

Pumbajaama juhtimine veebiserveri kaudu

Pump station control via web server

TELEMAATIKA JA ARUKATE SÜSTEEMIDE ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Jaroslav Savelitšev

Üliõpilakood: 183604EDTR

Juhendaja: Sergei Panomar, lektor

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt. Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“27” Detsember 2021.

Autor: Jaroslav Savelitšev /allkiri/

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“27” Detsember 2021.

Juhendaja: Sergei Ponomar /allkiri/

Kaitsmisele lubatud

“21” Detsember 2021.

Kaitsmiskomisjoni esimees Mare Roosileht

/nimi ja allkiri/

LÕPUTÖÖ LIHTLITSENTS

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Jaroslav Savelitšev

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose pumbajaama juhtimine web-serveri kaudu

mille juhendaja on Sergei Ponomar, 1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

27 Detsmber 2021

SISUKORD

AUTORIDEKLARATSIOON	2
LÕPUTÖÖ LIHTLITSENTS	3
TERMINID JA LÜHENDUSED	5
SISSEJUHATUS	6
1. ÜLESANNE KIRJELDUS	7
1.1 Ülesanne:	7
1. Ülesanne veebserveriks:	7
2. SEADMED KASUTATUD JUHTIMISEKS	9
2.1 Kontroller S7-1200	9
2.1.1. Kontrolleriga side	9
3. TARKVARA	11
4. OBJEKTI JUHTIMISE STRUKTUUR	12
4.1 Realiseerimine	12
4.1.1 Analoo signaal ja tema häired	14
4.1.2 Nivoo määratlemise	15
4.1.3 Pumpade juhtimine	16
4.1.4 Häired	17
4.1.5 Pumbade järjekord ja loogiline süsteem	20
4.1.6 Kellaaja määratlemine	22
4.1.7 Moto-tunnid	24
4.2 DataLogs	25
4.3 Veeb-serveri hälestamine	28
4.4 Veebitehnoloogia kasutamine	30
5. VEEB-SERVER	35
5.1 Uuendamine taustrežiimis	35
5.2. Animatsiooni JavaSkripti abil	36
5.3 Veebilehed	38
6. PROJEKTI TURVALISUS	43
7. KOKKUVÕTE	45
7.1 SUMMARY	46
8. KIRJENDUSED NIMEKIRJA	47
LISAD	49
1. Lisa 1	49
1. Lisa 2	49

TERMINID JA LÜHENDUSED

1. **URL** – Uniform Resource Locator – Standardne kirjeldamine vorm ressursi aadressi Internetis.
2. **AWP** – Anwenderprogrammierbare Webseiten – veebileht, mis programmeerib kasutaja.
3. **HTML** – Hyper Text Markup Language – standardne programmeerimise keel ülemaailses veebis.
4. **CSS** – Cascading Style Sheets – see on formaalne keel, kirjeldamiseks dokumendi ülemuseks, kasutatakse ühekoos märgistuskeelega.
5. **SCADA** – Supervisory control and data acquisition – dispetšeri juhtimine ja andmete kogumine, vajalik arendamiseks või reaajas toimimise pakkumiseks.
6. **HMI** – Human-Machine Interface, see on inimene-masina interfeis arvuti ja operaatori vahel.
7. **HTTPS** – see on HTTP protokoll laiendamise krüptimise toetamiseks turvalisuse parandamiseks.
8. **WEB** – internetiruum, globaalne võrk, ehitatud protokoll TCP/IP järgi.
9. **ETHERNET** – Tehnoloogia andmete ülekandmiseks seadmete kaudu arvutitesse ja tehasevõrkudesse.
10. **TCP/IP** – see on protokoll kogum, mis määratleb võrkude standarde arvutite vahel.
11. **CPU** – keskprotsessor, mis sooritab algoritmide ahelad võimalikult lühikese aja jooksul.

SISSEJUHATUS

Täna sel päeval veeb tehnoloogia on seotud inimese eluga. Interneti areng ja selle kasutamine on muutunud lihtsaks ja, sellega seoses, sai võimalikuks seadmete juhtimine interneti kaudu.

Täna päeval suhtlus ja teenus HMI kaudu, mis saab teha kohapeal, on enamlevinud, aga seoses IT-tööstuse arendamisega kaugjuhtimine saab realiseeruda interneti kaudu. Kaugdiagnostika ja kaugtehniliste seadmete häälestamine saavad igapäevaseks tegelikkuseks ja aitavad vähendada kulutusi hoolduseks.

See kõik mängib rolli veebtehnoloogiate kasutamisel ja arendamisel automatiseerimise valdkonnas.

ABC Service firma andis mulle järgmise ülesannet: on vaja realiseerida kontrolleriil Siemens S7-1214C DC/DC/DC programmi. Programm sisaldab projekti pumbajaama juhtimiseks. Samuti oli vaja realiseerida visualisatsioon veeb-tehnoloogiate abil.

Ülesandeks oli realiseerida visualisatsioon veeb-serveri tehnoloogia abil.

Ülesanne püstituse põhjuseks on ülemaailmne pooljuhtide kriis. Tehnilise protsessi visualiseerimine veebserveri kaudu teeb projekti odavamaks.

Pumbajaama arendamis-juhtimise süsteem veebserveri kaudu.

Et süsteemi arendada, on vaja:

1. Programmi koostamine kontrolleriil S7-1200 Siemens seeriast;
2. Looma veebserver, kirjutada skripte.

1. ÜLESANNE KIRJELDUS

ABC Service firma anti järgmist korraldust: oli vaja TIA portali arenduskeskkonnas koostada programm SIEMENS S7-1214 DC/DC/DC kontrolleri jaoks. Samuti oli vaja koostada ka visualiseerimis-süsteemi selle programmi jaoks, kasutades veeb-serveri tehnoloogiat.

1.1 Ülesanne:

- 1) On vaja organiseerida kahe pumba töörežiimi kordamööda. Kui vedeliku tase saab maksimaalseks, siis esimene pump hakkab töötama ja pumpama vett välja. Kui vee tase saab minimaalseks, siis esimene pump lülitatakse välja ja põhipumba funktsioon edastatakse teisele pumbale;
- 2) Avarii situatsioon: kui vee tase saab avarii väärtuseks, siis peavad mõlemad pumbad hakkama tööle ja pumpama vett, et saavutada tase minimaalseks;
- 3) Kui põhipump töötab ja tase on maksimaalne ja ei lange, siis on vaja lülitada sisse abipump ja pumpata vett minimaalse tase saavutamiseks.

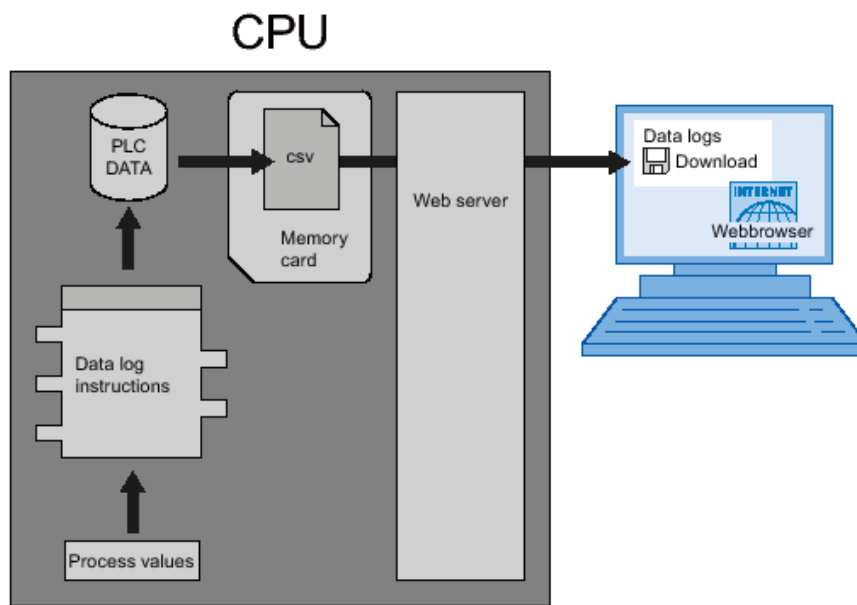
1. Ülesanne veebserveriks:

Realiseerida kolm töölehti:

- 1) Pealeht väärtuste monitoorimiseks;
- 2) Leht häirede väljalülitamise võimalusega;
- 3) Leht parameetrite muutmise võimalusega.

Pealehel peab olema realiseeritud seadmepilt, põhiparameetrite väljastus, kahe pumba juhtpaneel, mototundide väljastus.

Peavad olema kaks tabelit logiga. Esimene tabel koguaeg salvestab kirjendeid, teine tabel teeb kirjendeid ainult sündmustele. Esimene tabel peab tegema kirjendeid minutilise vaheajaga, ta peab salvestama põhisiltide väärtusi. Teine tabel teeb kirjendeid ainult möödunud häiretele. Andmed saab hoida .csv formaadis, veeb-serveris, reas „DataLogs“.



1.2. Joonis 1. DataLogi loomine

2. SEADMED KASUTATUD JUHTIMISEKS

2.1 Kontroller S7-1200

Süsteemi juhtimiseks, projektis on kasutatud programmeeritav loogiline kontroller S7-1200 perest. S7-1214C DC/DC/DC kontroller sisaldab: 14 digitaalseid sisendeid (DI) 24V, 10 digitaalseid väljundeid (DO) 24V, 2 analoogset sisendeid (AI) 0-10V, toiteallikas 20.4-28.8V. S7-1200 programmeeritavad kontrollerid on kompaktsed, nende turvalistase on IP20, töötavad temperatuuri diapsoonis 0-st +50-ni °C. Kontrollereid võib teenindada 10-st 284-ni diskreetseid ja 2-st 51-ni analoogseid sisend-väljund kanaaleid [\[9\]](#). Moodulne, kompaktne kontroller on vajalik süsteemide automatiseerimiseks, kus on nõutavad laienenud funktsionaalsus ja HMI ja võrgu ühendamise.

Kontroller sisaldab protsessorit, sisseehitatud toiteallikast, sisse- ja väljundahelaid.

S7-1200 kontrollerisse on sisseehitatud täisfunktsionaalne veebserver, mis lubab laadida veebilehti, veeb-brauseri tänu saada juurdepääsu protsessile.

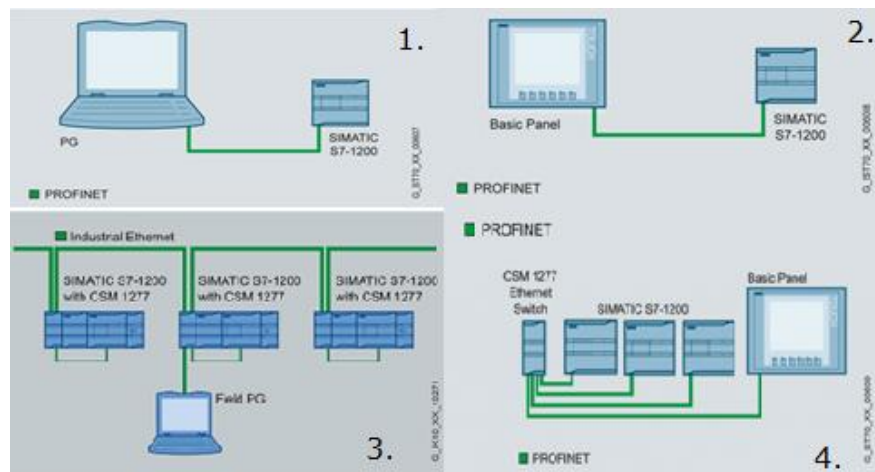
See võimaldab ka kasutada juba valmistatud võrgustruktuuri, ilma lisatud seadeteta, diagnoosida, hinnata ja kontrollida.

2.1.1. Kontrolleriga side

CPU-I S7-1200 on PROFINET-i port andmete vahetamiseks PROFINET-i võrgu kaudu. Profinet (**Process Field Network**) - see on tehase avatud standard, mida kasutatakse assotsiatsioonist „Ethernet Profibus and Profinet International“ automatiseerimiseks. Profinet kasutab TCP/IP standardi ja reaalaaja režiimi Ethernet. Kiirus on 100 Mbit/sekundis.

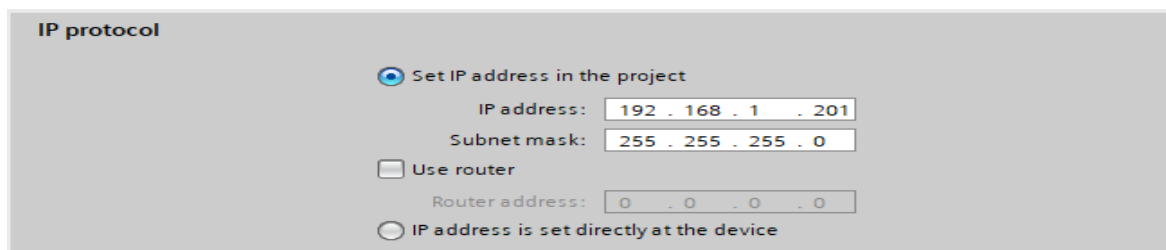
See võimaldab teha andmete vahetamist koos:

- 1) Programmatoriga
- 2) Seadmete ja süsteemid inimene-masina interfeisiga
- 3) Erinevate kontrolleritega võrgumooduli CSM 1277 kaudu
- 4) Seadmetega Ethernet CSM 1277 kommutaatori kaudu



2.1.1 Joonis 1. Ühendamise tüübid

Selles projektis on realiseeritud esimene punkt. Selle jaoks TIA Portalis on vaja määrata kontrolleri IP-aadress, näiteks (192.168.1.200). Arvutis on vaja määrata Ethernet porti IP-aadress, alamvõrk peab olema sama, nagu kontrolleri ja viimase numbrid peavad olema teised, mitte nagu kontrolleri, näiteks (192.168.1.xxx).



2.1.1 Joonis 2. IP-aadressi määramine

Kõik need parameetrid võib häälestada kontrolleri sees, TIA Portalis „PROFINET interface [X1] -> IP protocol“. [\[3\]](#) (vaata pilt 2)

3. TARKVARA

Kontrollerite konfigureerimiseks ja programmeerimiseks kasutatakse „TIA Portal V15.1“ tarkvara. Tarkvara „TIA Portal“ - see on arenduskeskkond, vajaliku süsteemi juhtimiseks. Seal on integreeritud protsesside automatiseerimise tarkvaraarendusekeskkond ajamite ja kontrollerite tasemelt kuni inimese ja masina interfeisi tasemeni ja SCADA süsteemid.

Arenduskeskkonnas samuti asub instrumendid programmi loomiseks mitmes programmeerimise keeles (LAD, FBD, SCL, STL, GRAPH). [\[18\]](#)

Projektis „Tia Portal“ oli kasutatud kahe programmeerimise keeled: LAD ja SCL.

- 1) **LAD** (Ladder diagram) - see on releeloogika keel, kasutatakse loogilise kontrolleri programmeerimiseks. Süntaksis on mugav loogikaskeeme asendamiseks;
- 2) **SCL** (Structured Control Language) - see on SIMATIC S7 kõrgetasemeline programmeerimiskeel, sobib keeruliste algoritmide programmeerimiseks või ülesannete jaoks.

3.1. Programmeerimised plokid

Efektive struktuuri loomiseks arenduskeskkonnal TIA Portal kasutatakse erinevad plokid:

1. **OB** – Organiseerimisblokk, see on peablokk, sest see määrab kogu programmi struktuur.
2. **FC, FB** – funktsioon ja funktsionaalplokkid, sisaldab programmi koodi, vastab konkreetsele ülesandele. Iga funktsioon ja funktsionaalne blokk annab sisse- ja väljundparameetrite kogum ülesannete funktsioneerimiseks.
3. **DB** – blokk, mis hoiab andmeid, mis saab kasutada teistes plokkides. [\[9\]](#)

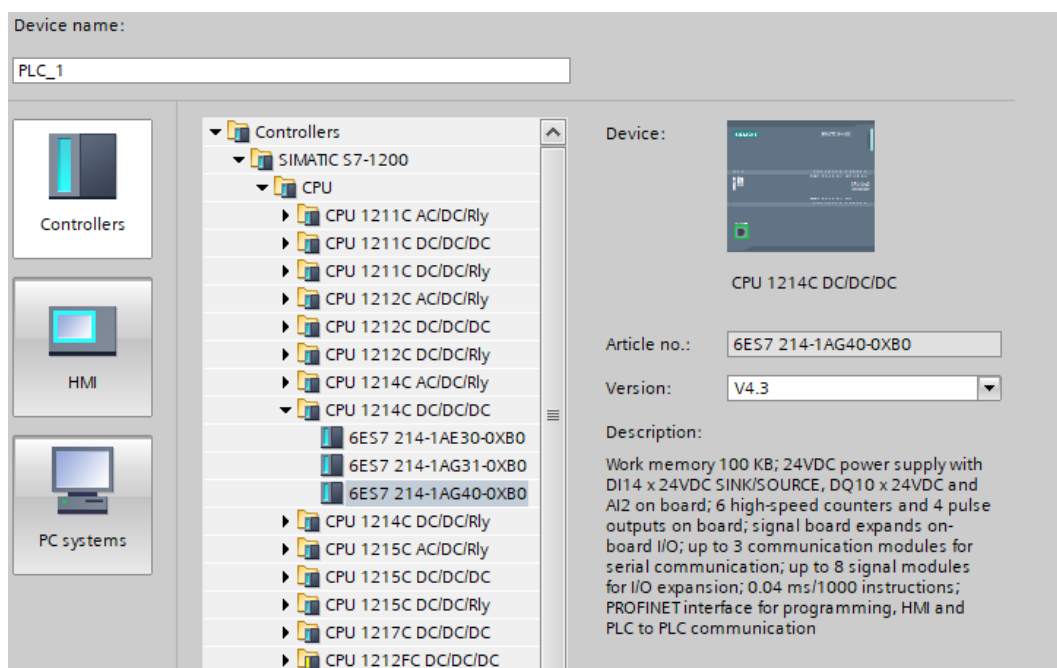
Programmi täidab ülesannet organiseerimisplokkide abil (OB), mis käivitatakse pärast RUN režiimis, töödeldakse üks kord. Organiseerimisblokk saab töödelda katkestava sündmust, mis võib olla standartne sündmus või sündmuse-häire.

4. OBJEKTI JUHTIMISE STRUKTUUR

Süsteemi juhtimiseks kasutatakse pumbajaama kahe pumpaga ja targa juhtimise süsteemiga. Seda pumbajaamat kasutatakse tehases. Pumbajaama põhiülesanneks on vee pumpamine kaevandusest välja, automaatse ja manuaalse töörežiimi võimalusega.

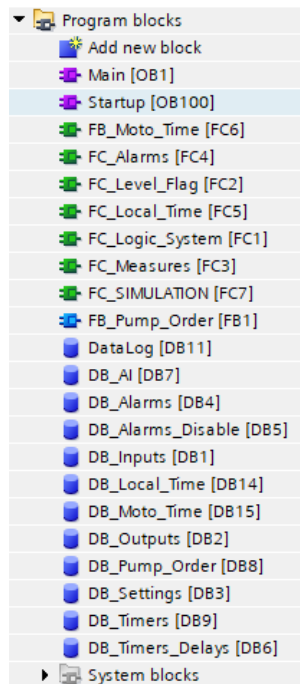
4.1 Realiseerimine

Pumbajaama programm oli realiseeritud arenduskeskkonnas TIA Portal V15.1. Esimene samm oli projekti loomine. Järgmine samm seade häälestamine. Lingil „Add new device“ on vaja valida kontrolleriit. Kasutatud kontrolleri on SIMATIC S7-1214C DC/DC/DC 6ES7 214-1AG40-0XB0 ja selle versioon V4.3. (vaata pilt 1)



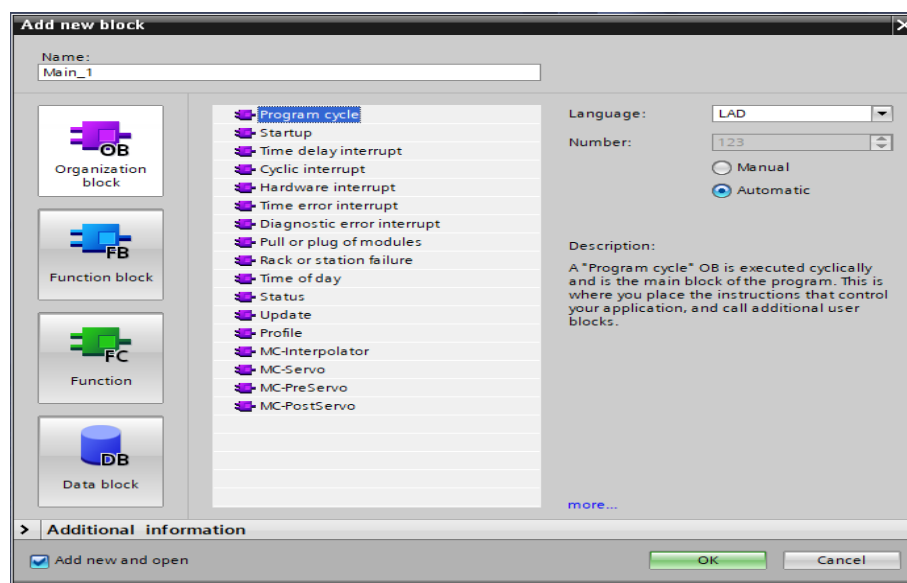
4.1 Joonis 1. Kontrolleri valimine

Programmi plokis on vaja organiseerida loogiline struktuur. See tekitab organisatsiooni, funktsiooni, data plokile abil. See loogiline struktuur töötab nagu hierarhia puu, kus igale plokile kuulub kindel piirkonna koodi. (vaata pilt 2)



4.1 Joonis 2. Hierarhiapuu

Et luua iga plokk, on vaja kahekordselt pressida „Add new block“ reases. [13] (vaata pilt 3)

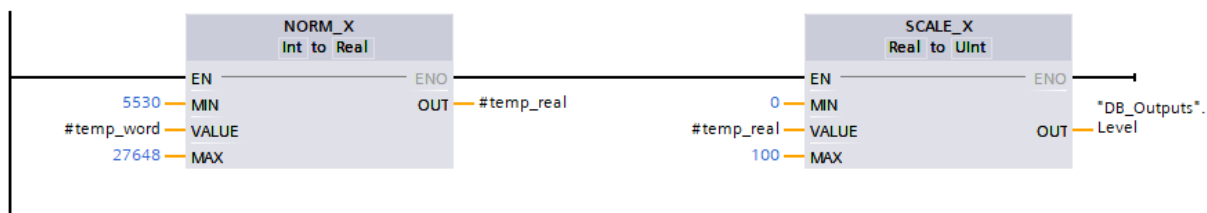


4.1 Joonis 3. Programmiplokkide valimine

Ilmunud aknas võib valida iga vajalik plokk. Projekti loomiseks automaatselt tekitatakse „Program cycle“ plokk. See on peaprogrammi plokk. Projektis ka on olemas „Startup“ plokk, mis töödeldakse ainult üks kord, kui PLC staatus muutub STOP-ist RUN-ini. „Startup“ plokil on eesõigus „Program cycle“ plokki ees. Esiteks, täidetakse kood, mis asub „Startup“-is. Pärast täitmist, „Program cycle“ OB saab töötada. Programmi struktuuri loomiseks oli tehtud algoritm. (asub LISAS 1)

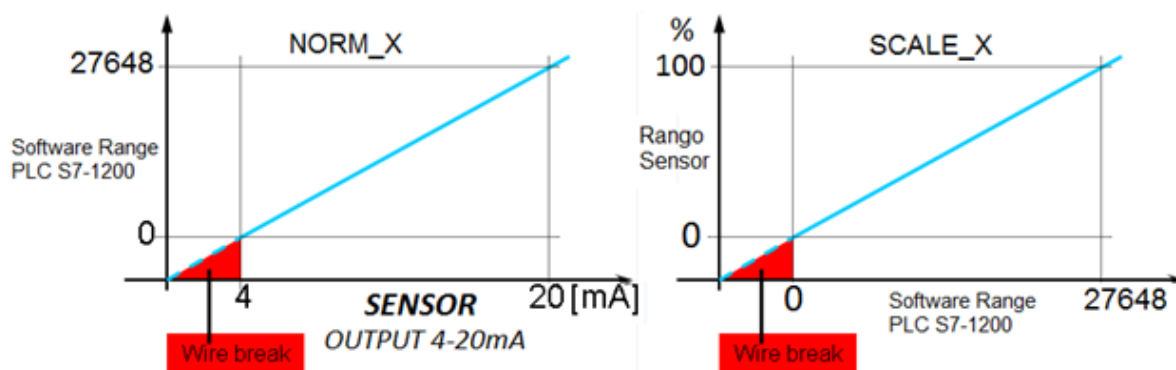
4.1.1 Analoo signaal ja tema häired

Programm alustab töötada plokis „FC_Measures“, kuhu saavad analoog signaalid reaalsest seadmetest. Analoo signaalie töödeldakse NORM_X ja SCALE_X funktsioonide abil, kus NORM_X normaliseerib signaal „Real“-setesse ühikutesse ja SCALE_X skaleerib saadud signaali 0-100 skaalaks. [\[10\]](#) (vaata pilt 1)



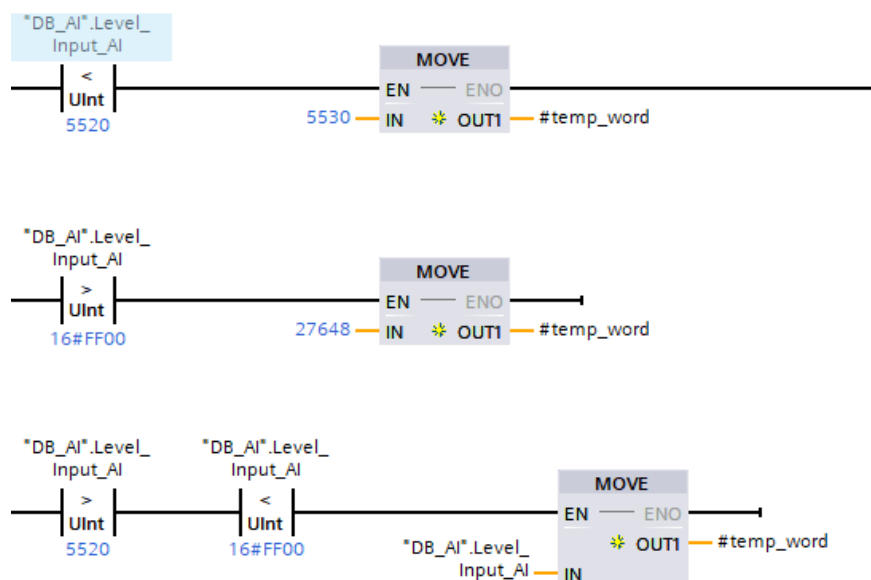
4.1.1 Joonis 1. Analoo signaali töödeldamine

5530 ja 27648 väärtused – need on inseneri-ühikud. Projektile on vaja kasutada inseneriühikud, mis seostavad voluga(4-20mA). Inseneriühikud tekkivad, kui 0/4-20 mA (füüsiline signaal) saabub kontrolleri AI moodulile ja kontrolleri konverteerib neid inseneriühikutele. Oletame, et analoogväljundiks on nivoo seadistus, kus analoogsisendi väärtus 0 tähistab 5530 ja 27648 tähistab 100. Mälus oleva nivooväärtuse, mis on vahemikus 0 kuni 100, teisendamiseks kui analoogväljund on vahemikus 5530 kuni 27648, on vaja normaliseerida (NORM_X) inseneri-ühiku väärtuse väärtusele 0,0 kuni 1,0 ja seejärel skaleerida (SCALE_X) selle analoogväljundi vahemikku 5530 kuni 27648. [\[11\]](#) (vaata pilt 2)



4.1.1 Joonis 2. Analoo signaali töödeldamine skeem

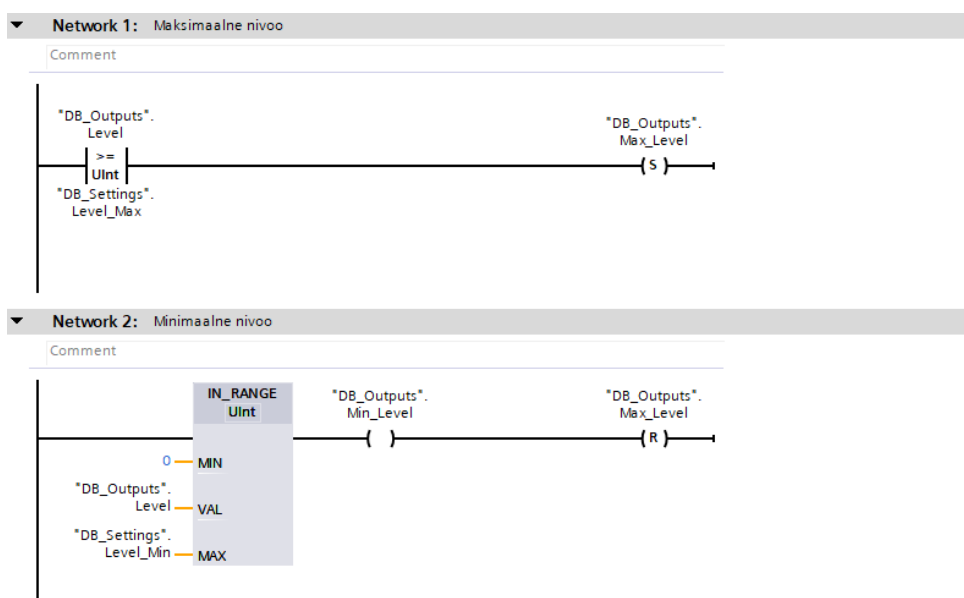
Väärtuse stabiilseks väljundiks juhe katkemise korral kasutatakse „MOVE“ funktsiooni. (vaata pilt 3)



4.1.1 Joonis 3. Matemaatiline võrdlemine

4.1.2 Nivoo määratlemise

„Level-Flag“ plokis määratakse vedeliku nivoo, kas see saab maksimaalse väärtus või minimaalse. Maksimaalne nivoo määratakse „set“ silti abil. Minimaalne nivoo määratakse „IN_RANGE“ funktsiooni abil. Funktsioonisse on vaja kirjutada minimaalne ja maksimaalne nivoo väärtused ja seale saadakse tema reaaliveärtus. (vaata pilt 1)

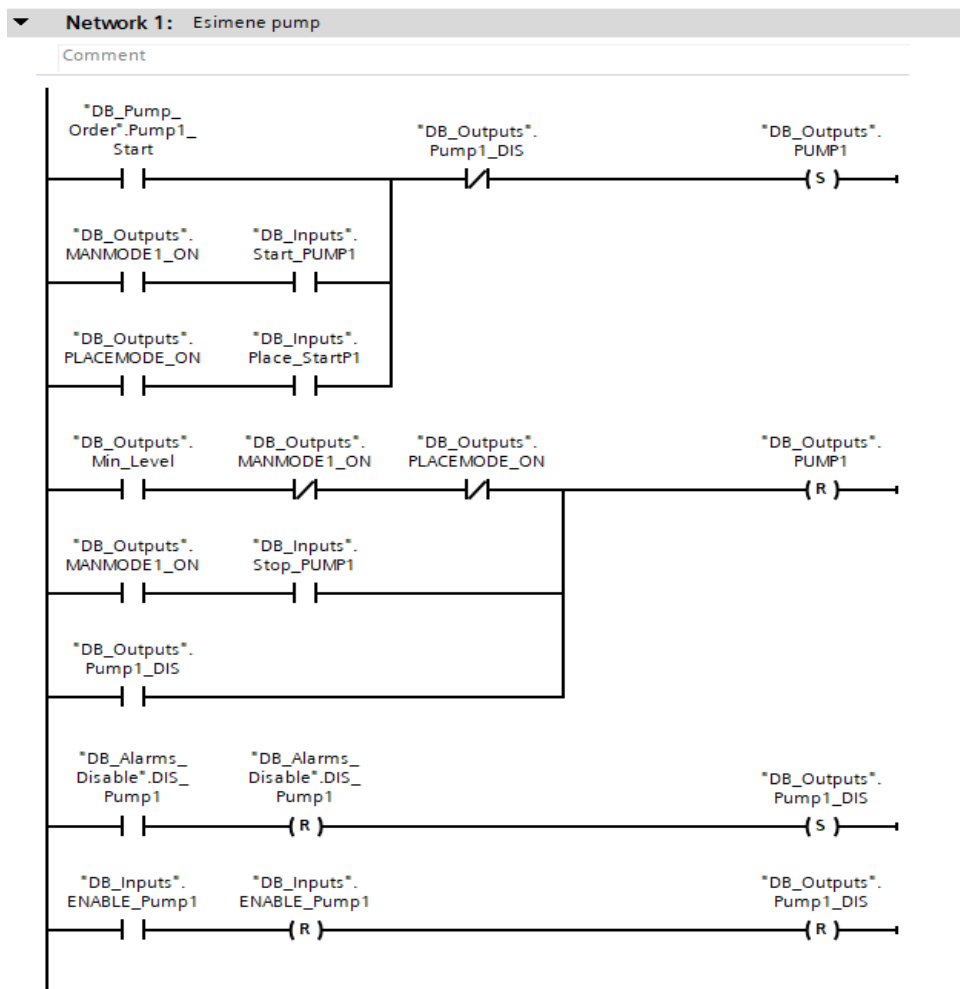


4.1.2 Joonis 1. Vedeliku nivoo määratamine

4.1.3 Pumpade juhtimine

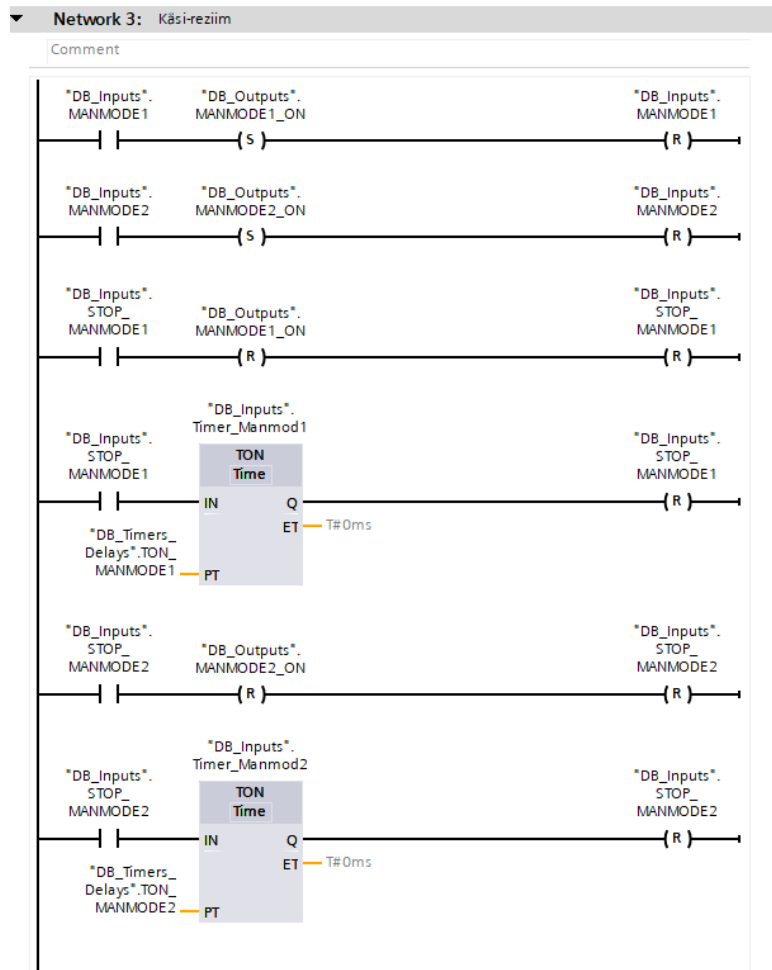
Kogu loogiline süsteem tehakse „Logic_System“ plokis. Plokis on tehtud juhtimissüsteem pumpade manuaalses ja automaatses režiimides. Juhtimist manuaalses režiimis võib rakendada kohapeal või operaatori panelist.

Skeemil olemas käivitamise varianti - see on automaatselt variant, käsi varaint, tagasisidega ja kohalik variant sama tagasisidega. Kui programmil pudivad häired, häire tagastamine, siis pump peab töötama. Allpool asuvad siltid häirete tagastamine. (vaata pilt 1)



4.1.3 Joonis 1. Pumbajaama juhtimine

Koodis käsi-režiimis kasutatakse taimerid. Nad on vajalikud veeb-serveriks, signaali pikaajaliseks saabumiseks. (vaata pilt 2)

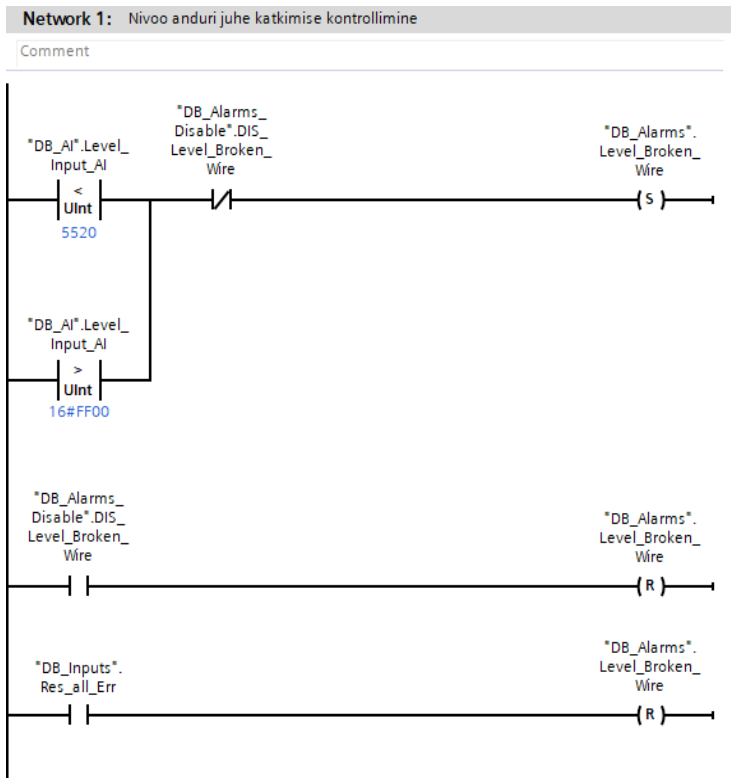


4.1.3 Joonis 2. Taimerid web-serveriks ja MANMODE tagastamine

4.1.4 Häired

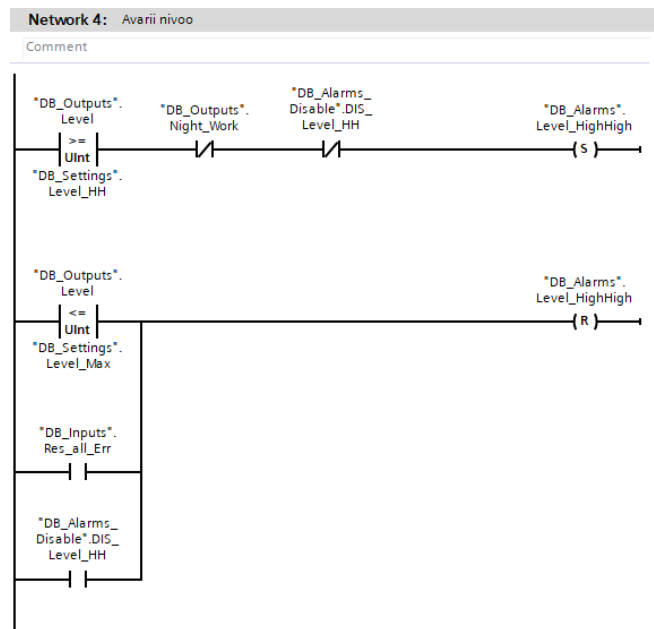
Funktsionaalplokkis „FC_Alarms“ on tehtud loogilise skeemid häirede tuvastamiseks. Igal skeemil on oma viis häirede tuvastamiseks. Programmis on kasutatud matemaatiline võrdlus: kui signaal, mis sattub inseneeri-ühikutes, suurem, kui 16#FF00 väärtus (see on väärtus suurem, kui 32767 inseneriühikud, see tähendab „overflow“), siis see tähendab, et juhe läks katki. (vaata pilt 1)

Kui inseneeri-ühikud on väiksemad, kui 5520 väärtus, siis see tähendab, et seadme juhe läks katki või on vaja seda kalebreerida. [\[12\]](#)



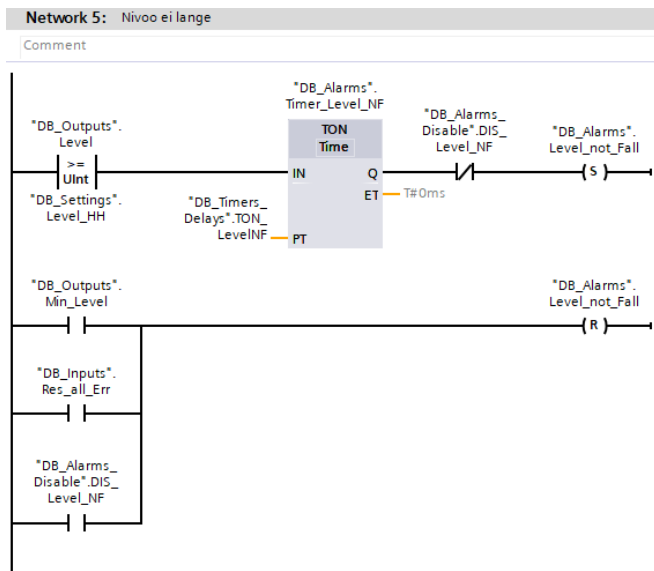
4.1.4 Joonis 1. Häirede tuvastamine

Kellaajast sõltub milline nivoo häire ilmub. Päeva avarii nivoo - 98% mahuti täitmine, öösel parameeter ei tööta, sest ööl pumbad töötavad iga väiskema nivoo muutusel (see koodi osa oli realiseeritud erinevates plokkides), sest elektroenergia on odavam sellel ajal. [\[12\]](#)



4.1.4 Joonis 2. Matemaatiline võrdlused

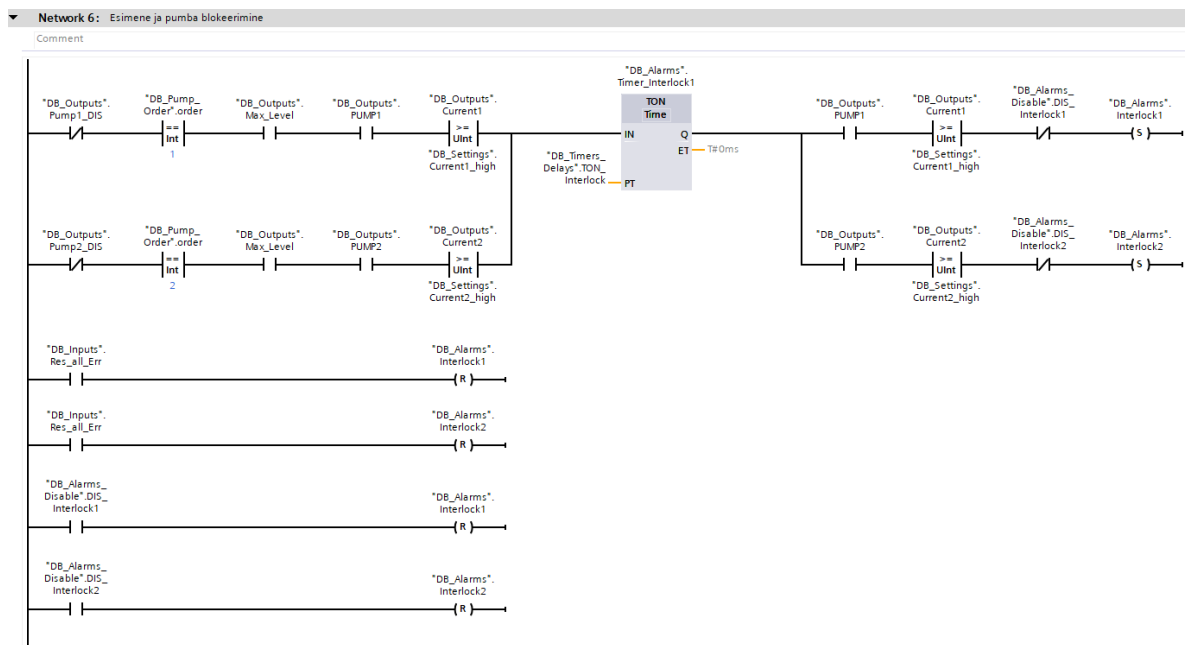
Neljandas Network-is, sama matemaatilise võrdluse abil, määratakse kas nivoo on jõudnud avarii väärtuseni. (vaata pilt 2)



4.1.4 Joonis 3. Avarii nivoo määramine

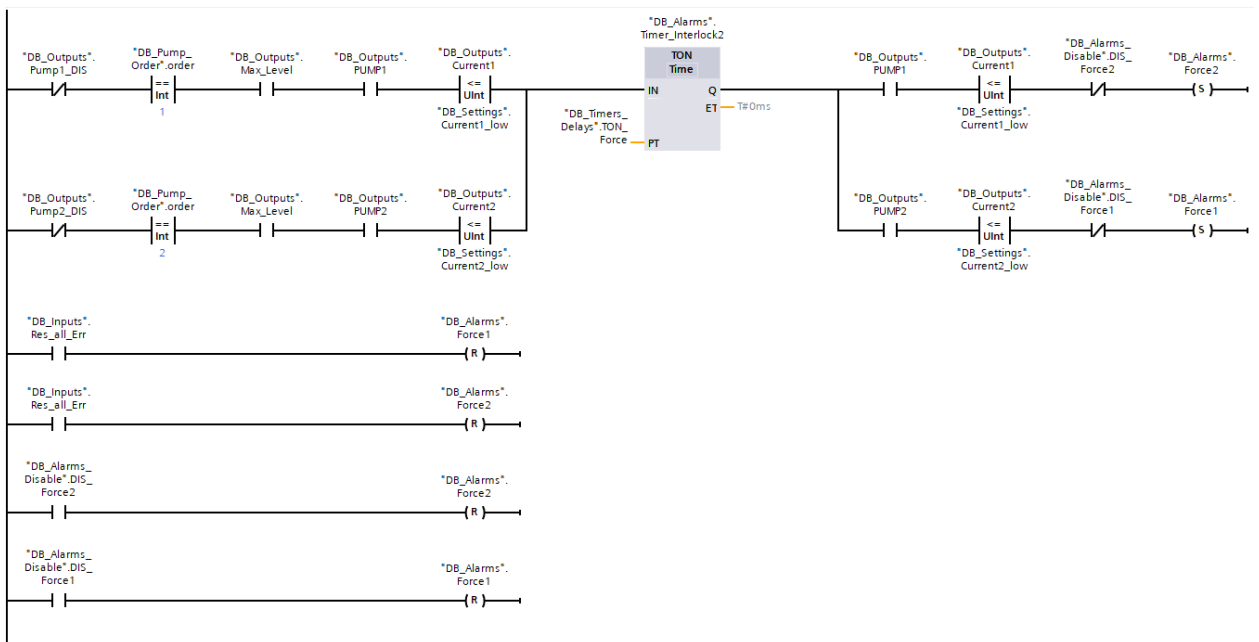
Kui nivoo väärtus on suurem, kui 90% (maksimaalne vedeliku nivoo mahutis), siis alustab tööd taimer ja kui nivoo ei lange taimeri lõppemisel, siis tuleb häire. Häire tulemusel hakkavad töötama kaks pumbad ning pumpama vett välja kuni minimaalse nivooni. (vaata pilt 3)

Samuti oli realiseeritud häired pumpade blokeerimiseks. Pumpad blokeeritakse, kui nivoo väärtus on kõrgem, kui normaalne. Minimaalsed, normaalsed, maksimaalsed väärtusi häälestatakse siltide abil.



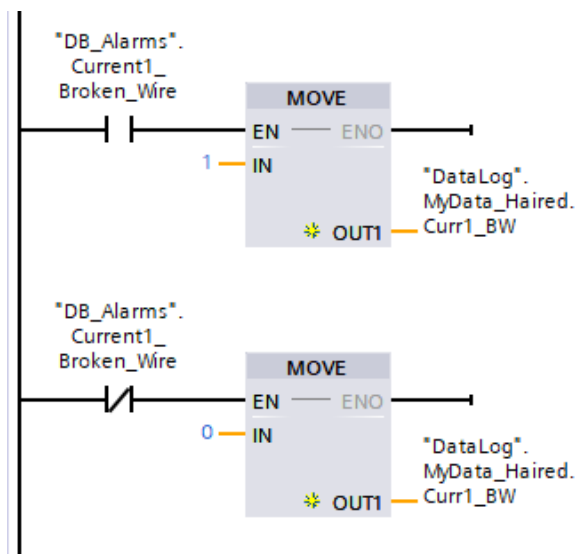
4.1.4 Joonis 4. Pumpade blokeerimine, kui vool on maksimaalne

Niisama on ehitatud struktuur pumpade tööjärjekorra muutmiseks. Voolu minimaalse väärtuse saavutamisel blokeeritakse peapump ja põhiülesanne kantakse üle teistele pumpadele. (vaata pilt 4,5)



4.1.4 Joonis 5. Pumpade blokeerimine, kui vool on minimaalne

„FC_Alarms“ plokis samuti kasutatakse „MOVE“ funktsioonid andmete kirjutamiseks DataLog-isse. See meetod rakendatakse sündmuse tabeli jaoks, mis asub veeb-serveris. (vaata pilt 6)



4.1.4 Joonis 6. Kirjutamine Datalogisse

4.1.5 Pumbade järjekord ja loogiline süsteem

„FB_Pump_Order“ plokis on kirjutatud kogu algoritm töötamiseks automaatses režiimis. Tööde tingimused pumbadeks, töödeks päeva- ja öö ajal oli kirjutatud SCL keeles. See oli tehtud sellepärast, et kood ei muutu mahukamaks ja see võimaldab mitte kasutada erinevaid

plokifunktsioone. Seal on kirjutatud tingimused pumbade töö alustamiseks ja lõppemiseks. [13] (vaata pilt 1)

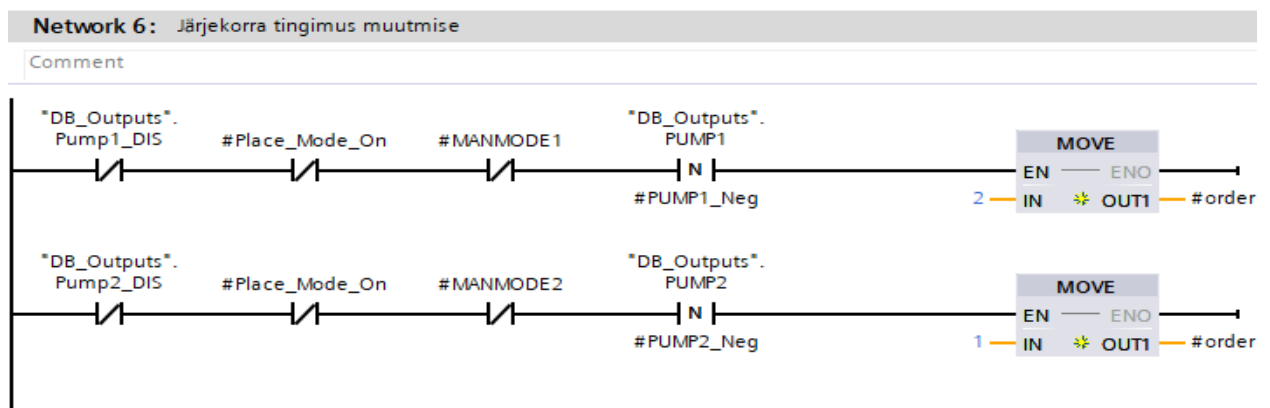
```

1 IF NOT #Night_Time AND (#Min_level AND NOT #MANMODE1 AND NOT #Place_Mode_On)
2   OR (#MANMODE1 AND #Man_StopP1) OR (#Place_Mode_On AND #Place_StopP1) OR (#Interlock1 OR #Force2) OR #DIS_Pump1
3 THEN
4   #Pump1_Start := FALSE;
5
6 END_IF;
7
8
9 IF NOT #Night_Time AND (#Min_level AND NOT #MANMODE2 AND NOT #Place_Mode_On)
10  OR (#MANMODE2 AND #Man_StopP2) OR (#Place_Mode_On AND #Place_StopP2) OR (#Interlock2 OR #Force1) OR #DIS_Pump1
11 THEN
12  #Pump2_Start := FALSE;
13
14
15
16 END_IF;
17 IF (#PUMP1_Neg) THEN
18  #order := 1;
19 IF (#PUMP2_Neg) THEN
20  #order := 2;
21 END_IF;
22
23 END_IF;
24
25

```

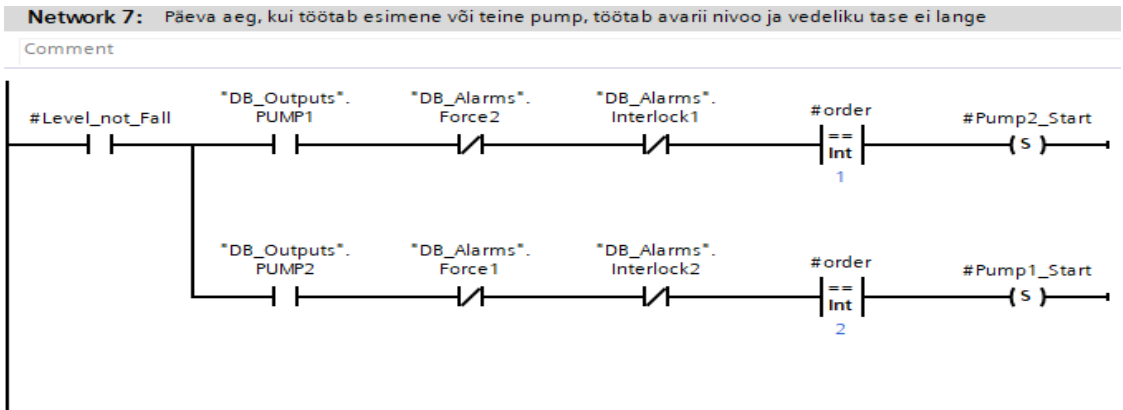
4.1.5 Joonis 1. Tingimised SCL keelel

Järjekorra muutmiseks oli ehitatud negatiivse signaaliga sturkuur. Pumbadel on töö järjekord. Kui pump lõppes oma tööd ära, siis selle pumba järjekord läheb teistele pumbadele. (vaata pilt 2)



4.1.5 Joonis 2. Järjekorra mutmine

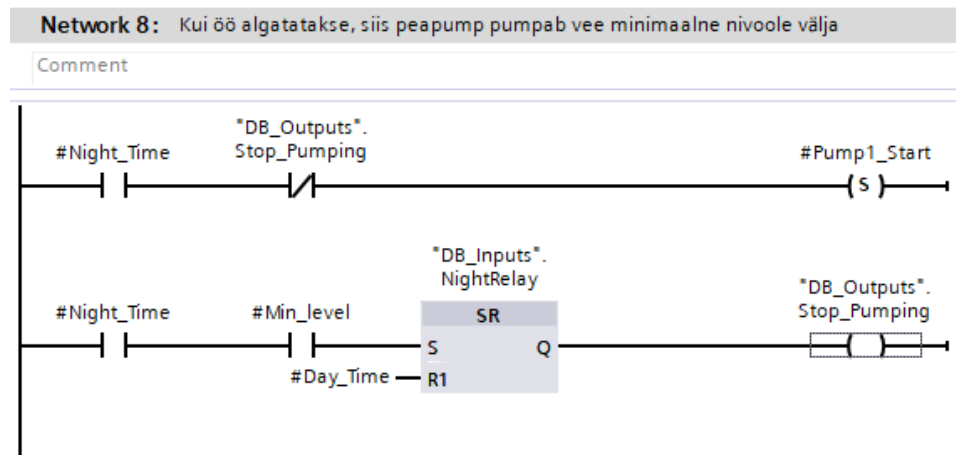
Edasi oli ehitatud tingimus, kui nivoo tõuseb „avarii nivooni“ ja ei lange. Siis hakkavad töötama kaks pumbad ja pumpama vedelikku minimaalse nivooni välja. (vaata pilt 3)



4.1.5 Joonis 3. Tingimus pumbade käivitamiseks

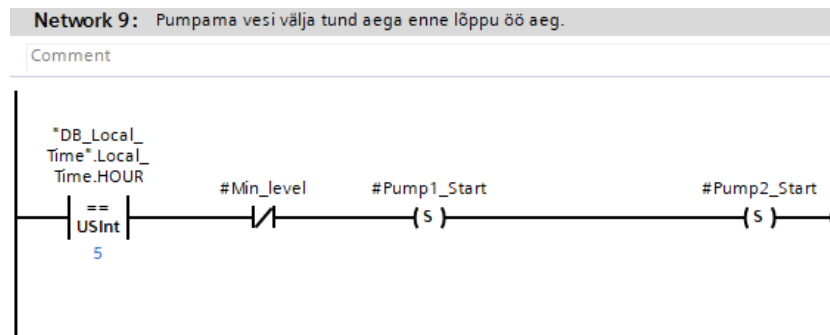
4.1.6 Kellaaja määratlemine

On vaja organiseerida vedeliku pumpamist, kui algab öö. Peafunktsioon läheb esimesele pumbale ja see pumpab vett minimaalse nivooni välja. (vaata pilt 1)



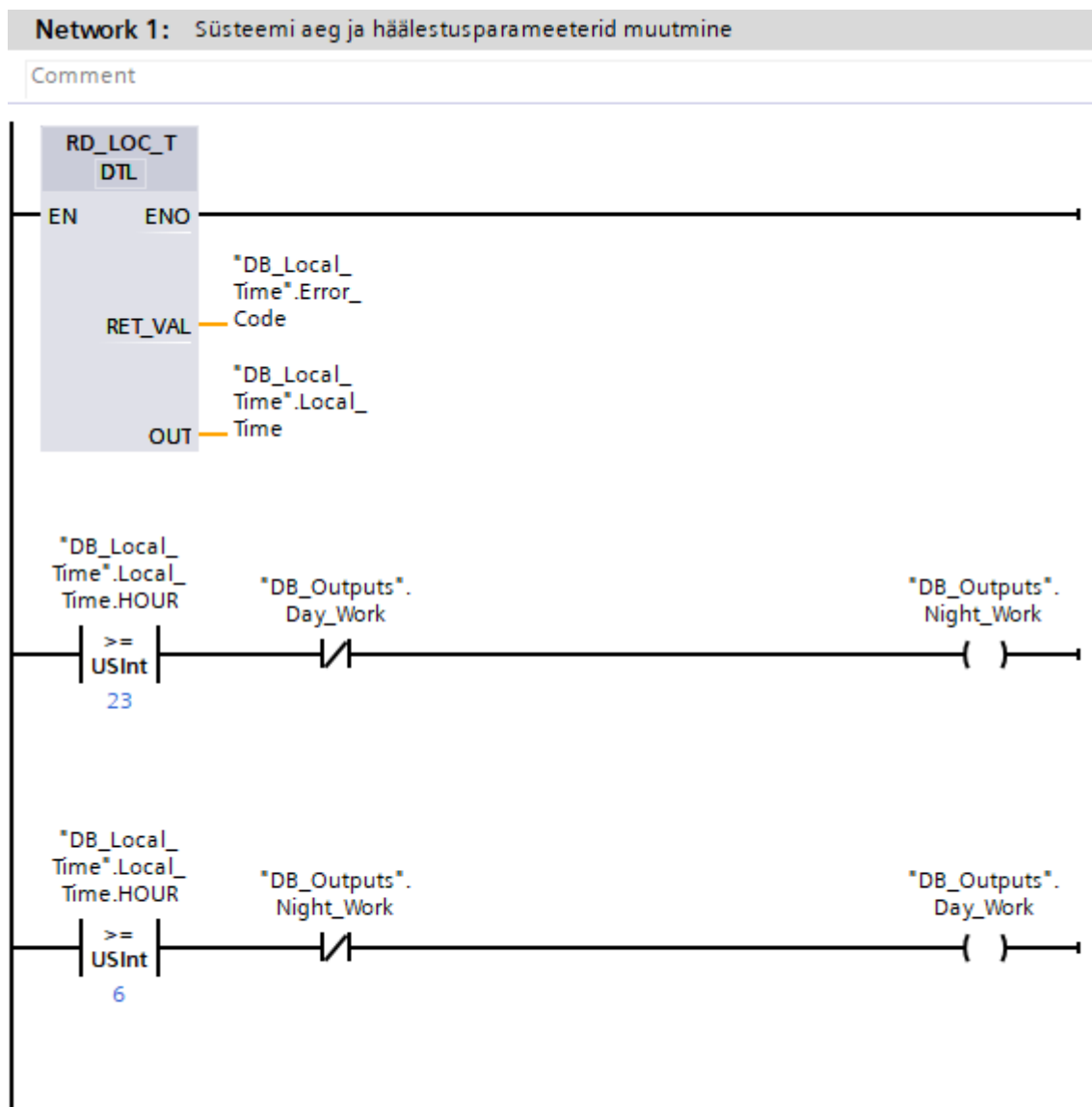
4.1.6 Joonis 1. Vedeliku pumpamine, kui öö

Üks tund enne hommikut, mõlemad pumbad peavad pumpama vett välja. (vaata pilt 2)



4.1.6 Joonis 2. Tingimus, et mõlemad pumpab pumpavad vett välja

Aega lugemiseks „FC_Local_Time[FC5]“ kasutatakse funktsioon, mis loeb kohalikku aja, mida edasi rakendatakse programmis. (vaata pilt 3)



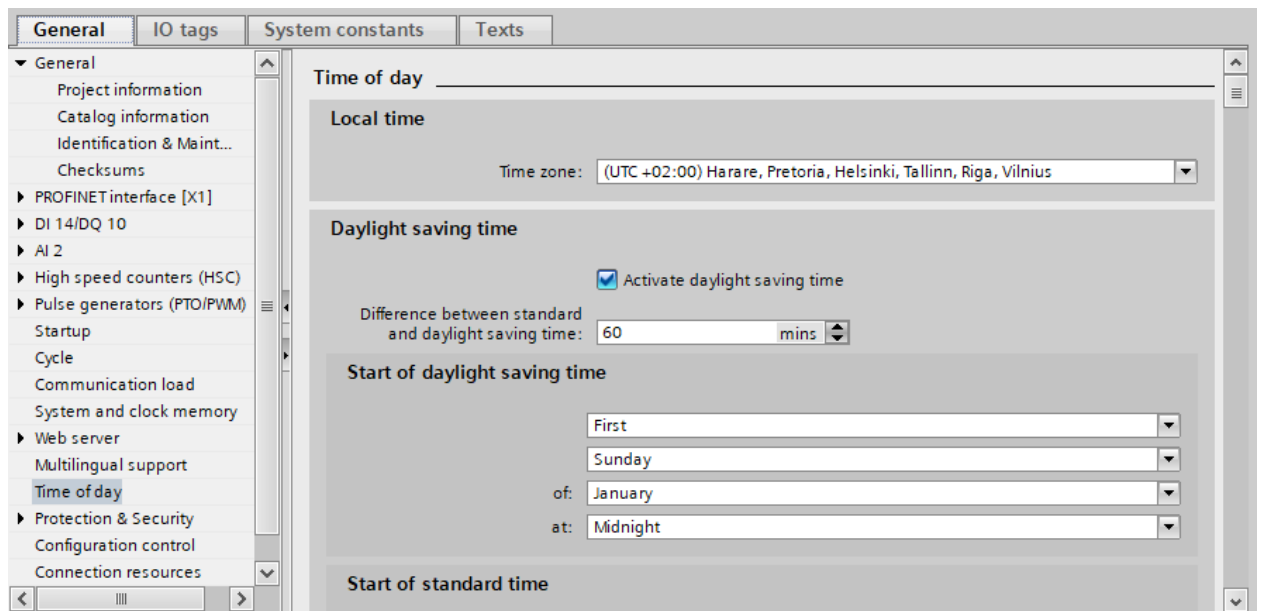
4.1.6 Joonis 3. Praegune aega lugemine

„RD_LOC_T“ funktsioon määrab praeguse kellaja. Kellaaeg määratakse „DTL“ väärtuse abil. „Data“ ploki „DB_Local_Time“ määratakse kõik sildid, mis viitavad ajale. (vaata pilt 4)

DB_Local_Time			
	Name	Data type	Start value
1	Static		
2	Local_Time	DTL	DTL#1970-01-01-4
3	YEAR	UInt	1970
4	MONTH	USInt	1
5	DAY	USInt	1
6	WEEKDAY	USInt	5
7	HOUR	USInt	0
8	MINUTE	USInt	0
9	SECOND	USInt	0
10	NANOSECOND	UDInt	0
11	Error_Code	Word	16#0

4.1.6 Joonis 4. RD_LOC_T funktsiooni sildid

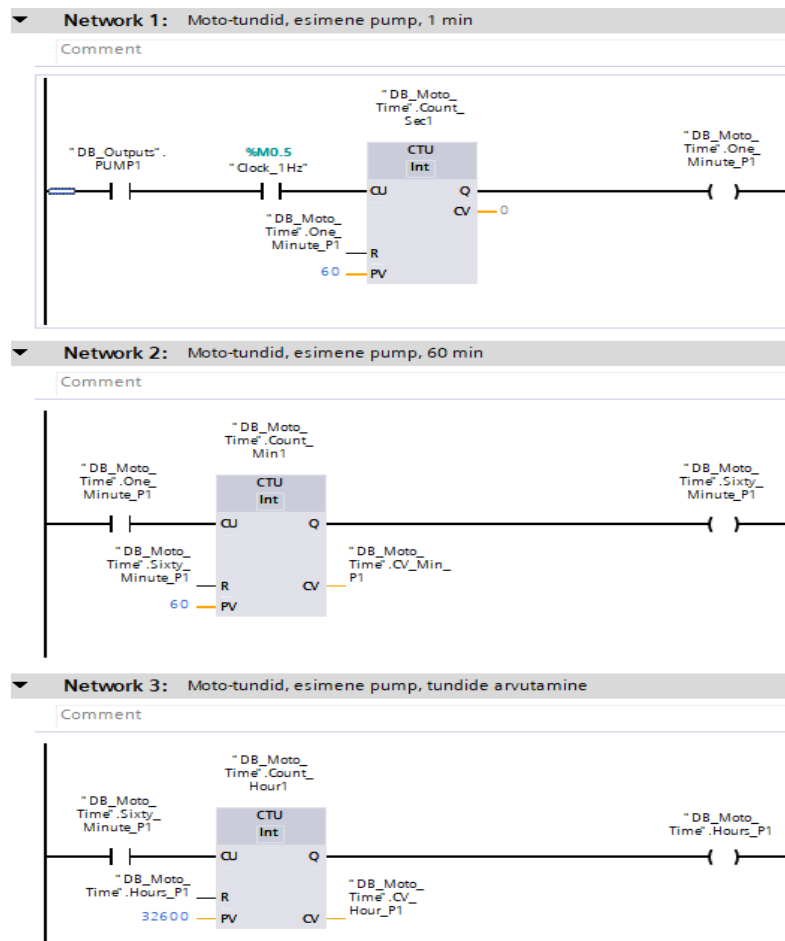
Sumuti, on vaja häälestada ajavööndit. Seda võib kontrolleris seadistada. [13] (vaata pilt 5)



4.1.6 Joonis 5. Ajavööndi häälestamine

4.1.7 Moto-tunnid

Samuti, programmis on vaja jälgida pumpade töötunde. Seda on vaja seadmete õigeaegse hoolduse jaoks. Arvutamine toimub arvestite ja sageduse sildi abil, mis annab signaal iga 1 sekund. (vaata pilt 1)



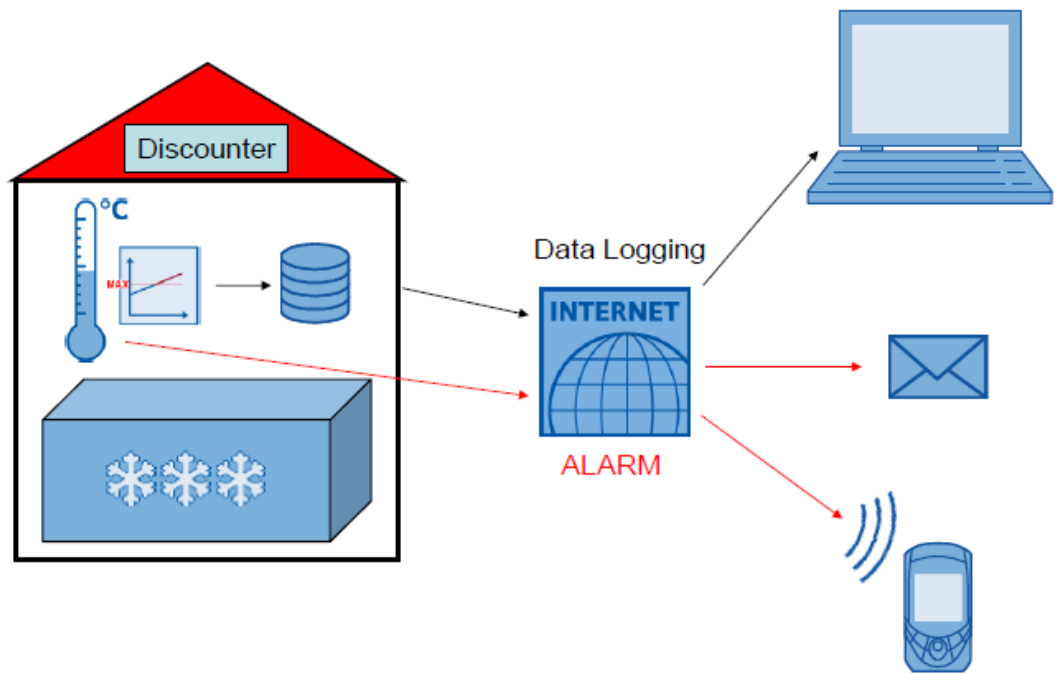
4.1.7 Joonis 2. Töötunnide arvutamine

4.2 DataLogs

Programmis kasutatakse datalogisid andmete kirjeldamiseks ja salvestamiseks.

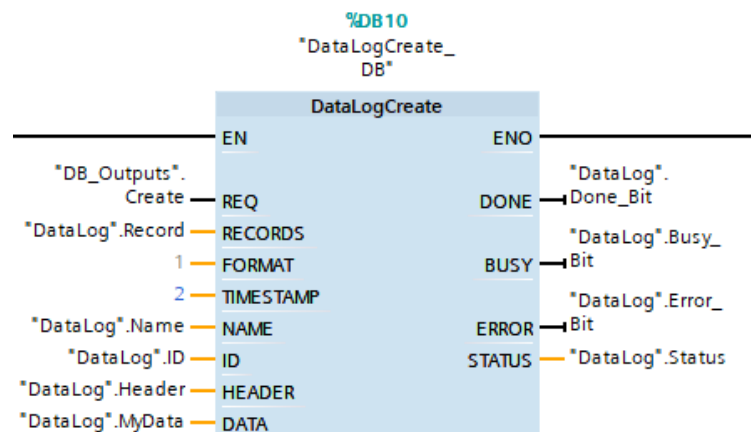
DataLog - see on organiseeritud struktuur, mis hoiab programmi andmeid. DataLog on aktiivselt kasutatud dünaamilisteks Interneti leheküljedeks. Andmeid tavaliselt hoitakse elektroonilises arvutisüsteemis. DataLogi põhielemendiks on tabel. Datalogi tabeli loomiseks on vaja määratleda selle struktuuri ja ridade arvu. Edasi, tabelisse sisestatakse ja muudetakse andmeid. DataLog loetakse loodud isegi siis, kui see on tühi.

Andmelogi juhiseid kasutatakse kasutajaprogrammis protsessiväärtuste salvestamiseks logifailidesse. Logifaile saab salvestada mälukaartile (MC) või CPU sisemällu (ainult S7-1200). Logifaile salvestatakse CSV-vormingus. [\[5\]](#) (vaata pilt 1)



4.2 Joonis 1. DataLogile andmete saatmine skeem

DataLogi loomiseks kasutatakse funktsioon „DataLogCreate“. (vaata pilt 2)



4.2 Joonis 2. DataLogCreate plokk

1	<	▼ Static				
2	<	■ Record	UDInt	4320		☑
3	<	■ Name	String	'MyDataLog'		☑
4	<	■ ID	DWord	16#0		☑
5	<	■ Header	String	'Level, Current ...'		☑
6	<	▼ MyData	Struct			☑
7	<	■ Level	UInt	0		☑
8	<	■ Current1	UInt	0		☑
9	<	■ Current2	UInt	0		☑
10	<	■ Done_Bit	Bool	false		☑
11	<	■ Busy_Bit	Bool	false		☑
12	<	■ Error_Bit	Bool	false		☑
13	<	■ Status	Word	16#0		☑
14	<	■ Set_Done_Bit	Bool	false		☑
15	<	■ Record_Haired	UDInt	2000		☑
16	<	■ Name_Haired	String	'HairedDataLog'		☑
17	<	■ ID_Haired	DWord	16#0		☑
18	<	■ Header_Haired	String	'Level Broken W...'		☑
19	<	▼ MyData_Haired	Struct			☑
20	<	■ Level_BW	Int	0		☑
21	<	■ Curr1_BW	Int	0		☑
22	<	■ Curr2_BW	Int	0		☑
23	<	■ Interlock1	Int	0		☑
24	<	■ Interlock2	Int	0		☑
25	<	■ Force1	Int	0		☑
26	<	■ Force2	Int	0		☑
27	<	■ Level_NF	Int	0		☑
28	<	■ Done_Bit_Haired	Bool	false		☑
29	<	■ Busy_Bit_Haired	Bool	false		☑
30	<	■ Error_Bit_Haired	Bool	false		☑
31	<	■ Status_Haired	Word	16#0		☑
32	<	■ Set_Done_Bit_Haired	Bool	false		☑

4.2 Joonis 3. Siltid DataLogiks

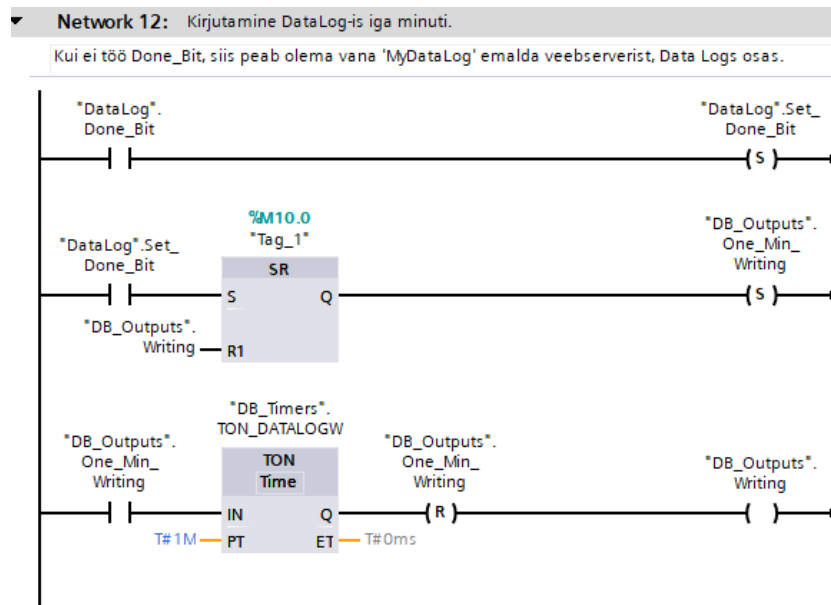
Selles funktsioonis määratletakse kirjendi format (FORMAT) ja kellaaeg (TIMESTAMP). Samuti, luuakse pealkirju, väärtuste nimesid ja kirjendite kogusid. DataLogi kirjutamiseks oli kasutatud REQ-signaal.

- 1) FORMAT - selle väärtus võib olema 0 (kirjendeid pole) ja 1 (kirjendid on CSV formadis).
- 2) RECORDS – kirjendite kogu.
- 3) TIMESTAMP – lokaalse aja kirjeldamine DataLogis.
- 4) NAME – DataLogi nimi.
- 5) ID – DataLogi ID (luuakse automaatselt).
- 6) HEADER – parameetride pealkirjad.
- 7) DATA – data struktuuri näitaja (1, 2, ..., n).

Oli realiseeritud kaks tüüpi DataLogid: tsüklilise ja sündmuse kirjendid. Tsükliline DataLog peab iga üks minut kirjutama nivoo ja volude väärtusi. Kirjutamine toimub loogilise süsteemi abil. (vaata pilt 3)

Andmete maht, mis on salvestatud DataLogis, sõltub saadaolevast ruumist või kasutatava protsessori sisemisest koormusmälust. CSV-faili maksimaalne suurus S7-1200 protsessori järgi 500 000 000 baiti, S7-1500 protsessori järgi 1 000 000 000 baiti. [\[5\]](#). [\[6\]](#)

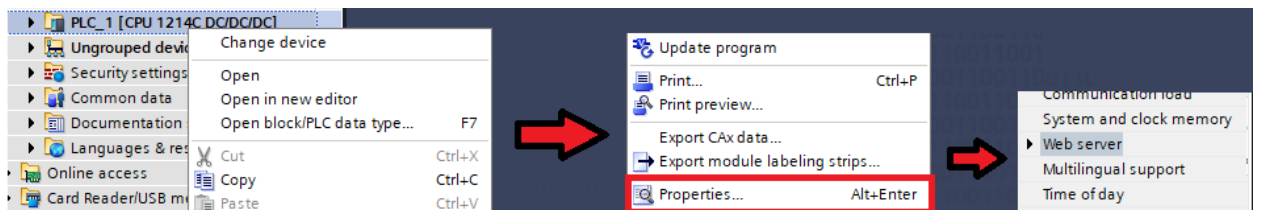
Iga minuti salvestamiseks kasutatakse käivitusviivitusega taimerit. "Done_Bit" näitab, et DataLogi tabel on edukalt loodud. (vaata pilt 4)



4.2 Joonis 4. Bit, mis määrab, et cvs. fail loodud

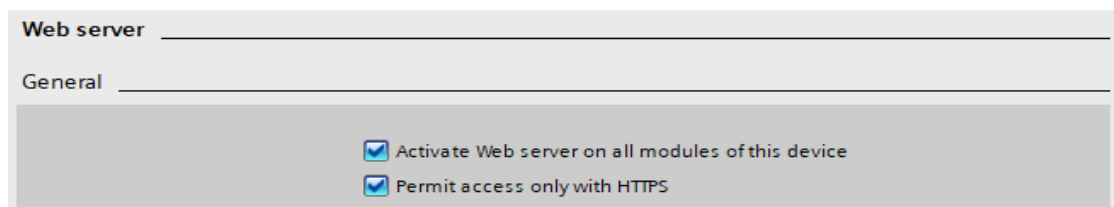
4.3 Veeb-serveri hälestamine

Pärast veeb-saidi loomisest, kontrollieris konfigureeritakse vajalikud plokid. Kontrolleri hälestustes valitakse „Web server“ link. (vaata pilt 1)



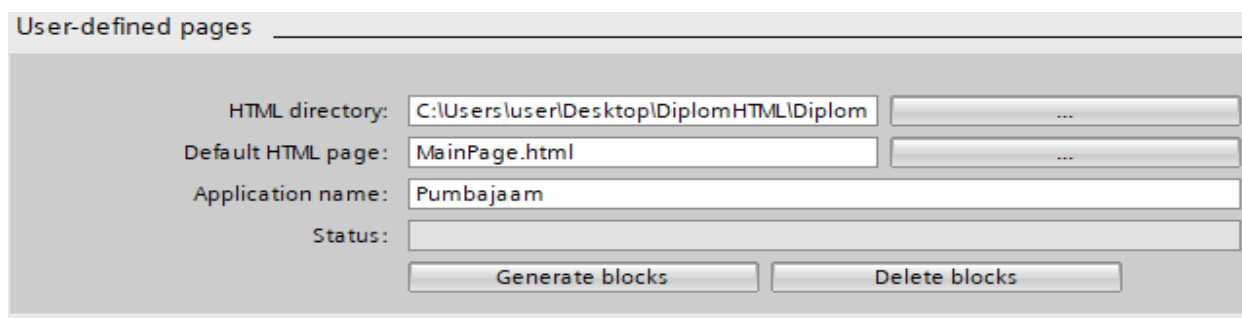
4.3 Joonis 1. Web-server link

Veeb-serveri aktiveerimiseks, valitakse „Enable Web server on this modul“. Turvalise juurdepääsu eest vastab „Permit access only with HTTPS“. (vaata pilt 2)



4.3 Joonis 2. Turvaline juurdepääs

Lingis „User-defined Web pages“ osutakse direktorium, kus asuvad veeb-server, algleht ja rakenduse nimi. Edasi nuppu „Generate blocks“ pressides, veeb-server genereerib (data)ploke. (vaata pilt 3)



User-defined pages

HTML directory: C:\Users\user\Desktop\DiplomHTML\Diplom ...

Default HTML page: MainPage.html ...

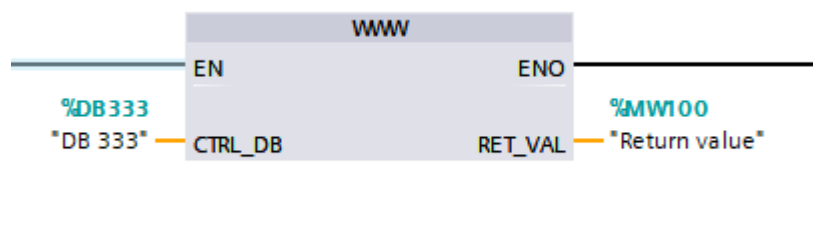
Application name: Pumbajaam

Status:

Generate blocks Delete blocks

4.3 Joonis 3. User-defined Web pages

Süsteemi funktsioneerimise tagab süsteemi funktsioon WWW, mis peab asuma peaplokis OB1. Initsialiseerimine toimub tsükliliselt OB1 plokis, siltide muudatuste visualiseerimiseks veeb-serveris. [4] (vaata pilt 4)



4.3 Joonis 4. Web-serveri plokk

4.4 Veebitehnoloogia kasutamine

Kogu projektis oli kasutatud järgnevad tehnoloogad: HTML, JQuery, JavaScript, CSS, SmoothieChart, JustGage+Raphael, AJAX, Bootstrap, DOM. [\[7\]](#), [\[15\]](#), [\[17\]](#), [\[20\]](#), [\[21\]](#)

- 1) HTML kasutatakse veebilehe loomiseks. See vajab brauserit, mis konverteerib hüpertexti ja kuvab lehe inimsõbralikus vormingus.
- 2) JavaScript - see on programmeerimise keel, spetsiaalselt interaktiivsete saitide loomiseks.
- 3) JQuery - see on Javaskripti teek, mõeldud joendamiseks, parandamiseks ja skriptimise lihtsustamiseks, kui töötatakse brauseris HTML.
- 4) CSS - see on kaskaadi stiilileht, mida kasutatakse HTML dokumenti välimuse kirjeldamiseks.
- 5) SmoothieChart – see on valmis JavaScripti šabloon graafiku loomiseks.
- 6) JustGage+Raphael - see on valmis JavaScripti šabloon animatsioonide loomiseks.
- 7) AJAX - see on serverile kohtlemine tehnoloogia ilma lehe uuendamiseta.
- 8) Bootstrap – see on tööriistakomplekt veebilehe loomiseks.
- 9) DOM – see on see on platvormist ja keelest sõltumatu programmeerimisliides, mis võimaldab programmidel ja skriptidel pääseda ligi HTML-, XHTML- ja XML-dokumentide sisule ning muuta selliste dokumentide sisu, struktuuri ja paigutust.

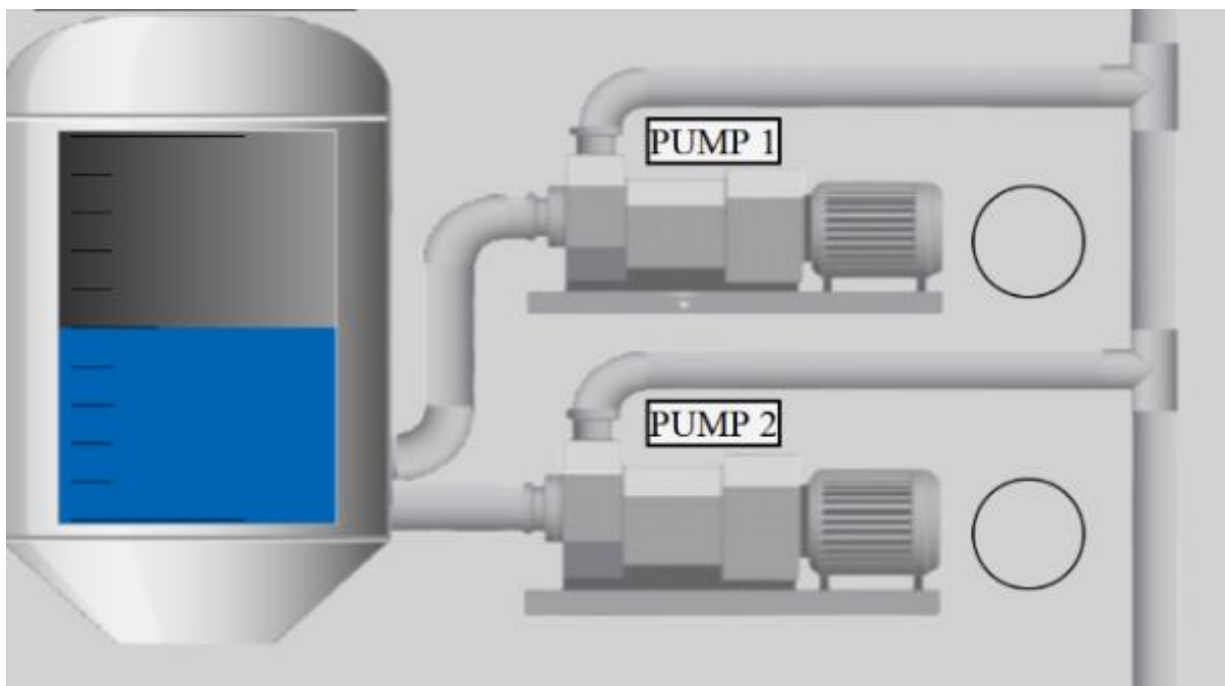
Tekstiredaktori abil, näiteks SublimeText, Notepad, ehitatakse HTML-fail, mis näitab brauseris kasutaja-veebilehe. See on nagu karkass veebilehele. Edasi on vaja formateerida HTML lehe kaskaadi stiililehe abil (CSS).

Veebilehel on vaja andmeid töödelda, suhelda kasutajaga, lisada efektid. Selleks kasutatakse JavaScripti. Kõige enam seda kasutatakse brauseri skriptikeelena, et lisada veebilehtedele interaktiivsust. Projektis Javaskript on kasutatud visuaalse animatsiooni jaoks, et lihtsustada programmi kasutamist. Näiteks, vedeliku nivoo pilti muutus, mis tekib JavaSkripti ja TiaPortali abil. JavaSkripti kood näeb selliselt välja:

```
1.   SetInterval(function(){
2.       var b = document.querySelector("#prbar");
3.       b.setAttribute("value", level);
4.   }, 500);
```

querySelector'i käsul valime esimene dokument nimega, mis on näidatud sulgudes. SetAttribute korraldusega valitakse atribuut, sel juhul „value“. Varem oli kasutatud AWP käsk, mis saab nivoo väärtuse kontrolleriist ja seda väärtust määratakse atribuudile. (vaata pilt 1) [\[3\]](#)

1. <!DOCTYPE html>
2. <!-- AWP_In_Variable Name='"DB_Outputs".Level' -->



4.4 Joonis 1. Pumbajaama pilt

Kasutaja peab teadma mis kell hetkel on. Selleks oli kirjutatud kood, mis määrab praeguse kellaaja.

```
1. setInterval(function(){
2.   var date = new Date();
3.   let curr_date = date.toLocaleString();
4.   /*           Plokk           id'ga           "current_date_time_block"           */
   document.getElementById('current_date_time_block').innerHTML = curr_date;
5. },0);
6.
```

Pumbajaama juhtiminst kontrollitakse ultraheli anduri abil, mis on paigaldatud mahutis teatud tasemel. Need andurid määratlevad, milline vee tase on ja, kui vaja, annab signaali, et hakata pumpama vett välja. Vee pumpamise tase muutub sõltuvalt kellaajalt. Juhtimisrežiimi muutmiseks kasutatakse navigatsiooni operatoori paneeli. Seda navigatsiooni paneeli kasutaja võib avada, pressides nuppu.[\[1\]](#), [\[2\]](#), [\[8\]](#)



4.4 Joonis 2. Esimine pumpa paneel

Selle funktsiooni abil võib kutsuma paneel ja kinni see, kus asutakse pumbajuhtimine ja indikatsiooni paneelid (vaata pilt 2). Samuti kalibreerida paneeli animatsioon (\$.fn.slideFadeToggle1). Skriptid nupudeks asuvad 464 reases.

```

1. $(function() {
2.   $('#pumpone').on('click', function() {
3.     if($(this).hasClass('selected1')) {
4.       deselect1($(this));
5.     } else {
6.       $(this).addClass('selected1');
7.       $('.pop1').slideFadeToggle1();
8.     }
9.     return false;
10.  });
11.  function deselect1(e1) {
12.    $('.pop1').slideFadeToggle1(function() {
13.      e1.removeClass('selected1');
14.    });
15.  }
16.
17.  $('.close1').on('click', function() {
18.    deselect1($('#pumpone'));
19.    return false;
20.  });
21. });
22.
23. $.fn.slideFadeToggle1 = function(easing1, callback1) {
24.   return this.animate({ opacity: 'toggle', height: 'toggle' }, 'fast', easing1, callback1);
25. };

```

Häirede kontrollimiseks oli loodud häirede tabel, häirede välja lülitamise võimalusega. Häirede mäaratlemiseks kasutatakse värvide indikatsioon. (vaata pilt 3)

Nivoo ei lange	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Nivoo on HH	<input type="checkbox"/>	välja lülitama
Nivoo juhe katkenud	<input type="checkbox"/>	välja lülitama
Pump 1 juhe katkenud	<input type="checkbox"/>	välja lülitama
Pump 2 juhe katkenud	<input type="checkbox"/>	välja lülitama
Interlock 1	<input type="checkbox"/>	välja lülitama
Interlock 2	<input type="checkbox"/>	välja lülitama
Force 1	<input type="checkbox"/>	välja lülitama
Force 2	<input type="checkbox"/>	välja lülitama
HÄIREDE TAGASTUS	HÄIRETE EMALDAMINE	

4.4 Joonis 3. Häirede tabel lülitamise võimalusega

```

1. $(document).ready(function(){
2.     $("#reserr").click(function(){
3.         name = 'DB_Inputs'.Res_all_Err';
4.         $.post("signals/ResetErrors.html",encodeURIComponent(name)+'= 1');
5.     });
6. });
7. var toggled1 = false;
8. function toggle1(){
9.     if(!toggled1){
10.         toggled1 = true;
11.         document.getElementById("dis1").style.backgroundColor = "green";
12.         $.post("signals/DIS_Level_NF.html",encodeURIComponent(name)+'= 1');
13.         return;
14.     }
15.     if(toggled1){
16.         toggled1 = false;
17.         document.getElementById("dis1").style.backgroundColor = "";
18.         $.post("signals/DIS_Level_NF.html",encodeURIComponent(name)+'= 0');
19.         return;
20.     }

```

Selle koodi abil, seda koodi kasutades saad nupu ühe klõpsuga peale jätta, järgmine kord vajutades lülitub see välja. POST päringute abil, saatakse 1 või 0 väärtuse kontrolleriil.

```

1. setInterval(function(){
2.     var s = new XMLHttpRequest();
3.     s.open("GET", 'signals/Level_Not_Fall.html', true);
4.     s.send();
5.     s.onreadystatechange = function(){
6.         if (s.readyState != 4) {
7.             return
8.         }
9.         if (s.status === 200) {
10.            if(s.responseText==1){
11.                document.getElementById("err1").style.backgroundColor = "red";
12.            }
13.            else{
14.                document.getElementById("err1").style.backgroundColor = "";
15.            }
16.        } else {
17.            console.log('err', s.responseText)
18.        }
19.    }
20. },1000);

```

Häirede välja lülitamise saab teha kahe sündmuse käitlejate abil. Esimene käitleja vastab nupu inkatsiooni eest ja saadab POST päringut kontrollerrisse. Teine käitleja vastab häirede indikatsiooni eest. Saadakse POST päring kontrollerrisse ja tuleb tagasi GET päringu abil. Kui häired ilmuvad, siis indikatsioon lünk saab punaseks. [4], [14]

Ülesannel oli võimalus muuta parameetreid distantsilt. Tabelis on kirjutatud kõik peaväärtused, samuti oli kujutatud need praegused parameetrid ja väli täiteks. Parameetride muutmise võimalik POST päringute abil. Formil määratakse päringu meetod, input lüngias kirjutatakse väärtus, mis hakkab edastatakse kontrollerris. (vaata pilt 4)

Minimaalne nivoo	10%	<input type="text"/>	Saatma
Maksimaalne nivoo	90%	<input type="text"/>	Saatma
High-High nivoo	98%	<input type="text"/>	Saatma
Vool 1 maksimaalne	280A	<input type="text"/>	Saatma
Vool 2 maksimaalne	280A	<input type="text"/>	Saatma
Vool 1 minimaalne	20A	<input type="text"/>	Saatma
Vool 2 minimaalne	20A	<input type="text"/>	Saatma
Timer nivoo ei lange	T#1m_40s	<input type="text"/>	Saatma
Timer Interlock	T#10s	<input type="text"/>	Saatma
Timer Force	T#10s	<input type="text"/>	Saatma

4.4 Joonis 4. Peaväärtuseid tabel muutmise võimalusega

```

1. <tr>
2.     <td>Minimaalne nivoo</td>
3.     <td class="param">
4.         <form method="post">
5.             <input name="'DB_Settings'.Level_Min' type="text"/>
6.             <button type="submit">Saatma</button>
7.         </form>
8.     </td>
9. </tr>
10.
```

5. VEEB-SERVER

Veeb-disaini kontekstis, termini „veeb-lehed“ kasutatakse nagu dokument, mis asub ülemaailmases veebis (WWW), mida saab kutsuda veeb-serverist veeb-brauseri abil, konkreetse URL väärtusega.

5.1 Uuendamine taustrežiimis

Veeb-lehe uuendamiseks taustrežiimis on vaja kasutada AJAX tehnoloogiat.

AJAX – („Asynchronous Javascript And Xml“) see on serverile kohtlemis tehnoloogia ilma lehe uuendamiseta. Selle tõttu vähendatakse kajaeg ja veeb-rakendus töötab nagu desktop. AJAX-i abil võidakse: [\[7\]](#), [\[20\]](#)

1. Otseotsingus
2. Dünaamilise andmete laadimises
3. Vähendatakse serveri koormus

AJAX-i päringu süntaksis on järmine vorm:

```
1. setInterval(function(){
2.   var name = new XMLHttpRequest();
3.   name.open("GET", 'link/link.html', true);
4.   name.send();
5.   name.onreadystatechange = function(){
6.     if (name.readyState != 4) {
7.       return
8.     }
9.     if (name.status === 200) {
10.    }
11.  else {
12.    console.log('err', name.responseText)}
13.  }
14. },500);
15.
```

XMLHttpRequest päringule, võib osutakse mis päring kasutatke POST või GET, kuhu tuleb see päring ja et URL loob ajalooolendisse uue kirje. Edasi if funktisooni abil kontrollitakse päringu staatus.

Samuti signaali töödeldamiseks kasutatakse JavaSkripti funktsioon, mille nimi on setInterval. Selle funktsiooni abil saab kogu aeg töödelda tuleneva signaali kontrollierist, ilma veeb-lehe uuendamiseta. setInterval funktsiooni süntaksis on järmine vorm:

```
1. setInterval(function(){
2. $.get("signals/name.html", function(result){
   var_name = parseInt(result.trim());
   //Teie kood
   });
3. },500} ;
```

On olemas kaks lahenduse varianti, kuidas kontrollis signaale töödelda. Mõlemas variandis kasutatakse funktsiooni `setInterval`, aga esimesel variandil kasutatakse proekteerimis-šabloon `XMLHttpRequest+HttpRequest+setInterval` + kood (`HttpRequest` vastab `POST` ja `GET` päringu eest). Teisel variandil kasutatakse funktsiooni `setInterval` + `jQuery`+kood. Need variandid töötavad samavõrra.

Kumb on parem selle projekti loomiseks - raske küsimus, sest ei saa projekterimis-šabloon ja funktsioon võrrelda omavahel. Need versioonid töötavad samamoodi, erinevus on ainult koodi pikkuses.

5.2. Animatsiooni JavaScripti abil

JavaScripti abil saab teha animatsiooni, mida ei saa realiseerida CSS abil. Näiteks liikumine või canvas animatsioonid. Operaatorite jaoks oli tehtud seadme pilt. Sellel pildil asub vedeliku nivoo skaala. See skaala on animeeritud, et kontrollida nivoo ja voolude väärtused reaalsajas.

Animatsiooniks oli valitud JustGage. See arhiiv on vabas juurdepääsus. [\[17\]](#) (vaata pilt 1)



5.2 Joonis 1. JustGage

On vaja luua `<div>` silt, unikaalse id-ga, laius ja kõrgus:

```
1. <div id="gague" style="width:400px; height:320px"></div>
```

Et käivitada animatsiooni, on vaja teha JustGage päringut:

```

1. var gauge = new JustGage({
2.     id: "gauge", // the id of the html element
3.     value: 100,
4.     min: 0,
5.     max: 300,
6.     levelColors: ["EBEB00", "EBEB00", "EBEB00"],
7.     title: "PUMP 1 VOOL"
8. });

```

JustGage siltil osutakse id, väärtus, minumum ja maksimum, RGB värveid ja pealkiri. Samuti on vaja saada kätte analoogsignaali kontrollerist, AWP korraldusele skripti abil, panna analoogväärtuse JustGage-sse. [\[17\]](#)

```

1.     var curr=parseInt(:"DB_Outputs".Current1:);
2.     setInterval(() => {
3.         $.get("signals/Current1.html",function(restult){
4.             curr=parseInt(restult.trim());
5.         });
6.         gauge.refresh(curr);
7.     }, 500);

```

Nivoo animatsiooniks oli kasutatud <progress></progress> silt. (vaata pilt 2)



5.2 Joonis 2. Nivoo animatsioon

AWP korralduse abil nivoo väärtus kontrollerist edastatakse veeblehele. Edasi querySelector edastab edenemisribal seda väärtust ja setAttribute edastab väärtust „value“ attributile, mis asub progress sildil.

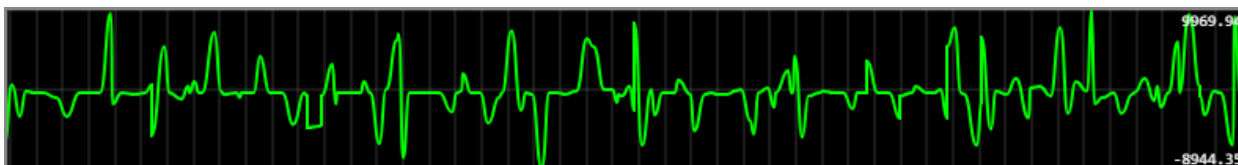
```

1. <div class="progress">
2.     <progress id="prbar" max="100" value="50" ></progress>
3. </div>
4. setInterval(function(){
5.     var b = document.querySelector("#prbar");
6.     b.setAttribute("value", level);
7. }, 500);

```

Nivoo muutuste jälgimiseks oli kasutatud teek, mida nimetatakse SmootheiChart. (vaata pilt 3)

Graafiku ühendamiseks on vaja SmoothieChart kaasata HTML failis ja kirjutada graafikut sablooni sisse. [15] [16]



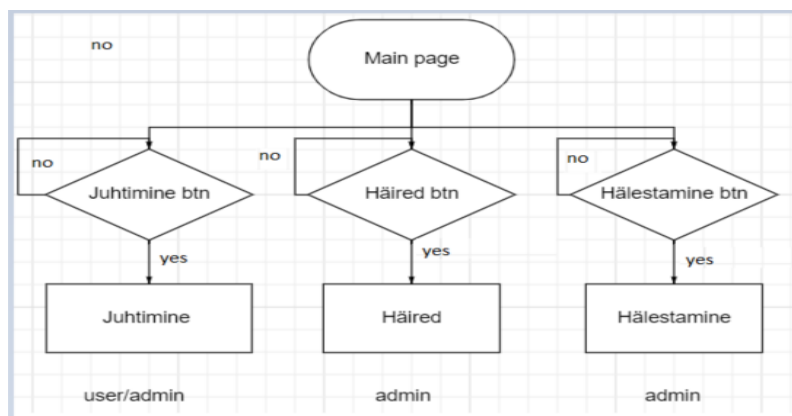
5.2 Joonis 3. SmoothieChart graafik

Koodis määratakse pikkus ja laius, rgb tagapaneeli värv ja joone värv. Koodis ka kirjutatakse maksimaal kõrgepiir ja madalapiir, voolkiirus.

```
1. <canvas id="smoothie-chart" width="1000" height="285"></canvas>
2. var chart = new SmoothieChart({grid:{fillStyle:'#ffffff'},labels:{fillStyle:'#000000'},
  millisPerPixel:45,maxValueScale:0.8,minValueScale:0.8,scaleSmoothing:0.012,minValue:0,maxValue:100,timestampFormatter:SmoothieChart.timeFormatter}),
3. canvas = document.getElementById('smoothie-chart'),
4. series = new TimeSeries();
5. chart.addTimeSeries(series,
  {lineWidth:2,strokeStyle:'#ff0000',fillStyle:'rgba(0,128,255,0.30)'});
6. chart.streamTo(canvas, 973);
```

5.3 Veebilehed

See on valimise stardileht, kus võib valida kasutaja lehe. Operatoril on ainult üks lehe variant, see on Juhtimise veebileht. Plokkskeemil on näidatud kuhu võib üle minna ja kes seda võib teha. Juhtimislehe võib külastada iga kasutaja, aga juhtida saab ainult administraator. (vaata pilt 1, 2)



5.3 Joonis 1. Main page plokkskeem

Pealehel võib valida töölehed.



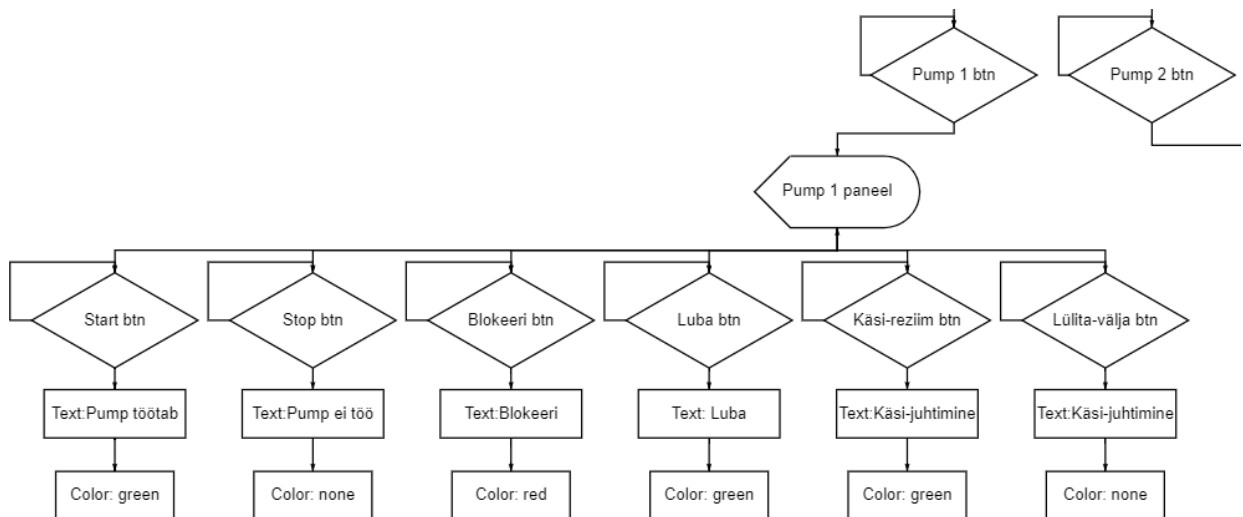
© Jaroslav Savellitsev

5.3 Joonis 2. Main page

Juhtimislehel toimuvad kõik visualiseerimisprotsessid. Lehel visualiseeritakse peaparametrid.

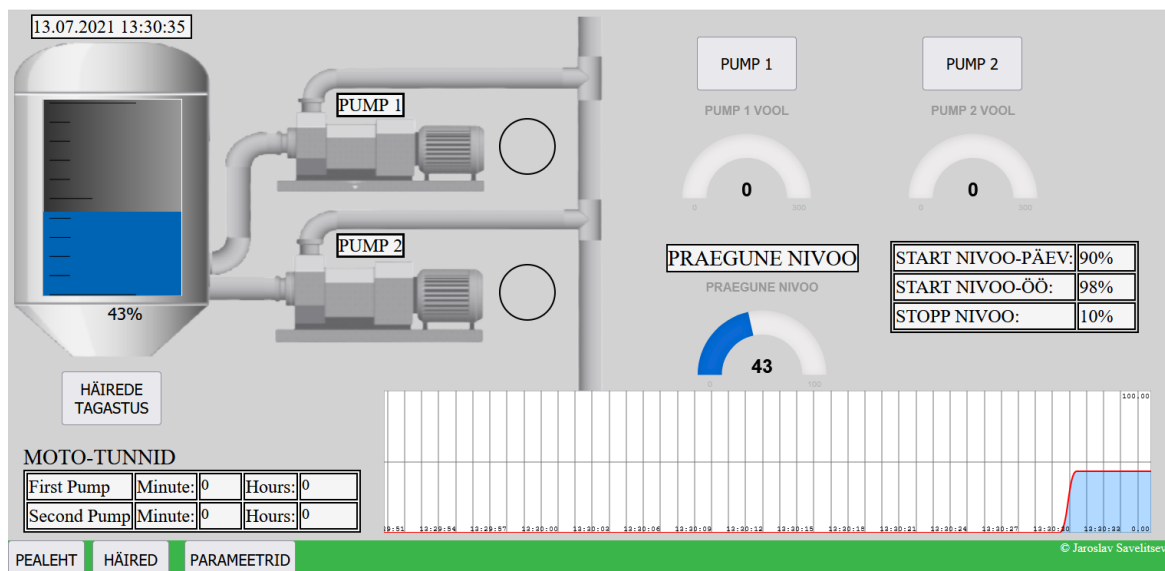
Siin operaator võib kontrollida pumbajaama peaparametrid.

Veeb-lehel on olemas kaks nupud operaatori paneelide kuvamiseks, pumbade juhtimiseks. (vaata pilt 3,4,5)



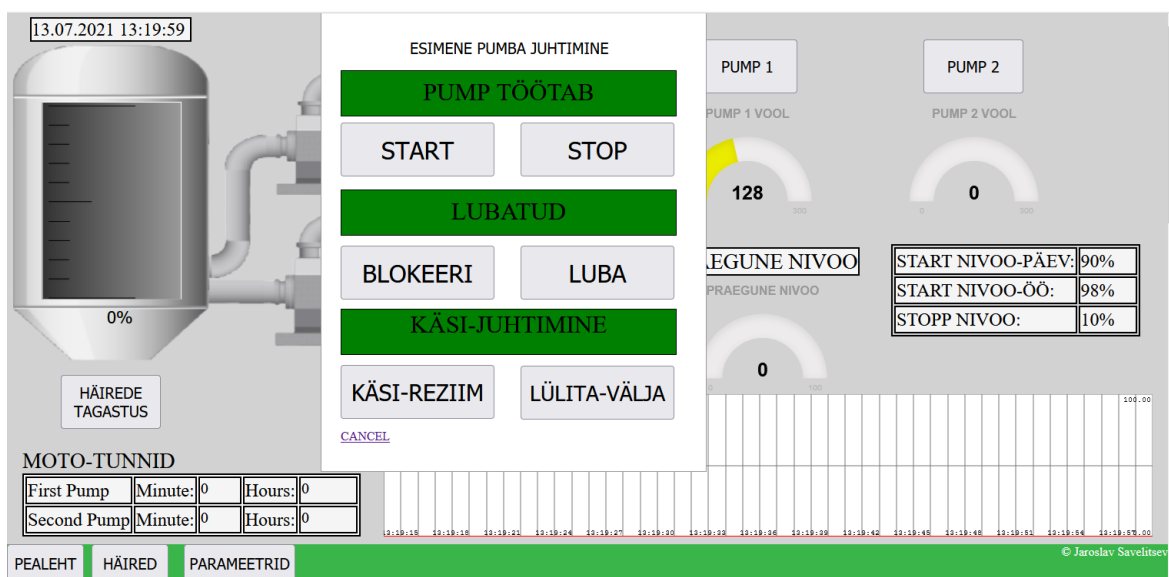
5.3 Joonis 3. Esimene pumba paneel plokk skeem

Siin on ette nähtud juhtimisleht põhivisualisatsiooniga.



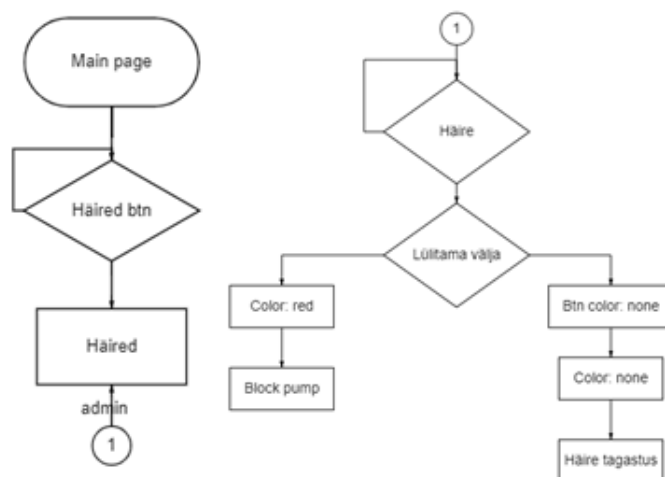
5.3 Joonis 4. Juhtimine leht

Siin ettenäitud juhtimisleht pumba juhtimis paneeliga.



5.3 Joonis 5. Juhtimine leht, kui ilmunud operaatori paneel

Häirede lehel võib häireid jälgida ja lülitada neid välja. Tabel töötab selle algoritmi järgi:



5.3 Joonis 6. Häirede lehe üleminek

Siin ette näitud juhtimisleht häirede tabeliga.

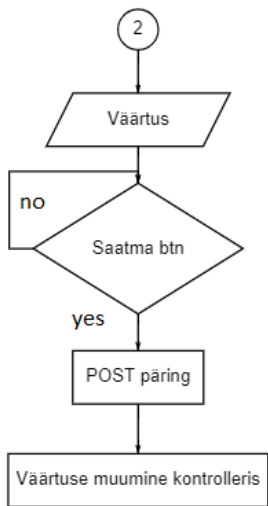
PEAB OLEMA KAHEKORDNE NUPPU PRESSIMA, SEST REALISEERITUD KAITSE JUHUSLIKU PRESSIMINE EEST

Nivoo ei lange	<input type="button" value="Valja lülitama"/>
Nivoo on HH	<input type="button" value="Valja lülitama"/>
Nivoo juhe katkenud	<input type="button" value="Valja lülitama"/>
Pump 1 juhe katkenud	<input type="button" value="Valja lülitama"/>
Pump 2 juhe katkenud	<input type="button" value="Valja lülitama"/>
Interlock 1	<input type="button" value="Valja lülitama"/>
Interlock 2	<input type="button" value="Valja lülitama"/>
Force 1	<input type="button" value="Valja lülitama"/>
Force 2	<input type="button" value="Valja lülitama"/>
HÄIREDE TAGASTUS	HÄIRETE EMALDAMINE

PEALEHT
JUHTIMINE
PARAMEETRID
© Jaroslav Saveltsev

5.3 Joonis 7. Häirede leht

Parameetrite häälestamise lehel võib muuta siltide väärtusi ilma kontrollerrisse sissepääsuta. Seda võib teha POST päringu abil.



Minimaalne nivoo	10%	<input type="text"/>	Saatma
Maksimaalne nivoo	90%	<input type="text"/>	Saatma
High-High nivoo	98%	<input type="text"/>	Saatma
Vool 1 maksimaalne	280A	<input type="text"/>	Saatma
Vool 2 maksimaalne	280A	<input type="text"/>	Saatma
Vool 1 minimaalne	20A	<input type="text"/>	Saatma
Vool 2 minimaalne	20A	<input type="text"/>	Saatma
Timer nivoo ei lange	T#1m_40s	<input type="text"/>	Saatma
Timer Interlock	T#10s	<input type="text"/>	Saatma
Timer Force	T#10s	<input type="text"/>	Saatma

PEALEHT JUHTIMINE HÄIRED © Jaroslav Saveltsev

5.3 Joonis 8. Parameetride häälestamine leht

6. PROJEKTI TURVALISUS

Projekti muutuste kaitsmiseks vastab funktsioon „No access“. See tähendab, et kõigil Tia Portal'i kasutajatel puudub jurdepääs funktsioonidele. Projekt on parooliga kaitstud ja programm on kaitstud koodimuutustest. [19] (vaata pilt 1)

Access level	Access			Access permi...
	HMI	Read	Write	Password
<input type="radio"/> Full access (no protection)	✓	✓	✓	*****
<input type="radio"/> Read access	✓	✓		
<input type="radio"/> HMI access	✓			
<input checked="" type="radio"/> No access (complete protection)				

No access (complete protection):
TIA Portal users and HMI applications will not have access to any functions.

Mandatory password:
For full access, TIA Portal users need to enter the "full access" password.

Optional password:
A "read access" password can be defined for read access to all functions.
For access by HMI applications, an "HMI access" password can be defined.

Some HMI devices do not support all possible characters. If you want to access the PLC from an HMI device, use only the standard characters. Please refer to the documentation of the device.

6. Joonis 1. Access level leht

Samuti oli kaitstud kõik projektide plokid erinevate paroolidega. (vaata pilt 2)

Know-how protection

The block is not protected.

Protection

Write protection

Write protection

The password for write protection has been defined.

Change password

Copy protection

No binding

Serial number is inserted when downloading to a device or a memory card.

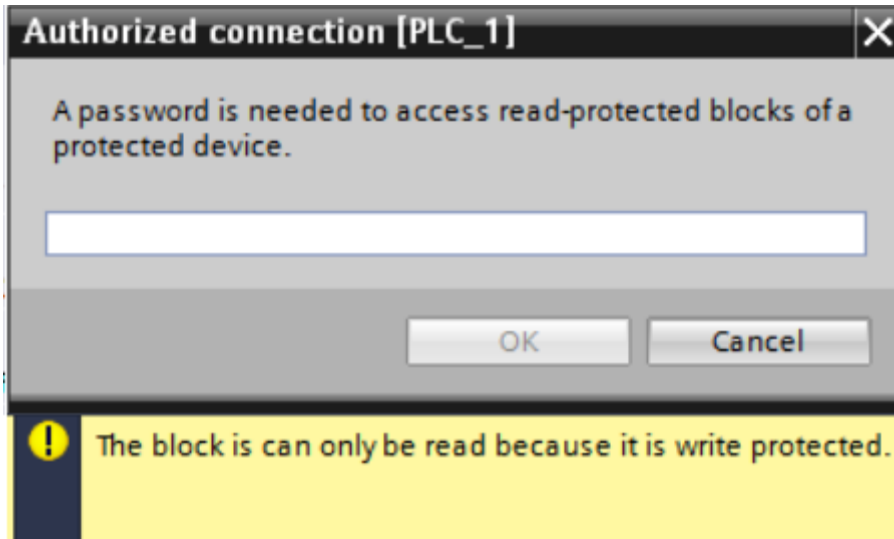
The password for copy protection has not been defined.

Define password

Enter serial number

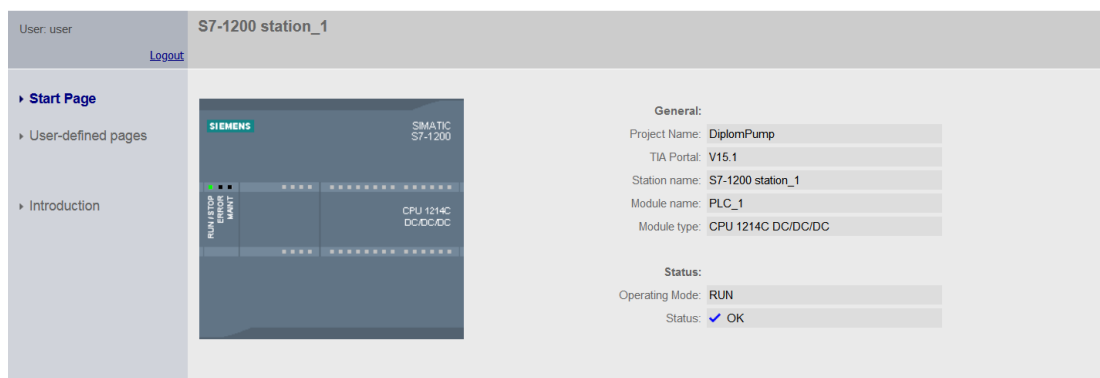
6. Joonis 2. Projekti kaitsmine

Nüüd projekti muutmisel kuvatakse aken parooli sisestamise nõudega. Programmploki üleval kuvatakse hoiatus, et programm on muutmistest kaitstud. (vaata pilt 3)



6. Joonis 3. Hoiatus, et programm on muutmise kaitstud

Veebiserveri kaitse tagamiseks on kasutusele võetud täiendav piiratud õigustega kasutaja. Ta saab lehte vaadata ainult visualiseerimisega, kuid ta ei saa seal midagi muuta. (vaata pilt 4)



6. Joonis 4. Web-serveri leht, kui on piiratud õigustega kasutaja

7. KOKKUVÕTE

See töö on pühendatud automaatsete sõlmede visualiseerimismeetodi loomisele. See töö valiti mitmel põhjusel:

1) Esiteks ABC Service'ilt tellimuse saamine visualiseerimissüsteemi arendamiseks ja pumbajaama programmi koostamiseks.

2) Teiseks on projekt ainulaadne selle poolest, et turul valitseva pooljuhtide nappuse ja tehniliste ülesannete esinemissageduse tõttu väikeste tehniliste sõlmede visualiseerimiseks on seda tüüpi visualiseerimine aitab kliendil säästa aega visualiseerimissüsteemi arendamiseks, kuna suur visualiseerimissüsteeme nagu SCADA on kahjumlik väike projektideks.

Nendel juhtudel on soovitatav kasutage kontrolleri sisseehitatud funktsioone, nagu "Veebiserver". Kuna kõik sellised objektid asuvad tavaliselt kaugetes kohtvõrkudes, siis autor ei saanud seda autorile, et kaitsta võrku volitamata juurdepääsu eest. Kaitse pakkus kliendi IT-osakond. Lõputöö koosneb 3 osast:

1) Esimene osa räägib ülesannest asetusest ja teooriast.

2) Teises osa kirjeldatakse pumbajaama programmi.

3) Kolmas osa sisaldab veebiserveri programmi ja projekti turvalise.

Projekti loomisel kasutus autor SIEMENSi maanualide kohandatud lehtede loomisel, kuid sellest ei piisanud potentsiaali avamiseks, see kontrollimeetod. Kuna vaikimisi värskendamine on sunnitud ja täielikult, mis on kasutajale ebamugav, samuti on lehe enda välimus vale. Olemasolevate probleemide kõrvaldamiseks kasutas autor kolledžist saadud teadmisi ning uuris ja tutvustas ka järgmisi olemasolevaid tehnoloogiaid: HTML, CSS, JQuery, JavaScript, SmoothieChart, JustGage + Raphael, AJAX, Bootstrap, DOM.

Pärast nende tehnoloogiate kasutuselevõttu on kasutaja lehe vaade paranenud, lehe värskendamine hakkas toimuma taustal, see sai võimalikuks jälgida näitu graafikute ja valmismallide abil.

Palju aega kulus materjali õppimisele, veebiserveri abil visualiseerimissüsteemi seadistamisele ja loomisele, kuid lõpuks sai töö edukalt lõpule viidud, kõik eesmärgid saavutatud.

Tulevikus saab projekti täiustada, teabekogumiseks ja salvestamiseks andmebaasi, samuti on võimalik ühendada kasutajalehti, võttes kasutusele uue kontrolleri S7-1500 seeriast ja tutvustades uut teek SIEMENS-st "S7 Framework". Selle teegi lõi SIEMENS, eriti veebiserveri jaoks. Teek võimaldab lahendada paljusid sellega seotud probleeme lehe värskendamise, serveri sooritusvõime, kuvaelementide loomise ja kontrolleri diagnostika.

7.1 SUMMARY

This work is dedicated to creating a method for visualizing automatic nodes. This work was chosen for several reasons:

1) Firstly, receiving an order from ABC Service for the development of a visualization system and the preparation of a pumping station program.

2) Second, the project is unique in that due to the prevailing shortage of semiconductors in the market and the frequency of technical tasks for visualizing small technical assemblies, this type of visualization saves the customer time to develop a visualization system. In these cases, it is recommended to use the controller's built-in functions, such as "Web Server". Because all such objects are usually located in remote LANs, the author did not send it to the author to protect the network from unauthorized access. Protection was provided by the customer's IT department. The dissertation consists of 3 parts:

1) The first chapter talks about general theory.

2) The second chapter describes the pumping station program.

3) The third chapter contains the web server program.

When creating the project, the author used SIEMENS to create custom pages for landscapes, but this was not enough to unlock the potential, this control method. Because the default update is forced and complete, which is inconvenient for the user, the appearance of the page itself is also incorrect. To solve the existing problems, the author used the knowledge gained from the college and also researched and introduced the following available technologies: JQuery, JavaScript, SmoothieChart, JustGage + Raphael, AJAX, Bootstrap, DOM.

After the introduction of these technologies, the user's page view has improved, the page has started to be updated in the background, it has become possible to monitor the display with graphs and ready-made templates.

It took a long time to learn the material, set up and create a visualization system using a web server, but in the end the work was completed successfully, all goals and objectives achieved.

In the future, the project can be improved to collect and store information in a database, and user pages can be connected by introducing a new controller from the S7-1500 series and introducing a new library from SIEMENS, the "S7 Framework". This library was created by SIEMENS, especially for the web server. The library solves many of the problems associated with updating the page, server performance, creating display elements, and controlling the controller.

8. KIRJENDUSED NIMEKIRJA

1) Creating User-Defined Web Pages for S7-1200/S7-1500 [WWW].

https://cache.industry.siemens.com/dl/files/496/68011496/att_1018669/v1/68011496_S7-1200_1500_Webserver_DOC_v4_en.pdf (03.10.2021)

2) Basics on Creating HTMLs for SIMATIC CPUs [WWW].

https://cache.industry.siemens.com/dl/files/496/68011496/att_14832/v1/68011496_html_basics_for_simatic_cpus_en.pdf (03.10.2021)

3) Examples for the SIMATIC S7-1200 / S7-1500 Web Server [WWW].

https://cache.industry.siemens.com/dl/files/496/68011496/att_997448/v1/68011496_Examples_for_S7WebServer_DOC_v3_en.pdf (03.10.2021)

4) Siemens S7-1200 Web Server Tutorial - From Getting Started to HTML5 User Defined Pages [WWW]. - Tim Jager

<https://www.dmcinfo.com/latest-thinking/blog/id/8567/siemens-s7-1200-web-server-tutorial--from-getting-started-to-html5-user-defined-pages> (03.10.2021)

5) Process Data Acquisition and Monitoring [WWW].

https://cache.industry.siemens.com/dl/files/156/64396156/att_992804/v1/64396156_S7-1200_DataLogging_DOC_v3d0d1_en.pdf (03.10.2021)

6) S7-1200 Data logging Siemens PLC [WWW]. - Ali Al-Mukhtar

https://www.youtube.com/watch?time_continue=2&v=E2tdtwrT_sE&feature=emb_logo&ab_channel=AliAl-Mukhtar (03.10.2021)

7) HTML, JS, CSS, AJAX õpetamine [WWW].

<https://www.w3schools.com/html/> (03.10.2021)

8) Нүпикакнад „Способы создания окон PopUp“ [WWW]. - Евгений Эдуардович @mronegin

<https://habr.com/ru/post/417873/> (03.10.2021)

9) S7-1200 manuaal [WWW]

https://cache.industry.siemens.com/dl/files/465/36932465/att_106131/v1/s71200_system_manual_ru-RU.pdf (03.10.2021)

10) Analooži sügnaali töötlemine, häirede fikseerimine [WWW] - Туманов А.В

http://www.maxplant.ru/article/siemens_tutorial_9.php (03.10.2021)

11) How to Programming Analog Inputs 4-20mA in PLC S7-1200 [WWW]

<https://www.electricalchile.cl/plcs71200siemens2en.php> (03.10.2021)

12) Analooži sügnaal [WWW]

[https://support.industry.siemens.com/cs/document/39334504/for-an-s7-1200-s7-1500-controller-in-step-7-\(tia-portal\)-how-do-you-scale-integer-values-in-real-numbers-and-vice-versa-for-analog-inputs-and-outputs-?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/39334504/for-an-s7-1200-s7-1500-controller-in-step-7-(tia-portal)-how-do-you-scale-integer-values-in-real-numbers-and-vice-versa-for-analog-inputs-and-outputs-?dti=0&lc=en-WW)

13) Analooži sügnaali moodustus, SCL keele õpimine, programmi struktureerimine [WWW]

https://www.youtube.com/channel/UC1P4ACs0hsr7AWclmWKQbQ/playlists?view=50&sort=dd&shelf_id=5 (03.10.2021)

14) Structuring the web with HTML [WWW]

<https://developer.mozilla.org/ru/docs/Learn/HTML> (03.10.2021)

15) Smoothie-cahrt [WWW] - Joe Walnes & Drew Noakes

<http://smoothiecharts.org/> (03.10.2021)

16) Smoothie-chart'i häälestamin [WWW] - Joe Walnes & Drew Noakes

<http://smoothiecharts.org/builder/> (03.10.2021)

17) JustGage+Raphael [WWW]

<https://toorshia.github.io/justgage/> (03.10.2021)

18) Siemens Step 7 manual [WWW]

<https://support.industry.siemens.com/cs/mdm/109747135?c=98342427147&dl=en&lc=ru-RU> (03.10.2021)

19) Source Code Protection For Your SIMATIC S7-1200 PLC in TIA Portal [WWW] - Jason Mayes

<https://www.dmcinfo.com/latest-thinking/blog/id/8884/source-code-protection-for-your-simatic-s7-1200-plc-in-tia-portal> (03.10.2021)

20) AJAX [WWW]

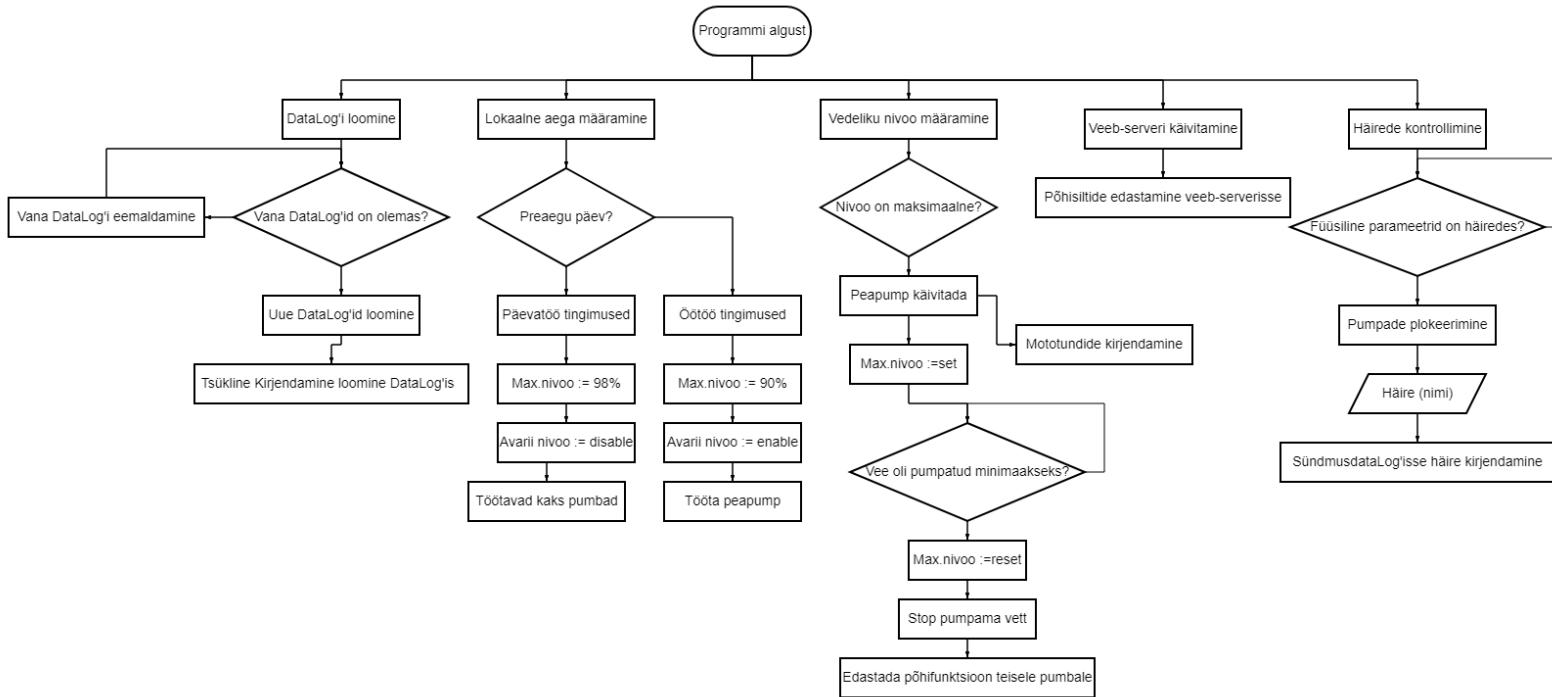
<https://ru.wikipedia.org/wiki/AJAX> (03.10.2021)

21)DOM [WWW]

https://www.w3schools.com/js/js_htmlDOM.asp (03.10.2021)

LISAD

1. Lisa 1



2. Lisa 2

```
1. <!DOCTYPE html>
2. <!-- AWP_In_Variable Name="DB_Outputs".Level' -->
3. <!-- AWP_In_Variable Name="DB_Outputs".Current1' -->
4. <!-- AWP_In_Variable Name="DB_Outputs".Current2' -->
5. <!-- AWP_In_Variable Name="DB_Moto_Time".CV_Min_P1' -->
6. <!-- AWP_In_Variable Name="DB_Moto_Time".CV_Min_P2' -->
7. <!-- AWP_In_Variable Name="DB_Moto_Time".CV_Hour_P1' -->
8. <!-- AWP_In_Variable Name="DB_Moto_Time".CV_Hour_P2' -->
9. <!-- AWP_In_Variable Name="DB_Settings".Level_Min' -->
10. <!-- AWP_In_Variable Name="DB_Settings".Level_Max' -->
11. <!-- AWP_In_Variable Name="DB_Settings".Level_HH' -->
12. <!-- AWP_In_Variable Name="DB_Alarms".Current1_Broken_Wire' -->
13. <!-- AWP_In_Variable Name="DB_Alarms".Current2_Broken_Wire' -->
14. <!-- AWP_In_Variable Name="DB_Alarms".Interlock1' -->
15. <!-- AWP_In_Variable Name="DB_Alarms".Interlock2' -->
16. <!-- AWP_In_Variable Name="DB_Alarms".Force1' -->
17. <!-- AWP_In_Variable Name="DB_Alarms".Force2' -->
18. <!-- AWP_In_Variable Name="DB_Outputs".PUMP1' -->
19. <!-- AWP_In_Variable Name="DB_Outputs".PUMP2' -->
20. <!-- AWP_In_Variable Name="DB_Outputs".Min_Level' -->
21. <!-- AWP_In_Variable Name="DB_Outputs".Night_Work' -->
22. <!-- AWP_In_Variable Name="DB_Outputs".Day_Work' -->
23. <!-- AWP_In_Variable Name="DB_Inputs".Res_all_Err' -->
24. <!-- AWP_In_Variable Name="DB_Inputs".MANMODE1' -->
25. <!-- AWP_In_Variable Name="DB_Inputs".MANMODE2' -->
26. <!-- AWP_In_Variable Name="DB_Inputs".STOP_MANMODE1' -->
27. <!-- AWP_In_Variable Name="DB_Inputs".STOP_MANMODE2' -->
28. <!-- AWP_In_Variable Name="DB_Inputs".Start_PUMP1' -->
29. <!-- AWP_In_Variable Name="DB_Inputs".Start_PUMP2' -->
30. <!-- AWP_In_Variable Name="DB_Inputs".Stop_PUMP1' -->
31. <!-- AWP_In_Variable Name="DB_Inputs".Stop_PUMP2' -->
32. <!-- AWP_In_Variable Name="DB_Alarms_Disable".DIS_Pump1' -->
33. <!-- AWP_In_Variable Name="DB_Alarms_Disable".DIS_Pump2' -->
34. <!-- AWP_In_Variable Name="DB_Outputs".Pump1_DIS' -->
35. <!-- AWP_In_Variable Name="DB_Outputs".Pump2_DIS' -->
```

```

36. <!-- AWP_In_Variable Name='DB_Inputs'.ENABLE_Pump1' -->
37. <!-- AWP_In_Variable Name='DB_Inputs'.ENABLE_Pump2' -->
38. <!-- AWP_In_Variable Name='DB_Outputs'.MANMODE1_ON' -->
39. <!-- AWP_In_Variable Name='DB_Outputs'.MANMODE2_ON' -->
40. <!-- AWP_In_Variable Name='DB_Outputs'.PUMP1_NOT_WORK(web)'' -->
41. <!-- AWP_In_Variable Name='DB_Outputs'.PUMP1_NOT_WORK(web)'' -->
42. <html>
43. <head>
44. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
45. <meta name="viewport" content="width=device-width, initial-scale=1">
46. <link rel="stylesheet" href="css/JuhtCSS.css">
47. <script type="text/javascript" src="js/jquery.js"> </script>
48. <script type="text/javascript" src="js/raphael-2.1.4.min.js"></script>
49. <script type="text/javascript" src="js/justgage.js"></script>
50. <script type="text/javascript" src="js/smoothiecharts.js"></script>
51. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
52. <title>Pumbajaama juhtimine</title>
53. </head>
54. <body>
55. <!--<div id="preloader_malc">
56. <div> </div>
57. </div-->
58. <div class="image">
59. 
60. 
61. </div>
62. <div class="names">
63. <span class="pname" id="p1">PUMP 1</span>
64. <span class="pname" id="p2">PUMP 2</span>
65. </div>
66. <div class="progress">
67. <progress id="prbar" max="100" value="50" ></progress>
68. </div>
69. <div class="buttons">
70. <button id="pumpone">Pump 1</button>
71. <button id="pumptwo">Pump 2</button>
72. <div id="gauge" style="width: 210px; height: 180px"></div>
73. <div id="gauge2" style="width: 210px; height: 180px"></div>
74. <div id="level">PRAEGUNE NIVOO</div>
75. <div id="gauge3" style="width: 210px; height: 180px"></div>
76. </div>
77. <div class="messagepop pop1">
78. <p id="esimene-pump">Esimene pumba juhtimine</p>
79. <div id="images">
80. <p id="pumpindik"></p>
81. </div>
82. <p>
83. <button id="startP1">START</button>
84. <button id="stopP1">STOP</button>
85. </p>
86. <p id="indikatsioon"></p>
87. <p>
88. <button id="blokeeri">BLOKEERI</button>
89. <button id="luba">LUBA</button>
90. </p>
91. <p id="manindik1">KÄSI-JUHTIMINE</p>
92. <p>
93. <button id="kasi-reziim1">KÄSI-REZIIM</button>
94. <button id="lulita-valja1">LÜLITA-VÄLJA</button>
95. </p>
96. <p>
97. <a class="close1" href="/">Cancel</a>
98. </p>
99. </div>
100. <div class="messagepop pop2">
101. <p id="teine-pump">Teine pumba juhtimine</p>
102. <div>
103. <p id="pumpindik2"></p>
104. </div>
105. <p>
106. <button id="startP2">START</button>
107. <button id="stopP2">STOP</button>

```

```

108.         </p>
109.         <p id="indikatsioon2"></p>
110.         <p>
111.             <button id="blokeeri2">BLOKEERI</button>
112.             <button id="luba2">LUBA</button>
113.         </p>
114.         <p id="manindik2">KÄSI-JUHTIMINE</p>
115.         <p>
116.             <button id="kasi-reziim2">KÄSI-REZIIM</button>
117.             <button id="lulita-valja2">LÜLITA-VÄLJA</button>
118.         </p>
119.         <p>
120.             <a class="close2" href="/">Cancel</a>
121.         </p>
122.     </div>
123.     <div class="lamps">
124.         <div class="circle" id="lamp1"></div>
125.         <div class="circle" id="lamp2"></div>
126.     </div>
127.     <div class="time">
128.         <div id="current_date_time_block"></div>
129.     </div>
130.     <button id="reserr">HÄIREDE TAGASTUS</button>
131.     <div class="moto-hours">MOTO-TUNNID
132.         <table id="table">
133.             <tr>
134.                 <td>First Pump </td>
135.                 <td>Minute:</td>
136.                 <td><div id="min1"></div></td>
137.                 <td>Hours:</td>
138.                 <td><div id="hour1"></div></td>
139.             </tr>
140.             <tr>
141.                 <td>Second Pump</td>
142.                 <td>Minute:</td>
143.                 <td> <div id="min2"></div></td>
144.                 <td>Hours:</td>
145.                 <td><div id="hour2"></div></td>
146.             </tr>
147.         </table>
148.     </div>
149.     <div id="nivood" class="nivood">
150.         <table>
151.             <tr>
152.                 <td>
153.                     START NIVOO-PÄEV:
154.                 </td>
155.                 <td id="start-nivoo-paev"></td>
156.             </tr>
157.             <tr>
158.                 <td>
159.                     START NIVOO-ÖÖ:
160.                 </td>
161.                 <td id="start-nivoo-oo"></td>
162.             </tr>
163.             <tr>
164.                 <td>
165.                     STOPP NIVOO:
166.                 </td>
167.                 <td id="stop-nivoo"></td>
168.             </tr>
169.         </table>
170.     </div>
171.     <div class="num">
172.         <p id="levelcurr"></p>
173.     </div>
174.     <canvas id="smoothie-chart" width="1000" height="285"></canvas>
175. </div>
176. </div>
177. <div class="footer">
178.     <div>

```

```

179.         <button id="pealeht" value=" Main Page "
onClick='location.href="MainPage.html"' formtarget="_blank">PEALEHT</button>
180.         <button id="haired" value=" Main Page "
formtarget="_blank">HÄIRED</button>
181.         <button id="parameetrid" value=" Main Page "
formtarget="_blank">PARAMEETRID</button>
182.     </div>@ Jaroslav Savelitsev</div>
183.     <script>
184.         $.ajaxSetup({cache:true});
185.         var gauge = new JustGage({
186.             id: "gauge", // the id of the html element
187.             value: 100,
188.             min: 0,
189.             max: 300,
190.             levelColors: ["EBEB00","EBEB00","EBEB00"],
191.             title: "PUMP 1 VOOL"
192.         });
193.         var curr=parseInt(:"DB_Outputs".Current1:);
194.         setInterval(() => {
195.             $.get("signals/Current1.html",function(restult){
196.                 curr=parseInt(restult.trim());
197.             });
198.             gauge.refresh(curr);
199.         }, 500);
200.
201.         var gauge2 = new JustGage({
202.             id: "gauge2", // the id of the html element
203.             value: 100,
204.             min: 0,
205.             max: 300,
206.             levelColors: ["EBEB00","EBEB00","EBEB00"],
207.             title: "PUMP 2 VOOL"
208.         });
209.         var curr2=parseInt(:"DB_Outputs".Current2:);
210.         setInterval(() => {
211.             $.get("signals/Current2.html",function(restult){
212.                 curr2=parseInt(restult.trim());
213.             });
214.             gauge2.refresh(curr2);
215.         }, 500);
216.
217.         var gauge3 = new JustGage({
218.             id: "gauge3", // the id of the html element
219.             value: 50,
220.             min: 0,
221.             max: 100,
222.             levelColors: ["0F6BD4","0F6BD4","0F6BD4"],
223.             title: "PRAEGUNE NIVOO"
224.         });
225.         var mes=parseInt(:"DB_Outputs".Level:);
226.         setInterval(() => {
227.             $.get("signals/Level.html",function(restult){
228.                 mes=parseInt(restult.trim());
229.             });
230.             gauge3.refresh(mes);
231.         }, 500);
232.
233.
234.         /*Загрузочная страница*/
235.         //window.onload = function() {
236.
237.             //setTimeout(function() {
238.
239.
240.                 //document.getElementById("preloader_malc").style.display = "none";
241.
242.                 //}, 400);
243.
244.             //});
245.             var chart = new
SmoothieChart({grid:{fillStyle:'#ffffff'},labels:{fillStyle:'#000000'},millisPerPixel:45,max

```

```

ValueScale:0.8,minValueScale:0.8,scaleSmoothing:0.012,minValue:0,maxValue:100,timestampForma
tter:SmoothieChart.timeFormatter}),
245.         canvas = document.getElementById('smoothie-chart'),
246.         series = new TimeSeries();
247.
248.         chart.addTimeSeries(series,
{lineWidth:2,strokeStyle:'#ff0000',fillStyle:'rgba(0,128,255,0.30)'}));
249.         chart.streamTo(canvas, 973);

```

```

1. <!DOCTYPE html>
2. <!-- AWP_In_Variable Name='DB_Alarms'.Level_not_Fall' -->
3. <!-- AWP_In_Variable Name='DB_Alarms'.Level_HighHigh' -->
4. <!-- AWP_In_Variable Name='DB_Alarms'.Level_Broken_Wire' -->
5. <!-- AWP_In_Variable Name='DB_Alarms'.Current1_Broken_Wire' -->
6. <!-- AWP_In_Variable Name='DB_Alarms'.Current2_Broken_Wire' -->
7. <!-- AWP_In_Variable Name='DB_Alarms'.Interlock1' -->
8. <!-- AWP_In_Variable Name='DB_Alarms'.Interlock2' -->
9. <!-- AWP_In_Variable Name='DB_Alarms'.Force1' -->
10. <!-- AWP_In_Variable Name='DB_Alarms'.Force2' -->
11. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Level_Broken_Wire' -->
12. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Current1_Broken_Wire' -->
13. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Current2_Broken_Wire' -->
14. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Level_HH' -->
15. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Level_NF' -->
16. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Interlock1' -->
17. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Interlock2' -->
18. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Force1' -->
19. <!-- AWP_In_Variable Name='DB_Alarms_Disable'.DIS_Force2' -->
20. <!-- AWP_In_Variable Name='DB_Inputs'.Res_all_Err' -->
21. <html lang="en">
22. <head>
23.   <meta charset="UTF-8">
24.   <meta name="viewport" content="user-scalable=no, width=device-width, initial-scale=1.0,
maximum-scale=1.0">
25.   <link rel="stylesheet" href="css/haired.css">
26.   <script type="text/javascript" src="js/jquery.js"> </script>
27.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
28.   <script type="text/javascript" src="js/Scripts for Haired.js"></script>
29.   <title>Häired</title>
30. </head>
31. <body>
32.   <div class="table">
33.     <table id="table">
34.       <tr>
35.         <th id="err1">Nivoo ei lange</th>
36.         <th id="th1">
37.           <button id="dis1" onclick="toggle1()">Välja
lülitama</button>
38.         </th>
39.       </tr>
40.       <tr>
41.         <th id="err2">Nivoo on HH</th>
42.         <th id="th2">
43.           <button id="dis2" onclick="toggle2()">Välja
lülitama</button>
44.         </th>
45.       </tr>
46.       <tr>
47.         <th id="err3">Nivoo juhe katkenud</th>
48.         <th id="th3">
49.           <button id="dis3" onclick="toggle3()">Välja
lülitama</button>
50.         </th>
51.       </tr>
52.       <tr>
53.         <th id="err4">Pump 1 juhe katkenud</th>
54.         <th id="th4">
55.           <button id="dis4" onclick="toggle4()">Välja
lülitama</button>
56.         </th>
57.       </tr>

```

```

58.         <tr>
59.             <th id="err5">Pump 2 juhe katkenud</th>
60.             <th id="th5">
61.                 <button id="dis5" onclick="toggle5()">Välja
        lülitama</button>
62.             </th>
63.         </tr>
64.         <tr>
65.             <th id="err6">Interlock 1</th>
66.             <th id="th6">
67.                 <button id="dis6" onclick="toggle6()">Välja
        lülitama</button>
68.             </th>
69.         </tr>
70.         <tr>
71.             <th id="err7">Interlock 2</th>
72.             <th id="th7">
73.                 <button id="dis7" onclick="toggle7()">Välja
        lülitama</button>
74.             </th>
75.         </tr>
76.         <tr>
77.             <th id="err8">Force 1</th>
78.             <th id="th8">
79.                 <button id="dis8" onclick="toggle8()">Välja
        lülitama</button>
80.             </th>
81.         </tr>
82.         <tr>
83.             <th id="err9">Force 2</th>
84.             <th id="th9">
85.                 <button id="dis9" onclick="toggle9()">Välja
        lülitama</button>
86.             </th>
87.         </tr>
88.         <tr>
89.             <th><button id="reserr">HÄIREDE TAGASTUS</button></th>
90.             <th>HÄIRETE EMALDAMINE</th>
91.         </tr>
92.     </table>
93.     <div class="selgitus">Peab olema kahekordne nuppu pressima, sest realiseeritud
        kaitse juhusliku pressimine eest</div>
94.     <div class="footer">
95.         <div>
96.             <button id="pealeht" value=""
                onClick='location.href="MainPage.html"'
                formtarget="_blank">PEALEHT</button>
97.             <button id="juhtimine" value=""
                onClick='location.href="Juhtimine.html"'
                formtarget="_blank">JUHTIMINE</button>
98.             <button id="parameetrid" value=""
                onClick='location.href="Parameetrid.html"'
                formtarget="_blank">PARAMEETRID</button>
99.         </div>
100.     <div> © Jaroslav Savelitsev</div>
101.     </div>
102. </div>
103. <script type="text/javascript">
104.     var name;
105.
106.     $("#dis1").click(function(){
107.         name = 'DB_Alarms_Disable'.DIS_Level_NF;
108.     });
109.
110.     $("#dis2").click(function(){
111.         name = 'DB_Alarms_Disable'.DIS_Level_HH;
112.     });
113.     $("#dis3").click(function(){
114.         name = 'DB_Alarms_Disable'.DIS_Level_Broken_Wire;
115.     });
116.     $("#dis4").click(function(){
117.         name
        =
        'DB_Alarms_Disable'.DIS_Current1_Broken_Wire;
118.     });
119.     $("#dis5").click(function(){

```

```

120.                                     name
    '"DB_Alarms_Disable".DIS_Current2_Broken_Wire';
121.                                     });
122.                                     $("#dis6").click(function(){
123.                                         name = '"DB_Alarms_Disable".DIS_Interlock1';
124.                                     });
125.                                     $("#dis7").click(function(){
126.                                         name = '"DB_Alarms_Disable".DIS_Interlock2';
127.                                     });
128.                                     $("#dis8").click(function(){
129.                                         name = '"DB_Alarms_Disable".DIS_Force1';
130.                                     });
131.                                     $("#dis9").click(function(){
132.                                         name = '"DB_Alarms_Disable".DIS_Force2';
133.                                     });
134.
135. </script>
136. </body>
137. </html>
138.

```

```

1. <!DOCTYPE html>
2. <!-- AWP_In_Variable Name='"DB_Settings".Level_Min' -->
3. <!-- AWP_In_Variable Name='"DB_Settings".Level_Max' -->
4. <!-- AWP_In_Variable Name='"DB_Settings".Level_HH' -->
5. <!-- AWP_In_Variable Name='"DB_Settings".Current1_high' -->
6. <!-- AWP_In_Variable Name='"DB_Settings".Current2_high' -->
7. <!-- AWP_In_Variable Name='"DB_Settings".Current1_low' -->
8. <!-- AWP_In_Variable Name='"DB_Settings".Current2_low' -->
9. <!-- AWP_In_Variable Name='"DB_Timers_Delays".TON_LevelNF'-->
10. <!-- AWP_In_Variable Name='"DB_Timers_Delays".TON_Interlock' -->
11. <!-- AWP_In_Variable Name='"DB_Timers_Delays".TON_Force' -->
12. <html lang="en">
13. <head>
14.   <meta charset="UTF-8">
15.   <meta name="viewport" content="user-scalable=no, width=device-width, initial-scale=1.0,
    maximum-scale=1.0">
16.   <link rel="stylesheet" href="css/parameetrid.css">
17.   <script src="https://code.jquery.com/jquery-1.9.1.min.js"></script>
18.   <script type="text/javascript" src="js/Scripts for Parameetrid.js"></script>
19.   <title>Parameetrid hälestamine</title>
20. </head>
21. <body>
22.   <div class="table"></div>
23.   <table id="table">
24.     <tr>
25.       <td>Minimaalne nivoo</td>
26.       <td class="param">
27.         <form method="post">
28.           <input name='"DB_Settings".Level_Min' type="text"
    />
29.           <button type="submit">Saatma</button>
30.         </form>
31.       </td>
32.     </tr>
33.     <tr>
34.       <td>Maksimaalne nivoo</td>
35.       <td class="param">
36.         <form method="post">
37.           <input name='"DB_Settings".Level_Max' type="text"
    />
38.           <button type="submit">Saatma</button>
39.         </form>
40.       </td>
41.     </tr>
42.     <tr>
43.       <td>High-High nivoo</td>
44.       <td class="param">
45.         <form method="post">
46.           <input name='"DB_Settings".Level_HH' type="text" />
47.           <button type="submit">Saatma</button>

```

```

48.         </form>
49.     </td>
50. </tr>
51. <tr>
52.     <td>Vool 1 maksimaalne</td>
53.     <td class="param">
54.         <form method="post">
55.             <input             name="'DB_Settings'.Current1_high'
type="text" />
56.                 <button type="submit">Saatma</button>
57.         </form>
58.     </td>
59. </tr>
60. <tr>
61.     <td>Vool 2 maksimaalne</td>
62.     <td class="param">
63.         <form method="post">
64.             <input             name="'DB_Settings'.Current2_high'
type="text" />
65.                 <button type="submit">Saatma</button>
66.         </form>
67.     </td>
68. </tr>
69. <tr>
70.     <td>Vool 1 minimaalne</td>
71.     <td class="param">
72.         <form method="post">
73.             <input             name="'DB_Settings'.Current1_low'
type="text" />
74.                 <button type="submit">Saatma</button>
75.         </form>
76.     </td>
77. </tr>
78. <tr>
79.     <td>Vool 2 minimaalne</td>
80.     <td class="param">
81.         <form method="post">
82.             <input             name="'DB_Settings'.Current2_low'
type="text" />
83.                 <button type="submit">Saatma</button>
84.         </form>
85.     </td>
86. </tr>
87. <tr>
88.     <td>Timer nivoo ei lange</td>
89.     <td class="param">
90.         <form method="post">
91.             <input             name="'DB_Timers_Delays'.TON_LevelNF'
type="text" />
92.                 <button type="submit">Saatma</button>
93.         </form>
94.     </td>
95. </tr>
96. <tr>
97.     <td>Timer Interlock</td>
98.     <td class="param">
99.         <form method="post">
100.            <input
name="'DB_Timers_Delays'.TON_Interlock' type="text" />
101.                <button type="submit">Saatma</button>
102.            </form>
103.     </td>
104. </tr>
105. <tr>
106.     <td>Timer Force</td>
107.     <td class="param">
108.         <form method="post">
109.             <input
name="'DB_Timers_Delays'.TON_Force' type="text" />
110.                 <button type="submit">Saatma</button>
111.         </form>
112.     </td>

```



```

113.         </tr>
114.     </table>
115.     <div class="values">
116.         <p id="value1">:="DB_Settings".Level_Min:%</p>
117.         <p id="value2">:="DB_Settings".Level_Max:%</p>
118.         <p id="value3">:="DB_Settings".Level_HH:%</p>
119.         <p id="value4">:="DB_Settings".Current1_high:A</p>
120.         <p id="value5">:="DB_Settings".Current2_high:A</p>
121.         <p id="value6">:="DB_Settings".Current1_low:A</p>
122.         <p id="value7">:="DB_Settings".Current2_low:A</p>
123.         <p id="value8">:="DB_Timers_Delays".TON_LevelNF:</p>
124.         <p id="value9">:="DB_Timers_Delays".TON_Interlock:</p>
125.         <p id="value10">:="DB_Timers_Delays".TON_Force:</p>
126.     </div>
127.     <div class="footer">
128.         <div>
129.             <button          id="pealeht"          value="          "
onClick='location.href="MainPage.html"' formtarget="_blank">PEALEHT</button>
130.             <button          id="juhtimine" value=""
onClick='location.href="Juhtimine.html"' formtarget="_blank">JUHTIMINE</button>
131.             <button          id="haired" value=""      Main      Page      "
onClick='location.href="Haired.html"' formtarget="_blank">HÄIRED</button>
132.         </div>
133.         © Jaroslav Savelitsev</div>
134.
135. </body>
136. </html>
137.

```