

Tartu ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Raimo Köidam

Valguslahenduse tarkvara väljatöötamine õpperobotile Robotont

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendaja:
PhD Veiko Vunder

Tartu 2024

Lühikokkuvõte

Valguslahenduse tarkvara väljatöötamine õpperobotile Robotont

Robotont on avatud tarkvara ja avatud riistvaraga omniliikuv õpperobot. Robotondist on välja töötatud 3. generatsioon. Sellele on lisatud valguslahendus, milleks on 60 adresseeritavat RGB LED-valgustit. Bakalaureusetöö eesmärgiks on luua Robotondi valguslahenduse juhtimiseks tarkvara. Töö käigus lisati valguslahenduse juhtimiseks vajalik funktsionaalsus robotondi püsivarasse, täiendati Robotondi pardaarvuti ROS2 draiverit valgustuse pistikprogrammiga ja loodi ROS2 kimp valgustite demonstreerimiseks. Püsivarasse loodi ka erinevaid valgustuse režiime, mis on Robotondi draiveri vahendusel ROS-i sõnumite kaudu juhitud. Valgustite juhtimiseks loodi püsivarasse valgustite üksikute valgustite ja valgustite segmentide juhtimise käsud ning valgustuse režiimi muutmise käsk ning nende töötlus. Töö käigus täiendati ka Robotondi spetsiifilisi sõnumeid sisaldavat ROS2 kimpu, kus täiendati olemasolevaid ROS-i sõnumeid üksikute valgustite ja segmentide juhtimiseks ning lisati uus režiimi muutmise sõnum.

CERCS: T125 Automatiseerimine, robootika, juhtimistehnika

Märksõnad: valguslahendus, LED visualiseerimine, robootika, ROS

Abstract

Development of light solution software for the educational robot Robotont

Robotont is an open-platform omnidirectional educational robot. The third-generation Robotont has been developed with the addition of 60 addressable RGB LED lights. The aim of this bachelor's thesis is to create the software for controlling the lighting solution of Robotont. During the project, firmware for controlling the lighting solution of Robotont was developed, the ROS2 driver package for Robotont was extended with a lighting module, and a ROS2 bundle for demonstrating the lights was created. Various lighting modes were also implemented into the firmware, which can be controlled via ROS messages through Robotont's driver. Commands for controlling individual lights and segments, as well as commands for changing lighting modes and parsing of all of the commands, were added to the firmware. Additionally, the ROS2 bundle containing Robotont-specific messages was extended, enhancing the existing ROS messages for controlling individual lights and light segments and adding a new message for changing modes.

CERCS: T125 Automation, robotics, control engineering

Keywords: light solution, LED visualisation, robotics, ROS

Sisukord

Lühikokkuvõte	2
Abstract	3
Jooniste loetelu	5
Tablelite loetelu	6
Lühendid, konstandid, mõisted	7
1 Sissejuhatus	8
2 Kirjanduse ülevaade	9
2.1 ROS	9
2.2 Adresseeritavad LED valgustid	10
2.3 Valguslahendused robotites	12
2.4 Olemasolevad valguslahendused	13
3 Robotont	16
4 Eesmärk ja nõuded	18
5 Lahendus	19
5.1 Teegi valik	19
5.2 Püsivara muudatused	20
5.3 Valgustuse režiimid	21
5.3.1 Visuaalsed režiimid	21
5.3.2 Funktsionaalsed režiimid	22
5.4 Robotondi ROS2 draiveri pistikprogramm	25
5.5 ROS2 kimp valgustuse demonstreerimiseks	26
6 Kokkuvõte	27
Kokkuvõte	27
Viited	29
Lisad	31
Lisa 1	31
Lihtlitsents	32

Jooniste loetelu

2.1	WS2812B ja SK6812RGBW valgustid [1]	10
2.2	Adresseeritavatele valgustitele andmete saatmine	11
2.3	WS2812B tüüpiline ühendamise viis [2]	11
2.4	Valgustite juhtimise ajastused	12
2.5	CoBot koos lisatud valgustitega [3]	13
2.6	ModLight valguslahenduse prototüüp [4]	14
2.7	BWIBot suunda näitamas [5]	14
2.8	Drooni liikumise suuna näitamise erinevad viisid [6]	15
3.1	Robotondi erinevad generatsioonid. (a) 1. generatsioon, (b) 2. generatsioon, (c) 2.1 generatsioon ja (d) 3. generatsioon	16
3.2	Robotondi komponentide vaheline suhtlus	17
3.3	Trükkplaadil olevad valgustid	17
5.1	STM32-ARGB-DMA teegi abil esimesed 7 valgustit seatud punaselt põlema	20
5.2	Mootori kiiruste režiim	23
5.3	Robotont MOTOR_SPEEDS režiimis vasakule pööramas	23
5.4	Joonisel on kuvatud SCAN_RANGES režiimi töö näidis. Nool näitab roboti vaate suunda.	24
5.5	ESP32 arendusplaat ja WS2813B valgustiriba	24
5.6	Robotondi valgustuse jaoks vajalike sõnumite teekond	26

Tablelite loetelu

5.1	Loodud käskude vormistused	20
5.2	LM käsu režiimide parameetrid	21

Lühendid, konstandid, mõisted

LED - *Light-Emitting Diode*, valgusdiod

RGB - *Red Green Blue*, punane roheline sinine

HSV - *Hue, Saturation, Value*, värvitoon, värvuse küllastatus, heledusväärtus

ROS - *Robot Operating System*, roboti operatsioonisüsteem

GPIO - *General-purpose input/output*, üldotstarbeline sisend/väljund

DMA - *Direct Memory Access*, otsemälupöördus

PWM - *Pulse Width Modulation*, pulsilaiusmodulatsioon

1 Sissejuhatus

Maailmas kasutatakse järjest rohkem roboteid, mistõttu muutub üha olulisemaks, et need robotid suudaksid suhelda oma ümbruses olevate inimestega ja edastada teavet enda kohta. Robotid võivad kasutada mitmeid erinevaid meetodeid info jagamiseks välismaailmaga. Näiteks saavad robotid edastada teavet kõlarite abil helisignaalide kaudu või ekraani kaudu, mis on roboti külge kinnitatud. Nendel viisidel on ka aga oma puudused. Helisignaali puhul võivad robotid enda ümber olevaid inimesi häirida ja ekraani puhul ei pruugi see informatsioon, mis sinna kuvatud on enam kaugelt loetav olla.

Üheks informatsiooni edastamise viisiks robotitel on veel RGB LED-valgustid, mis võimaldavad edastada teavet valguse ja värvide abil. Kui robotil on palju valgusteid ja need on individuaalselt adresseeritavad, siis saab nende abil kuvada erinevaid animatsioone, millega saab roboti kohta informatsiooni kuvada. Valgustite eeliseks on see, et need on suurema valguse intensiivsusega ja seega kaugemalt nähtavad kui ekraanid. Need ei häiri ka kõrvalisi isikuid nagu seda teeb heli signaali edastamine kui edastatakse pidevalt informatsiooni. Siiski on valgustitel ka omad miinused, näiteks kui valgustid on suunatud otse vaatajale silma, siis võivad need olla häirivad ja kui inimene robotit ei näe, siis oleks heli signaal kasulikum.

Robotite disaini ja funktsionaalsuse arendamisel mängivad olulist rolli ka haridusrobotid. Üheks selliseks näiteks on Robotont, mis on avatud platvormiga omniliikuvate ratastega õpperobot, välja töötatud Tartu Ülikoolis [7]. Robotondist on valminud 3. generatsioon, milles on toimunud võrreldes varasemaga suuri muudatusi. Uuel Robotondi generatsioonil on 3D prinditav kere ja 18 V aku süsteem. Kui varasemalt oli suuremad elektroonika komponendid jagatud erinevateks osadeks ja ühendatud juhtmetega, siis uues generatsioonis on kõik elektroonika koondatud kokku ühele suurele trükkplaadile, mis vähendab juhtmete vajadust. Uuel trükkplaadi alumisele äärele on ka paigutatud 60 adresseeritavad RGB LED-valgustit, mille juhtimiseks vastav tarkvara puudus.

Käesoleva bakalaureuse töö eesmärgiks on luua Robotondi valguslahenduse jaoks tarkvara, millega saab selle LED-valgusteid juhtida. Töös töötatakse välja Robotondi püsivaras valgustite juhtimise osa ning luua Robotont ROS-i draiverile pistikprogramm, mis võimaldab valgusteid juhtida ROS-i sõnumite teel.

2 Kirjanduse ülevaade

2.1 ROS

Üks oluline komponent, mis on võimaldanud robotika valdkonnas innovatsiooni ja õppimist, on ROS (*Robot Operating System*). ROS on avatud lähtekoodiga tarkvararaamistik, mis on loodud robotite juhtimiseks [8, 9]. Sellel on olemas funktsionaalsused, mis võimaldavad lihtsustatud riistvara juhtimist, pakuvad laialdaselt kasutatavaid funktsioone, toetavad protsessidevahelist suhtlust ning tarkvara kogumike ehk kimpude haldust [8, 9]. ROS-i kasutamine on robotikas laialt levinud, kuna see võimaldab arendajatel keskenduda oma unikaalsete robotilahenduste loomisele, jättes standardiseeritud funktsionaalsused ROS-i hoolde [10].

ROS-i tarkvara osadeks on kimbud, mis on tarkvarapakettide kogumid. ROS-i tarkvara on üles ehitatud sõlmedest, mis on konkreetseid ülesandeid täitvad protsessid. Sõnumid on andmestruktuurid, mida kasutatakse suhtluseks sõlmede vahel. ROS-i sõlmed saavad olla tellijad ja kuulutajad ning vastavalt võtta vastu ja saata sõnumeid, mida vahendatakse rubriikide kaudu. [9]

ROS-il on kaks versiooni: ROS1 ja ROS2. ROS1 on olnud olemas aastast 2007 ja sellele on välja töötatud palju robotikaga seonduvat tarkvara. Praegu ollakse üleminekuperioodis uuemale ROS2 versioonile ja uue funktsionaalsuse arendamisel vaadatakse peamiselt ROS2 suunas. ROS1 töötab ainult Unixi-põhistel operatsioonisüsteemidel ning selle tarkvara katsetatakse peamiselt Ubuntu ja Mac operatsioonisüsteemides. [8, 9, 11]

ROS2 on uuem versioon ROS-ist, mida on disainitud algusest peale kasutades teadmisi ja kogemusi, mis saadi ROS1 arendamisest. Seda ei tehtud lihtsalt ROS1 edasiarendusena, sest see oleks nõudnud liiga suuri muutusi, mis oleks vanemale versioonile loodud rakendused muutnud ebastabiilseks. ROS2 töötab lisaks Unixi-põhiste operatsioonisüsteemidele ka Windowsis. [12]

ROS2 toob kaasa mitmeid olulisi uuendusi, sealhulgas parema turvalisuse, reaalaaja töötlemise võimekuse ja parema süsteemi arhitektuuri. ROS2 on loodud olema modulaarsem ja paindlikum, et toetada keerukamaid robotisüsteeme. [10]

ROS2 kimbud on jagatud erinevateks distributsioonidesse, mis koosnevad paljudest kimpudest [13]. ROS2 distributsioonide eesmärk on tagada võimalikult stabiilne raamistik ROS-i arendajatele [14].

2.2 Adresseeritavad LED valgustid

ROS-i saab kasutada roboti erinevate komponentide juhtimiseks, milleks võivad olla robotite külge kinnitatud adresseeritavad valgustid. Adresseeritavad LED valgustid on valgustid, mida saab juhtida individuaalsete valgustite kaupa ning neid saab jadamisi ühendada üksteisega kokku. Adresseeritavatesse LED valgustitesse on sisse ehitatud kiip, mis võimaldab iga valgusti värvi ja eredust seada sõltumatult teistest valgustitest. Need valgustid koosnevad kolmest valgusdiodist: punane, roheline ja sinine. Mõned laialdaselt kasutusel olevad adresseeritavad RGB LED-valgusti tüübid on WS2812B ja SK6812. Need valgustid on tuntud ka nimega NeoPixel. [15, 16]

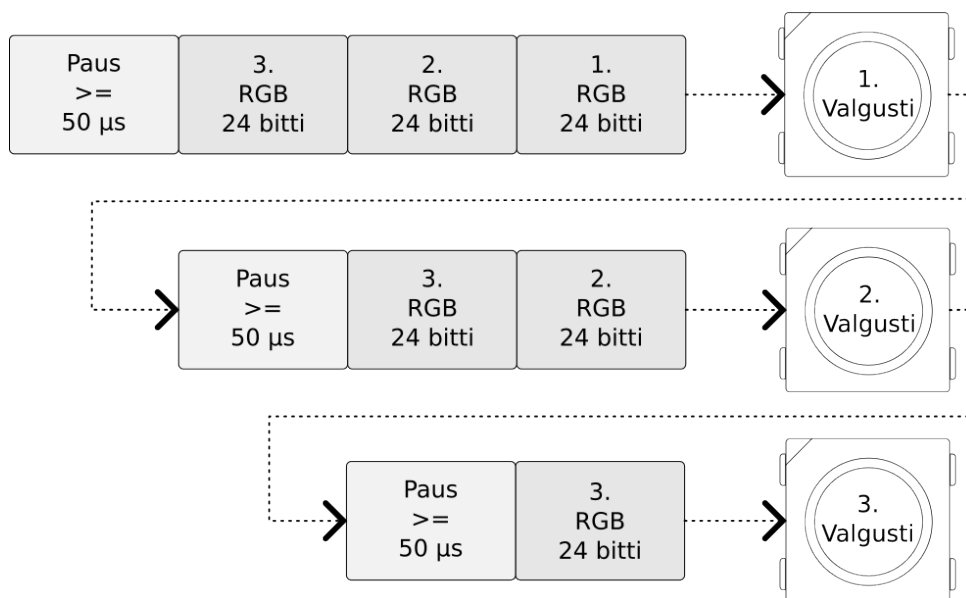
WS2812B ja SK6812RGB tüüpi valgustid on olemuselt väga sarnased ning nende juhtimine käib samamoodi, mis võimaldab neid koos kasutada [16]. Üheks nende valgustite erinevuseks on see, et SK6812 tüüpi valgustist on olemas SK6812RGBW versioon, millel on sisse ehitatud lisaks punasele, rohelisele ja sinisele valgusdiodile eraldi valge valgusdiod [1, 16]. Joonisel 2.1 on näha SK6812RGBW ja WS2812B valgusteid. Nende valgustite juhtimiseks on vaja mikrokontrollerit [15]. WS2812B ja SK6812RGB tüüpi valgustitele on vaja nende sisse lülitamiseks saata kolme värvi väärtused, milleks on punane, roheline ja sinine, aga SK6812RGBW valgusti jaoks on vaja saata üks lisa väärtus valge värvi jaoks [1].



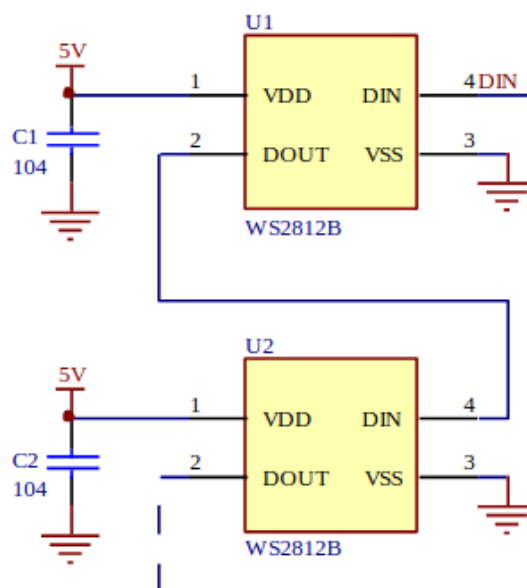
Joonis 2.1: WS2812B ja SK6812RGBW valgustid [1]

WS2812B ja SK6812RGB LED valgustitesse on sisse ehitatud kiip, millega valgustit juhitakse. LED valgusti ja kiip on paigutatud 5050 pakendisse. Sellele on sisse ehitatud signaalivõimendi. Pärast valgusti toite sisse lülitamist võtab DIN-viik vastu andmed kontrollerilt, esimene valgusti kogub esimesed 24 bitti andmeid ja saadab sellest järgnevad andmed pärast võimendamist DO-viigu kaudu edasi. Need 24 bitti koosnevad valgusti punasest, rohelisest ja sinisest väärtusest, mis on kõik 8 bitti pikad. Pärast iga valgusti läbimist on signaali pikkus 24 biti võrra

lühem. Valgustitele andmete saatmist on kujutatud joonisel 2.2. Kuna valgustid sisemiselt võimendavad edasi saadetavat signaali, siis valgusteid võib olla järjest kokku ühendatud piiramatult arv. Valgustid ühendatakse jädamisi ühendades ühe valgusti DO-viiku teise valgusti DIN-viigu külge nagu näha joonisel 2.3. [2, 17]



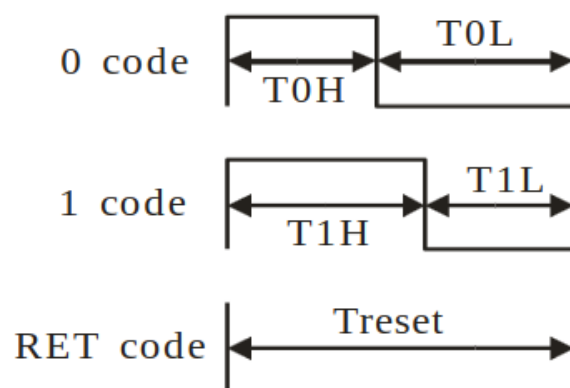
Joonis 2.2: Adresseeritavatele valgustitele andmete saatmine



Joonis 2.3: WS2812B tüüpiline ühendamise viis [2]

Valgustite juhtimine käib sagedusega 800 kHz. Iga biti saatmine võtab 1,25 µs. Kui saadetakse bitti väärtusega 1, siis on signaal kõrge 0.8 µs ± 150 ns ja ülejäänud aja madal ning kui saadetakse bit on 0, siis on signaal kõrge 0.45 µs ± 150 ns ja ülejäänud aja madal. Pärast kõikidele valgustitele andmete saatmist peab ootama vähemalt 50 µs, et saaks uuesti andmeid saata. Signaali kuju on näha joonisel 2.4. See paus annab valgusti sees olevatele kiipidele märku, et

andmete saatmine lõppes. Järgmine kord andmeid uuesti saates hakatakse jälle järjest esimesest valgustist pihta andmeid vastu võtma. [2]



Joonis 2.4: Valgustite juhtimise ajastused

2.3 Valguslahendused robotites

Inimesed puutuvad üha rohkem kokku erinevate robotitega. Kuna robotid liiguvad inimeste läheduses järjest enam ringi iseseisvalt, on saanud tähtsaks ümbruses olevatele inimestele teada robotite olekut ehk nende praegust tegevust või planeeritud tegevusi [3]. Robotite oleku teatamiseks on võimalik kasutada näiteks ekraane või kõne signaale, aga need ei ole pruugi olla kaugemalt arusaadavad [3]. Selle asemel on võimalik kasutada LED valgusteid, et anda edasi informatsiooni [3]. LED valgustite valguse intensiivsus on suurem kui ekraanidel, millel on LCD (*liquid-crystal display*) ehk vedelkristallekraanide puhul ekraani taga tausta valgususeks mõned valgustid või OLED (*organic light-emitting diode*) ehk orgaaniliste valgusdiodidega ekraani puhul, kus iga ekraani piksel käitub iseseisva väikse RGB valgustina [18]. Selle tulemusena on need valgustid võrreldes ekraanidega kaugemalt nähtavad ja annavad pidevalt informatsiooni edasi erinevalt kõne signaalidest [3]. Ekraani juhtimine on ka palju keerulisem kui väiksema koguse LED valgustite juhtimine. Näiteks kasutatakse valgusfoorides valgusallikaid, mis on kaugelt hästi nähtavad ja üheselt aru saadavad, et anda inimestele teada kas võib ristmikust edasi sõita või peab seisma jääma. Need on ka vähem häirivad kui valju heliga informatsiooni edasi andmine.

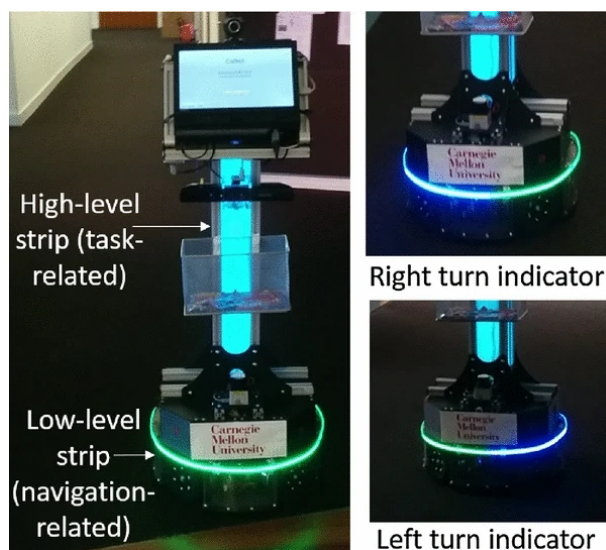
LED-valgustid on vastupidavad, odavad ja energiasäästlikud, mis teeb need heaks valikuks robotite külge panemiseks. Üksikuid valgusteid saab kasutada roboti sisse ja välja lülitatud oleku näitamiseks. Rohkem valgusteid ja erinevaid värve saab kasutada, et kuvada rohkem informatsiooni. [4] Suurema koguse valgustitega on ka võimalik edasi anda informatsiooni läbi animatsioonide [3]. Valgusteid saab juhtida, kui need ükshaaval GPIO viikude külge ühendada, aga selle puhul võivad olla piiratud viikude arv ja nende võimekus toimetada voolu valgustitele. Veel üheks võimaluseks on kasutada multiplekserit, mis võimaldab mitut seadet ühendada ühe sisend-väljund viigu külge, vähendades seeläbi vajalike viikude arvu, kuid see lahendus võib suurendada süsteemi keerukust ning põhjustada viivitusi valgustite juhtimisel. Roboti valguslahendusena saab kasutada ka adresseeritavat LED valgustiriba, millel saab ükshaaval muuta valgustite värvi ja eredust, kasutades ühte või mõnda vähest arvu GPIO viike. See on võimalik tänu sellele, et igal valgustil on sees kontroller, mis võimaldab juhtida iga valgustit eraldi.

Olekuid, mida valgustid saaksid näidata, on näiteks roboti aku olek või anda teada veast robotiga, et teavitada ümbruses olevaid inimesi [4]. Saab ka näiteks kuvada, mis suunas robot liigub, kas ta ootab inimeselt sisendit või kas roboti akut hetkel laetakse [3].

2.4 Olemasolevad valguslahendused

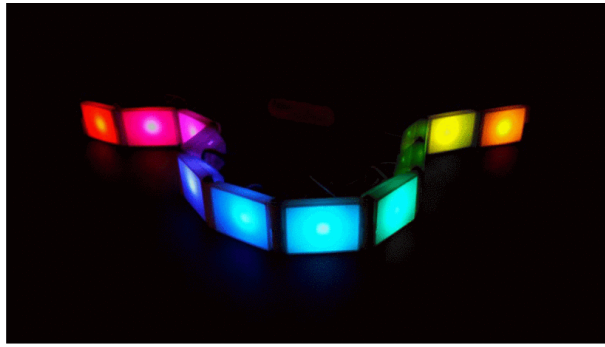
Selles alampeatükis antakse ülevaade erinevatest olemasolevatest valguslahendustest. Kirjeldatakse, milliseid roboti olekuid valgustite abil kuvatakse ja kuidas erinevaid olekuid nende abil esitatakse.

Baraka ja Veloso [3] lisasid autonoomsele mobiilsele teenindusrobotile CoBot kaks RGB LED-valgusti riba, kus ühel kuvati roboti olekut ja teisel roboti liikumisega seonduvat informatsiooni. Valgustite juhtimiseks kasutati Arduinot ja Adafruit NeoPixel teeki. Nad kasutasid erinevaid animatsioone ja värve, et näidata erinevaid roboti olekuid. Näiteks saab robot anda märku kui ta liigub, küsib abi või ootab inimeselt sisendit. Kui robot jääb kinni, siis ta annab kiire punast värvi valgustite vilkumisega sellest märku. Robot suudab ka näidata pika ajalise protsessi edenemist valgustiribal progressi ribana [3]. Joonisel 2.5 on näha CoBot robotit koos valgustitega.



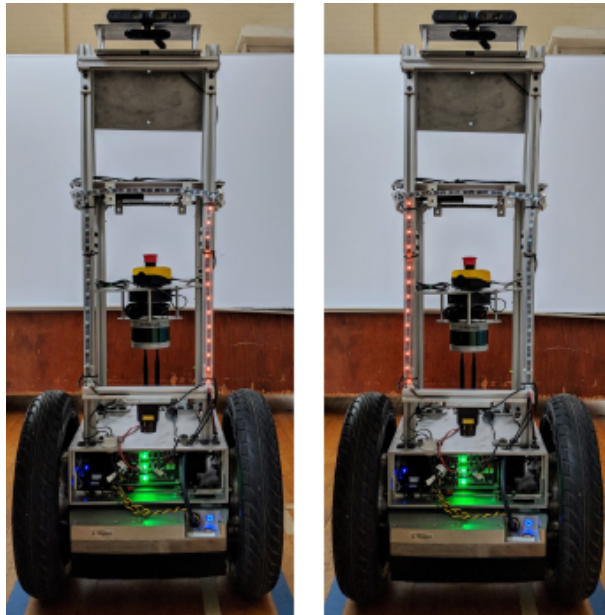
Joonis 2.5: CoBot koos lisatud valgustitega [3]

ModLight (Joonis 2.6) on üks avatud lähtekoodiga valguslahendus, mida on võimalik lisada külge erinevatele robotitele. Sellele on kaasa loodud samuti avatud lähtekoodiga tööriistad, mis teevad kasutajatele uute disainide tegemise lihtsaks. See on loodud teadlaste, disainerite, õpilaste ja harrastajate jaoks. Selle jaoks loodi C++ teek, mis võimaldab luua keerulisi valgusefekte. Teegi abil saab kuvada valgustitel erinevaid mustreid ja animatsioone, pannes valgustid vilkuma või värve vahetama. Robotites kasutamiseks on loodud ModLight-i jaoks ka ROS-i kimp. [4]



Joonis 2.6: ModLight valguslahenduse prototüüp [4]

BWIBotid on osa Building-Wide Intelligence (BWI) projektist Texase Ülikoolis Austinis, mille eesmärk on luua hoonesisene autonoomne robotite süsteem, mis teenindaks hoone elanikke ja küllastajaid [19]. BWIBot kasutab kahte WS2812B LED-valgusti riba, millel on mõlemal 60 valgustit. Üks riba on vasakul ja teine on paremal pool robotit. Valgustid näitavad millises suunas robot pöörab, vilkudes kollast värvi kaks korda sekundis (Joonis 2.7). [5]



Joonis 2.7: BWIBot suunda näitamas [5]

SMARTmBOT on avatud lähtekoodiga mobiilne robotplattvorm, mis põhineb ROS2 peal. Sellele on küljes 4 RGB LED-valgustit ning sellele on võimalik külge ühendada RGB LED-valgusti ribaga laiendus. Riba juhtimiseks on olemas ROS2 sõlm, mille kaudu saab muuta valgustite värvi, eredust ja kasutusel olevat valgusefekti. Nende valgustite abil on näiteks võimalik näidata roboti olekut. [20]

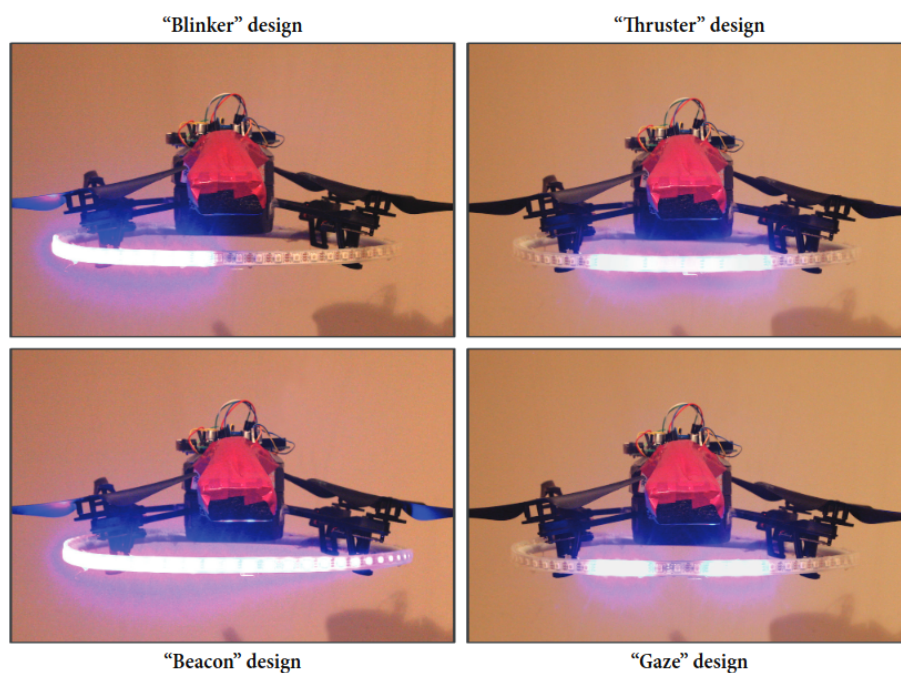
ROSbot on ROS-i jõul töötav neljarattaline autonoomne mobiilse robotplattvorm, millel on algselt küljes kolm sama värvi LED valgustit. Sellel on olemas pistik, kuhu saab ühendada WS2812B RGB LED-valgusti riba. Seda saab juhtida läbi ROS sõnumite, millega on võimalik valida viie erineva režiimi vahel: välja lülitatud, püsiv värv, sujuvalt eredust suuremaks-väiksemaks muutev, eest ja tagant vilkuv animatsioon ning värve vahetav režiim. Käsule saab ka lisada kuuteistkümnendkoodis värvi koodi, mis määrab valgustite värvuse.

Nende valgustite abil on võimalik visualiseerida roboti erinevaid olekuid. [21]

Szafir, Mutlu ja Fong [6] valmistasid valgusti lisamooduli, mida saab panna drooni külge. Selle lisandi küljes on Adafruit NeoPixel RGB LED-valgusti riba, mis ümbritseb drooni. Nad kasutasid valgusteid, et näidata drooni liikumise suunda. Kasutati nelja erinevat viisi, kuidas valgusdiodide ribal suunda näidata [6]:

- "blinker" - Esimene viis oli LED-valgusti ribal näidata ühe osa 1 Hz sagedusega vilkuvate valgustitega suunda, kuhu droon liigub.
- "thruster" - Teisel viisil näidati osade valgustitega, millisest suunast droon ennast tõukab, pannes põlema liikumis-suunale vastaspooles suunas olevad valgustid.
- "beacon" - Kolmas viis oli panna põlema kõik valgustid madala eredusega ja liikuma hakates, panna eredamalt põlema liikumissuunas olevad valgustid, mille tulemusel selle ümbruses olevate valgustite eredus muutus järjest väiksemaks.
- "gaze" - Neljandal viisil näidati roboti liikumise suunas kahte põlevat LED-valgusti riba osa, mis kujutasid roboti silmi.

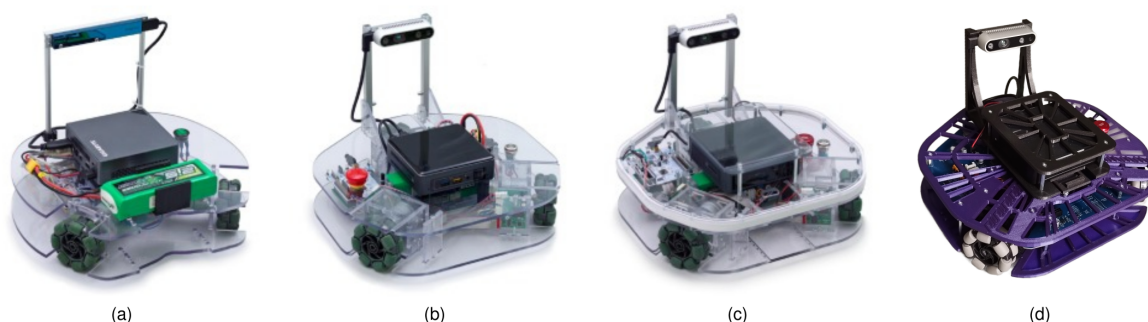
Joonisel 2.8 on näha loodud suuna näitamise viise.



Joonis 2.8: Drooni liikumise suuna näitamise erinevad viisid [6]

3 Robotont

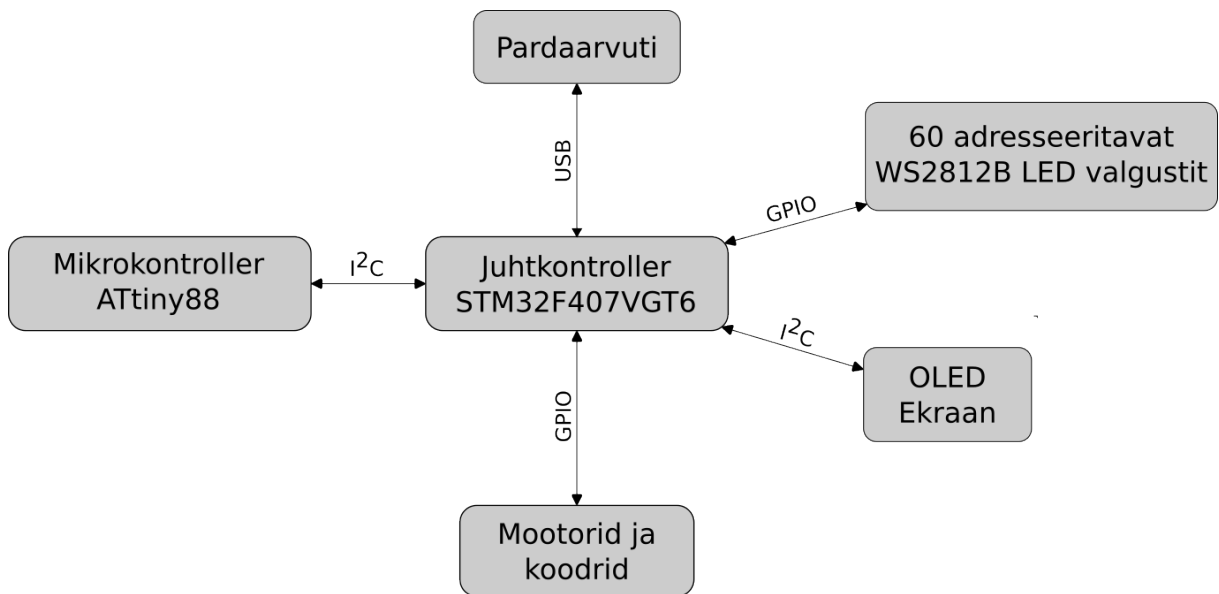
Robotont on Tartu Ülikoolis loodud avatud platvormiga omniliikuvate ratastega õpperobot [7]. Robotondi esimene ja teine generatsioon loodi Renno Raudmäe magistritöö raames [22]. Robotondi erinevaid generatsioone on kujutatud joonisel 3.1.



Joonis 3.1: Robotondi erinevad generatsioonid. (a) 1. generatsioon, (b) 2. generatsioon, (c) 2.1 generatsioon ja (d) 3. generatsioon

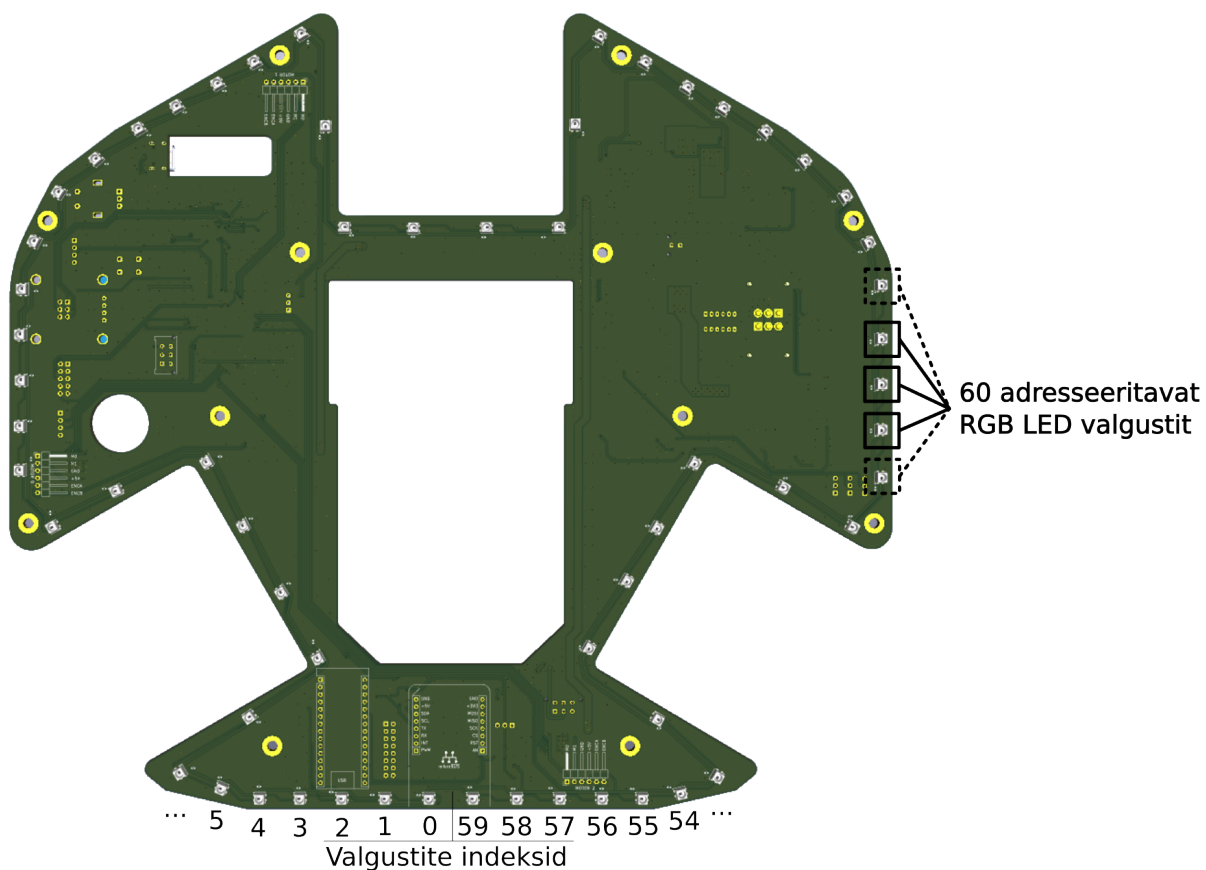
Robotondi kolmandal generatsioonil on uus kere disain ning kõik roboti kere osad on nüüd 3D prinditavad. Kui varasemalt olid mootorite juhtplaadid, mikrokontroller ja muud elektroonika osad eraldi roboti sees juhtmetega ühendatud, siis nüüd on kolmandas generatsioonis need komponendid pandud kokku ühe suure trükkplaadi peale, mis vähendab juhtmete vajadust.

Kolmanda generatsiooni Robotondi peal on pardaarvutiks 13. generatsiooni Intel NUC arvuti, millel jookseb Ubuntu operatsioonisüsteem. Pardaarvutil on peal ROS2 ning sellel jookseb Robotondi draiver, mis suhteb trükkplaadil oleva juhtkontrolleriga. Trükkplaadil on kaks mikrokontrollerit: STM32F407VGT6 mikrokontroller pardaarvutiga suhtlemiseks ja mootorite ning muu plaadil oleva elektroonika juhtimiseks ning ATtiny88 mikrokontroller toitehalduseks. Pardaarvuti suhtleb STM32 mikrokontrolleriga USB ühenduse teel. Mootorite juhtimiseks saadetakse pardaarvutilt käsud, mis parsitakse ja täidetakse STM32 mikrokontrolleris. Robotondi komponentide vahelist suhtlust on kujutatud joonisel 3.2.



Joonis 3.2: Robotondi komponentide vaheline suhtlus

Uuel trükkplaadil on 60 adresseeritavat WS2812B LED valgustit, mis on paigutatud trükkplaadi alla välimisele servale ning need on jaotatud trükkplaadi ümber ühtlaselt nagu on väha joonisel 3.3. Praeguseks robotondil puudub tarkvara uute valgustite juhtimiseks.



Joonis 3.3: Trükkplaadil olevad valgustid

4 Eesmärk ja nõuded

Käesoleva bakalaureusetöö eesmärk on välja töötada tarkvara, mis võimaldab juhtida kolmanda generatsiooni õpperoboti Robotont valgustuslahendust. Tarkvara koosneb kahest osast: STM32 mikrokontrolleri püsivara täiendusest ja Robotondi pardaarvuti ROS2 tarkvara täiendusest, et võimaldada valgustite juhtimist läbi ROS-i. Lisaks valgustuse juhtimise tarkvarale on eesmärk luua ROS-i kimp, mis demonstreerib erinevaid töö jooksul valmistatud valgustite režiime ja kasutab kõiki töös loodud valguslahenduse juhtimise funktsionaalsusi.

Töös püstitati valguslahenduse tarkvara jaoks järgmised nõuded:

- Lahendus peab võimaldama määrata robotile paigaldatud LED valgustite värvi ja heledust:
 - iga valgusti jaoks individuaalselt
 - korraga määratud kõrvutiasetsevate valgustite segmendile
- Valguslahendust saab juhtida roboti mikrokontrolleris eeldefineeritud režiimide abil, mis:
 - kirjeldavad erinevaid roboti olekuid (nt madala akutaseme hoiatus, mootorite kiirus, jne)
 - tõstaks roboti visuaalset atraktiivsust
- Valguslahenduse juhtimine peab olema võimalik roboti pardaarvutist ROS-i kaudu, nii valgustuse eelseadistatud režiimide valik kui ka üksikute valgustite ja segmentide juhtimine
- Valguslahendus peab olema programmeeritud sedavõrd ressursisäästlikult, et selle töö ei häiriks roboti muud funktsionaalsust (näiteks mootorite juhtimist).

5 Lahendus

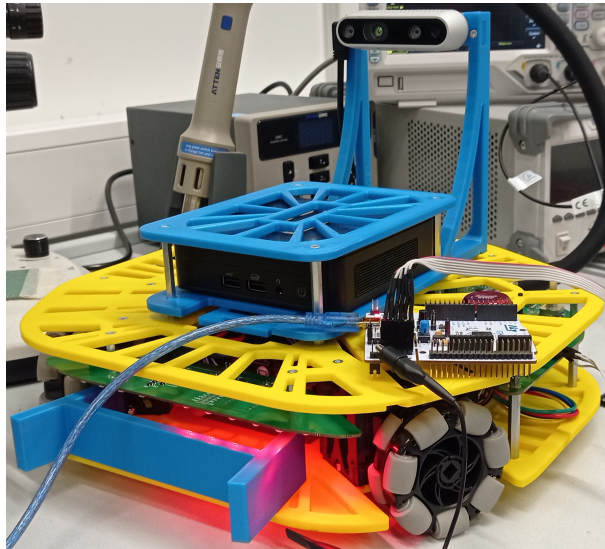
5.1 Teegi valik

Valgustite juhtimiseks oli vaja leida teek, millega saab juhtida WS2812B [2] adresseeritavaid LED valgusteid. Teek pidi olema loodud STM32 mikrokontrolleritel kasutamiseks, sest Robotondi 60 LED valgustit on ühendatud STM32F4 juhtkontrolleri külge.

Kuna Robotondi juhtkontrolleri püsivara põhitsükkel töötab sagedusel 50 Hz, seati autori poolt eesmärk, et valgustite uuenduse kood ei võtaks rohkem aega kui 5 ms. See jätkas vähemalt 75% kogu põhitsükliks ettenähtud ajast teiste ülesannete täitmiseks, et vältida teiste roboti funktsionaalsuste töö segamist. Selle eesmärgi täitmiseks teegi valikul otsiti teeki, mis kasutaks võimalikult efektiivselt ära juhtkontrolleri riistvaralist võimekust.

Näiteks on Robotondi juhtkontrolleris olemas eraldi DMA (*Direct Memory Access*) otsemälupöörduse kontroller, mis suudab vahendada andmed otse mälu ja seadme vahel ilma protsessori sekkumiseta. See oli ka põhiliseks kaalutluseks teegi valimisel. Teegiks valiti STM32-ARGB-DMA [23], mis kasutab valgustite juhtimiseks mikrokontrolleri DMA katkestusi ja taimerit PWM (*Pulse Width Modulation*) ehk pulsilaiusmodulatsiooni funktsionaalsust, et säästa protsessori aega. Teegis valgustite väärtusi muutes teisendatakse värvide bittide väärtused kaheks PWM täiteteguriks olenevalt kas tegemist on ühe või nulliga. Ühe puhul täitetegur on 56% ehk 0,7 μ s ja nulli puhul on täitetegur 28% ehk 0,35 μ s. Teisendatud täitetegurite väärtused kirjutatakse puhvrissi. Seejärel DMA võtab sellest puhvrast väärtused ning annab need taimerile, millega genereeritakse valgustite juhtimiseks PWM signaal sagedusega 800 kHz. Kuna DMA kontroller tegeleb ise andmete edastamisega taimerile, siis saab protsessor samal ajal tegutseda muude käskude täitmisega. STM32-ARGB-DMA teegis on olemas funktsioonid, millega saab muuta valgustite värvi kas üksaaval või kõikidel valgustitel korraga. Lisaks saab värvi seada nii RGB kui ka HSV värviruumis. Töö käigus mõõdeti STM32-ARGB-DMA teegi abil kõikide valgustite uuendamist, mis võttis protsessori aega 1 ms.

Antud teegi jaoks on vaja, et taimer töötaks sagedusega 32 MHz [23]. Robotondis on valgustite andmete DIN sisend ühendatud 1. taimerit 4. kanali külge, mille taksagedus sõltub APB2(*Advanced Peripheral Bus*) sagedusest. Teegi korrektseks toimimiseks, muudeti see seniselt 16 MHz pealt 32 MHz peale. Esimene kord teegi abil valgustite juhtimiseks kasutamist on näha joonisel 5.1.



Joonis 5.1: STM32-ARGB-DMA teegi abil esimesed 7 valgustit seatud punaselt põlema

5.2 Püsivara muudatused

Robotondi pardaarvuti suhtlus STM32 mikrokontrolleriga toimub üle USB ühenduse. Sõnumid algavad tähistega ja nendele järgnevad koolonitega eraldatud parameetrid. Tähisteks on kaks sümbolit näiteks RS, millega saab muuta roboti kiirust. Käsud töödeldakse mikrokontrolleri püsivaras. Antud bakalaureuse töö käigus lisati valgustitega seonduvad käskude tähised (LD, LM, LS) ja nende käskude töötus. Loodud käsud koos nende parameetritega on välja toodud tabelis 5.1.

Tabel 5.1: Loodud käskude vormistused

Käsk	1. Parameeter	2. Parameeter	3. Parameeter	4. Parameeter	...	N parameeter
LD	Indeks	Punane	Roheline	Sinine		
LS	Alguse	Lõpp	Värv 1	Värv 2	...	Värv n
LM	Režiim	Parameeter 1	Parameeter 2	Parameeter 3	...	Parameeter n

LD käsuga saab muuta üksikut valgustit, andes parameetritena käsule valgusti indeks ning punase, roheline ja sinise väärtused. Näiteks saab saata käsu LD:5:255:0:0, mis muudab valgusti indeksiga 5 punaseks.

LS käsuga saab muuta valitud segmendi valgustite väärtusi. Selle parameetriteks on segmendi alguse indeks, lõpu indeks ning punase, roheline ja sinise väärtused kokku pakituna 24 bitisesse väärtusesse. Kõrgemad 8 biti on punase väärtus, keskmised 8 biti on roheline väärtus ja viimased 8 biti on sinise väärtus. Värvide väärtuseid võib olla rohkem kui üks. Kui värve on antud vahemikuga sama palju, siis järjest iga valgusti väärtus selles vahemikus muudetakse värvide järjekorras samal positsioonil olevale värvile. Kui värve on vähem, siis valgustitele, millele ei ole värve järjekorras lisatakse viimase värvi väärtused. Näiteks saates mikrokontrollerile LS:5:6:255:65280, siis muudetakse 5. valgusti siniseks ja 6. valgusti rohelineks. Kui saata mikrokontrollerile käsk LS:5:7:85280, siis pannakse valgustid indeksitel 5, 6 ja 7 põlema roheline.

Valgustuse režiimide muutmiseks on käsk LM. Sellele saab parameetritena anda režiimi numbrit ning kolm täisarvulist väärtust.

5.3 Valgustuse režiimid

Töö käigus loodi valgustuse režiimid, mis jagunevad kahte gruppi, ühed on visuaalsed režiimid, mis on mõeldud roboti atraktiivsuse tõstmiseks ja teised, mis on seotud mingi konkreetse funktsionaalsusega. Visuaalseid režiime saab roboti püsivaras või ROS-i tasemel siduda roboti olekutega. Näiteks kasutatakse ühte visuaalset režiimi, et anda märku kui roboti aku on tühjas saanud.

Valmistatud režiimide indekse ning nendele kuuluvaid parameetreid näeb tabelis 5.2.

Tabel 5.2: LM käsu režiimide parameetrid

	Režiim	Nr	1. Parameeter	2. Parameeter	3. Parameeter	4. Parameeter
Visuaalsed	NONE	0				
	SPIN	1	Punane	Roheline	Sinine	Kiirus
	PULSE	2	Punane	Roheline	Sinine	Kiirus
	COLORS_SMOOTH	3	Kiirus			
	WHEEL_COLORS	4	Kiirus			
	COLORS_SPIN	5	Kiirus			
Funktsionaalsed	MOTOR_DUTY	6				
	MOTOR_SPEEDS	7				
	SCAN_RANGES	8	Vasak	Parem	Keskmine	

5.3.1 Visuaalsed režiimid

Visuaalseteks režiimideks käesoleva töö puhul on režiimid, mis ei võta enda töötamise käigus robotilt mingisugust sisendit ja kuvavad lihtsalt ettemääratud animatsiooni.

SPIN režiimis põleb üks valgusti korraga. Aktiivne valgusti valitakse järjekorras valgusti indeksi alusel liikudes 0-st 59-ni. Kui jõutakse viimase valgustini, alustatakse uuesti indeksist 0. Selle režiimi puhul LM käsu parameetritena saadatud värvi RGB väärtused määravad põleva valgusti värvi ja kiirus määrab põleva LED-i vahetamise sageduse vahemikus 1 Hz kuni 50 Hz. Tulemuseks on animatsioon, kus määratud värv teeb ringi ümber roboti.

PULSE režiimis kõik valgustid alustavad välja lülitatuna ning hakkavad sujuvalt järjest eredamaks muutuma. Kui valgustid on jõudnud maksimaalse ereduseni, siis hakkab nende eredustase langema kuni valgustid on jälle kustus. Selle režiimi puhul LM käsu parameetritena saadatud värvi RGB väärtused määravad valgustite värvi. Kiiruse parameeter määrab animatsiooni sammu sageduse. PULSE režiimi kasutatakse ka siis, kui Robotondi aku pinget on alla ette määratud miinimum piiri. Sellisel juhul hakkavad roboti valgustid punaselt PULSE režiimis tööle, vilkudes sagedusega 2,55 Hz, et näidata aku tühja olekut.

COLORS_SMOOTH režiim paneb kõik valgustid põlema sama värvi. Režiimi käigus valgustite värv sujuvalt muutub. Režiimi loomisel kasutati ära STM32-ARGB-DMA teegi poolt pakutavat võimekust seada valgustite värve HSV värvimudelit kaudu. HSV värvimudel is tähendavad H värvitooni, S küllastust ja V eredust. Värvide muutmiseks muudetakse igal animatsiooni sammul valgustitele seatavat värvitooni alates nullist ühe väärtuse võrra kuni jõutakse värvitooni väärtuseni 255 ning pärast seda hakatakse uuesti algusest pihta. Tulemuseks on sujuv värve muutev animatsioon.

WHEEL_COLORS on režiim, mis paneb roboti mootorite taga ja kõrval olevad valgustite segmendid põlema kõik sama värviga. Režiimi käigus värvid muutuvad sujuvalt samamoodi nagu COLORS_SMOOTH režiimis. Tulemuseks on rataste taga ja ümber sujuvalt värve muutev valgustite animatsioon.

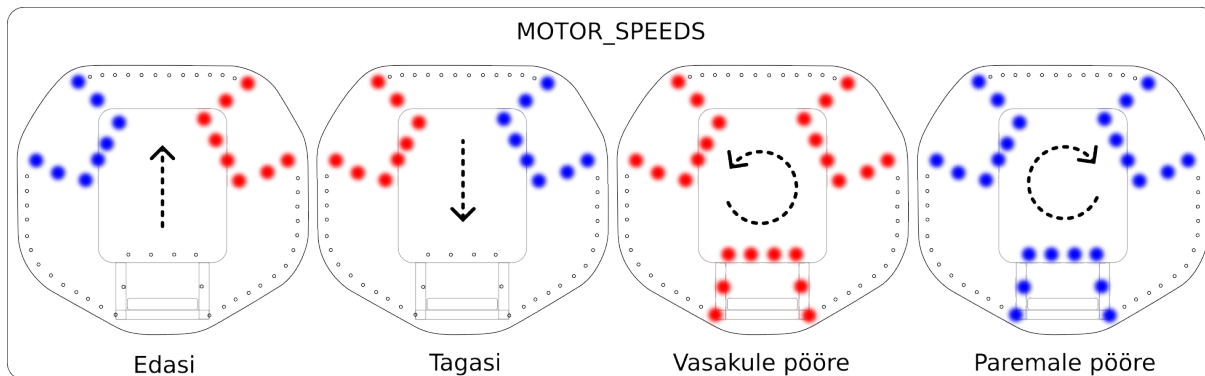
COLORS_SPIN režiimis muudetakse valgustite värvide väärtusi ühe kaupa. Alguses seatakse esimese valgusti väärtus, siis järgmise kuni jõutakse viimase valgusti väärtuse seadmiseni, pärast mida hakatakse uuesti esimesest valgustist pihta. Värvide seadmiseks kasutatakse HSV värvimudelit. Iga valgustile seatakse ühe värvitooni väärtuse võrra suurem värvitoon kuni jõutakse värvitoonini 255, mis juhul hakatakse uuesti värvitoonist 0 pihta. Tulemuseks on roboti ümber tiirlev ja värve muutev animatsioon.

Töös on loodud režiim NONE, mida valides ei ole ühtegi režiimi valitud. Seda režiimi kasutades saab valgusteid üksikhaaval või segmentide kaupa juhtida, sest siis ei juhtu sellist olukorda, kus seatakse valgustitele kindlad väärtused, aga roboti aktiivne valgustuse režiim kirjutab oma animatsiooni käigus soovitud valgustite väärtused üle. NONE režiimi valides kustutatakse ka kõik robotil olevad valgustid.

5.3.2 Funktsionaalsed režiimid

Funktsionaalseteks režiimideks on selle töö puhul režiimid, mille töö sõltub otseselt mõnest roboti anduri näidust või parameetrist. Näiteks roboti ees olevast takistuse kaugusest või roboti liikumiskiirusest.

MOTOR_SPEEDS on režiim, mis näitab mootorite taga ja kõrval olevate valgustitega mootori kiiruseid. Kui mootor pöörleb päripäeva, siis on vastava mootori ümbruses olevad LED-id punased ja kui vastupäeva, siis on need sinised. Värvide erendus näitab mootori kiirust vahemikus 0 kuni 1 m/s. Joonisel 5.2 on näidatud milliste mootorite valgustid ja milliste värvidega põlevad kui robot teeb erinevaid manöövreid. Joonisel 5.3 on näha, millised roboti rataste juures olevad valgustid välja näevad, kui robot teeb selles režiimis vasakule pöört.



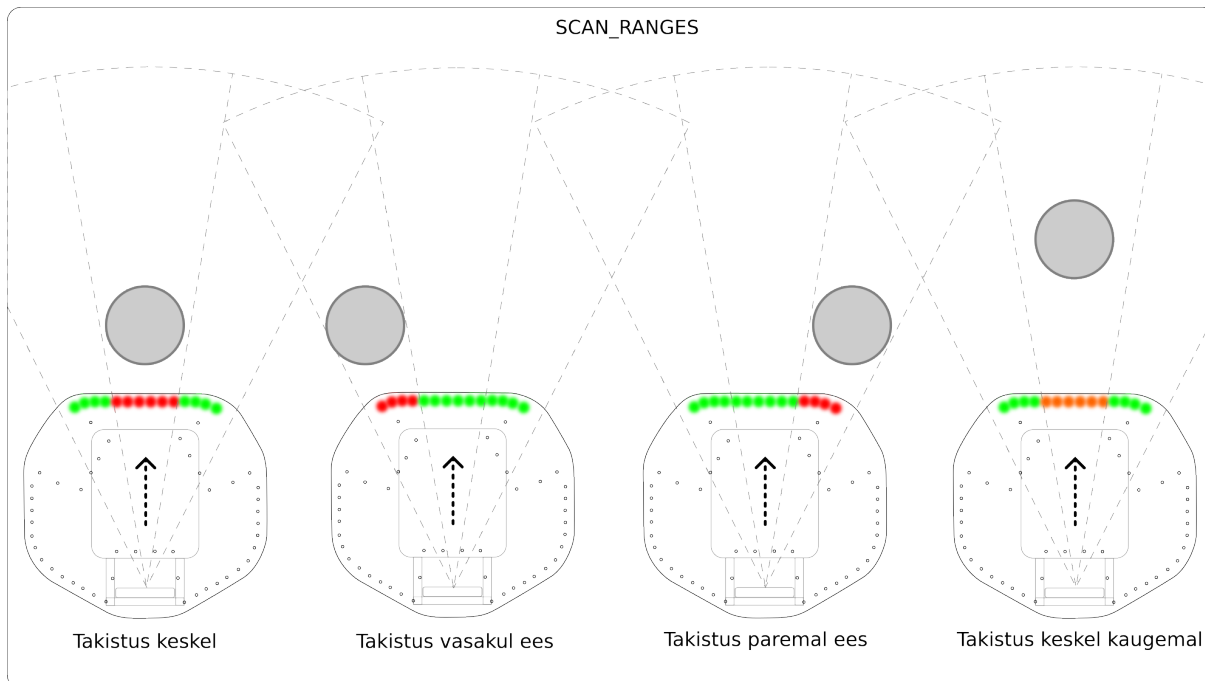
Joonis 5.2: Mootori kiiruste režiim

MOTOR_DUTY režiim näitab mootorite taga ning kõrval olevate valgustitega mootorite PWM signaali täitetegurit. Selles režiimis töötavad valgustid sarnaselt MOTOR_SPEEDS režiimile. Täitetegur kuvatakse valgustil kustus olevatest kuni täis eredusel põlevate valgustitega ning valgusti värvid on mootori ühes suunas pöörlemise puhul sinised ning teises suunas punased. See näitab kui suure jõuga ja millises suunas mootor pöörleb. Kui robot seisab ja seda käega edasi lükata, siis mootorid hakkavad vastassuunas vastu suruma, et taastada mootorite algne positsioon. Seda tehes on näha mootorite juures olevate valgustitega, millises suunas mootorid pöörelda proovivad.



Joonis 5.3: Robotont MOTOR_SPEEDS režiimis vasakule pööramas

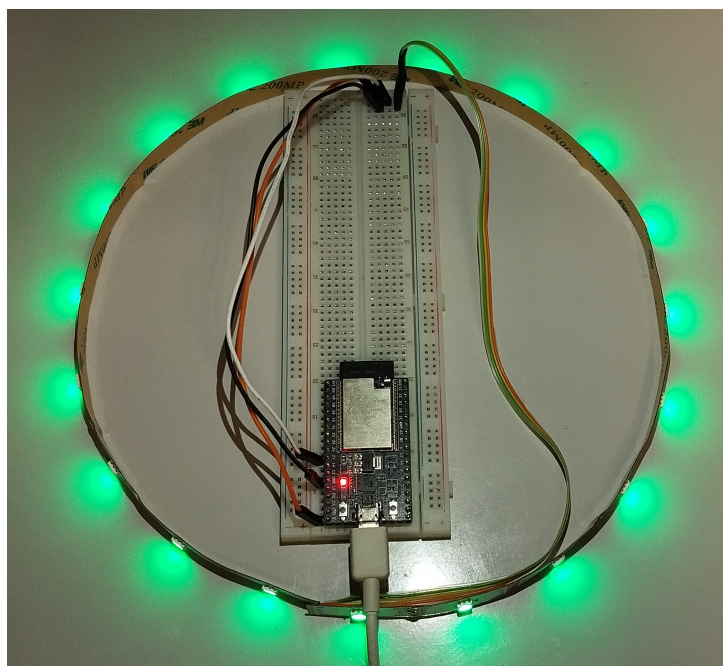
SCAN_RANGES on valgustite režiim, mis näitab roboti ees olevate valgustitega kolmes sektoris tuvastatud takistuste kauguseid. Kauguseid kuvatakse rohelisena, kui sektoris tuvastatud objekt on kaugel ja punasena siis, kui objekt on lähedal. Keskmisel distantsil olevad takistuste kuvamise värv jääb punase ja rohelise vahele. Joonisel 5.4 on näha kuidas selle režiimi puhul valgustid töötavad.



Joonis 5.4: Joonisel on kuvatud SCAN_RANGES režiimi töö näidis. Nool näitab roboti vaate suunda.

Lisaks loodud režiimidele seadistati Robotondi püsivara nii, et roboti sisse lülitamisel kuvavad valgustid esimesed 2 sekundit roboti aku pinget, vahemikus 13 V kuni 18 V, värvidega punasest roheliseni.

Töö käigus erinevate režiimide loogika välja töötamiseks kasutati ESP32 mikrokontrollerit ja WS2813B adresseeritavate valgustitega valgustiriba. See aitas kiiresti lahendust katsetada ja proovida palju erinevaid lahendusi. Katsetamiseks kasutatud komponente on näha joonisel 5.5.



Joonis 5.5: ESP32 arendusplaat ja WS2813B valgustiriba

5.4 Robotondi ROS2 draiveri pistikprogramm

Töös said loodud valmis režiimid ja käsud valgustite juhtimiseks ning režiimide muutmiseks. Siin alampeatükis kirjeldatakse loodud Robotont draiveri pistikprogrammi, mis võimaldab ROS-i sõnumite saabumisel edastada STM32 mikrokontrollerile käske valgustite juhtimiseks. Läbi selle Robotont draiveri pistikprogrammi muutub valguslahendus ROS-i kaudu juhitavaks.

Robotont draiver on ROS2 Humble Hawksbill distributsiooni jaoks loodud kimp. Robotondi draiveril on olemas funktsionaalsus saata sõnumeid STM32 mikrokontrollerile ning võtta vastu sellelt mikrokontrollerilt tulenevaid sõnumeid. Selles on olemas mootorite pistikprogramm, mille ülesandeks on võtta vastu `cmd_vel` rubriigi ROS-i sõnumeid, mis sisaldavad roboti kiiruseid ja vormistada need sõnumiks STM32 mikrokontrollerile ning saata need mikrokontrollerile. Drivers on olemas ka odomeetria pistikprogramm, mis võtab vastu mikrokontrollerilt tulevaid odomeetria sõnumeid ning saadab need ROS-i `nav_msgs/Odometry` sõnumina rubriiki `/odom`.

Antud töö käigus loodi Robotont draiverile valguslahenduse juhtmoodul nimega `plugin_led_module`. Mooduli ülesanne on tekitada valguslahenduse jaoks sobivad ROS-i rubriigid, kuulata nendel saabuval sõnumeid ning vormistada käsk, millega saadetakse andmed edasi STM32 juhtkonrollerile. Loodud moodul kuulab kolme sõnumi tüüpi: `led_pixel`, `led_segment` ja `led_mode`.

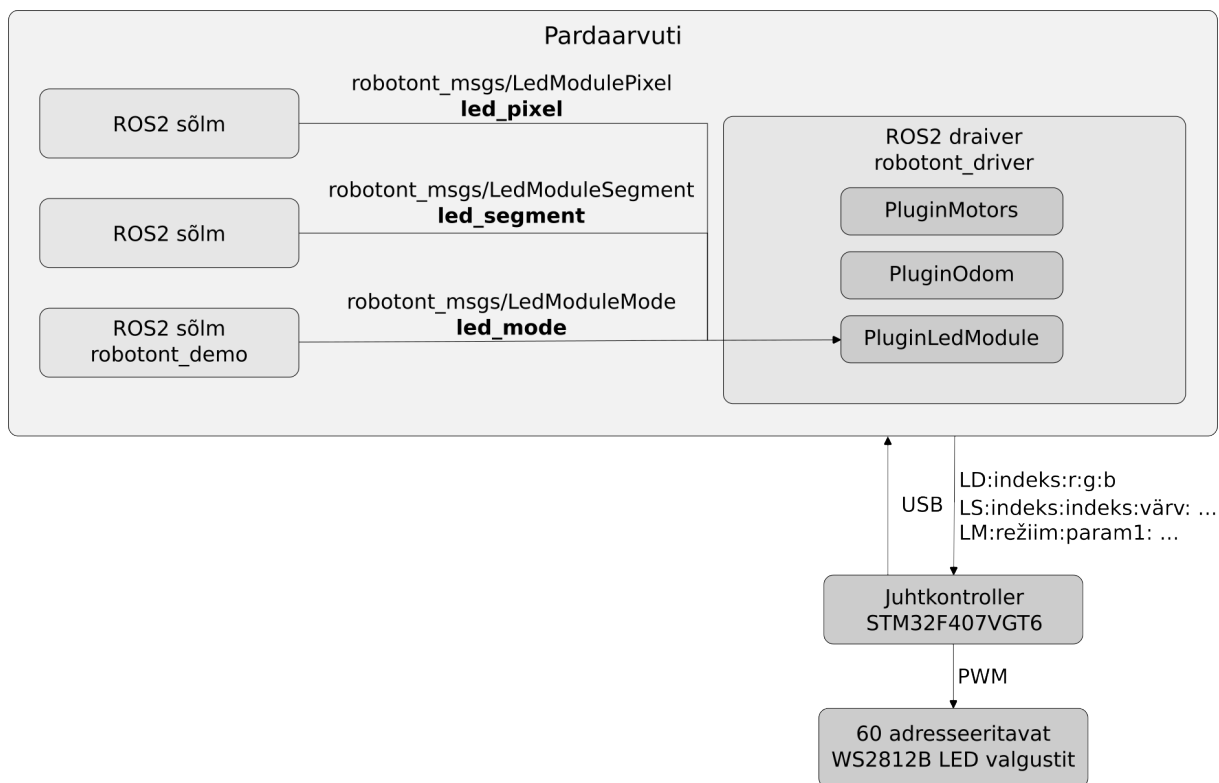
Kuna ROS-is puuduvad standardsõnumitena sellised sõnumid, mis sobiks valgustite juhtimiseks, siis vajalikud sõnumid valmistati töö käigus ja need on osa `robotont_msgs` kimbust. Sõnumite aluseks võeti ROS1 draiveri jaoks loodud valgustitega seonduvaid sõnumeid, mis loodi Ranno Mäesepa bakalaureuse töö raames [24]. Sõnumeid kohendati lisandunud funktsionaalsuse jaoks ning lisandus režiimide sõnum. Värvide väärtuste jaoks kasutati `ColorRGB` sõnumit. `ColorRGB` koosneb kolmest märgita 8 bitisest täisarvust, milleks on punase, roheline ja sinise värvi väärtused.

Sõnumi `led_pixel` sisuks on valgusti indeks ja `ColorRGB` sõnumina värv, mis koosneb RGB väärtustest. Sellega saab seada ühe valgusti värvi.

Sõnumi `led_segment` sisuks on algus ja lõpu indeksid ja `ColorRGB` värvide sõnumite massiiv, mis kooneb RGB väärtustest, millega määratakse antud vahemikus olevate valgustite värvid. Valgustite moodulis pakitakse RGB värvid ühte 24 bitilisse arvu kokku, et pikemate värvijadade saatmisel oleks vähem informatsiooni vaja saata.

Sõnumi `led_mode` sisuks on režiimi number, mis on esitatud märgita 8 bitise täisarvuna. Sõnumisse loodi ka iga režiimi numbrist vasteks režiimi nimi nii, et kui sõnumit kirjutada, siis saab näiteks kasutada `sõnum.mode = sõnum.SPIN`. See asendab sõnumit saates režiimi nime selle numbriga, milleks `SPIN` puhul on 1. Režiimide numbrid on nähtavad tabelis 5.2. Pärast režiimi numbrit on sõnumis parameetrite massiiv, mis koosneb 16 bitistest täisarvudest. Töö käigus kirjeldati `led_mode` sõnumi failis iga režiimi parameetreid, et need oleks dokumenteeritud ning sõnumit kasutades saaks vaadata, millised parameetrid erinevatel režiimidel on vaja sõnumiga koos saata.

Loodud sõnumite teekonda on kujutatud joonisel 5.6.



Joonis 5.6: Robotondi valgustuse jaoks vajalike sõnumite teekond

5.5 ROS2 kimp valgustuse demonstreerimiseks

Töö lõpuks oli vaja demonstreerida valgustite tööd ning kontrollida kas kõik loodud komponendid omavahel töötavad. Selle jaoks loodi töö käigus ROS2 kimp nimega `demo_lightning`, mis sisaldab kolme sõlme. Üks seab valgustitele järjest erinevaid režiime, et neid demonstreerida. Teine sõlm kuulab `simple_ranges` sõnumeid, mis tulevad Robotondil olevalt RealSense kaameralt läbi `depthimage_to_laserscan` ja `laserscan_to_ranges` sõlmede. Need sõnumid sisaldavad roboti ees oleva kolme sektori sees olevate objektide kaugusi. Kolmas sõlm võtab kasutajalt sisendit ja saadab nende abil vastavad valgustuse sõnumid. Sisendit on samad, mis on välja toodud tabelis 5.1. Näiteks indeksil 5 oleva valgusti roheliseks muutmiseks saab sisestada `LD:5:0:255:0`. Nende sõlmedega kasutatakse kõiki loodud sõnumeid ning demonstreeritakse, et valgusteid on võimalik juhtida ROS-i kaudu.

6 Kokkuvõte

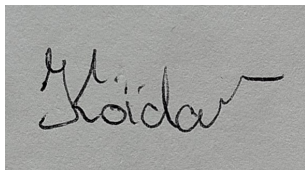
Bakalaureusetöö eesmärgiks oli luua tarkvara kolmanda põlvkonna õpperoboti Robotont valguslahenduse jaoks. Töö käigus täiendati Robotondi STM32 mikrokontrolleri püsivara, lisades sinnatarkvara, mille abil saab valgusteid juhtida individuaalselt või segmentide kaupa. Loodi ka 9 eelprogrammeeritud valgustuse režiimi, millest 6 on visuaalsed ja 3 funktsionaalsed režiimid. Loodud režiime kasutatakse erinevate roboti olekute kuvamiseks, näiteks kuvatakse kiiresti vilkuvate punaste valgustitega aku tühja olekut. Valgustite ja režiimide valimiseks loodi käsud püsivarasse. Loodud valguslahenduse püsivara kasutab valgustite juhtimisel otsemälupöördust, mistõttu võtab kõigi 60 valgusti uuendamine programmi põhitsüklist aega vaid 1 ms ja see ei sega muud püsivara funktsionaalsust.

Valguslahenduse juhtimiseks Robotondi parदारvutist koostati Robotondi ROS2 draiverile pistikprogramm, mis võtab vastu valgustitega seonduvaid sõnumeid ja vormistab ning saadab need STM32 mikrokontrollerile. Lõputöö eesmärk sai täidetud.

Bakalaureusetöö käigus valmis Robotondi 3. generatsiooni õpperoboti valguslahenduse juhtimiseks tarkvara. Loodud tarkvara abil on võimalik valgusteid juhtida ROS-i vahendusel. Töö käigus valmisid erinevad režiimid roboti püsivaras, mida saab ka läbi ROS-i seada ning nendele parameetreid anda.

Tänuavaldused

Veiko Vunder, juhendamise ja nõu eest kogu töö jooksul.

A square image containing a handwritten signature in black ink on a light gray background. The signature appears to be 'M. Koida'.

Viited

- [1] “The Difference Between Addressable RGBIC LED Strip WS2811, WS2812B, WS2813, WS2815, SK6812, SK9822,” <https://www.derunledlights.com/the-difference-between-addressable-rgb-led-strip-ws2811-ws2812b-ws2813-ws2815-sk6812-sk9822/>, (Vaadatud: 30. aprill 2024).
- [2] “WS2812B Datasheet,” <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>, (Vaadatud: 14. aprill 2024).
- [3] K. Baraka and M. M. Veloso, “Mobile service robot state revealing through expressive lights: formalism, design, and evaluation,” *International Journal of Social Robotics*, vol. 10, pp. 65–92, 2018.
- [4] E. Cha, T. Trehon, L. Wathieu, C. Wagner, A. Shukla, and M. J. Matarić, “Modlight: designing a modular light signaling tool for human-robot interaction,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1654–1661.
- [5] R. Fernandez, N. John, S. Kirmani, J. Hart, J. Sinapov, and P. Stone, “Passive demonstrations of light-based robot signals for improved human interpretability. in 2018 27th ieeeroman,” 2018.
- [6] D. Szafir, B. Mutlu, and T. Fong, “Communicating directionality in flying robots,” in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 2015, pp. 19–26.
- [7] Tartu Ülikool, “Robotont,” <http://robotont.ut.ee/>, (Vaadatud: 11. aprill 2024).
- [8] “ROS Introduction,” <http://wiki.ros.org/ROS/Introduction/>, (Vaadatud: 24. märts 2024).
- [9] M. Quigley, B. Gerkey, and W. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O’Reilly Media, 2015. [Online]. Available: <https://books.google.ee/books?id=Hnz5CgAAQBAJ>
- [10] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
- [11] O. Robotics, “ROS 2 Documentation,” <https://docs.ros.org/en/rolling/index.html>, (Vaadatud: 10. mai 2024).
- [12] T. R. Back-End, “Ros1 vs ros2, practical overview for ros developers,” https://roboticsbackend.com/ros1-vs-ros2-practical-overview/#Why_ROS2_and_not_keep_ROS1, (Vaadatud: 11. aprill 2024).

- [13] F. M. Rico, *A Concise Introduction to Robot Programming with ROS2*. Chapman and Hall/CRC, 2022.
- [14] “ROS2 Distributions,” <https://docs.ros.org/en/rolling/Releases.html>, (Vaadatud: 30. aprill 2024).
- [15] P. Burgess, “The Magic of NeoPixels,” <https://learn.adafruit.com/adafruit-neopixel-uberguide/>, (Vaadatud: 30. aprill 2024).
- [16] “SK6812 VS WS2812B,” <https://www.superlightingled.com/blog/sk6812-vs-ws2812b/>, (Vaadatud: 30. aprill 2024).
- [17] “SK6812RGB Datasheet,” <https://cdn-shop.adafruit.com/product-files/1138/SK6812+LED+datasheet+.pdf>, (Vaadatud: 10. mai 2024).
- [18] “OLED vs LCD,” <https://newhavendisplay.com/blog/oled-vs-lcd/>, (Vaadatud: 30. aprill 2024).
- [19] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz, J. K. Aggarwal, R. Mooney, and P. Stone, “Bwibots: A platform for bridging the gap between ai and human–robot interaction research,” *The International Journal of Robotics Research*, 2017. [Online]. Available: <http://www.cs.utexas.edu/users/ai-lab?khandelwal:ijrr17>
- [20] W. Jo, J. Kim, R. Wang, J. Pan, R. K. Senthilkumaran, and B.-C. Min, “Smartmbot: A ros2-based low-cost and open-source mobile robot platform,” *arXiv preprint arXiv:2203.08903*, 2022.
- [21] S. Szantula, “ROSbot with WS2812B LEDs signalization,” <https://husarion.com/tutorials/other-tutorials/rosbot-and-ws2812b-led-signalization/>, Husarion. (Vaadatud: 4. november 2023).
- [22] R. Raudmäe, “Avatud robotplattvorm robotont,” <http://hdl.handle.net/10062/64341>, 2019.
- [23] D. Semenov, “STM32-ARGB-DMA,” <https://github.com/Crazy-Geeks/STM32-ARGB-DMA>, (Vaadatud: 14. aprill 2024).
- [24] R. Mäesepp, “Takistuste vältimise lahendus õpperobotile robotont,” <http://hdl.handle.net/10062/72100>, 2020.

Lisad

Lisa 1

Töö käigus loodud tarkvara osad on Github koodihoidlas aadressil: <https://github.com/ut-ims-robotics/koidam-thesis-2024-robotont-lighting>

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Raimo Köidam

1. annan Tartu ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

“Valguslahenduse tarkvara väljatöötamine õpperobotile Robotont”

mille juhendaja on Veiko Vunder

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace'i kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Raimo Köidam

20.05.2024