

Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Tehnoloogiainstituut

Andres Sakk

**Avatud robotplatvormi Robotont 3 kasutajaliidese
väljatöötamine**

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:

PhD Veiko Vunder

Tartu 2024

Resüme

Avatud robotplatvormi Robotont 3 kasutajaliidese väljatöötamine

Tartu Ülikooli tehnoloogiainstituudis arendatakse haridusrobotika platvormi Robotont. Robotondi uusim versioon, Robotont 3 hõlmab endas uut trükkplaati, millel on ekraan ja koodernupp kasutajaliidese realiseerimiseks, kuid vastav püsivara oli puudu. Käesoleva töö eesmärk oli lisada roboti püsivarasse kasutajaliidese moodul, mis võimaldaks juhtida roboti tööd ja roboti pardaarvutile käske saata. Töö käigus uuriti erinevat tüüpi kasutajaliideseid, et järgida kasutajaliideste häid tavasid. Töö tulemusena valmis modulaarse struktuuriga kasutajaliides, mis kasutab mikrokontrolleri töökoormuse vähendamiseks otsemälupöördumise kontrollerit. Edasiste arendustööde käigus on võimalik luua liidestus kasutajaliidese ja roboti pardaarvutis töötava tarkvara ROS vahel.

CERCS: T125 Automatiseerimine, robotika, juhtimistehnika

Märksõnad: Robotont, kasutajaliides, haridusrobotika

Abstract

Development of a user interface for the open robotics platform Robotont 3

A workgroup in the Institute of Technology of the University of Tartu is developing an educational robotics platform named Robotont. Recently, the development of the third generation of Robotont has begun, which includes a redesign of the electronics. The control board now supports a screen module and an encoder knob. The goal of this thesis was to implement firmware for a user interface designed for the added devices. A study of user interfaces of various types of electronics was conducted and the best practices learned were kept in mind during the design process. The new user interface has expanded Robotont's functionality and allows for integration with the Robot Operating System framework running on the onboard computer.

CERCS: T125 Automation, robotics, control engineering

Keywords: Robotont, user interface, educational robotics

Sisukord

Resümee	2
Abstract	3
Jooniste loetelu	5
Lühendid, mõisted	6
1 Sissejuhatus	7
2 Ülevaade Robotondi platvormist	8
2.1 Robotondi ajalugu ja arendus	8
2.2 Robotont 3 riistvaraline konfiguratsioon	9
2.3 Robotont 3 kvadratuurkoodernupp	10
2.4 Robotont 3 ekraanimoodul	11
3 Ülevaade kasutajaliidestest	12
3.1 Robotite kasutajaliidesed	12
3.2 Muude seadmete kasutajaliidesed	14
4 Töö eesmärk ja nõuded	16
4.1 Töö eesmärk	16
4.2 Nõuded lahendusele	16
5 Robotondi kasutajaliidese arendus	17
5.1 Kasutajaliidese disain	17
5.2 Kasutajaliidese teostuse kirjeldus	18
Koodernupp	18
Ekraanile joonistamine	18
Menüü	19
Kasutajaliideseest parदारvutisse käskude saatmine	20
5.3 Ülevaade loodud kasutajaliideseest	21
6 Kokkuvõte	23
Viited	24
Lisad	27
Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	28

Jooniste loetelu

Joonis 1. Robotont 3

Joonis 2. Robotondi kasutajaliidese sisend- ja väljundseadmed

Joonis 3. Kvadratuurkoodri aegdiagramm ühes suunas keeramisel

Joonis 4. EV3 Mindstorms kasutajaliides

Joonis 5. Kuvatõmmis Duckiebot õpperoboti videost

Joonis 6. Kuvatõmmis KettyBoti tutvustavast videost

Joonis 7. Marlini kasutajaliides

Joonis 8. Nutikellade kasutajaliidesed

Joonis 9. Kasutajaliidese struktuur

Joonis 10. Pardaarvutisse kasutajaliidest käskude saatmise andmevoog

Joonis 11. Vasakul staatuseakna vaade, paremal peamenüü vaade

Joonis 12. Vasakul näide infoaknast, paremal arvulise väärtuse muutmise vaade

Lühendid, mõisted

DMA - *Direct Memory Access* ehk otsemälupöördumine

EXTI - *external interrupt* ehk väline katkestus

GPIO - *general purpose input-output* ehk mitmeotstarbeline sisend-väljund

PC - *Inter-Integrated Circuit*, jadasiin seadmete ühendamiseks manussüsteemides

LCD - *Liquid Crystal Display* ehk vedelkristallekraan

NUC - *Next Unit of Computing*, Inteli miniarvutite seeria

OLED - *organic light-emitting diode* ehk orgaaniline valgusdiod

ROS - *Robot Operating System*, vabavaraline tarkvararaamistik robotsüsteemidele

1 Sissejuhatus

Haridusrobotika turul on puudus platvormidest, millega õpetada edasijõudnud robotika õppijatele Robot Operating System tarkvararaamistikku. Selle probleemi lahendamiseks on loodud Tartu Ülikooli tehnoloogiainstituudis robotikaplatvorm Robotont, millele arendatakse kolmandat versiooni. Kolmanda Robotondi versiooni elektroonika võimaldab juhtida roboti trükkplaadil oleva mikrokontrolleri tööd kasutades kasutajaliidese jaoks mõeldud ekraani ja koodernuppu. Praegu puudub võimalus roboti pardaarvutisse kaugühendust tegemata roboti kohta infot kuvada ja roboti tööd juhtida. Kuna Robotonti kasutatakse tihti ka arvutitehnika eriala ja laiemalt inseneeriavaldkonna tutvustamiseks, siis oleks mugav, kui tutvustavate töötubade läbiviijatel oleks lahendus roboti funktsioone demonstreerida ilma suurema seadistusega.

Töö eesmärk on rakendada uue Robotont 3 trükkplaadil olevaid kasutajaliidese jaoks mõeldud seadmeid ja kirjutada kasutajaliidese tarkvara, et laiendada Robotondi funktsionaalsust, võimaldada edasine arendustöö ja tugi keerulisema konfiguratsiooni sätestamiseks vajaduseta kasutada välist arvutit. Töö eesmärgi saavutamiseks uuritakse teiste haridusrobotika platvormide ja muude seadmete kasutajaliideseid, et kasutada kasutajaliideste disainis levinumaid häid praktikaid.

Töö käigus valmib kergesti laiendatav kasutajaliidese tarkvara ja dokumentatsioon tulevastele arendajatele, kes soovivad luua liidestust roboti kere pealt juhitava kasutajaliidese ja roboti pardaarvutis töötava ROS tarkvara vahel. Kasutajaliidesele luuakse võimekus saata pardaarvutisse käsurea käske ning nende abil potentsiaalselt vahetada tööprogramme kasutades konteinertehnoloogiat pardaarvutis.

2 Ülevaade Robotondi platvormist

2.1 Robotondi ajalugu ja arendus

Robotont on Tartu Ülikoolis arendatav avatud robotikaplatvorm (joonis 1). Mõiste “avatud platvorm” tähendab seda, et igaüks, kes soovib endale robotit hankida, saab selle ise kokku panna. Roboti riistvara koostamiseks vajaminevad 3D-mudelid ja tehnilised joonised ning trükkplaadi tootmiseks vajalikud failid on võimalik alla laadida Robotondi veebilehelt. Robotondi tarkvaralises lahenduses kasutatakse vaid avatud lähtekoodiga tasuta kättesaadavat tarkvara. [1]



Joonis 1. Robotont 3

Robotondi arendus sai alguse soovist tekitada mobiilne robot, mis võimaldaks robotika kursustel õpetada tarkvararaamistikku ROS (Robot Operating System), mis on robotitele tarkvara arendamisel enim kasutatud lahendus nii haridusasutustes kui tööstuses [2]. Robotondi arendajad nägid turul saadaval olevates robotikaplatvormides puudust. Kuigi eksisteerivad töökindlad ning lihtsustatud õpperobotid, näiteks Tartu Ülikooli Robotika I

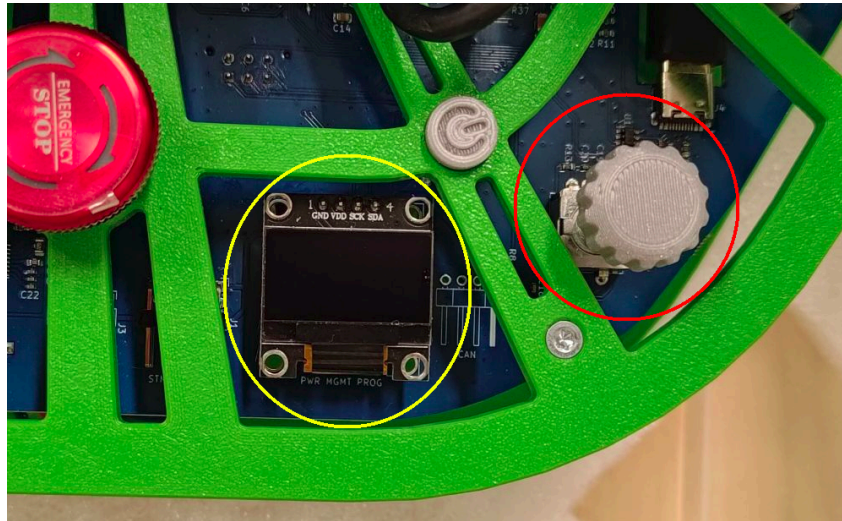
kursusel kasutatav GoPiGo [3], peamiselt esimesele kahele kooliastmele mõeldud Edison [4] ja Lego Mindstorms [5], on need edasijõudnud õppijate jaoks sobimatud, sest algajatele mõeldud robotikaplatvormid on tihti loodud eesmärgiga õpetada programmeerimise põhitõdesid ja algoritmikat – teemasid, millega Robotondi sihtgrupp peaks juba tuttav olema. On olemas ka keerukad platvormid (nt Clearpath Jackal [6], PAL Robotics TIAGo [7]), mis kätkevad endis põhjalikku konfigureerimisvõimalust ning võimekat riistvara, kuid sellise platvormi hankimise hind on enam kui 10 000 eurot. Nende kahe tootegrupi vahele jääb õpperobot Robotont, mille ehitamise kulu on suhteliselt väike, kuid mis on nii tarkvaraliselt kui riistvaraliselt piisavalt keerukas ja konfigureeritav ning seetõttu sobiv edasijõudnutele robotikalahenduste õpetamiseks. [8]

Robotonti arendatakse alates 2017. aastast ning on platvorm ajaga märkimisväärselt arenenud [9]. Robotondil on 3 versiooni. Neist uusima versiooni, Robotont 3 arendus algas 2023. aastal. Robotont 3 on nii mehaanika kui elektroonika poolest eelnevatest versioonidest küllalt erinev. Roboti kere tootmiseks kasutatakse 3D-printimist [10] varasema freesitud polükarbonaadi asemel [9]. Robotondi elektroonika paikneb nüüdsest ühtsel trükkplaadil varasema modulaarse laiendusplaate kasutava disaini asemel. Robotont 3 trükkplaat võimaldab uut funktsionaalsust, sh roboti töö integreerimist kerel paikneva ekraani ja koodernupuga. Käesolev töö keskendub just selle Robotont 3 võimekuse rakendamisele.

2.2 Robotont 3 riistvaraline konfiguratsioon

Robotont on omniliikuv ratasrobot. Robotondi kerele on kinnitatud *x86* arhitektuuril põhineva protsessoriga Intel NUC-tüüpi pardaarvuti, millele on paigaldatud Linuxi Ubuntu distributsioon koos ROS tarkvararaamistikuga. Pardaarvutiga on ühendatud roboti kerel paiknev Intel RealSense D435i kaamera, millel on sügavusmõõtmise võimekus. Robotondi trükkplaadil on STM32F407 seeria ARM Cortex-M4 arhitektuuriga mikrokontroller, mida kasutatakse peamiselt pardaarvutis jooksvast ROSi tarkvarast roboti mootoritele juhtsignaalide saatmise jaoks. Mikrokontroller haldab ka muud roboti tööks vajalikke tegevusi nagu odomeetria arvutamine, suhtlus lisamoodulitega, trükkplaadi valguslahenduse ja kasutajaliidese juhtimine. Trükkplaadil on ka AVR arhitektuuriga ATtiny88 mikrokontroller roboti toiteahela haldamiseks. Robotondi trükkplaadile on joodetud üherealine nelja viiguga standardse 2.54 mm sammuga viigupäis. Pesa on ühendatud vastavalt ekraanimooduli viikude järjestusele toitepinge, maanduse ja I²C (*Inter-Integrated Circuit*) jadaliidese viikudega

mikrokontrolleril. Kasutaja sisendi andmiseks on roboti trükkplaadil kvadratuurkoodernupp (joonis 2). [11]

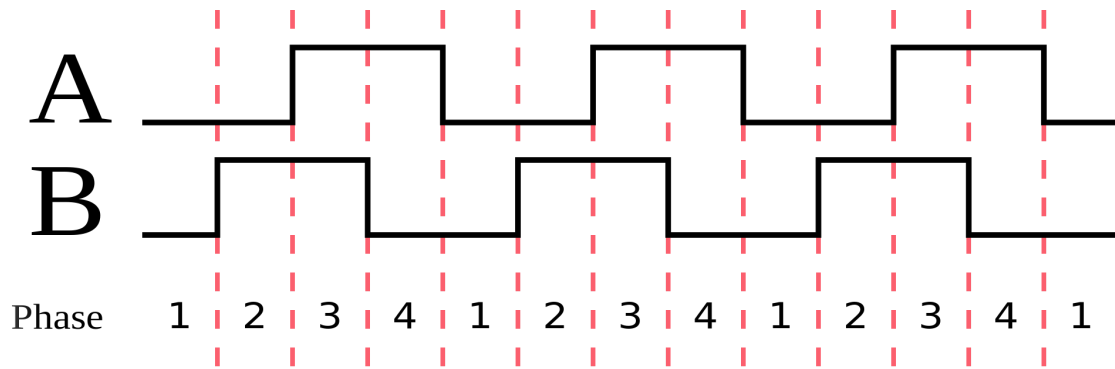


Joonis 2. Robotondi kasutajaliidese sisend- ja väljundseadmed. Kollase värviga on tähistatud ekraan, punasega koodernupp

2.3 Robotont 3 kvadratuurkoodernupp

Kvadratuurkoodernupul on mikrokontrolleriga kolm ühendust: nupp, A kanal, B kanal. Seda komponenti saab vajutada nagu tavalist surunuppu ja keerata diskreetsete sammudena lõpmatult nii päri- kui vastupäeva. Koodernupu täispöördes on 20 sammu.

Kvadratuurkoodri tööpõhimõte seisneb selles, et koodri keeramisel tekitatakse kaks kastsignaali, mille faasid on teineteise suhtes 90° võrra nihkes (joonis 3). Olenevalt sellest, kas koodrit keeratakse päri- või vastupäeva, tekitatakse kas A või B kanali kastsignaali front teisest signalist varem. Selle teadmise põhjal on võimalik mikrokontrolleril kindlaks teha, millist tüüpi sisendi kasutaja andis.



Joonis 3. Kvadratuurkoodri aegdiagramm ühes suunas keeramisel [12]

2.4 Robotont 3 ekraanimoodul

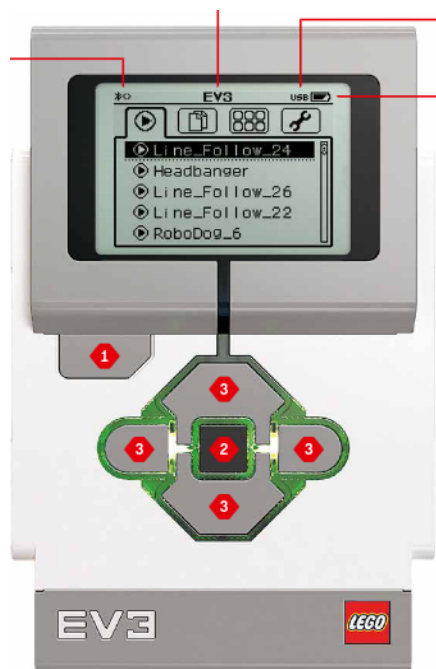
Töös loodava kasutajaliidese kuvamiseks kasutatakse SSD1306 ekraanimoodulit. SSD1306 ekraanimoodul põhineb orgaaniliste valgusdiodide (OLED) tehnoloogial ning neid mooduleid on erineva lahutusvõime, värvide ja ühendusliidese [13]. Robotont 3 kasutab valgete pikslitega moodulit, mille ekraani mõõtmed on 22 mm x 11 mm ning lahutusvõime 128x64 pikslit. Ekraani juhitakse kasutades I²C ühendusprotokolli.

SSD1306 ja sarnaste OLED-ekraanide kasutamine seadmetes on mugav, sest ekraanimoodulid on laialdaselt kättesaadavad, soodsad ja OLED-tehnoloogial põhinevate ekraanid on suure kontrastsusega [14]. See tuleneb sellest, et ekraanil valgustatakse vaid üksikuid piksleid ning ekraan taustvalgustust ei vaja. Samuti ei muutu OLED-ekraanidel kuvatava pildi kvaliteet, kui vaatleja muudab vaatenurka ekraani tasandi normaali suhtes [14]. Teistel ekraanitehnoloogiatel, nt paljudes manussüsteemides kasutusel olevatel vedelkristallekraanidel on kujutis vaatenurka muutes teist värvi ja kontrastsus madalam [15].

3 Ülevaade kasutajaliidestest

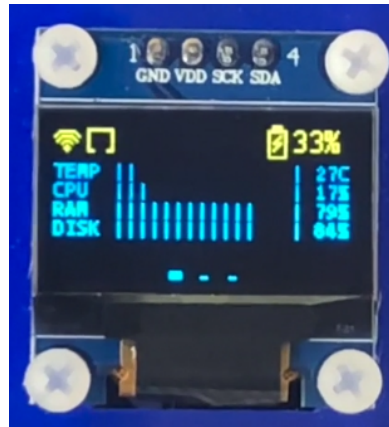
3.1 Robotite kasutajaliideseid

Robotondi kasutajaliidese teostamiseks uuriti esmalt teiste haridusrobotite kasutajaliideseid, et tutvuda ennast tõestanud lahendustega. Paljudel haridusrobotitel ei eksisteerigi välist kasutajaliidest, vaid ainuke viis roboti töö juhtimiseks on luua kaugühendus robotit juhtivasse arvutisse. Keerukamate operatsioonide teostamiseks on ka Robotondi puhul see vajalik. Õpperobotil Edison V3 on kasutajaliideseks vaid 3 programmeeritavat nuppu, mida kasutatakse roboti tööprogrammi käivitamiseks ja peatamiseks [16]. Populaarse haridusrobotika komplekti Lego Mindstorms kontrollploki on LCD, millel on kuvatud menüü erinevate tööprogrammide, failide, rakenduste ja seadistuse haldamiseks (joonis 4). Kasutajaliidest juhitakse kuue nupu abil – navigeerimisnupud ja nupud valimise ning tagasi minemise jaoks.



Joonis 4. EV3 Mindstorms kasutajaliides [17]

Õpperobot Duckiebot kasutab kasutajaliideseks sarnaselt Robotont 3 robotiga SSD1306 ekraanimoodulit, kuid seal kuvatakse vaid info roboti oleku kohta ning kasutajal robotile sisendit ilma kaugühendusega anda pole võimalik. Ekraanil kuvatakse infot võrguühenduse, aku laetuse astme, pardaarvuti jõudluse ja oleku kohta (joonis 5).



Joonis 5. Kuvatõmmis Duckiebot õpperoboti videost [18]

Tartu Ülikooli kliinikumis kasutataval teenindusrobotil KettyBot, mis pole mõeldud õppetöökseks ja arendustegevuseks, vaid vahetuks suhtluseks tavakasutajatega, on kasutusel hoopis puutekraan (joonis 6). Suuremate mõõtmetega robotitel on puutekraan kasutajaliidese hea lahendus, sest sellise kasutajaliidese juhtimine on nutitelefonide laia leviku tõttu intuitiivne.



Joonis 6. Kuvatõmmis KettyBoti tutvustavast videost [19]

3.2 Muude seadmete kasutajaliidesed

Kuna õpperobotite kasutajaliideste uurimisel Robotondiga piisavalt sarnaseid riistvarakonfiguratsioone ei leitud, vaadati etteantud sisend- ja väljundseadmete rakendamiseks teisi elektroonikaseadmete turul olemasolevaid lahendusi. Paljude vanemate 3D-printerite juhtpaneel kasutab printeri töö juhtimiseks samuti koodernuppu ning kasutajale tagasiside andmiseks ekraani [20]. Uuemad printerid kasutavad printeri juhtimiseks puuteekraani või Lego Mindstormsiga sarnast juhtnupustikku.

Tuntud 3D-printerite tootjate nagu Prusa Research ja Creality seadmete püsivarana kasutatakse modifitseeritud vabavara Marlin [21]. Marlini kasutajaliidesel kuvatakse ekraanil esmalt seadme olekut kirjeldav vaade ning nuppu vajutades hierarhiline menüü, milles on võimalik koodernuppu keerates üles-alla liikuda ning vajutades kinnitada ekraanil tähistatud valik (joonis 7). Kuna tagasi liikumiseks eraldi nuppu pole, on see paigutatud menüü esimeseks valikuks.



Joonis 7. Marlini kasutajaliides. Vasakul staatuseaken, paremal menüü vaade

Ideede leidmiseks vaadati ka nutikellasid, sest nende ekraani suurus on sarnasem Robotondi ekraani suurusega. Puuteekraanita nutikelladel on nupud menüüs navigeerimiseks, valimiseks, tagasi minemiseks. Tavaliselt on menüüdes võimalik liikuda vertikaalses suunas. Kasutaja valiku indikeerimiseks kasutatakse piirjooni, raame või värvide inverteerimist. Menüü valiku juures võib olla ka ikoon või piktogramm valiku illustreerimiseks (joonis 8).



Joonis 8. Nutikellade kasutajaliidesed. Vasakul Garmin Forerunner 255 [22] ja paremal Pebble [23]

4 Töö eesmärk ja nõuded

4.1 Töö eesmärk

Töö eesmärk on kirjutada Robotont 3 kasutajaliidese seadmete rakendamiseks püsivaramoodul, et laiendada Robotondi funktsionaalsust, võimaldada edasine arendustöö ja tugi keerulisema konfiguratsiooni sätestamiseks vajaduseta kasutada välist arvutit.

4.2 Nõuded lahendusele

Eesmärgist ja teiste Robotondi arendustiimi liikmete tööst lähtuvalt sõnastati kasutajaliidese nõuded.

Funktsionaalsed nõuded

- kasutab sisendina koodernuppu
- kasutab väljundina SSD1306 ekraanimoodulit
- peab sisaldama hierarhilist menüüd suvalise arvu alammenüüdega
- peab olema võimeline kuvama dünaamilist infot, nt aku pinget
- peab võimaldama arvuliste parameetrite muutmist
- peab võimaldama roboti valguslahenduse muutmist
- peab võimaldama parदारvutisse käskude saatmist

Mittefunktsionaalsed nõuded

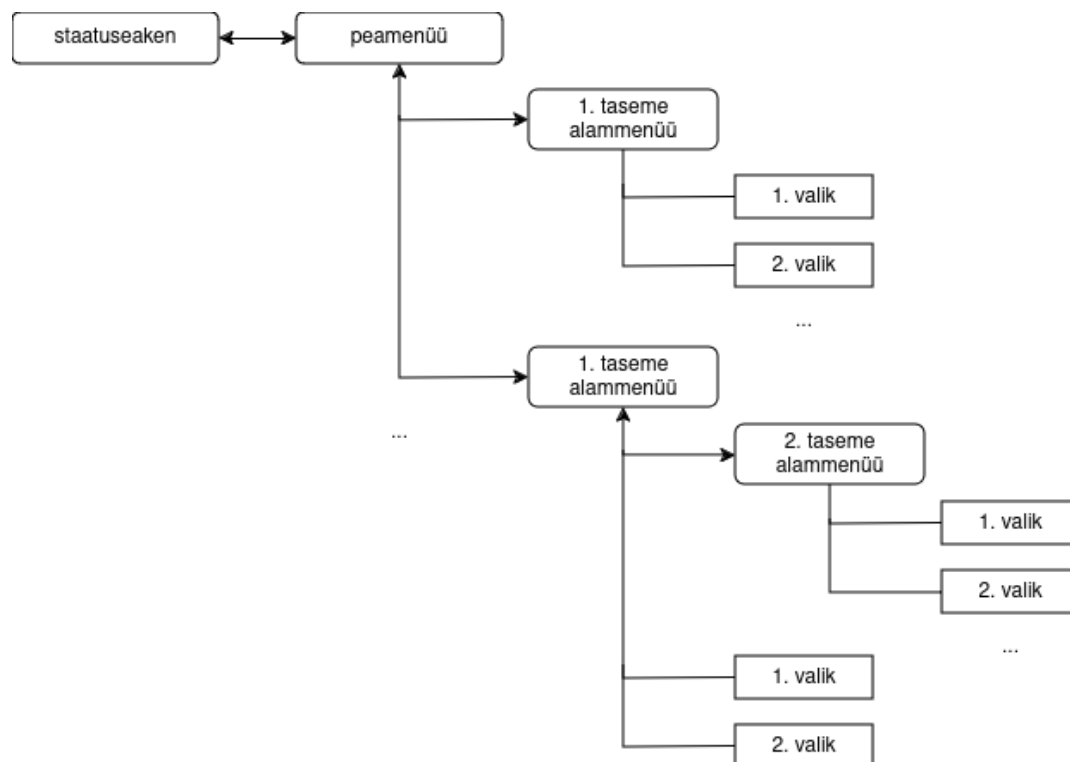
- kirjutatud programmeerimiskeeles C
- vastab Robotondi tarkvaraarhitektuurile
- vastab Robotondi koodibaasi stiilinõuetele
- peab olema dokumenteeritud
- peab olema teistele arendajatele laiendatav
- kasutajaliidese seotud tegevused peavad olema programmeeritud sedavõrd ressursisäästlikult, et nad ei häiriks roboti teisi ülesandeid, nt roboti liikumist

5 Robotondi kasutajaliidese arendus

5.1 Kasutajaliidese disain

Sisend- ja väljundseadmete sarnasuse ja püstitatud nõuete täitmiseks sobiva struktuuri tõttu otsustati jäljendada loodava kasutajaliidese Marlini kasutajaliidest. Laialdaselt levinud kasutajaliidese jäljendamine on otstarbekas, kuna paljudel inseneeriatudengitel on varasem kogemus 3D-printerite kasutajaliideste juhtimisega.

Roboti käivitamisel kuvatakse kasutajale staatuseaken, kus näidatakse enim vajaminevat informatsiooni. Staatuseaknast saab siseneda menüü vaatesse, kus asuvad alammenüüd roboti erinevate funktsioonide konfigureerimiseks, arvuliste parameetrite muutmise valikud ja lisainfo valikud (joonis 9).



Joonis 9. Kasutajaliidese struktuur

Koodernupu abil on kasutajal võimalik anda kasutajaliidesele kolme tüüpi sisendit: nupu keeramine päri- ja vastupäeva ning nupu vajutamine. Vahel kasutatakse piiratud sisendseadmete komplekti puhul nupu topeltvajutusi, eristatakse lühikesi ja pikki vajutusi. Selliste sisendsignaali töötlemine oleks olnud keerukam ning ka kasutajale vähem intuiitiivne. Topeltvajutuste rakendamise puhul väheneb kasutajaliidese reageerimisvõime, sest üksikvajutuse korral tuleb oodata püsivaras sätestatud ajahulka, et kindlaks teha, et kasutaja ei teosta topeltvajutust. Keerukamate sisendsignaali rakendamata jätmisel on ka kasutajal lihtsam, sest eraldi dokumentatsiooni kasutajaliidese opereerimise kohta pole vaja lugeda. Seetõttu valiti nupu keeramine menüüs navigeerimise ja nupu vajutamine valiku kinnitamise sisendiks.

5.2 Kasutajaliidese teostuse kirjeldus

Kasutajaliidese arendus jaotati etappidesse:

- koodernupu sisendi töötlemine
- ekraanile draiveri abil joonistamine
- menüü andmestruktuuri ja loogika kirjutamine

Koodernupp

Koodernupu kolm viiku on ühendatud mikroprotsessori lisafunktsionaalsuseta GPIO (*general purpose input-output*) viikude külge. Sisendi tuvastamiseks konfigureeriti püsivaras protsessori viikude EXTI (*external interrupt*) võimekus ehk signaali frondi tuvastamine suvalise viigu pealt. Katkestused on sisse lülitatud vaid ühel koodri kanalil. Katkestuse saabumisel loetakse teise kanali taset, mille järgi otsustatakse sisendi tüüp. Nupuvajutusi tuvastatakse samuti EXTI katkestuste abil.

Ekraanile joonistamine

Ekraanile joonistamiseks oli vaja leida sobiv draiver. Draiveri ülesanne on muuta ekraanile joonistamist programmeerija jaoks mugavamaks. Draiveris on lihtsustatud vajalike kontrollsignaalide ekraanimooduli juhtregistritesse saatmine ja ekraanimälusse kirjutamine. Samuti on draiveris abifunktsioonid teksti ja geomeetriliste kujundite joonistamiseks.

Vabalt kättesaadavaid SSD1306 ekraanimooduli draivereid, mis toetavad STM32F4 seeria protsessoreid, on mitmeid [24, 25, 26]. Algselt kasutati ainult blokeeruvat I²C saatmist

toetavat draiverit, kuid testimisel selgus, et ekraanile joonistamise protsess oli liiga aeganõudev ning programmi peatsükkel ei töötnud mootorite sujuvaks juhtimiseks piisavalt kiiresti. Blokeeruv I²C saatmine võttis ekraani tekstiga täitmisel ligikaudu 40 ms, mis rikkus 50 Hz sagedusega töötükli nõuet. Seetõttu tuli leida lahendus ekraanile saatmise kiirendamiseks.

Esmalt prooviti programmi peatsükli kiirendamiseks jaotada ekraani uuendamine osadeks. Kuna ekraanimooduli sees olev mälu on jaotatud lehekülgedeks, siis vähim uuendus kirjutab üle vaid ühe lehekülje. Antud lahendus küll vähendas uuendamisele kuluvat aega piisavalt, kuid kui ekraanil kuvatav muutus, oli selgelt näha, et pilt uueneb horisontaalsete sektsioonide kaupa. Ekraanipildi värelemine oli häiriv ja segas kasutajaliidese mugavat kasutamist.

Põhiprotsessori koormuse vähendamiseks on mikroprotsessoris erinevaid lisaseadmeid. Ekraanile kuvamise kiirendamiseks võeti kasutusele DMA ehk otsemälupöördumiste kontrolleri kasutamist võimaldav draiver [27]. Otsemälupöördumiste kontrolleri abil on võimalik põhiprotsessori sekkumiseta saata I²C siinile ekraanimälu puhver. Antud lahenduse korral kulus põhiprotsessori aeg ainult lokaalse ekraanipuhvri kirjutamisele. Ekraanipuhvri täitmine tekstiga võttis ligikaudu 10 ms, mis oli piisav, et täita peatsükli sageduse nõue.

Ekraanile joonistamise lähenemisi on mitmeid. Üks võimalustest on hoida mikrokontrolleri programmimälu varasemalt genereeritud *bitmap* andmestruktuure ehk staatilisi pilte, mida oleks võimalik tervikuna ekraanile saata. See tähendaks, et iga ekraani oleku jaoks on oma pilt. Antud lahendus on aga raskesti laiendatav ja sobib rohkem animeeritud graafika ja ikoonide näitamiseks. Kuigi see lahendus kasutab vähem arvutusjõudlust, kasutab see ka palju rohkem programmimälu. Samuti ei sobi see dünaamiliselt muutuvate arvandmete näitamiseks. Seetõttu otsustati kasutajaliides kujutada kasutades ekraani draiveris olevaid teksti ja geomeetriliste kujundite joonistamise abifunktsioone.

Menüü

Menüü loogika loomiseks uuriti erinevaid veebist kättesaadavaid lahendusi. Leitud programmid ja teegid olid piisavalt rakenduse- ja riistvaraspetsiifilised ning raskesti laiendatavad [28, 29]. Kasutajaliidese loogika otsustati kirjutada ise, kuna see oli hinnanguliselt lihtsam kui mõne olemasoleva lahenduse ümberkirjutamine.

Kasutajaliidese töös kasutatakse olekumasinat, millel on neli olekut: staatuseaken, menüü, arvandme muutmine, infokraan. Programm on jaotatud loogilisteks osadeks: menüü valikute tagasikutse funktsioonid, joonistamise funktsioonid ja sisendi töötlemine. Uue oleku lisamiseks on menüü arendajal vaja luua vastavas olekus sisendi töötlemine ja ekraanile joonistamine.

Kuigi menüü struktuur peegeldab puu andmestruktuuri, otsustati programmi kompleksuse vähendamise eesmärgil paigutada menüü kahemõõtmelisse massiivi. Iga massiivi element defineerib mingi alammenüü kõigi saadavalolevate valikutega. Iga valik koosneb valiku märgendist, valiku tagasikutse funktsioonist, alammenüü puhul alammenüü nimetusest ja muudetava väärtuse puhul viidast muudetava väärtuse mäluaadressile. Arendajate töö lihtsustamiseks loodi menüüsse valikute lisamiseks valiku tüübile vastavad makrod.

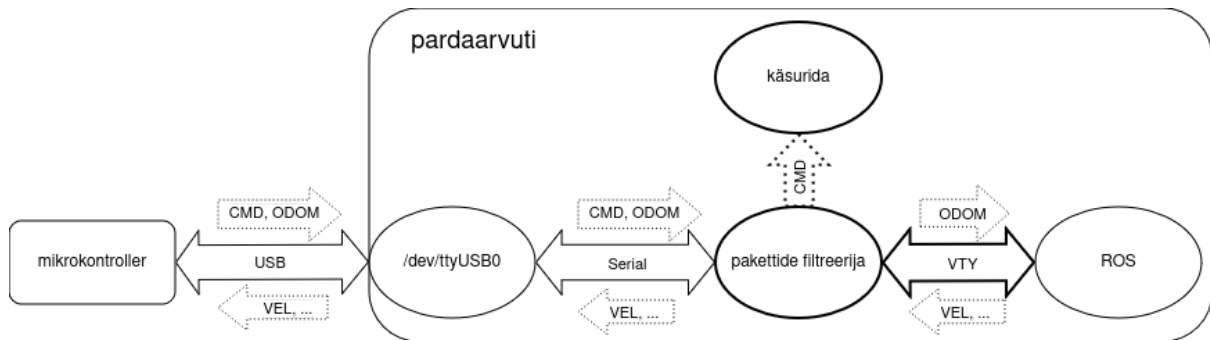
Programmi peatsüklis uuendatakse vastavalt olekumasina olekule ekraanipuhver ja töödeldakse kasutaja sisend. Kui DMA kontrolleri on ekraanipuhvri siinile saatmise operatsiooni valmis saanud, alustatakse seda uuesti. Loodud programmis on abifunktsioonid teksti paigutamise, raami, kerimisriba, keriva teksti joonistamise jaoks.

Töö käigus valmis ka abistav dokumentatsioon arendajatele, kes soovivad Robotondi kasutajaliidesele lisada uut funktsionaalsust. Dokumentatsioon kirjeldab staatuseaknale info lisamist, menüüsse valikute lisamist, uute alammenüüde ja olekute loomist. Nii loodud püsivara kui dokumentatsioon on leitav Robotondi püsivara Githubi repositooriumist (lisa 1).

Kasutajaliidese pardaarvutisse käskude saatmine

Üheks kasutajaliidese funktsionaalsetest nõuetest seati see, et selle vahendusel oleks võimalik saata pardaarvutisse käsurea käsk. Käskude saatmise eesmärk on juhtida ROSi tööd nii, et kasutajal oleks mugav pardaarvutisse kaugühendust loomata roboti tööprofiile ja funktsionaalsust muuta. Sellise võimaluse olemasolu oleks mugav Robotonti tutvustavate töötubade läbiviijatele. Pardaarvutis võiks kasutada kasutajaliidese käsu saabumisel konfiguratsiooni muutmiseks konteinereid, sest neid on võimalik käsurea käskudega luua, modifitseerida ja sulgeda. Konteineritehnoloogiat on juba Robotondiga seotud projektides kasutatud – Robotondi veebikursustes kasutatav keskkond loob igale õppijale sobiva ROS konfiguratsiooniga konteineri kursuse serveris. [30]

Kasutajaliidese abil pardaarvutisse käskude saatmise testimiseks kirjutati töö käigus lühike Pythoni programm. Programm võimaldab eristada ja töödelda kasutajaliidese käskude andmepakette ning saata roboti tööks vajalikule ROSi draiverile mikrokontrolleri edastatavat odomeetriainfot (joonis 10). Programm käivitatakse juurkasutaja õigustega, mis lubab kasutajaliidese abil ka pardaarvutit välja lülitada. Programm kasutab pakettide juhtimiseks ja filtreerimiseks virtuaalset terminali. Töö käigus ROSi funktsioonide juhtimist kasutajaliidese abil ei teostatud.



Joonis 10. Pardaarvutisse kasutajaliidest käskude saatmise andmevoog. Paksema joonega on tähistatud töö käigus testimiseks loodud komponendid

5.3 Ülevaade loodud kasutajaliidese

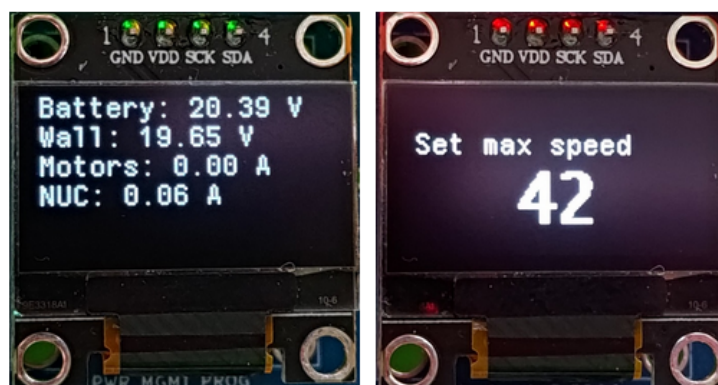
Roboti sisselülitamisel kuvatakse kasutajale roboti olekut kirjeldav aken, millele kuvatakse praeguses lahenduse etapis funktsionaalsetest väärtustest aku pinget ja roboti mootorite peatamiseks kasutatava ESTOP nupu olek (joonis 11). Tulevikus oleks kasulik kuvada seal ka näiteks pardaarvuti IP aadress kaugühenduse mugavamaks seadistamiseks, kuid selleks on vaja teostada suhtlusprotokoll pardaarvuti ja mikrokontrolleri vahel.

Nupuvajutusel kuvatakse ekraanile menüü, milles on võimalik koodrit keerates liikuda üles ja alla (joonis 11). Arvestades ekraani suhteliselt väikseid mõõtmeid ja ekraani draiveris saadaval olevate kirjatüüpide suuruseid, otsustati, et ekraanil kuvatakse loetavuse huvides korraga 3 menüü välja. Kasutaja valik on menüüs esile tõstetud raami abil ning intuitiivsemaks navigeerimiseks on ekraani paremas servas kerimisriba. Tagasi viivad menüü valikud on tähistatud üles suunatud noolega ja allmenüüd paremale suunatud noolega.



Joonis 11. Vasakul staatuseakna vaade, paremal peamenüü vaade

Menüü valikute hulgas on ka infoekraanid, millel saab kuvada harvemini vajaminevat infot, nt mootorite ja pardaarvuti voolutarvet (joonis 12). Samuti on olemas arvuliste väärtuste muutmise valikud, mida saab kasutada näiteks roboti maksimaalse kiiruse seadistamiseks.



Joonis 12. Vasakul näide infoaknast, paremal arvulise väärtuse muutmise vaade

6 Kokkuvõte

Käesoleva töö eesmärk oli robotplatvormi Robotont 3 kerel paikneva kasutajaliidese väljatöötamine. Töö nõuded püstitati nii, et edasise arendustöö käigus oleks võimalik kasutajaliidest kasutada pardaarvutis jooksva ROSi tarkvara juhtimiseks. Töö käigus uuriti teiste haridusrobotite ja muude elektroonikaseadmete kasutajaliideseid, et järgida laialdaselt levinud kasutajaliideste põhimõtteid.

Töö tulemusena valmis püstitatud nõuetele vastav Robotont 3 kasutajaliides. Kasutajaliidese disain sarnaneb uuritud kasutajaliidestest enim 3D-printerite tarkvarale Marlin. Loodud kasutajaliidese abil saab muuta roboti valguslahendust, muuta arvulisi parameetreid ja kuvada infot roboti töö kohta. Kasutajaliidest saab struktureerida alammenüüdega. Kasutajaliides on arvutusressusside suhtes tõhus, sest kasutab mikroprotsessori DMA lisaseadet. Kasutajaliidese püsivara on modulaarse struktuuriga ja dokumenteeritud ning lubab teistel Robotondi arendajatel edasiste tööde käigus välja arendada võimekus juhtida roboti pardaarvutit välise arvutita.

Viited

- [1] “robotont | avatud robotiplatvorm hariduses, teaduses ja tööstuses”, <http://robotont.ut.ee/> 04.11.2023, 19:14 (UTC)
- [2] P. Estefo, J. Simmonds, R. Robbes, J. Fabry, “The Robot Operating System: Package reuse and community dynamics”, *Journal of Systems and Software*, **151**, 2019, 226-242, <https://doi.org/10.1016/j.jss.2019.02.024>.
- [3] “GoPiGo Robot Kit for your Raspberry Pi”, <https://gopigo.io/>, 17.02.2024, 15:10 (UTC)
- [4] “Edison Programmable Robot”, <https://meet Edison.com/>, 17.02.2024, 15:20 (UTC)
- [5] Wikipedia, the free encyclopedia 2024. - Lego Mindstorms. https://en.wikipedia.org/wiki/Lego_Mindstorms, 17.02.2024, 15:30 (UTC)
- [6] “Jackal UGV”, <https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/>, 17.02.2024, 16:10 (UTC)
- [7] “TIAGo - Mobile Manipulator Robot”, <https://pal-robotics.com/robots/tiago/>, 17.02.2024, 16:15 (UTC)
- [8] R. Raudmäe, S. Schumann, V. Vunder, M. Oidekivi, M. K. Nigol, R. Valner, H. Masnavi, A. K. Singh, A. Aabloo, K. Kruusamäe, “ROBOTONT – Open-source and ROS-supported omnidirectional mobile robot for education and research”, *HardwareX*, **14**, 2023, DOI:10.1016/j.ohx.2023.e00436.
- [9] R. Raudmäe, “Avatud robotiplatvorm Robotont”, Tartu Ülikool, 2019
- [10] E. Mõtshärg, “3D-prinditava kere disain ja analüüs vabavaralisele haridusrobotile Robotont”, Tartu Ülikool, 2023
- [11] Veix123, IngvarD41, Norpz, markomuro, priitro, sudo-Cthulhufhtagn, robotont-electronics-mainboard. <https://github.com/robotont/robotont-electronics-mainboard> 04.11.2023, 19:38 (UTC)
- [12] Wikipedia, the free encyclopedia 2024. - Quadrature Diagram. https://en.m.wikipedia.org/wiki/File:Quadrature_Diagram.svg, 22.02.2024, 15:30 (UTC)
- [13] “SSD1306”, <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>, 22.02.2024, 15:30 (UTC)
- [14] Wikipedia, the free encyclopedia 2024. - OLED. <https://en.wikipedia.org/wiki/OLED#Advantages>, 22.02.2024, 15:30 (UTC)

- [15] Wikipedia, the free encyclopedia 2024. - Liquid-crystal display.
https://en.wikipedia.org/wiki/Liquid-crystal_display#Advantages_and_disadvantages
22.02.2024, 15:30 (UTC)
- [16] “Technical overview of the Edison V3 robot’s components”,
<https://meetiedison.com/technical-v3/>, 17.02.2024, 15:20 (UTC)
- [17] “Lego Mindstorms EV3 User Guide”,
https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MIN_DSTORMS_EV3_11_Tablet_ENUS.pdf, 22.03.2024, 17:30 (UTC)
- [18] “Handling - Duckiebot DB21”,
<https://docs.duckietown.com/daffy/opmanual-duckiebot/operations/handling/db21.html>,
22.03.2024, 17:30 (UTC)
- [19] “KettyBot”, <https://www.pudurobotics.com/product/detail/kettybot>, 02.04.2024, 17:30 (UTC)
- [20] “Original Prusa i3 MK3S+ 3D Printer kit”,
<https://www.prusa3d.com/product/original-prusa-i3-mk3s-3d-printer-kit/>, 24.03.2024, 17:30 (UTC)
- [21] “What is Marlin?”, <https://marlinfw.org/docs/basics/introduction.html>, 24.03.2024, 17:30 (UTC)
- [22] “Garmin Forerunner 255”, <https://www.garmin.com/pt-BR/p/780139/pn/010-02641-01>,
25.03.2024, 17:30 (UTC)
- [23] C. Gartenberg, “The Pebble’s hardware buttons couldn’t save smartwatches from touchscreens”, The Verge, 2021,
<https://www.theverge.com/22585819/pebble-smartwatch-hardware-button-touchscreen>
- [24] afiskon, stm32-ssd1306. <https://github.com/afiskon/stm32-ssd1306>, 29.03.2024, 17:30 (UTC)
- [25] 4ilo, ssd1306-stm32HAL. <https://github.com/4ilo/ssd1306-stm32HAL>, 29.03.2024, 17:30 (UTC)
- [26] olikraus, u8g2. <https://github.com/olikraus/u8g2>, 29.03.2024, 17:30 (UTC)
- [27] RobertoBenjami, stm32_ssd1306_i2c_dma_hal.
https://github.com/RobertoBenjami/stm32_ssd1306_i2c_dma_hal/tree/master,
29.03.2024, 17:30 (UTC)
- [28] 1999AZZAR, oled-display-menu. <https://github.com/1999AZZAR/oled-display-menu>,
30.03.2024, 17:30 (UTC)

- [29] upiir, arduino-oled-menu. https://github.com/upiir/arduino_oled_menu, 29.03.2024, 17:30 (UTC)
- [30] D. Krūmiņš et al., "Open Remote Web Lab for Learning Robotics and ROS With Physical and Simulated Robots in an Authentic Developer Environment," *IEEE Transactions on Learning Technologies*, **17**, 1325-1338, 2024, DOI: [10.1109/TLT.2024.3381858](https://doi.org/10.1109/TLT.2024.3381858)

Lisad

Lisa 1. Töös loodud tarkvara arhiveeritud versioon

<https://github.com/ut-ims-robotics/sakk-thesis-2024-robotont-firmware-menu>

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Andres Sakk

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

Avatud robotplatvormi Robotont 3 kasutajaliidese väljatöötamine

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 3.0, 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.

Andres Sakk

19.05.2024