



TALLINNA TEHNICAÜLIKOOL
INSENERITEADUSKOND
Tartu kolledž

**VIRU-NIGULA SAEVESKI AS UUE
LIPIKOGUMISLIINI AUTOMATISEERIMINE JA
VISUALISEERIMINE**

**VIRU-NIGULA SAEVESKI AS' NEW STICK COLLECTING
LINE AUTOMATION AND VISUALISATION**

RAKENDUSKÕRGHARIDUSTÖÖ

Üliõpilane: Johannes Laur

Üliõpilaskood: 207703EDTR

Juhendaja: Ago Rootsi, lektor
Priit Poopuu, ettevõtte juht

(Tiitellehe pöördel)

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.

11.2024

Autor: Johannes Laur

/ allkirjastatud digitaalselt /

Töö vastab bakalaureusetöö esitatud nõuetele

11.2024

Juhendaja: Ago Rootsi

/ allkirjastatud digitaalselt /

Kaasjuhendaja: Priit Poopuu

/ allkirjastatud digitaalselt /

Kaitsmisele lubatud

11.2024

Kaitsmiskomisjoni esimees: Aime Ruus

/ allkirjastatud digitaalselt /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina Johannes Laur

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Viru-Nigula Saeveski AS uue lipikogumisliini automatiseerimine ja visualiseerimine, mille juhendaja on Ago Rootsi ning kaasjuhendaja Priit Poopuu,

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.

11.2024

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Tartu Kolledž

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Johannes Laur, 207703EDTR

Õppekava, peeriala: Telemaatika ja arukad süsteemid EDTR17/18, Küberfüüsikalised süsteemid

Juhendaja(d): Ago Rootsi, lektor, +372 56629821

Kaasjuhendaja(d): Priit Poopuu, ettevõtte juht, tööstusautomaatik, +372 5567 8357

Lõputöö teema:

Viru-Nigula Saeveski AS uue lipikogumisliini automatiseerimine ja visualiseerimine

Viru-Nigula Saeveski AS' new stick collecting line automation and visualisation

Lõputöö põhieesmärgid:

1. Automatiseerida ning visualiseerida uus lipikogumisliin
2. Korralik pikaajaline veahaldus, süsteemi jälgimine
3. Tuua välja erinevused varasema lipikogumise lahendusega

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Automatiseerida ning visualiseerida uus lipikogumisliin.	30.04.23
2.	Veahaldus ning süsteemi jälgimine, andmete kogumine.	30.05.23
3.	Järeldused, analüüs, võrdlus varasema süsteemiga.	30.06.23

Töö keel: eesti keel **Lõputöö esitamise tähtaeg:** 11.01.2024

Üliõpilane: Johannes Laur 11.2024 a

Juhendaja: Ago Rootsi 11.2024 a

Kaasjuhendaja: Priit Poopuu 11.2024 a

Programmijuht: Aime Ruus 11.2024 a

Kinnise kaitsmise ja/või lõputöö avalikustamise piirangu tingimused formuleeritakse pöördel

SISUKORD

EESSÕNA	7
SISSEJUHATUS	8
1 KIRJANDUSE ÜLEVAADE	10
2 RIISTVARA VALIK	11
2.1 Kontrolleri valik	11
2.2 Liidese mooduli valik	12
2.3 Sagedusmuundurite valik	12
2.4 Seadmete suhtlusprotokollid.....	13
3 TARKVARA VALIK	15
3.1 Kontrolleri programmeerimine ning liini visualiseerimine	15
3.2 Sagedusmuundurite seadistamine.....	15
4 VÕRGU SEADISTAMINE	16
4.1 Sisendite ja väljundite nimekiri	16
4.2 TIA Portal seadmed ja ühendused	17
4.3 Sagedusmuundurid	17
4.4 Sisendide ja väljundite kontroll	19
5 PROGRAMMEERIMINE.....	20
5.1 Kontrolleri programmeerimine.....	21
6 VISUALISEERIMINE	24
6.1 Eeltöö visualiseerimise jaoks.....	24
6.2 Lipiliini külj- ja pealtvaade ning andurid	25
6.3 Visualisatsiooni dünamiseerimise näited.....	26
6.4 Interaktsioon lipiliiniga läbi visualisatsiooni.....	28
7 KONTROLLERI KOODI TESTIMA	31
7.1 Riistvaralised vead.....	31
7.2 Koodi vead.....	31
8 SÜSTEEMI TURVALISUS.....	33
8.1 Brauseripõhine visualisatsioon ja programm	33
8.2 Kohalik võrk.....	33
9 VÕRDLUS VARASEMA LAHENDUSEGA	34
9.1 Tehnoloogiline tööpuudus	34

9.2 Mõjud operaatorite tervisele	35
KOKKUVÕTE	36
SUMMARY	37
KASUTATUD KIRJANDUSE LOETELU	38

EESSÕNA

Eesti üks suuremaid puiduettevõtteid OÜ Nordwood omab kolme suurt saeveskit, millest üks asub Viljandimaal Viiratsis, teine Aegviidus ning kolmas Viru-Nigulas. Saeveskrite automaatika vajab pidevat veahaldust ja uuendamist, millega tegeleb ettevõtte PRT Automation OÜ. Kuna autor on antud tööstusautomaatika ettevõttes praktikal olnud juba kaks aastat, otsustas ka lõputöö valdkonna ning tööülesannetega siduda.

Autor soovib tänada oma juhendajaid Ago Rootsit ning Priit Poopuud. Samuti OÜ Nordwood kollektiivi.

Märksõnad: automaatika, visualiseerimine, puidutööstus, kontrollid, siemens, rakenduskõrgharidustöö, lõputöö.

SISSEJUHATUS

Taaskasutus on nüüdisaja tava: üha enam tööstusettevõtteid rakendab tootmises erinevaid protsesse, mis võimaldavad mingisuguses tootmisprotsessis kasutatavaid osi taastada ning uuesti kasutada. Samasugust praktikat kasutavad ka saeveskid. Üheks neist on OÜ Nordwood alla kuuluv AS Viru-Nigula Saeveski saeveski.

Viru-Nigula saeveski tootmistsükkel on sarnane paljudele teistele saeveskitele. Märjad palgid saetakse laudadeks, mis pakitakse lauapakiks. Lauapaki kihtide vahele asetatakse kiirema kuivamise soodustamiseks lipid ehk peened puupulgad. Lippidega täidetud niiske lauapakk läheb kuivatisse kuivama. Peale kuivamist saadetakse juba kuivanud, kuid siiski lippidega täidetud lauapakk kuivatist sorteeriliinile. Seal võetakse see kihthaaval lahti, lipid eraldatakse ning lauad sorteeritakse mõõtude järgi lauataaskutesse. Taskutest edasi lähevad kuivad lauad pakkimisliinile, kus seotakse ja pakitakse lauad viimast korda enne müüki.

Lauapakist lippide eraldamise etapi lihtsustamiseks ehitati Viru-Nigula saeveskisse uus lipikogumisliin. Enne ehitamist kooskõlastati liini kavand ning täpne asukoht saeveski juhtatuse, mehaanikute, elektrikute ning automaatikutega. Peale liini mehaanika ja elektroonika konstrueerimist kutsuti kohale meid, automaatikuid. Uuel lipiliinil puudusid võrguseadistused, automaatikakood ning visualisatsioon.

Lõputöö eesmärk oli automatiseerida ning visualiseerida kuivanud lauapakist eraldatud uus lippide kogumise liin ning võrrelda seda saeveskis eelnevalt kasutatud lippide kogumise meetodiga.

Enne uue lipiliini tulekut vedas etteandeliini all olev lintkonveier lipid majast välja teisele poole, kus sorteeris neid üks saeveski operaatoritest käsitsi. Lipid liikusid mööda lintkonveierit, läbi ava, majast väljas asuvasse kasti. Operaator sorteeris lipid seejärel kasutuskõlblikuse järgi. Mõranenud, hõrenenud või täiesti poolikud lipid viskas teise kõrvalseisvasse kasti ning piisavalt korras lipid ladus käsitsi lipikorvi.



Joonis 0.1 Lipiliin: Lippide sorteerimine enne automatiseeritud liini



Joonis 0.2 Sortliin: Lippidega täidetud kuivatist tulnud lauapakk

1 KIRJANDUSE ÜLEVAADE

Olemasolev kirjandus lippide ning nende kasutamise kohta on puudulik. Sama võib öelda ka uue lipiliini kohta, mis oli juba aimatav, kuna tegu on ainulaadse ja kohandatud lipikogumisliiniga, mille sarnast olen kohanud vaid, samuti Nordwood OÜ alla kuuluvas, AS Aegviidu Puit saeveskis. Et uurimistöö valdkonnast oleks endale suurem ülevaade, uurisin ka laiemat pilti. Kas ja millist kirjandust leiab Eesti või mõne lähedal asuva riigi puidutööstuste tootmisprotsesside, automaatika ja selle arengu kohta.

Rootsi Linné Ülikooli teadlaste sõnul on Rootsi puidutööstus tootlikuse poolest maailma juhtivate seas. Veel selgus, et nii Rootsi kui ka enamuse puidutööstuseid välisriikidest kannatavad vähese automatiseerituse ning tööjõu, ka koolitatud tööjõu, puuduse all. Samm järgmisesse arengufaasi Industry 4.0 on tegemata, viibitud ning selle tegemise täpsed mõjud ei ole teada. Uue digitaliseerimise ning automatiseerimise tagajärjel võib Rootsis töö kaotada ennustuste kohaselt ligilähedale 80% puidutööstuses töötavatest inimestest.[1]

Lappeenranta-Lahti Ülikooli üliõpilase lõputöö oli uurida Metsä Fibre Lappeenranta saeveski näidetel saeveski automatiseerituse taset ning jagada tulemuste põhjal soovitusi. Tema uuringutest selgus, et Lappeenranta saeveski süsteemi juhtisid pea 30 aastat vanad kontrollerid, mille tootmine ning tootja poolt uuendamine on lõpetatud. Kogu süsteem jooksis profibusi vahendusel. Lappeenranta saeveskil puudusid ka liinide juhtimiseks ka tänapäevased HMId: operaatoritel puudus liinist korralik ülevaade, mis muutis häirete märkamise ning kõrvaldamise tunduvalt raskemaks.[2]

Eesti Maaülikooli üliõpilase lõputöö oli leida ning fikseerida OÜ Vara Saeveski saeveski uue kombiliini seisakuid, nende põhjuseid ning koostada ettepanekuid tootlikuse tõstmiseks. Ta leidis, et suurem osa seisakuid oli seotud mehaaniliste vigade või elektrikatkestustega, mida on võimalik hooldustöödega ennetada. Teine põhjus seisakuteks oli ebapropotsionaalne materjal: väiksema ristlõikega, muljunud või katkine materjal tekitab liinil ummistusi. Ummistuste ennetamiseks oleks tema sõnul vaja praeguse liini lahenduse juures lisatöölist.[3]

2 RIISTVARA VALIK

Et üldse midagi ehitada, peab olema mingisugune plaan või kavand – selleks kasutame elektriskeeme. Elektriskeem ning joonised Viru-Nigula saeveski lipiliinist oli Smitech OÜ ülesanne. Tegemist on samamoodi Eesti tööstusautomaatikaettevõttega. Lipiliini ehitasid kokku enne võrku ühendamist ning automatiseerimist mehaanikud ning ühendused elektrikilbi, mootorite, sagedusmuundurite ja kontrolleri vahel tegid elektrikud. Kõik konstrueeriti elektriskeemi ning jooniste järgi.

Kasutame oma automaatikas peamiselt Siemensi kontrollereid ning Mitsubishi Electric sagedusmuundureid. Põhjuseid selleks on mitmeid. Üheks neist on kindlasti asjaolu, et paljud ettevõtted, kellele erinevaid lahendusi pakume, juba kasutavad erinevatel liinijuppidel Siemensi riist- ja tarkvara. Vana Siemensi riist- ja tarkvara uuema vastu vahetamine teeb meie jaoks automatiseerimist hõlpsamaks. See annab võimaluse näiteks konverteerida vana lähtekoodi uuemasse keskkonda, läbi mille saame eelneva töötamist paremini ning põhjalikumalt analüüsida. Samuti on Siemens ning Mitsubishi Electric riistvara kohta internetis väga palju avatud materjale, mis muudavad tööd kiiremaks.

2.1 Kontrolleri valik

Kuna tulevikus peab controller olema suuteline juhtima lisaks lipiliinile ka muid liine, pidin valima võimalikult võimsa variandi, mis oleks turvaline ning millel oleks piisavalt mälu, töövõimet ja kiirust.

Controller, mida uue liinijupi automatiseerimiseks kasutasin oli Siemens CPU S7-1517TF-3 PN/DP. Antud mudel on Siemensi S7-1500 seeria üks võimsamaid. Controlleril on integreeritud 11 megabaiti mälu, võimalus sisestada mälukaardiga kuni 32 gigabaiti mälu ning toetab maksimaalselt 16 384 sisend-väljund moodulit.[4]

Tähis „TF-3“ tähendab seda, et tegu on veakindla kontrolloriga, mis on võimeline vea korral süsteemi säästma. Muidugi sõltub antud tulest sellest, kuidas on turvasüsteem kontrolleri puhul veaolukorra jaoks programmeeritud. Samuti toetab veakindel kontroller turvamoodulite kasutamist. Kontrolleri nime järel märgis „PN/DP“ tähendab, et kontroller saab suhelda sagedusmuundurite ning arvutiga läbi profineti ja profibus-DP.[4]

2.2 Liidese mooduli valik

Kuna kontroller oli lipiliini elektrikilbist ning liinist endast piisavalt kaugel, paigaldasime liini elektrikappi liidese mooduli ehk hajus-IO, mille üks ja ainuke funktsioon oli olla kontrolleri pikendus ning säästa aega ja ressursi kaabelduse arvelt. Selle asemel, et sadu sisendi ja väljundi juhtmeid kaugel asuvasse kontrollerrisse vedada, oli elektrikutel kordades kergem neid ühendada liini kõrval olevasse hajus-IOsse, mis suhtles eemal oleva kontrolloriga läbi ainsa profineti kaabli.[5]

Kuna kasutame juba Siemensi kontrollert, oli ka hajus-IO valikul Siemensi enda hajus-IO kasutamine põhjendatud. Valik langes Siemens ET200SP 155-6PN ST kasuks, kuna mudel toetab veakindlust ning turvamoduleid, täpselt nagu eelnevalt valitud kontroller. Antud hajus-IO võimaldab kasutada maksimaalselt 32 sisend-väljund moodulit ning suurim vahemaa kontrolloriga sõltub profinet kaabli tüübist.[4], [5]

2.3 Sagedusmuundurite valik

Kuna lipiliinil oli palju positsioone, mille liigutamiseks ei piisanud ainult mootorist ja silindrist, kasutasime sagedusmuundureid. Nende rakendamine suurendab süsteemi eluiga ning efektiivsust, vähendab saeveski energiatarvet ja komponentide kulumist.[6]

Sagedusmuundurid võimaldasid läbi mootori toiteallika pinge ja sageduse muutmise kontrollida erinevaid lipiliini positsioonide kiirustega seotud parameetreid. Tähtsamad neist on näiteks liikumise kiirus, liikumise alustamise ja lõpetamise kiirus. Kuna oleme ka varasemalt Mitsubishi Electric sagedusmuunduritega kokku puutunud, valisime sama suuna ka see kord. Valituks osutus Mitsubishi Electric FR-E840-E seeria. [7]

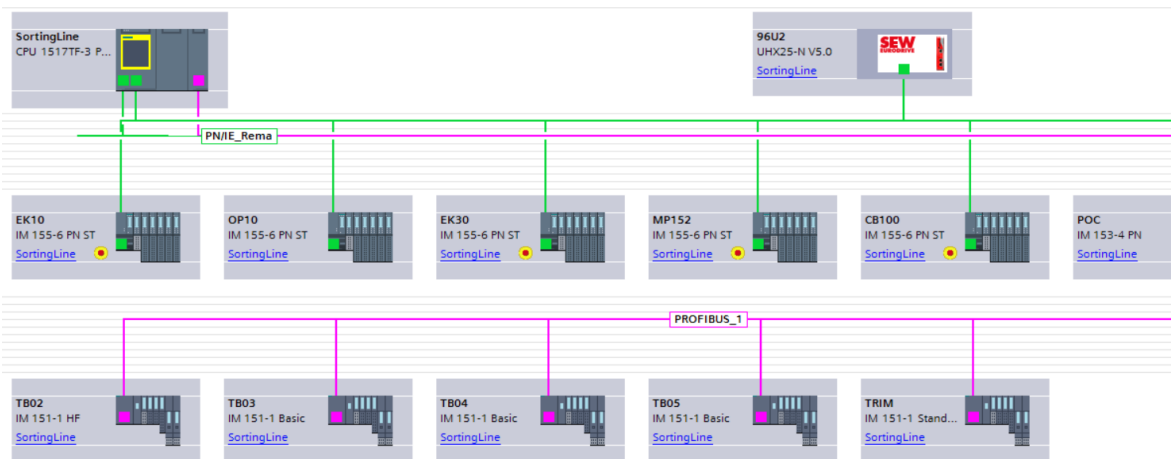
FR-E840-E seeria väljundvõimsus on 0,75 – 600 kW ning väljundsagedus 0,2 – 400 Hz. Märgis „-E“ tähistab seda, et muundur on võimeline suhtlema läbi etherneti. Kuigi seeria oli lipiliini sagedusmuunduritel sama, oli siiski iga sagedusmuundur teisest erinev, kuna igale mootorile valiti muundur vastavalt vajadusele. Liiga võimsa või nõrga sagedusmuunduriga mootori kontrollimine võib seda lõhkuda või vajamatult suurendada energia tarvet.[7]

2.4 Seadmete suhtlusprotokollid

Ka tööstustes käib seadmete infovahetus ning andmete edastamine üle etherneti. Et andmeside oleks nii veatu kui võimalik, peavad ka kaablid tööstuslikkudele oludele vastu pidama. Selleks mõeldi välja profinet ethernet (PN) ehk tugevdatud kaabliga ethernet.[8]

Suurem osa kõikidest Viru-Nigula saeveski sagedusmuunduritest, hajus-IODEst ning paneelidest suhtlevad kontrollite ning välismaailmaga üle profinet etherneti. Vanemad seadmed, millel profineti kasutamise võimalus puudub, on ühendatud üldiselt läbi profibus. profineti keskmine andmeedastuskiirus tööstuses on tavaliselt 100 megabitti sekundis ning olenevalt kaablist on maksimaalne vahe kahe seadme vahel alates 50 meetrist kuni 26 kilomeetrini. Kaabli tunneb tööstuses ära tema paksuse ning silmapaistva roheline värv järgi.[8], [9]

Enne etherneti ning profineti tulekut kasutati tööstustes laialdaselt profibus protokoll, mis põhines RS-485 võrgutüübil ning oli tööstuses ära tuntav lillade paksemate kaablite järgi. profibus on olenevalt oludele profinetist pea 10 kuni 100 korda aeglasem, mille tõttu kasutatakse seda seal, kus on uuendamata automaatika või vanemad seadmed, millel puudub profineti kasutamise võimalus.[8], [9]



Joonis 2.1 TIA Portal: osa Viru-Nigula saeveski võrgust

Pildil on näha osa Viru-Nigula saeveski võrgust. Rohelised jooned tähistavad profinet ühendust, lillad jooned tähistavad profibus ühendust, lühend CPU (*central processing unit*) tähistab põhikontrollerit, lühend IM (*interface module*) tähistab hajus-IO ning tähis U tähistab sagedusmuundurit.

3 TARKVARA VALIK

3.1 Kontrolleri programmeerimine ning liini visualiseerimine

Kuna kasutame süsteemide automatiseerimiseks Siemensi kontrollereid, peame nende konfigureerimiseks ning programmeerimiseks kasutama vastavat tarkvara. Kõige turvalisem on seda teha Siemensi enda poolt pakutavate programmidega. Võimalik on soetada ning alla laadida programmide pakett nimega Siemens STEP 7 and WinCC Engineering V17 mille südamikuks on Siemens TIA Portal.[10]

TIA Portalis konstrueerid võrgu, määrad ära selle komponendid, otsid nendega ühenduse, programmeerid, visualiseerid paneele ja brauseripõhist SCADAt. Samuti pakettis leiduvad WinCC Runtime Advanced ja WinCC Unified Certificate Manager teevad endast oleneva, et server läheks, püsiks töökorras ning litsentsid eksisteeriks. Paketti kuulub veel mitmeid teisi programme, mis aitavad arvuti-, programmi- või SCADA poolse seadistusega.[10]

3.2 Sagedusmuundurite seadistamine

Et lipiliini sagedusmuundurid võrgus kontrolleri ning teiste seadmete poolt leitavad oleks, ei piisa vaid nende elektri- ning profineti võrku ühendamisest – need tuleb ka seesiselt seadistada. Muundurile on küll integreeritud 4-kohaline LED-ekraan, kuid kolme nupuga kümnete, kui mitte sadade, parameetrite ära seadistamine võtaks tohutult aega. Kiiremaks seadistamiseks on Mitsubishi loonud tarkvara FR Configurator2.[7], [11]

Programm võimaldab muuta sagedusmuundurite parameetreid läbi arvuti. Selle asemel, et väiksel ekraanil parameetreid üksikshaaval muuta, pääseb neile ligi otse ning grupiti suurel ekraanil. FR Configurator2 toob välja lisaks parameetritele ka sagedusmuunduri nime, tüübi, MAC- ja IP-aadressi.[11]

4 VÕRGU SEADISTAMINE

Kui mehaanikud olid lipiliini riistvara monteerinud ning elektrikud olid ära ühendanud nii andurid, mootorid kui ka valmis saanud elektrikilbiga, kutsuti kohale automaatikuid, ehk meid. Minu töö algas automatiseerimise esimesest etapist, ehk võrgu seadistamisest.

Esimese asjana kontrollisin silmadega üle elektrikilbi, kuhu oli paigaldatud ET200SP 155-6PN ST hajus-IO, ning 10 Mitsubishi FR-E800 seeria sagedusmuundurit. Siemensi S7-1517TF-3 kontrolleri oli paigaldatud umbes 40 meetri kaugusele lipiliini elektrikapist, täpsemalt sortliini elektrikappi, kuna peale lipiliini lisatakse sinna tulevikus ka uus sortliini programm.



Joonis 4.1 Elektrikilp EK10: lipiliini sagedusmuundurid

4.1 Sisendite ja väljundite nimekiri

Iga projekti jaoks ehitame enne selle algust elektrikute koostatud elektriskeemide järgi Excelis nimekirja, kus on välja toodud liini positsioonid, kõik liiniga seotud komponendid, nende asukohad ning seotud elektrikilpide sisendite ja väljundite andmed. Nimekirjast saab vajadusel kiiresti välja lugeda täpset infot mingi positsiooni, komponendi, sisendi või väljundi kohta.

Lipiliini sisendite ja väljundite nimekirja koostasime enda jaoks umbes kuu enne seda, kui üldse midagi koha peal ehitama hakati.

4.2 TIA Portal seadmed ja ühendused

Kui esimene ülevaade oli elektrikilbist EK10 tehtud, alustasin Siemensi TIA Portalis lipiliini projektis võrgu komponentide ühendamist. Otsisin programmis üles kontrolleri ning hajus IO-t kirjeldavad GSD komponendid.

GSD fail ehk *profinet General Station Description* fail kirjeldab programmile, antud juhul Siemensi TIA Portalile, millise seadmega on täpsemalt tegu. TIA Portalis on olemas kõik Siemensi enda komponentide GSD failid, seega kontrolleri ning hajus-IO oma kuskilt internetist otsida ei olnud vajalik. Sagedusmuundurid olid Mitsubishi omad, seega nende GSD failid tuli otsida tootja enda leheküljelt.

Siemens TIA Portalis omistasin kontrolleri, hajus-IOle ning sagedusmuunduritele IT osakonna poolt määratud IPv4 aadressid ning profineti numbrid ja seadmenimed. Samuti määrasin veel ära kõik võrgu komponendid omavahel suhtlema üle profineti. Kuigi omistatud IPv4 aadressid ning profineti numbrid ja seadmenimed kirjutab Siemens TIA Portal oma seadmetele ise otse sisse, peab sagedusmuundurid veel eraldi seadistama, et TIA Portal need ära registreeriks.

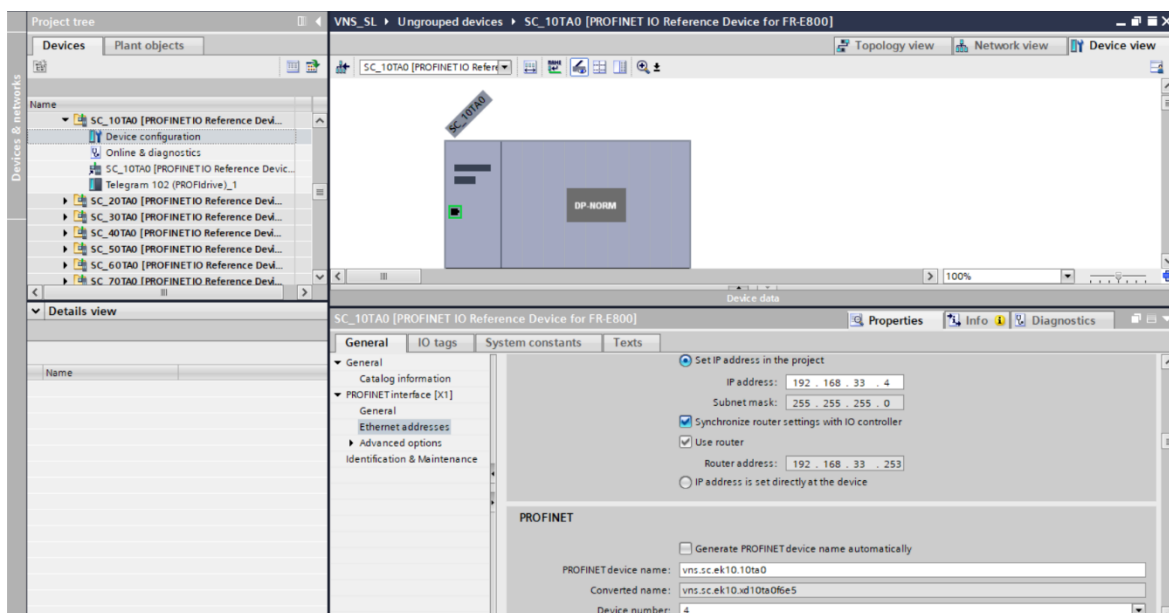
Peale algse võrgu loomist kirjeldasin TIA Portalis kontrolleri koodis ära veel kõik sisendid, väljundid, nende nimed ning aadressid. Sisendite ja väljundite kontrolli puhul saab programmist otse jälgida reaajas nende olekut.

4.3 Sagedusmuundurid

Kui algne võrk oli programmis loodud, pidin hakkama sagedusmuundureid seadistama. Sagedusmuundurite seadete muutmiseks on Mitsubishil olemas programm FR Configurator2, mis võimaldab muuta muunduri parameetreid kiirelt läbi arvuti. Enne seda tuli seadistada mõned parameetrid igal sagedusmuunduril käsitsi. Sagedusmuundurite käsitsi seadistamine käis läbi 4-kohalise integreeritud LED-ekraani.

Parameetri nr 1425 seadistamine määras ära muunduri isikliku seadmenumbri, läbi mille oskas FR Configurator2 neid eristada. Iga sagedusmuunduri number sõltus sellele antavast IPv4 aadressist. See ei olnud standard, vaid praktika, mida oma ettevõttes kasutame, et tulevikus oleks seadmed lihtsamalt leitavad ning eristatavad.

Parameetrid nr 1427 kuni 1430 olid mõeldud kommunikatsiooni jaoks. Igale neist sain valida vastavalt vajadusele protokollid, läbi mille oskas muundur teiste seadmetega suhelda. Kuna tööstusvõrk oli üles ehitatud profineti peal, kasutasin parameetri nr 1427 puhul muidugi seda. Ülejäänud kolm kommunikatsiooniparameetrit nr 1428 – 1430 jätsin muutmata, kuna nende tehaseseadistus oli juba sobiv. Parameetrid nr 1428 – 1429 vastutasid FR Configurator2 suhtlemise eest ning parameeter nr 1430 jäi reservi. Kui muunduri isikliku seadmenumbri ning kommunikatsiooniparameetrite seadistamine sai tehtud, oli muundur Siemensi TIA Portalile leitav. Läbi selle sain sagedusmuunduritele sisse kirjutada unikaalse IPv4 aadressi, profineti seadme nime ning numbrit.



Joonis 4.2 TIA Portal: sagedusmuunduri seadistamine

Iga muundur oli peale TIA Portalis seadistamist ka programmis FR Configurator2 leitav, läbi nii profineti kui ka FR-E800 seerial kasutatava USB B mini kaabli. Programmis oli võimalik kiirelt muuta ja kirjutada vajadusel kõiki teisi muundurisiseseid parameetreid kergelt läbi arvuti.

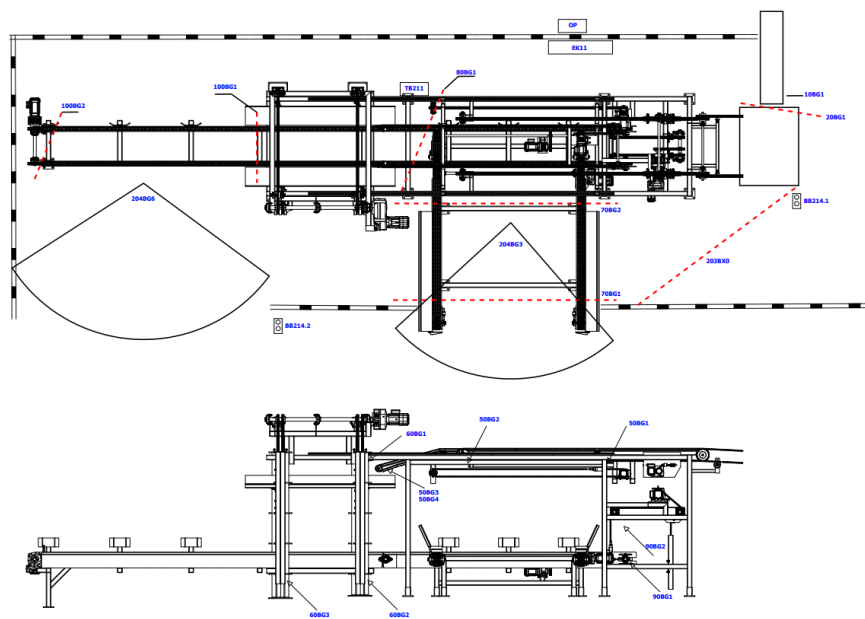
4.4 Sisendide ja väljundite kontroll

Et programmeerimise protsess liini automatiseerimise teises pooles sujuks, tuli läbida lipiliini elektrilbi sisendite ja väljundite kontroll. Üle tuli vaadata kõik ühendused erinevate lipiliini andurite, turvalülite, mootorikaitsmete, nupupaneeli nuppude ja hajus-IO vahel.

Iga sisend ja väljund tuli käsitsi aktiveerida ning jälgida läbi TIA Portali lipiliini programmi reaajas väärtuse muutumist. Samuti märkisime üles kontrollimise käigus iga anduri väljundi tüübi, mis on tähtis, sest iga andur on erinev. Väljundsignaaliks võib olla nii tõene ehk 1 või vale ehk 0. Kuigi väljast võib tunduda selline detail tähtsusetu, oli programmeerimise etapis see ülioluline.

5 PROGRAMMEERIMINE

Enne programmeerimise alustamist tuleb esimese asjana elektriskeemi järgi aru saada, mille jaoks on liin mõeldud, millised on positsioonid, andurid, nende paigutus ning kuidas on need mõeldud ühtse kooslusena toimima. Lipiliinil oli kokku 11 erinevat positsiooni ja 17 erinevat andurit.



Joonis 5.1 Elektriskeem: lipiliini külg- ning pealtvaade

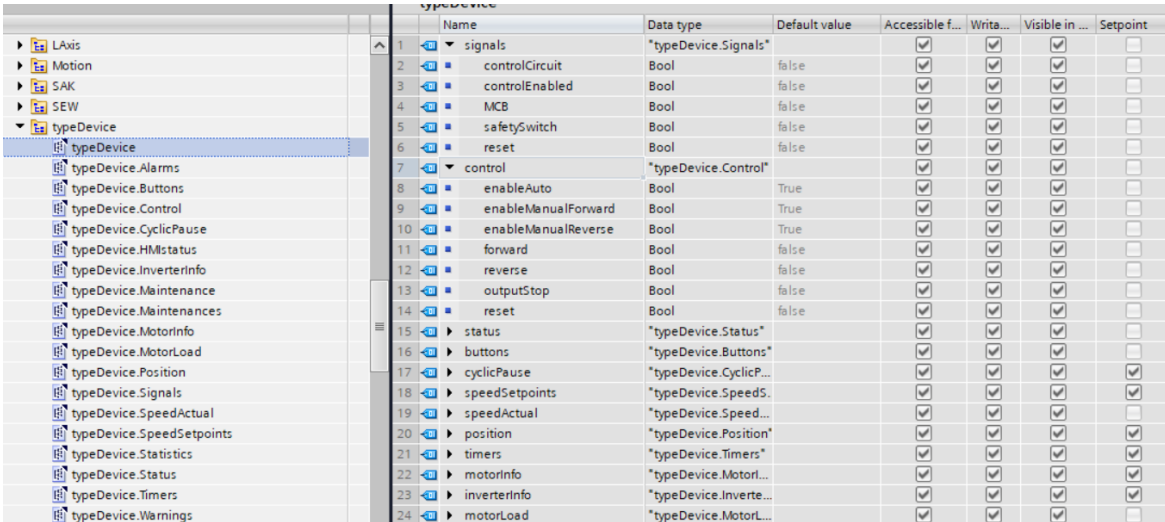
Kavandatava liini ülesanne oli koguda lippe metallist lipikorvidesse, mida tõstukijuhid liinile üks haaval tõstma hakkasid. Elektriskeemi lipiliini pealtvaate jooniselt oli kohe aru saada, et korvide jaoks oli planeeritud eraldi pikendatud puhver. Mitu korvi puhvrile mahub, oli skeemilt keeruline oletada, kuid ilmselt maksimaalselt kolm. Samuti silmasin külgvaate jooniselt seda, et lipiliinile mahub ka pikuti kolm lipikorvi: üks enne lifti, teine lifti alla, täitmisele, ning kolmas, juba täidetud lippidega korv, lifti järele.

Kuigi automaatikas kasutatakse tuhandeid erinevaid andureid, oli selles tööfaasis lipiliiniga kergem hakkama saada. Iga liiniandur oli mõeldud kindla positsiooni liikumise peatamiseks välja arvatud üks mootori pöörlemiskontrolli andur ja kaks induktiivandurit lifti otsas, mis kontrollisid seda, et lift enda piirianduritest mööda ei sõidaks. Samuti on elektriskeemi joonistel näha kahte liinivälise turvaradarit, millega mina antud projekti raames ei tegele.

5.1 Kontrolleri programmeerimine

S7-1500 seeria Siemensi kontrolleri programmeerimine käib TIA Portalis. Kood jaguneb suures osas kaheks osaks: andmeplokkideks ja koodiplokkideks. Andmeplokkides defineerid ära kontrollerisisesed muutujad, mille sisse hakkad informatsiooni salvestama. Koodiplokkidesse kirjutatakse kood, mis kogub sisenditest ja väljunditest informatsiooni, salvestab selle andmeplokkidesse ning manipuleerib nendega, et saavutada soovitud tulem. Koodiplokkid koosnevad parema loetavuse eesmärgil mitmest osast ehk nn võrgust. Lipiliini sisenditeks ja väljunditeks on andurid ning sagedusmuundurid. Sisendite ja väljundite tabeli koostas juba sisendite ja väljundite kontrolli etapis.

Alustasin TIA Portal keskkonnas programmi põhja komplementeerimist. Kasutame oma ettevõttes automatiseerimiseks enda loodud standardeid. Iga positsioon ja andur sai enda muutuja andmetüübi. Andmetüüp kirjeldab ära andmeploki sees oleva muutuja, mille sisse kogutakse temast kõik võimalik informatsioon: liikumiskiirused, enkooderite väärtused, veakoodid jne. Andmetüüpe oleme arendanud pikalt ning tihti lisandub sinna midagi uut.



Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1 signals	*typeDevice.Signals*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2 controlCircuit	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 controlEnabled	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 MCB	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 safetySwitch	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6 reset	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7 control	*typeDevice.Control*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8 enableAuto	Bool	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9 enableManualForward	Bool	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10 enableManualReverse	Bool	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11 forward	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12 reverse	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13 outputStop	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14 reset	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15 status	*typeDevice.Status*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16 buttons	*typeDevice.Buttons*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17 cyclicPause	*typeDevice.CyclicP...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18 speedSetpoints	*typeDevice.SpeedS...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19 speedActual	*typeDevice.Speed...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20 position	*typeDevice.Position*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21 timers	*typeDevice.Timers*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22 motorInfo	*typeDevice.Motori...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23 inverterInfo	*typeDevice.Inverte...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24 motorLoad	*typeDevice.MotorL...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Joonis 5.2 TIA Portal: osa positsiooni puhul kasutavast andmetüübist

Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1 name	WString	WSTRING#"	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2 signal	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 on	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 off	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 onOff	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6 onTime	Time	T# 100ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7 offTime	Time	T# 100ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8 sensorFault	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9 sensorFaultCounter	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10 sensorOnFault	Time	T# 1m	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11 sensorOffFault	Time	T# 1m	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12 onMeasuring	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13 offMeasuring	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14 onMeasuringTime	Time	T# 0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15 offMeasuringTime	Time	T# 0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16 onMeasuringLast	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17 offMeasuringLast	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18 onMeasuringLastTime	Time	T# 0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19 offMeasuringLastTime	Time	T# 0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20 color	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21 colorFiltered	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22 distanceTravelled	Int	1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23 force	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24 forceOn	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25 forceOff	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Joonis 5.3 TIA Portal: andurite puhul kasutatav andmetüüp

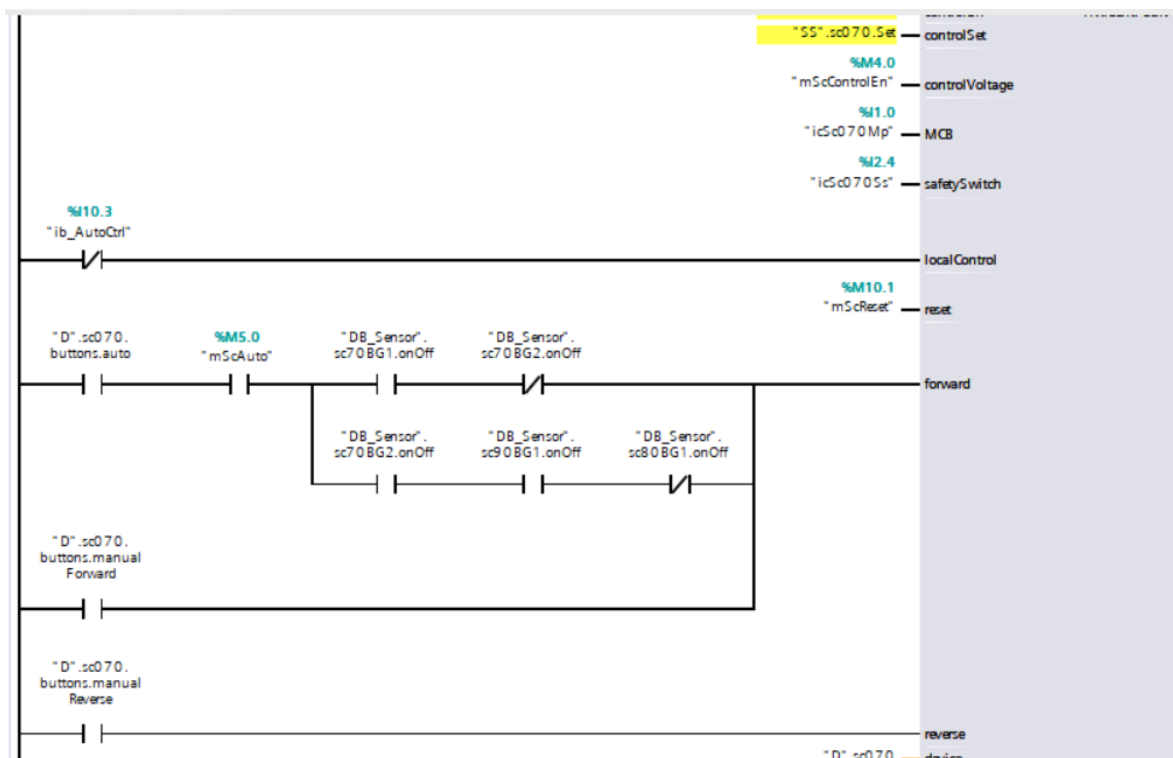
Järgmiseks kirjeldasin ära erinevad mälubitid ning koodiploki muutujad, mida hakkasin koodis vaheandmete salvestamiseks või mõne sündmuse toimumise tähistamiseks kasutama. Näiteks, lippe laob korvi edasi ja tagasi sõitev liikuvate kettidega laduja konveier. Iga kord, kui konveieri nn noka anduri ette jäi lipp, liigutas kontroller lifti veidi alla poole, et laduja konveieril oleks jälle võimalus liikuda ja ruum, kuhu lippe laduda. Iga lifti ning laduja konveieri kohta on olemas bitid, mis kirjeldavad, kas need on jõudnud oma liikumisega lõppu või on alles alguses.

Name	Data type	Address	Retain	Access...	Writa...	Visibl...	Supervision	Comment
1 mScControlEn	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Stick collecting Control Enabled
2 mScControlStart	Bool	%M4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC Control Start
3 mScAuto	Bool	%M5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC AUTO Enabled
4 mScAutoStart	Bool	%M5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC AUTO Start
5 mScSafetyReset	Bool	%M10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC Safety Reset
6 mScReset	Bool	%M10.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC Reset
7 mScAutoED	Bool	%M11.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC Button: Auto (CP03:SB01) ED
8 mScO61Brake	Bool	%M13.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		61 SC Stick lift Brake command
9 mScO605equenceRun	Bool	%M14.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		60 SC Stick lift Sequence running
10 mScO70MPE	Bool	%M14.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		70 SC Stick trolley chain Middle (220B211...
11 tBelt1StopOff	Timer	%T0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12 tBelt2StopOff	Timer	%T1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13 tScSiren	Timer	%T2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SC Siren H5S1 (H5S1) Duration
14 tScO60BrakeReleased	Timer	%T4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		60 SC Stick lift Brake released - Moving all...

Name	Data type	Start value	Retain	Access...	Writa...	Visibl...	Se...	Supervision	Comment
1 Static	Static		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2 O60PosDown	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O60 SC Stick lift Position Down
3 O60PosUp	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O60 SC Stick lift Position Up
4 O60NewTrolley	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O60 SC Stick lift Bring new trolley
5 O60FillingUp	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O60 SC Stick lift in Filling mode Up
6 O60FillingDown	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O60 SC Stick lift in Filling mode Down
7 O50PosB	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O50 SC Moving stick chain - moving Position Beginning
8 O50PosE	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		O50 SC Moving stick chain - moving Position End

Joonis 5.4 TIA Portal: Kirjeldatud mälubitid ja koodiploki muutujad

Kui andmetüübid, mälubitid ning üksikmuutujad olid defineeritud, lõin koodiplokid, milles hakkasin positsioonide ning andurite andmetüübi muutujaid sisenditega täitma. Iga anduri puhul määrame oma koodis alati ära tema sisendi ning selle suuna. Positsioonidele määrame ära võrgud selle nuppude, kiiruse ning kontrolli kohta. Samuti kirjutame positsiooni andmeplokki ära ka sagedusmuunduri info, mis seda mootorit kontrollib. Järgisin sama praktikat ka selles projektis.



Joonis 5.5 TIA Portal: osa pika lipiraami sisseveo keti automaatika koodist

Joonisel 5.4 olevast koodis on kirjas, et lokaalne kontroll käivitub siis, kui liini kõrval olevalt paneelilt lülitatakse lüliti automaatrežiimilt lokaalrežiimile. Samuti ütleb kood, et lipiraam saab automaatselt edasi liikuda vaid siis, kui liin on automaatrežiimil ning lipiraami sisseveoketi esimene andur näeb raami ja viimane andur ei näe raami või mõlemad lipiraami sisseveoketi andurid näevad lipiraame ja järgmise positsiooni lipiraami tuvastuse andur ei näe raami. Lipiraami sisseveoketti saab edasi ja tagasi liigutada käsitsi lokaalrežiimis liini kõrval oleva paneeli abil.

Terve lipiliini automaatikakoodi kirjutasin Ladder logic programmeerimiskeeles. Keel on välja mõeldud, et kujutada koodi graafiliselt ning lihtsustada selle lugemist. Automaatikas kasutatakse ka SCL ning STL programmeerimiskeeli juhul, kui Ladder logic keeles kirjutatud kood valgub liiga laiali ning kaotab oma loetavuse.

6 VISUALISEERIMINE

Pärast põhja loomist ning koodi kirjutamist tuli looma hakata algset visualisatsiooni. Visualiseerimise eesmärk on luua arusaadav vaade operaatorile liinist, selle olekust ning häiretest. Samuti võimaldada operaatoril mõningal määral või täielikult liini läbi selle kontrollida. Visualisatsiooni oli vaja ka lipiliini automaatika testimise jaoks.

6.1 Eeltöö visualiseerimise jaoks

Esmalt tuli minna tagasi elektriskeemi juurde (vt Joonis 5.1) ning analüüsida, kuidas on seal liini positsioone ja andureid kujutatud. Skeemilt nägin, et kujutatud oli liini pealtvaade ja külgvaade – küll aga raskendatud kujul. Visualisatsioonis tuli mõlemat vaadet korralikult lihtsustada: iga andur, positsioon ning nende nimed pidid olema kergelt eristatavad. Samuti tuli üle vaadata positsioonide ja andurite reaalelulised asukohad: kõik pidi parima jälgivuse tagamiseks olema vastavuses.



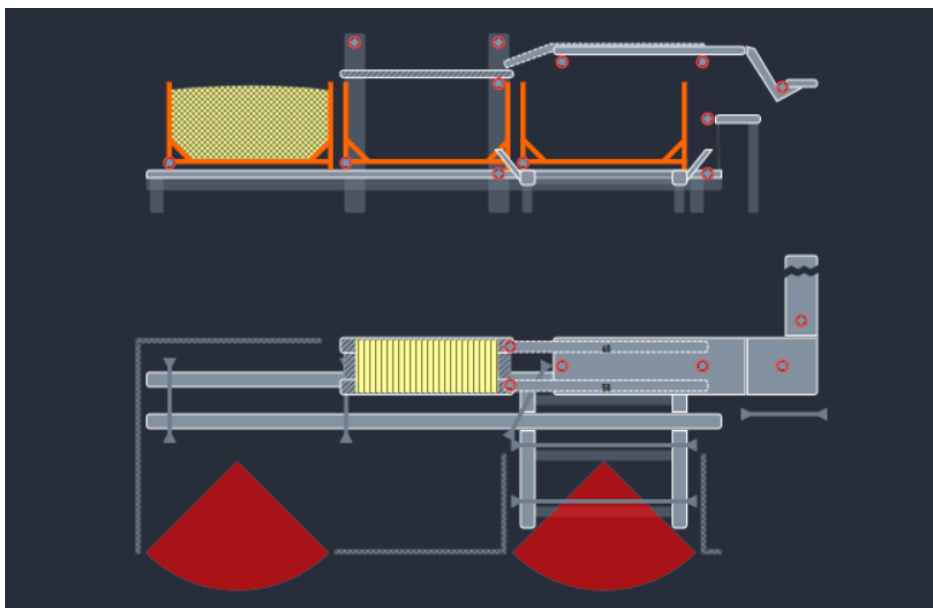
Joonis 6.1 Lipiliin: Valmis ehitatud lipiliin

Käisin ümber lipiliini, ronisin positsioonide vahel ja peal, skitseerisin jooksvalt kõikide positsioonide umbkaudse paigutuse ning märkisin ära andurite asukohad.

6.2 Lipiliini külg- ja pealtvaade ning andurid

Alustasin visualiseerimist TIA Portalis. Joonistasin TIA Portali objektide abil välja nii külgs- kui ka pealtvaate. Vaja oli mõlemat vaadet, kuna iga andur, positsioon või mõni muu element ei paista ainsast vaatest välja: näiteks, turvaradari koonused või lifti "üleväl-all" andurid.

Helehallid valge äärega objektid tähistasid positsioone. Tumehallid objektid tähistasid staatilisi osi nagu lipiliini raamistik või lipiliini ümbritsev aed. Oranžid joonistused tähistasid lipikorve ning kollane mass selle sees kogutud lippe. Väiksed hallid ringid punaste ääristega tähistasid andureid ja punased koonused turvaradareid.














Joonis 6.2 TIA Portal: lipiliini visualiseerimine, positsioonid ning andurid

Visualisatsioon tuli teha dünaamiline. Positsioonide, andurite värvid, lipiraamide visualisatsioonis kuvamine, nende täituvus ja lipilifti ning laduja konveieri asukoht pidid muutuma vastavalt nende reaallolulisele olekule. Dünamiseerimise ning jooksva info lugemise jaoks saingi kasutada kontrolleri muutujaid. Need sidusin vastava objekti atribuudiga, mida see kontrollima hakkas. Keerulistemate dünamiseerimiste, nagu positsiooni asukoha määramiseks, ei piisa lihtsalt muutuja sidumisest: kirjutama peab ka javascripti keeles koodi.

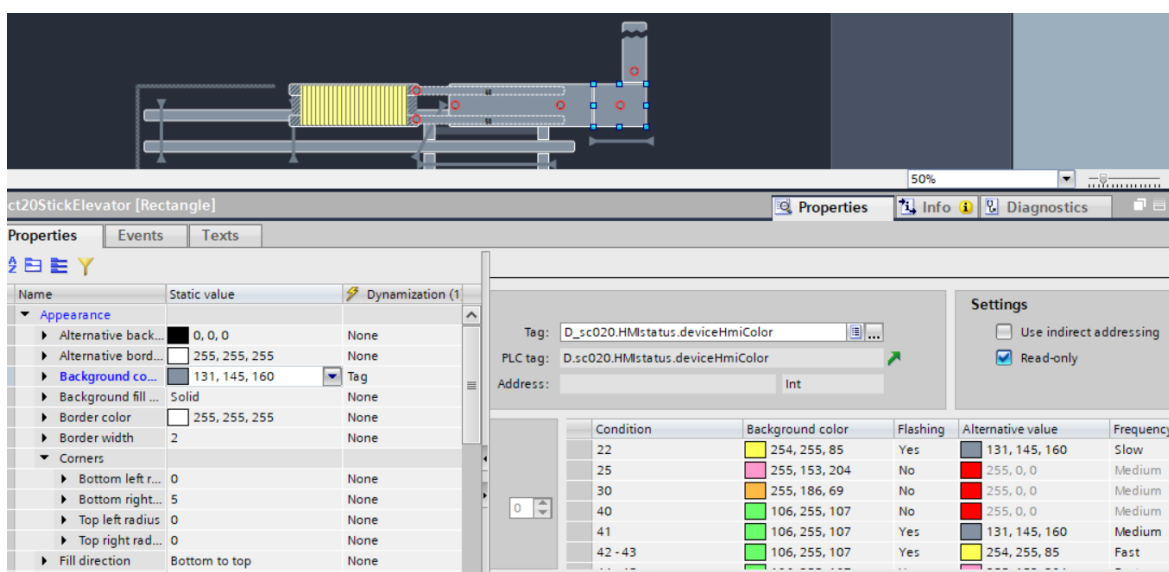
6.3 Visualisatsiooni dünamiseerimise näited

Operaatoritele tuleb võimalikult kiiresti edastada info erinevate olekute kohta. Üks parimaid mooduseid selleks on liini visualisatsioonis positsiooni värvimine. Koodis on igal positsioonil oma andmetüüp, mis väljastab vastavalt olekule värvikoodi. Visualisatsioonis pidin kõigest iga positsiooni puhul lugema selle andmetüübi värvikoodi ning vastavalt värvima.

Tabel 6.1 Visualisatsioon: positsioonide andmetüübi värvikoodid

	Liin peatatud, juhtahel maas
	Turva aktiveerunud
	Positsiooni viga (täpsem informatsioon häirete leheküljel)
	Käsirežiim
	Käsirežiimis tagasi liikumine
	Lokaalrežiim (liini saab juhtida vaid kõrval olevast puldist)
	Automaatrežiim
	Automaatrežiimis edasi liikumine
	Automaatrežiimis tagasi liikumine
	Käsirežiimis edasi liikumine
	Lokaalrežiimis edasi liikumine

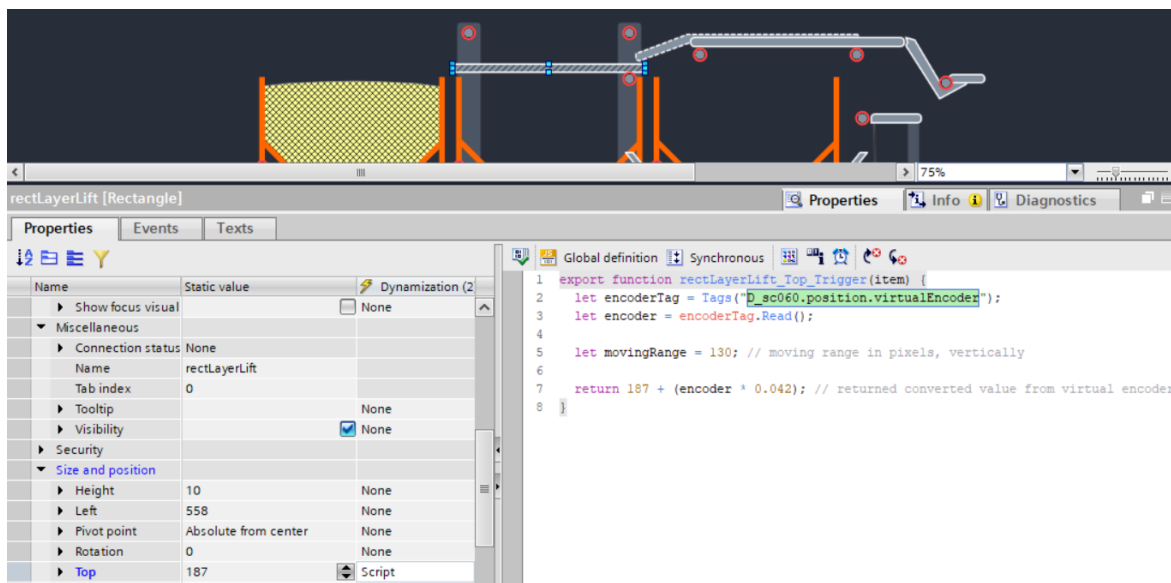
*kaks värvi tabelis tähistab positsiooni visualisatsioonis vilkumist



Joonis 6.3 TIA Portal: dünamiseerimine läbi muutuva otsese sidumise – positsiooni värvimine

Dünamiseerimisel ei piisa alati objekti atribuudi otse muutujaga sidumisest. Et kuvada visualisatsioonis dünaamiliselt näiteks lipilifti või laduja konveieri liikumist, pidin looma nendele kahele positsioonile virtuaalse enkooderi. Selleks pidin teadma positsiooni liikumiskiirust ning vahemaad, mille ta näiteks kümne sekundi jooksul läbis. Tulemuseks oli suhtarv, mille väikseimat ning suurimat arvu teades sain kindlaks teha lipilifti ning laduja konveieri asukoha. Virtuaalse enkooderi väärtuse arvutab välja vajalike andmete olemasolul meie andmetüüp.

Et visualisatsioonis positsiooni asukohta kuvada, pidin teadma, mitu pikslit on visualisatsioonis positsiooni liikumisulatus. Seejärel pidi arvutama kordaja virtuaalse enkooderi ulatuse ja pikslite arvu vahel ning teisendama javascripti abil enkooderi väärtuse piksliteks. Virtuaalseid enkoodereid kasutatakse siis, kui mootoritel puuduvad päris enkooderid ning operaatoritel on vaja ligilähedast teavet positsiooni asukoha kohta. Kui oluline on täpsus, siis selline lahendus ei sobi.

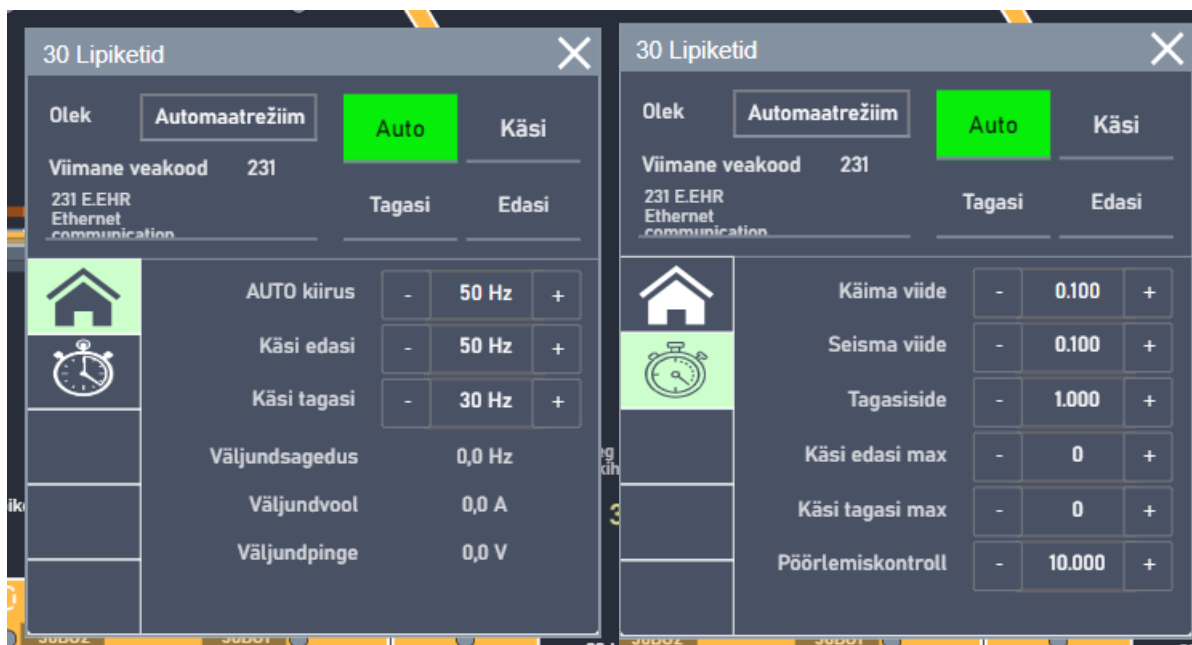


Joonis 6.4 TIA Portal: dünamiseerimine läbi javascripti - virtuaalne enkooder

6.4 Interaktsioon lipiliiniga läbi visualisatsiooni

Visualisatsioon pidi olema interaktiivne. Selleks lõin nii positsioonidele kui ka anduritele hüplikakna funktsioonid. TIA Portalis teatud kui *faceplate popup*. Kui operaator klikib mõnele positsioonile või andurile, avaneb hüplikaken. Aken on javascripti abil seotud selle positsiooni või anduri andmetüübiga.

Hüplikaken näitab kõige ülemises osas seotud positsiooni olekut, sagedusmuunduri viimast veakoodi ning režiiminuppe. Menüü esimene lehekülg kuvab informatsiooni positsiooni sagedusmuunduri kohta ning sealt saab samuti muuta kiirust automaat- ja käsirežiimis. Menüü teine lehekülg võimaldab muuta ooteaegu ning käsirežiimi liigutamise ja pöörlemiskontrolli seadeid.



Joonis 6.3 Visualisatsioon: Positsiooni hüplikaken (2 lehekülge)

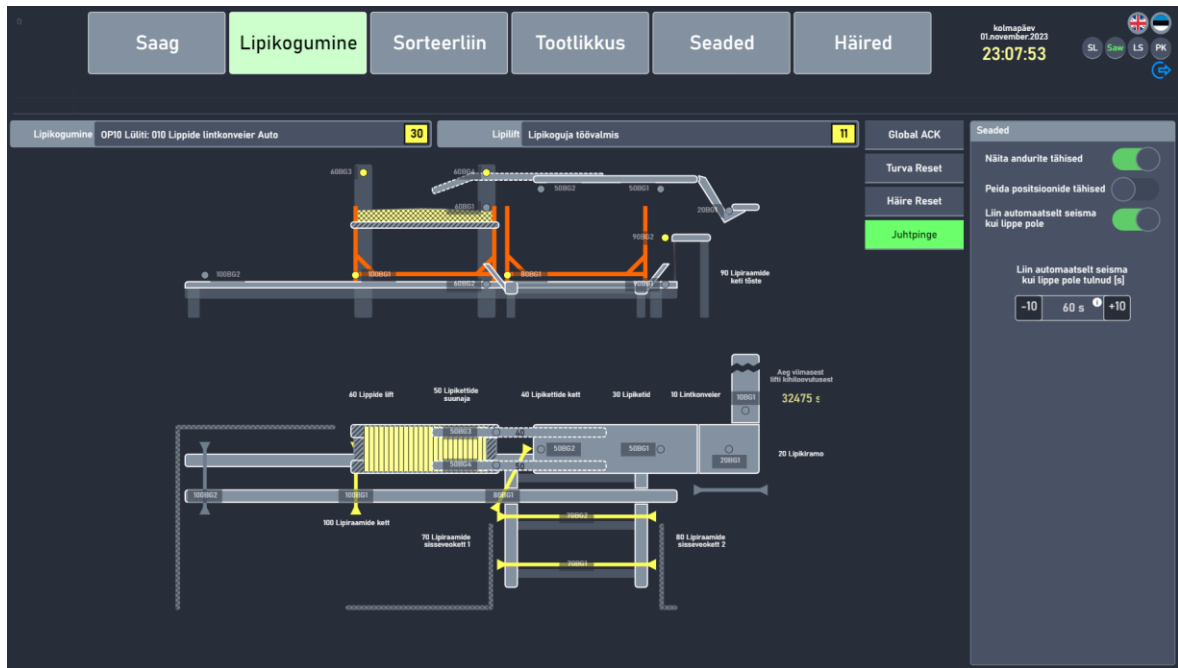
Anduri hüplikaken näitab kõige ülemises osas vastavat tähist, sees-väljas signaali ning filtreeritud signaali ehk anduri täpsemat olekut. Menüü esimesel leheküljel on võimalik muuta andurite ooteaegu ehk kui kaua läheb sellel aega, et reageerida muutusele või see unustada. Samuti kaks nuppu, millega saad vastava anduri sees või väljas olekut hoidma panna. Menüü teine lehekülg kuvab, kui kaua on või oli anduri signaal sees või väljas ning, kui tegu on optilise anduriga, mis on või oli selle mõõdetud distants.



Joonis 6.4 Visualisatsioon: Anduri hüpikaken (2 lehekülge)

Lisaks positsiooni ning andurite hüpikaknafunktsioonidele on lipiliini visualisatsioonis veel võimalik nelja nupu abil teostada osaliselt häire ja turva signaalide taastamist, anda globaalne nõusolek turvasignaali taastamiseks ning kontrollida juhtpinge olemasolu. Lipiliinil on ka üks parameeter, millega saab operaator ära määrata aja sekundites, peale mida jääb lipiliin seisma, kui sorteerliinilt lippe peale pole tulnud.

Erinevate häirete kohta saab operaator informatsiooni lehe üleväl servas asuvalt häireteribalt või häirete leheküljelt. Veel lõin lipiliini visualisatsiooni operaatorite soovil kaks olekuriba, millelt on võimalik saada informatsiooni nii lipilifti liikumise ning lipikorvi täitmise protsessi kohta.



Joonis 6.5 Visualisatsioon: lõplik lipiliini visualisatsioon, täisvaade

7 KONTROLLERI KOODI TESTIMA

Peale visualisatsiooni lõpetamist hakkasin oma kirjutatud koodi testima. Läbisin protsessi sujuvalt, kuna lipiliini automaatika on suhteliselt lineaarne. Sisuliselt teeb liin kogu aeg sama asja: kogub kihthaaval lipikorvi lippe ja kontrollib korvi täituvust ning korvi täitumise korral liigutab täis korvi liini lõppu ja tühja korvi olemasolul võtab puhvrist uue.

Testimise jaoks oli mul kogu aeg käes sülearvuti, millelt nägin otsepildis jooksvaid sisendeid ja väljundeid. Samuti sai sülearvutist kontrollida liini positsioone vajadusel läbi visualisatsiooni. Visualisatsiooniga kooskõlas testisin liini juhtimist ka liini kõrval olevast nupupaneelist.

7.1 Riistvaralised vead

Kõige rohkem tuli testimise käigus välja erinevaid riistvaraga seotud vigu. Tavaliselt olid nendeks väiksemad asjad, nagu andur konveieri ääres, mis pidi konveierit märgates liini seisma jätma. Selliste probleemide lahendamiseks pidin appi kutsuma kohaliku elektrikule, kelle abiga liigutasin või suunasin andurit, mis varasemalt valesti paigutatud oli.

Kõige suurem riistvaraline viga tuli välja kohe testimise alguses. Kõige pikemal konveieril oli olemas seda tõstev ja langetav mootor, mis ei sobinud: parameetrid ei ühtinud sagedusmuunduriga, mis tähendas, et mootorit ei saanud kasutada. Probleemi lahendamiseks kutsuti kohale kohalik mehaanik, kes asendas mootori ajutiselt silindriga. Koodi poole pealt asendasin positsiooni juhtploki sagedusmuunduri loogika lihtsa sisse-välja loogikavärvaga, mis konveieri tõstet ajutiselt juhtima hakkas.

7.2 Koodi vead

Ka koodi poolelt leidsin erinevaid vigu. Tavaliselt oli tegu mõningate vähearvestustega nagu anduri viite puudumine (konveier jäi lipikorvi nähes liiga vara seisma) või mõne loogikavärvaga puudumine. Kõige suurem koodipoolne viga oli kindlasti üks positsioon,

mis ei jäänud peale turvalüliti vajutust seisma. Sellised vead tuleb alati kohe tuvastada ning kõrvaldada, vastasel juhul võib mõni situatsioon kujuneda operaatorite jaoks eluohtlikuks.

8 SÜSTEEMI TURVALISUS

Nii automaatikakood kui ka visualisatsioonile ligipääs peavad olema kaitstud. Kuigi suurema töö selles valdkonnas teeb iga projekti puhul alati Nordwood OÜ ettevõtetes ära IT osakond, anname ka meie, automaatikud, oma panuse turvalisusesse.

8.1 Brauseripõhine visualisatsioon ja programm

Visualisatsiooni leheküljele pääseb ligi vaid siis, kui seda jooksutaval masinal on olemas vajalikud litsentsid. Avanenud leheküljel sisse logida saab vaid siis, kui olemas on programmis loodud kasutaja. Peale sisse logimist mõnel liinil muutusi teha ainult juhul, kui kasutajal on programmi poolt antud vajalikud õigused. Mitmetasemelisus tagab saeveski automaatikasüsteemide turvalisuse.

Programmiseselt on kaitstud vaid kontrolleri kood ning selles muudatuste tegemine. Visualisatsioonis muudatuste tegemine programmiseselt kaitstud ei ole. Kui terve programm peaks sattuma mõne halba sooviva osapoole kätte, ei pea siiski keegi midagi kartma. Millegi halva korda saatmiseks peab olema kurjategijal kõik Siemensi programmid, litsentsid programmi avamiseks ning ligipääs kohalikku võrku.

8.2 Kohalik võrk

Absoluutselt iga kontrolleri, arvuti, virtuaalne masin ning kõik ülejäänud kommuniqueerivad seadmed töötavad kohalikus võrgus. IP-aadressid neile, sealhulgas ka meie brauseripõhisele visualisatsioonile, määrab IT osakond. Kohalikku võrku on võimalik ligi pääseda läbi VPN teenuse, kuid ainult juhul, kui IT osakond loob sulle autoriseeritud konto. VPNi olemasolu soodustab meie tööd ning annab võimaluse väiksemaid vigu nii koodis kui ka visualisatsioonis parandada distantilt.

9 VÕRDLUS VARASEMA LAHENDUSEGA

9.1 Tehnoloogiline tööpuudus

Lipiliini automatiseerimine lihtsustas suuresti Viru-Nigula saeveski operaatorite tööd. Varasem lippide kogumine ning sorteerimine asendati automatiseeritud liiniga, mis teeb seda mitmeid kordi kiiremalt. Võib arutleda, et automaatne lipiliin annab võimaluse saeveski juhatusel rakendada varasemalt lippide sorteerimisega tegelenud operaatorit mujal, aga kui suur on oht, et operaatorile ei leia enam saeveskis tööd?

Tehnoloogiline tööpuudus on läbi ajaloo ning eriti viimaste aastate jooksul olnud väga päevakajaline teema. Kas automatiseerimine kaotab töökohti või soodustab vastupidiselt hoopis uute töökohtade teket ning spetsialiseerumist: selle küsimuse üle juurdlesid kaks majandusteadlast Hideki Nakamura ning Joseph Zeira. [12]

Nemad jagasid tööstuse uuendamise kaheks: mehaniseerimiseks ning automatiseerimiseks. Kuigi terminid võivad kõlada sarnaselt, on need sisult erinevad. Automatiseerimine eeldab tööstuses operaatori asendamist tehnoloogilise vastandiga ehk millegagi, mis oskab teha informatsiooni põhjal järeldusi ning nendele kiirelt ja õigesti reageerida. Mehaniseerimine aga eeldab kindla mehaanilise lahenduse lisamist, mis lihtsustab operaatorite tööd, kuid ei asenda neid täielikult. Operaatori uueks ülesandeks oleks sellisel juhul näiteks mehaanilise liidese kontrollimine. [12]

Järelduses selgus, et kartus tehnoloogilise tööpuuduse ees on liialdatud. Laialdane automatiseerimine vähendaks töökohtasid, suurendaks tööstust, palkasid ja vähendaks seeläbi vajadust automatiseerimise järele. Lisaks sellele väitsid nad, et praeguste näitajate põhjal töötute inimeste arv mitte ei vähene, vaid koondub nulli suunas. Tähtsa märkena mainisid majandusteadlased mitu korda, et tegemist on siiski teoreetilise paberiga, mis aitab mõista protsesse, mitte ennustada tulevikku. [12]

Kuigi puuduvad kindlad andmed töötajate arvu kohta, viibides ise sageli tööga seonduvatel põhjustel OÜ Nordwood alla kuuluvates saeveskites, julgen väita, et mõne liini automatiseerimine töökohti ei kaota. Tavaliselt eeldab uue liinijupi automatiseerimine seda, et operaator hoiab sellel pidevalt silma peal. Kui juhtub, et uus liinijupp on täiesti automaatne ning ei nõua järelvalvet, leitakse varasemalt kindlat ülesannet täitnud operaatorile saeveskis kiirelt uus rakendus.

9.2 Mõjud operaatorite tervisele

STAMI teadlaste uuring tõestas, et saeveskite töökeskkonna õhus leidub suures koguses kontaminatsiooni. Kolmandik saeveskite töötajatest puutuvad pidevalt töö jooksul kokku õhuga, mille mikroskoopiliste seente ning hallitusseente eoste sisaldus on soovitatust kõrgem. Õhus lendlevate eoste kogus ning koostis varieerub tugevalt aastaegade ning saeveski osakondade vahel. [13]

Oslo Ülikooli teadlased viisid läbi ühe Norra saeveski kümne töötaja peal läbi pikaajalise uuringu, mis uuris nende tööalalist kokkupuudet tolmu ning erinevate õhus lendlevate seente eostega. Kaheksa aasta vältel koguti 79 proovi uuringus osalevate töötajate töökeskkonna kohta. Samal ajal paluti töötajatel teada anda kõikidest tekkinud tervisehädadest. Samuti teostati kümme aastat pärast põhiuuringut järeluuring. [14]

Kõik töötajad kurtsid põhiuuringu ajal korduvatest kopsuhädadest. Keskmiselt esines neid igal töötajal kahe kuu jooksul umbes kaks korda. Kümme aastat hiljem järeluuringus selgus, et saeveski töötajate kopsuhädade sagedus oli vähenenud ning töökeskkonna õhk puhtam. Põhjusteks olid lühidalt uuemate tootmismeetodite kasutusele võtmine: näiteks uued moodused puidu hoiustamiseks, kuivatamiseks ning õhu filtreerimiseks. [14]

Peale Viru-Nigula saeveski lipiliini ehitust ja automatiseerimist ei pea saeveski operaatorid lippe sorteerima käsitsi, mis vähendab kordades nende otsest kontakti puiduga. Kuigi meil puuduvad andmed antud saeveski õhukvaliteedi ning töötajate tervise kohta, võime oletada, et uuendus mõjub töötajate tervisele ainult positiivselt.

KOKKUVÕTE

Käesoleva lõputöö eesmärk oli automatiseerida ning visualiseerida kuivanud lauapakist eraldatud lippide kogumise liin ning võrrelda seda saeveskis eelnevalt kasutatud lippide kogumise meetodiga.

Alustasin elektrikilbis sagedusmuundurite seadistamisega. Peale seda seadistasin Siemensi automaatikaprogrammis võrgu ning andsin IP-aadressid nii kontrolleri, sagedusmuundurile kui ka visualisatsiooni masinale. Seejärel tuli läbida lipiliini elektrikilbi sisendite ja väljundite kontroll. Üle tuli vaadata kõik ühendused erinevate lipiliini andurite, turvalülite, mootorikaitsmete, nupupaneeli nuppude ning hajus-IO vahel.

Peale liiniga tutvumist hakkasin kirjutama kontrollerrisse liinikoodi. Kui kood oli valmis, hakkasin looma interaktiivset visualisatsiooni, läbi mille oli võimalik lipiliini tööd kontrollida. Pärast koodi ja visualisatsiooni loomist tegin programmi poolt kõik, et programm oleks turvaline.

Loen lõputöö eesmärgi täidetuks. Automatiseerisin ning visualiseerisin uue lippide kogumise liini. Samuti võrdlesin seda leitud kirjanduse põhjal varasema lahendusega.

Uus lipiliin on täiesti automaatne ning nõuab minimaalset järelvalvet. Järeldasin, et operaatoritele leitakse suurema tõenäosusega saeveskis uus rakendus selle asemel, et neid vallandada. Samuti, uus automaatne lipiliin vähendab kordades töötajate otsest kontakti puiduga. Kuigi meil puuduvad andmed antud saeveski õhukvaliteedi ning töötajate tervise kohta, järeldasin leitud uurimistööde põhjal seda, et uuendus mõjub töötajate tervisele vaid positiivselt.

SUMMARY

The purpose of this thesis was to automate and visualize a new collecting line for sticks that are separated from a dried board pack and to compare it with the stick collecting method previously used in the sawmill.

I started by setting up the frequency converters in the electrical panel. After that, I configured the network in the Siemens automation program and gave IP addresses to the controller, frequency converter and visualization machine. Then the inputs and outputs of the stick line had to be checked. All the connections between the various stick collecting line sensors, safety switches, motor fuses, buttons on the button panel and distributed IO had to be reviewed.

After familiarizing myself with the stick collecting line, I started writing the line code in the controller. When the code was ready, I started to create an interactive visualization through which it was possible to check the operation of the stick collecting line. After finishing the code and the visualization, I did everything from my end inside the program for it to be as safe as possible.

I consider the aim of the thesis accomplished. I automated and visualized the new stick collecting line. I also compared it with the previous solution based on the literature I found.

The new stick collecting line is completely automatic and requires minimal supervision. I concluded that the operators were more likely to be found a new application rather than being fired. Also, the new automatic stick collecting line reduces workers' direct contact with dry and wet wood. Although we have no data on the air quality of this sawmill and the health of the workers, based on the research I found, I also concluded that the upgrade has only a positive effect on the health of the workers.

KASUTATUD KIRJANDUSE LOETELU

- [1] S. A. Landscheidt and M. Kans, "Automation Practices in Wood Product Industries: Lessons learned, current Practices and Future Perspectives," Lund University, Sweden, Lund, 2016. Accessed: Nov. 06, 2023. [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:1047705/FULLTEXT01.pdf>
- [2] T. Haaranen, "Bachelor's Programme in Electrical Engineering, Bachelor's Thesis," Lappeenranta–Lahti University of Technology, Finland, Lappeenranta, 2023. Accessed: Nov. 06, 2023. [Online]. Available: https://lutpub.lut.fi/bitstream/handle/10024/165365/Tuomas_Haaranen_Kandidaatinty%C3%B6.pdf?sequence=3&isAllowed=y
- [3] A. Oja, "ETTEPANEKUD OÜ VARA SAEVESKI KOMBILIINI TOOTLIKKUSE TÕSTMISEKS," 2016, Accessed: Nov. 14, 2023. [Online]. Available: https://dspace.emu.ee/xmlui/bitstream/handle/10492/2539/Andreas_Oja_2016BA_ME_t%C3%A4istekst.pdf?sequence=1&isAllowed=y
- [4] Siemens AG, "S7-1500 CPU 1517T-3 PN/DP (6ES7517-3TP00-0AB0) Equipment Manual." Siemens AG, Nov. 2022. Accessed: Nov. 06, 2023. [Online]. Available: https://cache.industry.siemens.com/dl/files/190/109739190/att_895934/v1/s71500_cpu_1517t_3_pndp_manual_en-US_en-US.pdf
- [5] Siemens AG, "IM 155-6 PN ST interface module (6ES7155-6AU01-0BN0) Equipment Manual." Siemens AG, Oct. 2020. Accessed: Nov. 06, 2023. [Online]. Available: https://cache.industry.siemens.com/dl/files/173/59768173/att_918231/v1/et200sp_im_155_6_pn_st_manual_en-US_en-US.pdf
- [6] T. Aditya, "Research to study Variable Frequency Drive and its Energy Savings," vol. 2, no. 6, p. 4, 2013.
- [7] Mitsubishi Electric Corporation, "FR-E800 Instruction Manual (Connection) (200V / 400V CLASS)." Mitsubishi Electric Corporation, 2004. Accessed: Nov. 06, 2023. [Online]. Available: https://www.lcautomation.com/wb_documents/Mitsubishi/Mitsubishi%20E800%20Manual%20-%20Connection.pdf
- [8] J. Kjellsson, A. E. Vallestad, R. Steigmann, and D. Dzung, "Integration of a Wireless I/O Interface for PROFIBUS and PROFINET for Factory Automation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4279–4287, Oct. 2009, doi: 10.1109/TIE.2009.2017098.
- [9] Siemens AG, "PROFINET with STEP 7 Function Manual." Siemens AG, Nov. 2022. Accessed: Nov. 06, 2023. [Online]. Available: https://cache.industry.siemens.com/dl/files/856/49948856/att_897210/v1/profinet_step7_v18_function_manual_en-US_en-US.pdf
- [10] Siemens AG, "STEP 7 and WinCC Engineering V17 System Manual." Siemens AG, May 2021. Accessed: Nov. 06, 2023. [Online]. Available: https://cache.industry.siemens.com/dl/files/671/109798671/att_1071920/v1/STEP_7_WinCC_V17_enUS_en-US.pdf
- [11] Mitsubishi Electric Corporation, "FR Configurator2 Instruction Manual." Mitsubishi Electric Corporation, 2014. Accessed: Nov. 06, 2023. [Online]. Available: <https://vfds.com/content/manuals/mitsubishi/mitsubishi-fr-configurator2.pdf>
- [12] H. Nakamura and J. Zeira, "Automation and unemployment: help is on the way," *J Econ Growth*, Sep. 2023, doi: 10.1007/s10887-023-09233-9.
- [13] K. A. Afanou, W. Eduard, H. B. Laier Johnsen, and A. Straumfors, "Fungal Fragments and Fungal Aerosol Composition in Sawmills," *Ann Work Expo Health*, vol. 62, no. 5, pp. 559–570, Jun. 2018, doi: 10.1093/annweh/wxy022.
- [14] K. Færden *et al.*, "Hypersensitivity pneumonitis in a cluster of sawmill workers: a 10-year follow-up of exposure, symptoms, and lung function," *Int J Occup Environ Health*, vol. 20, no. 2, pp. 167–173, Apr. 2014, doi: 10.1179/2049396714Y.0000000063.