

Martin Krechetov

DESTILLATSIOONISEADME AUTOMAATIKASÜSTEEMI UUENDAMINE VIRU KEEMIA GRUPI NÄITEL

LÕPUTÖÖ

Tehnikainstituut
Robotitehnika õppekava
Juhendaja: Eduard Brindfeld

Tallinn 2024

Autori deklaratsioon ja lihtlitsents

Mina, Martin Krechetov, tõendan, et lõputöö on minu poolt kirjutatud. Töö koostamisel kasutatud teiste autorite, sh juhendaja teostele on viidatud õiguspäraselt.

Kõik isiklikud ja varalised autoriõigused käesoleva lõputöö osas kuuluvad autorile ainuisikuliselt ning need on kaitstud autoriõiguse seadusega.

Juhendaja Eduard Brindfeld.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Martin Krechetov,

sünnikuupäev: 06.07.2000

annan Tallinna Tehnikakõrgkoolile (edaspidi kõrgkool) tasuta loa (lihtlitsentsi) enda loodud teose

DESTILLATSIOONISEADME AUTOMAATIKASÜSTEEMI UUENDAMINE VIRU KEEMIA GRUPI NÄITEL

1. elektroonseks avaldamiseks kõrgkooli repositooriumi kaudu;
2. kui lõputöö avaldamisele on instituudi direktori korraldusega kehtestatud tähtajaline piirang, lõputöö avaldada pärast piirangu lõppemist.

Olen teadlik, et nimetatud õigused jäävad alles ka autorile ja kinnitan, et:

1. lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid ega muid õigusi;
2. PDF-failina esitatud töö vastab täielikult kirjalikult esitatud tööle.

(allkirjastatud digitaalselt)

SISUKORD

LÜHENDID	4
SISSEJUHATUS	5
1 SEADME KIRJELDUS NING PROJEKTI ANALÜÜS	6
1.1 Generaatorõlide destillatsiooniseadme kirjeldus	6
1.2 Automatiseerimise mõju seadme tööle ja keskkonnale	7
1.3 Vana tarkvara kokkuvõte	7
1.4 Tehniline ülesanne uue programmi koostamiseks	8
2 RIIST- JA TARKVARA.....	9
2.1 Riistvara valik.....	9
2.2 Tarkvara	10
2.3 Kasutatud protokollid.....	11
2.3.1 TCP/IP.....	11
2.3.2 Profinet.....	11
2.3.3 Profibus DP	12
2.3.4 Modbus RTU	12
3 PROGRAMM.....	13
3.1 Programmi plokid.....	13
3.1.1 Organization Block	13
3.1.2 Function Block ja Function	13
3.1.3 Data Block	14
3.2 Profibus'ist ja Profinet'ist andmete võtmine	14
3.3 Modbus'ist andmete võtmine	16
3.4 Analoogsignaali skaleerimine	18
3.5 PID regulaatorid.....	19
3.6 Signalisatsioonid ja blokeering	22
3.7 Summeerivad loendurid	24
3.8 Proovivõtmise kraanid	25
3.9 Andmed teistest kontrollieritest	26
4 SCADA SÜSTEEM	28
KOKKUVÕTE	30
SUMMARY.....	31
VIIDATUD ALLIKAD.....	32

LÜHENDID

PLC - Programmable Logic Controller

VKG - Viru Keemia Grupp

TIA Portal - Totally Integrated Automation Portal

I/O - Input/Output

SCADA - Supervisory Control and Data Acquisition

RTU - Remote Terminal Unit

DP - Decentralized Periphery

TCP - Transmission Control Protocol

IP - Internet Protocol

SCL - Structured Control Language

RS-triger - Reset Set triger

SISSEJUHATUS

Käesoleva lõputöö põhiülesandeks on Viru Keemia Grupp kontserni territooriumil asuva generaatorõlide destillatsiooniseadme destillatsioonikolonna kontrolleri uuendamine.

Praegu seadmes on 5 kontrolleri, millest 2 tükki on juba vahetatud uute vastu. Neid lisati seadme uuendamise protsessis, kuna mõnikord ei olnud vanal kontrolleriil piisavalt ressursi või lisati neid lihtsalt selleks, et parandusi teha ilma seadet seiskamata. Käesoleva lõputöös käsitlev kontrolleri juhib fraktsioonide jahutamise protsessi.

Otsus kontrolleri vahetama oli põhjustatud sellega, et vana süsteem ei vasta enam tänapäevastele standarditele nii riistvara, kui ka tarkvara suhtes. Praegu on paigaldatud seadmes Siemens Simatic S7-300 sarja PLC-d, mis töötab seal tõrkumatult aastast 2008, aga vanuse tõttu ilmuvad probleemid nagu moodulitega sidet kadumine või iseeneslik välja lülitumine. Lisaks sellele, on plaan tulevikus terve seade modifitseerida, mis toob kaasa uue PLC paigalduse, millel on rohkem arvutusjõudlust ja laiendatud funktsioone.

Püstitatud probleemide lahendamiseks oli valitud uus Siemens Simatic S7-1500 sarja kontrolleri ning TIA Portal 18 arenduskeskkond. Kontrolleri moodulkonstruktsioon annab võimalust tulevikus süsteemi veel laiendada ja parandada. Töömahu vähendamiseks ja aja säästmiseks oli tehtud otsus olemasolevaid I/O mooduleid säilitada, kuna nende tehniline seisukord lubab neid veel ilma probleemideta kasutada. Lisaks sellele, uuendatakse ka sideprotokolle kontrolleri ja sagedusmuundurite vahel. Praegu 31-st sagedusmuundurist 11 kasutavad analoog signaali (4-20) mA sagedusmuundurite töö kontrollimiseks ja Profinet protokolliga andmete vastuvõtmiseks, 11 sagedusmuundurit kasutavad oma töös Profibus protokolliga, ülejäänud 9 muundurit kasutavad vaid analoogsignaali nii seiramiseks, kui ka juhtimiseks. Uues süsteemis planeeritakse analoog päringutest üldse loobuda ning integreeruda Profinet protokolliga andmete vastuvõtmiseks terve võrgu vältel, analoog signaal aga jääb muundurite kontrollimiseks. Jäevad ka 11 Profibus'i kasutavad sagedusmuundurit, kuna nende tarkvara Profinet'i praegu ei toeta.

Tulevikus tahetakse terve sidet ehitada Profineti baasil, nii andmete seire, kui ka kontroll, aga see teema ei ole selle lõputöö raames püstitatud. Veel üks tähtis protokoll, millega tuleb selle lõputöö käigus kokku puutuda, on ModBus RTU, mida kasutatakse voolumõõturite käsitlemises. Kasutatav PLC käsitleb 9 voolumõõtureid ModBus'i kaudu, ülejäänud näited võtab SCADA teistest kontrolleritest kas läbi TCP/IP protokolliga PUT/GET käsku kasutades, või analoog signaali kaudu.

1 SEADME KIRJELDUS NING PROJEKTI ANALÜÜS

1.1 Generaatorõlide destillatsiooniseadme kirjeldus

Viru Keemia Grupi generaatorõlide destillatsiooniseadme otstarbeks on puhastatud põlevkivi vaikude kihindumine. Fraktsioneeriva destillatsiooni käigus saadakse kätte järgmised vedelikud: bensiini fraktsioon, diisli fraktsioon, kerge masuut, raske masuut ning destillatsioonijääk. Seadme põhitoodanguks on puidu immutusõli, põlevkiviõli "VKG C", põlevkivi bituumen PB-2 ja põlevkivi bensiin. Lisaks sellele, toodab destillatsiooniseade veel vahepealsed keemilised toorkomponendid, mille kasutatakse teistes VKG seadmetes, näiteks põlevkiviõli komponendid "VKG sweet", "VKG D" ja "VKG D-1".

Destillatsioonikolonniide automatiseerimistööd olid teostatud 2008. aastal Viru Keemia Grupi tütarettevõtte Viru RMT poolt. Praegu seadmes kasutuses 3 kolonni, selle töö raames käsitletakse aga vaid 2 neist, K-1 ja K-2. Dehüdratatsiooni kolonn K-1 ettevalmistub (eemaldab vett ja vöörlisandeid) tooraine järgmiseks sammuks, mille käigus toimub jäme destillatsioon teises kolonnis.

K-2 koosneb kolmest sektsioonidest:

- auru sektsioon
- pesu sektsioon
- taldrikute sektsioon

Kolonniide mõõtmed on järgnevad:

	K-1	K-2
Kõrgus	21,85 m	33,20 m
Diameeter	2,2 m	1,0 - 2,5 m ¹
Maht	83000 l	127000 l

¹ Kolonni diameeter muutub astmeliselt

1.2 Automatiseerimise mõju seadme tööle ja keskkonnale

Enne seda, kui automatiseerimise protsess algas, juhtisid tootmisprotsessi operaatorid käsitsi. Protsess oli imelihtne, aga nõudis palju inimressursi ja tähelepanekut. Kuna operaatorid ei suutnud piisavalt keskenduda, juhtus nii, et andurid ja klapid jäid järelvalveta, see põhjustas avariisid ja/või ressursside raiskamist. Näiteks järelvalveta jäänud ahi (destillatsiooni eelsamm) põhjustab ülekuumenemist ja liigse kütte- ja maagaasi ületarbimist, mis omakorda põhjustab õhu saastumist. Lisaks sellele, "kõrgete pööretega" töö tekitab seadmete kiirkulumist, mis veel vähendab tootmise produktiivsust.

Automatiseerimine lahendab terve hulk probleeme, PID regulaatorid tagavad klappide ja pumpade optimaalset tööd, terve seadme seire ja kontroll toimub vaid ühel kuvaril, mis lubab operaatoritele tööst kõrvale mitte kalduda. Efektivsem töö tähendab väiksemat mõju loodusele ja turvalisemat töökeskkonda.

1.3 Vana tarkvara kokkuvõte

Olemasolu programm on tehtud juba vananenud Simatic Step 7 V5 keskkonnas, mille esimene versioon on päevavalgust näinud 30 aastat tagasi. Tänapäeval ei sobi see tehniliste protsesside juhtimiseks oma ebatäielikkuse ja vananenud funktsionaalsuse tõttu.

Lõputöö raames tehtud vana programmi analüüs näitab, et eksploatatsiooni ajal tehtud muudatused ja parandused moodustaksid liiga keerulise ning ebaselge koodi. Selle põhjuseks võib ka olla see fakt, et projektiga tegelesid erinevad programmeerijad, kes teevad tööd igaüks omamoodi ja tihti nende koodi kirjutamisstiilid ei klapi, mis omakorda tekitab segadust. Lisaks sellele, et programm on kirjutatud osaliselt vene keeles ladina kirja transliteratsioonis ja osaliselt inglise keeles, mis tekitab raskusi kommentaaridest arusaamisega.

1.4 Tehniline ülesanne uue programmi koostamiseks

Eelmise programmi puudutusega arvestades oli otsustatud maksimaalselt selget ja arusaadavat koodi kirjutada. Sealhulgas:

- kood plokkide sorteerimine kaustadesse, näiteks Profinetiga seotud kood tuleb "Profinet" kausta, PID regulaatoritega seotud kood tuleb "PID" kausta jne
- vene keele vältimine koodi seletustes, programmi kirjutamise standardiks on inglise keel
- võimalusel ainult sisseehitatud Siemens teekide kasutamine, et tulevikus ei ilmuks mittesobivuse probleeme

2 RIIST- JA TARKVARA

2.1 Riistvara valik

Uue süsteemi projekteerimisel olid püstitatud järgmised probleemid:

- Uue ja vana riistvara kokkusobivus
- Uue süsteemi vastupidavus ja vajadusel hooldatavus
- Süsteemi modernisatsiooni võimalus tulevikus

Neid probleeme arvesse võttes olin otsustanud kasutada Siemens S7-1500 sarja kontrollereid, täpsemalt Siemens CPU 1516F-3 PN/DP 6ES7516-3FN02-0AB0 mudelit. Otsustav argument oli see, et neid kontrollereid kasutatakse laialdaselt erinevatel VKG seadmetel. Selle PLC modulaarsus annab palju erinevaid võimalusi moodulite kombineerimiseks, mis on eriti tähtis tehase keskkonnas. See kombineerimis võimalus lubab näiteks kasutada olemasolevaid I/O mooduleid, mis ei ole võimalik teiste tootjate kontrolleritega.

Veel üks probleem, mis ilmus projekti raames, on vanad I/O moodulid. Vaatamata sellele, et nende seisukord on hea, siis nende vanus tekitab teatud probleeme, näiteks: aegunud pistikuühenduste tõttu ei saa neid otse kontrolleri ühendada. Selleks on vaja kasutada sidemoodulit, Siemens 6ES7 153-4BA00-0XB0, mida kasutatakse selle töö käigus.

Modbus RTU tööd tagab Siemens CM PtP RS422/485 HF mooduli seeria numbriga 6ES7 541-1AB00-0AB0. See on RS-422/485 liidest kasutatav sidemoodul, mida saab otse kontrolleri kokku liita oma konstruktsiooni tõttu. Modbus RTU'd kasutatakse seadistatud sidemoodul pooldupleks RS-485 režiimi TIA Portali kaudu.

Tervikuna selles projektis kasutatakse alljärgneva riistvara:

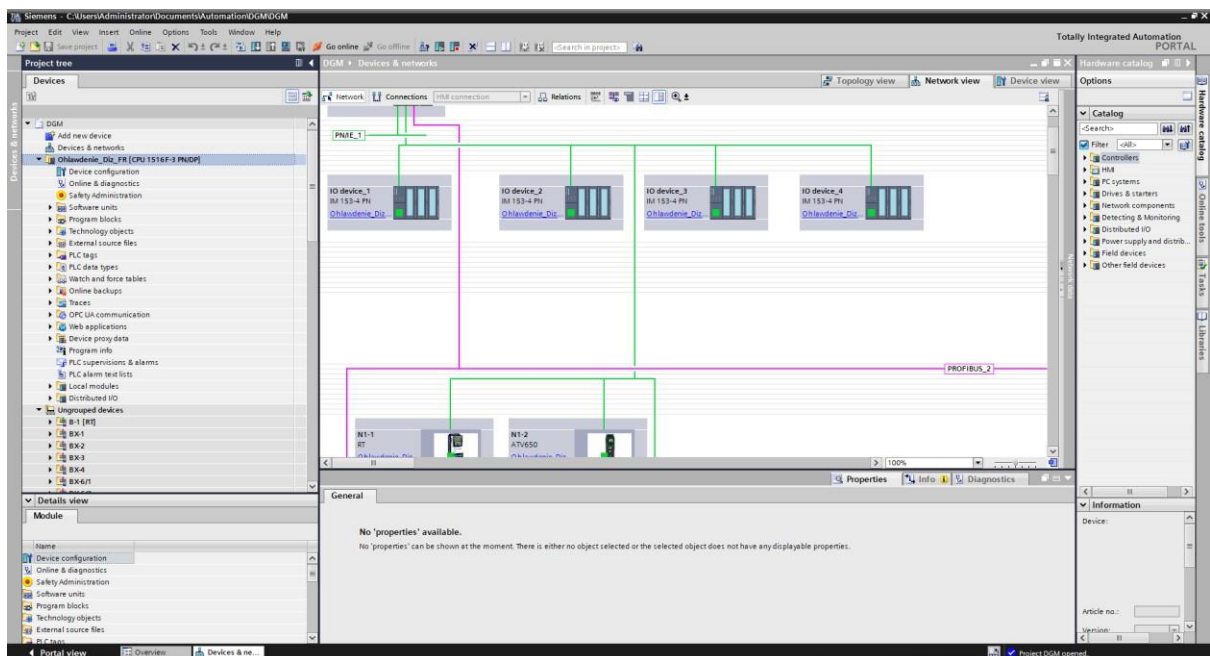
- S7-1516 PLC 6ES7 516-3FN02-0AB0 - 1 tükk;
- CM PtP RS422/485 HF kommunikatsiooni moodul 6ES7 541-1AB00-0AB0 - 1 tükk;
- AI 8xU/I/RTD/TC ST analoogsisendi moodul 6ES7 531-7KF00-0AB0 - 5 tükki;
- AQ 8xU/I HS analoogväljundi moodul 6ES7 532-5HF00-0AB0 - 2 tükki;
- DI 32x24VDC HF digitaalsisendi moodul 6ES7 521-1BL00-0AB0 - 1 tükk;
- DQ 32x24VDC/0.5A HF digitaalväljundi moodul 6ES7 522-1BL01-0AB0 - 1 tükk;
- IM 153-4 PN HF kommunikatsiooni moodul 6ES7 153-4BA00-0XB0 - 4 tükki;
- vanad S7-300 analoog- ja digitaalsisendi/väljundi moodulid, kokku 31 tükki.

Kokku selle kontrolleri sisendite/väljundite arv on:

- Analoogsed sisendid - 224 tk
- Analoogsed väljundid - 56 tk
- Digitaalsed sisendid - 96 tk
- Digitaalsed väljundid - 64 tk

2.2 Tarkvara

Projekti loomiseks oli valitud Simatic TIA Portal'i v.18 keskkond. See pole kõige uuem versioon, aga piisavalt uus, et tagada stabiilset tööd ja kõike vajalikke funktsioone.



Joonis 1. TIA Portal 18, Topograafia ekraan

TIA Portal'i eeliseks on selle tööriista universaalsus ja arusaadav graafiline kasutajaliides, mis teeb töö lihtsamaks. Siemens'i platvorm võimaldab laia diagnostika ja analüüsi võimalusi. See aitab, kui on vaja töötavas seadmes kiiresti parandusi või muudatusi teha. Näiteks "Monitoring" funktsioon annab võimalus enda projekti "live-režiimis" näha, kuidas erinevad koodi lõigud koos töötavad, mis üldiselt programmis toimub ja mis tuleb veel korda teha.

2.3 Kasutatud protokollid

Käesolevas peatükis antakse ülevaade kasutatud side protokollidest ja nende kasutusalaadest selles projektis.

2.3.1 TCP/IP

TCP ja IP on kaks erinevad protokollid, mis töötavad koos.

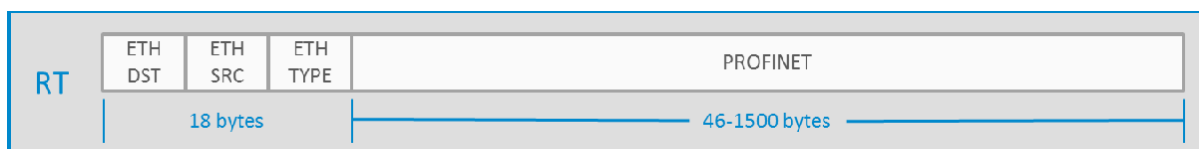
TCP, ehk Transmission Control Protocol, on side standard, mis lubab andmete vahetus erinevate kasutajate vahel. Enne andmete vahetust, tõendab TCP, et saatja ja saaja vahel on ühendus. Pärast seda jagab ta andmete hulga väiksematele tükidele (pakketidele) ja saadab neid võrgukihi protokolliga, tavaliselt selleks kasutatakse internetiprotokolli (IP). [1] [2]

IP, ehk Internet Protocol, on andmete saatmis meetod, mille abil saab andmeid saata üle võrgu. See vastutab andmete suunamise eest, ehk see määrab kuhu tuleb TCP poolt ettevalmistatud pakette saata, et nad saajani jõuaksid. Selleks annab IP kasutajatele IP-aadressi, mis omakorda võimaldab kliendi võrgus üksteist leida ja andmetevahetust alustada. [1] [3]

Seadme võrgus kasutatakse TCP/IP protokollide andmetevahetuseks kontrollerite vahel, samuti SCADA arvutite ühenduseks.

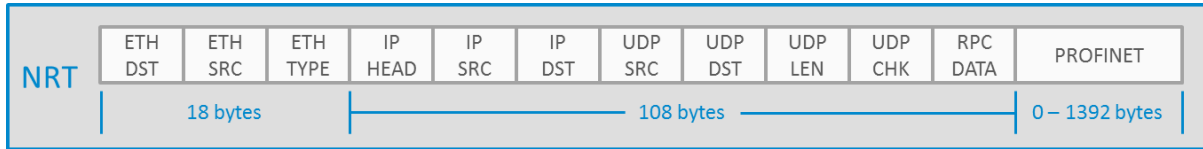
2.3.2 Profinet

Saab öelda, et Profinet on modifitseeritud Ethernet, millega iga tänapäevane inimene on hästi tuttav. Erinevuseks on see, et Profinet standard oli mõeldud tööstuse tarbeks, seega andmete vahetamise kiirus ja tõrkekindlus on prioriteediks. Selleks kasutatakse Profinet'i standardis nn "Real-Time Channel", mis põhimõtteliselt jäätakse kõik teised protokollid vahele ja saadab andmed otse saajale MAC aadressi kaudu. Sellega, aga, ilmub selline probleem, et ei saa andmeid teistele võrkudele saata, kuna protokollide vahele jäämisega kaotatakse ka üks tähtis eelis - IP-aadress. [4] [5]



Joonis 2. Profinet'i frame Real-time režiimis

Juhul, kui on ikkagi vaja teisest võrgust võrku pääseda, tuleb appi "Non Real-Time Channel". Selles režiimis saab kasutada kõiki OSI mudeli kihte, mis annab tavalise Etherneti võimalused ja eelised. Aga töötavad need võrgud aeglaselt, mis mõnikord ei sobi tööstusekeskkonnas kasutamiseks. [4]



Joonis 3. Profinet'i frame Non Real-time režiimis

Selles projektis kasutatakse Profinet'i võrku sagedusmuunduritega ja signaali moodulitega ühendamiseks.

2.3.3 Profibus DP

Profibus DP on sidestandard, mis oma töös kasutab RS-485 liidet. See on tänaseks üks kõige levinumaid sidestandardeid tööstusautomaatikas. Eeliseks on stabiilsus, suur andmete vahetuse kiirus ning süsteemi universaalsus. "DP" tähendab "Decentralized Periphery", mis on topoloogia raames tähendab, et kasutamiseks pole vaja midagi lisada, võrgus vaid kontrollid ja seadmed. Profibus kasutab Master/Slave tehnoloogiat, Master on võrgu keskpunkt, ehk kontrollid, ja Slave'id on juhitavad seadmed, näiteks andurid või elektriajamid. See on "lisa ja kasuta" standard, mis kasutab eelseadistatud failid seadmete programmi lisamiseks. See teeb programmi koostamise lihtsamaks, kuna programmeerija paneb seadmete aadressid programmi, kõik muud konfiguratsioonid on juba tehtud seadme tarnija poolt. [6] [7]

Selle projekti raames kasutatakse Profibus DP sagedusmuunduritega ühendamiseks.

2.3.4 Modbus RTU

Modbus RTU on järjestikliides kasutatav sidestandard, mis kasutab Master/Slave arhitektuuri. Seda liidetakse RS-232 või RS-485 pooldupleks standardi kaudu. Seadmetel on oma aadress 1-247 diapsoonis, 248-255 aadressid on reserveeritud süsteemi sisekasutamiseks. Andmete registritel on oma tootja poolt määratud aadressid, millele viitab Master. Nende aadresside loetelu tavaliselt võetakse seadme kasutusjuhendist. Kõik võrgus olevad seadmed peavad olla ühtemoodi seadistatud, need seadistused on: Bitrate (kiirus), Data Bit (bittide arv), Stop Bit (lõppbitt) ja Parity (paarsus). [8]

Käesolevas süsteemis kasutatakse Modbus'i voolumõõturitega ühendamiseks.

3 PROGRAMM

3.1 Programmi plokid

TIA Portal pakub enda kasutajatele 4 programmi plokkide liike. Igaühel plokil on oma ülesanne programmi sooritamise ajal ja reeglina ei saa neid omavahel vahetada. See tähendab, et programmeerija peab sujuvalt sotti saama mis ta teeb ja miks ta seda teeb, kuna igal plokil on oma omadused ja spetsiifika. [9] Plokkide liigid on järgmised:

- Organization Block (OB) - Organisatsiooni plokk
- Function Block (FB) - Funktsiooni plokk
- Function (FC) - Funktsioon
- Data Block (DB) - Andmete plokk

3.1.1 Organization Block

Organization Block on baas programmi plokk, mis ühendub kõik teised võrgud ja komponendid (sh plokid) kokku. Kõige levinumaid OB liigid on Main, Cyclic interrupt ja Startup. [9] [10]

Main (OB1) on põhiplokk, mis rakendab kõik teised funktsioonid, mis sinna on pandud. Main plokk on loodud vaikimisi ja peab igal programmil olemas olema kui esmane käivitus plokk. [9] [10]

Cyclic interrupt (OB30 kuni OB38) plokk töötab nagu Main, aga selle erinevusega, et kasutaja saab ise valida, mis ajavahemikuga plokk läheb tööle. See on eriti kasulik loendurite või teiste funktsioonide tegemiseks. [9] [10]

Startup (OB100) läheb tööle vaid üks kord kontrolleri stardil, programmi käigus ta enam ei tööta. [9]

3.1.2 Function Block ja Function

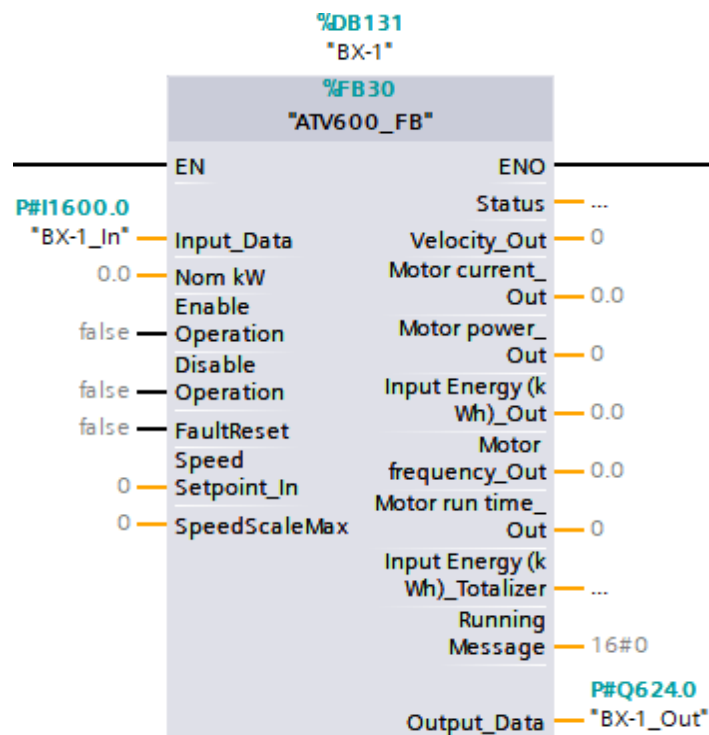
Function Block või Function kasutatakse signaalidega või teiste andmetega operatsioonidele. Funktsiooni ploki eeliseks funktsiooniga (FC) on see, et funktsiooni plokk omab oma andmete plokki, kus hoiakse kõik selle funktsiooniga seotud muutujaid. Selle abil saab näiteks vaikimisi anda väärtused programmile, et see tööle hakkaks. See teeb funktsiooni ploki funktsiooniga võrreldes suureks. See tähendab, et FB vajab rohkem mälu ja arvutusjõudlust. [12] [13]

3.1.3 Data Block

Data Block ehk andmete plokk salvestab ettenähtud andmed ja muutujad tabelivormis. Alternatiiviks on andmete salvestamine teistes plokkides, näiteks saab andmeid sisestada otse organisatsiooni plokki, aga puuduseks on see, et neid andmeid saab kasutada vaid selle ploki sees. Kui on vaja kasutada samu andmeid mitme ploki vältel, siis tuleb luua eraldi andmete plokkid. [11]

3.2 Profibus'ist ja Profinet'ist andmete võtmine

Protsessi lihtsustamiseks oli otsustatud eraldi teha funktsioon Profibusiga ühendamiseks. See funktsioon võtab sisendsignaali sagedusmuunduritest ja tagastab ettevalmistatud signaali SCADA süsteemile.



Joonis 4. Plokk andmete väljavõtmist Profibus'ist

Sisend BX-1_In tuleb sagedusmuundurilt telegrammi kaudu sisendaadressse kasutades (vt Joonis 5). Ploki sees teostatakse toiminguid signaaliga, konverteeritatakse andmetüübid, arvestatakse mõõtühikuid valemite järgi ning toimub andmete salvestamine. Selles plokkis näiteks normaliseeritakse ja skaleeritakse andmeid.

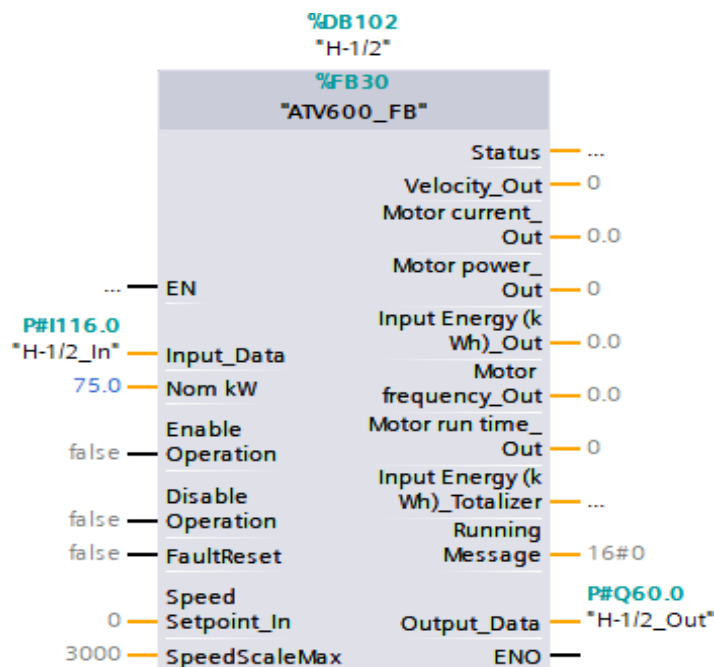
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision	Comment
1	▼ BX-1_In	"ATV650_Telegram106_In"	%I1600.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	PKW1	Word	%IW1600		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	PKW2	Word	%IW1602		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	PKW3	Word	%IW1604		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	PKW4	Word	%IW1606		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Status	Word	%IW1608		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3201
7	Velocity	Int	%IW1610		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		8604
8	Motor current	UInt	%IW1612		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3204
9	Motor power	Int	%IW1614		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3211
10	Motor frequency	Int	%IW1616		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3202
11	Motor run time	UDInt	%ID1618		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		3244
12	OMAB	Word	%IW1622		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Joonis 5. Tüüpiline Profibus telegrami sisu.

"Comment" reas saab näha mõningaid arve. Need on sagedusmuunduri sisesed aadressid, mida kasutatakse päringute saatmiseks. Näiteks siin, register numbriga 3201 vastab päringutele sagedusmuunduri staatusega, programm loeb seda ja kasutab.

Väljundisse saadakse sama telegramm, muudetud andmetega, mida kasutatakse edasi programmi käigus, näiteks siin BX-1_Out läheb edasi SCADA süsteemi, kuna mootoreid Profibus'i kaudu ei kontrollita. Juhul, kui ajameid kontrollitakse sama protokolliga kaudu, siis antakse need andmed tagasi sagedusmuundurile.

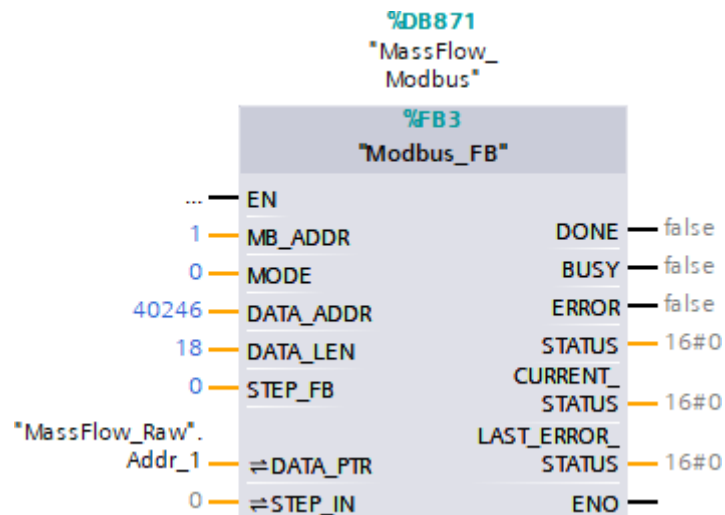
Selles projektis kasutatakse Profineti just samamoodi, erinevate standardite pärast tehakse koodis teatud muudatused, kuid need erinevused on ebaolulised.



Joonis 6. Profinet'i plokk

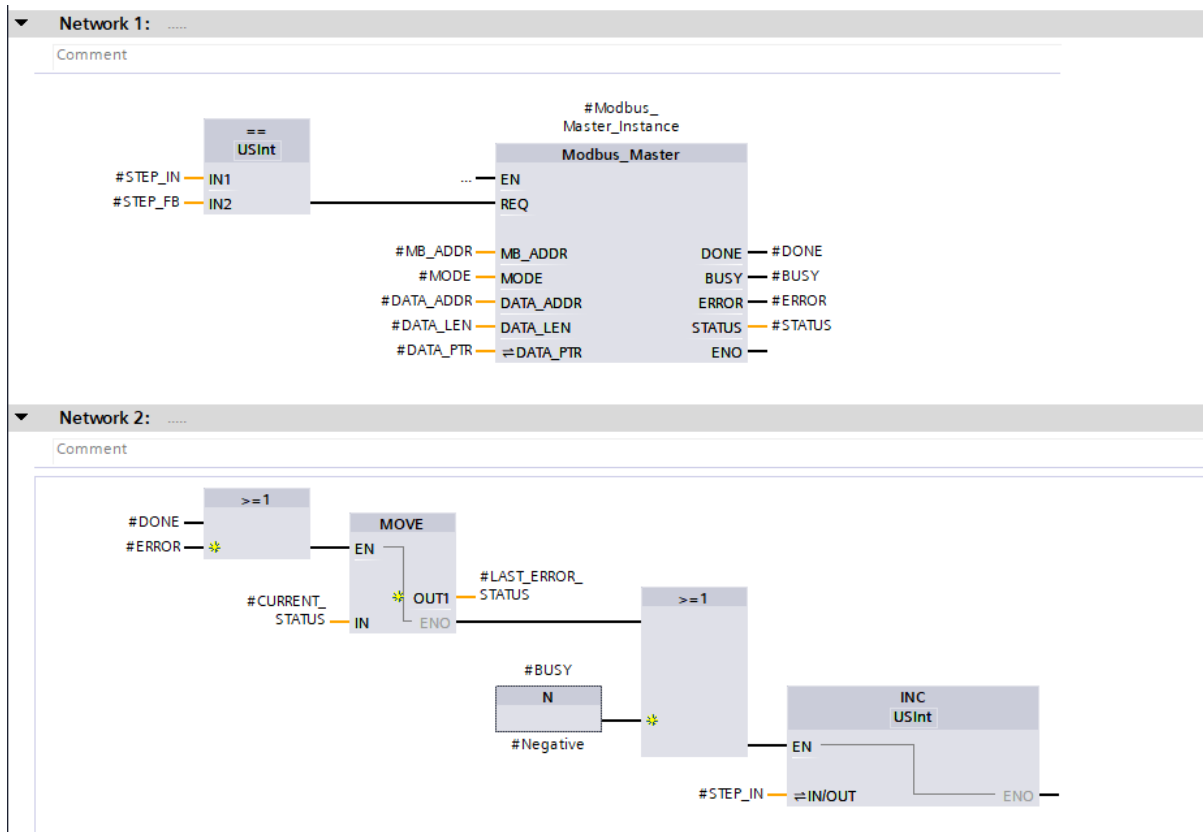
3.3 Modbus'ist andmete võtmine

Voolumõõtja näite vastuvõtmiseks kasutatakse Modbus RTU protokoll. Kõik selleks vajalikud sammud protsessi lihtsustamiseks olid ühendatud ühte funktsiooni.



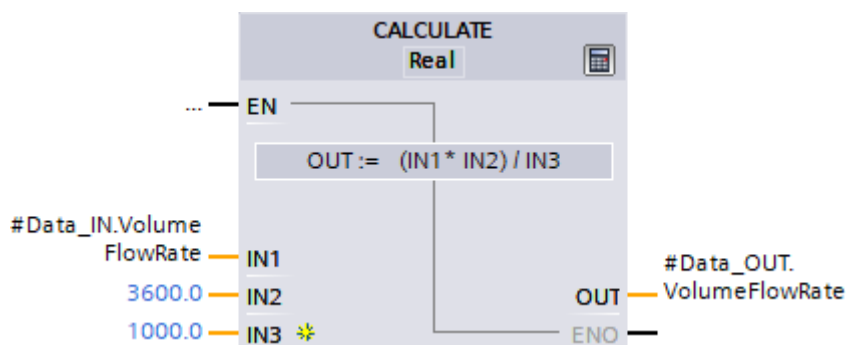
Joonis 7. Modbus funktsiooni plokk

Igal voolumõõtjal on oma andmete plokk, kuhu saabub kõik vajalik info. Uue Modbus ühenduse seadistamisel tuleb sisestada vaid Data Block'i aadress, seadme aadress, andmete aadressid ja STEP_FB parameeter, mis on mõeldud tsükli uuendamiseks ja praktikas on vaid funktsiooni ploki järjekorranumber. Plokis programm võrdleb STEP_FB STEP_IN'ga, mis on tsükli loendur. Kui STEP_FB võrdub STEP_IN'ga, siis teostatakse seadmega ühendust ja võetakse vastu väärtused. Kui kõik seadmed on küsitletud, algab tsükkel uuesti. See on mõeldud seadmete vaheliste konfliktide vältimiseks.



Joonis 8. Modbus'i tsükkel

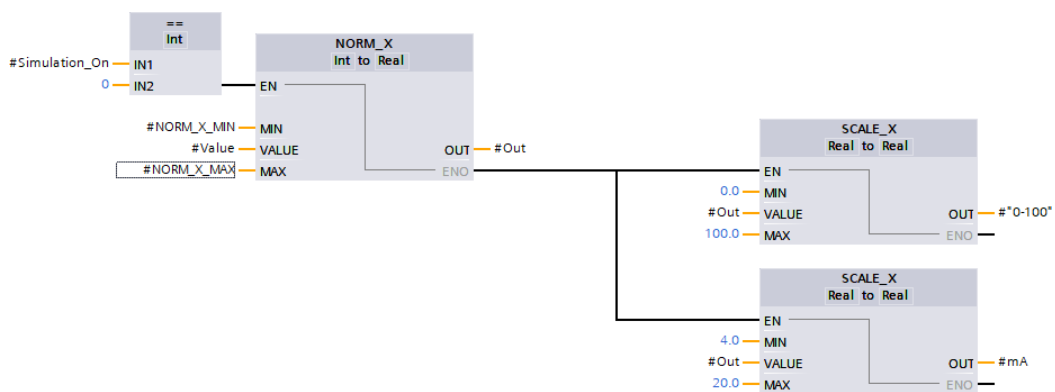
Pärast seda, kui toorandmed on salvestatud, tuleb neid töödelda. Andmed tulevad DWord formaadis ja Tonn sekundis, kasutamiseks andmed pööratakse Real formaati. Siis korrutatakse see arv 3600-ga selleks, et Tonn tunnis kätte saada. Arvu siis jagatakse 1000-ga. See arv tuleb voolumõõtja kasutusjuhendist, ja on mõeldud selleks, et pöörata naturaalarv ratsionaalarvuks. [13] Valmis arv salvestatakse andmete ploki, kust SCADA süsteem selle võtab.



Joonis 9. Arvuga operatsioonid

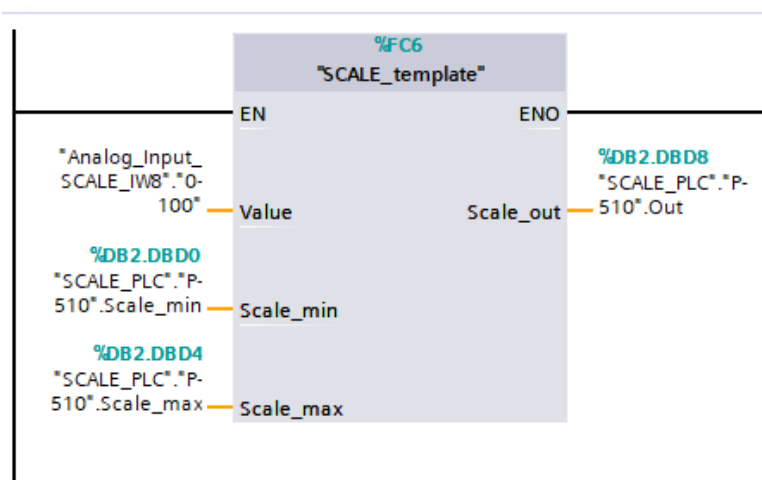
3.4 Analoogsignaali skaleerimine

Signaali skaleerimine toimub etapikaupa. Esimene samm on toorsignaali vastuvõtmine ja normaliseerimine. Selleks veendutakse selles, et simulatsiooni režiim on väljalülitatud (simulatsioon on tehtud seadistamiseks tulevikus). Pärast seda signaal "ära lõigatakse" teatud diapasoonis, see on normaliseerimine. Tavaliselt see diapason on 0 – 27648 volusignaali jaoks ja -27648 – 27648 pingesignaali jaoks. Need on inseneriühikud ja ei peegelda mingi tõelist väärtust. Pärast normaliseerimist, signaal skaleeritakse. Siin tehakse seda 3 korda. Esimesel sammul skaleerimine toimub 0 – 100 ja 4 – 20 diapasoonides. Esimene diapason kujutab ennast arvu protsentides ja teine mõõteseadme mõõtediapasoni milliamprites. Need skaneerimised on tehtud seadistamiseks ja kontrollimiseks, näiteks kontrollmõõteriistade jaoks.



Joonis 10. Skaleerimise esimene samm

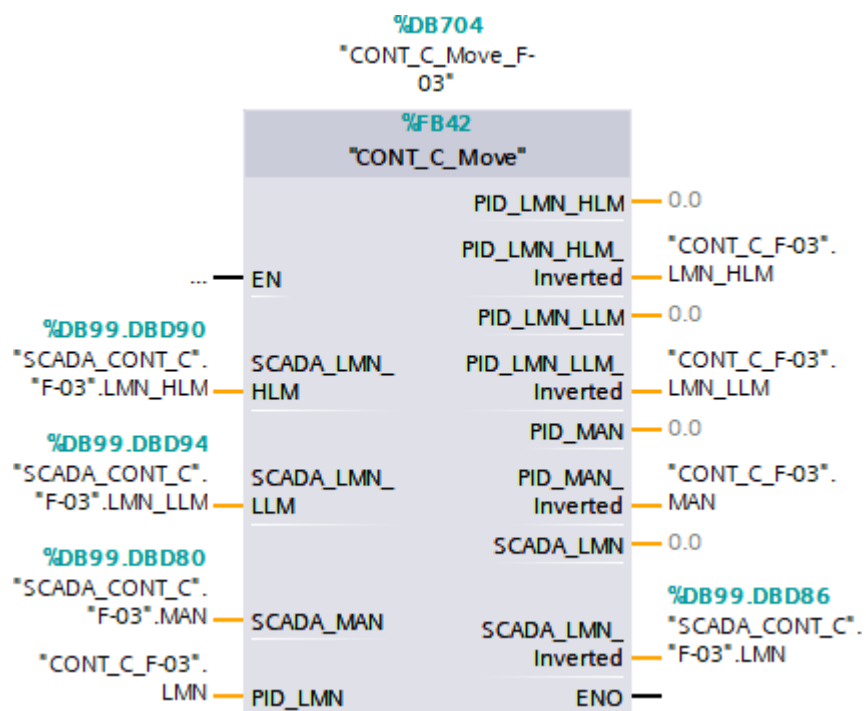
Viimane skaleerimine muutub arv reálnäiduks. Skaala moodustab metroloogia rühm, selle skaala järgi signaalist saadakse näit ja salvestatakse see andmebaasi PID kontrollerite poolt kasutamiseks või SCADA süsteemi tarbeks.



Joonis 11. Lõplik skaleerimine

3.5 PID regulaatorid

Käesolevas seadmes kasutatakse PID regulaatoreid klappide, pumpade ja jahutusventilaatorite töös. Pumpade ja ventilaatoritega töötades: 4 mA signaal tähendab, et seade seisab ja 20 mA signaal, et seade töötab täie jõuga. Klappidega aga olukord on vastupidi, kuna ilma signaalita klapid on avatud, mis võrdleb 100%-ga, signaali seadmele andes klapp sulgub, mis võrdleb 0%-ga. Normaalseks tööks tuleb skaala piirkondi inverteerida, et PID regulaator saaks normaalselt töötada. Selleks oli loodud "CONT_C_Move" funktsioon, mis võtab väärtused vastu ja inverteerib neid.



Joonis 12. Inverteerimisfunktsiooni plokk

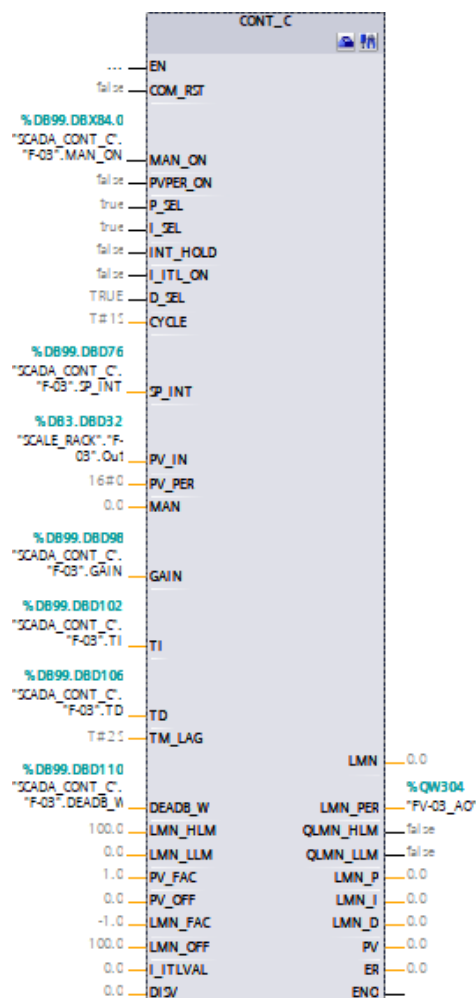
PID kontrollimiseks oli valitud Siemens'i CONT_C funktsioon. TIA Portal võimaldab erinevaid PID regulaatoreid. CONT_C eeliseks on see, et see funktsioon töötab klappidega stabiilselt, võrreldes PID_Compact-iga. Puuduseks aga automaatse seadistuse puudumine [14].

Funktsiooni sisend koosneb järgmistest parameetritest:

- MAN_ON – Käsitsijuhtimis režiimi lüliti
- SP_INT – Setpoint ehk nõutav väärtus
- PV_IN – Process value input ehk anduri näit
- GAIN – Proportsionaalne osa
- TI – Integraal osa
- TD – Diferentsiaal osa
- DEADB_W – Dead band ehk tundetuspiirkond

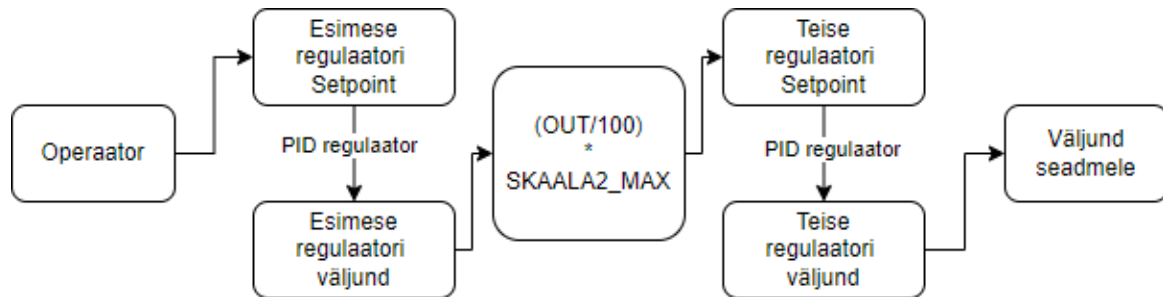
Kõik need väärtused, välja arvatud anduri näidu saab SCADA'st kontrollida ja seadistada. Regulaatori muutujad on võetud vanast süsteemist, peale uue süsteemi paigaldamise kontrollitakse neid ja vajadusel seadistatakse uuesti.

Kontrolleri väljundiks on vaid üks muutuja, LMN_PER, mis annab väljundi väärtust inseneriühikutes.



Joonis 13. CONT_C PID regulaator

Selles projektis kasutatakse ka kaskaadregulaatoreid. Need on samasugused PID regulaatorid selle erinevusega, et kui kaskaadrežiim on sisse lülitatud, kasutab regulaator esimese regulaatori väljundit enda nõutava väärtusena. Kui kaskaad on välja lülitatud, siis reguleerimiseks kasutatakse vaid esimest signaali.

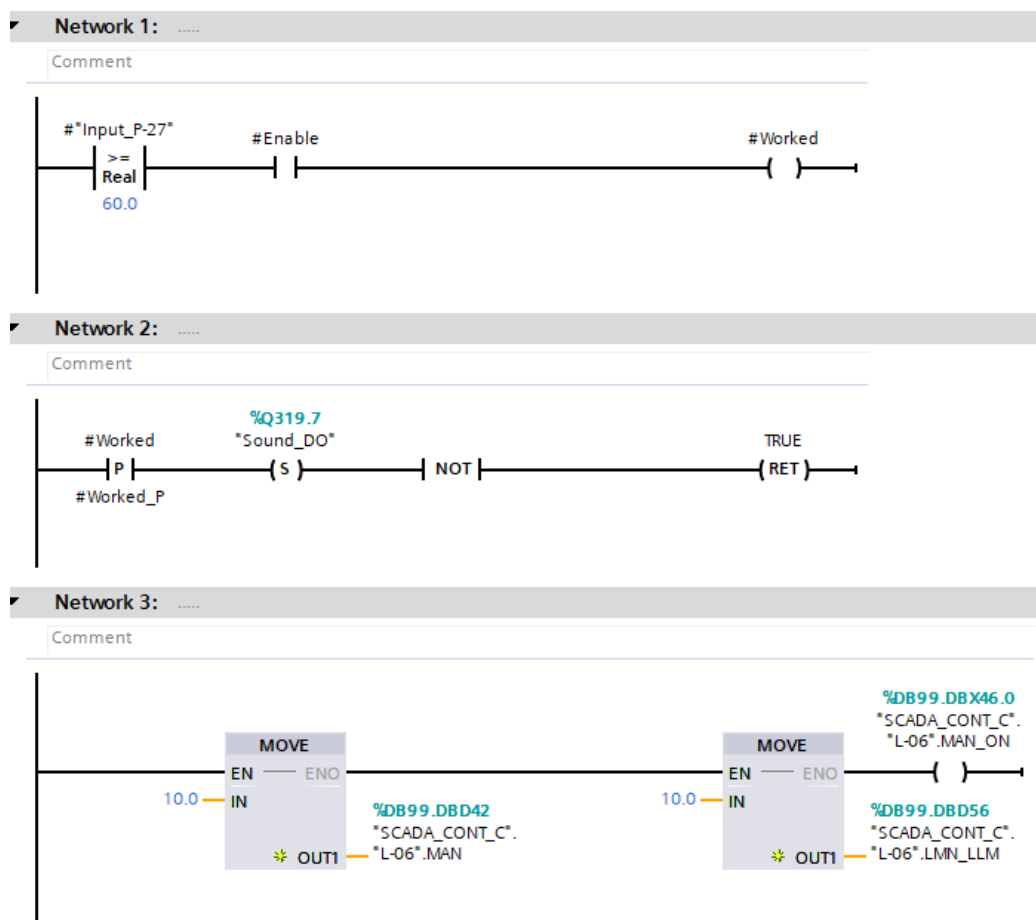


Joonis 14. Kaskaadrežiimi mudel

3.6 Signalisatsioonid ja blokeering

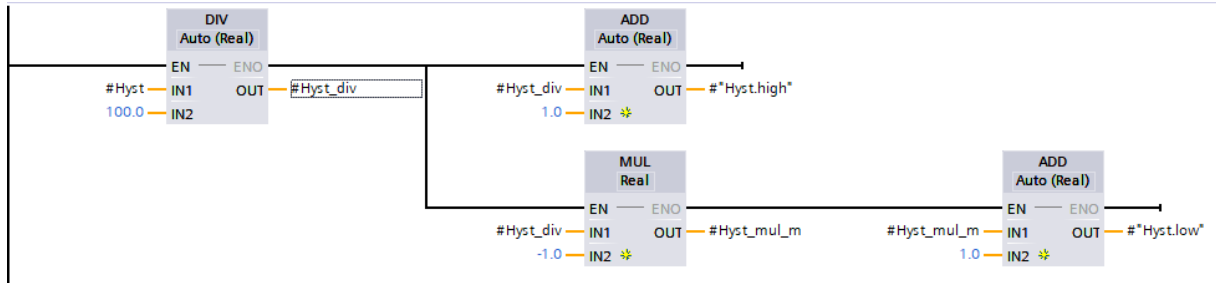
Ohutuse huvides on projektis realiseeritud signalisatsioonide ja blokeeringu süsteem. Põhierinevuseks nende vahel on see, et signalisatsioon ei sega protsessi, vaid teavitab operaatoritele, kui piirväärtused on saavutatud. Blokeering aga lisaks sellele peatab või muudab protsessi, õnnetuse vältimiseks.

Selles projektis on üks blokeering mis analüüsib rõhuanduri andmeid, kui ülempiir on saavutatud, lülitab sisse blokeeringu. See aktiveerib käsijuhtimisrežiimi ning paneb regulaatori ülesandeks 10%. See lubab seadmele rõhku alandada ning plahvatust või seadme kahjustamist ennetada.



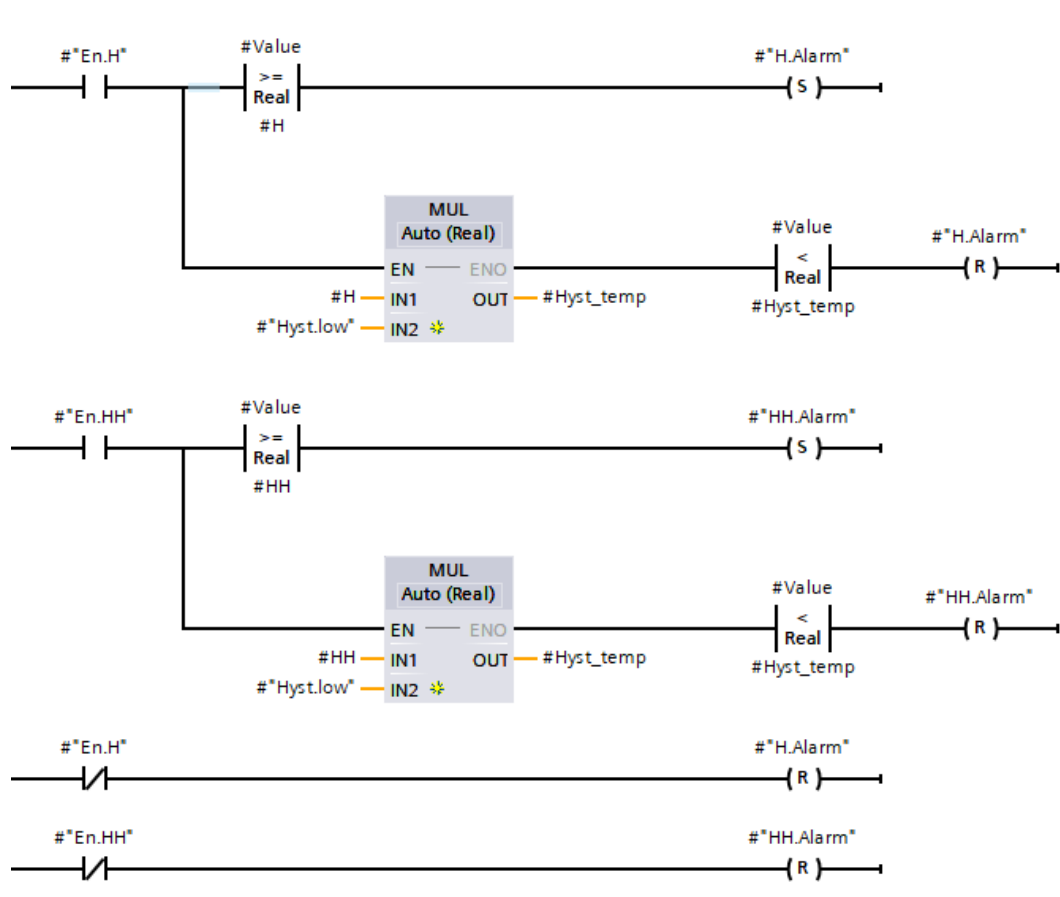
Joonis 15. Blokeeringu programm

Veel üks erinevus on see, et signalisatsioonidele on arvatud hüsterees. See on tehtud kasutajate mugavuseks, et helisignalisatsioon ei lülitaks uuesti sisse just pärast seda, kui see oli tühistatud.



Joonis 16. Hüsteresi arvutamine

Kokku igal anduril on 4 signalisatsiooni, ülempiiri eelteavitus, ülempiiri saavutamine ja samad signalisatsioonid alampiiri jaoks. Piire määratakse metroloogia rühmas ning edastatakse edasi programmeerijatele.



Joonis 17. Signalisatsioonide programm. H - eelteavitus, HH - ülempiiri saavutatud

3.7 Summeerivad loendurid

Selle projekti loendurid on kirjutatud SCL keele abil. SCL ehk Structured Control Language on Pascal'i baasil tehtud programmeerimiskeel Siemens'i kontrolleri programmeerimiseks. Tööprintsip on sama, nagu LAD keeles, aga sobib SCL selles näites paremini, kuna on vaja kirjutada palju samalaadseid koodilõike, LAD-iga kirjutades võtaks see liiga palju ruumi ja tööd, kui SCL-ga kirjutades. Lühidalt, sobib SCL väga hästi arvutamiseks. [15]

```
1 IF #Input_LReal >= 0 THEN
2   #statAccum_lH := ((#Input_LReal * #tempCycle) / #tempIntervall_lH) + #statAccum_lH; // Hour Counter
3   #statAccum_Shift := ((#Input_LReal * #tempCycle) / #tempIntervall_lH) + #statAccum_Shift; // Shift Counter
4   #statAccum_lD := ((#Input_LReal * #tempCycle) / #tempIntervall_lH) + #statAccum_lD; // Day Counter
5   #statAccum_Month := ((#Input_LReal * #tempCycle) / #tempIntervall_lH) + #statAccum_Month; // Month Counter
6   #statAccum_Total := ((#Input_LReal * #tempCycle) / #tempIntervall_lH) + #statAccum_Total; // Total Counter
7
8   /////
9   #UDT_Output.Hour := #statAccum_lH;
10  #UDT_Output.Shift := #statAccum_Shift;
11  #UDT_Output.Day := #statAccum_lD;
12  #UDT_Output.Month := #statAccum_Month;
13  #UDT_Output.Total := #statAccum_Total;
14 END_IF;
15
```

Joonis 17. Loendurid

Valem loendurite jaoks on lihtne. Esiteks võetakse voolumõõtja näit, mille ühikuks on ühik (tavaliselt Tonn või kuupmeeter) tunnis. Seda arvu korrutatakse 1/3600-ga, mis vastab ühele sekundile. See on vool sekundis. Ja siis see arv liidetakse juba loendatud arvuga. See mehhanism töötab kuna funktsioon hakkab tööle iga sekundi tagant, selleks kasutatakse Cyclic interrupt plokki. Nagu nähtav, igal ajavahemikul on oma muutuja. Selleks, et loendurid töötaksid õigesti, on vaja neid muutujaid nullida. Viimane koodilõik salvestab muutujaid massiivi, et neid edasi programmi saata.

```
1
2 IF "LGF_AstroClock_DB".actualLocalTime.MINUTE = 0 THEN
3   IF "LGF_AstroClock_DB".actualLocalTime.SECOND = 0 THEN
4     #UDT_Output.Prev_Hour := #statAccum_lH;
5     #statAccum_lH := 0;
6   END_IF;
7 END_IF;
8
```

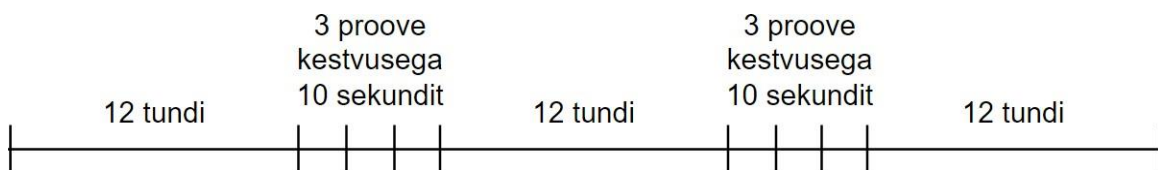
Joonis 18. Loendurite nullimine

Ainuke väliseteek, mida selles projektis kasutatakse, LGF_AstroClock, see teek on mõeldud loendurite jaoks. See teek annab rohkem stabiilsust ja mugavust, kui sisseehitatud kella teegid.

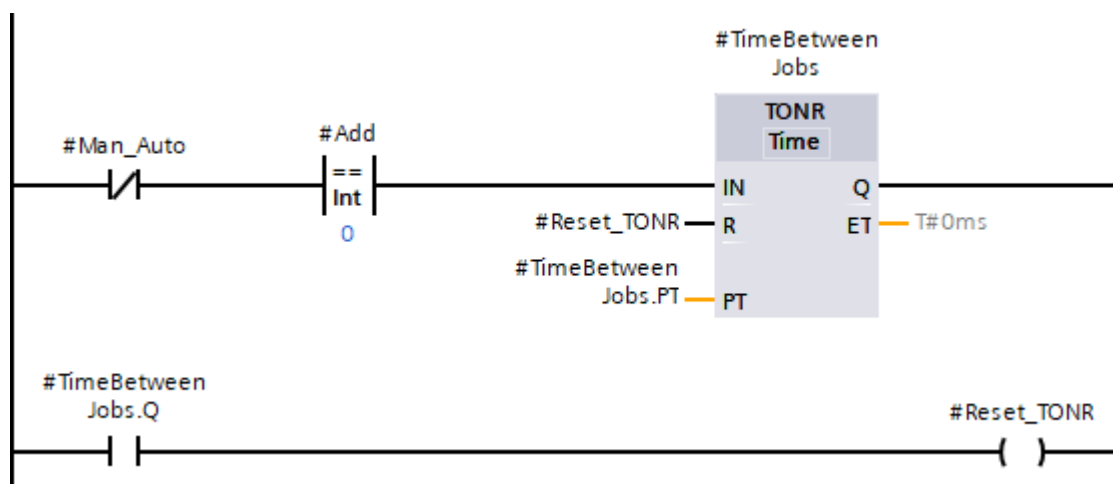
Joonis 18 näitab koodilõiku, kus programm võrdleb AstroClock'ist võetud praegust minutit ja sekundit nulliga. Kui mõlemad on null, siis see tähendab, et tund on möödunud ja loendur tuleb nullida. Loendatud number kirjutatakse Prev_Hour muutuja sisse ja tema asemel tuleb null. Pärast seda algab tsükkel uuesti. Kuu, päeva ning vahetuse loendurid töötavad just samamoodi, erinevad vaid ajavahemikud, millal loendurit nullitakse.

3.8 Proovivõtmise kraanid

Tootekvaliteedi tagamisest võtab keemia labor regulaarselt bensiini ja vaigu proove. Selles seadmes on ettenähtud automaatsed proovivõtmise kraanid. SCADA süsteemis seadistatakse intervallid, mille vahel proovid on võetud, proovi võtmise kestus ja nende arv.

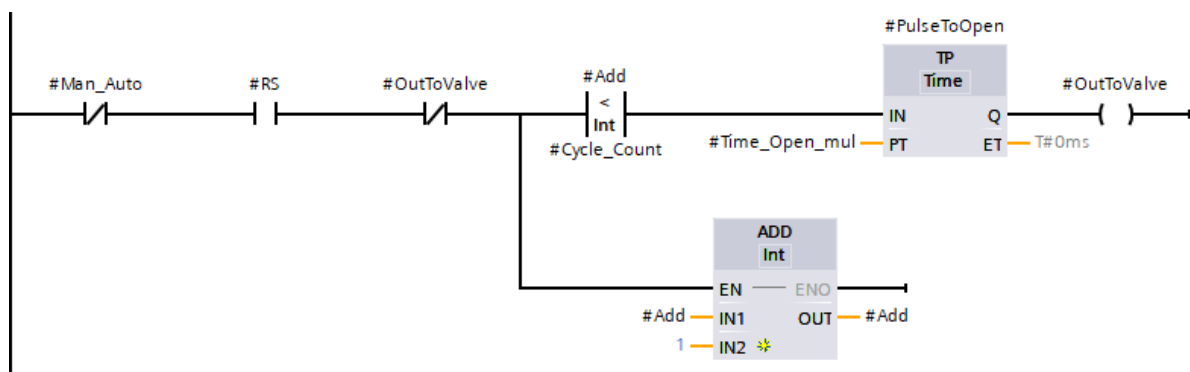


Joonis 19. Proovivõtmise algoritmi skeem



Joonis 20. Proovivõtmise intervallide taimer

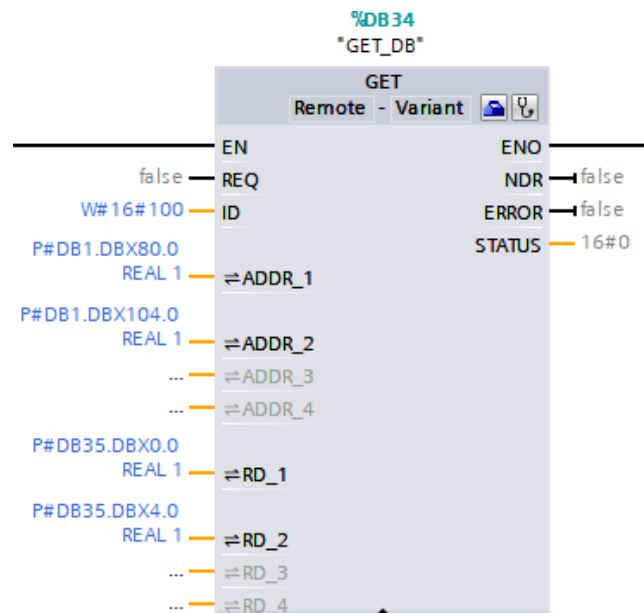
Kui muutuja "Add" võrdub nulliga, siis hakkab tööle taimer "TimeBetweenJobs", mille nõutavaks väärtuks "PT" on nõutav aeg proovide vahel. Kui ettenähtud aeg on läbi, siis saab signaal RS-trigeri S sisendile ja alustab proovide võtmise protsess. Iga proovivõtmise tsükli jooksul töötab TP taimer ja kui tsükkel on lõppenud, suurendab muutuja "Add" 1 kaupa. See jätkub seni muutuja "Add" on väiksem, kui tsüklite nõutav arv. Pärast seda "Add" nullitakse ja intervall proovide vahel algab uuesti.



Joonis 21. Kraani avamine proovivõtmiseks

3.9 Andmed teistest kontrollieritest

Side kontrollierite vahel on tehtud PUT/GET käskude abil. Üks kontrollier funktsioneerib nagu server ja saadab andmed teisele, klient kontrollierile. Selle projekti kontrollier võtab andmeid vastu, saatmist siin ei toimu. Selleks kasutatakse GET käsku.





Joonis 22. GET plokk

PUT/GET plokid töötavad nagu sillad kontrollerte vahel. GET plokki sisestatakse andmete aadressid teises kontrollereis (ADDR sisendid) ja aadressid vastuvõtja kontrollereis, mille sees GET plokk asub, et sinna vastu võetud andmed sisse kirjutada. Side kontrollerte vahel seadistatakse ploki menüüs. Adresseerimine toimub IP aadressi abil ühes võrgus. [16]

Connection parameter

General

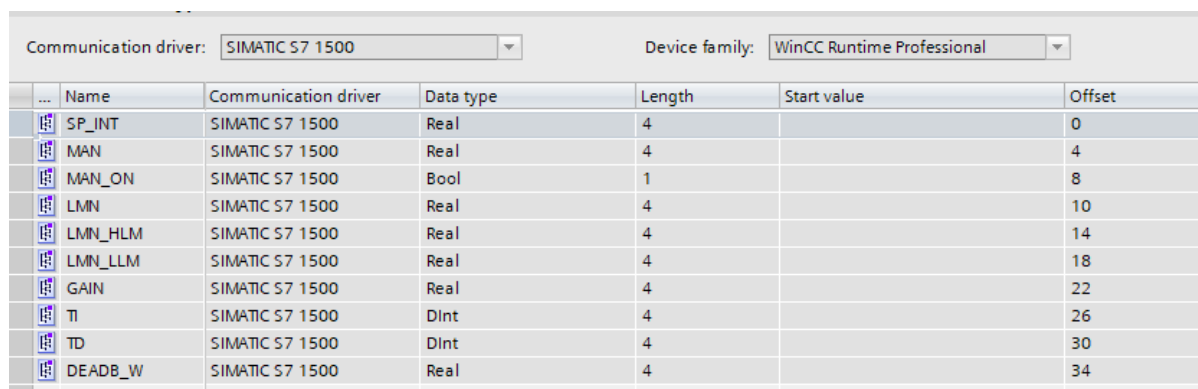
Local	Partner
End point: Ohlawdenie_Diz_FR [CPU 1516F-3 PN/DP]	Unknown
	
Interface: Ohlawdenie_Diz_FR, Main Connection[X2]	
Subnet: Ethernet	Ethernet
Subnet name: PN/IE_1	
Address: 10.10.100.30	10.10.100.40
Connection ID (hex): 100	
Connection name: S7_Connection_1	
<input checked="" type="checkbox"/> Active connection establishment	
<input checked="" type="checkbox"/> One-way	

Joonis 23. GET ploki võrgu parameetrid

4 SCADA SÜSTEEM

Selle seadme SCADA süsteem funktsioneerib aastast 2020, millal oli tehtud suur süsteemi modernisatsioon. Eelmine süsteem töötas Trace Mode tarkvara baasil. Kokkusobivuse huvides oli uueks platvormiks valitud WinCC 15.1 Professional, mis on Siemens'i oma tarkvara, visualiseerimiseks. See tarkvara oli välja töötanud just Siemens'i kontrollritega töötamiseks, mis vähendab programmi arendamise ajal tekkivaid probleeme.

Selle projekti jooksul suuri muudatusi SCADA'le tehtud ei ole. Suurim muudatus oli aadresside muutmine uute vastu. Selleks ettevalmistati uued andmete arhitektuurid, et programmi lihtsustada ja korrastada. Varasemalt olid PID regulaatoril erinevad Tag'id. PID regulaatoritel oli neid 10 tk, see tähendab, et iga regulaator võtab 10 rida Tag'ide loetelust. Selleks, et seda korda teha olin otsustanud teha andmete struktuure, ehk massiive, kuhu kõik need andmed tulevad, just nagu kausta. See lahendus lubab andmetega tööd lihtsustada, kuna kasutaja saab ise valida, mida ta soovib programmiga töötades näha.



...	Name	Communication driver	Data type	Length	Start value	Offset
	SP_INT	SIMATIC S7 1500	Real	4		0
	MAN	SIMATIC S7 1500	Real	4		4
	MAN_ON	SIMATIC S7 1500	Bool	1		8
	LMN	SIMATIC S7 1500	Real	4		10
	LMN_HLM	SIMATIC S7 1500	Real	4		14
	LMN_LLM	SIMATIC S7 1500	Real	4		18
	GAIN	SIMATIC S7 1500	Real	4		22
	TI	SIMATIC S7 1500	DInt	4		26
	TD	SIMATIC S7 1500	DInt	4		30
	DEADB_W	SIMATIC S7 1500	Real	4		34

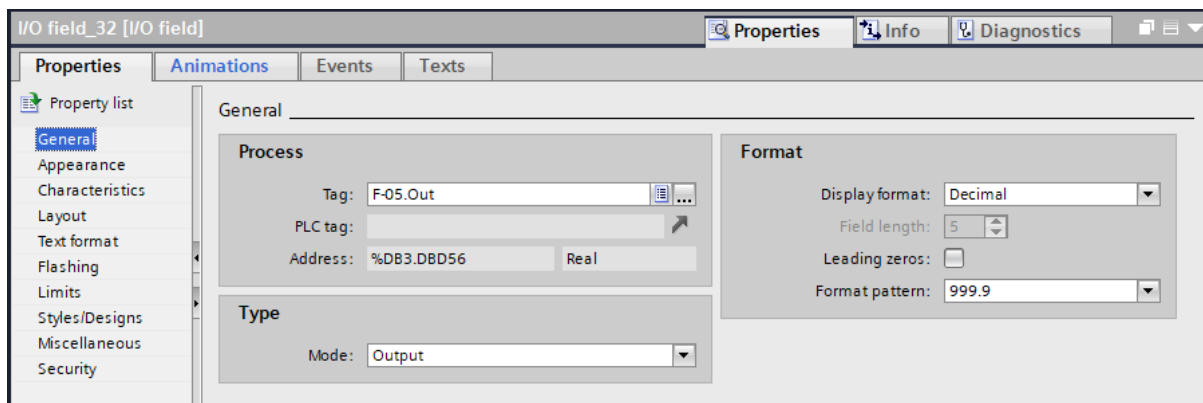
Joonis 24. Andmete struktuuri šabloon programmis

Selle lähenemisviisi eeliseks on see, et tuleb teha vaid üks šabloon, pärast seda saab struktuuri kasutada, seda loetelust valides ja programm rakendub kõik automaatselt. See on eriti kasulik, kui on vaja teha palju samalaadset tööd.

SCADA_PID_F-03		SCADA_PID_Struct...	Ohl_diz_Fr
SP_INT	Real		Ohl_diz_Fr
MAN	Real		Ohl_diz_Fr
MAN_ON	Bool		Ohl_diz_Fr
LMN	Real		Ohl_diz_Fr
LMN_HLM	Real		Ohl_diz_Fr
LMN_LLM	Real		Ohl_diz_Fr
GAIN	Real		Ohl_diz_Fr
TI	DInt		Ohl_diz_Fr
TD	DInt		Ohl_diz_Fr
DEADB_W	Real		Ohl_diz_Fr

Joonis 25. Massiivi kasutamise näidis

Joonis 25 näitab, kuidas see andmete struktuuri kasutada programmis. Massiivi loomise algoritm on järgmine: esiteks luuakse üks muutuja, siin see on "SCADA_PID_F-03", mis tähendab, et see on PID regulaator, mis kontrollib kolmandat voolumõõtjat. Pärast seda antakse sellele muutujale "SCADA_PID_Struct" andmete struktuur. See on see struktuuri nimi, mida koostati varem. Automaatselt saab see muutuja massiiviks ja lisanduvad kõik ettenähtud muutujad selle sisse. Kõiki neid andmeid saab visualisatsioonis kasutada, igaühel on oma aadress ja nimi, nime struktuur on siis "MUUTUJA.objekt". Näiteks, kui on vaja kuvada selle massiivi SP_INT objekti, mis on Setpoint, ehk nõutav väärtus, lihtsalt tuleb kirjutada SCADA_PID_F-03.SP_INT ekraanil loodus tekstiväljas ja see ilmub seal. Joonis 26 näitab, kuidas see töötab. Siin on kasutatud voolumõõtja F-05 "Out" objekt, mis on anduri väljund.



Joonis 26. Tekstivälja seaded

KOKKUVÕTE

Lõputöö käigus oli koostatud ja programmeeritud uus riistvara destillatsiooni seadmele, mis asub Viru Keemia Grupi territooriumil Kohtla-Järve linnas. Samuti olid tehtud muudatused SCADA süsteemile uue riistvaraga korrektseks tööks. Programmeerimine ja visualiseerimine toimus ühes programmis, TIA Portal'is versiooniga 18. Püstitatud eesmärgid olid saavutatud ja pärast testimist probleemeid ei ilmunud.

Eesmärkide hulgas oli:

- Uue riistvara valik
- Programmi koostamine olemasolu programmi baasil
- Programmi osa ümber organiseerimine ja lihtsustamine
- SCADA süsteemi seadistamine

Lõputöös anti ka teoreetilised baasteadmisi kontrollerite kohta, näiteks protokollidest ja programmi koostamisest.

Testiks kasutati PLCSim tarkvara, mis on samuti Siemens'i poolt loodud kontrolleri simulatsioon. Lõppkatseks kasutati päris kontrollerit, selle abil katsetati programmi ja kontrollerit nõ "laua peal". Selle testi tulemuseks oli stabiilne ja korralik töö.

Uuendatud kontrolleri eeliseks vanaga võrreldes on laiendatud arvutusjõudlus, mis toob kaasa ka suurema funktsionaalsust ja stabiilsust. Lisaks annab selle kontrolleri kasutamine rohkem võimalusi terve süsteemi tulevikus moderniseerimiseks.

Lõputöö kirjutamise ajal ei lähe koostatud süsteem tööle, kuna selleks on vaja oodata destillatsiooniseadme seismapanekut. Eeldatavasti pannakse uus süsteem kokku suve alguses, pärast seda läheb uuendatud kontroller tööle.

SUMMARY

The topic of this Bachelor's Thesis is *Distillation Plant's Automation System Upgrade on the Example of Viru Keemia Grupp*. At the moment there are 5 Programmable logic controllers, or PLCs, 2 of which already have been replaced with new ones. The main point of replacing the PLCs is that old ones no longer meet all the requirements and cannot work as intended anymore.

The PLC described in this thesis is responsible for control of Fraction cooling process. It is an outdated Siemens S7-300 series controller, which has been working there from the year of 2008 and is in need of replacement. The problem with it is periodical self-shutdown, connection disruption and out of date software with chaotically written code. In addition to problems mentioned above, new PLC is capable of future improvements of entire plant's system, such as new communication protocols with frequency inverters and providing more opportunities for extra modules and systems added to it. For such task was chosen a newer Siemens S7-1500 PLC, these PLCs have shown great results in Viru Keemia Grupp before, so it is a good choice for this plant as well.

During the work on this project new programme was created. In this programme, problems occurred in the previous version were eliminated, such as unorganized code, written mostly using Russian comments, which made it barely readable for non-Russian speakers. In the new version author uses English comments, with code organized in logical order into folders for better navigation inside the programme. In addition to that, programme was made with the use of functions, which also simplified the programme, meaning that a potential user does not need to know or understand the entire code, only the necessary segment, which became easier to find and make changes.

Finished programme passed a few tests using simulation software and a real PLC. Tested programme works well and ready for further integration into plant's environment. This project will be implemented in early summer of 2024. It is not possible to install and launch the system right now due to need to halt the entire plant to do that.

VIIDATUD ALLIKAD

- [1] Fortinet, „Cyber Glossary,” 11 03 2024. [Võrgumaterjal]. Available: <https://www.fortinet.com/resources/cyberglossary/tcp-ip>.
- [2] T. Kroll, „Kommunikatsiooni põhimõtted,” 11 03 2024. [Võrgumaterjal]. Available: <https://www.scribd.com/doc/78025933/TA>.
- [3] Kaspersky, „What is an IP Address,” 11 03 2024. [Võrgumaterjal]. Available: <https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address>.
- [4] Profinet University, „PROFINET Communication Channels,” 11 03 2024. [Võrgumaterjal]. Available: <https://profinetuniversity.com/profinet-basics/profinet-communication-channels/>.
- [5] Profinet, „Technology Description,” 12 03 2024. [Võrgumaterjal]. Available: <https://www.profinet.com/profinet-explained/technology-description>.
- [6] J. Worth, „What is PROFIBUS and how does it work?,” 12 03 2024. [Võrgumaterjal]. Available: <https://toolbox.igus.com/4410/what-is-profibus-and-how-does-it-work>.
- [7] F. d. Andrade, „Everything about a PROFIBUS network and how to use it with IIoT services!,” 15 03 2024. [Võrgumaterjal]. Available: <https://netilion.endress.com/blog/profibus-network-iiot-services/>.
- [8] S. Sharma, „Modbus RTU: A Comprehensive Guide to Understanding and Implementing the Protocol,” 15 03 2024. [Võrgumaterjal]. Available: <https://www.wevolver.com/article/modbus-rtu-a-comprehensive-guide-to-understanding-and-implementing-the-protocol>.
- [9] TTHK, „PLC programmeerimine,” 15 03 2024. [Võrgumaterjal]. Available: <https://www.tthk.ee/PLC/plcprog.html>.
- [10] Instrumentation Blog, „Siemens PLC Programming Blocks How to use it?,” 15 03 2024. [Võrgumaterjal]. Available: <https://instrumentationblog.com/siemens-plc-programming-blocks/>.

- [11] Automation Fair, „Siemens TIA Portal program blocks,“ 21 03 2024. [Võrgumaterjal]. Available: <https://www.automation-fair.com/siemens-tia-portal-program-blocks/>.
- [12] V. Edema, „Understanding Function Block Instances (Single, Multi and Parameter) in Siemens TIA Portal,“ 23 03 2024. [Võrgumaterjal]. Available: <https://www.solisplc.com/tutorials/function-block-instances-in-siemens-tia-portal>.
- [13] B. Cooper, „TIA Portal, S7 – Using Function Blocks,“ 24 03 2024. [Võrgumaterjal]. Available: <https://theautomationblog.com/using-function-blocks-in-the-s7-1200/>.
- [14] M. Salama, „What is a Data Block? Global Data Blocks in PLC,“ 24 03 2024. [Võrgumaterjal]. Available: <https://instrumentationtools.com/global-data-blocks-in-plc/>.
- [15] Micro Motion, *Modbus Interface Tool Integer Scaling*, Micro Motion.
- [16] Siemens, „PID Control: PID_Compact vs CONT_C,“ 24 03 2024. [Võrgumaterjal]. Available: <https://support.industry.siemens.com/forum/WW/en/posts/pid-control-pid-compact-vs-cont-c/235408>.
- [17] M. Aghajani, „Introduction to SCL Programming in Siemens TIA Portal,“ 24 03 2024. [Võrgumaterjal]. Available: <https://www.solisplc.com/tutorials/introduction-to-scl-programming-in-tia-portal>.
- [18] Siemens, „S7 Communication,“ 24 03 2024. [Võrgumaterjal]. Available: https://cache.industry.siemens.com/dl/files/115/82212115/att_108330/v2/82212115_s7_communication_s7-1500_en.pdf.