

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Bo Aaron Kooser 193552IACB

Prügikasti automaatse pesusüsteemi juhtimisprogramm prügiautole

Bakalaureusetöö

Juhendaja: Andres Rähni
Tehnikateaduste
magister

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Bo Aaron Kooser

16.05.2022

Annotatsioon

Bakalaureusetöö on kirjutatud projektist, mille eesmärk oli luua koos ettevõttega Eritehnik OÜ pesusüsteem prügiautole, mis peseb peale prügi kallamist ära prügikasti. See töö käsitleb täpsemalt programmi loomist sellele süsteemile. Süsteemi juhib Siemensi kontrolleri, mis on programmeeritud tarkvaraga *Logo! Soft comfort V8* [1]. Programm luuakse *LOGO! 24CE* [2] kontrolleri jaoks, mille valimist põhjendatakse töös “Prügikastide automaatse pesusüsteemi juhtiva loogikakontrolleri valik ja sellele loodava tarkvara testimine” [3] autori Jürgen Kolli poolt, kellega koostöös projekt valmis. Käesolev töö tutvustab programmi vaheversioone ja arenguetappe, mis viisid töö lõpptulemuseni. Ülevaade antakse kasutatud arendamis meetodist ning ka üldistest kontrolleri programmeerimise viisidest. Kuigi simuleerides sai programm testitud ja tehtud ja vastas tellija ootustele, ei jõutud antud töö raames selle testimiseni ega seadistamiseni praktikas, sest tellija ei saanud õigeaegselt kätte teisi vajalikke vahendeid. Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 22 leheküljel, üheksa peatükki, kümme joonist, ühte tabelit.

Abstract

Management Program for Garbage Truck Trash Bin Washing System

The thesis is written about a project that aimed to create, together with the company Eritehnik OÜ, a washing system for a garbage truck that washes the trash box after pouring it. This work more specifically is about creating a program for this system. The system is controlled by Siemens PLC, which is programmed with the software Logo! Soft comfort V8 [1]. The program is created for the LOGO! 24CE [2] controller, the choice of which is justified in the work “Selecting a Programmable Logic Controller for an Automatic Washing System for Garbage Cans and Testing the Software Created for the System” [3] by the author Jürgen Koll, with whom the project was completed. The work introduces the intermediate versions of the program and the development stages that led to the result of the work. An overview is given of the development method used as well as the general PLC programming methods. Although the program was tested and performed in accordance with the customer's expectations during the simulation, it was not possible to test or set it up in practice within the given work, because the customer did not receive other necessary resources in time. The thesis is in Estonian and contains 22 pages of text, nine chapters, ten figures, one table.

Lühendite ja mõistete sõnastik

<i>AND</i>	Konjunktsioon
Arendaja	Programmi looja
FBD	<i>Function Block Diagram</i> ehk funktsioon plokkiagramm
Kamm	Prügiauto prügipressi osa, mis tõstab üles prügikasti
<i>NAND</i>	Konjunktsiooni inversioon
<i>OR</i>	Disjunktsioon
PCB	Trükkplaat
PLC	Programmeeritav loogika kontrolleri
RLL	<i>Relay Ladder Logic</i> ehk redeldiagramm
ST	<i>Structured Text</i> ehk struktureeritud tekst
Testija	Programmi testija
<i>XOR</i>	Disjunktsiooni inversioon

Sisukord

1 Sissejuhatus	10
2 Projekti taust.....	11
3 Arendusmeetodid.....	12
3.1 Agiilne arendusmeetod	12
3.2 Kanban arendusmeetod.....	13
3.3 Kosk arendusmeetod.....	13
4 Valitud arendusmeetod	15
5 Projekti nõuded.....	16
5.1 Versioon 1.....	16
5.2 Versioon 2.....	16
5.3 Versioon 3.....	17
5.4 Versioon 4.....	17
5.5 Versioon 5.....	17
5.6 Versioon 6.....	18
6 Kontroller	19
7 Programm	20
7.1 Programmeerimistarkavara tutvustus	20
7.2 Programmi esitlusvormid	20
7.2.1 FBD	20
7.2.2 RLL.....	21
7.2.3 ST	22
7.3 Valitud meetod	23
7.4 Programm	23
7.4.1 Esimene osa	23
7.4.2 Teine osa.....	24
7.4.3 Kolmas osa	25
7.4.4 Neljas osa.....	25
7.4.5 Viies osa	26
7.5 Edasiarendusvõimalused	26

8 Kasutus- ja seadistusjuhend.....	28
8.1 Seadistusjuhend	28
8.2 Kasutusjuhend	30
9 Kokkuvõte	31
Kasutatud kirjandus	32
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	34
Lisa 2 – Programmi ülevaade	35

Jooniste loetelu

Joonis 1. Agiilse arendus meetodi töö tsükkel [8].....	12
Joonis 2. Kosk meetoditöö tsükkel [10]	14
Joonis 3. Funktsioon plokk diagramm.....	21
Joonis 4. Redel diagramm	22
Joonis 5. Struktureeritud teksti põhine programmi näide [16]	23
Joonis 6. Sisendite kontroll.....	24
Joonis 7. Pesupeade valik	25
Joonis 8. Pesu pikkuse valimine	25
Joonis 9. Pesu pea sõidu asendisse liigutamine.....	26
Joonis 10. Reovee süsteemi juhtimine.....	26

Tabelite loetelu

Tabel 1. Seadistusjuhend	129
--------------------------------	-----

1 Sissejuhatus

Töö eesmärk oli luua prügiauto prügipressi pesusüsteemile programm, mille toimetel peseb süsteem prügikasti. Loodav süsteem toob ajalise ja rahalise võidu tarbijale. Probleemiks on soodsa ning töökindla pesusüsteemi puudumine turul. Senised lahendused ei ole vastupidavad või on liialt kallid nii energia kui ka tööjõukulu poolest. Töö tellijaks on Eritehnik OÜ, kellega koostöös projekt valmib. Ettevõtte on edukalt toimetanud nii Baltikumis kui ka Skandinaavias pea 20 aastat ehk omab põhjalikku teadmiste baasi antud valdkonna kohta. Töötav süsteem loodeti saada valmis auto peal maikuu lõpuks, et saata loodud prototüüp testimisele sihtriiki. Süsteemi valmimistähtaeg on liikunud 2022 aasta juuli kuuks, sest tellija ei ole kätte saanud kõiki vajalikke osasid süsteemi kokkupanekuks. Programm süsteemi juhtimiseks sai valmis õigeaegselt ning on tellijale üle antud. Töö tegemisel kasutati agiilset arendusmeetodit, mis valiti välja kolme laialt levinud meetodi vahel. Programm tehti tarkvaraga *Logo! Soft comfort V8* [1] ning see on Siemensi enda tarkvara spetsiaalselt valitud kontrolleri jaoks. Programmeeriti *LOGO! 24CE* kontrolleri [2]. Programm otsustati luua kasutades funktsioon plokkidel põhinevat viisi kuna sobis kõige paremini loodu kiireks simuleerimiseks.

Selles töös tutvustatakse lähemalt projekti tausta, mis kirjeldab lähemalt lahendatavat probleemi ning turul pakutavaid variante. Peatükis kolm, kirjeldatakse enim tuntuimaid arendusmeetodeid ning põhjendatakse, miks valiti antud töö arendamiseks agiilne arendusmeetod. Neljandas peatükis tutvustatakse kuidas jõuti projekti lõppnõueteni ning töö varaseimaid versioone. Viiendas punktis tuuakse välja erinevad võimalused kasutatud kontrolleri programmeerimiseks ning põhjendatakse, miks otsustati kasutada funktsiooni plokkidel põhinevat meetodit. Samas peatükis selgitatakse kuidas on programm koostatud ning samuti lühike tutvustus ka edasiarenduse võimalustest. Viimases osas selgitatakse kuidas läbi puldi süsteemi juhtida ning kuidas tuleb ülejäänud süsteem ühendada kontrolleriiga.

2 Projekti taust

Skandinaavias ja osades teistes riikides on kohustus pesta prügikasti peale selle tühjendamist. See kohustus kehtib aegadel millal väljas on plusskraadid ning kohustus erineb riigiti, piirkonniti ning prügikasti tüübi poolest [4]. Tiheda asustusega piirkondades tuleb pesta kõik prügikastid. Turul on see probleem lahendatud mitmel viisil, peamine neist on meetod, kus sarnaselt loodavale süsteemile on ehitatud tavalise prügiauto prügipressi peale pesusüsteem. Need süsteemid on valdkonnas kahe suurema tegutseva ettevõtte KEOX OY ja JOAB AS poolt. Mudelid vastavalt „Urbaser 20“ [5] ja „Pesuauton 30“ [6] täidavad küll nõutud funktsionaalsused, kuid on hinna poolest kallid ning pole ennast tellija jaoks praktikas tõestanud. On olemas mudelid, mis on ehitatud ainult pesuautoks ning need sõidavad prügiauto järel ning pesevad tühjad prügikastid. See lahendus ei ole sobiv masina ja tööjõu hinna poolest. Viimasena on ka ehitatud eraldi süsteemid väikse järelhaagise peale ning hinna poolest on need kõige mõistlikumad, kuid praktilisuse ja töökindluse poolest jäävad alla varem väljatoodud lahendustele. Sellest on ka tekkinud tellija poolne huvi arendada pesusüsteem, mis on loodud vältides teiste tehtud vigu. Hea töökindluse kõrval saab olema suurim erinevus ka toote hinnas, mis saab olema odavam seni turul olevatest lahendustest.

3 Arendusmeetodid

Töö on valminud koostöös Jürgen Kolli lõputööga „Prügikastide automaatse pesusüsteemi juhtiva loogikakontrolleri valik ja sellele loodava tarkvara testimine“ ning kogu projekti raames on olnud lisaeesmärgiks õppida kasutama ja järgida agiilse arendusmeetodi põhimõtteid. Projekti ja meetodi kasutamisega sai alustatud Arvutisüsteemide projekti aine raames. Alustades lõputöö sprinte kaaluti kolme meetodit ning millisega oleks mõistlikum jätkata parima tulemuse saavutamiseks.

3.1 Agiilne arendusmeetod

Agiilne arendusmeetod on kasutusel IT-valdkonnas kui töö jaotamise ja planeerimise viis, kus iga kindla perioodi tagant, reeglina kahe nädalaga saadakse valmis mingi osa või versioon kogu projektist, mis kantakse ka osapooltele ette tagasiside saamiseks [7]. Neid tsükleid nimetatakse sprintideks. Reeglina koosneb arendusmeetod kuuest tegevusest. Esimesena neist vaadatakse ja kinnitatakse üle projekti nõuded ning seejärel planeeritakse edasine sprint. Kolmandaks alustatakse uue info põhjal arendust, mis tehakse üheskoos testimisega. Viimaseks avaldatakse tehtud töö kliendile üle vaatamiseks, mille järel algab protsess uuesti [8] kuni projekti valmimiseni (Joonis 1).



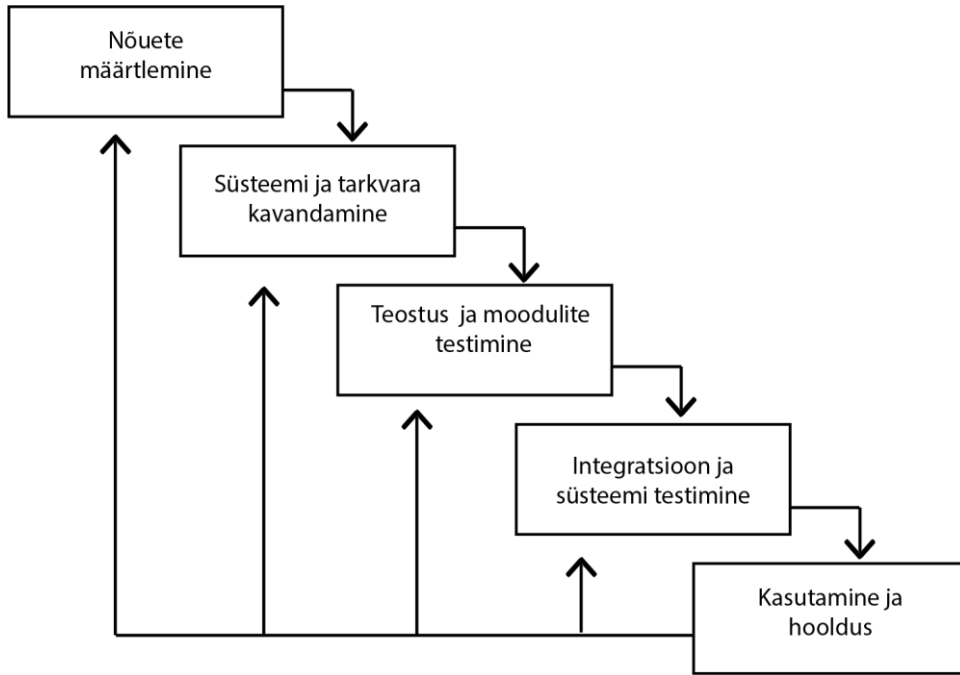
Joonis 1. Agiilse arendus meetodi töö tsükel [8]

3.2 Kanban arendusmeetod

Kanban arendusmeetod on tegelikult agiilse arendusmeetodi üks alamliike, kuid siiski selgelt eristatav [9]. See viis keskendub ülesannete visuaalsele planeerimisel, mis aitab kaasa paremale arusaamisele kokkulepetest. Kanban meetod ei käi alati läbi agiilse meetodi puhul välja toodud etappidest, vaid valib vastavalt vajadusele kiireima tee lõpp tooteni. Tihti peale küsitakse tagasiside jooksvalt ka arenduse keskel, mis võib hoida aega hiljem kokku. Seeläbi ei planeerita ka töö alguses põhjalikult, vaid seda tehakse töö käigus.

3.3 Kosk arendusmeetod

Levinuim ja vanim arendusmeetod IT sektoris on koskmudel [7]. Koskmudel põhineb rangelt erinevatel etappidel ning järgmist etappi ei alustata enne kui viimasega on lõpetatud. Iga etappi lõpus tehtud töö dokumenteeritakse. Koskmudeli üks algtingimusi on, et projekti nõuded on selgelt määratletud ega muutu peale projekti alustamist. Põhjus selleks on lihtne- kui projekti nõuded muutuvad tuleb kogu projekti algusest alustada. Eeliseks sellel meetodil on, et kogu projekti ei pea tegema sama arendustiim ehk näiteks kui disain on valmis ning dokumenteeritud võib anda selle sisendiks teisele ettevõttele kes teeb järgmise etapi [10]. Samal põhimõttel saab toimida ka firma siseselt, kus on oluline inimressursside jaotamine ehk tippspetsialisti ei hoita kinni kogu projekti perioodiks, vaid ainult soovitud etapiks. Koskmudel koosneb viiest osast nagu joonisel (joonis 2) on näha. Esimeses etapis pannakse paika projekti nõuded nõnda, et neid enam hiljem ei muudetakse. Seejärel liiguvad need nõuded IT arhitektide töörühma, kus tarkvara kavandatakse ja saadetakse edasi arendusse. Peale arendust toimub loodud tarkvara integratsioon ja testimine. Sealt edasi läheb tarkvara kliendile kasutusse, millega tavaliselt käib kaasas ka hooldus, mis omakorda võib tähendada muudatusi, mistahes eelnevates etappides.



Joonis 2. Kosk meetoditöö tsükkel [10]

4 Valitud arendusmeetod

Selle projekti jaoks valiti agiilne arendusmeetod, sest see tagab, et projekt saab pidevalt tagasisidet ning muudatused saab sisse viia kergemalt, kui seda oleks teha ainult projekti lõppfaasis. Kuna tellija poolt oli algusest peale selge, et algtingimused muutuvad nii kaua kuni ei ole olemas füüsilisi seadmeid, siis sobis see kokku arendus meetodi valikuga. Kanban arendusmeetod oli samuti valikus, kuid korrapäratu struktuuri tõttu oli meeskonnale lihtsam hoida projektil järge valitud meetodit kasutades. Samuti oli valik mõistlik, sest meeskond oli ainult kaheliikmeline lisaks tellijale. See tähendas, et kommunikatsioon toimus hästi nii meeskonna siseselt kui ka tellijaga ning muudatused sai kiiresti sisse viia. Järgides agiilset meetodit olid tellijaga kohtumised iga kahe nädala tagant kas kõne või kohapeal kohtumise vormis. Kohtumistel esitati küsimusi tekkinud kitsaskohtade osas ning esitleti seni tehtud tööd, millele sai antud tagasiside tellija poolt edasisteks parandusteks ja arenduseks. Iga kohtumise eesmärk oli ka paika saada muutunud nõuete järgi juhtalgoritm, mille aluselt hiljem programm järgi teha. Kahe töö siseselt olid planeeritud eraldi iganädalased koosolekud, mis olid ka tähtsajad kokkulepitud tööde esitlemiseks kui ka tagasiside saamiseks teineteiselt. Käesoleva töö raames oli põhieesmärgiks luua juhtprogramm ning teise töö raames seda testida ning anda tagasisidet, mida on vaja veel parandada ja mis on programmi funktsionaalsed puudused. Selle alusel sai järgmiseks kohtumiseks tehtud parandused parimal võimalikul viisil, sest alati ei olnud võimalik soovitud nõudmisi sisse viia. Tehtud soovitused ja parandused pandi kirja ka logina ning salvestati eraldi programmi versioonid, et oleks olemas nii tellijale kui meile ülevaade tehtust või kui on vaja programm taastada varasemas versioonis. Kokku toimus lõputöö projekti raames üheksa sprinti.

5 Projekti nõuded

Projekti põhisiht oli küll alati paigas, kuid detailid ja projekti maht oli pidevas muutuses. Tellija poolelt on nõuded muutunud nii eelarve kui kogemuse tõusmisest olenevalt. Erinevaid variante on üldises vaates olnud kuus ning kõigile nendele sai tehtud programmi versioon või juhtalgoritm.

5.1 Versioon 1

Projekti algnõuded sai paika pandud 2021 aasta septembri kuus. Tehniliste nõuetena ja funktsioonidena pandi eesmärgiks teha kogu prügiauto prügi pressi süsteem ja sellele lisanduva pesusüsteem kokku ühele controllerile. See oleks taganud paremad juhtimisvõimalused ning vähendanuks oluliselt juhi poolse sisendi vajadust, mis omakorda andnuks suurema ajavõidu. Küll aga oli prügiauto prügi pressi juhtimine ise ehitatud PCB peale, mille programmile ligipääsu polnud, sest puudus selleks vajalik riist- ja tarkavara. Seetõttu oleks ka prügi pressi enda funktsioone keerukas olnud ümber teha kuna puudus detailne ülevaade. Kuna tellijal on eesmärk kasutada ka kogu loodud süsteem teiste prügi autode prügi presside peal, siis polnud mõistlik selle lähenemisega jätkata, sest prügi pressid ei ole standardsed ning tähendanuks vajadust luua igale mudelile eri versioon.

5.2 Versioon 2

Tellijaga arutades esimese variandi puudusi, sai otsustatud muuta algtingimusi, uus eesmärk oli luua eraldi seisev süsteem, mis on ühendatud juba olemas oleva PCB-iga. Uus programm peab juhtima automaatselt vastavalt vajadusele ühte või kahte pesupead soovitud suundades, puhta vee pumpa vastavalt valitud pesuperioodile ja kontrollima eri liikuvaid prügi pressi osi ohutuse tagamiseks läbi sisendandurite, mis tulevad controllerile. Programm peab suutma täita ilma lülitite algasendisse naasmata ehk vool süsteemis on pidev. Ühendus PCB-iga tundus endiselt liialt keeruline, sest polnud leida seadistusjuhendit ega muud infot mille alusel ühendused luua. Samuti polnud lahendatud musta vee ära vool. Ühendus PCB-iga on ka eri prügi pressi mudelitel erinev ehk süsteemi tulnuks kohandada iga mudeli jaoks. Kuna süsteem tuleb miinus kraadide saabudes autolt eemaldada oli ka see üks põhjustest miks ikkagi kahte taheti eraldi hoida.

5.3 Versioon 3

Viimase variandi puudusi arvesse võttes otsustati teha kaks süsteemi täielikult teineteisest sõltumatult nõnda, et see oleks kohaldatav mis tahes prügiautole prügipressile. Selle variandi puhul oli ka lisandväärtuseks asjaolu, et autokast töötab ka talvel, kuid siis tuleb pesusüsteem eemaldada, et keskkonna mõjud seda ära ei lõhuks. Eraldi seisva süsteemina on seda lihtsam ära võtta ja lisada hooaja vahetudes. Selles versioonis soovis tellija lisada musta vee äravoolupumba, mida juhitakse läbi kontrolleri, et vesi liiguks ära prügipressist reovee paaki.

5.4 Versioon 4

Viimase süsteemi esitlemisel otsustas tellija töökindluse tõttu, et pesupead liiguksid ainult töö ja sõidu asendisse ning mitte prügikasti sees. Pesupeade liikumist prügikasti sees võimaldava seadme purunemisel oleks parandus protsess liiga aeglane. See tegi programmi lihtsamaks kontroll funktsioonide osas ning ka lihtsamini testitavaks. Samuti otsustati selles versioonis lisada kontrollandurid veepaagile ehk oleks nii hoiatussignaal kui ka programmi katkestus, kui vesi otsa saab. Vastasel juhul oleks vee lõppemine põhjustanud pumba katki minemise. Selles versioonis vahetati välja ka kontrolleri, sest varem valitud polnud enam laialdaselt kätte saadav ning programmeerimistarkvara oli küllaltki aegunud kiire arendus jaoks.

5.5 Versioon 5

Tellija poolne praktika on varasemalt näidanud, et induktiivanduritel, mis ise on küll parima hinna ja töökindluse suhtega, on ka ette tulnud riknemisi ehk kas midagi satub anduri vahele või mistahes põhjusel lõpetab see oma töö. On ette tulnud olukordi, kus kogu prügiauto prügipressi pole võimalik kasutada, sest üks induktiivandur ei tööta, mis tähendab ka terve tööpäeva raisku minemist. Selleks otsustati üheskoos lisada pesusüsteemile funktsioon, mis kõrvaldaks selle vea vähemalt loodaval süsteemil „override“ funktsiooniga. Nii nimetatud „override“ on peidetud juhtpuldil ehk seda saab kasutada ainult äärmisel juhul, et tagada ohutus ning auto juht ei hakkaks seda väärkasutama.

5.6 Versioon 6

Eelmises versioonis lisatud „override“ funktsioon otsustati viimases versioonis ära võtta, sest vähendati kontrollitavaid sisendeid millest mööda minna. Tulenevalt sellest otsustati see asendada lihtsa LED pirnidel põhineva ekraaniga, millel kuvatakse andurite kontakte. See aitab purunemise korral määrata koheselt milline andur on katki ning see kohapeal asendada. Alles jäi kaks pesu varianti ning valik mitu pesupead kasutada. Otsustati ümber teha reovee pumba süsteem, kus asendati klassikaline pump hoopis suruõhul põhineva äravooluga. Nimelt sõidab programmi lõpedes võre ette klapp, mis piirab suruõhu väljapääsu ning vesi surutakse otse reovee paaki. Selline viis välistab seadme rikke, mis pumba puhul oli võimalik. Tulenevalt sellest juhib programm nii klappi kui õhku. Samas süsteemi tuli lisandurina ujuvandur, mis aktiveerub kui vesi tõuseb. Pesupeade liikumine jäi samaks.

6 Kontroller

Süsteemi juhtimiseks kasutatakse Siemensi *LOGO! 24CE* [2]. Logo kontroller on loodud väiksemate automatiseerimis projektide jaoks ning selle jõudluse ja hinna suhe sobib projekti nõuetega. Esialgu oli plaanis kasutada ilma ekraanita kontrollerit, sest asukoha tõttu pole võimalik seda kasutada. Ekraan võib prototüübi seadistusel olla kasulik, kui see pole veel paigaldatud, sest muudatusi programmis on võimalik teha ka läbi sisendnuppude kontrolleril [11]. Tarneraskuste tõttu oli saadaval ainult ekraaniga variant ning esimese prototüübi loomise jaoks sobib ka see. Valitud kontrollerile ei ole küll protsessormoodulil piisav arv sisendeid ega väljundeid, kuid projekti nõuete pidevas muutumises oli see sobiv, sest moodulite lisamine on võimalik kontrollerile [12]. Programmi seisukohalt on olulise tähtsusega veel induktiivandurid, mis annavad sisendsignaali programmi tööks. Nende anduritena kasutatakse „XS2 M18“ *Telemecanique Sensors* [13], mille osas on tellijal kogemus. Kontrolleri valikut põhjendatakse lähemalt bakalaureusetöös „Prügikastide automaatse pesusüsteemi juhtiva loogikakontrolleri valik ja sellele loodava tarkvara testimine“ [3].

7 Programm

7.1 Programmeerimistarkavara tutvustus

Kontrolleri programmeerimiseks kasutati tarkavara nimega *Logo! Soft comfort V8* [1]. See on Siemensi poolt toodetud spetsiaalselt valitud mudeli programmeerimiseks ning on loodud nagu ka kontroller ise, väiksemate süsteemide automatiseerimiseks. Tarkvaras on võimalik programmeerida nii funktsioonplokkide abil kui ka redeldiagrammis ning selles töös kasutati esimest võimalust selle omaduste poolest. Keskkonnas on võimalik ka tehtut simuleerida ja kontrollida, nii keskkonnas endas kui ka läbi võrgu [14]. Tarkvara testimisvõimekust kasutatakse töö raames küll vähe, sest kõik sisendid on digitaalsed, kuid on olemas võimekus ka analoogsignaalide puhul simuleerida reaalse väärtustega nagu temperatuur ja muud ilmastikutingimused. Kokku on tarkvaras 63 erineva funktsionaalsusega plokki, mille abil programm luua, sellest töös kasutati enim erinevaid loogika ja taimerite plokkide [15].

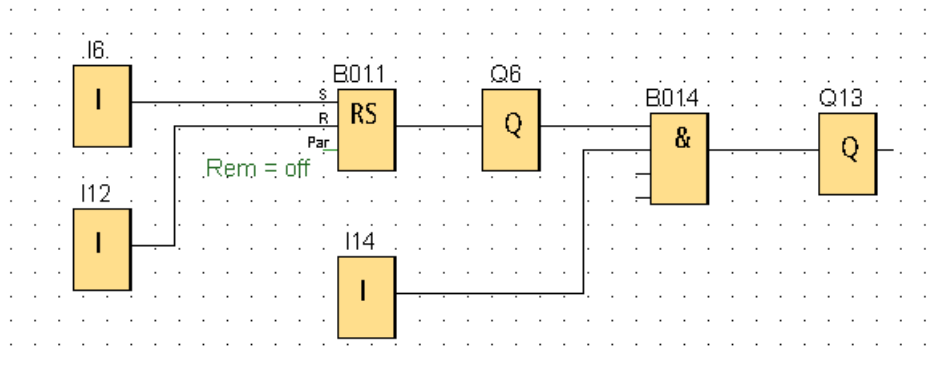
7.2 Programmi esitlusvormid

Kontrolleri programmeerimiseks on mitmeid erinevaid meetodeid, neist enim levinumad on redel- ja funktsioonplokk diagramm ning struktureeritud teksti põhine programm [16]. *Logo! Soft comfort V8* [1] võimaldab programmeerida redel- ja funktsioonplokk diagrammi. Neid standardprogrammi esitus vorme arendab tööstusautomaatikas tegutsev organisatsioon PLCOpen [17], mille eesmärk on parandada efektiivsust erinevate avatud standardite loomisega ning sellega tuua süsteemide maksumused alla. PLCOpen tegutseb juba aastast 1992 ehk standardid on juba ennast tõestanud. Põhiline standard mida järgivad programmitavad loogikakontrollerid on IEC 61131-3 [18]. Selle järgi on ka kasutatud ja loodud põhilised kontrolleri programmeerimismeetodid ja nende süntaks. Seda standardit järgivad ka kõik allpool kirjeldatud meetodid.

7.2.1 FBD

Function Block Diagram (FBD) ehk funktsioonplokk diagramm, mida kasutati ka selle töö jaoks, põhineb nii nagu nimi ka ütleb erinevatel plokkidel, millel igal ühel on oma funktsioon. Selle meetodi eeliseks on teksti põhise programmeerimise puudumine ehk kompileerimine ja vea leidmine on lihtsam kirjavigade puudumise tõttu (joonis 3).

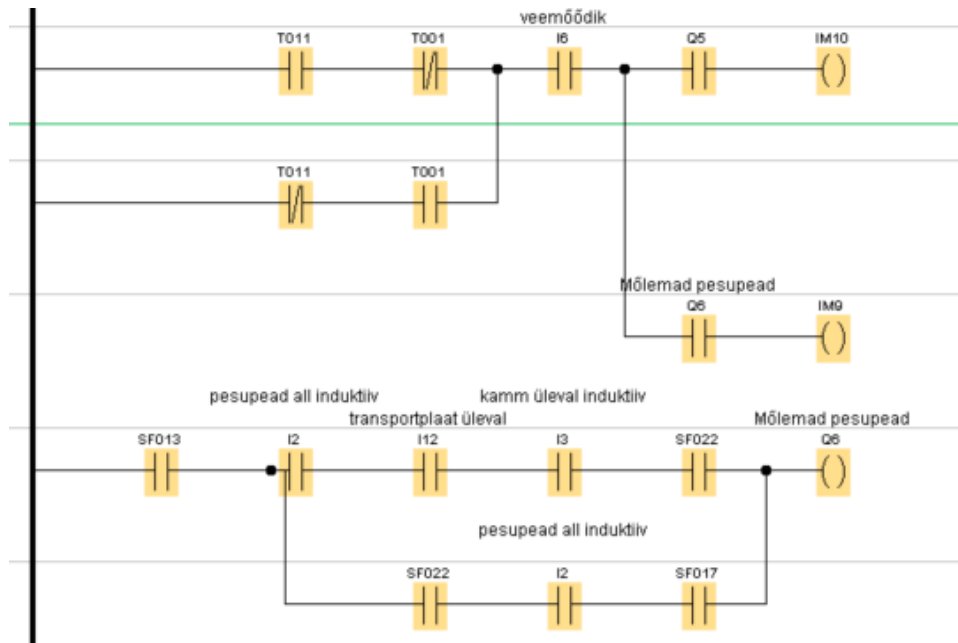
Tarkvara võimaldab liigutada plokkide, mis annab võimaluse grupeerida funktsioone, mis teeb visuaalselt programmi kergesti arusaadavaks. Samuti on kergem jälgida signaali liikumist ja leida ning parandada vigu. Vajadusel saab konverteerida ka lihtsalt redeldiagrammi vaatesse kasutades programmeerimise tarkvarasse sisseehitatud funktsionaalsust. Eelnevalt on programmeeritud taimerid, kõik peamised loogika tehted, sisend-väljund signaalid ja muud lihtsamad funktsioonid sõltuvalt tarkvarast.



Joonis 3. Funktsioon plokki diagramm

7.2.2 RLL

Relay Ladder Logic (RLL) ehk redeldiagramm on enim kasutatud meetod kontrolleri programmeerimiseks. Meetod põhineb relee loogikal ning seetõttu osutus see ka populaarseks, sest ei vajanud põhjalikku ümber õpet programmeerijale [19]. Struktuuri poolest on RLL rangema üles ehitusega ehk formaat on redeli põhine (joonis 4). Ka sellega kasutatakse tarkvara poolt antud funktsioon plokkide, kuid üles ehitusest tingituna on erinevaid valikuid vähem. Formaadi poolest on redeldiagrammi keerulisem ka mõista, sest puudub kogu projekti üldine vaade ning on raske ette kujutada erinevaid ühendusi. Samade funktsioonide teostamine RLL meetodil võtab mahu poolest rohkem aega kui FBD-s. *Logo! Soft comfort V8* [1] võimaldab konverteerimist kahe meetodi vahel, mis teeb arenduse lihtsamaks.



Joonis 4. Redel diagramm

7.2.3 ST

Structured Text (ST) ehk struktureeritud teksti põhine programmeerimise meetod on sarnane *Python* või *C* keeles programmeerimisele. Sarnaselt laialt levinud programmeerimiskeeltega on read eraldatud semikoolonitega, tühik ei oma tähtsust ning suurel ja väiksel tähel pole vahet [20]. Ülejäänud tehte märgid kehtivad nagu ka teistes keeltes (joonis 5). Kuigi see meetod vajab juba head väljaõpet, on seda hea kasutada keeruliste matemaatiliste võrrandite kirjutamiseks, mis saavad sisendina suure hulga infot. Plokkidel põhinevad meetodid on üldiselt paremad kasutamiseks väiksemate süsteemide puhul, süntaksi põhine programmeerimine on parem suurte süsteemi puhul, kus on juba oluline ka andmete maht [21]. Tekstipõhine programmeerimine annab suurema paindlikkuse, kui juba tarkvara poolt ette kirjutatud plokid. Samuti lisandväärtuseks sellele meetodile on võimalus seda kasutada üheskoos varasemalt kirjeldatud meetoditega.

```

PROGRAM
VAR
A: BOOL;
END_VAR
A: = TRUE;
REPEAT
A: = FALSE;
UNTIL A: = FALSE;
END_REPEAT;
END_PROGRAM;

```

Example 1

```

PROGRAM
IF tank.temp>300 THEN
pump.fast: =1;
pump.slow: =0;
pump.off: =0;
ELSEIF tank.temp>200 THEN
pump.fast: =0;
pump.slow: =1;
ELSE
pump.slow: =0;
pump.off: =1;
ENDIF;
END_PROGRAM;

```

Example 2

Joonis 5. Struktureeritud teksti põhine programmi näide [16]

7.3 Valitud meetod

FBD meetod sobis kõige paremini selle töö teostamiseks oma visuaalsete omaduste poolest. Plokkide abil on lihtne näha eri seoseid ning jälgida süsteemi simuleerimisel signaali liikumist, mis andis ka redeldiagrammi ees eelise. Redeldiagrammi puudus, näha üldist vaadet, sai ka siin kohal otsustavaks. Kuna antud süsteem ei pea teostama keerulisi matemaatikatehteid, vaid teostama palju lihtsaid loogika tehteid siis FBD meetod sobis selleks kõige paremini. Valitud keskkonnas puudus ka struktureeritud teksti programmeerimise võimekus, mis välistas viimase meetodi kasutamise.

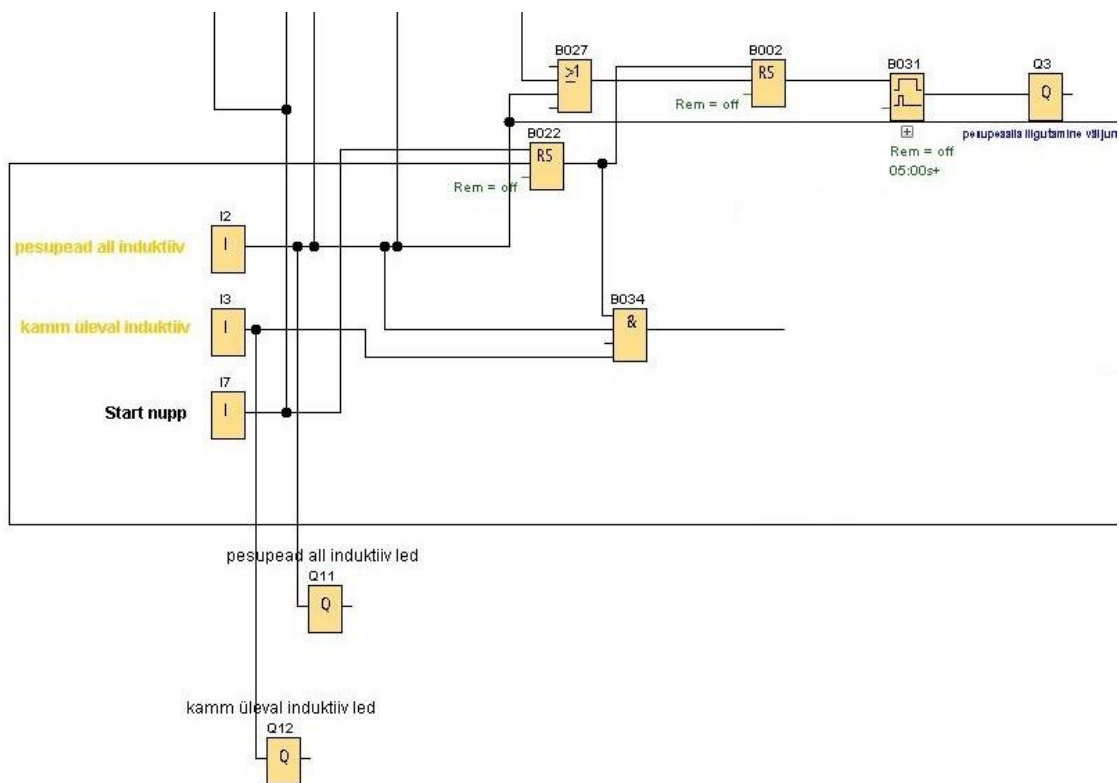
7.4 Programm

Funktsioonplokkidena on kasutatud erinevaid loogika, taimer, sisend ja väljund plokkide. Loogika plokkidest on kasutatud *AND*, *NAND*, *XOR* ja *OR* plokkide. Eri tüüpe taimereid on kolm mis on oma signaali läbilaske poolest erinevad. Programm juhib pesupäid tööning sõiduasendisse, musta vee süsteemi, survepesurit ja veevoolu kahele või ühele pesupeale (lisa 2).

7.4.1 Esimene osa

Programm koosneb viiest erinevast osast. Taimerid saab seadistada alles siis, kui süsteemi on praktikas võimalik katsetada, kuid funktsionaalsus sellest ei muutu. Programmi esimene osa (joonis 7) algab peale start nupu I7 vajutamist ning lüüti puhul on tegu hetkelise signaaliga, mille puhul tuli kasutada pinget säilitavat plokk B022. Peale selle

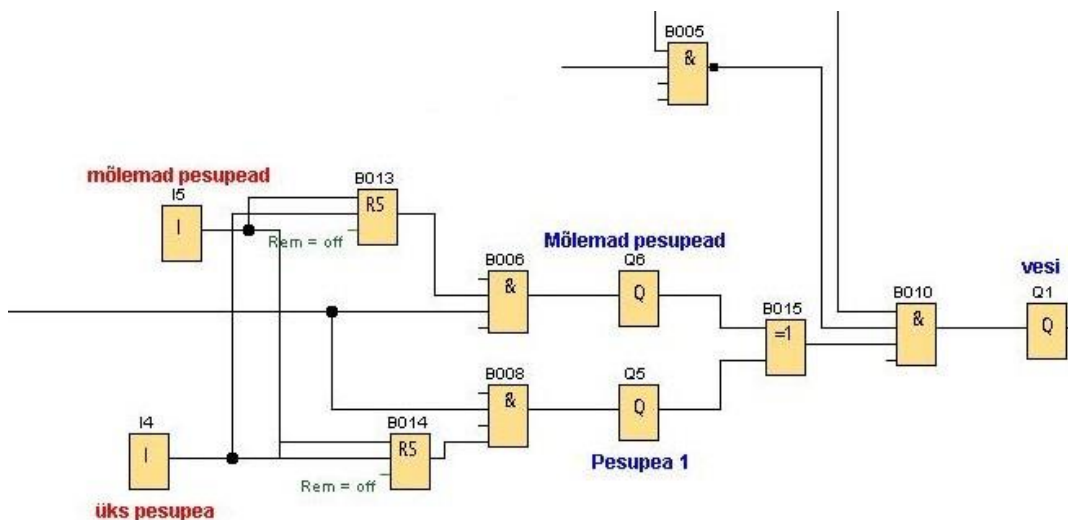
sisendi andmist kontrollitakse kammi I3 ja pesupea I2 asendit. Kahe väljundi kontrollimiseks on väljundid LED tuled kujul juht puldil ehk Q11 ja Q12. Kui pesupea ei ole õiges kohas, liigutatakse see alla väljundis Q3. Kui kontrollid on kõik täidetud ehk plokis B034 on loogika tehe täidetud, liigub vool edasi.



Joonis 6. Sisendite kontroll

7.4.2 Teine osa

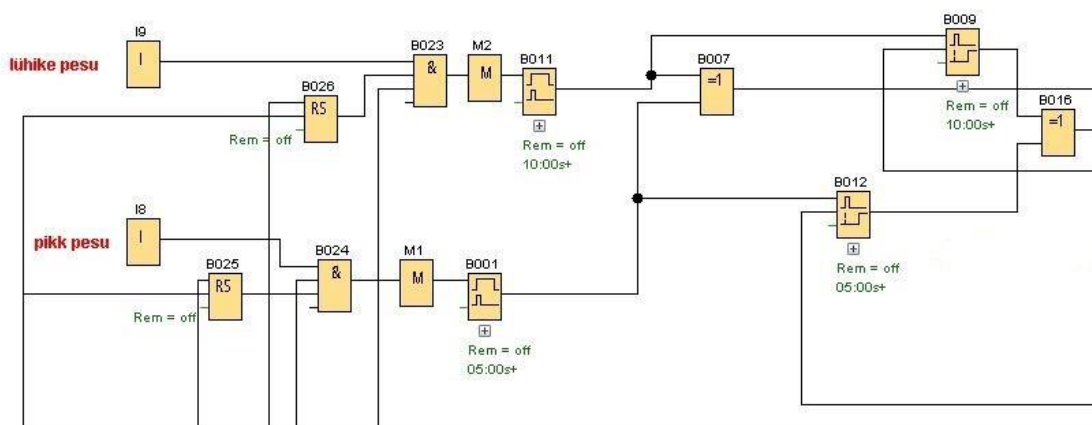
Programmi teises osas (Joonis 8) valitakse töötavad pesupead. Tegemist on pöördlülitiga, kus on võimalus valida kas soovitakse kasutada ühte I4 või mõlemat pesupead I5. Praktikas sõltub see prügikastide arvust või suuruselt. Kui valik on tehtud liigub vool edasi plokki B010, kus oodatakse signaali pesu pikkuse aega.



Joonis 7. Pesupeade valik

7.4.3 Kolmas osa

Kolmandas osas (joonis 9) valitakse pesuprogrammi pikkus. I9 on pikk ja I8 on lühike pesu ning ka need on juhtpuldil pöördlülitina. Kui valik on tehtud liigub signaal läbi B007 plokki B010, mis annab signaali survepesurile ning algab pesu. Taimerid B009 ja B012 on taimerid, mis töötavad vastassuunaliselt, ehk need ei lase enne loogikavoolu läbi kui pesu on lõppenud.

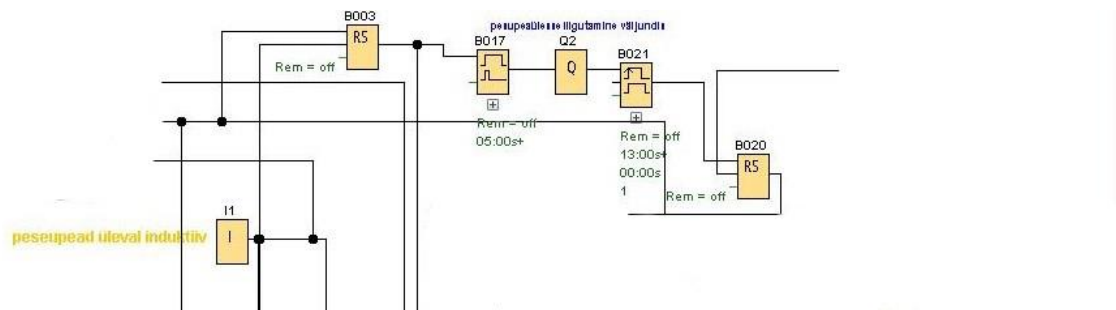


Joonis 8. Pesu pikkuse valimine

7.4.4 Neljas osa

Neljandas osas (joonis 10) liigutatakse peale pesu lõppemist pesupea tagasi sõiduasendisse väljundiga Q2 kuni I1 induktiivandur annab signaali. B017 on

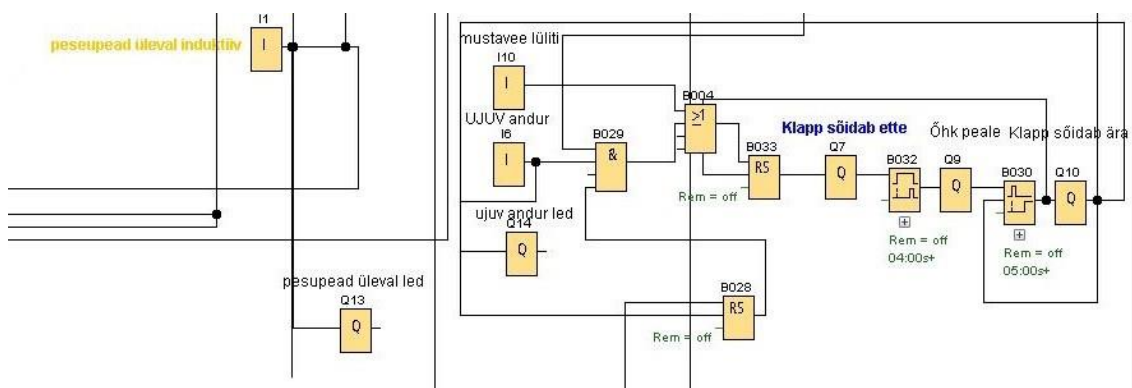
kontrolltaimer juhiks kui induktiivandur ei tööta. B021 annab signaali reovee äravoolu süsteemile.



Joonis 9. Pesu pead sõidu asendisse liigutamine

7.4.5 Viies osa

Viies ja viimane osa käsitleb reovee äravoolu (joonis 11). Lisaks B017 signaalile ootab süsteem ka ujuva anduri signaali, mille järel sõidab ette klapp Q7. Pärast klapi ette liikumist surutakse õhuga vesi reovee paaki väljundiga Q9. Selle järel liigutatakse klapp jälle pealt ära väljundiga Q10. Q10 ja I1 annab ka nii-öelda taaskäivituse signaali kogu süsteemile, millega paneb süsteem ennast tagasi algasendisse. Lisaks sellele on võimalik reovee äravoolu juhtida ka eraldi nupu vajutamisega ehk I10.



Joonis 10. Reovee süsteemi juhtimine

7.5 Edasiarendusvõimalused

Edasiarendusvõimalustena nähakse süsteemi automatiseerituse taseme tõstmist eesmärgiga tõsta ühe pesuperioodi kiirust.

Üks variant seda teha on võttes ära käsitsi valikust kui mitu pesupead töötaks ehk see asendataks anduritega, mis tuvastavad prügikastide arvu ise. Lisaandur tuleb panna ka tuvastamiseks, millal on prügikast tühjendatud ehk pesu hakkab koheselt peale kallamise osa. Sellega kaasneb jällegi risk, et osa prügist on prügikasti kinni jäänud ega tule peale esmast kallamist ära, kuid pesu hakkab juba peale ning on asjata. See on ka üks põhjustest, miks seda varianti veel tehtud ei ole. Teisena lisades andurid kammile on praktika näidanud, et töökindlus pole kuigi hea ehk purunemise puhul tuleb kasutada “override”, kuid see juba teeb omakorda kogu selle funktsiooni kasutuks.

Üks võimalus optimeerida programmi, on kasutades tarkvarasse sisse ehitatud plokki, mis kogub andmeid praktikas soovitud sisendite ja väljundite kohta [22]. Selle lisamine programmi aitaks koguda andmeid, et kui kaua läheb aega erinevate tegevuste vahel, kui tihti on pidanud autojuht teist korda prügikaste pesema ja sama ka reovee äravoolu süsteemi kasutuse kohta. Andmed on tarkvara abil lihtne kanda Excelisse või muude soovitud andmetöötlusplatvormile nende analüüsimiseks [23]. See annab eelkõige võimaluse seadistada taimereid, kuid aitab välja tuua kitsaskohad, mida saaks lahendada uute funktsioonidega programmis või riistvaraga päris süsteemil.

8 Kasutus- ja seadistusjuhend

8.1 Seadistusjuhend

Juhend on kirja pandud tabeli kujul, et süsteemi kokkupanekuks ei oleks vaja kõrvale selle programmeerijat. Tabelis (tabel 1) on välja toodud eraldi sisendid ja väljundid koos selgitustega, mis tuleb ühendada kontrollerial vastavas kontakti.

Tabel 1. Seadistusjuhend

Sisendid	
I1	Induktiivandur – pesupead sõidu asendis
I2	Induktiivandur – pesupead töö asendis
I3	Induktiivandur – kamm all asendis
I4	Pöördlüüti – üks pesupea
I5	Pöördlüüti – kaks pesupea
I6	Ujuv andur – reovee süsteem
I7	Start lüüti
I8	Pikk pesu režiim
I9	Lühike pesu režiim
I10	Reovee süsteemi eraldi lüüti
Väljundid	
Q1	Survepesur
Q2	Pesupeade liigutamine sõidu asendisse
Q3	Pesupeade liigutamine töö asendisse
Q5	Vee vool läbib ühte pesupead
Q6	Vee vool läbib mõlemat pesupead
Q7	Reovee süsteemis sõidab klapp ette
Q9	Reovee süsteemis tuleb suru õhk peale
Q10	Reovee süsteemis sõidab klapp tagasi
Q11	Induktiivandur – pesupead töö asendis LED tuli
Q12	Induktiivandur – kamm all asendis LED tuli
Q13	Induktiivandur – pesupead sõidu asendis LED tuli
Q14	Ujuv andur – reovee süsteem LED tuli

8.2 Kasutusjuhend

Juhend on mõeldud selgitamaks, kuidas süsteem programmi abil praktikas töötab. Enne pesu algust peab olema prügikast viidud pesu asendisse. Seejärel tuleb juhtpuldil teha soovitud valikud pesu pikkuse ja pestavate prügikastide osas. Mõlema puhul on tegu pöördlülititega. Peale valikute tegemist tuleb vajutada start nuppu, mille järel sõidavad pesupead töö asendisse ning algab pesu. Kui pesu ei alga tuleb üle kontrollida LED indikaatorite osas kas „pesupead töö asendis“ ja „kamm üleval“ induktiivanduri LED tuled põlevad. Kui LED indikaatorid ei põle, tuleb üle kontrollida vastavad induktiivandurid visuaalselt, et vajadusel eemaldada prügi mis võib sattuda induktiivanduri ja metalli vahele. Kui visuaalsel vaatlusel viga ei tuvasta ja induktiivi enda tuli ka ei põle, tuleb induktiivandur välja vahetada. Kui pesu on hakanud tööle tuleb oodata selle lõpuni, kuni pesupead on läinud tagasi sõidu asendisse. Seejärel tohib prügikasti alla lasta ja ära võtta. Kuigi prügikast on juba eemaldatud võib käia veel reovee ära suunamine. Kui reovett on endiselt alles saab reovee nuppu kasutades aktiveerida see süsteemi osa eraldi. Kui reovee süsteemilt on klapp pealt sõitnud tähendab see programmi ringi lõppu. Uut tsüklit saab alustada ka ainult start nuppu uuesti vajutades, kui eelnevad valikud soovitakse jätta samaks. Kui pesu ei alga tuleb kontrollida, kas pesupead üleval induktiivandur on saanud vahepeal kontakti. Hädaolukorra puhul on võimalik kasutada prügipressi stopp nuppu mis peatab kogu protsessi.

9 Kokkuvõte

Käesoleva töö eesmärgiks oli luua juhtprogramm, mis juhib prügiauto prügi pressi küljes olevat pesusüsteemi. Programm juhib seadmeid läbi Siemensi kontrolleri ja kontrolleri programmi koostamise arendustarkvaraks kasutati Siemensi *Logo! Soft comfort V8*. Projekti arendusmeetodiks kasutati agiilset arendusmeetodit, mis sobis oma omaduste poolest kõige paremini projekti nõuete ning eesmärgiga. Programmeerimiseks kasutati funktsioonplokkidel põhinevat viisi, sest oli enda visuaalsete ja teiste omaduste poolest kõige sobivam töö teostamiseks. Programm koosneb viiest peamisest osast. Need viis osa on jaotatud vastavalt peamistele protsessidele mida juhitakse. Programmil on 10 sisendit ja 13 väljundit, mida juhitakse erinevate taimerite ja loogika tehete abil.

Süsteemi osade seadmete saabumise hilinemine takistas juhtprogrammi reaalsetel seadmetel testimist, aga see ei mõjutanud programmi funktsionaalsuse tagamist. Hiljem vajavad reaalsusega täpsemat sobitamist vaid programmis kasutatud taimerid. Programm on antud üle tellijale ning simuleerimisel vastas see püstitatud tingimustele ja sellega said kõik eesmärgid täidetud.

Peale esimese prototüübi valmimist ja testimist on võimalik süsteemi edasi arendada lisades programmi andmete kogumise funktsioon. See annab võimalusel koguda infot süsteemi kasutuse kohta, mille alusel on võimalik seda optimeerida. Teise võimalusena saab lisada kammile kaks induktiivandurit ning programmis vastav funktsioon, mille abil vähendatakse kasutaja poolset sisendi vajadust.

Kasutatud kirjandus

- [1] LOGO! Software. Version 8, Siemens. [Online]. Loetud aadressil: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-software.html> Kasutatud: 10.02.2022.
- [2] Siemens, 6ED1052-1CC08-0BA1, 2020, [Online]. Loetud aadressil: <https://www.industry-mobile-support.siemens-info.com/#/en/product/6ED1052-1CC08-0BA1> Kasutatud: 17.03.2022.
- [3] J. Koll, „Prügikastide automaatse pesusüsteemi juhtiva loogikakontrolleri valik ja sellele loodava tarkvara testimine“, [Bakalaureusetöö], Infotehnoloogia teaduskond, TalTech, Tallinn, Eesti, 2022.
- [4] Ekokymppi, Jätehuoltomääräykset, 2018, [Online]. Loetud aadressil: <https://www.ekokymppi.fi/tietopankki/jatehuoltomaaraykset.html> Kasutatud: 27.01.2022.
- [5] Tekninen erittely & käyttöohje, kasutusjuhend, Keox Oy, Soome, 2020. Kasutatud: 01.10.2021.
- [6] PESUAUTO 30 HUOLTO-OHJEITA, hooldusjuhend, JOAB Oy, Soome, 2020. Kasutatud: 13.10.2021.
- [7] I. Majas, „Tarkvaraarenduse meetodika valiku protsess“, [Bakalaureusetöö], Informaatika teaduskond, Tartu Ülikool, Tartu, Eesti, 2016. [Online]. Loetud aadressil: <https://dspace.ut.ee/handle/10062/56239>
- [8] Medium, Agile Software Development — A game changer of software development process, 2021, [Online]. Loetud aadressil: <https://mzuhdiz.medium.com/agile-software-development-a-game-changer-of-software-development-process-d52933246c28> Kasutatud: 20.01.2022.
- [9] GeeksforGeeks, Difference between Agile and Kanban, 2020, [Online]. Loetud aadressil: <https://www.geeksforgeeks.org/difference-between-agile-and-kanban/> Kasutatud: 03.02.2022.
- [10] EUCIP, Koskmudel, 2011, [Online]. Loetud aadressil: <https://eopearhiiv.edu.ee/e-kursused/eucip/arendus/index.html> Kasutatud: 21.01.2022.
- [11] Electrical Chile, Configure and Monitor from Display Screen TDE Logo!, 2017, [Online]. Loetud aadressil: <https://www.electricalchile.cl/plclogosiemens7en.php> Kasutatud: 19.03.2022.
- [12] Siemens, LOGO! Expansion Modules, 2020, [Online]. Loetud aadressil: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-expansion-modules.html> Kasutatud: 05.03.2022.

- [13] Inductive proximity sensors, kataloog, Schneider Electric Industries SAS., Prantsusmaa, 2022. [Online]. Loetud aadressil: https://download.schneider-electric.com/files?p_enDocType=Catalog&p_File_Name=Catalogue+Inductive+proximity+sensors+XS+range+Easy+Series.pdf&p_Doc_Ref=DIA4ED2180501EN Kasutatud: 14.12.2021.
- [14] LOGO! Soft Comfort Online Help, kasutusjuhend, Siemenes AG, Saksamaa, 2022. [Online]. Loetud aadressil: https://cache.industry.siemens.com/dl/files/807/100782807/att_924632/v1/Help_en-US_en-US.pdf Kasutatud: 11.02.2022.
- [15] LOGO SOFT COMFORT V8 TUTORIALS. 2018. [Online]. Loetud aadressil: <https://plc-courses.com/logo-soft-comfort-v8-tutorials/> Kasutatud: 03.04.2022.
- [16] Ezautomation, PLC Ladder Logic vs. Everything Else, 2018, [Online]. Loetud aadressil: <https://www.ezautomation.net/industry-articles/PLC-Ladder-Logic-vs-Everything-Else-Part-2.htm> Kasutatud: 17.02.2022.
- [17] Wikipeedia, PLCOpen, 2017, [Online]. Loetud aadressil: <https://en.wikipedia.org/wiki/PLCopen> Kasutatud: 16.04.2022.
- [18] IEC 61131-3: Programming Languages. IEC 61131-3, PLCOpen. [Online]. Loetud aadressil: <https://plcopen.org/iec-61131-3> Kasutatud: 17.04.2022.
- [19] Control Engineering, Relay Ladder Logic, 1998, [Online]. Loetud aadressil: <https://www.controleng.com/articles/relay-ladder-logic/> Kasutatud: 19.02.2022.
- [20] PLC Academy, Structured Text Programming: A Step by Step Guide (With Examples), 2015, [Online]. Loetud aadressil: <https://www.plcacademy.com/structured-text-tutorial/> Kasutatud: 25.02.2022.
- [21] Motion control tips, IEC 61131-3 programming: Where's structured text more appropriate than ladder logic?, 2019, [Online]. Loetud aadressil: <https://www.motioncontroltips.com/iec-61131-3-programming-are-there-uses-for-which-structured-text-is-more-appropriate-than-ladder-logic/> Kasutatud: 25.02.2022.
- [22] Siemens, LOGO!Soft Comfort Online Help, kasutusjuhend, Siemens AG, Saksamaa, 2022. [Online]. Loetud aadressil: https://support.industry.siemens.com/dl/dl-media/807/100782807/att_1066579/v2/153568113035_en-US/en-US/index.html#treeId=a5e1f566d78b5066d1ac77e3129019ae Kasutatud: 18.04.2022.
- [23] Open Automation Software, How to Log Siemens Data to a CSV or Excel File, 2018, [Online]. Loetud aadressil: <https://openautomationsoftware.com/knowledge-base/log-siemens-csv-excel/> Kasutatud: 18.04.2022.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Bo Aaron Kooser

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Prügikasti automaatse pesusüsteemi juhtimisprogramm prügiautole“, mille juhendaja on Andres Rähni.
 - 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.

16.05.2022

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 tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

