



TALLINNA TEHNIKAÜLIKOO  
INSENERITEADUSKOND  
Virumaa kolledž

**PUMBAJAAMA AUTOMATISEERIMINE ELEKTRI  
BÖRSIHINDA ARVESTADES**  
**AUTOMATION OF A PUMPING STATION, DEPENDING ON THE  
EXCHANGE PRICE OF ELECTRICITY**  
TELEMAATIKA JA ARUKAD SÜSTEEMID ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Aleksandr Hinn  
Üliõpilaskood: 193239EDTR  
Juhendaja: Sergei Pavlov, 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.

"...." ..... 20.....

Autor: .....

/ allkiri /

Töö vastab rakenduskõrgharidusõppe lõputööle/magistritööle esitatud nõuetele

"...." ..... 20.....

Juhendaja: .....

/ allkiri /

Kaitsmisele

lubatud

"...." ..... 20.....

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

## **LIHTLITSENTS LÕPUTÖÖ ÜLDSUSELE KÄTTESAADAVAKS TEGEMISEKS JA REPRODUTSEERIMISEKS**

Mina Aleksandr Hinn (sünnikuupäev: 30.08.1977)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Pumbajaama automatiseerimine elektri börsihinda arvestades“ mille juhendaja on Sergei Pavlov,
  - 1.1. reprodutseerimiseks säilitamise ja elektroonilise avaldamise eesmärgil, sealhulgas 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 punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta kolmandate isikute intellektuaalomandi ega isikuandmete kaitse seadusest ja teistest õigusaktidest tulenevaid õigusi.

# TalTech Inseneriteaduskond Virumaa kolledž

## LÕPUTÖÖ ÜLESANNE

**Üliõpilane:** Aleksandr Hinn, 193239EDTR

Õppekava, peeriala: EDTR17/18 Telemaatika ja arukad süsteemid, Protsesside automatiseerimine

Juhendaja(d): : Lektor, Sergei Pavlov, [sergei.pavlov@taltech.ee](mailto:sergei.pavlov@taltech.ee)

### Lõputöö teema:

(eesti keeles) Pumbajaama automatiseerimine elektri börsihinda arvestades.

(inglise keeles) Automation of a pumping station, depending on the exchange price of electricity.

### Lõputöö põhieesmärgid:

1. Pumbajaama pumpade töö automaatjuhtimine elektri börsihinda arvestades.

### Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Lõputöö teema valik, andmete kogumine, tehnoloogiate ja automatiseerimisevahendite valimine.	02.09.2022 - 30.09.2022
2.	Tarkvara loomine ja testimine, tõhususe analüüs.	1.10.2022 - 31.10.2022
3.	Lõputöö kirjutamine ja vormistamine.	1.11.2022 - 30.11.2022
4.	Lõputöö viimased parandused.	01.12.2022 - 12.12.2022

**Töö keel:** Eesti keel

**Lõputöö esitamise tähtaeg:** "....."..... 20.....a

**Üliõpilane:** Aleksandr Hinn

/allkiri/

"....." ..... 20.....a

**Juhendaja:** Sergei Pavlov

/allkiri/

"....." ..... 20.....a

**Programmijuht:** Žanna Gratševa

/allkiri/

"....." ..... 20.....a

# SISUKORD

EESSÕNA .....	7
LÜHENDITE JA TÄHISTE LOETELU .....	8
SISSEJUHATUS .....	9
1 ESTONIA KAEVANDUSE PUMBAJAAM.....	11
1.1 Kirjeldus.....	11
1.2 Pumpade parameetrid .....	11
1.3 Tõhususe uurimine ja analüüs.....	12
1.3.1 Pythoni rakenduse loomine .....	13
2 PROTSESSI JUHTIMISSÜSTEEM.....	20
2.1 Protsessi juhtimissüsteemi kirjeldus.....	20
3 PROJEKTI REALISEERIMINE .....	21
3.1 Seadmete ja tehnoloogiate valik.....	21
3.2 Kontrolleri programmeerimine.....	21
3.2.1 Veebiserver .....	22
3.2.2 Taseme andur .....	23
3.2.3 Käsitsi töörežiim.....	25
3.2.4 "Auto" töörežiim.....	27
3.2.5 "ECO" töörežiim .....	28
3.3 HTML rakenduse kirjutamine.....	30
3.3.1 Side kontrolloriga .....	31
3.3.2 Veebi API. ....	31
3.3.3 Protsessi juhtimine .....	33
3.3.4 Visualiseerimine .....	35
KOKKUVÕTE .....	39
SUMMARY.....	41
KASUTATUD KIRJANDUSE LOETELU .....	43
Lisa 1 HTML-rakendus index.html .....	44

## **EESSÕNA**

Minu lõputöö eesmärgiks on kaevanduse pumbajaama töö automatiseerimine elektri börsihinda arvestades. Viimase aja kõrged elektri hinnad on valusalt mõjutanud kõiki sh ettevõtteid ning tekitanud selle kaudu ootamatult suurt kulude kasvu. Estonia kaevanduses on üheks energiamahukaks protsessiks põhjavee pumpamine. Mina kogusin ja analüüsisin kaevanduse pumbajaama nr.8 töö statistikat ning jõudsin järeldusele, et elektrikulusid rahalises väärtuses on võimalik oluliselt vähendada. Selle jaoks tuleb pumpa seadmeistiku töörežiimi automatiseerida. Avaldan tänu minu lõputöö juhendajale: telemaatika ja arukate süsteemide lektor Sergei Pavlovile. Täna kaevanduse vanemmehaanikut Marek Tulkot, kes abistas konsultatsioonide ja andmete kogumisega.

Võtmesõnad: elektri börsihind, kontrollid, programmeerimine, moderniseerimine, automaatika, diplomitöö.

## LÜHENDITE JA TÄHISTE LOETELU

- **API** (Application Programming Interface) – masinliides;
- **SCADA** (Supervisory control and data acquisition) - tarkvarapakett andmete kogumiseks ja tööstusprotsesside juhtimiseks;
- **TIA Portal** (Totally Integrated Automation Portal) - Kontrolleri Siemens S-1200 programmeerimiseks kasutakse Siemens AG arenduskeskkonna Tia Portal versiooni 15;
- **Python** – programmeerimiskeel;
- **HTML** - (HyperText Markup Language) on standardiseeritud hüpertexti märgistuskeel veebilehtede vaatamiseks brauseris.
- **jQuery** - JavaScripti teek, mis keskendub JavaScripti ja HTML-i vastastikusele mõjule.
- **JavaScript** - veebi programmeerimiskeel;
- **JSON (JavaScript Object Notation)** – teksti andmevahetuse vorming;
- **GIF (The Graphics Interchange Format)** - graafiline vorming, mis võimaldab pildist luua animatsiooni;



## SISSEJUHATUS

Viimasel ajal kogu maailmas on oluliselt tõusnud energiakandjate hinnad. Selle tulemusel eri ettevõtete ning organisatsioonide finantsilised kulutused aina kasvavad. Paraku on mõned ettevõtted sunnitud oma tegevuse lõpetama kuna nt tootmise protsessi käigus kulutatava energia kogumaksumus ei võimalda viimastel pakkuda oma tooteid või teenust konkurentsivõimelise hinnaga avalikul turul.

Elektriarvete vähendamiseks tuleb antud tingimustes alandada, kas tarbimist või tarbitava elektri hinda. Ettevõtetele, kus tootmiskulud sõltuvad rohkem elektri hinnast, toob elektritarbimise vähenemine kaasa toodangumahu vähenemise, mis omakorda negatiivselt mõjub nii konkurentsivõimele, kui ka asutuse jätkusuutlikkusele.

Küll aga on võimalik börsipaketi kasutades elektriarvet vähendada. Börsihind kujuneb elektribörsil, milleks Põhjamaades on Nord Pool. Börsipakettide puhul võib hind sõltuda igal tunnil tarbitud elektrienergia kogusest ja sellel tunnil kehtivast börsihinnast. Kasutades tunnipõhist börsipaketti on tarbijal võimalik tarbimist reguleerida ja kavandada oma tegevuse odavamatele tundidele juhul, kui ettevõtte protsessid saavad olla nii paindlikud.

Börsi tunnihinnad on diskreetsed, igal tunnil on eri hind, mis väljendub eurodes megavatt-tunni kohta. Tähtis, et tunni elektri hinnad on teada täpselt ööpäev ette. Elektrienergia tarbimisest tekkivat kulu saab vähendada, kui energiamahukate protsside käivitamist kavandada soodsa tunnihinna kehtivuse perioodiks.

Minu lõputöö eesmärk on odavamatel tundidel elektrimasinate käivitamise automatiseerimine ning loogika sidumine elektrienergia maksumusega. Automatiseerimiseks mina valisin Estonia kaevanduse allmaa pumbajaama, sest põhjavee pumpamine on üks energiamahukamatest protsessidest ettevõttes, mis võimaldab aga kindlates piirides paidlikku töögraafikut kasutada.

Lõputöö analüüsi ülesandeks on kaevanduse SCADA andmebaasist tööprotsessi statistiliste andmete allalaadimine ja nende alusel uue töörežiimi algoritmi leidmine. Teine ülesanne on elektrienergia tarbimiskulude võrdlemine, vana ja uue automatiseerimise puhul. Selle jaoks tuleb võrrelda kaheksanda pumbajaama tavalise töörežiimiga kuu elektriarvet ning uue automatiseerimisalgoritmi kasutamise tulemusel saadud elektri kuarvet.

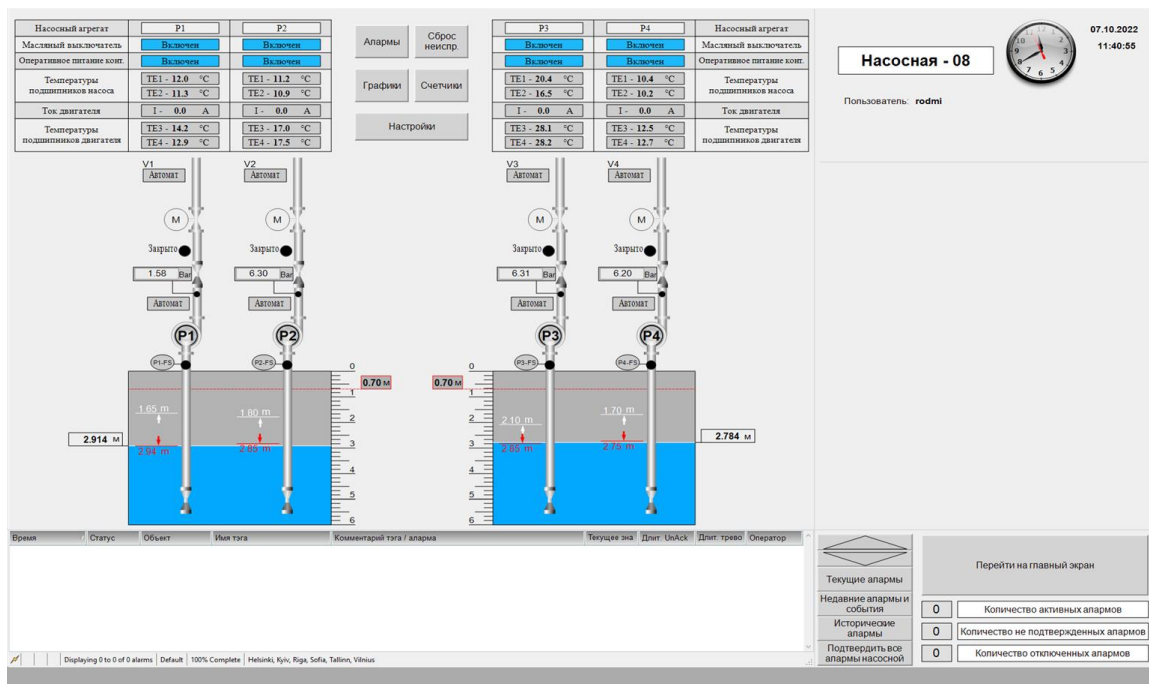
Minu lõputöö eesmärgiks on kaevanduse pumbajaama automatiseerimine elektrienergia börsihinda arvestades. See tähendab, et pumpamise seadmeid saab käivitada sõltuvalt börsi elektritunnihinnast soodsamal perioodil. Pumbajaam on juba automatiseeritud, mis vähendab ka automaatikaseadmete ostmise kulusid, lisatakse ainult uue režiimi juhtimiseks ning visualiseerimiseks tööstusarvutit ja vastavat tarkvara.

Kuna Estonia kaevanduse pumbajaama nr. 8 automatiseerimiseks juba kasutatakse Siemensi seadmeid, nagu kontroller ja SCADA juhtimissüsteem, siis kasutan oma töös ka Siemensi poolt toodetud komponente ja seadmeid. Automaatika skeemi realiseerimiseks kasutatakse kontrollerit Simantic S7, milles on veebi server ja loodud veebi rakendus, mis käivitub Siemensi tööstusarvutis IPS brauseri kaudu. Protsessi juhtimiseks uues režiimis "ECO" on vajalik ka HTML rakenduse loomine.

# 1 ESTONIA KAEVANDUSE PUMBAJAAM

## 1.1 Kirjeldus

Pumbajaam nr. 8 asub Estonia kaevanduses allmaa osas. Põhjavesi kogutakse allmaa veekoguriusse ja teatud taseme saavutamisel pumbatakse maapinnale. Pumblas on neli pumpa ja ultraheli andur veetaseme mõõtmiseks, mille näitude kaudu käivituvad pumbad. Igal pumbal on olemas oma käivitamise ning väljalülitamise veetaseme piir. Kui esimene pump töötab, aga veetase ikka tõuseb, siis käivitub teine pump ja nii edasi. Pumpla juhtimiseks ja protsessi visualiseerimiseks kasutatakse SCADA süsteemi. (vt Joonis 1.1)



Joonis 1.1 Kaheksanda pumbajaama tööprotsessi juhtimine ning visualiseerimine

## 1.2 Pumpade parameetrid

Analüüsi jaoks on vajalikud pumpade põhiparameetrid nagu pumba elektrimootori võimsus ning pumbaseadme tootlikkus. (vt Joonis 1.2)

ВОДООТЛИВНЫЕ УСТАНОВКИ ШАХТЫ „ЭСТОНИЯ“														
ПО СОСТОЯНИЮ НА 01.01.2020 Г.														
Местоположение	Напорные трубопроводы			Насосы				Всасыв. трубы		Электродвигатели				
	№	Диаметр, метр	Длина, м	№	Тип	Произв. м <sup>3</sup> /ч	Год монтажа	Диаметр, метр	Длина, м	Тип	Р, квт	U, в	I, ном. А	Вес, кг.
8	1	300	83	1	Д2000/100	1840	1979	600	8	A13-59-6	800	6000	94	4170
	4	300	83	4	Д2000/100	1330	1979	600	8	A13-59-6	800	6000	94	4170
	2	400	83	2	DHV.300-560PA	1390	2012	500	7	DM1-HV450-4	560	6000	65.7	3900
	3	400	82	3	DHV.300-560PA	1450	2012	500	7	DM1-HV450-4	560	6000	65.7	3900

Joonis 1.2 Kaevanduse kaheksanda pumbajaama pumpade parameetrid

### 1.3 Tõhususe uurimine ja analüüs

Uurimine koosneb andmete kogumisest ja analüüsist. Analüüsi eesmärk on kulude vähendamiseks parema lahenduse leidmine. Analüüsis kasutatakse kaevanduse pumbajaama nr. 8 andmeid oktoobri kuu eest.

SCADA andmete põhjal on näha, et oktoobris töötas ainult kolmas pump. Pumba tööaeg oli 84 tundi ja 27 minutit (89 sisselülitamist). Pumba oktoobri elektritarbimine oli 45 432 kWh. Pumbatud veemaht ja ka vee sissevool oli 113 163 m<sup>3</sup>. (vt Joonis 1.3)

P3	Данные за день				Данные за месяц				Суммарное значение		Сброс
	Текущий		Предыдущий		Текущий		Предыдущий				
Q = 1340 м3/ч	2 ч	13 м	3 ч	25 м	87 ч	2 м	84 ч	27 м	44150 ч	36 м	
Моточасы	2 вкл		4 вкл		90 вкл		89 вкл		7935 вкл		
Коп-во включений	2 вкл		4 вкл		90 вкл		89 вкл		7935 вкл		
Активная энергия	1214 кВтч		1858 кВтч		47080 кВтч		45432 кВтч		2246880 кВтч		
Выкачано воды	2970 м3		4578 м3		116625 м3		113163 м3		59161804 м3		

Joonis 1.3 Kaheksanda pumbajaama kolmanda pumba andmed oktoobri kuu eest

Sellest tulenevalt oli päeva keskmine elektritarbimine:

$$45\,432\text{kWh} / 31\text{päev} = 1\,466\text{ kWh päevas}$$

Kui käivituvad kõik neli pumpa, siis tunni elektritarbimine võrdub nelja pumba koguvõimsusega:

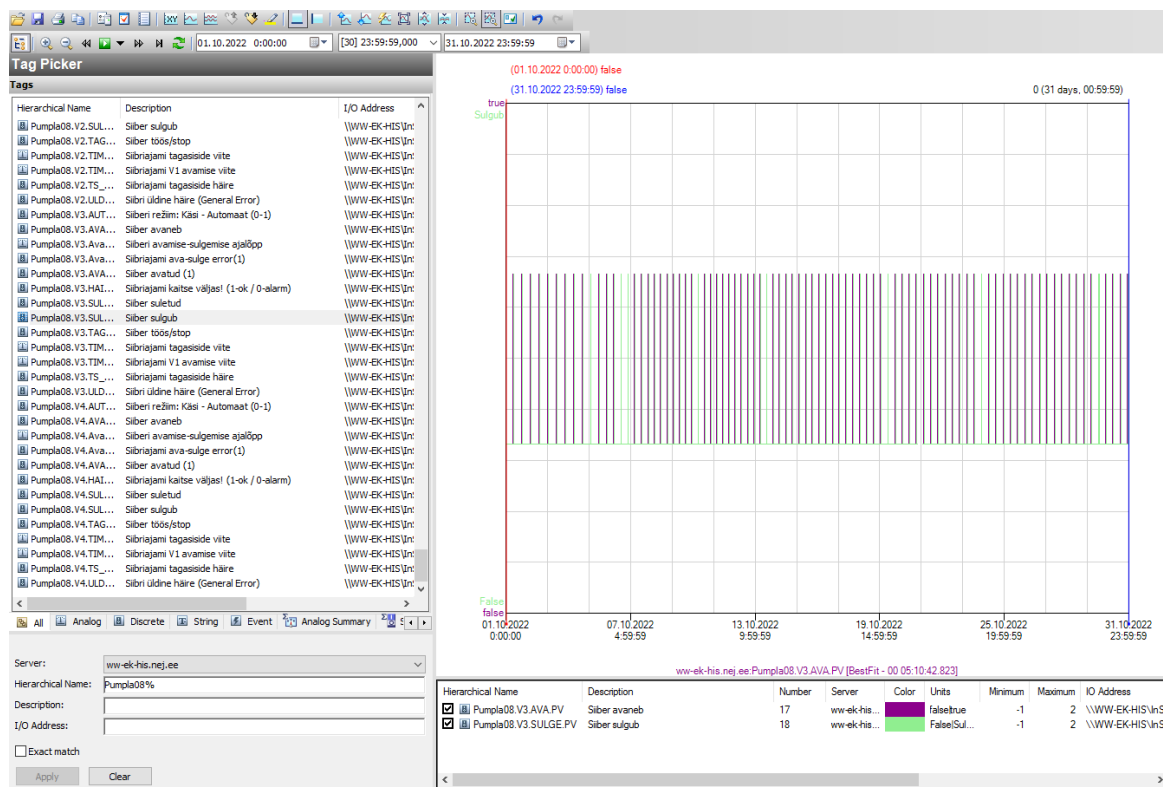
$$1\,840 + 1\,390 + 1\,450 + 1\,330 = 6\,010\text{ kWh}$$

Börsil on teada järgmise ööpäeva tunnielektrihinnad ette. Kui börsi tunnielektrihind on kõige madalam, siis saab mitu pumpa ühel ajal käivitada, pumbates välja varasema sissevoolava veehulga kogurist. Üheks põhitingimuseks aga on see, et kõikide pumpade tootlikkus peab olema kõrgem kui perioodil sissevoolava vee maht.

### 1.3.1 Pythoni rakenduse loomine

Analüüsi lihtsustamiseks ja automatiseerimiseks on kirjutatud skript Pythoni programmeerimise keele abil.

Suurema pumba ööpäevase tööaja leidmiseks vaja oktoobri kuu andmeid SCADA-st importida ja analüüsida. (vt Joonis 1.4)



Joonis 1.4 Kaheksanda pumbajaama kolmanda pumba graafik oktoobri kuu eest

Andmed kaevanduse allmaa pumplast nr.8 asuvad SCADA andmebaasis, mille väljavõtet saab CSV faili kujul alla laadida. Analüüsi aluseks on võetud ajahetked, kus pumba siiber avaneb/sulgub. Sellisel juhul on näha vee pumpamise protsessi algusest lõpuni. Failis andmed hoitakse tabeli kujul, kus iga rida omab järgmisi siibrite töö väärtusi:

- kuupäev ja kellaaeg;
- parameetrid: „Pumpla08.V3.AVA.PV“ on siibri avamine ja „Pumpla08.V3.SULGE.PV“ on siibri sulgumine;
- andmebaasi asukoht;
- siibri liikumise etapi lõpp;
- andmete saamise tulemus, kus „Good“ on edukas ja „Bad“ on mitteedukas. (vt Joonis 1.5)

Time;Hierarchical Name;Server;Value;Quality
01.10.2022 8:25:08.860;Pumpla08.V3.AVA.PV;ww-ek-his.nej.ee>true;Good
01.10.2022 8:26:10.198;Pumpla08.V3.AVA.PV;ww-ek-his.nej.ee>false;Good
01.10.2022 9:21:25.323;Pumpla08.V3.SULGE.PV;ww-ek-his.nej.ee;Sulgub;Good
01.10.2022 9:22:21.635;Pumpla08.V3.SULGE.PV;ww-ek-his.nej.ee;False;Good

Joonis 1.5 CSV failis on kolmanda pumba SCADA andmed

Põhjavee pumpamise protsessi võib etapidena järjestada:

- 1) pump käivitub;
- 2) kui veesamba rõhk tõuseb, siis hakkab elektrisiiber avanema ja SCADA parameeter "Pumpla08.V3.AVA.PV" saab "true" väärtuse;
- 3) kui elektrisiiber on avatud, siis "Pumpla08.V3.AVA.PV" saab "false" väärtuse;
- 4) pumpamise lõppetapina siiber hakkab sulguma ja parameeter "Pumpla08.V3.SULGE.PV" saab "Sulgub" väärtuse;
- 5) kui siiber on suletud, siis parameeter "Pumpla08.V3.SULGE.PV" saab "False" väärtuse;
- 6) toimub pumba väljalülitumine tööst.

SCADA-st polnud võimalust pumpade käivitamise aja andmeid alla laadida. Aga võib kasutada pumba siibri andmeid. Analüüsi jaoks vaja teada pumba siibri eduka avamise ja sulgumise aegu oktoobri kuu jooksul.

Pythoni rakenduse töö etapid on

- lugemine failist ja vajalike andmete salvestamine massiivi: selliseid, nagu siibri eduka avamise aeg ning siibri eduka sulgumise aeg;
- leida kõige suurema ööpäevase vee juurdevoolu hulga oktoobris;
- lugemine teisest failist oktoobri börsi tunnihindu ja nende salvestamine massiivi;
- võrdlemine elektrienergia kasutamisest tuleneva rahalise kulu mahtu oktoobri kuu eest vana ja uue automatiseerimismeetodi puhul.

Kolmanda pumba andmeid laaditakse alla CSV failina. Failist võetakse ainult siibri eduka avamise/sulgumise aega ja salvestatakse analüüsiks loendisse "mlist". (vt Joonis 1.6)

```

6 #Pumba käigu oktoobri statistikaandmeid SCADA CSV failist lugemine
7 with open('pump3.csv', 'r', newline='') as f:
8     reader = csv.reader(f, delimiter=',')
9     mlist = []
10    for row in reader:
11        if row[3] == 'true' or row[3] == 'Sulgub':
12            mlist.append(datetime.strptime(row[0][:-4], '%d.%m.%Y %H:%M:%S'))

```

Joonis 1.6 CSV failist andmete lugemine

Kuna elektrienergia börsil igal tunnil on uus elektrihind, siin analüüsiks tuleb teada kui palju pump töötas igal tunnil ööpäevas. Kui pump käivitub ühes tunnis, aga lülitub välja teises, siis tema tööaega on vaja erinevatesse tundidesse jagada. Siis on teada, kui kaua töötas kolmas pump igas tunnis oktoobri kuu jooksul. (vt Joonis 1.7)

		tund																									
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
k u p ä e v	1	0	0	0	0	0	0	0	0	0	2092	1285	0	0	0	0	0	0	0	519	2818	0	0	0	0		
	2	0	0	0	0	0	2765	603	0	0	0	0	0	0	0	0	1310	2052	0	0	0	0	0	0	0		
	3	0	0	3357	0	0	0	0	0	0	0	0	0	2024	1345	0	0	0	0	0	0	0	0	0	645	2738	
	4	0	0	0	0	0	0	0	0	0	2989	404	0	0	0	0	0	0	0	0	1873	1489	0	0	0	0	
	5	0	0	0	0	0	1410	1957	0	0	0	0	0	0	0	0	2280	1101	0	0	0	0	0	0	0	0	
	6	1726	1665	0	0	0	0	0	0	0	2460	983	0	0	0	0	0	0	624	2847	0	0	0	0	0	0	
	7	0	0	3203	267	0	0	0	0	0	0	2805	665	0	0	0	0	0	0	2871	604	0	0	0	0	0	
	8	0	0	3301	199	0	0	0	0	0	549	2963	0	0	0	0	0	0	2030	1496	0	0	0	0	0	0	
	9	98	3432	0	0	0	0	0	0	2031	1518	0	0	0	0	0	449	3057	0	0	0	0	0	0	0	2612	
	10	909	0	0	0	0	0	1201	2313	0	0	0	0	0	0	3452	19	0	0	0	0	0	0	2071	1406	0	
	11	0	0	0	0	775	2739	0	0	0	0	0	0	3021	502	0	0	0	0	0	1683	1837	0	0	0	0	
	12	0	0	281	3239	0	0	0	0	0	0	2409	1089	0	0	0	0	0	802	2676	0	0	0	0	0	0	
	13	0	2594	896	0	0	0	0	0	740	2746	0	0	0	0	0	0	2394	1013	0	0	0	0	0	0	324	
	14	3115	0	0	0	0	0	0	1526	1932	0	0	0	0	0	0	2583	883	0	0	0	0	0	0	0	3312	
	15	161	0	0	0	0	0	378	3086	0	0	0	0	0	0	920	2554	0	0	0	0	0	0	0	1181	2292	
	16	0	0	0	0	0	0	1324	2136	0	0	0	0	0	0	1362	2104	0	0	0	0	0	0	0	0	1223	2255
	17	0	0	0	0	0	0	851	2638	0	0	0	0	0	0	295	3171	0	0	0	0	0	0	0	0	0	3226
	18	230	0	0	0	0	0	0	2377	1094	0	0	0	0	0	0	1360	2110	0	0	0	0	0	0	0	0	279
	19	3193	0	0	0	0	0	0	0	2690	786	0	0	0	0	0	0	1372	2067	0	0	0	0	0	0	0	0
	20	20	3418	0	0	0	0	0	0	2022	1403	0	0	0	0	0	0	0	372	2996	0	0	0	0	0	0	0
	21	0	0	2186	1233	0	0	0	0	0	0	157	3227	0	0	0	0	0	0	0	1650	1750	0	0	0	0	0
	22	0	0	0	0	2891	522	0	0	0	0	0	0	494	2929	0	0	0	0	0	0	0	0	1740	1667	0	
	23	0	0	0	0	0	0	3107	330	0	0	0	0	0	0	0	798	2659	0	0	0	0	0	0	0	1932	
	24	1508	0	0	0	0	0	0	0	2941	485	0	0	0	0	0	0	0	327	3106	0	0	0	0	0	0	0
	25	0	1312	2094	0	0	0	0	0	0	0	2599	801	0	0	0	0	0	0	280	3103	0	0	0	0	0	0
	26	0	0	0	1475	1930	0	0	0	0	0	0	0	2660	744	0	0	0	0	0	0	0	173	3224	0	0	
	27	0	0	0	0	0	1143	2279	0	0	0	0	0	0	0	1909	1500	0	0	0	0	0	0	0	0	2769	
	28	639	0	0	0	0	0	0	0	3403	0	0	0	0	0	0	0	669	2745	0	0	0	0	0	0	0	0
	29	0	1393	2050	0	0	0	0	0	0	0	2090	1376	0	0	0	0	0	0	0	2832	667	0	0	0	0	0
	30	0	0	0	3488	0	0	0	0	0	0	0	773	2717	0	0	0	0	0	0	0	0	1662	1795	0	0	0
	31	0	0	0	0	0	2729	703	0	0	0	0	0	0	0	288	3122	0	0	0	0	0	0	0	1599	1801	

Joonis 1.7 Kolmanda pumba töö oktoobrikuu jooksul

Kolmanda pumba tunni töö oktoobri kuus salvestatakse kahemõõtmelisse massiivi "mas[kuupäev][tund]", sekundites. Edasi analüüs toimub nende andmete alusel. (vt Joonis 1.8)

```

13 #Pumba käigu tunniplanograam, oktoobri eest
14 mas = [0] * 32
15 for i in range(0, 32):
16     mas[i] = [0] * 24
17     i = 0
18 while i < len(mlist):
19     time = mlist[i]
20     if i and i % 2:
21         if mlist[i].hour == mlist[i - 1].hour:
22             mas[mlist[i].day][mlist[i].hour] = (mlist[i] - mlist[i-1]).seconds
23         else:
24             d = copy.copy(mlist[i])
25             d = d.replace(minute=00, second=0)
26             mas[mlist[i-1].day][mlist[i-1].hour] = (d - mlist[i - 1]).seconds
27             mas[mlist[i].day][mlist[i].hour] = (mlist[i] - d).seconds
28     i += 1
29     i += 1

```

Joonis 1.8 Kolmanda pumba töögraafiku koostamine oktoobri kuus

Nüüd tuleb suureima ööpäevase põhjavee juurdevoolu mahu välja selgitada. Selle jaoks tuleb iga oktoobri ööpäeva elektri koguhinda arvutada.(3.1)

$$\sum_{i=0}^{23} t_i h_i , \tag{3.1}$$

kus

$t_i$  – pumba tööaeg igas ööpäevas, sekundites;

$h_i$  – börsi tunnielektrihind, iga ööpäeva tunni eest;



Suurema ööpäevase põhjavee sissevoolu leidmist teostatakse Python-i skripti kaudu. (vt Joonis 1.9)

```
30     # max päeva veesissevool
31     m = []
32     for j in range(1, 32):
33         sum = 0
34         for i in range(0, 24):
35             sum += mas[j][i]
36         m.append(sum)
37     m.sort()
38     print(m[len(m)-1])
```

Joonis 1.9 Oktoobri kuu päev suurima põhjavee juurdevoolu mahuga

Suurim pumba ööpäeva tööaeg oli 13 351 sekundit, elektrimootori võimsus on 560 kW. Elektritarbimine on:

$$13\,351 * 560 / 3\,600 = 2\,077 \text{ kW/h}$$

Pumpla nelja pumba elektrimootorite koguvõimsus on:

$$560\text{kW} + 560\text{kW} + 800\text{kW} + 800\text{kW} = 2\,720 \text{ kW}$$

Sellest tulenevalt võib järeldada, et lülitades korraga neli pumba ööpäeval kindlal perioodil, saab maapinnale pumbata kogu põhjavee juurdevoolu mahu kogurist. Järelkult saab pumpade käivitamiseks valida tunde kõige odavama elektri hinnaga, et pumpamisel elektrienergia tarbimisest tulenevaid kulutusi vähendada.

Teine analüüsi ülesanne on kulude võrdlemine vana ja uue automatiseerimismeetodi puhul. Kodulehelt <https://dashboard.elering.ee/v2/api-docs> võib näha statistilisi andmeid börsi tunnielektrihindadest ja JSON faili kujul andmed alla laadida. Oktoobri tunnihinnad salvestatakse loendisse "prices". Tunnihinnad on väljendatud eurodes megavatt-tunni eest. (vt Joonis 1.10)

```
39     #JSON failist oktoobri börsi tunnihindu lugemine
40     with open("my.json", "r") as read_file:
41         dat = json.load(read_file)
42         data = copy.copy(dat['data'])
43         ee = copy.copy(data['ee'])
44         prices = []
45         for pr in ee:
46             prices.append(float(pr['price']))
```

Joonis 1.10 JSON failist andmete lugemine ja loendi loomine

Vana automatiseerimise skeemi puhul elektri tarbimiskulude leidmiseks summeeritakse kõik maksekulud elektritarbimise eest iga tunni kohta oktoobri kuu jooksul. Arvutus toimub börsi elektritunnihinna ning kolmanda pumba planogrammi alusel. (3.2)

$$\sum_{p=1}^{31} h_p , \quad (3.2)$$

kus

$h_p$  – pumpamise elektritarbimise rahalised kulud iga oktoobri ööpäeva eest;

Valemi tulemuse leidmiseks kasutakse Pythoni skripti. (vt Joonis 1.11)

```

47     #elektri kulud vana automaatiseerimise kujul
48     ii = 0
49     sum = 0
50     for j in range(1, 32):
51         for i in range(0, 24):
52             sum += mas[j][i]*prices[ii]*0.56/3600
53             ii +=1
54     print(sum)

```

Joonis 1.11 Elektri arve vana automaatiseerimise skeemi puhul oktoobri kuu eest

Tulemus on 7 900.20 eurot.

Uue automatiseerimise meetodi juurutamisest tulenevate elektritarbimise rahaliste kulude leidmiseks on vaja teada kõige odavama ööpäeva tunni hinnalist väärtust nelja pumba üheaegselt käivitamiseks. Arvestus toimub börsi tunnielektrihinna ning kõikide pumpade koguvõimsuse alusel. (3.3)

$$\sum_{p=31}^1 P h_{mp} t_p , \quad (3.3)$$

kus

$P$  – pumpade koguvõimsus, 2 720 kW;

$h_{mp}$  – madalam ööpäeva börsi elektrienergia tunnihind;

$t_p$  – pumba tööaeg igas ööpäeva tunnis;

Arvutus toimub ka Pythoni rakenduse abil. (Joonis 1.12)

```
55     # elektri kulud uue automaatiseerimise kujul
56     ii = 0
57     mm = []
58     sum1 = []
59     for j in range(1, 32):
60         sum = 0
61         sum1.clear()
62         for i in range(0, 24):
63             sum += mas[j][i]
64             sum1.append(prices[ii])
65             ii += 1
66         sum1.sort()
67         mm.append(sum*0.56/3600*sum1[0])
68     w = 0.0
69     for i in mm:
70         w += i
71     print(w)
```

Joonis1.12 Elektri arve uue automaatiseerimisskeemi puhul oktoobri kuu eest

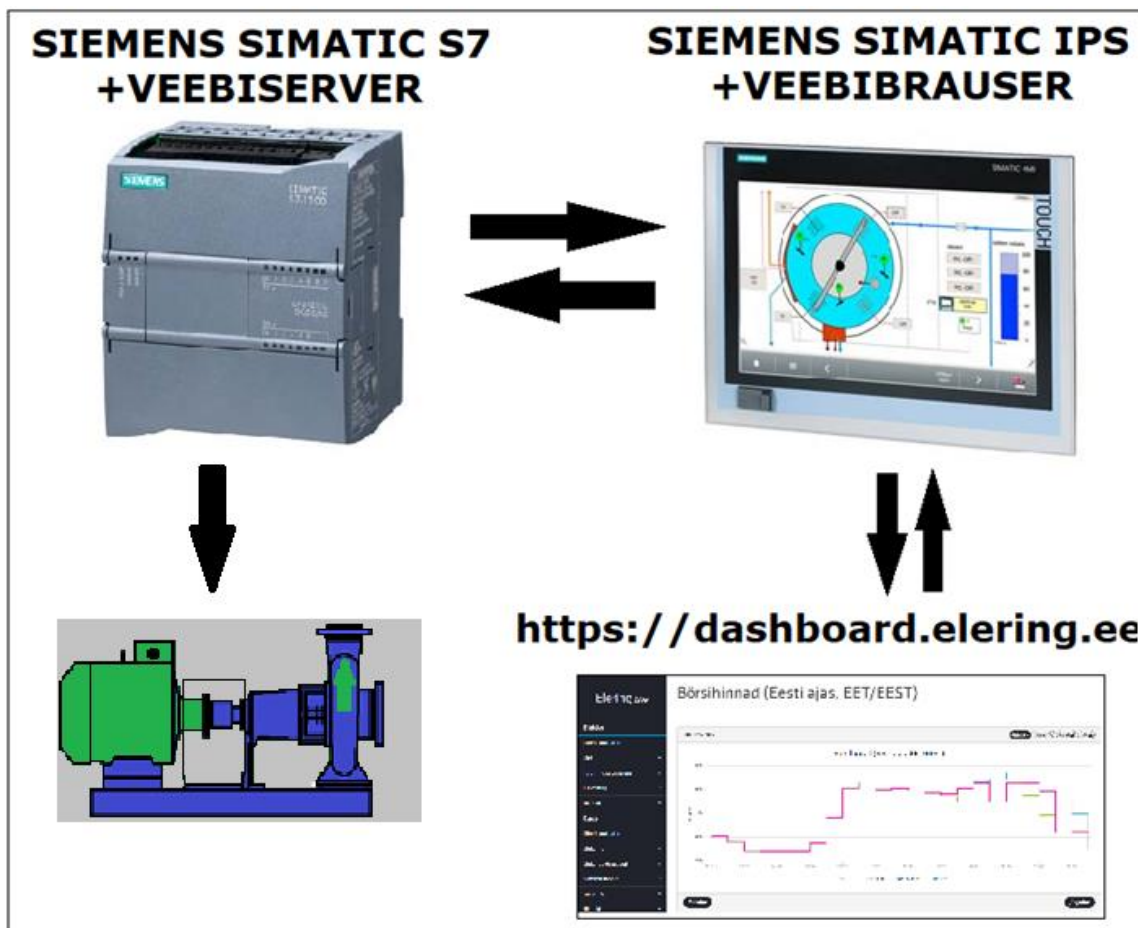
Tulemuseks on 2614.28 eurot, mis on 66.9% võrra vähem, kui vana automatiseerimismeetodi puhul (7 900.20 eurot).

Uue automatiseerimisskeemi kasutamisega saavutatakse positiivne majanduslik efekt. Automatiseerimiseks kasutatakse olemasolevaid pumbajaama seadmeid. Muutub ainult tarkvara ning paigaldatakse tööstusarvuti protsessi juhtimiseks ning visualiseerimiseks.

## 2 PROTSESSI JUHTIMISSÜSTEEM

### 2.1 Protsessi juhtimissüsteemi kirjeldus

Automaatika skeemi realiseerimiseks kasutatakse kontrolleriit Simantic S7, kus on olemas veebiserver ja HTML veebirakendus, mis käivitub Siemensi IPS tööstusarvutis veebibrauseri kaudu. Rakendus võtab vastu börsihindade andmeid <https://dashboard.elering.ee/et> kodulehelt ja nende alusel otsustatakse, millal pumbad tuleb käivitada. (vt Joonis 2.1)



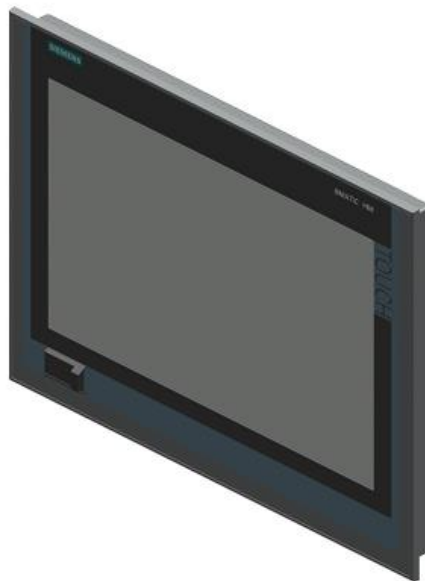
Joonis 2.1 Protsessi juhtimissüsteem

Veebirakendus realiseeritakse HTML ning JavaScript-I programmeerimiskeele abil. (vt Lisa 1)

## 3 PROJEKTI REALISEERIMINE

### 3.1 Seadmete ja tehnoloogiate valik

Kuna automatiseerimise eesmärk on pumpamise elektritarbimisest tulenevate rahaliste kulude vähendamine, siis projektis kasutatakse kõiki kaheksanda pumbajaama olemasolevaid seadmeid ning automaatika vahendeid, minimiseerides lisakulude teket. Lisatakse ainult uus tööstusarvuti ja tarkvarasse režiimi "ECO" süsteemi juhtimiseks ja visualiseerimiseks. Tööstusarvuti SIMATIC IPC277E paigaldatakse pumbajaama automaatikakilbile. [4] Tööstusarvuti internetibrauser laadib HTML rakenduse kontrolleri SIMATIC S7 serverist. (vt Joonis 3.1)



Joonis 3.1 Tööstusarvuti SIMATIC IPC277E [4]

### 3.2 Kontrolleri programmeerimine

Kontrolleri programmeerimiseks kasutatakse Siemens AG arenduskeskkonda Tia Portal versiooni 15. [1] Automatiseerimise algoritm realiseeritakse LAD programmeerimiskeele kaudu. Pumpade juhtimise protsess koosneb neljast režiimist: "0", "Käsitsi", "Auto", "ECO" – uus režiim, mille loomine on selle projekti ülesanne. Režiimide teostamiseks kasutakse kontrolleri SIEMENS SIMATIC S7-1200. [5] (vt Joonis 3.2)



Joonis 3.2 Kontroller SIEMENS SIMATIC S7-1200 [5]

### 3.2.1 Veebiserver

Kontrolleris on realiseeritud sisemine veebiserver. Kasutajate HTML veebilehtede abil, mis laaditakse üles veebiserverile, saab luua rakendusi seadmaks nende tööd distantiliselt läbi veebibrauseri. Rakendus omab ligipääsu kontrollerile ning on suuteline infovahetuseks, see omakorda võimaldab tööprotsessi juhtida ning luua visualisatsioon. Veebiserveri ühendamisel Tia Portal keskkonnas tuginedes esitatud HTML dokumentidele ja muudele tugifailidele genereeritakse ja lisatakse kontrolleri programmile www plokk. (vt Tabel 3.1)

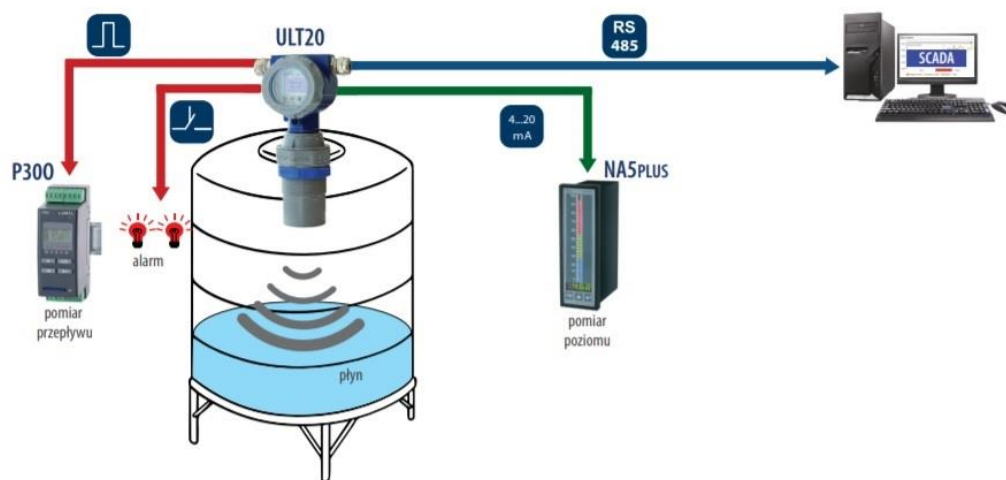
Tabel 3.1 Projekti failid

Faili nimi	Kirjeldus	Faili tüüp
index.html	HTML rakendus, kus JavaScript-i algoritmi kaudu realiseeritakse põhiline 'ECO' tööprotsessi loogika ning visualiseerimine.	HTML dokument
IOpump1.html	Esimese pumba tagasiside: 1-töötab, 0-ei tööta.	HTML dokument
IOpump2.html	Teise pumba tagasiside	HTML dokument
IOpump3.html	Kolmanda pumba tagasiside	HTML dokument
IOpump4.html	Neljanda pumba tagasiside	HTML dokument
IOrezim.html	Töörežiim: 0, 1-Käsitsi, 2-Auto, 3-ECO	HTML dokument
IOstart.html	Käsk pumpade käivitamiseks 'ECO' töörežiimis: 0-Start, 1-Stop	HTML dokument
IOveemaht_p.html	Vee maht veekoguris, protsentides	HTML dokument
IOveemaht_m.html	Vee maht veekoguris, m3	HTML dokument
Visio.jpg	Tagataust protsessi visualiseerimiseks	Pilt
PumpStop.png	Pump on "Stop" seisundis	Pilt
PumpStartGIF.gif	Pump on "Start" seisundis	Animeeritud pilt

### 3.2.2 Taseme andur

Veekoguri veetaseme mõõtmiseks kasutatakse ultraheli ULT20 tüüpi tasemeandurit.

ULT20 on mõeldud kauguse/taseme mõõtmiseks ning pärast mõõdetud vahemaa väärtust töötlevate funktsioonide programmeerimist saab seda kasutada mahutite täituvuse mõõtmiseks, vooluhulga ja taseme mõõtmiseks avatud kanalites. [2] Sisseehitatud LCD ekraan, mis koos nuppudega loob lokaalse kasutajaliidese, võimaldab muuta tööparameetreid ja vaadata seadme hetkeseisu, s.h. lugeda mõõdetud väärtusi (vt Joonis 3.3) [2].



Joonis 3.3 Ultrahelianduri ühendamise skeem [2]

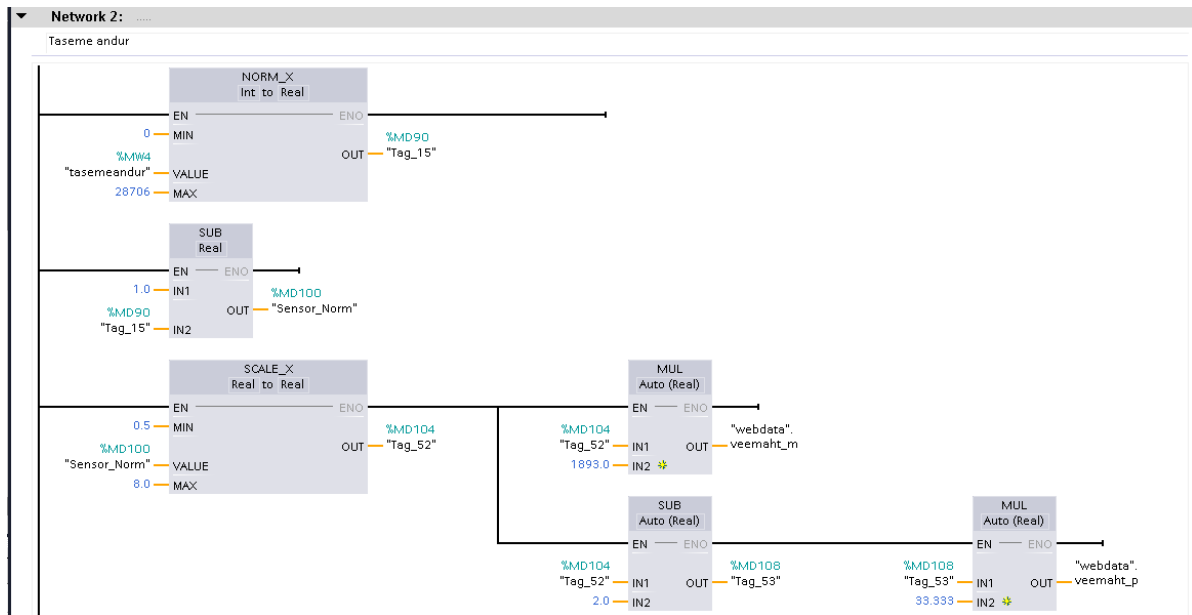
Põhiparameetrid on

- vedelate ja pulbriliste materjalide taseme mõõtmine vahemikus kuni 8 m, automaatne temperatuuri kompensatsioon;
- mahuti täituvuse või vooluhulga arvutamine avatud kanalites;
- standartne galvaaniliselt eraldatud analoogväljund 0/4..20 mA;
- digitaalliides rs-485 modbus RTU protokolliga;
- 2 häirereleed;
- siseandmed ja seadistusmälu;
- kaks 32-punktilist individuaalset tunnust (funktsioonide ümberarvutamine). [2]

Analoogsisendi moodulid AI 8x13-bitine 6ES7 231-4HF32-0XB0 kasutatakse anduri ULT-20 kontrolleriiga ühendamiseks. [3]

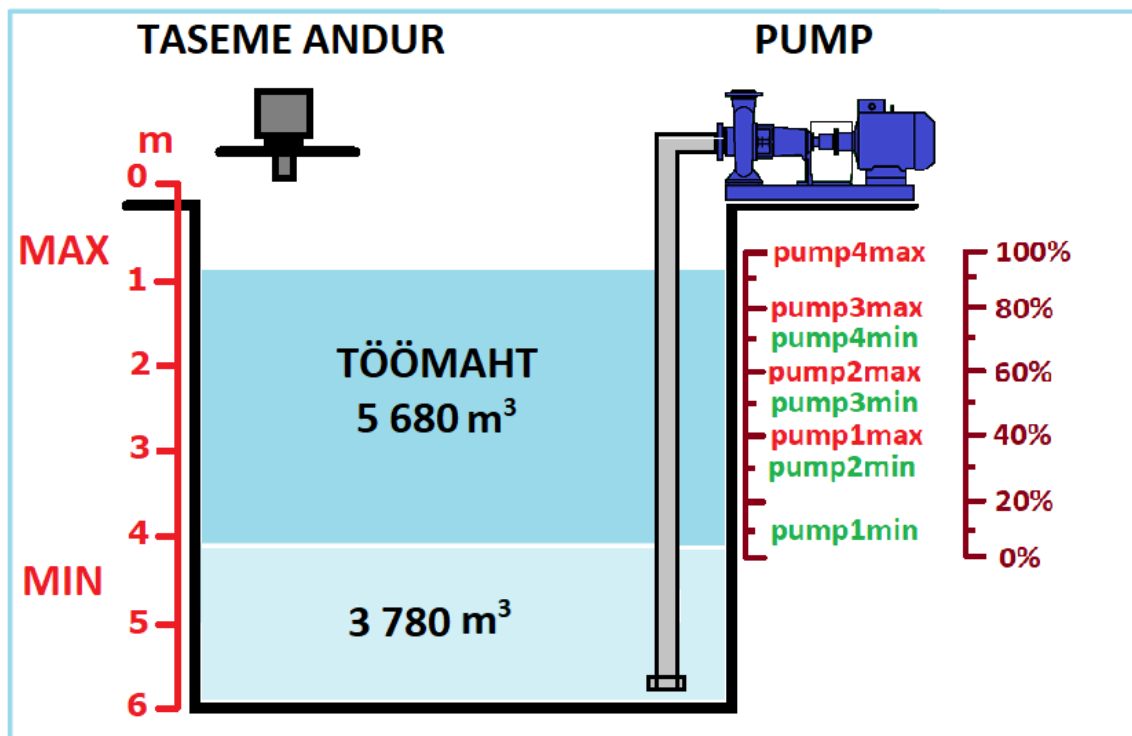
Ultraheliandur on ühendatud 4-20mA analoogkanaliga, mille minimaalne väärtus on 0 ja maksimaalne väärtus on 27 648 mõõtühikut. Programmiplokk NORM\_X teisendab tegeliku sisendväärtuse INT normaliseeritud väärtuseks REAL (0,0 kuni 1,0).

Väärtuse saamiseks protsentides ja m<sup>3</sup>-s kasutatakse programmiplokki SCALE\_X. (vt. Joonis 3.4)



Joonis 3.4 Ultraheli anduri signaali skaleerimine

Skaleerimisel on vaja arvestada minimaalse ja maksimaalse veetasemega veekoguris anduri asukoha suhtes. Andmed salvestatakse andmeplokki DB1 "webdata", muutujad "veemaht\_p" ja "veemaht\_m". Veekoguri töömaht on 5 680 m<sup>3</sup>, mis võrdub 1 kuni 4 meetrit tasemeanduri väärtusega või 0 kuni 100 protsenti. (vt. Joonis 3.5)



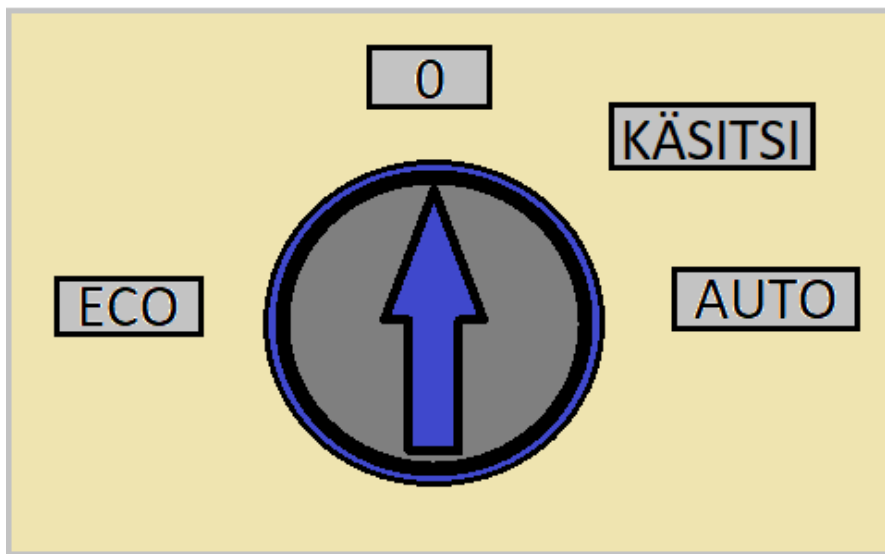
Joonis3.5 Veetased pumppla veekoguris



### 3.2.3 Käsitsi töörežiim

Manuaalrežiimi kasutatakse pumpade kontrollimiseks, näiteks pärast remonti või hooldustöid korraliseks käivituseks. Juhtimiskilp käsitsi juhtimiseks asub pumpade lähedal, tänu millele võib pumpade tööd visuaalselt kohal kontrollida.

Pumpade käivitamise režiimi valimiseks peajuhtimiskilbil on olemas töörežiimi lüliti. (vt. Joonis 3.6)

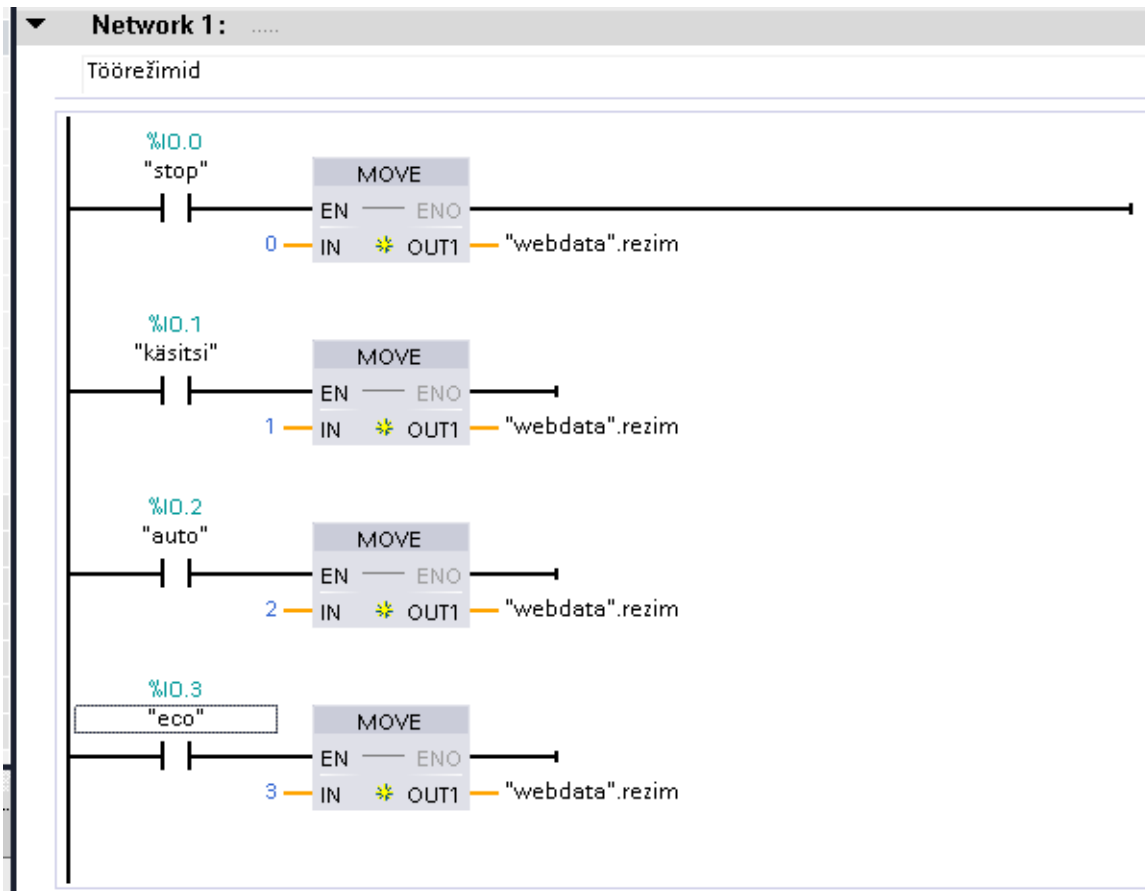


Joonis 3.6 Töörežiimi lüliti automaatikakilbil

Töörežiimi lüliti võib olla sisse lülitatud ühte neljast positsioonist:

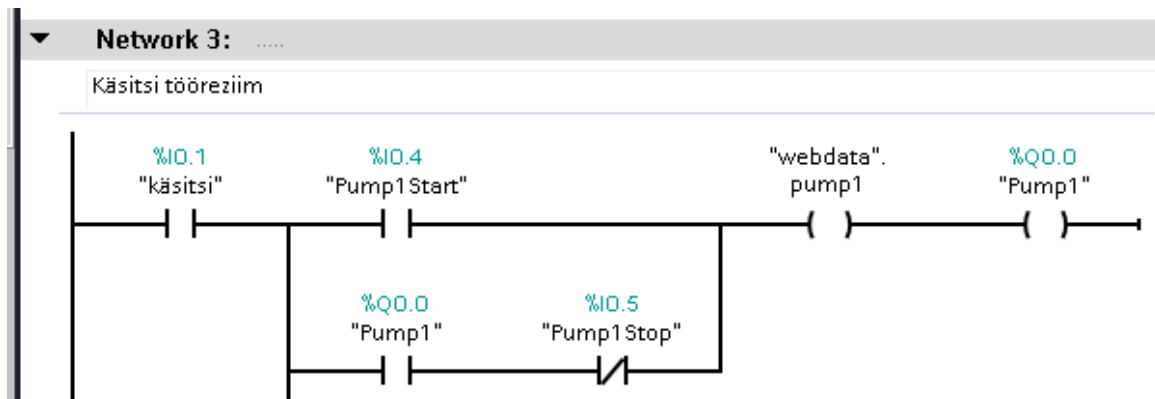
- "0" – pumbajaam ei tööta;
- "käsitsi" – manuaalrežiim;
- "auto" – automaatrežiim, pumbad töötavad tuginedes veetasemele;
- "eco" – automaatrežiim, pumbad töötavad arvestades elektri börsihinda.

Töörežiimi lüliti asend salvestatakse "webdata.rezim" muutujasse. (vt. Joonis 3.7)



Joonis 3.7 Töörežiimide lüliti realiseerimine

Töörežiimi lüliti on "käsitsi" asendis. Kui nupp "Pump1Start" vajutatud, siis esimese pumba kontaktor on sisse lülitatud ja pump käivitub. Pumba väljalülitamiseks on vaja vajutada nupule "Pump1Stop". Esimese pumba kontaktor lülitatakse välja ja pump ei tööta. (vt. Joonis 3.8)



Joonis 3.8 Töörežiimi "käsitsi" realiseerimine

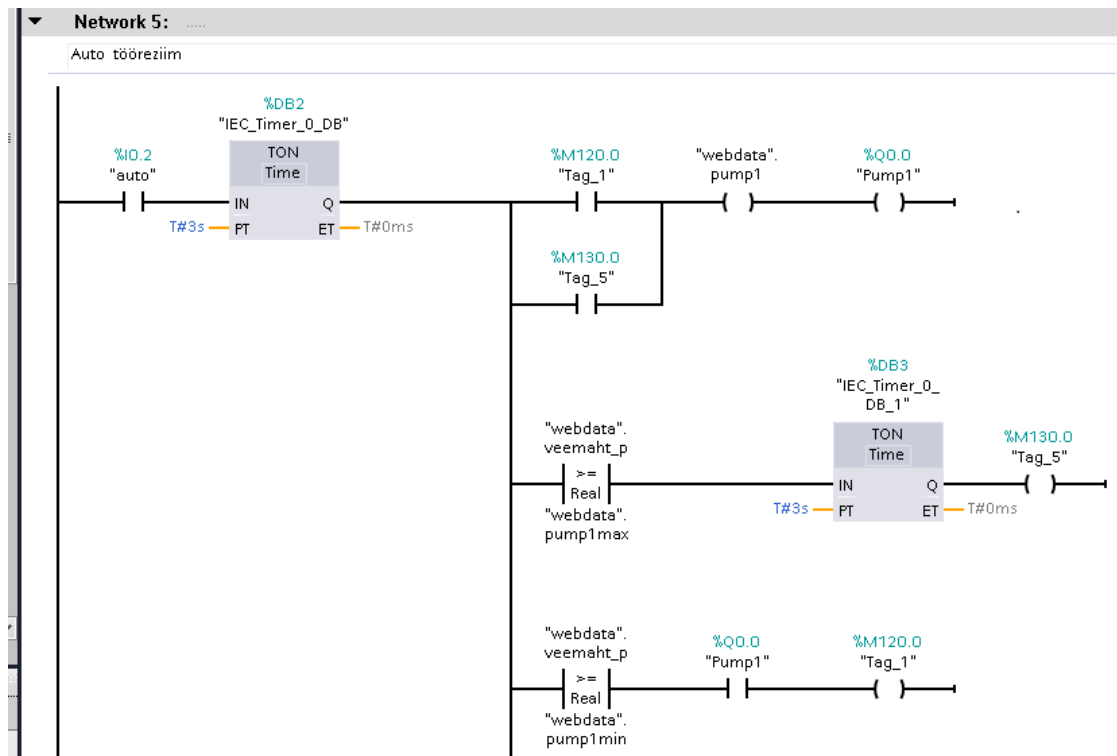
### 3.2.4 "Auto" töörežiim

Automaatne töörežiim on mõeldud pumpade automaatseks käivitamiseks, olenevalt veetasemest allmaa veekoguris. Iga neljast pumbast käivitub ja seiskub sõltuvalt oma teatud veetasemest. Taseme min/max väärtused protsentides hoitakse muutujates, mis asuvad "webdata" andmeplokis. (vt Tabel 3.2)

Tabel 3.2 Andmeploki "webdata" veetaseme protsessimuutujad

Protsessi muutuja	Andme tüüp	Kirjeldus	Väärtus, protsentides (veetase)
Pump1max	Int	Pump1 veetase käivitamiseks	40
Pump1min	Int	Pump1 veetase peatamiseks	70
Pump2max	Int	Pump2 veetase käivitamiseks	60
Pump2min	Int	Pump2 veetase peatamiseks	50
Pump3max	Int	Pump3 veetase käivitamiseks	80
Pump3min	Int	Pump3 veetase peatamiseks	30
Pump4max	Int	Pump4 veetase käivitamiseks	100
Pump4min	Int	Pump4 veetase peatamiseks	10

Töörežiimi lüliti on "auto" asendis. Kui vee tase veekoguris võrdub, või kõrgem "pump1max" väärtusega kolme sekundi jooksul, siis pump käivitub ja "webdata.pump1" muutuja saab „true" väärtuse. Paus kolme sekundi jooksul on vajalik, et täpselt teada - töörežiimi lüliti pööramine on lõpetatud ja töörežiim valitud. Kui veetase langeb alla "pump1min" väärtuse, siis pump lülitub välja ja "webdata.pump1" muutuja saab "false" väärtuse. Kui esimene pump pumpab, kuid veetase ei lange (näiteks, suurenes juurdevool sõltuvalt n.t. aastaajast), siis käivitub teine pump ja nii edasi. (vt. Joonis 3.9)



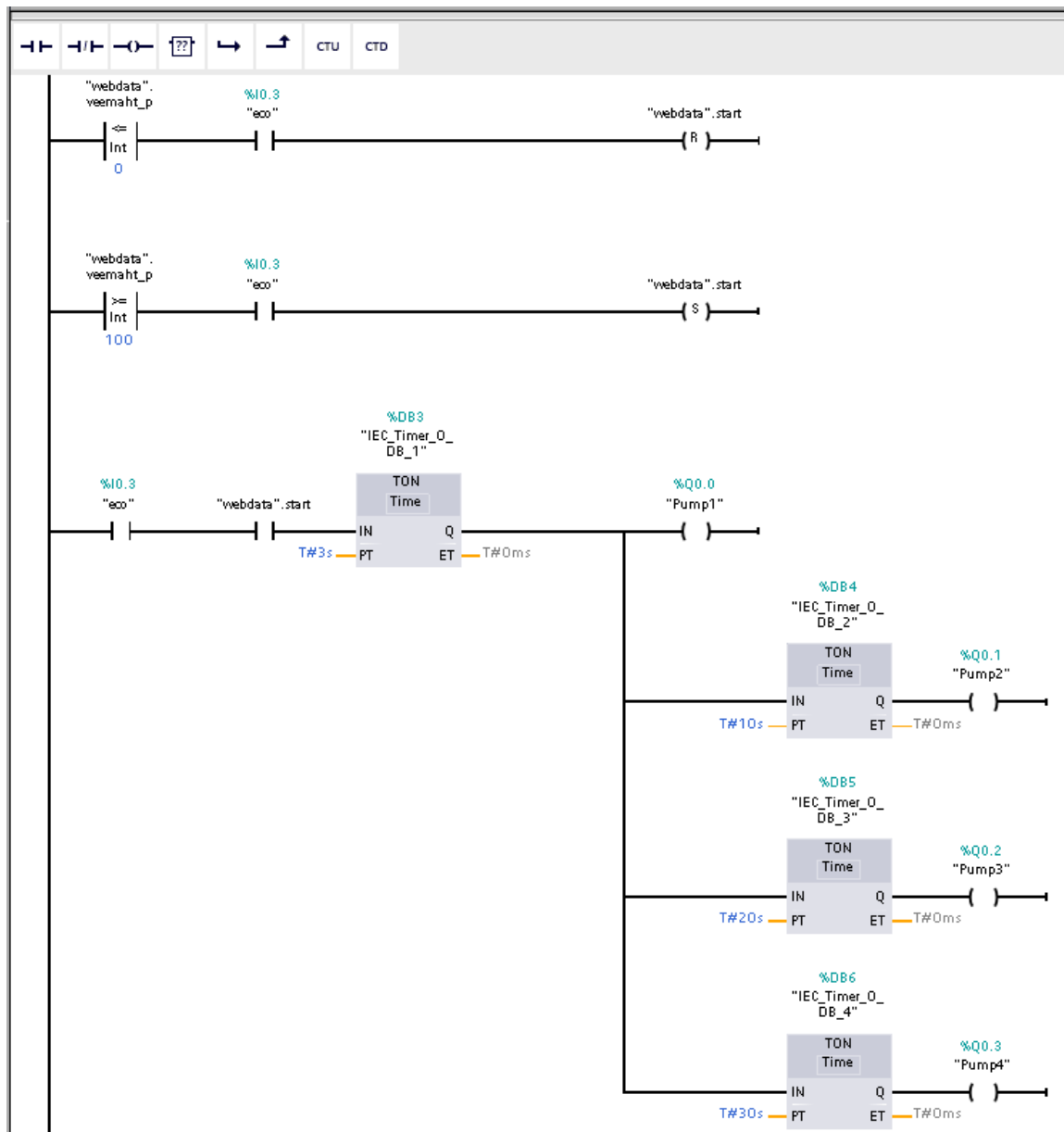
Joonis 3.9 Auto töörežiimi realiseerimine

### 3.2.5 "ECO" töörežiim

"Eco" töörežiimi kaudu realiseeritakse selle projekti põhieesmärk: pumbajaama automatiseerimine, arvestades elektri börsihinda. "ECO" töörežiimis pumpade käivitamiseks on vaja positiivset signaali veebirakendusest, kus otsustatakse, sõltuvalt börsi elektritunnihinnast, millal pump sisse lülitada. Andmeplokis DB1 asub muutuja "start", selle "true" väärtuse alusel käivituvad järjestikku kõik neli pumpa.

Töörežiimi lüliti on "ECO" asendis. Muutuja "'webdata'.start" saab "true" väärtuse. Pärast kolme sekundit lülitakse sisse esimese pumba elektrimootori kontaktor ja pump käivitub. Paus kolme sekundi jooksul on vajalik, et täpselt teada - töörežiimi lüliti pööramine on lõpetatud ja töörežiim valitud. Pärast kümnet sekundit käivitub teine pump j.n.e. Paus kümne sekundi jooksul on vajalik, et välistada elektrivõrgustiku ülekoormamist pumpade elektrimootorite üheaegsete käivitusvooludega.

Kõik neli pumpa töötavad, kuni veetase veekoguris "webdata'.veemaht\_p" langeb nullväärtuseni ja pärast seda toimub pumpade tööst välja viimine. (vt. Joonis 3.10)



Joonis 3.10 "Eco" töörežiimi realiseerimine

Kui veetase kasvab 100% või suuremaks, siis muutuja "webdata'.start" saab "true" väärtuse. Kõik pumbad käituvad ja töötavad sellisel juhul. See olukord võib tekkida erinevatel põhjustel, need on

- puudub side Elering serveriga või server ei vasta;
- ei käivitu üks või mitu pumpa;
- liiga suur põhjavee juurdevool.

Kui "ECO" töörežiimi pole eri tõrgete või tingimuste põhjusel võimalik kasutada, siis operaator lülitab sisse "AUTO" režiimi.

### 3.3 HTML rakenduse kirjutamine

HTML rakendus index.html on loodud selleks, et teha otsus pumpade käivitamiseks režiimis "ECO", samuti protsessi visualiseerimiseks. Rakendus suhtleb kontrolloriga. Annab käsu käivitada pumbad, kui börsi elektri tunnihind on madalaim. Samal ajal jälgib põhjavee juurdevoolu mahtu veekogurisse, vältides ületäitumist. Veetaseme juhtimine veekoguris on diskreetne, pumpade käivitamise otsus tehakse iga uue tunni alguses, kuna igale tunnile määratakse elektrikörsil oma hind. Börsi hindu uuendatakse üks kord ööpäevas. Nii saab 12 kuni 36 tundi ette teada elektrihindu ja käivitada pumbad madalaima hinnaga. Veekoguri ületäitumise vältimiseks on vaja kontrollida ka põhjavee juurdevoolu, mida pidevalt jälgitakse.

Rakendusel on kolm tsükliit.

Iga sekund on

- protsessiandmete vastuvõtt kontrollierilt ja protsessi visualiseerimine (pumba olek, veetase veekoguris, juhtimisrežiim).

Iga tund on

- Elering-i serverilt börsi tunnielektrihinna saamine ja visualiseerimine;
- tunni veevoolu arvutamine ja visualiseerimine;
- pumpade käivitamise aja arvutamine ja visualiseerimine;
- vajadusel käsk kontrollierille pumpade käivituseks;

Esimisel rakendusel käivitamiseks või uuenduseks on

- samad sammud nagu iga tund, ainult tunni vee juurdevool saab keskmise väärtuse  $193.0 \text{ m}^3/\text{h}$ .

### 3.3.1 Side kontrolleriiga

Andmevahetus kontrolleriiga toimub asünkroonses režiimis, ilma veebilehte uuesti laadimata. "Jquery 3.6.0" teegi meetodite kaupa realiseeritakse asünkroonne andmeedastus, samuti dünaamilised muudatused lehe stiilides ja parameetrites. Kontrolleri protsessi muutujad asuvad andmeplokis "webdata". (vt Tabel 3.3)

Tabel 3.3 Protsessi muutujad ja sellega seotud html-dokumendid

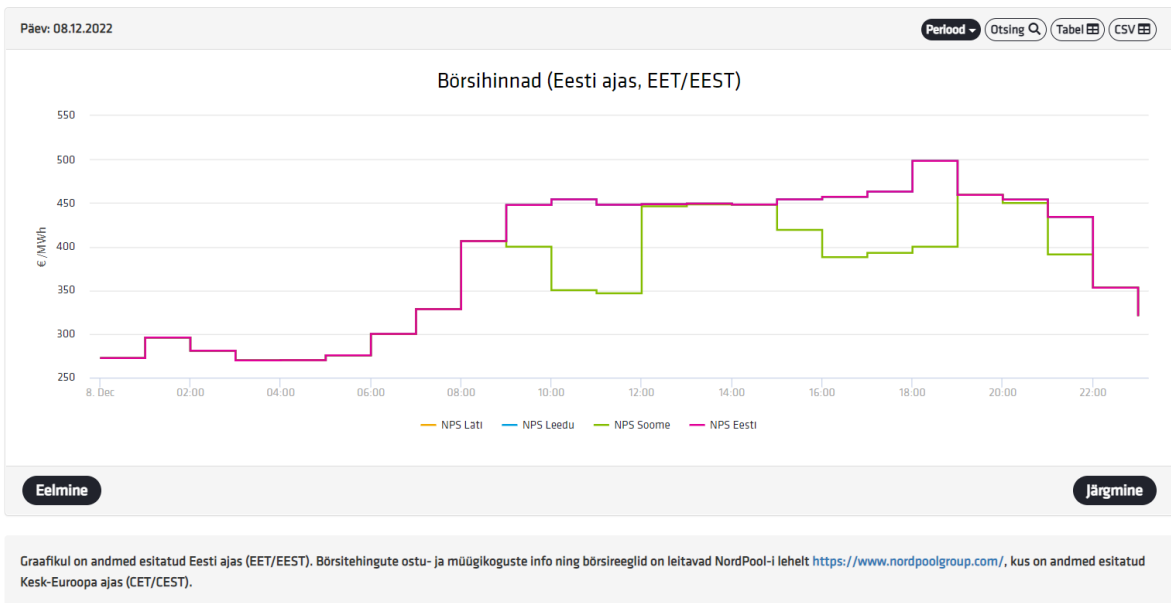
Muutuja nimetus	Muutuja tüüp	Muutuja väärtused	HTML dokument
rezim	Int	Töörežiim: 0, 1-Käsitsi, 2-Auto, 3-ECO	IOrezim.html
start	Bool	Käsk pumpade käivitamiseks: 'True'-Start, 'False'-Stop	IOstart.html
veemaht_p	Int	Veemaht veekoguris, protsentides	IOveemaht_p.html
veemaht_m	Real	Veemaht veekoguris, m3	IOveemaht_m.html
pump1	Bool	Pump1 tagasiside: 1-töötab, 0-ei tööta.	IOpump1.html
Pump2	Bool	Pump2 tagasiside: 1-töötab, 0-ei tööta.	IOpump2.html
Pump3	Bool	Pump3 tagasiside: 1-töötab, 0-ei tööta.	IOpump3.html
Pump4	Bool	Pump4 tagasiside: 1-töötab, 0-ei tööta.	IOpump4.html

Nende edastamiseks HTML rakendusse ja tagasi kasutatakse täiendavaid HTML-dokumente, millega saab muutujate väärtust teada või muuta. Näiteks, muutuja "webdata.rezim" järgi kasutakse HTML dokument "IOrezim.html", mis koosneb koodist: "<!-- AWP\_In\_Variable Name="webdata".rezim' -->:="webdata".rezim:". Teiste muutujate jaoks luuakse samad HTML dokumendid ja laaditakse need kontrolleriisse.

### 3.3.2 Veebi API.

Börsihind kujuneb elektribörsil, milleks Põhjamaades on Nord Pool. Oluline aspekt on aga see, et tunnipõhised elektrihinnad on teada järgmise ööpäeva perioodiks. Tunnihinnad on diskreetsed, iga tund omab hinda, mis väljendub eurodes megavatt-tunni kohta. Teatud perioodil saab börsihinnad Eleringi infoportaalist leida.

Elering on ettevõtte, mis haldab Eesti Vabariigi elektri edastamise põhivõrke, ehk teisi sõnu võrguettevõtja.(vt Joonis 3.11)



Joonis 3.11 Elering OÜ infoportaal, börsihinnad

Börsihindu saab ka HTML-rakendusest teada, Eleringi serverisse päringu saates. Selleks kasutakse jQuery teegi meetodit \$.ajax(), millele tuleb järgmisi andmeid edastada:

- url – veebibrauser edastab Eleringi serverile aadressi, mille põhjal Eleringi server väljastab vastava info;
- method – kasutakse meetodit "get", serveri andmete päring;
- dataType – andmetüüp, HTML;
- success: function(json){} – funktsioon, mis käivitub eduka andmete edastamise puhul;
- error: function (jqXHR, exception) {} – funktsioon, mis käsitleb vigu.

URL koosneb "https://dashboard.elering.ee/api/nps/price?" aadressist ja ajaperioidist. Ajaperioid algab praegusest kuupäevast ning kellaaajast, ja kestab 36 tundi. URL-i moodustamisel tuleb arvestada, et näiteks kuupäevas tundide ja kuude väärtuste ees seisab null, mis tuleneb Java Script-i date-tüübist. Samuti kuud date-tüübist on märganud 0 kuni 11 arvuga.

URL võib välja näha selline: „https://dashboard.elering.ee/api/nps/price?start=2020-05-31T20%3A59%3A59.999Z&end=2020-06-30T20%3A59%3A59.999Z“

Server tagastab börsihinna loendi JSON-i kujul.

Kui andmete saamine Eleringi serverilt ebaõnnestus, siis käivitub tõrkedagnostika, mis alert meetodi abil kuvab ekraanile veateate koos vea tüübiga.



Selle tulemusel operaator saab täpselt teada, mille põhjusel ei toimunud andmete edastamine ja vajadusel viib pumbad üle „Auto“ töörežiimile (see režiim ei sõltu serveriga toimiva side olemasolust) (vt Joonis 3.12)

```
error: function (jqXHR, exception) {
  if (jqXHR.status === 0) {
    alert('Not connect. Verify Network.');
```

```
  } else if (jqXHR.status == 404) {
    alert('Requested page not found (404).');
```

```
  } else if (jqXHR.status == 500) {
    alert('Internal Server Error (500).');
```

```
  } else if (exception === 'parsererror') {
    alert('Requested JSON parse failed.');
```

```
  } else if (exception === 'timeout') {
    alert('Time out error.');
```

```
  } else if (exception === 'abort') {
    alert('Ajax request aborted.');
```

```
  } else {
    alert('Uncaught Error. ' + jqXHR.responseText);
  }
}
```

```
});
```

Joonis 3.12 Törkediagnostika realiseerimine

Rakendus saab andmeid elektrienergia hindadest börsil Eleringi serverilt, kuid on võimalus saada samad andmed ka börsi Nord Pool serverilt, läbides autentimise.

### 3.3.3 Protsessi juhtimine

HTML rakenduse peamiseks ülesandeks on pumpade käivitamiseks optimaalse aja tuvastamine. Samas on mõjuteguriteks vee juurdevool ja jooksvad elektrienergia hinnad.

Rakenduse käivitamisel ning iga tunni alguses võetakse vastu otsus pumpade käivitamisest. Antud aspekt on tingitud sellest, et elektrienergia börsihind on diskreetne – igal tunnil oma hinnaline väärtus. Kui otsus kujuneb positiivseks, siis kontrolleri muutuja “webdata.start” saab “True” väärtuse ja pumbad käivituvad, kui süsteem on „ECO” režiimis. Väärtuste muutmiseks taustarežiimis ilma lehe uuendamiseta kasutatakse HTML abidokumenti IOstart.html ja meetodit \$.post(). (vt Joonis 3.13)

```
94 | url="IOstart.html";
```

```
95 | name=' "webdata".start';
```

```
96 | val=true;
```

```
97 | sdata=escape (name)+'='+val;
```

```
98 | $.post (url,sdata,function (result) {});
```

Joonis 3.13 Muutuja “webdata.start” väärtuse muutmine



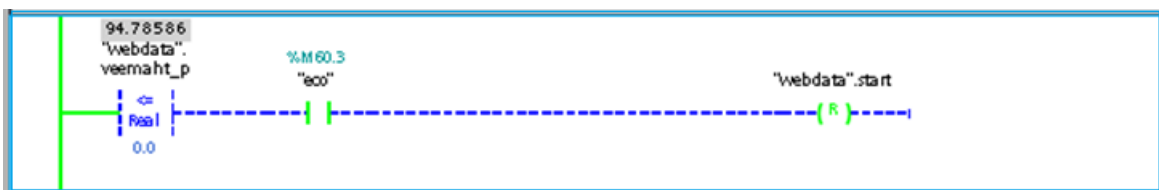
```

89     let first = prices[0];
90     var m_array = prices.slice(0, hours);
91     m_array.sort((a, b) => a - b);
92     $("#ON").text(m_array[0]);
93     if (first <= m_array[0] and rezim_eco){
94         url="IOstart.html";
95         name="webdata.start";
96         val=true;
97         sdata=escape(name)+'='+val;
98         $.post(url,sdata,function(result){});
99     };

```

Joonis 3.15 Pumpade käivitamiseks otsuse tegemine

Kui veekoguris oleva vee tasemeks on saavutatud "webdata.veemaht\_p = 0%", siis pumbad lülituvad välja ja muutuja "webdata.start" saab väärtuse "False". Antud algoritm on kontrollieris realiseeritud. (vt Joonis 3.16)



Joonis 3.16 Vee pumpamise peatamine

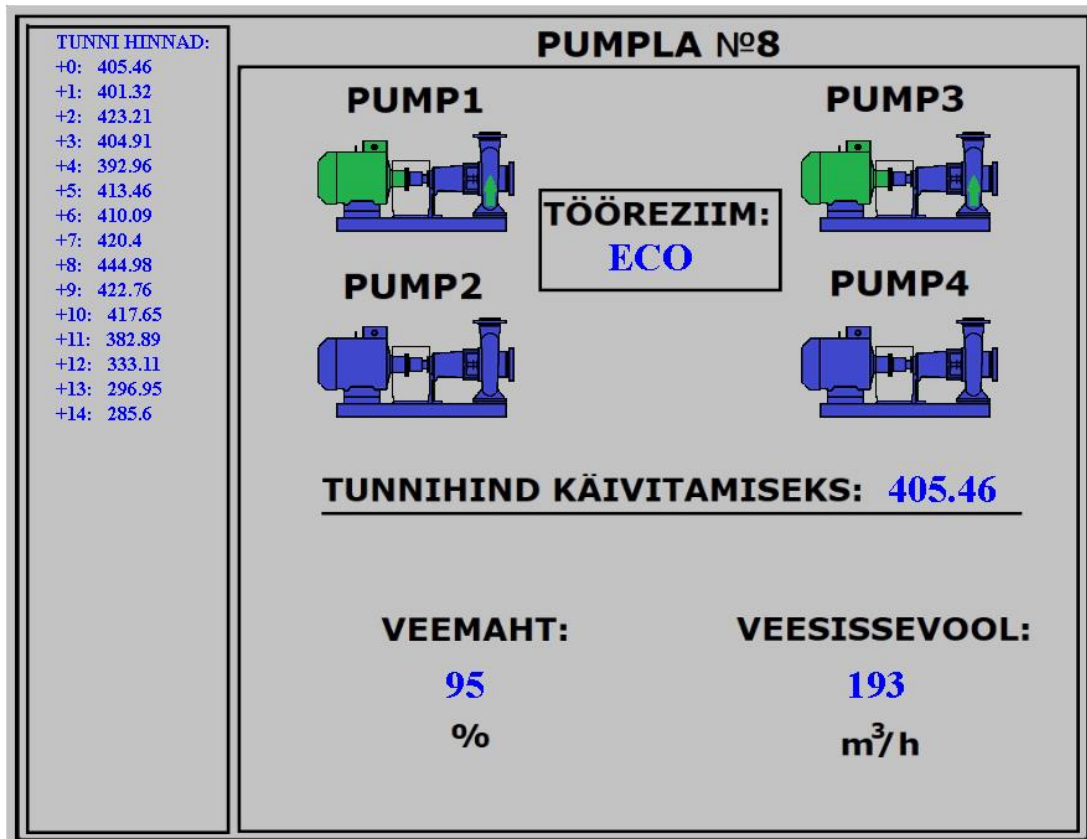
Peale eelnevalt kirjeldatud tunnise tsükli on programmis realiseeritud tööprotsessi visualisatsioon. Tsükkel kordub igal sekundil ning saades kontrollierilt muutujate väärtusi „webdata“ plokk visualiseerib neid.

### 3.3.4 Visualiseerimine

“ECO” töörežiimil visualiseerimise teostamiseks kasutakse tööstusarvutit 15 tollise ekraaniga, mis asub pumbajaama juhtimiskilbil.

Ekraanil kuvatakse järgmised andmed:

- elektribörsi tunnihinnad;
- töörežiim;
- pumpade töö staatus;
- pumpade käivitamiseks valitud tunnihind;
- tase veekoguris, protsentides;
- tunnipõhine põhjavee juurdevool. (vt Joonis 3.17)



Joonis 3.17 Protsessi visualiseerimine

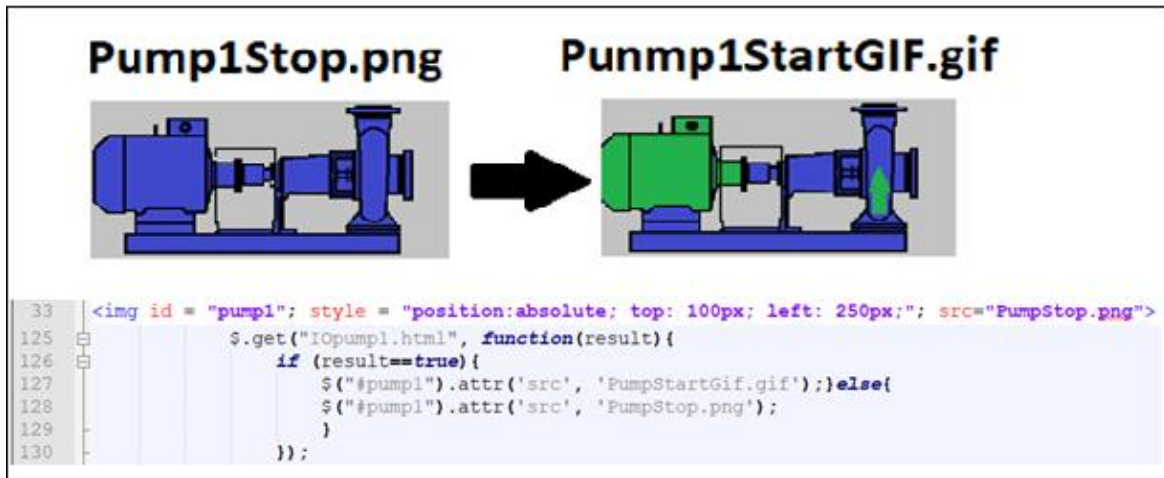
Börsihinnad tundide kaupa kuvatakse alates praegusest tunnist kuni viimase, hetkel teadaoleva hinnamääraneni. Samuti on näha hind, mille saabumisel pumbad käivituvad. Ekraanil on märgitud, millisel tunnil see toimub. Iga sekundi tagant uuendatakse veekoguri veetaseme infot ja pumpade staatuseid. Pumpade töö animatsioon toimub GIF formaadis piltide abil.

Protsessi visualisatsioon toimub JavaScript-i teegi jQuery abil. Näiteks esimese pumba käivitumise animatsioon toimub järgmiselt:

- 1) meetodi `setInterval(function() {},1000)` abil seatakse tsükli perioodilisus 1000ms. Sellisel viisil allolev kood toimib kord sekundis;
- 2) kontrolleri paigutab oma muutuja väärtuse „webdata.pump1“ HTML dokumenti `Iopump1.html`, mis omab selleks ainult ühte rida `<!-- AWP_In_Variable Name=""webdata".pump1' -->:=""webdata".pump1:"`. Märgendi nime ees on `":=""` ja järelliitega `":"` See on võti muutujate sisestamiseks kasutaja lehele. Kui leht renderdatakse, asendatakse see tegeliku kontrolleri muutujaga. HTML dokument on eelnevalt laetud kontrolleri sisemisele veebiserverile;
- 3) kui esimene pump töötab, siis muutuja „webdata.pump1“ väärtus on `"true"`. Meetodi `$()` abil pumba joonis, mis asub failis `'PumpStop.png'`, asendatakse animeeritud joonisega `'PumpStartGif.gif'`;(Joonis)

- 4) kui esimene pump ei tööta, muutuja „webdata.pump1“ väärtus on „false“, pumba joonis asendatakse sarnaselt eelmise näitega.

Sama põhimõtte järgi visualiseeritakse kõik teised pumbad. (vt Joonis 3.18)



Joonis 3.18 Pumpade visualiseerimine

Kõik vajalikud HTML dokumendid ning joonised on eelnevalt üles laetud kontrolleri sisemisele veebiserverile Siemens TIA Portal keskkonna abil.

Jooksva režiimi visualiseerimiseks tuleb failist IOrezim.html saada muutuja „webdata.rezim“ väärtus. Järgmisena tuleb väärtus konverteerida arvvaatusest operaatorile mugavalt vastuvõetavasse vormi, kujutades rida mille nimeks on režiimi nimetus. Režiimi nimetus kuvatakse ekraanile meetodi \$.post(url,sdata,function(result){}) abil jOeri raamatukogust. Funktsioon function(result){} kutsutakse esile vaid siis, kui on teostatud edukas päring \$.post() ja argumendina saab selle täitmise tulemuse, ehk muutuja „webdata.rezim“ väärtuse.

Edasi tulemus läbib töötlemise ning ekraanile kuvatakse jooksva režiimi nimetus (vt Joonis 3.19)

```
148     $.get("IOrezim.html", function(result) {
149         if (result=="0") {
150             $("#REZIIM").text("0");
151             rezim_eco = false;
152         }
153         else if (result=="1") {
154             $("#REZIIM").text("KÄSITSI");
155             rezim_eco = false;
156         }
157         else if (result=="2") {
158             $("#REZIIM").text(" AUTO ");
159             rezim_eco = false;
160         }
161         else {
162             $("#REZIIM").text(" ECO ");
163             rezim_eco = true;
164         }
165     });
```

Joonis 3.19 Jooksva režiimi visualiseerimine

## KOKKUVÕTE

Käesoleva töö eesmärgiks oli Estonia kaevanduse allmaa pumbajaama automatiseerimine elektrienergia börsihinda arvestades. See tähendab, et pumbad tuleb käivitada sõltuvalt elektribörsi tunnihinnast soodsal ajal. Kaevandusel põhjavee pumpamine on üks energiamahukamaid protsesse, kuid on olemas võimalus börsipaketi kasutades elektriarvet oluliselt vähendada.

Oma töös automatiseerimise kulude vähendamiseks, mina kasutasin kaheksanda pumbajama automaatikaseadmeid, lisasin uue režiimi "ECO" juhtimiseks ning visualiseerimiseks tööstusarvuti ja vajaliku tarkvara. Tööstusarvuti SIMATIC IPC277E paigaldatakse pumbajaama automaatikakilpi. Tööstusarvuti internetibrauser laadib HTML rakenduse kontrolleri SIMATIC S7 serverist.

Uues režiimis pumpade juhtimiseks ja protsessi visualiseerimiseks mina lõin HTML rakenduse, mis suhtleb kontrolleri ja. Ta annab käsu käivitada pumbad, kui börsi elektri tunnihind on madalaim. Samal ajal jälgib põhjavee sissevoolu veekogurisse, vältides selle ületäitumist. Rakendus võtab börsi tunnihindu Eleringi serverisse päringu saates ja põhjavee tunnise juurdevoolu arvestades, arvutab pumpade käivitamise ajamomenti. Protsessi visualiseerimiseks, mina lõin veebilehe disaini ja kõik kasutatud pildid.

Rakenduse kirjutamise käigus mina tutvusin Java Scripti veebiprogrammeerimiskeele ning jQuery teegiga. Selle kaudu on realiseeritud kogu protsessi loogika ning arvutused. Õppisin asünkroonseid meetodeid, mis võimaldavad ka mitte peatada programmi täitmist enne oma protsessi lõppu, näiteks serverist andmete vastuvõtmise jooksul. jQuery teegi meetodid saavad muuta veebilehe sisu, ilma seda uuesti laadimata.

Kontrollerit Simantic S7 kasutatakse protsessi töörežiimide realiseerimiseks. Nendeks on käsitsi, automaatne ja uus režiim "ECO", mis realiseeritakse elektribörsi tunnihinna alusel. Mina lõin kontrolleri juhtimisprogrammi pumpade käivitamiseks kõikides režiimides LAD-i programmeerimise keele kasutamisega. Pumpade tööd kontrollisin erinevates režiimides. Selle jaoks mina lisasin juhtimisprogrammile põhjavee sissevoolu simulatsiooni. Juhtimisprogramm töötas korrektselt, ilma vigadeta.

Lõputöö analüüsi ülesandeks oli kaevanduse SCADA andmebaasist tööprotsessi statistiliste andmete allalaadimine ja nende alusel uue töörežiimi algoritmi leidmine. Analüüsi arvutuse automatiseerimiseks mina kirjutasin scripti Python-i programmeerimiskeelega. Selgitasin välja, et võib pumpade käivitamiseks parimat elektribörsi tunnihinda oodata vähemalt 24 tundi, vältides veekoguri ületäitumist.

Teiseks analüüsi ülesandeks oli elektritarbimise rahaliste kulude võrdlemine, vana ja uue automatiseerimissüsteemi puhul. Selle jaoks mina võrdlesin kaheksanda pumbajaama elektritarbimise oktoobri kuu arvet uue automatiseerimismeetodi

rakendamisel saadud elektri kuarvega. Oktoobri elektrikulud uue automatiseerimismeetodi puhul olid 2 614.28 eurot, mis on 66.9% võrra vähem, kui vana automatiseerimispehimoette puhul (7 900.20 eurot).

Minu poolt loodud automatiseerimissüsteemi ja protsesside kasutamisega saavutati suur positivne majanduslik efekt. Börsipaketi kasutades on võimalik elektriarvet oluliselt vähendada tigitimisel, et tehnoloogiline protsess võimaldab oma olemuselt rakendada paindlikku juhtimismeetodit jälgides seejuures korruga eri muutujaid. See on hea võimalus eri ettevõtete jaoks. Minu töö arendamise üheks teekonnaks võib olla elektrienergia akkumuleerimistehnoloogia kasutamine. Elektrienergia akkumuleeritakse kui elektri börsihind on madalam ja kasutatakse, kui hind tõuseb.

Virumaa Kolledži õppimise käigus omandatud teadmised ja oskused aitasid minul luua efektiivse automaatse juhtimissüsteemi. Olen arvamusel, et kõik projekti algul püstitatud eesmärgid on saavutatud. Mul õnnestus pumbajaama tööd automatiseerida elektri börsihinda alusel ja majanduslikult positiivse tulemuse saavutamist tõestada.



## **SUMMARY**

The topic of the thesis is "Automation of a pumping station, depending on the exchange price of electricity". The author of the work is Aleksandr Hinn. The supervisor of the thesis is Sergei Pavlov, a lecturer at Virumaa College.

The aim of this work was the automation of the underground pumping station of the Estonia mine, considering the stock market price of electricity. This means that the pumps must be started at a favorable time depending on the hourly price of the electricity exchange. Pumping groundwater at a mine is one of the most energy-intensive processes, but there is an opportunity to significantly reduce the electricity bill using the exchange package.

To reduce the cost of automation for this work, the author used automation equipment of the eighth pump, where he added an industrial computer and the necessary software for control and visualization of the new "ECO" regime. The industrial computer SIMATIC IPC277E is installed in the automation panel of the pumping station. The Internet browser of the industrial computer loads the HTML application from the SIMATIC S7 server of the controller.

Aiming to control the pumps and visualize the process in the new mode, an HTML application that communicates with the controller was created. It gives an order to start the pumps when the hourly price of electricity on the exchange is the lowest. At the same time, it monitors the inflow of groundwater into the water body, preventing it from overflowing. The application takes stock exchange hourly prices by sending a request to the Elering's server and calculates the moment of starting the pumps taking into the account the hourly groundwater inflow. To visualize the process, the author created the website design and all the images used.

While writing the application, the author got acquainted with the Java Script web programming language and the jQuery library through which the logic and calculations of the entire process have been realized. The author learned about asynchronous methods, which also allow not to stop the execution of the program before the end of its process, for example, while receiving data from the server. Methods in the jQuery library can change the content of a web page without reloading it.

The Simantic S7 controller is used to realize the operating modes of the process. These include manual, automatic and the new "ECO" mode, which realize based on the hourly price of the electricity exchange. The controller control program was created to run the pumps in all modes using the LAD programming language. During the research, the operation of the pumps in different modes was checked. For this, a groundwater inflow

simulation was added to the control program. The control program worked correctly, without errors.

The purpose of the thesis analysis was to download the statistical data of the work process from the SCADA database of the mine and to find a new work mode algorithm based on them. To automate the calculation of the analysis, the author wrote a script in the Python programming language. The possibility was found out to wait at least 24 hours for the best hourly price on the electricity exchange to start the pumps, avoiding overflowing the water tank.

Another task of the analysis was to compare the financial costs of electricity consumption in the case of the old and new automation system. For this purpose, the author compared the electricity consumption bill of the eighth pumping station for the month of October with the monthly electricity bill resulting from the implementation of the new automation method. October electricity costs of the new automation method were 2,614.28 euros, which is 66.9% less than when using the old automation principle (7,900.20 euros).

A considerable positive economic effect was achieved by using the automation system and processes created by the author. By using the exchange package, it is possible to significantly reduce the electricity bill, given that the technological process, by its very nature, enables the application of a flexible management method while monitoring different variables at the same time. This is a good option for different companies. One of the ways to develop the completed research could be the use of electrical energy storage technology. Electric energy is accumulated when the electricity exchange price is lower and used when the price rises.

The knowledge and skills acquired during the studies at Virumaa College helped the author create an effective automatic control system. The author is of the opinion that all the goals set at the beginning of the project have been achieved. The author managed to automate the work of the pumping station based on the stock exchange price of electricity and prove the achievement of an economically positive result.

## KASUTATUD KIRJANDUSE LOETELU

1. Siemens industry "SIMATIC STEP 7 and WinCC Engineering V15.1" System Manual, 16.10.2018.  
[https://support.industry.siemens.com/cs/attachments/109755202/STEP\\_7\\_WinCC\\_V15\\_1\\_enUS\\_en-US.pdf?download=true](https://support.industry.siemens.com/cs/attachments/109755202/STEP_7_WinCC_V15_1_enUS_en-US.pdf?download=true)
2. Lumel Automation devices and systems "Ultrasonic level transducer ULT20", 2022.  
<https://www.lumel.com.pl/en/catalogue/product/ultrasonic-level-transducer-ult20>
3. Siemens industry mall, "6ES7231-4HF32-0XB0", 2022.  
<https://mall.industry.siemens.com/mall/en/ww/catalog/product/6es7231-4hf32-0xb0> (11.10.2022)
4. Siemens industry mall, "6AV7882-0DB50-4CA0", 2022.  
<https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6AV7882-0DB50-4CA0> (11.10.2022)
5. Siemens industry mall, "6ES7215-1AG40-0XB0", 2022.  
<https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7214-1AG40-0XB0> (11.10.2022)

## Lisa 1 HTML-rakendus index.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Language" content="en" >
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
<meta name="Aleksandr Hinn 193239EDTR">
<title>PUMPLA</title>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body bgcolor = "white";>
<div style = "position:relative;margin-left: auto;
margin-right: auto; height: 680px;
width: 880px; background-image:
url('Visio.jpg');
background-repeat: no-repeat;
background-size: 100% 100%;
">
<div style = "font-weight: bold; font-size: 16px; color: BLUE; position:absolute; top:
20px; left: 40px;";>
<b>TUNNI HINNAD:<br>
<label id = "PRICES";>
</label>
</div>
<label id = "REZIIM";
style = "position: absolute; font-weight: bold; font-size: 32px; color: BLUE;
position:relative; top: 185px; left: 485px;";
>ECO</label>
<label id = "VEEMAHT";
style = "position: absolute; font-weight: bold; font-size: 32px; color: BLUE;
position:relative; top: 540px; left: 290px;";
>100%</label>
<label id = "SISSEVOOL";
style = "position: absolute; font-weight: bold; font-size: 32px; color: BLUE;
position:relative; top: 545px; left: 510px;";
>00.00</label>
```

```


</img>

</img>

</img>

</img>
<label id = "ON";
    style = "position: absolute; font-weight: bold; font-size: 32px; color: BLUE;
position:relative; top: 372px; left: 490px;";
    >+0</label>
</div>
</body>
<script type="text/javascript">
    $(document).ready(function(){
        $.ajaxSetup({ cache: false });
        let prices = 0.00;
        let sissevool_tund = 193.00;
        let veemaht_alg = 0.00;
        let veemaht_m = 0.00;
        let veemaht_p = 0.00;
        let rezim_eco = false;
        setInterval(function() {
            let today = new Date;

            if (prices == 0 || (today.getUTCMinutes() == 0 &&
today.getUTCSeconds() < 2)){
                let tomorrow = new Date;
                tomorrow.setDate(tomorrow.getDate() + 1);
                let today_iso = today.getUTCFullYear() + '-' +
(today.getUTCMonth()<10?'0:') + (today.getUTCMonth() + 1) + '-' +
(today.getUTCDate()<9?'0:') +
                    today.getUTCDate() + "T" +
(today.getUTCHours()<10?'0:') + today.getUTCHours());

```



```

m_array.sort((a, b) => a - b);
$("#ON").text(m_array[0]);
if (first <= m_array[0] and rezim_eco){
    url="IOstart.html";
    name=""webdata".start";
    val=true;
    sdata=escape(name)+'='+val;
    $.post(url,sdata,function(result){});
};
});
},
error: function (jqXHR, exception) {
if (jqXHR.status === 0) {
    alert('Not connect. Verify Network.');
```

```

} else if (jqXHR.status == 404) {
    alert('Requested page not found (404).');
```

```

} else if (jqXHR.status == 500) {
    alert('Internal Server Error (500).');
```

```

} else if (exception === 'parsererror') {
    alert('Requested JSON parse failed.');
```

```

} else if (exception === 'timeout') {
    alert('Time out error.');
```

```

} else if (exception === 'abort') {
    alert('Ajax request aborted.');
```

```

} else {
    alert('Uncaught Error. ' +
jqXHR.responseText);
};
};
$.get("IOveemaht_p.html", function(result){
$('#VEEMAHT').text(result);
});
}

```

```

$.get("IOpump1.html", function(result){

    if (result==true){
        $("#pump1").attr('src', 'PumpStartGif.gif');}else{
        $("#pump1").attr('src', 'PumpStop.png');
        }
    });

$.get("IOpump2.html", function(result){

    if (result==true){
        $("#pump2").attr('src', 'PumpStartGif.gif');}else{
        $("#pump2").attr('src', 'PumpStop.png');
        }
    });

$.get("IOpump3.html", function(result){

    if (result==true){
        $("#pump3").attr('src', 'PumpStartGif.gif');}else{
        $("#pump3").attr('src', 'PumpStop.png');
        }
    });

$.get("IOpump4.html", function(result){

    if (result==true){
        $("#pump4").attr('src', 'PumpStartGif.gif');}else{
        $("#pump4").attr('src', 'PumpStop.png');
        }
    });

$.get("IOrezim.html", function(result){
    if (result=="0"){
        $("#REZIIM").text("0");
        rezim_eco = false;
        }
    else if (result=="1"){
        $("#REZIIM").text("KÄSITSI");
        rezim_eco = false;
    }
});

```



```
        }
        else if (result=="2"){
            $("#REZIIM").text(" AUTO ");
            rezim_eco = false;
        }
        else {
            $("#REZIIM").text(" ECO ");
            rezim_eco = true;
        }
    });
    $.get("IOveemaht_p.html", function(result){
        $('#VEEMAHT').text(result);
    });
    },1000);
});
</script>
</html>
```