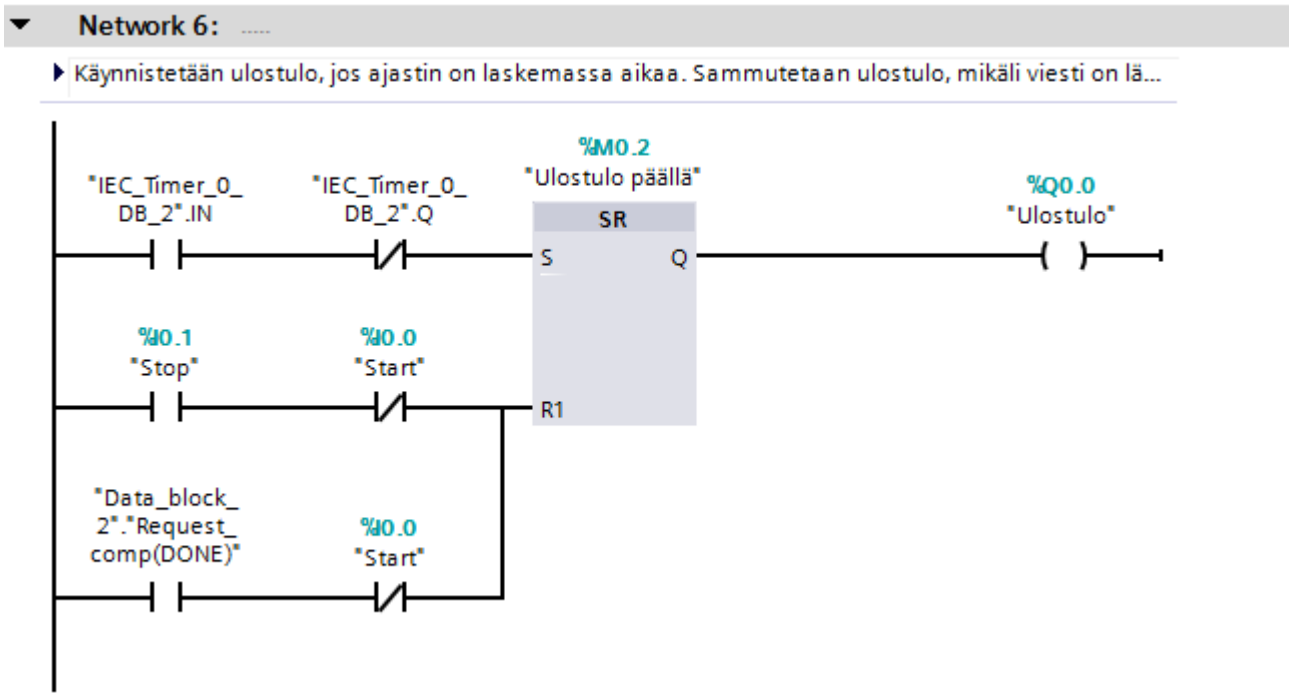

Kuva 31. Pääohjelman Network 4/9

Neljänteen piiriin on lisätty toinen ajastin, jota käytetään myöhemmässä vaiheessa indikoimaan viestin-siirrossa tapahtuva viive.




Kuva 32. Pääohjelman Network 5/9

Piirissä viisi asetetaan logiikan ulostulo Q0.1 todeksi, mikäli aiemmissa piireissä olleet ajastimet ovat molemmat laskeneet loppuun asti. Ulostulo Q0.1 indikoi syttyessään siis sitä, että viestin kulkemiseen kului aikaa yli 100 ms. Ulostulo voidaan sammuttaa sisääntulolla I0.3.

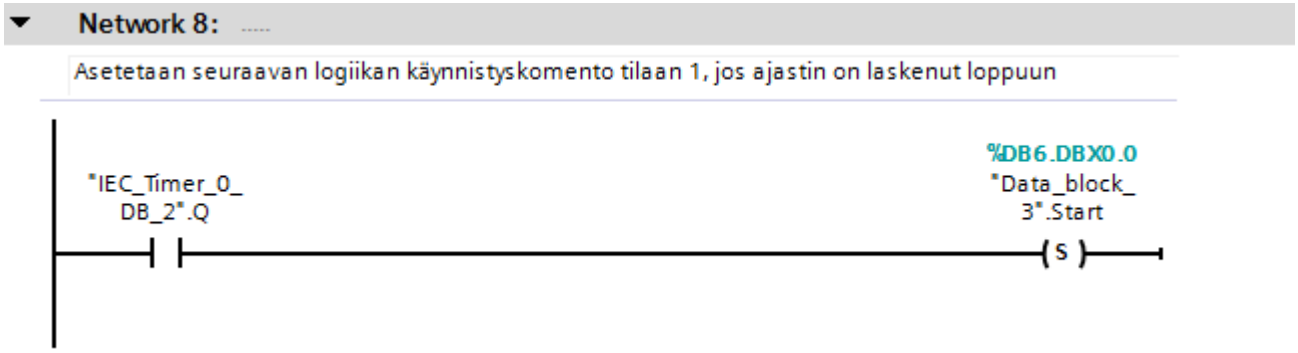



Kuva 33. Pääohjelman Network 6/9

Kuudennessa piirissä asetetaan logiikan ulostulo Q0.0 todeksi, mikäli ajastin IEC_Timer_0 on käynnissä, mutta ei ole vielä laskenut loppuun. Ulostulo voidaan sammuttaa sisääntulolla I0.1. Ulostulo sammuu myös, kun PUT-komento on suorittanut viestinsiirron loppuun.

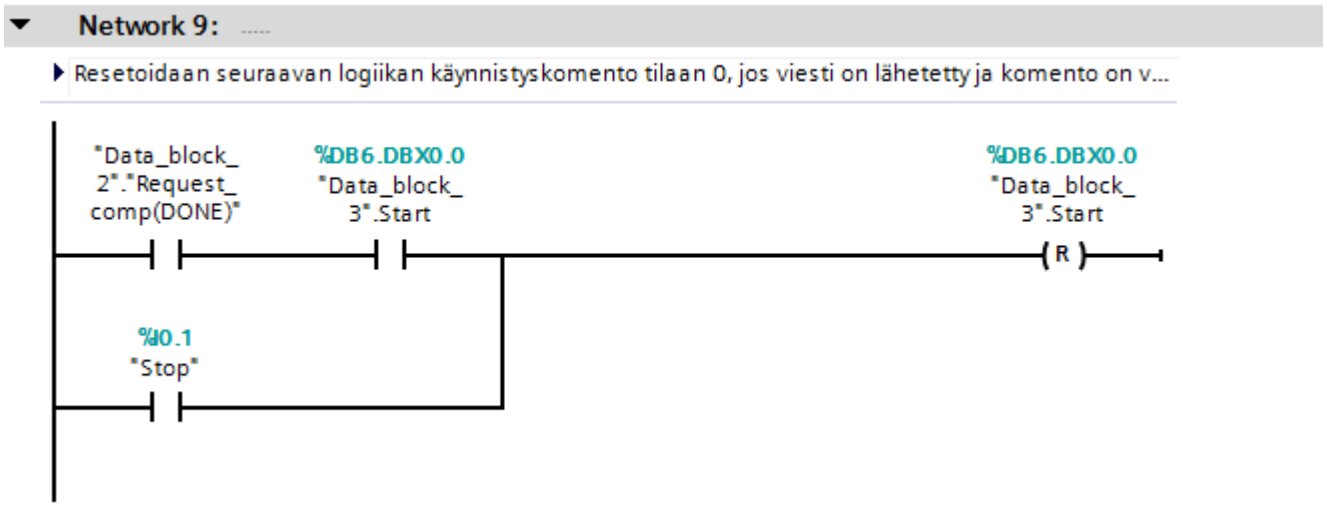

Kuva 34. Pääohjelman Network 7/9

Seitsemännessä piirissä asetetaan Start_request(REQ) arvo todeksi, kun ajastin IEC_Timer_0 on käynnissä ja on myös laskenut loppuun asti, eikä PUT-komento ole saanut viestiä lähetettyä. Start_request(REQ) käynnistää PUT-komennon tiedonsiirron.




Kuva 35. Pääohjelman Network 8/9

Kahdeksannessa piirissä asetetaan logiikan datablokissa DB6 oleva Start-arvo todeksi, kun IEC_Timer_0 on laskenut loppuun asti.




Kuva 36. Pääohjelman Network 9/9

Viimeisessä piirissä asetetaan datablokin DB6 Start-arvo takaisin epätodeksi, mikäli PUT-komento on saanut tiedonsiirron suoritettua.

Jokaisessa logiikassa on sama ohjelma ja samat datablokit, joten seuraavan logiikan ulostulon kytkeytymiseen tarvitaan jokin muuttuja, tämä muuttuja on datablokkiin DB6 asetettu Start-arvo. Ensimmäisessä logiikassa Start-arvo käy arvossa tosi, jolloin koko datablokin DB6 sisältö eli yksi rivi lähetetään toiseen logiikkaan. Lähetyksen jälkeen logiikan omassa datablokissa oleva Start asetetaan takaisin epätodeksi.

**LÄHTEET**

Fonselius, J., Pekkola, K., Selosmaa, S., Ström, M. & Välimaa, T. 1999. Automaatiolaitteet. Koneautomaatio. 199 s. Viitattu: 11.4.2016.

Siemens, 2014. Programming Guideline for S7-1200/S7-1500.
Saatavissa: http://www.xilinx.com/company/about/programmable.html. Viitattu 2.3.2016.

BlackHat 2011 - Siemens Simatic S7 PLC Exploitation, S7-FU with Metasploit.
Saatavissa: https://www.youtube.com/watch?v=33kouEKm0zo.

Exploiting Siemens Simatic S7 PLCs. 2011.
Saatavissa: https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf.

How-to Penetration Testing and Exploiting with Metasploit + Armitage + msfconsole.
Saatavissa: https://www.youtube.com/watch?v=lZlqr2PFJIo.

Siemens. SIMATIC Controllers. The innovative solution for all automation tasks.
Saatavissa: http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/automaatiotekniikka/ohjelmoitavat_logiikat/brochure_simatic-controller_en.pdf. Viitattu 10.3.2016.

Advanced micro controls INC. What is a programmable logic controller (PLC)?.
Saatavissa: http://www.amci.com/tutorials/tutorials-what-is-programmable-logic-controller.asp.

Siemens.
Saatavissa: http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/teollinen_tiedonsiirto_esim_profinet/teollisuus_ethernet.htm.

Spectrum, 2013.
Saatavissa: http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet. Viitattu 10.3/2016.

VTT.
Saatavissa: http://www.vtt.fi/medialle/uutiset/suomesta-teollisen-internetin-piilaakso-n%C3%A4emmek%C3%B6-teollisen-internetin-tuovan-vai-tuhoavan-ty%C3%B6paikkoja. Viitattu 3.3.2016.

VTT.
Saatavissa: http://www.vtt.fi/img/Media/Uutiset/2015/Suomi_Teollisen_Internetin_Piilaakso.pdf. Viitattu 3.3.2016.

Cnet.com, 2006.
Saatavissa: http://www.cnet.com/news/u-k-outlaws-denial-of-service-attacks/, Viitattu 4.3.2016.

Teollisuusautomaation tietoturva, verkottumisen riskit ja niiden hallinta, Suomen Automaatioseura ry, Turvallisuusjaosto 2005, Verkkopainos 2010.
Saatavissa: http://docplayer.fi/1237507-Teollisuusautomaation-tietoturva-verkottumisen-riskit-ja-niiden-hallinta.html. Viitattu 1.4.2016.

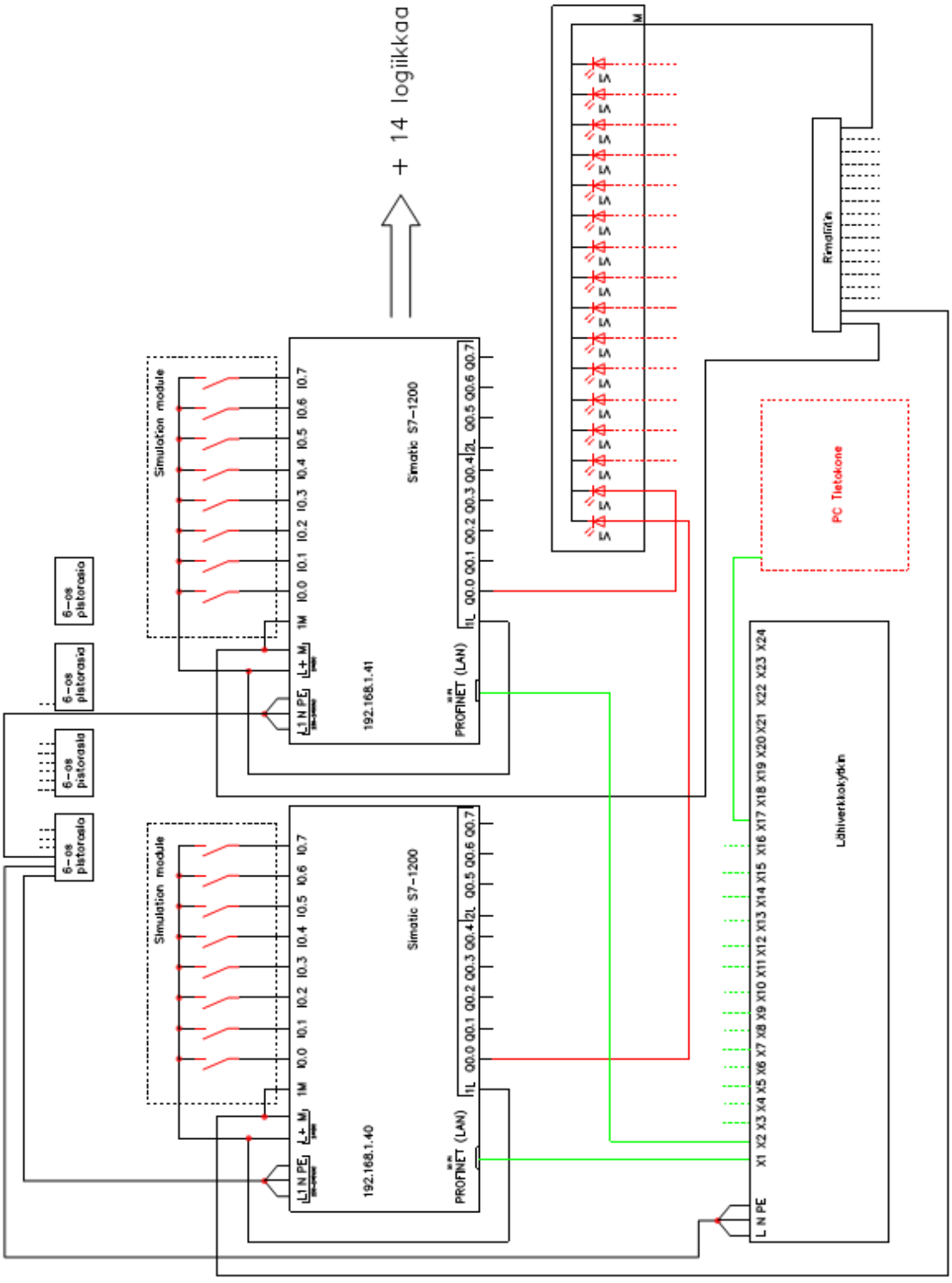
Belden, 2016.
Saatavissa: http://www.belden.com/blog/industrialsecurity/SCADA-Security-Basics-Why-are-PLCs-so-Insecure.cfm. Viitattu 4.4.2016.

Tiilikainen S., Manner J., 2013. Suomen automaatioverkkojen haavoittuvuus - Raportti Internetissä julkisesti esillä olevista automaatiolaitteista.

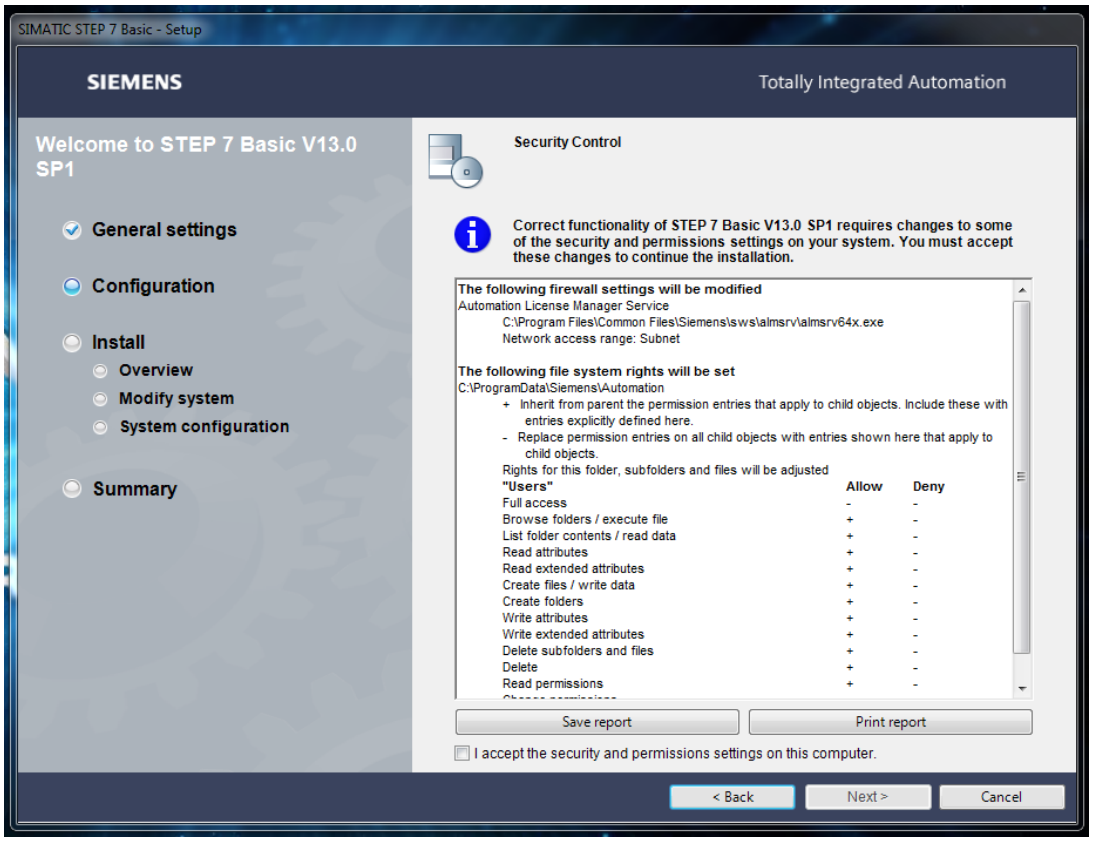
Saatavissa: https://research.comnet.aalto.fi/public/Aalto-Shodan-Raportti-julkinen.pdf. Viitattu: 12.4.2016.

PLC Mentor.com. History of the programmable logic controller (PLC). Saatavissa: http://www.plcmentor.com/Articles/Newsletters/Programmable-Logic-Controller-PLC-History.aspx. Viitattu: 25.4.2016.

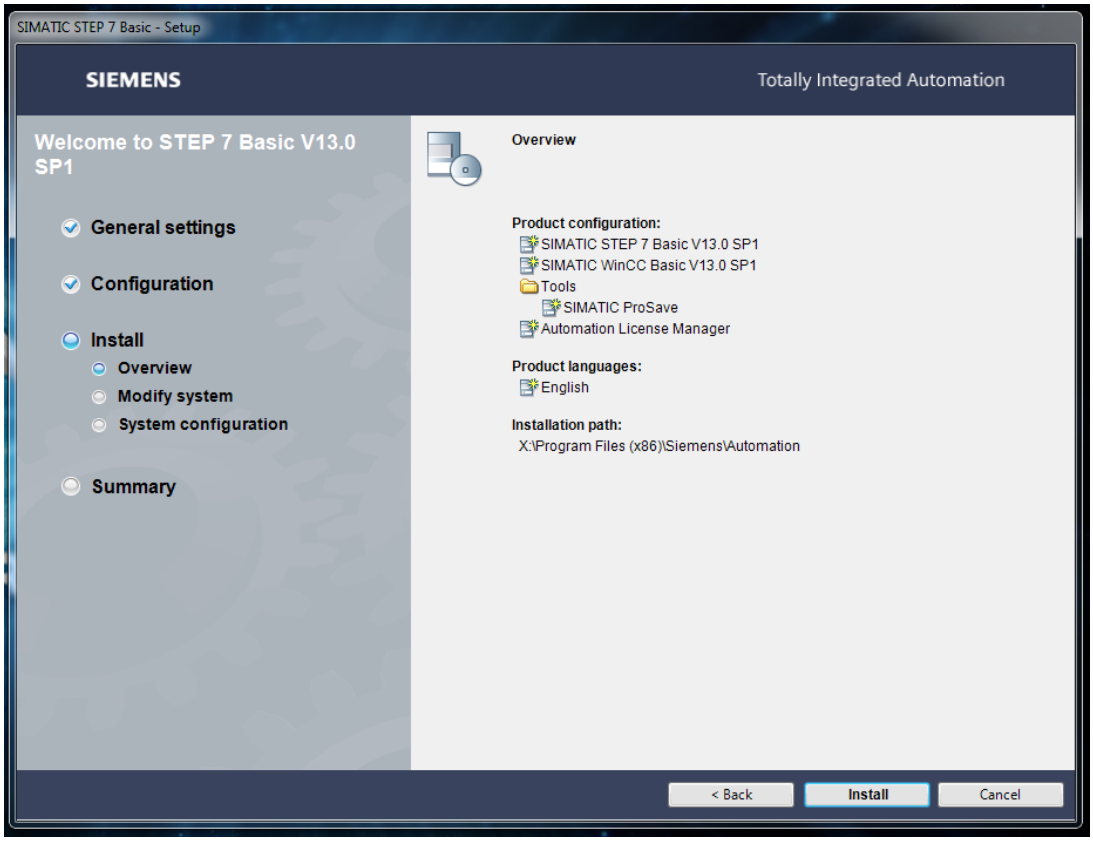
**LIITTEET**



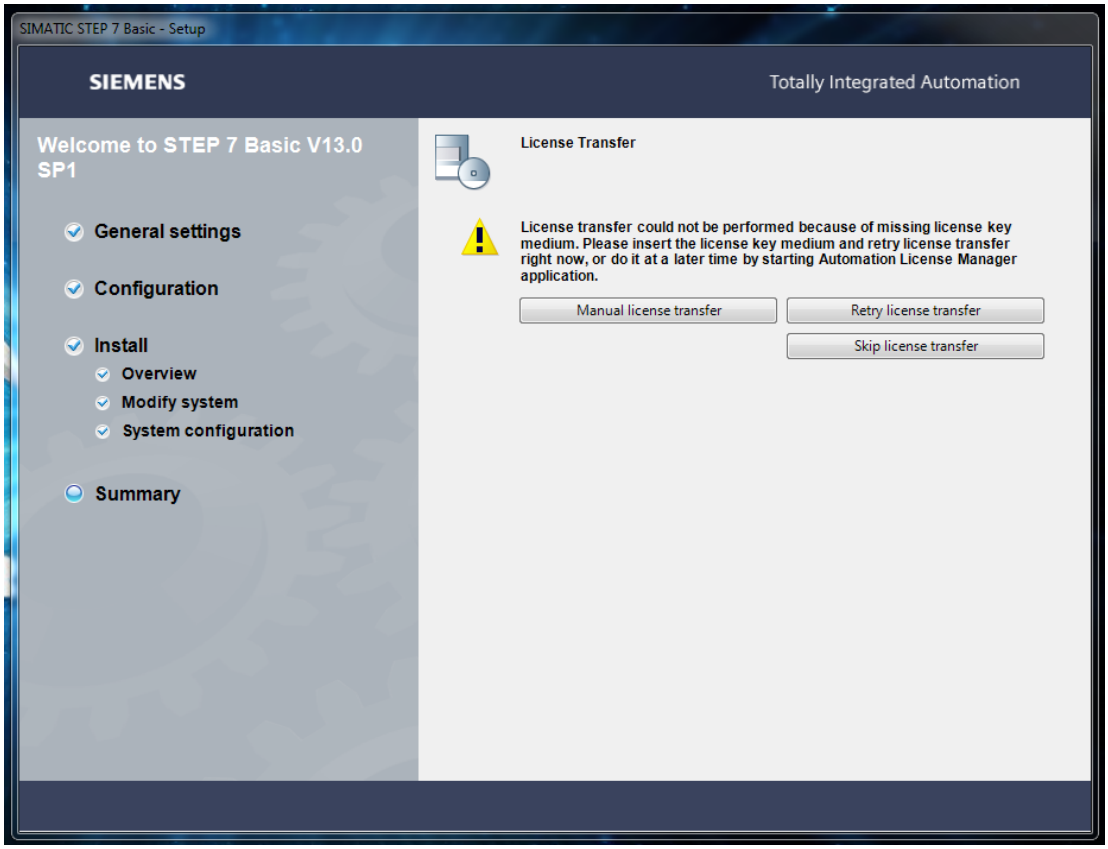

Demonstraatioalustan kytkentäkaavio



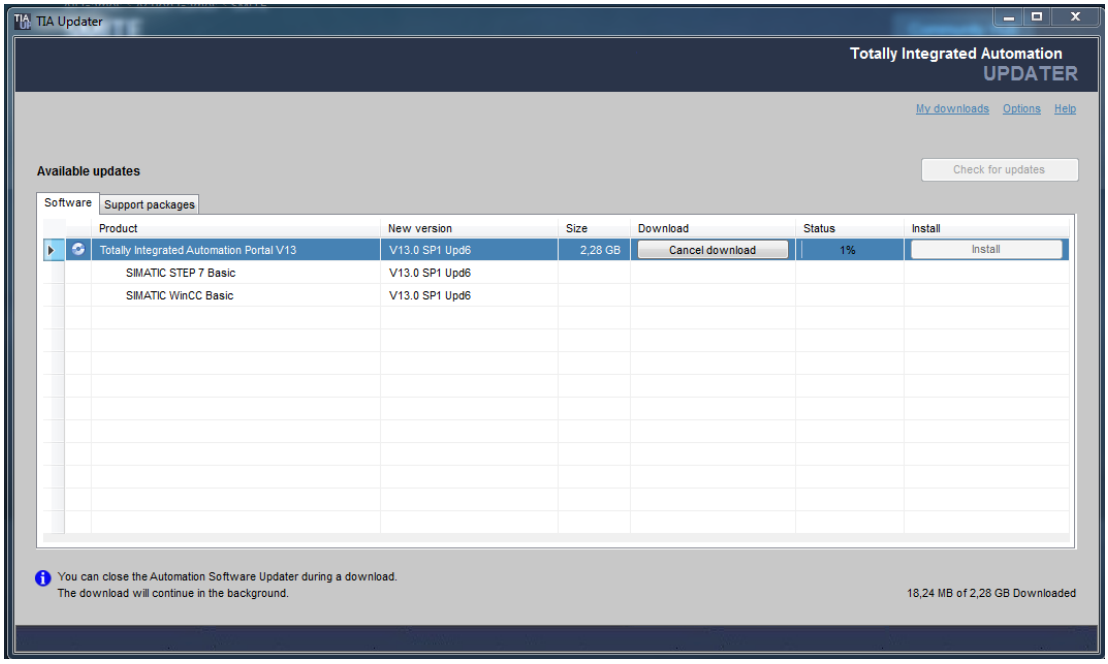

Ohjelmiston asennus, konfigurointi



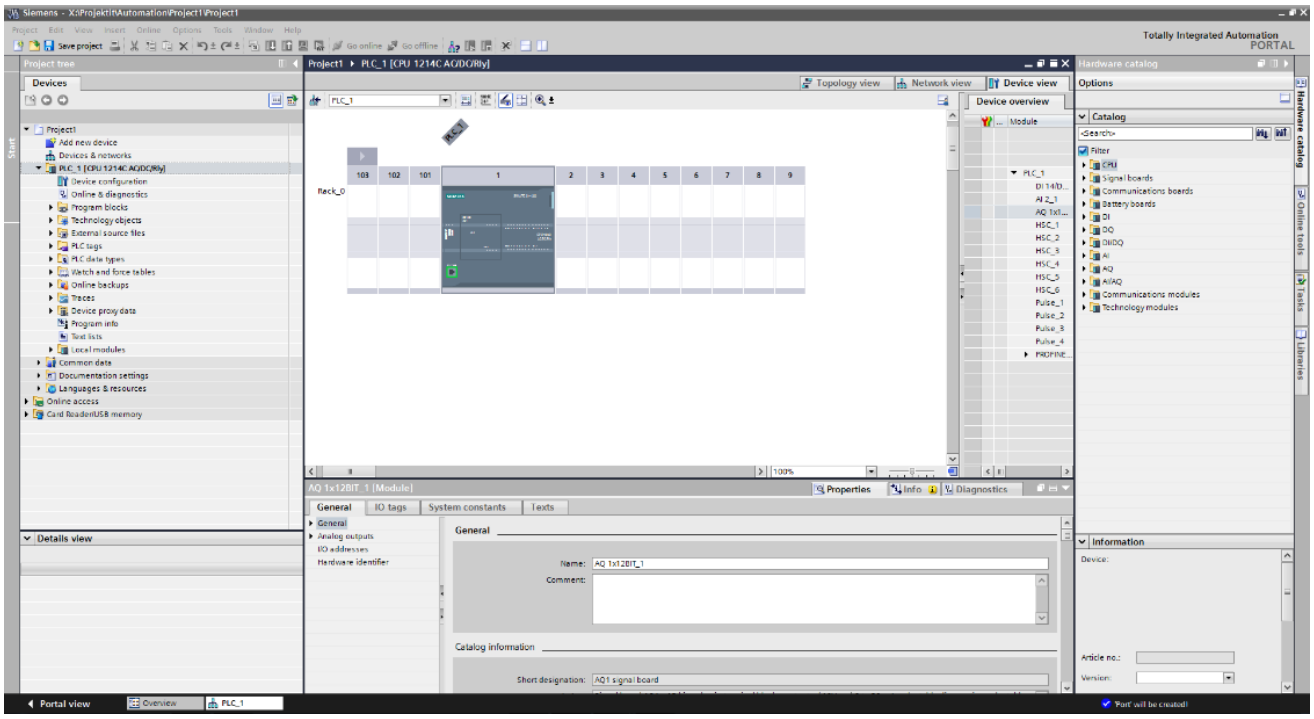

Ohjelmiston asennus



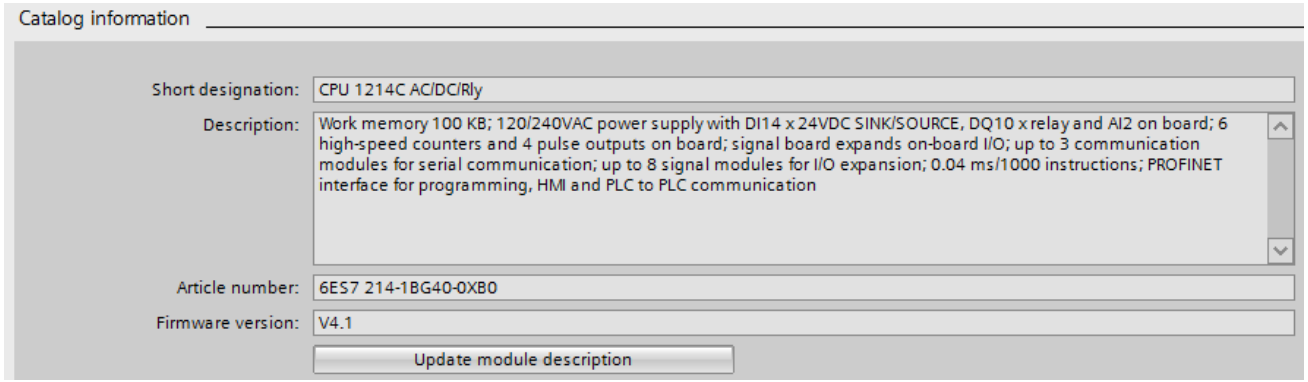

Ohjelmiston lisenssin siirto USB-muistitikulta koneelle




Ohjelmiston päivitys

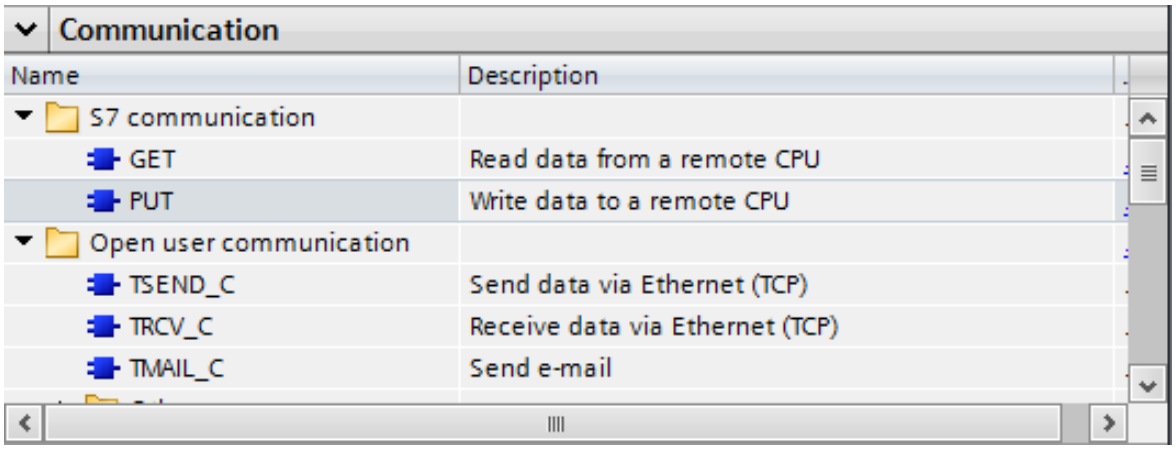



Projektinäkymä - Device view

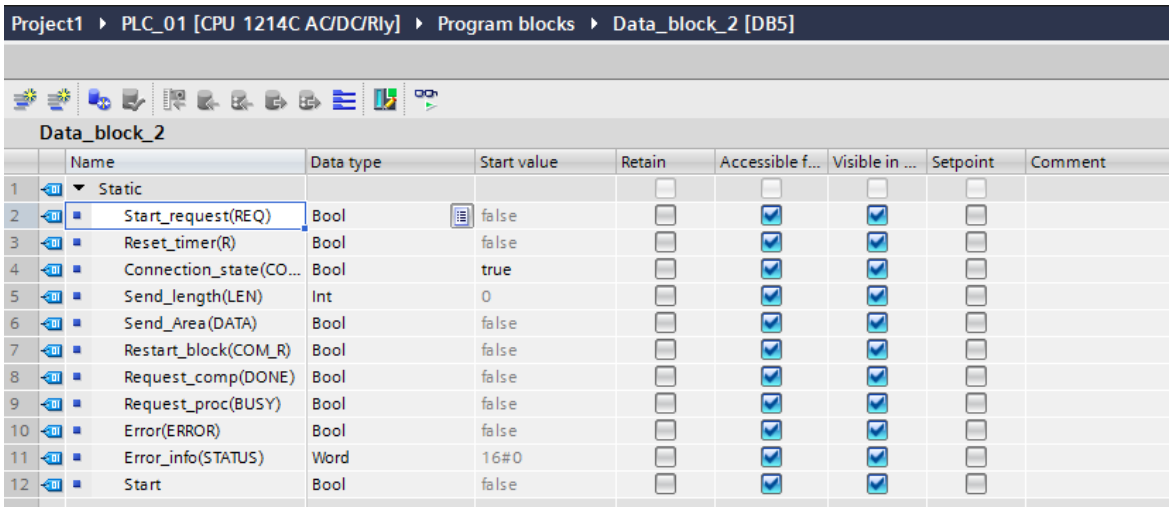



S7-1200 ohjelmoitavan logiikan tiedot


Siemens Simatic S7-1200

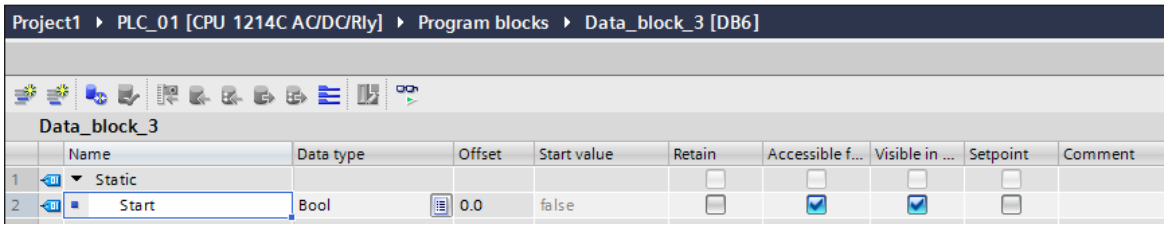



Logiikan tiedonsiirtomenetelmiä



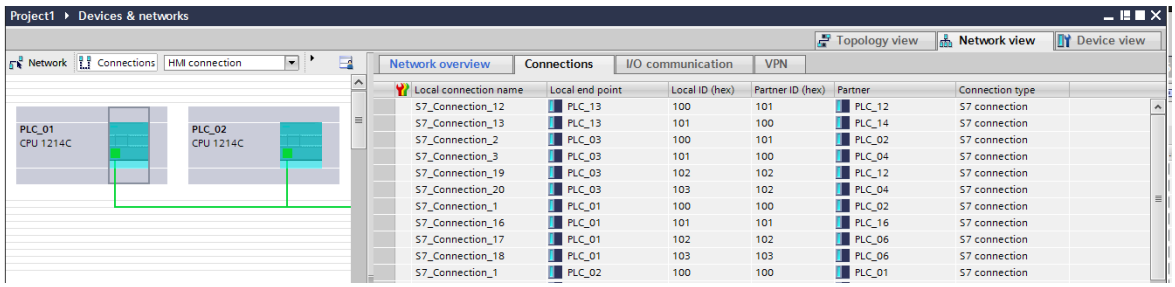

| | Name | Data type | Start value | Retain | Accessible f... | Visible in ... | Setpoint | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | Static | | | | | | | |
| 2 | Start_request(REQ) | Bool | false | | | | | |
| 3 | Reset_timer(R) | Bool | false | | | | | |
| 4 | Connection_state(CO... | Bool | true | | | | | |
| 5 | Send_length(LEN) | Int | 0 | | | | | |
| 6 | Send_Area(DATA) | Bool | false | | | | | |
| 7 | Restart_block(COM_R) | Bool | false | | | | | |
| 8 | Request_comp(DONE) | Bool | false | | | | | |
| 9 | Request_proc(BUSY) | Bool | false | | | | | |
| 10 | Error(ERROR) | Bool | false | | | | | |
| 11 | Error_info(STATUS) | Word | 16#0 | | | | | |
| 12 | Start | Bool | false | | | | | |

PUT-komennon käyttämän DB5-tietokannan sisältö

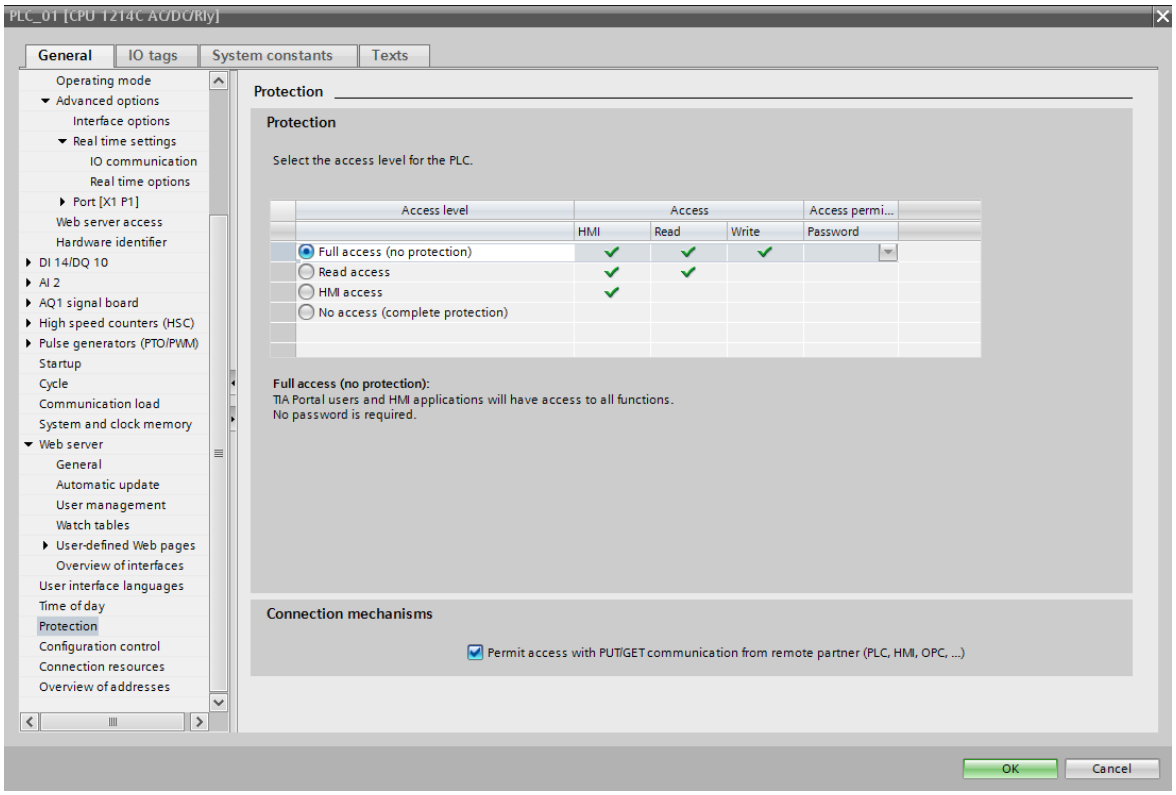



| | Name | Data type | Offset | Start value | Retain | Accessible f... | Visible in ... | Setpoint | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Static | | | | | | | | |
| 2 | Start | Bool | 0.0 | false | | | | | |

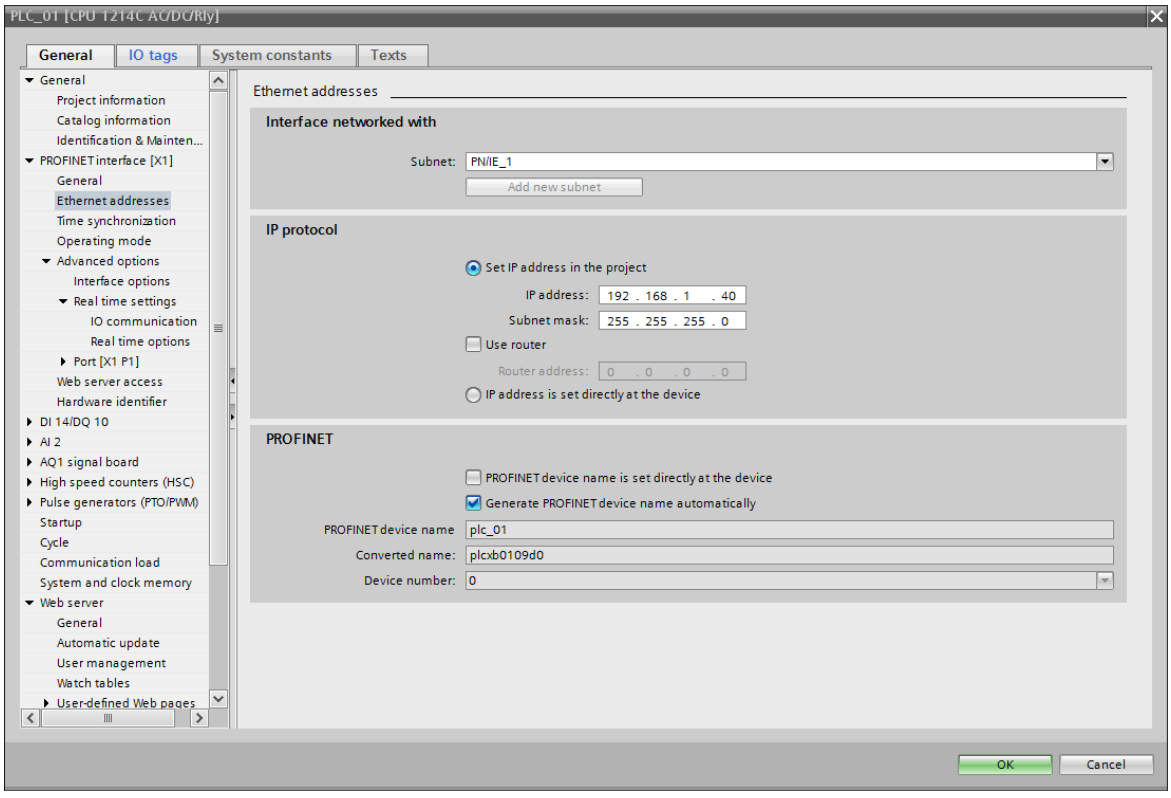
Tietokanta DB6 ja Start-arvo



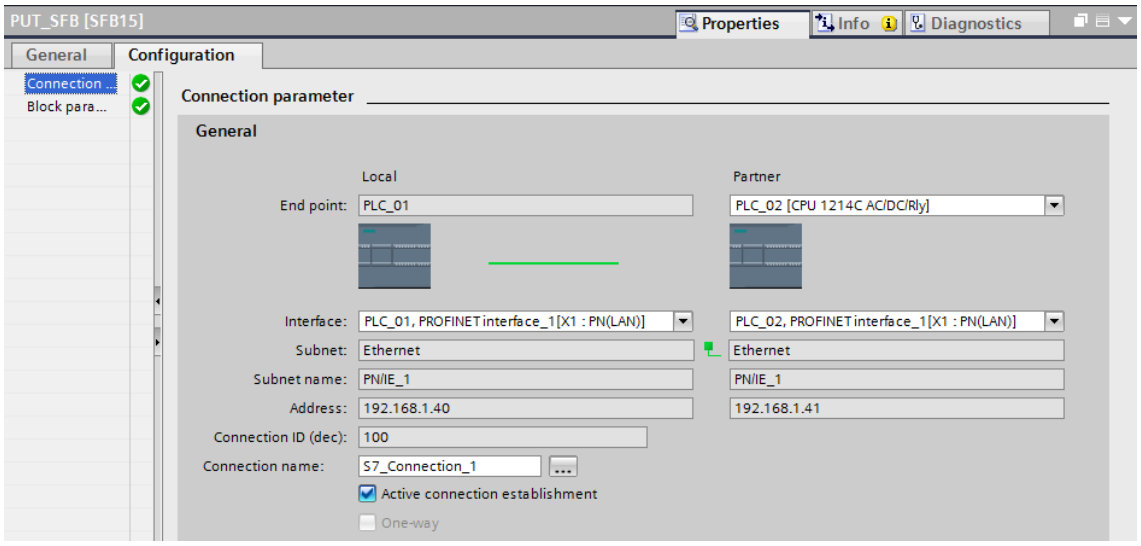

Devices & Networks S7 connections



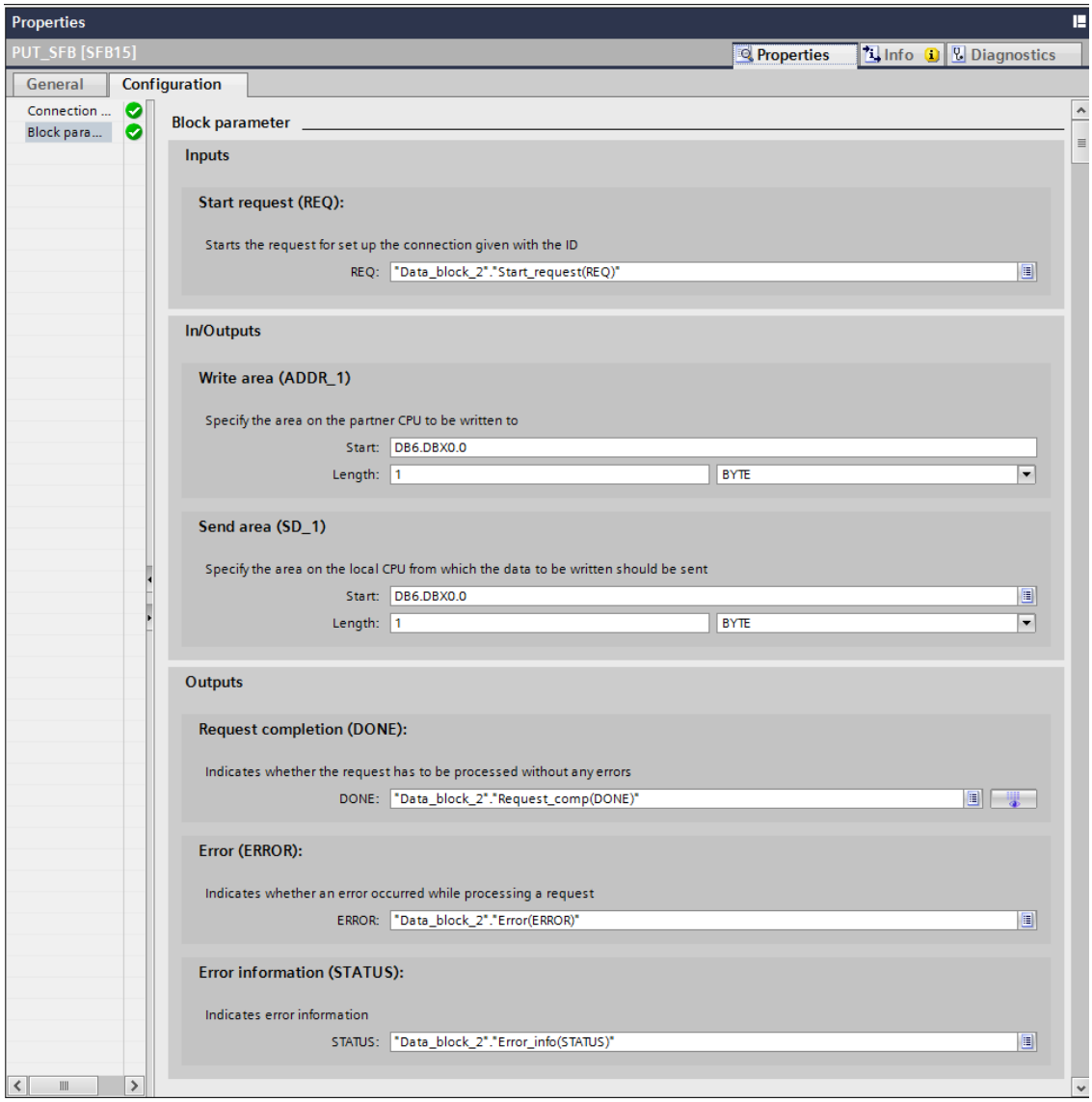

PLC1 asetuksista sallitaan PUT/GET-keskustelun käyttö




PLC_01 asetuksista asetetaan logiikan IP-osoite




PUT-komennon asetukset




PUT-komennon parametrit