

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl Kivi 179408IACB

TUDENGIVORMELI FEST20 MAAPINNA KIIRUSE ANDUR

bakalaureusetöö

Juhendaja: Peeter Ellervee
Ph.D.

Tallinn 2020

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: Karl Kivi

16.05.2020

Annotatsioon

Käesolev bakalaureusetöö annab ülevaate tudengivormeli FEST20 maapinna kiiruse anduri disainimisest ja tegemisest. Eesmärk on luua esmane versioon maapinna kiiruse andurist (edaspidi GSS), mis paigaldatakse vormeli FEST20 külge. Lõputööraames valmib isearendatud LED valgustrükkplaat ja kõikidele GSS süsteemi komponentidele 3D modelleeritud korpused. Lisaks arendatakse pilditöötlus tarkvara kiiruse arvutamiseks, muutes eelmise aasta programmkoodi ning luuakse LED valgustrükkplaadi mikrokontrollerile tarkvara.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 85 leheküljel, 9 peatükki, 57 joonist.

Abstract

Ground speed sensor for formula student racecar FEST20

The intention for building our own ground speed sensor (GSS) came from the unavailability of speed sensors in the commercial markets that would suit the needs of a formula student racecar. The applicable solution would consist of low cost, low output delay, maximum speed of 120km/h, low working height and CAN bus interface for data transfers. Thus, the in-house GSS was built that could satisfy all these considerations.

Due to four-wheel drive and in accordance with formula student rules, the speed sensor uses machine vision to determine the actual speed. Components of ground speed sensor are following – a camera with a lens, a computer for template matching algorithm, the in-house developed printed circuit board for image brightening and distance sensors for height measurement. This level of freedom enables us to modify the ground speed sensor to meet our formula student racecar's requirements.

The thesis is in estonian and contains 85 pages of text, 9 chapters, 57 figures.

Lühendite ja mõistete sõnastik

GSS	Ground Speed Sensor, Maapinna kiiruse andur
FEST _{xx}	20xx hooajal ehitatud tudengivormel
CAN	Controller Area Network
MCU	Mikrokontroller
DAC	Digitaal-analoogmuundur
LDO	Low-dropout regulaator
GPIO	General purpose input/output
ADC	Analoog-digitaalmuundur
HAL	Hardware Abstraction Layer
EXTI	External interrupt
LED	Light-Emitting Diode, valgust kiirgav diod
I2C	Inter-integrated circuit,, kahejuhtmeliides
N-MOSFET	Metal-oxide-semiconductor field-effect transistor, isoleeritud paisuga väljatransistor
FPS	Frames per second, kaadrit sekundis
DMA	Otsemällupöördus
CV_TM_SQDIFF	Sum of Square Differences
SSH	Secure Shell, turvakest

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract Ground speed sensor for formula student racecar FEST20	4
Lühendite ja mõistete sõnastik	5
Sisukord.....	6
Jooniste loetelu	8
1 Sissejuhatus	10
2 GSS süsteemi ülevaade.....	13
2.1 Riistvara.....	14
2.2 Tarkvara.....	18
3 Teooria.....	20
3.1 Kiiruse arvutamine	20
3.2 <i>Template Matching</i> kasutamine.....	22
3.3 Kuidas kaamera sensor töötab	24
3.4 Kaamera piksli suurus maapinnal.....	25
3.5 Maapinna kõrguse anduri disainimine.....	26
4 Varasema aasta lahendus	28
4.1 Riistvara.....	28
4.2 Tarkvara.....	29
5 GSS tarkvaraarendus Jetson TX1	31
5.1 Tarkvara paketid ja teegid	31
5.2 Programmi vooskeem.....	31
5.3 Arendamise etapid	32
6 LED valgustus trükkplaadi arendus.....	34
6.1 Elektriskeemide disainimine ja simuleerimine.....	35
6.2 Elektriskeemide koostamine.....	38
6.3 Trükkplaadi joonised	51
6.4 Testimine	55

6.5 Revisjon 2	61
7 Mikrokontrolleri tarkvaraarendus	67
7.1 Mikrokontrolleri perifeeria seadistamine	68
7.2 Mikrokontrolleri programmeerimine	70
8 Korpused	73
Trükkplaat	73
Jetson TX1	74
Kaamera hoidja	75
9 Kokkuvõte	76
Kasutatud kirjandus	77
Lisa 1. Trükkplaadi komponentide nimekiri (BOM)	78
Lisa 2. Trükkplaadi ülemine ja alumine korpus	83
Lisa 3. Trükkplaadi ja kaamera eemaldamine	84
Lisa 4. Valgustrükkplaat korpuses, kaamera ja kõrguseanduriga	85

Jooniste loetelu

Joonis 1. GSS parameetrid	13
Joonis 2. GSS süsteemi ülevaade	14
Joonis 3. Jetson TX1 [3].....	15
Joonis 4. Auvideo J120 laiendusplaat [4]	15
Joonis 5. Jetson TX1 koos laiendusplaadiga [5]	16
Joonis 6. Adafruit VL6180X kõrguseandur [6].....	16
Joonis 7. LED Valgustustrükkplaat	17
Joonis 8. Kaamera	18
Joonis 9. Objektiiv.....	18
Joonis 10. <i>Template Matching</i> [2]	23
Joonis 11. CV_TM_SQDIFF meetod.....	23
Joonis 12. GSS programmi vooskeem.....	32
Joonis 13. GSS trükkplaadi toitepingete tabel.....	36
Joonis 14. Trükkplaadi plokskeem	38
Joonis 15. Revisjon 1 ülemskeem.....	39
Joonis 16. Revisjon 1 analoog elektriskeem.....	40
Joonis 17. Revisjon 1 mikrokontrolleri elektriskeem.....	41
Joonis 18. Revisjon 1 silumise elektriskeem	42
Joonis 19. Revisjon 1 mikrokontrolleri LED indikaator elektriskeem	42
Joonis 20. Revisjon 1 väline taktsignaali generaatori elektriskeem	43
Joonis 21. Revisjon 1 mikrokontrolleri toite elektriskeem.....	43
Joonis 22. Revisjon 1 ühenduspistiku elektriskeem	44
Joonis 23. Revisjon 1 LED toite elektriskeem	45
Joonis 24. Revisjon 1 LED toite juhtimine elektriskeem	46
Joonis 25. Revisjon 1 toitemuundur 5V elektriskeem.....	48
Joonis 26. Revisjon 1 toitemuundurid 3V3 ja 3V3A elektriskeemid	48
Joonis 27. Revisjon 1 toitemuundurid 15V ja 12V elektriskeemid.....	49
Joonis 28. Revisjon 1 toitemuundurite LED indikaatorid elektriskeemid	50
Joonis 29. Revisjon 1 LED moodulite juhtimine elektriskeem	51

Joonis 30. Revisjon 1 LED moodul elektriskeem	51
Joonis 31. Revisjon 1 trükkplaadi ülemine kiht	52
Joonis 32. Revisjon 1 trükkplaadi alumine kiht	54
Joonis 33. Revisjon 1 24V trükkplaadi kiht	55
Joonis 34. Revisjon 1 0V trükkplaadi kiht	55
Joonis 35. Trükkplaadi ülemine kiht ilma komponentideta	56
Joonis 36. LED mooduli testimine	57
Joonis 37. Mikrokontrolleri LED indikaatorid testimine	57
Joonis 38. Kaamera katkestussignaali testimine trükkplaadiga	58
Joonis 39. Revisjon 1 välise DAC ühendus viga ülemskeemis.....	59
Joonis 40. Revisjon 1 LED toite vale COMP viigu disain	60
Joonis 41. Revisjon 1 LED toite COMP viigu parendus.....	60
Joonis 42. Revisjon 1 I2C suhtlussiinis viga	61
Joonis 43. Revisjon 2 ülemskeem.....	62
Joonis 44. Revisjon 2 ühenduspistiku elektriskeem	63
Joonis 45. Revisjon 2 LED toitemuunduri tagasiside elektriskeem	63
Joonis 46. Revisjon 2 trükkplaat, ülemine kiht	64
Joonis 47. Revisjon 2 trükkplaat, alumine kiht	65
Joonis 48. Revisjon 2 trükkplaadi 24V kiht	66
Joonis 49. Revisjon 2 trükkplaadi 0V kiht	66
Joonis 50. Mikrokontrolleri LED indikaatorid	68
Joonis 51. Mikrokontrolleri ADC perifeeria seadistus.....	69
Joonis 52. I2C perifeeria seadistus	70
Joonis 53. Mikrokontrolleri programmi vooskeem	71
Joonis 54. Trükkplaadi ja kaamera kinnitus korpused	73
Joonis 55. Jetson TX1 arvuti korpus	74
Joonis 56. Põhjaava kinnitus korpus	75
Joonis 57. Kaamera hoidja korpus.....	75

1 Sissejuhatus

Käesoleva bakalaureusetöö eesmärk on parendada tudengivormeli meeskonna FS Team Tallinn kõige uuema vormeli FEST20 maapinna kiiruse andurit (edaspidi GSS). GSS-i eesmärk on arvutada vormeli tegelikku kiirust maapinna suhtes. Lõputöö raames ei jõua FEST20 vormeli ehitus valmis ning seetõttu ei kirjeldata lõputöös GSS-i testimistulemusi. GSS-i loomine hõlmab järgmiste ülesannete lahendamist:

1. FEST19 vormeli GSS süsteemi ja teooria tundma õppimine;
2. valgustrükkplaadi disainimine, koostamine ja testimine;
3. pilditöötlusprogrammi täiustamine;
4. GSS korpuste 3D modelleerimine.

Lõputöös aitasid mitmed tudengivormeli liikmed GSS-i ülesandeid lahendada. Kristjan Kõuts kontrollis valgustrükkplaadi elektriskeeme ja trükkplaadi jooniseid ning genereeris tootmisfailid. Hans Anniste andis nõu elektriskeemide koostamisel ning Henri J. Norden selgitas, kuidas GSS FEST19 vormeli peal töötab. Täpsemalt selgitatakse ülesannete olemust järgnevates peatükkides.

Ülesanne

GSS täidab vormelis mitut ülesannet, mis aitavad vormeli kiiremini ja paremini sõitma panna. Vormel on neljarattaveoga, mis tähendab, et iga hetk veavad vormeli nelja ratast neli mootorit. Seetõttu ei ole võimalik mõõta vormeli tegelikku kiirust ratastelt, sest näiteks kiirendusel vormeli kõik rattad libisevad läbi. GSS-i ülesanded on mõõta vormeli tegelikku otse- ja külgliikumise kiirust. Seejärel saab vormeli juhtimissüsteem kasutada seda, et piirata vormeli rataste läbilibisemist ja suunata paremini veokontrolli erinevatele ratastele. Sensori andmeid saab ka kasutada vormeli ala- ja ülejuhitavuse kontrollimiseks. Viimasena on GSS kasutusel täpse odomeetria arvutamiseks ja salvestamiseks. Käesolev lõputöö selgitab GSS valmistamise protsessi ning kiiruse

arvutamise protsessi. GSS on disainitud tuvastama kiirust kuni 120km/h, mis on vormeli maksimaalne kiirus.

Tulemus

Käesoleva lõputöö raames on plaanis valmistada GSS esialgne versioon, mis paigutatakse vormelisse. Esialgne versioon eeldab, et on valmistatud kõik GSS korpused, mis hoiavad vormelis paigal trükkplaati, kaamerat koos objektiviiga ja arvutit. Valmis on isearendatud valgustustrükkplaat, mis aitab valgustada kaameraga tehtud pilte. Samuti, isearendatud tarkvara Jetson TX1 arvutile ning mikrokontrollerile.

Bakalaureusetöö ülevaade

Kõigepealt antakse GSS süsteemi ülevaade. Ülevaade kirjeldab tarkvara ja riistvara, mida GSS kasutab. Tarkvaraarendus jaguneb kaheks. Esiteks, Jetson TX1 arvutil toimub arendus C++ keeles ja programmi eesmärk on arvutada vormeli kiirus. Kiiruse arvutamine toimub pilditötluse abil ja tulemuse mõõtmistäpsust parandab vedrustus- ja kõrguseandur. Programm kirjutab kiiruse vormeli üldisele CAN suhtlussiinile. Teiseks, mikrokontrolleri programmikoodi kirjutamine LED valgustustrükkplaadile. Riistvara ülevaade selgitab täpsemalt, millised arvutisüsteemid on vajalikud GSS-i töötamiseks.

Teooria peatükk kirjeldab, kuidas kiiruse arvutamine toimib ning millised parameetrid on olulised. *Template matching* on funktsioon, mida GSS kasutab, et pilt A pildi B pealt tuvastada. Kaamera sensori tööpõhimõte on väga oluline, sest see aitab arendada isearendatud trükkplaati. Alampeatükk, mis kirjeldab kaamera piksli suurust maapinnal, annab olulise arusaamise, miks kaamera kõrguse muutumine sõiduajal mõjutab kiiruse täpset arvutamist. Viimase teemana tuuakse välja, millest alustada ja kuidas disainida maapinna kõrguse andur.

Varasema aasta lahenduse peatükk selgitab, millised muutused FEST20 vormeli GSS-i puhul tehakse võrreldes eelmise süsteemiga. Vastupidiselt toon välja, mida ei muudeta eelmise aasta süsteemis. Eraldi alampeatükis selgitan tarkvaraarendust Jetson TX1 arvutil ja tarkvaraarendust LED valgustustrükkplaadi mikrokontrolleril. Trükkplaadi disainimine ja mikrokontrolleri tarkvaraarendus olid kõige mahukamad tegevused, sest need arendati nullist. Seetõttu on need peatükid kõige rohkem lahti seletatud. LED

valgustuse trükkplaadiarenduse peatükk kirjeldab mitmeid teemasid, mis on olulised trükkplaadi tegemisel. Näiteks, elektriskeemide disainimine ja simuleerimine, elektriskeemide koostamine ning trükkplaadi jooniste tegemine. Trükkplaat revisjon 2 alampeatükk selgitab esimeses revisjonis välja tulnud vigasid ning tehtud parandusi. Mikrokontrolleri tarkvaraarenduse peatükk annab ülevaate, millises keskkonnas programmikood kirjutati. Viimane peatükk on pühendatud GSS süsteemis seadmete 3D modelleeritud korpustele.

2 GSS süsteemi ülevaade

Käesolev peatükk annab ülevaate kogu GSS süsteemist. Joonisel 1 on välja toodud GSS-i parameetrid. Süsteem koosneb mitmest alamsüsteemist – isearendatud trükkplaat, Jetson TX1¹, kaamera koos objektiiviga ja kõrguseandur. Need on ühendatud omavahel erinevate suhtlus- ja andmesuhtlusega. I2C suhtlussuhtlusele on ühendatud Jetson TX1 arvuti, kõrguseandur ja valgustustrükkplaat. Jetson TX1 arvuti ja kaamera on omavahel ühendatud USB 3.0 kaabliga ning kaamera annab pildi tegemise signaali läbi väljundliidese valgustustrükkplaadile. GSS süsteemi komponendid ja nende vahelised ühendused on kirjeldatud joonisel 2. Süsteemi riistvara ja tarkvara lahti seletamine toimub järgmistes lõikudes.

	GSS
tüüp	masinägamine
Liidesed	CAN (võimalik: Ethernet, Wi-Fi, USB, I2C, SPI)
min kiirus	0.25km/h
max kiirus	120km/h
resolutsioon	0.25km/h

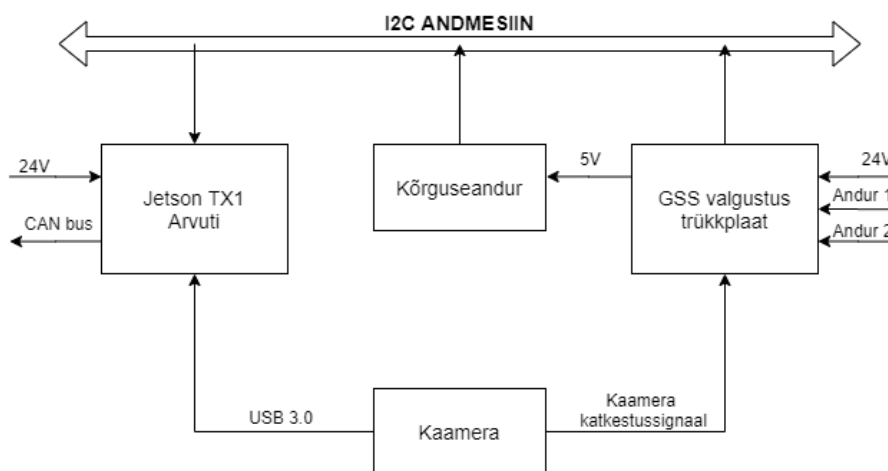
Joonis 1. GSS parameetrid

Jetson TX1 on kompaktne arvuti, mille põhiline eesmärk on maapinnast tehtud piltide töötlemine ning auto vertikaal- ja horisontaalsihis kiiruse arvutamine. Arvutatud tulemuse mõõtevea parendamiseks kasutab GSS valgustustrükkplaadilt saadud kõrguse ning vedrustuse andmeid. Need kirjeldavad kaamera kõrguse muutu maapinnast sõiduajal.

¹ <https://developer.nvidia.com/embedded/jetson-tx1>

Kõrguseandur on Adafruit VL6180X¹, mis on paigutatud auto põhja alla ning loeb auto kõrgust maapinnast. Kõrguseandurid töötavad programmikoodis koos vedrustusandmetega, sest mõlemad annavad programmile informatsiooni auto hetke kõrgusest. Kuna kõrguseandur ei ole piisavalt täpne, siis täpsust parandatakse vedrustusanduritega.

Isearendatud valgustrükkplaadi põhiline funktsioon on valgustada maapinda, kui kaamera teeb pilti. Kaamera pildi tegemise ajal kogub kaamera sensorile footoneid, mis ergastavad sensoril piksleid. Kui ergastamise kestvus on väga väike, siis päevavalgus ei jõua piisavalt palju kaamera sensoril piksli elemente ergastada ning pilt tuleb väga tume. Seetõttu tuleb kasutada eraldi valgusallikat, mis lülitab LED valgustuse sisse hetkeks, kui kaamera teeb pilti. Isearendatud trükkplaadi funktsioon on veel tõlkida auto esitelje vedrustusandurite analoogpinged digitaalsele kujule. Need andmed saadetakse I2C suhtlussiinile, kust Jetson TX1 arvuti saab neid sisse lugeda ja kasutada.



Joonis 2. GSS süsteemi ülevaade

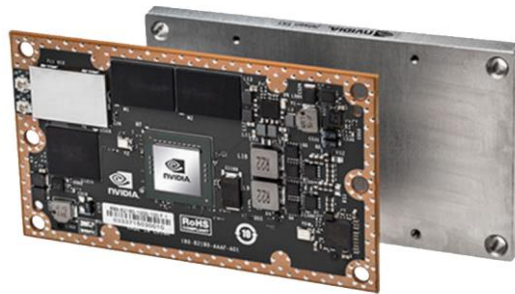
2.1 Riistvara

See alampeatükk kirjeldab täpsemalt, mis riistvaralised komponendid on GSS süsteemis kasutusel. Samuti, annab ülevaate nende kirjeldusest.

¹ <https://www.adafruit.com/product/3316>

Jetson TX1 arvuti

Jetson TX1 ei ole täielikult sobilik arvuti (joonisel 3). Juurde on ostetud arendusplaat, mis laiendab selle funktsionaalsust. Vormelitiim kasutab Auvidea J120¹ laiendusplaati. Arvuti toide on 18V. Kuna madalvoolupinge on 24V ja Jetson TX1 arvuti toitepinge peab olema madalam, siis kasutame toitemuundurit, mis teeb 24V pingest umbes 18V.



Joonis 3. Jetson TX1 [3]

Auvidea J120 laiendusplaat joonisel 4 on valitud seetõttu, et tal on olemas erinevad perifeeriad, mis sobivad meie GSS süsteemi arendamisel. Sellel on olemas CAN suhtlussiin, USB väljundliidesed ning programmeerimine toimub läbi Ethernet väljundi. Mini-HDMI liidest on mugav kasutada, et Jetson TX1 külge ühendada monitor tarkvaraarenduse ajal.



Joonis 4. Auvidea J120 laiendusplaat [4]

Joonisel 5 on välja toodud Jetson TX1 arvuti koos laiendusplaadi ning jahutusega. Kuna Jetson TX1 kasutatakse täie võimsuse juures, siis arvuti läheb väga kuumaks ning seetõttu on tal vaja jahutust. See süsteem on valitud mitme aasta eest teise tudengivormeli liikme poolt. Autor kasutab sama Jetson TX1 arvutit.

¹ <https://auvidea.eu/j120/>



Joonis 5. Jetson TX1 koos laiendusplaadiga [5]

Adafruit VL6180X kõrgusandur

Kõrguseandur on väga oluline seade, mis aitab määrata kaamera kõrguse maapinnast mõne millimeetri täpsusega. Adafruit VL6180X (joonisel 6) trükkplaat kasutab kõrguse mõõtmiseks objektilt tagasi peegeldunud signaali aega ning selle kaudu määrab objekti kauguse sensorist. Anduri täpsus on $\pm 2\text{mm}$. Kõrguse täpsust parandatakse auto esitelje vedrustuse lineaaranduritega.



Joonis 6. Adafruit VL6180X kõrguseandur [6]

Vedrustusandurid

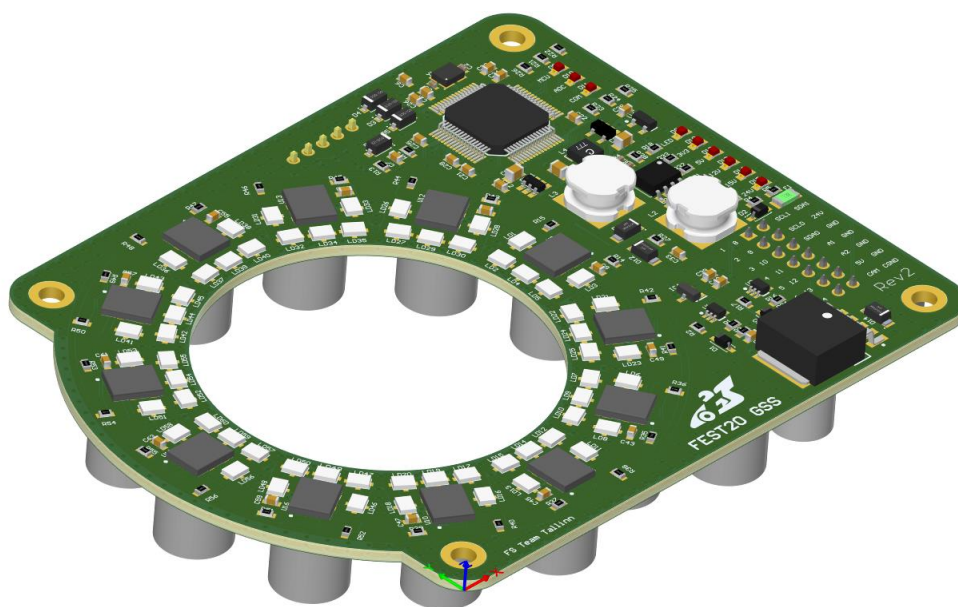
Vedrustusandurid on tudengivormeli liikme poolt valitud ning GSS kasutab ainult nende analoog andmeid. Mõlema vasaku ja parema ratta jaoks on eraldi lineaarandur, mis loeb vedrustuse kokku ja lahti liikumist ning tõlgendab seda elektrilise pingena.

Maapinna kiiruse andur kasutab FEST20 vormeli esitelje mõlema vedrustusanduri andmeid, et saada teada auto tegelik kõrgus pildi tegemise hetkel. See on väga oluline, sest sellest oleneb, kui suur ala maapinnast kaamera kaadrisse jäi. Kiiruse mõõtmine on väga tundlik kaamera vertikaalsihi muutustele ehk kaamera kõrguse muutumine maapinnast. Vedrustusandurid näitavad algselt kõrguseanduriga mõõdetud kõrguse muutust, mille põhjal on võimalik arvutada auto tegelik kõrgus täpsemalt.

Auto esitelje vedrustusandurite digitaalsed andmed ei liigu auto üldises suhtlusvõrgus (CAN suhtlussiin), seetõttu analoogpinged tõlgitakse digitaalsele kujule. Sellega tegeleb mikrokontroller LED valgustrükkplaadil.

Isearendatud LED valgustrükkplaat

Isearendatud LED valgustrükkplaadil (joonisel 7) on 12 LED moodulit. Igas moodulis on 5 jadamisi ühendatud LED valgustit, mida lülitatakse ühe N-MOSFET lülitiga. Moodulil on üks 1mF kondensaator, mis töötab kiiresti reageeriva vooluallikana, kui LED valgustust lülitatakse mõneks mikrosekundiks. Mikrokontroller kontrollib LED lülitusi ning nende sisse lülitatud ajaperioodi pikkust, milleks on paar mikrosekundit (oleneb, kui eredalt maapinda peab valgustama). Trükkplaadil on LED valgustuse heleduse kontrollimiseks elektriskeem, mida juhitakse mikrokontrolleri tarkvaras. Kaamera väljundliides annab katkestussignaali mikrokontrollerile, kuna peab LED valgustust lülitama. Analoog elektrihaelad on mõeldud puhverdama sisse tulevat nõrka signaali ning pingejagur tõlgib pinget mikrokontrollerile loetavaks pingevahemikuks 0V kuni 3V3.



Joonis 7. LED Valgustrükkplaat

XIMEA Kaamera ja objektiiv

GSS kasutab piltide tegemisel XIMEA kaamerat, mudel MQ003MG-CM¹ (joonisel 8). Kaamera maksimaalne kaadrisagedus on veidi üle 500 kaadri sekundis. Kaamera sensor on 1/3" suur ning on mõeldud koos töötama C/CS *mount* objektiiviga (joonisel 9). Üksiku piksli suurus on 7.4µm ning resolutsioon 648x488px. Madalaim seadistatav säriaeg on 54 µs, mis näitab kui pikk aeg on kaamera sensori valgusega ergastamine.



Joonis 8. Kaamera



Joonis 9. Objektiiv

2.2 Tarkvara

Antud töös kasutati kahte erinevat programmeeriskeelt C++ ja C. C++ programmeerimiskeelt kasutati Jetson TX1 arvutis, kus arendati kiiruse leidmise tarkvara, mis koosnes pilditötlusest ja suhtlussiinide CAN ja I2C rakendamisest programmikoodis. Mikrokontrolleri kood kirjutati C-keeles. Mikrokontrolleri kood kontrollib kõiki trükkplaadiga seoses olevaid funktsioone ning kuulab kaamera katkestussignaali.

Jetson TX1 arvutile arendatud tarkvara

Pilditöötlus ning kiiruse arvutamised toimuvad Ubuntu operatsioonisüsteemiga Jetson TX1 arvutis. CAN siinile kirjutatakse kiiruse arvutusi ning I2C suhtlussiiniga loetakse sisse vedrustusandmed ning kõrguseanduri tulemused. GSS kasutab logisüsteemi, mis on testimisel ja programmi silumisel väga kasulikud.

¹ <https://www.ximea.com/en/products/cameras-filtered-by-sensor-types/mq003mg-cm>

Mikrokontrolleri tarkvara LED valgustrükkplaadil

GSS valgustrükkplaadi funktsionaalsust juhitakse mikrokontrolleriga, millel ei ole operatsioonisüsteemi. Programmikood laaditakse otse mikrokontrollerile ning käivitatakse pinge lülitamisest. Neli põhilisemat funktsiooni mikrokontrolleril on kaamera signaali kuulamine, analoog andmete digitaliseerimine, suhtlussiin Jetson TX1 ja trükkplaadi vahel ning viimasena LED valgustuse lülitamine.

3 Teooria

Teooria peatükk selgitab, kuidas GSS arvutab auto kiirust. Kiirust arvutatakse nii auto liikumise sihis kui ka risti liikumist ehk auto vertikaal- ja horisontaalsihis liikumist. GSS ei arvuta ainult kiirust, vaid leiab ka läbitud vahemaa kiirusest. Need on kaks hetkel põhilisemat väärtust, mida GSS arvutab ja kirjutab vormeli üldisesse CAN suhtlussiinile. Kiiruse arvutamine hõlmab endas mitmelt GSS süsteemi komponendilt saadud info töötlust ning kokku panemist, et arvutada võimalikult täpne auto liikumiskiirus. Esmalt selgitan, kuidas arvutatakse kiirust kaameraga. Seejärel toon näite, kuidas leida lähteandmetega kiirused.

3.1 Kiiruse arvutamine

Kiiruse all mõeldakse, kui kiiresti auto liigub nii vertikaal- kui horisontaalsihis. Kiiruse arvutamise põhimõte on see, kui suure vahemaa liigub auto kindla aja jooksul, mida kirjeldab valem (1). Kui auto liigub ühe meetri ühe sekundiga, siis auto kiirus on 1m/s. See tähendab, et kiiruse arvutamiseks on vaja kahte komponenti. On vaja mõõta liikuva auto distantssi ning aega, kui kaua selleks kulub.

$$\frac{\text{auto liigutud vahemaa (m)}}{\text{vahemaa läbimise aeg (s)}} = \text{auto kiirus} \left(\frac{\text{m}}{\text{s}} \right) \quad (1)$$

Distants ja aeg – parameetrid vormeli hetkkiiruse arvutamiseks

Liigutud auto distantssi saame kasutades kaamerat. Kaamera teeb kaks järjestikust pilti. Pilti A kasutatakse musterpildina, mis tähendab, et pilt lõigatakse programmis väiksemaks. GSS puhul lõigatakse 648x488px pilt umbes 150x100px suuruseks. Musterpildi puhul jäetakse meelde koordinaadid, kust kohast pildi A pealt see muster välja lõigati. Pilt B on väga oluline, kuna pildi A pealt lõigatud musterpilt üritatakse sobitada pilt B peale. Selleks kasutatakse OpenCV teeki, kus saab kasutada *template matching* funktsiooni, mis üritab leida muster pilti A pildi B pealt. Kui funktsioon leiab sobiva tulemuse, siis ta tagastab koordinaadid, kust ta selle mustri leidis. Sellise meetodi põhjal saab võrrelda algselt võetud mustri koordinaate ning hiljem teise pildi pealt leitud mustri koordinaate. Nende koordinaatide vahe on liigutud vormeli distantss kahe pildi vahel. Järgmiseks on väga oluline teada, kui palju aega kulus kahe pildi tegemise vahel. Seda kirjeldab kaadrisagedus, mis näitab, kui sagedasti kaamera pildikaadreid teeb.

Toon näite, mis kirjeldab eelmises lõigus selgitatud kiiruse leidmist, kui me teame kaameralt saadud auto liigunud distantsi kindla aja jooksul, milleks on kahe kaadri vaheline aeg. Ütleme, et kaamera on 500 FPS (kaadrit sekundis), see tähendab, et valemi (2) põhjal on kahe kaadri vahel 2ms.

$$\frac{1}{500 \text{ fps}} = 2\text{ms} \quad (2)$$

Vormel sõitis kindla kiirusega. Kaamera tegi kaks pilti 2 ms vahega. Esimesena tehtud pildi A muster võeti koordinaatidelt (100, 100). Kui pilditöötlus töö lõpetas, leidis ta mustri teise pildi B koordinaatidelt (255, 100). See tähendab, et 2ms jooksul liikus kaamera maapinna suhtes 155 pikslit (255px-100px=155px). Seeläbi saab öelda, et auto kiirus oli 155 pikslit 2ms kohta. Nüüd jõuamegi järgmise olulise aspekti juurde – meil on vaja teada, kui palju millimeetreid maapinnal vastab ühele pikslile kaameras. Kui me saame teada, kui palju on üks piksel, siis saame ka teada kui kiiresti auto liikus. Näiteks, ütleme, et kaamerasensori ühele pikslile vastab maapinnal 0.10mm. Valem (3) näitab meie varasema näite puhul, et auto liikus 15.5mm 2ms kohta.

$$155\text{px} * 0.10 \frac{\text{mm}}{\text{px}} = 15.5\text{mm} \quad (3)$$

Kui see teisendada rohkem tuntud kiiruse ühikuks, siis $15.5/2\text{mm/ms} = 7.75\text{m/s} = 27.9 \text{ km/h}$. Seega auto kiirus mõõdeti selles punktis 27.9km/h. Piksli suurus maapinnal oleneb kahest asjast – objektiivist või kaamera kõrgusest maapinnast. Mida suurem on kaamera kõrgus maapinnast, seda suurem ala maapinnal on nähtav ühele pikslile kaamera sensoril. Kaamera kõrgust tõstes maapinna suhtes, siis kaamera vaateväli maapinnast suureneb. Samamoodi, kui kaamera kõrgus jätta samaks ning muuta kaamera objektiivi fookus kaugust väiksemaks, siis kaamera vaateväli läbi objektiivi muutub maapinnal suuremaks. Kaamera vaateväli määrab, kui suurt kiirust kaamera üldse mõõta saab.

Vertikaal- ja horisontaalsihi kiiruse arvutamine

Kaameraga tehtud pilditöötlusel saab arvutada mõlema telje kiirust vormelil. See tähendab auto vertikaalsihi kui ka horisontaalsihi liikumise kiirust (vastavalt valemid (4) ja (5)). Kui auto liigub sirgjooneliselt otse, siis vertikaalsihis liikumine on maksimaalne ja horisontaalsihis null. Sellisel juhul saab mõõta auto liikumist ainult vertikaalsihis liigunud kiirusest. Kuid vormeli kurvis liikumine on keerulisem, sest auto ei liigu ainult

vertikaalsihis, vaid ka horisontaalsihis. Seega resultaatkiiiruse arvutamiseks tuleb vertikaal- ja horisontaalsihi vektorid kokku liita, mille kaudu saame täpse auto kiiruse.

$$\frac{x_{vahemaa}}{t_{kaadrite\ vaheline\ aeg}} = x_{kiirus} \quad (4)$$

$$\frac{y_{vahemaa}}{t_{kaadrite\ vaheline\ aeg}} = y_{kiirus} \quad (5)$$

Kiiruse mõõtmises arvutatakse vertikaalsihis liigutud kiirus kaamera pildil x-koordinaatist ning horisontaalsihis liigutud kiirus y-koordinaadist. Nende vektorite summa on resultaatkiiirus valemis (6), mis kirjeldab auto tegelikku kiirust maapinna suhtes.

$$\sqrt{x_{kiirus}^2 + y_{kiirus}^2} = v_{resultaatkiiirus} \quad (6)$$

Tegemist on hetkkiirusvektoriga, mis tähendab, et kiirust mõõdeti kahe pildi toimumise ajal. See, kui kiiresti kiirusevektoreid arvutatakse oleneb arvutis jooksva programmi kiirusest. Eelmise aasta autol arvutati 70Hz, mis tähendab, et igas sekundis arvutati 70 korda auto kiirust.

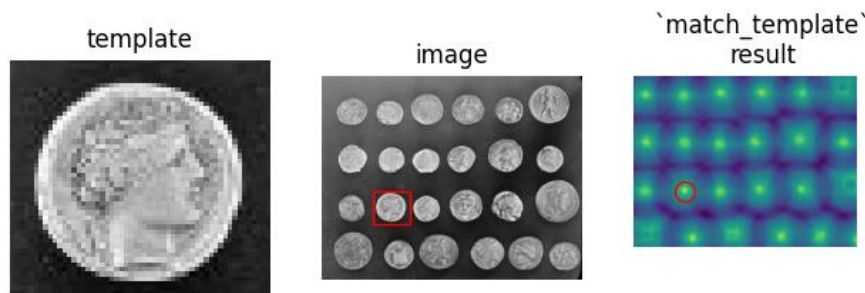
3.2 *Template Matching* kasutamine

*Template Matching*¹ (edaspidi pildimustri tuvastus) funktsiooni kasutamiseks on vaja kahte pilti, pilti A ja B. Joonisel 10 vastavad pilt A ja B pildile „template“ ja „image“. Üks piltidest on pilt A, mida tahetakse pildi B pealt leida. Pilti B kasutatakse, et tuvastada kõige tõenäolisem koht, kus pilt A võib asuda pildil B. Selleks kasutatakse nihutamise meetodit. Otsitavat asukohta alustatakse pildi B algus koordinaatidest (0,0) ning iga pikslit juures arvutatakse kindla meetodiga, kui hästi pilt sobib sinna asukohta ning tagastatakse number. Järgmisena nihutatakse pilti A edasi pildi B peal samas reas ning arvutatakse igale pildi pikslile arvuline väärtus. Pildist B moodustatakse maatriks (joonisel 10 välja toodud pilt nimega „match template result“), kus iga rea ja tulba pikslile on arvutatud arvuline väärtus, mis vastab pildile B. Maatriksi väärtused on arvutatud eelnevalt defineeritud meetodiga. Olenevat kasutatavast meetodist on kõige

1

https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

tõenäolisem piksli punkt, kas suurim või väiksem punkti väärtus (punktiväärtused on 0 kuni 1 ehk pildil piksel on must või valge).



Joonis 10. *Template Matching* [2]

Kõige tõenäolisema pildi tuvastus koordinaadi valimiseks kasutakse funktsiooni, mis leiab maatriksis kõige suurema või väiksema punkti ning tagastab selle koordinaadid, kus see maatriksis asub. See koordinaat näitabki, kus kohas leiti kõige täpsemini pildile A vastav koht pildil B. Tuleb tähele panna, et kui pilti ei leita, siis tagastatakse ikkagi pildi koordinaat, kus ta võib asuda. Seetõttu tuleks võrrelda maatriksis arvatud arve, et kui lähedal on see vastavalt meetodile kas 0 või 1. Näiteks, kui kaamera ei saa piisavalt valgust ja pildid on tumedad, siis pildimustri tuvastus algoritm annab väga vale tulemuse. GSS kasutab pildituvastamise algoritmina $CV_TM_SQDIFF^1$ meetodit (joonisel 11), mille põhjal arvutatakse maatriksi väärtused. Meetodi puhul võetakse piksli paar (pildilt A ja B) ning lahutatakse omavahel. Järgmiselt leitud arvude vahe võetakse ruutu ning kõik piksli paari vahe ruudud liidetakse kokku. Hiljem arvud normaliseeritakse 0 ja 1 vahele. Selle meetodi korral on kõige tõenäosem pildituvastus koordinaatideks väiksem arvutatud maatriksi väärtus. See tähendab, et mida väiksem on maatriksi arv nullile, seda tõenäolisem, et pilt A tõesti leiti pildi B pealt.

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

Joonis 11. CV_TM_SQDIFF meetod

1

https://docs.opencv.org/2.4/modules/imgproc/doc/object_detection.html?highlight=matchtemplate#matchtemplate

3.3 Kuidas kaamera sensor töötab

Tähtis on selgitada, kuidas kaamera mõned komponendid töötavad ja miks see oluline on GSS-i juures. Kasutatav kaamera kasutab *global shutter*'it, mis tähendab, et kogu kaamera sensor kogub valgust korraga ja samal ajal ehk iga valgustundlik piksel kaamera sensoril hakkab koguma pildi kohta informatsiooni samal ajahetkel. See on väga oluline, sest pildi tegemise ajal vormel liigub.

Kaamera sensori säriaeg

Järgmine oluline teema on säriaeg (ingl.k *exposure time*), mis on aeg, mille jooksul pääseb valgus sensorile kaameras. Kui liiga vähe valgust langeb säriajal kaamera sensorile, siis pilt tuleb liiga tume ja detaile ei ole näha. Kui liiga palju saabub valgust kaamera sensorile, siis pilt on üle helendatud ning samuti ei ole pildi pealt detaile näha. Mõlemal juhul pildimustri tuvastus algoritm ei leia head tulemust ning arvutab vale piksli nihke. Oluline on saada just õige valgus hulk kaamera sensorile, et pilt oleks selge.

Säriaeg peaks olema võimalikult väike, sest liikuva pildi korral tuleb pilt udune, kui kaamera säriaeg on väga pikk. Udune pilt on jällegi väga kehv pilditöötuseks ning seda tuleks kindlasti vältida. Süsteemis kasutatava kaamera säriaeg on 54 mikrosekundit. See tähendab, et kaamera kogub pilditegemise ajal sensorile valgust 54 mikrosekundi jooksul. Kui kaamera pildistab 54µs säriajaga, siis üldjuhul päevavalgusest ei piisa, et pilt tuleks piisavalt hele. Seetõttu on vaja lisa valgusallikat, mis tekitaks piisavalt valgust kaamera sensorile väikse säriaja jooksul.

GSS disainiti auto liikumissihis arvutama kuni 120km/h. See on väga suur kiirus kaamera jaoks. Toon näite, kui palju liigub vormel 54 mikrosekundi jooksul, kui auto liigub 120km/h. Lõigu all toodud valem (7) näitab, et vormel liigub edasi 1.8mm pildi tegemise ajal. See ei tundu palju, kuid arvestades, et umbes ühe piksli suurus maapinnal on 0.10mm, siis see tähendab, et pildi tegemise ajal kirjutatakse ühe piksli andmed üle järgneva 18 piksli poolt, mis tähendab, et pilt on väga udune ning detailsus puudub.

$$\frac{120\text{km/h}}{3.6} \times 54 \times 10^{-6}\text{s} = 1.8\text{mm} \quad (7)$$

Idealis oleks hea, kui kaamera säriaeg on nii väike, et vormel liigub vähem kui ühe piksli võrra edasi maksimaalse kiiruse korral. Sellisel juhul tuleks väga selge pilt. Kuna kaamera säriaega ei saa vähendada 54 mikrosekundilt väiksemaks (see on kaamera riistvara poolt vähim, mis on võimalik), siis kasutatakse kaamera objektiivi ees värvifiltrit. Värvifilter on kindla lainepikkusega, mis laseb läbi ainult konkreetse lainepikkusega valgust (621nm). GSS mõõdab kiirust vormeli põhja all, mis tähendab, et sinna jõuab väga vähe päevavalgust. Seetõttu, kui värvifilter on paigutatud kaamera objektiivi ette ning kaamera säriaeg on seadistatud kõige madalamaks (54 μ s), siis pildi tegemise ajal kaamera sensorile jõutud footonite hulk on nii väike, et pilt tuleb täiesti must. Seetõttu on vaja isarendatud trükkplaati, millel on LED valgustus. See valgustab maapinda just samal lainepikkusel, mida värvifilter kasutab. See tähendab, et kaamera sensorile pääseb valgus ainult siis, kui LED valgustus töötab. Trükkplaat on kaameraga ühendatud ning kuulab, kuna pilti tehakse ning siis lülitab sisse eelnevalt defineeritud ajaperioodiks LED valgustuse. Lülituse aega kontrollides saame ise kontrollida, kui kaua kaamera sensori valgustamine toimub. Sellise meetodiga on võimalik kaamera säriaega vähendada. Kaamera säriaega vähendatakse 54 μ s kuni 3.5 μ s. See tähendab, et kui kaamera sensor lülitab sisse pildi tegemise ehk hakkab valgust koguma sensorile, siis trükkplaat lülitab ainult 3.5 mikrosekundiks LED valgustuse sisse, mis tähendab, et kaamera sensorile jõuab ainult selle aja sees maapinnalt valgustatud valgus. Seeläbi on kaamera säriaega vähendatud ning pildid ei tule enam udused.

3.4 Kaamera piksli suurus maapinnal

Eelnevas peatükis sai mainitud, et kiiruse arvutamisel on oluline teada, kui palju millimeetreid vastab ühele pikslile kaamera sensoril. Piksli suurus oleneb kaamera objektiivist või kaamera kõrgusest maapinnast.

Kaamera objektiiv

Kaamera objektiividel on erinevad fookus kaugused, mis muudavad kaamera vaatevälja suurust. Kaamera fookus kaugusest oleneb, kui laia või kitsa nurgaga objektiiv on. GSS-i puhul oleks parem laiema vaatevälja nurgaga objektiiv, sest siis saab kaamera koos objektiiviga viia võimalikult lähedale maapinnale. Muidugi kaamera saab nii lähedale viia, kuni pilt on veel fookuses ja selge. Kaamera objektiivi on mõistlik vahetada, kui kaamerat ei ole võimalik kõrgemale viia, kuid on vaja saada suurema vaateväljaga pilt.

Kuna FEST20 autol on GSS paigutatud auto põhja alla, siis seal on väga vähe ruumi kõrguse muutmiseks. Seetõttu pidi see aasta ostma uue objektiivi, millel oleks väiksem fookus kaugus (12mm asemel 8mm). See suurendas kaamera vaatevälja kaamera kõrgust muutmata.

Kaamera kõrgus maapinnast

Eelmine aasta FEST19 auto peal oli kaamera seadistatud auto taha. See lubas rohkem ruumi kaamera paigutamiseks erinevatele kõrgustele. Seetõttu FEST19 peal kaamera objektiivi vaatevälja nurk oli kitsam, aga kaamera paigutatud kõrgemale maapinnast.

3.5 Maapinna kõrguse anduri disainimine

Kõige esimesena tuleks paika panna, kuhu GSS tuleks paigutada autos ja milline oleks maksimaalne mõõdetav kiirus. Sellest oleneb, milliste parameetritega tuleb GSS-i riistvara (kaamera, objektiiv ja trükkplaat). Järgmisena tuleb valida kaamera, sest vastavalt maksimaalsele valitud lõppkiirusele oleks hea valida kaamera, millel oleks just GSS-i jaoks head parameetrid. Kui kaamera on valitud, siis viimasena tuleks valida objektiiv, mis mõjutab lõplikku kaamera vaatevälja. Kaamera vaateväljast oleneb lõplikult GSS-i maksimaalne mõõdetav kiirus. Isearendatud trükkplaati saab teha nii, et sobituks kokku ülejäänud süsteemiga.

GSS paigutus vormelis

Eelnevalt juba mainitult on kõige olulisem esimese asjana paika panna GSS-i asukoht vormelis. Asukoha leidmisega on alati probleeme, sest vormelis ei ole kunagi väga palju ruumi. Sellegipoolest tuleb leida võimalikult hea koht, kuhu GSS sobiks. Eelmine aasta, FEST19 auto peal, oli GSS paigutatud vormeli tagaosasse, masinast väljapoole. See aasta sai valitud GSS asukohaks vormeli esitelg. See tähendab, et kaamera kõrgus muutus kõvasti maapinnale lähemale, sest paigutati põhja alla. Esiteljel on kõige suurem muutus horisontaalsihis, sest kui auto rattaid keeratakse, siis kõige esimesena reageerib esitelg.

Kaamera olulised parameetrid

GSS süsteemi puhul on kaameral päris palju olulisi parameetreid, mida tuleks kaamera ostmisel valida. Igal parameetril on jällegi oma tugevad ja nõrgad küljed. Selgitan

lühidalt, kuidas valida õige kaamera kõige paremate parameetritega. Kuna kahjuks sellise GSS süsteemi jaoks ei ole müügil just õige riistvara näitudega kaamerat, siis tuleb valida just see, mis sobiks kõige paremini.

Esiteks, kaamera kaadrisagedus, mis näitab, kui kiiresti suudab kaamera teha kahte järjestikust pilti. See tähendab, et milline on kiiruse mõõtmiseks aeg kahe pildi tegemise vahel. Näiteks 500 FPS kaameral on 2ms kaadrite vaheline aeg, kuid 1000 FPS oleks ainult 1ms. See aeg on oluline, sest väiksem aeg tähendab väiksemat kaamera vaatevälja. Oluline on, et kui vormel liigub kindla kiirusega, siis pildi A tehtud musterpilt oleks pildi B pealt leitav ehk pilt A peab mahtuma kaamera vaatevälja maapinnal. Teisi öeldes, vormel liigub nii kiiresti, et pilti A ei ole võimalik leida pildi B pealt, sest vormel liikus terve kaamera vaatevälja jagu edasi. Kui kaamera teeks kaks korda kiiremini pilte, siis sama kiiruse arvutamiseks piisab kaks korda väiksemast vaateväljast.

Teiseks, kaamera sensori suurus on oluline vastavalt objektiivile. Mida suurem on kaamera sensori vaateväli sama objektiivi korral, seda suurem on ka kaamera vaateväli. Valiku puhul tuleks eelistada suuremat kaamera sensorit, kui valik baseerub ainult sensori suurusel. Sensori suurusega tuleb ka resolutsioon. Tavaliselt on suurema resolutsiooniga kaamerad aeglasema kaadrisagedusega. Kaamera resolutsioon mõjutab, kui väikse sammuga on võimalik tuvastada kiiruse muutust. Hetke GSS süsteemil on kiiruse sammuks 0.25km/h.

Kaamera objektiivi valik

Kaamera objektiivi valik on samuti keeruline. GSS-i puhul soovitaks jätta selle pärast kaamera ja GSS asukoha valikut. Sellisel juhul on need parameetrid fikseeritud ja siis on mõistlik alles kaamera objektiivi valima hakata. Kui valitud sai parim kaamera ja GSS-i asukoht, siis objektiiviga saab veel muuta kaamera vaatevälja suurust, millest oleneb maksimaalne mõõdetav kiirus.

Objektiivi valikul tuleks tähele panna minimaalset töötavat distantsi ehk kui lähedale saab kaamera maapinnale panna enne, kui pilt enam ära ei fokuseeri. Teiseks, tuleb tähele panna, kui suured on objektiivi vaatevälja nurgad. Vaatevälja nurk tuleneb otseselt objektiivi fookus kaugusest. Mida väiksem on fookus kaugus, seda suurem on objektiivi vaatevälja nurk. Üldjuhul on 6mm kuni 12mm fookus kaugustega objektiivide

minimaalne fookuseerimiskaugus 100mm. See tähendab, kui paigutada kaamera koos objektiiviga lähemale fookus kaugusest 100mm, siis pilt on udune. Objektiiv ei tekita pilti täpselt kaamera sensorile. Seetõttu objektiividel on minimaalne fookuseerimise kaugus piiravaks teguriks, kui lähedale on võimalik üldse kaamerat maapinnale viia.

4 Varasema aasta lahendus

FEST19 ja FEST20 vormeli maapinna kiiruse anduritel on mitmeid erinevusi. Esimene suur erinevus on GSS-i paiknemine vormelis. FEST19 GSS [1] puhul paiknes see vormeli tagaosas ning oli kõrgemal maapinnast. FEST20 GSS on auto esiteljel ning paigutatud auto põhja alla, rohkem maapinnale lähemale. Teine suur erinevus on trükkplaat. Selle aasta GSS süsteemil on tehtud täiesti uus trükkplaat, mille puhul lisati funktsioone ning muudeti põhimõtteid, kuidas funktsioonid trükkplaadil töötavad. Kolmas suurem muudatus on korpused. Uus trükkplaat on teiste mõõtmetega ning asetseb auto põhja all, mis tähendab, et eelmise aasta korpuseid ei saa kasutada. Samuti ei saa kasutada arvuti korpust, sest arvuti paikneb see aasta auto eesosas. Eelmine aasta paiknes see auto tagaosas.

Jetson TX1 arvutit ja kaamerat ei vahetatud muu seadme vastu. Kaameral otsiti väga palju alternatiivne, kuid eelneval aastal kasutatud kaamera oli kõige paremate näitajatega. Kaamera objektiivi muudeti, sest kaamerat viidi maapinnale lähemale, mis tähendas, et sama vaatevälja saamiseks tuli osta väiksema fookus kaugusega objektiiv.

4.1 Riistvara

Selgitan täpsemalt, mis muutus võrreldes eelmise aasta riistvaraga ning mis jäi samaks. Toon välja põhjused, miks oli vaja midagi muuta.

Muutused

Esimene suurem muutus on LED valgustustrükkplaadis. Kuna selle aasta GSS trükkplaadi funktsioone tuli juurde ja need muutusid võrreldes eelmise aasta omaga, siis kahjuks ei saanud kasutada eelmise aasta trükkplaati. Trükkplaadi funktsionaalsuse

tahtsin lahendada paremini ja töökindlamana, kui seda tehti eelnevalt. Eelmine aasta ei kasutatud mikrokontrollerit plaadil, vaid kasutati loogikalülitusi ja 555 taimer kiipi, mis lülitas kindla perioodiga LED lülitusi, kui kaameralt tuli signaal. FEST20 GSS puhul tegeleb kaamera katkestussignaali lugemise ja LED lülitusega mikrokontroller. Samamoodi muutus LED lampide konfiguratsioon skeemis kahe jadamisi LED-lambi asemel 5 jadamisi ühendatud LED-lambi peale. LED koguarvu tõsteti 40 LED-lambi pealt 60 LED-lambi peale.

FEST20 GSS-le lisasin ka analooghela elektriskeemid, mis lugesid vedrustusandureid ning mikrokontroller tõlkis need digitaalsele kujule. Vedrustusandureid kasutan resultaatkiruse leidmisel kõrguse vea kompenseerimiseks. Korpused tuli disainida täiesti nullist ning vastavalt uue asukoha paigutusele.

Mida ei muudetud

GSS riistvaras ei muudetud Jetson TX1 arvutit, mis tegeleb pilditöötlusega, sest arvuti on väga hea jõudlusega ning füüsiliste dimensioonidega. Vormelis on ruumi vähe ja seetõttu on oluline, et arvuti oleks kompaktne ja võimas.

Eelmine aasta kasutati ka kõrguseandureid alguses, kuid GSS projekti tegija mainis, et need olid nii ebatäpsed, et ta lõpetas nende kasutamise ning seadis programmi konstanti, mida kasutati kõrguse mõõduna. See aasta kasutan kõrguseandureid ainult vormeli algekõrguse lugemisel, kui auto käivitatakse. Ülejäänud kõrguse muutumine esiteljel loetakse vedrustusanduritelt.

Kaamera puhul otsiti väga palju alternatiivne, mida saaks kasutada. Eelkõige üritati leida kõrgema kaadrisagedusega kaamerat, millel oleks vähemalt sama resolutsioon ja kaamera sensori suurus. Kuid kahjuks GSS süsteemi jaoks ei olnud ühtegi õiget kaamerat, mis rahuldaks kõiki vajalikke parameetreid. Seetõttu kasutatakse sama kaamerat, mis eelmine aastagi.

4.2 Tarkvara

Tarkvara alampeatükk selgitab, millised süsteemi komponendid muutusid ja millised ei muutunud. GSS tarkvara on valgustustrükkplaadil ja Jetson TX1 arvutis.

Muutused

Jetson TX1 arvutis ei kasutanud ma eelmise aasta programmi, sest programmi töö selgeks tegemisel kirjutasin kõik tarkvara ümber. Kuna esialgse kiiruse arvutamise programmi tööle saamisel ei ole väga mahukat koodi vaja, siis võtsin eelmise aasta koodi baasiks ning kirjutasin olulise osa ümber. Sellise meetodiga saan kontrollida iga funktsiooni veakatesusi ja oskan kontrollida programmikoodi käitumist. Eelmise aasta programmikoodi puudus oli stabiilne kiiruse arvutus. Vormeliga oli tehtud testsõite, kus kiiruse arvutused olid väga head, kuid tihti viskas sisse väga valesid kiiruse arvutusi ning märja pinna peal ei tundnud GSS üldse töötavat. Programmikoodis oli väga hästi tehtud programmikoodi käivitamiseks süsteem. See tähendas, et programmi sai käivitada mitmete erinevate valikutega, kuid ma leian, et see ei ole väga oluline enne, kui GSS töötab hästi. See teeb ainult koodi keerulisemaks. Hetke GSS eesmärk on esmalt kood saada nii töökindlaks, et kiiruse arvutamine oleks võimalikult täpne ja stabiilne. Seejärel lisada juurde põhjalikum silumissüsteem ning sellejärel programmi töötsükli kiiremaks optimeerida.

Eelmise aasta GSS-il ei olnud trükkplaadil mikrokontrollerit. See tähendab, et selle aasta LED valgustrükkplaadil tuli kirjutada nullist ka mikrokontrolleri kood, mis kontrolliks kõiki trükkplaadi funktsioone.

Mida ei muudetud

Kuna Jetson TX1 arvutit ei vahetatud uue arvutisüsteemi vastu välja, siis kasutusel on ikka Ubuntu operatsioonisüsteem ning muud varasemad seadistused. Arvutiga ühendus seadistamiseks ja programmeerimiseks on läbi SSH. Programmi jooksub teenus, mis käivitatakse automaatselt, kui vormelis lülitatakse sisse madalvooluahel ehk 24V.

5 GSS tarkvaraarendus Jetson TX1

Käesolev peatükk annab ülevaate GSS-i pilditöötamise esmasest tarkvara arendusest, sest vormel ei ole veel valmis ning seetõttu ei saa kiiruse andurit testida realses keskkonnas. Kirjeldan ainult programmeerimise etapid, mis esmalt peavad olema pilditöötamise tarkvara töötamiseks. Täielik tarkvaraarendus ei mahu lõputöö raamesse.

5.1 Tarkvara paketid ja teegid

Jetson TX1 tarkvaraarenduses kasutan Tegra Ubuntu Visual Studio Code¹ programmi, kuhu saab C++ programmi koodi kirjutada. Programmi kompileerimiseks kirjutasin GNU Make² faili, mille abil on mugav kokku koondada kõik programmifailid.

Programmi töötamiseks kasutan kolme põhilisemat teeki, milleta ei oleks arendustöö võimalik. Esiteks, OpenCV³, millega toimub *template matching*. Teiseks, kaamera teek⁴, millega saab kontrollida kaamera erinevaid parameetreid. Kolmandaks, kõrguseanduri VL6180x teek⁵, millega saab suhelda kõrguse anduriga ning lugeda sisse andmeid millimeetrites.

5.2 Programmi vooskeem

Programmi vooskeem (joonisel 12) selgitab, kuidas GSS pilditöötamise programm töötab. Programmitöö algab algandmete initsialiseerimisega, mis tähendab, et algandmed algväärtustatakse. Seejärel algab programmi põhiline lõputu tsüklil, mis hakkab arvutama kiirust. Kõige esimese sammuna teeb kaamera kaks järjestikust pilti. Mõõdetakse kahe kaadri vaheline aeg, mis on hiljem oluline kiiruse arvutuses. Edasi kasutab OpenCV teek kaamera tehtud pilte ning tõlgib need graafikakaardile arvutamiseks maatriksiteks. Jetson TX1 integreeritud graafikakaardil tehakse *template matching* arvutus, sest see matemaatiline ülesanne on lihtsam ja kiirem kui protsessoril.

¹ <https://code.visualstudio.com/docs/setup/linux>

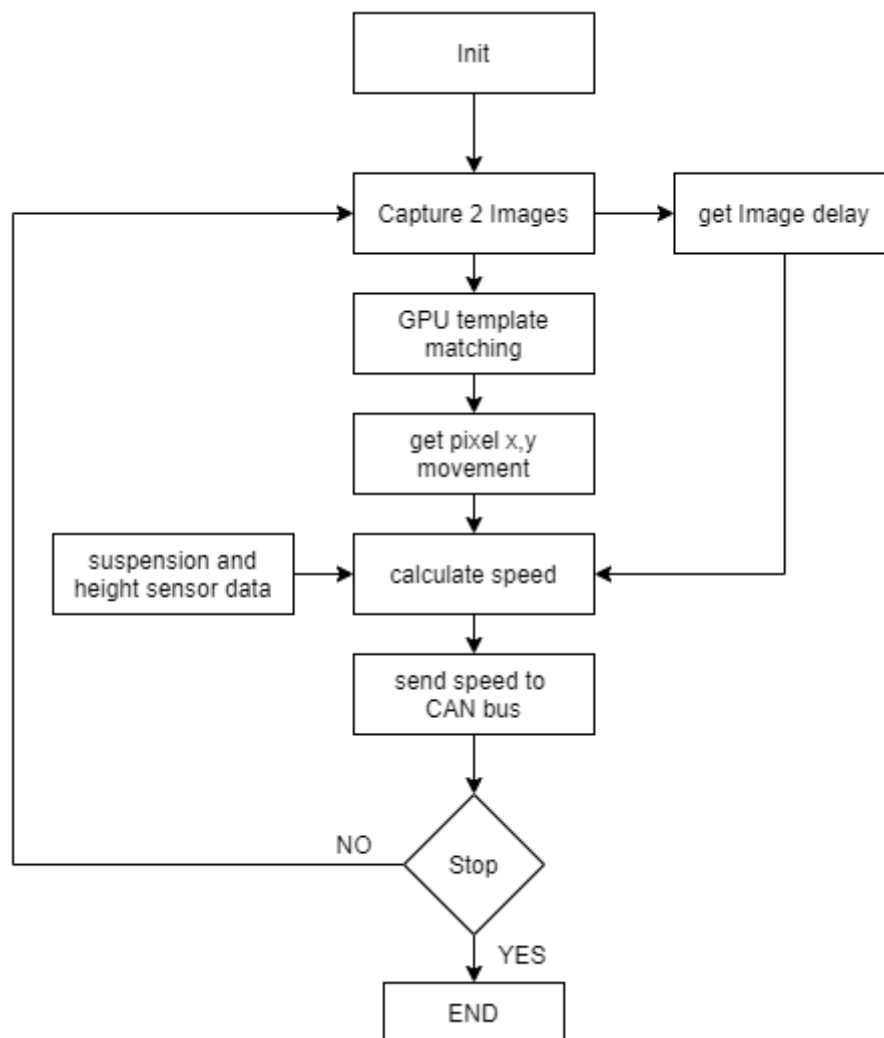
² <https://www.gnu.org/software/make/>

³ <https://opencv.org/>

⁴ <https://github.com/nwadedx/ofxXimea/blob/master/include/xiApi.h>

⁵ <https://www.st.com/en/embedded-software/stsw-img003.html>

Template matching funktsioon tagastab resultaattmaatriksi. Resultaattmaatriksilt arvutatakse kõige tõenäolisem koordinaat, kus musterpilt võib asuda. Kiiruse arvutamine on nüüd võimalik, sest kõik parameetrid on olemas. Olemas on algne koordinaat, kus musterpilt võeti ja *template matching* andis koordinaadi, kus musterpilt leiti. Nende koordinaatide vahe annab auto liigutud vahemaa pikslites – nii vertikaalkui horisontaalsihil. Seejärel kasutatakse algselt meelde jäetud kaadri vahelist aega, et leida mitu pikslit auto sekundis liigub. Viimasena leitakse piksli suurus kõrguse- ja vedrustusanduritelt loetud andmetelt, et arvutada lõplik kiirus.



Joonis 12. GSS programmi vookeem

5.3 Arendamise etapid

Jetson TX1 arvutis ei arendatud edasi eelmise aasta koodi. Kood oli mahukas ning ta ei täitnud kahte kõige olulisemat eesmärki – arvutada kiirust võimalikult täpselt ning et kiiruse arvutus oleks stabiilne. See tähendab, et arvutustel ei teki anomaaliaid. Koodi

tundma õppimisel mõisteti, et esimese testkoodi töötamine ei vaja väga mahukat koodi ning seetõttu kirjutati uus kood ümber. Ümberkirjutamine toimus vana koodi baasil, kuid üritati koodi ülesehitust muuta loogisemaks. Lisati objekt orienteeritud programmeerimise keelele omaseid struktuure.

GSS koodi tarkvaraarendus

GSS projekti tegemine algas eelmise aasta programmikoodi tundma õppimisega. Uuriti esmalt kõiki teeke, mida programm kasutab. Seejärel uuriti eelmise aasta tudengivormeli liikmelt, kuidas programm on ehitatud ja kuidas ta pealiskaudselt töötab. Järgmiselt tehti vooskeemi, mille kaudu üritati välja selgitada, kuidas programm töötab ehk millised funktsioonid kutsutakse millal välja ja millises järjekorras. Programmi erinevate moodulite testimisega taheti välja selgitada, kuidas funktsioonid täpselt töötavad ehk funktsioonide vajalikkus, nende töö põhimõte ja veakäitlus. Idee programmimoodulite ümber kirjutamises oli enda õpetamine ning süsteemi selgeks tegemine. Hiljem teadmiste rakendamine GSS tarkvara parendamisel.

Kui endale koodi selgeks tegemine oli valmis, siis alustasin uue projektiga ning hakkasin GSS programmikoodi kirjutama. Esimese töötava tarkvara versiooni mõte ei tohi olla keerukas. Eesmärk on saada tööle töökindlam kiiruse arvutamine FEST20 vormelil kui eelmine aasta. Pilditöötlus tarkvara arvutab kiiruse kasutades kõrguseandurit ja vedrustusanduri andmeid, et parendada arvutatud tulemust.

6 LED valgustus trükkplaadi arendus

GSS FEST20 LED valgustrükkplaadil on kolm põhilist eesmärki, mis on väga olulised maapinna kiiruse anduri töötamiseks.

Esiteks, trükkplaadile on paigutatud 12 LED valgustusmoodulit, mis igaüks koosneb viiest jadamisi LED-lambist ehk kokku on trükkplaadil 60 LED-valgustit. Valgustust lülitatakse korruga ning ainult pilditegemise ajaperioodiks – see tähendab mõne mikrosekundi vahemikus. Valgustuse juhtimine toimub MOSFET draiveriga, millele annab signaali mikrokontroller. Mikrokontroller ootab kaameralt pilditegemise käsklust ning seejärel lülitab valgustust. Valgustuse sisselülitatud aeg ning valgustustugevus on kontrollitavad mikrokontrolleri tarkvaraga.

Teiseks, GSS on väga tundlik kaamera üles-alla liikumise suhtes ning seeläbi mõjutab see arvutatud auto kiirust. Esimene mõte oli paigutada kõrguseandur. Mis loeks auto kõrgust iga kiirusepunkti arvutamise juures, kuid eelmise aasta projekti põhjal sai kindlaks, et väga keeruline on leida suurema täpsusega kõrguseandurit kui $\pm 2\text{mm}$. Kasutatav kõrguseandur oli Adafruit VL6180X, mis sobis kõige paremini GSS süsteemi, kuid ei saavutanud piisavat täpsust ning tekitas lõplikus kiiruse arvutustes liiga ebastabiilse tulemuse. Seetõttu, see aasta kasutan kõrguseandurit ainult auto algekõrguse mõõtmiseks ning ülejäänud kompenseeritakse vedrustusandurite andmetega. Trükkplaadil on kaks analoogskeemi, mis loevad esitelje vedrustusandurite signaale. Mikrokontroller tõlgib need digitaalsele kujule 12-bitise analoog-digitaalmuunduriga.

Kolmandaks, GSS koosneb Jetson TX1 arvutist ning disainitud trükkplaadist. Nende omavahelisel suhtlusel on kasutatud I2C protokollit. Selle protokollit kasuks sai otsustatud seetõttu, et vaja on ainult kahte juhet (SCL ja SDA) ning see annab võimaluse mitme seadme paigutamiseks samale suhtlussiinile. Jetson TX1, GSS valgustrükkplaat ning kõrguseandur on kõik sama andmesiini peal. Jetson TX1 loeb kõrguse andurilt kõrguse andmeid ning trükkplaadilt vedrustusandmeid. Jetson TX1 arvutist on võimalik ka seadistada LED valgustustugevust ning valgustuse sisselülitatud ajaperioodi pikkust.

Trükkplaadi esimese revisjoni tootmine toimus Brandner OÜ firmas. Tootmisfailid on genereeritud tudengivormeli liikme poolt.

6.1 Elektriskeemide disainimine ja simuleerimine

Elektriskeemide disainimine algas trükkplaadi kõikide funktsioonide üles kirjutamisega. Funktsioonid määravad, mida on vaja, et trükkplaadil hakkasid tööle disainitud elektriskeemid. Valgustustrükkplaadil on neli põhilisemat funktsiooni. Esiteks, valgustustrükkplaat peab valgustama maapinda kaamera pildi tegemise ajal. Maapinna valgustamine toimub ainult mõni mikrosekund. Kuna FEST20 on elektrivormel, siis trükkplaat peab olema väga efektiivse voolutarbega. Seetõttu tuleb valgustust lülitada ainult kaamera pildi tegemise ajal väga eredalt sisse ja välja. Teiseks, FEST20 vedrustusandurite analoogsignaali tuleb tõlkida analoog kujult digitaalseks, et see oleks kasutatav kiiruste arvutamises. Vedrustusandurite andmed on autos ainult analoog kujul, mis tähendab, et trükkplaat peab ise andmed digitaliseerima ning arvutile edasi saatma. Kolmandaks, pildi tegemise ajal kaamera väljundliidesest katkestussignaali kuulamine. Kaamera katkestussignaali põhjal teab, kuna trükkplaat peab valgustust lülitama. Valgustusaeg on programmeeritav mikrokontrolleri tarkvaras. Viimane funktsioon on valgustustrükkplaadil suhtlus Jetson TX1 arvutiga. Suhtlus toimub I2C andmesiini kaudu.

Kõik eelnevas lõigus mainitud funktsioonid tuli elektriskeemidega lahendada. Elektriskeemidele tuli luua erineva pingetasemega toiteallikad. Elektriskeemide simuleerimine toimus „Falstad Circuit Simulator“¹ keskkonnas. Toitemuunduri elektriskeemide lahendused disainiti vastavalt toitekiipide andmelehe näidete põhjal. Üks toiteskeem on kopeeritud eelmise aasta FEST19 auto ECU trükkplaadilt.

¹ <https://www.falstad.com/circuit/>

Toite elektriskeemid

Kokku oli vaja kasutada seitset erinevat pingemuundurit. Toitemuundurid on välja toodud joonisel 13.

Sisendpinge	Väljundpinge	Kommentaar
24V	15V	Andurite signaali puhverdamine
24V	12V	LED MOSFET paisu lülitamine
24V	5V	Välise DAC toide
24V	4.5V kuni 21.5V	LED valgustus
5V	3V3	Mikrokontrolleri toide
5V	3V3	MCU ADC võrdluspinge
5V	4.096V	DAC kiibile võrdluspinge

Joonis 13. GSS trükkplaadi toitepingete tabel

Trükkplaadi toiteskeemi eesmärk on tekitada pingestabilisaatoritega erinevaid pingetasemeid, mida komponendid trükkplaadi peal kasutavad. Pingestabilisaatorid jagunevad kaheks: lineaarstabilisaator ja impulss-stabilisaator. Nende kasutamisest sõltub, kui palju voolu komponendid kasutavad. Kui voolutarve on suur, siis on mõistlik kasutada impulss-stabilisaatorit, mis kasutab efektiivsemat pingelange meetodit ning protsessi käigus ei eritata väga palju soojust. Nende negatiivne külg on keerulisem elektriskeem. Näiteks on see kasutusel mikrokontrolleri ja LED valgustuse toiteskeemides. Lineaarstabilisaatorid on vähem efektiivsemad pingelange ehk tekitavad suurema voolu kasutamise juures rohkem sooja, kuid on väga lihtsa skeemiga. Seetõttu kasutan neid skeemides, mis tarbivad väga väikeses koguses voolu ning kuna lineaarstabilisaatorid on müravabamad, siis kasutan neid võrdluspingeallikatena mikrokontrolleri analoog-digitaalmuunduri ning välise digitaal-analoogmuunduri kiibi võrdluspingeallikatena.

Analoog elektriskeemid

Analoog elektriskeemid jagunesid kaheks. Esiteks, kaks auto esitelje vedrustus lineaaranduri signaali tuleb trükkplaati sisse lugeda. Kuna trükkplaadile tuleb andurilt ainult 0V ehk GND ja signaal, siis puhverdati signaali, et see oleks piisavalt tugev. Lineaarandurid töötavad 12V pingega ja puhverdamiseks on kõrgemat pinget vaja kui 12V. Auto 24V madalvooluahela pinget vähendati 15V pingeni, mida kasutati operatsiooni võimendil toiteks. Mikrokontrolleri viigule võib anda pinget vahemikus 0V

kuni 3.6V, seega tuleb teha pingejagur 15V pingelt madalamale pingele 3V3. Analoo elektriskeemi on sisse ehitatud ülepingekaitse, mida tehakse tavalise TVS diodiga. Samuti ahelas on madalpääsfilter, mis filtreerib kõrgema sagedusega müra. Madalpääsfilter on disainitud siluma pinget kõrgema sagedusega kui 1000Hz. Teiseks, mõõdetakse esimeses revisjon plaadil temperatuuri. Teisel revisjoni asendati see LED toitepinge mõõtmisega.

LED valgustus lülitamise elektriskeem

Kuna LED-valgus lülitub väga väikse ajaperioodi jooksul, siis see tähendab, et LED-lampidele lülitatakse korraga peale väga suur vool. LED valgustuse lülitamine toimub N-MOSFET lülitusega, mis peab kannatama suurt voolu. LED-lambid kannatavad väikse töötükli (ingl.k *duty cycle*) juures piikvooluks 2.5 kuni 3 amprit. Kuna tavaliselt sellise suure vooluga N-MOSFET lülitil on väga suure mahtuvusega pais, siis mikrokontrolleri 3V3 pingega laetakse seda väga kaua täis. See võib hakata piirama transistori sisse ja välja lülitamise kiirust. Selleks tuleb luua eraldi elektriskeem, mis lülitaks paisu 12V pingega. Kõrgem pinget aitab täita paisu kondensaatori mahtuvust kiiremini, mille järel sulgeb transistor LED valgustuse elektriahela. See tähendab, et mikrokontrolleriga tuleb kontrollida 12V peale lülitamist N-MOSFET lülitile ning seda saab teha MOSFET draiveriga. MOSFET draiveri sisendiks on loogika signaal, milleks sobib mikrokontrolleri viigult kontrollitav 3V3 pinget. Väljundiks on kiibile seadistav pinget, milleks on 12V. Mikrokontroller kontrollib MOSFET draiverit, mis omakorda lülitab kõrgema pingega N-MOSFET lülitit.

Elektriskeemi teeb keeruliseks asjaolu, et LED jadaahel tuleb lülitada väga väikse töötükliga. Väga keeruline on arvutada, kui palju kaamera vajab sensorile valgust, kui kaamera pildi tegemise aeg on mõni mikrosekund. Kaamera sensorit on lihtne üle- ja alavalgustada. Seetõttu peab olema võimalik valguse intensiivsuse muutmine sama töötükli korral. See tähendab, et LED ahelale tuleb rohkem pinget peale lülitada, mis lubab suuremat voolu LED-lampidest läbi.

LED valgustuse heleduse muutmine

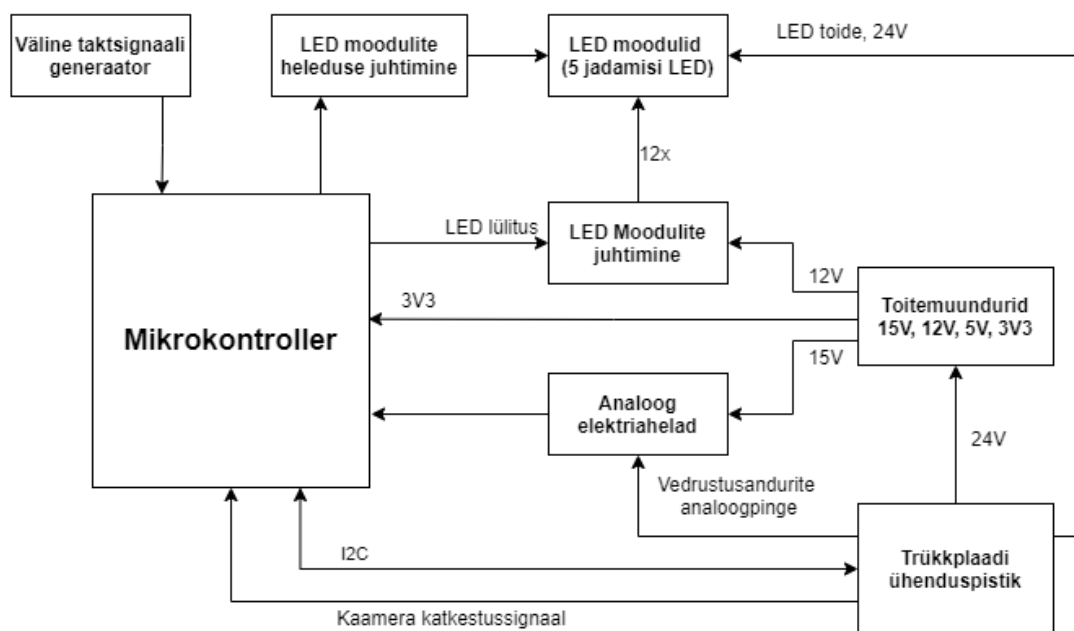
LED valgustugevuse muutmise eesmärk on vähendada ja suurendada valguse intensiivsust kaamera sensorile. Lihtne LED skeem siin ei tööta, kus jadamisi on ühendatud LED ja voolupiirav takisti, sest kui LED-lampi väga väikse töötükliga

lülitada, siis LED ei jõua põlema hakatagi. See tähendab, et pidevalt peab muutma voolupiiravat takistit. Seetõttu kasutan sama põhimõtet, mida kasutati ka eelmine aasta. LED valgustust juhitakse kondensaatorile laetud pingega ning pinge muutmisega muutub LED valgustugevus. Sellega peab olema väga ettevaatlik, liiga palju pinget muutes võib LED väga kiiresti läbi põleda. Mainitud põhimõte võimaldab väikse töötükli korral tõsta kondensaatorile laetud pingetaset, mis suurendab LED-lambi läbivat voolu ning paneb valgustuse eredamalt põlema. Eelmine aasta FEST19 projektis sai muuta valgustugevust muutes toiteskeemi tagasiside ahela reostaattakistit. See aasta plaaniti see disainida mikrokontrolleriga juhitavaks, mis tegi elektriskeemi oluliselt keerulisemaks.

Tuli mõelda skeem, mis tuleks paigutada LED toiteskeemi tagasiside ahelasse. Disainisin elektriskeemi, mis kasutab välist digitaal-analoogmuundurit ning takisteid, et muuta LED toiteskeemi tagasiside ahelat.

6.2 Elektriskeemide koostamine

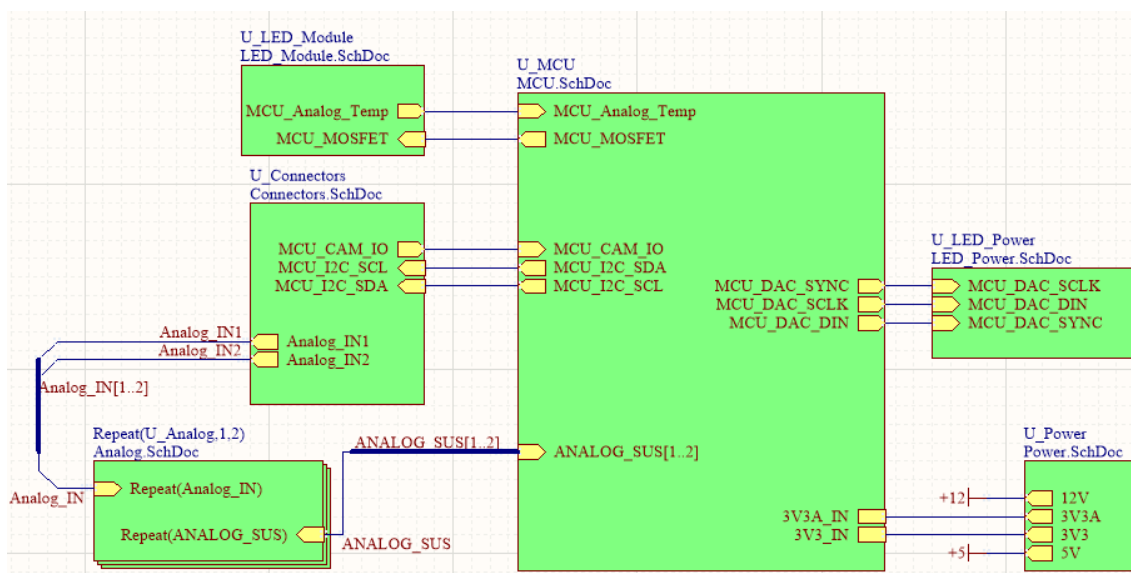
Elektriskeemide koostamise alameatükk toob ülevaate, kuidas ja millised elektriskeemid koostati. Selgitab täpsemalt, kuidas kõik elektriskeemid omavahel ühendatud on ja millised funktsioonid neil on. Joonisel 14 on ülevaade mikrokontrolleriga trükkplaadi alamosadest ning nende vahelised ühendid.



Joonis 14. Trükkplaadi plokk skeem

Ülemelektriskeem

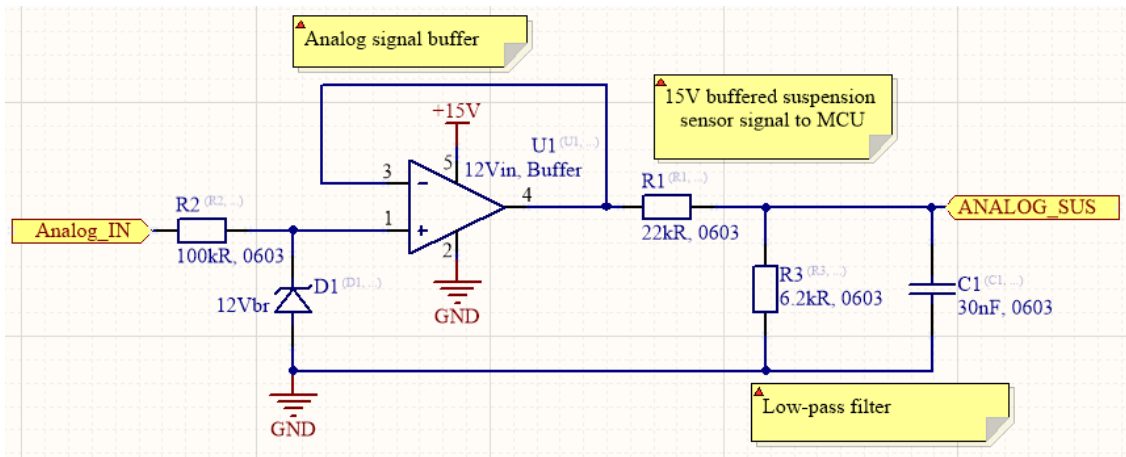
Ülemskeem kirjeldab kogu GSS valgustustrükkplaadi alamskeeme ning kuidas need omavahel seotud on. Trükkplaat koosneb järgmistest alamskeemidest: mikrokontroller, analoogahelad, ühenduspistik, LED moodul, LED toide ja toiteskeemid. See on illustreeritud joonisel 15.



Joonis 15. Revisjon 1 ülemskeem

Analoog elektriskeem

Trükkplaadil on kokku kaks analoog elektriahelat (joonisel 16), mis loevad vormeli esitelje vedrustus lineaarandurite analoogsignaali. Analoogetsendist, mis on ühendatud pistiku viigule, tulevale signaalile on seatud volupiirav 100k oomine takisti R2 ning TVS kaitse diod D1. Signaal on puhverdatud operatsiooni võimendiga, kus negatiivne sisend on ühendatud otse operatsiooni võimendi väljundisse. Puhverdamiseks kasutatakse 15V toiteallikat. Puhvri väljundist tulev 15V on liiga kõrge pinge, et MCU seda sisse loeks ning seetõttu moodustavad R1 ja R3 pingejaguri, mis tekitavad 3V3 pinge. Viimane analoogahela osa on madalpääsfilter, mis silub pinget kõrgematel sagedustel kui 1000Hz.



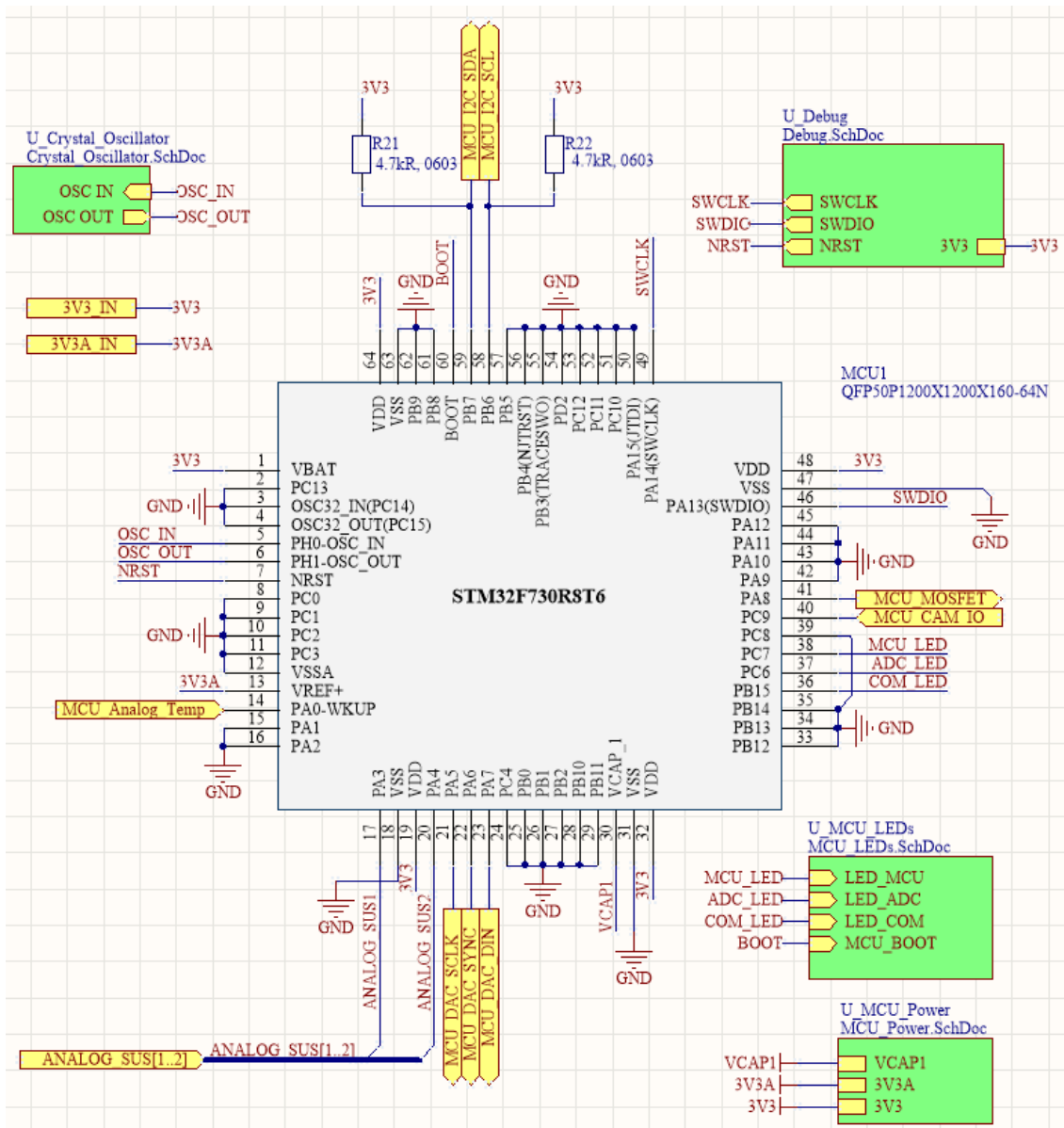
Joonis 16. Revisjon 1 analoog elektriskeem

Mikrokontrolleri elektriskeem

Eelnevates peatükkides on mainitud ja põhjendatud mikrokontrolleri rolli trükkplaadil.

Joonisel 17 on mikrokontrolleri STM32F730R8T6 viigud defineeritud järgmiselt:

- (PA8), LED valgustuse lülitamine
- (PA9), kaamera väljundliidesel katkestussignaali kuulamine
- (PA0), analoog temperatuur (revisjon 2 on eemaldatud),
- (PA3, PA4), vedrustusandurid
- (PB6, PB7), andmesiin I2C
- (PA13, PA14), programmi peale laadimine ja silumine
- (PH0, PH1), väline taktsignaali generaator
- (PA5, PA6, PA7), välise DAC kiibiga suhtlus
- (PC6, PC7, PC8), LED indikaatorid
- (NRST), mikrokontrolleri restart viik
- (VBAT, VDD), toiteviigud
- (VREF+, VSSA), analoog võrdluspingeallikas
- (BOOT), programmikoodi asukoha käivitamise viik



Joonis 17. Revisjon 1 mikrokontrolleri elektriskeem

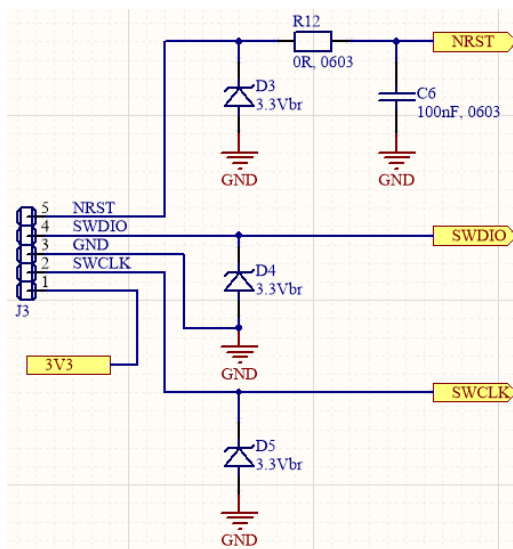
Mikrokontrollerile tarkvara peale laadimine ning programmikoodi silumine toimub SWD liidesega (joonisel 18), mis koosneb viiest juhtmest. Programmi peale laadimiseks on meil kasutusel STM32Discoveryboard¹, millel on eraldi ST-Link Debugger trükkplaadil. Sellel on SWD liides².

NRST viik on mikrokontrolleri restart signaal, sellele juhtmele on kaitse zeneri diod, mis hoiab konstantselt 3V3 pinget. Silumisahel on kopeeritud varasema tudengivormeli mikrokontrolleriga elektriskeemi trükkplaadilt ning seal oli kasutatud null oomist

¹ <https://www.st.com/en/evaluation-tools/stm32f4discovery.html>

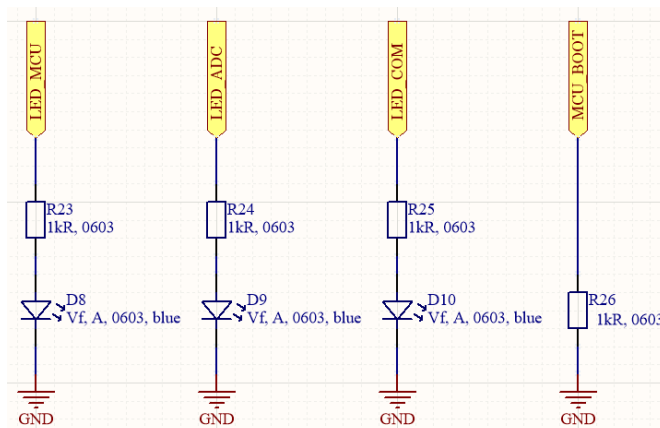
² <http://www.ti.com/lit/wp/spmy004/spmy004.pdf>

takistit, mis hoiab 0603 korpusega takistile vajadusel kohta. Igaks juhuks pandi R12 takisti skeemi (joonisel 18), kuid revisjon 2 eemaldati. SWDIO on andmesiin, mis töötab nii sisend- kui väljundliidesena. GND on 0V, mis on ST-Link Debuggeril ja GSS valgustrükkplaadil ühine. SWCLK on kellatakti signaal. Viimane viies viik on 3V3, mis annab aimu ST Link Debuggerile, millise toitepinge tasemega trükkplaadi mikrokontroller töötab.



Joonis 18. Revisjon 1 silumise elektriskeem

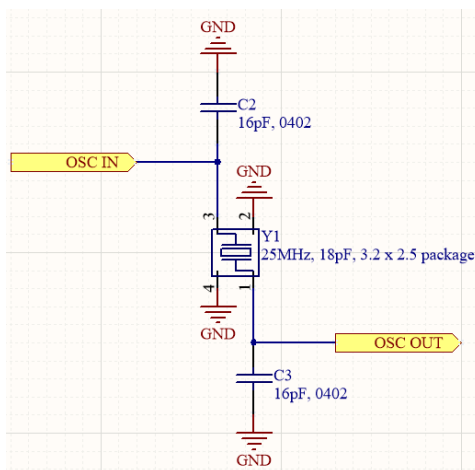
LED indikaatorid (joonisel 19) näitavad mikrokontrolleri erinevate programmifragmentide tööolekut. Kuna LED indikaatorite voolutarve on paar milliamprit, siis neid juhitakse otse mikrokontrolleri viigult. BOOT signaal on 0V ühendatud läbi 1k oomise takisti. See tähendab, et kui mikrokontroller tööle hakkab, siis programmikood asub Flash mälus kindlal mäluaadressil.



Joonis 19. Revisjon 1 mikrokontrolleri LED indikaator elektriskeem

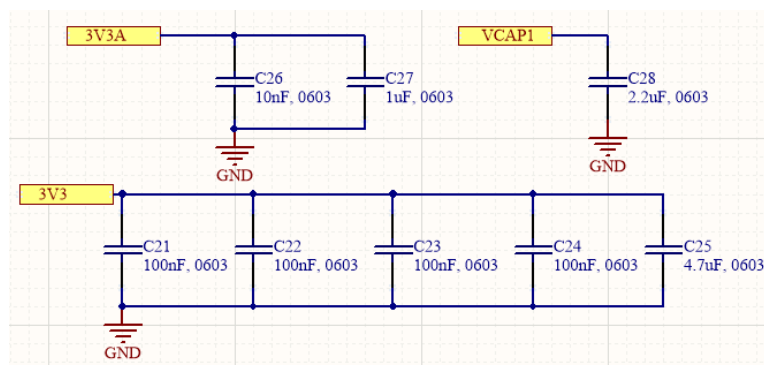
Taktsignaali generaatori elektriskeem (joonisel 20) on võetud eelmise aasta ühe mikrokontrolleriga trükkplaadi pealt. Ei ole mõistlik uut disainima hakata, kui vajalik

skeem on olemas ning on teada, et see on töökindel. Väline taktsignaali on 25MHz, mis on väga täpne. Sellist täpset taktsignaali suurendab mikrokontrolleri PLL¹ loogika, mis tõstab mikrokontrolleri maksimaalse taktsignaali 216Mhz.



Joonis 20. Revisjon 1 väline taktsignaali generaatori elektriskeem

Mikrokontrolleril on neljal erineval viigul vaja ühendada VSS ja VDD vastavalt GND ja 3V3 (joonisel 21). Igale voolupaarile on vaja ühendada möödaviigu kondensaator (ingli.k *bypass capacitor*), mis siluks mikrokontrolleri 3V3 pinget ning hoiaks seda ühtlasemalt 3V3 pinge juures. C21 kuni C24 on vaja paigutada iga voolupaari juurde ning C25 ühe voolupaari juurde paralleelselt lisaks. Mikrokontrolleri analoog võrdluspinge jaoks on vaja asetada C26 ja C27 kondensaatorid. VCAP1 signaal tuleb ühendada läbi C28 GND liinile.



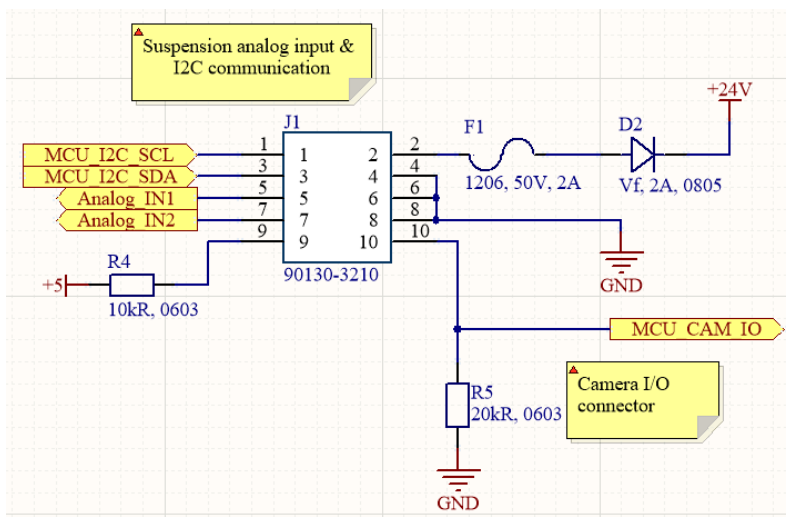
Joonis 21. Revisjon 1 mikrokontrolleri toite elektriskeem

¹ https://en.wikipedia.org/wiki/Phase-locked_loop

Ühenduspistiku elektriskeem

Ühenduspistikuna kasutan Molex Crid III, sest tudengivormeli elektroonika meeskond omab tööriista, millega saab sellele pistikule juhtme otsasid teha. Pistik on 2x5 konfiguratsiooniga ehk kaks rida ning mõlemas reas viis viiku.

Ühenduspistiku (joonisel 22) viikudele 1 ja 3 ühendatakse I2C andmesiin, mille kaudu LED valgustus trükkplaat suhtleb Jetson TX1 arvutiga. Vedrustus lineaarandurite signaali juhtmed tulevad analoog sisendiks viikudele 5 ja 7. Auto madalvoolupinge tuleb viigule 2, mis on 24V. Enne kui 24V trükkplaadil toitemuunduritele jõuab on liini kaitse ja diodiga kaitstud. Kaamera väljundliides kasutab 5V süsteemi, mis sulgeb elektriahela kui toimub tarkvara katkestussignaal. Toide peab tulema trükkplaadilt kaamerale, mis peab olema 5V. Trükkplaadi pealt antud 5V ei sobi otse mikrokontrolleri viigult sisse lugemiseks, seetõttu tuleb pingejaguriga 5V teha 3V3 pingeks. Pingejagur tehakse takistitega R4 ja R5. Seega võib mõelda, et kaamera ühendatakse ühenduspistiku viikude 9 ja 10 vahele ning kui toimub kaamera poolt katkestussignaal, siis kaamera lühistab pistikult viigud 9 ja 10 kokku. Revisjon 2 trükkplaadil tulevad väiksed muudatused ühenduspistiku viikude osas.

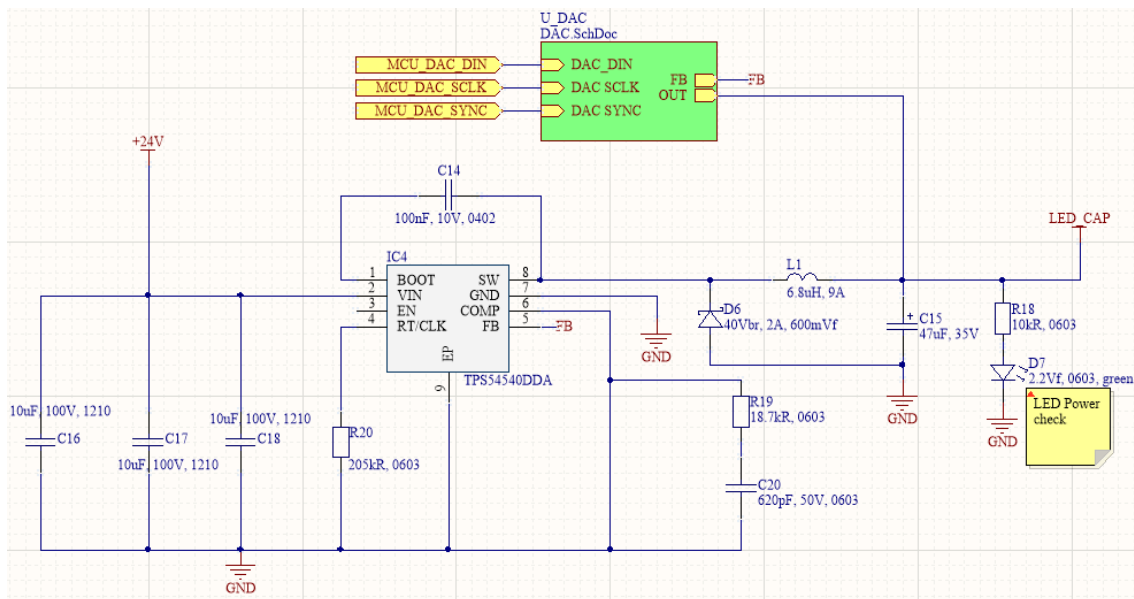


Joonis 22. Revisjon 1 ühenduspistiku elektriskeem

LED toite elektriskeem

LED valgustuse toiteks tuli luua paindlik toitemuundur, mis suudaks muuta 24V pingevahemikuks 4.5V kuni 21.5V. Täpselt need arvud ei olnudki olulised. Oluline oli, et LED mooduli saaks välja lülitada testimiseks nii, et LED-lampidest juhitakse läbi väga väike pinge, mis ei lülita LED-lampe sisse. 4.5V on piisav, et 5 jadamisi

ühendatud LED ei lülituks sisse. Kasutasin sama toitemuunduri kiipi, mis eelmine aasta kasutati IC4 (joonisel 23), aga toitemuunduri skeemi muutsin täielikult ümber. Eelmine aasta juhiti selle toitemuunduriga umbes 6-7V vahel kahe jadamisi LED-i väga kõrge voolupiigiga. See tähendab, et elektriskeem pidi olema efektiivne just selles pingevahemikus. Kuna selle aasta GSS valgustrükkplaat juhib korraga viit LED-i, siis pinge on kõvasti suurem. Seetõttu üritasin toitemuunduri väljundpinge disainida vastavalt eelmise aasta LED pinget arvesse võttes. Teadsin, et 3.25V lülitati LED-lambile peale, kui teda läbis 2,5A. Arvestasin, et juhin viit jadamisi LED-lampi ehk toitepinge võiks olla 16.25V. Eelmise aasta projekti tegija ütles, et võiks isegi kõrgema pingega LED juhtida, sest töötükkel on niivõrd väike, et see ei tohiks kahjustada LED-lampe. Seetõttu disainisin WEBENCH Power Designer¹ tööriistaga selle kiibi 19V juurde, mis tähendab, et toitemuundur võiks selle pinge juures kõige efektiivsem olla ning toita LED valgustust 2-3A voolupiigiga. See tähendab, et kui valgustus töötab auto peal ettenähtult ehk väga väikse töötükkli juures, siis 19V juures kasutatakse kõige suuremat voolu.



Joonis 23. Revisjon 1 LED toite elektriskeem

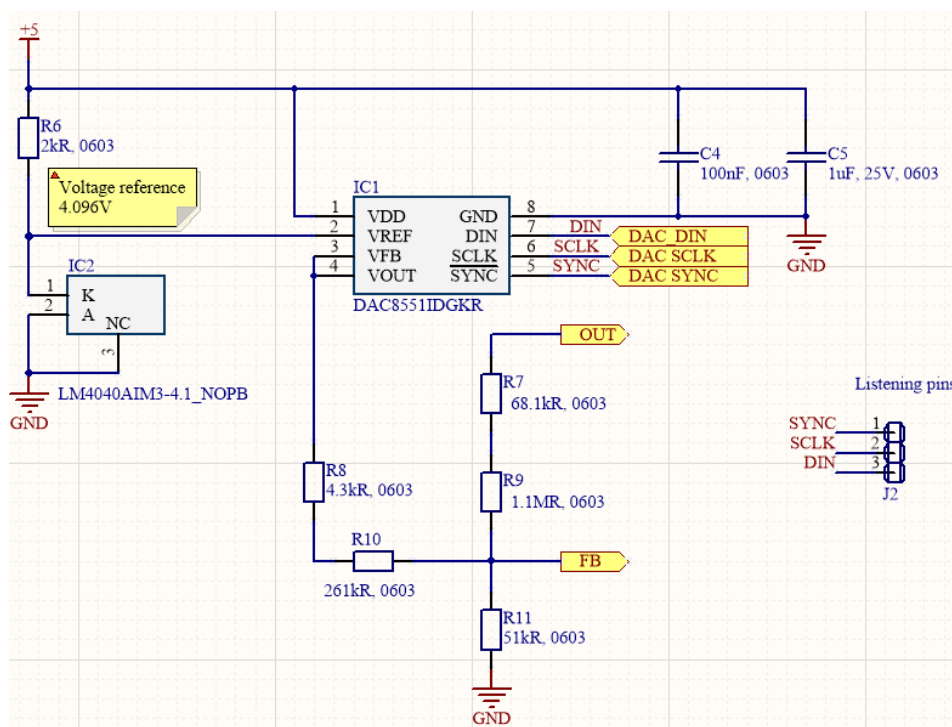
Pinge muutmine oli keeruline ülesanne. Eelmine aasta tehti seda toitemuunduri tagasiside pingejaguris ühe takisti väärtuse muutmisega.. See aasta taheti seda lahendada mikrokontrolleriga. Selle jaoks ehitasin väljundpinge ja toitemuunduri

¹ <http://www.ti.com/design-resources/design-tools-simulation/webench-power-designer.html>

tagasiside viigu (FB) vahele elektriskeemi, mille abil sai muuta toitemuunduri väljundpinget. Selle muutmine toimus välise digitaal-analoogmuunduriga.

Elektriskeem LED toitepinge muutmine ei ole väga keeruline põhimõtte poolest. Digitaal-analoogmuundur muudab digitaalse väärtuse analoogpingeks. IC1 on 16-bitine DAC¹ (joonisel 24), mis tähendab, et see digitaal-analoogmuundur suudab kiibile seadistatud võrdluspinge (viik 2 VREF joonisel 24) jagada 65 536 võrdseks osaks. Kiibile on seadistatud võrdluspingeks 4.096V, mis tähendab, et 16-bitise resolutsiooni korral saab mikrokontroller kontrollida DAC väljundpinget 62.5uV sammuga 0V kuni 4.096V pingevahemikus. Mikrokontrolleriga saab DAC programmeerida läbi SPI liidese, mis hõlmab endas kolme erinevat viiku. Esiteks, DIN, mis on andmebiti jaoks, siis teiseks SCLK, mis saadab DIN liinil olevad andmebitid DAC andmeregistrisse ning SYNC viik on kiibi programmeerimisluba. Kui SYNC viik on viidud 0V pinge juurde, siis DAC saab programmeerida, vastasel juhul ei juhtu midagi. Näiteks, kui DAC programmeeritakse arvuga 30000, siis valemi (8) järgi on vastus 1.875V.

$$\frac{4.096V}{2^{16}} \times 30000 = 1.875V \quad (8)$$



Joonis 24. Revisjon 1 LED toite juhtimine elektriskeem

¹ <http://www.ti.com/lit/ds/symlink/dac8551.pdf>

Nüüd vaadates elektriskeemi joonist 24, näeme, et DAC väljundpinge viikudel 3 ja 4 on ühendatud takistid. Kuna selle elektriskeemis oli oluline, et arvatud väärtused takistitel oleksid täpsed, siis on kasutatud kahte takistit jadamisi, et nende summana saaks täpsema väärtuse kokku. Seega R8 ja R10 on jadamisi ühendatud ning summeeritud ning R7 ja R9 on summana arvestatavad.

LED toiteskeemi toite muutmiseks vaatame takistite konfiguratsiooni (joonisel 24). Skeemil on kolm pingepunkti, mis on olulised. Üks on nimetusega OUT, teine FB ja kolmas on DAC väljundpinge (VFB ja VOUT). FB on toitemuunduri tagasiside viik, mis tähendab, et see on fikseeritud pinge. Toitemuunduri andmeleht¹ ütleb, et see on alati 0.8V. Seega DAC kiibiga muutes tema väljundpinget, siis peab OUT pinge muutuma. Seetõttu, kuna FB on fikseeritud pinge, siis DAC väljundiga saabki lineaarselt muuta OUT pinget. OUT viigu pingevahemik on seadistatav takistite (R7 kuni R10) väärtustega. Otsustati, et pingevahemik oleks 5V kuni 23V. Seega, kui DAC väljundpinget muuta pingevahemikus 0-4.096V, siis muutub ka OUT pingevahemik vastavalt pingevahemikus 5-23V ja pingevahemik on võimalik jaotada 65 535 osaks.

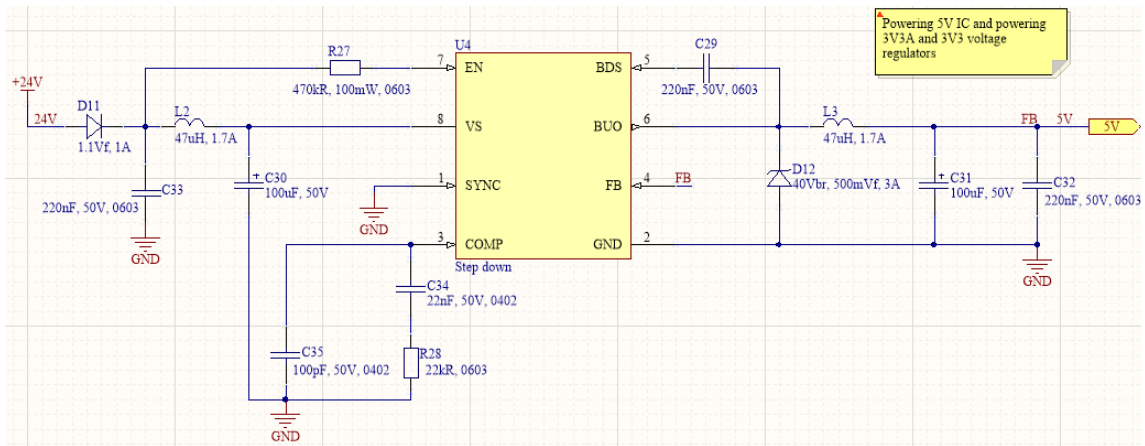
$$\frac{23V-5V}{65536} = 275\mu V \quad (9)$$

Väikseim pinge resolutsioonisamm on valemi (9) järgi 275uV, see on oluline, kuna viit jadamisi LED-lampi ei piirata voolupiirava takistiga, seega on vaja pinget tõsta väga väikse sammu kaupa, et LED-lambid läbi ei põleks. Sellise tagasiside meetodiga saab kontrollida mikrokontrolleri tarkvaraga toitemuunduri väljundpinget.

Toite elektriskeemid

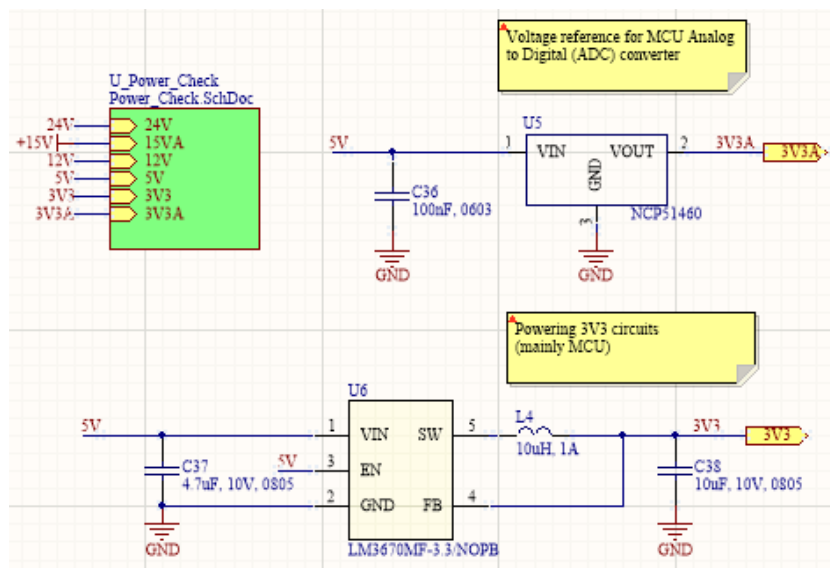
Toiteskeemi (joonisel 25) 5V on kasutuses mitmes kiibis ning sisendiks 3V3 ja 3V3A lineaar pingestabilisaator kiipidele. Samuti kasutan DAC võrdluspinge 4.096V tekitamiseks ning temperatuurianduri kiibi toiteks 5V. Kõige suurem voolutarbimine on läbi 3V3 impulss-pingestabilisaatori, mis annab toite mikrokontrollerile. DAC võrdluspinge ja 3V3A (analoogpinge) tarbivad olematus koguses voolu ning ongi mõeldud ainult võrdluspinge loomiseks.

¹ <http://www.ti.com/lit/ds/symlink/tps54540.pdf>, lk 14



Joonis 25. Revisjon 1 toitemuundur 5V elektriskeem

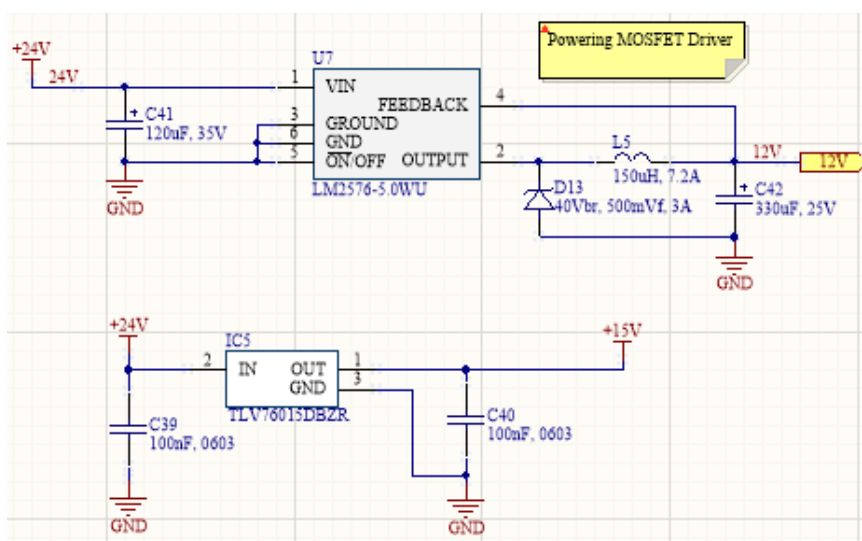
Toiteskeemi (joonisel 26) 3V3 kasutan mikrokontrolleri toiteks. Kuna voolutarbimine on võrdlemisi suur (ca 200mA), siis kasutan impulss-pingestabilisaatorit, mis ei erita suurema voolutarbe juures palju soojust. Mikrokontrolleri analoog võrdluspinge tekitamiseks kasutan LDO, kus 5V tehakse 3V3.



Joonis 26. Revisjon 1 toitemuundurid 3V3 ja 3V3A elektriskeemid

Toiteskeem 15V (joonisel 27) on vajalik ainult operatsioonivõimendi toiteks. Vedrustusandurite signaalipinge on kuni 12V ning seetõttu operatsioonivõimendi, mida kasutatakse skeemis puhvrina, peab saama kõrgema toitepinge kui operatsioonivõimendi sisendid. Selletõttu kasutasin lineaarset pingestabilisaatorit, mis langetab auto 24V pinge 15V peale. Linearsel pingestabilisaatoril on stabiilne pingeväljund, mis ei ole mürane ning seetõttu sobib suurepäraselt analoogsignaalidega. Kasutasin LDO-d, sest operatsioonivõimendi tarbib väga vähe voolu ning seetõttu LDO ei tekita väga palju võimsust, mis võiks tekitada soojust.

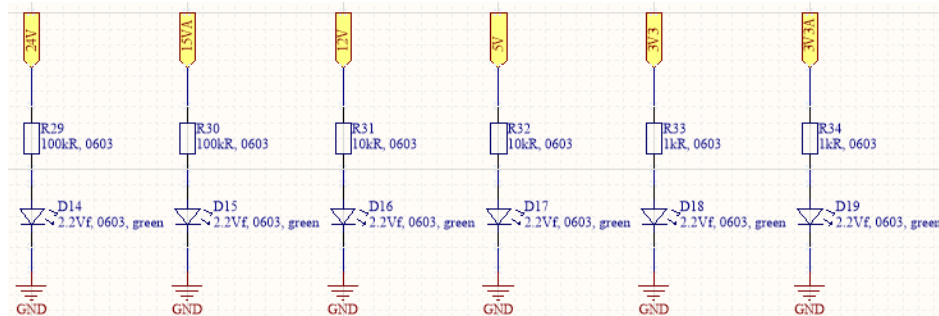
Toiteskeem 12V (joonisel 27) on mõeldud LED-lampide MOSFET draiveri lülitamiseks, mis lubavad suurt voolu MOSFET paisule. LM2576 kiip¹ on impulss-pingestabilisaator, mis suudab anda väljundvooluks 3A. Nii suurt väljundvoolu on vaja sellepärast, et võimalikult kiiresti laadida 12 LED moodulil MOSFET paisukondensaator täieliku juhtimispingeni, mis on 12V.



Joonis 27. Revisjon 1 toitemuundurid 15V ja 12V elektriskeemid

¹ <http://www.ti.com/lit/ds/symlink/lm2576.pdf?&ts=1589624639623>

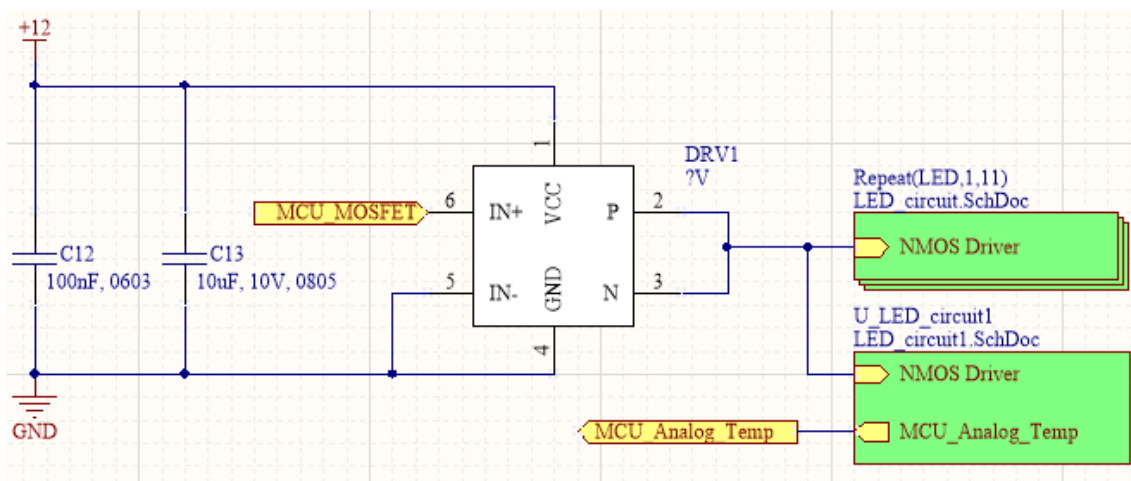
Toitekontroll (joonisel 28) on visuaalseks indikaatoriks trükkplaadi peal, mis annab kiire tagasiside, kas kõik toiteahelate skeemid töötavad ning kiibid annavad välja väljundpinget.



Joonis 28. Revisjon 1 toitemuundurite LED indikaatorid elektriskeemid

LED MOSFET draiver elektriskeem

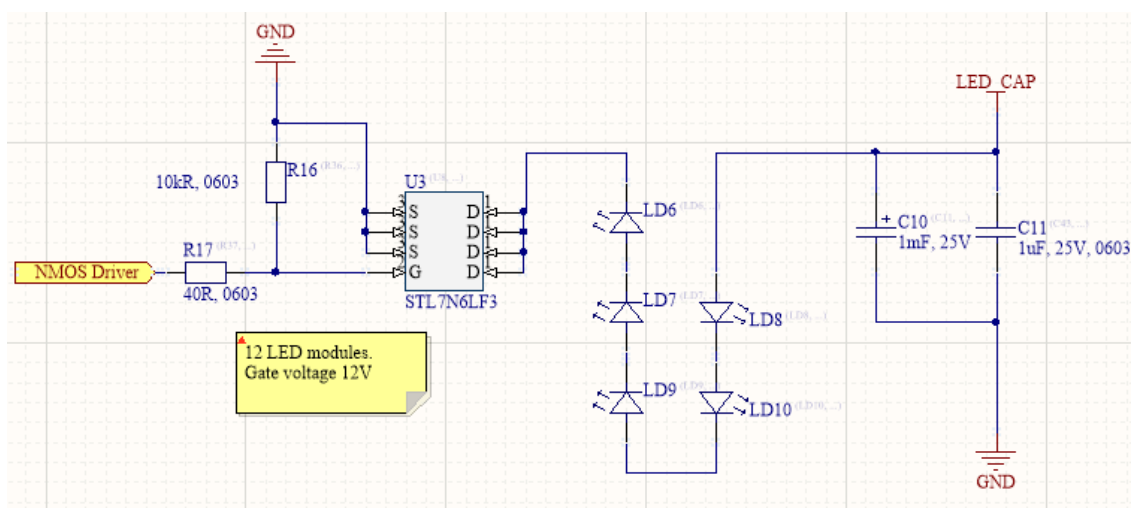
LED moodul juhib viit jadamisi ühendatud LED-lampi (joonisel 30), mis on kõik juhitavad ühe N-MOSFET lülitusega. Kõiki LED moodulite N-MOSFET lülitusi juhitakse ühe MOSFET draiveriga (joonisel 29). Kokku on 12 LED moodulit. MOSFET draiver lubab juhtida kõrgemat pinget N-MOSFET paisukondensaatoritele, kui näiteks mikrokontrolleri väljundpinge 3V3 võimaldaks. Kuna N-MOSFET lülitused juhivad suurt voolu LED ahelasse, siis on oluline, et paisukondensaatorit laaditakse kõrgema pingeni, näiteks 12V, kus lülitus avatakse täielikult ning suur vool saab läbi N-MOSFET lülituse liikuda. Ühe LED moodulite küljes on temperatuuri andur, mis mõõdab GND raja peal olevat temperatuuri. Kuna ma ei kasuta LED-lampide juhtimiseks voolupiiravaid takisteid, vaid kondensaatorile laetud pinget, siis võib LED-lampide ahel väga kiiresti kuumaks minna. LED-lampidele ei meeldi suur kuumus, sest siis nad hakkavad kaotama oma eredust ning lõpuks nad enam ei tööta. Kuigi temperatuuri mõõtmine ei ole tegelikult väga kiireloomuline protsess, siis temperatuuriandur on lisatud skeemi rohkem monitooringu jaoks. Revisjon 2 eemaldas selle.



Joonis 29. Revisjon 1 LED moodulite juhtimine elektriskeem

LED moodul elektriskeem

LED-valgus elektriskeem (joonisel 30) koosneb viiest jadamisi ühendatud LED-lambist, mis saavad toite 1mF kondensaatorilt. N-MOSFET on valitud, et ta kannataks suurt voolu ning oleks võimalikult väikse paisukondensaatori mahtuvusega, et pais laetaks kiiresti täis. See tähendab, et N-MOSFET juhiks LED ahelat võimalikult kiiresti. Takisti R16 on mõeldud paisu tühjaks laadimiseks ehk kasutuses *pull-down* takistina. MOSFET draiveri poolt tuleva voolupiigi limiteerimiseks kasutan R17 takistit, mis piirab voolupiigi 3 amprini.



Joonis 30. Revisjon 1 LED moodul elektriskeem

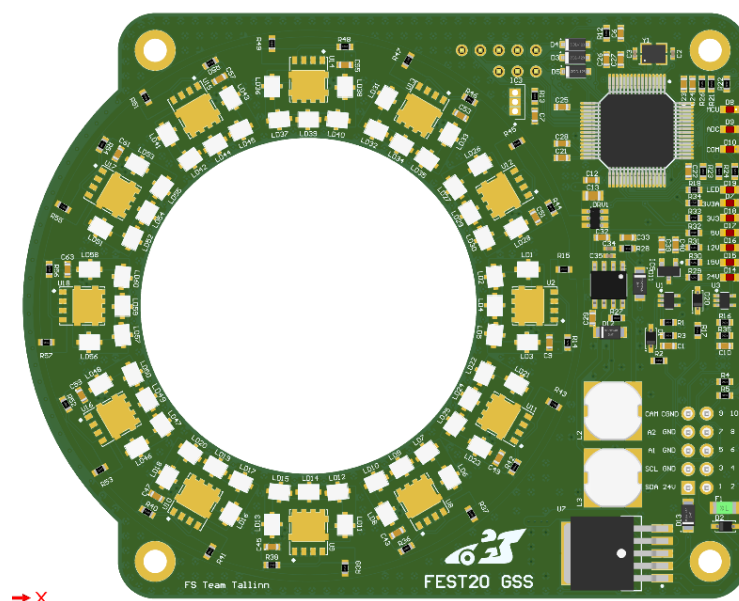
6.3 Trükkplaadi joonised

Esimese revisjoni trükkplaadi joonistel oli raske luua head komponentide paigutust, sest ei olnud varasemalt sellist plaati, mille pealt oleks hea vaadata. Kuna trükkplaat tuli teha

võimalikult väike, siis tuli väga hoolikalt komponendid ära paigutada trükkplaadile. Paigutamine tuli teha loogiliselt ning elektriskeemide kaupa. See tähendab, et ühe elektriskeemi detailid tuli koos paigutada ning võimalikult lähedale üksteisele. Trükkplaat koosneb mitmest erinevast elektriskeemist. Esiteks LED moodulid ja nende N-MOSFET lülitused, seejärel mikrokontroller ja kõik mikrokontrolleriga seotud toitekondensaatorid ja LED indikaatorid. Järgmiseks, kõik toiteskeemid ning viimaseks analoog elektriskeemid. Esialgselt plaanisin paigutada nii palju komponente ülemisele kihile kui võimalik ning kõik ülejäänud lähevad kõige alumisele kihile. Kuna elektriskeemides oli väga palju suuri kondensaatoreid, siis arvestasingi kohe alguses, et need tuleb paigutada trükkplaadil alumisele kihile. Samamoodi ühenduspistik on alumisel kihil. Trükkplaat on nelja kihiline. Trükkplaadil kasutatavad komponendid on välja toodud lõputöö lõpus (vt lisa 1).

Ülemine kiht

Kõige esimeseks tuli paika panna LED valgustrükkplaadi kuju ning dimensioonid (joonisel 31), kuhu hakkavad komponendid paigutama. Järgmiseks tuli välja mõelda, kui suure ava trükkplaadile teen, et kaamera sealt läbi mahub. Kui dimensioonid ja trükkplaadi keskel ava mõõtmed ja asukoht paigas, siis alustasin ühe LED mooduli paigutusega. Proovisin mitu erinevat viisi, kuidas oleks kõige mõistlikum LED-lampe paigutada. Joonisel 31 on näha, millise lõpliku LED mooduli konfiguratsiooni kasuks otsustati. LED moodulid paigutati ümber keskmise ava.



Joonis 31. Revisjon 1 trükkplaadi ülemine kiht

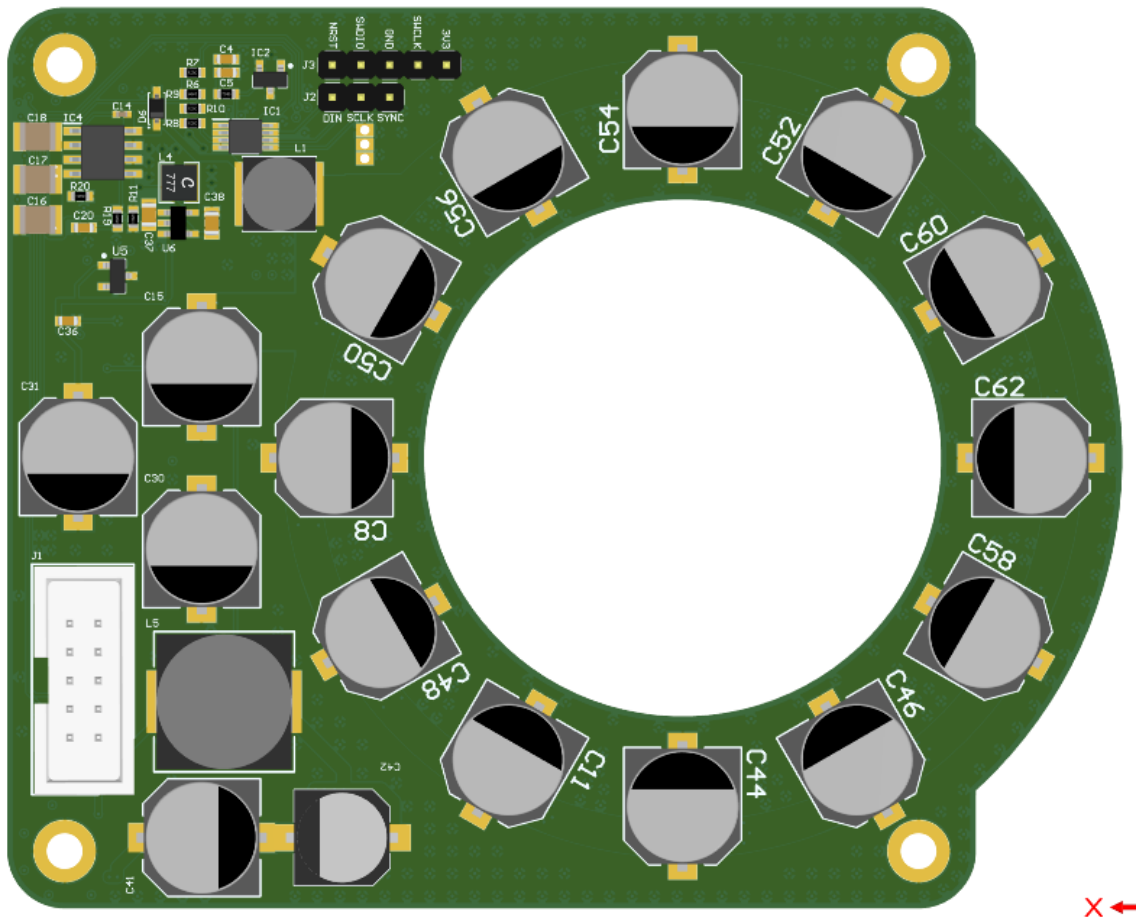
Pärast LED moodulite paigutuse konfiguratsiooni väljamõtlemist ning trükkplaadi peale paigutamist alustasin järgmise suure komponendi trükkplaadi peale paigutamise, milleks on mikrokontroller. Mikrokontrolleri elektriskeemil on väga palju erinevaid väiksemaid komponente, mille puhul on oluline, et need paigutuksid väga lähedal mikrokontrollerile. Näiteks väline taktisignaali generaator, millel on oluline, et vaserajad oleksid võimalikult lühikesed trükkplaadi peal. Teiseks, kõik möödaviigu kondensaatorid tuli paigutada nii lähedale trükkplaadile kui võimalik. Samuti mikrokontrolleri elektriskeemile lähedale on paigutatud programmi peale laadimise ja silumiseks viie viiguga ühenduspistik. Mikrokontrolleri lähedale paigutasin esimeses revisjonis ka temperatuuri anduri ning MOSFET draiveri.

Kuna trükkplaadi peal on väga vähe ruumi, siis toitemuundurite komponendid asuvad vabadel kohtadel, mis trükkplaadil on. Üritasin need paigutada võimalikult loogiliselt ning lähedale just nendele kiipidele, mis neid toiteväljundeid vajavad. Ülemisel kihil on paigutatud kaks toitemuundurit – 5V ja 12V.

Viimane elektriskeemide grupp on analoog elektriskeemid. Need on paigutatud mikrokontrolleri elektriskeemide ja ühenduspistiku viikude vahele. Ühenduspistikust tuleb sisendina analoogsignaali ja pärast puhverdamist liigub signaal mikrokontrollerisse.

Alumine kiht

Alumisele kihile (joonisel 32) paigutasin kõik kõrgema korpusega komponendid ning LED moodulite kondensaatorid, mis on suurte korpustega. LED mooduli kondensaatorid peavad olema väga lähedal LED moodulitele, sest need on kiired vooluallikad. Samuti paigutasin ühenduspistiku alumisele kihile. Kõik väiksema korpusega komponendid, mis ei mahtunud ülemisele kihile, tõin alumisele kihile. Väiksema korpusega komponendid olid LED toitemuundur, DAC ja 3V3 toitemuunduri elektriühendused. Lisaks ühenduspistikule on trükkplaadi alumisel kihil programmeerimiseks SWD ühenduspistik.



Joonis 32. Revisjon 1 trükkplaadi alumine kiht

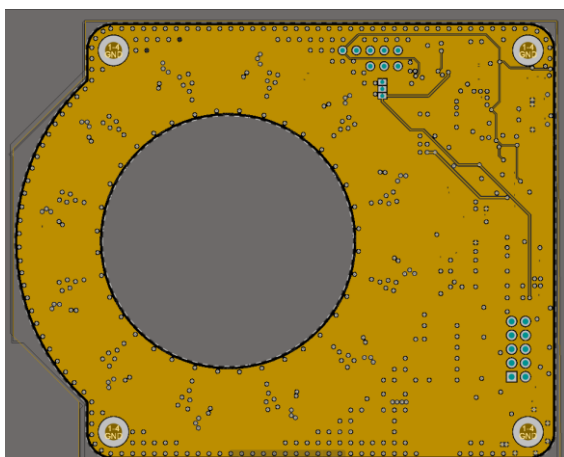
Alumise kihi komponentide paigutamist alustasin LED moodulite kondensaatorite paigutamisega trükkplaadi keskmise ava ümber. See oli lihtne, sest eelnevalt olin paika pannud ülemisel kihil LED moodulite asukohad. Järgmiseks paigutasin ära ühenduspistiku plaadi peale. Pärast ühenduspistiku paigutamist oli trükkplaadi peal väga vähe ruumi, et toitemuundurite sisend- ja väljundkondensaatorite paigutust teha võimalikult efektiivselt. Suurte induktorite ja kondensaatorite paigutus tuli teha vastavalt trükkplaadil järele jäänud ruumile. Mis ei tulnud üldse nii efektiivne, kui oleksin tahtnud. Kuid, väga suur aeg oli juba paigutamisele peale pandud, siis esimese revisjoni kohta oli rahul. Teadsin, et kindlasti tuleb teine revisjon, kus saab teha parendusi skeemidele ning kindlasti proovin parandada trükkplaadi jooniseid ja komponentide paigutusi.

Signaalikihid

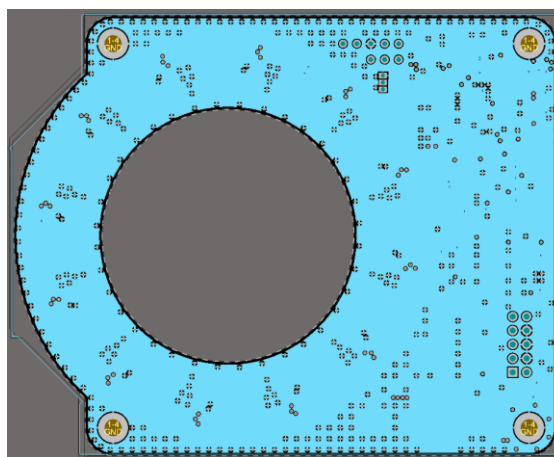
LED valgustrükkplaat on nelja kihiline. See tähendab, et trükkplaat koosneb neljast erinevast vasekihist, kuhu saab joonistada radasid. Kaks sisemist signaalikihti on

kasutusel tegelikult toitekihtidena. Üks kiht on tehtud 24V kihiks ning teine 0V ehk GND kihiks.

Joonisel 33 on signaalikihile valatud 24V toitepinge polügoon. Esimeses revisjonis on samuti paigutatud sellele kihile mõned teised signaalid, mis ei mahtunud alumisele ja ülemisele kihile ära. Joonisel 34 on valatud 0V polügoon. Terve kiht ühendab trükkplaadil 0V läbiviike. See on oluline, sest siis on 0V pinge kõikidele komponentidele võimalikult lähedal. Teises revisjonis signaalikihid muutuvad natukene.



Joonis 33. Revisjon 1 24V trükkplaadi kiht



Joonis 34. Revisjon 1 0V trükkplaadi kiht

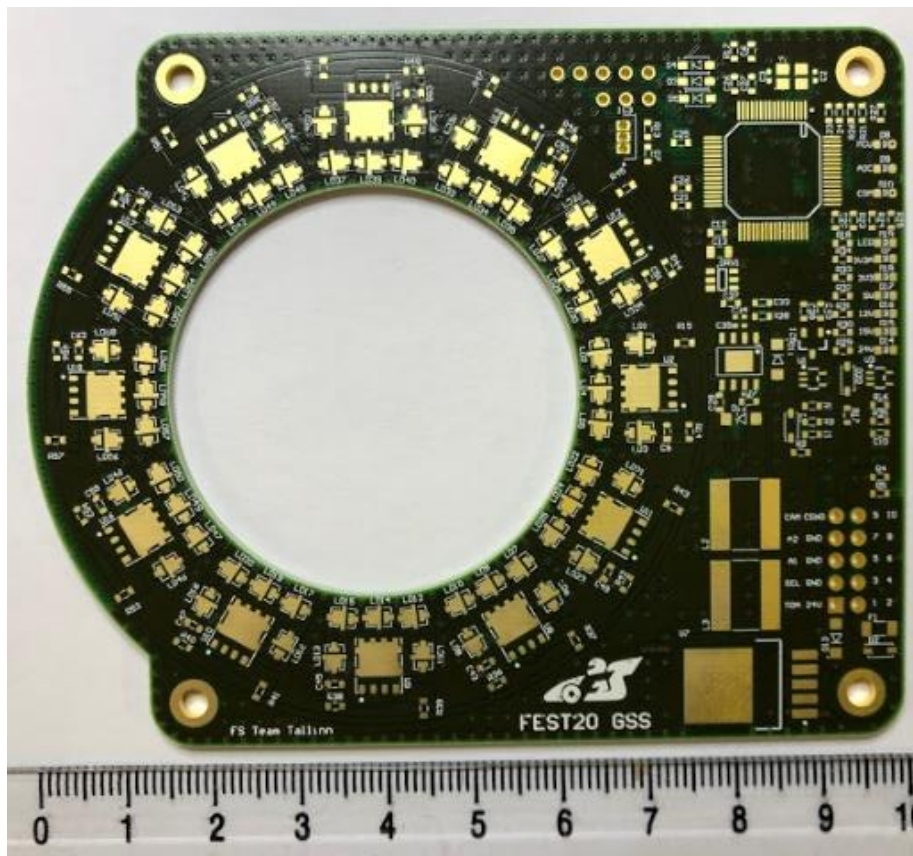
6.4 Testimine

Käesolevas peatükis selgitan, kuidas testisin trükkplaadi funktsioone. Samuti selgitan, millised vead tulid välja ning kuidas neid lahendasin.

Funktsioonide testimine

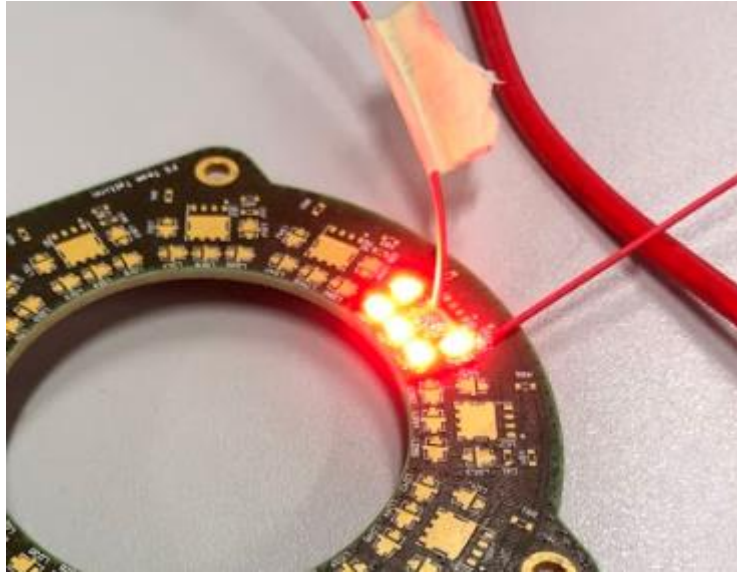
Testimine algas, kui revisjon 1 trükkplaat saabus tootmisest (joonisel 35). Esimene skeem, mis peale jootsin trükkplaadile oli toiteahelad. Alustasin toitemuundurist, mis muundab 24V pingest 5V. Kui olin valideerinud, et 5V töötab, siis alustasin teiste toitemuunduritega. Sellise meetodiga, kus iga toitemuunduri töötamise jootmise järel valideerisin, jootsin ära kõik toitemuundurid. Korraga jootes oleks vea leidmisel olnud väga raske öelda, milline elektriskeem seda põhjustab ning väga palju aega oleks kulunud komponentide maha ja uuesti peale jootmisele. Koos toitemuunduritega jootsin peale LED indikaatorid, mis näitasid, kas toitemuundurid töötavad. See visualiseeris, kas toitemuunduri väljundis on pinge. Järgmiseks jootsin peale analoog elektriskeemid

ning valideerisin välise toiteploki, kas puhverdatud signaalid töötavad ning pingejagur teeb 15V pingest 3V3 mikrokontrollerile.



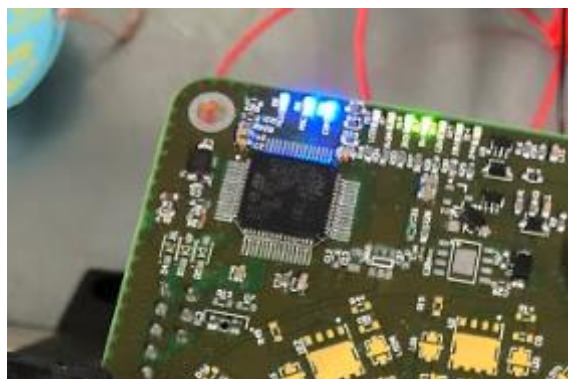
Joonis 35. Trükkplaadi ülemine kiht ilma komponentideta

Kui analoog elektriahelad ja toitemuundurid olid paigas ja töötasid, siis jootsin plaadi külge ühe LED mooduli (joonisel 36) ning testisin seda välise toiteploki. Proovisin leida pingelanguse, kus LED-lambid hakkavad üldse valgust näitama. Viiel jadamisi ühendatud LED-lambil on pingelang 7.2V, see tähendab, et sellest pingest edasi hakkavad LED-lambid tööle.



Joonis 36. LED mooduli testimine

Kui LED moodul oli peale joodetud, siis hakkasin jootma mikrokontrolleriga seotud elektriskeeme. Kõigepealt jootsin kiibi ning seejärel kõik toiteskeemid, et mikrokontroller hakkaks tööle. Plaadi peal on väga raske aru saada, kas mikrokontroller töötab, sest LED indikaatorid tuleb tarkvaras enne ära programmeerida, et need tööle hakkaksid. Selletõttu ainuke viis teada, kas mikrokontroller hakkas tööle või mitte, on joota plaadi peale ka programmi peale laadimiseks ja silumiseks mõeldud SWD liides. Kui proovisin STM32CubeIDE¹ trükkplaadiga ühendada ning algset koodi peale laadida, siis IDE andis positiive vastuse, et programmi kood on edukalt mikrokontrolleri peale laetud. Algses programmifragmendis programmeerisin lihtsalt sinised LED indikaatorid sisse lülitama, et valideerida, et programmikood ka töötab (joonis 37).

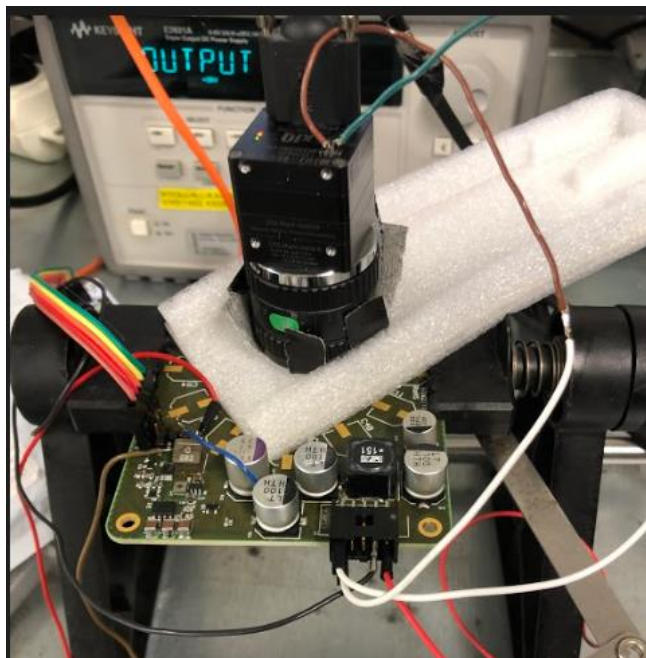


Joonis 37. Mikrokontrolleri LED indikaatorid testimine

¹ <https://www.st.com/en/development-tools/stm32cubeide.html>

Välise taktisignaali generaatori töötamise valideerimises on mõistlik seadistada taimer ning panna LED indikaatorid mingi sagedusega lülitama. Ostsilloskoobiga saab täpselt mõõta, kui pikk on LED indikaatorite periood. Kui taimer seadistus vastab ostsilloskoobi pealt loetud andmetele, siis on kõik töökorras.

Viimaseks testisin ka kaamera katkestussignaali trükkplaadi LED lülituse. Seadsin mikrokontrolleriga LED lülitusajaks $3.5\mu\text{s}$ ja ühendasin kaamera väljundliidese juhtmed trükkplaadi külge. (Joonisel 38) kaamerast tulev pruun ja sinine juhe on vajalikud katkestussignaali püüdmiseks. Testi käigus tuli välja, et $3.5\mu\text{s}$ LED-lampide sisse lülitamisel oli kaamera pilt natukene üle helendatud, mis tähendab, et trükkplaadil on veel võimekust varuks. Kuna testi tehti ühe LED mooduliga, siis peaks trükkplaat kindlasti olema võimeline piisavalt kaamera pilti helendama ka teiste parameetritega.



Joonis 38. Kaamera katkestussignaali testimine trükkplaadiga

Leitud vead, komplikatsioonid ja lahendused

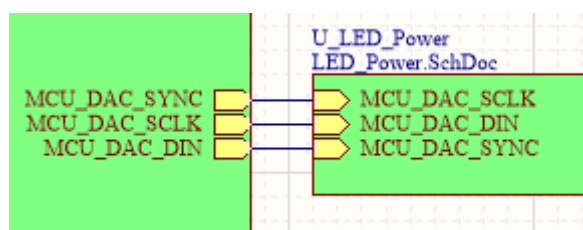
Esimese revisjon trükkplaadi funktsioonide testimisel tuli välja mitu viga, mis olid skeemide ja trükkplaadi jooniste disainis. Selgitan järgmises lõigus täpsemalt, mis vead tekkisid ning kuidas ma need ajutiselt parandasin, et testida kas trükkplaadil on veel midagi katki või puudu. Testimine ei läinud ka ilma komplikatsioonideta, mis tulid autori enda disainimisvigadest. Nendest lähemalt järgmistes lõikudes.

Esimesed komplikatsioonid tekkisid, kui üritasin joota peale 5V toitemuundurit. Kui 5V toitemuunduri peale jootsin, siis toitemuunduri väljund pinget oli 6.2V ning kui umbes

100mA koormust taha ühendada, siis väljundpinge langes 1.6V peale. Üritasin välja vahetada kiipi, kuid midagi ei aidanud. Päris palju aega hiljem avastasin, et olin unustanud väljundisse peale joota diodi. See tähendas, et kõik kiibid, mis peale jootsin, läksid kohe katki ja näitasidki mulle väga valesid pingeid ning koormust ei kannatanud üldse. Kui diodi peale jootsin ning uue kiibi peale jootsin, siis toitemuundur andis väga täpse 5.02V väljundpingeks.

Teine viga, mis välja tuli, oli 3V3A elektri ahelas, mis andis välja natukene üle 2V. See 3V3A oli mõeldud mikrokontrolleri analoogahela võrdluspingeallikaks. Kuid olin disaininud LED indikaatori sellesse ahelasse. See 3V3A toiteväljund ei ole mõeldud tarbima üldse voolu ning seetõttu langeski pinge, kui trükkplaadi peal oli talle järgi ühendatud LED indikaator. Järgmises revisjonis parandasin selle vea.

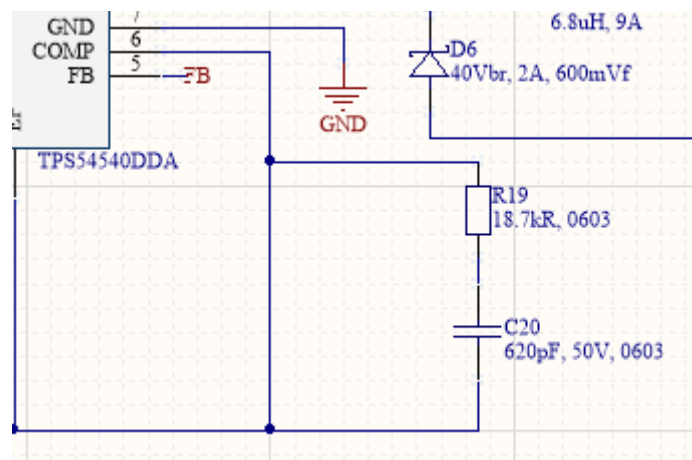
Järgmine viga disainis oli LED toite juures. LED toiteväljund andis välja 1.9V. LED toite elektri ahelat oli väga keeruline testida. Keeruliseks tegi see, et LED toite tagasiside ahelaks oli enda disainitud välise DAC kiibiga. Kuna ei olnud seda ära testinud, siis ei teadnud, kas asi võib olla selles. Seega alustasin LED toiteahela testimist DAC kiibist. See tähendas, et üritasin DAC programmeerida, et saada DAC väljundisse 0V kuni 4.096V väljundpinget. Avastasin, et SPI liin ei tööta, sest juhtmed olid omavahel valesti ühendatud. Kui vaatasin oma ülemskeemi, siis märkasin kohe viga. Joonisel 39 on näha, et olen ülemskeemis kahe skeemi omavahelised ühendused kogemata valesti kokku ühendanud. Seetõttu testimisel ei saanud kasutada SPI mikrokontrolleri riistvaralist tuge, vaid programmeerisin GPIO viikudena DAC programmeerimisviigud. Siis hakkas DAC tööle ning väljundpingesse tekkisid vastavalt 0V ja 4.096V. Samuti kõik töötas, kui 16-bitne DAC programmeerida vähima ja suurima arvuga, mis on 0 ja 65 535.



Joonis 39. Revisjon 1 välise DAC ühendus viga ülemskeemis

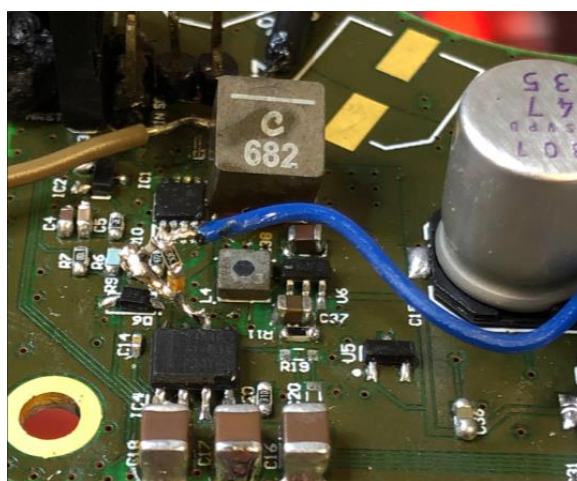
Kui DAC oli testitud, siis LED toitemuundur andis ikkagi 1.9V väljundisse. Vaatasin skeemi üle ning avastasin, et olin teinud toitemuunduri kiibi COMP viigu elektriskeemis vea. Joonisel 40 on näha, et olen ühendanud LED toitemuunduri COMP

otse GND liinile. Seal tegelikult oli enne vahel väga väikse mahtuvusega kondensaator, kuid olin kogemata ära kustutanud.



Joonis 40. Revisjon 1 LED toite vale COMP viigu disain

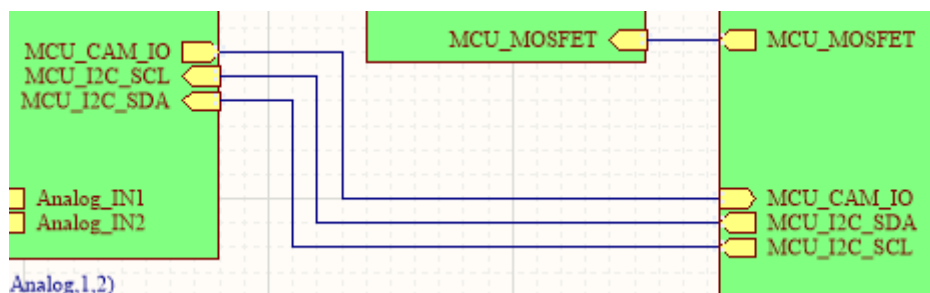
Selle vea parandamiseks üritasin plaadi peal tõsta natukene COMP viigu jalga korpusel üles ning joota sinna külge R19 ja C20 (joonisel 40) ning ühendada see GND liinile (joonisel 41). Lisasin veel varem ära kustutatud kondensaatori paralleelselt nende kahe R19 ja C20 komponendiga. Pärast seda hakkas toitemuundur tööle ning andis väljundpingeks normaalsed väärtused. Seejärel testisin mikrokontrolleriga välise DAC programmeerimise ning LED toitemuunduri väljundpinge muutus täpselt disainitud elektriskeemile. Kuna välise DAC väljundis takisti väärtused ei olnud nii täpsed, siis disainitud 5V kuni 23V pingevahemik oli reaalsuses hoopis vahemikus 4.5V kuni 21.5V, mis ei olnud probleem.



Joonis 41. Revisjon 1 LED toite COMP viigu parendus

Kui väljundpinge oli käes, siis hakkasin testimas koormust ja siis tekkis järgmine probleem. Toitemuunduri korpus läks väga kuumaks. Kui COMP viigule tehtud

juhtmetele vastu läksin (joonisel 41), siis muutus väga kergelt väljundpinge. Tuli välja, et COMP viik on väga tundlik ning selline parendus plaadil ei olnud piisav, et toitemuundur tööle hakkaks. Arvan, et juhtmemahtuvus ja muud elektrilised tegurid võisid seda signaali mõjutada. Probleem lahenes alles siis, kui plaadi peal üritasin vea parandada. Lõikasin vasest juhtmerajad läbi plaadil, mis signaali 0V lühistas ning sain komponendid uuesti plaadile joota. Põhjus, miks kiip väga kuumaks läks oli see, et kiip tekitas induktoril väga kõrged voolupiigid. Jootsin 100mOhm takisti induktori ja väljundpinge kondensaatori vahele ja mõõtsin ostilloskoopiga takistil pinge langu. Sain teada, et voolupiigid oli kuni 6.5A. Probleem on selles, et ikkagi voolupiigid tekivad induktoril. Kuid 19V kuni 21V juures ei olnud voolupiigid suured ning toitemuunduri korpus ei läinud ka kuumaks. GSS valgustus hakkabki ise 19-21V juures LED lampe lülitama. Seega see ei ole probleem, sest toitemuundurit ei kasutata väiksema väljundpinge juures. Lihtsalt tähelepanekuks järgmisele projekti tegijale ütlesin, et toitemuundur muutub järjest ebaefektiivsemaks 19V alla poole. Viimane viga, mis plaadi peal leidsin oli seotud I2C suhtlussiiniga. Nimelt olin jällegi ülemskeemis ühendanud kaheskeemi juhtmed valepidi kokku (Joonis 42).



Joonis 42. Revisjon 1 I2C suhtlussiiniviga

Pärast kõiki nende vigade üle vaatamist ning parendamist, hakkasid kõik funktsioonid plaadi peal tööle ning töötasid laua peal nii nagu disainifaasis sai kirja pandud. Vormeli peal ei ole saanud plaati proovida, sest vormel ei ole valmis. Usun, et vormeli peal võib tulla veel vigu välja.

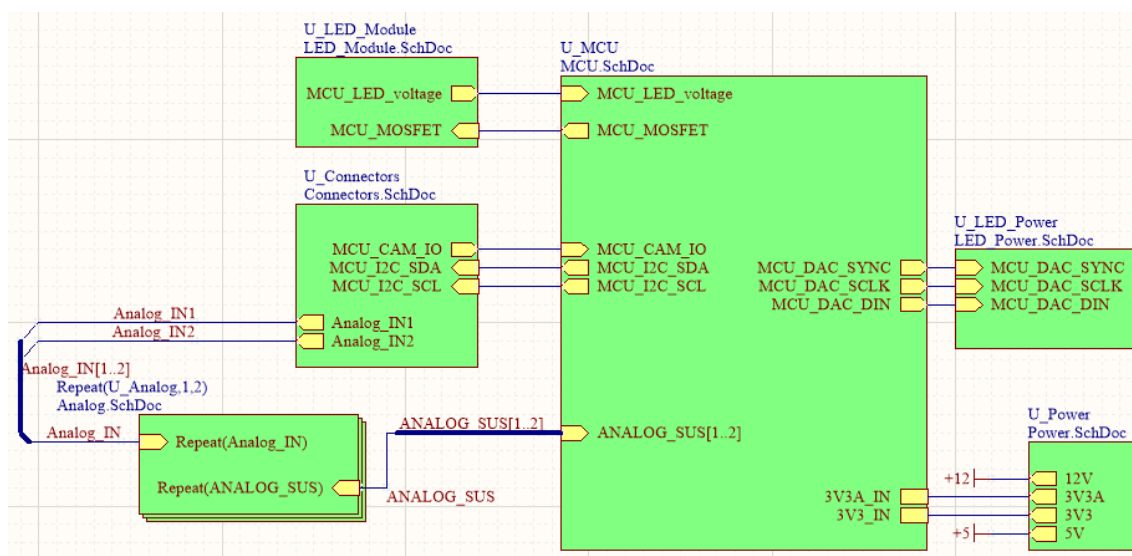
6.5 Revisjon 2

Trükkplaadi teine versioon on oluline, et esimeses revisjonis välja tulnud vead parandada. Pärast esimest revisjoni skeemide üle kontrollimist ja testimist tulid välja mõned lihtsad vead, mis vajaksid nii skeemides kui ka trükkplaadi joonistes parandamist. Samuti oli lisatud testimiseks lisa pistik, millega sai kuulata pealt välise DAC suhtlust. Kuna see sai testitud ja tööle, siis seda ei ole enam teises revisjonis vaja.

Veel kaotati ära temperatuuri andur, mille vajalikkust ei näinud. Teises revisjonis lisasin selle asemel analooghela, mis annab tagasisidet, kui palju DAC pinget peale seadistas. Samuti muutus ühenduspistik, kuhu lisandusid 4 viiku kõrgusanduri juhtmete jaoks. Välja tulnud vead esinesid LED ja muudes toiteskeemides. Mikrokontrolleri ja DAC vahelised signaalijuhtmed olid ka valesti ühendatud omavahel ning samuti oli vahetuses I2C kellatakt ja andmesiin.

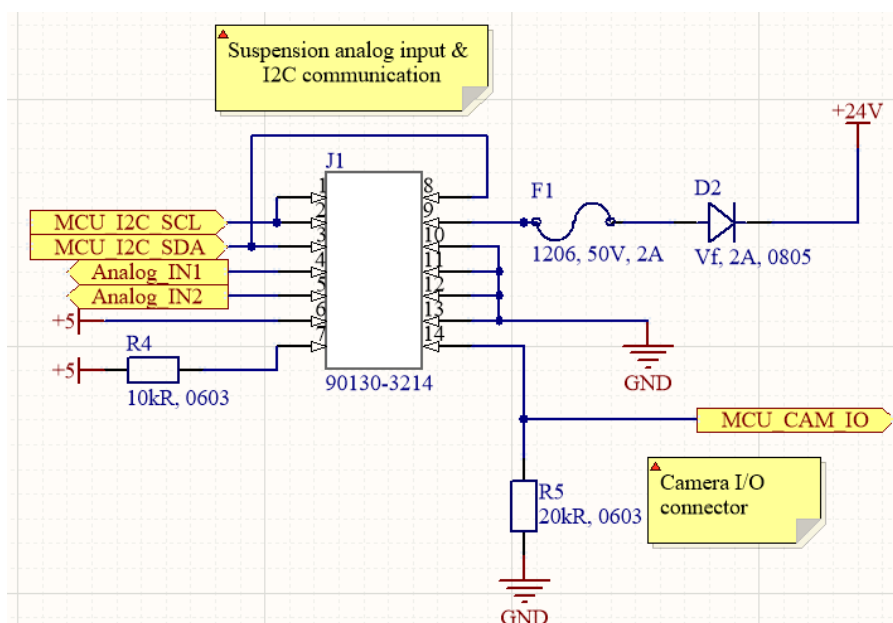
Muudatused elektriskeemides

Ülemskeemis (joonisel 43) tuli teha mitu muudatust. Muudatused põhinesid alamskeemide vahelistes ühendustes. Olin ühendanud valesti kokku paar ühendust. Parendused toimusid MCU ja LED toite skeemi vahel, kus nüüd ühenduvad omavahel kokku SYNC signaalid, SCLK signaalid ning DIN signaalid. Samuti olin omavahel segamini ajanud I2C andmesiinid. Muutsin signaalid õigeks, et SDA on omavahel kokku ühendatud ja samuti ka SCL signaalid. Muid suuremaid muutusi ei teinud ülemskeemis.



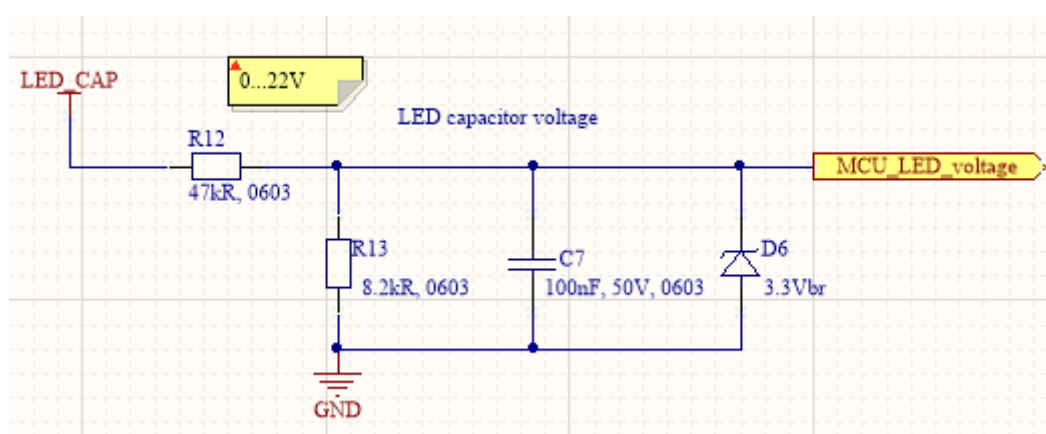
Joonis 43. Revisjon 2 ülemskeem

Ühenduspistiku skeemis (joonisel 44) tein mitu muudatust. Esiteks muutus ühenduspistik ise. Eelmises revisjonis oli 2x5 ühenduspistik. Revisjon 2 lisasin selle asemel 2x7 ühenduspistiku. Lisa ühendusviigud lisasin kõrguseanduri juhtmete jaoks. Kuna kõrguseandur asub samas kohas kus trükkplaat, siis kõige lihtsam tundus ühendada kõrguseanduri pistik LED valgustustrükkplaadile külge. Kõrguseanduri andmesiin ühendub otse trükkplaadi I2C külge. Samuti saab kõrguseandur 5V toitepinge ja ühise 0V liini trükkplaadilt.



Joonis 44. Revisjon 2 ühenduspistiku elektriskeem

LED toitemuunduri pinge, mis on vahemikus 4.5V kuni 21.5V, loetakse sisse analoog elektriabeluga (joonisel 45). Elektriabel pingejagur on disainitud lugema 0V kuni 22V, mis on piisav. R12 ja R13 on pingejaguri takistid, mis tekitavad lineaarselt 0 kuni 22 pingele 0 kuni 3V3 pinget. Samuti on lisatud madalpääsfilter kondenssaator C7 skeemile, mis silub kõrgemaid sagedusi, et ADC lugemine mikrokontrolleris oleks võimalikult müravaba. Kaitsedioid D6 kaitseb mikrokontrolleri viiku kõrgemate kui 3V3 pingete eest.



Joonis 45. Revisjon 2 LED toitemuunduri tagasiside elektriskeem

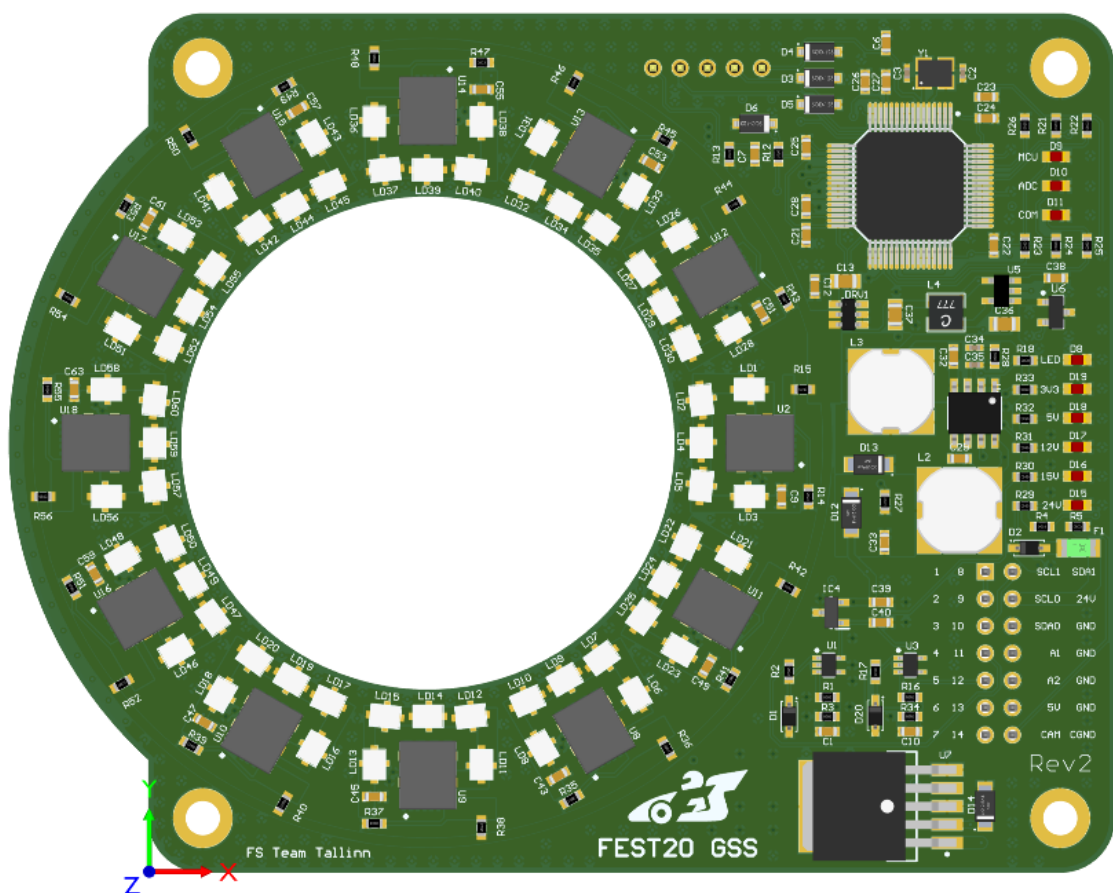
Muudatused trükkplaadi joonistes

Esimeses revisjonis oli väga keeruline paigutada kõiki komponente kohe võimalikult efektiivselt, sest trükkplaadi dimensioonid olid valitud võimalikult väikesed. Teises

trükkplaadi revisjonis parandasin mitmeid trükkplaadi jooniseid ja tõstsin nende paigutuse ringi, et need oleksid loogilisemad. Teine trükkplaat on ühe küljega 10mm pikem, et mahutada komponente paremini trükkplaadil.

Ülemine kiht

Ülemise kihi muutmisel (joonisel 46) pidin otsustama, millise osa ma trükkplaadi joonistest ringi teen. Ma ei tahtnud liigutada LED mooduleid, sest need tundusid olevat väga hästi paigutatud. Veel ei tahtnud liigutada mikrokontrollerit, sest selle asukoht tundus ka õiges kohas. Ühenduspistik ja 12V toitemuundur jäid ka muutmata. Mikrokontrolleril muutsin möödaviigu kondensaatorite asukohtasid, üritasin neid lihtsalt paremini ja lähemale paigutada mikrokontrolleri viikudele.



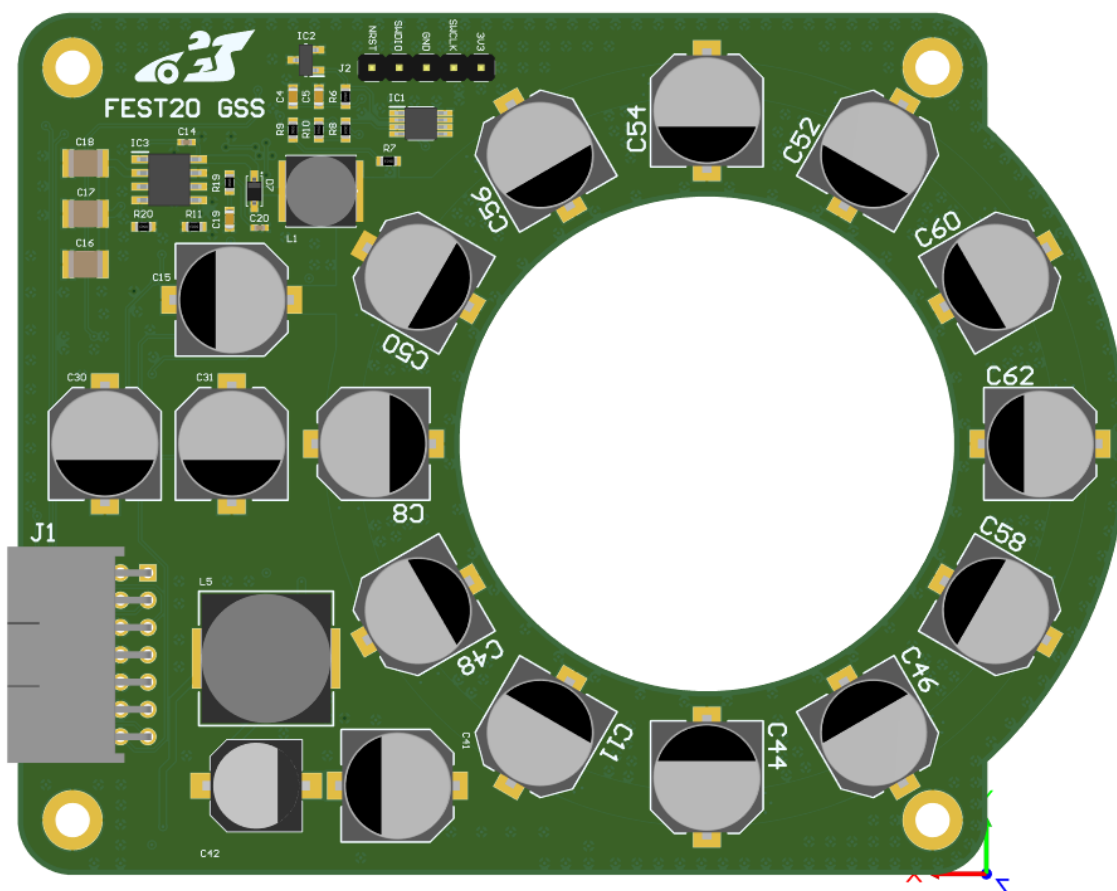
Joonis 46. Revisjon 2 trükkplaat, ülemine kiht

Suurim muutus toimus analoogelektriahelates ning 5V ja 3V3 toitemuunduritega. Analoog elektriskeemid viisin ühenduspistiku kõrvale. Mainitud toitemuundurid paigutasin ümber. Tõin 3V3 toitemuunduri trükkplaadi alumiselt kihilt ülemisele. Kuna 3V3 toidet kasutab põhiliselt mikrokontroller, siis otsustasin, et see peab olema võimalikult lähedal mikrokontrollerile. Seega 5V toitemuunduri ja mikrokontrolleri

vahele paigutasin 3V3 toitemuunduri. 5V toitemuunduril tõin induktorid kiibile lähemale. Muid suuremaid paigutus muudatusi ülemisel kihil ei tehtud.

Alumine kiht

Alumisel kihil (joonisel 47) ei toimunud nii palju muudatusi komponentide liigutamises kui ülemisel kihil. Alumisele kihile jätsin jäädavalt suuremate korpustega komponendid. Kuna trükkplaat läks 10mm pikemaks, siis oli natukene rohkem ruumi, et suurte korpusega komponente paremini paigutada.



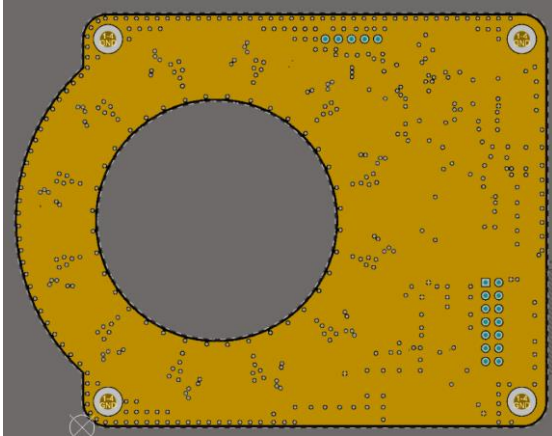
Joonis 47. Revisjon 2 trükkplaat, alumine kiht

Alumiselt kihile jäi LED toitemuundur, millele tehti muudatused, mis leiti testimisel. Samuti kõik toitemuunduri sisend- ja väljundkondensaatid jäid alumisele kihile, sest neil on kõrged korpused.

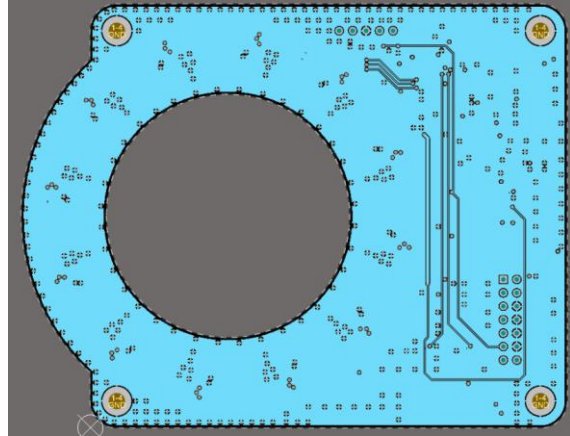
Signaalikihid

Signaalikihtide põhimõte ei muutunud. Üks signaalikiht on kasutusel toitepingega 24V (joonisel 48) ning teine 0V (joonisel 49). Võrreldes eelmise revisjoniga, siis toitepinge 24V signaalikihilt paigutati kõik trasseeritud signaalid ja muud toiterajad 0V

signaalikihile (joonisel 49). Muid olulisi muudatusi ei tehtud. Null pingega signaalikihil trasseeriti rajad, mis ei mahtunud ülemisele ja alumisele kihile ära. Null kihi põhieesmärk on ühendada kokku kõik null pingega ühendused plaadil.



Joonis 48. Revisjon 2 trükkplaadi 24V kiht



Joonis 49. Revisjon 2 trükkplaadi 0V kiht

7 Mikrokontrolleri tarkvaraarendus

Mikrokontrolleri tarkvara arendamisel kasutan STM32CubeIDE¹ integreeritud arenduskeskkonda. Eelmistel aastatel on kasutatud Atollic TrueSTUDIO² programmeerimiseks ning mikrokontrolerile programmi peale laadimiseks mõeldud keskkonda. Mikrokontroleri perifeeria konfiguratsioon tehti programmi STM32Cube abil, mis hoidis väga olulist aega kokku. Kasutasin programmi STM32CubeIDE, mis on ühitanud eelnevalt mainitud kaks keskkonda kokku üheks integreeritud programmeerimiskeskonnaks ning põhiliselt mõeldud STM32 mikrokontrolerite arendamiseks. Tudengivormelis on oluline, et kõik süsteemid oleksid programmeeritud ühise keskkonna põhiselt. See teeb kogu tarkvaralist süsteemi ühtsemaks. GSS trükkplaadile on disainitud STM32F730R8T³ mikrokontroller.

Programmikood on kirjutatud C-keeles. Programmeerimisel on kasutatud HAL teeki⁴, mis on riistvara peale defineeritud abstraktne tase. See on ülioluline, sest iga perifeeria seadistamiseks vajaminevate registrite mäluaadresside otsimine on väga ajamahukas ning väga lihtne on teha viga kindla mäluaadressi registriväärtuse muutmisel. Seega HAL teek on väga kasulik, sest on nimetanud kõik mikrokontroleri seadistamisega seotud registrid. Autor kasutas HAL teegi poolt defineeritud funktsioone, mis aitasid suhelda otse mikrokontroleri raudvaraga. STM32CubeIDE tarkvara keskkonnas saab defineerida kõik perifeeria seadistuse graafilise liidese kaudu, mis genereerib algse lähtekoodi, kasutades HAL teeki. Seadistati I2C, ADC, DMA, taktsignaali konfiguratsioon, taimer ning GPIO viigud. Mikrokontroller kasutab väga täpset välist 25Mhz taktsignaali generaatorit, mis aitab mikrokontroleri sisemisel PLL loogikal tõsta mikrokontroleri taktsignaali 216Mhz kiiruseni. Tarkvara silumiseks ning mikrokontroleri programmeerimiseks kasutatakse SWD (ingl.k *Serial Wire Debug*) liidest.

¹ <https://www.st.com/en/development-tools/stm32cubeide.html>

² <https://atollic.com/truestudio/>

³ <https://www.st.com/en/microcontrollers-microprocessors/stm32f730r8.html>

⁴ https://www.st.com/resource/en/user_manual/dm00189702-description-of-stm32f7-hal-and-lowlayer-drivers-stmicroelectronics.pdf

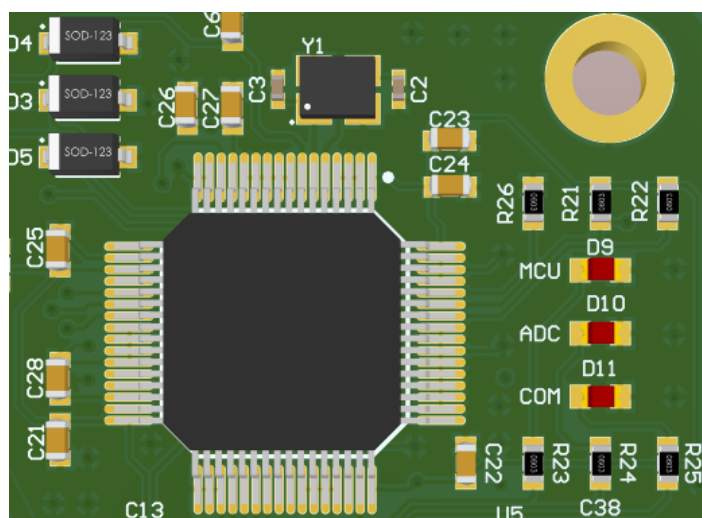
7.1 Mikrokontrolleri perifeeria seadistamine

Käesolev alampeatükk kirjeldab mikrokontrolleri perifeeria konfiguratsiooni. Kõik algne lähtekood on seadistatud STM32CubeIDE programmi graafilise liidese kaudu.

GPIO

GPIO (ingl.k *general purpose input/output*) on üldise otstarbega sisend ja väljund viigud mikrokontrolleeril, mida saab vastavalt programmeerida väljundina või sisendina. Sisendi puhul saab näiteks sisse lugeda nupuvajutuse olekut ja väljundi puhul saab lülitada väikse voolutarbega LED-lampi.

GSS trükkplaadil (joonisel 50) on kolm LED indikaatorit, mis näitavad mikrokontrolleri programmi tööolekut. MCU LED näitab, et põhitsüklitel töötab. ADC LED näitab, et programm tegeleb analoog-digitaalmuundurist andmete töötlemisega ja andmete mällu salvestamisega. Viimaseks COM LED näitab, et mikrokontrolleri ja Jetson TX1 arvuti vaheline suhtlus I2C liinil töötab.



Joonis 50. Mikrokontrolleri LED indikaatorid

ADC

Mikrokontrolleri sisemine analoog-digitaalmuundur on maksimaalselt kasutatav 12-bitise resolutsiooniga. Tõlgitakse mikrokontrolleri viigule rakendatud analoogpinge digitaalsele kujule. Mikrokontrolleer töötab pinges vahemikus 0V kuni 3V3.

Trükkplaadil on kasutusel 3 analoog elektriskeemi. Kaks skeemi on mõlema vedrustusanduri signaali lugemiseks ning kolmas skeem loeb LED toiteahela pinget. ADC parameetrid kõigi kolme signaali sisse lugemiseks on defineeritud

STM32CubeIDE programmis joonisel 51. DMA on seadistatud suunaga perifeeriast mällu ning ringpuhvri. Andmesõna suurus on 16-bit. Katkestusvektoreid ei ole kasutatud.

▼ ADCs_Common_Settings		
Mode		Independent mode
▼ ADC_Settings		
Clock Prescaler		PCLK2 divided by 8
Resolution		12 bits (15 ADC Clock cycles)
Data Alignment		Right alignment
Scan Conversion Mode		Enabled
Continuous Conversion Mode		Enabled
Discontinuous Conversion Mode		Disabled
DMA Continuous Requests		Enabled
End Of Conversion Selection		EOC flag at the end of all conversions
▼ ADC_Regular_ConversionMode		
Number Of Conversion		3
External Trigger Conversion Sou...		Regular Conversion launched by software
External Trigger Conversion Edge		None
> Rank		1
> Rank		2
> Rank		3

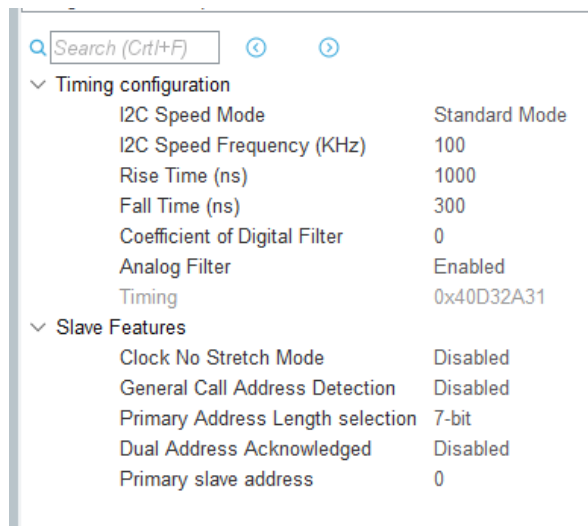
Joonis 51. Mikrokontrolleri ADC perifeeria seadistus

SPI

SPI andmesiini kasutatakse, et juhtida välist DAC kiipi. See sai valitud, sest DAC andmelehes oli etteantud protokollid, millega on mugav kiibiga suhelda. Mikrokontrolleeril on SPI riistvaraline tugi, seetõttu kasutan seda.

I2C

Tegemist on andmesuhtlus protokolliga, mis kasutab kahte kahesuunalist avatud kollektoriga ühendust. Üks juhe on andmesiin ning teine juhe kellataktisignaali. Vähenenud juhtmete kasutamine on ka üheks põhjuseks, miks see protokoll sai valitud. Teine põhjus on omadus ühendada mitu seadet sama suhtlussiini peale. Konfigureeritud parameetrid on illustreeritud joonisel 52.



Joonis 52. I2C perifeeria seadistus

EXTI

Mikrokontrolleril on seadistatud ühele viigule väline katkestussignaali. See tähendab, et mikrokontrolleri viik tunneb ära pinge muutumise ning kutsub programmi töös välja katkestustöötlusfunktsiooni, mis omakorda käivitab LED lülituse eelnevalt defineeritud ajaperioodiks.

Silumine

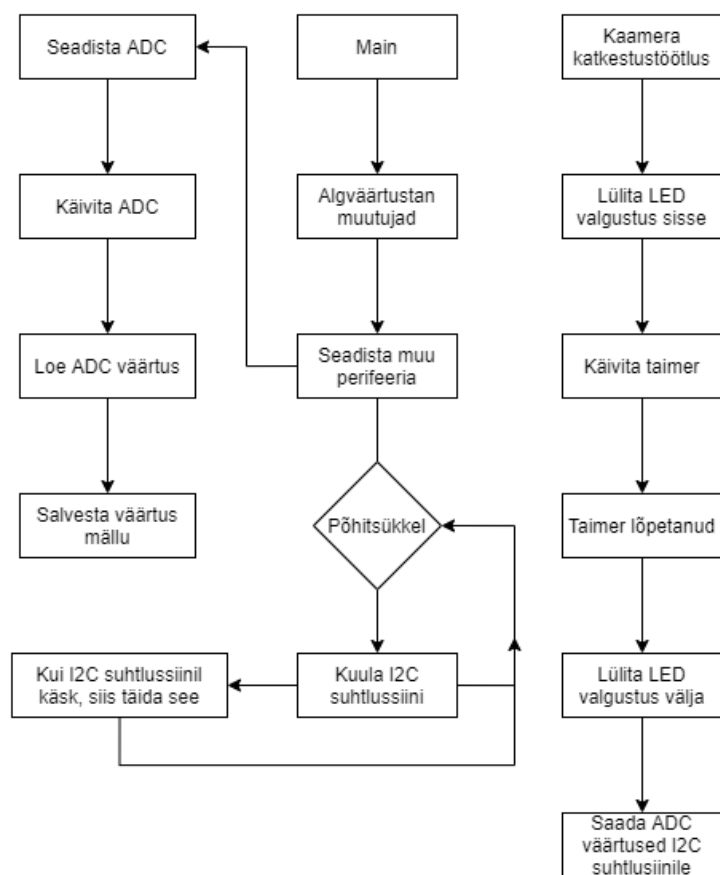
Silumine ja programmi peale laadimine toimub SWD liidese kaudu. Liidesel on viis juhet kokku.

7.2 Mikrokontrolleri programmeerimine

Mikrokontrolleri programmeerimise alampeatükk selgitab, kuidas mikrokontroller programmeeriti. Selgitatakse programmivooskeemi ning kõiki funktsioone, mida tarkvara kontrollib.

Programmi vooskeem

Programmivoo skeem näitab, kuidas programm töötab (joonisel 53). Programmitöös esmalt väärtustatakse algandmed ja perifeeria. Seejärel astub programm põhitöö tsüklisse.



Joonis 53. Mikrokontrolleri programmi vooskeem

Analoog-digitaalmuundur

Analoog-digitaalmuundur käivitatakse programmivoo alguses ning jäetakse käima. Kuna ADC kasutab otsemällupöörduskanaleid (DMA), siis see ei koorma ega sega mikroprotsessori tööd. Iga kord, kui toimub kaamera poolt katkestus, siis toimub ADC andmete mälust lugemine ja I2C suhtlussiini kaudu Jetson TX1 saatmine.

Valgustugevuse kontrollimine

Valgustugevuse kontrollimine toimub läbi välise digitaal-analoogmuunduri, mida saab programmeerida erinevate väärtusega läbi SPI liidese. Valgustugevuse muutmine on võimalik mikrokontrolleerile uuesti koodi peale laadimisega või hiljem ka läbi I2C suhtluskanali. Jetson TX1 saadab kindla sõnumi, millega defineeritakse valgustugevus ja käsk seda muuta.

LED valgustuse lülitamine

Valguse lülitamise kontroll on ainult kaamera genereeritud katkestussignaalil. See tähendab, et valgustust lülitatakse katkestustöötlusfunktsioonis, mis lülitab MOSFET draiverit, mis omakorda lülitab LED valgustuse sisse ja välja.

Suhtlussiin I2C

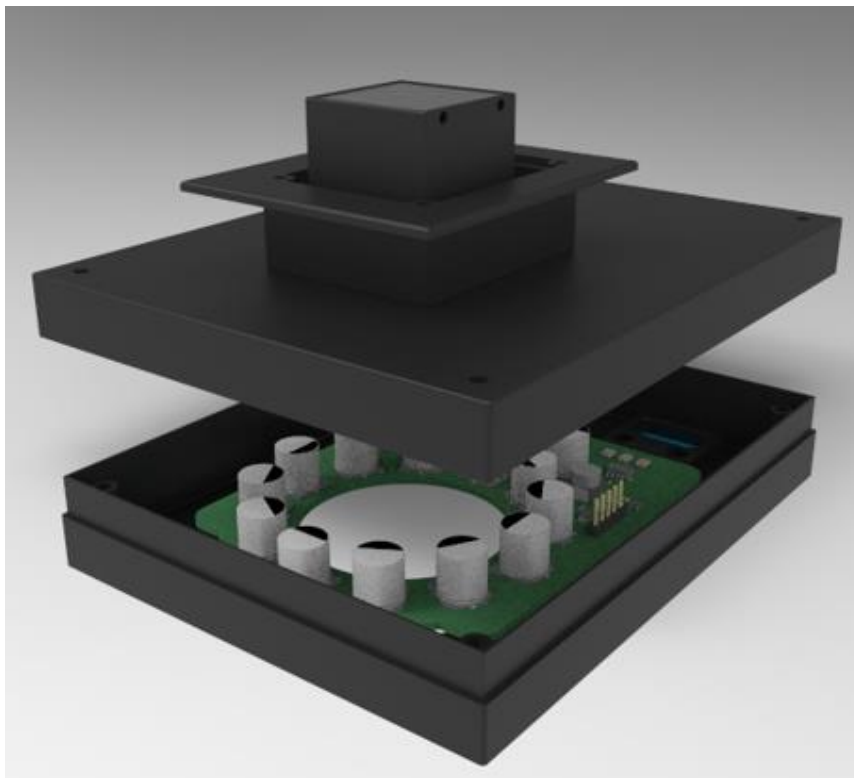
Suhtlussiin on I2C, mis kasutab katkestustöötlusfunktsioone, et saata ning võtta vastu sõnumeid. GSS süsteemis on ülemsõlmeks Jetson TX1 arvuti ning alamsõlmedega isearendatud trükkplaat ja kõrguseandur.

LED indikaatorid

LED indikaatorid näitavad töötava programmi olekut. Esimene LED indikaator on programmeeritud vilkuma, kui mikrokontrolleri põhitsükkel töötab ning ei ole kuhugile kinni hangunud. Teine LED indikaator on pandud vilkuma, kui I2C suhtlus toimub ning viimane on seadistatud vilkuma, kui analoog-digitaalmuundur tõlgib analoogpingeid digitaalseks ning salvestab neid mälli.

8 Korpused

GSS koosneb mitmest riistvaralisest komponendist ehk on vaja modelleerida mitu korpust, mis kaitseks ja aitaks kinnitada neid. Kuna FEST20 vormelis on väga vähe ruumi, siis tuli luua korpused, mis oleksid võimalikult efektiivsed ning hoiaksid detaile kinni. Kõik korpused on modelleeritud CATIA V5 programmiga. Joonisel 54 on GSS-i korpused, mis paigutatakse vormeli põhja alla. Need korpused hoiavad trükkplaati, kõrguseandurit ja kaamerat koos objektiiviga paigal.



Joonis 54. Trükkplaadi ja kaamera kinnitus korpused

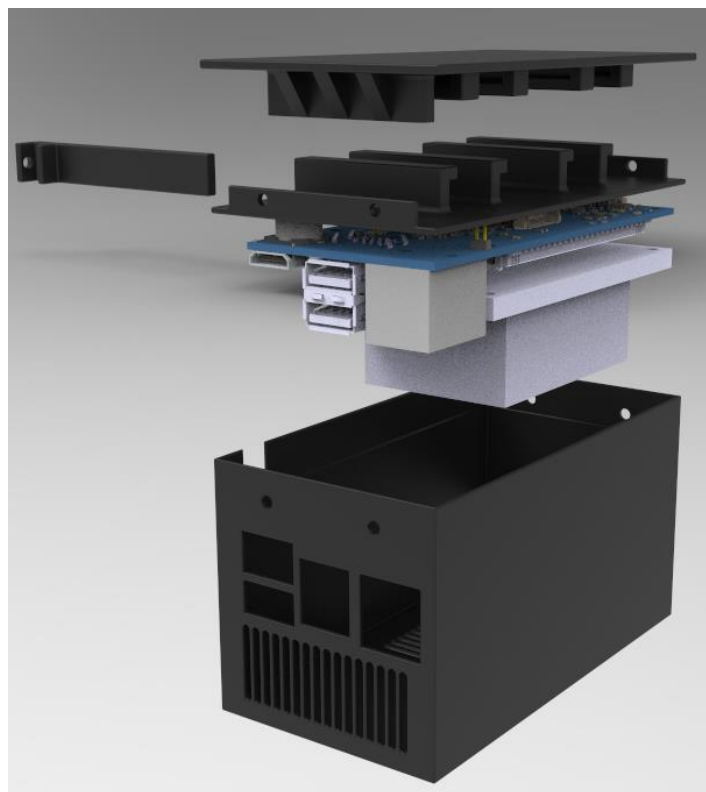
Trükkplaat

Trükkplaat paikneb auto põhja all, mis tähendab, et trükkplaadile tuleb teha korpus, mis kaitseks teda erinevate ilmastiku tingimuste eest ning muude auto alt läbi lendvate objekti eest. Kuna toimub pidev testimine, siis korpus peab mugavalt lahti käima, et see oleks võimalikult lihtsasti eemaldatav auto põhja alt. Trükkplaadi korpuses on veel kõrguseandur ja kaamera koos objektiiviga. Trükkplaadi korpus on tehtud trükkplaadi mõõtmete järgi ning jäetud natukene ruumi ka juhtmete jaoks. Trükkplaadi korpus koosneb ülemisest ja alumisest korpusest (vt lisa 2). Alumisel korpusel on trükkplaadi

valgustuse jaoks suur ümmargune ava ning kõrguseanduri jaoks väike ava, kuhu saab selle kinnitada. Trükkplaadi ülemine korpus kinnitatakse vormeli põhja külge liimiga. Kui ülemine korpus ja põhjaava kinnitus on vormeli põhja küljes kinni, siis kaamera saab kaamera hoidjaga kinnitada põhja alla. Kuna trükkplaadi alumise korpuse külge kinnitub trükkplaat, siis koos alumise korpusega kinnitatakse trükkplaat auto põhja külge (vt lisa 3). Lisas 4 on toodud pilt, kuidas trükkplaat ja kaamera paiknevad korpuses.

Jetson TX1

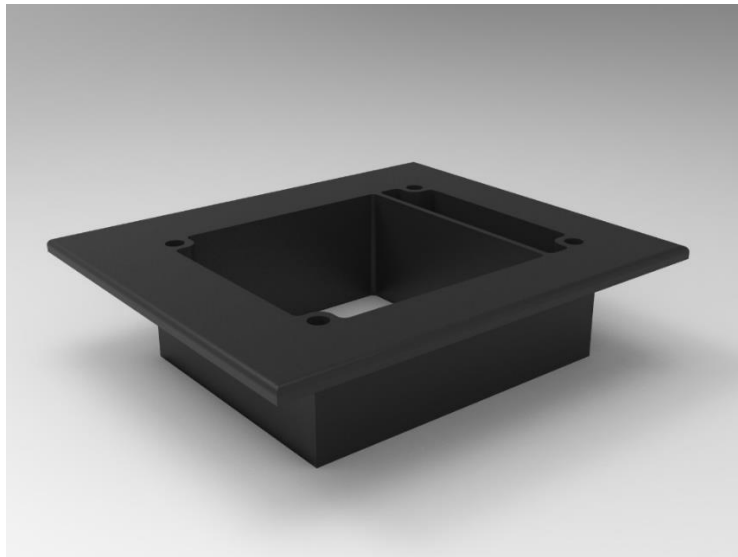
Jetson TX1 arvuti korpus (joonisel 55) asub kokpitis sees. Täpsemalt juhi jalgade kohal laes. Samamoodi nagu trükkplaadi korpusega on siin oluline, et korpust oleks võimalikult lihtne ära võtta, et saaks mugavalt teha muudatusi süsteemis. Joonisel 55 kõige ülemine detail kinnitatakse vormeli lakke ning ülejäänud korpus kinnitub selle detaili külge. Ilmastiku poolt on arvuti korpus rohkem kaitstud ning ei pea nii veekindel olema. Jetson TX1 korpusega peab tähele panema, et arvuti läheb väga kuumaks. See tähendab, et korpuses peab olema piisavalt avasid, kus õhuvool saaks takistuseta liikuda.



Joonis 55. Jetson TX1 arvuti korpus

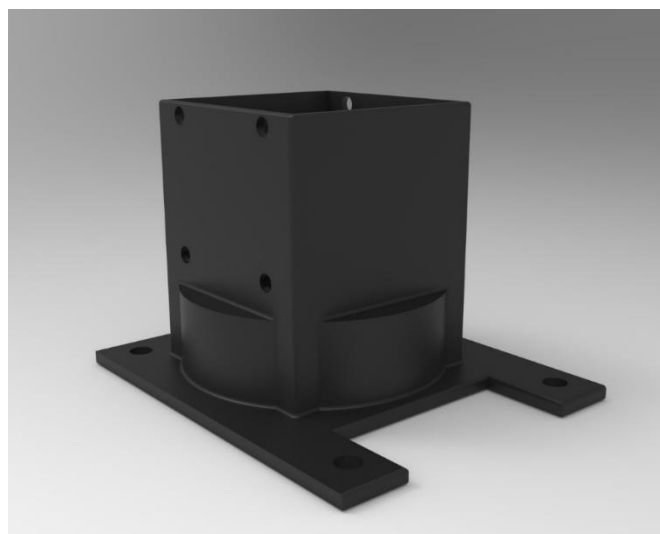
Kaamera hoidja

Kaamera hoidja mõte on see, et trükkplaati ja kaamerat saaks võtta eraldi välja ning et nad kinnituksid eraldi vormeli monokoki põhja külge. Vormeli monokoki põhja on tehtud GSS jaoks 50x40mm ava, millesse valmistasin põhjaava korpuse (joonisel 56). Põhjaava korpus on täpselt vormeli põhjapaksuse kõrgune.



Joonis 56. Põhjaava kinnitus korpus

Korpus liimitakse avasse ning korpusel on kinnitamisaukud nurkades, kuhu saab kinnitada trükkplaadi korpuse ja kaamera hoidja (joonisel 57).



Joonis 57. Kaamera hoidja korpus

9 Kokkuvõte

Käesoleva bakalaureusetöö käigus toimus eelmise aasta tudengivormeli maapinna kiiruse anduri edasi arendamine. Maapinna kiiruse anduri arendus toimub tudengivormeli FEST20 vormelil. Süsteemi juures tehti väga palju muudatusi, mis muudaksid GSS kiiruse arvutamise töökindlamaks. Samuti muudeti GSS asukohta vormelis, mis tähendas, et kiiruse andurile tuli teha uued korpused. GSS arvutatud kiirus kirjutatakse auto üldisele CAN suhtlussiinile, kus ülejäänud vormeli arvutisüsteemid saavad kiiruse andmetele ligipääsu. GSS süsteemis muudeti järgmisi komponente: kiiruse arvutamise tarkvara, objektiiv, uus isearendatud trükkplaat, korpused, vedrustusandurid ja trükkplaadi mikrokontrolleri tarkvara. Välja ei vahetatud Jetson TX1 arvutit, kaamerat ja kõrguseandurit.

Töö raames ei saanud valmis vormeli FEST20 ehitus, seega ei saanud GSS-i testida reaalses tingimustes. GSS-i esimene versioon sai valmis: tarkvaraarendus ning isearendatud trükkplaat on töökorras, et vormeli peal esimesed testid teha. Kõigi korpuste joonised on 3D modelleeritud, kuid pole toodetud. Toodetud on trükkplaadi esimene revisjon, mille testimise põhjal sai valmis arendatud ka teine trükkplaadi revisjon, mis töö kirjutamise ajal ei ole veel tootmisesse saadetud. GSS-i kõik komponendid on testitud nii üksikult kui ka koos. Vormeli peal tuleb kindlasti veel erinevaid vigasid välja, mida saab alles reaalse testimise ajal teada. Lõputöö raames seatud eesmärk – saada valmis esimene GSS versioon – on täidetud.

Töö käigus valmis kogu süsteemist ülevaade tudengivormeli meeskonna „Confluence“¹ leheküljele. See on väga oluline järgmisele liikmele, kes seda süsteemi hakkab edasi arendama. Samuti on järgnevatel aastatel kasulik materjal nendele, kes soovivad selle süsteemi käitumisest ja teooriast aru saada.

¹ <https://www.atlassian.com/software/confluence>

Kasutatud kirjandus

- [1] H. J. Norden, "FEST19 GSS disainidokument rev4", MTÜ Tudengivormel, 2019
- [2] https://scikit-image.org/docs/dev/_images/sphx_glr_plot_template_001.png
- [3] <https://developer.nvidia.com/sites/default/files/akamai/embedded/images/jetsontx1/TX1fixedFinal.png>
- [4] https://auvidea.eu/download/manual/J120/J120_technical_reference_1.6.pdf
- [5] <https://a70ad2d16996820e6285-3c315462976343d903d5b3a03b69072d.ssl.cf2.rackcdn.com/54b54aa5bf34eb421952501658f844ee>
- [6] https://grobotronics.com/images/detailed/101/3316-00_-_%CE%91%CE%BD%CF%84%CE%AF%CE%B3%CF%81%CE%B1%CF%86%CE%BF.jpg

Lisa 1. Trükkplaadi komponentide nimekiri (BOM)

ProjectName	Manufacturer	Product Name	Supplier	Supplier Number	Quantity	Order Step	Price (1st level)
GSS_FEST20- EV_Rev1.PrjPCB	AVX	06035C333JAT2A	Farnell	1740618	10	10	0.16
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MC0402N160J500CT	Farnell	1758946	10	10	0.05
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MC0603B104K500CT	Farnell	1759122	20	10	0.04
GSS_FEST20- EV_Rev1.PrjPCB	PANASONIC	EEEFT1E102UV	Farnell	3126815	25	1	1.85
GSS_FEST20- EV_Rev1.PrjPCB	KEMET	C0805C106J8RACAUTO	Farnell	2776879	5	5	0.99
GSS_FEST20- EV_Rev1.PrjPCB	PANASONIC	35SVPD47M	Farnell	2491821	2	1	2.96
GSS_FEST20- EV_Rev1.PrjPCB	TAIYO YUDEN	UMK325C7106MM-T	Farnell	2112860	6	1	0.93
GSS_FEST20- EV_Rev1.PrjPCB	KEMET	C0603C621J5GACTU	Farnell	2819768	10	10	0.35
GSS_FEST20- EV_Rev1.PrjPCB	MURATA	GRM188R61A475KE15D	Farnell	2362098	10	10	0.08
GSS_FEST20- EV_Rev1.PrjPCB	TDK	C1608X7R1H224K080AB	Farnell	2346906	10	10	0.14

GSS_FEST20- EV_Rev1.PrjPCB	RUBYCON	50THV100M10X10.5	Farnell	2102438	4	1	0.74
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MC0402F223Z500CT	Farnell	2627377	10	10	0.02
GSS_FEST20- EV_Rev1.PrjPCB	KEMET	C0402C101J5GACTU	Farnell	1414572	10	10	0.03
GSS_FEST20- EV_Rev1.PrjPCB	TDK	CGA4J3X7R1A475K125AB	Mouser	810- CGA4J3X7R1A475AB	2	1	0.25
GSS_FEST20- EV_Rev1.PrjPCB	RUBYCON	50THV100M10X10.5	Farnell	2102438	2	1	0.74
GSS_FEST20- EV_Rev1.PrjPCB	PANASONIC	EEEFK1E331P	Farnell	9695796	2	1	0.87
GSS_FEST20- EV_Rev1.PrjPCB	BOURNS	CDSOD323-T12C	Farnell	1824863	4	1	0.70
GSS_FEST20- EV_Rev1.PrjPCB	VISHAY	SD103AWS-E3-08	Farnell	2393466	10	5	0.27
GSS_FEST20- EV_Rev1.PrjPCB	NXP	PTVS3V3S1UR	Farnell	1829253	10	5	0.33
GSS_FEST20- EV_Rev1.PrjPCB	VISHAY	SD103AWS-E3-08	Farnell	2393466	10	5	0.27
GSS_FEST20- EV_Rev1.PrjPCB	KINGBRIGHT	KP-1608MGC	[NoValue], Farnell	8529825	15	5	0.35
GSS_FEST20- EV_Rev1.PrjPCB	KINGBRIGHT	KPT-1608QBC-D	Farnell	2217978	10	5	0.22
GSS_FEST20- EV_Rev1.PrjPCB	TAIWAN SEMICONDUCTOR	S1AL	Farnell	1559148	5	5	0.06
GSS_FEST20- EV_Rev1.PrjPCB	ON SEMICONDUCTOR	MBRS340T3G	Farnell	9557377	5	5	0.27

GSS_FEST20- EV_Rev1.PrjPCB	MAXIM INTEGRATED PRODUCTS	MAX5048CAUT+T	Farnell	2516393	3	1	1.16
GSS_FEST20- EV_Rev1.PrjPCB	LITTLEFUSE	0466002.NR	Farnell	9921877	10	10	0.49
GSS_FEST20- EV_Rev1.PrjPCB	Texas Instruments	DAC8551IDGKR	Farnell	3005678	2	1	6.03
GSS_FEST20- EV_Rev1.PrjPCB	Texas Instruments	LM4040AIM3-4.1/NOPB	Farnell	3009188	1	1	1.56
GSS_FEST20- EV_Rev1.PrjPCB	Texas Instruments	LM35DZ/LFT4	Mouser	926-LM35DZ/LFT4	1	1	1.85
GSS_FEST20- EV_Rev1.PrjPCB	TEXAS INSTRUMENTS	TPS54540DDAR	Farnell	3007358	2	1	4.75
GSS_FEST20- EV_Rev1.PrjPCB	Texas Instruments	TLV76015DBZR	Mouser	595-TLV76015DBZR	2	1	0.41
GSS_FEST20- EV_Rev1.PrjPCB	Molex	90130-3210	Farnell	9734015	2	1	1.83
GSS_FEST20- EV_Rev1.PrjPCB	Coilcraft	994-XAL6060-682MEB	Mouser	XAL6060-682MEB	2	1	2.50
GSS_FEST20- EV_Rev1.PrjPCB	BOURNS	SRN8040TA-470M	Mouser		4	1	0.44
GSS_FEST20- EV_Rev1.PrjPCB	COILCRAFT	XFL3012-103MEB	Farnell	2289191	2	1	1.58
GSS_FEST20- EV_Rev1.PrjPCB	Würth Elektronik	710-7447709151	Mouser	7447709151	2	1	2.22
GSS_FEST20- EV_Rev1.PrjPCB	OSRAM OPTO SEMICONDUCTORS	KR DMLS31.23-5K8L-26- M3W3	Farnell	2981648	130	5	0.95
GSS_FEST20- EV_Rev1.PrjPCB	STMicroelectronics	STM32F730R8T6	Farnell	2980932	2	1	4.06

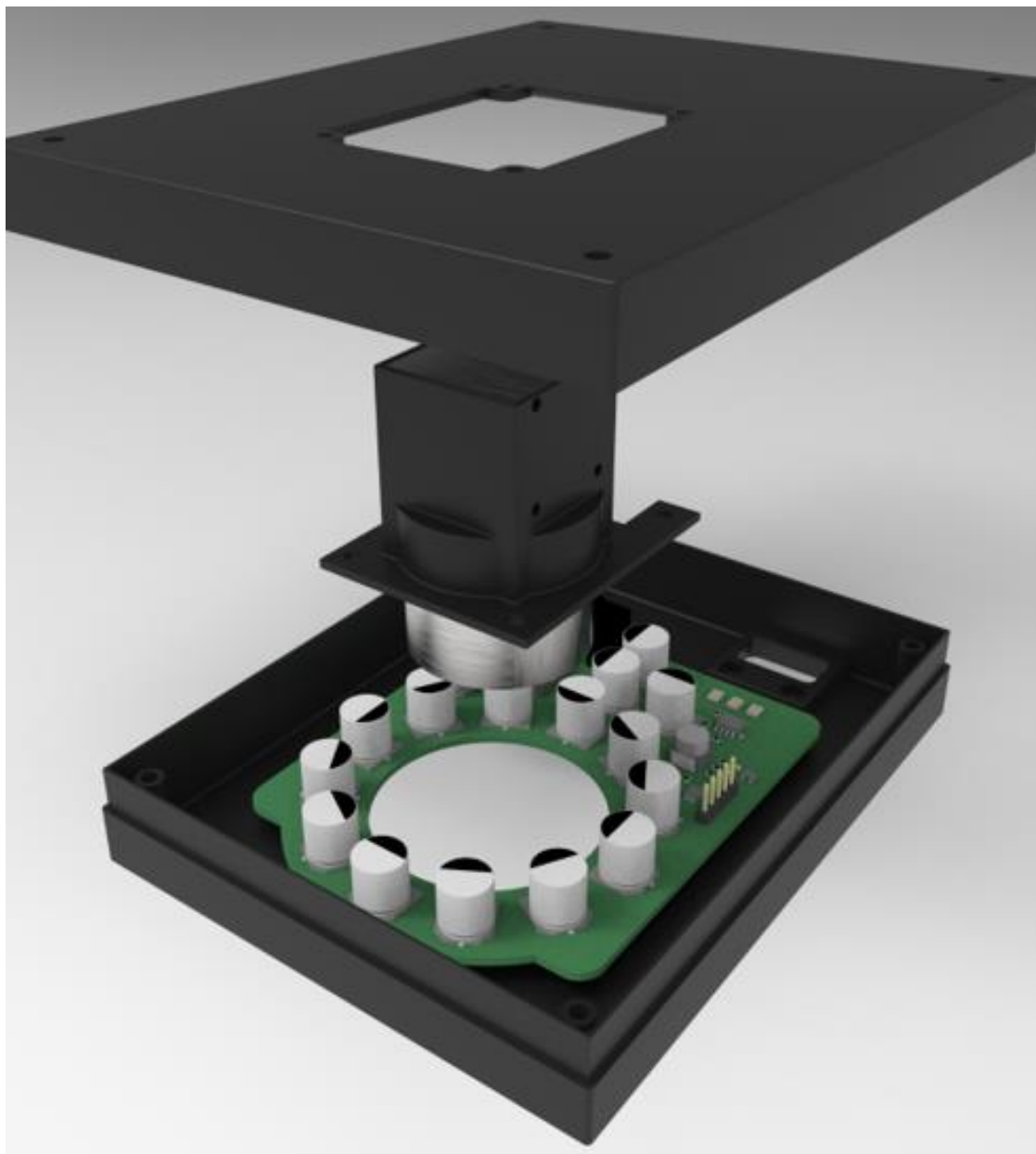
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MCWR06X6201FTL	Farnell	2447425	10	10	0.00
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MCWR06X2002FTL	Farnell	2447293	10	10	0.00
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MCWR06X2001FTL	Farnell	2447319	10	10	0.00
GSS_FEST20- EV_Rev1.PrjPCB	VISHAY	MCT06030C1104FP500	Farnell	2325294	10	10	0.12
GSS_FEST20- EV_Rev1.PrjPCB	TE CONNECTIVITY	CPF0603B261KE1	Farnell	2331077	3	1	0.34
GSS_FEST20- EV_Rev1.PrjPCB	PANASONIC	ERJ3EKF5102V	Farnell	2059471	10	10	0.07
GSS_FEST20- EV_Rev1.PrjPCB	BOURNS	CR0603-FX-1502ELF	Farnell	2333567	10	10	0.02
GSS_FEST20- EV_Rev1.PrjPCB	Panasonic	ERJPA3F40R2V	Farnell	2562021	30	10	0.08
GSS_FEST20- EV_Rev1.PrjPCB	MULTICOMP PRO	MCWR08X1872FTL	Farnell	2447585	10	10	0.01
GSS_FEST20- EV_Rev1.PrjPCB	MUTLICOMP PRO	MCWR06X2053FTL	Farnell	2694765	10	10	0.00
GSS_FEST20- EV_Rev1.PrjPCB	TEXAS INSTRUMENTS	TLV172IDCKT	Farnell	3005236	4	1	1.11
GSS_FEST20- EV_Rev1.PrjPCB	STMicroelectronics	STL7N6LF3	Farnell	2807205	25	1	0.73
GSS_FEST20- EV_Rev1.PrjPCB	Infineon Technologies	TLE8366EV50XUMA1	Mouser	726- TLE8366EV50XUMA1	2	1	2.10
GSS_FEST20- EV_Rev1.PrjPCB	ON SEMICONDUCTOR	NCP51460SN33T1G	Farnell	2464416	2	1	0.48

GSS_FEST20- EV_Rev1.PrjPCB	TEXAS INSTRUMENTS	LM3670MF-3.3/NOPB	Farnell	1685757	5	5	0.73
GSS_FEST20- EV_Rev1.PrjPCB	Microchip	LM2576-12WU	Farnell	2920801	2	1	1.93
GSS_FEST20- EV_Rev1.PrjPCB	ABRACON	ABM8G-25.000MHZ-18- D2Y-T	Farnell	2101339	2	1	0.93

Lisa 2. Trükkplaadi ülemine ja alumine korpus



Lisa 3. Trükkplaadi ja kaamera eemaldamine



Lisa 4. Valgustrükkplaat korpuses, kaamera ja kõrguseanduriga

