

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Raigo Seppago 240680IACM

# **Tudengivormeli veoajami elektrimootori kontroller**

Magistritöö

Juhendaja: Peeter Ellervee  
PhD

Tallinn 2025

## **Autorideklaratsioon**

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

Autor: Raigo Seppago

12.05.2025

## **Annotatsioon**

Käesolev magistritöö käsitleb elektrimootori kontrolleri arendust tudengivormeli veoajamile. Kontrolleri ülesandeks on tagada veoajami elektrimootoril soovitud väändemoment. Selleks teisendatakse kontrolleri momendi seadeväärtus pingevaheldi transistoride juhtsignaalideks, mille tulemusel tekitatakse mootorile kolmefaasiline pinge. Ühtlasi tagab kontroller veoajami võimalikult efektiivse ning ohutu juhtimise. Efektiivsus tagatakse efektiivsusele optimeeritud väljavektorjuhtimisega. Ohutuse tagamiseks jälgitakse reaal-ajas veoajami olekut ning vea tekkimisel ajami töö peatatakse ja vea põhjus edastatakse süsteemi.

Töös on kirjeldatud kogu kontrolleri arendusprotsess. Esimese sammuna analüüsiti võimalike lahendusi ning defineeriti nõuded. Sellele järgnes elektriskeemide disain, mille baasil disainiti kontrolleri trükkplaat. Ühtlasi on kirjeldatud kontrolleri tarkvara arendus, mis käsitleb juhtimisloogikat ning selle juurde kuuluvaid toetavaid funktsioone. Viimase sammuna katsetati kontrolleriit veoajami katsepingis, kus valideeriti disainitud liidesed ning veenduti juhtimisalgoritmi toimimises.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 69 leheküljel, 7 peatükki, 41 joonist, 14 tabelit.

## **Abstract**

### **Electric Motor Controller for Formula Student Class Car Powertrain**

This thesis focuses on the development of an electric motor controller for Formula Student car powertrain. The controller's purpose is to ensure the desired torque output of the electric motor. To achieve this, the torque setpoint is converted into inverter transistor gate signals, resulting in a three-phase voltage applied to the motor. The controller also ensures efficient and safe operation of the powertrain. Efficiency is achieved by field-oriented control, which is optimized for efficiency. Safety is ensured by continuous monitoring of powertrain state and in the event of a fault, inverter operation is stopped, and cause is communicated to the system.

Motor controller is based on two microcontrollers and multiple measurement circuits for sensors inputs. One microcontroller purpose is to calculate field-oriented control algorithm and ensure 20 kHz switching frequency for the inverter. To ensure fast measurement process for control algorithm feedback signals, measurement circuits like phase currents, DC link and motor position measuring are connected to that microcontroller. Second microcontroller purpose is to handle supporting functions for control algorithm like temperatures measuring, communication with formula CAN network and setpoint transformations.

The thesis describes entire development process of the motor controller. Firstly, possible solutions were analysed, which was followed by defining requirements. Based on requirements, electrical schematics and PCB were designed. Thesis covers also development of the control software, including the control algorithm and supporting functions. Finally, the controller was tested on a powertrain test bench to validate the designed interfaces and verify the operation of the control algorithm.

The thesis is in Estonian and contains 69 pages of text, 7 chapters, 41 figures, 14 tables.

## Lühendite ja mõistete sõnastik

ADC	<i>Analog-to-digital Converter</i> , analoog-digitaalmuundur
CAN	<i>Controller Area Network</i> , suhtlusstandard
CS	<i>Chip Select</i> , suhtlust lubav liin SPI liideses
ECU	<i>Electronic Control Unit</i> , elektrooniline juhtaju
EMC	<i>Electromagnetic Compatibility</i> , elektromagnetiline ühilduvus
EMI	<i>Electromagnetic interference</i> , elektromagnetiline häiring
ESD	<i>Electrostatic discharge</i> , elektrostaatiline lahendus
FOC	<i>Field Oriented Control</i> , väljavektorjuhtimine
I <sup>2</sup> C	<i>Inter-integrated Circuit</i> , suhtlusstandard
IGBT	<i>Insulated Gate Bipolar Transistor</i> , isoleeritud paisuga bipolaartransistor
MSPS	<i>Mega Samples per Second</i> , mõõtmiste arv ühes sekundis
MTTR	<i>Mean Time to Repair</i> , aeg mis kulub töökorda seadmiseks
MTPA	<i>Maximum Torque per Ampere</i> , minimaalsele faasivoolule optimeeritud mootori juhtimine
MTPV	<i>Maximum Torque per Voltage</i> , minimaalsele faasipingele optimeeritud mootori juhtimine
PWM	<i>Pulse Width Modulation</i> , pulsilaiusmodulatsioon
RTD	<i>Resistance Temperature Detector</i> , takistustermoandur
SiC	<i>Silicon Carbide</i> , ränikarbiid
SPI	<i>Serial Peripheral Interface</i> , järjestikisuhtluse liidese standard
VCU	<i>Vehicle Control Unit</i> , sõiduki juhtaju

## Sisukord

1 Sissejuhatus .....	12
2 Elektrivormeli veoajam .....	14
2.1 Veoajami ülevaade .....	14
2.2 Varasemad lahendused .....	15
2.2.1 AMKmotion kontrollrite ja mootorite komplekt .....	16
2.2.2 AMKmotion kontrollrit kolmanda osapoole mootoriga .....	17
2.3 Alternatiivsed lahendused.....	17
2.4 Mootori kontrolleri valik .....	20
2.5 Nõuded arendatavale kontrollerile.....	22
3 Riistvara arendus .....	26
3.1 Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel.....	26
3.2 Enkoodri liides.....	28
3.3 Resolveri liides .....	29
3.4 Mootori temperatuuri mõõteahel .....	32
3.5 CAN liides .....	33
3.6 Pingevaheldi sisendvoolu mõõteahel.....	34
3.7 Mikrokontrollerid .....	36
3.7.1 Juhtimisloogika mikrokontroller .....	36
3.7.2 Toetavate funktsioonide mikrokontroller .....	38
3.8 Toiteahel .....	39
3.9 Liideste pistikud .....	41
3.9.1 Kaitseahelad .....	42
3.10 Trükkplaadi disain .....	43
4 Juhtimisloogika tarkvara arendus .....	46
4.1 Väljavektorjuhtimise algoritm.....	47
4.2 Sisendid sensoritelt .....	48
4.2.1 Mootori faasivoolude ning pingevaheldi sisendpinge mõõtmine.....	49
4.2.2 Mootori võlli nurga lugemine enkoodriga.....	50
4.2.3 Mootori võlli nurga ja kiiruse lugemine resolveriga .....	52

4.3 Mootori pöörlemiskiiruse arvutamine .....	53
4.4 Sisend süsteemist.....	54
4.5 Vealukordade käsitlemine .....	56
4.6 Funktsioonide ajaline planeerimine.....	56
4.7 Pingevaheldi käivitamise protseduur.....	58
5 Toetavate funktsioonide tarkvara arendus.....	60
5.1 Temperatuuride mõõtmine .....	61
5.2 Sisendvoolu mõõtmine .....	62
5.3 Seadeväärtuste teisendamine .....	64
5.4 Vealukordade käsitlemine .....	66
5.5 Kaitseahelate olekusignaalid .....	67
5.6 CAN liides .....	67
5.7 Funktsioonide ajastamine .....	68
6 Prototüübi katsetamine .....	70
6.1 Veoajami katsepink .....	70
6.2 Kiiruse arvutuse valideerimine.....	72
6.3 Faasivoolude mõõtmise valideerimine .....	72
6.4 Vooluregulaatori seadistamine .....	74
6.5 Voolu seadeväärtuste katseline leidmine.....	74
6.6 Tulemuste analüüs .....	76
6.7 Edasised arendused.....	77
7 Kokkuvõte .....	79
Kasutatud kirjandus .....	81
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	85
Lisa 2 – Mootori kontrolleri trükkplaadi elektriskeemid .....	86
Lisa 3 – Mootori kontrolleri trükkplaadi esimene kiht.....	98
Lisa 4 – Mootori kontrolleri trükkplaadi teine kiht .....	99
Lisa 5 – Mootori kontrolleri trükkplaadi kolmas kiht .....	100
Lisa 6 – Mootori kontrolleri trükkplaadi neljas kiht .....	101
Lisa 7 – Mootori kontrolleri trükkplaadi viies kiht .....	102
Lisa 8 – Mootori kontrolleri trükkplaadi kuues kiht .....	103
Lisa 9 – Mootori kontrolleri trükkplaadi 3D mudeli pealtvaade .....	104
Lisa 10 – Mootori kontrolleri trükkplaadi 3D mudeli altvaade.....	105

Lisa 11 – Mootori kontrolleri veakoodid..... 106



## Jooniste loetelu

Joonis 1. Tudengivormeli FEST25 veoajami põhimõtteskeem.....	15
Joonis 2. AMKmotion RACING KIT 4WD pingevaheldite, mootori kontrolleri ja toitemuundurite komplekt [2]. .....	16
Joonis 3. Drivetrain Innovation F-SIC kontrolleri komplekt [4]. .....	18
Joonis 4. Bucher DCU/DCU 60/60 kontrolleri [5]. .....	19
Joonis 5. Kontrolleri valiku kaalutud tulemuste võrdlus. ....	22
Joonis 6. Mootori kontrolleri plokkskeem.....	25
Joonis 7. Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahela põhimõtteskeem.....	28
Joonis 8. Enkoodri liidese põhimõtteskeem. ....	29
Joonis 9. Resolveri liidese põhimõtteskeem.....	31
Joonis 10. Mootori temperatuuri mõõteahela põhimõtteskeem.....	33
Joonis 11. CAN liini terminatsiooni skeem.....	34
Joonis 12. CAN liidese põhimõtteskeem.....	34
Joonis 13. ADC sisendsignaali puhverahel. ....	35
Joonis 14. Pingevaheldi sisendvoolu mõõteahela põhimõtteskeem. ....	35
Joonis 15. Mikrokontrolleri sisendsignaali kaitseahel pingevaheldi ülevoolu kaitseahela olekusignaali näitel. ....	42
Joonis 16. Trükkplaadi pealmise kihi funktsionaalsed tsoonid. ....	44
Joonis 17. Trükkplaadi alumise kihi funktsionaalsed tsoonid. ....	45
Joonis 18. Juhtimisloogika tarkvara lihtsustatud plokkskeem.....	46
Joonis 19. FOC põhimõtteskeem [31]. ....	47
Joonis 20. PWM taimerite konfiguratsioonide programmikood. ....	48
Joonis 21. Mootori faasivoolude ning pingevaheldi sisendpinge mõõtmise protsess. ...	49
Joonis 22. Mootori faasivoolude ning pingevaheldi sisendpinge ADC andmete lugemise programmikood. ....	50
Joonis 23. EnDat 2.1 protokollide ülesehitus [34]. ....	50
Joonis 24. Enkoodri andmete lugemise protsess. ....	51
Joonis 25. Enkoodri vea sõnumite genereerimise programmikood.....	52

Joonis 26. Resolveri mõõtetulemuste lugemise programmikood.....	53
Joonis 27. Mootori pöörlemiskiiruse arvutamise programmikood.....	54
Joonis 28. Veakoodide genereerimise programmikood. ....	56
Joonis 29. Juhtimisloogika mikrokontrolleri funktsioonide ajastamine.....	58
Joonis 30. Pingevaheldi käivitamise protseduur. ....	59
Joonis 31. Toetavate funktsioonide tarkvara lihtsustatud ülevaade. ....	60
Joonis 32. Mootori temperatuuri mõõtmise protsess.....	62
Joonis 33. Pingevaheldi sisendvoolu mõõtmise protsess. ....	63
Joonis 34. Pingevaheldi sisendvoolu piirväärtuste kontrollimise programmikood.....	64
Joonis 35. Momendi seadeväärtuse teisendamine voolu seadeväärtusteks. ....	66
Joonis 36. Kaitsehelate olekusignaali kontrollimise protsess. ....	67
Joonis 37. Koostatud mootori kontrolleri trükkplaat.....	70
Joonis 38. Veoajami katsestendi põhimõtteskeem. ....	71
Joonis 39. Q-telje voolu juhtimine PI-kontrolleriga.....	74
Joonis 40. D- ja q-telje seadeväärtuste kombinatsioonid positiivse väändemomendi näitel. ....	75
Joonis 41. Mootori väändemomendi sõltuvus pöörlemiskiirusest katseliselt leitud seadeväärtustega. ....	76

## Tabelite loetelu

Tabel 1. Kontrolleri valiku analüüsi tulemused.....	21
Tabel 2. Mootori kontrolleri nõuded. ....	24
Tabel 3. Juhtimisloogika mikrokontrolleri vajalikud ühendused. ....	37
Tabel 4. Juhtimisloogika mikrokontrolleri valik. ....	37
Tabel 5. Toetavate funktsioonide mikrokontrolleri vajalikud ühendused. ....	38
Tabel 6. Toetavate funktsioonide mikrokontrolleri valik. ....	39
Tabel 7. Mootori kontrolleri toiteskeemi tarbijad arvestades maksimaalset tarbimist... 40	
Tabel 8. Kontrolleri juhtsõnumi struktuur. ....	55
Tabel 9. Juhtimisloogika mikrokontrolleri peamiste funktsioonide ajalised kestvused. 57	
Tabel 10. Kontrolleri CAN sõnumid. ....	68
Tabel 11. Toetavate funktsioonide esinemis sagedused. ....	69
Tabel 12. Katsesendis kasutatud mõõtevahendid [39], [40], [41]. ....	71
Tabel 13. Kiiruse arvutuse valideerimise tulemused. ....	72
Tabel 14. Faasivoolude mõõtmise valideerimise tulemused. ....	73

# 1 Sissejuhatus

Antud magistritöö ajendiks on tudengivormeli meeskonna FS Team Tallinn pidev arendustöö uute lahenduste suunas, mis tagaksid igal võistlushooajal võiduka vormeli [1]. Sealjuures pannakse olulist rõhku vormeli elektrilise veoajami võimekuse ja efektiivsuse parendamisele. Vormeli veoajam määrab sõiduki maksimaalse kiiruse, kiirenduse ning juhitavuse dünaamilistes sõiduolukordades. Ühtlasi mõjutab veoajami efektiivsus oluliselt vormeli akupaki ning jahutussüsteemi massi, mis moodustavad märgatava osa vormeli kogumassist. Nimetatud põhjustest tulenevalt on vormeli veoajami arendamine määrava tähtsusega kogu vormeli arendusprotsessis.

Käesoleva töö eesmärgiks on arendada mootori kontroller tudengivormeli elektrilisele veoajamile. Kontrolleri arendamine hõlmab selleks vajaliku riist- ja tarkvara arendamist ning disainitud prototüübi katsetamist. Sealjuures tuleb jälgida piiranguid ning nõudeid, mille seavad peamiselt võistlussarja reeglid, veoajami mootor, pingevaheldi ning vormeli elektrisüsteem. Mootori kontrolleri ülesandeks on tagada mootorilt soovitud väändemoment, teisendades soovitud väändemomendi kuueks PWM (*Pulse Width Modulation*) signaaliks pingevaheldi transistoride juhtimiseks. Väändemomendi seadeväärtuse arvutab iga mootori jaoks vormeli VCU (*Vehicle Control Unit*). Kuna iga mootorit juhitakse eraldi, on kokku vajalikud neli mootori kontrollerit ja neli pingevaheldit. Sealjuures on oluline, et iga mootori kontroller tagab juhitava pingevaheldi ja mootori ohutu ning normaaltalitluses töö.

**Lahendatavaks ülesandeks** on disainida tudengivormeli veoajami nõuetele ja võistlussarja reeglitele vastav elektrimootori kontroller. Selle juurde kuuluvateks alamülesanneteks on:

- Võimalike lahenduste analüüs;
- Nõuete defineerimine;
- Elektriskeemide disain ja komponentide valik;
- Tarkvara arendus;
- Testimine ja tulemuste analüüs.

Töö kirjeldab kogu mootori kontrolleri arendusprotsessi, mis jaguneb neljaks peamiseks etapiks: kontseptsioon ning nõuete analüüs, riistvara arendus, tarkvara arendus ning prototüübi katsetamine. Töö esimeses osas antakse ülevaade vormeli veoajamist, seni kasutatud kontrolleri lahendustest, alternatiivsetest lahendustest ning kirjeldatakse nõuded arendatavale kontrolleri. Teises osas kirjeldatakse mootori kontrolleri riistvara arendust, mis käsitleb nii elektriskeemide kui ka trükkplaadi disaini. Töö kolmas ja neljas osa kirjeldab tarkvara arenduse protsessi kontrolleri juhtimisloogika ning liideste jaoks. Viimases osas viiakse läbi prototüübi katsetamine ning tulemuste analüüs.

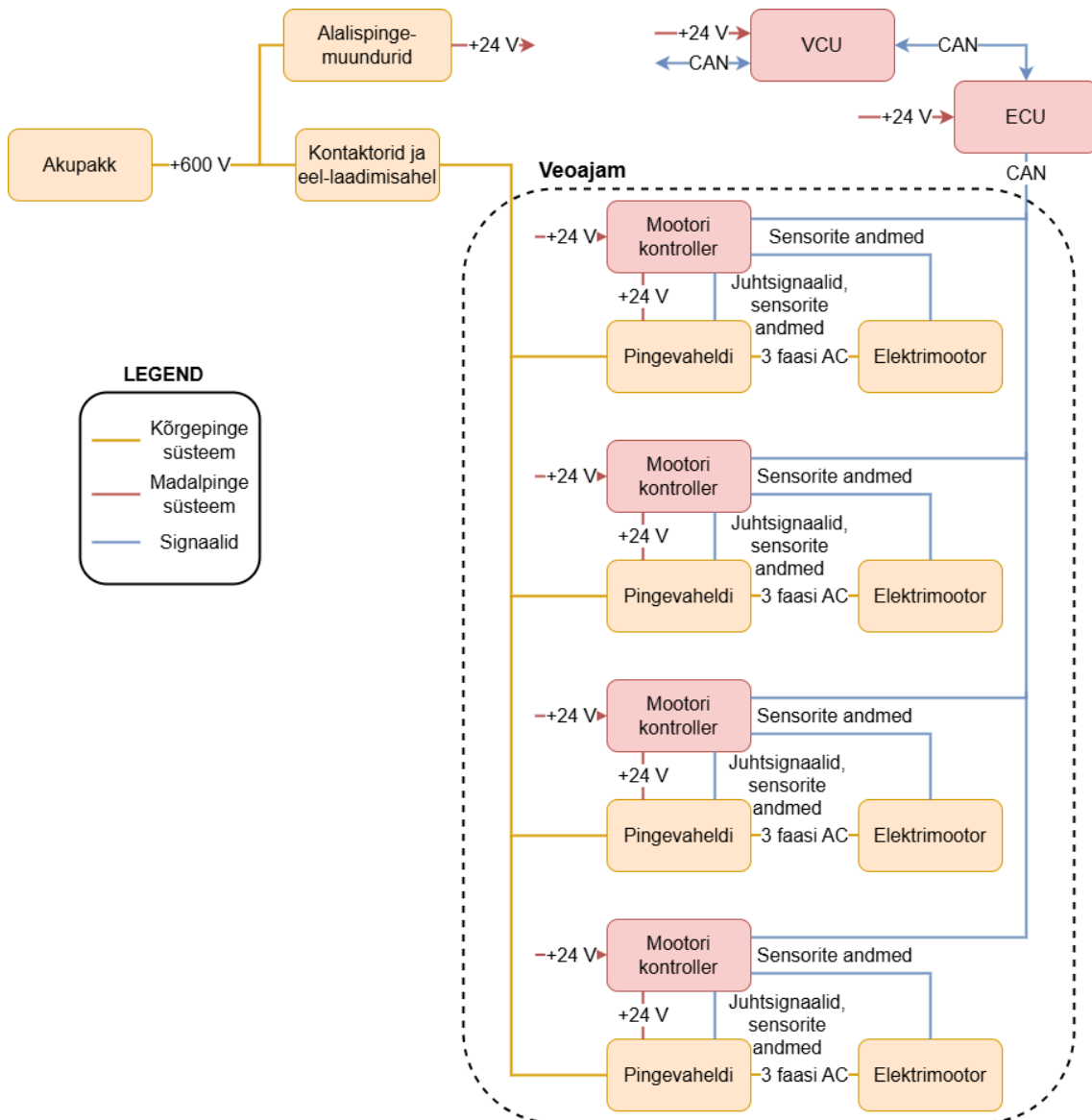
## 2 Elektrivormeli veoajam

Käesoleva töö kirjutamise hetkel arendab tudengivormeli meeskond kolmeteistkümneks elektrivormelit nimega FEST25. Nimetatud vormeli ning selle eelkäija FEST24 puhul on üheks eesmärgiks olnud veoajami võimekuse parendamine. Vormeli FEST24 veoajami võimekust ja efektiivsust parendati uute mootorite kasutusele võtuga. Arendatava vormeli FEST25 puhul otsitakse võimalusi võimekuse parendamiseks mootori kontrolleri ja pingevaheldi näol. Järgnevates peatükkides analüüsitakse varasemaid lahendusi ning võimalike alternatiive, millele järgneb sobivaima lahenduse valik ning nõuete esitamine.

### 2.1 Veoajami ülevaade

Tudengivormeli FEST25 puhul on tegu nelikveolise elektrivormeliga, mille veoajam koosneb neljast mootori kontrolleri, pingevaheldist ning mootorist. Kogu süsteemi toiteallikaks on kõrgepinge akupakk, mille maksimaalne pinge on 600 V. Viimane on piiratud võistlussarja reeglitega [2]. Akupaki väljund on ühendatud läbi kontaktorite ja eel-laadimisahela nelja pingevaheldiga. Kontaktorite eesmärgiks on eraldada akupakk veoajamist kui vormel on madalpinge režiimis. Kõrgepinge režiimi käivitades kontaktorid suletakse ning läbi eel-laadimisahela laetakse pingevaheldite kondensaatorid. Ühtlasi on akupaki väljundiga ühendatud alalispingemuundurid, mille väljundiks on 24 V ning on vormeli madalpinge süsteemi toiteallikaks. Antud madalpinget kasutatakse ühtlasi ka mootorite kontrolleri toiteks. Kontrollerid on lisaks ühendatud läbi CAN-võrgu (*Controller Area Network*) vormeli ECU-ga (*Electronic Control Unit*), mille läbi saadetakse kontrolleri juhtsõnumeid ning saadakse tagasisidet veoajami hetkeseisu kohta. Seadeväärtused igale mootorile arvutatakse vormeli VCU-s, mis peegeldatakse läbi ECU kontrolleri. Veoajami hetkeseisu info saamise jaoks on kriitiliste parameetrite mõõtmiseks paigaldatud erinevad sensorid. Mootorite puhul mõõdetakse nende temperatuure ning mootori võlli asendit ja kiirust. Pingevahelditest mõõdetakse pooljuhtide temperatuure, mootori faasivoolusid, pingevaheldi sisend alalispinget ja -voolu. Nimetatud mõõteandmeid koguvad mootori kontrollerid veoajami juhtimiseks

ning ohutuse tagamiseks. Kontrolleri väljund on juhtsignaalid pingevaheldile, mille tulemusel tekitatakse kolmefaasiline vahelduvpinge mootori juhtimiseks. Kirjeldatud veoajami põhimõtteskeem on toodud Joonis 1.



Joonis 1. Tudengivormeli FES25 veoajami põhimõtteskeem.

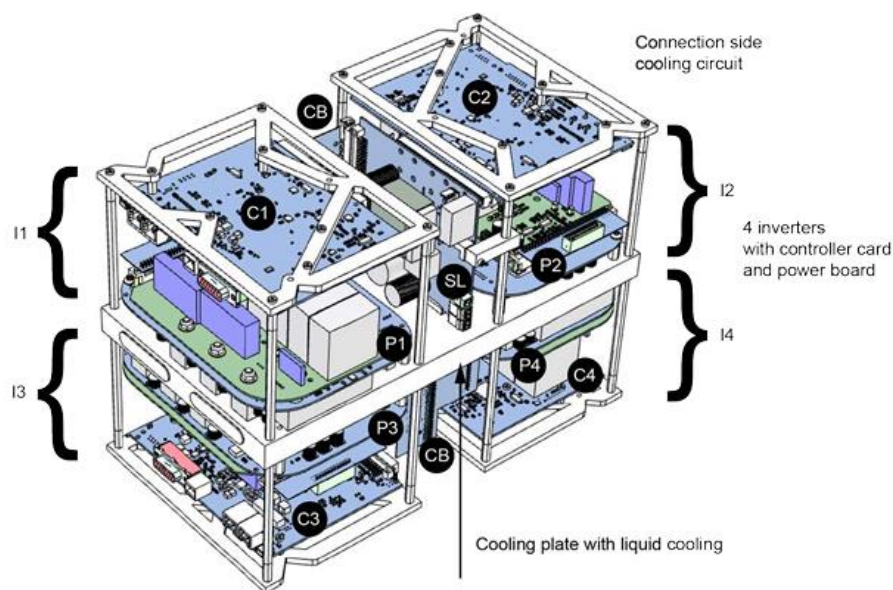
## 2.2 Varasemad lahendused

Meeskonna esimene nelikveoline vormel FES15 valmis aastal 2015. Alates sellest on seni kasutatud iga hooaeg „AMKmotion RACING KIT 4WD“ kontrolleri, mis koosneb neljast pingevaheldist, mootori kontrolleri ning toitemuundurist. Ühtlasi kasutati aastani 2023 nimetatud komplekti juurde kuuluvaid püsिमagnetmootoreid. Seega nii kontrolleri kui ka mootori olid sama tootja poolt. 2024.aasta hooajal otsustati kasutusele võtta

osaliselt meeskonna poolt arendatud mootorid, kuid jätkata endiselt AMKmotion kontrollritega. Kui varasemalt kasutati sama tootja poolt komplekti kuuluvat kontrollereid ja mootoreid, siis kontrolleri seadistamine oli dokumentatsiooni põhjal lihtsasti teostatav. Uue mootori puhul oli seadistamine keerukam, sest tegu on kontrolleri jaoks kolmanda osapoole mootoriga.

### 2.2.1 AMKmotion kontrollrite ja mootorite komplekt

Antud komplekti kuulub neli mootorit, pingevaheldit, mootori kontrollrit ja toitemuundurit [3]. Mootorite puhul on tegu peitpoolustega püsिमagnetmootoriga, mille maksimaalseks võimsuseks on 35 kW. Maksimaalne moment mootori võllil ja pöörlemiskiirus on vastavalt 21 Nm ning  $20\,000\text{ min}^{-1}$  [4]. Iga mootori juhtimiseks on eraldi pingevaheldi, mootori kontrollrit ning toitemuundur. Viimase ülesandeks on muundada 24 V toitepinge madalamateks toitepingeteks. Illustratsioon vaheldite, kontrollrite ja toitemuundurite kompleksis on toodud Joonis 2. Mootori kontrolleri seadistamiseks on spetsiaalne kasutajaliidesega programm, kus on võimalik muuta mootori juhtimiseks vajalike parameetreid [3]. Antud tarkvara puuduseks on siiski piiratud seadistamisvõimalus ning ebavajalike parameetrite üleküllus kasutajaliideses. Viimane tuleneb ilmselt sellest, et programm on mõeldud ka teise otstarbega AMKmotion kontrollrite seadistamiseks. Eeliseks on küllaltki põhjaliku dokumentatsiooni olemasolu nii seadistavate parameetrite selgituste kui ka veakoodide tähenduste osas.



Joonis 2. AMKmotion RACING KIT 4WD pingevaheldite, mootori kontrollrite ja toitemuundurite komplekt [3].



### **2.2.2 AMKmotion kontrolleri kolmanda osapoole mootoriga**

2024.aasta hooajal, kui otsustati kasutusele võtta uued mootorid seni kasutatud AMKmotion mootorite asemel, selgitati välja et AMKmotion kontrolleri on võimalik juhtida ka kolmanda osapoole mootorit. Uute mootorite kasutuselevõtu pealmisteks põhjusteks oli suurem disainivabadus ning parem efektiivsus. Disainivabadust suurendas asjaolu, et antud mootorite puhul tuli mootori korpus ning võll meeskonnal endal disainida ning toota. Tänu kõrgemale efektiivsusele on võrreldes eelmise mootoriga väljund mehaaniline võimsus suurem sama sisendvõimsuse korral. Kuna mõlema mootori kiirus- ning võimsusvahemik on sama, siis suurem väljundvõimsus avaldub kõrgema mehaanilise momendi näol, mis kasvas senisest 21 Nm pealt 26 Nm-ni. Tulenevalt sellest, et AMKmotion kontrolleri õnnestus uut mootorit kasutada, saab väita et antud kontrolleri on piisav seadistamisvabadus erinevate mootorite juhtimiseks. Siiski mitte piisav, et mootori juhtimist optimeerida maksimaalsele efektiivsusele kogu selle töövahemikus.

### **2.3 Alternatiivsed lahendused**

Tulenevalt uute mootorite kasutuselevõtust ning vajadusest leida viise, kuidas tõsta vormeli veoajami võimekust, otsustati uurida seni kasutatud AMKmotion kontrolleri võimalike alternatiivseid lahendusi. Kontrolleri valikule on võimalik läheneda mitmeti. Üheks võimaluseks on kasutada kontrolleri, millel on juba sobiva mootori integratsioon tehtud. Viimane tähendab seda, et kontrolleri on mootori juhtimiseks seadistatud ning kasutaja ei pea selleks vajalike protseduure läbi viima. Kirjeldatud lahendusele vastab meeskonnas pikalt kasutuses olnud AMKmotion kontrolleri ja mootorite komplekt. Antud lahenduse eelisteks on põhjaliku seadistamise protsessi puudumine, tegu on testitud lahendusega ning lihtne integratsioon ülejäänud vormeliga. Puudustena saab välja tuua piiratud optimeerimise võimalus, võimalike probleemide korral pikk MTTR (*Mean Time to Repair*) ning fikseeritud füüsiline kujutegur. Lisaks saab puudusena välja tuua asjaolu, et AMKmotion on üks väheseid ettevõtteid, kes taolist tudengivormeli sarjale sobivat lahendust pakub. Sarnast mootorite ja kontrolleri komplekti pakub ka ettevõtte Drivetrain Innovation. Komplekti kuulub neli kontrolleri, mis on paigutatud koos pingevahelditega kahte identsesse korpusesse nagu on näidatud Joonis 3. Sellest tulenevalt on kontrolleri disain võrreldes AMKmotion kontrolleri kompaktsem ning kergem. Lisaks on Drivetrain Innovation pingevaheldil teoreetiliselt väiksemad kaod, mis

suuresti tuleneb SiC (*Silicon Carbide*) tüüpi transistoride kasutamisest IGBT (*Insulated Gate Bipolar Transistor*) asemel [5], [3].



Joonis 3. Drivetrain Innovation F-SIC kontrolleri komplekt [5].

Võimalik variant on ka valida ise sobiv kontrolleri mootorile. Sellisel juhul kasvab oluliselt valikuvabadus, sest kontrolleri valik turul on väga lai. Valikul tuleb arvestada, et kontrolleri oleks võimalik mootorit juhtida kogu selle töövahemikus ning on tagatud seadistamisvõimalus. Sellest tulenevalt tuleb arvestada järgmisi olulisimaid parameetreid: mootori võimsus, süsteemi pingevahemik, mootori kiirusevahemik, mootori asendi kindlakstegemine, faaside arv. Ühtlasi tuleb arvestada, et kontrolleri tuleb mootori juhtimiseks seadistada, mille jaoks võib kuluda oluliselt aega. Viimase saab jagada kolmeks etapiks. Esiteks tuleb välja selgitada, millist juhtimisalgoritmi kontrolleri kasutab ning milliseid parameetreid tuleb selleks kontrolleri seadistada. Teiseks kontrolleri seadistamine mootoriga, milleks üldjuhul kasutatakse katsestendi, kus on võimalik mootorile ka koormust tekitada. Viimaseks sammuks on kontrolleri testimine kogu mootori töövahemikus, mille käigus kogutakse mõõteandmeid kontrolleri ja mootori kohta ning valideeritakse kontrolleri sobivust. Ühe variandina on võimalik kasutada AMKmotion kontrolleri nagu vormelil FEST24. Ühtlasi sobib kontrolleri ka Drivetrain Innovation F-SIC kontrolleri, mida on võimalik kasutada samuti kolmanda osapoole mootoriga. Võrreldes AMKmotion kontrolleri on see universaalsem, tagades ühilduvuse erinevate positsioonianduritega ning annab võimaluse kasutajal digitaalseid sisendeid ja väljundeid defineerida [5]. Nimelt AMKmotion kontrolleri on võimalik kasutada ainult EnDat 2.1 protokollil baseeruvat enkoodrit ning ligipääs digitaalsetele sisenditele ja väljunditele puudub [3]. F-SIC kontrolleri ühildub nii erinevate enkoodritega

kui ka resolveriga ning võimalik on konfigurērida ühte digitaalset sisendit ning väljundit [5]. Mõlema nimetatud kontrolleri eeliseks on see, et need on mõeldud tudengivormeli sarjale, seega on tagatud vastavus võistlusreeglitega. Sobiv alternatiiv on ka Bucher DCU/DCU 60/60 kontrolleri, mis sarnaselt F-SIC kontrolleri sisaldab ühes korpusis kahte kontrolleri ja pingevaheldi. Nimetatud alternatiiv on näidatud Joonis 4. Kuna tegu pole spetsiaalselt tudengivormeli sarjale mõeldud lahendusega, siis tegu on robustsema ning universaalsema kontrolleri erinevate apliatsioonide jaoks. Robustsus väljendub suuremas massis, milleks on 7,4 kg kahe kontrolleri kohta ning suuremas võimsusvahemikus, milleks on kuni 60 kW [6]. AMKmotion ja F-SIC kontrolleriil on vastavad näitajad 5,5 kg ja 1,1 kg ning 35kW ja 32kW [3], [5].



Joonis 4. Bucher DCU/DCU 60/60 kontrolleri [6].

Lisaks ostutoodetele on võimalik alternatiiv kontrolleri ise arendada. Sellele on võimalik läheneda kahte moodi: täielikult või osaliselt arendatud kontrolleri. Osalise arenduse puhul on võimalik näiteks pingevaheldi osa osta ning mootori kontrolleri osa ise disainida või vastupidi. Antud variant on vähe levinud ning teeb arenduse keerukaks, sest ostutoote osa seab disainile ning optimeerimisele piirangud. Täielikult arendatud kontrolleri puhul on jällegi puuduseks arenduseks kuluv aeg ning ressurss. Ühtlasi vajab taoline lahendus põhjaliku testimist, et tagada vajalik töökindlus. Eelistena saab välja tuua täieliku disaini vabaduse, mis annab võimaluse disainida nii kontrolleri kui ka pingevaheldi konkreetsete vajaduste ja nõuete järgi. Ühtlasi tagab arendus põhjaliku teadmise mootori juhtimisest, mis annab võimaluse innovatsioonile ehk leida viise kuidas mootorit juhtida nii, et kogu veoajami efektiivsus pareneks. Sellised teadmised ja oskused annavad panuse ka

tudengivormeli võistlustel, sest tegu on tootearendus võistlusega, kus lisaks sõidualadele on võimalik punte teenida vormeli disainikaitsmisel [2]. Viimase puhul hinnatakse kõrgelt innovaatilisi lahendusi ning eristumist konkurentidest.

## **2.4 Mootori kontrolleri valik**

Tulenevalt sellest, et meeskond plaanib jätkata osaliselt meeskonna arendatud mootoritega, siis variandid, mille puhul on tegu kontrollrite ja mootorite komplektiga jäävad valikust välja. Kuna mõlema nii AMKmotion kui ka Drivetrain Innovation komplektides sisalduvad kontrollereid on võimalik kasutada ka kolmanda osapoole mootoriga, siis saab neid eraldiseisvate kontrollriteritena siiski arvestada. Ühtlasi sobib kasutuses oleva mootori juhtimiseks Bucher kontrolleri. Neljanda valikuna saab välja tuua variandi sobiv kontrolleri välja arendada. Seega lisaks seni kasutusel olevale AMKmotion kontrolleri on potentsiaalseid alternatiive kolm. Selleks, et leida sobivaim lahendus hinnati neid kuue kriteeriumi alusel. Igale kriteeriumile määrati kaalutegur vahemikus 0-1, mis iseloomustab aspekti olulisust. Kõige suurema kaaluteguriga kriteeriumiteks valiti mass ning töökindlus. Mass seetõttu, et see mõjutab oluliselt vormeli võimekust nii energiakulu kui ka ringiaja vaatenurgast. Ühtlasi on sama oluline töökindlus, sest dünaamilistelt võistlusaladelt on võimalik punkte võita ainult juhul kui vormel tehniliste probleemide tõttu ei katkesta [2]. Oluliselt järgmised kriteeriumid on kujutegur ja efektiivsus. Kujutegur mängib olulist rolli kompaktsuse eesmärgil, sest mida suurem on kontrolleri seda keerulisem on seda vormelisse paigutada. Vormeli efektiivsust saab kontrolleri mõjutada kahel viisil: pingevaheldi efektiivsus ning mootori efektiivne juhtimine. Madalamad kaod annavad võimaluse projekteerida väiksema mahutusega akupaki ning optimeeritud jahutussüsteemi, mis mõjutavad oluliselt vormeli massi. Keskmise kaaluteguriga kriteeriumiteks valiti ühilduvus ning seadistatavus. Ühilduvuse all peetakse silmas ühildumist ülejäänud vormeliga ehk kontrollerist tulenevad piirangud suhtlusprotokollile, toitepingele ja mootori asendiandurile. Seadistatavus määrab milliste seadeväärtustega on võimalik mootorit juhtida ning kui palju on võimalik kasutajal juhtimisalgoritmi muuta. Kriteeriumite alusel hinnati võimalike alternatiive viie punkti skaalal, kus hindele 1 vastab hinnang väga kehv ning hindele 5 vastab hinnang suurepärase. Kaaluteguri ja hinde korrutisena leitakse kaalutud tulemus. Saadud kaalutud tulemused liidetakse kokku ja saadakse iga alternatiivi koguskoor: [7]

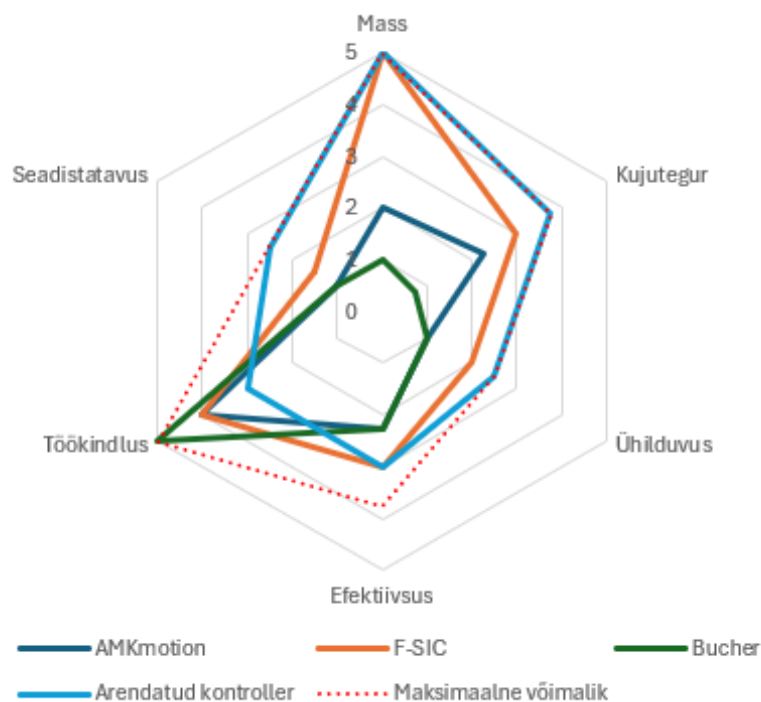
$$Skoor = w_1u_1 + w_2u_2 + w_3u_3 + w_4u_4 + w_5u_5 + w_6u_6. \quad (2.1)$$

kus:  $w$  – kaalutegur,  
 $u$  – hinne.

Võrdlusanalüüsi tulemused on toodud Tabel 1 ja Joonis 5, kust selgub et nimetatud kriteeriumite alusel sobib parimaks lahenduseks kontrolleri arendada. Siiski lähedase skoori sai ka F-SIC kontrolleri jäädes arendatud kontrolleri 1,25 punktiga alla. Arendatud kontrolleri eelisteks F-SIC ees kujunesid kujutegur, ühilduvus ning seadistatavus. F-SIC on küll mõõtmelise kompaktne, kuid siiski paneb piirangud vormeli kerele. Arendatud kontrolleri puhul on võimalik nii kogu, liidesed kui ka juhtimisalgoritm arendajal vastavalt nõuetele projekteerida. F-SIC eeliseks arendatud kontrolleri ees kujunes töökindlus, sest arendatud kontrolleri puhul on tegu prototüübiga. Nii AMKmotion kui ka Bucher kontrolleri said kahe eelnevalt mainituga võrreldes oluliselt madalama tulemuse. Peamisteks põhjusteks on kõrge mass, suured mõõtmed ning piiratud ühilduvus ja seadistatavus.

Tabel 1. Kontrolleri valiku analüüsi tulemused.

<b>Valik</b>		AMKmotion	F-SIC	Bucher	Arendatud kontroller	
<b>Kriteerium</b>	<b>Kaalu- tegur</b>	<b>Kaalutud tulemus</b>	<b>Kaalutud tulemus</b>	<b>Kaalutud tulemus</b>	<b>Kaalutud tulemus</b>	<b>Maksimaal- ne võimalik tulemus</b>
Mass	1	2	5	1	5	5
Kujutegur	0,75	2,25	3	0,75	3,75	3,75
Ühilduvus	0,5	1	2	1	2,5	2,5
Efektiiivsus	0,75	2,25	3	2,25	3	3,75
Töökindlus	1	4	4	5	3	5
Seadistatavus	0,5	1	1,5	1	2,5	2,5
<b>Summa</b>		12,5	18,5	11	<b>19,75</b>	22,5



Joonis 5. Kontrolleri valiku kaalutud tulemuste võrdlus.

Võttes aluseks saadud tulemused ning tudengivormeli sarja eripärasusi otsustati valik arendatud kontrolleri kasuks. Antud lahenduse puhul on suurimaks murekohaks töökindlus, kuid seda riski on võimalik maandada põhjaliku testimisega enne kasutuselevõttu.

## 2.5 Nõuded arendatavale kontrollerile

Kontrolleri saab jagada kaheks osaks: mootori kontrolleri osa ja pingevaheldi. Käesolev töö keskendub mootori kontrolleri osa välja arendamisele. Kontrolleri juurde kuuluva pingevaheldi arendamise ülesanne anti tudengivormeli liikmele Hans Annistele [8]. Sellest tulenevalt on lisaks vormeli poolsetele nõutele ka pingevaheldist ja juhtimisalgoritmist tulenevad nõuded. Kõik nõuded on esitatud Tabel 2 ning mootori kontrolleri plokskeem on toodud Joonis 6.

Vormeli poolsetest piirangutest tuleb arvestada peamiselt ühilduvusele seatud nõuetega. Esiteks vormeli madalpinge süsteemis on kasutusel kolm toitepinget: 5 V, 12 V ja 24 V. Seega kontrolleri tuleb disainida nii, et toitepingena kasutatakse ühte või rohkem antud valikust. Suhtlus VCU ja ülejäänud vormeliga käib üle CAN-võrgu, seega vajalik on CAN liides. Mootorist tulenevad nõuded mootori võlli asendianduri ja mähise

temperatuuri mõõtmise võimekuse olemasolule. Esimesel prototüübil peab olema võimalik liidestada nii resolverit kui ka enkoodrit mootori võlli asendi määramiseks. Sealjuures tuleb arvestada, et kontrolleri peab piirama ja kontrollima mootori pöörlemiskiirust. Mootori maksimaalseks lubatud pöörlemiskiiruseks on  $20\,000\text{ min}^{-1}$ . Ühtlasi, et mootori temperatuur ei tõuseks üle maksimaalse lubatud piiri, milleks on  $130\text{ °C}$  [9]. Viimaks, kuna tegu on nelikveolise vormeliga, siis iga mootori jaoks on vajalik eraldi pingevaheldi ning mootori kontrolleri. Mootoreid juhitakse momendi põhiselt, seega VCU saadab igale kontrolleri soovitud momendi väärtuse. Seadeväärtus võib olla nii negatiivne kui ka positiivne.

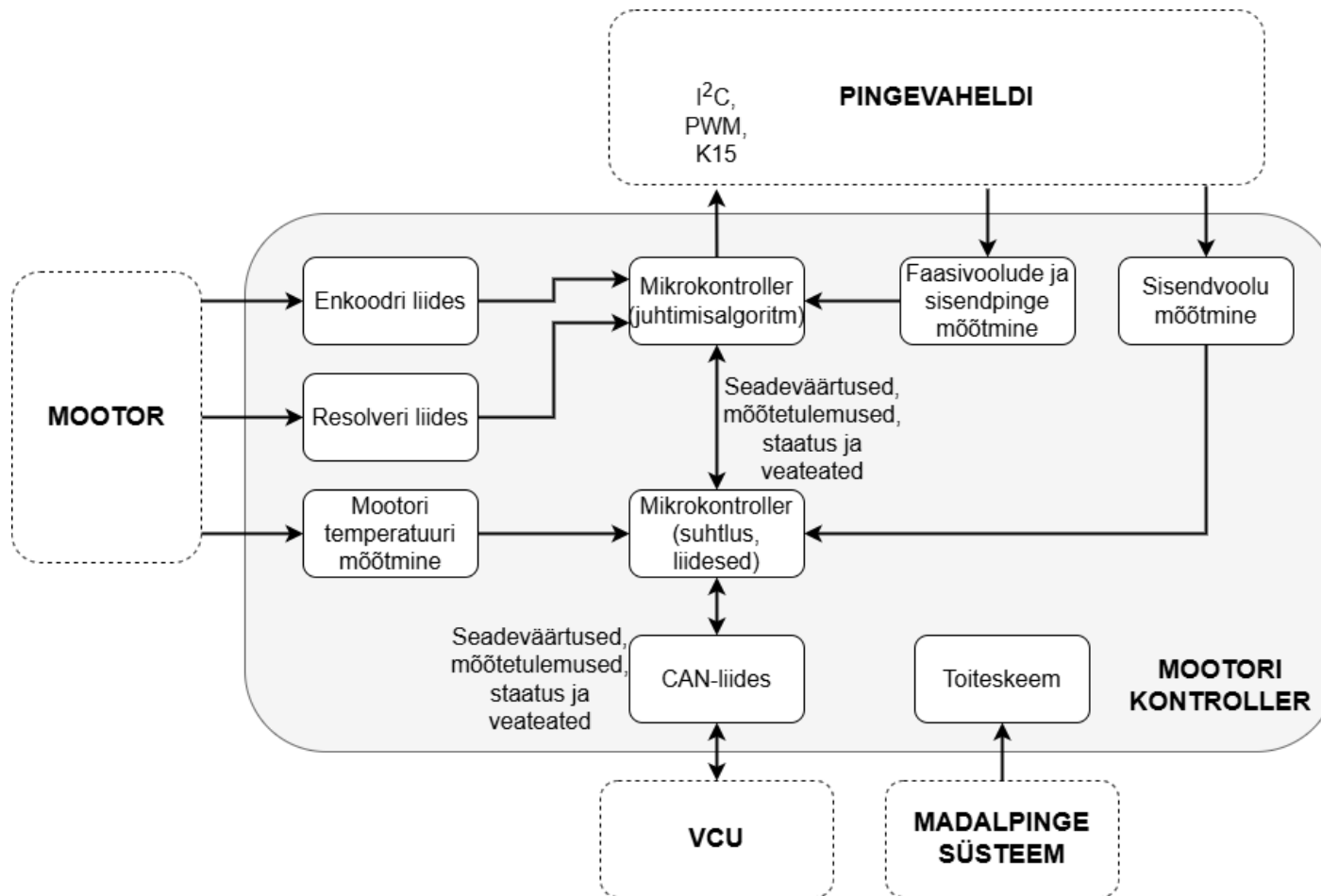
Pingevaheldi ülekoormuse kaitseks ning juhtimisalgoritmi toimimiseks jaoks peab mootori kontrolleri mõõtma mootori kolme faasivoolu ning pingevaheldi sisendpinget – ja voolu. Mootori maksimaalseks lubatud faasivoolu efektiivväärtuseks on  $61\text{ A}$  [9]. Faasivoolude andurid paiknevad pingevaheldis ning nende väljundsignaalid teisendatakse diferentsiaalsignaaleks. Pingevaheldi sisendpinge mõõteahel paikneb samuti pingevaheldis ning väljund viiakse diferentsiaalsignaali kujule. Pingevaheldi sisendvoolu sensor paikneb küll pingevaheldis, kuid sensori signaali mõõtmine tuleb täielikult teha kontrolleri. Ühtlasi peab kontrolleri sisse lugema ning kontrollima kuue transistori temperatuuri. Temperatuuride mõõtmine toimub pingevaheldis ning mõõtetulemused saadetakse SPI (*Serial Peripheral Interface*) liidesega kontrolleri. Transistoride juhtimiseks kasutatakse pingevaheldis spetsiaalseid draivereid, millega suhtlemiseks on vajalik I<sup>2</sup>C (*Inter-integrated Circuit*) liides. Lisaks on transistoride lülitussageduseks  $20\text{ kHz}$ , millest tulenevalt peab kontrolleri väljundiks pingevaheldile olema kuus PWM signaali antud sagedusega.

Ohutuse tagamiseks peab kontrolleri vea tuvastamisel kontrolleri pingevaheldi suutma ohutult välja lülitada ning salvestama vea põhjuse. Vea täpsustus tuleb saata vormeli CAN-võrku.

Tabel 2. Mootori kontrolleri nõuded.

Nõue		Väärtus	Täpsustus
Ühilduvus vormeliga	Toitepinge	24 V / 12 V / 5V	Kõikumine $\pm 6,5$ % (24 V)
	Liidesed	CAN-võrk	Kiirus 1 Mbit/s
Ühilduvus mootoriga	Mootori võlli asendiandur	Enkooder, resolver	Täpsus 18 bitti (enkooder), 16 bitti (resolver)
	Temperatuuri andur	PT1000	Täpsus $\pm 0,1$ °C
Ühilduvus pingevaheldiga	Temperatuurid	SPI liides	Täpsus 16 bitti
	Mootori faasivoolud, pingevaheldi sisendpinge	Diferentsiaalsignaali -3,3 kuni +3,3 V	Täpsus $\pm 1$ %
	Pingevaheldi sisendvool	Analoogsignaali 0-5 V	Täpsus $\pm 1$ %
	Liidesed	I <sup>2</sup> C	Kiirus 100 kHz
	Sisend	6x PWM 0-3,3 V	Sümmeetriline PWM
Funktsionaalsed nõuded	Maksimaalsete lubatud piiride jälgimine	Mootori pöörlemiskiirus, temperatuurid, pingevaheldi voolud ja pinged, seadeväärtused	Piiride ületamisel pingevaheldi peatamine ja veateate genereerimine
	Vea olukordade tuvastamine	Lubatud piiridest väljumine, suhtluse kaotus, liideste veasignaalid/sõnumid	Veateated tuleb edastada CAN võrku
	Mootori juhtimine	Kogu mootori tööpiirkonnas	$\pm 25$ Nm, 0 – 20 000 min <sup>-1</sup>
	Pingevaheldi juhtimine	Normaaltalitus, vea olukord	Ohutu talitus mõlemal juhul
Mitte-funktsionaalsed nõuded	Kontrollerite arv	4	Iga mootori jaoks eraldi
	Lülitussagedus	20 kHz	Muudetav





Joonis 6. Mootori kontrolleri plokk skeem.

## 3 Riistvara arendus

Mootori kontrolleri riistvara arendamiseks kasutati tarkvara Altium Designer. Arendus jagunes kolmeks peamiseks etapiks. Esimese sammuna disainiti elektriskeem ning valiti sobivad komponendid. Järgmiseks disainiti vastavalt elektriskeemile trükkplaat. Ning viimase sammuks oli trükkplaadi tootmine. Koostatud elektriskeemid on toodud Lisas 2, trükkplaadi kihid Lisades 3 – 8 ning trükkplaadi 3D vaated Lisades 9 - 10.

### 3.1 Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel

Tagasisidestatud juhtimisalgoritmi toimimiseks ning pingevaheldi ja mootori ülekoormuse kaitseks on vajalik mootori faasivoolude mõõtmine. Selleks on pingevaheldisse paigutatud kõigi kolme mootori faasivoolu mõõtmiseks sensorid. Nimetatud põhjustel on vajalik ka pingevaheldi sisendpinge mõõtmine, mille jaoks on pingevaheldis samuti vastav mõõteahel. Tagamaks mõõdetud signaalide häirekindlus, muudetakse pingevaheldis voolusensorite analoogsignaale ning sisendpinge mõõteahela signaal diferentsiaalsignaalideks. Viimased on vahemikus  $-3,3\text{ V}$  kuni  $+3,3\text{ V}$ . Selleks, et mõõdetud suuruseid saaks mikrokontrolleri abil kasutada, tuleb need ADC (*Analog-to-digital Converter*) abil digitaalseteks väärtusteks teisendada.

Juhtimisalgoritmi toimimiseks on oluline, et faasivoolude väärtused fikseeritakse kõik samaaegselt ning igal arvutustsüklil. Kuna valitud lülitussageduseks on  $20\text{ kHz}$ , siis valitud ADC peab suutma analoogväärtuseid fikseerida vähemalt iga  $50\text{ }\mu\text{s}$  tagant. Sobivaks muunduriks valiti Analog Device muundur AD7380-4, mis suudab samaaegselt nelja diferentsiaalsignaali kuni  $4\text{ MSPS}$  (*Mega Samples per Second*) mõõta. Antud muunduril on iga sisendi mõõtetulemuse jaoks eraldi SPI andmeliin, mis annab võimaluse kiirelt ning sünkroonselt kõigi nelja mõõteahela väärtused mikrokontrollerile edastada. ADC toitepingeks on vajalik  $3,3\text{ V}$  ning eraldi tuleb tagada toidepinge digitaalsignaale ning mõõteahela referentsile, mis on samuti  $3,3\text{ V}$  [10].

Tagamaks selle, et muunduri referentspinge oleks võimalikult stabiilne, otsustati kasutada eraldiseisvat referents pingeaallikat. Sobivaks pingereferentsiks valiti Maxim Integrated

MAX6070 pingesallikas [11]. Valikul arvestati asjaoluga, et valitud ADC muunduril on nõue, et pingereferents tuleb muundurisse lubada, siis kui muunduri toitepinge ja digitaalsignaali toitepinge on olemas [10]. Antud nõude täitmiseks kasutatakse pingereferentsi lubamisklemmi ning AND-loogikalülitust U1. Täpsemalt kui muunduri toitepinge ja digitaalsignaali toitepinge on kõrges olekus samaaegselt, siis on kõrges olekus ka AND-loogikalülituse U1 väljund, mis omakorda on ühendatud pingereferentsi lubamisklemmiga.

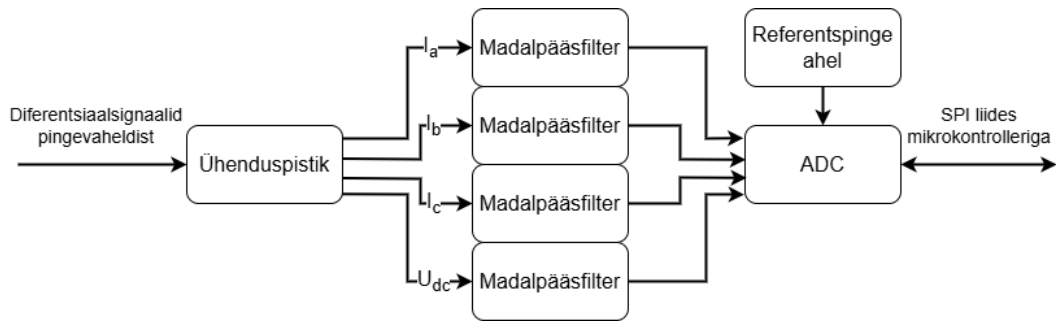
Kuna mootori kontrolleri paikneb pingevaheldi läheduses, mis võib põhjustada häiringuid, siis sisendidiferentsiaalsignaale paigutati madalpääsfiltrid. Viimaste puhul arvestati, et faasivoolu maksimaalseks sageduseks on 1333 Hz [9]. Filtrite löikesageduseks valiti pingevaheldi lülitussagedusest viis korda kõrgem sagedus ehk 100 kHz, et filter ei moonutaks oluliselt mõõdetavate faasivoolude kuju. Filtrite jadatakistite väärtused valiti 75 Ω, et ADC muunduri lekkevoolu tõttu takistitel märgatavat pingelangu ei tekiks. Filtrite rööpmahtuvused arvutati valemiga (3.1):

$$C = \frac{1}{2\pi R f_c}. \quad (3.1)$$

Kus  $C$  – madalpääsfiltri rööpmahtuvus, F,  
 $R$  – madalpääsfiltri jadatakistus, Ω,  
 $f_c$  – madalpääsfiltri löikesagedus, Hz.

Lisaks paigutati filtri eesmärki täitvad kondensaatorid ka diferentsiaalliinide vahele, mille väärtus valiti 10 korda kõrgem kui madalpääsfiltri rööpmahtuvus. Diferentsiaal madalpääsfiltri löikesageduseks on sellisel juhul 10 kHz.

Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahela skeem on toodud Lisas 2 skeemil tiitliga „Phase current & DC link sense“ ning põhimõtteskeem Joonis 7.

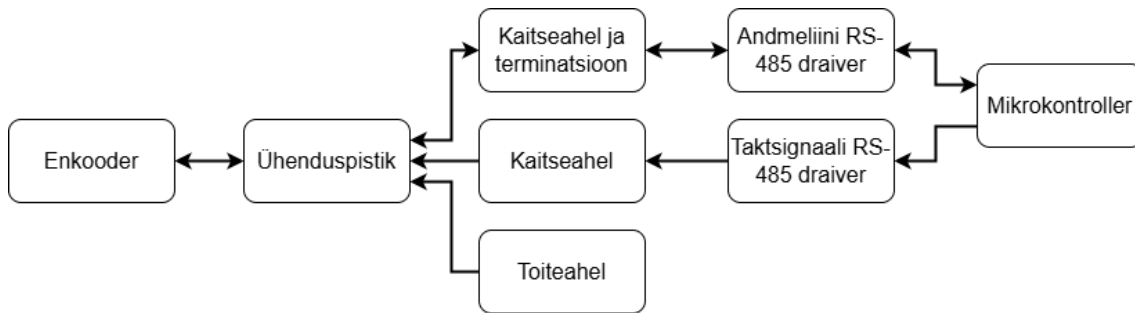


Joonis 7. Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahela põhimõtteskeem.

### 3.2 Enkoodri liides

Mootori võlli asendi määramise üheks võimaluseks on kasutada enkoodrit. Seni on meeskond kasutanud mootorites Heidenhain ECI 1118 enkoodreid. Antud enkoodri andmevahetuse protokolliks on EnDat 2.1, mille füüsiline kiht baseerub diferentsiaal andme- ja taktsignaali liinidel. Ühtlasi on enkoodri toimimiseks vajalik toitepinge vahemikus 4,75-10 V [12].

EnDat 2.1 realiseerimiseks otsustati kasutada mikrokontrolleri SPI liine, mille andme- ja taktsignaali liinid viiakse üle diferentsiaalkujule. Viimase realiseerimiseks kasutatakse Texas Instruments RS-485 draiverit, mis võimaldab pool-dupleks suhtlust [13]. Tulenevalt sellest, et taktsignaali genereeritakse ainult mikrokontrollerist ehk tegu on antud liinis ühesuunalise signaali liikumisega, siis on taktsignaali skeemis draiveri U5 vastuvõtu andme – ning lubamisklemmid ühendatud läbi takistite R22 ja R23 toitepingega. Draiver lubab signaali vastuvõtmist kui vastuvõtu lubamisklemm on madalas olekus ning signaali saatmist kui saatmise lubamisklemm on kõrges olekus [13]. Mõlema draiveri U5 ja U4 diferentsiaalsignaali liinidele on paigutatud madalpääsfiltrid lõikesagedusega 48 MHz vastavalt draiveri referentsdisaini soovitusel [14]. Tulenevalt sellest, et andmeliinidel on pool-dupleks suhtlus, siis on andmeliini diferentsiaalliinidele lisatud terminaator takisti R19, mis tagab et andmeliin on mõlemast otsast termineeritud. Enkoodri liidese põhimõtteskeem on näidatud Joonis 8.



Joonis 8. Enkoodri liidese põhimõtteskeem.

Sobivaks enkoodri toitepingeks valiti 5 V, mis viiakse enkoodrini kontrolleri trükkplaadilt. Kuna enkoodri maksimaalseks tarbimiseks võib olla kuni 0,6 W, siis otsustati vastav toiteskeem ülejäänud trükkplaadi toiteskeemist eraldiseisvana disainida [12]. Selleks kasutati alalispingemuundurit U7, mille sisendiks on 24 V ning väljundiks vajalik 5 V. Muunduri väljundisse on ühendatud *eFUSE* tehnoloogial põhinev kaitseahel U6, mis kaitseb enkoodrit ebastabiilse pinge ning ülekoormuse eest. Täpsemalt piiratakse väljundpinge 5,4 V juurde ning kui sisendpinge peaks tõusma üle 7,6 V, siis väljund lülitatakse välja kuniks viga kestab. Ülekoormuse piiri on võimalik määrata takistiga R30 kasutades valemit (3.2) [15]:

$$R_{LIM} = \left( \frac{97399}{I_{LIM} + 30} \right)^{\frac{1}{1,015}} \quad (3.2)$$

kus  $R_{LIM}$  – vajalik takisti väärtus, k $\Omega$ ,

$I_{LIM}$  – soovitud voolupiir, mA.

Soovitud voolupiiriks määrati 300 mA. Vea tuvastamisel läheb kaitseahela veaklemmi signaal madalaks, mis annab võimaluse mikrokontrolleriga monitoorida kaitseahela seisundit.

Enkoodri liidese skeem on toodud Lisas 2 skeemil tiitliga „Encoder interface“.

### 3.3 Resolveri liides

Meeskonna arendatud mootoriga on võimalus kasutada mootori võlli asendi määramiseks enkoodri asemel ka resolverit. Selgitamaks välja kumb sensor mootori kontrolleri perspektiivist sobiks, otsustati esimesel kontrolleri prototüübil mõlemad võimekused tagada. Potentsiaalseks kasutatavaks resolveriks on TE Connectivity V23401-S1001-

B101. Tegu on ühe pooluspaariga resolveriga, mille ergutuspinge efektiivväärtus on 7 V sagedusega 10 kHz. Kuna antud resolveri ülekandetegur on 0,5, siis siinus ja koosinus väljundsignaalide efektiivväärtus on 4,95 V ning samuti sagedusega 10 kHz [16].

Resolveri liidestamiseks kasutati Analog Device AD2S1210 muundurit, millega on võimalik tekitada resolverile ergutuspinget ning ühtlasi sisse lugeda resolveri siinus ja koosinus signaale. Muundurit on võimalik seadistada digitaalsete sisendite ning SPI liidese läbi. Viimast kasutatakse ka mõõdetud mootori võlli asendi ning kiiruse edastamiseks [17].

Tulenevalt sellest, et muunduri väljundis ergutuspinge efektiivväärtus on 2,5 V, mis on väiksem kui resolverile vajalik pinge, siis lisati ergutusahelasse võimendi U8 [17]. Vajaliku võimendusteguri saab leida kasutades valemit (3.3):

$$A = \frac{U_{\text{väljund}}}{U_{\text{sisend}}}. \quad (3.3)$$

kus  $A$  – võimendi võimendustegur,

$U_{\text{väljund}}$  – võimendi väljundpinge, V,

$U_{\text{sisend}}$  – võimendi sisendpinge, V.

Seega vajalikuks võimendusteguriks on 2,8. Kuna ergutussignaali puhul on tegu diferentsiaalsignaali, siis võimendada tuleb mõlemat signaali. Võimendi U8, mis sisaldab endas kahte võimendit, võimendustegur on määratud sisendi takistitega R36 ja R53 ning tagasiside takistitega R31 ja R52 kasutades valemit (3.4):

$$A = \frac{R_{fb}}{R_{in}}. \quad (3.4)$$

kus  $R_{fb}$  – tagasiside takisti väärtus,  $\Omega$ ,

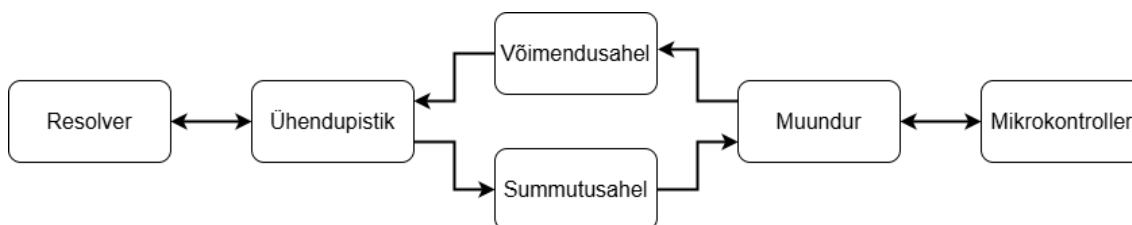
$R_{in}$  – sisendi takisti väärtus,  $\Omega$ .

Muunduri väljundis oleva ergutuspinge keskpinge on 2,47 V [17]. Kuna soovitud ergutuspinge amplituudväärtus on 9,9 V, siis tuleb tõsta ka keskpinget, et võimendi U8 suudaks soovitud signaali kuju tekitada. Viimase väljundis saab signaal olla vahemikus 0-12 V. Keskpinge tõstmiseks on võimendi U8 positiivsesse sisendisse ühendatud pingejagur, mille väljundpinge on 3,4 V. Seega tekitatav ergutuspinge on

keskpingega 5,9 V maa suhtes, mis tagab võimendi ootuspärase töö. Lisaks on tagasiside takistitega R31 ja R28 ühendatud rööbiti kondensaatorid C34 ja C36, mis moodustavad koos nimetatud takistitega madalpääsfiltrid. Viimaste eesmärgiks on vähendada võimalike häiringuid, mis muunduris ergutuspinge moduleerimisel võivad esineda. Lõikesageduseks valiti nimetatud filtritel vastavalt muunduri tootja soovitusel 86 kHz, mis on piisavalt kõrge, et ergutuspinges ei tekiks märgatavat faasinihet [17]. Kondensaatorite väärtuste arvutamiseks kasutati valemit (3.1).

Tulenevalt sellest, et muundur suudab siinus ja koosinus sisendeid mõõta maksimaalselt pingeamplituudiga 4 V, siis tuleb resolveri väljundsignaale summutada [17]. Selleks on siinuse mõõteahelasse pandud takisti R44 ning koosinuse mõõteahelasse takisti R45. Takistid summutavad sisendsignaale efektiivväärtusega 3,5 V sobivasse vahemiku nii, et signaalide efektiivväärtused enne muundurit on efektiivväärtusega 1,2 V. Ühtlasi nõuab muundur, et sisendsignaale ei oleks alla 0 V [17]. Viimase tagamiseks viiakse siinuse signaalide keskpinge takistite R32 ja R33 ning koosinuse signaalide keskpinge takistite R34 ja R35 abil 2,5 V peale. Nii siinuse kui ka koosinuse ahelasse on lisatud ka jadatakistid ning diodid, et kaitsta muundurit ESD (*Electrostatic discharge*) ning lühiste eest. Kuna resolver paikneb mootori küljes, mis võib põhjustada häiringuid, siis mõõteahelatesse on pandud rööpkondensaatorid C35 ja C38 siinuse ahelas ning C36 ja C39 koosinuse ahelas, mis moodustavad jadatakistitega madalpääsfiltrid. Viimaste lõikesageduseks on 97 kHz, mis vastavalt muunduri tootja soovitusel jääb piisavalt kõrgele, et mitte tekitada liigset faasinihet sisendsignaalides [17].

Resolveri liidese skeem on toodud Lisas 2 skeemil tiitliga „Resolver interface“ ning liidese põhimõtteskeem on näidatud Joonis 9.



Joonis 9. Resolveri liidese põhimõtteskeem.

### 3.4 Mootori temperatuuri mõõteahel

Kasutatava mootori mähisesse on paigutatud PT1000 tüüpi temperatuuri andur [9]. Mootori temperatuuri jälgimine on oluline, et vältida kõrgete temperatuuride tulemusel tekkivaid isolatsiooni ning rootori magnetiliste omaduste riknemist. Kasutatav andur on mõeldud mõõtma vahemikus 0-200 °C omades samas vahemikus takistust 1000-1759 Ω [18].

Temperatuuri anduri takistuse mõõtmiseks valiti plaatina RTD (*Resistance Temperature Detector*) andurite jaoks mõeldud Analog Device muundur MAX31865. Muundur mõõdab sensori takistust referents takisti R64 suhtes ning väljastab nende vahelise suhte läbi SPI liidese mikrokontrollerile [19]. Muunduri andmeleht soovib valida referents takisti neli korda suurema kui temperatuuri anduri takistus 0 °C juures, ehk PT1000 puhul 1000 Ω, seega referents takisti R64 väärtuseks valiti 4 kΩ [18], [19]. Muunduriga on võimalik mõõta sensori väärtust kolmes erinevas konfiguratsioonis: 2-, 3- või 4-juhtme meetodiga. Kõige täpsema tulemuse saab 4-juhtme meetodiga, sest elimineeritakse mõõteahelast mõõtejuhtmete takistus. Antud konfiguratsiooni puuduseks on suurem ühenduste arv. Selgitamaks välja, kui palju reaalses erinevad mõõtekonfiguratsioonid mõõtetulemust vormeli aplikatsioonis mõjutavad, jäeti skeemi riistvaraline konfigureerimise võimekus. Nimelt on võimalik takistitega R57-R62 määrata, millist konfiguratsiooni kasutatakse [19].

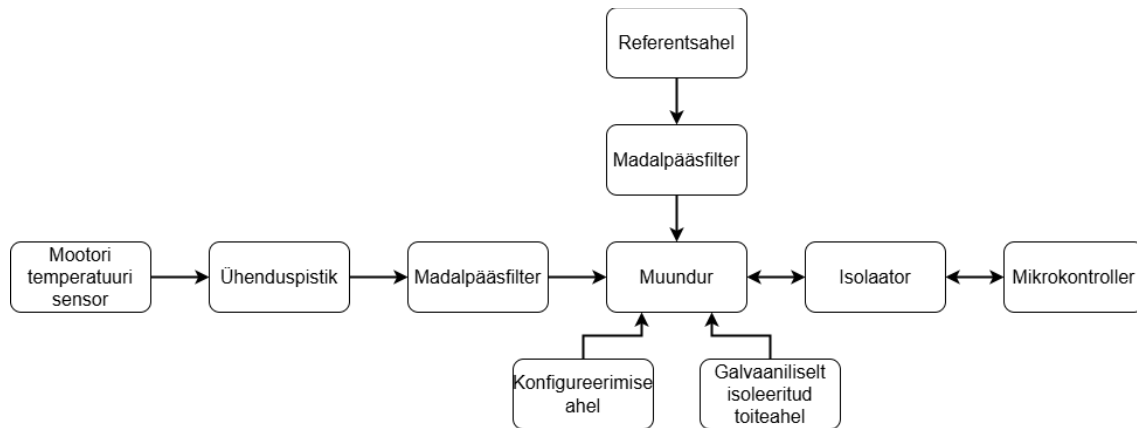
Kuna temperatuuri andur paikneb mootori staatori mähisel, siis on oht, et mootor tekitab sensori mõõtejuhtmetesse häiringuid. Antud riski maandamiseks on jäetud skeemi madalpääsfiltri võimekus takistite R66 ja R67 ning kondensaatorite C71 ja C73 näol. Ühtlasi soovib muunduri tootja kahe mõõteotsa vahele muunduri vahetuslähedusse paigutada PT1000 tüüpi anduri kasutamisel 10 nF kondensaatori [19]. Viimane on skeemis kondensaator C72.

Lisaks võimalikele häiringutele on sensori asukohast tulenev oht isolatsioon kõrgepinge ja madalpinge süsteemi vahel. Antud riski maandamiseks on kogu temperatuuri sensori mõõteahel ülejäänud kontrollerist galvaaniliselt isoleeritud. Digitaalsignaale isoleerimiseks on skeemis isolaator U11. Toitesüsteemi isoleerimiseks on isoleeritud alalispingemuundur U13, mille sisendpinge on 24 V ning väljundiks on isoleeritud 5 V [20]. Kuna muundur U12 vajab digitaalsignaale ning analoog mõõteahela jaoks eraldi



3,3 V, siis on muunduri U13 väljundisse ühendatud vastavalt kaks lineaarregulaatorit U14 ja U16 [19].

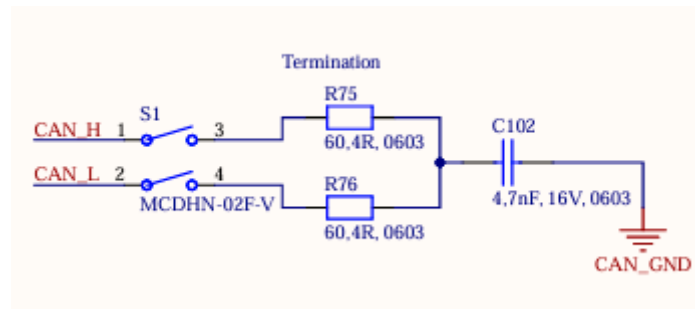
Mootori temperatuuri mõõteahela skeem on toodud Lisas 2 skeemil tiitliga „Motor temperatuure“ ning põhimõtteskeem on näidatud Joonis 10.



Joonis 10. Mootori temperatuuri mõõteahela põhimõtteskeem.

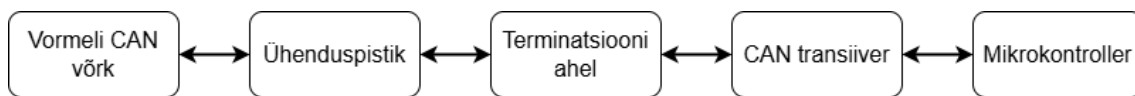
### 3.5 CAN liides

VCU-ga suhtlemiseks on kontrollerial vajalik CAN liides. Viimase füüsilises kihis on kaks diferentsiaalsignaali liini CANH ja CANL ning maa liin. Antud liidese riistvara disainiks kasutati tudengivormeli meeskonna standardset CAN-transiiveri skeemi, mis pole autori loodud. Skeemi keskmeks on CAN-transiiver U17, mis teisendab mikrokontrolleri CAN liinid diferentsiaalsignaalideks. See tagab selle, et mikrokontroller saab CAN-liinile andmeid saata ning ka vastu võtta. CAN võrgu toimimiseks on vajalik, et CAN liin oleks liini kaugeimatest punktidest termineeritud 120  $\Omega$  takistiga ehk liini kogutakistus oleks 60  $\Omega$  [21]. Selleks on skeemi lisatud CANH liinile takisti R75 ning CANL liinile takisti R76, kumbki 60  $\Omega$  (vt. Joonis 11). Kuna kontrollereid on lõpuks kokku neli ning terminatsiooni on suure tõenäosusega vaja ainult ühel, siis terminatsiooni takistid on CAN liinile ühendatud läbi lüliti S1. Viimane annab võimaluse lihtsasti kasutajal takistid liinile lülitada.



Joonis 11. CAN liini terminatsiooni skeem.

Kogu CAN liidese skeem on toodud Lisas 2 skeemil tiitliga „CAN“ ning põhimõtteskeem on näidatud Joonis 12.



Joonis 12. CAN liidese põhimõtteskeem.

### 3.6 Pingevaheldi sisendvoolu mõõteahel

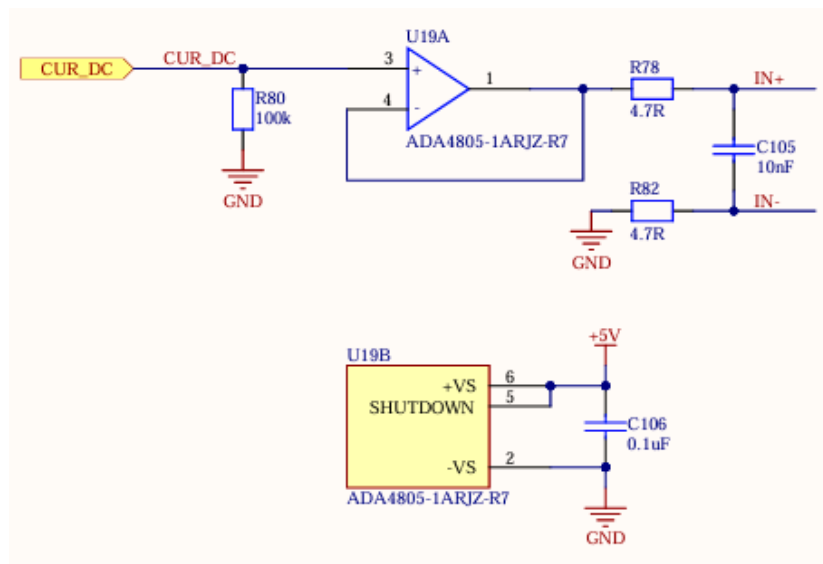
Pingevaheldi ülekoormuse tuvastamiseks on lisaks mootori faasivoolu anduritele paigutatud voolusensor ka pingevaheldi sisendisse. Viimane mõõdab pingevaheldisse minevat alalisvoolu. Sensori väljundiks on analoogsignaali vahemikus 0-5 V, mis tuleb ADC abil mikrokontrollerile edastada, et signaal voolu väärtuseks teisendada.

Sobivaks ADC muunduriks valiti Texas Instruments ADS8339 muundur, mille mõõtevahemik on 0 V kuni referentspinge ning mõõtetulemuse edastamiseks on SPI liides. Referentspingeks valiti 5 V, mis katab kogu sisendsignaali vahemiku ning ühtlasi jääb muunduri soovitatud referentspinge vahemiku. Referentspinge tekitamiseks kasutatakse skeemis pingesallikat U20 [22].

Antud muunduril on võimekus mikrokontrollerile märku anda, et muundur valmis mõõtetulemuse läbi SPI liidese saatma. Kui mõõtetulemused on valmis edastamiseks, siis muundur viib SPI andmeliini madalale tasemele. Et antud funktsionaalsus toimiks, siis on andmeliin ühendatud läbi takisti R77 digitaalsüsteemi toitepingesse. Viimane hoiab andmeliini kõrges olekus, kui andmete edastust ei toimu ning kui muundur pole valmis mõõteandmeid edastama. Lisaks on muunduri tootja soovitusel tuginedes SPI

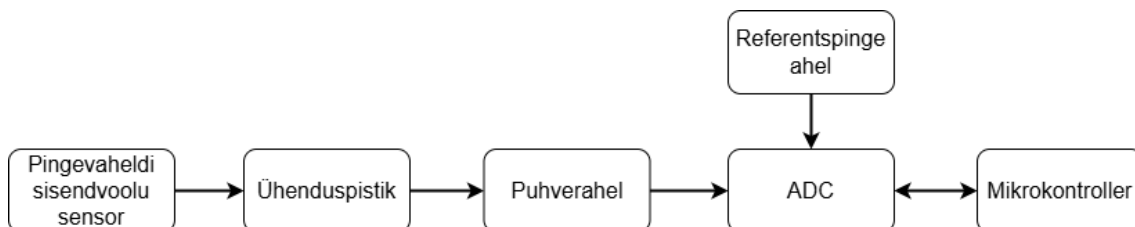
taktsignaali liinile ühendatud takisti R79 ning CS (*Chip Select*) liinile takisti R81, et suurendada häirekindlust [22].

Voolusensori väljundsignaal on ühendatud ADC muunduriga läbi puhverahela, mis on näidatud Joonis 13. Puhvri funktsionaalsust täidab võimendi U19, mille negatiivsesse sisendisse on ühendatud võimendi väljund. Viimase eesmärgiks on mõõtmise täpsuse suurendamine. Täpsemalt tagab võimendi U19 kõrge sisendimpedants selle, et mõõteahel ei koorma sensori väljundsignaali. Ühtlasi võimendi U19 madal väljundimpedants tagab stabiilse signaali ADC mõõteahelale. Tuvastamaks olukorda, kus sensor on lahti ühendatud, on sensorist saabus signaal läbi takisti R80 ühendatud maaga, mis viib signaali nimetatud olukorras madalale nivoole.



Joonis 13. ADC sisendsignaali puhverahel.

Kogu pingevaheldi sisendvoolu mõõteahela skeem on toodud Lisas 2 skeemil tiitliga „DC current measurement“ ning põhimõtteskeem on näidatud Joonis 14.



Joonis 14. Pingevaheldi sisendvoolu mõõteahela põhimõtteskeem.

## 3.7 Mikrokontrollerid

Juhtimisalgoritmi täpsuse seisukohast on oluline, et kogu arvutustsükli on võimalik ära teha ühe pingevaheldi lülitusperioodi jooksul. Kuna pingevaheldi lülitussageduseks on valitud 20 kHz, siis ühe juhtimisalgoritmi arvutustsükli jaoks on aega maksimaalselt 50 µs. Sellest tulenevalt otsustati kasutada kontrolleris kahte mikrokontrollerit: üks juhtimisalgoritmi arvutamiseks ning teine kõikide muude funktsionaalsuste tagamiseks. Antud lähenemine tagab selle, et kogu ühe mikrokontrolleri ressurss on mõeldud ainult juhtimisalgoritmi jaoks ning ajalise piirangu jälgimine on oluliselt lihtsam. Kahe mikrokontrolleri vaheliseks andmete vahetamiseks kasutatakse SPI liidest. Alternatiivina on võimalik kasutada mitme tuumalist mikrokontrollerit, kuid arendustöö lihtsustamiseks otsustati mitte antud varianti kasutada.

### 3.7.1 Juhtimisloogika mikrokontroller

Esimese sammuna loetleti üles kõik vajalikud ühendused ja liidesed, mis juhtimisloogika mikrokontrolleriga peavad ühendatud olema. Sealjuures lähtuti sellest, et antud mikrokontrolleriga ühendatakse ainult need liidesed, mis on otseselt vajalikud juhtimisalgoritmiks. Kontrolleri enda trükkplaadil olevatest skeemidest peavad juhtimisloogika mikrokontrolleriga olema järgmised ahelad: pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel, enkoodri liides, resolveri liides ning teine mikrokontroller. Sealjuures tuleb arvestada, et samaaegselt enkoodri ja resolveri liideseid ei kasutata, mis annab võimaluse sama andmeliini mõlema liidese jaoks kasutada. Lisaks on vajalikud ühendused järgmiste pingevaheldi ahelatega: transistoride draiverid, madalpinge toite lubamisklemm ning riistvaraline ülevoolu kaitseahel. Esimeste eesmärgiks on lülitada juhtimisalgoritmi väljundite mikrokontrollerist ehk 6 PWM signaali põhjal jõutransistore ning tagada mikrokontrolleri eraldatus jõuelektronikast. Toite lubamisklemmi kõrge nivoo lubab toitepinge transistoride draiveritele ning on ühendatud juhtimisloogika mikrokontrolleriga, sest kõik draiveritega seotud ühendused on seotud otseselt juhtimisalgoritmiga. Lisaks tarkvaralisele ülekoormuse kaitsele on pingevaheldisse disainitud riistvaralised kaitseahelad, mille olekusignaale on võimalik mikrokontrolleriga monitoorida. Juhtimisalgoritmi perspektiivist on iga arvutustsükkel oluline teada, kas mõni kaitseahel on rakendunud, et selle põhjal vastu võtta juhtimisotsus. Vajalike ühenduste arv iga nimetud skeemiga on toodud Tabel 3.

Tabel 3. Juhtimisloogika mikrokontrolleri vajalikud ühendused.

Mikrokontrolleriga ühilduv ahel	Vajalikud ühendused ja liidesed
Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel	1x digitaalne väljund, 4x SPI
Enkoodri liides	1x digitaalne sisend, 2x digitaalset väljundit, 1x SPI (ühine resolveri ja enkoodri jaoks)
Resolveri liides	7x digitaalset väljundit, 6x digitaalset sisendit, 1x SPI (ühine resolveri ja enkoodri jaoks)
Toetavate funktsioonide mikrokontroller	1x SPI, 1x digitaalne väljund
Transistoride draiverid	6x PWM (ühise taimeriga), I <sup>2</sup> C, 2x digitaalset sisendit
Pingevaheldi madalpinge lubamisklemm	1x digitaalne väljund
Riistvaralised ülevoolu kaitseahelad	4x digitaalset sisendit

Ühtlasi seab mikrokontrollerile nõuded juhtimisalgoritm, mille nõuded on tuletatud varasemast arendusest [23]. Nimelt juhtimisalgoritmi toimimiseks peab mikrokontrolleri taktsagedus olema vähemalt 216 MHz. Ühtlasi algoritmi väljundiks olevad 6 PWM signaalide moduleerimiseks on vajalik 32 bitine taimer, mis suudab samaaegselt kõiki 6 PWM signaali moduleerida.

Sobiv mikrokontroller otsustati valida STMicroelectronics'i STM32 F-seeria seast, sest antud seeria peal viidi läbi varasem algoritmi katsetamine ning on tüüpiline seeria valik tudengivormeli projektides [23]. Antud seeriast jäi valikusse neli mikrokontrollerit, mis vastavad eeltoodud kriteeriumitele [24], [25], [26], [27]. Valikusse jäänud mikrokontrollerid on toodud Tabel 4. Sobivaks valiti nende seast STM32F777ZIT6, sest sellel on teistega võrreldes rohkem muut- ning väälmälu.

Tabel 4. Juhtimisloogika mikrokontrolleri valik.

Mikrokontroller	Digitaalsed sisendid/väljundid	SPI liidesed	Maksimaalne taktsagedus	Muutmälu	Väälmälu
STM32F777ZIT6	114	6	216 MHz	512 kB	2 MB
STM32F745ZGT6	114	6	216 MHz	320 kB	1 MB
STM32F746ZGT7	114	6	216 MHz	320 kB	1 MB
STM32F745ZET6	114	6	216 MHz	320 kB	512 kB

Antud mikrokontrolleri skeem on toodud Lisas 2 skeemil tiitliga „Main MCU“.

### 3.7.2 Toetavate funktsioonide mikrokontroller

Kõik muud liidesed, mille andmeid pole juhtimisalgoritmi jaoks iga arvutustsükkel vaja uuendada, ühenduvad toetavate funktsioonide mikrokontrolleriga. Kontrolleri enda trükkplaadil on nendeks järgmised ahelad: mootori temperatuuri mõõteahel, CAN liides, pingevaheldi sisendvoolu mõõteahel ning teine mikrokontroller. Pingevaheldi ahelatest on vajalikud järgmiste skeemide ühendused: transistoride temperatuuride mõõteahel, transistoride draiverite olekusignaaliid ning riistvaralised ülevoolu kaitseahelad. Kaks viimast on dubleeritud kahe mikrokontrolleri peale, et info jõuaks mõlema mikrokontrollerini ka siis, kui suhtlus kahe mikrokontrolleri vahel peaks katkema. Toetavate funktsioonide mikrokontrollerile on lisaks ühendatud antud skeemide olekute lähtestusviigud. Vajalike ühenduste arv iga nimetud ahelaga on toodud Tabel 5.

Tabel 5. Toetavate funktsioonide mikrokontrolleri vajalikud ühendused.

Mikrokontrolleriga ühilduv ahel	Vajalikud ühendused ja liidesed
Mootori temperatuuri mõõteahel	1x digitaalne sisend, 1x digitaalne väljund, 1x SPI (transistoride temperatuuridega ühine)
CAN liides	1x CAN
Pingevaheldi sisendvoolu mõõteahel	1x digitaalne sisend, 1x digitaalne väljund, 1x SPI
Juhtimisloogika mikrokontroller	1x SPI, 1x digitaalne sisend
Transistoride temperatuuri mõõteahel	1x digitaalne väljund, 1x SPI (ühine mootori temperatuuri mõõteahelaga)
Transistoride draiverite olekusignaaliid	3x digitaalset sisendit, 2x digitaalset väljundit
Riistvaralised ülevoolu kaitseahelad	4x digitaalset sisendit, 1x digitaalne väljund

Sobiv mikrokontroller valiti jällegi STMicroelectronics'i STM32 F-seeria seast. Valikusse jäid kolm mikrokontrollerit, mis suudavad täita Tabel 5 toodud ühenduste nõudmisi [28], [29], [30]. Antud mikrokontrollerid on toodud Tabel 6. Sobivaks valiti STM32F446RET7, sest sellel on teistega võrreldes rohkem väikmälu, mis on oluline erinevate funktsionaalsuste tagamiseks ning teisendustabelite talletamiseks programmikoodis.

Tabel 6. Toetavate funktsioonide mikrokontrolleri valik.

Mikrokontroller	Digitaalsed sisendid/väljundid	SPI liidesed	Muutmälu	Välkmälu
STM32F446RCT7TR	50	3	128 kB	256 kB
STM32F446RCT7	50	3	128 kB	256 kB
STM32F446RET7	50	3	128 kB	512 kB

Toetavate funktsioonide mikrokontrolleri skeem on toodud Lisas 2 skeemil tiitliga „Com MCU“.

### 3.8 Toiteahel

Eelnevalt kirjeldatud kontrolleri skeemid vajavad erinevaid toitepingeid. Nendeks on: 3,3 V digitaalsüsteemidele, 3,3 V analoogsüsteemidele, 5 V, 12 V ja 24 V. Analoo- ja digitaalsüsteemidele on eraldi toiteahelaid vaja häirekindluse suurendamiseks. Ühenduste arvu minimaalseks hoidmiseks ning stabiilse ja häiringute vaba toitepingete saamiseks otsustati kasutada kontrolleri trükkplaadil lokaalseid alalispingemuundureid. Selliselt vajab kontrolleri vormeli madalpinge süsteemist ainult 24 V toitepinget.

Sobivate alalispingemuundurite valimiseks selgitati esmalt välja vajalik võimsustarve kõikide toitepingete jaoks. Kõikide ahelate maksimaalsed võimalikud võimsustarbed on toodud Tabel 7. Madalamate pingete tekitamiseks kasutatakse sellele lähimat kõrgemat pinget, et hoida pinge muundamine võimalikult efektiivsena. Teisisõnu 3,3 V muundamiseks kasutatakse 5 V sisendpinget ning 5 V muundamiseks kasutatakse 12 V sisendpinget. Seega iga toitepinge juures tuleb tarbijana arvestada ka madalamate pingete võimsustarvet. Antud info on toodud Tabel 7 kahel viimasel real.

Tabel 7. Mootori kontrolleri toiteskeemi tarbijad arvestades maksimaalset tarbimist.

Skeem	3,3 V (analoog)	3,3 V (digitaal)	5 V	12 V	24 V
Pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel	35 mA	30 mA	1,5 mA		
Enkoodri liides		2 mA			63 mA
Resolveri liides		2 mA	47 mA	120 mA	
Mootori temperatuuri mõõteahel		6 mA			0,5 mA
Juhtimisloogika mikrokontroller		250 mA			
CAN liides			260 mA		
Pingevaheldi sisendvoolu mõõteahel			4,8 mA		
Toetavate funktsioonide mikrokontroller		110 mA			
<b>Voolutarve kokku</b>	35 mA	408 mA	313,3 mA	120 mA	63,5 mA
<b>Võimsustarve kokku</b>	0,12 W	1,35 W	1,57 W	1,44 W	1,52 W
<b>Võimsustarve arvestades pingemuundamisi</b>	0,12 W	1,35 W	3,03 W	4,47 W	5,99 W
<b>Voolutarve arvestades pingemuundamisi</b>	35 mA	408 mA	606 mA	372 mA	250 mA

Toitepingete 12 V, 5 V ning digitaalsüsteemi 3,3 V muundamiseks kasutatakse Recom Power tootja alalispingemuundureid, mis on skeemil märgitud vastavalt U21, U22 ja U24. Vastavalt tootja soovitusel on alalispingemuundurite U21, U22 ja U24 ette lisatud PI-filtrid, mis koosnevad kahest rööpkondensaatorist ning ühest jadainduktorist. Nende eesmärgiks on EMC (*Electromagnetic Compatibility*) häiringute vähendamine [31]. Ühtlasi on nimetud muundurite väljunditesse Tabel 7 viimaselt realt saadud info põhjal, lisatud sulavkaitsed F1, F2 ja F3. Tulenevalt sellest, et analoogsüsteemi 3,3 V toitepinge ahelas on üpris madal voolutarve, siis antud pinge muundamiseks otsustati kasutada lineaarregulaatorit U23.

Kogu toiteahela skeem on toodud Lisas 2 skeemil tiitliga „Supply“.



### 3.9 Liideste pistikud

Tagamaks lihtsama juhtmestuse ning suurema häirekindluse saavutamiseks on kontrolleri liidesed jaotatud seitsme pistiku vahele. Nendest kaks J1 ja J4 on mõeldud pingevaheldi ühenduste jaoks. Üks analoog- ja teine digitaalsignaali jaoks. Mõlemad on valitud TE Connectivity MQS-seeriast, mis on autotööstuses laialt levinud. Pistikute valimisel peeti silmas, et tagatud oleks vibratsioonikindlus lukustusmehhanismi näol ning kannataksid vähemalt 100 ühendamise tsükli. Pingevaheldi analoogsignaali pistikusse J4 ühenduvad pingevaheldi sisendvoolu ja -pinge ning mootori faasivoolude mõõteahela signaalid. Digitaalsignaali pistikusse J1 ühenduvad kõik ülejäänud pingevaheldi ühendused ehk PWM, I<sup>2</sup>C, transistoride temperatuuri SPI liides ning kaitseahelate olekusignaali.

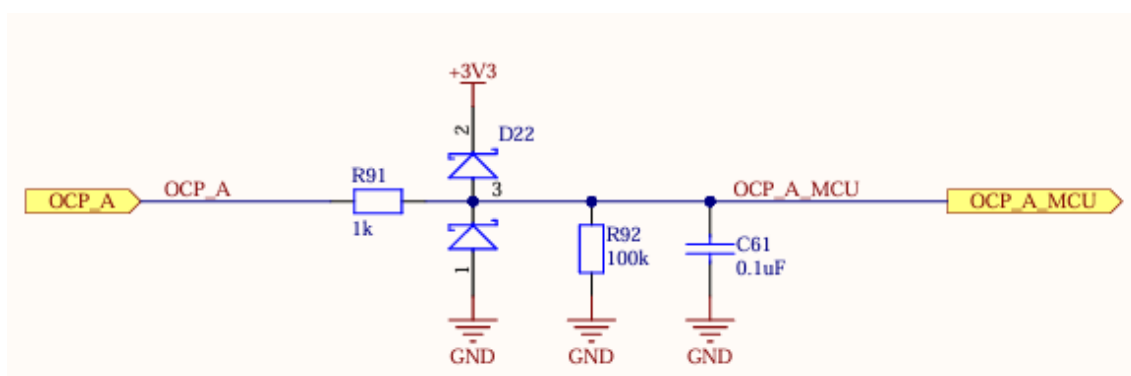
Ühtlasi on kaks pistikut J2 ja J7 mõeldud vormeli poolsete signaalide ühendamiseks. Mõlemad valiti jällegi MQS-seeriast ning lähtuti eelnevalt nimetatud nõuetest. Pistikusse J7 ühenduvad mootori temperatuuri mõõtmiseks vajalikud ühendused. Vaja minevate ühenduste arv sõltub sellest, millisesse konfiguratsiooni on temperatuuri mõõteahel seadistatud. Teise vormeli poolsesse pistikusse J2 ühenduvad kontrolleri toiteahel, CAN ühendused ning resolveri signaalid. Vormeli poolsete signaalide ühendamiseks on lisaks ka pistik J6, kuhu ühenduvad enkoodri signaalid. Antud pistikuks valiti DSUB9-tüüpi pistik, et tagada signaalijuhtmete varjestus. Nimelt on pistiku J6 korpus metallist ning on ühendatud trükkplaadi maakihiga, mille tulemusena on pistikusse ühenduva juhtme varjestus rüüs samuti ühendatud maakihiga. Varjestuse olemasolu enkoodri signaalidel on vajalik häirekindluse suurendamiseks, sest enkooder paikneb mootori küljes, mis on potentsiaalselt häiringute allikaks.

Mõlema mikrokontrolleri programmeerimiseks kasutatakse *Serial Wire* liidest, mis vajab minimaalselt viite ühendust. Juhtimisloogika mikrokontrolleri programmeerimiseks on pistik J3 ning toetavate funktsioonide mikrokontrolleri jaoks pistik J5. Mõlemad pistikud valiti Molex Microfit 3.0 seeriast, mis on tudengivormelis tüüpiliseks valikuks kujunenud programmeerimisliidese pistiku puhul.

Pistikutesse minevad ühendused on välja toodud Lisas 2 skeemil tiitliga „Connectors & input protection“.

### 3.9.1 Kaitseahelad

Tagamaks kontrolleri mõõteahelate ning skeemide kaitse võimalike häiringute nagu ESD ja EMI (*Electromagnetic interference*) eest, on pistikutesse ühenduvatele signaalidele lisatud vastavad kaitseahelad. Pingevaheldi ülevoolu kaitseahelate ning transistoride draiverite olekusignaale, mis ühenduvad mikrokontrolleritega on pistiku ning mikrokontrolleri vahele disainitud Joonis 15 näitena toodud kaitseahel. Viimane koosneb jadatakistist R91, mille eesmärgiks on suurenenud sisendpinge korral piirata voolu ning ühtlasi moodustab koos rööpkondensaatoriga C61 madalpääsfiltri lõikesagedusega 1,5 kHz. Lisaks on skeemis diod D22, mis piirab sisendsignaali pinget vahemiku 0-3,3 V. Rööptakisti R92 eesmärgiks on viia signaal madalale nivoole kui ühendus katkeb või puudub. Sarnaselt on kaitstud ka mikrokontrolleri väljundsignaalid, kuid ilma diodi ning rööptakistita.



Joonis 15. Mikrokontrolleri sisendsignaali kaitseahel pingevaheldi ülevoolu kaitseahela olekusignaali näitel.

Andmeliinidele nagu SPI ja CAN on lisatud jadatakistid, mille eesmärgiks on kiirete pingeniivoode muutusel tekkivate voolupiikide piiramine. Analoo signaalide puhul on kasutusel eelnevates peatükkides käsitletud madalpääsfiltrid. Toiteahela kaitseahel koosneb kolmest komponendist. Esiteks sulavkaitse F4, mis katkestab toiteühenduse ülekoormuse korral. Teiseks diod D17, mis kaitseb kontrolleri toite vale polaarsuse eest. Ning viimaks diod D18, mille eesmärgiks on pingepiikide summutamine.

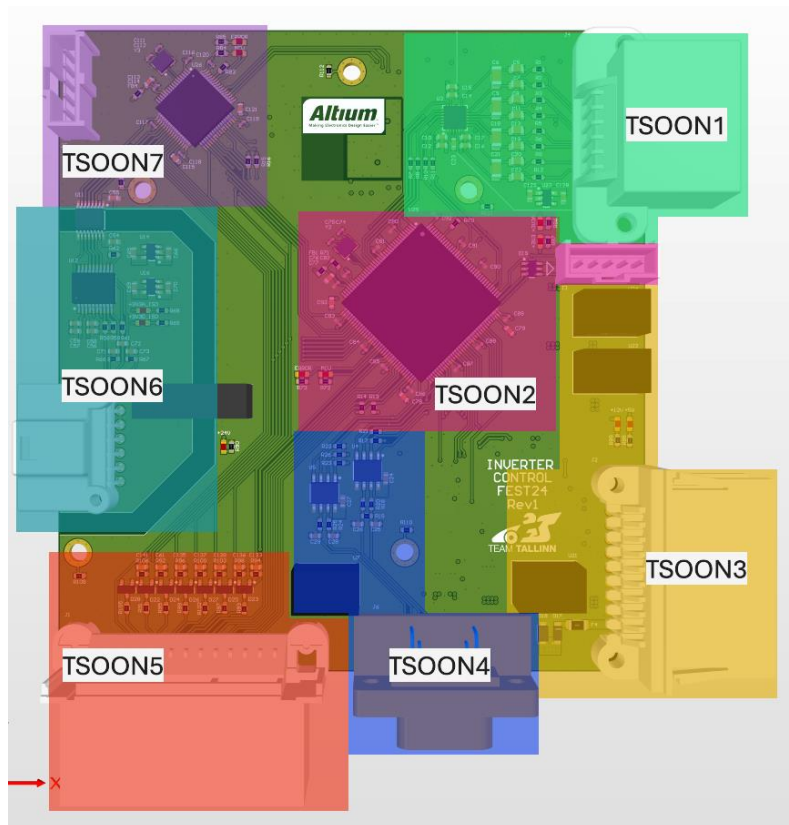
Kirjeldatud kaitseahelad on toodud peamiselt Lisas 2 skeemidel tiitlitega „Connectors & input protection“ ja „MCU IO protection“.

### 3.10 Trükkplaadi disain

Mootori kontrolleri trükkplaadil on mitmeid skeeme, mis sisaldavad endas nii digitaalseid kui ka analoog signaale. Lisaks paikneb kontroller pingevaheldi läheduses, mis on potentsiaalseks häiringute allikaks. Sellest tulenevalt on trükkplaat kuue kihiline, mis tagab ühtlased maa ning toite vasekihid ja lihtsustab signaalide vedamist. Trükkplaadi kihid on toodud Lisades 3-8. Kihtide paigutus on järgnev:

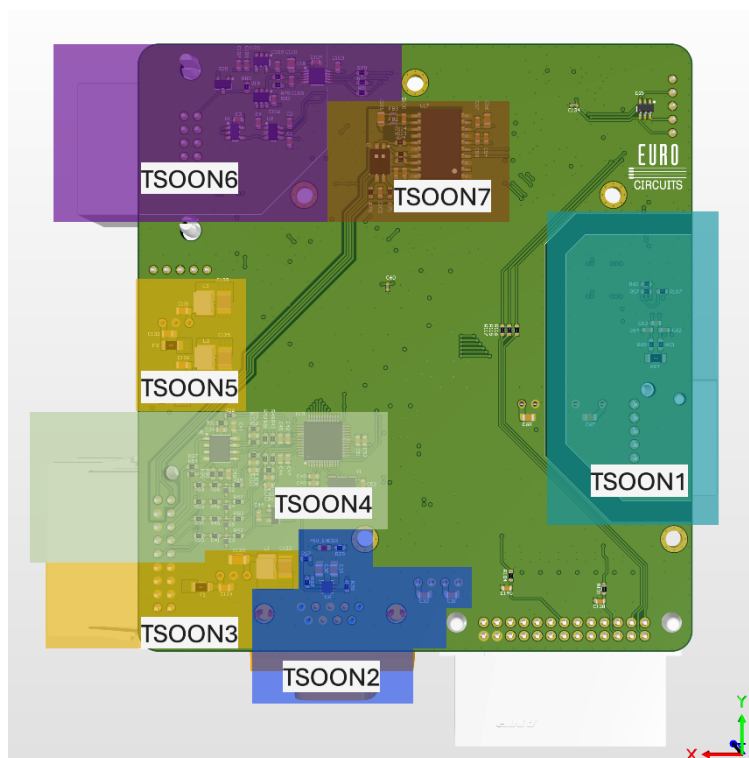
- 1.kiht – signaalide kiht;
- 2.kiht – maa vasekiht;
- 3.kiht – digitaalsüsteemide 3,3 V toite vasekiht;
- 4.kiht – toite polügoonid;
- 5.kiht – maa vasekiht;
- 6.kiht – signaalide kiht.

Komponentide paigutamisel jaotati trükkplaat funktsionaalseteks tsoonideks, et erinevate skeemide signaalid üksteist ei mõjutaks. Pealmine ehk 1.kiht jaguneb seitsmeks funktsionaalseks tsooniks nagu näidatud Joonis 16. Esimeses tsoonis on pingevaheldi sisendpinge ja mootori faasivoolude mõõteahel, mis on paigutatud koos madalpääsfiltritega signaalide pistikule võimalikult lähedale. Teises tsoonis paikneb juhtimisloogika mikrokontroller koos vastava programmeerimispistikuga ning on paigutatud selliselt, et sellega ühenduvad ahelad saaks võimalikult lühikeste radadega mikrokontrolleri viikudega ühendada. Kolmandas tsoonis paiknevad toiteahela muundurid, mis on paigutatud analoog mõõteahelatest eemale. Neljandas tsoonis on enkoodri liides, mis on samuti eraldatud analoog mõõteahelatest. Viiendas tsoonis on pingevaheldi digitaalsed signaalid, mille kaitseahelad on paigutatud vastavale pistikule võimalikult lähedale. Kuuendas tsoonis on isoleeritud mootori temperatuuri mõõteahel koos vastava pistikuga. Seitsmendas tsoonis on toetavate funktsioonide mikrokontroller koos programmeerimispistikuga.



Joonis 16. Trükkplaadi pealmise kihi funktsionaalsed tsoonid.

Funktsionaalsete tsoonide jaotamine tehti ka trükkplaadi alumisel ehk 6.kihil. Alumine kiht jaotati seitsmeks funktsionaalseks tsooniks nagu on näidud Joonis 17. Esimeses tsoonis paikneb isoleeritud mootori temperatuuri mõõteahel. Teises tsoonis enkodeeri liidese toite kaitseahel. Kolmandas tsoonis on osa toiteahela filtritest, mis on paigutatud võimalikult lähedale toitemuunduri viikudele. Neljandas tsoonis paikneb resolveri liides, mille analoogsignaali on paigutatud pistikule võimalikult lähedale, et vähendada võimalike häiringuid trükkplaadil. Viiendas tsoonis on jällegi osa toiteskeemi filtritest, mis on toitemuundurite viikude lähedusse paigutatud. Kuuendas tsoonis paikneb pingevaheldi sisendvoolu mõõteahel, mille kaitseahel on paigutatud pistiku viigu lähedale ning komponentide paigutamisel jälgiti, et analoogsignaali oleks eraldatud digitaalsignaalist. Seitsmendas tsoonis paikneb CAN liides, mis on paigutatud trükkplaadi mõõteahelatest eemale, et vähendada häiringute tõenäosust.



Joonis 17. Trükkplaadi alumise kihi funktsionaalsed tsoonid.

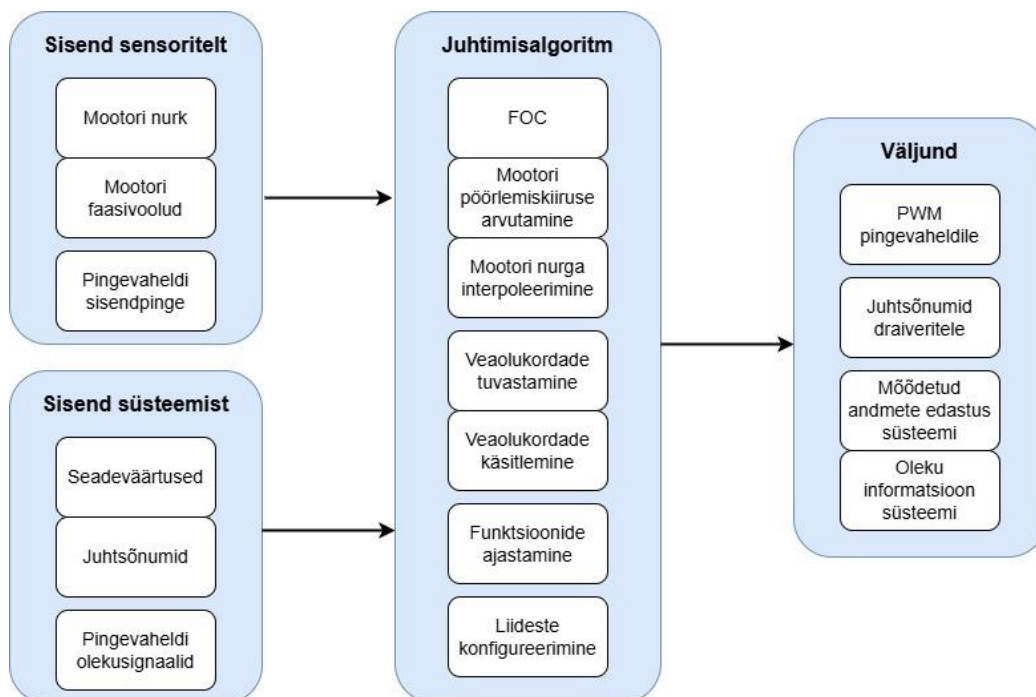
Trükkplaadi 3D vaated on töö Lisades 9-10.

## 4 Juhtimisloogika tarkvara arendus

Juhtimisloogika mikrokontrolleri tarkvara on kirjutatud C programmeerimiskeeles ning selleks kasutati STM32CubeIDE ja STM32CubeMX tarkvara. Juhtimisloogika peamiseks eesmärgiks on teisendada arvulised seadeväärtused kuueks PWM signaaliks kasutades tagasisidestatud vektorjuhtimise FOC (*Field Oriented Control*) algoritmi. Seadeväärtusteks antud mikrokontrolleerile on mootori q- ja d-telje voolu seadeväärtused, mis defineerivad soovivat väändemomenti mootori võllil. Lisaks sellele peab tarkvara tagama järgmisi funktsionaalsusi:

- Andmevahetus teise mikrokontrolleeriga;
- Veaolukordade tuvastamine;
- Draiverite konfigureerimine.

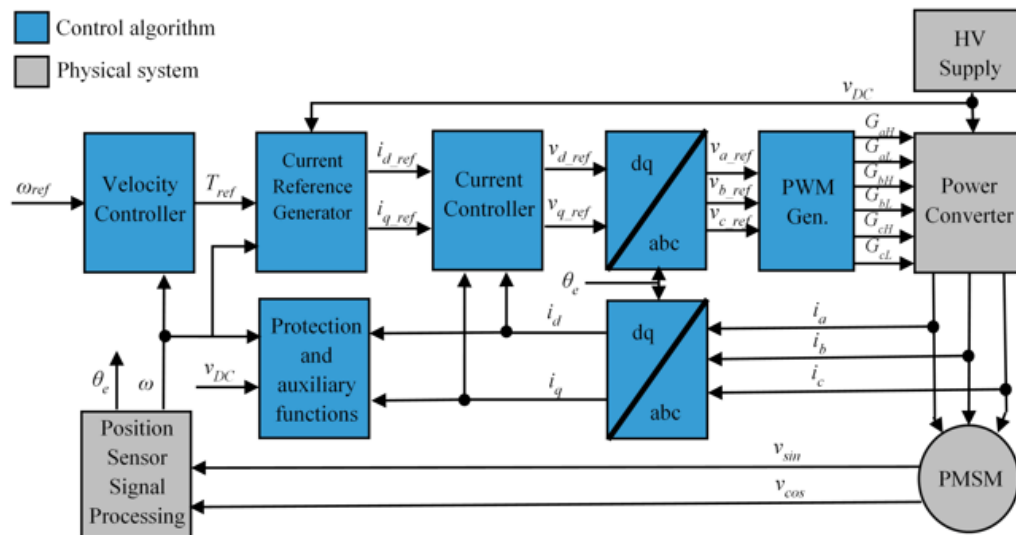
Ülevaatic plokkskeem juhtimisloogika mikrokontrolleeri tarkvara arhitektuurist on toodud Joonis 18. Täpsem sisu Joonis 18 plokkskeemi osadest on kirjeldatud järgnevates peatükkides.



Joonis 18. Juhtimisloogika tarkvara lihtsustatud plokkskeem.

## 4.1 Väljavektorjuhtimise algoritm

Elektrimootori juhtimiseks kasutab kontrolleri FOC ehk väljavektorjuhtimise algoritmi, mille põhimõtteskeem on toodud Joonis 19. Mootori juhtimine baseerub antud algoritmis mootori staatori voolul [32]. Selleks mõõdetakse mootori kõigi kolme faasi voolud ning teisendatakse Park'i ja Clarke'i teisenduste abil d- ja q-telje voolukomponentideks [33], [34]. Teisendused on vajalikud selleks, et siinuseline vool viia üle pöörlevasse koordinaatsüsteemi ehk d- ja q-telje tasanditele. See annab omakorda võimaluse neid kasutada voolukontrolleri PI-kontrollerite tagasisidena. Voolukontroller ise koosneb kahest PI-kontrollerist, üks q-telje ja teine d-telje voolu juhtimiseks. PI-kontrolleri toimimiseks on ühtlasi vaja seadeväärtusi, mille põhjal arvutatakse viga seadeväärtuse ning mõõdetud tulemuse vahel. Voolukomponentide seadeväärtused tuletatakse momendi seadeväärtusest, kasutades tabelleid või reaal-ajas arvutust. Voolukontrolleri väljundiks on d- ja q-telje pinge seadeväärtused, mis teisendatakse tagasi kolmefaasiliseks kasutades vastupidiseid Park'i ja Clarke'i teisendusi. Kolmefaasilise pinge seadeväärtuste põhjal moduleeritakse koos PWM signaali pingevaheldi transistoridele [32].



Joonis 19. FOC põhimõtteskeem [32].

FOC algoritm arvutatakse läbi iga pingevaheldi lülitusperiood. Kuna pingevaheldi lülitussageduseks on 20 kHz, siis maksimaalseks arvutustsükli ajaks on teoreetiliselt 50  $\mu$ s. Lülitusperiood on programmikoodis määratud PWM taimeri konfigureerimisega, mis on toodud Joonis 20. Taimer on konfigureeritud selliselt, et igal poolperioodil tehakse

programmi katkestus. Katkestuse ajal toimub algoritmi arvutamine, mis tähendab et maksimaalseks arvutustsükli ajaks on 25  $\mu$ s. Sellise lähenemise korral on rakendatavad PWM signaalid arvutatud pool perioodi tagasi mõõdetud andmete põhjal. Eeliseks on lühem viivitus tagasiside mõõtmise hetke ja selle põhjal arvutatud PWM signaalide rakendamise hetke vahel.

```
htim1.Instance = TIM1;
htim1.Init.Prescaler = 0;
htim1.Init.CounterMode = TIM_COUNTERMODE_CENTERALIGNED1;
htim1.Init.Period = 5400;
htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim1.Init.RepetitionCounter = 0;
htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
```

Joonis 20. PWM taimeri konfigureerimise programmikood.

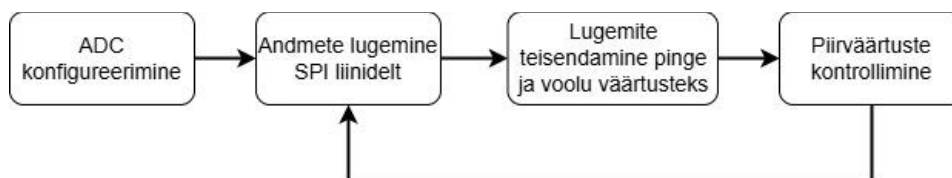
## 4.2 Sisendid sensoritelt

FOC algoritmi toimimiseks ning riistvara kaitseks on vajalik pingevaheldi sisendpinge, mootori faasivoolude ning võlli nurga tagasiside. Sisendpinge väärtust kasutatakse PWM signaalide juhtivusaegade arvutamiseks, voolukontrolleri referentsväärtuste leidmiseks ning kontrollimaks, et pingevaheldi sisendpinge poleks liiga madal ega kõrge. Arendatav pingevaheldi on mõeldud töötama pinge vahemikus 300-800 V [8]. Kasutatav mootor on, aga mõeldud töötama kuni 600 V sisendpingega, mis on ühtlasi ka võistlussarja poolt määratud kõrgeim lubatud pinge [2], [9]. Seega alumiseks piiriks on 300 V ning ülemiseks 600 V. Mootori faasivoolude tagasiside on vajalik voolukontrolleri PI-kontrollerite toimimiseks ning kontrollimaks, et faasivool ei ületaks pingevaheldi ega mootori maksimaalset lubatud piiri. Pingevaheldi puhul on maksimaalseks lubatud faasivoolu amplituud väärtuseks 113 A ning mootoril 86 A, seega arvestatakse madalamat ehk mootori piirväärtust [8], [9]. Mootori võlli nurga mõõtmine on vajalik Park teisenduse tegemiseks, et viia faasivoolud pöörlevale koordinaatasandile ehk d- ja q-telje komponentideks. Ühtlasi arvutatakse nurga muutuse põhjal välja mootori pöörlemiskiirus, mille alusel kontrollitakse, et pöörlemiskiirus ei ületaks maksimaalset lubatud kiirust. Kasutataval mootoril on maksimaalseks lubatud pöörlemiskiiruseks 20 000  $\text{min}^{-1}$  [9]. Juhtimisalgoritmi täpsuse jaoks on oluline, et nimetatud sensorite sisendid mõõdetakse vahetult enne algoritmi arvutamist ehk samuti iga 25  $\mu$ s tagant.



#### 4.2.1 Mootori faasivoolude ning pingevaheldi sisendpinge mõõtmine

Pingevaheldi sisendpinge ning mootori faasivoolud mõõdetakse ADC muunduriga, mis võimaldab kõiki nelja suurust samaaegselt mõõta ning ka läbi SPI liidese edastada [10]. Nimetatud suuruste mõõtmine jaguneb neljaks sammuks, mis on näidatud Joonis 21.



Joonis 21. Mootori faasivoolude ning pingevaheldi sisendpinge mõõtmise protsess.

ADC konfigureerimine on ühekordne protsess, mida tehakse programmikoodi alguses initsialiseerimise osas. Antud protsess on vajalik, et konfigureerida ADC nelja SPI andmeliiniga režiimi. Viimase tulemusel on võimalik neljalt SPI andmeliinilt samaaegselt vastavalt nelja mõõtetulemust lugeda, mis kiirendab oluliselt andmete lugemise protsessi [10]. Edasised kolm sammu tehakse iga juhtimisalgoritmi arvutustsükli korral. Esmalt loetakse neljalt SPI andmeliinilt mõõtetulemused kasutades Joonis 22 toodud programmikoodi. Mõõtetulemuste lugemiseks viiakse SPI liini CS signaal madalale tasemele ning genereeritakse andmete lugemiseks taktsignaali. Tulemused kirjutatakse mikrokontrolleri nelja SPI andmeliini andmeregistritesse, kust iga sisendi info peale CS liini kõrgeks viimist maha loetakse. Peale mõõtetulemuste lugemist teisendatakse ADC lugemid mõõdetud pinge ning voolude väärtusteks. Viimaseks sammuks kontrollitakse, et pinge ning voolude väärtused jäävad määratud piiridesse. Kui lugem on piiridest väljas, siis genereeritakse veasõnum ning pingevaheldi töö peatatakse.

```

void CurrentADC(){
    __HAL_SPI_ENABLE(&hspi5);
    __HAL_SPI_ENABLE(&hspi4);
    __HAL_SPI_ENABLE(&hspi6);
    __HAL_SPI_ENABLE(&hspi3);

    HAL_GPIO_WritePin(ADC_CS_GPIO_Port, ADC_CS_Pin, GPIO_PIN_RESET);
    hspi5.Instance->DR = 0;
    while (__HAL_SPI_GET_FLAG(&hspi5, SPI_FLAG_RXNE) != SET);
    HAL_GPIO_WritePin(ADC_CS_GPIO_Port, ADC_CS_Pin, GPIO_PIN_SET);

    dataBuf[0] = hspi5.Instance->DR;
    dataBuf[1] = hspi4.Instance->DR;
    dataBuf[3] = hspi3.Instance->DR;
    dataBuf[2] = hspi6.Instance->DR;

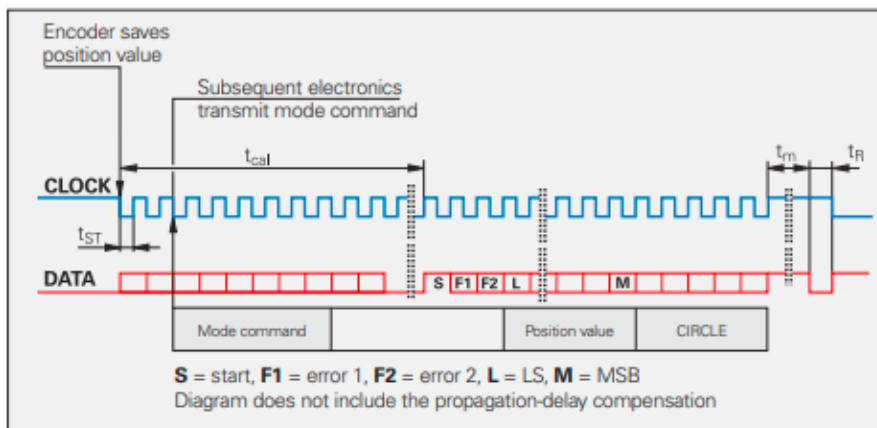
    __HAL_SPI_DISABLE(&hspi5);
    __HAL_SPI_DISABLE(&hspi4);
    __HAL_SPI_DISABLE(&hspi6);
    __HAL_SPI_DISABLE(&hspi3);
}

```

Joonis 22. Mootori faasivoolude ning pingevaheldi sisendpinge ADC andmete lugemise programmikood.

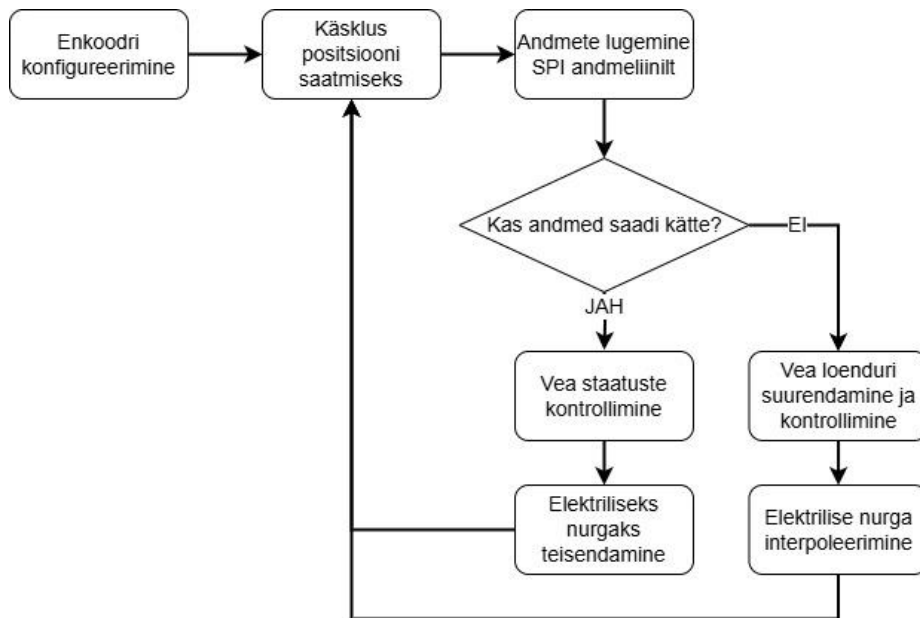
#### 4.2.2 Mootori võlli nurga lugemine enkoodriga

Kasutatava enkoodri suhtlus baseerub EnDat 2.1 protokollil, mille füüsiline kiht koosneb andme- ja taktsignaali diferentsiaalsignaalist [12]. Protokoll realiseerimiseks kasutatakse mikrokontrolleri SPI liidest. EnDat 2.1 protokoll andmepaketi ülesehitus on toodud Joonis 23. Positsiooni andmete saamiseks tuleb esmalt saata peale kahte taktsignaali takti käsklussõnum. Seejärel tuleb oodata kuni enkooder on valmis positsiooni edastama ehk kuni andmeliinile saadetakse andmepaketi algus bitt. Viimasele järgneb kaks veastaatuse bitti ning seejärel enkoodri positsiooni väärtus [35].



Joonis 23. EnDat 2.1 protokoll ülesehitus [35].

Enkoodrilt mootori nurga lugemise protsess juhtimisloogika tarkvaras on näitatud Joonis 24. Esmalt tehakse enkoodri ühekordne konfigureerimine, mille käigus tehakse enkoodrile tarkvaraline taaskäivitus ning vea staatuste eemaldamine. Edasised sammud tehakse iga arvutustsükli korral alustades enkoodrile käsklussõnumi saatmisest. Seejärel loetakse SPI andmeliinilt saadetud andmepakett ning kontrollitakse, kas sisse loetud andmed vastavad Joonis 23 toodud struktuurile. Antud teadmine on oluline, et otsustada kas uut positsiooni väärtust saab kasutada. Kui andmepakett vastab oodatud struktuurile, siis kontrollitakse andmepaketi kahte vea staatuse bitti, mis inditseerivad enkoodri enda poolt tuvastatud veaolukorda. Viimaseks teisendatakse mõõdetud nurk elektriliseks nurgaks, sest enkooder mõõdab mootori võlli mehaanilist positsiooni ning FOC jaoks on vajalik just mootori elektriline nurk.



Joonis 24. Enkoodri andmete lugemise protsess.

Püsimagnetmootor teeb ühe mehaanilise pöörde jooksul pooluspaaride arv korda elektrilist pööret [36]. Sellest tuletatuna saab mootori elektrilise nurga tuletada valemiga (4.1):

$$\theta_e = N \cdot \theta_m. \quad (4.1)$$

kus  $\theta_e$  – mootori elektriline nurk, rad,  
 $N$  – mootori pooluspaaride arv,  
 $\theta_m$  – mootori mehaaniline nurk, rad.

Kui tekib olukord, kus saadetud andmepaketist ei õnnestunud positsiooni andmeid saada, siis koheselt vea signaali ei genereerita vaid suurendatakse kadunud lugemite loendurit. Vea signaal saadetakse kui viielt järjestikult positsiooni küsimiselt ei õnnestunud andmeid saada. Lisaks on veel kaks vea olukorda, mille puhul saadetakse vea sõnum. Esiteks kui enkooder annab vea staatuse bitiga teada veaolukorrast. Teiseks kui enkoodri toiteahelas olev kaitseahel annab märku veast toiteahelas. Vea sõnumite genereerimiseks kasutatakse Joonis 25 toodud programmikoodi.

```
void CheckFault(FaultCondition *faultCondition, uint32_t message){
    if(message&&0x02){
        ErrorOccured(faultCondition, ENCODER_ERROR, ENCODER_INTERNAL);
    }
    if(missedReadings >= ENCDR_MAX_MISSED_READINGS){
        ErrorOccured(faultCondition, ENCODER_ERROR, ENCODER_TIMEOUT);
    }
    if(!(HAL_GPIO_ReadPin(ENCDR_FLT_GPIO_Port, ENCDR_FLT_Pin))){
        ErrorOccured(faultCondition, ENCODER_ERROR, ENCODER_SUPPLY);
    }
}
```

Joonis 25. Enkoodri vea sõnumite genereerimise programmikood.

Kui positsiooni andmeid ei õnnestunud saada ja kadunud andmete loendur jääb alla viie, siis arvutatakse uus elektriline nurk interpoleerimise teel kasutades valemit (4.2). Selleks on vajalik teada mootori pöörlemiskiirust ning aega viimasest nurga väärtuse uuendamisest.

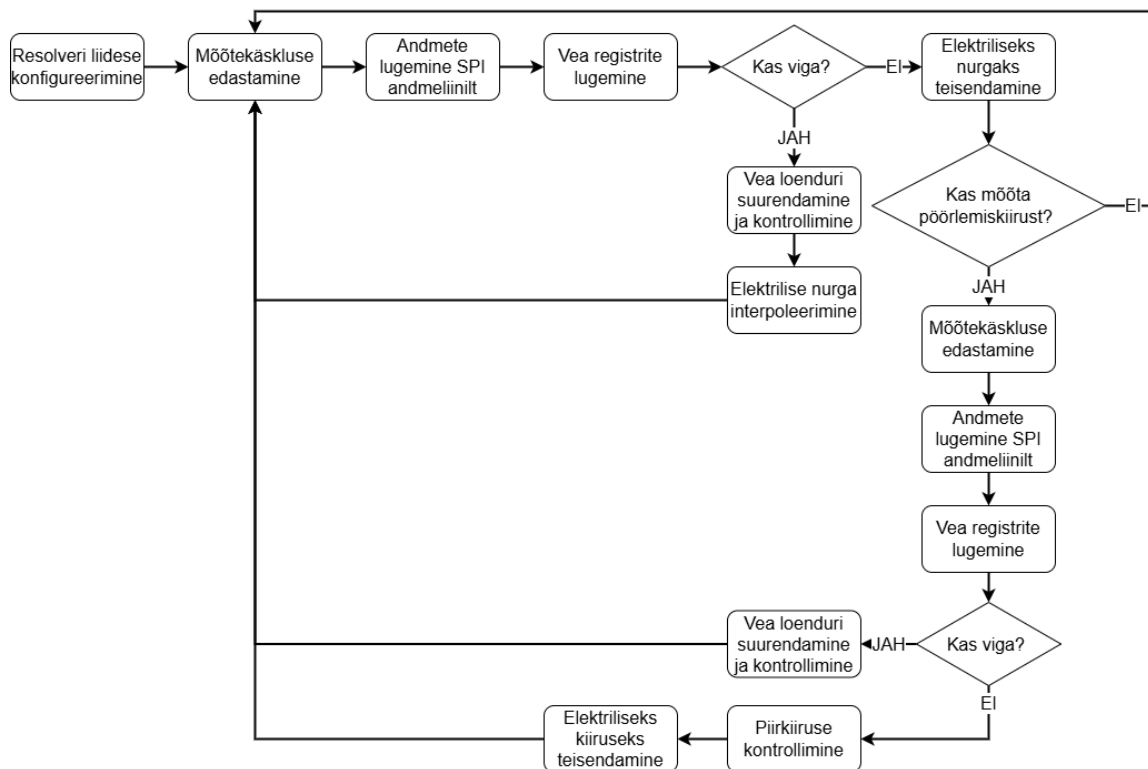
$$\theta_{uus} = \omega_e t + \theta_{vana}. \quad (4.2)$$

kus  $\theta_{uus}$  – interpoleeritud mootori elektriline nurk, rad,  
 $\omega_e$  – mootori elektriline kiirus, rad/s,  
 $t$  – aeg viimasest nurga muutusest, s,  
 $\theta_{vana}$  – eelmine mootori elektriline nurk, rad.

### 4.2.3 Mootori võlli nurga ja kiiruse lugemine resolveriga

Muundur AD2S1210, mida kasutatakse resolveri liidestamiseks võimaldab edastada nii mootori võlli asendit kui ka pöörlemiskiirust [17]. Andmete lugemiseks kasutatakse mikrokontrolleri SPI liidest ning mõõtmise protsess on toodud Joonis 26. Esimese sammuna tehakse resolveri ühekordne konfigureerimine programmikoodi initsialiseerimise osas. Konfigureerimise käigus määratakse resolveri ergutuspinge

sagedus, mõõtetulemuste resolutsioon ning vea registrite piirväärtused. Resolveri liides suudab küll edastada nii mootori nurka kui ka pöörlemiskiirust, kuid mitte sama mõõtetsükliga. Kuna mootori nurk on ajas pidevalt muutuv suurus, siis antud suurust mõõdetakse tihedamini kui mootori pöörlemiskiirust. Mõlemal juhul algab mõõteprotsess käskluse edastamisega, mille tulemusel arvutab ja salvestab soovitud mõõtetulemuse. Sellele järgneb mõõtetulemuste edastamine läbi SPI liidese. Mõlemal juhul kontrollitakse esmalt vea registrit, mille põhjal otsustatakse, kas saadetud andmeid saab kasutada. Kui mitte, siis suurendatakse kadunud andmete loendurit ning genereeritakse viga kui viis järjestikust mõõtetulemust on puudu. Kui loendur jääb alla viie piiri, siis nurga puhul interpoleeritakse elektrilist nurka kasutades valemit (4.2). Kui probleeme ei esine siis teisendatakse mehaanilised suurused elektriliseks ning pöörlemiskiiruse puhul kontrollitakse piirkiirust.



Joonis 26. Resolveri mõõtetulemuste lugemise programmikood.

### 4.3 Mootori pöörlemiskiiruse arvutamine

Mootori pöörlemiskiirus on vajalik peamiselt kolmel eesmärgil: mootori maksimaalse pöörlemiskiiruse jälgimine, nurga interpoleerimine ning momendi teisendamisel d- ja q-telje voolu seadeväärtusteks. Kuna resolveri liides suudab edastada mootori

pöörlemiskiirust, siis arvutamine on vajalik enkoodri kasutamise puhul. Pöörlemiskiiruse saab välja arvutada mootori nurga muutuse põhjal, kui on teada kahe nurga mõõtmise vaheline aeg kasutades valemit (4.3):

$$\omega = \frac{\Delta\theta}{t}. \quad (4.3)$$

kus  $\omega$  – mootori mehaaniline pöörlemiskiirus, rad/s,  
 $\Delta\theta$  - mehaanilise nurga muutus, rad,  
 $t$  – kahe nurga mõõtmise vaheline aeg, s.

Elektrilise nurga interpoleerimiseks on vajalik mootori elektriline kiirus, mis on mehaanilisest pöörlemiskiirusest pooluspaaride arv korda suurem. Seega elektrilisel teisendusel saab lähtuda nurga teisendamise valemist (4.1) asendades nurga kiirusega. Programmikood pöörlemiskiiruse arvutamiseks ning elektriliseks teisenduseks on toodud Joonis 27. Aja mõõtmiseks kasutatakse arvutustsükli perioodi ning loendurit, mis kirjeldab mitu arvutustsükli oli kahe nurga mõõtmise vahel.

```
void CalculateSpeed(Motor *motor, uint8_t counterVal, float period){
    float deltaAngle;
    float deltaTime;

    deltaAngle = motor->mechAngle - motor->prevMechAngle;
    if(deltaAngle > PI_VALUE){
        deltaAngle -= TWO_PI_VALUE;
    }
    else if(deltaAngle < NEG_PI_VALUE){
        deltaAngle += TWO_PI_VALUE;
    }
    deltaTime = counterVal * period;

    motor->mechSpeed = deltaAngle/deltaTime;
    motor->elSpeed = motor->mechSpeed * motorParameter.polePairs;
}
```

Joonis 27. Mootori pöörlemiskiiruse arvutamise programmikood.

#### 4.4 Sisend süsteemist

Sisendid süsteemist saab jagada kaheks: sisendid toetavate funktsioonide mikrokontrollerilt ning sisendid pingevaheldilt. Üheks sisendiks toetavate funktsioonide mikrokontrollerilt on d- ja q-telje voolu seadeväärtused PI-kontrolleritele. Teiseks 8-bitine juhtsõnum, mille struktuur on toodud Tabel 8. 1.biti oleku määrab toetavate

funktsioonide mikrokontroller, mis jälgib transistoride draiverite vastavat olekusignaali. 2.biti saavad kõrgeks kirjutada mõlemad mikrokontrollerid, kui on tuvastatud veaolukord. Käsklus pingevaheldi käivitamiseks tuleb toetavate funktsioonide mikrokontrollerilt, kuid juhtloogika mikrokontroller kontrollib lisaks, kas kõik tingimused selleks on täidetud. Transistoride draiverite registreite lugemise käsklus tuleb samuti teiselt mikrokontrollerilt, kui kasutaja soovib manuaalselt draivereid diagnoosida. Vastasel juhul loetakse vastavaid registreid ainult vea olukorras, sest kuue draiveri registreite lugemine I<sup>2</sup>C liidesega on ajamahukas tegevus, mille jaoks kulub mitmeid arvutustsükleid. 5.biti määrab samuti toetavate funktsioonide mikrokontroller kui soovitakse vea teated kustutada. Viimast tehakse programmikoodi initsialiseerimise osas ning kui vastav sisend tuleb kasutajalt.

Tabel 8. Kontrolleri juhtsõnumi struktuur.

Juhtsõnumi bitt	Biti väärtuse tähendus
1.bitt	1 – transistoride draiverid on valmis PWM signaali vastu võtma; 0 – draiverid ei ole valmis PWM signaali vastu võtma
2.bitt	1 – kontroller on vea seisundis; 0 – vigu pole
3.bitt	1 – käsklus pingevaheldi käivitamiseks; 0 – pingevaheldi töö peatatud
4.bitt	1 – käsklus transistoride draiverite registreite lugemiseks; 0 – registreid loetakse vea tekkimisel
5.bitt	1 – kustuta veateated; 0 – veateated salvestatakse

Sisendid pingevaheldilt on ülekoormuse kaitseahelate ning transistoride draiverite olekusignaalid. Viimaste jälgimise eesmärgiks on vea olukordade tuvastamine. Kuna antud signaalid on dubleeritud ka teisele mikrokontrollerile, siis juhtimisloogika mikrokontroller loeb signaalid küll sisse ning vea korral lülitab pingevaheldi välja, kuid vea sõnumit ei genereerita. Vea sõnumid genereeritakse ainult transistoride draiverite vea põhjuste edastamisel. Nimelt kui transistoride draiveri olekusignaali annab märku veast, siis juhtimisloogika mikrokontroller loeb läbi I<sup>2</sup>C liidese draiverite vea registreid, mis sisaldavad infot vea tekke kohta. Viimane edastatakse edasi toetavate funktsioonide mikrokontrollerile.

## 4.5 Vealukordade käsitlemine

Vealukorra tuvastamisel kutsutakse programmikoodis välja esimese sammuna Joonis 28 toodud veakoodi genereerimise funktsioon. Antud funktsioonile antakse sisendiks, millises ahelas on viga ning mis on täpsem põhjus. Mõlemad on kaheksa bitised väärtused, mis salvestatakse vigade puhvrissse ning hiljem kombineeritakse üheks 16 bitiseks veakoodiks. Võimalikud veakoodid ning kirjeldused on toodud Lisas 11.

```
void ErrorOccured(ControlData *controlData, FaultCondition *faultCondition,
uint8_t errorType, uint8_t errorDesc){
    uint8_t i = 0;

    faultCondition->inverterError = true;
    controlData->status |= 0x02;
    HAL_GPIO_WritePin(ERROR_LED_GPIO_Port, ERROR_LED_Pin, GPIO_PIN_SET);

    while(faultCondition->errorType[i] && (i < TOTAL_ERRORS)){
        if((faultCondition->errorType[i] == errorType) &&
(faultCondition->errorDesc[i] == errorDesc)){
            return;
        }
        else{
            i++;
        }
    }
    if(i < TOTAL_ERRORS){
        faultCondition->errorsDetected++;
        faultCondition->errorType[i] = errorType;
        faultCondition->errorDesc[i] = errorDesc;
    }
}
```

Joonis 28. Veakoodide genereerimise programmikood.

Veakoodi genereerimise käigus muudetakse ka juhtsõnumi vea bitt kõrgeks, mis inditseerib kontrolleri veaseisundit. Viimase tulemusel lülitatakse pingevaheldi välja kuniks vea olukord on lahendatud.

## 4.6 Funktsioonide ajaline planeerimine

Juhtimisloogika mikrokontrolleri puhul on oluline, et algoritm arvutatakse läbi ning väljund PWM signaalide juhtivusajad 25 µs jooksul. Selleks uuriti kui kaua algoritmi erinevad funktsioonid aega võtavad. Katse käigus kasutati mootori võlli asendi määramiseks enkoodrit, sest viimane oli paigaldatud katsetustel kasutatud mootorile. Seega analüüs ei võta arvesse resolveri liidesega seotud funktsioonide ajalisi kestvusi.



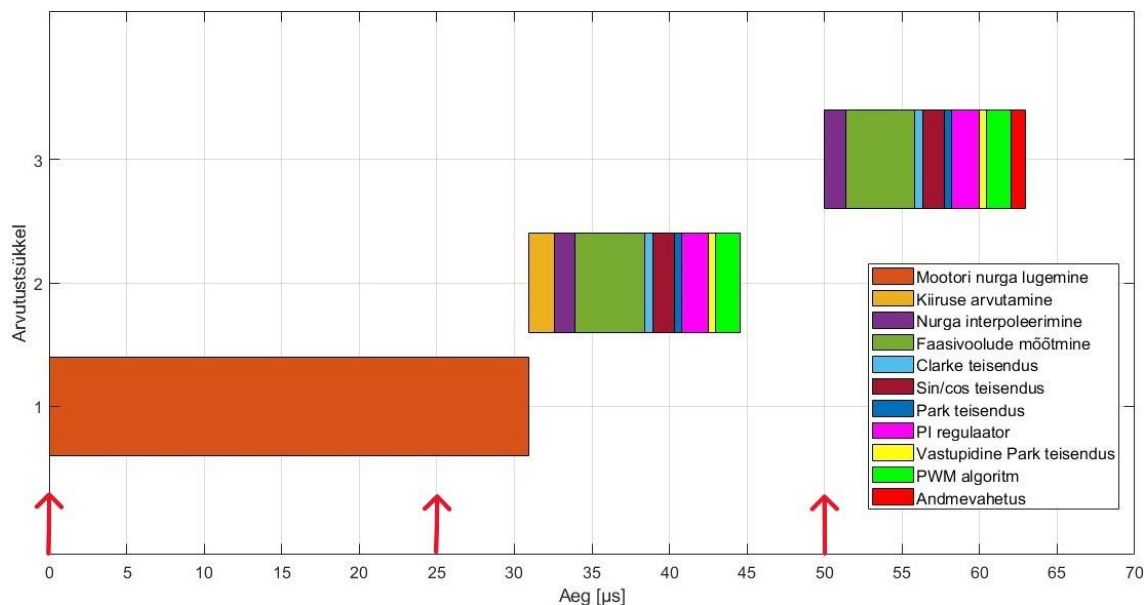
Tulemused mõõdeti prototüübi esimestel katsetustel, lisades programmikoodi taimerit, millega mõõdeti funktsioonide ajalisi pikkusi täpsusega 9,3 ns. Tabel 9 on näidatud funktsioonide ajalised varieeruvused. Kõige enam varieeruvad enkoodri lugemine vahemikus 30,4 – 30,9  $\mu$ s ning PWM algoritm vahemikus 1 – 1,6  $\mu$ s. Kõige enam kulub aega mootori võlli nurga mõõtmiseks enkoodri abil, mis ühtlasi ületab ka soovitud 25  $\mu$ s piiri. Ülejäänud funktsioonid jäävad vahemiku 1 – 2  $\mu$ s, välja arvatud faasivoolude mõõtmine, mis võtab aega 4,4  $\mu$ s.

Tabel 9. Juhtimisloogika mikrokontrolleri peamiste funktsioonide ajalised kestvused.

Funktsioon	Väikseim mõõdetud aeg [ $\mu$ s]	Suurim mõõdetud aeg [ $\mu$ s]
Andmevahetus	0,93	0,93
Mootori nurga lugemine	30,39	30,91
Kiiruse arvutamine	1,37	1,65
Nurga interpoleerimine	1,31	1,37
Faasivoolude mõõtmine	4,37	4,46
Clarke'i teisendus	0,52	0,54
Sin/cos teisendus	1,26	1,39
Park'i teisendus	0,43	0,44
PI regulaatorid	1,70	1,74
Vastupidine Park'i teisendus	0,39	0,48
PWM algoritm	1,00	1,61

Kuna enkoodri kasutamisel kulub üle ühe arvutustsükli aja nurga lugemisele, siis otsustati nurga väärtust uuendada igal kolmandal arvutustsüklil. Ülejäänud tsüklite jooksul interpoleeritakse nurga väärtust kiiruse põhjal kasutades valemit (4.2). Joonis 29 on toodud funktsioonide ajastamine kolmel järjestikul arvutustsüklil. X-teljel on märgitud punaste nooltega programmi katkestused, mis inditseerivad uue arvutustsükli algust. Kui sellel hetkel on mõni algoritmi funktsioon pooleli, siis uus tsükkel algab siis kui kõik algoritmi funktsioonid on lõpetatud. Seda seetõttu, et kogu algoritm tehakse läbi katkestuse sees. Esimene katkestus kulub täielikult enkoodri nurga väärtuse lugemisele, mis ühtlasi lükkab edasi ka teise tsükli algusaega. Teine ja kolmas tsükkel erinevad selle poolest, et teise tsükli alguses arvutatakse mõõdetud mootori nurga alusel välja mootori pööremiskiirus. Viimast kolmandas tsüklis uuesti ei arvutata, vaid kasutatakse nurga

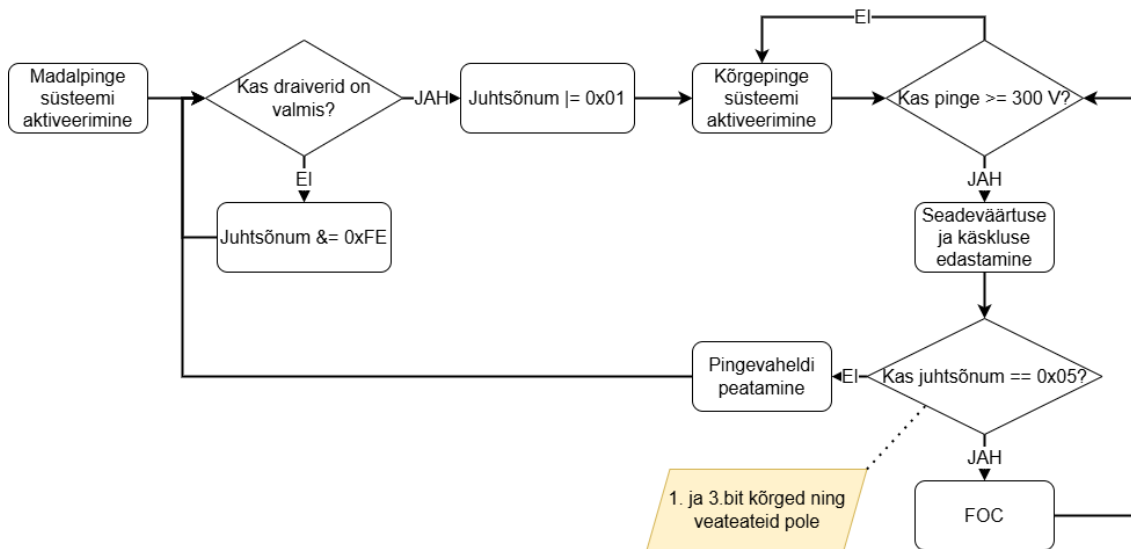
interpoleerimiseks. Peale kolmandat tsüklit kordub uuesti mootori nurga lugemine enkoodriga.



Joonis 29. Juhtimisloogika mikrokontrolleri funktsioonide ajastamine.

## 4.7 Pingevaheldi käivitamise protseduur

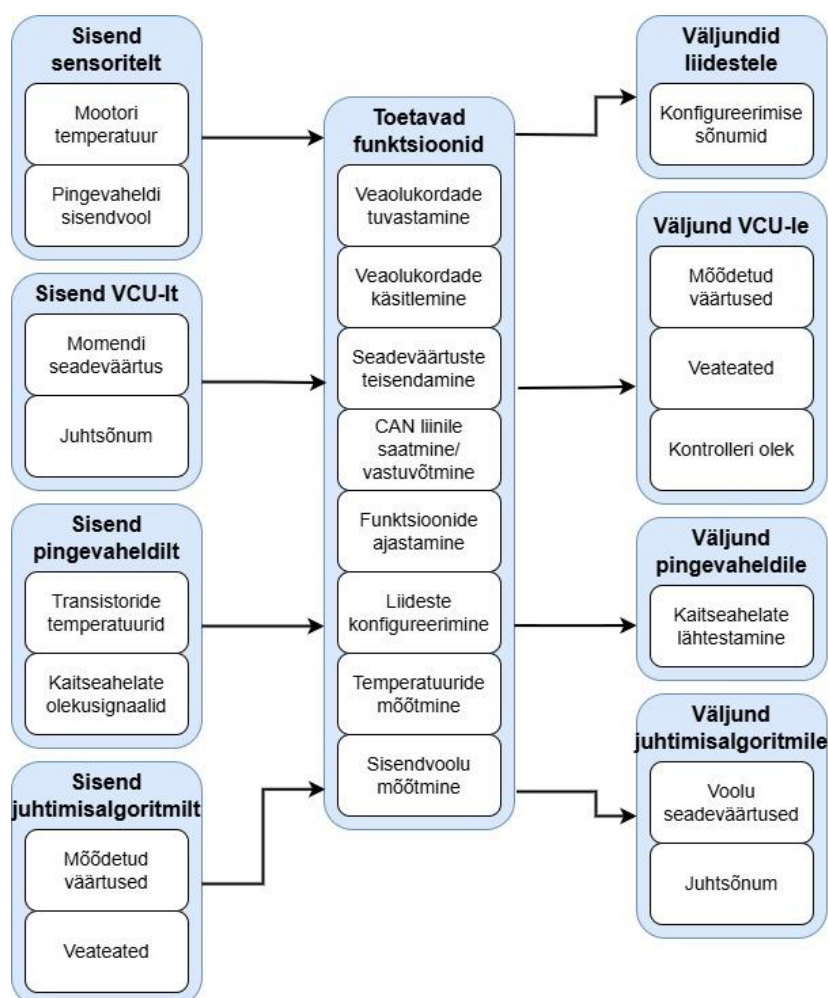
Pingevaheldi käivitamiseks tuleb esmalt rakendada nii kontrolleri kui ka pingevaheldile madalpinge. Kui mõlemal on madalpinge olemas, siis transistoride draiveritele veel PWM signaale ei saadeta. Ühtlasi kasutatakse FOC algoritmist ainult tagasiside osa ehk Park'i ja Clarke'i teisendusi. Kui transistoride draiverid on madalpinge toitega ühendatud ning vea seisundit pole, siis annavad draiverid olekusignaaliga märku, et need on valmis PWM signaale vastu võtma. Selle info põhjal kirjutab toetavate funktsioonide mikrokontroller Tabel 8 juhtsõnumi esimese biti kõrgeks. Järgmiseks tuleb vormelis aktiveerida kõrgepinge süsteem, mille käigus laetakse pingevaheldi kondensaatorid. Peale seda tuleb edastada toetavate funktsioonide mikrokontrollerile läbi CAN võrgu pingevaheldi käivitamise käsklus ning momendi seadeväärtus. Viimane teisendatakse d- ja q-telje voolu seadeväärtusteks ning edastatakse koos käsklusega juhtimisalgoritmi mikrokontrollerile. Kui vealukorda pole, siis käivitatakse pingevaheldi ning seadeväärtused edastatakse FOC algoritmile. Pingevaheldi töötab seni kuni tuleb käsklus peatamiseks, kas VCU või algoritmi enda poolt. Kirjeldatud protseduur on toodud Joonis 30.



Joonis 30. Pingevaheldi käivitamise protseduur.

## 5 Toetavate funktsioonide tarkvara arendus

Toetavate funktsioonide tarkvara arenduseks kasutati C programmeerimiskeelt ning samu programme, mis juhtloogika tarkvara arenduse puhul. Toetavate funktsioonide mikrokontrolleri ning selle juurde kuuluva tarkvara peamiseks eesmärgiks on andmevahetus VCU ning juhtimisalgoritmi mikrokontrolleriga. Sealjuures ka mootori ning transistoride temperatuuride mõõtmine, momendi seadeväärtuste teisendamine q- ja d-telje voolu seadeväärtusteks ning veaolukordade tuvastamine. Ülevaade toetavate funktsioonide tarkvarast on näidatud Joonis 31 ning täpsem sisu on kirjeldatud järgnevas peatükis.

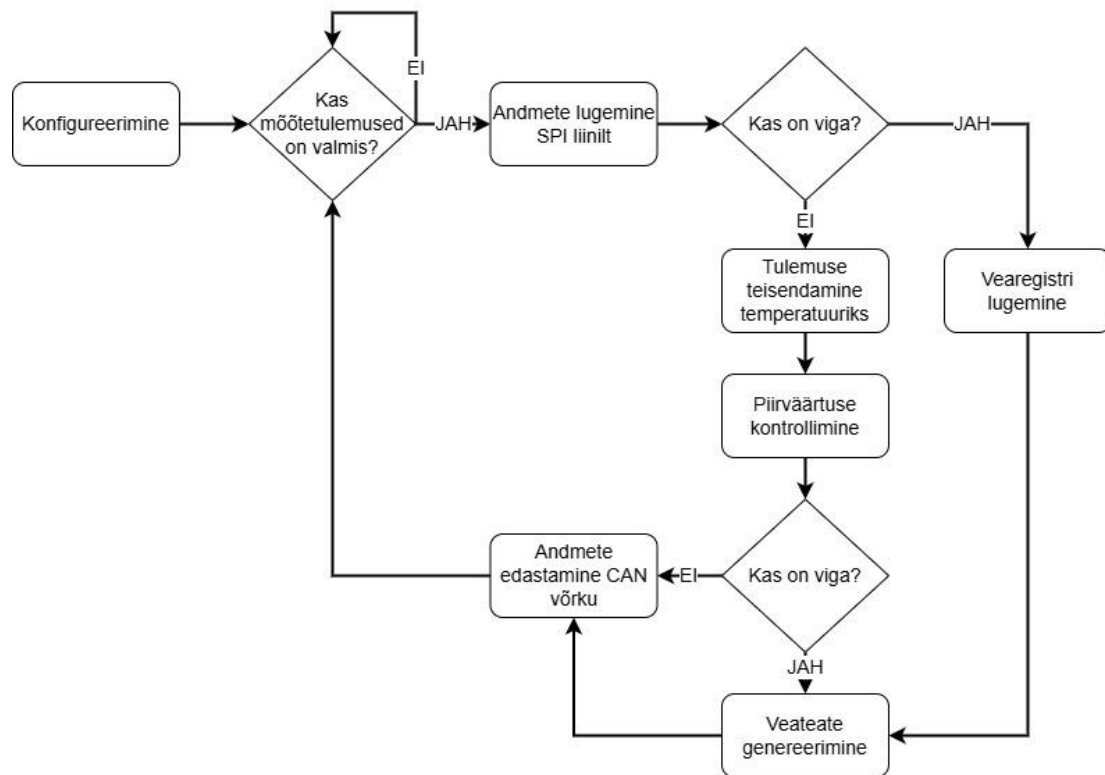


Joonis 31. Toetavate funktsioonide tarkvara lihtsustatud ülevaade.

## 5.1 Temperatuuride mõõtmine

Vältimaks transistoride ning mootori ülekuumenemist, mõõdab mootori kontrolleri nimetatud komponentide temperatuure. Transistoride temperatuuri mõõteahel koos ADC-ga on paigutatud pingevaheldisse ning kontrolleri edastatakse andmed üle SPI liidese. Kokku edastatakse kaheksa mõõtetulemust: kuue transistori pinna temperatuur ning transistoride jahutusplaati siseneva ja väljuva vee temperatuurid. Kontrolleri ülesandeks on kuue transistori maksimaalse temperatuuri jälgimine ning piirväärtuse ületamise korral veateate saatmine. Ühtlasi kõigi kaheksa temperatuuri lugemi CAN võrku edastamine.

Sama SPI liidese kaudu toimub ka mootori temperatuuri andmete saatmine. See kumma mõõteahela andmeid edastatakse, määratakse CS signaaliga. Mootori temperatuuri mõõtmise protsess on toodud Joonis 32. Esimese sammuna tehakse programmikoodi initsialiseerimise faasis ühekordne konfigureerimine MAX31865 muundurile. Selle käigus konfigureeritakse muundur automaatse mõõtmise režiimi, määratakse sensori ühendusmeetod ning madalpääsfiltri lõikesagedus. Automaatse mõõtmise režiimis annab muundur digitaalse väljundiga märku, kui mõõtetulemused on valmis edastamiseks. Ühendusmeetodiga määratakse, kas sensorit mõõdetakse 2-,3- või 4-juhtme meetodiga. Madalpääsfiltri lõikesageduseks saab valida 50 või 60 Hz. Kuna temperatuuri muutused mootoris on aeglase iseloomuga, siis sageduseks valiti 50 Hz. Kui muundur annab märku andmete olemasolust, siis loetakse need SPI liidese abil sisse. Saadetud andmed sisaldavad mõõtetulemust ning vea indikatsiooni bitti. Kui bitt annab märku veast, siis loetakse lisaks muunduri vearegistrilt, mille alusel genereeritakse veakood. Kui vea bitt viga ei näita, siis teisendatakse mõõtetulemus temperatuuriks. Teisendamiseks kasutatakse lineaarset teisendust, kasutades valemit (5.1) [19]. Saadud tulemust kontrollitakse ning kui temperatuur on suurem kui 130 °C, siis genereeritakse jällegi veakood. Nii mõõtetulemused kui ka veateated saadetakse vormeli CAN võrku.



Joonis 32. Mootori temperatuuri mõõtmise protsess.

ADC lugem teisedatakse temperatuuriks järgmise valemiga [19]:

$$T \approx \frac{x}{32} - 256. \quad (5.1)$$

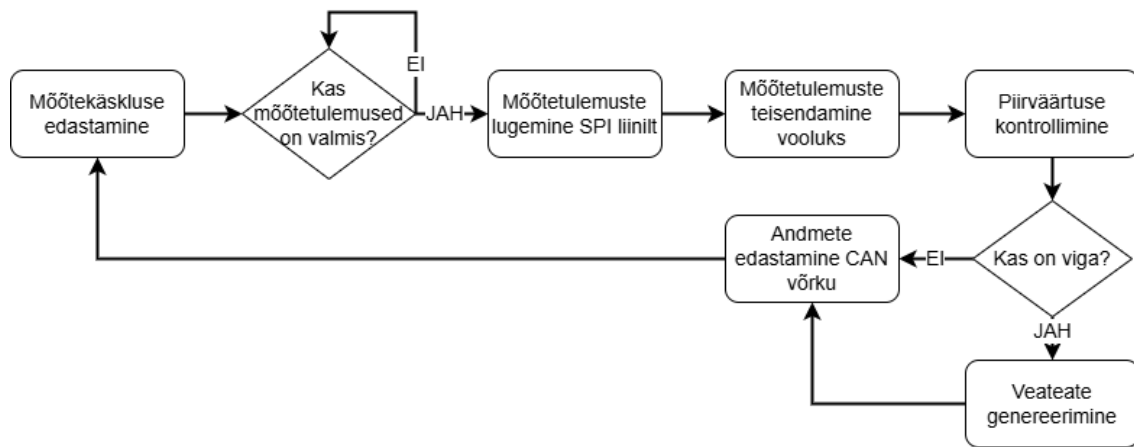
kus  $T$  – temperatuur, °C,

$x$  – ADC lugem.

## 5.2 Sisendvoolu mõõtmine

Kuna pingevaheldi sisendvool pole FOC algoritmi toimimise jaoks oluline, siis mõõdetakse seda toetavate funktsioonide mikrokontrolleriga. Sisendvoolu info on vajalik pingevaheldi ülekoormuse tuvastamiseks ning kasutatava võimsuse arvutamiseks. Mõõtetulemuste edastamiseks kasutatakse SPI liidest ning mõõtmise protseduur on toodud Joonis 33. Esimese sammuna edastatakse ADC-le mõõtekäsklus viies SPI liini CS signaal hetkeks kõrgele nivoole ning madalaks tagasi. Peale seda oodatakse kuni mõõtetulemused on valmis edastamiseks. ADC annab sellest märku SPI andmeliini madalaks viimisega. Kui mõõtetulemused on valmis, siis loetakse need SPI liinilt sisse ning teisendatakse voolu väärtuseks. Järgmiseks kontrollitakse, kas sisendvool jääb lubatud piiridesse. Esimesel pingevaheldi prototüübil on lubatud piiriks 100 A. Kui

mõõdetud tulemus ületab lubatud piiri, siis genereeritakse veateade ning pingevaheldi töö peatatakse. Nii mõõtetulemus kui ka veateade edastatakse vormeli CAN võrku.



Joonis 33. Pingevaheldi sisendvoolu mõõtmise protsess.

Sisendvoolu piirväärtuse kontrollimisel arvestatakse, et sisendvool võib olla ka negatiivne ehk mootor on regeneratiivse pidurdamise režiimis. Vältimaks veateate genereerimist lühiajaliste häiringute ning voolu piikide peale on kontrollahelasse lisatud puhver. See tähendab, et veateade genereeritakse siis kui viis järjestikust mõõtmist on olnud piirväärtustest väljas. Vigade tuvastamiseks kasutatakse Joonis 34 toodud programmikoodi.

```

void CheckOverCurrent(int16_t current, ControlData *controlData,
FaultCondition *faultCondition){
    if(current > MAX_DC_CURRENT){
        currentBuffering.ovlCount++;
        if(currentBuffering.ovlCount >= OVER_CURRENT_ALLOWED_READINGS){
            ErrorOccured(controlData, faultCondition,
                DC_CURRENT_ERROR, DC_OVERCURRENT);
        }
    }
    else{
        currentBuffering.ovlCount = 0;
    }

    if(current < MAX_REGEN_CURRENT){
        currentBuffering.regenOvlCount++;
        if(currentBuffering.regenOvlCount >=
OVER_CURRENT_ALLOWED_READINGS){
            ErrorOccured(controlData, faultCondition, DC_CURRENT_ERROR,
                DC_REGEN_OVERCURRENT);
        }
    }
    else{
        currentBuffering.regenOvlCount = 0;
    }
}

```

Joonis 34. Pingevaheldi sisendvoolu piirväärtuste kontrollimise programmikood.

### 5.3 Seadeväärtuste teisendamine

VCU sisend mootori kontrolleri jaoks on soovitatav väändmoment mootori võllil. FOC algoritmi toimimiseks tuleb momendi seadeväärtus teisendada d- ja q-telje seadeväärtusteks. Viimase jaoks leidub mitmeid erinevaid viise. Üheks levinuimateks on MTPA (*Maximum Torque per Ampere*) ja MTPV (*Maximum Torque per Voltage*). MTPA algoritmiga arvutatakse voolu seadeväärtused mootori konstantse väändmomenti piirkonnas ning MTPV algoritmiga väljanõrgestamise piirkonnas. Nimetatud meetodite puhul arvutatakse voolu seadeväärtused mootori parameetrite põhjal [37]. Viimaste puuduseks on asjaolu, et mootori parameetrid ei ole realsuses konstantsed ning parameetrite reaal-ajas muutuste arvesse võtmine vajab oluliselt arvutusressurssi. Alternatiivseks lahenduseks on leida voolu seadeväärtused katseliselt. Selleks kaardistatakse kogu mootori tööpiirkond, mille käigus selgitatakse välja millise d- ja q-telje voolu seadeväärtuste kombinatsiooniga erinevate kiiruste juures on mootori väändmoment suurim. Viimane on aega nõudev protsess, kuid selline lähenemine võtab



arvesse mootori parameetrite muutumist erinevate tingimuste juures [36]. Arendatava kontrolleri puhul otsustati kasutada katselist lähenemist.

Seadeväärtuste teisendamine tehakse seega varasemalt mõõdetud tulemuste põhjal. Antud tulemused talletatakse programmikoodis tabeli kujul, kus rea indeks iseloomustab pöörlemiskiirust ning veeru indeks väändemomenti. Pöörlemiskiiruse väärtused on tabelisse talletatud iga  $100 \text{ min}^{-1}$  tagant ning väändemoment  $0,1 \text{ Nm}$  tagant. Rea indeksile üks vastab kiirus  $100 \text{ min}^{-1}$ , kaks  $200 \text{ min}^{-1}$  jne. Veeru indeks momendi puhul jälgib sarnast seaduspära, kuid alustatakse null veerust, kuhu on talletatud väikseim negatiivne momendi seadeväärtus. Seega kiiruse vahemiku  $0 - 20\,000 \text{ min}^{-1}$  ja momendi vahemiku  $-25 - 25 \text{ Nm}$  talletamiseks on vajalik  $200 \times 500$  tabel. Kuna seadeväärtused tuleb leida nii q- kui d-telje voolule, siis vaja on kahte tabelit. Ühes q-telje voolu seadeväärtused ja teises d-telje.

Veeru indeks arvutatakse sisendiks saadud väändemomendi seadeväärtuse põhjal kasutades valemit (5.2). Sealjuures lähtutakse asjaolust, et momendile  $-25 \text{ Nm}$  vastav voolu seadeväärtus asub tabelis veerus null ning momendile  $25 \text{ Nm}$  vastav seadeväärtus asub veerus 500. Ühtlasi ümardatakse momendi seadeväärtus eelnevalt kümnendiku täpsusele.

$$x = T_{ref} \cdot 10 + 250. \quad (5.2)$$

kus  $x$  – veeru indeks,

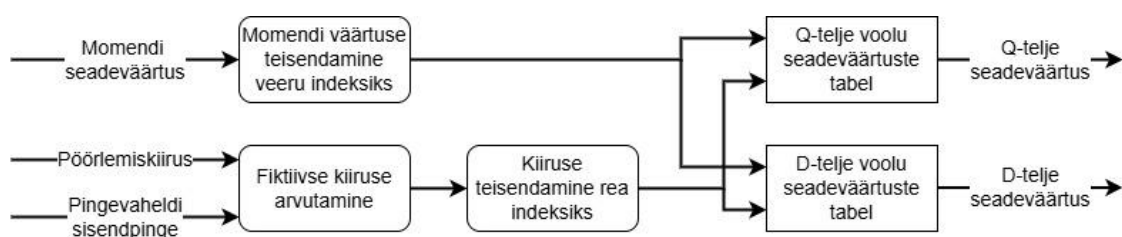
$T_{ref}$  – momendi seadeväärtus, Nm.

Rea indeks leitakse hetke pöörlemiskiiruse ning pingevaheldi sisendpinge põhjal. Tulenevalt sellest, et pöörlemiskiirus milleni mootor suudab tekitada maksimaalset momenti sõltub mootorile rakendavast pingest, tuleb arvesse võtta muutuvat pingevaheldi sisendpinget. Teisisõnu, mida madalam on mootorile rakendatav pinge, seda madalamalt kiiruselt tuleb mootori magnetvälja nõrgestada, mille tulemusel väheneb ka mootori väljundmoment [36]. Kuna katseliselt leitud voolu seadeväärtused on saadud konstantse pingevaheldi sisendpinge korral, siis üheks võimaluseks on kasutada fiktiivset pöörlemiskiirust. Selleks tuleb arvutada esiteks hetkeväärtuste põhjal mootori magnetvoog kasutades valemit (5.3): [36]

$$\lambda = \frac{U_{dc}}{\sqrt{3} \cdot \omega_e} \quad (5.3)$$

kus  $\lambda$  – mootori magnetvoog, Wb,  
 $U_{dc}$  – pingevaheldi sisendpinge, V,  
 $\omega_e$  – mootori elektriline pöörlemiskiirus, rad/s.

Saadud magnetvoo alusel arvutatakse pöörlemiskiirus pinge järgi, mida kasutati seadeväärtuste leidmise katses. Selleks saab kasutada jällegi valemit (5.3), kuid avaldades viimasest välja kiiruse. Saadud tulemus on fiktiivne kiirus, mida saab kasutada rea indeksi leidmiseks. Selleks tuleb kiirus teisendada  $\text{min}^{-1}$  kujule ning ümardada lähima sajaliseni. Kirjeldatud seadeväärtuste teisendamine on ühtlasi näidatud Joonis 35.



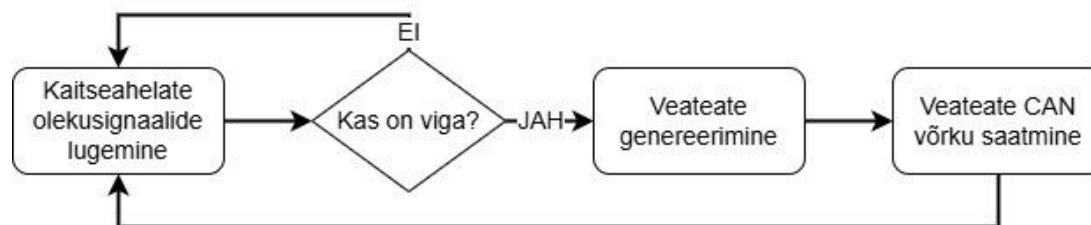
Joonis 35. Momendi seadeväärtuse teisendamine voolu seadeväärtusteks.

## 5.4 Vealukordade käsitlemine

Vealukordade käsitlemisel kasutatakse sama lähemist, mis juhtimisalgoritmi tarkvara puhul. Lisas 11 toodud võimalike veakoodide tabelis on seega toodud ka toetavate funktsioonide mikrokontrolleri poolt genereeritavad veakoodid. Erinevus kahe tarkvara vahel on see, et toetavate funktsioonide mikrokontroller lisaks vigade tuvastamisele ja veakoodide genereerimisele, võtab vastu veateated ka teiselt mikrokontrollerilt ning edastab kõik veateated CAN võrku. Selleks, et kõik veateated saaksid CAN võrku edastatud on programmikoodis veateadete puhver, kus on veateated järjestatud tuvastamise järjekorras. Ühtlasi saadetakse koos veakoodiga CAN võrku veateate järjekorra number puhvris ning kogu vigade arv. Seega CAN võrgust on võimalik välja lugeda, mis järjekorras vead tuvastati, mis omakorda aitab lihtsustada vea põhjuse tuvastamist.

## 5.5 Kaitseahelate olekusignaaliid

Pingevaheldi ülekoormuse kaitseahelate olekuid jälgitakse nende olekusignaali põhjal. Viimaste veaolukordade tuvastamise protsess on toodud Joonis 36. Ülekoormuse kaitseahelate olekusignaale on kokku neli: kolm faasivoolude ning üks sisendvoolu kaitseahela signaal. Lisaks on kaks olekusignaali, mis kirjeldavad transistoride draiverite seisundit. Üks nendest iseloomustab, kas transistoride draiverid on valmis PWM signaale vastu võtma ning teine annab märku draiverite veaseisundist. Esimese signaali põhjal määratakse Tabel 8 toodud juhtsõnumi 1. biti väärtus. Kui mõni kaitseahela olekusignaali või transistori draiveri veaseisundi signaal annab märku veaolukorrast, siis genereeritakse vastav veateade ning pingevaheldi töö peatatakse. Ühtlasi saadetakse veateade vormeli CAN võrku.



Joonis 36. Kaitseahelate olekusignaali kontrollimise protsess.

## 5.6 CAN liides

Vormeli CAN võrk kasutab CAN 2.0A protokollit ehk sõnumitel on 11 bitised ID-d. Sõnumite prioriteet on määratud ID-ga ehk mida madalam on ID väärtus, seda kõrgema prioriteediga sõnumiga on tegu [38]. Selleks, et hoida CAN liini koormus kontrolli all, siis on mootori kontrollid ühendatud vormeli üldisest CAN võrgust eraldi CAN võrku. Sealjuures on iga kontrolleri jaoks mõeldud erineva ID-ga sõnumid. ID-d on jaotatud nii, et ID viimane number määrab millisele kontrollile sõnum mõeldud on: 1 – esimene kontroll, 2 – teine kontroll, 3 – kolmas kontroll ning 4 – neljas kontroll. Kontrollid kasutavad CAN ID-sid vahemikus 0x101-0x171 ning iga kontrolli kohta on kokku kaheksa sõnumit, millest seitse on tagasiside ning üks juhtsõnum. Iga sõnumi täpsem sisu on kirjeldatud Tabel 10, kus kontrolleri number on asendatud tähega „N“.

Tabel 10. Kontrolleri CAN sõnumid.

Sõnumi ID	Sõnumi nimetus	Sõnumi kirjeldus
0x10N	INVERTER_N_Control	Momendi seadeväärtus, pingevaheldi käivitamise käsklus, vigade lähtestamine, transistoride draiverite veateadete lugemise käsklus.
0x11N	INVERTER_N_Status	Kontrolleri olek, mida kirjeldab Tabel 8 toodud sõnumi struktuur; lisaks veakoodid, vea number ning vigade arv.
0x12N	INVERTER_N_Motor	Mootori pöörlemiskiirus, elektriline nurk ja väändmoment.
0x13N	INVERTER_N_VoltageCurrent	Möödetud d- ja q-telje voolud, pingevaheldi sisendpinge ja -vool.
0x14N	INVERTER_N_Temp1	Mootori temperatuur ning kolme ülemise öla transistoride temperatuurid.
0x15N	INVERTER_N_Temp2	Kolme alumise öla transistoride temperatuurid.
0x16N	INVERTER_N_Temp3	Transistoride jahutusplaati siseneva ja väljuva vee temperatuur.
0x17N	INVERTER_N_Debug	Kontrolleri katsetamiseks mõeldud sõnum, kuhu saab andmeid vastavalt vajadusele lisada.

## 5.7 Funktsioonide ajastamine

Toetavate funktsioonide tarkvara funktsioonid on jagatud kolme gruppi: kriitilised, keskmised ning aeglased. Kriitiliste funktsioonide esinemis sageduseks on 500 Hz, keskmistel 200 Hz ning aeglastel 10 Hz. Funktsioonide jaotamine kolme kategooria vahel on näidatud Tabel 11. Funktsioonide ajastamiseks kasutatakse mikrokontrolleri *Systick* taimerit, mis tekitab programmi katkestuse iga 1 ms tagant. Katkestuse ajal tõstetakse lipp, kui möödunud on 2, 5 või 100 ms. Lippude olekuid kontrollitakse programmikoodi lõpmatus tsüklis ning lipu kõrge oleku korral täidetakse, kas kriitilised, keskmised või aeglased funktsioonid, sõltuvalt milline lipp kõrge on. Prioriteet sealjuures on suurema

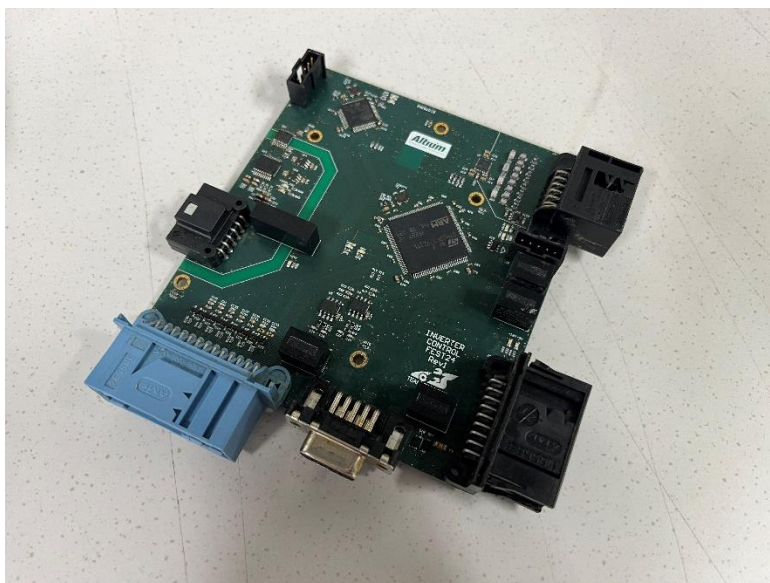
sagedusega funktsioonidel. Lisaks toimub 1 kHz sagedusega andmevahetus teise mikrokontrolleriga. Viimase sageduse määrab juhtimisloogika mikrokontroller ning toetavate funktsioonide mikrokontrolleri programmikoodis toimub katkestus kui andmevahetus algatatakse. Katkestuse käigus loetakse SPI liinilt andmed ning kirjutatakse saadetavad andmed SPI saadetavate andmete puhvrissi.

Tabel 11. Toetavate funktsioonide esinemis sagedused.

<b>Funktsioon</b>	<b>Sagedus</b>
Andmevahetus teise mikrokontrolleriga	1 kHz
Kaitseahelate olekusignaali kontrollimine	500 Hz
Pingevaheldi sisendvoolu mõõtmine	500 Hz
CAN sõnumite 0x10N – 0x13N saatmine	200 Hz
Seadeväärtuste teisendamine	200 Hz
Temperatuuride mõõtmine	10 Hz
CAN sõnumite 0x14N – 0x16N saatmine	10 Hz

## 6 Prototüübi katsetamine

Mootori kontrolleri funktsionaalsuste testimiseks ning arendamiseks katsetati seda tudengivormeli veoajami katsepingis koos pingevaheldi ning mootoriga. Selleks koostati esmalt kontrolleri trükkplaat, mille tulemus on näidatud Joonis 37. Koostamise käigus testiti ja verifitseeriti ahelad eraldiseisvalt. Katsepingi testide eesmärgiks oli integratsioon pingevaheldi ja mootoriga ning funktsionaalsuste testimine ja arendamine, mis on kirjeldatud järgnevatel peatükkidel.

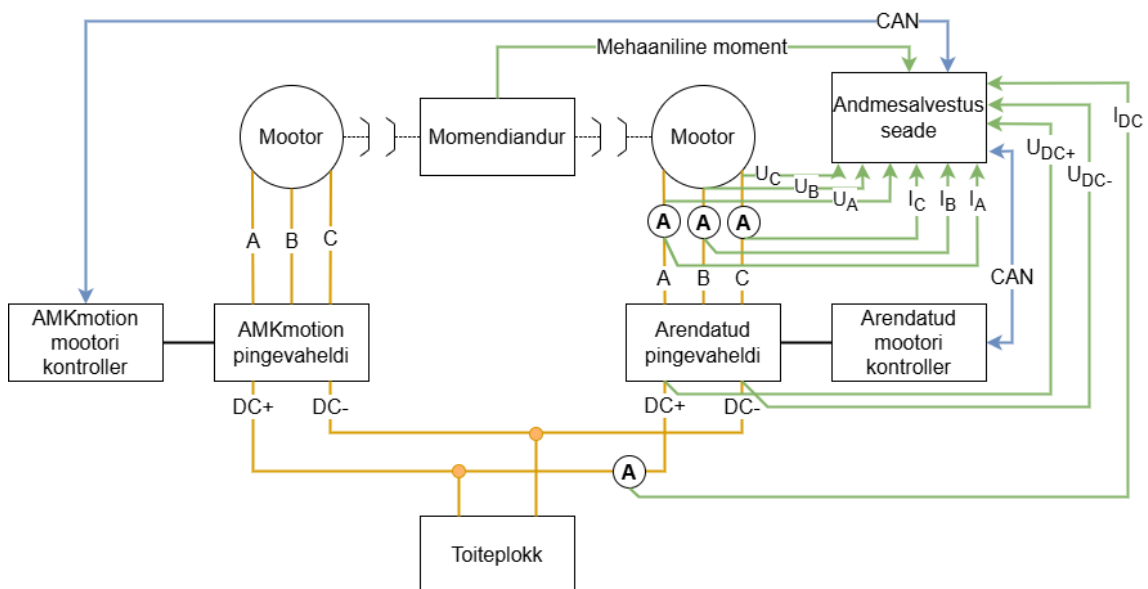


Joonis 37. Koostatud mootori kontrolleri trükkplaat.

### 6.1 Veojami katsepink

Kontrolleri testimiseks kasutati tudengivormeli veoajami katsestendi TalTech'i elektrisõidukite laboris [39]. Katsepingi põhimõtteskeem on näidatud Joonis 38. Katsepinki on ühendatud kaks tudengivormeli mootorit, millest ühte juhib AMKmotion kontrolleri ning teist käesolevas töös arendatud kontrolleri. Sealjuures juhitakse AMKmotion kontrolleri ühendatud mootorit kiiruse põhiselt ning arendatud kontrolleri ühendatud mootorit momendi põhiselt. Sellisel määral määratakse ühe mootoriga pöörlemiskiirus ning teisega avaldatakse, kas siis kiirendavat või aeglustavat momenti. Kiirendava momendi ehk arendatud kontrolleri jaoks positiivse momendi korral hakkab kiirust määrav mootor

regeneratiivset energiat tootma, et pöörlemiskiirust hoida. Aeglustava ehk negatiivse momendi korral on olukord vastupidine. Kuna mõlemad pingevaheldid on ühendatud sama alalispinge liinile, siis toiteplokkist kasutatakse ainult energiat, mis läheb katsepingis kadudeks. Selleks, et välja selgitada kui suured on veoajami komponentide kaod on süsteemi ühendatud erinevad mõõteseadmed, mille informatsioon on toodud Tabel 12. Mehaanilise momendi ning võimsuse mõõtmiseks on mootorite völlide vahele ühendatud momendiandur. Mootori sisendvõimsuse ehk pingevaheldi väljundvõimsuse arvutamiseks mõõdetakse mootori faasidelt pinget ja voolu. Pingevaheldi sisendvõimsuse arvutamiseks mõõdetakse samuti selle sisendvoolu ja -pinget. Mõlemale kontrolleriile saadetakse seadeväärtused üle CAN võrgu, mis on samuti ühendatud andmesalvestus seadmega.



Joonis 38. Veoajami katsestendi põhimõtteskeem.

Tabel 12. Katsestendis kasutatud mõõtevahendid [40], [41], [42].

Seade	Tootja	Mudel	Mõõtevahemik	Täpsus
Andmesalvestus	Dewesoft	SIRIUS HS	$\pm 800$ V $\pm 5$ V	$\pm 0,1$ %
Voolusensor	Signaltec	CT 100	$\pm 100$ A	0,2 %
Momendiandur	Lorenz messtechnik	DR-2643	$\pm 50$ Nm	0,1 %

## 6.2 Kiiruse arvutuse valideerimine

Tulenevalt sellest, et arvutatud kiirust kasutatakse nii nurga interpoleerimisel kui ka seadeväärtuste teisendamisel, siis veenduti arvutuse õigsuses. Selleks võrreldi CAN võrku saadetud arvutatud kiirust mootori faasivoolu sagedusega. Nimelt mootori faasivoolu sagedus on võrdne mootori elektrilise pöörlemissagedusega. Tulemusi võrreldi iga 2000 min<sup>-1</sup> tagant vahemikus 0 – 20 000 min<sup>-1</sup>. Tulemused on toodud Tabel 13, kust on näha, et viga on varieeruv, kuid marginaalne. Saadud tulemuste põhjal on kiiruse arvutamise täpsuseks ± 0,4 %. Antud täpsust peeti esimese prototüübi jaoks sobivaks. Katse käigus oli näha, et kiiruse arvutuse tulemus varieerub stabiilses olukorras keskmiselt ± 50 min<sup>-1</sup>, seega Tabel 13 toodud tulemustes kasutati keskmestatud väärtust. Varieeruvuse vähendamiseks on üheks võimaluseks kiiruse arvutusel kasutada libisevat keskmist.

Tabel 13. Kiiruse arvutuse valideerimise tulemused.

Arvutatud kiirus [min <sup>-1</sup> ]	Faasivoolu sagedus [Hz]	Kiirus faasivoolust [min <sup>-1</sup> ]	Viga [min <sup>-1</sup> ]	Täpsus [%]
2001,4	133,3	2000,3	1,1	0,05
4001,1	266,5	3999,9	1,2	0,03
6005,6	400,1	6001,5	4,1	0,07
7971,8	533,1	7996,7	-24,9	-0,31
9972,3	666,6	9999,0	-26,7	-0,27
11954,8	800,2	12003,0	-48,2	-0,40
13995,4	933,2	13998,0	-2,6	-0,02
15983,6	1066,4	15996,0	-12,4	-0,08
17993,7	1204,8	18072,0	-78,3	-0,43
19991,5	1334,6	20019,0	-27,5	-0,14

## 6.3 Faasivoolude mõõtmise valideerimine

Valideerimaks, kas mõõdetud faasivoolud vastavad reaalsusele, siis võrreldi kontrolleri poolt mõõdetud tulemusi katsepingi voolusensoritega. Kuna faasivoolude sagedus ulatub kuni 1333,3 Hz, siis üle CAN võrgu on tulemusi keeruline võrrelda. Seega valideerimiseks kasutati mõõdetud voolude d- ja q-telje komponente, mis arvutatakse



juhtimisalgoritmis faasivooludest kasutades Park'i ja Clarke'i teisendusi. Faasivoolu amplituudi ja voolude d- ning q-telje komponentide vahel kehtib seos: [36]

$$I_{amplituud} = \sqrt{I_q^2 + I_d^2}. \quad (6.1)$$

kus  $I_{amplituud}$  – faasivoolu amplituud, A,  
 $I_q$  – faasivoolu q-telje komponent, A,  
 $I_d$  – faasivoolu d-telje komponent, A.

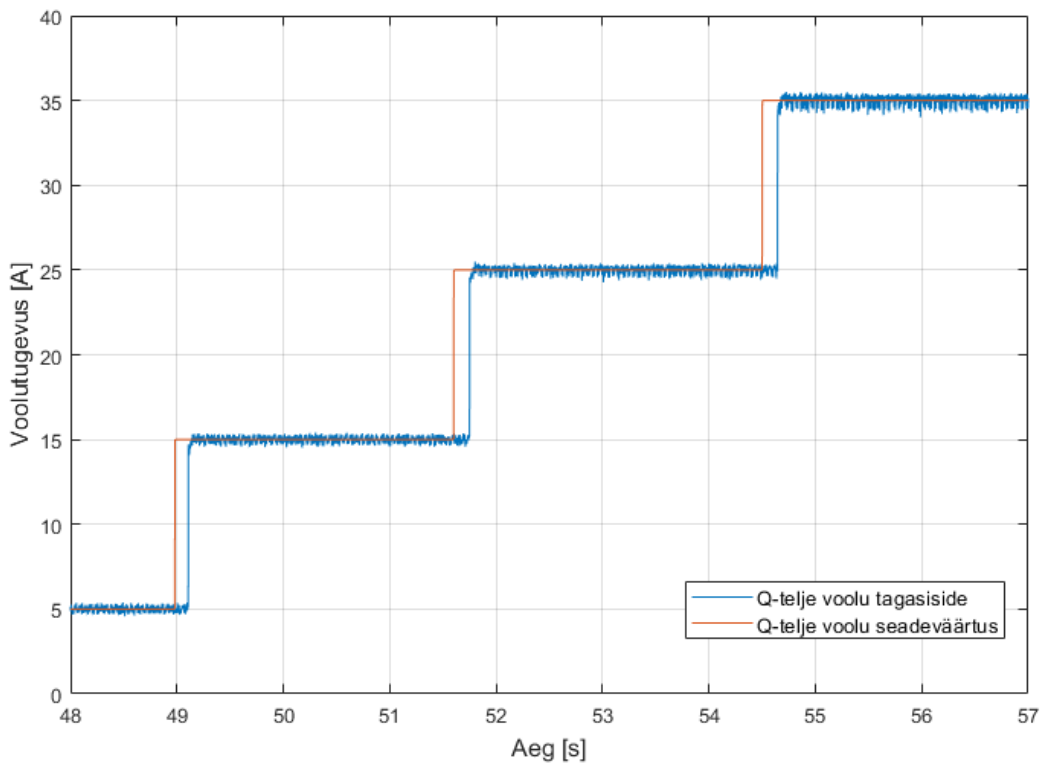
D- ja q-telje voolu komponentide eeliseks on see, et stabiilses olukorras on nende väärtused samuti stabiilsed. Ühtlasi annab antud lähenemine võimaluse kontrollida Park'i ja Clarke'i teisenduste õigsust. Katse viidi läbi pöörlemiskiirusel  $4000 \text{ min}^{-1}$  ning kasutati nii q- kui ka d-telje voolu. Tulemused on esitatud Tabel 14, kust selgub, et voolu amplituudi suurenedes viga suureneb. 90 A amplituudiga faasivoolu korral on viga suurim ehk 1,5 %. Täpsuse parandamise üheks võimaluseks on kasutada kalibreerimise funktsiooni, kus mõõtetulemus on funktsiooni argument. Funktsiooni välja selgitamiseks on otstarbekas vaadelda igat mõõtetulemust eraldi ja kalibreerida kogu mõõtevahemik.

Tabel 14. Faasivoolude mõõtmise valideerimise tulemused.

$I_q$ [A]	$I_d$ [A]	Arvutatud amplituud [A]	Mõõdetud amplituud [A]	Viga [A]	Täpsus [%]
9,90	-1,34	9,99	10,04	-0,05	-0,50
19,59	-3,98	19,99	20,14	-0,15	-0,74
29,17	-7,00	30,00	30,29	-0,29	-0,96
39,21	-8,00	40,02	40,43	-0,41	-1,02
49,28	-8,88	50,07	50,59	-0,52	-1,02
58,57	-12,16	59,82	60,75	-0,93	-1,53
68,68	-14,07	70,11	70,91	-0,80	-1,13
78,47	-15,63	80,01	81,07	-1,06	-1,31
88,20	-18,01	90,02	91,37	-1,35	-1,48

## 6.4 Vooluregulaatori seadistamine

Vooluregulaatori kahe PI-kontrolleri seadistamine viidi läbi katseliselt. Selleks seadistati integraatorite võimendustegurid esialgu marginaalseteks ning katsetati erinevaid proportsionaalseid võimendusi. Selliselt pandi proportsionaalsete teguritega esialgu paika vooluregulaatori reageerimiskiirus ning lubatud üle reguleerimine. Järgmiseks suurendati integraatorite võimendustegureid kuniks püsiviga jäi alla 0,1 A mõlema telje puhul. Viimaseks toodi sisse telgede lahtisidestus osa, mille tulemusel paraneb vooluregulaatori dünaamilisus ning tagab q- ja d-telgede eraldiseisva reguleerimise [43]. PI-regulaatorite toimimist katsetati kogu mootori tööpiirkonnas, mille käigus pandi paika lõplikud võimendused. Lõplik tulemus q-telje voolu näitel on toodud Joonis 39, kust on näha, et PI-kontroller suudab seadeväärtuse saavutada ning seda stabiilselt hoida. Graafikul on seadeväärtuse ja tagasiside muutuse vahel 100 ms hilistus, mis on tingitud andmesalvestist.

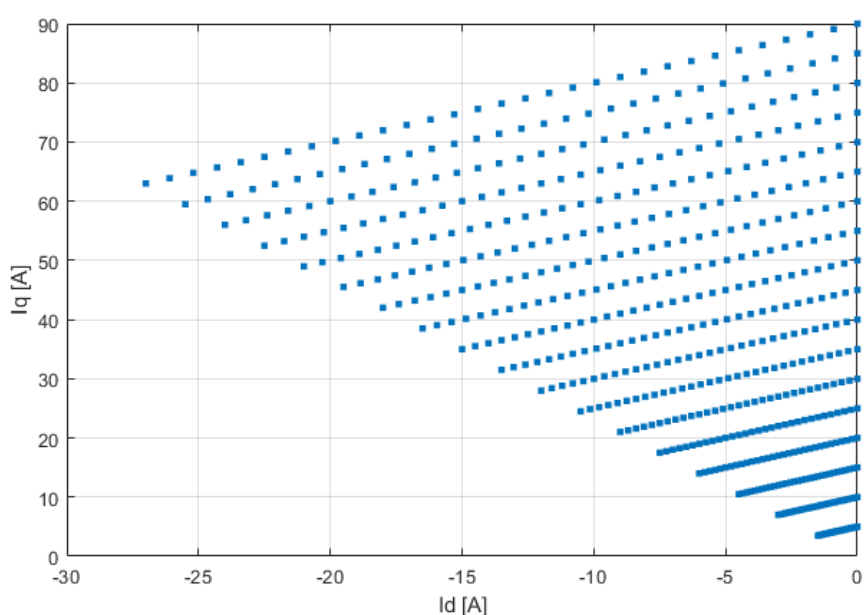


Joonis 39. Q-telje voolu juhtimine PI-kontrolleriga.

## 6.5 Voolu seadeväärtuste katseline leidmine

Väändemomendi seadeväärtuste teisendustabelite loomiseks uuriti, milliste d- ja q-telje voolude kombinatsioonide korral on võimalik kõige suuremat väändemomenti saavutada.

Valemi (6.1) kohaselt määravad d- ja q-telje voolud faasivoolu amplituudväärtuse. Viimasest tulenevalt otsustati kombinatsioonide otsida fikseeritud amplituudväärtuste korral. Teisisõnu d- ja q-telje voolude kombinatsioone katsetati erinevate faasivoolude amplituudväärtuste korral vahemikus 5 – 90 A iga 5 A tagant nagu on näidatud Joonis 40. Antud lähenemine tagab ka selle, et seadeväärtused mille puhul on mootori väljundmoment suurim on ka veoajami efektiivsus suurim. Seda seetõttu, et fikseeritud faasivoolu amplituudväärtuse korral on mootori sisendvõimsus erinevate d- ja q-telje voolukombinatsioonide juures sama. Joonis 40 näidatud kombinatsioonid katsetati läbi kogu mootori pöörlemiskiiruse vahemikus 0 – 20 000 min<sup>-1</sup> iga 500 min<sup>-1</sup> tagant.

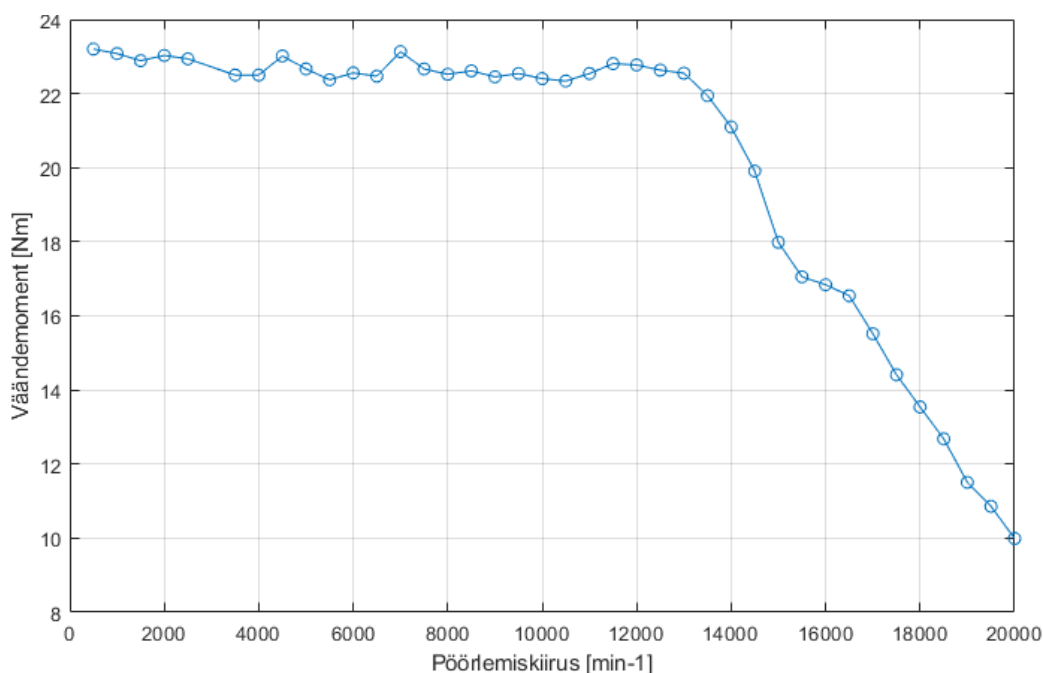


Joonis 40. D- ja q-telje seadeväärtuste kombinatsioonid positiivse väändemomendi näitel.

Katsetusest selgus, et kuni väljanõrgestamiseni ehk konstantse momendi piirkonnas saavutatakse suurim väändemoment sõltumata pöörlemiskiirusest samade d- ja q-telje voolu seadeväärtuste korral. Väljanõrgestamise piirkonnas suureneb d-telje voolu osakaal kõrgemate faasivoolude amplituudide korral pöörlemiskiiruse kasvades. Viimast saab põhjendada asjaoluga, et kõrgemal pöörlemiskiirusel kasvab mootori vastuelektromotoorjõud ning mootori magnetvälja nõrgestamiseks on vajalik suurem d-telje voolu osakaal. Selle tulemusel saab mootorit kõrgemal pöörlemiskiirusel küll juhtida, kuid väheneb maksimaalne väändemoment [44].

## 6.6 Tulemuste analüüs

Mootori kontrolleri prototüübi katsetamiste käigus veenduti funktsionaalsuste toimimises ning leiti sobivad d- ja q-telje voolu seadeväärtuste kombinatsioonid, mida kasutada momendi seadeväärtuste teisendamisel. Mõõdetud maksimaalse mehaanilise momendi tulemus pingevaheldi 600 V sisendpinge korral on toodud Joonis 41. Ühtlasi integreeriti kontrolleri pingevaheldi ning mootoriga, mille käigus valideeriti disainitud liidesed. Mootori puhul veenduti, et kontrolleri suudab mõõta mootori temperatuuri ning lugeda enkoodri poolt mõõdetud mootori võlli asendit. Pingevaheldi puhul kontrolliti mootori faasivoolude, pingevaheldi sisendpinge, transistoride temperatuuride ja pingevaheldi sisendvoolu mõõtmiste toimimist. Ühtlasi veenduti transistoride draiverite I<sup>2</sup>C liidese toimimises lugedes draiverite registreid ning saates konfigureerimise sõnumeid. Katsetuste käigus tuli esile ka veaolukordi, mis andis võimaluse kontrollida, kas pingevaheldi töö peatatakse ootuspäraselt.



Joonis 41. Mootori väändemomendi sõltuvus pöörlemiskiirusest katseliselt leitud seadeväärtustega.

Lisaks teostati katsetuste käigus veoajami efektiivsuse mõõtmisi ning võrreldi tulemusi seni kasutatud AMKmotion kontrolleri ja arendatud pingevaheldi, mille juhtimiseks kasutati käesolevas töös arendatud mootori kontrolleri on seni kasutatud AMKmotion pingevaheldist 2-3 korda väiksemate kadudega. Ühtlasi on arendatud kontrolleri komplekt AMKmotion omast üle kahe korra kergem, kaaludes koos

jahutusplaadiga 623 g [8]. Antud mass on võrdeline kergeima pakutud alternatiiviga Drivetrain Innovation F-SIC kontrolleriiga.

Siiski tuli esimese prototüübi testimisel esile probleeme, mis tuleb töökindluse suurendamiseks lahendada. Nimelt Joonis 41 toodud maksimaalse momendi graafikult on näha väändemomendi järsk langus väljanõrgestamise piirkonnas. Tuginedes varasematele katsetustele AMKmotion kontrolleriiga ning mootori maksimaalsele lubatud võimsusele on 600 V pingevaheldi sisendpinge korral 20 000 min<sup>-1</sup> pöörlemiskiiruse juures maksimaalne võimalik väändemoment kuni 16,7 Nm. Seega languskõver peaks olema oluliselt laugem. Kontrolleri katsetustel seda saavutada ei suudetud, sest lähenedes mootori maksimaalsele võimsusele läksid transistoride draiverid vearežiimi. Viimane tuli esile just väljanõrgestamise piirkonnas, kus on vajalik suurem d-telje voolu osakaal, mis on suure tõenäosusega probleemi põhjusega seotud. Sarnaseid olukordi tuli suurema sagedusega esile ka negatiivse momendi seadeväärtuste katsetamistel. Käesolevas töös antud probleemile veel täpset põhjust ning lahendust välja ei selgitatud.

## 6.7 Edasised arendused

Järgmisteks olulisteks sammudeks kontrolleri arendamisel on töökindluse suurendamine ning seejärel disaini optimeerimine. Töökindlust on võimalik tõsta viies kõik suhtlus – ja mõõteliinid pingevaheldiga diferentsiaalkujule, mille tulemusel tõuseb häiringukindlus. Ühtlasi on võimalik häiringukindlust suurendada kaotades pingevaheldi ja kontrolleri vahel juhtmeühendused asendades need klemmühendustega selliselt, et kontrolleri trükkplaat ühendub otse pingevaheldi külge. Viimane teeb ühenduste teekonna oluliselt lühemaks, kuid sellisel juhul tuleb olulist rõhku panna kontrolleri tundlike mõõteahelate varjestamisele. Lisaks tuleb leida põhjused ning lahendused käesolevas töös esile tulnud probleemidele. Käesoleva töö raames ei proovitud katsepingis resolveri liidest resolveriga koostatud mootori puudumise tõttu. Seega edasiste arenduste käigus tuleb veel välja selgitada kumba mootori asendi määramise meetodit on otstarbekam kasutada.

Ühtlasi on võimalik suurendada kontrolleri töökindlust juhtimisalgoritmi vaatest. Üheks võimaluseks on lisada kaitseahelatesse funktsionaalsus, mis piirab veoajami võimekust kui lähenetakse piirväärtustele. Viimast saaks rakendada näiteks pöörlemiskiiruse ning temperatuuride puhul. Näiteks kui lähenetakse maksimaalsele lubatud pöörlemiskiirusele, siis piiratakse maksimaalset lubatud momenti selliselt, et lubatud

pöörlemiskiirust ei ületata. Sarnast lähenemist saab kasutada temperatuuride puhul, kus vähendatakse maksimaalset lubatavat väändemomenti kui temperatuurid lähenevad maksimaalselt lubatud väärtustele. Lisaks on võimalik kontrolleri töökindlust hinnata veoajami katsestendis, imiteerides reaalseid sõidu olukordi, kus on dünaamiliselt muutuvad momendi seadeväärtused ning pöörlemiskiirused. Viimast on võimalik teostada andes mõlemale katsepingi kontrollerile seadeväärtuseid varasemalt salvestatud sõiduprofiili põhjal.

Töökindluse saavutamisel on võimalik disaini optimeerida. Nimelt kui on tehtud valik resolveri ja enkoodri vahel, siis on võimalik üks liidestest välja jätta. Ühtlasi on võimalik kasutada kahe mikrokontrolleri asemele ühte mitme tuumalist mikrokontrollerit. Nimetud muudatused võimaldavad muuta trükkplaadi disaini oluliselt kompaktsemaks.

## 7 Kokkuvõte

Magistritöö eesmärgiks oli arendada elektrimootori kontrolleri tudengivormeli veoajamile. Arendustöö käsitles endas nii võimalike lahenduste analüüsi, riist – ja tarkvara disaini kui ka prototüübi katsetamist. Mootori kontrolleri peamiseks eesmärgiks on tagada vormeli elektrimootori väändemomendi põhine juhtimine. Sealjuures saavutatakse soovitud väändemoment selliselt, et veoajami kaovõimsus oleks võimalikult väike. Viimane tagab ka selle, et veoajami võimekust kasutatakse maksimaalselt ära.

Töö esimeses osas kirjeldatakse tudengivormeli veoajami ülesehitust ning analüüsitakse eelnevaid lahendusi. Ühtlasi pakutakse välja võimalike mootori kontrolleri alternatiive, mida võrreldi omavahel kuue kriteeriumi alusel. Sobivaimaks lahenduseks osutus arendatud kontrolleri. Järgnevalt pannakse paika nõuded arendatavale mootori kontrolleri ning kontseptsioon.

Teises osas luuakse vastavalt esitatud nõuetele kontrolleri riistvara disain. Selleks koostatakse iga vajaliku liidese jaoks elektriskeem ning valitakse sobivad komponendid. Skeemide disainimisel arvestati asjaoluga, et nii veoajami pingevaheldi kui ka elektrimootor võivad põhjustada signaalides häiringuid. Elektriskeemi disainile järgneb trükkplaadi disain.

Tarkvara arendus on jaotatud kaheks peatükiks. Esimene neist ehk töö kolmas osa kirjeldab mootori kontrolleri juhtimisloogika tarkvara arendust. Sealjuures käsitletakse kasutatavat juhtimisalgoritmi ning milliseid sisendeid on sensoritelt viimase toimimiseks vaja. Järgnevalt kirjeldatakse, kuidas toimub mõõteahelatest andmete lugemine ning veaolukordade käsitlemine. Ühtlasi uuriti, kui pikk on iga funktsiooni arvutamiseks vajalik aeg, mille põhjal teostati funktsioonide ajaline planeerimine juhtimisloogika tarkvaras. Viimaseks antakse ülevaade pingevaheldi käivitamise protseduurist.

Töö neljandas osas kirjeldatakse mootori kontrolleri toetavate funktsioonide tarkvara arendust. Viimane käsitleb nende liideste tarkvara, mida ei kasutata väljavektorjuhtimise algoritmi jaoks, vaid suhtluseks vormeliga ning lisa ohutuse tagamiseks. Ühtlasi kirjeldatakse, kuidas teisendatakse väändemomendi seadeväärtus juhtimisalgoritmi jaoks

sobivale kujule arvestades veoajami muutuvat hetkeseisu. Järgnevalt antakse ülevaade, milliseid andmeid saadab mootori kontrolleri vormeli CAN võrku.

Viimases osas kirjeldatakse mootori kontrolleri põhiliste funktsionaalsuste testimist ja valideerimist. Selleks antakse esmalt ülevaade kasutatud katsestendist. Järgnevalt valideeritakse juhtimisalgoritmi jaoks vajalike mõõteahelaid ning arvutusi. Lisaks kirjeldatakse protsessi, mille alusel kaardistati võimalikult efektiivsed seadeväärtuste teisendused. Viimaseks antakse ülevaade avastatud puudujääkidest ning võimalikest edasistest arendustest.

Käesoleva magistritöö tulemusena valmis tudengivormeli veoajami mootori kontrolleri prototüüp, mida on võimalik kasutada vormeli veoajami juhtimiseks. Kontrolleri tagab soovitud seadistatavuse ning ühildub vormeli elektrisüsteemiga. Ühtlasi on võimalik juhtimisalgoritmi edasi arendada ning muuta kujutegurit vastavalt vajadusele.



## Kasutatud kirjandus

- [1] "Formula Student Team Tallinn," MTÜ Tudengi Vormel, [Võrgumaterjal]. Saadaval: <https://www.formulastudent.ee/et/>. [Kasutatud 07 05 2025].
- [2] "Formula Student Rules 2025," [Võrgumaterjal]. Saadaval: [https://www.formulastudent.de/fileadmin/user\\_upload/all/2025/rules/FS-Rules\\_2025\\_v1.1.pdf](https://www.formulastudent.de/fileadmin/user_upload/all/2025/rules/FS-Rules_2025_v1.1.pdf). [Kasutatud 27 02 2025].
- [3] AMK Motion, "FSE Racing Kit 2022," 2022. [Võrgumaterjal]. Saadaval: [https://amk-motion.com/sites/default/files/FSE/RK22en/01\\_Device%20Description/PDK\\_205481\\_KW26-S5-FSE-4Q\\_en.pdf](https://amk-motion.com/sites/default/files/FSE/RK22en/01_Device%20Description/PDK_205481_KW26-S5-FSE-4Q_en.pdf). [Kasutatud 03 03 2025].
- [4] "DD5-14-10-POW Motor-Datenblatt," 27 11 2018. [Võrgumaterjal]. Saadaval: [https://www.amk-motion.com/amk-dokucd/dokucd/en/content/resources/pdf-dateien/fse/motor\\_data\\_sheet\\_a2370dd\\_dd5.pdf](https://www.amk-motion.com/amk-dokucd/dokucd/en/content/resources/pdf-dateien/fse/motor_data_sheet_a2370dd_dd5.pdf). [Kasutatud 03 03 2025].
- [5] Drivetrain Innovation, "F-SIC," [Võrgumaterjal]. Saadaval: <https://www.drivetraininnovation.com/fs-sic>. [Kasutatud 09 03 2025].
- [6] BUCHER hydraulics, "Double inverter DC/AC + DC/AC," [Võrgumaterjal]. Saadaval: <https://www.bucherdrives.com/96169/Products/MOBILE-DCUDCU/index.aspx>. [Kasutatud 14 03 2025].
- [7] A. Kossiakoff, W. N. Sweet, S. J. Seymour and S. M. Biemer, Systems Engineering Principles and Practice, John Wiley & Sons, Inc., 2011.
- [8] H. Anniste, "Traction Inverter Power Electronics Development for Formula Student Class Car," 2025.
- [9] Fischer Elektromotoren, "TI085-052-070-04B7S-07S04BE2," 2018.
- [10] Analog Devices, "AD7380-4 Data Sheet," 4 2024. [Võrgumaterjal]. Saadaval: [https://www.mouser.de/datasheet/2/609/ad7380\\_4-2821747.pdf](https://www.mouser.de/datasheet/2/609/ad7380_4-2821747.pdf). [Kasutatud 22 03 2025].
- [11] Maxim Integrated, "MAX6070/MAX6071 Data Sheet," 12 2020. [Võrgumaterjal]. Saadaval: [https://www.mouser.ee/datasheet/2/609/MAX6070\\_MAX6071-3129973.pdf](https://www.mouser.ee/datasheet/2/609/MAX6070_MAX6071-3129973.pdf). [Kasutatud 22 03 2025].
- [12] HEIDENHAIN, "ECI 1118 Product Information," 11 2018. [Võrgumaterjal]. Saadaval: [https://www.heidenhain.com/fileadmin/pdf/en/01\\_Products/Produktinformationen/PI\\_ECI1118\\_1119\\_EQI1130\\_1131\\_ID1270494\\_en.pdf](https://www.heidenhain.com/fileadmin/pdf/en/01_Products/Produktinformationen/PI_ECI1118_1119_EQI1130_1131_ID1270494_en.pdf). [Kasutatud 22 03 2025].
- [13] Texas Instruments, "SN65HVD78DR Data Sheet," 03 2019. [Võrgumaterjal]. Saadaval: [https://www.ti.com/lit/ds/symlink/sn65hvd78.pdf?HQS=dis-mouser&ts=1701349624820&ref\\_url=https%253A%252F%252Fwww.mouser.ee%252F](https://www.ti.com/lit/ds/symlink/sn65hvd78.pdf?HQS=dis-mouser&ts=1701349624820&ref_url=https%253A%252F%252Fwww.mouser.ee%252F). [Kasutatud 22 03 2025].

- [14] Texas Instruments, “TIDM-1008 EnDat 2.2 Absolute Encoder Master Interface Reference Design for C2000 MCUs,” [Võrgumaterjal]. Saadaval: <https://www.ti.com/tool/TIDM-1008>. [Kasutatud 22 03 2025].
- [15] Texas Instruments, “TPS25200-Q1,” 03 2015. [Võrgumaterjal]. Saadaval: [https://www.ti.com/lit/ds/symlink/tps25200-q1.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1701357400724&ref\\_url=https%253A%252F%252Fwww.mouser.ee%252F](https://www.ti.com/lit/ds/symlink/tps25200-q1.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1701357400724&ref_url=https%253A%252F%252Fwww.mouser.ee%252F). [Kasutatud 22 03 2025].
- [16] TE Connectivity, “2-1393047-5 DATA SHEET - HOLLOW SHAFT RESOLVER,” 26 06 2022. [Võrgumaterjal]. Saadaval: [https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtv&DocNm=2-1393047-5\\_DATA\\_SHEET&DocType=Data%20Sheet&DocLang=English&DocFormat=pdf&PartCntxt=2-1393047-5](https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtv&DocNm=2-1393047-5_DATA_SHEET&DocType=Data%20Sheet&DocLang=English&DocFormat=pdf&PartCntxt=2-1393047-5). [Kasutatud 23 03 2025].
- [17] Analog Devices, “AD2S1210 Data Sheet,” 03 2021. [Võrgumaterjal]. Saadaval: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad2s1210.pdf>. [Kasutatud 23 03 2025].
- [18] Thermal Micro Control, “PT1000 S series Technical specifications,” 2016. [Võrgumaterjal]. Saadaval: [https://tmc.eu/images/Data\\_sheet\\_S\\_series.pdf](https://tmc.eu/images/Data_sheet_S_series.pdf). [Kasutatud 25 03 2025].
- [19] Analog Devices, “MAX31865 RTD-to-Digital Converter,” 07 2015. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/609/MAX31865-3128729.pdf>. [Kasutatud 25 03 2025].
- [20] Traco Power, “TBA 1-2411HI Unregulated DC/DC Converter,” 19 02 2025. [Võrgumaterjal]. Saadaval: [https://www.mouser.ee/datasheet/2/687/tba1hi\\_datasheet-3049731.pdf](https://www.mouser.ee/datasheet/2/687/tba1hi_datasheet-3049731.pdf). [Kasutatud 26 03 2025].
- [21] Computer Solutions Ltd, “CAN and CAN FD - brief tutorial,” [Võrgumaterjal]. Saadaval: [https://www.computer-solutions.co.uk/info/Embedded\\_tutorials/can\\_tutorial.htm](https://www.computer-solutions.co.uk/info/Embedded_tutorials/can_tutorial.htm). [Kasutatud 28 03 2025].
- [22] Texas Instruments, “ADS8339 Data Sheet,” 10 2014. [Võrgumaterjal]. Saadaval: [https://www.ti.com/lit/ds/symlink/ads8339.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1701379733186&ref\\_url=https%253A%252F%252Fwww.mouser.ee%252F](https://www.ti.com/lit/ds/symlink/ads8339.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1701379733186&ref_url=https%253A%252F%252Fwww.mouser.ee%252F). [Kasutatud 28 03 2025].
- [23] R. Seppago, “Tudengivormeli veojami vektorjuhtimine,” Tallinn, 2023.
- [24] STMicroelectronics, “STM32F777ZIT6 Datasheet - production data,” 09 2017. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f777bi-1851232.pdf>. [Kasutatud 29 03 2025].
- [25] STMicroelectronics, “STM32F745ZGT6 Datasheet - production data,” 02 2016. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f745ie-1851206.pdf>. [Kasutatud 29 03 2025].
- [26] STMicroelectronics, “STM32F746ZGT7 Datasheet - production data,” 02 2016. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f745ie-1851206.pdf>. [Kasutatud 29 03 2025].

- [27] STMicroelectronics, “STM32F745ZET6 Datasheet - production data,” 02 2016. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f745ie-1851206.pdf>. [Kasutatud 29 03 2025].
- [28] STMicroelectronics, “STM32F446RCT7TR Datasheet - production data,” 01 2021. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f446mc-1760997.pdf>. [Kasutatud 29 03 2025].
- [29] STMicroelectronics, “STM32F446RCT7 Datasheet - production data,” 01 2021. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f446mc-1760997.pdf>. [Kasutatud 29 03 2025].
- [30] STMicroelectronics, “STM32F446RET7 Datasheet - production data,” 01 2021. [Võrgumaterjal]. Saadaval: <https://www.mouser.ee/datasheet/2/389/stm32f446mc-1760997.pdf>. [Kasutatud 29 03 2025].
- [31] Recom Power, “R-78-0.5 Data Sheet,” 06 2023. [Võrgumaterjal]. Saadaval: <https://recom-power.com/pdf/Innoline/R-78-0.5.pdf>. [Kasutatud 29 03 2025].
- [32] Mathworks, “PMSM Field-Oriented Control,” [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/sps/ref/pmsmfieldorientedcontrol.html>. [Kasutatud 08 04 2025].
- [33] Mathworks, “Park Transform,” [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/sps/ref/parktransform.html>. [Kasutatud 08 04 2025].
- [34] Mathworks, “Clarke Transform,” [Võrgumaterjal]. Saadaval: <https://uk.mathworks.com/help/mcb/ref/clarktransform.html>. [Kasutatud 08 04 2025].
- [35] HEIDENHAIN, “EnDat 2.2 – Bidirectional Interface for Position Encoders,” 09 2011. [Võrgumaterjal]. Saadaval: [https://www.heidenhain.us/addl-materials/enews/stories\\_1012/EnDat.pdf](https://www.heidenhain.us/addl-materials/enews/stories_1012/EnDat.pdf). [Kasutatud 11 04 2025].
- [36] K. H. Nam, AC Motor Control and Electrical Vehicle Applications, 2019.
- [37] C. Bianchini, G. Bisceglie, A. Torreggiani, M. Davoli, E. Macrelli, A. Bellini and M. Frigieri, “Effects of the Magnetic Model of Interior Permanent Magnet,” MDPI, 2023.
- [38] Kvaser, “The CAN Protocol Tour,” [Võrgumaterjal]. Saadaval: <https://kvaser.com/about-can/the-can-protocol/can-messages-13/>. [Kasutatud 22 04 2025].
- [39] T. Tammeveski, “Elektrijami katsestendi projekteerimine,” 2023.
- [40] Dewesoft, “SIRIUS Technical Reference Manual,” 31 01 2025. [Võrgumaterjal]. Saadaval: <https://downloads.dewesoft.com/manuals/dewesoft-sirius-manual-en.pdf>. [Kasutatud 23 04 2025].
- [41] Signaltec, “High Precision Current Transducers,” 03 2022. [Võrgumaterjal]. Saadaval: [https://signaltec.de/wp-content/uploads/DS\\_CT\\_Current\\_Transducers\\_Ver.1.6.3.pdf](https://signaltec.de/wp-content/uploads/DS_CT_Current_Transducers_Ver.1.6.3.pdf). [Kasutatud 23 04 2025].
- [42] Lorenz messtechnik, “DR-2643 Data Sheet”.

- [43] Renesas, “Vector Control for Permanent Magnet Synchronous Motor with Encoder,” 31 10 2018. [Võrgumaterjal]. Saadaval: [https://www.renesas.com/en/document/apn/vector-control-permanent-magnet-synchronous-motor-encoder-algorithm-rev102?srsltid=AfmBOoqFFeEkNqw845pQW-RDdPyy7HhC\\_an-qktkqABt9as5gyKFI-tO](https://www.renesas.com/en/document/apn/vector-control-permanent-magnet-synchronous-motor-encoder-algorithm-rev102?srsltid=AfmBOoqFFeEkNqw845pQW-RDdPyy7HhC_an-qktkqABt9as5gyKFI-tO). [Kasutatud 23 04 2025].
- [44] R. Maulana, Y. F. Sidik and H. R. Ali, “Field Weakening Control of a Permanent Magnet,” ICITEE, 2023.

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Raigo Seppago

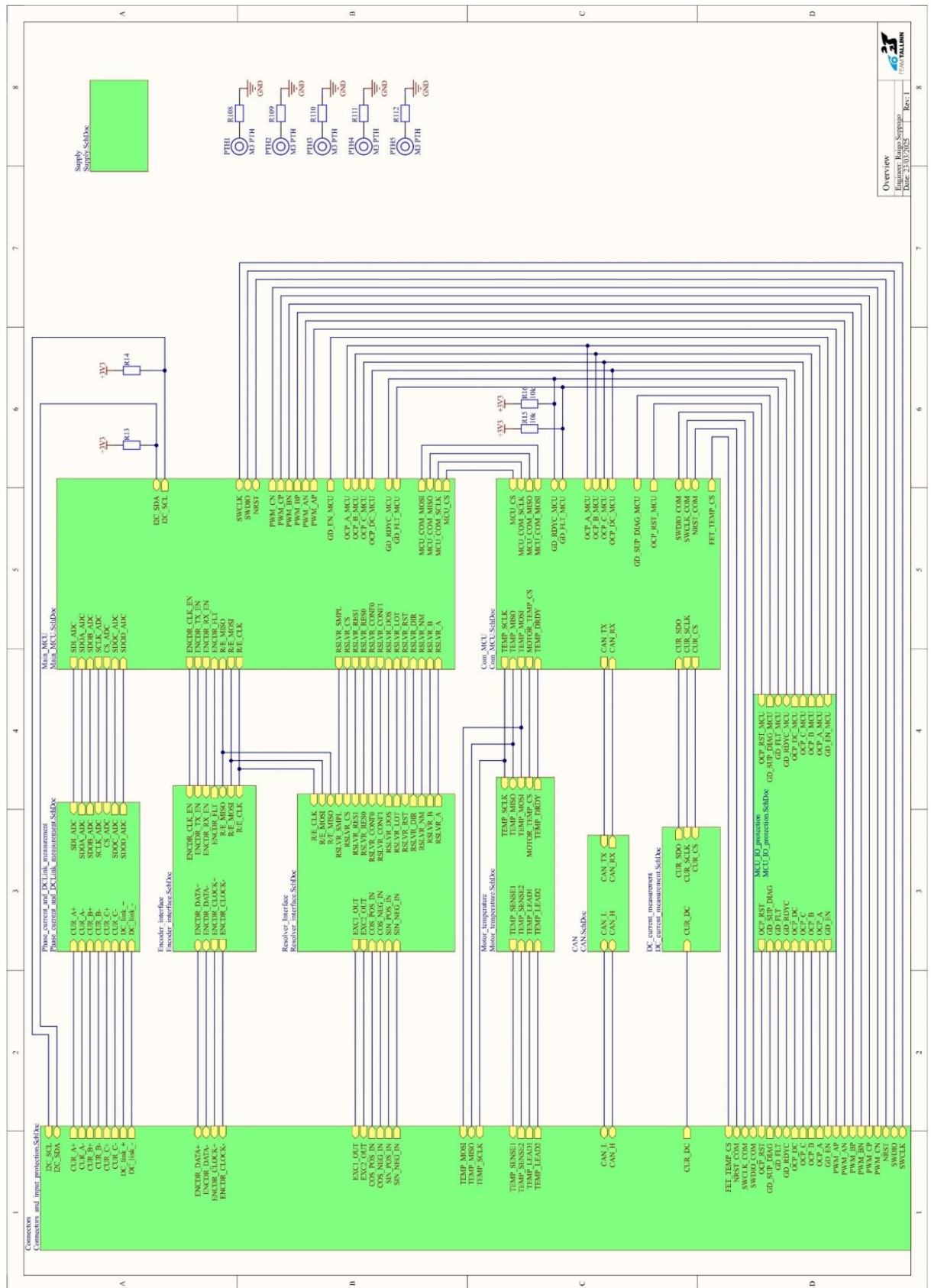
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Tudengivormeli veoajami elektrimootori kontrolleri“, mille juhendaja on Peeter Ellervee.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

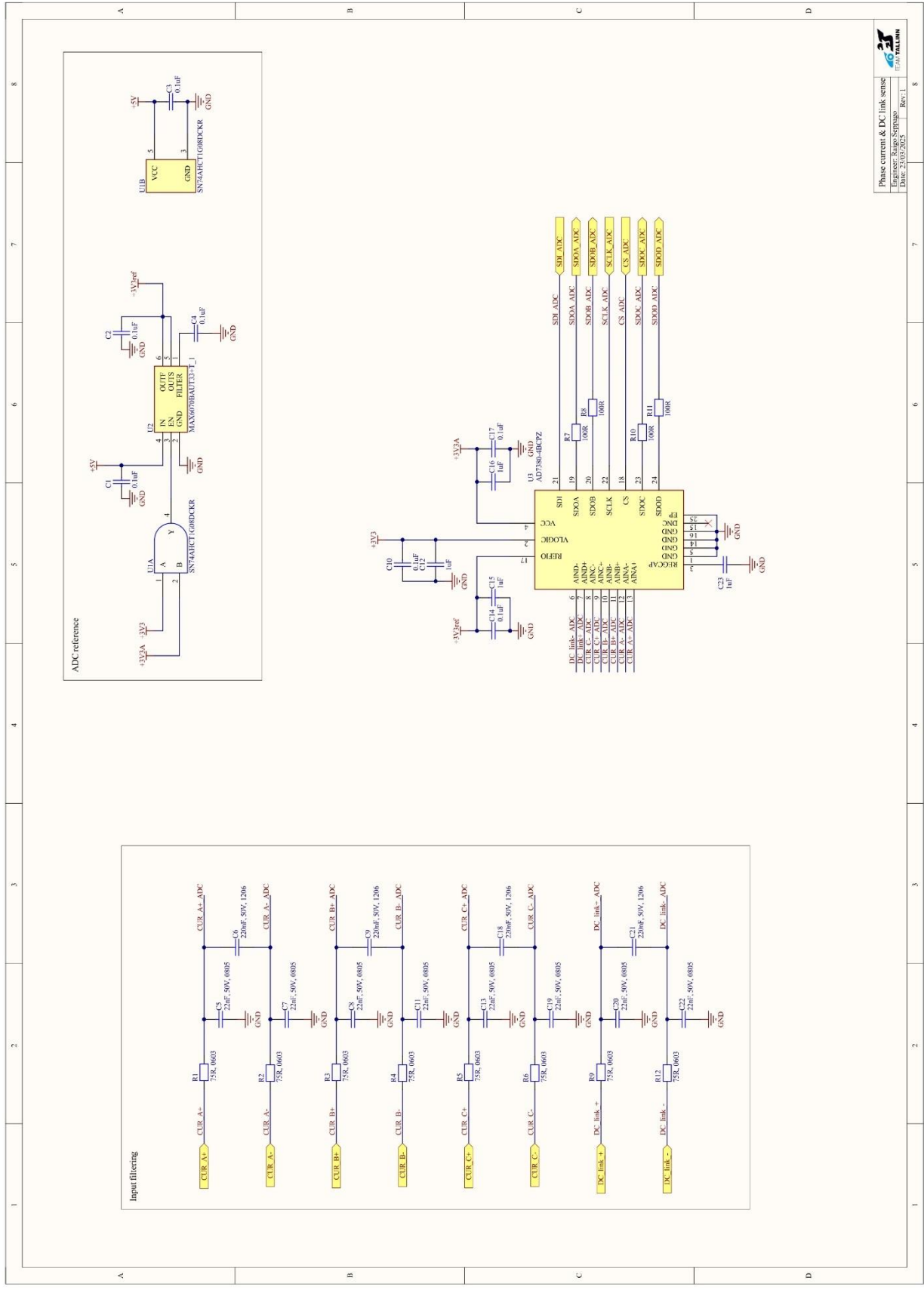
12.05.2025

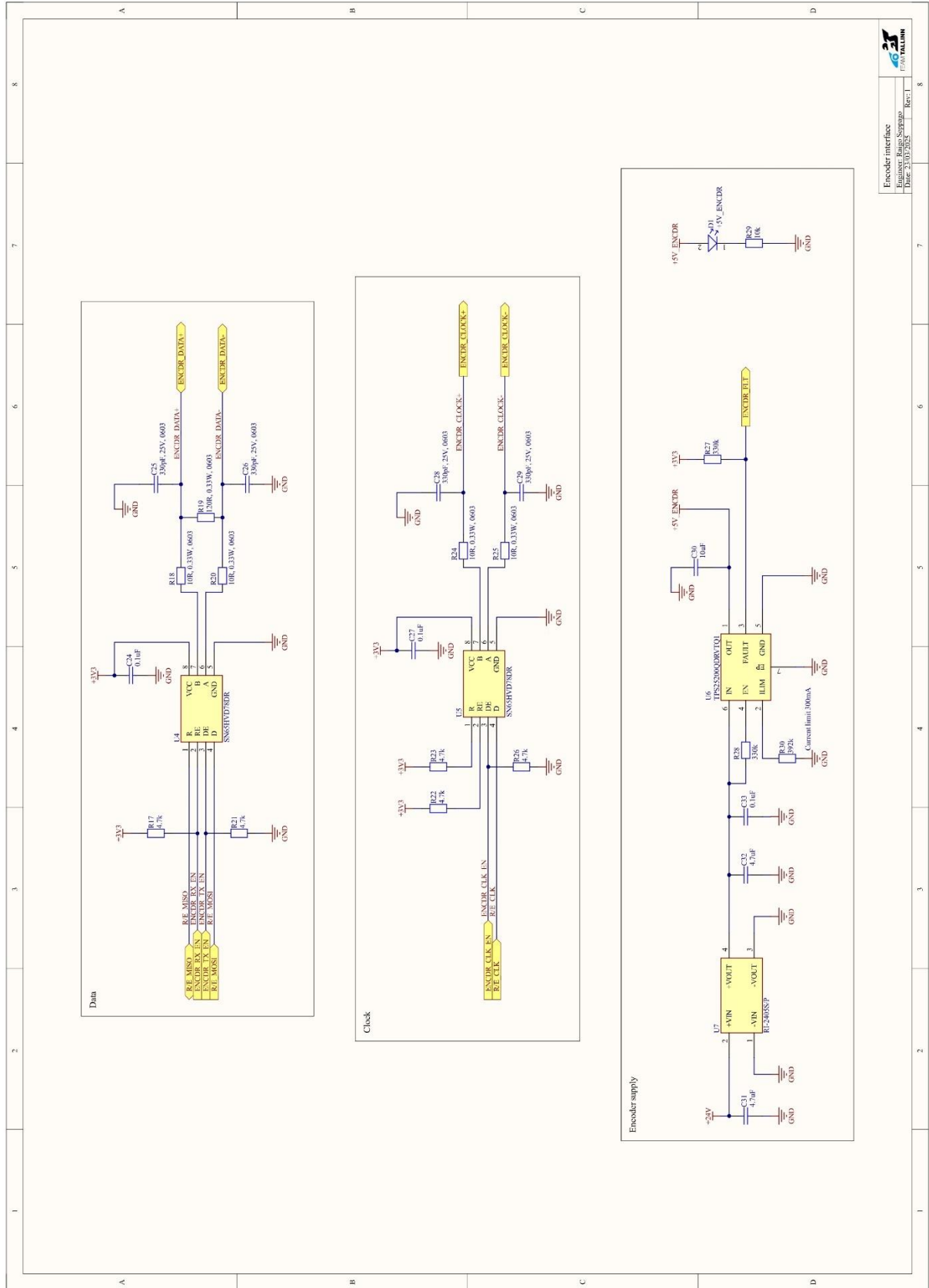
---

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

# Lisa 2 – Mootori kontrolleri trükkplaadi elektriskeemid

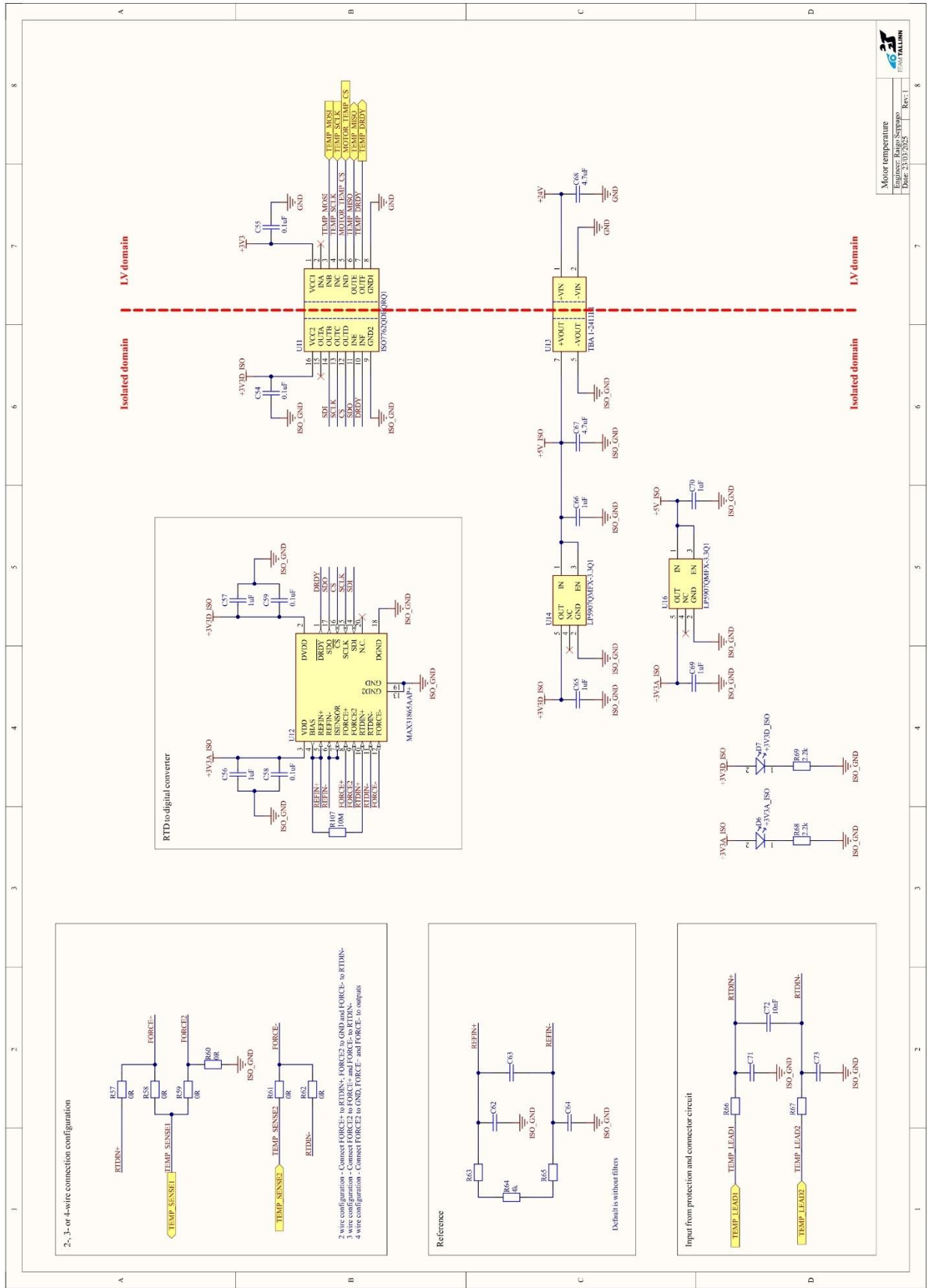


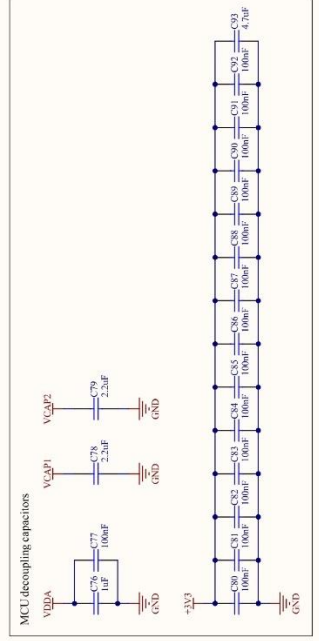
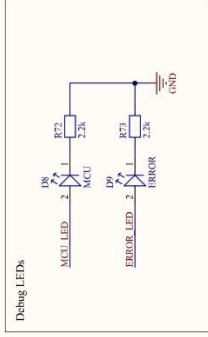
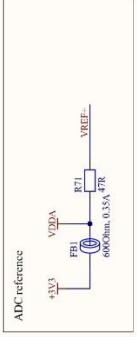
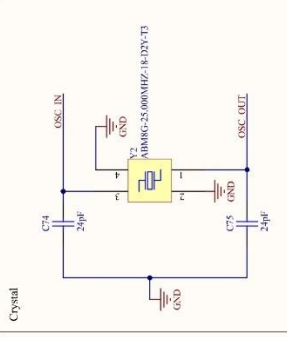
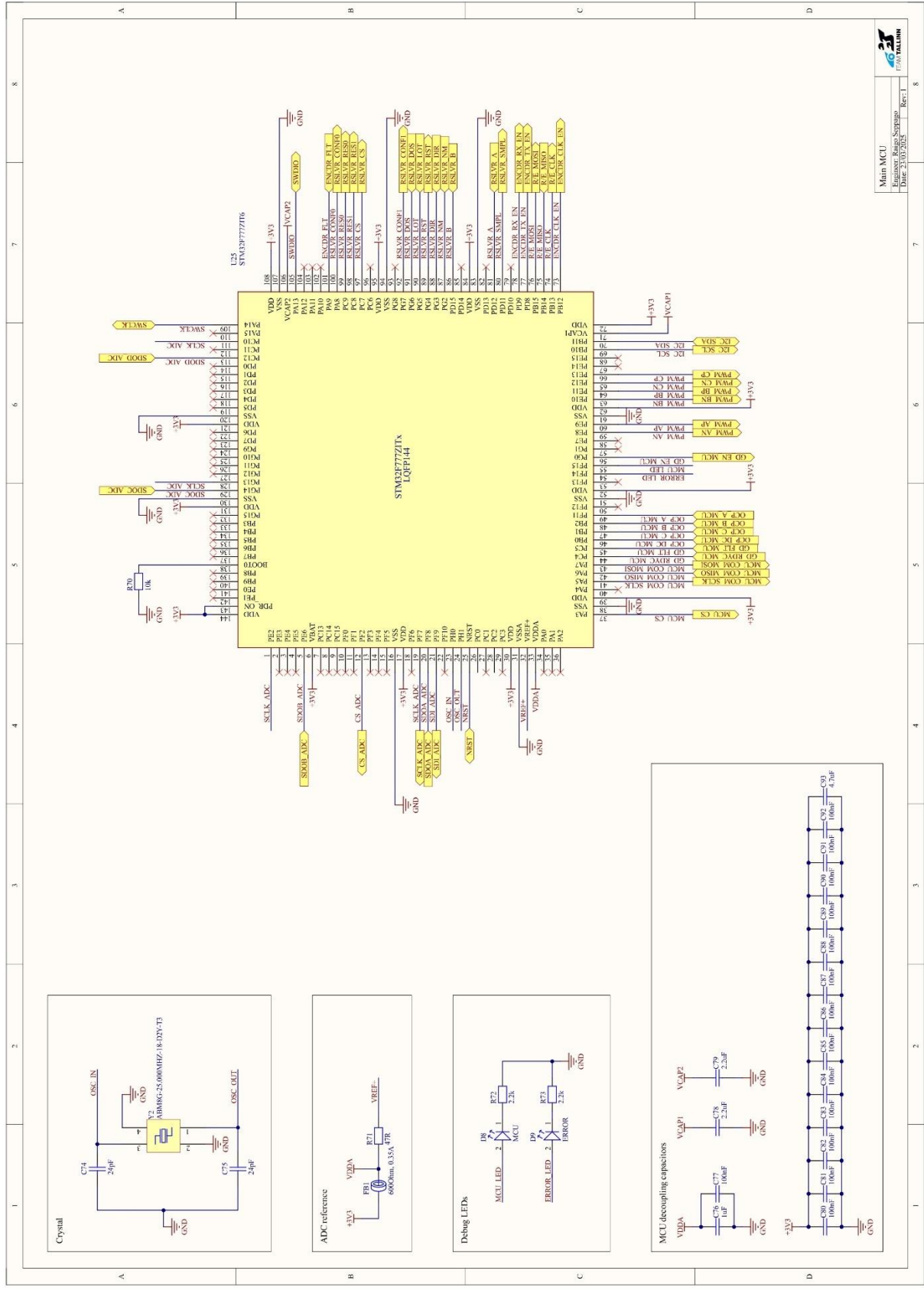


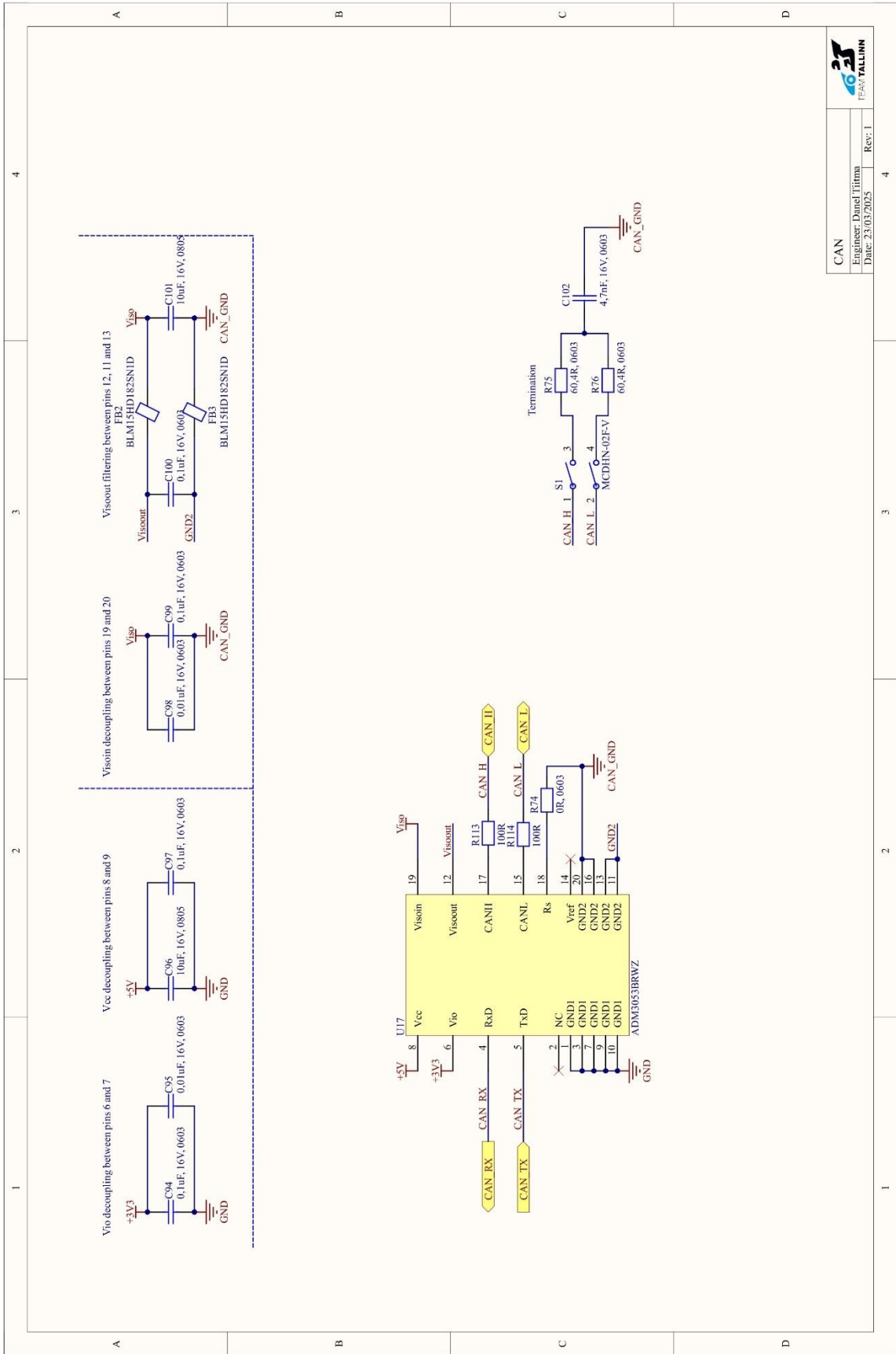


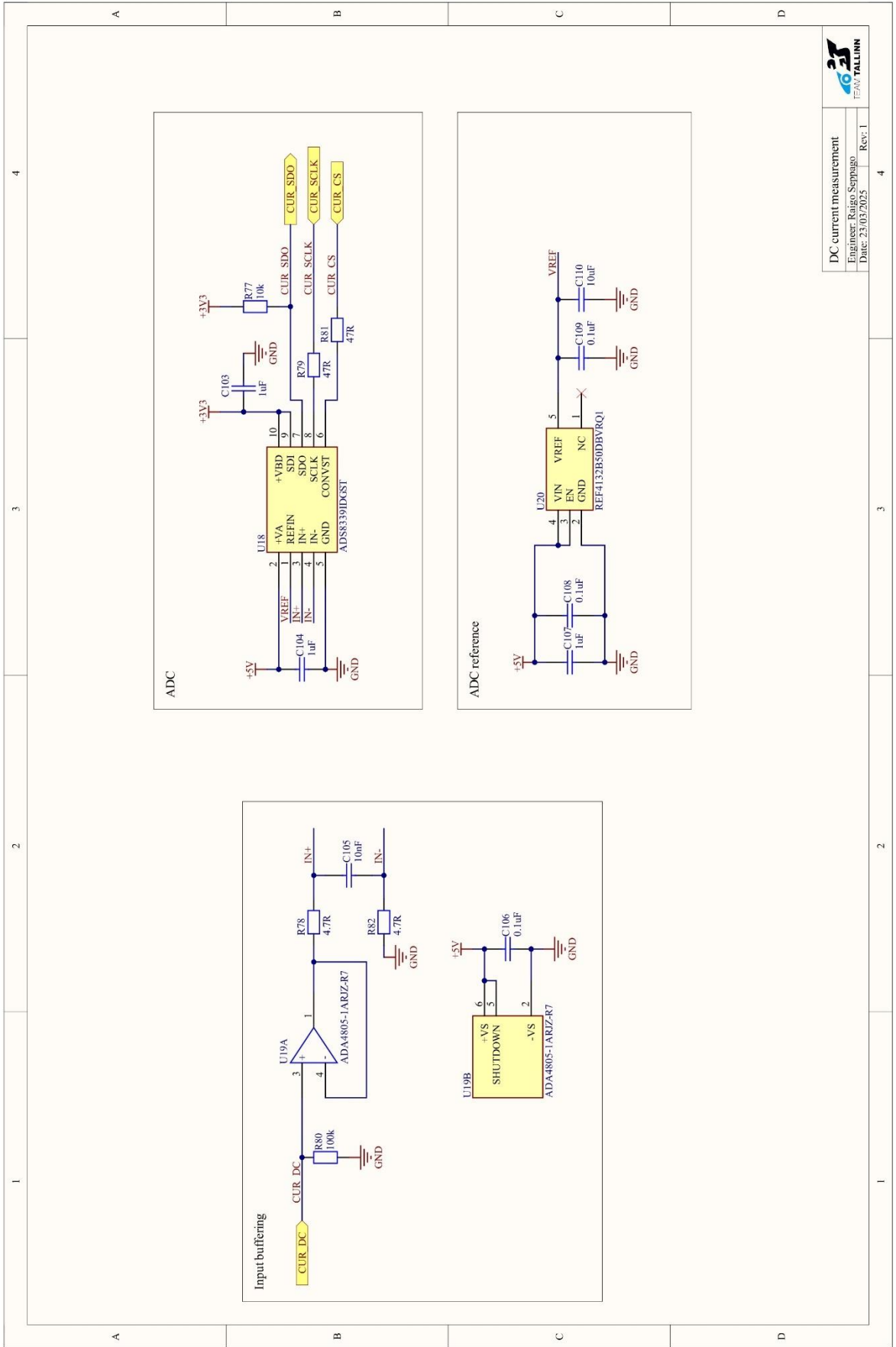




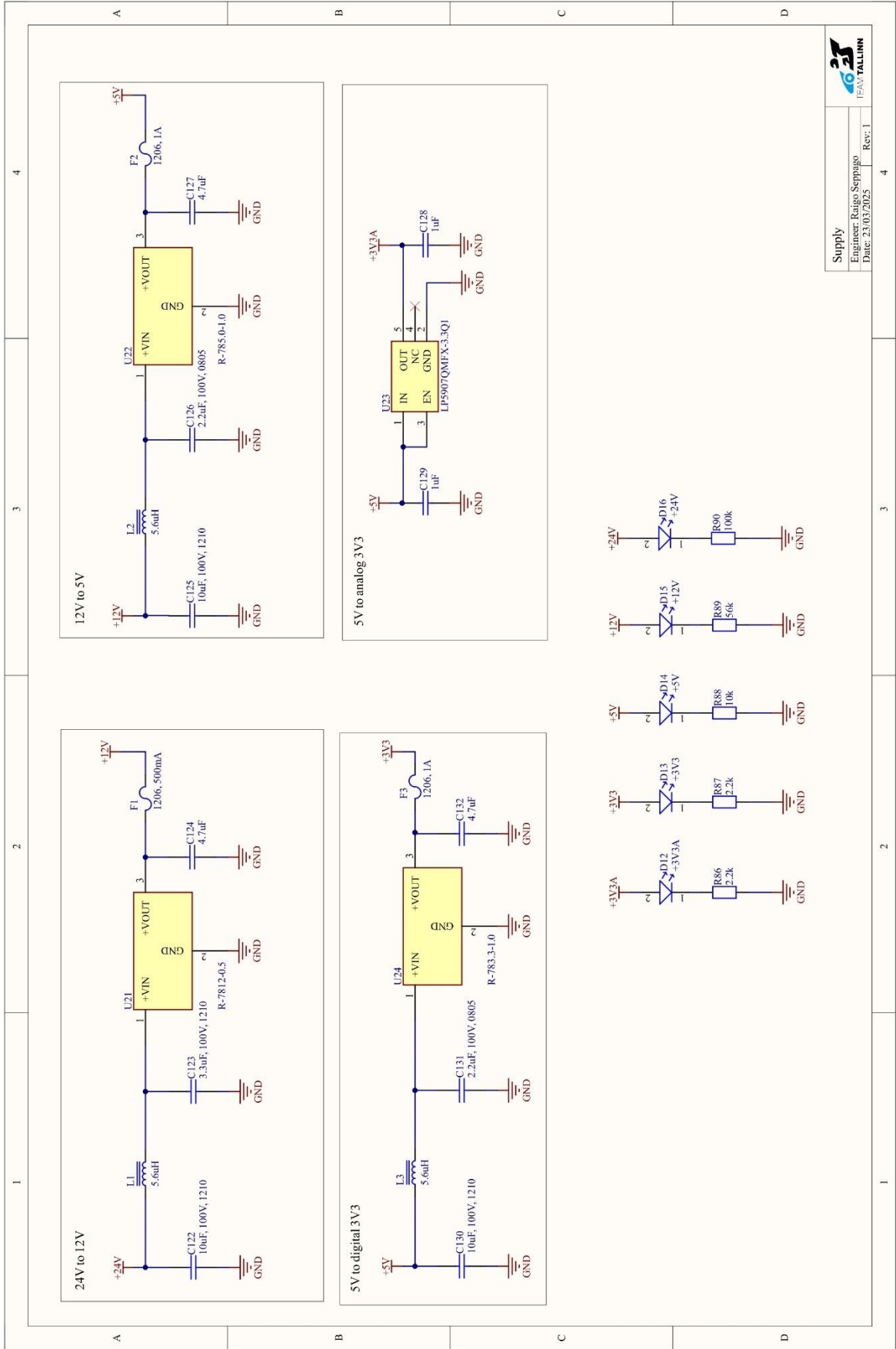






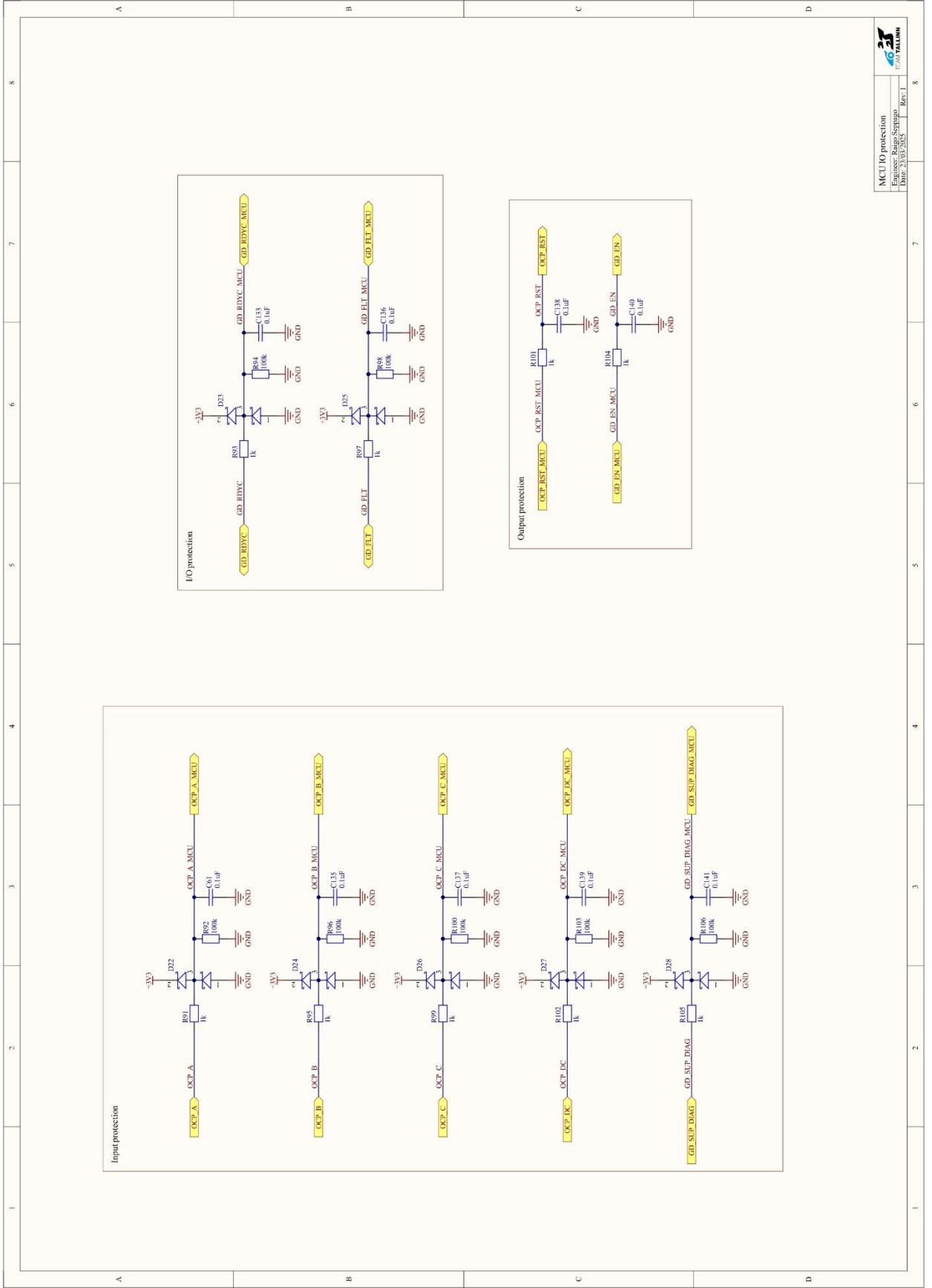




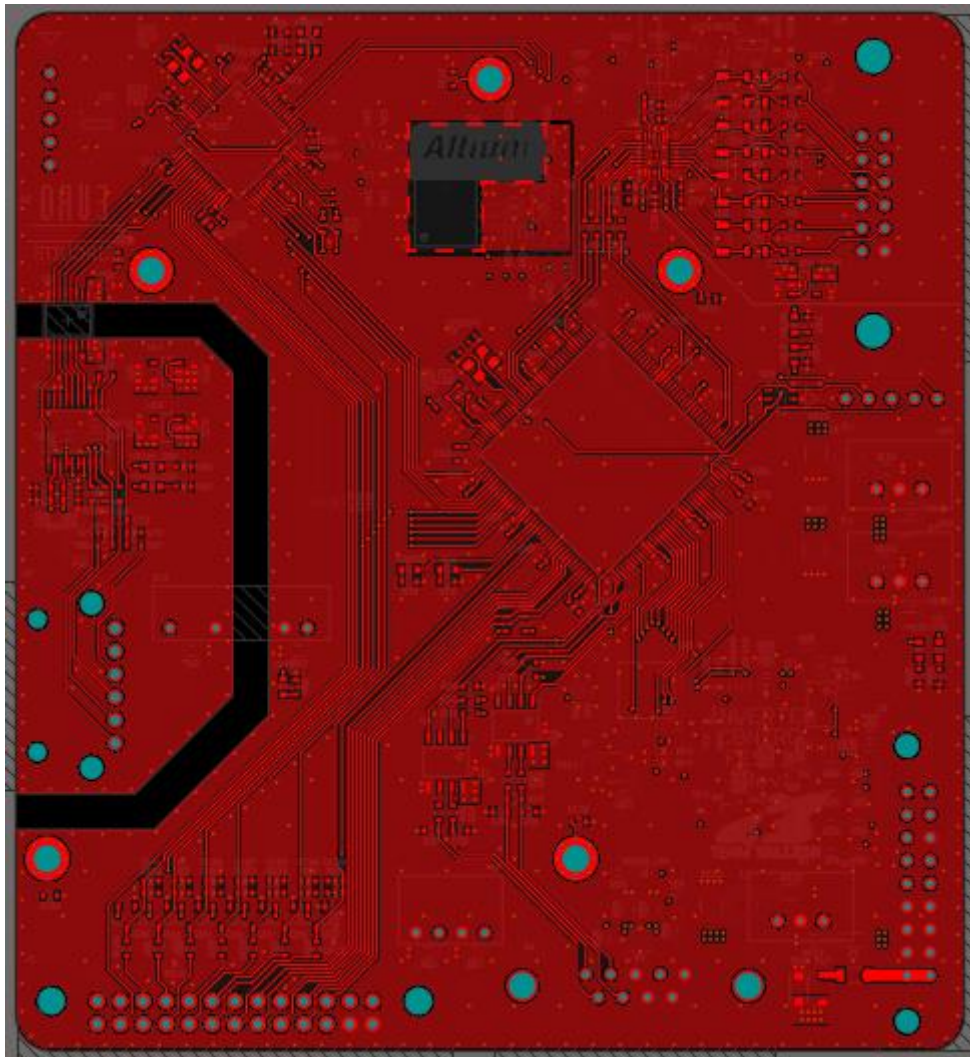




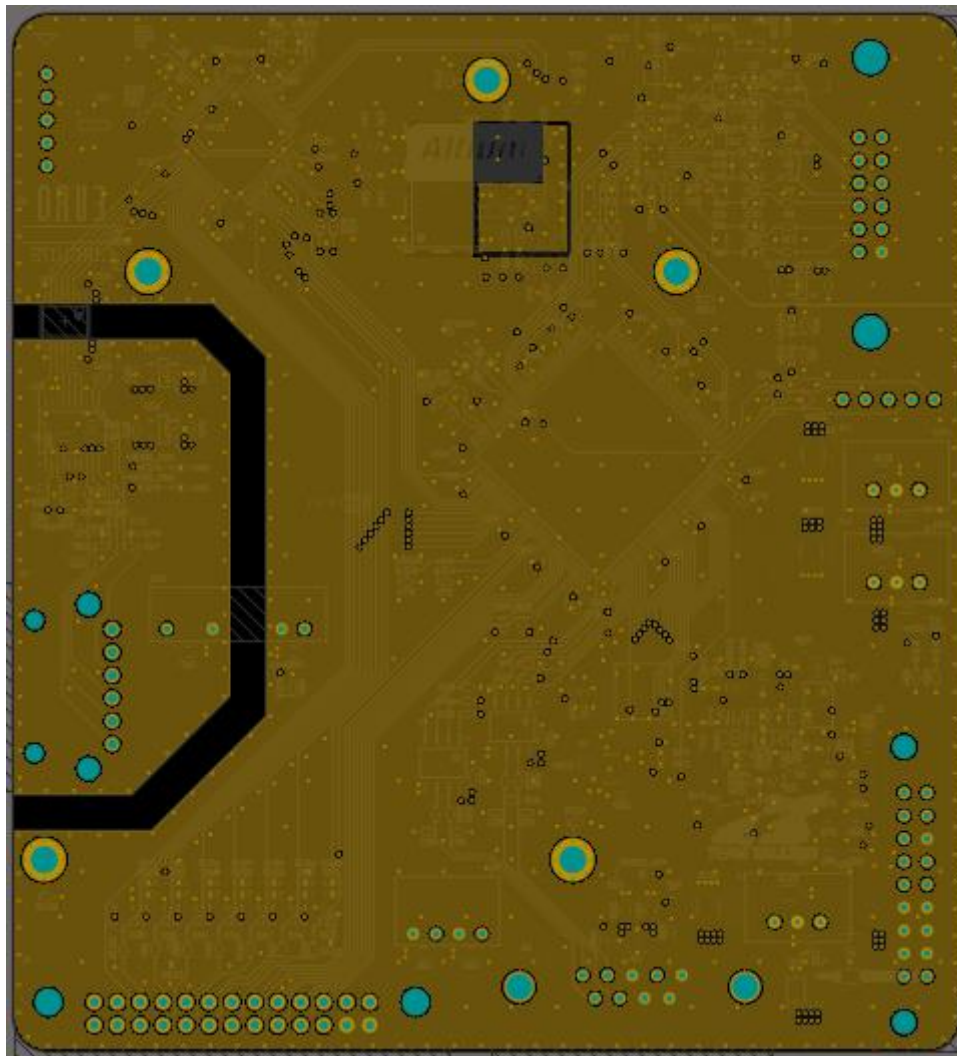




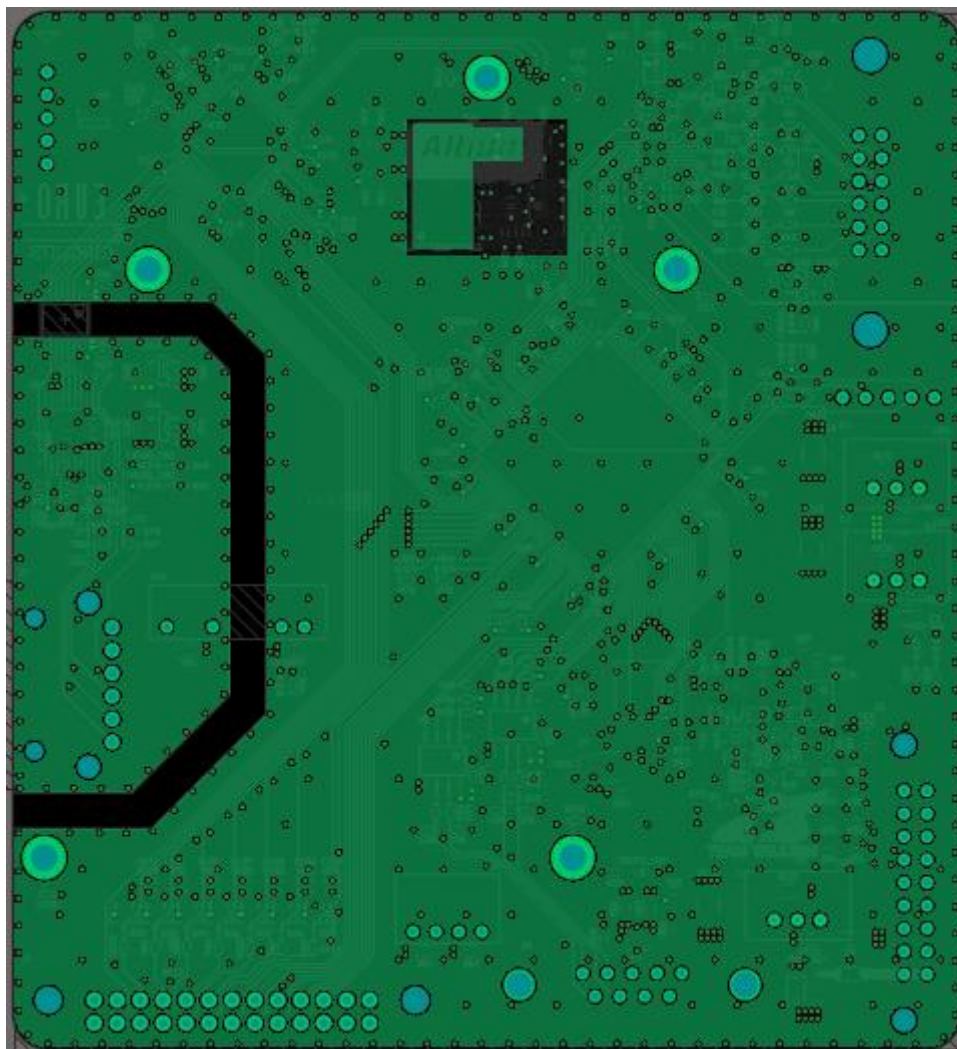
### Lisa 3 – Mootori kontrolleri trükkplaadi esimene kiht



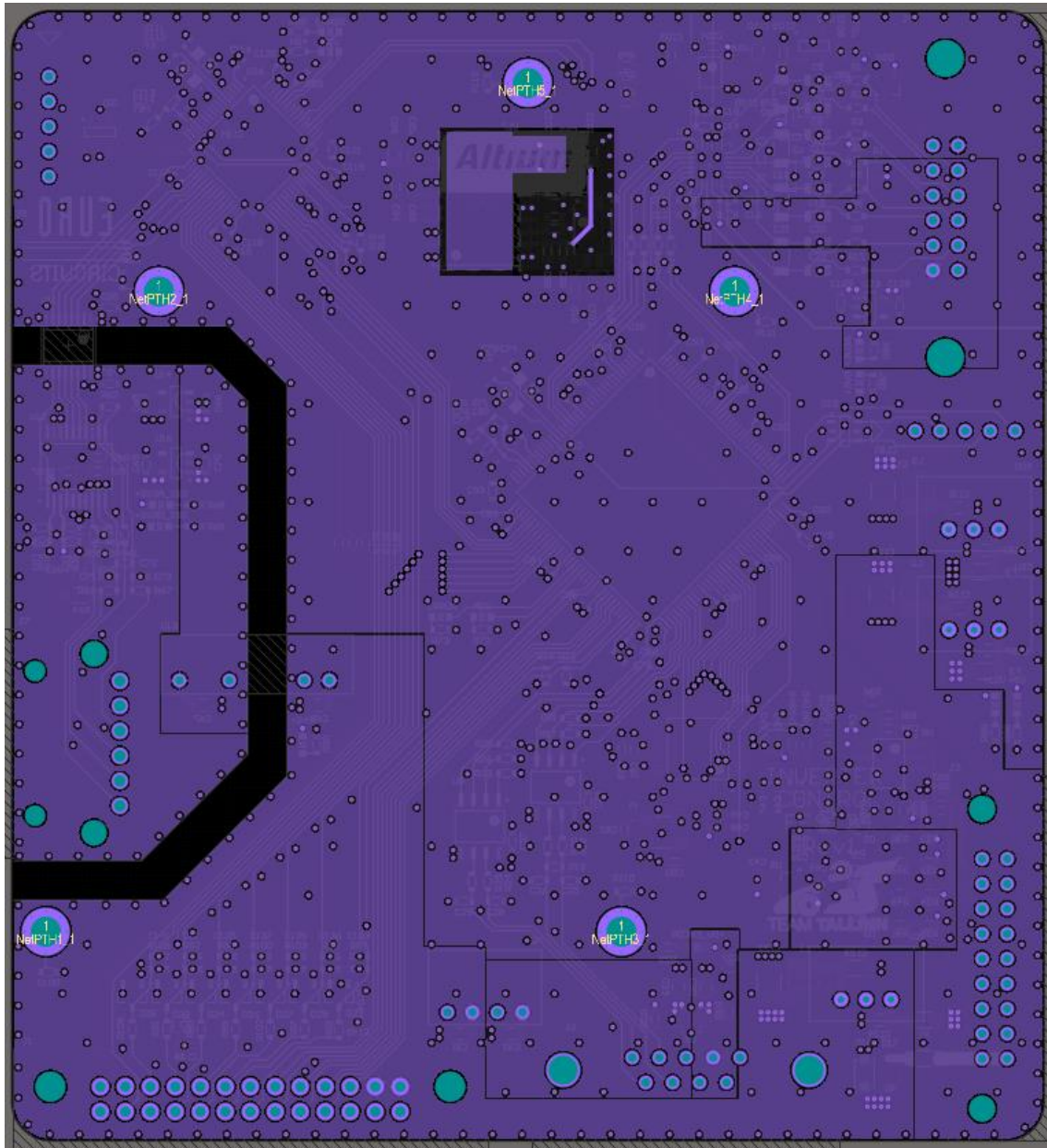
## Lisa 4 – Mootori kontrolleri trükkplaadi teine kiht



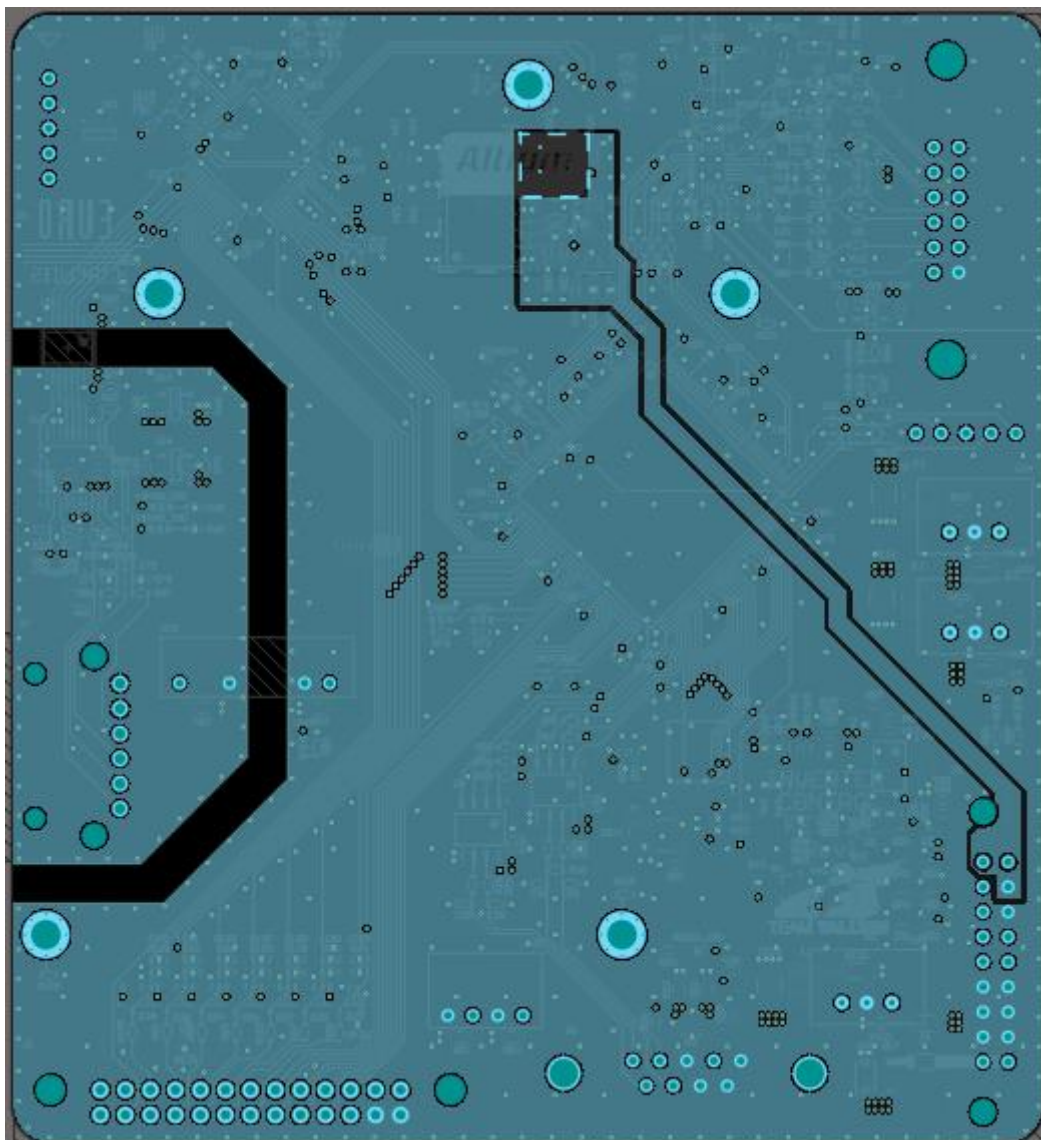
## Lisa 5 – Mootori kontrolleri trükkplaadi kolmas kiht



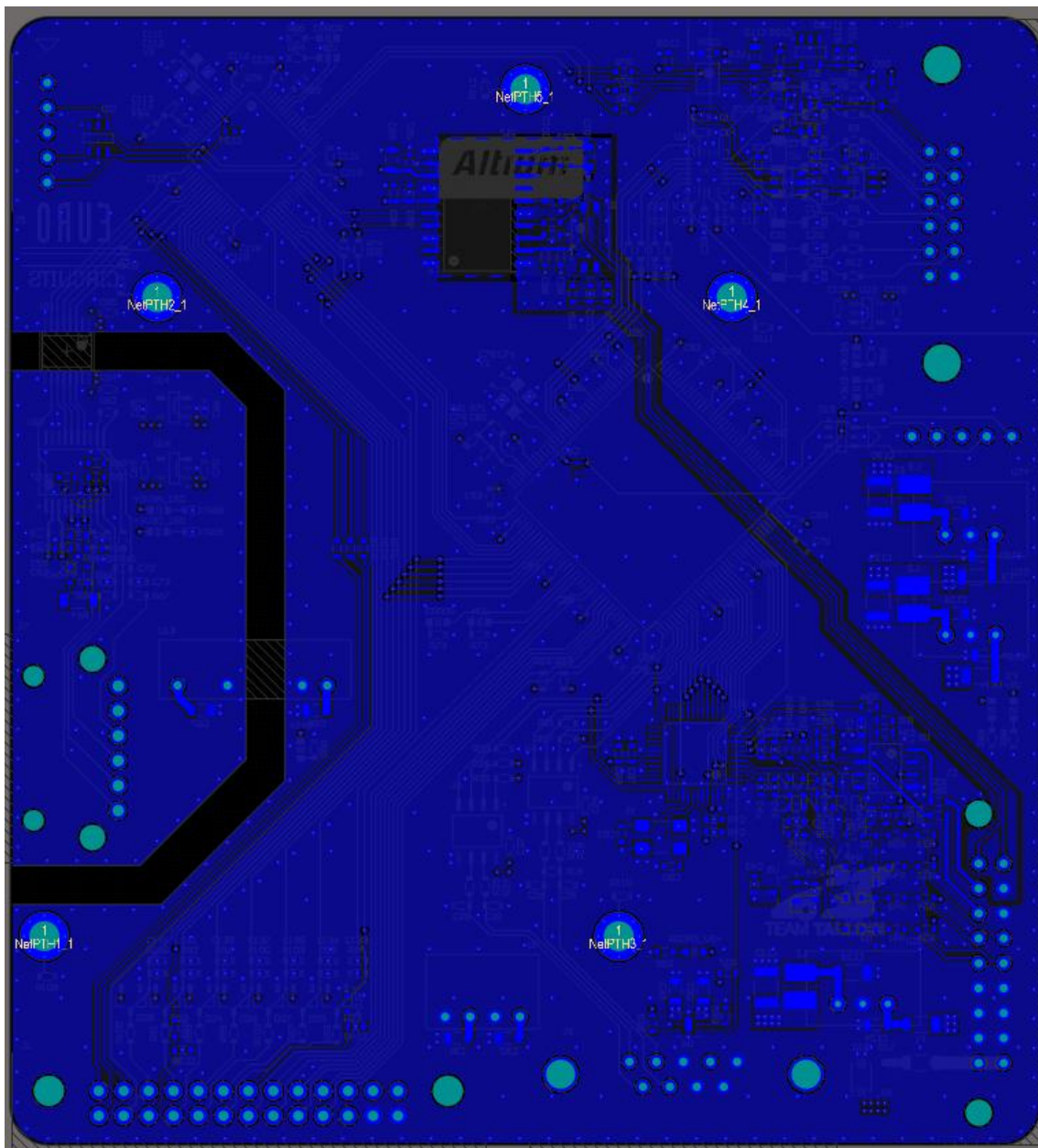
## Lisa 6 – Mootori kontrolleri trükkplaadi neljas kiht



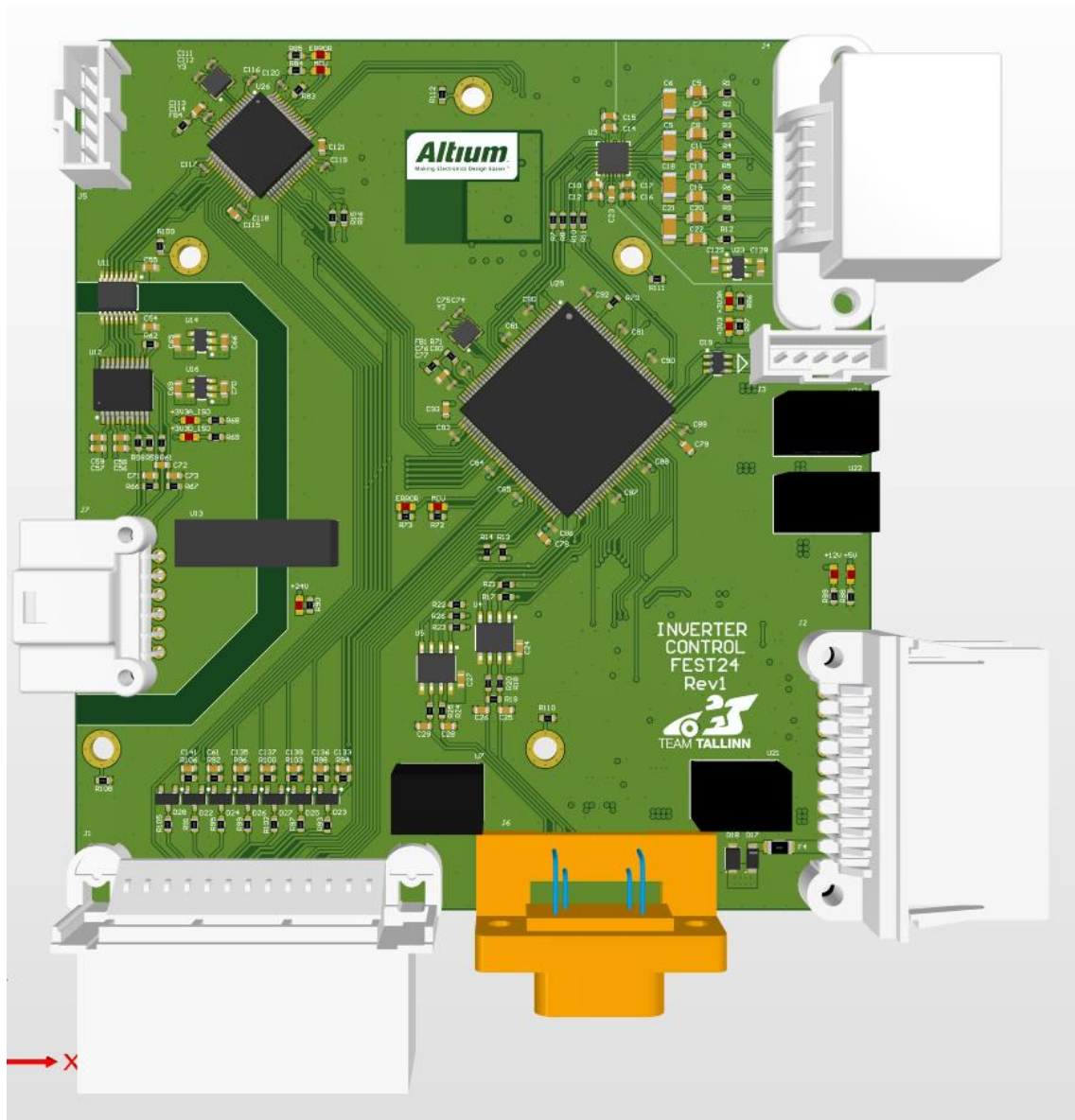
## Lisa 7 – Mootori kontrolleri trükkplaadi viies kiht



## Lisa 8 – Mootori kontrolleri trükkplaadi kuues kiht

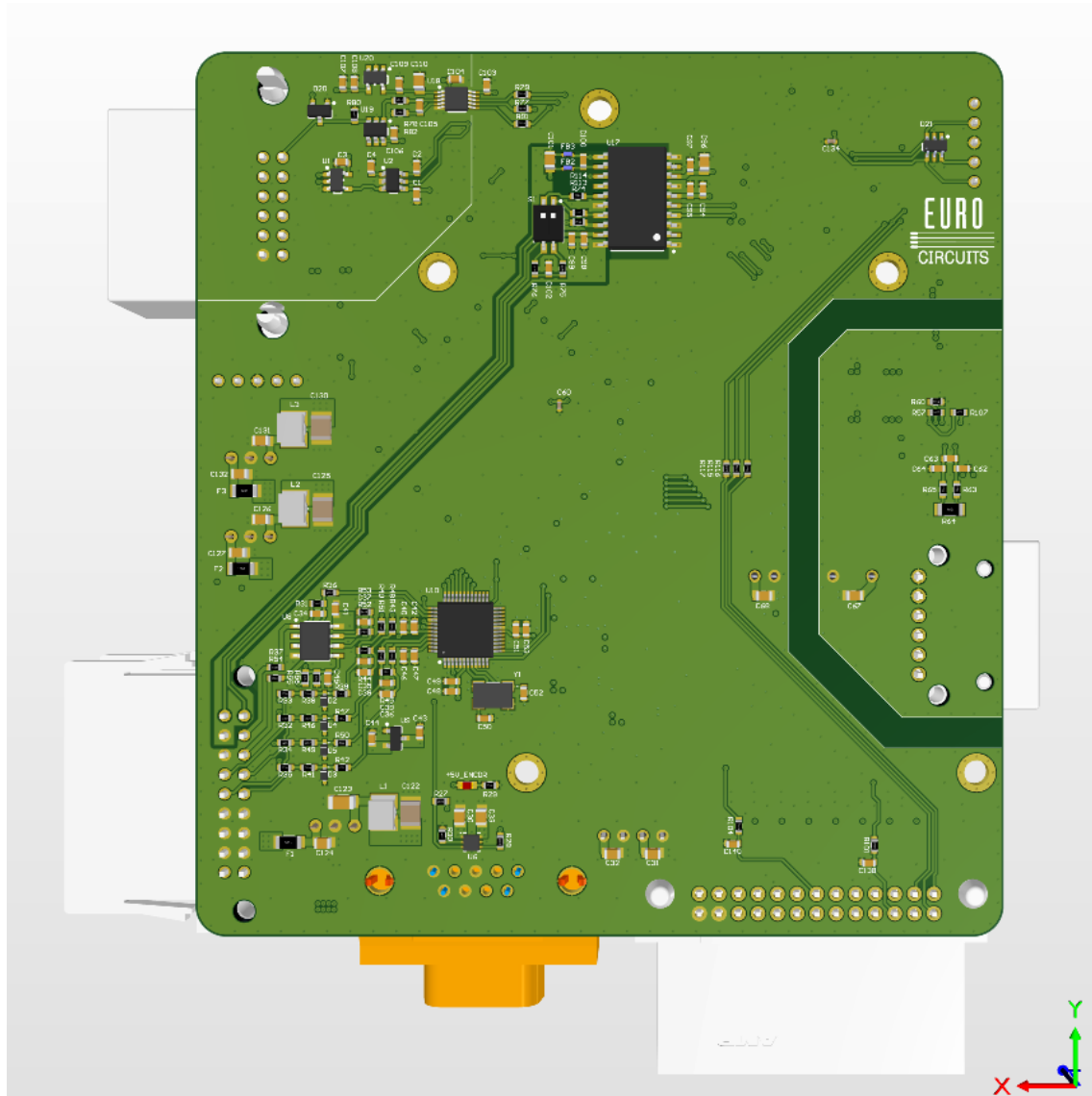


## Lisa 9 – Mootori kontrolleri trükkplaadi 3D mudeli pealtvaade





## Lisa 10 – Mootori kontrolleri trükkplaadi 3D mudeli altvaade



## Lisa 11 – Mootori kontrolleri veakoodid

Veakood	Vea kirjeldus
257	Enkoodri andmeid pole määratud aja jooksul saadud
513	Enkoodri sisemine viga
769	Viga enkoodri toiteahelas
258	Faasivoolude ADC sisemine viga
514	Faasivoolude ADC konfigureerimine ebaõnnestus
770	A faasi ülekoormus
1026	B faasi ülekoormus
1282	C faasi ülekoormus
1538	Pingevaheldi sisendpinge on liiga kõrge
8451	A faasi ülemise õla transistori draiveri I2C SDA on püsivalt madalal nivool
16643	A faasi alumise õla transistori draiveri I2C SDA on püsivalt madalal nivool
24835	B faasi ülemise õla transistori draiveri I2C SDA on püsivalt madalal nivool
33027	B faasi alumise õla transistori draiveri I2C SDA on püsivalt madalal nivool
41219	C faasi ülemise õla transistori draiveri I2C SDA on püsivalt madalal nivool
49411	C faasi alumise õla transistori draiveri I2C SDA on püsivalt madalal nivool
8707	A faasi ülemise õla transistori draiveri I2C SCL on püsivalt madalal nivool
16899	A faasi alumise õla transistori draiveri I2C SCL on püsivalt madalal nivool
25091	B faasi ülemise õla transistori draiveri I2C SCL on püsivalt madalal nivool
33283	B faasi alumise õla transistori draiveri I2C SCL on püsivalt madalal nivool
41475	C faasi ülemise õla transistori draiveri I2C SCL on püsivalt madalal nivool
49667	C faasi alumise õla transistori draiveri I2C SCL püsivalt madalal nivool
8963	A faasi ülemise õla transistori draiveri I2C viga
17155	A faasi alumise õla transistori draiveri I2C viga
25347	B faasi ülemise õla transistori draiveri I2C viga
33539	B faasi alumise õla transistori draiveri I2C viga
41731	C faasi ülemise õla transistori draiveri I2C viga
49923	C faasi alumise õla transistori draiveri I2C viga

<b>Veakood</b>	<b>Vea kirjeldus</b>
9219	A faasi ülemise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
17411	A faasi alumise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
25603	B faasi ülemise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
33795	B faasi alumise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
41987	C faasi ülemise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
50179	C faasi alumise õla transistori draiveri vearežiim (põhjustatud seadistava kaitseahela poolt)
9475	A faasi ülemise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
17667	A faasi alumise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
25859	B faasi ülemise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
34051	B faasi alumise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
42243	C faasi ülemise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
50435	C faasi alumise õla transistori draiveri lühisekaitse rakendatud 1. lühisekaitse mooduli poolt
9731	A faasi ülemise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
17923	A faasi alumise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
26115	B faasi ülemise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
34307	B faasi alumise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
42499	C faasi ülemise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
50691	C faasi alumise õla transistori draiveri lühisekaitse rakendatud 2. lühisekaitse mooduli poolt
9987	A faasi ülemise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud

<b>Veakood</b>	<b>Vea kirjeldus</b>
18179	A faasi alumise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud
26371	B faasi ülemise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud
34563	B faasi alumise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud
42755	C faasi ülemise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud
50947	C faasi alumise õla transistori draiveri riistvaraline ülekuumenemise kaitse rakendunud
10243	A faasi ülemise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
18435	A faasi alumise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
26627	B faasi ülemise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
34819	B faasi alumise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
43011	C faasi ülemise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
51203	C faasi alumise õla transistori draiveri programmeeritav ülekuumenemise kaitse rakendunud
10499	A faasi ülemise õla transistori draiveri ülepinge transistori paisul
18691	A faasi alumise õla transistori draiveri ülepinge transistori paisul
26883	B faasi ülemise õla transistori draiveri ülepinge transistori paisul
35075	B faasi alumise õla transistori draiveri ülepinge transistori paisul
43267	C faasi ülemise õla transistori draiveri ülepinge transistori paisul
51459	C faasi alumise õla transistori draiveri ülepinge transistori paisul
10755	A faasi ülemise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
18947	A faasi alumise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
27139	B faasi ülemise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
35331	B faasi alumise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
43523	C faasi ülemise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
51715	C faasi alumise õla transistori draiveri sisendi-väljundi andmevahetuse tõrge
11011	A faasi ülemise õla transistori draiveri sisendi ja väljundi CRC ei ühti

<b>Veakood</b>	<b>Vea kirjeldus</b>
19203	A faasi alumise õla transistori draiveri sisendi ja väljundi CRC ei ühti
27395	B faasi ülemise õla transistori draiveri sisendi ja väljundi CRC ei ühti
35587	B faasi alumise õla transistori draiveri sisendi ja väljundi CRC ei ühti
43779	C faasi ülemise õla transistori draiveri sisendi ja väljundi CRC ei ühti
51971	C faasi alumise õla transistori draiveri sisendi ja väljundi CRC ei ühti
11267	A faasi ülemise õla transistori draiveri sisendi CRC viga
19459	A faasi alumise õla transistori draiveri sisendi CRC viga
27651	B faasi ülemise õla transistori draiveri sisendi CRC viga
35843	B faasi alumise õla transistori draiveri sisendi CRC viga
44035	C faasi ülemise õla transistori draiveri sisendi CRC viga
52227	C faasi alumise õla transistori draiveri sisendi CRC viga
11523	A faasi ülemise õla transistori draiveri väljundi CRC viga
19715	A faasi alumise õla transistori draiveri väljundi CRC viga
27907	B faasi ülemise õla transistori draiveri väljundi CRC viga
36099	B faasi alumise õla transistori draiveri väljundi CRC viga
44291	C faasi ülemise õla transistori draiveri väljundi CRC viga
52483	C faasi alumise õla transistori draiveri väljundi CRC viga
11779	A faasi ülemise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
19971	A faasi alumise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
28163	B faasi ülemise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
36355	B faasi alumise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
44547	C faasi ülemise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
52739	C faasi alumise õla transistori draiveri sisendi-väljundi PWM signaali tõrge
12035	A faasi ülemise õla transistori draiveri tõrge draiveri töös
20227	A faasi alumise õla transistori draiveri tõrge draiveri töös
28419	B faasi ülemise õla transistori draiveri tõrge draiveri töös
36611	B faasi alumise õla transistori draiveri tõrge draiveri töös
44803	C faasi ülemise õla transistori draiveri tõrge draiveri töös
52995	C faasi alumise õla transistori draiveri tõrge draiveri töös
306	A faasi riistvaraline ülekoormuse kaitse on rakendunud
562	B faasi riistvaraline ülekoormuse kaitse on rakendunud
818	C faasi riistvaraline ülekoormuse kaitse on rakendunud

<b>Veakood</b>	<b>Vea kirjeldus</b>
1074	Pingevaheldi sisendvoolu riistvaraline ülekoormuse kaitse on rakendunud
307	Transistoride draiverid on vea seisundis
308	Mootori temperatuuri mõõteahela ADC konfigureerimine ebaõnnestus
564	Mootori temperatuur on liiga kõrge
820	Mootori temperatuuri mõõteahela ADC toitepinge viga
1076	Mootori temperatuuri mõõteahela ADC mõõteühendus on maaga lühises
1332	Mootori temperatuuri mõõteahela ADC negatiivne mõõteühendus on maaga lühises
1588	Mootori temperatuuri mõõteahela ADC mõõteühendus on toitega lühises
1844	Mootori temperatuuri sensor on lühises
2100	Mootori temperatuuri sensor on lahti ühendatud
309	Pingevaheldi transistoride temperatuur on liiga kõrge
310	Pingevaheldi sisendvool on liiga kõrge
566	Pingevaheldi väljundvool (regeneratiivne) on liiga kõrge
311	Vooluregulaator on saturatsioonis