



Argo Ilves

LIN JUHTIMISEGA GENERAATORI ÕPPESTEND

LÕPTUÕÕ

Erialal: AUTOTEHNIKA

Tehnikainstituut

Õpperühm: KAT2019

Juhendaja: Msc Henri Vennikas

Tallinn 2023

Autori deklaratsioon ja lihtlitsents

Mina, **Argo Ilves**, tõendan, et lõputöö on minu kirjutatud. Töö koostamisel kasutatud teiste autorite, sh juhendaja teoste le on viidatud õiguspäraselt. Kõik isiklikud ja varalised autoriõigused käesoleva lõputöö osas kuuluvad autorile ainuisikuliselt ning need on kaitstud autoriõiguse seadusega.

Annan Tallinna Tehnikakõrgkoolile (edaspidi kõrgkool) tasuta loa (lihtlitsentsi) enda loodud teose „**LIN juhtimisega generaatori õppestend**“:

- reprodutseerimiseks paber kandjal kõrgkooli raamatukogus avaldamise ja säilitamise eesmärgil;
- elektroonseks avaldamiseks kõrgkooli repositooriumi kaudu;
- kui lõputöö avaldamisele on instituudi direktori korraldusega kehtestatud tähtajaline piirang, lõputöö avaldada pärast piirangu lõppemist;
- ning kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid ega muid õigusi.

Autor: Argo Ilves

/allkirjastatud digitaalselt, kuupäev digiallkirjas/

Juhendaja kinnitus

Lõputöö vastab lõputöö ja kirjalike tööde vormistamise juhendile. Lubada lõputöö kaitsmisele instituudi direktori korraldusega.

Juhendaja: Henri Vennikas, MSc, juhtivlektor, autotehnika õppekava juht

/allkirjastatud digitaalselt, kuupäev digiallkirjas/

SISUKORD

MÕISTED.....	4
SISSEJUHATUS.....	6
1 LIN PROTOKOLL.....	8
1.1. Sõidukite andmevahetuse protokollidest üldiselt	8
1.2. Andmeedastuse protokollide kiirus	9
1.3. LIN protokollide ülevaade	10
1.4. Juhtmoodulid LIN võrkudes.....	14
1.5. Täiturmoodulid LIN võrkudes.....	15
1.6. Generaatori väljundpinge juhtimine	16
2 ÕPPESTENDI KAVANDAMINE.....	21
2.1. Turul pakutavad õppestendid	21
2.2. Õppestendi disain lähtuvalt õpiväljunditest	21
2.3. Detailide valik stendi koostule	22
2.4. Riist- ja tarkvara	24
3 LIN MOODULITELE TARKVARA ARENDAMINE.....	26
3.1. Tarkvara arendusest sõidukitele üldiselt	26
3.2. UML skeemid LIN protsessi visualiseerimiseks	27
3.3. Tarkvara arendus ja koodinäited.....	29
4 STENDI KOOST.....	35
4.1. Stendi komponendid	35
4.2. Stendi moodulid.....	40
4.3. Näidiskava praktikumi läbiviimiseks.....	40
KOKKUVÕTE.....	43
SUMMARY	44
KASUTATUD ALLIKAD	45
LISAD	48

MÕISTETE JA LÜHENDITE LOETELU

AUTOSAR - (*Automotive Open System Architecture*) on autotootjate konsortsiumi poolt standardiseeritud autodele loodava tarkvara arhitektuuriline raamistik [1].

ADAS – (*Advanced Driver Assistance Systems*) juhiabi süsteemid sõidukis.

BARR – (*Best Automotive Repair and Regulatory Practices*)

BUS – (*Binary Unit System*) kahendsüsteemis info edastamise füüsiline osa, mille kaudu (kaabel vms seade, mida kutsutakse ka *SIINIKS*) edastatakse info teistele seadmele.

CAN – (*Controller Area Network*) kaablipaarist ja kontrollieritest (*ECU*) koosnev andmeedastuse võrk, mida kasutatakse autotööstuses ja tööstuslikus automaatikas kontrollierite ühendamiseks, mis arendati välja Boschi poolt 1980ndatel

CHECKSUM – ehk kontrollräsi on matemaatiline tehe edastatavas sõnumis, mille abil saaja saab kontrollida sõnumi vastavust algele sõnumile [2].

CPU – (*Central Processing Unit*) ehk keskne arvutusseade, mis sooritab etteantud tehteid [3].

ECU - (*Electronic Control Unit*) on sõiduki elektrooniline juhtseade, mis vastutab erinevate süsteemide juhtimise eest [1].

HUD – (*Head-Up Display*) – Ekraan, mida saab vaadata pea püsti ehk pilku sõidusuunas hoides ja võib olla lahendatud kas läbipaistva ekraaniga või peegeldades infot tuuleklaasile [1].

IDE – (*Integrated Development Environment*) – Ehk terviklik tarkvara arenduskeskkond, mis sisaldab koodi kirjutada, käivitada ja omab ka sisse ehitatud veatuvastus ja näiteks ka ühendus koodihoidlaga.

LIN – (*Local Interconnect Network*), sõidukisse paigaldatud ühe kaabliga ühendatud kontrollierite võrgustik, mis on kasutusel lihtsama elektroonika juhtimiseks vastava protokolliga kaudu [1].

MISRA – (*Motor Industry Software Reliability Association*) ISO 26262 standardile vastav raamistik nõuetega eelkõige turvalisusele suunatud autonduse tarkavale, mis sisaldab tehnoloogiast sõltumatut juhendmaterjali arendusele, testimisele ja ka dokumenteerimisele.

PWM – (*Pulse Width Modulation*) on elektroonika juhtimise protokoll, kus muudetakse impulsi laiuse "on" aega ehk impulsi kestvust, mida võrreldakse kogu impulsi kestvusega, see on hea võimalus analoogsignaali defineerimiseks digitaalsena [1].

RPM – (*Revolutions per minute*) – Pöörete arv minutis.

SERIAL – ehk jadaühendus tähendab et, samaaegselt edastatakse 1 sõnum korraga.

SOC – (*State of Charge*) aku laetuse tase.

SOF – (*State of Failure*) vea staatus, CAN (*Controller Area Network* puhul ka *Start of Frame*).

SOH – (*State of Health*) aku toimevõime hinnang.

UART - (*Universal Asynchronous Receiver-Transmitter*) ehk universaalne andmete edastamise ja vastuvõtmise riistvaraline moodul [4].

UML – (*Unified Modelling Language*) on tarkvara arenduses standardne visuaalne modelleerimiskeel mida kasutatakse tarkvara disaini etapis süsteemi funktsionaalsuse kirjeldamiseks diagrammina, mis näitab igas etapis protsesside järjekorda ning interaktsioonide [5].

SISSEJUHATUS

Käesoleva lõputöö eesmärk on LIN juhtimisega generaatori õppetendi loomise protsessi võimalikult detailne kirjeldus ja tulemuste analüüs. Stendi vajadus on aktuaalne ning selle loomine õppetöö käigus on hea alternatiiv ostutootele, et anda õppimisvõimalus tudengitele kui ka kogemus juhendajatele. Valmistootte puudus on see, et LIN juhtimise õppimiseks või õpetamiseks on väga vähe valikut, kus oleks pakkuda ainult nii kitsa ülesandega stende.

Töös on kirjeldatud nii tehnilise konstruktsiooni, kasutajaliidese disaini kui ka LIN protokollid kaudu seadmete juhtimise teooria ja praktiline osa. Lisaks on kaetud riistvaraliste ja koostu detailide valik ja tarkvara loomine vastavalt standarditele. Ajend lõputöö teemaks tulenes võimalusest vajalikud detailid õppeprotsessiks partneritelt saada (sh generaator) ning valmistootte hankimise asemel on vastava õppevahendi loomine rätsepatööna iseseisev osa õppeprotsessist ning võimalus koostada võimalikult täpselt õpiväljundeid toetav õppevahend tudengitele LIN tehnoloogia õppimiseks. Lisaks on LIN juhtimisega stendi loomisel saadav kogemus kirjalikus vormis hea õppematerjal edaspidi tudengitele LIN protokollid teooria omandamisel või alusmaterjaliks sarnaste projektide töös.

Töös on kasutusel erinevad autotehnika erialal omandatud teadmised, mille ühendamise võimaldab luua õppetendi näol õppeprotsessi rikastava toote, mille loomisel on võetud arvesse kõik vajalikud nõuded autotehnikas loodavate seadmete ja tarkvarade loomisel aga lisaks on lähtutud ka õppevahendi loomise didaktika teoreetilisest poolest, et toetada erinevaid õpistiile ja praktilises osas tagada vahendi kasutamisel nii turvalisus kui ka hea kasutaja- või ka õpikogemust.



Pilt 1. Sõiduki andmevõrgustiku näide [6].

Praktilise tööna stendi loomisel saab omandada teema nii teoorias kui praktikas. Stend aitab tudengitel käsitledavat teemat lihtsamalt ja samm-sammult omandada ilma sõiduki võrgustiku keerukusest (Pilt 1) end ehmatada laskmata.

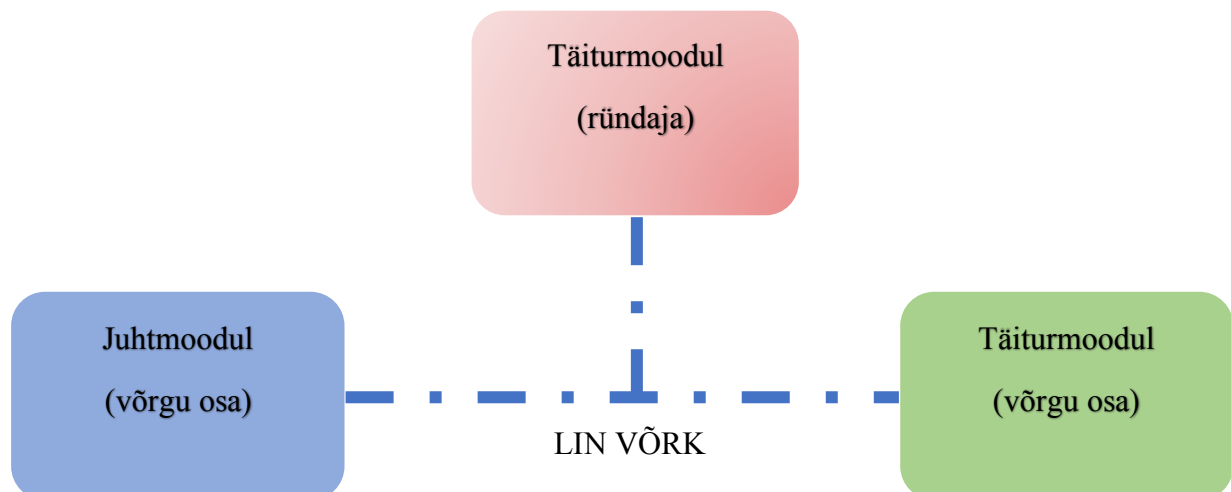
Käesoleva töös skoobis valminud stendi prototüübi võimekus piirdub pinge reguleerimisega LIN signaali kaudu. Edasiarendusena saab samale stendile lisada võimekuse muuta mootori pöördeid ja vajadusel kasutada stendi sõidukite generaatorite töökorra testimiseks jne.

1. LIN PROTOKOLL

1.1. Sõidukite andmevahetuse protokollidest üldiselt

Esmaste andmevahetuse süsteemidena tuli sõidukites kasutusele CAN 1980-date lõpus (andmeedastuse kiirus kuni 150 Kbit/s) ning hiljem Audi, BMW, Daimler-Chrysler, Volcano, Volvo ja Volkswageni koostööna LIN, mille andmeedastuskiirus kuni 20Kbit/s aastal ning jäi CAN kõrval lihtsamate funktsionaalsuste andmevahetuse katmiseks. Hiljem lisandus veel süsteeme ja protokolle andmete edastamiseks nagu näiteks Byteflight, mille kasutuselevõtt oli tingitud vajadusest tagada kiirem andmevahetus passiivse ohutuse süsteemidele - näiteks turvapadi [7].

Tänapäeval on autotootjatele suurim väljakutse lisaks auto töökindlalt toimivatele (sh elektroonika) osadele arendada turvaline ühendus ka auto komponentide omavahelise ja välise võrguühenduse kaudu toimuva võrguliikluse tagamiseks. Sealjuures tuleb tagada turvalisus nii uute ühenduste, kauguuenduste kui ka olemasolevate lihtsate autosiseste võrkude jaoks, et tagada sõiduki ja reisijate turvalisus (Joonis 1). Probleemiks on see, et viimasel kümnendil on ka autonduses kasutusel olevate võrkude vastu rünnakute tase hüppeliselt kasvanud. Üheks viimasena toimunud ründeks on hiljuti Toyota CAN võrku tungimine esitule kaudu, kuid selliseid näiteid on palju [8]. Teisalt on tootjatel ka surve toota samaväärset või paremat tehnoloogiat, mis suudab kõrgema turvalisuse juures tagada adekvaatsed andmete edastamise kiirused. [9]



Joonis 1. Turvarisk LIN võrgus.

1.2. Andmeedastuse protokollide kiirus

Andmete edastuse kiirus mistahes infotehnoloogilises süsteemis toimub üldise loogika alusel, kus andmed bitt-haaval (sisuliselt 0 või 1 ehk LIN näitel kas 0 volti või 12 volti) edastatakse saatjalt vastuvõtjani etteantud kanalit kasutades. Andmete edastamisel üle võrgu eristatakse ja vastavalt tähistatakse kiirust bittide ja sõnumi pikkust omakorda baitide kaudu. Ehk siis sõnumi pikkuses 1 bait vastab sõnumi sisule pikkusega 8 bitti ja kui tuua näiteks LIN sõnum pikkusega 64 baiti, siis bittides on selle sõnumi pikkus 512 bitti ehk 0,512 Kbit ning andmeedastuse kiiruse 20Kbit/s juures kulub sõnumi edastamiseks ligikaudu:

$$\frac{0,0512 \text{ Kbit}}{20 \text{ Kbit/s}} = 0,0256 \text{ s} \quad (1)$$

ehk ligikaudu 25,6 millisekundit. Selliselt kiiruse arvutamisel tuleb alati võtta arvesse ka võimaliku vea koefitsient, sest mida pikem on sõnum ja andmete edastamise vahemaa, vaheseadmete arv jmt., seda rohkem võib reaalses kiiruses tekkida kõikumisi.

Sõiduki elektrooniliste seadmete ehk moodulite vahel toimuva andmevahetuse puhul on oluline nii kiirus kui ka sõnumite pikkus, mis iga protokollivi või ka sama protokollivi erineva versiooni puhul on erinevad. Kiirus on seda olulisem, mida tähtsam on seadme poolt edastatava sõnumi sisu ja selle kontekst sõiduki turvalisele toimimise mõistes. Sõnumi edastamise ajale lisandub andmete töötlemisele moodulite poolt kuluv aeg ja vajadusel ka vastava vastuse saatmine, kas juhile kuvatava infona või vastussõnumine mõnele kolmandale seadmele.

Tabel 1. Andmeedastuse protokollide kiirused sõidukites [10].

Protokoll	Suurim võimalik kiirus	Kasutusvaldkond
LIN	20 Kbit/s	Kuvarid, valgustus, alarm, kliima, istmed ja peeglid, elektriaknad, kojamehed, tuled
CAN	1 Mbit/s	
CAN_FD	8 Mbit/s	Suurema kiiruse ja pikema (64 baiti vs CAN 8)
TTCAN	10Mbit/s	Mootor, käigukast, pidurid, roolisüsteem, vedrustus, assisteerivad süsteemid, turvasüsteemid, diagnostika
FlexRay	10Mbit/s	
Ethernet	1 Gbit/s	ADAS, HUD, mootor, veermik, kere, pilvelahendused, iseautod jne.

Näiteks mootori juhtimine, käiguvahetus, roolisüsteem, turvasüsteemid, assisteerivad süsteemid jmt. puhul kasutusel FlexRay (10 Mbit/s), aeglasemad LIN ja CAN protokollid aga kontrollivad monitore, valgustust, alarmsüsteeme, kliimaseadet jne. –**Error! Reference source not found.**

1.3. LIN protokoll ülevaade

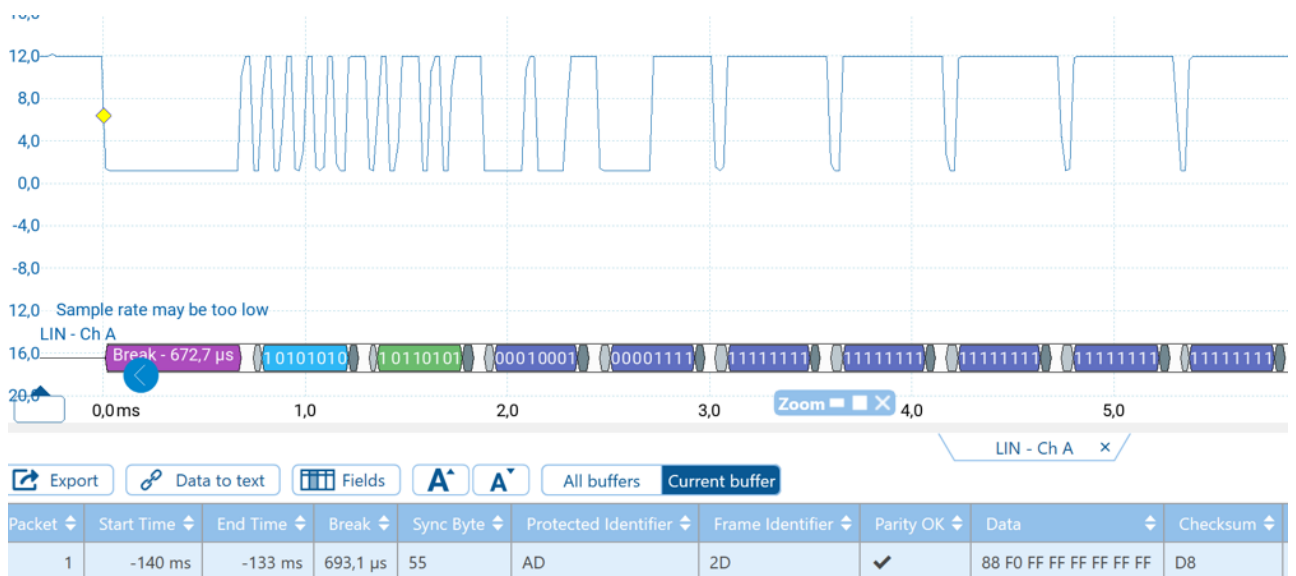
LIN-protokoll on lihtne, töökindel ja kulutõhus andmevahetuse protokoll väikese koguse elektrooniliste süsteemide vahel, kus ei ole oluline väga suur andmeedastuse kiirus. Protokoll võimaldab andmeid edastada mööda ühe kaabliga andmekanalit (maksimaalse pikkusega 40m), millega on ühendatud alati vähemalt üks juhtmoodul ja kuni 16 täiturmoodulit (versioon 2.2A) [4]. LIN protokoll andmeedastus saab toimuda kiirusel kuni 20 Kbit/s. Üldisemalt on tänaseks kasutusel LIN protokollist 2 erinevat perekonda 1 ja 2, millel omakorda 5 tüüpi ning igal tüübil omakorda 5 (A-E) erinevat versiooni, mis teeb seadmete valikul LIN võrgu ehitamiseks oluliseks iga mooduli versiooni jälgimise, sest varasema versiooni täiturmooduleid on võimalik juhtida uuema versiooni juhtmoodulitega, aga mitte vastupidi. Kõik LIN tüübid toimivad üldjuhul kiirustel 9600 või 19200 bit/s.

Infomüra vältimiseks on LIN protokollis andmeedastuse kiiruse lubatud viga piiratud 14 protsendilise kõikumisega [11]. LIN protokollis eeliseks on andmevahetuse toimumine autos kasutusel oleva pingele, ehk 12 volti või rohkem, abil signaali edastades vastavalt pingele tõstmise või langetamisega, mis loogilise digitaalse infona on edastatud nagu eespool kirjeldatud: 0 volti on digitaalne väärtus 0 ja 12 või kõrgem pingele vastab loogilisele väärtusele 1. Sõiduki tavapinge kasutamine lihtsustab ja teeb soodsamaks info edastamise ilma pinget muundamata 3- või 5-voldisele pingele, mis on tavapärane pingele digitaalsetes süsteemides. Kui sõnumit ei edastata, on liinis 12v e digitaalne 1.

Kõrgema pingele kasutamine andmete edastamisel paneb suuremad nõudmised ka vastavatele elektroonikakomponentidele, mis peavad suutma toimida pingetel kuni 30 volti olenevalt konkreetsest kasutusest [12]. Ka selliste komponentide töö lihtsus on koht, kus tekib säästuefekt, sest mikroskeemide keerukus on suhteliselt madal moodulites.

LIN sõnumi edastamiseks vajab iga moodul vastava sõnumi võrku bitikaupa saatmiseks (1 või 0) lisaks mikroprotsessorile, mis sõnumi loob, ka nõ. väravat, ehk füüsilist osa, mis edastab (TX e *transceiver* e saatja) genereeritud sõnumid vastavalt jadamisi võrku ehk näiteks kaablisse, mida mööda sõnum liigub saajani, mis edastab ja neid sealt vastu võtab (RX e *receiver* e vastuvõtja). LIN protokollis on selliseks väravaks UART moodul, mis võib olla osaks põhikiibis või eraldi kiibis. UART ehk universaalne asünkroonne andmevahetuse moodul, mille ülesanne on kõik sõnumid vastu

võtta ja edastada võrku. Kokkuvõtvalt on mooduli ülesanne kiibist saadud sõnum muuta selliseks, et see oleks võrgus üheselt arusaadav (algus, kellade sünkroonsus, andmeedastuse kiirus), mis võimaldab ka teistel võrgu moodulitel sõnumi tervikut mõista võimaliku müra seest. Seejuures peab UART olema võimeline samaaegselt võrgust saabuvasid sõnumeid lugema ja neid terviklikuna kätte saades ehk siis kontrollinud nii ajalist vastavust kui ka *checksum*, edastama sõnumi sisu CPU-le.



Pilt 2. LIN sõnumi osad.

LIN-protokolli sõnumi põhiosad on (Pilt 2):

- *break* e paus, millega antakse teistele võtku ühendatud kontrolleritele märku, et algas uus sõnum.
- *sync* sünkroniseerimine, mille põhjal saavad kõik võrku ühendatud kontrollerid aru, millise andmeedastuse kiirusega sõnumid liiguvad ja saavad nii end teiste seadmetega sünkroonis hoida.
- *Protected identifier* ehk teatav identifikaator (edaspidi PID), mis annab teada, mis sõnumiga on tegu ja millisele moodulile või ka mitmele on sõnum suunatud. Siinkohal tuleb tähele panna, et identifikaator võib olla ka nõ mitmele ehk *broadcast* ehk nõ levisõnum, mis on suunatud mitmele saajale.
- *Data* ehk sõnumi sisu, mille pikkus ja sisu on varieeruv sõltuvalt sellest, mis võrguga on tegu ja mis on juhtmooduli eesmärk, kuid üldjuhul on see seotud sõnumi ID-ga. Andmevoog on üldjuhul kahekihiline sisaldades nii, juhtmooduli algatatud käsku kui ka täituri poolt lisatud infot käsu täitmise kohta ehk *response* osa.

- *Checksum* ehk kontroll, mis on matemaatiline tehe, mis koosneb sõnumist ja identifikaatorist, mille põhjal sõnumit vastu võttev moodul saab veenduda, et saadud sõnum on korrektne [13].

Sõnumi saatmise loogika toimib LIN protokollis vastavalt sõnumi kujule nii, et sõnumi algataja, mis võib olla võrgus (peamiselt) juhtmoodul, aga ka täiturmoodul, kui tal on vaja edastada sõnum juhtmoodulile. Saatjalt edastatud sõnumiga on selline, et see edastatakse LIN võrku, kus selle võtavad vastu kõik võrku ühendatud moodulid. Nagu mainisime sisaldub igas sõnumis lisaks *break* ja *sync* väljadele ka *identifier* ehk identifikaator. Kui kõik LIN võrku ühendatud moodulid kuuluvad nõo LIN sõnumi eetrit pidevalt, siis seda, kas ja kellele sõnum on suunatud määratleb ära sõnumi identifikaator. See tähendab, et iga võrgus olev moodul omab oma identifikaatorit ning kui saadud sõnumis sisaldub vastav identifikaatorit, asub moodul sõnumit täitma [14].

Sõnumi täitmise osa toimub eelkõige seeläbi, et moodul täidab vastava funktsiooni ja seejärel saadab võrku vastuse, et juhtmoodul saaks kinnituse korralduse täitmise õnnestumise või ka ebaõnnestumise kohta, mis sisaldab sama identifikaatorit, mille *data* ehk andmeväljale lisab täiturmoodul info, mille põhjal juhtmoodul saab aru, kas ülesanne täideti või tekkis tõrge vms.

Tabel 2. LIN sõnumi osad ja pikkused erinevate protokollide versioonide puhul [15] [14]

Sõnumi osa	Protokollide tüüp	
	LIN 1.x	LIN 2.x
<i>Sync</i> väli	1 bit	1 bit
<i>Break</i> väli	13-31 bitti	13-31 bitti
PID	1 bit	1 bit
Andmevälja pikkus	1 bit	2 bitti
Andmeväli	8 bitti (1 bait)	1-64 bitti (8 baiti)
		Süsteemne ID: 1 bit
		Sõnum: 1-46 bitti
		ID: 1 bit
		Tootja ID: 1 bit
		Funktsiooni ID: 1 bit
		Versioon: 1 bit
		Jääk: 1 bit
Vastus (valikuline)	1-8 bit	Osa andmeväljast
Kontrollräsi	1 bit	1 bit

LIN sõnumivahetuses *data* ehk andmete osa saab omakorda sisaldada erinevaid tüüpi andmevälju, igal väljal on vastavalt sisule, mida sellega edastatakse oma ID ja iga väli sisaldab kasutatavast versioonist alati 1-64 bitti olenevalt protokollidest versioonist. Vaikimisi on andmevahetuses andmeväljad sõltumatud ehk ilma eritingimusteta väljad, kus juhtmoodul saadab identifikaatori põhjal päringu konkreetsele täiturmoodulile ja vastav moodul reageerib käsu täitmisega ja raporteerimisega.

Iga versiooni puhul on andmete edastuse loogika erinev, millega peavad arvestama võrgu koostajad sh Detailide tootjad, seejuures kõrgemat versiooni toetavad juhtmoodulid toetavad eelmise versiooni protokollidega täiturmoodulitega suhtlemist, aga vanemad juhtmoodulid uuemate täituritega ei ühildu. All olevas tabelis (**Error! Reference source not found.**) on näha peamine võimalik andmemaht sõnumi osade ja LIN perekondade lõikes [15].

Lisaks sõnumi osade pikkusele on erinevad ka kontrollräsidsid, kus 1.x versioonide korral on tegemist klassikalise kontrollräsiga, mis saadakse andmebaitide ja ID summeerimise ning summast pöördkoodi [16] leidmisega kahendsüsteemis. Uuemate protokollide puhul on, sealhulgas ka turvalisuse eesmärgil, kasutusel keerukamad kontrollräsidsid, kus andmevälja pikkusele liidetakse klassikaline kontrollräs, millele omakorda liidetakse võtmesõna baitide summa ning lõpuks leitakse ka siin saadud lõppsumma pöördkood [15]. Kasutada saab ka kombineeritud varianti, kus mõlemad summeeritakse ja leitakse pöördkood. Keerukust lisab sellise turvalisuse tõhustamise juures see, et ka moodulid, kes sõnumit saavad, peavad omama vastavalt suuremat arvutusvõimsust, et vajalik tehe jõuda teha ja sõnumisse kaasa panna.

Mistahes LIN võrgus või sinna saadetud välise info (ECU kaudu juhtmoodulile edastatud info teistelt anduritelt) põhjal algatatud sõnum, milles juhtmoodul edastab identifikaatori põhjal sõnumi mitmele täiturmoodulile, soovides moodulitelt vastavalt kas teavet või mingi korralduse täitmist. Täiturmoodulid, mis saadud sõnumi ID põhjal vastavad sõnumile, kui vastavale päringule on neil vastus pakkuda. Kui sellises päringus tekitab võrgus müra ehk mitu moodulit üritab sama sõnumit saata üheaegselt, lülitub juhtmoodul klassikalisele sõnumivahetusele ja saadab sama päringu võrku igale seotud moodulile eraldi.

Järgmiseks sõnumi tüübiks on *sporadic* ehk „ise vastan, sest tean“, kus andmevälja edastab juhtmoodul ise, kui talle on selge, et mõne täiturmooduli andmed on uuenenud ehk toimides ise täiturmoodulina saadab info võrku koos uue infoga, mida ta ei pidanud küsima täiturmoodulilt vaid uuendab info ise kõigi teiste võrku ühendatud seadmete jaoks kuna info oli talle juba teada.

Diagnostikaks kasutatakse alates LIN versioonist 2.0-st ID-sid 60 (juhtmooduli päring) ja 61 (täiturmooduli vastus) diagnostika info kogumiseks. Kasutaja poolt seadistatud väli ID-ga nr 62, mis võib sisaldada sellist infot, mis kasutajale antud hetkel vajalik. Reserveeritud ID-ga väli on 63

Nagu mainitud on võrgus oht, eriti mitme täiturmooduli puhul, et sõnumid põrkuvad või siis edastatakse samaaegselt, Sellisel juhul nagu mainisime, pöördub juhtmoodul uuesti iga täiturmooduli poole eraldi vastavalt tema ID-le, et saada vastus soovitud päringule ja vältida müra. Sõnumite samaaegsus võib tekkida ka muudel juhtudel. Näiteks võrgus olevad täiturmoodulid annavad teavet seisundi muutusest. Sellises olukorras jälgib iga sõnumi edastaja eetrit ja kui tema poolt saadetava sõnumi ID on madalama prioriteetsusega, siis edastab vastav moodul oma sõnumi peale kõrgema tasemega sõnumit. Kui eeter „vabaneb“, edastatakse madalama prioriteediga (ID-ga) sõnum uuesti vastava mooduli poolt.

Lisaks annab LIN võrgu ja protokollile tervikuna kuluefektiivsuse mõõtme, võrgu võimekus iseseisvalt peale 4 sekundit kestvat vaikust eetris minna puhke režiimi. Puhke režiimi saab algatada ka juhtmoodul. Süsteemis saab eetri ja sõnumivahetuse uinunud olekust äratada mistahes võrku ühendatud moodul, millel on vajadus sõnumi edastamiseks.

1.4. Juhtmoodulid LIN võrkudes

LIN Juhtmooduli ülesandeks on LIN võrgus juhtida ja jälgida teiste samasse võrku ühendatud moodulite tööd. Tegemist on kontrollieriga, millel üldjuhul võimekus lisaks muude protokollide (CAN eelkõige autonduses) signaalide saatmisele ja vastuvõtule ka vastavalt olla ühendusees ka näiteks CAN võrguga [13]. LIN juhtmoodul võib saada lähteülesande ka CAN võrgust, milles on antud LIN juhtseade kui CAN täiturmoodul. Sellise korralduse saamisel peab LIN juhtseade selle suutma vastu võtta (käesoleva töö eesmärk ei ole LIN moodulile antava välise võrgu nagu näiteks CAN juhtimiskäskluse loogika detailne analüüs) ja näiteks edastada vajadusel teisele LIN täiturmoodulitele, mis asuvad temaga samas LIN võrgus [13].

Lisaks vastavate LIN protokollide signaalide edastamisele täiturmoodulitele, peab juhtmoodul jälgima teiste LIN täiturmoodulite tööd ning vajadusel reageerima nende tõrgetele, sõnumitele vms. Sealhulgas juhtima nende toimet ning võimaldama neil vajadusel sõnumit korrata, aga kui on ilmnenud tõrge seadme tuleb LIN juhtmoodulil veast teavitada vajadusel ka teisi süsteeme nii, et vajadusel ka sõiduki juht oleks tekkinud probleemist teadlik – näiteks CAN võrku ühendatud signaallambi kaudu. Samuti tuleb selline viga edastada auto juhtmoodulile, mille põhjal sõiduki hooldamisel on võimalik viga defekteerida ja vigane moodul ka kas asendada või parandada. LIN

juhtmooduli ülesanne on tagada andmete kao või konflikti korral vastavate täiturmoodulitele uute või korduvate korralduste jagamine, vea tuvastamist võimaldab vigase sõnumi korral ka *checksum*. Nii saadab LIN juhtmoodul näiteks korraldusel mingi seadme sisse lülitamiseks, kuid kui esmasel katsel vastav täiturmoodul ei vasta korrektselt kui kõik vastavad nõuded on täidetud [1].

LIN juhtmoodul töötab ka vastupidi, võttes vastu sõnumeid täiturseadmetelt. Näiteks kui mootori kontrollimise moodul saadab LIN juhtmoodulile mootori kohta teavet, siis viimane töötleb selle teabe ning vastavalt vajadusele kas vastab sellele vastavalt või annab järgmisele täiturmoodulile uue ülesande või saadab hoopis CAN kanalisse vastava teate.

Juhtmoodul on vajadusel muutma ka oma võrgus toimivate andmete struktuuri, kui selline juhis tuleb väljastpoolt / kõrgemalt. Lisaks on LIN juhtmooduli ülesanne juhtida võrgus unerežiimi, kui see on võimalik. Nagu mainitud peab juhtmoodul käsitlema kõiki veateateid ja näiteks ka samaaegseid sõnumeid täituritelt, misjärel on tema ülesanne eeter peatada ja küsida igalt täiturilt üle, mis sõnumiga tegu. Lisaks on juhtmooduli ülesanne võrgu diagnostika, mis tähendab kindla korrapärasusega sõnumite edastamise ja teiste moodulite toime hindamine ja vajadusel sekkumine [5].

1.5. Täiturmoodulid LIN võrkudes

LIN täiturmoodul on sarnaselt juhtmoodulile elektrooniline seade, mis LIN võrgus võib olla väga erinevate funktsioonide täitmiseks. Peamiselt on sellised täiturmoodulid kasutusel lihtsamate ja juhi vaates mitte-elutähtsate (turvavööde pingutid, turvapadjad vms) jaoks. Näiteks võib mainida:

- Roolil või selle juures asuva lülitussüsteemi juhtimine - raadio, püsikiirusehoidja, ohutuled, klaasipuhastid, kliimaautomaatika;
- Mugavusseadmete juhtimine - temperatuur, katuseluuk, salongivalgustus, vihasensor;
- Jõuülekandes - kiiruse, asukoht sõidurajal, rehvirõhu;
- Mootori juhtimises – väiksema olulisusega andurid, väikesed mootorid, jahutusventilaatori mootorid;
- Auto kere - küljepeeglid, aknad, lukud;
- Istmed - istumisandurid, istmete asendimootorid jne. [17]

LIN täiturmoodulid on tavaliselt väikesed, lihtsad ja odavad elektroonilised komponendid, kuna nende ülesanne on vastu võtta juhtmoodulilt sõnum, see täita ja edastada tagasi võrku sõnum kas täitmise või vea kohta. Selline infokogus ei vaja suurt andmemahut ega arvutusjõudlust. Sellised moodulid võivad tihti koosneda vaid ühest mikroskeemist. Keerukamad seadmed võivad sisaldada

ka komponente, mis võimaldavad mooduli erinevaid staadiume või veateateid salvestada, et neid oleks võimalik hilisema diagnostika käigus uurida [17].

Näiteks saab selline moodul, mis suudab toimida täiturina generaatori väljundpinge reguleerimiseks ka LIN võrgus ka kõiges ca 10mm lai ja 5 mm paksusega [12]. Mis näitab, et füüsilises mõttes saab LIN võrgu täitus olla väga väike elektrooniline element. Olenevalt protokollist on vastuse viis erinev, kuid see sõltub ka igakordsest rakendamise kohast. Täiturmoodul võib seega täiesti iseseisvalt infot anda, kui talle vastav (andur) info muutuse kohta saabub. Sõnum võib olla saadetud vastuseks juhtmooduli poolt vastava ID-ga saadetud päringule nii, et vastav juhtmooduli saadetud sõnumit täiendatakse vastuse väljadega [15].

1.6. Generaatori väljundpinge juhtimine

Generaatori ülesanne on sõiduki vooluringis, mille peamine toiteallikas on aku või akud, vajaliku pinge hoidmine, et katta tarbijate vajadus. Pinge hoidmisel tuleb lähtuda mitmest aspektist. Loomulikult on esmane kriteerium konkreetse sõiduki vooluringi disain. Sõltuvalt, kas tegu on 12 voldise või 24 voldise või mingi kolmanda lahendusega süsteemiga, peab ka generaatori väljundpinge vahemik olema akule ja muudele tarbijatele sobiv. Generaatori töö üldisel põhimõttel siinkohal pikemalt ei peatu rohkem, kui selles vaates, et klassikalisel kujul on generaator rihm- või muud tüüpi ajamiga seade, mida mootori töötades vastav ajam samuti ringi ajab – suhtega 1:3 [1]. Ringliikumine generaatori osade (staator ja rootor) vahel tekitab nimetatud seadmete ehitusest ja elektromagneetilise induktsiooni printsibist (ehk Faraday seadusest):

$$U_{ind} = \frac{d\phi}{dt} \quad (2)$$

pinge staatori mähistel [1]. Siit tuleb välja ka peamine kitsaskoht, kus generaatori väljundpinge U_{ind} on sõltuvuses magnetvälja muutusest asjas, täpsemalt generaatori puhul pöörlemiskiirusest. Ehk kogu sõidu vältel võib vedav mootor (milleks klassikalisel juhul on sama mootor, mis auto sõitma paneb) pöörelda väga erinevatel kiirustel, mis omakorda eelpool mainitud elektromagneetilise induktsiooni toimel võib pinget muuta väga suures vahemikus. See võib viia pikemaajaliselt olukorrani, kus tarbijad või ka aku ei pea ülepingele vastu ja purunevad või süsteemid ei tööta nagu soovitud, kui pinge auto vooluvõrgus langeb alla nende nimipinge või ületab seda liiga suurel määral.

Seadmed on sõidukis alalisvoolul toimivad, aga pöörlemisel tekkiv vool on vahelduvvool, sest generaator toimib ergutusvooluga sünkroongeneraatorina. Vahelduvvoolu saab süsteemis alaldada vastavate dioodsildade ja kondensaatorite abil, kuid pinge reguleerimine on keerukam. Täpsustuseks

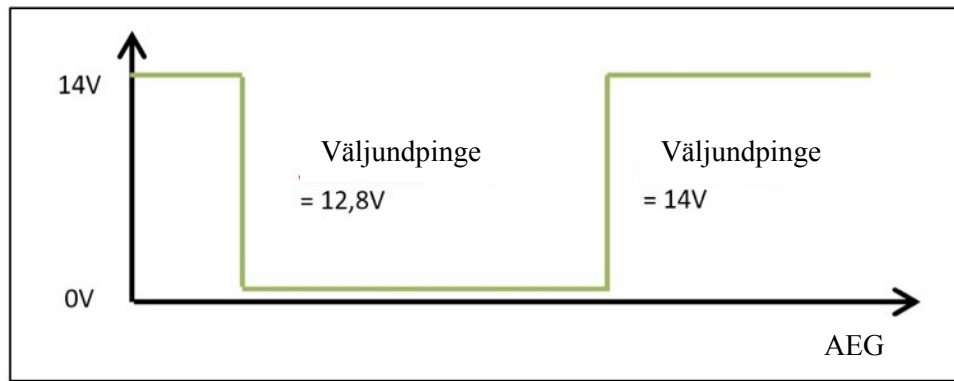
tuleb mainida, et Pinge staatoris tekib tänu ergutusvoolule rootoris, mis muudab rootori pöörlevaks magnetväljaks.

Sarnane loogika toimiks ka püsimagnetite korral, kuid elektrimasinate ja generaatorite üldisel toimel siin pikemalt ei peatu. Suures plaanis on sõiduki generaatori toime sarnane klassikalise sünkroongeneraatori tööpõhimõttele, mis koosneb pöörlevast magnetväljaga rootorist ja staatorist, milles tekib vahelduvvool vastavuses ergutusvoolu suurusele, mis omakorda mõjutab magnetvälja rootoris ning oluline on tähele panna, et sellises generaatoris on magnetväljapöörlemine arusaadavalt sünkroonis rootori enda pöörlemisega [18].

Kõige lihtsam ja algelisem lahendus, mis võeti kasutusele ergutusvoolu reguleerimiseks oli loogika, kus teatavatel pööretel seadmed ergutusvoolu kontaktide ühendust vastavalt generaatori väljundpinge tõusule katkestati, et eemaldada ergutusvool. Generaatori väljundis vahelduvvool tänu sellele alanes, et magnetväli aeg-ajalt kadus, kuid tänu vastavale voolu stabiliseerimisele ja dioodsildadele oli võimalik siiski toota ühtlast pinget. Kuna sõiduauto vajab normaalseks toimeks pinget vahemikus 12-14 volti, siis algsed regulaatorid andsidki vastavalt kas 12 või 14 volti väljundpingeks. Sellised lahendused võeti kasutusele 1960. aastatel. Hiljem konstrueeriti sellised regulaatorid osana generaatori staatorist.

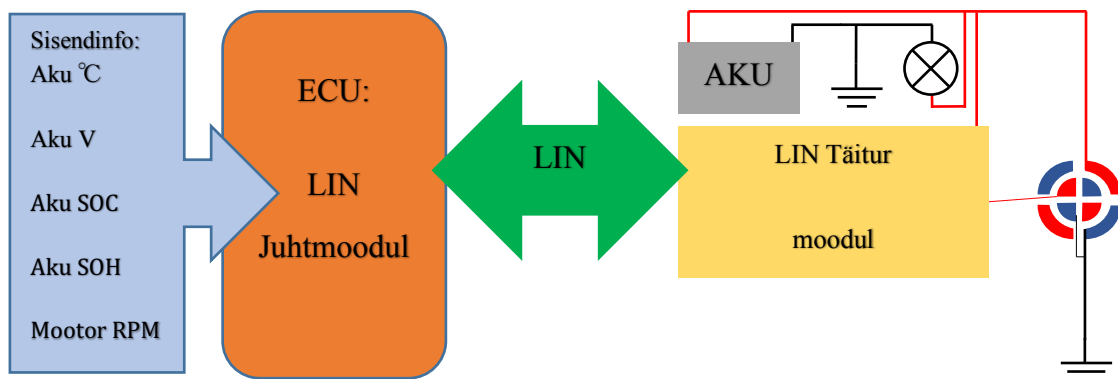
Nagu antud töö teema viitab on tänapäevastel sõidukitel kasutusel palju keerukam lahendus. Ühelt poolt toimub regulaator keerukamalt, sest täienenud auto ECU-de süsteemi tõttu suudab sõiduki pardaarvuti jälgida ja vastavalt ka seadistada reguleerima vahelduvvoolugeneraatori pingeväljundit reaajas ergutusvoolu muutmise kaudu. Vajadus sellise kontrolli järele on ühest küljest seotud eesmärgiga säästa kütust, sest ka generaatori ringiajamine voolu toomiseks kõrgema pingega, võtab rohkem energiat, kui madalama pingetootmine, ent arvestada tuleb paljusi muid näitajad.

LIN juhtimise peamine eesmärk on laadimispinge optimeerimine, et pikendada kõikide komponentide (sh Aku) eluiga seda korrektse pingega ja õigel ajal laadides. LIN juhtimisele eelnesid nõ. „targad“ protokollid, kus generaatori väljundpinge suurust juhiti PWM signaali kaudu, mis toimus põhimõttel, mida laiem (ajaliselt) impulss, seda suuremat väljundpinget generaatorilt „telliti“ ECU poolt. LIN puhul on aga võimalik täpsemalt arvesse võtta kõiki vajalikke sisendeid. Näiteks hindab ECU LIN signaaliga ergutusvoolu juhtides, milline on aku laetus, sisetakistus, koormus, temperatuur, mootori pöörlemiskiirus jne.



Joonis 2 PWM kaudu väljundpinge juhtimine [19].

Aku täpsem ehitus ei ole konkreetse töö kontekstis oluline, aga keerukamad protokollid (sh LIN) juhivad laadimispinget väga mitmete parameetrite põhjal, mida sõiduki töö ajal ECU andurite kaudu kogub, kus on lisaks voolu tarbimisele jälgimisel ka erinevate andurite abil aku enda seisukord (Joonis 3). ECU-le anduritelt saadud info põhjal saadetakse pinge kontrolleriile vastav LIN sõnum, mille põhjal kontrolleri muudab ergutusvoolu rootoris ja selle kaudu muutub väljundpinge.

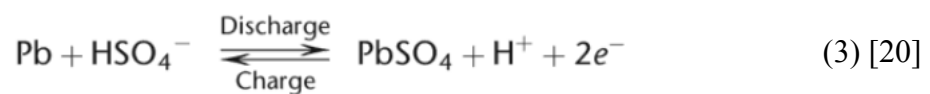


Joonis 3 LIN juhtimisega generaatori skeem.

LIN süsteemi rikke puhul on mõningad kaasaegsemad kontrolleriid ka iseseisvalt võimelised automaatrežiimil reguleerima ergutusvoolu. Selline režiim on siiski erandlik ja üldise eesmärgi täitmiseks, milleks on sõiduki heitmete vähendamine, peab ECU täpselt juhtima LIN kaudu ergutusvoolu võttes arvesse paljusid näitajaid [12] [20]. Lisaks võib kontrolleri olla võimeline mootori käivitamise ja süüte sisse lülitamisel (pedaali vajutus) põhjal ECU poolt antava korralduse põhjal tekitama generaatori rootoris ka enne mootori käivitamist (ehk generaatori pöörlemist) ka ettevalmistava ergutusvoolu, et kohe mootori pöörlemisel saaks lisavoolu ka generaatorist.

Sellised näitajad võivad olla näiteks SOC, mille madala näitaja puhul sõiduki ECU tõstab vastavalt laadimispinget, et lisaks tarbijate toimimisele saaks ka aku ise piisavalt laetud. Keerukamad näitajad on veel SOH, SOF, aga ka temperatuur ja voolutarve, mis omakorda muudavad vajadust generaatori

vajaliku väljundpinge osas. Lisaks suudavad uuemate sõidukite juhtmoodulid kohanduda laadimist ehk pinge juhtimist juhi sõiduvõtete, distantside jmt. põhjal õppida, kuidas optimeerida voolutarvet ning seeläbi vähendada kütusekulu kui ja säästa sõiduki aku eluiga. Selleks optimeeritakse aku parameetrite, koormuse ja mootori pöörete põhjal parim aeg, millal ja millist pinget kasutada auto laadimiseks. Lühidalt akus toimuvat kokku võttes võib selgitada nii, et kui aku kas liiga pikalt seisab (näiteks ka üle talve) pidevalt tühjenedes või ka sõitval autol voolu tarbimise osa on ajaliselt või mahuliselt suurem, kui tagasi laadimine, tekib aku anoodidele kristalne sulfaadi kiht, mis pikapeale muudab aku kasutuskõlbmatuks. Valemite kujul saab plii ja pliioksiidi ning katalüsaatori toime välja tuua laadimise ja tühjakslaadimise (ka iseenesliku) ajal järgmiselt:



Generaatori poolt laadimise ehk väljundpinge aja ja tugevuse loogika huvides on peab aru saada, et igas olukorras käitub süsteem erinevalt ja muudab näitajad vastavalt olukorrale. Peamised meetodid selliseks automaatseks, tõsi ECU poolt juhitud, väljundpinge peamisteks seadetekks generaatoris on:

- Aku desulfatsiooni režiim
- Käivitusrežiim
- Kütuse säästmise režiim
- Tuledega sõidu režiim
- Pinge vähendusrežiim
- Laadimis ehk tavarežiim
- Aeglustusrežiim [21].

Desulfatsiooni režiimis suurendab laadimissüsteemi väljundpinget teatud aja möödudes (tavaliselt umbes 45 minutit), kui aku laetus (SOC) on endiselt madal, et vältida pliiaku anoodidele plii-sulfaadina ladestuvate kristallide suurenemist ja vältida seeläbi aku jõudluse toimet .

Käivitusrežiimil kasutab generaator pärast sõiduki käivitamist ECU poolt tõstetud mootori pööreid ja annab väljundpingeks vähemalt 14,5 volti ja seda vähemalt 30 sekundi jooksul. Pöörded tõusevad peale sõiduki käivitamist mootori kiiremaks soojenemiseks ja ettevalmistamiseks sõiduks, sel ajal kõrgendatud pingega laeb auto starteri kasutamise järsult kaotatud suurt hulka aku mahtu. Kütuse säästmise režiimil ECU alandab pinget 13 voldini või alla selle, et vähendada koormust mootorile ja parandada rihma eluiga ning säästa kütust.

Tulede režiim rakendub võimsate esitulede kasutamisel ehk tarbimise suurenemisel vooluvõrgus, sarnane pinge tõstmine toimub auto klaasisoojenduste või istmesoojenduste kasutamisel. Pinge vähendusrežiim - vähendab väljundpinget, kui aku on laetud vähemalt 80% või rohkem ja voolutarve on madal. Laadimis ehk tavarežiim - mõeldud aku 80%-ilise laetuse taseme säilitamiseks. Aeglustusrežiim toimib osadel sõidukitel nii, et aeglustamisel auto liikumise energiat kasutatakse aku laadimiseks [18].

Õppesendis ei ole algselt plaanis täita täies mahus generaatori väljundpinge juhtimise ja laadimise loogika, mis muu hulgas peaks arvestama nii mootori pöörlemiskiirust kui ka näiteks aku temperatuuri jmt. Kindlasti on eesmärk luua terviklahendus, kus juhtmooduli poolt saadetud korraldus ehk väljundpinge suurus on korrektne ning jääb püsima olenemata koormuse muutusest samasuguseks uue korralduse saamiseni juhtmoodulilt.

2. ÕPPESTENDI KAVANDAMINE

2.1. Turul pakutavad õppestendid

Turul pakutavad stendid on väga võimekad ja kindlasti iga konkreetse stendi korral täidavad sellele esitatud eesmärgid õppurile vajalike õpiväljundite omandamise näiteks vastavate laboritööde kaudu.

Töö koostamisel silma jäänud pakkujad on:

- Consulab [22]
- TECHNOLAB SA [24]
- Initek [24] jne.

Kahjuks palju snitti pakutavate pealt võtta ei saa, sest kõik materjalid on kiivalt varjatud ja ka hindade osas saab vaid päringuid teha. Enim jäi silma Thepra ehk Technonlab SA, Saksa tootja [24], mis pakub väga laia valikut ja kõrget modulaarsust, kus näiteks sama elektrilise veomootori külge on võimalik tellida mitu erineva toimega generaatorit jne. Kahjuks on sellise komplekti, mis sisaldab stendi koos kolme erineva reguleerimise loogikaga generaatoriga ja sisaldab ka veomootorit orienteeruvalt 35000€.

2.2. Õppestendi disain lähtuvalt õpiväljunditest

Autotehnika õppekavale õppevahendi disainimisel tuleb lähtuda vajadustest ja õppekava eesmärkidest. Rakenduskõrghariduses on iga õppekava eesmärk tagada õppijate praktilised oskused ning teadmised õpitavas valdkonnas. Seetõttu peab õppevahend olema koostatud nii, et see vastaks õppijate taustale, erinevatele õpieelistustele, teoreetilisele taustale ja tasemele ehk eelteadmistele.

Õppestendi loomise idee rätsepatööna tuli varasemate kogemuste najal, kus õppevahendi koostamise käigus on saanud väga tugeva teooriabaasi valdkonnas tööd juhendavad õppejõud. Varasemalt on kooli õppejõudude praktilisel ja teoreetilisel juhendamisel valminud sarnase lähenemisega CAN juhtimisega sõiduki kliimasüsteemi (salongi soojendus / jahutus koos juhtsüsteemiga) õppestend, mis valmis mitme tudengi ühisprojektina, kus toimeloogika pidi lõpuni selgeks saama lisaks autoritele ka töö akadeemilisele juhendajale. LIN võrk on veidi teistsuguse toime põhimõttega, kui CAN, aga sarnane loogika on hea lähenemine stendi loomisele, kus süsteemi sisuline töö saavad läbi testida igakülgsest nii teoorias kui ka praktikas nii autor kui ka juhendajad.

Disainimisel tuleb arvestada õppimise erinevaid õpistiile ja õpieelistusi. Näiteks peavad õppevahendid sisaldama nii visuaalseid (näidikud, signaallambid) kui ka praktilisi tegevusi

(ühendamine, mõõtmised, seadistamine), et toetada erinevate õpistiilidega õppijaid. Samuti tuleb kasutada interaktiivseid elemente, et muuta õppimine huvitavaks ja kaasahaaravaks. Tervikuna peab disainitav õppevahend toetama õppija varem omandatud faktiteadmiste alust, mis ilmselgelt eeldab teooriaõpet enne praktikumis stendi kasutamist. Lisaks peab õppevahend aitama mõista teoreetilisi fakte kontseptuaalse raamistiku kontekstis ning toetama teooriateadmiste struktureerimist ja seoseid õpitud teooria ja praktikumi tööde vahel viisil, mis hõlbustab teadmiste omandamist ja edaspidist praktilist rakendamist [25].

Õppestend peab omama lisaks väga selget ja loogiliselt struktureeritud juhendit, et tagada õppijatele hea arusaam praktilise töö teemade sisust, nende omavahelisest seosest ja samuti seosest varem õpituga. Juhendi keel peab olema selge, konkreetne ja üheselt mõistetav koos vajadusel viidetega muudele materjalidele joonistele või teoreetilisele materjalile. Selge sõnastus tagab praktilise töö sujuvuse ja eesmärgi selge mõistmise, kus näiteks tööd sooritab mitte emakeeles õppiv tudeng. Juhend peab kindlasti sisaldama praktikumis läbitavate tööde illustreeritud juhendit samm-sammult, koos igat etappi toetava skeemi, joonise, graafiku vms kaudu, et muuta õppeprotsess ka visuaalselt paremini arusaadavaks.

Täpsemalt saab käsitleda juhendi koostamist koos laboritöö näidiskavaga järgmistes peatükkides, võimalusel tuleb õppevahendi disainimisel kasutada erinevaid keerukuse tasemeid, kui stendi funktsionaalsus seda võimaldab.

Lisaks eelmainitule peab mistahes stend olema eelkõige turvaline ja vastama ning olema ka testitud valdkonnas kehtivate standardite põhjal. Ning lisaks õppevahendi turvalisusele tuleb testida stendi ka testõppuritega, kelle tegevuse jälgimisele või ka muul moel tagasiside põhjal on võimalik vajadusel kas seadet või juhendeid kohandada või turvalisemaks muuta.

2.3. Detailide valik stendi koostule

Nagu mainitud sissejuhatuses, siis algmaterjaliks on garantiikorras partneri, ELKE Auto, poolt sõidukilt eemaldatud generaator. Kasutatud generaatori kasutamise esmane miinus on see, et ei saa kindel olla lõpptulemuse toimimises, sest teadmata on täpne põhjus, miks konkreetne seade sõidukilt ma ha võeti ehk mis täpsemalt on seadme tehniline probleem.

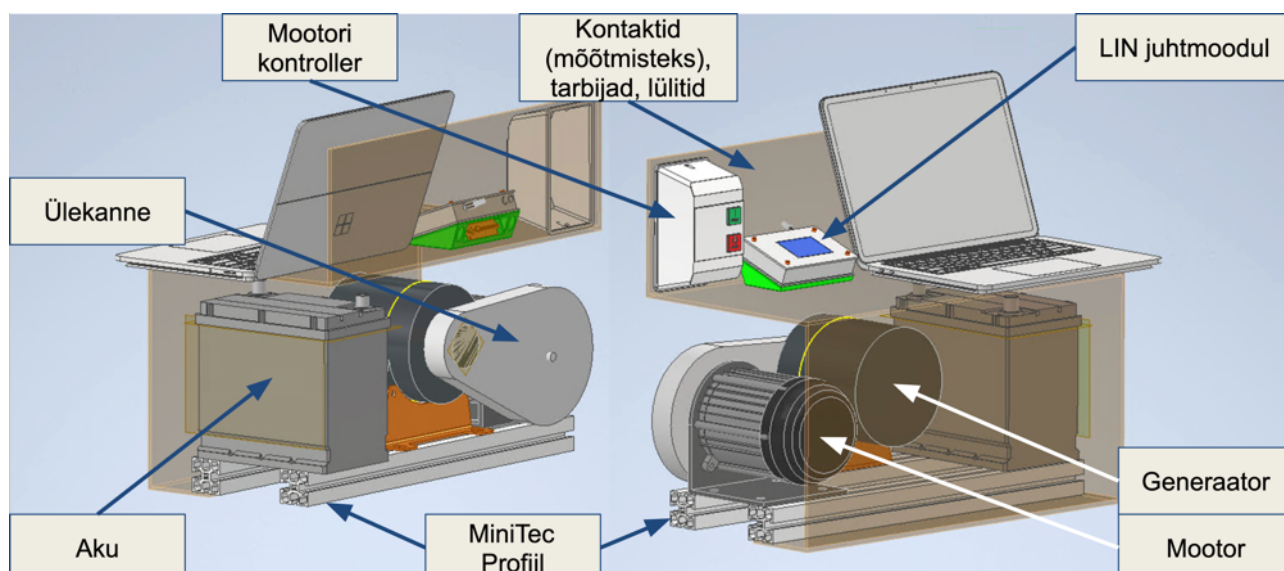
Sellegi poolest on ka ebakindel generaator esimese stendi prototüübi loomiseks ideaalne variant, et aru saada kuidas seadmed stendis tööle panna ja vastavalt valida sobivad – elemendid ajamina toimiv mootor, stendi raam ja seada paika vajaliku kaabelduse paigutus. Varasema kogemuse pinnalt, kus lahendus toimis, tuli mõtte kasutada stendi raamina samuti valmisprofiile (Näiteks: Minitec).

Muude elementide valikul on tähtis tagada kasutajale ohutus, aga mõnes olukorras võib olla vajalik kasutada ka läbipaistvat materjali (Näiteks: Pleksiklaas), mis võimaldaks kasutajal jälgida süsteemi tööd vms.

Stendi koost peab tervikuna sisaldama vähemalt:

- Raam, mille põhiosa koosneb valmisprofiilidest, mis oma universaalsuse tõttu on head prototüübina loodava stendi koostuks.
- Elektrimootor, mille väljundvõimsus on piisav, et anda süsteemi ühendatud generaatorile piisavalt väljundvõimsust, et tagada stendi toiteahelale vajaliku toitepinge ja voolu saamine generaatorist, mille juures tuleks arvestada ka kadudega.
- Ülekanne, mida ilmselt on kompaktsuse ja universaalsuse mõistes lahendada rihmülekanega.
- Aku, millele vajadusel saab lisada hiljem ka temperatuuri anduri, sest terviksüsteemis peab generaatori väljundpinge valimisel juhtkontroller võtma arvesse ka aku temperatuuri [1] väljundpinge reguleerimisel. Aku valikult tuleb silmas pidada seda, et kasutusel olevad tarbijad töötaks süsteemis ka ilma generaatorit käivitamata, et saada võimalikud ja vajalikud näitajad laboritöö täitmiseks.
- Generaator Toyota Proace sõidukilt
- Tarbijad, mille juurde sobivad lülitid, et tagada ohutu tarbijate ühendamine mistahes tööhetkel stendi kasutamisel. Tarbijad peavad olema igaüks eraldi lülitatav. Tarbijate võimsus peaks olema selline, et nende ahelasse lülitamisel oleks generaatori poolt saadava võimsuse muutus astmeliselt mõõdetav. Näiteks kolm tarbijat nimivõimsustega 35, 55 ja 200 vatti.
- Ühenduspaneel, koos sobivate kaablitega, et ohutult mõõta laboritöös vajalikke näitajaid generaatori stendiga töötamisel

Järgnevates peatükkides viin läbi täpsemad arvutused, et piiritleda valitavate toodete nõuded. Sealhulgas profiilid, veomootor, vajalikud ülekannete arvutused jne. Algne idee on näha all oleval pildil (**Error! Reference source not found.**), kus kogu ringi aetav mehhanismide osa on kaetud ja a sub stendiga töötavast tudengist kaugemal küljel. Tudengi käeulatusse on planeeritud mootori kontroller, et muuta vajadusel veomootori pöörlemiskiirust ja LIN juhtmoodul, mille kaudu saab tudeng reguleerida LIN sõnumeid, mida saadetakse generaatori kontrollerile ja vastavalt veenduda, kas generaatori väljundpinge muutub. Stendi algasel kavandil ei ole näha tarbijaid koos lülititega ja mõõtmiste läbiviimiseks vajalike kontaktidega paneeli.



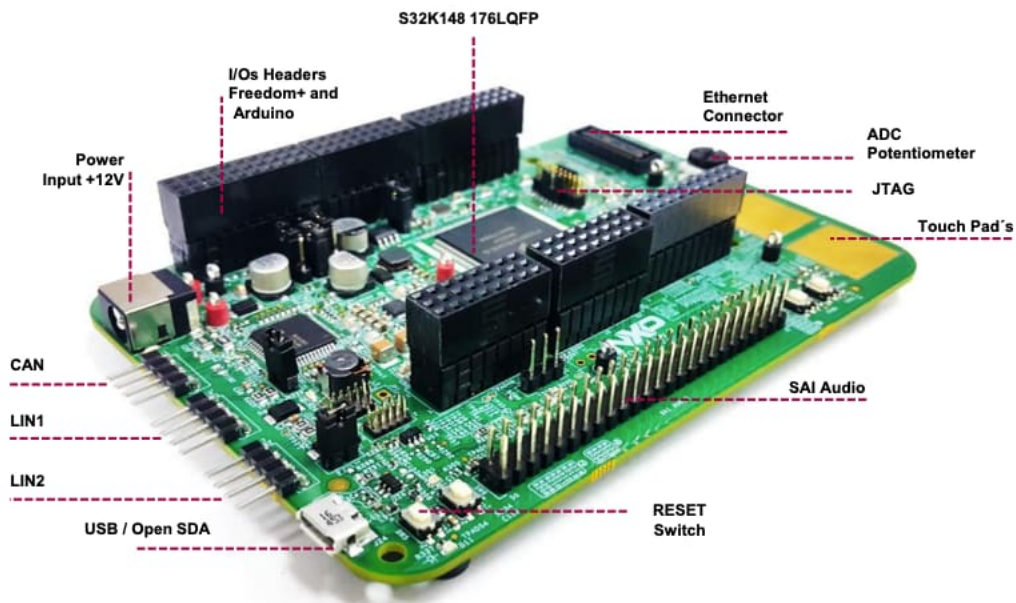
Pilt 3. Stendi algne kontseptsioon.

2.4. Riist- ja tarkvara

Sarnaselt generaatorile, oli valikus arendusplaat NXP SK32K148EVB-Q176, millel on LIN moodulina toimimise võimekus, mistõttu antud stendi loomiseks on see hea valik [26]. Sarnase võimekusega arendusplaatide tootjad on mitmeid, aga NXP kasuks rääkis ka mõningane varasem kogemus nimetatud arendusplaadiga töötamisel. Nimetatud arendusplaat on suunatud tarkvaraarendusele ja prototüüpimisele Arm Cortex-M4 kiibiga arendusplaat. Mille eesmärk on auto- ja tööstusrakendustele pakkuda võimalust uute rakenduste arendamiseks ja testimiseks.

Cortex-M4 tuuma, töösagedus on kuni 112 MHz ja sellega on võimalik lihtsalt ühendada väliseid juba varem mainitud protokolle kasutavaid seadmeid nagu CAN, LIN jne. Sarnaselt lihtsamate arendusplaatidega on sellel olemas ka mitmeid andurid, nagu kiirendusmõõtur, güroskoop ja magnetomeeter, mida saab kasutada liikumis- ja suunaanduritega seonduvate toodete arendamiseks, mis ei ole olulised küll käesoleva projekti raames, kuid annab aimu arendusplaadi laiast kasutusvõimalusest [26].

Antud arendusplaadiga tarkvara arendamiseks on kasutusel NXP S32 Design Studio ja S32 Software Development Kit (SDK), mis on terviklik arenduskeskkond ja suunatud NXP arendusplaatidele tarkvara või prototüüpide arendamiseks. Tarkvara põhineb avatud lähtekoodiga platvormil Eclipse IDE, mida kasutatakse tarkvara arenduseks väga mitmetes programmeerimiskeeltes ja erinevatele tehnoloogiatele [27].



Pilt 4. Kuvatõmmis arendusplaadi dokumentatsioonist [26].

Arenduskeskkond sisaldab kõiki vajalikke komponente tarkvara arendamiseks, et kirjutada lähtekood, seda samas kompileerida, kontrollida (*debug*) ja ka käivitada testimiseks emulaatoris. Ning lisaks on tarkvaral tugi, et valmis kood kanda üle arendusplaadile ja see käivitada testimiseks otse plaadil. Lisaks täisfunktsionaalsele tarkvara arenduse tööriistale sisaldab Design Studio for ARM ka AUTOSAR standardile vastava koodi genereerimiseks vajalikke mooduleid ja korrektse ülesehitusega näidisprojekte.

3. LIN MOODULITELE TARKVARA ARENDAMINE

3.1. Tarkvara arendusest sõidukitele üldiselt

Sarnaselt tavapärasele tarkvara arendusele toimib ka autonduses üldine loogika, et enne töö alustamist tuleb selgeks teha nõuded loodavale tarkvarale. Peamiselt on selles kontekstis oluline tarkvara funktsionaalsete nõuete kirjeldamine, mis on teiste sõnadega esialgne analüüsi etapis toimuv kirjeldus, kuidas programmikood peab toimima ehk millist sõiduki funktsiooni peab konkreetne loodav tarkvara täitma. Lisaks tuleb kirjeldada, millistes moodulites töö käib, kuidas programmi tegevused esile kutsutakse jne.

Sõiduki moodulitele koodi kirjutamine nõuab lisaks ohutuse ja ühtlustamise eesmärgil järgmiste standardite järgimist: AUTOSAR, BARR ja MISRA [15]. Mis kirjeldavad põhjalikult, kuidas peavad funktsionaalsed nõuded olema selged ja üheselt mõistetavad. Kood peab olema ühtlaselt struktureeritud ja selgelt loetav koos tekstisiseste kommentaaridega iga koodilõigu kohta, et vajadusel oleks sama lõik taaskasutatav. Igas etapis peab olema kogu arendustegevus ka dokumenteeritud, et vastavad muutused oleks korratavad ja vajadusel lihtsasti muudetavad.

Järgmise sammuna tuleks koodi kirjutamisel, mida üldjuhul tehakse tarkvaraettevõtetes, lähtuda konkreetsest funktsioonist. Alati tuleks koodi loomisel eelistada objektorienteeritud lähenemist, kus süsteemi käitav kood pöördub iga funktsiooni täitmiseks vastavalt funktsiooni ehk konkreetse objekti poole. Näiteks funktsioon: „salvesta(mida, kuhu)“, on toodud põhikoodis, aga mida täpsemalt tähendab „salvesta“ koodis, on täpsemalt kirjeldatud eraldi failis, kuidas see salvestamine täpsemalt peaks toimuma.

Lisaks objektorienteeritud lähenemisele on tänapäeval kasutusel ka koodihoidlad. Ühelt poolt tähendab selline hoidla turvalist (eeldame, et tarkvara tootja on rakendanud kõiki turvalisust tagavad meetmeid) kohta, kus mitme arendaja vahel koodi saab jagada või ka kolmandatele osapooltele või testijatele. Teisalt on koodihoidlatel üldjuhul võimekus ka tarkvara versioneerida, mis on lisaks koodi ajaloo automaatsele dokumenteerimisele ka mistahes probleemis kiirelt tagasi pöörduda versioonile, milles viga ei esinenud.

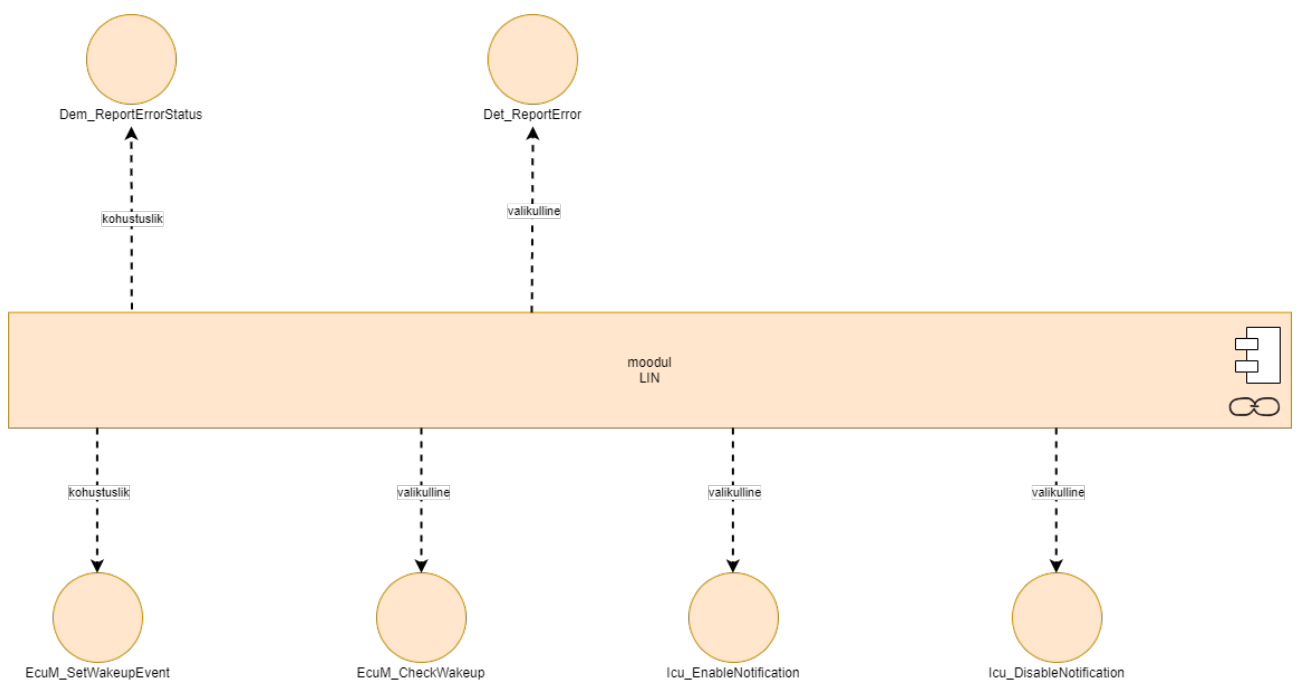
Järgmiseks koodi testimine, mis eriti autonduses toimub väga põhjalikult ja mitmes etapis. Testimine peab hõlmama testimist funktsionaalsuse osas ehk ühiktestimist, kus testitakse, kas funktsionaalsus käitub oodatult. Üldjuhul koostatakse iga testiühiku jaoks uus testistsenaarium, mis on automaatne, ehk samuti tarkvaraline osa, mis peab omakorda olema vastavuses kehtestatud nõuetega.

Kui funktsionaalsus üksikult on testitud, dokumenteeritud, vajadusel parandatud, uuesti testitud jne. siis tuleb testimist korrata terviksüsteemis ehk koostöös teiste moodulitega või samas moodulis koos teiste funktsionaalsustega. Lõpuks tuleb kogu testimine läbi viia ka sõiduki peal, kus selle toimet saab testida reaalses olukorras, eh sõidukiga sõites. [5]

3.2. UML skeemid LIN protsessi visualiseerimiseks

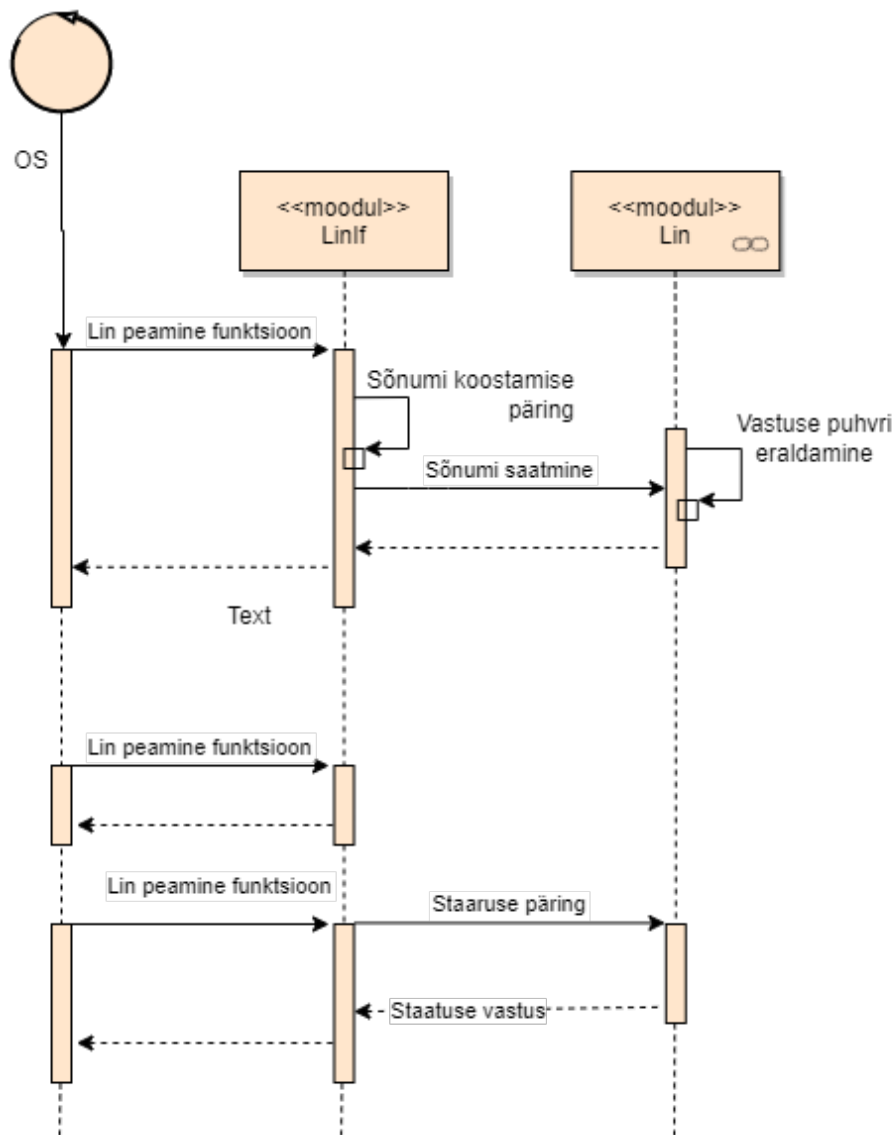
Tarkvara planeerimisel on hea tava alustada funktsionaalsuse loogika läbimängimist lisaks nõuete üldisele kirjeldamisele („*mida peab kood tegema?*“), ka UML skeemina, mis aitab suhteliselt lihtsalt ja soodsalt (sarnaselt prototüüpimisele visuaalse tarkvara loomisel) tarkvara funktsionaalsusi läbi mõelda ning vajadusel enne tarkvara kirjutamisega alustamist viia sisse vajadusel muudatused.

AUTOSAR on vastavad nõuded kirjeldanud kogu autoduse tarkvara arendusele [15], mis sisaldab nii nõuet sidumise tarkvara spetsifikatsiooniga, koos vastava osa kirjeldusega. UML kasutamise eeliseks on visuaalselt kirjeldada ära seosed koodi erinevate funktsionaalsuste vahel, koodis toimuvad tööprotsessid ja muu (Joonis 4), mis vajalik selleks, et mõista konkreetse osa tööd. UML on universaalne ja paindlik ka sellepärast, et selliste skeemide juures on lihtne mõista sisu kõigil – nii süsteemi arhitektil, disaineril kui ka arendajal [28].



Joonis 4. LIN mooduli UML Autosar spetsifikatsiooni järgi [15].

Lisaks on defineeritud järjestusdiagramm, mis näitab keerukamate koodiosade puhul täpsemalt, milline operatsioon millises järjekorras toimub ja kuidas see on seotud teiste protsessidega Joonis 5.



Joonis 5. LIN protokollide tegevuste jada UML.

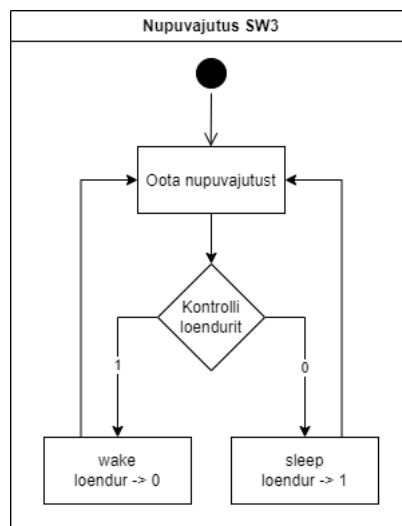
Kõige lihtsama loogikaga on seadme staatuste või tegevuste jada kirjeldavad diagrammid, milles vastavalt on näha kas mingi tegevuse järel seadme või funktsionaalsused staatus ning tegevuste jada puhul on vaadatud samad funktsionaalsused vastavalt tegevuste (sisemiste või välimiste alusel).

Käesoleva töö puhul on stendi tarkvara juhtmooduli funktsionaalsus piiratud ainult pinge seadistamisele, sest antud stendi loomisel eeldame, et laboritöös seadistatav pinge on ette antud ECU poolt, ning selle tõesuses ei ole põhjust kahelda, sest on võetud arvesse nii aku seisundit, mootori pöördeid ja muud olulised näitajad (sh Säästlik sõit), et seada soovitud pinge.

3.3. Tarkvara arendus ja koodinäited

Nagu selgitatud eelmises peatükis, siis esimene dokument on tarkvara arendusel nõuete kirjeldamise dokument, mille põhjal luuakse töövooskeemid (UML) ja sellele järgneb reaalne funktsionaalsuste programmikoodi loomine. Iga funktsionaalsus, mis koodina valmib ja mis ei ole otseselt juba täitur- või juhtmooduli koodi vajalik osa ja juba vastavuses AUTOSAR nõuetega, ehk kõik need koodi osad, mis kirjutatakse lisaks, tuleb vastavalt kommenteerida koodis ja märkida ära vastavalt koodi dokumentatsiooni dokumendis.

Teine ja autonduse tarkvara arendamise olulisimaid etappe on testimine, milles veendutakse, et funktsionaalsus toimib selliselt nagu planeeritud. Tagasi tulles plaanimise ja UML juurde, siis nende põhjalikkus annab ka testimisel ajavõitu, sest paljud võimalikud loogikavead on selliselt juba elimineeritud. Testimisel on paljude funktsionaalsuste puhul kasutusel ühiktestimine, kus automaattest ehk vastav tarkvara testib funktsionaalsuse toimet ja nii iga ühiku ehk funktsionaalsuse kohta eraldi. Täpne testloogika valik sõltub konkreetsest rakendusest. Ühiktestide tulemused ehk testimiskoodi poolt loodud raportid või muul moel tulemid vaatab üle ka inimene ehk testija. Antud töö skoobis testimist ei ole, aga siiski sai kõiki loodud funktsionaalsusi manuaalselt testitud ning need testid läbis loodud kood edukalt.

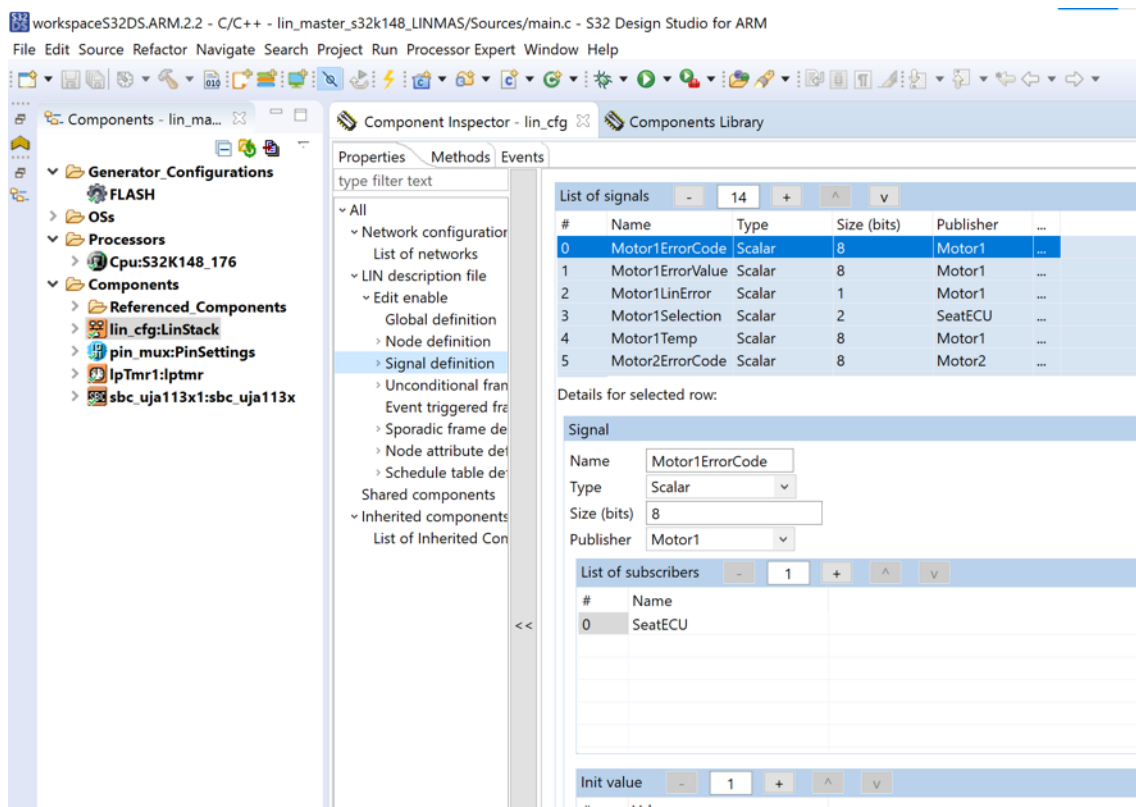


Joonis 6. UML diagramm nupu SW3 muutmiseks.

Näidiskoodi loomist on hea alustada, nagu selgitatud, loogikavigade vältimiseks, UML skeemist, millega saab lihtsustada esialgset funktsionaalsust visuaalselt, et näha, mis on ootused riistvara toimele ja seejärel alustada selle põhjal koodi funktsionaalsuste programmeerimist. Nii on protsesside

ja koodi funktsionaalsuste töö loogika ja järjekord kõigile protsessi osapooltele lihtsalt ja üheselt selge ilma süüvimata täpselt igasse koodis kirjutatud käsku või funktsionaalsuse dokumentatsiooni.

LIN protokollide parameetreid kirjeldab failide süsteemis LDF (*LIN description file*), milles on ära toodud kõik võrgu moodulid, protokollide kiirus, versioonid, kõikide signaalide mahud, raamide (*frame*) suurus ja sisu ning ajaline kestvus. Töös kasutatud NXP mooduli jaoks arendamise tarkvara koostab kogu LDF faili sisu automaatselt selle põhjal, mis kasutaja vastava mooduli juures seadistab (Pilt 5).



Pilt 5. Kuvatõmmis NXP S32 Design Studio tarkvarast.

Näidisoodi kirjutamisel on aluseks võetud arenduskeskkonnas leitavad näidisprojektid. Näited on üles ehitatud erinevate mootorite temperatuuri kontrollile ning vastava tagasiside andmisele näiteks erinevat värvi *led* diode kuvamise kaudu. Näiskoodi eesmärk on juhtida mootoreid kui täitureid nende pöörlemiskiiruse muutmisega vastavalt vajadusele ja muudele sisenditele sh ka temperatuur. Näidise põhjal oma lahenduse loomine lihtsustab tööd ja aitab võita aega, eriti kui varasem kogemus C-keele, konkreetse arendusplaadi või LIN sõnumi saatmise loogikaga. Need osad, mis panevad kokku korrektse LIN sõnumi ja saadavad selle õigel moel välja, on juba erinevate draiverite ja moodulite poolt näidisprojektis sees.

```

199 /*!
200 * @brief Interrupt Service Routine for buttons
201 * Depending on which buttons were pressed LIN scheme will be
202 * set to sleep mode or normal mode.
203 */
204 void BTNPORT_IRQHandler(void)
205 {
206     /* If SW3/Button 1 is pressed */
207     if (PINS_DRV_GetPortIntFlag(BTN_PORT_NAME) & (1 << PORT_BTN0_INDEX))
208     {
209         PINS_DRV_ClearPinIntFlagCmd(BTN_PORT_NAME, PORT_BTN0_INDEX);
210         l_sch_set(LI0, LI0_GOTO_SLEEP_SCHEDULE, 0u);
211
212         button_press_count++; // Increment button press count
213
214         if (button_press_count == BTN_PRESS_MAX) // If button pressed twice
215         {
216             PINS_DRV_TogglePins(LED_GPIO_PORT, PORT_LED1_MASK);
217             /* Switch to normal table */
218             l_ifc_wake_up(LI0);
219             l_sch_set(LI0, LI0_NormalTable, 0u);
220
221             button_press_count = 0; // Reset button press count
222         }
223     }
224 }

```

Pilt 6. Kuvatõmmis tarkvarast uuendatud koodiga.

Näidisprojekti muutmise UML skeem (Joonis 6) oli aluseks mille põhjal näidiskoodi muuta. Eesmärgiks oli senise SW3 (arendusplaadil nupp e *switch* 3) kaudu anda lisaks korraldusele „*uinuta*“ (*sleep*) samale nupule teine käsklus nii, et nupu teistkordne vajutus äratab LIN võrgu. Selleks lisati koodi projekti *Sources* kausta faili *main.c* failis maksimaalse nupuvajutuste lugeja väärtuse *BTN_PRESS_MAX*, mille väärtuseks on 2 ning uue muutuja *button_press_count*, mille kaudu saab salvestada nupuvajutuste viimast seisut ja selle algväärtuseks on 0. Järgmise vajutusega muudetakse nupu staatus algseks ja äratatakse (*wake*) LIN võrk – vastav kood on lisatud samas failis alates reast 195. Kuna näidiskoodide loomisel ei kasutatud koodi- või versioonihoidlat, siis kõik vana koodi töötavad osad on välja kommenteerituna lähtekoodis alles (**Error! Reference source not found.**) – a lgne kood jääb samas failis ridadele 225-230.

Järgmiseks eesmärgiks sai teisele arendusplaadi nupule (SW4 e *button* 2) olemasoleva normaaltoime asemel anda funktsioonile ülesanne pinge tõstmiseks rotatsiooni põhimõttel, kus iga nupuvajutus saadab välja uut tüüpi sõnumit, andes generaatori regulaatorile e täiturile ülesande tõsta pinget 0,5 volti iga vajutuse järel. Peale maksimumpingeni jõudmist (antud projektis 14,5 volti) algab järgmisel vajutusel sõnumi saatmine taas eesmärgiga saada väljund pingeks 10,5 volti jne. Nii saab saata LIN võrku erinevaid sõnumeid, mille reaalsel toimimist on võimalik jälgida esialgu näiteks pikoskoobi

(signaalide mõõteseade, töös oli kasutusel seade PICOSCOPE 2205A MSO koos tarkvaraga Pico7) kaudu ja samuti prototüüpides mõõta, kuidas muutub generaatori väljundpinge.

Mainitud loogika arenduse aluseks on samuti UML skeem, mis annab ülevaate ka neile, kes koodi sisu ei mõista või kellele selle toime programmikoodi tasemel ei ole relevantne. Aga lisaks on UML oluline osa dokumentatsioonist, kus algne soov analüüsi põhjal liigub kogu tarkvara elutsüklis testimisse ja lõpuks on osana dokumentatsioonist ka võimalik lisada toote kasutusjuhendisse, et toote rakendaja (autotootja) jaoks oleks iga komponendi lisamine tervikusse ja töövoos ühtlustamine sellevõrra lihtsam.

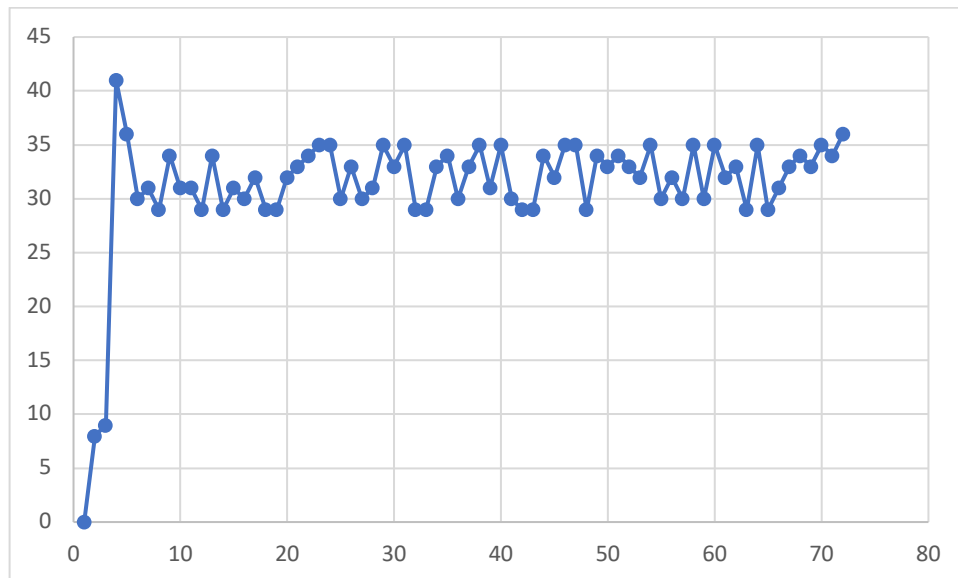
UML väljundpinge tõstmise loogika kuvamiseks, ehk regulaatorile suurema ergutusvoolu sõnumi saatmise loogika on sarnane eelmisele skeemile, selle erisusega, et iga nupuvajutuse järel saadetakse kontrollerile sammu võrra suurema ergutusvoolu väärtusega sõnum. Loogika tervikuna peaks veidi keerukam olema, aga kuna terviksüsteemi seni ei ole testida olnud võimalik, siis ei saa UML skeemi lisada kontrolle ja täiturmooduli ehk generaatori regulaatori poolt vastu saadav sõnum.

Antud programmi kõige keerulisem osa oli välja selgitada, milline peaks olema iga pinge tõstmise korral käsklus LIN sõnumi kujul, mis võrku saadetakse. Aluseks, et sõnumi sisu kokku panna, on võetud mitmed allikad. Varasema praktika käigus kogutud teave, milles töötava Toyota Proace sõiduki generaatori pistiku külge ühendatud sõnumite lugemisel saadi suur kogus andmeid, mis ECU välja saadab. Stendis asendab ECU-d NXP arendusplaat. Teine allikas on NXP AR6000 [12] pingeregulaatori dokumentatsioon, mille põhjal sõnumite loogika kokku panna. NXP probleemiks on see, et seade toetab väga mitmeid protokolle versioone. Abiks on ka video, mis on salvestatud Proace mõõtmisel, kus ekraanil on näha nii LIN sõnumid, kui ka sõiduki hooldustarkvara näidud, millel on näha pinge muutus aku klemmidel ja generaatori pingel.

Videotötlustarkvara DaVinci Resolve abil sai võetud referentsiks sõiduki käivitamisele järgnenud ajavahemik, kus pingel muutus ajas kõige kiiremini ja proovisin sealt samaaegselt saadetud sõnumeid jälgides mõista, millised on pingel reguleerimiseks suunatud sõnumid. Varasemalt olin sama mõõtmise ajal 3 salvestuse tulemused kokku pannud ja kogu mõõtmise ajal saabus ECU-lt generaatori regulaatorile viie erineva PID-ga sõnumeid: 37, 73, 97, A8 ja EC.

Siit edasi tekkis aga suluseis, sest neid raame analüüsid erinevate normatiivide vastu ei olnud näha selget loogikat, millisega neist peaks näidissõidukis olema juhitud väljundpinge. Kõik nimetatud PID-d on LIN põhjal vabad kasutamiseks mistahes soovitud skeemis sõidukis [29].

Järgnevalt analüüsisin videos kõiki enim infot andvaid PID-sid ca 40 sekundi vältel peale sõiduki käivitamist. Selle tulemusena saab kokkuvõtvalt öelda, et kõige loogilisem tundus PID 73, mille kolmanda baidi väärtus tundus olevat info, millega antakse regulaatorile ette ergutusvoolu suurus, kui sõiduk on käivitunud (Joonis 7).



Joonis 7. PID 73 3. baidi väärtused peale sõiduki käivitamist.

ECU võtab arvesse seejuures pinget aku klemmidel, temperatuuri akus ja ka ilmselt teises komponendis, millel ülekuumenemise oht vooluringis. Lisaks kindlasti tarbimisvõimust ja mootori pöördeid. Sarnaste teiste regulaatorite manuaalid ei anna samuti ette konkreetseid, PID ja sõnumi sisu infot, mida peab edastama andmeväljal, kuid teise tootja juhendis leidsin viite, et LIN 2.2A puhul toimub sõnumis ergutusvoolu juhtimine [20]. Et saada detailsem arusaam LIN sõnumi andmeväljast, tuleb sarnaseid mõõtmisi teha veel, et igat PID-d sõnumit täpsemalt vaadelda.

Eelmainitud info põhjal saab katseeksituse meetodi abil testimiseks kokku panna sõnumi, mis vahetab nupuvajutusel 73 PID sõnumis vastavalt soovitud ergutusvoolule. Siis saab juba konkreetsemalt mõõta, mis muutub, kui muuta mootori kiirust ja mis muutub kui tarbitav võimsus tõuseb. Et antud konteksti täpsemalt analüüsida on vajalik saada ligipääs konkreetsete moodulite juhendmaterjalidele ja eelkõige LIN tarkvara dokumentatsioonidele, mille põhjal saaks täpsemalt hinnata, kas ja millised sõnumi osad on olulised generaatori väljundpinge juures.

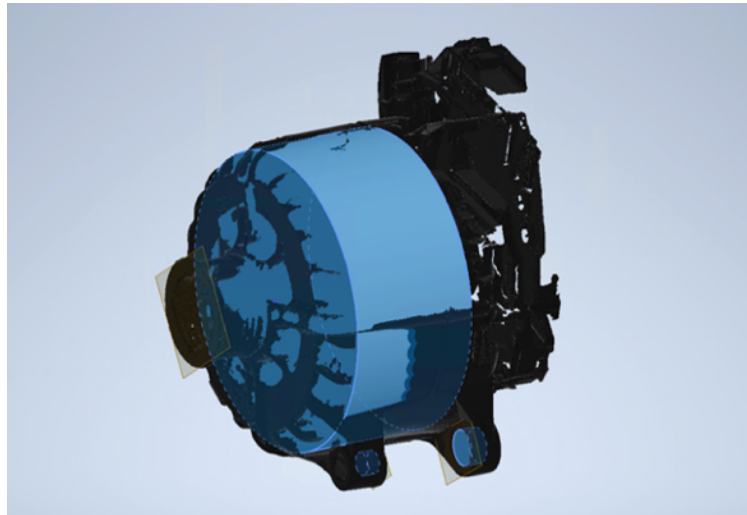
Antud kontekstis ei ole mõistlik tervet näidiskoodi, mis juhib mootorit, ümber teha. Kui algselt proovisin kuidagi muuta sõnumi elemente *lin_cfg.c* faili, siis sellisel kujul lahendust tööle ei saanud. Sain palju vigaseid kirjed vastavas *.h* failis, mis tekib kompileerimisel. Peale mõningast testimist ja

avastamist pöördusin tagasi tarkvaras asuva LIN seadistuste juurde ja seal sain muuta igas sõnumite tabelis saadetavaid sõnumeid. Nii sain esmakordselt mingil kujul plaadist välja uue sõnumi. Tõsi *SLEEP* sisse/välja lülitamine oli juba korras. Kasutasin *Switch Case* loogikat c-keeles, kus nupuvajutuse peale antakse moodulile uus käsk ja tema vastavalt edastab selle põhjal võrku uue sõnumi. Selleks loome eraldi funktsiooni, mis kasutab olemasolevaid LIN raamistiku valmiskomponente et sõnum korrektselt koostada ja edastada võrku.

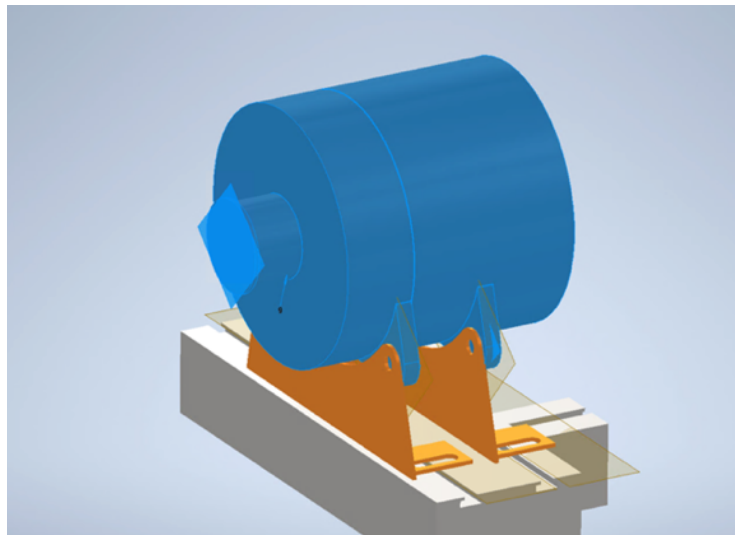
4. STENDI KOOST

4.1. Stendi komponendid

3D skaneerimisel saadud stendis kasutatava generaatori 3D mudel (Pilt 7) on aluses muude jooniste ja koostude disainiks. Selliselt on võimalik üsna täpselt saada kasutatava generaatori oluliste osade suurus ja paigutus, et disainida vajalikud lisadetailid koostule.



Pilt 7. Kuvatõmmis 3D skaneerimisel saadud generaatori mudelist.



Pilt 8. Kuvatõmmis projekteeritud kinnitusdetailidest.

Kuna visuaalselt oli näha, et generaatori kinnitamine samade kinnituskohtadega, millega ta on kinnitatud sõidukis, õppetendi külge, on keeruline, sai esmaseks ülesandeks teha vastavad ühendusdetailid (Pilt 8), millega ühendada generaator profiilidest raami – pildil oranžid.

Eesmärgiks oli luua võimalikult lihtne kinnitus selliselt, et see ühilduks maksimaalselt hästi valitud profiiliga (*MiniTec*), võimaldades sealhulgas kõikide osade lihtsat ringipaigutamist, aga samas ka koostu konstrueerimine tervikuna oleks võimalikult lihtne.

Vedava elektrimootori kinnitus, mis peab analoogselt olema kohandatavate kinnitustega, et rihmülekanne saaks toimida ühel joonel üle erinevate pöörlevate osade. Varasema kogemuse põhjal kliimastendi ja käigukasti stendi koostude pealt (Pilt 9 ja Pilt 10) saab komponentide koguse ja tervikliku mahu tõttu stendi lahendada nagu varem mainitud, lauale paigaldatavana. Olenevalt koostu stabiilsusest täispöoretel, tuleb edasise töö käigus otsustada, kas alusaam võib olla lihtsalt laud või muu alus või vajab stand jäigemal ja stabiilsemal alusraami, kui tavaline laud.

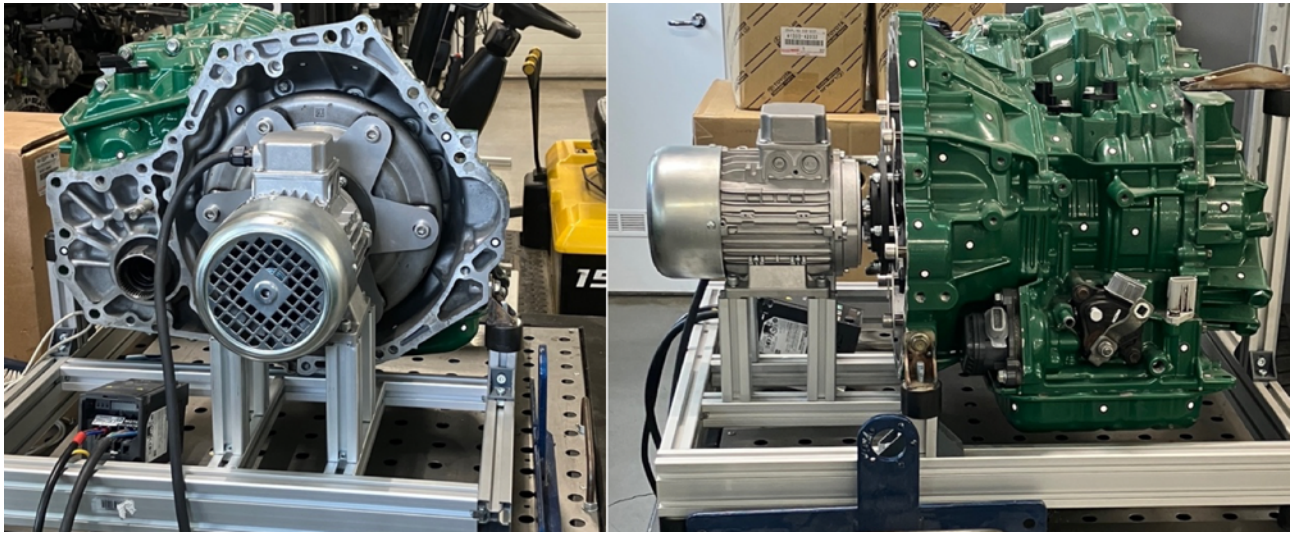


Pilt 9. Sõiduki kliimaseadme õppestand.

Mootori, kui kõige raskema detaili kinnituse raamile peaks lahendada nii, et profiilid on seatud vastavusse mootori vertikaalsete suunaga kinnitusavadega. Juhtpaneeli loogika on esialgu kavas lahendada laual lahtiselt asuva arendusplaadina, millele asuvate lülitite abil saab sõnumiedastust kontrollida. Selleks, et sõnumeid visuaalselt näha, on võimalik algselt kasutada pikoskoopi [31], mille tarkvara abil saab arvutist jälgida sõnumi kuju ja sisu, mis võrgus liiguvad

Lauapealsena, sarnaselt olemasoleva käigukasti stendi koostuga, on võimalik õppuril ka ergonomilises mõttes mugav laboritöö läbi viia. Ülekande lahendamisel rihmaga on stendi algses

disaini lisaks ergonoomikale ohutuse mõttes jäetud kõik liikuvad osad nii, et need võimalikult vähe saaks ohustada laboritööd läbiviijat.



Pilt 10. Sõiduki käigukasti õpepeend.

Kuna töö koostamise hetkel ei ole olemas sobilikku elektrimootorit, mida kasutada stendi generaatori veomootorina, saab edasise arvutusloogika kokku panna üsna üldiselt ja see vajab testimist ja vajadusel üle arvutamist, et saavutada soovitud toime. Veomootori pöörete vahemiku ja võimsuse arvutamise aluseks on erinevate osade võimsustarve.

Et mitte stendis voolu mitte liiga kõrgeks ajada, on hea kasutada näiteks kolme erineva võimusega tarbijat vooluvõrgus, et nende koos ja eraldi lülitamise kaudu jälgida, kuidas muutuvad vooluvõrgus vool, pinge ja milline on LIN sõnum. Koormuste lihtsaim lahendus on esmalt kasutada stendile kinnitamiseks 2 erineva võimusega 12 voldise nimipingega lampi või näiteks ühte tule moodulit, kus käib näiteks H4 tulepirn, millel on 2 võimsuse astet ja lülitades n.ö lähi- ja kaugtulede vahel on tarbitava võimsuse muut juba ca 50 vatti.

Kolmanda elemendina võiks olla kasutusel näiteks 200 w võimsusega elektriline kütteseade vms. Nii saab läbi erinevate lülituskombinatsioonide tõsta stendi vooluvõrgus võimuse vajaduse ca 300 vatini, mis annab amprites maksimaalseks voolutugevuseks näiteks 16 voldi puhul:

$$I = \frac{P}{V}; \text{ millest } \frac{300w}{16v} = 18.75 A \quad (4)$$

Regulatsioonide põhjal on vajalik varutegur 1,5 korda ja seega peab kasutatavate kaabelduste juures arvestama vastavalt iga ühendatava koormus nimivoolu ja kogu süsteemi maksimaalset voolu. Kuna

stend on üsna kompaktne, siis pingelangu kaablis saab hinnata näiteks maksimaalselt 2m pikkuse kaabli põhjal. Loogiliselt võiks ühendamisest lahendada nii, et tarbijateni jõuab akult üks kaabel ning vastavast lülitite moodulist, läheb iga tarbijani vastavalt tarbija võimusele valitud diameetriga kaabel. Võttes aluseks, et kaabli paigaldus on selline, et ei teki olulist ülekuumenemise ohtu vaba õhu juurdepääsu puudumise tõttu ja kasutada vaskkaablit, mille eritakistus (ρ) on $1,78 \times 10^{-8} \Omega m$, kaabli pikkus 2m, maksimaalse voolu suuruseks koos varuteguriga (1,5), saame $1,5 \times 18,75 = 28,13A$.

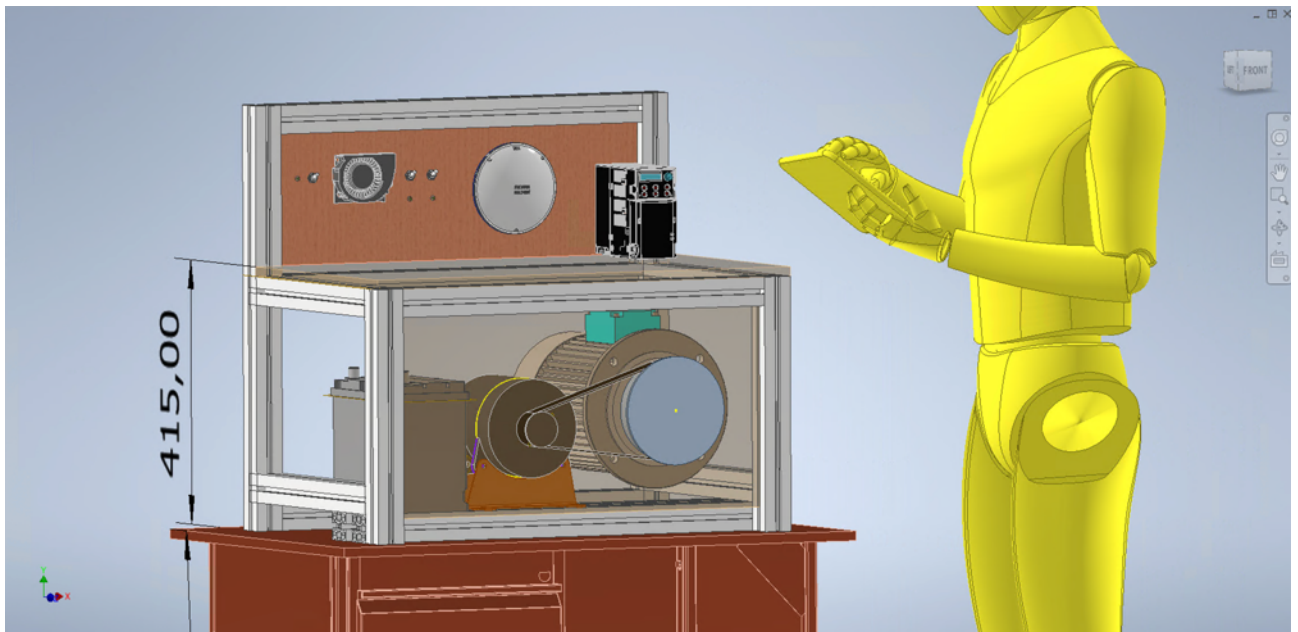
Vaskkaabli eritakistuse põhjal saab hinnata, et tarbijatele kasutatavad kaablid peavad olema vähemalt 2,6 mm läbimõõduga. Stendi koostu tuleb sarnaselt sisse arvutada generaatori enda kaabelduse nõuded, mis peavad vastama generaatori enda spetsifikatsioonile, sest generaator on ise tunduvalt suurema väljundvõimsusega, kui stendi planeeritud tarbijad. Kõikide tarbijate lülitid peavad samuti vastama iga tarbija maksimaalsele voolule. Lisaks tuleb iga tarbija juurde lisada vastava nimivoolu mahus kaitse, mis võimaldab lisaks muudele oskustele omandada tudengitel ka oskuse, kuidas vooluringis probleeme defekteerida. Stendi mõistes on see tagurpidi kaitseks tarbijatele, kui pinge peaks stendis tõusma liiga kõrgeks. Kaitsmed võivad asuda stendil iga seadme lüliti juures, et neid oleks võimalik ka visuaalselt hinnata.

Generaatori ringivedamiseks on tellitud MT 2,2 KW võimsusega elektrimootor koos Delta MS300 seeria vastava sagedusmuunduriga. Mootoril on võimalik sagedusmuunduri abil reguleerida pöördeid. Esialgu on elektrimootori pöördeid võimalik muuta otse kontrollierist, kuid hilisemas arenduses on see mõislik siduda stendi juhtmooduliga, kus pöördeid reguleerib ECU ja generaator arvestab pööretele vastava väljundpinge (ergutusvoolu) seadistamisel. Esimeses etapis võib mootori kiirus olla konstantne, kuid peab antud pööretel andma vajaliku väljundvõimsuse (min 30A).

Mootori valiku teine kriteerium lisaks nimivõimsusele ja pöörlemise reguleerimisele on pööremiskiiruste vahemiku ja võimsuste suhe. Pöörlemiskiiruse ülekandmisel generaatorile tuleb arvestada, et mootor oleks generaatorile vajalike pöörete juures, mis on üldjuhul mootori suhtes 1:1,5 kuni 1:3, mis tähendab, et generaator vajab mootoris suuremat kiirust. Mootori valikul on seega vaja hinnata, kas valida kohe sobiva pöörete ja võimsuse vahemikuga mootor või arvestada sobitamisel ülekannete loomisega vajalike pöörete ja võimsuse suhtega.

Ülekande kaudu peab seega generaatori kiiruse saama vähemalt esimeses faasis ca 1000 RPM minimaalselt või ka rohkem, et saaks alustada vastavalt ergutusvoolu ja väljundpinge suhete testimist. Stendi koostus lõpplahenduses tuleb vajadusel teha sellised muudatused, et vastav ülekanne oleks võimalik etteantud pöörete ja võimsuse suhte vahemikes. Kuigi algsel eskiisil on raami osal näha

ainult alumine koost, siis sarnaselt tuleb jäikuse mõttes lahendada kogu raam, et see moodustaks kuubi, mille tagumised postid on tasapinnast kõrgemal ja nende külge saab kinnitada tagaseina, millele kinnitada omakorda kontrollid, tarbijad ja ka mõõtekontaktid (Pilt 11).



Pilt 11. Kuvatõmmis stendi koostu lõppdisainist.

Stendi tasapinnaks ja seinteks, mis jäävad kasutaja poole, on hea kasutada kas sobivat vineerplaati või pleksiklaasi, kui kasutajale peaks õpioskuste omandamiseks on oluline näha, kuidas koost toimib, aga kaetud pöörlevad osad lisavad kindlasti stendile turvalisust. Viimase disaini juures on lähtunud sellest, et tudeng näeks reaalselt pöörlevate osade liikumist. Disaini viimases versioonis on arvestatud juba tellitud mootori ja regulaatoriga (sagedusmuunduriga).

Sellega on peamised stendi komponendid üles loetud, et mootori, ülekande ja generaatori kaudu toota tarbijatele, mida on võimalik sisse ja välja lülitada, voolu. Järgmises peatükis pisut veel LIN võrgu moodulitest. Muudest komponentidest on mainimata veel aku, mille valik ei ole hetkel selge. Eesmärk on aku valikul anda stendile veel üks mõõde, kus tudeng saab kõiki mõõtmisi stendil teha ka ilma veojõu mootorit käivitamata. Hetkel ei ole planeeritud küll sellist osa, kus stend imiteeriks „süüde sees“ seisu, aga hiljem saaks selles seisus stendis seadistada generaatori juhtmoodulit ilma käsuta ergutusvoolu tootma, ehk siis peaks kontrollid iseseisvalt olema valmis, et kui starter mootorit ringi ajab, ka ise vooluringi lisavõimsust anda.

Eelneva põhjal peaks aku võimsus olema selline, et suudaks ilma generaatori lisatoiteta katta tarbijate vajadused vähemalt $\frac{1}{2}$ - 1 h jooksul. Meie 300 vatise ja veidi rohkem arvutusliku võimsustarbe juures

võiks selline maht olla minimaalselt 12,5 Ah, aga mõistlik oleks varuga kasutada vähemalt 25Ah ehk üsna tüüpilist 12v mootorratta akut. Selline aku annab võimaluse stendil uurida ja kõiki näitajaid ja ka vastavale seisule programmeeritud LIN sõnumeid jälgima.

4.2. Stendi moodulid

Nagu varasemates punktides mainitud, siis stendi peamine juhtmoodul on NXP arendusplaat, millele kirjutatud tarkvara abil saab muuta generaatori pingeregulaatorisse saadetavat LIN sõnumit. Algses arendusfaasis ei ole mõistlik kasutusele võtta keerukaid lülite loogikat. Kasutada on võimalik plaadi olemasolevaid lüliteid, et reguleerida sõnumite väljastamist.

Mõõtekontaktid on algselt võimalik lahendada nii, et tudeng saab mõõtmisi teha näiteks otse aku klemmidelt - pinge. Sõnumite lugemiseks vajalikud ühendused saab teostada otse täiturmooduli ühendamisel prototüüpimise kaablite kaudu, kus vastavad moodulid on omavahel ühendatud min 1mm läbimõõduga kaablitega ning sama ühenduse vahele on ühendatud ka pikoskoobi mõõteotsikud.

Kogu stendi maanduse osa vajab täpsemat läbimõttlemist, sest generaatoris asuv täiturmoodul ei oma eraldi maandusliini vaid kasutab sõiduki maandust. Mis tähendab, et kogu õppestendi loogikas peab maandus olema lahendatud tervikuna, generaatorile, tarbijatele ja juhtmoodulile.

Täiturmooduliks algses kontseptsioonis on kindlasti ainult generaatori korpuses asuv reguleerimismoodul, mis tuleb ühendada LIN võrku juhtmooduliga. Nagu varasemalt mainitud, saab stendi hiljem vajadusel täiendada ka mootori juhtimise osaga, kus LIN juhtmoodul jälgib mootori pöördeid ja vajadusel näiteks imiteerides seisu „käik väljas“ nõuda mootorilt kõrgemaid pöördeid, et laadida kriitiliselt madala laetuse tasemega akut vms. Hetkel ei sisaldu ka see loogika terviklahenduses, aga järgnevalt peaks juhtmoodul saama mingil moel infot aku pinge ja muude aku omaduste kohta. See eeldab võrku lisamooduleid või siis juhtmooduliga lisaandurite ühendamist, mis annaks vajadusel vastavat infot ning lisaarendust, et juhtmoodul pinge juhtimisel arvestaks ka neid parameetreid.

4.3. Näidiskava praktikumi läbiviimiseks

Stendi kasutava tudengi jaoks on täidetud õpiväljundid, mis katavad ühelt poolt sõiduki elektrisüsteemi toime ja teisalt saavad nad jälgida, kuidas moodsamates sõidukites pinge reguleerimine käib vastavate LIN sõnumite kaudu, mille sisu omakorda genereerib sõiduki juhtmoodul või konkreetse võrgu juhtmoodul vastavalt kõigile sisendparameetritele.

LIN juhtimisega generaatori regulaatori tööpõhimõtet saab tudeng samuti stendil testida, mõõtes erineva sõnumi korral kontrolleri poolt saadava väljundpinget / ergutusvoolu generaatori mähistel või harjadel. Lisaks saab stendi kasutaja harjutada stendiga mõõteseadmete ühendamist etteantud juhendi põhjal. Siin tuleks enne valmistoote kasutusse andmist kindlasti läbi mõtleva võimalikud veakohad ja looma võimalikud takistused või kaitsemehhanismid, et vältida vigastusi või stendi osade rikkumist, mis võivad tekkida eksitusest või ebapiisavast juhendiga tutvumisest.

Stendi generaatori väljundpinge, LIN sõnumi ja ergutusvoolu suuruse seose omandamise jätkuks saab õppija lisaks jälgida, kuidas voolutarve ehk ahelasse ühendatud tarbijad mõjutavad pinget ja voolu stendi ahelates. Selliselt saab õppija testida, kas regulaator hoiab mistahes koormuse korral vajalikku ergutusvoolu, et tagada alati etteantud pinge.

Nagu mainitud eelmises peatükis, saab edasiarenduses lisada mootori pöörlemiskiiruse jälgimise, järgimise ja muutmise, mille ajal regulaator peab suutma hoida vajalikku pinget aga ka samas saab pöördeid tellida vms. Aga tervikpildis peaks tudeng saama aru, miks ja kuidas mistahes koormuse ja mistahes pöörete juures tagatakse reguleerimise kaudu vajalik pinge sõiduki elektrisüsteemis. Samuti tuleb aku juurde või väljavõtuna stendil tekitada võimalus voolu muutuse järgmiseks. Ka selle ahela võib edasiarendustes lisada süsteemi nii, et ka juhtmoodul saaks aru tarbimisvõimsuse muutusest.

Näidisülesanded tudengile:

1. Mõõda pinge aku klemmidel ja märgi see töölehele
2. Mõõda ampertangidega voolutugevus vooluringis ja märgi see töölehele
3. Käivita veomootor pööretele N.
4. Mõõda uuesti pinge aku klemmidel ja vool ahelas ja märgi tulemused töölehele.
5. Tõsta juhtmooduli abil süsteemis ergutusvoolu N amprini ja veendu, et LIN sõnumi sisu pikoskoobis vastaks programmile.
6. Teosta uuesti pinge ja voolu mõõtmised ja märgi need töölehele.
7. Lülita ahelasse tarbija ja teosta kõik mõõtmised uuesti, märgi tulemused töölehele.
8. Põhjenda, mis on muutunud ja miks.
9. Suurenda ergutusvoolu ja teosta veelkord kõik mõõtmised ja märgi töölehele.

Nagu varasemalt mainitud saab vastavalt edasisele stendi arendusele laiendada ka vastavalt tudengile pakutavaid tegevusi stendil. Keerukamate funktsionaalsuste arendamisega (pöörete jälgimine, pöörete muutmise, aku jälgimine, pinge ja aku olukorra jälgimine) muuta stendi üsna sarnaseks reaalse sõiduki pingeregulatsiooni süsteemiga. Kindlasti saab lisada ka võimaluse teise sõiduki

generaatori kontrollimiseks või juhtimiseks, mis vajab vastavate ühenduste disaini. Kindlasti on hea lisada erinevaid väljavõtte, kus mõõtmisi saaks teha ka otse generaatori kontrolleriilt, et mõõta ergutusvoolu otse rootori harjade juurest. Suureks miinuseks antud juhul on see, et lõpliku juhtimisloogika väljatöötamine vajab iga generaatori korral nende spetsiifika tundmist, mis ka antud projekti juures on seni suur väljakutse, sest dokumentatsioonid ei ole avalikud ja puudub ka kogemus varasemalt sarnaste koodide väljaselgitamise või andmebaasidest otsimisega.

KOKKUVÕTE

Käesoleva lõputöö eesmärk on LIN juhtimisega generaatori õppetendi loomise protsessi kirjeldamine, analüüs ja näidiskava loomine laboritöök. Töös on välja toodud erinevad õppevahendi loomise eripärad, kui selline seade on suunatud harjutamiseks ja LIN signaali põhimõtete omandamiseks tudengitele.

Töös antakse ülevaade LIN ja teiste sarnaste protokollide ajaloost ja toime loogikast, mille teadmine on LIN juhtimise õppimisel oluline alus, et saada aru juht ja täitur loogikast, sõnumite edastamise füüsilisest kihist ja kiirustest. Stendi loomisel on võetud suund vältida valmistoodete hankimist, et anda tudengitele, sealhulgas käesoleva töö autorile võimalus detailselt süsteemidest aru saada ja saada ning info dokumenteeritult ka edasi anda.

Töös kirjeldatud prototüübi jaoks valiti NXP arendusplaat ja Toyota Proace generaator. Hetkel on puudu reguleeritava kiirusega mootor ja tarbijad, ent sellele vaatamata saab loogilised osad kokku panna. Varasemalt on sarnase generaatori peal läbi viidud mõõtmisi, mille põhjal oli käesolevas töös vajalike programmide koostamine pisut kergem, sest oled olemas mõningad algandmed, mis viitavad kuidas antud toode täitur moodulina toimib. Siiski võib öelda, et tarkvara arenduse protsess autonduses on suhteliselt keeruline, kui iga moodul jaoks ei ole võtta täpset juhust. Hea on mõista, kui põhjalikult ja täpselt on reguleeritud tarkvara arendus autonduses kasutatavatele moodulitele.

Töös kirjeldatud viisil stendi loomine on mitmekordselt kasulik andes õppimisvõimaluse nii stendi valmistajale, juhendajatele kui ka järgmistele tudengitele. Lisaks saab sellise arendustöö käigus rakendada mitmeid õppimise ajal omandatud oskusi. Näiteks joonestamist, 3D modelleerimist aga ka reaalselt praktilist tööd seadmete valmistamisel, materjalide valikul ja koostu kokkupanekul.

Lõputöös on kompaktselt kokku pandud mitmed distsipliinid alates toote füüsilisest disainist ja lõpetades riistvaraliste komponentide valikult ning tarkvara arendamisele oluliste aspektide uurimine alustades nõuete kirjeldamise ja UML jooniste kasutamisest kuni testimise ja käivitamiseni.

Kuna stend ei saanud lõpuni valmis, siis on oluline sisend ka stendil õppetöö läbiviimiseks praktikumi kava, mis on heaks aluseks õppe, läbiviimisel, ei ole kindlasti lõplik, aga käesoleva töö näol on kindlasti olemas väga põhjalik alusmaterjal õppetöö jätkamiseks.

SUMMARY

The objective of current thesis “LIN-Controlled Alternator Stand for Education” is to describe, analyze, and create a sample prototype for working in laboratory environment and to use as a training stand for the LIN-controlled alternator. The paper highlights the unique features of designing a training tool for students in practice and how to learn about LIN signal principles.

When trying to understand more about the LIN-controlled systems, for understanding the operational logic of controllers and actuators, the physical layer, and message transmission speeds, it is crucial to have a basic grasp of the history and operational logic of LIN and other protocols that are used in the automotive industry. To provide students, including the author of this work, the chance to comprehend the systems in detail and to document and pass on the information in an organized way, the stand was constructed from parts instead of buying already assembled and ready to use stand.

A Toyota Proace alternator and an NXP development board were chosen for the prototype. The variable speed motor and controller are currently missing, but the logical components can still be assembled. The construction of the essential programs in this study was slightly facilitated by measurements previously performed on an alternator of a similar design since some preliminary data was suggesting how the product functions as an actuator module shall work. However, if there isn't a detailed manual provided for each module, it may be claimed that the software development process in automobile engineering is rather complicated. However, it's critical to comprehend how stringently and precisely the creation of software for modules utilized in automobile engineering is governed.

In addition to offering learning opportunities for the stand builder, teachers, and future students, creating a stand in such way as outlined in the article has several advantages. Additionally, a variety of learning-related abilities, including sketching, 3D modeling, as well as practical work in equipment production, material selection, and assembly, can be used during such development work.

The investigation of significant aspects of software development, beginning with the description of requirements and the use of UML diagrams and ending with testing and launch, are all brought together concisely in the thesis. These disciplines range from the physical design of the product to the selection of hardware components.

The practical training plan for the stand, which should serve as the foundation for delivering the training, is undoubtedly insufficient because the stand was not finished after the work for a variety of reasons. However, this work offers a solid platform on which to build future studies.

KASUTATUD ALLIKAD

- [1] K.-h. Dietsche ja K. Reif, *Automotive Handbook*, Karlsruhe: Robert Bosch GmbH, 2018.
- [2] I. Koren ja M. Krishna, „Fault-tolerant systems,“ *Elsevier Science & Technology*, 2007.
- [3] C. Ünsalan, H. D. Gürhan ja M. E. Yü, *Embedded System Design with ARM Cortex-M Microcontrollers: Applications with C, C++ and MicroPython*, Springer International Publishing, 2023.
- [4] *Road vehicles -- Local Interconnect Network (LIN) -- Part 8: Electrical physical layer (EPL) specification: LIN over DC powerline (DC-LIN) ISO 17987-8:2019*, 2019.
- [5] N. Navet ja F. Simonot-Lion, *Automotive Embedded Systems Handbook*, New York: CRC Press Taylor & Francis Group, 2009.
- [6] K. Van Cleave, „A Survey of Automotive Ethernet Technologies and Protocols,“ 12 2019. [Võrgumaterjal]. Saadaval: https://www.cse.wustl.edu/~jain/cse570-19/ftp/auto_eth/index.html. [Kasutatud 02 04 2023].
- [7] „Infineon,“ [Võrgumaterjal]. Saadaval: <https://www.infineon.com/cms/en/applications/automotive/chassis-safety-and-adas/sensor-fusion/?redirId=67681>. [Kasutatud 4 3 2023].
- [8] CVE, „CVE,“ 05 04 2023. [Võrgumaterjal]. Saadaval: <https://www.cve.org/CVERecord?id=CVE-2023-29389>. [Kasutatud 12 04 2023].
- [9] F. Páez ja H. Kaschel, „Design and Testing of a Computer Security Layer for the LIN Bus †,“ %1 2021 *IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Valparaíso, Chile, 2021.
- [10] Department of Electronics and Communication, Nitte Meenakshi Institute of Technology, Bengaluru, India, „Comparison of CAN, LIN, FLEX RAY and MOST In-vehicle bus protocols,“ *JETIR*, kd. 6, nr 4, pp. 160-165, 2019.
- [11] E. Hackett, „Application Note LIN Protocol and Physical Layer Requirements,“ [Võrgumaterjal]. Saadaval: https://www.ti.com/lit/an/slla383a/slla383a.pdf?ts=1679582025211&ref_url=https%253A%252F%252Fwww.google.com%252F. [Kasutatud 23 03 2023].
- [12] „Alternator Regulator with LIN,“ [Võrgumaterjal]. Saadaval: <https://www.nxp.com/products/power-management/motor-and-solenoid-drivers/powertrain-and-engine-control/alternator-regulator-with-lin:AR6000>. [Kasutatud 19 03 2023].
- [13] M. Falch, „LIN Bus Explained - A Simple Intro [2023],“ [Võrgumaterjal]. Saadaval: <https://www.csselectronics.com/pages/lin-bus-protocol-intro-basics>. [Kasutatud 19 03 2023].

- [14] „LIN Bus Explained - A Simple Intro [2023],“ CSS Electronics, 2023. [Võrgumaterjal]. Saadaval: <https://www.csselectronics.com/pages/lin-bus-protocol-intro-basics>. [Kasutatud 12 03 2023].
- [15] AUTOSAR Foundation, „Specification of LIN Interface,“ 28 11 2019. [Võrgumaterjal]. Saadaval: https://www.autosar.org/fileadmin/standards/classic/19-11/AUTOSAR_SWS_LINInterface.pdf. [Kasutatud 23 03 2023].
- [16] A. Ariste, Digitaalseadmete aritmeetika ja loogika I. Loengukonspekt, Tallinn: TPI rotaprint, 1971.
- [17] I. Costachescu, „101: Local Interconnect Network (LIN),“ [Võrgumaterjal]. Saadaval: <https://community.nxp.com/t5/Blogs/101-Local-Interconnect-Network-LIN/ba-p/1284877>. [Kasutatud 02 03 2023].
- [18] M. Katsan, Elektrimasinad ja transformormaatid, Tallinn: Valgus, 1971.
- [19] Franck, „ALTERNATOR REGULATION : What are the different communication protocols on the market ?,“ [Võrgumaterjal]. Saadaval: <https://www.testmyalternator.com/en/news/technical-articles/252-different-communication-protocols-regulation-alternators-tma-technical.html>.
- [20] „Alternator voltage regulator with LIN interface for 12 V applications,“ [Võrgumaterjal]. Saadaval: <https://www.st.com/resource/en/datasheet/19918.pdf>. [Kasutatud 24 04 2023].
- [21] T. Denton, Automobile Electrical and Electronic Systems, Routledge, 2013.
- [22] S. Garrett, „High Output: A Look at Today’s Charging Systems,“ [Võrgumaterjal]. Saadaval: <https://gearsmagazine.com/magazine/high-output-a-look-at-todays-charging-systems/>.
- [23] Consulab, „Automotive Service Technology,“ Consulab, [Võrgumaterjal]. Saadaval: https://consulab-prod-contenu.s3.amazonaws.com/Media/catalogue/EM-CAT54-EN_099105_WEB.pdf. [Kasutatud 12 03 2023].
- [24] Thepra LIN juhtimisega generaator, „Generator with LIN bus controller,“ [Võrgumaterjal]. Saadaval: <https://www.thepra.de/Generator-with-LIN-bus-controller>. [Kasutatud 02 04 2023].
- [25] „TITANE tester by Intitek,“ [Võrgumaterjal]. Saadaval: <https://www.testmyalternator.com/en/products/titane-tester-heavy-duty-automatic-alternator-starter-test-bench.html>. [Kasutatud 23 03 2023].
- [26] National Research Council., How People Learn: Brain, Mind, Experience, and School: Expanded Edition., Washington, D.C.: The National Academies Press., 2000.
- [27] „S32K148-Q176 Evaluation Board for Automotive General Purpose,“ [Võrgumaterjal]. Saadaval: <https://www.nxp.com/design/development-boards/automotive-development->

platforms/s32k-mcu-platforms/s32k148-q176-evaluation-board-for-automotive-general-purpose:S32K148EVB. [Kasutatud 26 03 2023].

- [28] „Desktop IDEs,“ [Võrgumaterjal]. Saadaval: <https://www.eclipse.org/ide/>. [Kasutatud 26 03 2023].
- [29] A. Nyßen ja P. Könemann, „Model-based Automotive Software Development using Autosar, UML, and Domain-Specific Languages,“ %1 *embedded world Conference* , Nürnberg, Germany, 2013.
- [30] „LIN Specification Package Revision 2.2A,“ [Võrgumaterjal]. Saadaval: https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf. [Kasutatud 23 04 2023].
- [31] „LIN bus decoder and LIN controlled alternators,“ [Võrgumaterjal]. Saadaval: <https://www.picoauto.com/library/training/lin-bus-decoder-and-lin-controlled-alternators>. [Kasutatud 30 04 2023].
- [32] „Generator with LIN bus controller,“ [Võrgumaterjal]. Saadaval: <https://www.thepra.de/Generator-with-LIN-bus-controller>. [Kasutatud 12 03 2023].

LISAD

1. Muudetu programmikood
2. Generaatori kinnituse joonis
3. Stendi koostu joonis

Lisa 1. Programmikood

```
/*DEFINEERITUD VAJALIKUD LISAMUUTUJAD*/
#define BTN_PRESS_MAX 2
#define COMMAND1 1
#define COMMAND2 2
#define COMMAND3 3
volatile uint8_t button_press_count = 0;
volatile uint8_t button2_press_count = 0;
volatile uint8_t currentcommand = 0;
/*DEFINEERIMISE LÖPP*/

/*!
 * @brief MUUDETUD NUPPUDE VAJUTAMISE LOOGIKA
 */
void BTNPORT_IRQHandler(void)
{
    /* If SW3/Button 1 is pressed */
    if (PINS_DRV_GetPortIntFlag(BTN_PORT_NAME) & (1 << PORT_BTN0_INDEX))
    {
        PINS_DRV_ClearPinIntFlagCmd(BTN_PORT_NAME, PORT_BTN0_INDEX);
        l_sch_set(LI0, LI0_GOTO_SLEEP_SCHEDULE, 0u);
        PINS_DRV_WritePin(LED_GPIO_PORT, PORT_LED1_INDEX, 0U);
        PINS_DRV_WritePin(LED_GPIO_PORT, PORT_LED0_INDEX, 0U);
        PINS_DRV_WritePin(LED_GPIO_PORT, PORT_LED2_INDEX, 0U);

        // SIIN SUURENDATAKSE NUPUVAJUTUSE LOENDURIT
        button_press_count++;

        // KUI NUPPU ON VAJUTATUD 2 KORDA, NULLITAKSE LOENDUR
        if (button_press_count == BTN_PRESS_MAX)
        {
            l_ifc_wake_up(LI0);
            l_sch_set(LI0, LI0_NormalTable, 0u);
            button_press_count = 0;
        }
    }

    /* If SW4/Button 2 is pressed */
    if (PINS_DRV_GetPortIntFlag(BTN_PORT_NAME) & (1 << PORT_BTN1_INDEX))
    {
        PINS_DRV_ClearPinIntFlagCmd(BTN_PORT_NAME, PORT_BTN1_INDEX);

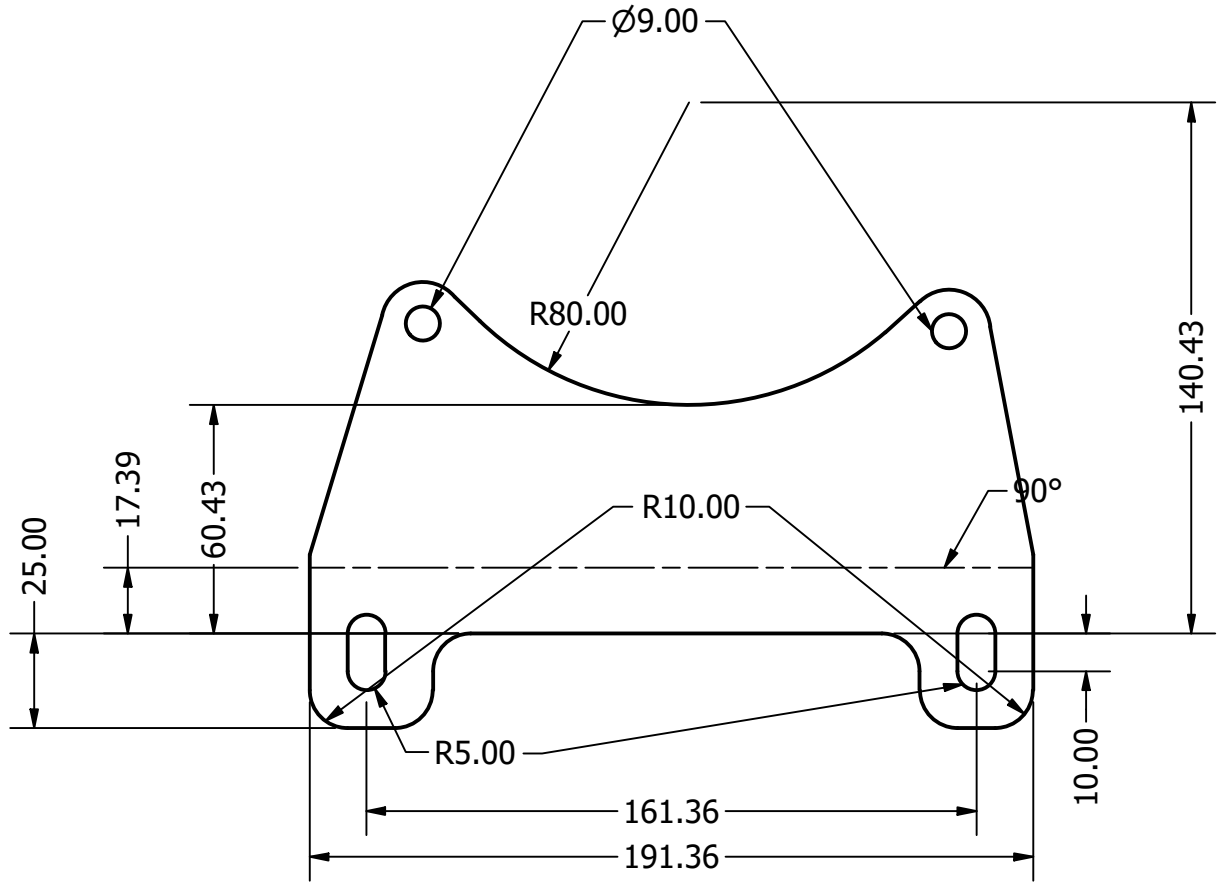
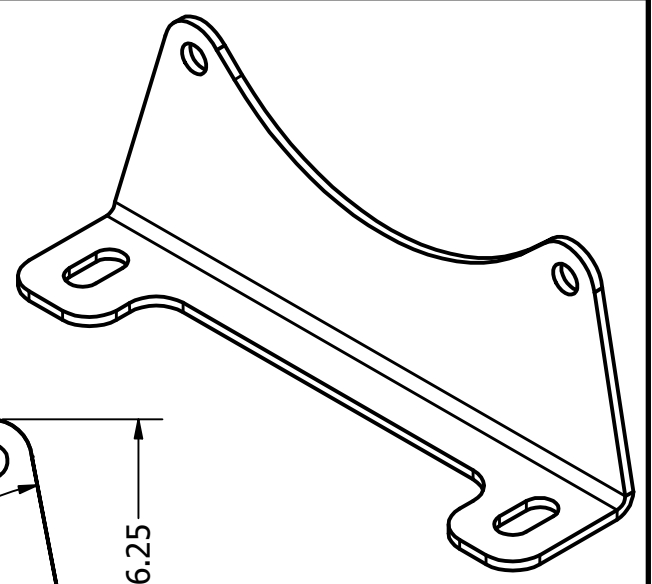
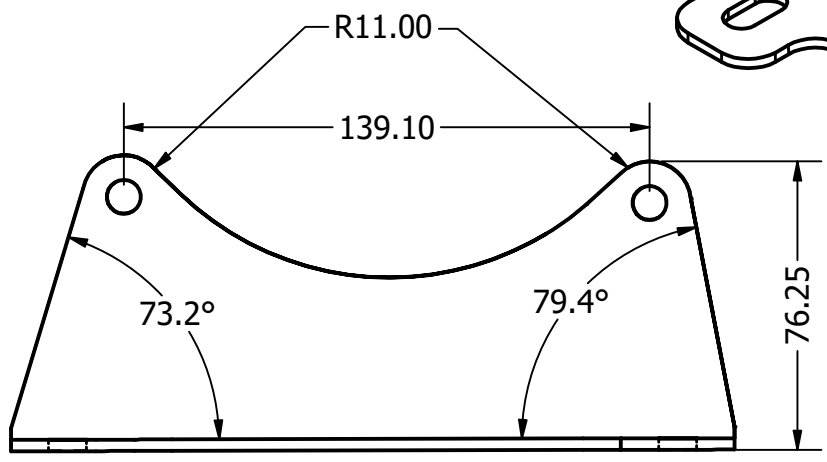
        // KÄSU LOENDURI KONTROLL
        if (button2_press_count > 3) {
            button2_press_count = 0;
            currentcommand = 0;
        }
        else {
            // SUURENDATAKSE KÄSU MUUTUJUAT
            currentcommand++;
        }
    }
}
```

```

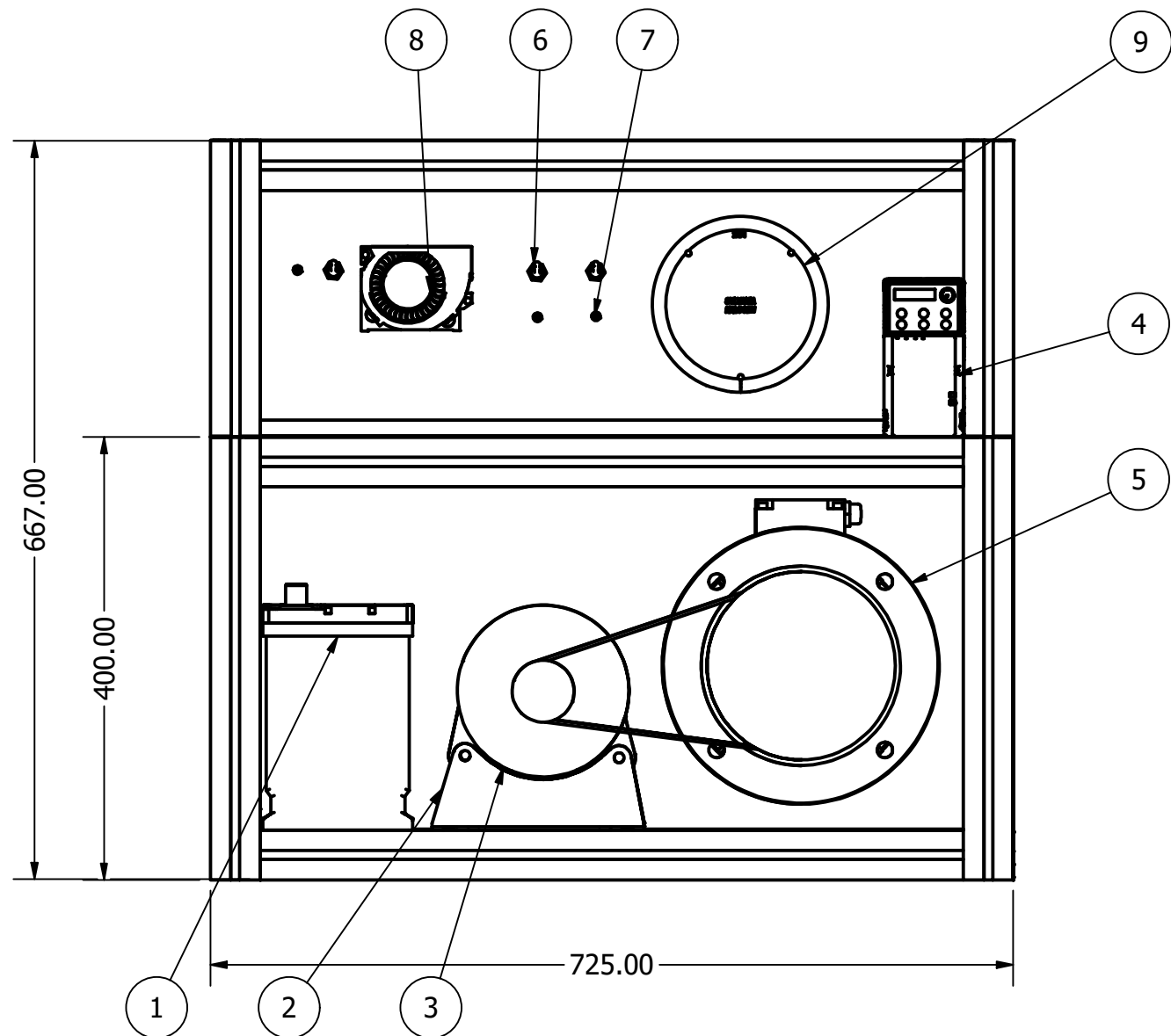
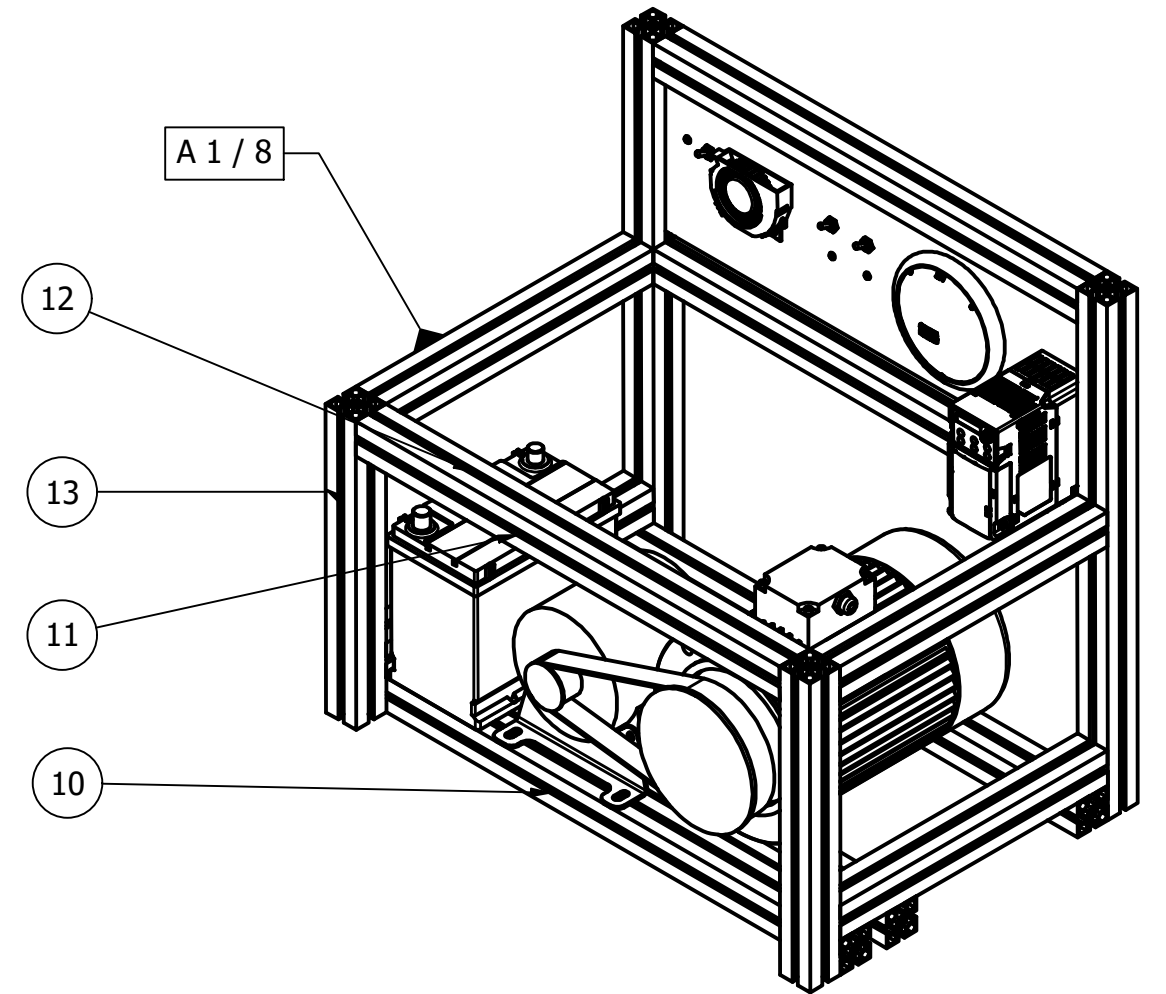
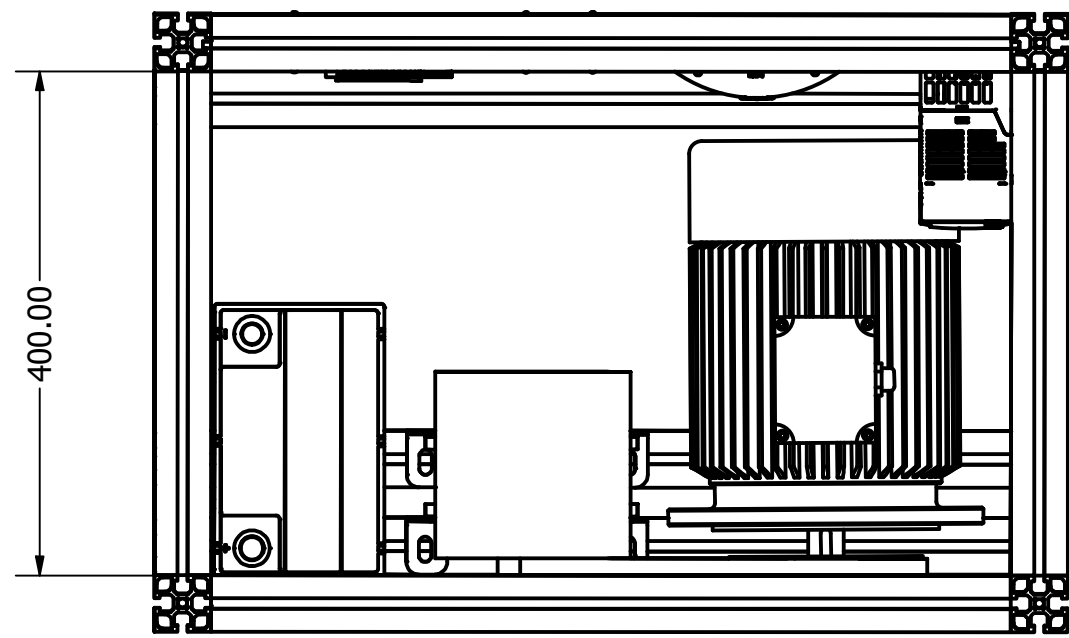
/ * !
/ * @brief TESTIMISEKS NUPUVAJUTUSELE LISATUD ERINEVAD KÄSUD.
/ *
void lin_master_task(void)
{
    / * Initialize LIN network interface */
    l_sys_init();
    l_ifc_init(LI0);
    / * Set Schedule table to Normal */
    l_sch_set(LI0, LI0_NormalTable, 0u);

    / * Infinite loop */
    for (;;)
    {
        switch (currentcommand)
        {
            case COMMAND1:
                l_sch_set(LI0, LI0_NormalTable, 0u);
                break;
            case COMMAND2:
                l_u8_wr_LI0_Motor1Selection(MOTOR_SELECTION_DECREASE);
                break;
            case COMMAND3:
                l_u8_wr_LI0_Motor1Selection(MOTOR_SELECTION_INCREASE);
                /SIIN MUUDETAKSE SÖNUMITE TABELIT
                l_sch_set(LI0, LI0_AbnormalTable, 0u);
                break;
            default:
                break;
        }
    }
}

```



	Materjal:	Stainless Steel	Näitamata piirhälbed:	Mass:	Mööd:
	Teostas:	Argo Ilves	Nimetus:	0,350	1 / 2
	Kontrollis:		Generaatori kinnitus	kg	
Kinnitas:					
Tallinna Tehnikakõrgkool	1	Leht:	Tähis:	00-00-01.ipt	



PARTS LIST				
ITEM	QTY	PART NUMBER	DESCRIPTION	
1	1	Aku		
2	2	00-00-01	Generaatori kinnitus	
3	1	9826550280-00	Generator	
4	1	MS300	Sagedusmuundur	
5	1	MT 2,2KW	Elektrimootor	
6	3	Lülitid		
7	3	FXB6F-LED	signaallamp	
8	1	Soojapuhur 12v	Tarbija	
9	1	Esituli	Tarbija	
10	3	MiniTec 45 x 45	725 mm	
11	2	MiniTec 45 x 45	667 mm	
12	3	MiniTec 45 x 45	635 mm	
13	6	MiniTec 45 x 45	400 mm	
	Materjal:		Näitamata piirhälbed:	Mass: N/A
				Mõõt: 1 / 6
Teostas:	Argo Ilves	Nimetus:		
Kontrollis:		Generaatori stand		
Kinnitas:				
Tallinna Tehnikakõrgkool		1	Leht:	Tähis: 00-01-00.iam