

Veeti Hiukka

MODBUS TCP PLC-KOMMUNIKOINNISSA

MODBUS TCP PLC-KOMMUNIKOINNISSA

Veeti Hiukka
Opinnäytetyö
Kevät 2024
Sähkö- ja automaatiotekniikan tutkinto-
ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Sähkö- ja automaatiotekniikan tutkinto-ohjelma, automaatiotekniikka

Tekijä: Veeti Hiukka

Opinnäytetyön nimi: Modbus TCP PLC-kommunikoinnissa

Työn ohjaaja: Timo Heikkinen

Työn valmistumislukukausi ja -vuosi: kevät 2024

Sivumäärä: 31 + 1 liite

Opinnäytetyön tavoite oli saada kahden eri valmistajan ohjelmoitavat logiikkaohjaimet (PLC) kommunikoidaan keskenään käyttäen Modbus TCP:tä (Transmission Control Protocol). Työn tilaaja oli THT Control Oy. Työssä käytetyt PLC:t olivat Beckhoff Automationin CX9020 ja THT Controllin Cromi S1000. Beckhoff-automaatiojärjestelmä ohjelmoitiin käyttäen Beckhoff TwinCAT 3 -ohjelmaa ja Cromi S1000 selainpohjaista ohjelmointiympäristöä.

Projektissa luotiin etäohjaus- ja valvontamahdollisuus valmiiseen Beckhoff-automaatiojärjestelmään käyttäen Cromi S1000 -logiikkaa ja Modbus TCP:tä. Modbus TCP toimii asiakas-palvelinmenetelmällä (client-server), jossa asiakas lähettää yhdelle tai useammalle palvelimelle kyselyn, ja palvelimet palauttavat vastauksen takaisin asiakkaalle. Tässä projektissa Cromi S1000 toimi asiakkaana ja Beckhoff-järjestelmä palvelimena.

Opinnäytetyön tavoitteisiin päästiin ja lopputuloksena oli toimiva etäyhteys automaatiojärjestelmään. Etäyhteys mahdollisti Beckhoff-järjestelmän mittauksien valvomisen ja asetusarvojen ohjauksen.

Asiasanat: Modbus TCP, PLC, automaatio

ABSTRACT

Oulu University of Applied Sciences
Electrical and Automation Engineering, Automation Engineering

Author: Veeti Hiukka
Title of thesis: Modbus TCP in PLC Communication
Supervisor: Timo Heikkinen
Term and year when the thesis was submitted: spring 2024
Number of pages: 31 + 1 appendix

The goal of the thesis was to get programmable logic controllers (PLC) from two different manufacturers to communicate using Modbus TCP (Transmission Control Protocol). The thesis was commissioned by THT Control Oy. The PLCs used in the project were Beckhoff Automation's CX9020 and THT Control's Cromi S1000. The Beckhoff automation system was programmed using TwinCAT 3 software and the Cromi S1000 browser-based programming environment.

A remote control and monitoring capability was created for an existing Beckhoff automation system using Modbus TCP, where Cromi S1000 was the client and CX9020 was the server. Client sends a request to the server, and the server send an answer back to the client.

The objectives of the thesis were achieved, and the result was a working remote connection to the automation system. The remote connection enabled the Beckhoff system to monitor measurements and control setpoints.

Keywords: Modbus TCP, PLC, automation

SISÄLLYS

1	JOHDANTO	6
2	BECKHOFF-AUTOMAATIOJÄRJESTELMÄ	7
2.1	Beckhoff TwinCAT 3.....	7
2.2	TF6250 TwinCAT 3 Modbus TCP -funktio.....	8
2.3	Beckhoff CX9020 PLC	8
3	CROMI S1000	9
3.1	Cromi S1000 -käyttöliittymä.....	9
3.2	Cromi S1000 -käyttöliittymän I/O view.....	9
3.3	Cromi S1000 -käyttöliittymän Apps	10
4	MODBUS.....	11
4.1	Modbus TCP	11
4.2	Modbus TCP:n toimintaperiaate	11
5	OHJELMOITAVIEN LOGIIKKAOHJAINTEEN VÄLISEN KOMMUNIKOINNIN TOTEUTUS ..	12
5.1	Modbus TCP -väylän testaus	12
5.1.1	Modbus TCP -väylän testaus Beckhoff-järjestelmässä	12
5.1.2	Modbus TCP -väylän testaus Cromi S1000:ssa.....	15
5.2	Etävalvonnan ja -ohjauksen toteutus Beckhoff-järjestelmään	18
5.2.1	Globaalin muuttujalistan lisääminen projektiin	18
5.2.2	Beckhoff-järjestelmän etäohjattavat muuttujat	19
5.2.3	Beckhoff-järjestelmän etävalvottavat muuttujat.....	20
5.3	Etävalvonnan ja -ohjauksen toteutus Cromi S1000:n puolelle.....	21
5.3.1	Cromi S1000 I/O-moduulien lisäys.....	22
5.3.2	Etävalvottavat rekisterit S1000:n puolella	22
5.3.3	Etäohjattavat rekisterit S1000:n puolella	25
5.3.4	Rekistereiden etävalvonta ja -ohjaus S1000:n käyttöliittymästä.....	26
6	YHTEENVETO	30
	LÄHTEET.....	31

1 JOHDANTO

Automaatiotekniikassa kommunikointiprotokollat mahdollistavat tiedonsiirron useiden eri laitteiden välillä. Tässä opinnäytetyössä hyödynnetään Modiconin (nykyisin Schneider Electric) kehittämää Modbus TCP:tä (tunnetaan myös nimellä Modbus TCP/IP). Ethernet-verkon yli toimiva TCP/IP kommunikointiprotokolla on yksi yleisimmin käytetyistä kommunikaatioprotokollista automaatiotekniikassa. Modbus TCP toimii client-server-menetelmällä (asiakas-palvelin), jossa client-laite lähettää pyynnön yhdelle tai useammalle server-laitteelle, jotka lähettävät vastauksen takaisin. (1.)

Tässä opinnäytetyössä käytetään Modbus TCP:tä kahden eri valmistajan PLC:iden (Programmable Logic Controller) väliseen kommunikointiin. PLC:t ovat olennainen osa automaatiojärjestelmiä, sillä niiden avulla hallitaan monimutkaisia prosesseja ja laitteita.

Työn tilaaja on THT Control Oy ja työ on osa projektia, jossa luodaan etäohjaus- ja valvontamahdollisuus olemassa olevaan automaatiojärjestelmään. Etäohjaus ja valvonta toteutetaan THT Controlin Cromi S1000 PLC:llä hyödyntämällä Modbus TCP -rekistereitä. Projektin automaatiojärjestelmää ohjaa Beckhoff Automationin CX9020 PLC.

PLC:t ohjelmoidaan eri ohjelmointiympäristöissä laitteiden mukaan. Cromi S1000 käyttää selainpohjaista sovellusta ja Beckhoff-järjestelmä TwinCAT 3 -sovellusta. Molempien PLC:iden ohjelmointikielenä käytettiin ST:tä (Structured Text).

PLC:iden Modbus TCP -toimintojen selvittämiseksi luotiin aluksi testausympäristö, jolla simuloitiin valmista automaatiojärjestelmää, johon etäohjaus ja -valvonta toteutettiin. Testauksen tuloksista tehtiin taulukot, joista saatiin hyvä ymmärrys Modbus-rekisteriosoitteiden käytöstä eri PLC:iden välillä.

THT Control tilasi opinnäytetyöhön sisältyvän ohjeen Beckhoff TwinCAT 3 -ohjelmiston käyttämiseen. Tämä ohje sisältää PC:n (Personal Computer) ja PLC:n välisen yhteyden luomisen, logiikkaohjelman lataamisen PC:n ja PLC:n välillä molempiin suuntiin ja projektin tallentamisen varmuuskopioksi.

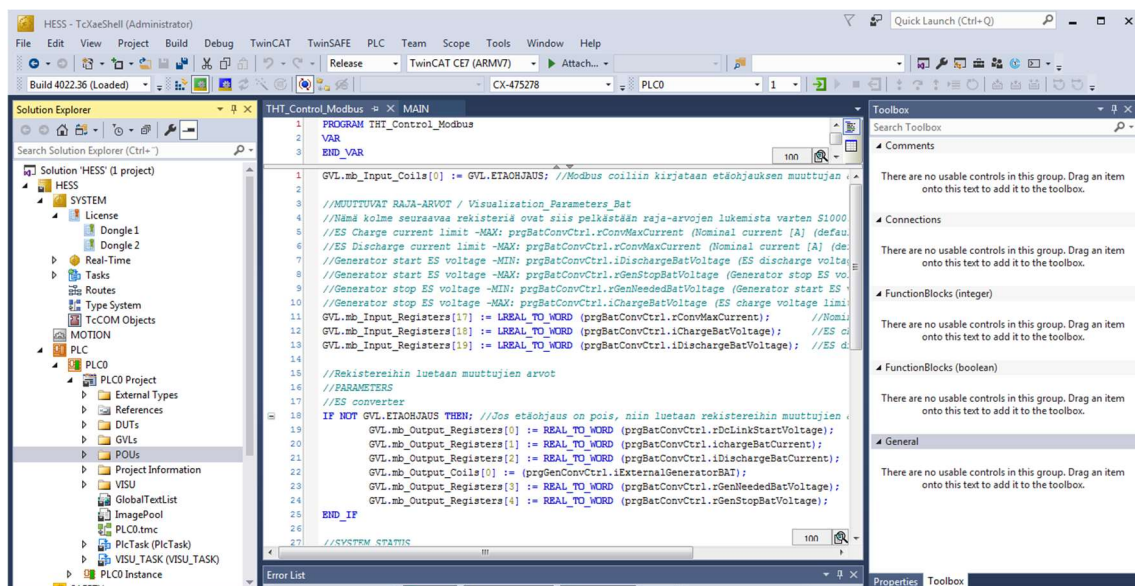
2 BECKHOFF-AUTOMAATIOJÄRJESTELMÄ

Beckhoff Automation on saksalainen yritys automaatiotekniikan alalla, joka perustettiin vuonna 1980. Beckhoff toteuttaa avoimia automaatiojärjestelmiä, jotka perustuvat PC-pohjaiseen ohjaus- tekniikkaan. Tuotevalikoima sisältää teollisuus-PC:t, I/O- ja kenttäväyläkomponentit, liikkeenoh- jaustuotteet, automaatio-ohjelmistot, ohjauskaapittoman automaation sekä konenäkölaitteistot. (2.)

2.1 Beckhoff TwinCAT 3

TwinCAT 3 (The Windows Control and Automation Technology) on Beckhoff Automationin tekemä uusien versio ohjelmistosta, jota käytetään automaatiojärjestelmien ohjelmointiin. Ohjelmisto on Windows-pohjainen ja se on rakennettu Microsoft Visual Studio -ohjelmointiympäristöön. (3.)

Tässä opinnäytetyössä käytetty versio oli TwinCAT 3 XAE (eXtended Automation Engineering) (kuva 1), jonka ohjelmointikieliin kuuluu IEC 61131-3-standardin lisäksi C/C++ ja MATLAB® / Si- mulink®. Ohjelma on modulaarinen, eli sitä voidaan laajentaa tarpeen mukaan eri funktioilla, kuten tässä työssä käytetyllä TF6250 Modbus TCP -lisäosalla, joka mahdollistaa PLC:n käytön Modbus- serverinä ja -clientinä hyödyntäen kirjastoja ja eri toimilohkoja. (3.)



KUVA 1. TwinCAT 3 XAE Shell -ohjelmisto

2.2 TF6250 TwinCAT 3 Modbus TCP -funktio

TF6250-funktio toimii yhdyskäytävänä Modbus TCP -laitteiden ja TwinCAT-ajonaikaisten järjestelmien välillä. Funktio mahdollistaa Beckhoff-järjestelmän toiminnan Modbus TCP -serverinä ja -clienttinä. Server-tilassa TwinCAT-ajonaikaiset järjestelmät voidaan yhdistää suoraan Modbus-muistialueisiin. Client-toimintaa varten on PLC-kirjasto, joka sisältää eri toimilohkoja, jotka mahdollistavat Modbus TCP -laitteen muistialueiden käyttämisen. (4.)

2.3 Beckhoff CX9020 PLC

Beckhoff Automationin kehittämä CX9020 CPU-moduuli (Computer Processing Unit) on kompakti, DIN-kiskoon asennettava Ethernet-ohjausjärjestelmä. Tässä työssä käytetyn laitteen tarkka malli on CX9020-0115. Tässä mallissa on Windows Embedded Compact 7 -käyttöjärjestelmä ja TwinCAT 3 runtime (XAR) -ajonaikainen järjestelmä. (5.)

3 CROMI S1000

Cromi S1000 on ohjelmoitava logiikkaohjain (PLC), joka tarjoaa prosessin ohjausominaisuudet sekä web-pohjaisen käyttöliittymän. S1000 on suunniteltu toimimaan tyypillisessä teollisuusympäristössä ja se on yhteensopiva useimpien eri signaalityyppien kanssa. S1000 PLC sisältää sisäänrakennettuja I/O-portteja, Ethernet-portin ja mahdollisuuden eri laajennusmoduulien käyttämiseen, esimerkiksi langattoman verkon ja energianvalvonnan.

3.1 Cromi S1000 -käyttöliittymä

Cromi S1000 web-pohjaiseen käyttöliittymään (kuva 2) voidaan kirjautua sisään, kun PC ja S1000 ovat samassa aliverkossa kirjoittamalla S1000:n IPv4 osoite PC:llä avattuun nettiselaimen. Kaikki S1000:n toimenpiteet tehdään käyttöliittymän kautta, kuten logiikkaohjelmointi, laitteen konfigurointi ja eri sovellusten luominen ja käyttö.



KUVA 2. Cromi S1000 -käyttöliittymä monitor-näkymä

3.2 Cromi S1000 -käyttöliittymän I/O view

I/O Cromi S1000:n ollessa konfigurointitilassa päästään käyttöliittymän I/O-valikosta lisäämään eri I/O-moduuleita projektiin. Vaihtoehtoihin kuuluvat sisäänrakennetut I/O:t, laajennusmoduulit, Modbus-toiminnot sekä langattoman verkon ominaisuudet (kuva 3).

Options

I/O

Logic

Apps

cTalk

Network

Users

Backup

System

Filesystem

Add I/O module

Onboard I/O

Module type: Digital inputs

Friendly name:

Add

Expansion I/O

Module type: E10 Energy Analyzer

Friendly name:

Add

Modbus I/O

Module type: Modbus TCP Scanner

Friendly name:

Add

Wireless

Module type: W1 6LowPAN Module

Friendly name:

Add

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 3. Cromi S1000 -käyttöliittymän I/O-moduulin lisäys

3.3 Cromi S1000 -käyttöliittymän Apps

S1000:n ollessa konfigurointitilassa voidaan käyttöliittymään lisätä sovelluksia (Apps), joilla suoritetaan eri toimenpiteitä (kuva 4). Sovelluksen tyypin valinta määrittää, mihin tarkoitukseen sovellusta käytetään. Tässä projektissa käytettiin HMI Values -sovellusta ja HMI Parameters Control -sovellusta, jotka mahdollistivat Modbus-rekisterien lukemisen ja kirjoittamisen.

Options

I/O

Logic

Apps

cTalk

Network

Users

Backup

System

Filesystem

Enter new application data

Application friendly name:

Application type: DPWS Points

Create

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 4. Cromi S1000 -sovelluksen lisäys käyttöliittymään

4 MODBUS

Modbus-protokolla on Modiconin (nykyisin Schneider Electricin omistama) vuonna 1979 kehittämä viestirakenne. Sitä käytetään älylaitteiden väliseen asiakas-palvelin-viestintään yleisimmin teollisuuden valmistusympäristössä. Se mahdollistaa diskreetti- ja analogi-I/O (input/output) sekä rekisteridatan siirtämisen laitteiden välillä ja toimii yhtenäisenä kielenä eri valmistajien välillä. (1.)

Modbus on suosittu teollisuusympäristöissä, sillä se on avoimesti julkaistu ja lisenssimaksuton. Se on kehitetty teollisuussovelluksia varten, helppo ottaa käyttöön ja ylläpitää muihin standardeihin verrattuna, ja se asettaa vain vähän rajoituksia siirrettävien tietojen muodolle. (1.)

4.1 Modbus TCP

Modbus TCP tai Modbus TCP/IP on muunnelma Modbus-tuoteperheen valmistajariippumattomista viestintäprotokollista, jotka on tarkoitettu automaatiolaitteiden valvontaan ja ohjaukseen. Se kattaa Modbus-viestien käytön intranet- tai internet-ympäristössä TCP/IP-protokollia käyttäen. Protokollan yleisin käyttö tällä hetkellä on PLC:iden ja I/O-moduulien liittäminen Ethernetin kautta muihin kenttäväyliin tai I/O-verkkoihin. (6.)

4.2 Modbus TCP:n toimintaperiaate

Modbus TCP -kommunikointi tarjoaa client-server-viestinnän Ethernet TCP/IP-verkkoon liitetyille laitteille. Client-server-malli perustuu neljään viestityyppiin:

1. Modbus Request (pyyntö), joka on clientin lähettämä viesti verkkoon datan siirron käynnistämiseksi.
2. Modbus Indication (osoitus), joka on serverin vastaanottama pyyntö clientiltä.
3. Modbus Response (vastaus), joka on serverin lähettämä vastaus takaisin clientille.
4. Modbus Confirmation (vahvistus), joka on clientin vastaanottama vastaus.

TCP/IP:ssä käytetään omaa otsikkoa Modbus-sovellustietoyksikön (ADU, Application Data Unit) tunnistamiseen, jonka nimi on MBAP-otsikko (Modbus Application Protocol header). (7.)

5 OHJELMOITAVIEN LOGIIKKAOHJAINTEEN VÄLISEN KOMMUNIKOINNIN TOTEUTUS

Työssä luotiin etäohjaus- ja valvontamahdollisuus olemassa olevaan Beckhoff-automaatiojärjestelmän ja Cromi S1000:n välille käyttäen Modbus TCP:tä. Etävalvonta vaati, että Beckhoff-automaatiojärjestelmän projektiin luotiin ohjelma, joka luki haluttujen muuttujien arvoja Modbus-rekistereihin. Etäohjaus toteutettiin luomalla olemassa oleviin ohjelmiin ehtoja, jotka mahdollistivat Modbus-rekisterien arvojen kirjoittamisen ohjelmien muuttujiin.

5.1 Modbus TCP -väylän testaus

Modbus TCP -väylän toiminnan selvittämistä varten luotiin simuloitu ympäristö, jossa testattiin laitteiden väliset yhteydet Modbus TCP:llä. Projekti tuli osaksi valmista automaatiojärjestelmää, jossa oli jo valmiiksi datansiirtoa laitteistojen välillä käyttäen Modbus TCP:tä.

5.1.1 Modbus TCP -väylän testaus Beckhoff-järjestelmässä

Tässä testissä Beckhoff-järjestelmä (client) lähetti pyyntöjä serverille, joissa luettiin kuutta eri input-rekisteriä ja kirjoitettiin yhteen output-rekisteriin (taulukko 1). Projektissa oli valmiiksi tehty Modbus TCP client -ohjelma (kuva 5), jota käytettiin tässä testissä. Ohjelmassa input-rekistereiden arvot luettiin muuttujiin `arrData[1]–[6]` käyttäen toimilohkoa `FB_MBReadInputRegs`. Muuttujan `sentData` arvo kirjoitettiin output-rekisteriin toimilohkolla `FB_MBWriteRegs`.

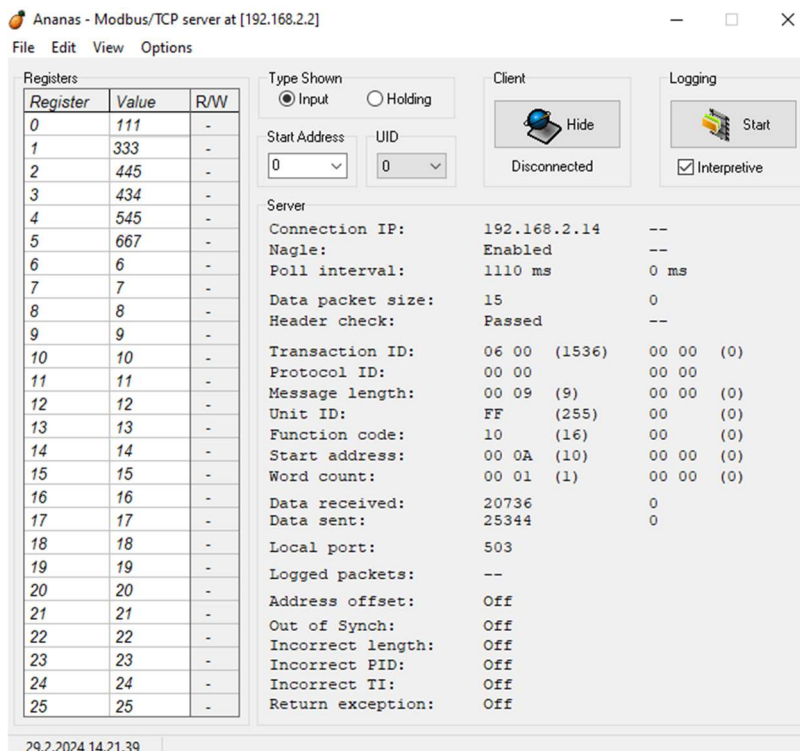
```

ModbusInit  THT_Control_Modbus  prgBatConvCtrl  prgGenConv
2  VAR
3      ip : STRING(15) := '192.168.2.2';
4      port : UINT := 503;
5      timeoutvalue : TIME := T#5S;
6      READ_reg : FB_MBReadInputRegs;
7      WriteRegs : FB_MBWriteRegs;
8
9      SentTimer : TON;
10     ReadTimer : TON;
11
12
13 END_VAR
14 VAR_INPUT
15     sentData : WORD;
16 END_VAR
17 VAR_OUTPUT
18     arrData : ARRAY [1..6] OF WORD;
19     modbusError : BOOL;
20 END_VAR
21
22 SentTimer(IN:=NOT SentTimer.Q , PT:=T#1S , Q=> , ET=> );
23 ReadTimer(IN:=NOT ReadTimer.Q , PT:=T#1S , Q=> , ET=> );
24
25 READ_reg(
26     sIPAddr:=ip ,
27     nTCPPort:=port ,
28     nUnitID:=16#FF ,
29     nQuantity:= 6 ,
30     nMBAAddr:= 0 ,
31     cbLength:=SIZEOF(arrData) ,
32     pDestAddr:=ADR(arrData) ,
33     bExecute:=ReadTimer.Q ,
34     tTimeout:=timeoutvalue ,
35     bBusy=> ,
36     bError=> ,
37     nErrId=> ,
38     cbRead=> );
39
40 WriteRegs(
41     sIPAddr:=ip ,
42     nTCPPort:=port ,
43     nUnitID:=16#FF ,
44     nQuantity:=1 ,
45     nMBAAddr:=10 ,
46     cbLength:=SIZEOF(sentData) ,
47     pSrcAddr:=ADR(sentData) ,
48     bExecute:=SentTimer.Q ,
49     tTimeout:=timeoutvalue ,
50     bBusy=> ,
51     bError=> ,
52     nErrId=> );

```

KUVA 5. Beckhoff Modbus TCP client -ohjelma

Testauksen Modbus server -laitteeksi otettiin käyttöön toinen tietokone, jolle asennettiin Ananas Modbus/TCP server -ohjelmisto. Server-ohjelmisto konfiguroitiin vastaamaan Beckhoffin Modbus TCP client -ohjelmassa määriteltä IP-osoitetta ja porttia. Ohjelmalla simuloitiin Modbus serverin toimintaa muuttamalla rekistereiden arvoja. (Kuva 6.)



KUVA 6. Ananas Modbus TCP server -ohjelmisto

Kun yhteys oli luotu, laitettiin ohjelma käyntiin. Beckhoffin ohjelma kirjoitti sendData-muuttujan arvon output-rekisterien osoitteeseen 10. Ohjelma luki myös input-rekistereiden arvoja osoitteista 0–5 kirjasi ne muuttujiin arrData[1]–[6]. (Kuva 7.)

HESS.PLC0.ModbusInit		
Expression	Type	Value
SentTimer	TON	
ReadTimer	TON	
sendData	WORD	0
arrData	ARRAY [1..6] OF WORD	
arrData[1]	WORD	111
arrData[2]	WORD	333
arrData[3]	WORD	445
arrData[4]	WORD	434
arrData[5]	WORD	545
arrData[6]	WORD	667

KUVA 7. Beckhoff Modbus TCP client -ohjelman muuttujien arvot

Lopuksi tehtiin taulukko (taulukko 1), jossa näkyi client-ohjelman muuttujat, input-rekistereiden osoitteet ja niistä luettavat arvot ja output rekisterin osoite ja siihen kirjoitettava arvo. Testauksesta saatiin hyvä ymmärrys, miten Beckhoff-järjestelmä toimi clienttinä.

TAULUKKO 1. Beckhoff-Ananas välisen yhteyden muuttujat ja rekisterit

Beckhoff (client)				Ananas (server)
rekisteri-osoite	rekisterin tyyppi	R/W	muuttujan nimi	rekisterin arvo
0	input-rekisteri	R	arrData[1]	111
1	input-rekisteri	R	arrData[2]	333
2	input-rekisteri	R	arrData[3]	445
3	input-rekisteri	R	arrData[4]	434
4	input-rekisteri	R	arrData[5]	545
5	input-rekisteri	R	arrData[6]	667
10	output-rekisteri	W	sentData	0

5.1.2 Modbus TCP -väylän testaus Cromi S1000:ssa

Seuraavaksi testattiin, miten Cromi S1000 käyttää Modbus TCP:tä. Tässä testissä Cromi S1000 (client) lähetti pyyntöjä Beckhoff-järjestelmälle (server), joissa luettiin kahta input-rekisteriä, yhtä output-rekisteriä ja kirjoitettiin toiseen output-rekisteriin (taulukko 2).

Koska Beckhoff-järjestelmä toimi väylän testauksessa samaan aikaan clienttinä Ananas-serverille, käytettiin yhteyksissä eri portteja, jotta Modbus rekisterien osoitteet eivät menisi sekaisin. Beckhoff-järjestelmän client-ohjelma käytti porttia 503 ja server-tiedosto porttia 504. Server-puolen portti vaihdettiin TF6250-funktion luomaan TcModusSrv.xml-konfiguraatitiedostoon (kuva 8).

```

<Configuration>
  <Port>504</Port>
  <IpAddr/>
  <Mapping>
    <InputRegisters>
      <MappingInfo>
        <AdsPort>851</AdsPort>
        <!-- Modbus Address 32768 = 0x8000 -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>GVL.mb_Input_Registers</VarName>
      </MappingInfo>
    </InputRegisters>
    <OutputRegisters>
      <MappingInfo>
        <AdsPort>851</AdsPort>
        <!-- Modbus Address 12288 = 0x3000 -->
        <StartAddress>12288</StartAddress>
        <EndAddress>24575</EndAddress>
        <!-- IndexGroup 16416 = 0x4020 -> plc memory area %M -->
        <IndexGroup>16416</IndexGroup>
        <IndexOffset>0</IndexOffset>
      </MappingInfo>
    </OutputRegisters>
    <InputCoils>
      <MappingInfo>
        <AdsPort>851</AdsPort>
        <!-- Modbus Address 32768 = 0x8000 -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>GVL.mb_Input_Coils</VarName>
      </MappingInfo>
    </InputCoils>
    <OutputCoils>
      <MappingInfo>
        <AdsPort>851</AdsPort>
        <!-- Modbus Address 32768 = 0x8000 -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>GVL.mb_Output_Coils</VarName>
      </MappingInfo>
    </OutputCoils>
  </Mapping>
</Configuration>

```

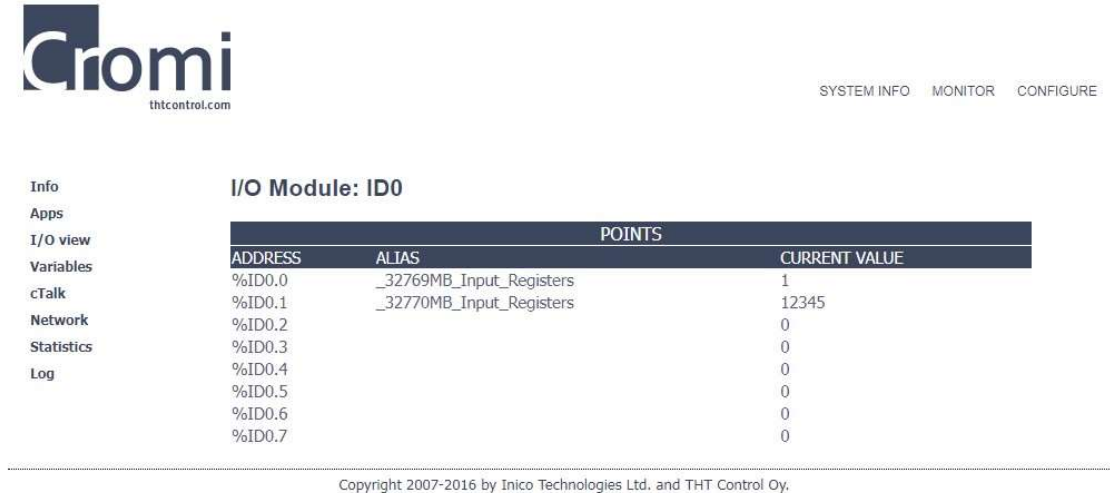
KUVA 8. TcModusSrv.xml-konfiguraatiodieto

Beckhoff-järjestelmän server-puolella rekisterien arvoja muokattiin tekemällä globaali muuttujalista TF6250-funktion luoman TcModusSrv.xml-konfiguraatiodoston mukaan. Tiedosto pidettiin muuten oletusarvoilla, mutta portiksi vaihdettiin 504. Tämä tarkoitti, että input- ja output-rekisterien aloituss osoitteet olivat 32768 ja globaalin muuttujalistan sisällä olevien muuttujien nimeksi laitettiin "mb_Input_Registers" ja "mb_Output_Registers" (kuva 9).

ModbusTestiRegToQ [Online] GVL [Online]			
HESS.PLC0.GVL			
Expression	Type	Value	Prepared value
mb_Input_Coils	ARRAY [0..4] OF BOOL		
mb_Output_Coils	ARRAY [0..4] OF BOOL		
mb_Input_Registers	ARRAY [0..4] OF WORD		
mb_Input_Registers[0]	WORD	1	
mb_Input_Registers[1]	WORD	12345	
mb_Input_Registers[2]	WORD	0	
mb_Input_Registers[3]	WORD	0	
mb_Input_Registers[4]	WORD	0	
mb_Output_Registers	ARRAY [0..4] OF WORD		
mb_Output_Registers[0]	WORD	1	
mb_Output_Registers[1]	WORD	54321	
mb_Output_Registers[2]	WORD	0	
mb_Output_Registers[3]	WORD	0	
mb_Output_Registers[4]	WORD	0	

KUVA 9. Beckhoff GVL-muuttujalista testauksen aikana

S1000 ja Beckhoff-järjestelmä käyttävät Modbus-rekistereiden osoitteita eri tavalla. Beckhoffin ensimmäinen Modbus-rekisteri alkaa osoitteesta 32768, joka vastaa S1000:n rekisteriä 32769. S1000:n käyttöliittymään lisättiin kaksi I/O-moduulia (kuva 10), joihin luotiin Modbus input- ja output-rekisterit.



I/O Module: ID0		
ADDRESS	ALIAS	CURRENT VALUE
%ID0.0	_32769MB_Input_Registers	1
%ID0.1	_32770MB_Input_Registers	12345
%ID0.2		0
%ID0.3		0
%ID0.4		0
%ID0.5		0
%ID0.6		0
%ID0.7		0

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 10. S1000 input-rekistereiden I/O-moduuli testauksen aikana

Jotta output-rekistereihin arvojen kirjoittaminen oli mahdollista, piti ohjelmaan luoda lisäksi sovellys, muuttujat ja logiikkaohjelma. Sovelluksella kirjoitettiin muuttujiin arvoja ja logiikkaan tehtiin ohjelma, joka luki muuttujan arvon output-rekisteriin. Lopuksi varmistettiin, että kaikki Modbus-pyyntö ja vastaukset menevät perille ja kaikki toimivat niin kuin oli tarkoitus. Lopuksi luotiin taulukko arvojen tarkastusta varten (taulukko 2).

TAULUKKO 2. S1000-Beckhoff välisen yhteyden muuttujat ja rekisterit

S1000 (client)				Beckhoff (server)	
rekisteri-osoite	rekisterin tyyppi	R/W	rekisterin arvo	muuttujan nimi	rekisteri-osoite
32769	input-rekisteri	R	1	mb_input_Registers[0]	32768
32770	input-rekisteri	R	12345	mb_input_Registers[1]	32769
32769	output-rekisteri	R	1	mb_Output_Registers[0]	32770
32770	output-rekisteri	W	54321	mb_Output_Registers[1]	32771

5.2 Etävalvonnan ja -ohjauksen toteutus Beckhoff-järjestelmään

Beckhoff TwinCAT 3 -ohjelma on modulaarinen, eli sitä voidaan laajentaa eri funktiolla, joiden asentaminen tapahtuu esimerkiksi Beckhoffin sivuilta. Tähän projektiin oli valmiiksi asennettuna TF6250 Modbus -funktio, jonka avulla CX9020 PLC pystyi toimimaan Modbus-clienttinä ja -serverinä.

TF6250-funktio laajentaa TwinCAT 3 -ohjelmistoa lisäämällä kirjaston, joka sisältää Modbus client -toimilohkoja, joilla kirjoitettiin ja luettiin rekistereitä. Etäohjausta ja -valvontaa varten tarvittavat server-ominaisuudet toimivat funktion asentamalla konfiguraatitiedoston avulla (kuva 8), jossa määriteltiin Modbus yhteyden IP-osoite, portti, rekisterien osoitteet ja ohjelmaan lisättävien muuttujalistan ja muuttujien nimet.

Beckhoff-järjestelmässä on HMI-paneeli (Human Machine Interface), josta seurataan järjestelmän muuttujien arvoja. Paneelin System status- ja PARAMETERS-välilehdillä olevia arvoja valvottiin ja ohjattiin tässä projektissa.

5.2.1 Globaalin muuttujalistan lisääminen projektiin

TF6250-funktion asentamassa xml-konfiguraatitiedostossa oli määritelty, kuinka CX9020 PLC käyttää Modbus-coileja ja -rekistereitä. Koska työssä käytettiin oletusasetuksia, projektiin lisätyn globaalin muuttujalistan nimeksi laitettiin GVL. Muuttujat lisättiin GVL-muuttujalistaan konfiguraatitiedoston määrittämällä tavalla (kuva 11):

- mb_Input_Coils
- mb_Input_Registers
- mb_Output_Coils
- mb_Output_Registers.

Coils-muuttujien datatyyppi oli BOOL ja Registers-muuttujien WORD. Lisäksi GVL-muuttujalistaan lisättiin BOOL-tyypin muuttuja nimeltä ETAOHJAUS, joka määrittää etäohjauksen käytön. Modbus-rekisterien ja -coilien ensimmäinen osoite on 32786, joka määrittää ensimmäisen GVL-muuttujan [0].

```

GVL  X THT_Control_Modbus prgGenConvCtrl prgBatCor
1 {attribute 'qualified_only'}
2 //Modbus rekisterit THT Control
3 VAR_GLOBAL
4     mb_Input_Coils      :   ARRAY [0..1] OF BOOL;
5     mb_Output_Coils    :   ARRAY [0..1] OF BOOL;
6     mb_Input_Registers :   ARRAY [0..19] OF WORD;
7     mb_Output_Registers :   ARRAY [0..4] OF WORD;
8
9     ETAOHJAUS : BOOL;
10
11 END_VAR

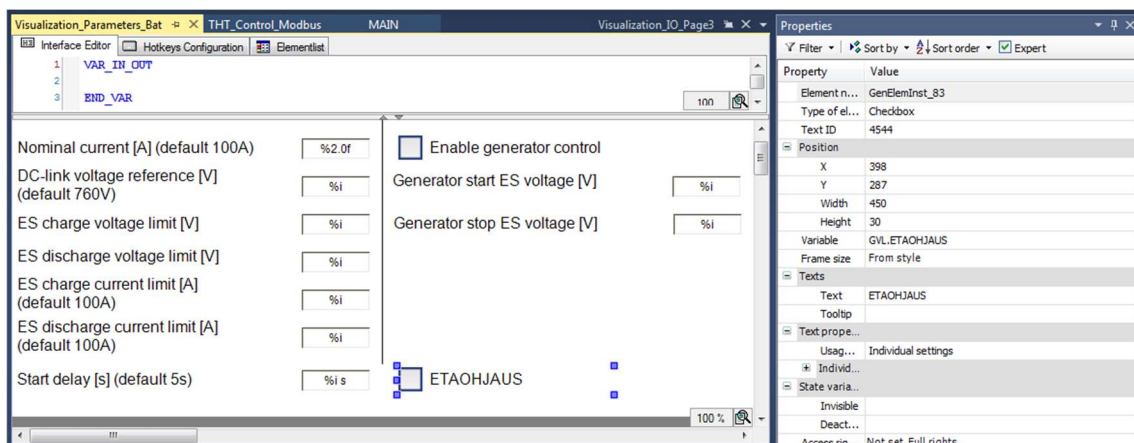
```

KUVA 11. Globaali muuttujalista GVL sisältäen muuttujat

5.2.2 Beckhoff-järjestelmän etäohjattavat muuttujat

Beckhoff-järjestelmässä oli kuusi muuttujaa, joihin luotiin etäohjausmahdollisuus. Viisi näistä muuttujista oli datatyyppiä REAL ja yksi BOOL. Tämä tarkoitti, että tarvittiin viisi Modbus output-rekisteriä ja yksi output-coil, joista kirjoitettiin arvot Beckhoff-järjestelmän ohjelmien muuttujiin. Etäohjauksen ollessa pois päältä muuttujien arvot luettiin output-rekistereihin.

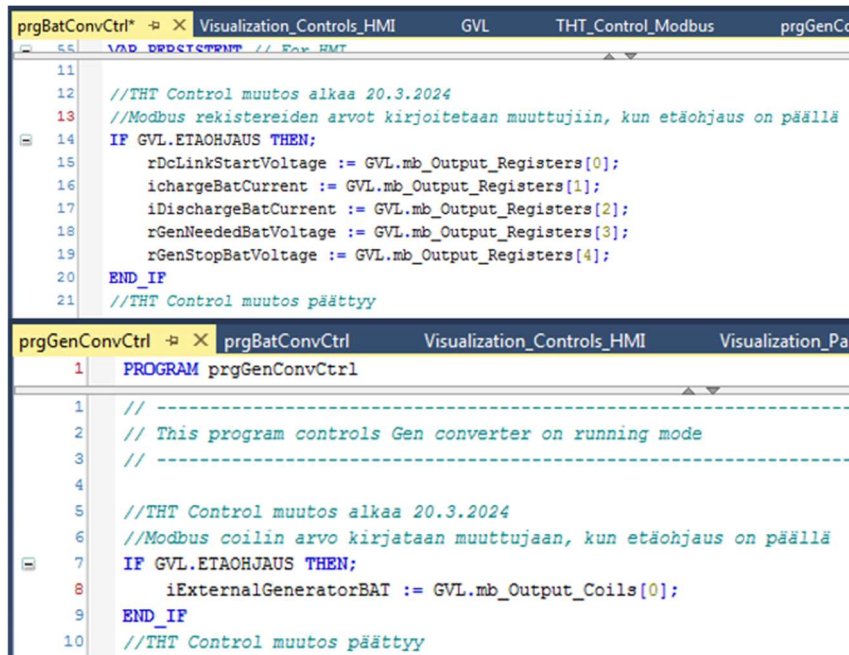
Viisi REAL-tyypin muuttujaa oli PARAMETERS-välilehden asetusarvoja, joihin luotiin etäohjaus. Yksi BOOL-tyypin muuttuja oli etäohjauksen valintapainike, joka luotiin välilehdelle (kuva 12). Etäohjauspainike kirjoitti globaaliin muuttujaan ETAOHJAUS arvon 1 tai 0.



KUVA 12. ETAOHJAUS-painike PARAMETERS-välilehdellä

Etäohjattavien muuttujien arvoja kirjoitettiin Modbus-rekistereistä etäohjauksen ollessa päällä. Tämä toteutettiin luomalla ehto Beckhoff-järjestelmän ohjelmiin, joissa muuttujat olivat. Muuttujien arvojen kirjoitus Modbus-rekistereistä toteutuu, kun globaalin muuttujan ETAOHJAUS arvo on 1.

Tämä ehto luotiin kahteen Beckhoff-ohjelmaan nimiltä prgBatConvCtrl ja prgGenConvCtrl (kuva 13).



```
prgBatConvCtrl* -> Visualization_Controls_HMI GVL THT_Control_Modbus prgGenCo
11 //THT Control muutos alkaa 20.3.2024
12 //Modbus rekistereiden arvot kirjoitetaan muuttujiin, kun etäohjaus on päällä
13 IF GVL.ETAOHJAUS THEN;
14   rDcLinkStartVoltage := GVL.mb_Output_Registers[0];
15   iChargeBatCurrent := GVL.mb_Output_Registers[1];
16   iDischargeBatCurrent := GVL.mb_Output_Registers[2];
17   rGenNeededBatVoltage := GVL.mb_Output_Registers[3];
18   rGenStopBatVoltage := GVL.mb_Output_Registers[4];
19 END_IF
20 //THT Control muutos päättyy

prgGenConvCtrl -> prgBatConvCtrl Visualization_Controls_HMI Visualization_Par
1 PROGRAM prgGenConvCtrl
2 // -----
3 // This program controls Gen converter on running mode
4 // -----
5 //THT Control muutos alkaa 20.3.2024
6 //Modbus coilin arvo kirjataan muuttujaan, kun etäohjaus on päällä
7 IF GVL.ETAOHJAUS THEN;
8   iExternalGeneratorBAT := GVL.mb_Output_Coils[0];
9 END_IF
10 //THT Control muutos päättyy
```

KUVA 13. Beckhoff-järjestelmän ohjelmat, joihin luotiin etäohjauksen ehto

5.2.3 Beckhoff-järjestelmän etävalvottavat muuttujat

Kaikkiin muuttujiin luotiin etävalvonta, mutta vain osaan luotiin ohjaus. Muuttujia, joille ei luotu ohjausta, oli yhteensä 21 kappaletta. 17 näistä muuttujista sijaitsi System Status -välilehdellä ja niiden datatyyppi oli LREAL. Näiden lisäksi luettiin PARAMETERS-välilehdeltä kolmea REAL-datatyyppin muuttujaa etäohjauksen raja-arvoja varten ja yhtä BOOL-tyypin muuttujaa etäohjauksen valintaa varten.

Muuttujien arvojen lukeminen toteutettiin luomalla uusi ohjelma nimeltä THT_Control_Modbus (kuva 14). Ohjelma luki etävalvottavien muuttujien arvoja input-rekistereihin jatkuvasti. Etäohjauksen ollessa pois päältä, ohjelma luki myös etäohjattavien muuttujien arvoja output-rekistereihin.

Etävalvottavien muuttujien datatypit olivat REAL ja LREAL, mutta globaalin muuttujalistasta Modbus-rekisterien muuttujat olivat datatyyppiä WORD. Koska datatypit eivät täsmänneet, käytettiin ohjelmassa funktioita *REAL_TO_WORD()* ja *LREAL_TO_WORD*, joilla muutettiin REAL- ja LREAL-datatypit WORD-muotoon.

THT_Control_Modbus	prgBatConvCtrl	prgGenConvCtrl	GVL	Visualization_Controls_HMI	Visualization_Parameters_Bat
--------------------	----------------	----------------	-----	----------------------------	------------------------------

```

1 GVL.mb_Input_Coils[0] := GVL.ETAOHJAUS; //Modbus coilin kirjataan etäohjauksen muuttujan arvo (BOOL) S1000 varten.
2
3 //MUUTTUVA RAJA-ARVOT / Visualization_Parameters_Bat
4 //Nämä kolme seuraavaa rekisteriä ovat siis pelkästään raja-arvojen lukemista varten S1000:lle
5 //ES Charge current limit -MAX: prgBatConvCtrl.rConvMaxCurrent (Nominal current [A] (default 100A))
6 //ES Discharge current limit -MAX: prgBatConvCtrl.rConvMaxCurrent (Nominal current [A] (default 100A))
7 //Generator start ES voltage -MIN: prgBatConvCtrl.iDischargeBatVoltage (ES discharge voltage limit [V])
8 //Generator start ES voltage -MAX: prgBatConvCtrl.rGenStopBatVoltage (Generator stop ES voltage [V])
9 //Generator stop ES voltage -MIN: prgBatConvCtrl.rGenNeededBatVoltage (Generator start ES voltage [V])
10 //Generator stop ES voltage -MAX: prgBatConvCtrl.iChargeBatVoltage (ES charge voltage limit [V])
11 GVL.mb_Input_Registers[17] := LREAL_TO_WORD (prgBatConvCtrl.rConvMaxCurrent); //Nominal current [A] (default 100A)
12 GVL.mb_Input_Registers[18] := LREAL_TO_WORD (prgBatConvCtrl.iChargeBatVoltage); //ES charge voltage limit [V]
13 GVL.mb_Input_Registers[19] := LREAL_TO_WORD (prgBatConvCtrl.iDischargeBatVoltage); //ES discharge voltage limit [V]
14
15 //Rekistereihin luetaan muuttujien arvot
16 //PARAMETERS
17 //ES converter
18 IF NOT GVL.ETAOHJAUS THEN; //Jos etäohjaus on pois, niin luetaan rekistereihin muuttujien arvot.
19     GVL.mb_Output_Registers[0] := REAL_TO_WORD (prgBatConvCtrl.rDcLinkStartVoltage); //DC-Link voltage reference
20     GVL.mb_Output_Registers[1] := REAL_TO_WORD (prgBatConvCtrl.iChargeBatCurrent); //ES Charge current limit
21     GVL.mb_Output_Registers[2] := REAL_TO_WORD (prgBatConvCtrl.iDischargeBatCurrent); //ES Discharge current limit
22     GVL.mb_Output_Coils[0] := (prgGenConvCtrl.iExternalGeneratorBAT); //Enable generator control
23     GVL.mb_Output_Registers[3] := REAL_TO_WORD (prgBatConvCtrl.rGenNeededBatVoltage); //Generator start ES Voltage
24     GVL.mb_Output_Registers[4] := REAL_TO_WORD (prgBatConvCtrl.rGenStopBatVoltage); //Generator stop ES voltage
25 END_IF
26
27 //SYSTEM STATUS
28 //EV converter
29 GVL.mb_Input_Registers[0] := LREAL_TO_WORD (Inputs_Main.rIn_PV InputVoltage); //PV array voltage
30 GVL.mb_Input_Registers[1] := LREAL_TO_WORD (Inputs_Main.rIn_PV_ES_Current); //PV array current
31
32 //AC Input Converter
33 GVL.mb_Input_Registers[2] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_L1); //L1 in
34 GVL.mb_Input_Registers[3] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_L2); //L2 in
35 GVL.mb_Input_Registers[4] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_L3); //L3 in
36 GVL.mb_Input_Registers[5] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_IN_L1_Current); //L1 current
37 GVL.mb_Input_Registers[6] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_IN_L2_Current); //L2 current
38 GVL.mb_Input_Registers[7] := LREAL_TO_WORD (Inputs_Main.lIn_GEN_IN_L3_Current); //L3 current
39
40 //ES converter
41 GVL.mb_Input_Registers[8] := LREAL_TO_WORD (Inputs_Main.rIn_BAT_InputVoltage); //ES voltage
42 GVL.mb_Input_Registers[9] := LREAL_TO_WORD (Inputs_Main.BAT_ES_currentReal); //ES current
43
44 //BESS system
45 GVL.mb_Input_Registers[10] := LREAL_TO_WORD (Inputs_Main.rIn_DCLC_DCLinkVoltage); //DC link voltage
46
47 //Off-Grid Inverter
48 GVL.mb_Input_Registers[11] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L1); //L1 out
49 GVL.mb_Input_Registers[12] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L2); //L2 out
50 GVL.mb_Input_Registers[13] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L3); //L3 out
51 GVL.mb_Input_Registers[14] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L1_Current); //L1 current
52 GVL.mb_Input_Registers[15] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L2_Current); //L2 current
53 GVL.mb_Input_Registers[16] := LREAL_TO_WORD (Inputs_Main.lIn_OUT_L3_Current); //L3 current

```

KUVA 14. Beckhoff-järjestelmään luotu THT_Control_Modbus-ohjelma

5.3 Etävalvonnan ja -ohjauksen toteutus Cromi S1000:n puolelle

Etävalvonta ja -ohjaus toteutettiin luomalla käyttöliittymäsovellukset ja logiikkaohjelmat S1000:een. Käyttöliittymäsovelluksista valottiin input-rekisterien arvoja ja määriteltiin asetusarvoja output-rekistereihin. Arvojen kirjoittaminen ja lukeminen tapahtuivat logiikkaohjelmissa, joissa muuttujien arvot kirjoitettiin output-rekistereihin ja input-rekisterien arvot luettiin muuttujiin. TCP-yhteyttä varten määritettiin laitteen IP-osoite ja Modbus-yhteyden portti.

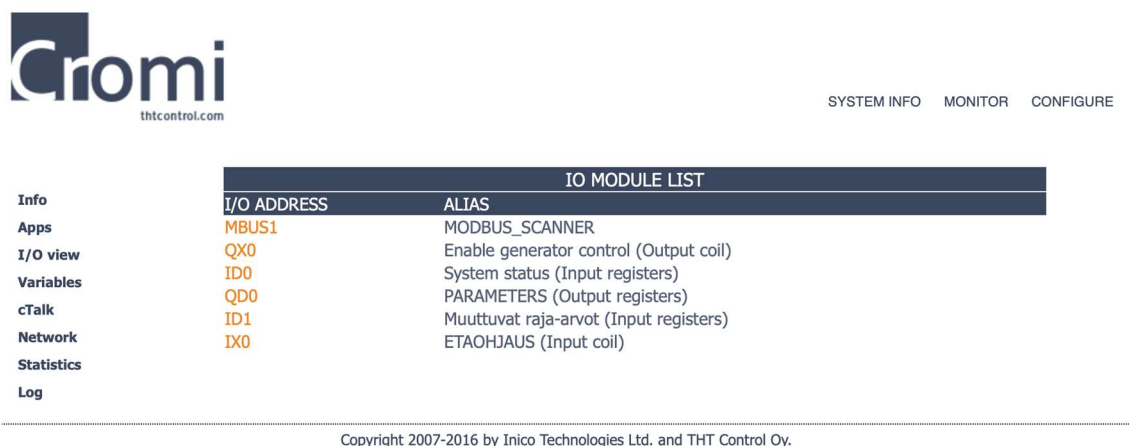
Beckhoff-järjestelmässä oli asetusarvoille määritetty raja-arvot, jotka estivät liian suuren tai liian pienen luvun kirjoittamisen muuttujiin. Nämä raja-arvot luotiin myös S1000:n logiikkaohjelmiin. Osa raja-arvoista oli staattisia ja osa luettiin Beckhoff-järjestelmästä input-rekistereinä. Logiikkaohjelmat asettivat alarajaksi määritetyn arvon, kun syötetty luku alitti alarajan ja ylärajaksi määritetyn arvon, kun syötetty luku ylitti ylärajan.

5.3.1 Cromi S1000 I/O-moduulien lisäys

Cromi S1000:n ohjelmistoon lisättiin kuusi I/O-moduulia (kuva 15) seuraavilla Modbus-toiminnoilla:

- Modbus TCP Scanner, jossa määritettiin IP-osoite ja portti yhteyden luomista varten
- Discrete inputs (Read), joka mahdollisti input-coilin lukemisen
- Coils (R/W), joka mahdollisti output-coilin lukemisen ja kirjoittamisen
- Kaksi Register inputs (Read) -moduulia, jotka mahdollistivat input-rekisterien lukemisen
- Holding Registers (R/W), joka mahdollisti output-rekisterien lukemisen ja kirjoittamisen.

Modbus I/O-moduulit nimettiin ja niihin lisättiin kaikki input-rekisterit, output-rekisterit, input-coilit ja output-coilit, joille annettiin omat nimet ja osoitteet, jotta niitä voitiin käyttää logiikkaohjelmissa.



The screenshot shows the Cromi S1000 web interface. At the top left is the 'Cromi' logo with 'thtcontrol.com' below it. At the top right are links for 'SYSTEM INFO', 'MONITOR', and 'CONFIGURE'. On the left side is a navigation menu with items: Info, Apps, I/O view, Variables, cTalk, Network, Statistics, and Log. The main content area displays a table titled 'IO MODULE LIST' with two columns: 'I/O ADDRESS' and 'ALIAS'. The table contains the following data:

I/O ADDRESS	ALIAS
MBUS1	MODBUS_SCANNER
QX0	Enable generator control (Output coil)
ID0	System status (Input registers)
QD0	PARAMETERS (Output registers)
ID1	Muuttuvat raja-arvot (Input registers)
IX0	ETAOHJAUS (Input coil)

At the bottom of the page, there is a copyright notice: 'Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.'

KUVA 15. Cromi S1000 lisätyt I/O-moduulit

5.3.2 Etävalvottavat rekisterit S1000:n puolella

Muuttujia, joille ei luotu etäohjausta, oli yhteensä 21 kappaletta. 17 näistä muuttujista sijaitsi Beckhoff-järjestelmän System Status -välilehdellä ja loput 4 PARAMETERS-välilehdellä. Rekistereitä varten luotiin kaksi Register inputs (Read) -moduulia ja yksi Discrete inputs (Read) -moduuli S1000:n käyttöliittymään.

System Status -välilehden muuttujien arvoja luettiin Beckhoffin puolella GVL-muuttujiin mb_Input_Registers[0]–[16]. Konfiguraatiotiedosto TcModusSrv.xml määrittä nämä muuttujat input-rekisterien osoitteisiin 32768–32784. Cromi S1000 käyttää Modbus-rekisterien osoitteita eri tavoin kuin Beckhoff-järjestelmä. S1000:n rekisteriosoite 32769 vastaa Beckhoff-järjestelmän rekisteriosoitetta 32768. Tästä syystä määritettiin nämä input-rekisterit S1000:n I/O-moduulissa osoitteisiin 32769–32785 (kuva 16).

Configuration: ID0

Friendly name:

REMOTE REGISTERS	
Register type:	Register inputs
Modbus scanner:	TCP-1-1 / 192.168.2.3
Slave address:	0 (Serial: 1-247, TCP: 0/255)
Register quantity:	17
Register type:	Unsigned 16-bit
Max request size:	120 (1-120)
Allow gaps in request:	No

			POINTS			
ADDRESS	REGISTER	ALIAS	SCALE	RAW L	RAW H	SCALED L
%ID0.0	32769	PV_converter_PV_array_voltage	<input type="checkbox"/>			
%ID0.1	32770	PV_converter_PV_array_current	<input type="checkbox"/>			
%ID0.2	32771	AC_Input_Converter_L1_in	<input type="checkbox"/>			
%ID0.3	32772	AC_Input_Converter_L2_in	<input type="checkbox"/>			
%ID0.4	32773	AC_Input_Converter_L3_in	<input type="checkbox"/>			
%ID0.5	32774	AC_Input_Converter_L1_current	<input type="checkbox"/>			
%ID0.6	32775	AC_Input_Converter_L2_current	<input type="checkbox"/>			
%ID0.7	32776	AC_Input_Converter_L3_current	<input type="checkbox"/>			
%ID0.8	32777	ES_converter_ES_voltage	<input type="checkbox"/>			
%ID0.9	32778	ES_converter_ES_current	<input type="checkbox"/>			
%ID0.10	32779	HESS_system_DC_link_voltage	<input type="checkbox"/>			
%ID0.11	32780	Off_Grid_Inverter_L1_out	<input type="checkbox"/>			
%ID0.12	32781	Off_Grid_Inverter_L2_out	<input type="checkbox"/>			
%ID0.13	32782	Off_Grid_Inverter_L3_out	<input type="checkbox"/>			
%ID0.14	32783	Off_Grid_Inverter_L1_current	<input type="checkbox"/>			
%ID0.15	32784	Off_Grid_Inverter_L2_current	<input type="checkbox"/>			
%ID0.16	32785	Off_Grid_Inverter_L3_current	<input type="checkbox"/>			

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 16. Etävalvottavat input-rekisterit S1000:n I/O-moduulissa

Etäohjauksen raja-arvojen muuttujat PARAMETERS-välilehdeltä luettiin Beckhoffin puolella GVL-muuttujiin mb_Input_Registers[17]–[19]. Konfiguraatiedostossa nämä muuttujat vastasivat input-rekisterien osoitteita 32785–32787. Nämä rekisterit lisättiin S1000:n toiseen Input registers-I/O-moduuliin osoitteisiin 32786–32788 (kuva 17).

Configuration: ID1

Friendly name:

REMOTE REGISTERS	
Register type:	Register inputs
Modbus scanner:	TCP-1-1 / 192.168.2.3
Slave address:	(Serial: 1-247, TCP: 0/255)
Register quantity:	3
Register type:	Unsigned 16-bit
Max request size:	120 (1-120)
Allow gaps in request:	No

POINTS					
ADDRESS	REGISTER	ALIAS	SCALE	RAW L	RAW H
%ID1.0	32786	rConvMaxCurrent_REG	<input type="checkbox"/>		
%ID1.1	32787	iChargeBatVoltage_REG	<input type="checkbox"/>		
%ID1.2	32788	iDischargeBatVoltage_REG	<input type="checkbox"/>		

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 17. Etäohjauksen raja-arvojen input-rekisterit S1000:n I/O-moduulissa

Etäohjauspainikkeella kirjoitettiin globaaliin muuttujaan ETAOHJAUS arvo 1 tai 0. Tämä määrittä, oliko etäohjaus päällä vai pois päältä. Tämä muuttuja luettiin input-coiliin mb_Input_Coils[0], jotta S1000:n puolella tiedettiin, oliko etäohjaus käytössä. Konfiguraatiodiedosto määrittä tämän muuttujan input-coiliin 32768. Tämä coili luotiin Discrete inputs -moduulin osoitteeseen 32789 (kuva 18).

Configuration: IX0

Friendly name:

REMOTE REGISTERS	
Register type:	Discrete inputs
Modbus scanner:	TCP-1-1 / 192.168.2.3
Slave address:	0 (Serial: 1-247, TCP: 0/255)
Register quantity:	1
Max request size:	120 (1-120)
Allow gaps in request:	No

POINTS		
ADDRESS	REGISTER	ALIAS
%IX0.0	32769	ETAOHJAUS_COIL

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 18. Etäohjauksen valinnan muuttujan coili S1000:n I/O-moduulissa

5.3.3 Etäohjattavat rekisterit S1000:n puolella

Etäohjattavia muuttujia oli kuusi kappaletta ja ne sijaitsivat Beckhoff-järjestelmän PARAMETERS-välilehdellä. Rekistereiden lukemista ja kirjoitusta varten lisättiin S1000:n käyttöliittymään yksi Holding Registers R/W -I/O-moduuli ja yksi Coils R/W -I/O-moduuli. Etäohjattavien muuttujien arvoja luettiin output-rekistereihin THT_Control_Modbus-ohjelmassa etäohjauksen ollessa pois päältä.

PARAMETERS-välilehden etäohjattavat REAL-tyyppin muuttujat luettiin ja kirjoitettiin Beckhoff-järjestelmässä globaaliin muuttujiin mb_Output_Registers[0]–[5]. Konfiguraatiodiedosto määrittä nämä output-rekistereihin 32768–32772. Nämä rekisterit lisättiin S1000:n Holding registers R/W -I/O-moduuliin osoitteisiin 32769–32773 (kuva 19).

Cromi
tthcontrol.com

SYSTEM INFO MONITOR CONFIGURE

Options
I/O
Logic
Apps
cTalk
Network
Users
Backup
System
Filesystem

Configuration: QD0

Friendly name:

REMOTE REGISTERS

Register type: Holding registers R/W
Modbus scanner:
Slave address: (Serial: 1-247, TCP: 0/255)
Register quantity:
Register type:
Max request size: (1-120)
Allow gaps in request:

POINTS

ADDRESS	REGISTER	ALIAS	SCALE	RAW L	RAW H	SCALED L	SCALED H
%QD0.0	32769	DC_Link_voltage_reference_REG	<input type="checkbox"/>				
%QD0.1	32770	ES_charge_current_limit_REG	<input type="checkbox"/>				
%QD0.2	32771	ES_discharge_current_limit_REG	<input type="checkbox"/>				
%QD0.3	32772	Generator_start_ES_voltage_REG	<input type="checkbox"/>				
%QD0.4	32773	Generator_stop_ES_voltage_REG	<input type="checkbox"/>				

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 19. REAL-datatyypin etäohjattavat rekisterit S1000:n I/O-moduulissa

PARAMETERS-välilehdellä oli lisäksi yksi BOOL-datatyypin muuttuja, joka luettiin globaaliin muuttujaan mb_Output_Coils[0]. Konfiguraatiodiedosto määrittä tämän muuttujan coil-osoitteeksi 32768. Tämä coil lisättiin Coils R/W -I/O-moduulin osoitteeseen 32769 S1000:n puolella (kuva 20).

Configuration: QX0

Friendly name: Enable generator control (C

REMOTE REGISTERS		
Register type:	Coils R/W	
Modbus scanner:	TCP-1-1 / 192.168.2.3	
Slave address:	0	(Serial: 1-247, TCP: 0/255)
Register quantity:	1	
Max request size:	120	(1-120)
Allow gaps in request:	No	

POINTS		
ADDRESS	REGISTER	ALIAS
%QX0.0	32769	Enable_generator_control_COIL

Save

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

KUVA 20. BOOL-datatyypin etäohjattava coili S1000:n I/O-moduulissa

5.3.4 Rekistereiden etävalvonta ja -ohjaus S1000:n käyttöliittymästä

S1000:n käyttöliittymään luotiin sovellukset, joista Modbus-rekistereitä valvottiin ja ohjattiin. Etäohjaukseen käytettiin HMI-parameters-sovellusta ja etävalvontaan HMI-values-sovellusta. HMI-values-sovellukseen lisättiin suoraan input-rekisterit ilman ohjelman sisäisiä välimuuttujia, sillä niihin ei luotu ohjausta (kuva 21).

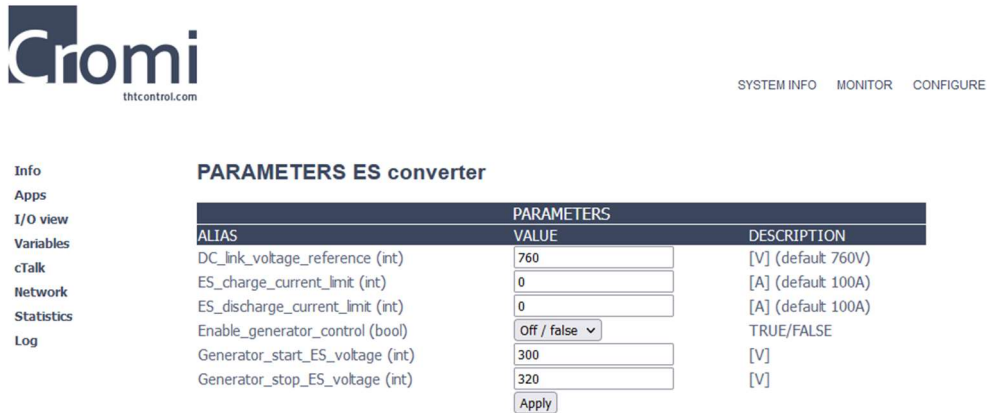
System status

VALUES			
ALIAS	VALUE	DESCRIPTION	
PV_converter_PV_array_voltage	0		
PV_converter_PV_array_current	0		
AC_Input_Converter_L1_in	0		
AC_Input_Converter_L2_in	0		
AC_Input_Converter_L3_in	0		
AC_Input_Converter_L1_current	0		
AC_Input_Converter_L2_current	0		
AC_Input_Converter_L3_current	0		
ES_converter_ES_voltage	0		
ES_converter_ES_current	0		
HESS_system_DC_link_voltage	0		
Off_Grid_Inverter_L1_out	0		
Off_Grid_Inverter_L2_out	0		
Off_Grid_Inverter_L3_out	0		
Off_Grid_Inverter_L1_current	0		
Off_Grid_Inverter_L2_current	0		
Off_Grid_Inverter_L3_current	0		

Copyright 2007-2016 by Inico Technologies Ltd. and THT Control Oy.

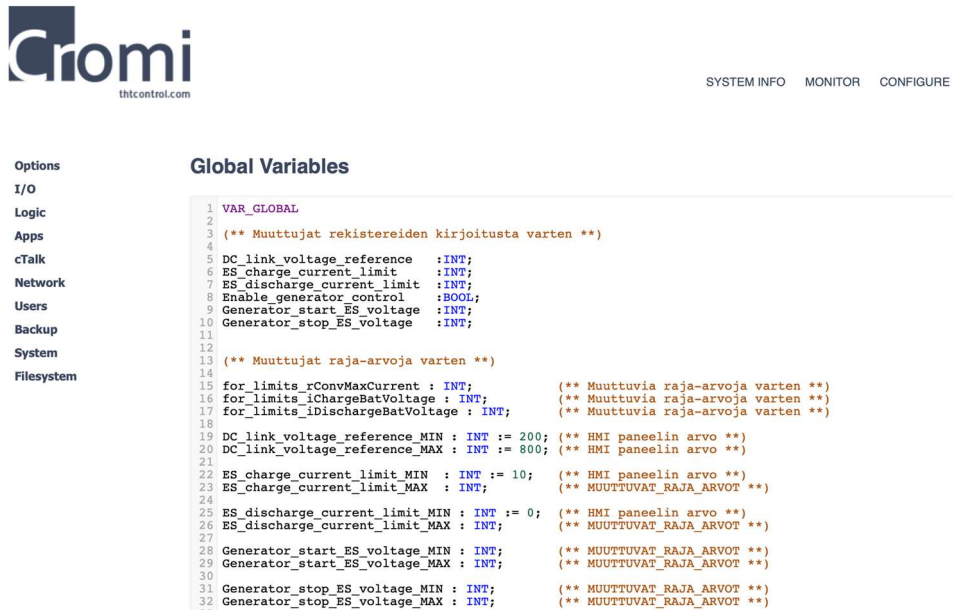
KUVA 21. Etävalvottavat rekisterit S1000:n käyttöliittymän System Status -sovelluksessa

HMI-parameters-sovellus sisälsi kaikki etäohjattavat muuttujat. Etäohjauksen ollessa päällä sovelluksella muokattiin välimuuttujien arvoja (kuva 22), jotka kirjoitettiin logiikkaohjelmassa oikeaan output-rekisteriin.



KUVA 22. Etäohjattavat muuttujat S1000:n käyttöliittymän PARAMETERS-sovelluksessa

S1000:n logiikkaan luotiin globaali muuttujalista (kuva 23), joka sisälsi kaikki tarvittavat välimuuttujat etäohjattavia output-rekistereitä varten. Lisäksi muuttujalistaan lisättiin etäohjattavien rekistereiden raja-arvomuuuttujat.



KUVA 23. Etäohjattavien rekistereiden välimuuttujat S1000:n globaalissa muuttujalistassa

Projektiin luotiin logiikkaohjelmia muuttujien kirjoitusta ja lukemista varten. Luotiin yksi ohjelma etäohjattavien rekistereiden arvojen kirjoitusta ja lukemista varten, jossa etäohjauksen ollessa päällä globaalien muuttujien arvot kirjoitettiin rekistereihin ja etäohjauksen ollessa pois päältä rekisterien arvot luettiin globaaleihin muuttujiin (kuva 24).

mb_OUTPUT_REGS

```

1 PROGRAM mb_OUTPUT_REGS
2 VAR
3   (** Locals here **)
4 END_VAR
5
6   (** for limits x muuttujiin kirjoitetaan rekistereiden arvot.
7     Nämä kolme ovat MIN/MAX raja-arvoja varten **)
8
9   for_limits_rConvMaxCurrent := rConvMaxCurrent_REG; (** Nominal current [A] (default100A)
10  for_limits_iChargeBatVoltage := iChargeBatVoltage_REG; (** ES charge voltage limit [V]
11  for_limits_iDischargeBatVoltage := iDischargeBatVoltage_REG; (** ES discharge voltage lim
12
13
14
15  IF ETAOHJAUS_COIL THEN; (** Jos ETAOHJAUS_COIL on päällä, kirjoitetaan rekistereihin
16    muuttujien arvot.
17    Jos ei, niin muuttujiin kirjoitetaan rekisterien arvot. **)
18    (** "_REG/COIL" ovat rekistereitä **)
19
20    DC link voltage reference_REG := DC link voltage reference;
21    ES charge current limit_REG := ES charge current limit ;
22    ES discharge current limit_REG := ES discharge current limit;
23    Enable_generator_control_COIL := Enable_generator_control;
24    Generator_start_ES_voltage_REG := Generator_start_ES_voltage;
25    Generator_stop_ES_voltage_REG := Generator_stop_ES_voltage;
26
27  ELSE
28    DC link voltage reference := DC link voltage reference_REG;
29    ES charge current limit := ES charge current limit_REG ;
30    ES discharge current limit := ES discharge current limit_REG;
31    Enable_generator_control := Enable_generator_control_COIL;
32    Generator_start_ES_voltage := Generator_start_ES_voltage_REG;
33    Generator_stop_ES_voltage := Generator_stop_ES_voltage_REG;
34
35  END_IF
36
37 END PROGRAM

```

KUVA 24. S1000:n output-rekisterien luku-kirjoitus-ohjelma

Toinen logiikkaohjelma esti liian suuren tai pienen arvon syöttämisen etäohjattavaan rekisteriin käyttämällä ylä- ja alarajoja. Ohjelmaan luotiin ehto, joka syötti ylä- tai alarajan arvon muuttujaan, kun ne ylitettiin tai alitettiin (kuva 25). Osa raja-arvoista oli laskennallisia ja ne tuotiin input-rekistereistä ja osa staattisia, joille määritettiin arvo globaalissa muuttujalistassa.

mb_Output_REG_LIMITS

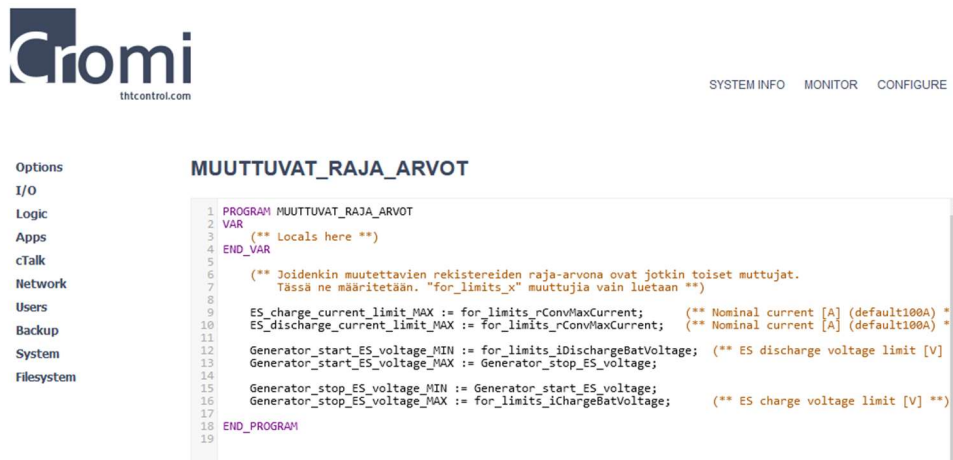
```

1 PROGRAM mb_Output_REG_LIMITS
2 VAR
3   (** Locals here **)
4 END_VAR
5
6   (** REG 0 **)
7
8   IF DC link voltage reference < DC link voltage reference_MIN THEN;
9     DC link voltage reference := DC link voltage reference_MIN;
10  END_IF
11
12  IF DC link voltage reference > DC link voltage reference_MAX THEN;
13    DC link voltage reference := DC link voltage reference_MAX;
14  END_IF
15
16  (** REG 1 **)
17  IF ES charge current limit < ES charge current limit_MIN THEN;
18    ES charge current limit := ES charge current limit_MIN;
19  END_IF
20
21  IF ES charge current limit > ES charge current limit_MAX THEN;
22    ES charge current limit := ES charge current limit_MAX;
23  END_IF
24
25  (** REG 2 **)
26  IF ES discharge current limit < ES discharge current limit_MIN THEN;
27    ES discharge current limit := ES discharge current limit_MIN;
28  END_IF
29
30  IF ES discharge current limit > ES discharge current limit_MAX THEN;
31    ES discharge current limit := ES discharge current limit_MAX;
32  END_IF
33
34  (** REG 3 **)
35  IF Generator_start_ES_voltage < Generator_start_ES_voltage_MIN THEN;
36    Generator_start_ES_voltage := Generator_start_ES_voltage_MIN;
37  END_IF
38  IF Generator_start_ES_voltage > Generator_start_ES_voltage_MAX THEN;
39    Generator_start_ES_voltage := Generator_start_ES_voltage_MAX;
40  END_IF

```

KUVA 25. S1000:n output-rekisterien raja-arvojen ohjelma

Laskennallisia raja-arvoja käytettiin, sillä osassa Beckhoffin etäohjattavien muuttujien ylä- ja alarajoina oli jonkin toisen muuttujan arvo. Näiden muuttujien arvot olivat laskennallisia Beckhoff-järjestelmässä, joten ne tuotiin input-rekistereinä S1000:n logiikkaan. (Kuva 26.)



KUVA 26. Laskennallisten raja-arvojen määrittäsohjelma S1000:ssa

6 YHTEENVETO

Opinnäytetyössä luotiin yhteys kahden eri valmistajan ohjelmoitavien logiikoiden välille käyttäen Modbus TCP:tä. Yhteys mahdollisti etäohjauksen ja -valvonnan ominaisuuksien toteuttamisen valmiiseen automaatiojärjestelmään. Etäohjaus mahdollisti useiden eri asetusarvojen säätämisen järjestelmästä ja etävalvonta mahdollisti haluttujen mittauksien seuraamisen.

Valmista automaatiojärjestelmää ohjasi Beckhoff Automationin CX9020 PLC ja etäohjauksen ja -valvonnan ominaisuudet toteutettiin THT Controlin Cromi S1000 PLC:llä. Beckhoff-järjestelmä ohjelmoitiin käyttäen Beckhoffin omaa TwinCAT 3 -sovellusta ja Cromi S1000 web-pohjaista ohjelmointiympäristöä. Beckhoff-järjestelmään oli valmiiksi asennettuna Modbus TCP -ominaisuudet, sillä valmiissa automaatiojärjestelmässä käytettiin niitä laitteiden väliseen kommunikointiin. Cromi S1000 PLC sisälsi Modbus TCP -ominaisuudet ja ne otettiin käyttöön laitetta ohjelmoidessa.

Opinnäytetyötä tehdessä selvitettiin Modbus TCP:n, TwinCAT 3:n ja Cromi S1000:n ominaisuuksien soveltamisesta käytännössä. Modbus TCP:n ominaisuuksien ja toimintojen osaaminen on hyödyllistä, sillä se on todella laajasti käytetty kommunikointiprotokolla teollisuuden automaatiossa. TwinCAT 3:n käyttäminen syvensi tietämystä PLC:iden ohjelmointiympäristöstä ja niiden ominaisuuksista. Cromi S1000 mahdollisti eri muuttujien ja Modbus-rekistereiden arvojen säätämisen käyttöliittymästä ja niiden vaikutuksen seuraamisen valmiissa automaatiojärjestelmässä.

Työn alkuvaiheessa olisi voinut aikatauluttaa paremmin, jotta olisi ollut selkeämpi järjestys työn eri vaiheita varten. Opinnäytetyön lähtökohtana oli valmiin automaatiojärjestelmän ohjelma tuntemattomassa ohjelmointiympäristössä, josta lähdin opettelemaan Modbus-teoriaa, ohjelmointia ST-kielellä ja samaan aikaan uusien sovelluksien käyttämistä. Tästä seurasi ongelmia ohjelmointiympäristöjen perustoimintoja käyttäessä ja niiden selvittämiseen kului ylimääräistä aikaa.

LÄHTEET

- 1 Modbus Organization 2024. Modbus FAQ. Hakupäivä 21.3.2024. <https://modbus.org/faq.php>.
- 2 Beckhoff Automation 2024. Beckhoff Automation. Hakupäivä 8.4.2024. <https://www.beckhoff.com/fi-fi/company>.
- 3 Beckhoff Automation 2024. TwinCAT automation software. Hakupäivä 28.2.2024. <https://www.beckhoff.com/en-en/products/automation/twincat>.
- 4 Beckhoff Automation 2024. TF6250 TwinCAT 3 Modbus TCP. Hakupäivä 27.3.2024. <https://www.beckhoff.com/en-en/products/automation/twincat/tfxxx-twincat-3-functions/tf6xxx-connectivity/tf6250.html>.
- 5 Beckhoff Automation 2024. CX9020 Basic CPU module. Hakupäivä 27.3.2024. <https://www.beckhoff.com/fi-fi/products/ipc/embedded-pcs/cx9020-arm-cortex-a8/cx9020.html>.
- 6 Real Time Automation 2024. An introduction to Modbus TCP/IP. Hakupäivä 5.3.2024. <https://www.rtautomation.com/technologies/modbus-tcpip>.
- 7 Modbus Organization 2024. Modbus TCP/IP Hakupäivä 27.3.2024. <https://modbus.org/specs.php>.

Veeti Hiukka

Beckhoff import/export/download

Projektin tuonti, tallennus ja lataus ohje

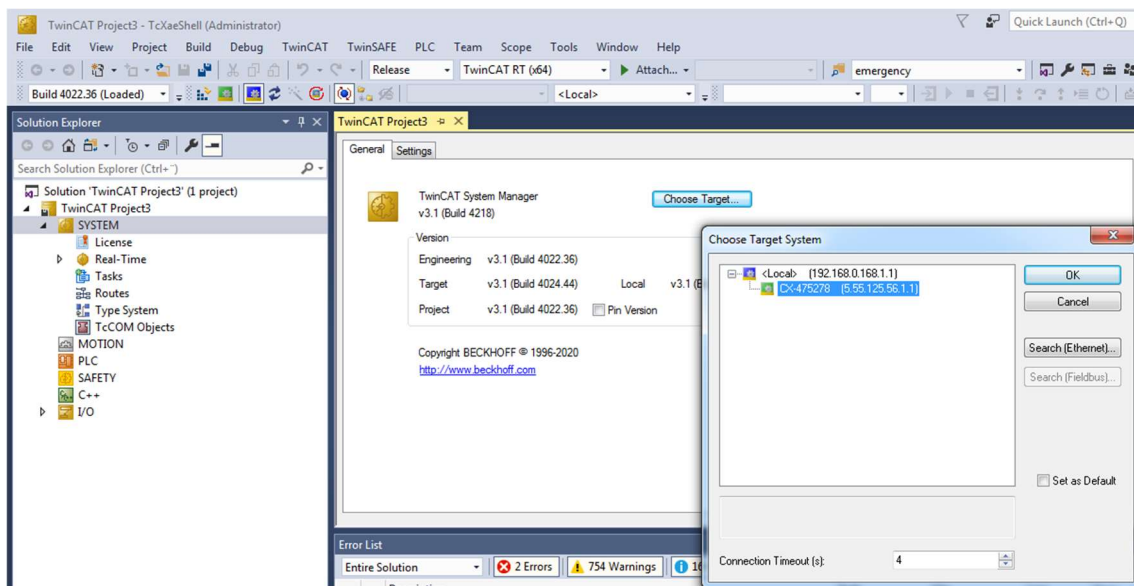
SISÄLLYS

1	PROJEKTIN TUOMINEN PLC:STÄ PC:LLE	3
1.1	PLC:n ja PC:n välisen yhteyden luominen ja PLC:n config-tila.....	3
1.2	PLC:n projektin tuonti PC:lle.....	5
2	PROJEKTIN TALLENNUS ARCHIVEKSI.....	7
3	PROJEKTIN AVAAMINEN ARCHIVE:STÄ.....	8
4	UUDEN PROJEKTIN LUOMINEN JA LATAAMINEN LOGIIKKAAN	10
4.1	Kohteen valinta TwinCAT 3.....	10
4.2	Laitteiden tuominen TwinCAT 3 -projektiin.....	11
4.3	Uuden ohjelman lataaminen logiikkaan.....	14

1 PROJEKTIN TUOMINEN PLC:STÄ PC:LLE

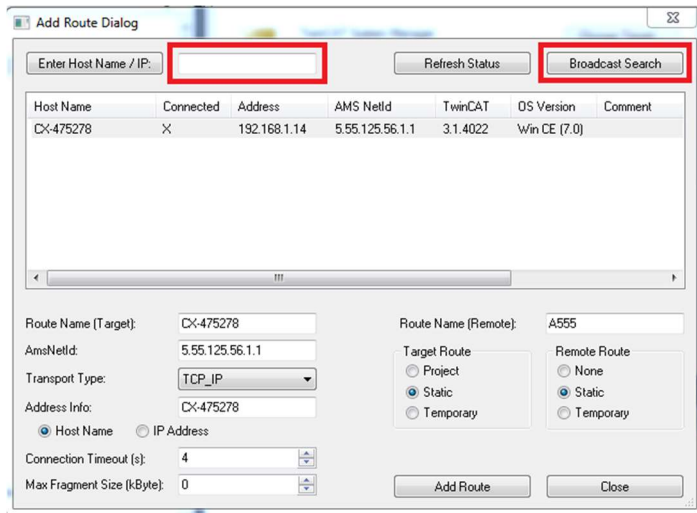
1.1 PLC:n ja PC:n välisen yhteyden luominen ja PLC:n config-tila

PC:n ja PLC:n ollessa samassa aliverkossa, voidaan luoda yhteys niiden välille valitsemalla PLC projektin kohteeksi. Ohjelman ikkunassa *Solution Explorer* kaksoisklikataan kohtaa *SYSTEM*, jonka jälkeen avataan kohteen valintaikkuna *Choose Target...* (kuva 1) ja valitaan PLC.



KUVA 1. Kohteen valinta

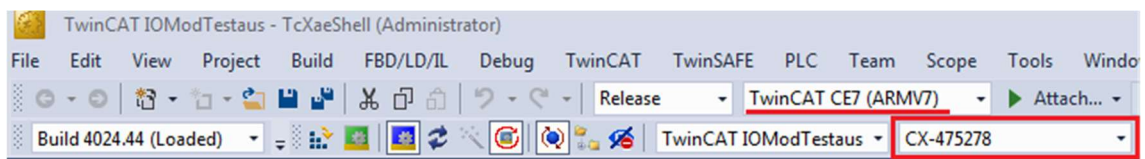
Jos PLC:tä ei löydy *Choose Target System* -valikosta, suoritetaan *Search (Ethernet)*. Seuraavaksi aukeaa valintaikkuna, josta voidaan etsiä PLC:n verkosta. Valitsemalla *Enter Host Name / IP*: voidaan PLC hakea sen nimellä tai IP-osoitteella. Laitteen voi myös hakea luettelosta valitsemalla *Broadcast Search*, joka etsii kaikki laitteet PC:n aliverkosta (kuva 2).



KUVA 2. PLC:n yhteyden luominen

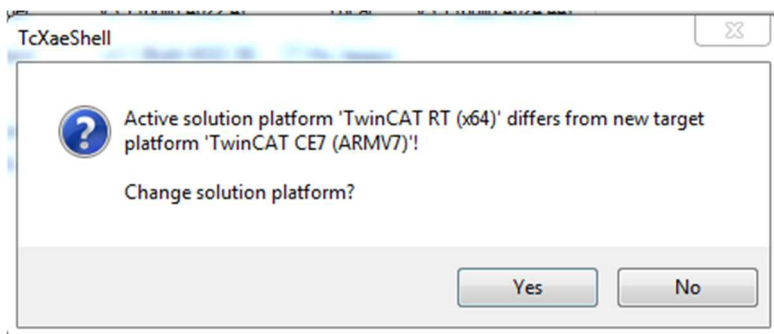
Kun haluttu PLC on löydetty, valitaan se ja suoritetaan *Add route*. Seuraavaksi ohjelma kysyy kirjautumistietoja, jotka ovat oletuksena "Administrator" ja "1". Yhteys on luotu, kun *Connected*-sarakeessa näkyy X.

Lopuksi varmistetaan, että projektin kohteena on haluttu PLC (kuva 3).



KUVA 3. PLC valittu kohteeksi

Ohjelma kysyy alustan vaihtoa, valitaan Yes (kuva 4).

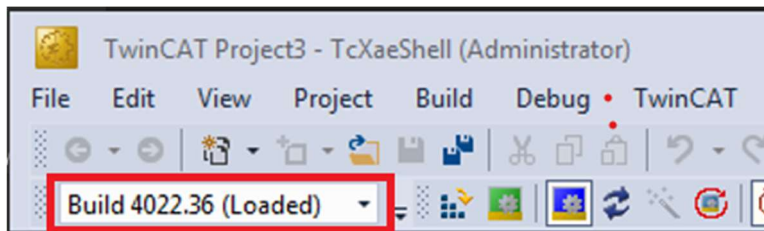


KUVA 4. Alustan vaihtaminen

Nyt voidaan PLC laittaa *config*-tilaan painamalla *config*-painiketta  oikeasta yläreunasta.

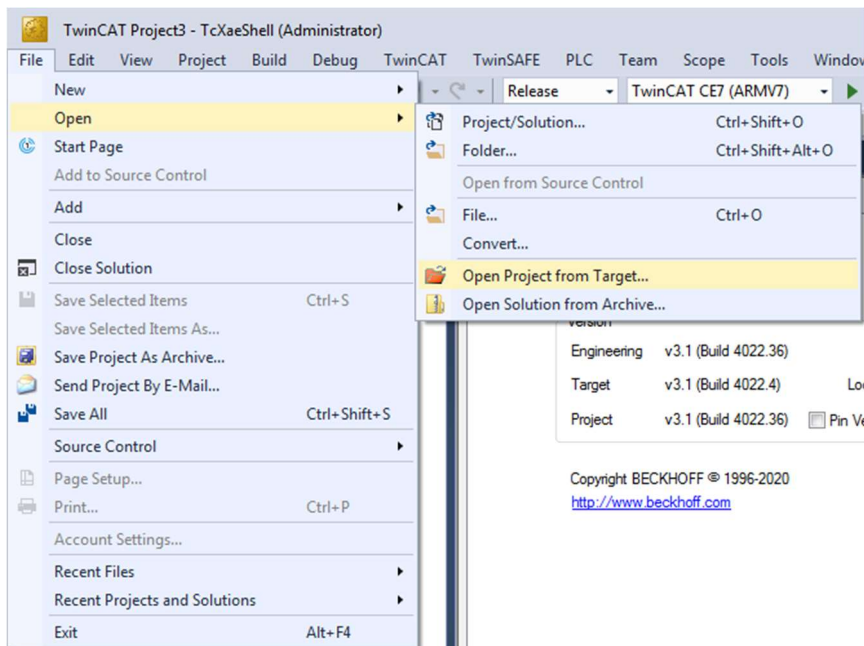
1.2 PLC:n projektin tuonti PC:lle

Kun PLC:stä halutaan tuoda projekti PC:lle, pitää molemmissa olla sama TwinCAT 3 -versio. PLC:n version voi tarkistaa, kun PLC on *config*-tilassa valitsemalla PLC:n sisäisen Windows-käyttöjärjestelmän *start*-menusta *Control Panel*->*CX Configuration* -ikkunassa kohta *TwinCAT*. Esimerkiksi *Versio 3.1, TC Build 4022.4*. PC:llä pitää olla valittuna sama *Build*, kuin PLC:ssä (ainakin alkuosan pitää olla sama, esim 4022.xx, tarkista yhteensopivuus valmistajalta) (kuva 5). PC:n TwinCAT 3 versioita voi vaihtaa esimerkiksi lataamalla Beckhoff Remote Managerin.



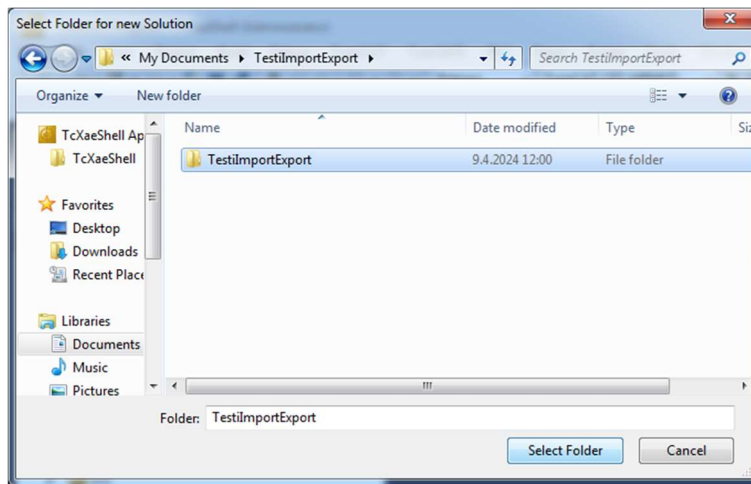
KUVA 5. TwinCAT 3 Build-versio

Kun PLC:n ja PC:n Build-versiot täsmäävät, voidaan projekti tuoda PC:lle. Tämä tapahtuu valitsemalla valikosta *File*->*Open*->*Open Project from Target...* (kuva 6).



KUVA 6. Ohjelman tuonti PLC:stä PC:lle TwinCAT 3:seen

Seuraavaksi ohjelma kysyy mihin kansioon tuotu projekti tallennetaan (kuva 7).

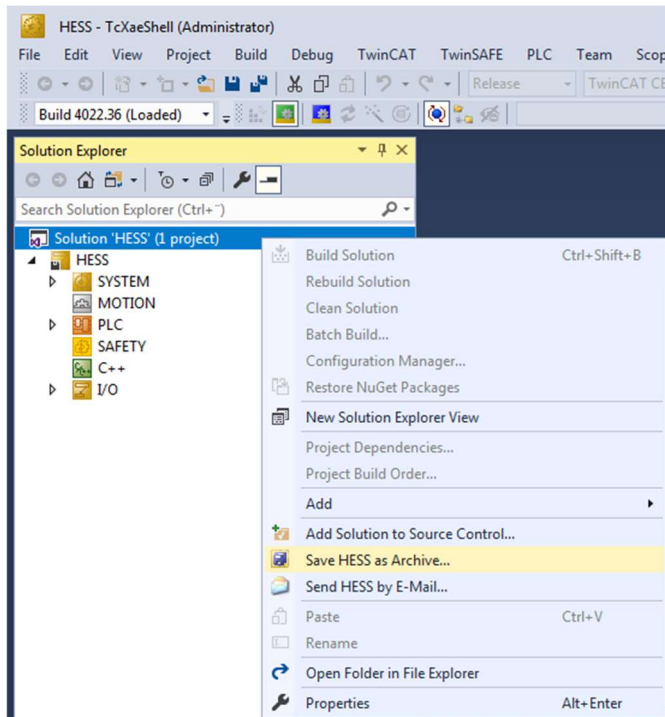


KUVA 7. Kansion valinta, johon tuotu projekti tallennetaan

Nyt PLC:ssä oleva projekti on tuotu PC:lle, sen tallennuspolku on määritetty.

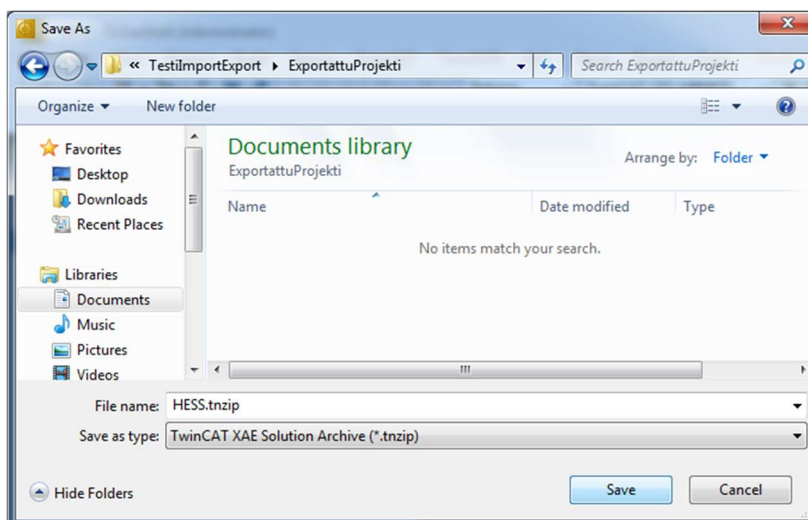
2 PROJEKTIN TALLENNUS ARCHIVEKSI

Solution, joka sisältää kaikki projektit voidaan tallentaa Archiveksi valitsemalla *Solution Explorer* -ikkunasta *Solutions* hiiren oikealla painikkeella ja suorittamalla *Save (projektin nimi) as Archive* (kuva 8).



KUVA 8. Koko projektin tallennus: Save as Archive

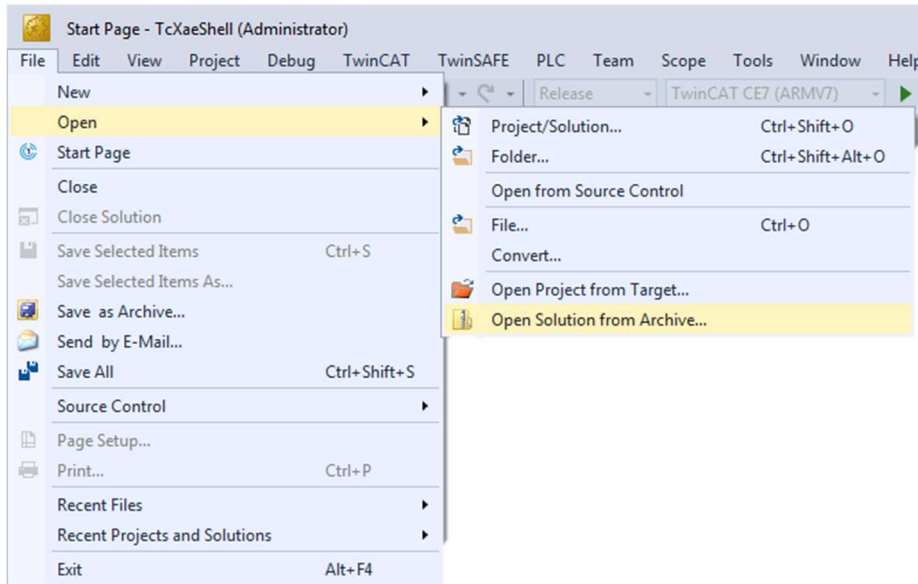
Seuraavaksi ohjelma kysyy, mihin kansioon *Archive* tallennetaan. Kun tallennetaan koko *Solution* sisältäen kaikki projektit, pakatun kansion muoto pitää olla *(projektin nimi).tnzip* (kuva 11).



KUVA 9. Kansion valinta tallentamista varten

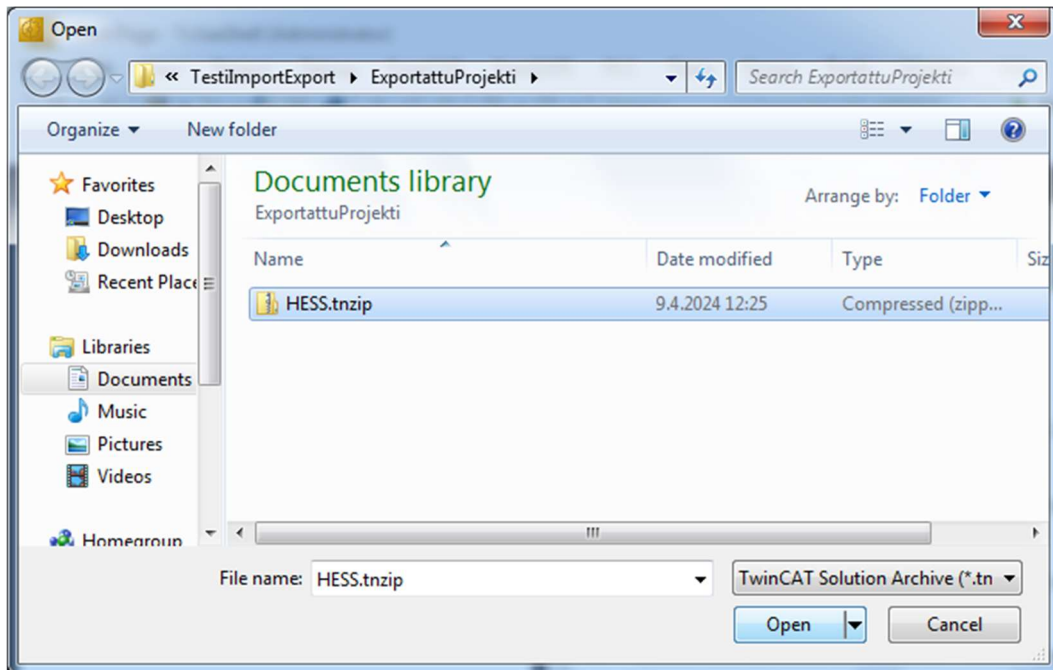
3 PROJEKTIN AVAAMINEN ARCHIVE:STÄ

Solution sisältäen kaikki projektit, voi tuoda pakatusta kansioista (tnzip) valitsemalla *File->Open->Open Solution from Archive...* (kuva 10).



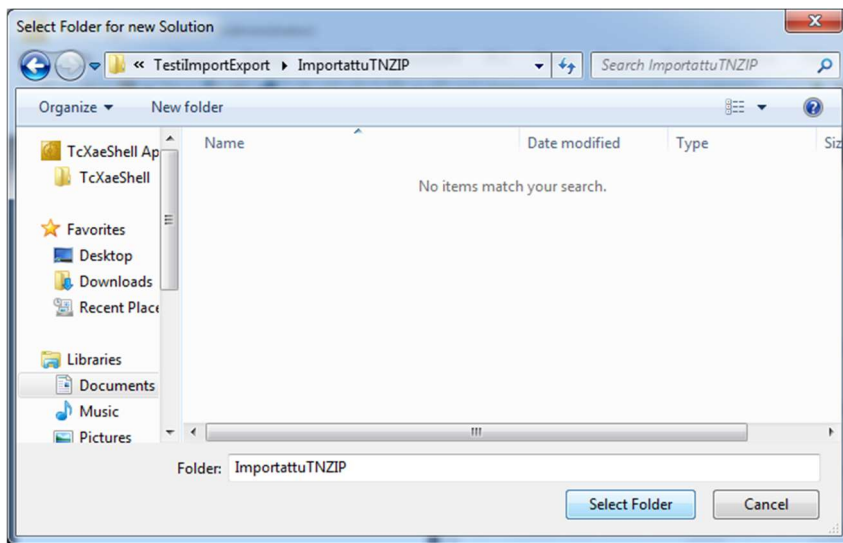
KUVA 10. *Solution* avaaminen tnzip-kansiosta

Seuraavaksi valitaan *Archive*, joka halutaan avata (kuva 11).



KUVA 11. Archiven valinta avaamista varten

Ohjelma kysyy, minne tallennetaan avattu projekti, luodaan uusi kansio (kuva 12).



KUVA 12. Avatun Archiven uusi tallennuspaikka

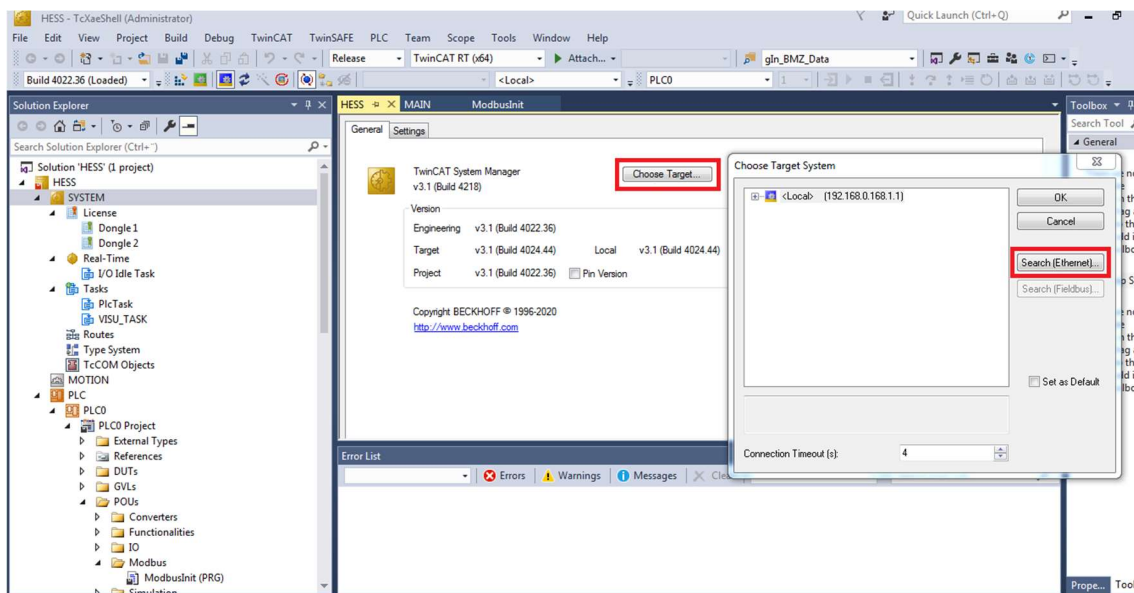
Projekti aukeaa kokonaisuudessaan PC:lle.

4 UUDEN PROJEKTIN LUOMINEN JA LATAAMINEN LOGIIKKAAN

4.1 Kohteen valinta TwinCAT 3

TwinCAT 3 -sovellusta voidaan käyttää paikallisesti ilman PLC:tä mm. simulointitarkoituksiin tai fyysisen PLC:n kanssa. Kun halutaan ohjelma ladata PLC:n sisälle, pitää se valita TwinCAT 3 -projektin kohteeksi.

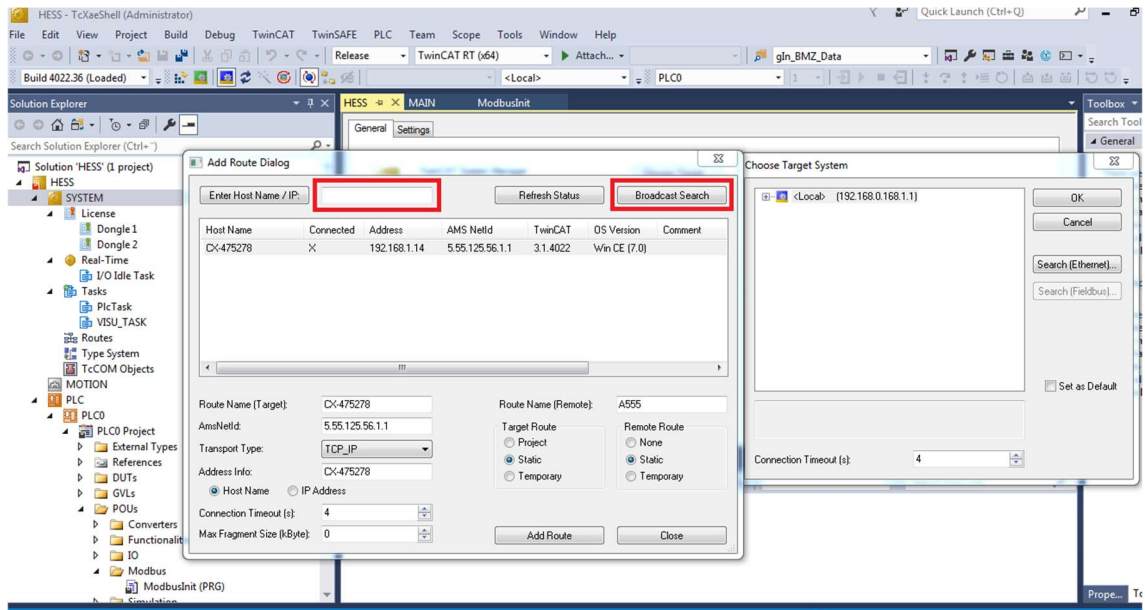
PC:n ja PLC:n ollessa samassa aliverkossa, voidaan luoda yhteys niiden välille valitsemalla PLC projektin kohteeksi. Ohjelman ikkunassa *Solution Explorer* valitaan kohta *SYSTEM*, jonka jälkeen avataan kohteen valintaikkuna *Choose Target...* ja etsitään Ethernet laitteista haluttu PLC painamalla *Search (Ethernet)...*-painiketta (kuva 13).



KUVA 13. TwinCAT 3 kohteen valinta

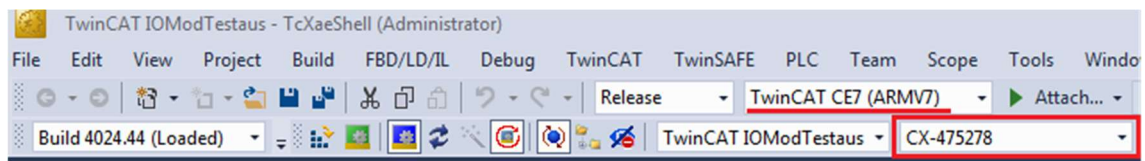
Seuraavaksi aukeaa valintaikkuna, josta voidaan itse PLC löytää. Valitsemalla *Enter Host Name / IP*: voidaan PLC hakea sen nimellä tai IP-osoitteella. Laitteen voi myös hakea luettelosta valitsemalla *Broadcast Search*, joka etsii kaikki laitteet PC:n aliverkosta. (kuva 14).

Kun haluttu PLC on löydetty, valitaan se ja suoritetaan *Add route*. Seuraavaksi ohjelma kysyy kirjautumistietoja, jotka ovat oletuksena "Administrator" ja "1". Yhteys on luotu, kun *Connected*-sarakeessa näkyy X.



KUVA 14. TwinCAT 3 kohteen löytäminen

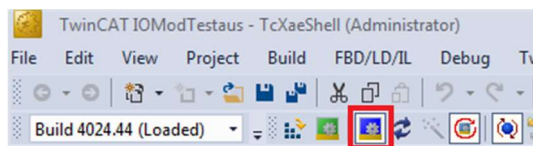
Lopuksi varmistetaan, että projektin kohteena on haluttu PLC (kuva 15).



Kuva 15. TwinCAT 3 kohteen valinta projektiin

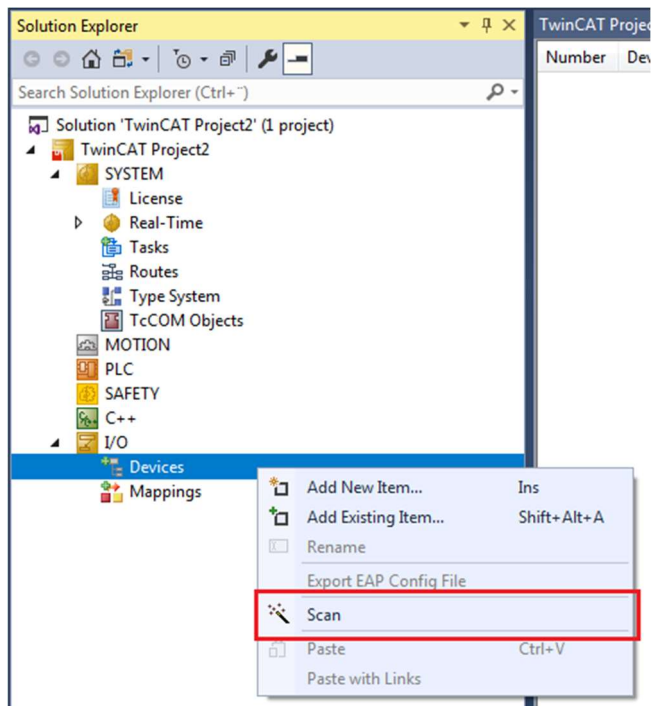
4.2 Laitteiden tuominen TwinCAT 3 -projektiin

Kun projektin kohteena on PLC ja TwinCAT 3 on *config*-tilassa (kuva 16), tuodaan projektiin laitteet.



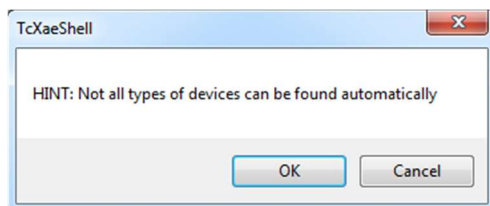
Kuva 16. TwinCAT 3 -ohjelmisto config-tilassa

Tämä tapahtuu valitsemalla *Soulution explorer*-ikkunasta *I/O*-valikko. Valitaan kohta *Devices* hiiren oikealla painikkeella, jonka jälkeen suoritetaan *Scan* aukeavasta valikosta (kuva 17).



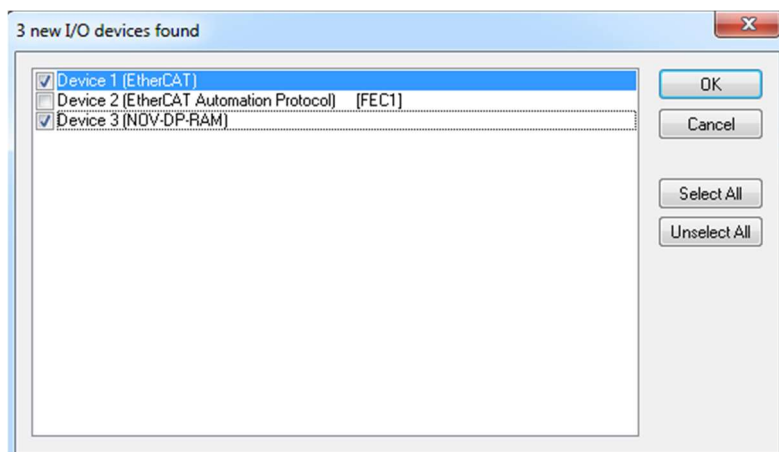
Kuva 17. TwinCAT 3 Laitteiden skannaus

Seuraavaksi TwinCAT 3 ilmoittaa aukeavassa valikossa, että kaikkia laitteita ei voida välttämättä löytää automaattisesti ja valitaan painike OK (kuva 18).



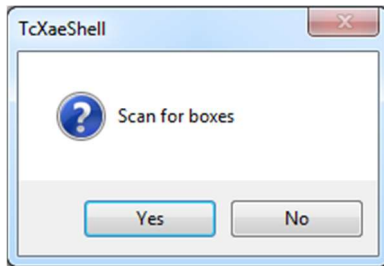
Kuva 18. TwinCAT 3 vihjevalikko

Tämän jälkeen TwinCAT 3 aukaisee ikkunan, jossa näkyy kaikki automaattisesti löydetyt I/O-laitteet (kuva 19). Tässä ikkunassa valitaan halutut laitteet ja hyväksytään OK-painikkeesta.



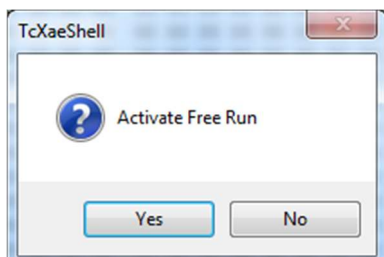
Kuva 19. TwinCAT 3 Löydetyt I/O laitteet -ikkuna

Seuraavaksi TwinCAT 3 kysyy, skannataanko moduuleita (*boxes*) varten (kuva 20). Valitsemalla Yes edetään seuraavaan vaiheeseen.



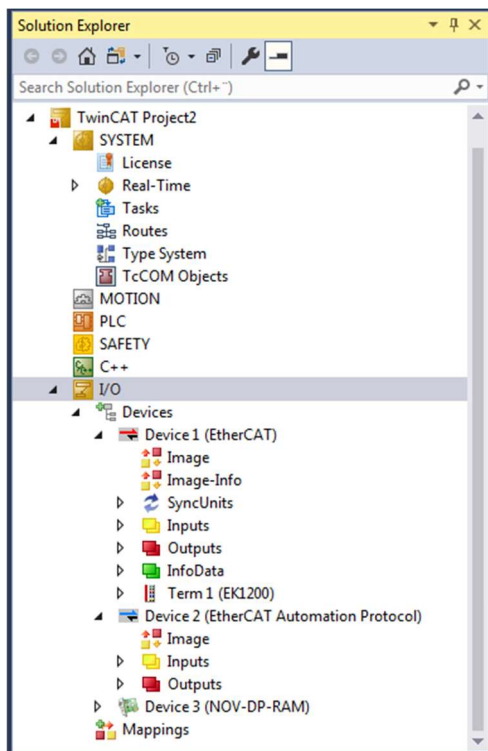
Kuva 20. TwinCAT 3 skannaus moduuleita varten -valikko

Viimeiseksi ohjelma kysyy, laitetaanko *Free Run* päälle (kuva 21), johon valitaan Yes.



Kuva 21. TwinCAT 3 aktivoidaanko Free Run -valikko

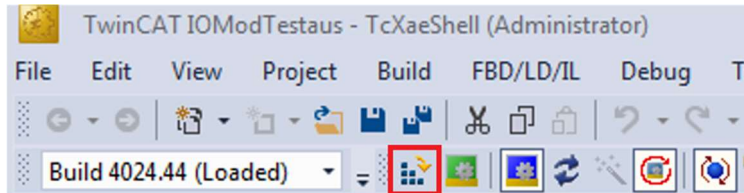
Nyt *Solution Explorer* -ikkunassa näkyy lisätyt laitteet ja niihin voidaan mm. liittää ohjelman tuloja ja lähtöjä (kuva 22).



Kuva 22. TwinCAT 3 *Solution Explorer* -ikkuna projektiin lisätyillä laitteilla

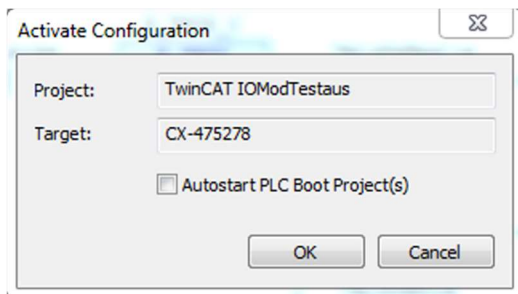
4.3 Uuden ohjelman lataaminen logiikkaan

Tämä korvaa logiikan sisällä olevan ohjelman. TwinCAT 3:n ollessa *config* -tilassa ja projektin kohteena on valittu haluama PLC, voidaan valmis ohjelma ladata PLC:lle. Valitaan yläpalkista *Activate Configuration* -painike (kuva 23).



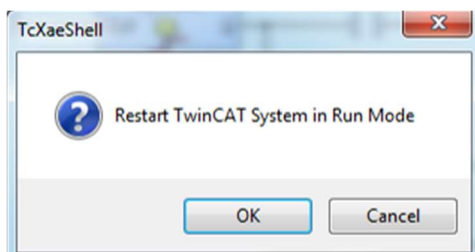
Kuva 23. TwinCAT 3 *Activate Configuration* -painike

Ohjelma avaa valintaikkunan, jossa näkyy projekti ja sen kohde. Aktivoidaan konfiguraatio valitsemalla OK-painike (kuva 24).



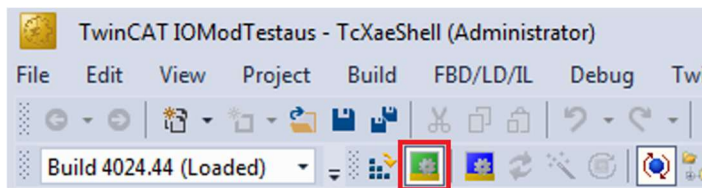
Kuva 24. TwinCAT 3 *Activate Configuration* -valintaikkuna

Seuraavaksi TwinCAT 3 käynnistetään *Run*-tilassa aukeavasta valintaikkunasta painamalla OK (kuva 25).



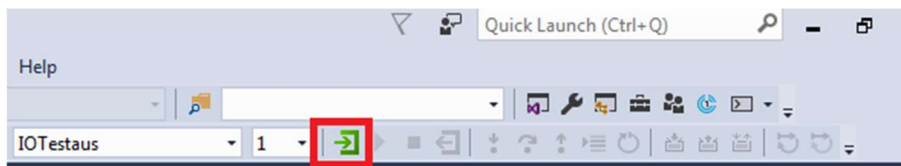
Kuva 25. TwinCAT3 *Run mode* -valintaikkuna

TwinCAT 3 on nyt *run*-tilassa. (kuva 26)



Kuva 26. TwinCAT 3 -ohjelmisto *run*-tilassa

Seuraavaksi kirjaututaan logiikkaan sisään valitsemalla yläpalkin oikeassa reunassa sijaitseva *Login*-painike (kuva 27). Jos ohjelma kysyy salasanaa, se on oletuksena "1".



Kuva 27. TwinCAT 3 *Login*-painike

Seuraavaksi TwinCAT 3 avaa *PLC Control* -valintaikkunan, jossa kysytään portin luomisesta sen puuttuessa, johon valitaan *Yes* (kuva 28).



Kuva 28. TwinCAT 3 *PLC Control* -valintaikkuna

Nyt ohjelma voidaan laittaa käyntiin painamalla *Login*-painikkeen vieressä sijaitsevaa *Start*-painiketta (kuva 29).



Kuva 29. TwinCAT 3 *Start*-painike

Ohjelma on nyt käynnissä logiikassa.