



TALLINNA TEHNIKAÜLIKOOL

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

MULTIFUNKTSIONAALSE MOBIILSE ROBOTPLATVORMI ARENDAMINE

**Development of a multi-functional mobile robot
platform**

BAKALAUREUSETÖÖ

Üliõpilane: Märtn Laanemets

Üliõpilaskood: EAAB206619

Juhendaja: Vahur Maask, insener

Kaasjuhendaja: Taavi Möller, insener

Tallinn 2023

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud.

Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

20. detsember 2023

Autor:

/ allkiri /

Töö vastab bakalaureusetöö esitatud nõuetele

20. detsember 2023

Juhendaja:

/ allkiri /

Kaasjuhendaja:

/ allkiri /

Kaitsmisele lubatud

".....".....20..... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina Märtn Laanemets

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Multifunktsionaalse mobiilse robotplatvormi arendamine“, mille juhendaja on Vahur Maask

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

20.12.2023

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

Elektroenergeetika ja mehhatroonika instituut

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Märtn Laanemets, EAAB206619

Õppekava, peeriala: EAAB, Mehhatroonika

Juhendaja(d): Insener, Vahur Maask, 6203703

Insener, Taavi Möller, 6203706

Lõputöö teema:

Multifunktsionaalse mobiilse robotplatvormi arendamine

Development of a multi-functional mobile robot platform

Lõputöö põhieesmärgid:

1. Olemasoleva robotplatvormi puuduste tuvastamine
2. Uue kontrollerplaadi välja arendamine
3. Robotplatvormile tarkvara kirjutamine ja selle katsetamine

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Lähteandmete defineerimine ja komponentide valimine	detsember 2022
2.	Trükkplaadi elektroonika disainimine	märts 2023
3.	Trükkplaadi ja komponentide tellimine	aprill 2023
4.	Trükkplaadi kokku jootmine ja elektroonika tööle saamine	august 2023
5.	Põhifunktsionaalsuste programmeerimine ja testimine	oktoober 2023
6.	Demo programmi loomine	november 2023
8.	Juhendajale läbilugemiseks saatmine	november 2023
9.	Paranduste sisseviimine	detsember 2023
10.	Töö lõplik versioon valmis	detsember 2023

Töö keel: eesti keel **Lõputöö esitamise tähtaeg:** 20. detsember 2023 a

Üliõpilane: Märtn Laanemets ".....".....20.....a
/allkiri/

Juhendaja: Vahur Maask ".....".....20.....a
/allkiri/

Kaasjuhendaja: Taavi Möller ".....".....20.....a
/allkiri/

Programmijuht: Marek Tull ".....".....20.....a
/allkiri/

SISUKORD

AUTORIDEKLARATSIOON.....	2
SISUKORD.....	5
EESSÕNA	7
Lühendite ja tähiste loetelu.....	8
SISSEJUHATUS	9
1 ROBOTI LÄHTEANDMED.....	10
1.1 Olemasoleva lahenduse kirjeldus	11
1.1.1 Pisi-Xbee.....	11
1.1.2 Pisi-Bot.....	12
1.2 Probleemid olemasoleva lahendusega.....	14
1.3 Turul saadaolevad lahendused.....	15
2 KOMPONENTIDE VALIK.....	18
2.1 Mikrokontroller	18
2.2 Ajamid	19
2.3 Andurid	21
2.3.1 Kaugus ja seinte tuvastus	22
2.3.2 Inertsiaalandur	22
2.3.3 Joonejärgimine	23
2.3.4 Enkoodrid.....	23
3 KONTROLLERPLAADI KOOSTAMINE.....	24
3.1 Elektriskeem	24
3.1.1 Viiguread	26
3.1.2 Andmeside ühendused	27
3.1.3 Kasutajaliides	28
3.1.4 Mootori muundur.....	29
3.1.5 Toiteahel.....	29
3.2 Trükkplaadi disain.....	30
3.2.1 Komponentide jälgpinnad.....	30
3.2.2 Trükkplaadi mõõtmed	31
3.2.3 Trükkplaadi kihid.....	32
3.2.4 Komponentide vahelised ühendused	33
3.2.5 Ohukohad	33
3.3 Trükkplaadi kooste.....	34
4 ROBOTI KATSETAMINE	36
4.1 Komponentide ja funktsionaalsuse kontroll.....	36

4.1.1 Mootorid	36
4.1.2 Kaugusandurid.....	37
4.1.3 Jooneandurid	38
4.1.4 Raadiomoodul.....	38
4.2 Näidisprogrammi koostamine	39
KOKKUVÕTE	42
SUMMARY.....	43
KASUTATUD KIRJANDUSE LOETELU	45
LISAD	48
Lisa 1 Programmide lähtekoodid.....	48
Lisa 2 Pisi-Xbee mikrokontrolleri elektroonika skeem	49
Lisa 3 Pisi-Xbee mootori muunduri elektroonika skeem	50
Lisa 4 Pisi-Xbee toiteahela elektroonika skeem.....	51
Lisa 5 Pisi-Xbee raadiomooduli elektroonika skeem.....	52
Lisa 6 Pisi-Xbee inertsiaalanduri elektroonika skeem	53
Lisa 7 Pisi-Xbee trükkplaadi disain	54

EESSÕNA

Lõputöö teema tekkis Robotiklubi algatusel eesmärgiga kaasajastada vana Pisi-Bot robotplatvorm. Projekti idee käis välja Robotiklubi liige ja antud projektijuht Erki Meinberg, kes defineeris esialgsed puudused ja võimalikud lahendused. Töös kasutati vana robotplatvormi algandmeid, mis olid loodud endiste Robotiklubi liikmete poolt. Siinkohal tahaks autor tänada Robotiklubi, antud projekti rahastamise ja töövahendite eest. Samuti tahaks tänada Erki Meinbergi, kes oskas alati tekkinud probleemide korral autorit suunata ja probleemidele lahendusi leida.

Lühendite ja tähiste loetelu

ADC – analoog-digitaal muundur (*analog-to-digital converter*)

CPU – protsessor (*central processing unit*)

DAC – digitaal-analoog muundur (*digital-to-analog converter*)

Flash – välmälu ehk säilmälu

GPIO – mitmeotstarbeline sisend-väljund (*general purpose input output*)

HAL – riistvara abstraksioonikiht (*hardware abstraction layer*)

I2C – kahejuhtmeliides (*inter-integrated circuit*)

IDE – integreeritud programmeerimiskeskond (*Integrated development environment*)

IMU – inertsiaal mõõtmisseade (*inertial measurement unit*)

IOC – IDE seadistuste konfiguratsioon (*IDE Option Configuration*)

IR – infrapuna (*infrared*)

LED – valgusdiod (*light emitting diode*)

MOSFET – isoleeritud paisuga väljatransistor (*metal-oxide-semiconductor field-effect transistor*)

PWM – pulsilaiusmodulatsioon (*pulse width modulation*)

RGB – punase, roheline ja sinise liitvärvimudel (*red, green, blue*)

RTC – reaalaaja kell (*real time clock*)

RX – andmete vastuvõtja

STM – STMicroelectronics

SWD – jadaliides programmi silumiseks (*serial wire debug*)

TX – andmete edastaja

UART – jadaside liides, universaalne asünkroonne vastuvõtja/saatja (*universal asynchronous receiver transmitter*)

SISSEJUHATUS

Robotiklubi on Tallinna Tehnikaülikooli organisatsioon, kus on võimalik läbi praktiliste koolituste ja projektide, arendada teadmisi programmeerimises, elektroonikas ja mehaanikas. Klubi liikmeks saamiseks on vajalik läbida baaskursus, kus omandatakse algteadmised, läbi mille on võimalik hiljem teha otsus, mis sorti projektide peal soovitakse jätkata.

Klubis tekib pidevalt uusi ideid projektide jaoks, kus on vaja kasutada roboteid, millega need ellu viia. Iga projekti jaoks uute robotite välja töötamine on liiga kulukas ja tudengitest huvilistele ajamahukas. Näiteks on hetkel Robotiklubis käsil projekt, milles on vaja juhtida kolme kuni nelja robotit läbi raadiomoodulite, kus robotite juhtkäskude genereerimine toimub masinnägemise kaudu. Taolise projekti puhul ei ole mõtet kulutada ressursse uute robotite loomisele, kuna masinnägemise algoritmi programmeerimine on juba piisavalt keeruline ja ajamahukas. Selliste projektide puhul on hea võtta kasutusele olemasolevad robotid, millel on vajaminev funktsionaalsus juba tagatud ning lihtsalt modifitseerida selle koodi vastavalt vajadustele. Samuti on kasulik taolisi multifunktsionaalseid roboteid kasutada õppe eesmärkidel, kus saab roboti programmeerimise kaudu tutvuda andurite lugemisega ja täiturite kontrollimisega.

Taoline robotplatvorm on Robotiklubis olemas, mis kannab nime Pisi-Bot ning mida kontrollib sellele lihtsasti ühendatav Pisi-Xbee kontrollierplaad. Pisi-Xbeel on aga teatud puudused, mistõttu on tarvis sellest valmis teha uus versioon. Antud lõputöö eesmärgiks on analüüsida olemasoleva robotplatvormi puuduseid ning need likvideerida, arendades välja uus Pisi-Xbee kontrollierplaad.

1 ROBOTI LÄHTEANDMED

Robotplatvormi koostamiseks on tarvis lähteandmeid, et robot vastaks soovitud vajadustele. Olemasoleva Pisi-Bot roboti mehaanilised omadused on välja toodud tabelis 1.1.

Tabel 1.1 Robotplatvormi mehaanilised omadused.

Mehaaniline omadus	Suurus
Pisi-Bot plaadi suurus	100 x 100 mm
Pisi-Xbee plaadi suurus	45 x 50 mm
Roboti kaal (akuta)	120 g
Maksimaalne kiirus	0,7 m/s
Kiirendus maksimaalse kiiruseni	1 m/s ²
Ratta raadius	12,5 mm

Põhi eesmärk antud töö juures on välja arendada uus versioon roboti kontrollerplaadist Pisi-Xbee, mille baasil saaksid tulevikus Robotiklubi võistlusrobotid ja Pisi-Botid opereerida. Peamised vajaminevad funktsionaalsused, mille roboti kontrollerplaat tagab ja mille uus versioon peab tagama, on välja toodud tabelis 1.2. Kontrollerplaati peab olema võimalik ühendada erinevatele robotitele ühte moodi. Sellel on fikseeritud viikude arv, millega ühenduvad robotite andurid, täiturid, toide ja maandus.

Robotplatvormi puhul võrreldakse erinevusi olemasolevate lahendustega, mis turul eksisteerivad, et näha millised on Pisi-Boti eelised. Samuti tuleb uurida, mis probleemid või puudused eksisteerivad olemasoleval versioonil ja kuidas neid parandada.

Tabel 1.2 Kontrollerplaadi vajaminev funktsionaalsus koos vana versiooni andmetega [1].

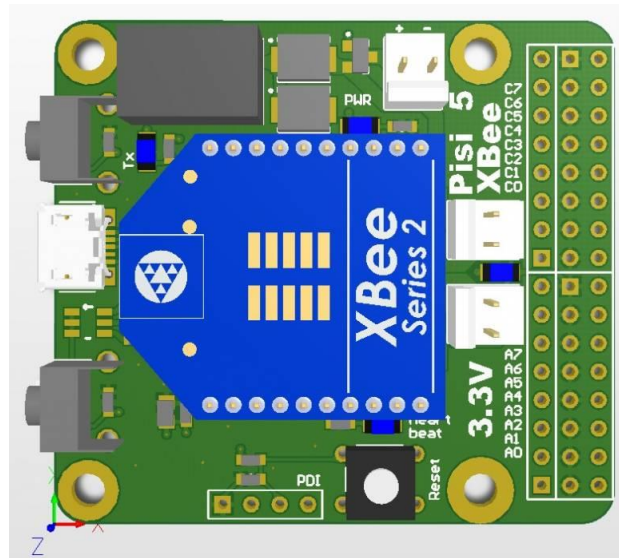
Funktsionaalsus	Vajaminev	Vana
Ühendusviis teiste robotitega	Kokku ühendatavus viikudega	Kokku ühendatavus viikudega
Mikrokontroller	32 bit	8 bit
Säilmälu (Flash)	Vähemalt 32 kB	32 kB
ADC sisendid	8	8
PWM taimerid	3	3
Enkoodri taimerid	2	2
USB 2.0	1	1
UART	1	1
I2C	1	1
GPIO	6	6

1.1 Olemasoleva lahenduse kirjeldus

Olemasolev robotplatvorm on välja arendatud endiste Robotiklubi liikmete poolt ning järgmised peatükid kirjeldavad täpsemalt millest platvorm koosneb ja milline on selle tööpõhimõte.

1.1.1 Pisi-Xbee

Kontrollerplaat nimega Pisi-Xbee eksisteerib Robotiklubis juba pikemat aega ning sellest on tehtud varasemalt mitu versiooni. Pisi-Xbee on multifunktsionaalne roboti kontrollerplaat, mille põhieesmärk on juhtida Pisi-Bot robotit, kuid mida on samuti võimalik kasutada teiste robotite juhtimiseks. Teades Pisi-Xbee viikude paigutust on võimalik arendada robot, mis saab ära kasutada kontrollerplaadi võimekust. Joonisel 1.1 on näha, milline nägi välja eelmine Pisi-Xbee versioon ning milline oli viikude paigutus. Mikrokontrolleri perifeeriatega ühenduvad viigud on kolmest järjestikusest viiguveerust kõige vasakpoolsemad. Joonisel on need nummerdatud tähistustega C0-C7 ja A0-A7. Nende viikude kõrval on joonisel veel näha kahte veergu, millest keskmine on 3,3 V pinge ja parempoolne on maandus. Et Pisi-Xbee ühtiks eelmise versiooniga ning selle paigutuse baasil loodud robotitega, peavad olema igal plaadil täpselt samal asukohal eelnevalt mainitud viiguread ning toite- ja mootori väljundviigud. Toite ja mootori ühendused on joonisel tuntavad valgete aluste järgi ning kõik on kahe ühendusega.



Joonis 1.1 Pisi-Xbee 5 3D mudel [1].

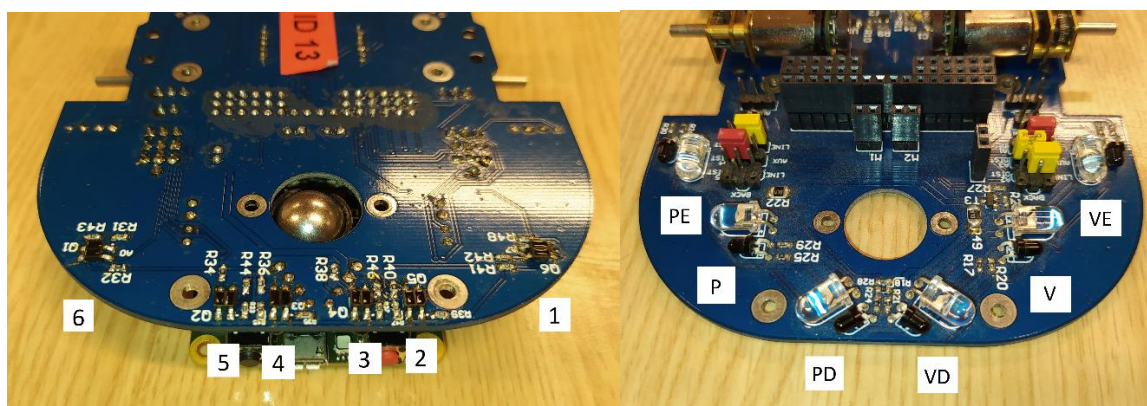
Plaadi küljes on erinevat värvi LED-id (*light emitting diode*), et näiteks näha kas toide on olemas või andes infot raadio kommunikatsiooni kohta. Samuti on olemas RGB LED, mis on kasutaja poolt programmeeritav ja Xbee pesa, mis võimaldab kasutada sobiva ühendusega raadiomoodulit (XRF, Xbee). Kontrollerplaadil on veel kuni kahe mootori kontrollimiseks topelt H-sild mootori muundur, IMU (*inertial measurement unit*), roboti orientatsiooni määramiseks, pingeregulaator, õige toitepinge saavutamiseks ja mikrokontrolleriks Microchipi ATxmega32A4U [2]. Lisaks akule on võimalik toita plaati mikro-USB pistikust tuleva 5 V pingega ja sealt kaudu ka mikrokontrollerit programmeerida.

1.1.2 Pisi-Bot

Pisi-Xbee on üks pool Pisi-Bot robotist, mis on multifunktsionaalne robot. Robotit kasutatakse eri projektidel tema raadio võimekuse pärast ja sellel on kahe Robotexi võistluse funktsionaalsused tagatud. Pisi-Botil on olemas võimekus läbida labürindi rada, kasutades sellel olemasolevaid kaugusandureid ja intertsiaalandurit ning samuti joonejärgimiserada, kasutades joonelugemisandureid. Tavaliselt aga arendatakse võistlustel osalemiseks, kindlale rajale spetsialiseeritud robotid.

Samuti on üks Pisi-Boti potentsiaalsetest eesmärkidest kasutada neid Robotiklubi läbiviidavatel kursustel, kus õpetatakse olemasoleva roboti programmeerimist. Lisaks kasutatakse Pisi-Bote veel erinevates projektides, kus on vaja lihtsasti kontrollitavaid liikuvaid roboteid.

Joonisel 1.2 on näha milline näeb välja Pisi-Bot ilma kontrollerplaadita ja antud juhul ka ilma ratasteta. Jooniselt on samuti näha lühenditega, kus paiknevad joone ja kaugusandurid. Jooneandurid on plaadi alumisel ja kaugusandurid ülemisel poolel.



Joonis 1.2 Pisi-Boti jooneandurite (vasakul) ja kaugusandurite (paremal) positsioonid.

V – vasak küljele

VD – vasak diagonaal

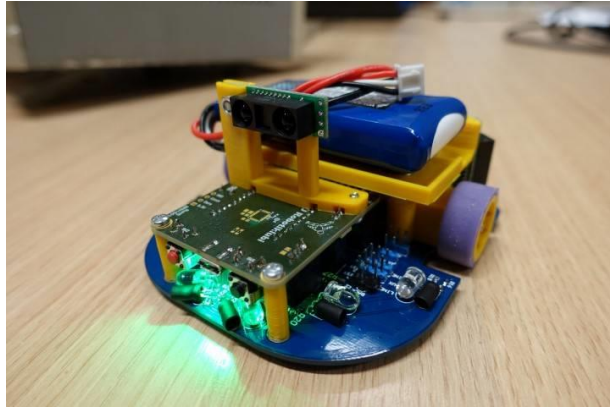
VE – vasak ette

PE – parem ette

PD – parem diagonaal

P – parem küljele

Hetkel kasutusel olev Pisi-Bot on liikumisvõimeline kahe mootori abil, millele on otseselt ühendatud rattad ning kolmandaks toetuspunktiks on roboti eesotsas paiknev kuul. Mootorite külge on ühendatud enkoodrid, mille abil on võimalik mootoreid täpsemalt kontrollida saades infot kiiruse, kiirenduse ja läbitud distantsi kohta. Pisi-Boti külge on ühendatud IR (*infrared*) kaugusandurid, mida on võimalik kasutada labürindi rajal seinte tuvastamiseks. Veel on roboti küljes joonejärgimisandurid, et läbida joonejärgimise võistlusradu ja piipar ehk lihtne helisignaali generaator, mis on läbi PWM (*pulse-width modulation*) signaali kasutaja poolt programmeeritav. Milline näeb Pisi-Bot robot välja kokkupandult koos Pisi-Xbee kontrollerplaadi ja akuga, on näha jooniselt 1.3.



Joonis 1.3 Pisi-Bot robotplatvorm [3].

1.2 Probleemid olemasoleva lahendusega

Peamised põhjused, miks eelnev kontrolleraud tuleb välja vahetada:

1. Mikrokontrolleri võimekus
2. Mikrokontrolleri arendusvõimalused
3. USB toiteahela ülevoolu kahjustus

Esimeseks väljavahetamise põhjuseks on mikrokontroller, mis Pisi-Xbee plaadil kasutusel on. Tegemist on Microchipi (varasemalt Atmel) kaheksa bitise mikrokontrolleriga ATxmega32A4U. Kaheksa bitised mikrokontrollerid, ei ole nii võimekad kui enamused tänapäevaseid 32 bitised mikrokontrollerid. Nende puhul võib mikrokontrolleri opereerimiskiirus aeglustuda, kuna tihti on vaja anduritest loetavate muutujatega teha palju kalkulatsioone. Kui tegemist on kaheksa bitise protsessoriga, siis tuleb suuremaid väärtuseid kirjutada mälusse mitmesse kaheksa bitisesse registritesse, mis nõuab kalkulatsioonideks rohkem operatsioone ja kokkuvõttes võtab rohkem aega. Vanal mikrokontrolleri protsessori kiirusel otseselt viga ei ole (maksimaalselt 32 MHz), aga kaheksa bitilise loogika kasutamine teeb opereerimise aeglasemaks. Mikrokontrolleri säilmälu on hetkel 32 kB, mida võiks samuti rohkem olla.

Teiseks miinuseks on vanal versioonil ATxmega mikrokontrolleri arendusvõimalused. Antud seeria mikrokontrollerid ei ole toetatud Arduino IDE-ga (*Integrated development environment*) töötamiseks nagu on paljud ATmega mikrokontrollerid. See teeb mikrokontrolleri programmeerimise palju keerulisemaks eriti veel siis, kui enamused Robotiklubi liikmetest ei ole kunagi sellisel tasemel Microchipi mikrokontrollerite programmeerimisega kokku puutunud. STM mikrokontrolleritega on seevastu enamik Robotiklubi liikmetest kokku puutunud, kuna neid kasutatakse juba Robotiklubi

baaskursusel, mille läbimine on kohustuslik, et Robotiklubi liikmeks saada. STM mikrokontrolleritel on väga võimekas arenduskeskkond ja HAL (*Hardware Abstraction Layer*), kus erinevate mikrokontrolleri perifeeriade algseadistamine ja kasutamine on lihtsamaks tehtud. STM mikrokontrolleri puhul on võimalik teha kogu algseadistus ära STM32CubeIDE tööriista abil, kus tuleb lihtsalt sisse lülitada vajaminevad funktsionaalsused ja algseadistustega kood genereeritakse automaatselt [4].

Kolmandaks uue kontrollerplaadi loomise põhjuseks oli USB liidese maha põlemine, mida põhjustas spekulatsioonide kohaselt toite võtmine läbi USB, kui aku ei olnud robotiga ühendatud. Mootorite tõttu läheb tarbitav vool kõrgeks ning plaadile ühendatud USB komponent ei kannata seda välja. Teoreetiliselt, USB toite puhul ei tohiks mootorite kontroll üldse töötada, kuna sealt ei jõua vool mootori muunduri varustusviiguni. Küll aga põlesid sellest olenemata aeg-ajalt USB komponendid maha. Spekulatsioonide põhjal jõuab USB toide ikka mootoriteni läbi muunduri loogikapinge viigu. Selle vältimiseks vahetatakse välja mootori muundur ja implementeeritakse lahendus, kus muundurini ei jõuaks juhtsignaalid, kui aku ei ole robotiga ühendatud.

1.3 Turul saadaolevad lahendused

Tänapäeval on müügil erinevaid roboteid, mida oleks samuti võimalik sarnastel eesmärkidel kasutada. Peamised funktsionaalsused, mis peaksid olema roboti puhul tagatud on:

1. Liikumisvõime
2. Kaugusandurid
3. Joonejärgimisandurid
4. Raadio võimekus.

Näiteks müüb Makeblock mBot roboteid, mida on võimalik kontrollida läbi *bluetooth*'i ning millel on ka joonejärgimis sensorid ja üks ultraheli kaugussensor [5]. Sellel robotil on sarnased võimekused Pisi-Botiga, aga kahjuks ei ole ainult ühe kaugussensoriga võimalik labürindi rada läbida. MBot roboteid omab ka Robotiklubi, aga peamiselt on need mõeldud laste õpetamiseks. Selliste robotite programmeerimine on tehtud väga lihtsaks. Nende puhul on kasutusel Scratch programmeerimiskeel, mis on kõrgetasemeline, visuaalne programmeerimismeetod. Robotite programmeerimiseks tuleb lihtsalt lohistada värvilisi kaste üksteise otsa, et panna neid tegema, mida soovid. Kuna robotite programmeerimine on tehtud nii lihtsaks, siis ei ole nende juures paindlikkust kasutada roboteid eri otstarvetel nagu on otse mikrokontrollerit

programmeerides. Seetõttu kasutabki Robotiklubi mBot-e ainult demoesinemisteks ja laste õpetamiseks.

Pisi-Botiga sarnaste võimekustega roboteid müüakse veel näiteks Sumozade roboti e-poes [6]. Seal on võimalik leida erinevaid joonejärgimis- ja *folkrace* radade läbimiseks loodud roboteid, mis baseeruvad Arduinodel. Plussiks nende robotite puhul on Arduinode kasutamine. Arduino baasil töötavad robotid on väga kasutajasõbralikud algajale programmeerijale, kellega on Robotiklubis peamiselt tegemist. Nende puhul ei ole tegemist täiesti laste tasemel programmeerimisega nagu on mBot-ide puhul. See tähendab, et robotite programmeerimine on paindlik, aga samas piisavalt lihtne. Probleemiks aga antud juhul on see, et robotitel on kas ainult joonejärgimis või takistuste vältimise võimekus, ehk ei ole olemas kombineeritud varianti. Samuti puudub nende juures inertsiaalidur, mida on tarvis labürindi raja läbimiseks. Kõige odavamad variandid nendest on 50\$, millele lisandub veel transpordi kulu. Bluetooth võimekuse tagamiseks tuleb raadiomoodul eraldi veel juurde osta. Pisi-Botide ise tootmine on küll mõnevõrra odavam, aga tegemist ei ole siiski kõige kallimat sorti robotitega, mis teeb nendest üpris hea alternatiivi. Mainitud robotite võimekused on võrdlemiseks välja toodud tabelis 1.3.

Tabel 1.3 Pisi-Boti ja turul saadaval olevate lahenduste võrdlus [3][5][6].

Funktsionaalsus	Pisi-Bot	mBot	Sumozade
Jooneandurid	6	2	0
Kaugusandurid	6	1	2
Raadiovõimekus	✓	✓	✓
Mootorid	2	2	2
Enkoodrid	2	0	0
Inertsiaalidur	1	0	0

Hetkel, kuna Pisi-Xbeel on omad puudused, on hakanud Robotiklubi baseeruma kõiki oma võistlusroboteid STM mikrokontrollerite Nucleo arendusplaatidele. Nucleod võimaldavad kasutada ära STM mikrokontrollerite funktsionaalsusi, ilma et oleks vajadust disainida eraldi trükkplaati. Tavaliselt ühendatakse Nucleo, läbi juhtmete, andurite ja mootoritega, mille lõpptulemus ei ole suure juhtmerägastiku tõttu kõige elegantsema välimusega. On ka võimalik disainida oma trükkplaat, kuhu ühendada andurid, Nucleo plaat ja mootorid, mis on keerukam variant, kuid siiski lihtsam, kui arendada trükkplaat Nucleo asemel puhta mikrokontrolleri kiviga. Arvesse aga tuleb võtta, et enamus Robotiklubi poolseid võistlejaid on algkursuse osalejad, kellel üldiselt

puudub varasem kogemus robotite arendamisest ehk tõenäoliselt nemad enda trükkplaati disainima ei hakka.

2 KOMPONENTIDE VALIK

Robotplatvormi arendamisega kaasneb olemasolevate komponentide analüüs, mille alla kuuluvad kogu Pisi-Boti põhikomponendid. Analüüsi põhjal otsustatakse, kas komponendid tuleb välja vahetada või sobivad need ka edaspidiseks kasutamiseks.

2.1 Mikrokontroller

Nagu eelnevas peatükis sai mainitud, ei ole olemasolev mikrokontroller nii heal tasemel nagu võiks. Peamisteks probleemideks eelmisel versioonil olid halvad arendusvõimalused ja et tegemist oli kaheksa bitise mikrokontrolleriga. Kuna Robotklubis kasutatakse juba algkursusel STM mikrokontrollereid, et tudengitele programmeerimist õpetada, siis paljud liikmed on nende mikrokontrollerite arendamisega juba tuttavad. STM mikrokontrollerid on enamasti kõik 32-bitised ja neil on hea arenduskeskkond, kus on võimalik mikrokontroller lihtsasti algseadistada vastavalt oma vajadustele.

Mikrokontrolleri valikul tuleb vaadata, et lisaks eelnevatele puudustele, oleks tagatud võimekus, mis eksisteeris varasemal versioonil. Vajaminevad ja vana versiooni võimekused, mis puudutasid ka mikrokontrollerit on välja toodud tabelis 1.1. Kasutusele jääb samade sensorite arvu tõttu, sama arv ADC (*Analog-to-digital converter*) kanaleid ja mootorite arvu tõttu on vaja sama palju taimereid, mis on võimelised PWM signaali genereerima. Samamoodi on uuel versioonil tarvis I2C liidest IMU-ga suhtlemiseks, UART liidest raadiomooduliga suhtlemiseks, USB ühendust toitmiseks ja kommunikatsiooni jaoks, veel ühte taimerit piipari kontrollimiseks PWM signaaliga ning GPIO viike, nii sisenditena nupu vajutuste lugemiseks, kui ka väljunditena RGB LED-i juhtimiseks.

Õnneks on tänapäeva mikrokontrollerid juba nii arenenud, et minimaalsed vajaminevad omadused on enamikel neist tagatud. Näiteks on sobilikud järgnevad kolm STM mikrokontrollerit, mis on kõik erinevate seeriade mudelid ning mille peamised funktsionaalsused on toodud välja tabelis 2.1.

Tabel 2.1 Sobilike mikrokontrollerite funktsionaalsuste võrdlustabel [7][8][9].

Funktsionaalsus	STM32L151C6T6A	STM32F303CB	STM32G473CB
ADC kanalid	14	15	20
I2C	2	2	4
UART	3	5	5
USB	1	1	1
PWM kanalid	4	22	32
Flash	32 kB	128 kB	128 kB
Max. protsessori sagedus	32 MHz	72 MHz	170 MHz

Kuna mikrokontrollerite kättesaadavus, eriti STM mikrokontrollerite, ei olnud kiibi puuduse tõttu tellimise hetkel kõige parem, siis oli suureks teguriks mikrokontrolleri valikul selle kättesaadavus. Valituks osutus antud juhul STM32L151C6T6A, kuna see täitis kõik vajalikud nõuded. Ainuke koht mille üle võib tehtud valiku puhul kurta on mikrokontrolleri säilmälu, mis võib keerulisemate programmide puhul jääda väikseks. Õnneks on samast seeriast olemas täpselt sama jälgpinnaga ja suurema mälu variante nagu STM32L151C8 ning STM32L151CB, mille ainsaks erinevuseks ongi mälu suurus [10][11]. Viigujaotus ja ülejäänud perifeeriad on täpselt samad ning need variandid on nüüdseks ka turul saadaval. Selliselt on võimalik tulevikus trükkplaadil mikrokontroller vähese vaevaga ümber vahetada.

Põhiomadustelt on uus STM mikrokontroller väga sarnane vana mikrokontrolleriga. Mõlemal on 32 kB säilmälu ja on sama maksimaalse protsessori kiirusega 32 MHz.

Esialgu osutus valituks hoopis STM32F303CB, mille järgi alustati ka elektroonikaskeemide koostamist. Tegemist on igat pidi võimekama mikrokontrolleriga, aga kahjuks ei olnud komponent tellimise hetkel kiibipuuduse tõttu kättesaadav.

2.2 Ajamid

Pisi-Boti kasutuseesmärkide ja väikeste mõõtmete tõttu, ei ole tarvis väga võimekaid mootoreid. Roboti põhieesmärk ei ole otseselt võistlustest osa võtmine, mistõttu ei ole suurte kiiruste saavutamine erilisel tähtsusel. Vana roboti versiooni mass on ligikaudu 120g (Pisi-Bot + Pisi-Xbee). Olenevalt sellest, mis suuruses aku robotiga ühendatakse (väike = ~30g, suur = ~60g), jääb kogumass alla 200 grammi. Hetkel kasutusel olevad mootorid on Pololu 50:1 ülekandega DC mootorid, mis on võimelised ilma koormuseta saavutama 650 pööret minutis [12].

Et leida kas kasutusel olevad mootorid on piisavalt võimelised saavutama soovitud kiirust ja kiirendust, tuleb leida kui võimsaid mootoreid on tarvis. Arvestades, et kasutusel on suurem aku, tuleb roboti massiks umbes 180 grammi.

Kuna roboti mass jaguneb peamiselt kahe ratta peale ja eesmisele tugipunktile palju raskust ei mõju, jagatakse lihtsustamise jaoks roboti kaal kahega, et saada teada kui palju raskust mõjub ühele rattale. Kuna roboti kogu mass on 180g, siis ühele rattale mõjub 90g.

Võimsuse leidmiseks kasutatakse valemit:

$$P = \omega * T, \quad (2.1)$$

kus ω – nurkkiirus (rad/s),
T – pöördemoment (Nm).

Et leida pöördemoment, tuleb teada ratta raadiust (r) ja jõudu (F), mis leitakse vajamineva kiirenduse kaudu. Vajaminevaks kiirenduseks võetakse $a = 1 \frac{m}{s^2}$. Teades ratta raadiust (r = 0,0125m), saab leida kiirendamiseks vajamineva jõu:

$$F = m_{\frac{1}{2}} * a = 0,09 * 1 = 0,09 N. \quad (2.2)$$

Pöördemoment leitakse ratta raadiuse ja jõu kaudu valemiga:

$$T = r * F = 0,0125 * 0,09 = 0,00125 Nm. \quad (2.3)$$

Teiseks, et leida nurkkiirus on vaja teada ratta raadiust (r) ja roboti liikumiskiirust (v) ning see leitakse valemiga:

$$\omega = \frac{v}{r} = \frac{0,7}{0,0125} = 56 \frac{rad}{s}. \quad (2.4)$$

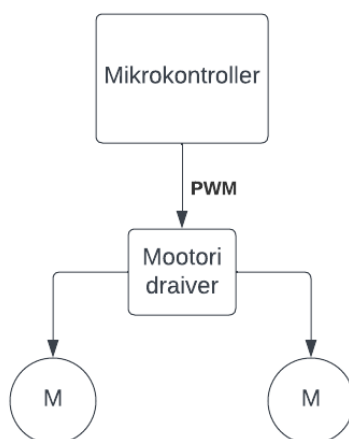
Nüüd kasutades leitud andmeid, saab valemi 2.1 kaudu leida vajaliku võimsuse ühe mootori jaoks:

$$P = 80 * 0,00125 = 0,07 W. \quad (2.5)$$

Kuna kasutusel on kaks mootorit, tuleb antud võimsus korrutada kahega ehk mootoritele kulub kiirendamiseks kokku 0,14 W. Antud kalkulatsioonide puhul on tegemist ideaalse stsenaariumiga, ehk ei ole arvestatud esinevate kadudega. Et mootorid oleksid kindlasti võimelised algandmetena välja toodud kiirendust ja kiirust saavutama, tuleb võtta mootorid mis on võimsuselt piisava varuga. Kuna Pisi-Botil on

juba kasutusel 1,2 W võimsusega mootorid, mis on võimelised antud kiirendust ja kiirust saavutama, siis ei ole vajadust uusi mootoreid soetada.

Mootorite kontrollimiseks on vajalik ka mootori muundur, mille abil mootoreid oleks võimalik juhtida edaspidi ja tagurpidi. Kuna meil on tarvis kontrollida kahte mootorit eraldi, et keeramine oleks võimalik, siis osutus valituks topelt H-sild mootori muundur DRV8835 [13]. Selle abil on võimalik kontrollida mootoreid sõltumatult üksteisest ja muuta signaalide kaudu pöörlemis-suunda. Muunduri valikul oli tähtis, et mootorite toitepinge ei ületaks lubatud toitepinget ja et loogikapinge oleks samuti õiges suurusjärgus. Antud muunduri maksimaalne lubatud vool on 1,5 A kanali kohta, mis on täpselt sama nagu mootorite maksimaalne vool. Muunduril on olemas veel liigvoolu kaitse, mis väldib komponendi kahjustada saamist, kui vool peaks lubatust suuremaks minema. Kui aga sillad ühendada paralleelselt, oleks võimalik kontrollida ühte mootorit, kuni 3 A. Mikrokontrolleri, muunduri ja mootorite vaheline ühendus on lihtsustatud kujul toodud joonisel 2.1.



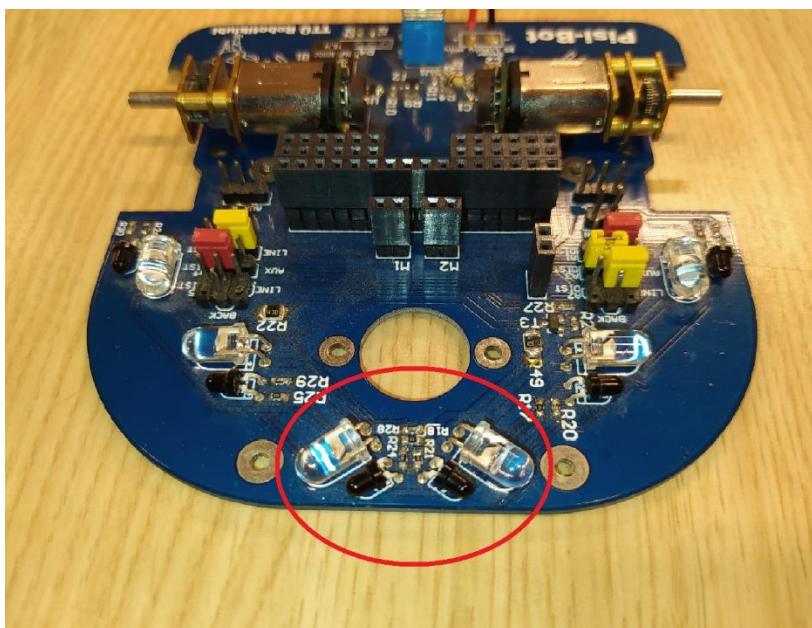
Joonis 2.1 Mootorite juhtimise loogika plokk skeem.

2.3 Andurid

Et Pisi-Bot oleks võimeline läbima erinevaid võistlusradu nagu labürint, *folkrace* ja joonejärgimine, on robotil tarvis sobilikke andureid. Nii labürindi, kui *folkrace* radade läbimiseks on vaja kaugusandureid ja joonejärgimise raja läbimiseks on vaja joonejärgimisandureid. Samuti on väga kasulik inertsiaalandur (IMU – *inertial measurement unit*), mille abil on võimalik määrata roboti liikumiskiirendusi x , y ja z -telgedel ning samuti leida roboti pööramist ümber kõigi kolme telje.

2.3.1 Kaugus ja seinte tuvastus

Vanal roboti versioonil olid kasutusel isetehtud IR sensorid, kus kontrollerplaat käivitas GPIO väljundi abil IR LED-ide ning vastuvõtjaga loeti tagasipõrkunud kiirgust, mis edastas loetud kiirguse pingena Pisi-Xbee ADC kanalisse. Selle meetodi abil on võimalik määrata lühikestel distantsidel, tuvastatavate objektide kaugusi. Kuna roboti kasutuseesmärkidel ei ole kaugusandurite täpsus kõige olulisem ja loetavad väärtused saab sensoritest lihtsasti kätte, siis hetkeseisuga otsustati neid mitte välja vahetada. Kaugusandurite asukoht Pisi-Botil on näidatud joonisel 2.2.



Joonis 2.2 Roboti eesmised diagonaalsed IR kaugusandurid.

2.3.2 Inertsiaalandur

Eelmisel Pisi-Xbee versioonil oli IMU, mis koosnes kolme teljelisest kiirendusandurist, koos kolme teljelise güroskoobiga. Selline andur on väga kasulik erinevates roboti operatsioonides, kus on vaja olla teadlik roboti manöövrimest. Näiteks on see vajalik Robotexi labürindi rajal ja samuti *Folkraace* ringrajal, kus tuleb olla kindel, et robot läbib rada õiget pidi. Kuigi peamiselt kasutatakse seda tüüpi robotitel ainult güroskoopi, pööramissuuna määramiseks, siis seda tüüpi inertsiiaalandur on võimeline rohkemaks, andes andmeid ka kiirenduste kohta. Valituks osutus uus sarnaste võimekustega inertsiiaalandur, mida kasutati varasemalt, milleks on TDK Invensense ICM-20689 [14]. Andmete kätte saamine on andurist võimalik kas läbi I2C või SPI kommunikatsiooni. Valituks osutus andur seetõttu, kuna antud IMU on Robotiklubis varasemalt kasutatud ning sellel on olemas valmis kirjutatud draiver STM mikrokontrolleril kasutamise jaoks.

2.3.3 Joonejärgimine

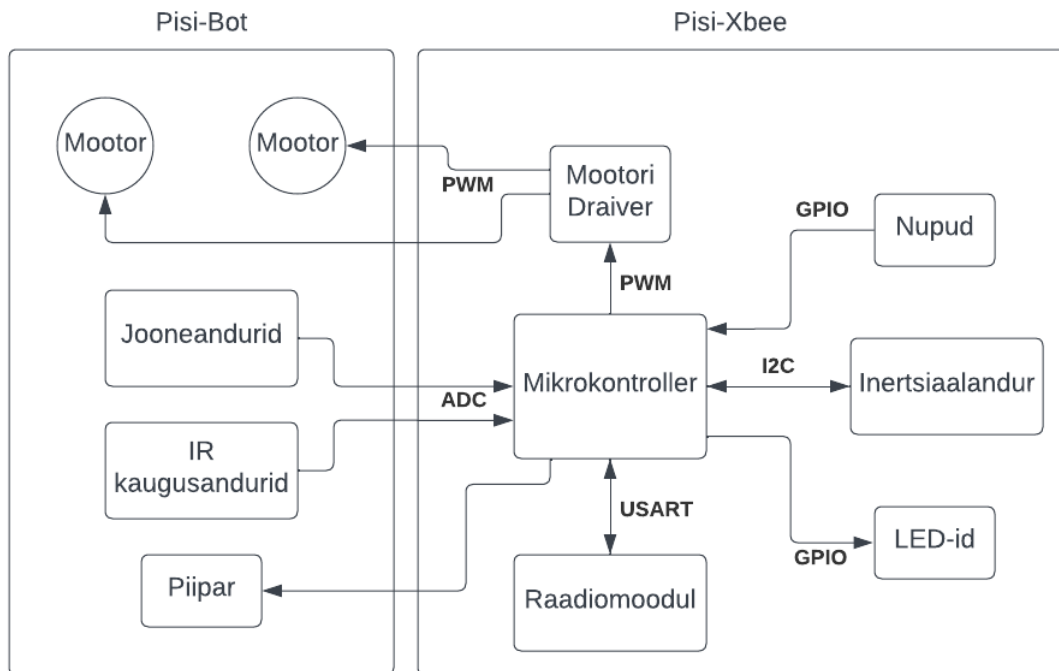
Joonejärgimise võimekuse tagamiseks on robotil tarvis IR joonejärgimis andureid, et teha vahet tumedal joonel, mida mööda robot peab sõitma, ja valgel ümbritseval pinnal. Sellised andurid töötavad lihtsal põhimõttel, kasutades kaugusanduritele sarnaselt LEDe, mis kiirgavad infrapunast valgust maapinnale ja lugedes tagasi põrkunud valguse intensiivsust. Sõltuvalt loetud intensiivsusest on võimalik mikrokontrolleri ADC kanalitega määrata pinna tumedust. Kui pind on tume, on tagasipõrkunud kiirgus nõrgem kui valget värvi peegeldavama pinna puhul. Sensorid, mida Pisi-Bot kasutab, töötab piisavalt hea täpsusega ning kuna võistlusradadel on joone kontrast taustpinnaga väga suur, siis ei ole vajadust paremate sensorite peale üle minna.

2.3.4 Enkoodrid

Enkoodrid on andurid, mis annavad infot mootorite pöörlemise kohta, mille abil on võimalik leida infot läbitud vahemaa, kiiruse ja ka kiirenduse kohta. See info võib kasuks tulla näiteks juhul, kui ei kasutata pidevat mootori kontrolli, ehk pannakse robot läbima fikseeritud vahemaid. Sellisel juhul võib ka juhtuda, et mõlemad mootorid ei tööta täpselt ühtemoodi ning robot võib hakata teelt kõrvale kalduma. Enkoodritega on võimalik seda vältida, jälgides mootorite pöörlemiskiiruseid ning tekkivat viga kompenseerida. Enkoodrid, mis vanal roboti versioonil kasutusel olid, täitsid oma tööülesannet väga hästi, mistõttu ei ole neid vaja välja vahetada.

3 KONTROLLERPLAADI KOOSTAMINE

Lihtsustatud plokkskeemina on näha jooniselt 3.1 kuidas roboti erinevad komponendid omavahel ühendatud peavad olema. Roboti terviku moodustavad Pisi-Boti andurite ja täiturite trükkplaat ning Pisi-Xbee kontrolleraud. Kõik sisendid jõuavad mikrokontrollerini, mis roboti ajuna annab väljundeid ja kontrollib täitureid. Komponentide vahelised ühendused on kujutatud lihtsustatud kujul.



Joonis 3.1 Roboti elektroonika ülesehitus.

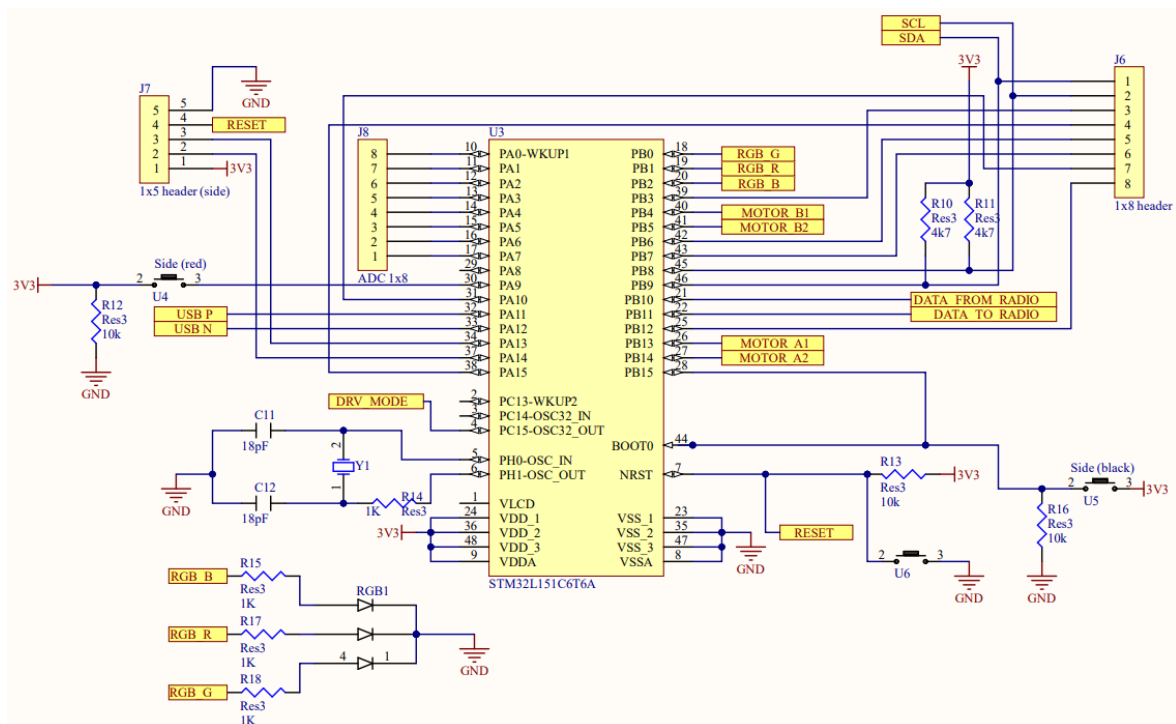
3.1 Elektriskeem

Pisi-Xbee trükkplaadi uue versiooni loomiseks kasutati Altium Designer tarkvara [15]. Trükkplaadi loomisel, tuli alustada elektroonikaskeemide loomisest. Selleks, et skeemid luua, on vaja teada mis põhikomponente roboti juures vaja läheb. Kuna mikrokontrolleri kätte saamise garanteerimiseks, telliti see esimesena ära, sai alustada mikrokontrollerist ja seda ümbritsevast. Et teada, kuhu viikudesse, mis ühendused jooksma peavad tuli esmalt konfigurereida STM arenduskeskkonnas kõik perifeeriad (funktsionaalsuse moodulid), mida on näha tabelis 3.1.

Tabel 3.1 Mikrokontrolleri sisendite ja väljundite tabel.

Viik	Kasutus	Perifeeria
PA0-PA7	Joone- ja kaugusandurid	ADC
PA9, PB15	Kasutaja nupud	GPIO
PA10	Kaugusandurite IR LED	GPIO
PA11, PA12	USB kommunikatsioon	USB
PA13, PA14	Silur	SWD
PA15, PB3, PB6, PB7	Enkoodrid	Taimer
PC15	Muunduri seadistus	GPIO
PH0, PH1	Kella signaaligeneraator	RTC
PB0-B2	RGB LED	GPIO
PB4, PB5, PB13, PB14	Mootorite juhtimine	Taimer (PWM)
PB8, PB9	I2C suhtlus/IMU	I2C
PB10, PB11	Raadiomoodul	UART
PB12	Piipar	Taimer (PWM)

Kui kõikide viikude funktsionaalsused said määratud, mindi edasi mikrokontrolleerit ümbritseva elektroonika skeemi loomisega. Skeemi on näha jooniselt 3.2, mis pandi kokku vastavalt vajaminevatele funktsionaalsustele ja mikrokontrolleeri andmelehele. Andmelehest sai infot passiivide kohta, nagu näiteks kui suuri väliseid kondensaatoreid mikrokontrolleer vajab korrektseks toimimiseks. Samuti tuli kasuks Robotiklubi poolne Pisi-Xbee dokumentatsioon, kust sai teada milliste väliste viikudega peavad erinevad ühendused olema [1].



Joonis 3.2 Mikrokontroller ja selle ühendused.

Järgmistes alapeatükkides selgitatakse täpsemalt ära kõik peamised trükkplaadi elektronika osad.

3.1.1 Viigured

Joonejärgimis- ja kaugusandurite sisendid peavad ühenduma mikrokontrolleri ADC kanalitega, mis jõuavad kontrolleriplaadini läbi viigurea (komponent J8 Joonisel 3.2). Kuna Pisi-Boti ja Pisi-Xbee vahel on ühendus kaheksa anduriga, siis on vaja seadistada mikrokontrolleri sisendiks kaheksa ADC kanalit.

Tabel 3.2 Pisi-Xbee analoog ühendusviigud.

Pesa viigu nr	MCU ühendusviik	Jooneandur	IR kaugusandur	Sharp kaugusandur
8	PA0	1	-	-
7	PA1	2	V	-
6	PA2	3	VD	VD
5	PA3	-	VE	V
4	PA4	-	PE	P
3	PA5	4	PD	PD
2	PA6	5	P	-
1	PA7	6	-	-

V – vasak küljele
 VD – vasak diagonaal
 VE – vasak ette
 PE – parem ette
 PD – parem diagonaal
 P – parem küljele

Läbi teise viigurea (J6), ühenduvad mikrokontrolleriga enkoodrid. Kahe mootori küljes oleva enkoodri lugemiseks on vaja seadistada kaks erinevat taimerit, millel on võimalus kasutada kombineeritud kanaleid enkoodri režiimis. See võtab kasutusele neli erinevat viiku, ehk kaks kanalit ühe taimeri jaoks. Viimased ühendused selles viigureas on mõeldud GPIO väljundiks labürindi kaugusandurite LED-ide sisse lülitamiseks ja piipari kontrollimiseks PWM signaali teel. Piipar on lihtne helisignaali generaator, mis teeb heli vastavalt PWM kanalist tulevale signaalile. Mida suurem on täitetegur, seda kõrgem on heli.

Tabel 3.3 Pisi-Xbee digitaalsed ühendusviigud.

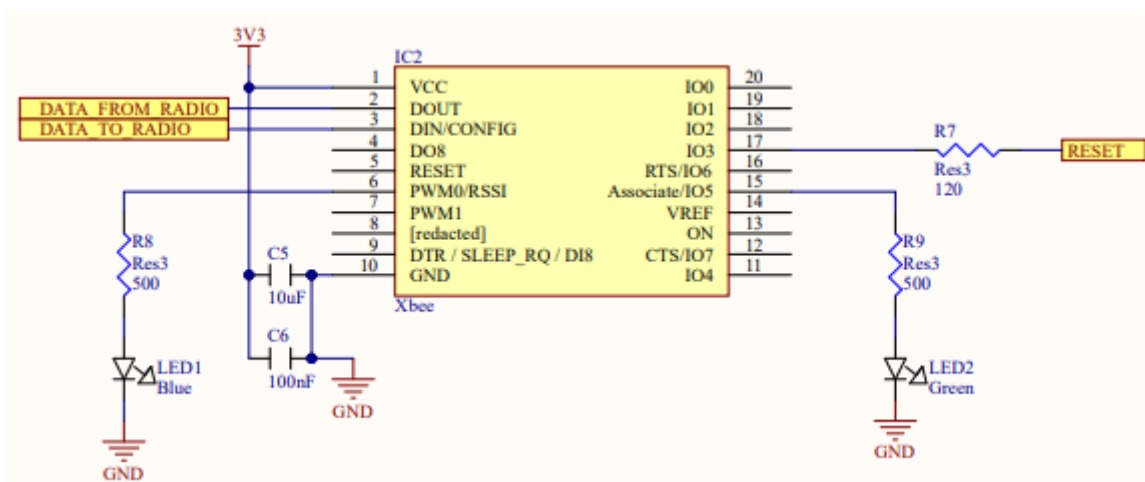
Pesa viigu nr	MCU ühendusviik	Kasutus
1	PB9	I2C (SDA)
2	PB8	I2C (SCL)
3	PB3	Enkooder 1
4	PA15	Enkooder 2
5	PB6	Enkooder 3
6	PB7	Enkooder 4
7	PA10	Kaugusandurite LED-de lülitus
8	PB12	Piipar

Kolmas viigurida mis on kasutusel (J7), on siluri ühendamiseks. Et kasutada SWD tüüpi silurit, on tarvis viite viiku, millest kolm ühenduvad mikrokontrolleri viikudega (SWCLK, SWDIO ja RESET). Teised on ühendused 3,3 V toite ja maandusega.

3.1.2 Andmeside ühendused

Ühendatava raadiomooduli kommunikatsiooniks on tarvis mikrokontrolleril üles seadistada UART (*universal asynchronous receiver transmitter*) moodul. UART jadaside kommunikatsiooniks on vaja ainult kahte liini vastuvõtja (RX) ja saatja (TX), mis ühenduvad skeemil mikrokontrollerist Xbee pesaga. Elektroonika skeemil on UART liinid

tähistatud teksti kastidega DATA FROM RADIO ja DATA TO RADIO, mis tähistavad mikrokontrolleri UART ühenduse RX ja TX liine. Xbee pesa on elektroonika skeemil kujutatud Xbee raadiomooduli komponendina, et näha kuhu täpsemalt mis ühendused jooksevad. Raadio mooduli elektroonika skeemi on näha jooniselt 3.3, mis on kokku pandud kooskõlas andmelehele [16].



Joonis 3.3 Raadiomooduli elektroonika skeem.

Teiseks kommunikatsiooni meetodiks on mikrokontrolleeril I2C, mille abil on võimalik suhelda Pisi-Xbee inertsaalanduriga või mõne muu seadmega. Kuna I2C liinid ühenduvad peale inertsaalanduri ka viiguritta, on võimalik seeläbi suhelda veel teiste ühenduvate anduritega. Pisi-Botil selleks vajadust ei ole, kuna andurid kus veel teist I2C suhtlust vaja oleks puuduvad. See võimekus on jäetud muudele robotitele, mis tahavad Pisi-Xbeed kontrollina kasutada. I2C liinide hõljumise vältimiseks on nende puhul kasutatud väliseid 4,7 kΩ üles tõmbe (*pull-up*) takisteid.

3.1.3 Kasutajaliides

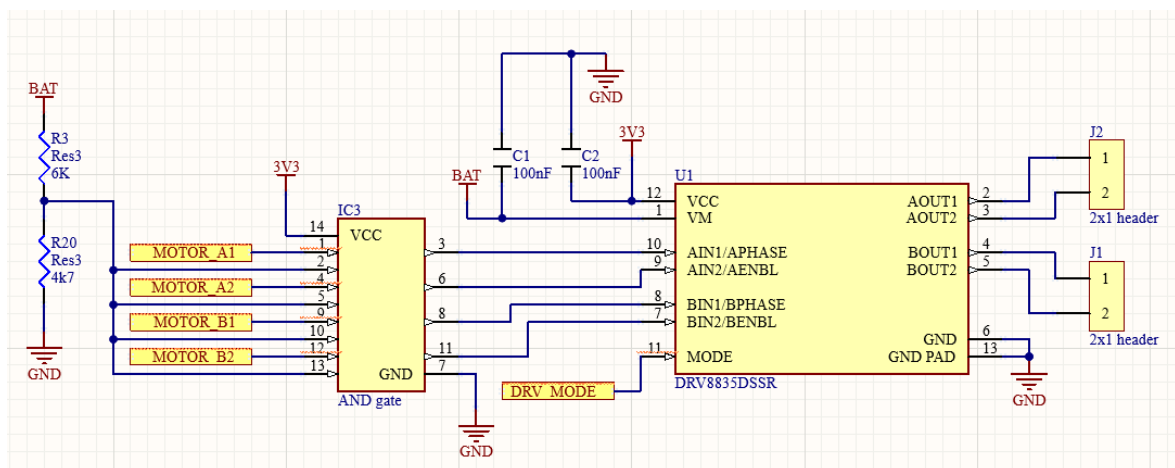
Kasutaja ehk tulevase programmeerija jaoks on kontrolleraalil üks RGB LED. Hetkel, kuna RGB viigud konfigureeriti GPIO-na, on võimalik genereerida kuut erinevat värvi. Seda sellepärast, et GPIO-d saavad olla ainult kas sisse või välja lülitatud olekus. Mikrokontrolleeril oleks olnud võimalus konfigureerida RGB-d kontrollivad viigud DAC (*digital-to-analog converter*) kanalitena, ehk kasutaja oleks saanud genereerida ükskõik millist värvi. DAC väljundi erinevus GPIO-st on reguleeritav väljundpinge. Selle abil saab genereerida väljundpinge vahemikus 0 - 3,3 V.

Nagu eelmisel Pisi-Xbee versioonil, tuleb kasutajale programmeerimiseks jätta kaks nuppu. Need tuleb STM arenduskeskkonnas seadistada GPIO sisenditena, et lugeda nupu olekut. Nuppe saab kasutada näiteks programmi käivitamiseks või mis iganes

muuks eesmärgiks. Kokkuvõttes tuleb kontrolleriplaadile kolm nuppu, koos taaskäivitus nupuga.

3.1.4 Mootori muundur

Mootorid ühenduvad kontrolleriplaadiga läbi kahe eraldi seisva 1x2 viigu ühenduse. Mootorite juhtimiseks on tarvis, sõltuvalt valitavast juhtimismeetodist, kas kahte PWM väljundit ja kahte GPIO väljundit või nelja PWM väljundit. Juhtimismeetodi info on võetud kasutatava muunduri DRV8835 andmelehest [13]. Opereerimismeetodit saab muuta läbi mikrokontrolleri GPIO, tõmmates MODE viigul pinge kas üles või alla. See ühendus on tähistatud joonisel 3.4 tähisega DRV MODE. Mikrokontrolleri mootori juhtimis väljundid ühenduvad läbi kaheksa sisendilise ja nelja väljundilise NING lüli, mootori muunduriga, mida on samuti näha joonisel 3.3.



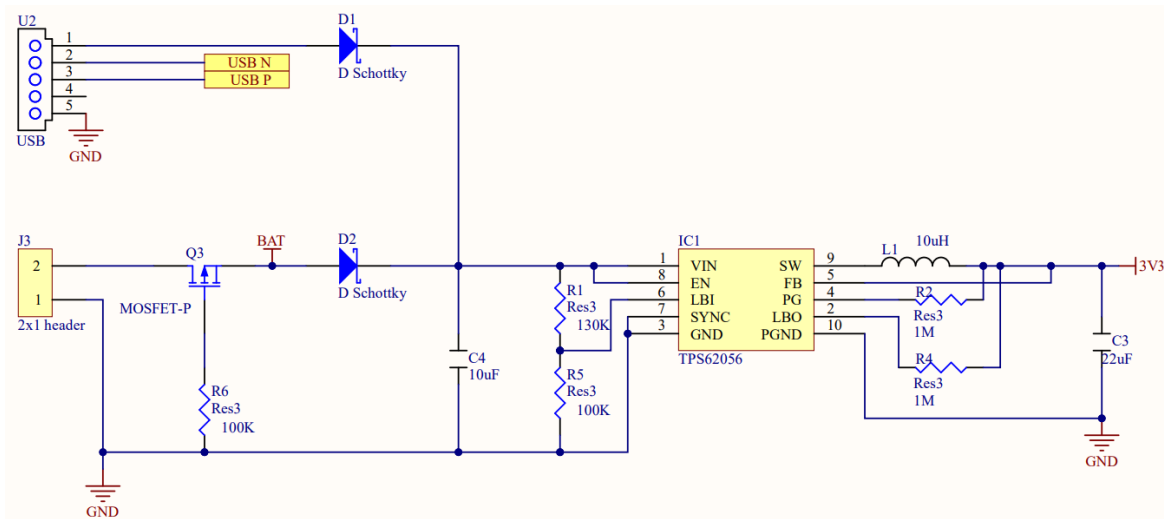
Joonis 3.4 Mootori muunduri elektriskeem.

Põhjus miks uuel Pisi-Xbee versioonil on pandud NING lüli muunduri ja mikrokontrolleri vahele on selles, et varasemalt, kui aku ei olnud ühendatud ja mootorid võtsid toidet läbi USB, siis hakkasid USB ja mikrokontroller viga saama. Nüüd on iga mootori juhtsignaali väljund sõltuvuses sellest, kas aku on ühendatud või mitte. Kui aku pinget pole, ei edastata ka mootori muundurisse ühtegi juhtsignaali. NING lülisse jõuab aku pinge läbi pingejaguri, sest soetatud lüli sisendpinge peab olema vahemikus 2 - 6 V, aga kasutatakse 8 V akut [17]. Pingejaguri jaoks valiti 6 k Ω ja 4,7 k Ω takistid, mille abil saadakse pinge suurusjärgus 3,3 V, kuid see sõltub veidi sellest, kui laetud aku on. Aku pinge otsustati muundada 3,3 V juurde, et see ühtiks kogu ülejäänud elektroonika loogika pingega.

3.1.5 Toiteahel

Kontrollerplaadil on kaks võimalikku toite allikat, milleks on USB (5 V) ja 1x2 viigurida, kuhu ühendub 8 V aku (J3). Kuna mikrokontroller vajab loogika jaoks 3,3 V, siis on vaja

pingeregulaatorit, mis muundaks sisendpinge sobilikuks. Valituks osutus TI TPS62056DGS pingeregulaator [18]. Tegemist on madaldava (*step-down*) pingeregulaatoriga, mis on võimeline muutma 3,3 V – 10 V sisendpinge 3,3 V pingeks. Kuna nii akust, kui ka USB kaudu tulev sisendpinge on mõlemad väiksemad kui 10 V, siis sobib see regulaator ideaalselt. Pingeregulaatori skeemi ühendusi on näha jooniselt 3.4.



Joonis 3.5 Toiteahela skeem.

Akust tulev pinge läbib ühte P-tüüpi MOSFET-i (Q3). Selle MOSFET-i eesmärk polaarsus kaitse, ehk kaitsta skeemi, selle eest kui aku toide ühendatakse valet pidi. Vanal Pisi-Xbee versioonil, oli seal MOSFET-i asemel ainult diod, mis takistas voolu liikumist valet pidi. Diode peal läheb aga palju energiat raisku võrreldes MOSFET-iga. USB-st tulev toide läbib schottky diodi, et vältida akust tuleva pinge jooksmist USB toitesse, sest seal on tegemist väiksema pingega. Samamoodi läbib ka akust tulev toide diodi, mis on asetatud aku pinge ühenduse järele, et vältida USB 5 V toite edastamist mootoritesse.

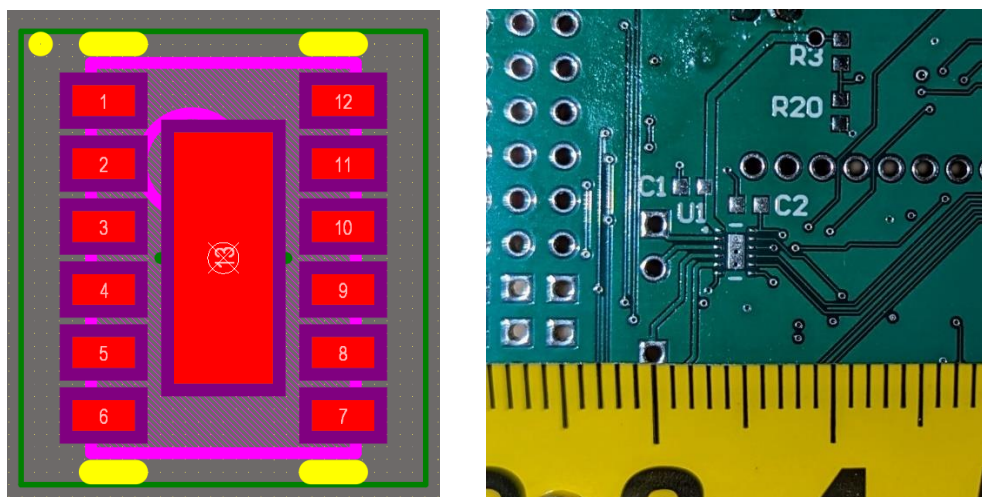
3.2 Trükkplaadi disain

Kui elektroonika skeemid said valmis, oli järgmiseks sammuks alustada trükkplaadi disainiga. Trükkplaadi disaini juures paigutatakse Altiumis kõik komponendid plaadile ja luuakse radadega ühendused kõikide komponentide vahel, mis elektroonikaskeemis lihtsustatud kujul loodi.

3.2.1 Komponentide jälgpinnad

Et trükkplaadi disainiga alustada, tuli esialgu kõikidele skeemis olevatele komponentidele määrata trükkplaadi jälgpinnad (inglise keeles *footprint*). Jälgpindade

külge joodetakse trükkplaadil tellitud komponendid. Jälgpinnad peavad ühtima tellitud komponentide mõõtmetega, kuna muidu elektriahel ei toimi. Jälgpindade puhul on tegemist väga väikeste mõõtmetega. Näiteks on kahe IMU jala ühenduse vahemaa üksteisest 0,25 mm. Milline näeb näiteks välja mootori muunduri jälgpind on võimalik näha jooniselt 3.6.



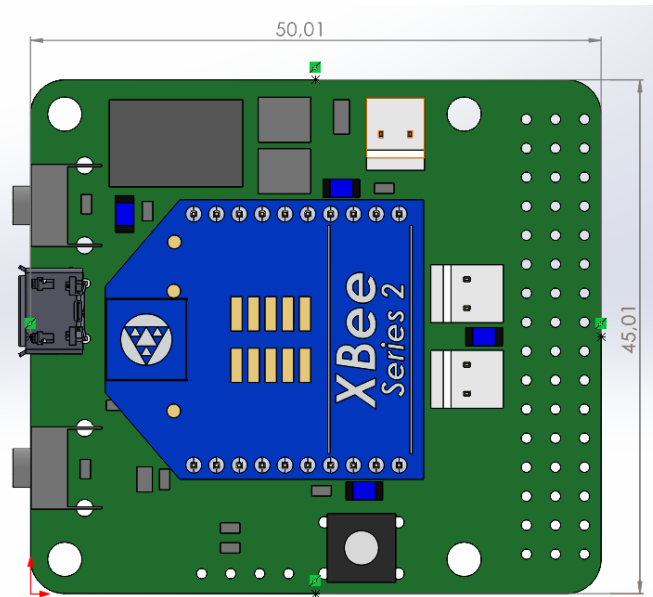
Joonis 3.6 Mootori muunduri jälgpind Altiumis (vasakul) ja trükkplaadil (paremal).

Lihtsamatel komponentidel nagu takistid, kondensaatorid ja viiguread, oli võimalik jälgpinnad leida Altiumi enda komponendi kogumist. Et kõikidele takistitele ja kondensaatoritele saaks ühed jäljed määrata, valiti e-poest kõik need komponendid ühe mõõduga (1,6 x 0,8 mm). Samuti oli võimalik osade keerukamate komponentide jälgpinnad leida kas Altiumi komponentide otsingust (*Manufacturer Part Search*) või snapEDA-st [19]. Kõiki komponente nende meetoditega ei olnud võimalik üles leida, mistõttu tuli osad jälgpinnad ise valmis teha. Altiumis on olemas tööriist, millega oli näiteks võimalik genereerida mootori muunduri ja veel mõne teise komponendi jäljed. Selleks tuli teada komponendi mõõtmeid, mis on leitavad andmelehtedest, ja tööriist teeb jälgpinnad ise valmis. Antud tööriista sai kasutada ainult juhul, kui valitud komponent oli koostatud teatud standardi järgi. Kui mõne valitud komponendi standardit Altium ei toetanud, tuli nende komponentide jälgpinnad täiesti nullist luua.

3.2.2 Trükkplaadi mõõtmed

Kui kõikidele komponentidele olid jälgpinnad määratud, sai edasi minna trükkplaadi disainiga. Alustada tuli trükkplaadi mõõtmete määramisest. Et uus Pisi-Xbee versioon ühenduks samamoodi Pisi-Boti külge nagu varasemalt, tuli leida vana versiooni mõõtmed eelnevate Robotiklubi liikmete poolt loodud 3D mudelist. Vana trükkplaat oli mõõtmetega 50 mm x 45 mm. Tähtsaimaks osaks, et trükkplaadid omavahel kokku

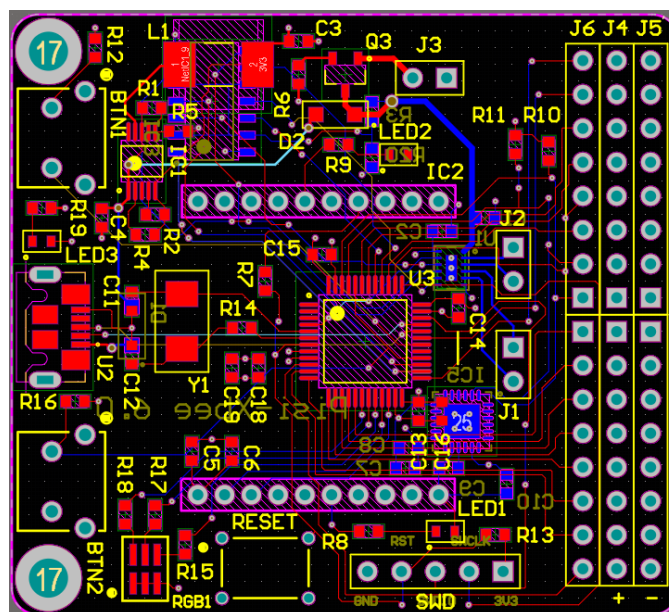
ühenduks, on viikudevahelised ühendused. Need tuli väga täpselt 3D mudelist leida, kuna viikude asetusel ei olnud lähtunud kindlast mustrist.



Joonis 3.7 Vana Pisi-Xbee versiooni, 3D mudel.

3.2.3 Trükkplaadi kihid

Kuna tegemist on üpris väikeste mõõtmetega trükkplaadiga, siis tuli kasutada komponentide ja radade jaoks mõlemat plaadi poolt. Enamus komponentidest läks trükkplaadi pealmisele poolele, aga töö lihtsustamiseks pandi IMU, mootori muundur, NING lüli ja kõik nendega ühenduvad passiivid, plaadi alumisele poolele. Trükkplaadi disaini koos komponentide ühenduste ja radadega on näha jooniselt 3.7.



Joonis 3.8 Trükkplaadi disain Altiumis.

Trükkplaat otsustati teha nelja kihiline, millest pealmine (punased ühendused) ja alumine kiht (sinised ühendused) on komponentide jaoks ning keskmised, ehk plaadi sisesed kihid, on mõeldud 3,3 V toite ja maanduse jaoks. Keskmiseid kihte kasutati samuti mõne üksiku ühenduse loomiseks, mis vajasis signaali tundlikkuse tõttu võimalikult otsest rada ja oleks muidu liiga keeruliseks läinud (näiteks USB andmeliinid). Trükkplaadi pealmisel kihil asetsevad Pisi-Botiga ühenduvad viiguread, mikrokontroller, mikro-USB pistik, nupud, RGB LED, XBee raadiomooduli pistik ja nende komponentidega ühenduvad passiivid. Nagu eelnevalt mainitud on ainsad komponendid, millel on kindlalt määratud positsioonid, Pisi-Botiga ühenduvad viigud. Kui need mõõtmed omavahel ei klapiks, ei oleks Pisi-Botiga antud kontrollerplaati võimalik kasutada. Ülejäänud komponentide paigutus ei pidanud olema nii täpne, aga tuli ikkagi silmas pidada, et nappudele, USB ja siluri ühendustele oleks võimalik ligi pääseda.

3.2.4 Komponentide vahelised ühendused

Enamik ühendustel, mis trükkplaadi disainimisel veeti, kasutati minimaalset raja paksust, milleks on 0,102 mm. Seda saab teha ühenduste puhul, mis ei vaja palju voolu, ehk kui tegemist on signaalidega. Rajad mis peavad kindlasti jämedamad olema, on mootoritest sõltuvad ühendused, sest mootorid on kogu kontrollerplaadi juures kõige suuremad voolu tarbijad. Sinna alla kuuluvad ühendused aku ja mootori muunduri vahel ning mootori muunduri väljundid. Mootoritega ühenduvate liinide puhul võeti paksuseks mootori muunduri viikude laiused, milleks on 0,25 mm. Kuna maksimaalne võimalik vool mootori kohta on 1,5 A, siis kasutades trükkplaadi raja kalkulaatorit, tuleb välja, et 0,25 mm raja laiusega ja standardse 0,035 mm raja paksusega on potentsiaalne temperatuuri tõus 34° C [20]. Jämedamat rada aga ei olnud võimalik vedada, kuna mootori muunduri enda ühendus on nii kitsas. Toiteliini puhul tuleb aga arvestada, et kahe mootori puhul on maksimaalne vool kahekordne, mistõttu peab aku toite liin olema kaks korda paksem (0,5 mm). USB toiteliin tehti samuti igaks juhuks pisut jämedam (0,25 mm), et garanteerida voolu tolerantsus.

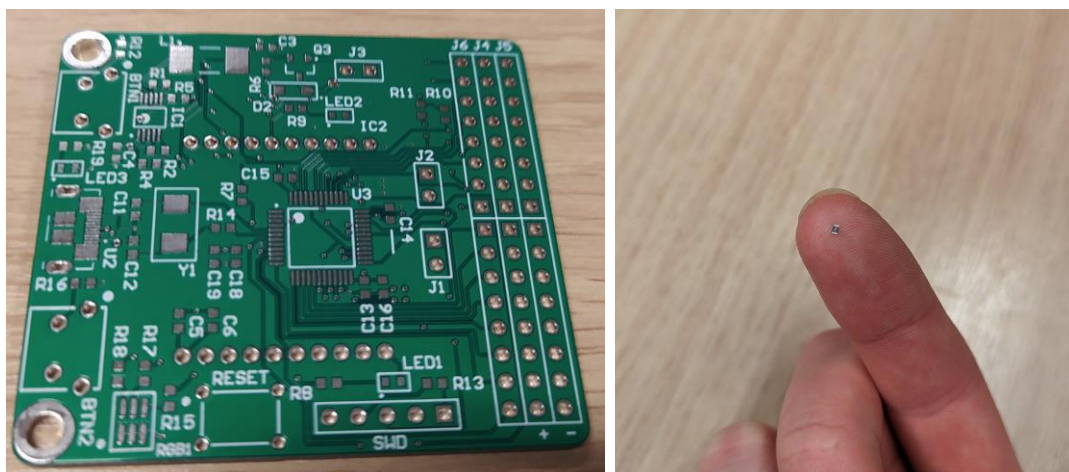
3.2.5 Ohukohad

Veel on trükkplaadi disainil mõned reeglid mida tuleks järgida. Drossel, mis on mõeldud pingeregulaatori väljundi silumiseks, peab olema võimalikult lähedal pingeregulaatorile, et vältida teiste signaalide mõjutamist. Väline kella signaali generaator on üpris habras, mistõttu tuleks vältida kihtide vahelisi läbimisi (*via*) ja ühendused peaksid olema mikrokontrolleri lähedal. Samamoodi on habras USB andmete signaalid, kuna tegemist on diferentsiaal paariga (*differential pair*). Liinid peaksid olema mõlemad sama pikad ja soovituslik oleks jällegi vältida kihtide vahelisi läbimisi.

Trükkplaat otsustati tellida ettevõttest JLCPCB, sest sealt tellib tavaliselt Robotiklubi kõik oma trükkplaadid [21]. Et disainitud trükkplaat ühtiks tootja poolsete võimekustega, tuli Altiumis importida JLCPCB poolsed reeglid. Reeglid määravad näiteks ära kui peenikesed võivad loodud rajad minimaalselt olla ja kui suured peavad olema komponentide ja radade vahed. Nende reeglite põhjal on võimalik jooksutada Altiumis disainireeglite kontroll (*Design Rule Check*), mis näitab ära disaini käigus tehtud rikkumised.

3.3 Trükkplaadi kooste

Trükkplaadi ja komponentide kohale jõudes, tuli joota vastavad komponendid neile määratud ühenduskohtadele. Ilma komponentideta trükkplaat on välja toodud joonisel 13. Kuna tegemist on väikest mõõtu trükkplaadiga, siis olid enamus komponentidest miniatuursete mõõtmetega, pindliitealaga ühenduvad komponendid (*SMD – Surface-Mount Device*). Näiteks olid kõik takistid ja kondensaatored dimensioonidega 1,6 mm x 0,85 mm, mille suurust on võimalik näha joonisel 3.9.



Joonis 3.9 Pisi-Xbee trükkplaat komponentideta (vasakul) ja takisti suurus (paremal).

Tavaliselt alustatakse komponentide jootmist toiteahelast, et näha kas toiteahel toimib õigesti ja annab väljundiks soovitud pinge, piisava stabiilsusega. Toiteahela juures tuli kohe alguses välja, et polaarsuskaitse eesmärgiga MOSFET, ei töötanud ootuspäraselt. Põhjuseks oli see, et MOSFET-i jälgpinna ühendused ei olnud vastavuses päris komponendi jalgadega. Probleemist mööda minemiseks lühistati MOSFET-i ühenduskoht, kuid kahjuks sellisel viisil polaarsuskaitset toiteahelal ei ole tagatud.

Peale toiteahela tööle saamist ühendati mikrokontroller ja kõik sellega seotud komponendid, mis on vajalikud mikrokontrolleri programmeerimiseks. Tarkvara peale laadimiseks kasutati SWD (*Serial Wire Debugging*) ühendust, mille abil on võimalik ka

programmi siluda. SWD kasutamiseks oli tarvis ST-Link liidest, mille abil arvuti on võimeline mikrokontrolleriga suhtlema.

Trükkplaat saadi terviklikult tööle kolmandal kokkupanemisel. Esimesel trükkplaadil oli tuvastamata viga, mis ei lasknud mikrokontrollerit programmeerida ja tõenäoliselt põletas uusi mikrokontrollereid ühendades need läbi. Katsetamiste käigus põles läbi kolm mikrokontrollerit. Teisel korral kõik uuesti nullist jootes, tuli lahti mootori muunduri pindliiteala, mida ei olnud väikeste mõõtmete tõttu võimalik parandada. Lõpuks kolmanda korraga saadi trükkplaat tervikuna tööle ja sai alustada funktsionaalsuste testimisega. Valmis joodetud trükkplaati on näha jooniselt 3.10.



Joonis 3.10 Pisi-Xbee trükkplaat kokku joodetuna.

4 ROBOTI KATSETAMINE

Roboti kokku saades tuli katsetada, kas kõik ettenähtud funktsionaalsused töötavad korrektselt. Ühendades Pisi-Xbee kontrolliplaat läbi viikude Pisi-Boti andurite ja täiturite plaadiga oli võimalik hakata roboti võimekusi tervikuna kontrollima. Et kontrollida Pisi-Boti funktsionaalsusi, oli tarvis luua arenduskeskkonnas STM32CubeIDE mikrokontrollerile vastav projekt. Projekti aluseks tuli võtta varasemalt konfigureeritud IOC fail, kus elektroonika loomiseks olid kõik funktsionaalsused ära seadistatud. Selle abil sai genereerida STM32CubeIDE abil programmi baasi, kus kõik algseadistused sooritas arenduskeskkond ise ära. Roboti katsetamisel hakati üks haaval funktsionaalsuseid tööle panema ja testima kas need töötavad korrektselt, samal ajal luues Pisi-Boti jaoks eraldi teegid, et tulevikus roboti programmeerimine oleks lihtsam.

4.1 Komponentide ja funktsionaalsuse kontroll

4.1.1 Mootorid

Kui veenduti, et mikrokontrolleri programmeerimine töötab, oli esimeseks sammuks testida mootorite juhtimist. Mootorite tööle saamisel tuli ette takistusi nii elektroonika kui tarkvara poolelt. Esiteks oli mootori muunduri jootmine üpris keeruline, selle väikeste mõõtude ja pindliiteala tõttu. Kuna tegemist oli WSON (*Very-Very-thin Small Outline Non-leaded*) tüüpi komponendiga, siis puudusid sellel välja ulatuvad jalad ning ühenduste kontakte ei olnud võimalik silmaga näha.

Kui mootori muunduri ühendused tundusid korras olevat, sai edasi liikuda tarkvara poolele. Mootorite kontroll toimub mikrokontrolleri pulsi-laius modulatsiooni (PWM) abil. Eialgu oli probleemiks mootorite voolu ületarbimine. Kuna ohutuse mõttes kasutati toiteallikana toiteplokki, millel on võimalik voolu piirata, siis tuli välja, et robot vajab väga palju voolu mootorite juhtimiseks. Selle põhjuseks võis olla kaks faktorit. Esiteks tuli välja, et PWM-i sagedus oli konfigureeritud liiga madalale sagedusele, kuna arvutustes oli tehtud viga. Mootoreid kontrolliv PWM sagedus arvutati välja järgneva valemiga [22]:

$$F_{PWM} = \frac{F_{CLK}}{(ARR + 1) * (PSC + 1)}. \quad (4.1)$$

Kus taimerisse määratud kellasedus:

$$F_{CLK} = 16 \text{ MHz.}$$

Eelskaleerija (*PSC - Prescaler*), millega vähendatakse taimeri opereerimissagedust:

$PSC = 16$.

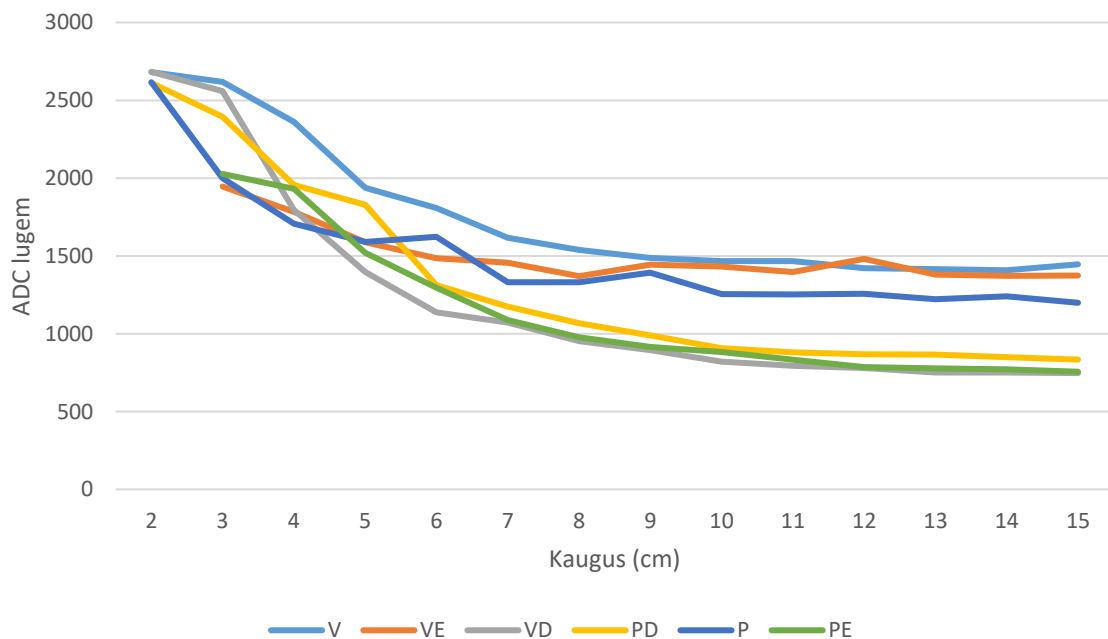
Taimeri nullimisväärtus (*ARR - Auto-Reload Register*), milleni lugemiseni jõudes loendur nullitakse. Andmelehe järgi liidetakse ARR väärtusele automaatselt +1 ja kuna eesmärk on saada mootori kiirust kontrollida vahemikus 0 - 100, siis

$ARR = 99$.

Samuti ei olnud mootorite kontrollimine esialgu korrektne, mis tuli välja põhjalikumal muunduri andmelehe uurimisel. Kui need probleemid said lahendatud, töötas mootorite juhtimine veatult.

4.1.2 Kaugusandurid

Pisi-Botil paiknevate IR kaugusandurite lugemiseks tuleb esialgu sisse lülitada infrapuna LED-id, mille objektidelt tagasipõrkunud kiirguse intensiivsust mõõdetakse, kui need jõuavad vastuvõtjasse. Vastuvõtjasse jõudnud intensiivsust loeb mikrokontroller ADC kanalite kaudu. Kui lugemine on ära tehtud lülitatakse LED-id uuesti välja, et vähendada voolutarbimist. Joonisel 4.1 on näha kuidas erinevad anduri lugemid varieeruvad fikseeritud distantsidel. Iga väärtus on kümne lugemi keskmine ja väärtuste varieeruvus oli vahemikus ± 10 ühikut.

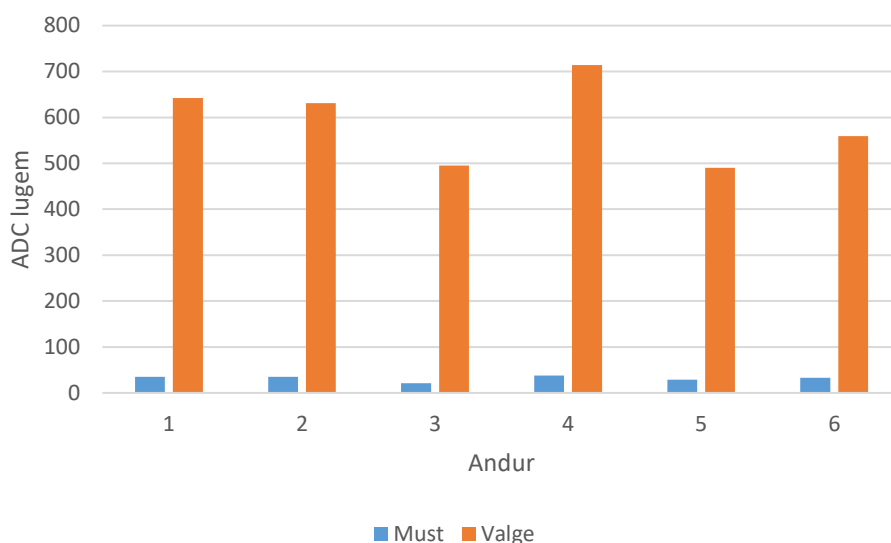


Joonis 4.1 IR kaugusandurite lugemid vahemikus 2 - 15 cm.

Andureid testides tuli välja, et nende efektiivne mõõtekaugus on väga väike. Kaugemalt kui 8 cm ei ole andurid enam efektiivsed, mistõttu sobivad need paremini labürindi raja läbimiseks, kuid mitte nii hästi *folk*race võistluseks. Jooniselt on ka näha kuidas kõigi andurite puhul mõõtevahemikud ei ühti. Selleks, et kõik andurite lugemid omavahel paremini ühtiksid, saab programmikoodis lisada igale andurile enda parandusteguri. Samuti on jooniselt näha, et PE ja VE positsioonidel ei ole tehtud mõõtmist 2 cm kauguselt, sest mõõtmisel kasutatud sein jõudis enne vastu roboti äärt.

4.1.3 Jooneandurid

Jooneandureid on Pisi-Botil kuus tükki. Nende ainuke eesmärk on joonejärgimis rajal olla võimeline tuvastada, kas tegemist on valget või musta värvi pinnaga. Jooneandurite mõõtetulemusi on näha jooniselt 4.2, kus mõõdeti ADC lugemeid mustal ja valgel pinnal. Selleks võeti päris võistlusel kasutusel olnud raja tükk, kus on valgel pinnal must joon. Andmete järgi on selgelt näha millal on tegemist valge ja millal musta värvi joonega. Et andureid joonejärgimiseks kasutada, tuleb antud juhul lihtsalt määrata piirväärtus, mille järgi arvestada kas tegemist on joonega või mitte.



Joonis 4.2 Jooneanduri lugemid valgel ja mustal pinnal.

4.1.4 Raadiomoodul

Mikrokontrolleri ja raadiomoodul kasutavad suhtluseks asünkroonset UART kommunikatsiooni. See tähendab, et nii saatja, kui vastuvõtja peavad olema ühtmoodi sätestatud, et andmeedastus toimiks korrektselt. Kasutatud UART konfiguratsiooni andmeid on näha tabelist 8. Mikrokontroller seadistatakse ära programmi koodis ja iga raadiomoodul mis kasutusele võetakse seadistatakse ära Xbee raadiomoodulitele loodud programmis XCTU. [23]

Tabel 4. UART Protokollid seadistus.

Seadistus	Väärtus
Edastuskiirus	115 200 b/s
Start bitt	1
Andmebitid	8
Stop bitt	1

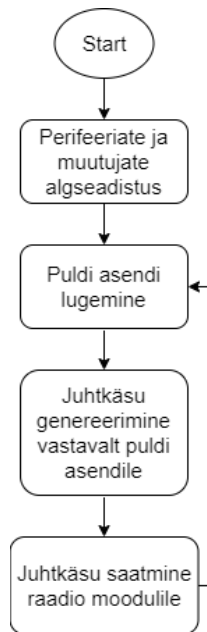
Raadio moodulit tööle saades tuli välja, et mikrokontrolleri saatja ja vastuvõtja viigud olid vastupidiselt raadiomooduliga ühendatud. Et andmeedastus toimiks õigesti, peab olema mikrokontrolleri saatja viik ühendatud raadio mooduli vastuvõtja viiguga ja sama teistpidi. Et probleem lahendada tuli trükkplaadil antud ühendusrajad läbi lõigata ja juhtmetega ühendades korrektsioon teha. Peale paranduse sisse viimist töötas UART suhtlus ja toimis ka raadiosuhtlus. Pisi-Xbeed, millele on ühendatud Xbee raadio moodul, on näha jooniselt 4.3.



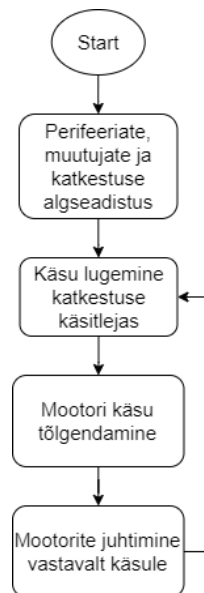
Joonis 4.3 Pisi-Xbee kontrolliplaat, millele on ühendatud Xbee raadiomoodul.

4.2 Näidisprogrammi koostamine

Et demonstreerida roboti töötamist tervikuna, otsustati luua programm, kus kontrollitakse roboti sõitmist läbi raadioside. Kontseptsiooni mõtte on kasutada kahte Pisi-Xbee kontrolliplaati, millest esimene on ühendatud Pisi-Botiga kontrollides roboti liikumist ja on raadio ühenduses vastuvõtja rollis. Teine kontrolliplaat töötab olemuselt juhtpuldina, saates robotile raadioühenduse teel liikumiskäsk. Selle plaadi külge on ühendatud juhtkang, mis edastab käsk vastavalt kangi asendile ja saadab raadio teel liikumiskäsu edasi. Kontseptsiooni eesmärk on demonstreerida, kuidas Pisi-Xbeed on võimalik kasutada nii sisendipõhise andmete saatjana, kui ka vastuvõtjana, roboti juhtimiseks. Juhtpuldi ja Pisi-Boti kontrolleri näidisprogrammi loogika on toodud välja joonistel 4.4 ja 4.5.



Joonis 4.4 Juhtpuldi kontrolleri näidisprogrammi loogika.



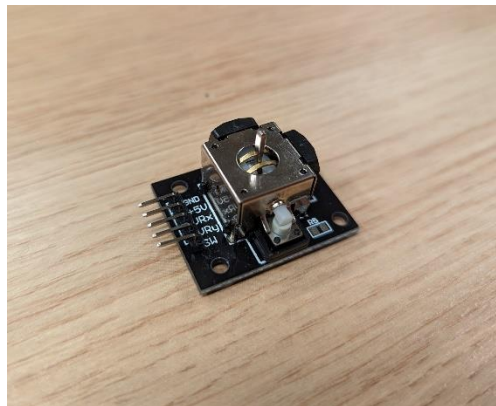
Joonis 4.5 Pisi-Boti kontrolleri näidisprogrammi loogika.

Nagu jooniselt näha, on tarvis kirjutada programm kahele kontrolleri, millest üks töötab juhi ja teine alluva rollis. Juhiks on antud juhul puldiga Pisi-Xbee kontrolleri ja alluvaks on Pisi-Bot. Juhtpuldi tööle saamiseks oli vaja järgnevat komponente:

1. Pisi-Xbee kontrolleriplaat
2. Juhtimiskang

Juhtimiskang on võetud Arduinole mõeldud arenduskomplektist ja selle kasutamiseks on vaja nelja viiku. Juhtimiskangi on võimalik näha jooniselt 4.5. Komponenti peal on kirjas, et seda on vajalik toita 5 V pingega. Küll aga, näeb kang välja nagu oleks

kasutusel lihtsalt kaks potentsiomeetrit, mis annavad X ja Y koordinaadi positsiooni kohta infot analoog pingena. Seetõttu prooviti kas komponent töötab ka siis, kui seda toita sama 3,3 V pingega, millel opereerib Pisi-Xbee. Tuli välja, et töötab, mistõttu ei ole tarvis välist pingemuundurit. X ja Y asendi väljundpinge on antud juhul vahemikus 0 – 3,3 V ning seda on võimalik lugeda mikrokontrolleri ADC kanalitega. Loetud asendite järgi on võimalik genereerida roboti kahele mootorile juhtimiskäsud, mis raadio teel edastada. Antud juhul X koordinaadi asendit kasutatakse, et määrata roboti pööramissuunda ning Y väljundit roboti liikumiskiiruse määramiseks.



Joonis 4.6. Kasutatav juhtimiskang.

Programmikood kirjutati valmis STM32CubeIDE arenduskeskkonnas ja mõlema programmi lähtekoodid on kättesaadavad GitLabi projektist, mis on välja toodud lisa 2. Pisi-Boti jaoks sai kasutada baas projekti ja teeke, mis loodi roboti katsetamise käigus. Puldi kontrolleri jaoks loodi aga eraldi projekt, kus konfigureeriti UART, raadiosuhtluseks ja kaks ADC kanalit, puldi X, Y koordinaatide mõõtmiseks.

KOKKUVÕTE

Töö eesmärgiks oli defineerida TTÜ Robotiklubis kasutatava Pisi-Bot robotplatvormi puudused ja need lahendada. Puuduseid analüüsidis tuli välja, et enamusest nendest on seotud robotit kontrolliva trükkplaadiga, mis kannab nime Pisi-Xbee. Et puudused lahendada, loodi uus versioon robotit kontrollivast trükkplaadist.

Trükkplaadil oli tuvastatud kolm peamist probleemi. Nendeks olid mikrokontrolleri võimekus, arendusvõimalused ja USB ahela kahjustamise oht ülevoolu tõttu. Pidades silmas neid puudusi, valiti vana Microchipi ATxmega32A4U mikrokontrolleri asemele STMicroelectronicsi STM32L151C6 mikrokontroller. Kuna tegemist oli teise tootja mikrokontrolleriga ja nende viigujaotused on erinevad, tuli luua kogu elektroonika nullist. Elektroonika loomiseks kasutati Altium arenduskeskkonda. Komponentide valimisel ja elektroonika loomisel tuli pidada silmas, et uus versioon säilitaks vana kontrollerplaadi omadused, lahendades ära tuvastatud vead.

Uue trükkplaadi valmides tuli kirjutada mikrokontrollerile teegid, mille abil lugeda robotplatvormi andureid ja juhtida täitureid, et tulevastel platvormi programmeerijatel sellega lihtsam töötada oleks. Selleks loodi Pisi-Boti baasprojekt, kus mikrokontroller on konfigureeritud töötama kõikide tema funktsionaalsustega ja kus paiknevad funktsionaalsuste kontrollimise teegid. Programmeerimiseks ja mikrokontrolleri konfigureerimiseks kasutati STM arenduskeskkonda STM32CubeIDE. Samuti tuli programmeerimise kaudu läbi testida Pisi-Boti funktsionaalsused, et kõik töötaks ootuspäraselt. Sinna alla kuulusid andurite lugemised ADC kanalite ja GPIO-de kaudu, ajamite juhtimine läbi PWM signaalide ja kommunikatsioon UART ning I2C kaudu.

Robotplatvorm saadi tööle uue kontrollerplaadiga ning selle demonstreerimiseks loodi näidisprogramm Pisi-Botile, kus robotit juhiti teise Pisi-Xbee plaadiga, millele oli ühendatud väike juhtimiskang. Juhtimiskangi positsiooni kaudu kontrolliti roboti liikumist, saates Pisi-Botile käske raadioside teel.

Probleemid, mis esinesid eelmisel kontrollerplaadi versioonil said lahendatud, kuid paar parandust tuleb veel sisse viia. Esiteks oli trükkplaadil kaks ühendusega seotud viga, mis küll ajutiselt said käsitsi parandatud, kuid tuleb uue trükkplaadi tellimisega lõplikult lahendada. Samuti on mõistlik võtta kasutusele sama mikrokontrolleri seeria suurema säilmäluga versioon, et oleks võimalik luua suurema mahulisi programmikoode. See ei vaja elektroonika poolset muudatust, kuna tegemist oleks sama viigujaotusega mikrokontrolleriga.

SUMMARY

The purpose of the thesis was to define existing problems with Pisi-Bot robot platform, which is used in TUT Robotics Club and fix them. After analyzing the problems, it turned out that most of the concerns were rooted in the controller board of the platform, which is called Pisi-Xbee. To fix the problems a new version of Pisi-Xbee's PCB had to be made.

There were three main problems detected with the board. These were the performance of the used microcontroller, the difficulties of developing software for the microcontroller and damaging of the USB electronics circuit. Keeping in mind these problems, a new STMicroelectronics microcontroller STM32L151C6 was chosen to replace the old Microchip ATxmega32A4U. Since the microcontroller is from another manufacturer and their pinouts are different, a new PCB had to be developed from scratch. Altium was used for designing the new electronics for the PCB. When selecting the components and designing the new PCB, the same functionality had to be kept, while solving the detected problems.

After the new PCB was ready, new software libraries had to be written for the new microcontroller, to make development easier for future programmers working with the platform. For that, a new Pisi-Bot base project had to be made, where the microcontroller was configured to work with all its provided sensors and actuators, and which also contained the libraries to work with them. For configuring and programming the microcontroller, STM development environment STM32CubeIDE was used. With programming the libraries, all the robots functionalities had to be tested, to make sure they work as expected. That included reading sensor values with ADC channels and GPIO-s, controlling the actuators with PWM signals and communication through UART and I2C.

The robot platform started working with the newly developed controller board. For demonstration a program was made for Pisi-Bot, where the robot was controlled with another Pisi-Xbee which had a small joystick attached to it. Pisi-Bot was controlled by the joystick's position, by sending commands through a radio module.

Problems which were detected on the previous version of Pisi-Xbee got fixed, but a few adjustments still needed to be made. Firstly, the PCB had two incorrect trace connections, which temporarily got fixed by hand, but should still be corrected by ordering a new PCB without these errors. Also another version of the same microcontroller could be used for more flash memory, to support more complex software

solutions. This does not need any additional changes in the PCB, because the microcontrollers have the same pinout.

KASUTATUD KIRJANDUSE LOETELU

- [1] Robotiklubi, „Pisi-Xbee 5“ [Võrgumaterjal].
[https://wiki.robotiklubi.ee/projektid/arendusprojektid/pisi_xbee5?s\[\]=pisixbee5](https://wiki.robotiklubi.ee/projektid/arendusprojektid/pisi_xbee5?s[]=pisixbee5) [Kasutatud 09.02.2023].
- [2] Microchip Tehnology, „Atmel 8/16-bit Atmel XMEGA Microcontroller ATxmega128A4U/ATxmega64A4U/ATxmega32A4U/ATxmega16A4U“ [Võrgumaterjal]. http://ww1.microchip.com/downloads/en/devicedoc/atmel-8387-8-and16-bit-avr-microcontroller-xmega-a4u_datasheet.pdf [Kasutatud 09.02.2023].
- [3] Robotiklubi, „Pisi-Bot“ [Võrgumaterjal].
[https://wiki.robotiklubi.ee/projektid/robotid/pisi-bot?s\[\]=pisi&s\[\]=bot](https://wiki.robotiklubi.ee/projektid/robotid/pisi-bot?s[]=pisi&s[]=bot) [Kasutatud 09.02.2023].
- [4] STMicroelectronics, „Integrated Development Environment for STM32“ [Võrgumaterjal]. <https://www.st.com/en/development-tools/stm32cubeide.html> [Kasutatud 14.02.2023].
- [5] Makeblock, „Makeblock mBot“ [Võrgumaterjal].
<https://www.makeblock.com/pages/mbot-robot-kit> [Kasutatud 14.02.2023].
- [6] Sumozade, „Alpha Obstacle Avoidance Robot Kit – Assembled“ [Võrgumaterjal].
<https://www.sumozade.com/product/alpha-obstacle-avoidance-robot-kit-assembled> [Kasutatud 14.02.2023].
- [7] STMicroelectronics, STM32L151x6/8/B „Ultra-low-power 32-bit MCU ARM based Cortex-M3 128KB Flash, 16KB SRAM, 4KB EEPROM, LCD, USB, ADC, DAC“ [Võrgumaterjal]. <https://www.st.com/resource/en/datasheet/stm32l151c6.pdf> [Kasutatud 28.02.2023].
- [8] STMicroelectronics, STM32F303CB, „Mainstream Mixed signals MCUs Arm Cortex-M4 core with DSP and FPU, 128 Kbytes of Flash memory, 72 MHz CPU, MPU, CCM, 12-bit ADC 5 MSPS, PGA, comparators“ [Võrgumaterjal].
<https://www.st.com/en/microcontrollers-microprocessors/stm32f303cb.html> [Kasutatud 28.02.2023].
- [9] STMicroelectronics, STM32G473CB, „Mainstream Arm Cortex-M4 MCU 170 MHz with 128 Kbytes of Flash memory, Math Accelerator, High Analog level integration“ [Võrgumaterjal]. <https://www.st.com/en/microcontrollers-microprocessors/stm32g473cb.html> [Kasutatud 28.02.2023].
- [10] STMicroelectronics, STM32L151C8, „Ultra-low-power Arm Cortex-M3 MCU with 64 Kbytes of Flash memory, 32 MHz CPU, USB“ [Võrgumaterjal].
<https://www.st.com/en/microcontrollers-microprocessors/stm32l151c8.html> [Kasutatud 28.02.2023].

- [11] STMicroelectronics, STM32L151CB, „Ultra-low-power Arm Cortex-M3 MCU with 128 Kbytes of Flash memory, 32 MHz CPU, USB“ [Võrgumaterjal]. <https://www.st.com/en/microcontrollers-microprocessors/stm32l151cb.html> [Kasutatud 28.02.2023].
- [12] Pololu Robotics and Electronics, „50:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft“ [Võrgumaterjal]. <https://www.pololu.com/product/3073/specs> [Kasutatud 08.03.2023].
- [13] Texas Instruments, „DRV8835 Dual Low-Voltage H-Bridge IC“ [Võrgumaterjal]. https://www.ti.com/lit/ds/symlink/drv8835.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1681138538361&ref_url=https%253A%252F%252Fwww.mouser.ee%252F [Kasutatud 08.03.2023].
- [14] TDK Invensense, „High Performance 6-Axis MEMS MotionTracking™ Device“ [Võrgumaterjal]. <https://invensense.tdk.com/wp-content/uploads/2021/03/DS-000143-ICM-20689-TYP-v1.1.pdf> [Kasutatud 08.03.2023].
- [15] Altium, Altium Designer [Võrgumaterjal]. <https://www.altium.com/altium-designer> [Kasutatud 08.03.2023].
- [16] Digi, „Digi Xbee 3 RF Module“ [Võrgumaterjal]. <https://www.digi.com/resources/documentation/digidocs/pdfs/90001543.pdf> [Kasutatud 13.03.2023].
- [17] ON Semiconductor, „Quad 2–Input AND Gate“ [Võrgumaterjal]. <https://www.farnell.com/datasheets/76582.pdf> [Kasutatud 13.03.2023].
- [18] Texas Instruments, „TPS6205x 800-mA Synchronous Step-Down Converter“ [Võrgumaterjal]. https://www.ti.com/lit/ds/symlink/tps62051.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-ww&ts=1681134859851&ref_url=https%253A%252F%252Fwww.mouser.ee%252F [Kasutatud 13.03.2023].
- [19] SnapEDA [Võrgumaterjal]. <https://www.snapeda.com/altium-libraries/> [Kasutatud 17.03.2023].
- [20] 4PCB Advanced Circuits, Inc. „PCB Trace Width Calculator“ [Võrgumaterjal]. <https://www.4pcb.com/trace-width-calculator.html> [Kasutatud 10.04.2023].
- [21] JLCPCB. <https://jlcpcb.com/> [Kasutatud 12.04.2023].
- [22] STMicroelectronics, „AN4776 Application note. General-purpose timer cookbook for STM32 microcontrollers“ [Võrgumaterjal].

https://www.st.com/resource/en/application_note/an4776-generalpurpose-timer-cookbook-for-stm32-microcontrollers-stmicroelectronics.pdf [Kasutatud 12.10.2023].

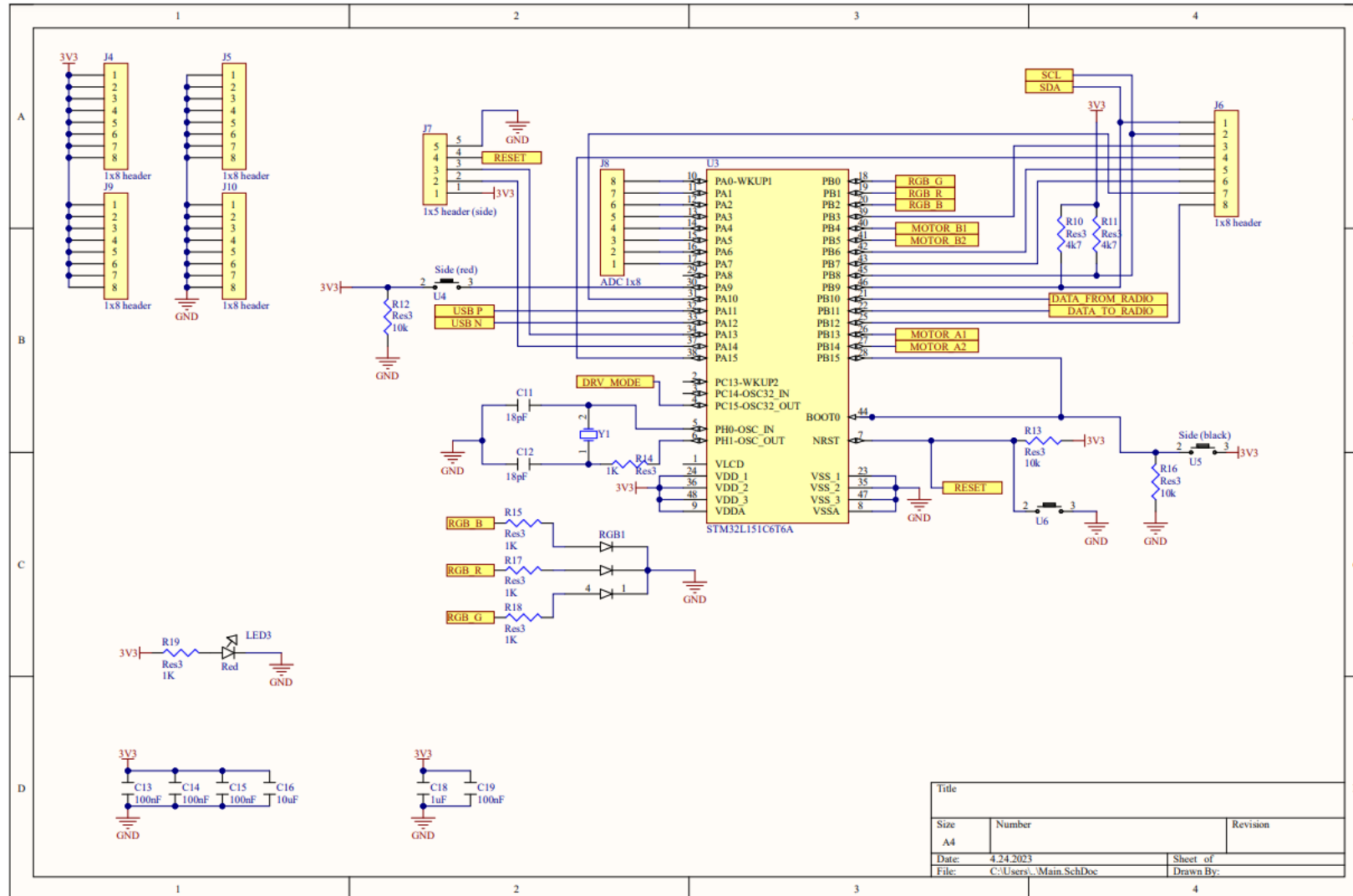
[23] Digi, XCTU [Võrgumaterjal]. <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu> [Kasutatud 18.10.2023].

LISAD

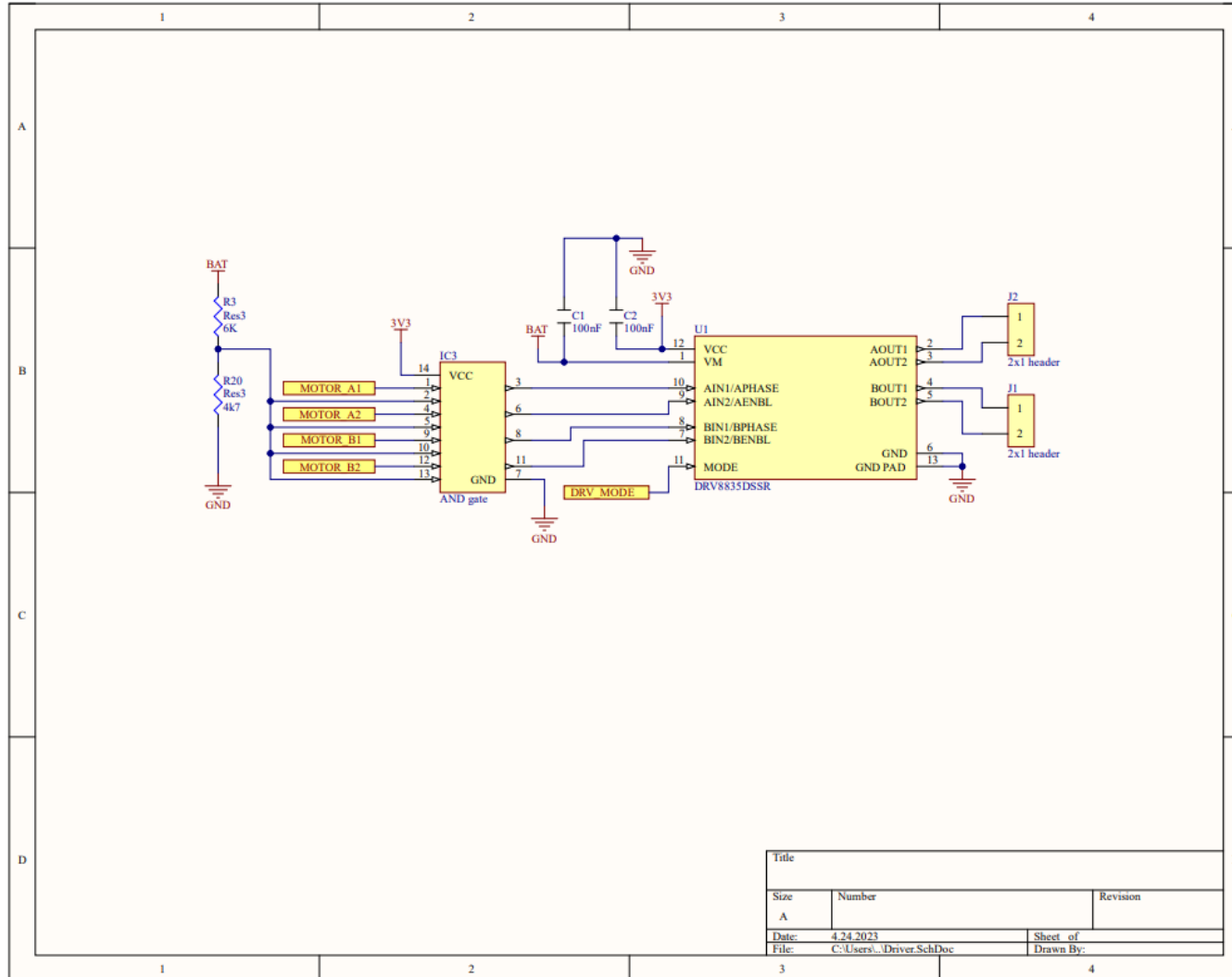
Lisa 1 Programmide lähtekoodid

Näidisprogrammide lähtekoode, koos Pisi-Boti baasprojektiga on võimalik vaadata GitLabist: <https://gitlab.com/martinlaanem/pisi-xtbee-6>

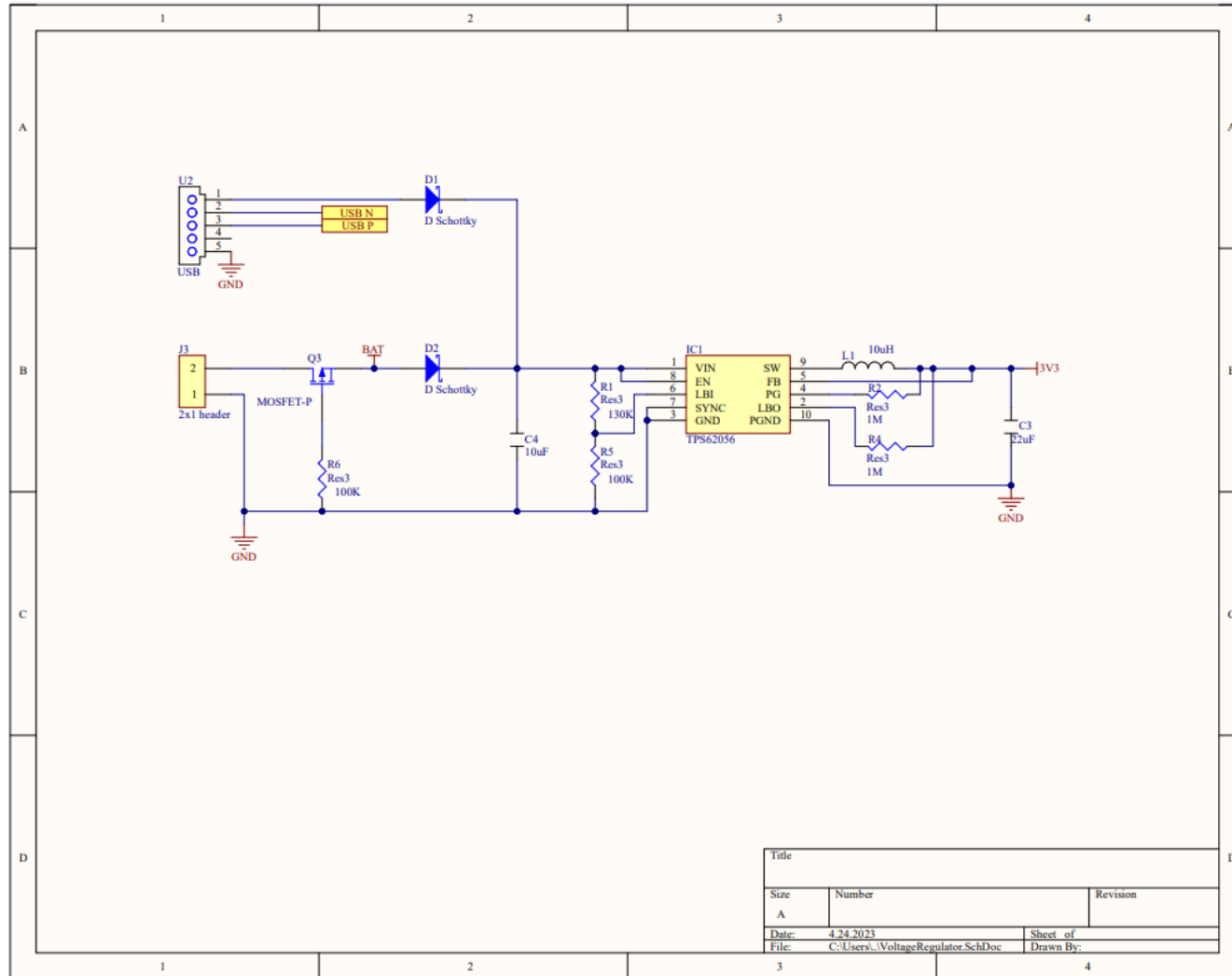
Lisa 2 Pisi-Xbee mikrokontrolleri elektronika skeem



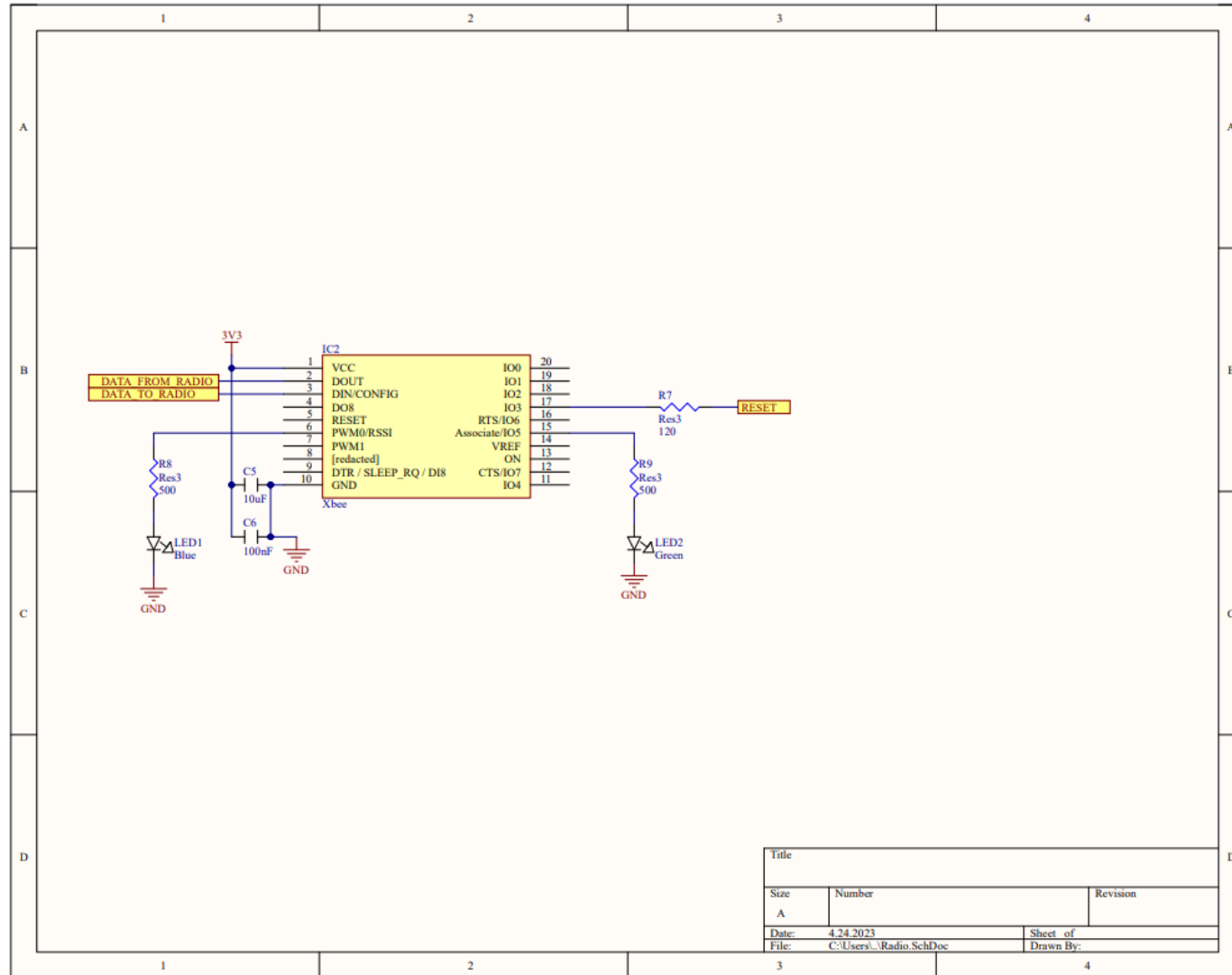
Lisa 3 Pisi-Xbee mootori muunduri elektroonika skeem



Lisa 4 Pisi-Xbee toiteahela elektroonika skeem

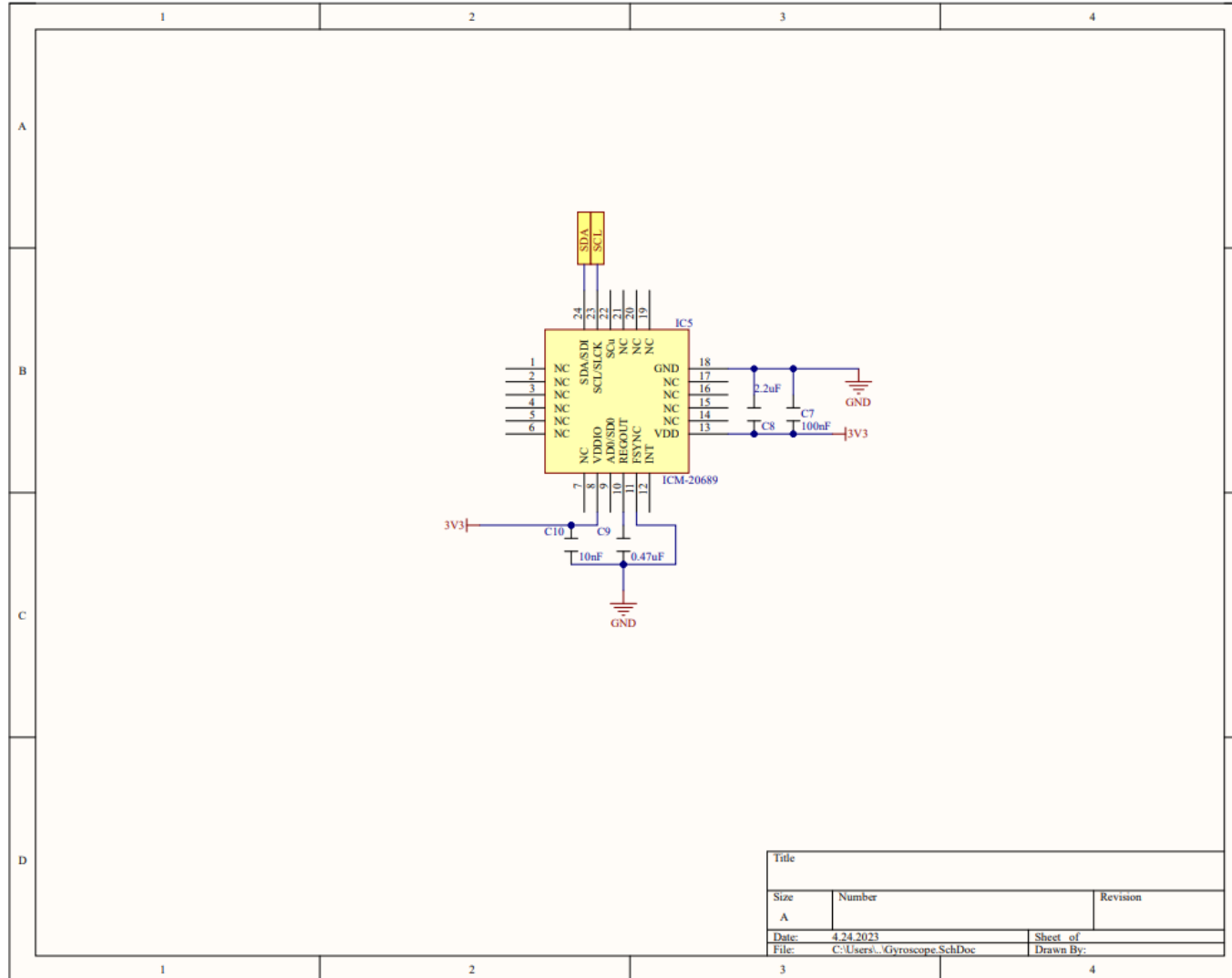


Lisa 5 Pisi-Xbee radiomooduli elektroonika skeem



Title		
Size	Number	Revision
A		
Date:	4.24.2023	Sheet of
File:	C:\Users\...Radio.SchDoc	Drawn By:

Lisa 6 Pisi-Xbee inertsiaalanduri elektroonika skeem



Lisa 7 Pisi-Xbee trükkplaadi disain

