

TARTU ÜLIKOOL
LOODUS- JA TEHNOLOOGIATEADUSKOND

Füüsika instituut

Antti Jaaniste

**ANDROID OPERATSIOONISÜSTEEMIL PÕHINEV
MÕÖTESÜSTEEM TREENINGUSEADMETELE**

Bakalaureusetöö (12 EAP)

Juhendaja: Fred Valk, MSc

Kaitsmisele lubatud:

Juhendaja:

allkiri, kuupäev

Tartu 2014

Sisukord

TÄHISED, LÜHENDID JA DEFINITSIOONID.....	4
SISSEJUHATUS.....	6
1. MANUSSÜSTEEMIDE ÜLEVAADE.....	7
1.1. Android operatsioonisüsteem.....	8
1.1.1. Ülevaade.....	8
1.1.2. Eelised teiste mobiilsete platvormide ees.....	8
2. MÕÖTESÜSTEEMI PROJEKTEERIMINE.....	10
2.1. Treeningseadme ülevaade ja tööpõhimõte.....	10
2.2. Nõuded loodavale mõõtesüsteemile.....	12
2.2.1. Nõuded riistvarale.....	12
2.2.2. Nõuded tarkvarale.....	13
2.3. ATmega328P mikrokontrolleri ülevaade.....	13
2.4. Honeywell SS460S magnetvälja anduri ülevaade.....	14
2.5. Atmel AT42QT1010 puuteanduri ülevaade.....	16
2.6. Tahvelarvuti A13 ülevaade.....	17
3. ANDMETE LUGEMINE, TÖÖTLEMINE NING ESITAMINE.....	18
3.1. Nurkkiiruse leidmine.....	18
3.2. Nurkkiiruse filtreerimine.....	18
3.3. Androidi rakendus.....	22
3.3.1. Alamprogramm SettingsActivity.....	23
3.3.2. Alamprogramm DeviceListActivity.....	24
3.3.3. Alamprogramm ActivityGraph.....	24
3.3.4. Teek BarGraph.....	25
4. TULEMUSED JA ANALÜÜS.....	26
KOKKUVÕTE.....	27
SUMMARY.....	28
VIIDATUD ALLIKAD.....	29
LISAD.....	31
Lisa 1. Andmehõivemooduli ühendusskeem.....	31
Lisa 2. Androidi rakendus. Seadete menüü.....	32
Lisa 3. Androidi rakendus. Tehtud tõmmete graafik.....	33
Lisa 4. Androidi rakendus. Tehtud tõmmete ajalugu.....	34
Lisa 5. Manussüsteemi erinevad olekud.....	35

Tähised, lühendid ja definitsioonid

ARM (ingl advanced RISC machine)	– ARM on odavate ning väikese energiakuluga 32-bitiste RISC protsessorite sari manussüsteemide, andmetöötuse, digitaalse signaalitöötuse ning laiatarbe-multimeedia jaoks.
FIR-filter (ingl finite impulse response)	– Lõpliku siirdega filter, ka FIR-filter.
FTDI	– Future Technology Devices International, mikrokiipide tootja.
GND (ingl ground)	– Maaühendus.
IIR-filter (ingl infinite impulse response)	– Lõpmatu siirdega filter, ka IIR-filter.
Jadavärat (ingl COM port)	– Liides, mida kasutatakse digitaalsignaali järjestikedaastuseks. Jadavärat on universaalne liides, mida võib kasutada peaaegu igasuguse seadme arvutiga ühendamiseks.
Kõrge ja madal olek	– Signaal on kõrge, kui see vastab loogilisele ühele (tõene) ja madal, kui see vastab loogilisele nullile (väär).
LED (ingl light-emitting diode)	– Valgusdiod.
Mikrokontroller (ingl. microcontroller unit)	– Ühe kiibi sisse ehitatud arvuti.
RISC (ingl reduced instruction set computer)	– Kärbitud käsustikuga arvuti.
SRAM (ingl static random-access memory)	– Staatile juhupeördusega mälu.
Tegevus (ingl activity)	– Ühe kindla kujundusega rakenduse osa, mis suhtleb kasutajaga.
TTL (ingl transistor–transistor logic)	– Transistor-transistor loogikaskeem – digitaalne mikroskeem, mis koosneb bipolaarsetest transistoritest ning takistitest.

Tx, Rx (ingl transmission, recepton)	– Jadaühenduse viigud, mida kasutatakse vastavalt andmete saatmiseks ning vastuvõtmiseks.
UART (ingl Universal Asynchronous Receiver/Transmitter)	– Universaalne asünkroontransiiver – Programmeeritud mikrokiip, mis juhib arvutit ja välisseadmeid ühendavat jadaliidest.
USB (ingl Universal Serial Bus)	– Universaalne järjestiksiin.
Vdd, Vss	Vdd tähistab toiteviiku, Vss tähistab maaühenduse viiku.
Sisendi või väljundi muutumine	– sisend või väljund läheb kõrgest olekust madalasse või vastupidi.
α , rad/s ²	– Nurkkiirendus.
B , T	– Magnetiline induktsioon.
E , J	– Energia.
$e = 1.6 \cdot 10^{-19}$ C	– Elektroni laeng.
I , kg·m ²	– Inertsimoment.
n , 1/m ³	– Vabade laengukandjate kontsentratsioon.
T , s	– Periood.
U , V	– Pinge.
ω , rad/s	– Nurkkiirus.

Sissejuhatus

Arvutusvõimsuse kiire kasv ja mitmete tarkvara- ja riistvaraplatvormide laialdane levik on loonud olukorra, kus tootjad otsivad aina enam oma toodetele lahendusi, mis lisaks oma tavaülesannetele suudaksid hakkama saada mitmete lisaülesannetega. Näiteks võib siin tuua tavalise mobiiltelefoni, mis lisaks helistamisfunktsioonile peab täitma ka muid ülesandeid, et see oleks toote lõpptarbijale atraktiivne.

Käesoleva bakalaureusetöö eesmärgiks on projekteerida mõttesüsteem treeningseadmele, mis koosneb tahvelarvutist ning andmehõivemoodulist. Lisaks riistvarale on vaja luua ka tarkvara andmehõivemoodulile ning tahvelarvutile. Loodava tarkvara ülesandeks on tagada andmehõivemooduli stabiilne ja tõrgeteta töö ning selle külge ühendatud anduritest lugemite võtmine, edasine töötlus ja salvestamine. Oluliseks lisanõudeks on juhtprogrammi kõrge töökindluse tagamine.

Süsteem mõõdab hooratta pöörlemise kiirust, kasutades Halli efektil põhinevat magnetvälja andurit, arvutab selle põhjal hooratta energia ning kuvab tulemused ekraanil. Andmehõivemooduli tarkvara arendatakse C-keeles ning tahvelarvuti rakendus Javas, mis on mõeldud töötama operatsioonisüsteemil Android.

Andmehõivemoodul on ühendatud tahvelarvutiga üle virtuaalse COM-pordi. Androidi rakendust uuendatakse automaatselt üle WiFi võrgu.

Töö esimeses osas antakse ülevaade manussüsteemides kasutatavast riist- ja tarkvarast ning operatsioonisüsteemist Android. Seejärel kirjeldatakse süsteemile esitatud nõudeid. Edasi antakse ülevaade juba konkreetsest töös kasutatavast riistvarast, kus kirjeldatakse, mida vastavad komponendid teevad ning miks nad valiti. Lisaks tutvustatakse andmehõivemooduli tarkvara ning Androidi rakendust. Tarkvara juures kirjeldatakse nende ülesehitust ning tööloogikat.

Viimases peatükis räägitakse arenduse käigus tekkinud probleemidest ning nende lahendamisest – muuhulgas, kuidas tagada puuteanduri tõrgeteta töö või magnetvälja anduri signaali õigsus. Lisaks kirjeldatakse, kuidas on lahendatud tahvelarvuti rakenduse automaatne uuendamine serverist.

1. Manussüsteemide ülevaade

Manussüsteemid on väga laialdaselt levinud: kui igal aastal toodetakse miljoneid protsessoreid, mis on mõeldud lauarvutite või sülearvutite jaoks, siis toodetakse miljardeid protsessoreid manussüsteemide jaoks. Manussüsteemide alla kuuluvad peaaegu kõik seadmed, mis sisaldavad endas protsessorit ning ei ole lauarvutid, sülearvutid või serverid, näiteks tavakasutaja elektroonika: mobiiltelefonid, fotoaparaadid, videokaamerad ning kalkulaatorid. Koduseadmetest: mikrolaineahjud, külmkapid ning pesumasinad. Kontoriseadmetest: printerid, koopiamasinad ning skännerid. Autodest: pidurisüsteem, kliimaseade, kütuse sissepritse ning ABS. [11]

Manussüsteemide omadused:

- **Ühe funktsiooniga.** Tavaliselt töötab manussüsteemis pidevalt üks kindel programm. Näiteks mikrolaine ahi on alati mikrolaineahi, samas lauarvutiga saab teha tekstitöötlust, mängida, kuulata muusikat või vaadata filme. Lisaks saab programme uuendada ning lisada.
- **Ranged nõuded.** Manussüsteemidel on ranged nõuded funktsionaalsusele, hinnale, suurusele, jõudlusele ning võimsusele. Sellised süsteemid peavad olema võimalikult odavad, makstes mõned eurod, mahtuma ühele mikrokiibile, olema võimalikult kiired, et reaajas andmeid töödelda ning olema piisavalt väikese energiatarbega, et suurendada aku kestvust või vähendada vajadust aktiivjahutuse järele.
- **Kiiresti reageeriv ning reaalgajaline.** Paljud manussüsteemid peavad pidevalt registreerima keskkonnas toimuvaid muutusi ning töötleva andmeid reaalgajaliselt. Näiteks auto püsikiiruse hoidja peab pidevalt lugema kiirust ning vastavalt reguleerima pealeantava kütuse hulka. Kui see ei toimuks reaalgajaliselt, ei liiguks auto püsiva kiirusega ning võib ka mõningatel juhtudel kaotada juhitavuse. Samas kui lauarvutites ei tööta näiteks klaviatuuri sisendi lugemine reaalgajaliselt, on see kasutajale ebamugav, aga ei põhjusta õnnetusi. [11]

1.1. Android operatsioonisüsteem

1.1.1. Ülevaade

Androidi asutajaks on firma Android Inc., mis loodi 2003. aasta oktoobris, eesmärgiga toota targemad seadmed, mis on teadlikud kasutaja eelistustest ning asukohast. Android Inc. töötas kaks aastat saladuskatte all ning teada oli ainult see, et tegeletakse mobiilse operatsioonisüsteemi loomisega. 2005. aasta juulis ostis firma Google plaaniga siseneda mobiiltelefonide turule. [2]

Androidi operatsioonisüsteem on loodud modifitseeritud Linuxi kernelile. Rakendused on kirjutatud Javas ning jooksevad Dalvici virtuaalmasinas, operatsioonisüsteemi enda osad on kirjutatud nii Javas, C-s, C++-is kui ka XML-is. Androidi seadmega tuleb kaasa suur hulk rakendusi, näiteks veebibrauser, e-posti klient ja kaardirakendus, mida tarbija saab kohe kasutama hakata. [3]

1.1.2. Eelised teiste mobiilsete platvormide ees

Praegu on populaarsed mitmed mobiilsed platvormid, näiteks iOS, Windows Mobile ja BlackBerry, ning tekib küsimus, miks peaks rakenduse just Androidile looma. Kuigi Android sisaldab vahendeid, mis leiduvad ka teistes platvormides, on see esimene nii mõneski aspektis. [4]

Android on mobiilsete seadmete tootjate seas populaarne, kuna platvormi saab oma vajaduste järgi muuta, maksmata tasusid, mis on vajalik suletud platvormi korral. Lisaks kindlustab platvormi avatus selle, et Android ei ole seotud ühe kindla tootjaga, mis võib näiteks pankrotti minna ning tarbijad ei saa enam seda platvorm kasutada. [4]

Mobiilse operatsioonisüsteemi populaarsus sõltub ka sellest, kui lihtne on vastaval platvormil uusi rakendusi turustada. Rakenduste jagamine Google Play (endise Android Marketi) kaudu on vaba, see tähendab, et arendajad saavad programmid üles laadida ning kasutajad saavad neid kasutada. Apple App Store'i puhul otsustab Apple, millised rakendused nende rakendustepoes avalikustatakse. Kuna Google Play'sse üleslaadimine on vaba, on see endaga kaasa toonud palju madala kvaliteediga rakendusi. Google teeb praegu parema kvaliteediga rakendusi lihtsamalt leitavaks. [3]

Kõik Androidi rakendused on modulaarsed ning asendatavad: programmide osi saab kasutada teistes rakendustes ning viisidel, milleks need osad alguses mõeldud ei olnud. Lisaks on võimalus kasutada paljusid sisseehitatud teenuseid: [4]

- Asukoha määramine – kasutades GPS-i või mobiilimastide abil trianguleerimist, on võimalik luua rakendusi, mis olenevalt kasutaja asukohast käituvad erinevalt. [4]
- SQL andmebaas – võimaldab luua lihtsalt kohaliku andmebaasi ning sünkroniseerida seda teistega. [4]
- Rakenduste integreerimine – on võimalik oma rakenduses sisseehitatult kasutada näiteks veebibrauserit või kaardirakendust. See tagab suurema funktsionaalsuse ning võimaldab alandada arenduskulusid. [4]

Android haldab automaatselt rakenduste elutsüklit. Lõppkasutaja ei pea enam teadma, millised rakendused hetkel jooksevad ning millised tuleb sulgeda, et oleks võimalik uusi programme käima panna. Android hoolitseb ka selle eest, et jõude olevad programmid suletaks, kui tekib vajadus rohkema mälu järele. Platvorm on optimeeritud, töötamaks madala jõudluse ja vähese mäluiga seadmetel – see on midagi, millele teised tootjad varem tähelepanu pööranud ei ole. [4]

Programmid jooksevad üksteistest eraldatult ning saavad ligipääsu teiste rakenduste andmetele ainult juhul, kui kasutaja seda lubab. Samas on iPhone'i rakendustel vaikimisi ligipääs paljudele süsteemiressurssidele ning kasutaja ei saa seda keelata. Kuna Androidi kasutajad saavad määrata, milliseid teenuseid rakendus kasutada tohib, saavad nad paremini kontrollida oma privaatsust ning turvalisust. [3]

Androidil on sisseehitatud tugi kvaliteetsele graafikale ja helile. Olemas on nii kahemõõtmelise vektorgraafika tugi, kasutades Adobe Flashi, kui kolmemõõtmelise graafika tugi, kasutades OpenGLi. Toetatakse laialtkasutatavad heli- ja videokodekid, näiteks H.264, MP3 ja AAC. [4]

Toetatud on suur hulk riistvara. Kuna programmid on kirjutatud Javas ning töötavad Dalviku virtuaalmasinas, saab programme käivitada erinevatel arhitektuuridel, näiteks ARM või x86. Toetatakse ka erinevad sisendusviise, näiteks klaviatuuri, puudutust ning juhtkuuli. Kasutajaliidest on võimalik kohandada, töötamaks erinevatel ekraanilahutustel. [4]

2. Mõõtesüsteemi projekteerimine

2.1. Treeningseadme ülevaade ja tööpõhimõte

Käesoleva bakalaureusetöö eesmärgiks oli prototüübi projekteerimine ettevõttele, mille peamiseks tegevusalaks on treeningseadmete väiketootmine. Loodava süsteemi ülesandeks on, sarnaselt teistele sellistele süsteemidele, kuvada ekraanil kasutaja treeningaktiivsus. Eriliseks teeb antud lahenduse asjaolu, et erinevalt tavalistest treeningseadmetest jõusaalis, kus nn vastujõu tekitavad erinevad raskused, kasutatakse siin vastujõu tekitamiseks hooratta pöörlemisenergiat. Sellise lahenduse üheks eeliseks on asjaolu, et sellised seadmed töötavad ka keskkonnas, kus kehade kaal võrdub nulliga.

Treeningseadme tööpõhimõte on järgmine. Hoorattaga sama telje külge on kinnitatud vastupidav rihm, mis treenimise käigus järjest ühte- ja teistpidi tagasi telje ümber keritakse. Alguses on rihm telje ümber keritud ning rihma tõmmates pannakse hooratas liikuma. Kui rihm on kogu pikkuses teljelt maha keritud, jätkab hooratas inertsit tõttu liikumist ning kerib rihma teistpidi telje peale. Treenija peab energiat kulutama nii hooratta liikuma panemiseks kui ka selle pidurdamiseks.

Raskuste muutmise asemel saab treenija vajadusel muuta hooratate arvu ja seega kogu inertsimomenti. Suurema inertsimomendi korral tuleb treenijal rakendada suuremat jõudu, et sooritada harjutusi sama kiirusega. Hooratta energia on määratud järgmiselt:

$$E = \frac{I \cdot \omega^2}{2} \quad (1)$$

Siin on I kasutatud hooratate inertsimoment. Kogu inertsimoment on kõikide hooratate inertsimomentide summa:

$$I = I_1 + I_2 + I_3 + \dots \quad (2)$$

Süsteem ei mõõda automaatselt telje inertsimomenti. Kuna inertsimomendi sisestamine on tülikas, siis esitatakse kasutajaliidesel hooratate valik koos vastavate piltidega, mis teeb hooratate valimise kasutaja jaoks mugavamaks. Kui hoorattad on õigesti valitud, siis arvutab süsteem automaatselt vastava koguinertsimomendi vastavalt valemile 2.

Hooratta energia leidmiseks on vaja teada ka hooratta pöörlemise kiirust (nt valem 1). Selle mõõtmine käib juba automaatselt. Inertsimomendi automaatne mõõtmine pole vajalik, kuna

hooratase vahetamisi ei tule sageli ette.

Hooratta nurkkiirust ω arvutatakse Halli efektil põhineva magnetvälja anduri tekitatud katkestustele vastavate ajaintervallide Δt järgi. Vastavalt nurkkiiruse definitsioonile võime kirjutada:

$$\omega = \frac{\Delta \varphi}{\Delta t} = \frac{2\pi \text{ rad}}{16 \cdot \Delta t} = \frac{2\pi \cdot 1000}{16 \cdot \Delta t(\text{ms})} = \frac{392.699}{\Delta t(\text{ms})} \quad (3)$$

Hoorattaga sama telje küljes on teine ketas, mille sees on kuusteist magnetit, mis asuvad üksteisest võrdsel kaugusel. See tähendab, et 2π radiaani kohta on kuusteist magnetit ning kui magnetanduri kahe katkestuse vahe on Δt (millisekundites), siis saab leida konstandi, mis tuleb läbi jagada Δt -ga, et leida hooratta nurkkiirus ω . Näiteks, kui kahe katkestuse vaheline aeg on $\Delta t = 1s = 1000 \text{ ms}$, siis nurkkiirus on $\omega = 0.39 \text{ rad/s}$. Teiste sõnadega, kui hooratta kiirus on $2\pi \text{ rad/s}$, siis peab magnetvälja andur registreerima 16 oleku muutust. Lisaks nurkkiirusele saame leida ka nurkkiirenduse α :

$$\alpha = \frac{\Delta \omega}{\Delta t} \quad (4)$$

Andmehõivemooduli tööpõhimõte on järgmine. Jõud, mida treenija peab rakendama hooratta liikuma panemiseks sõltub viimase inertsimomendist I . Teades hooratta nurkkiirust ω , on võimalik arvutada teisi parameetreid vastavalt valemitele 1 - 4. Kõige olulisem nendest on hooratta energia E , aga saab leida ka rakendatud jõu F ning kulutatud kalorid. Nende arvutamiseks on kõigepealt vaja leida hooratta nurkkiirus ω . Kuna viimane on võrdne telje kiirusega, saab teljele kinnitada lisaratta ning mõõta selle kiirust. Lisaratta kiiruse mõõtmiseks on sellesse puuritud kuusteist auku, millesse on liimitud magnetid. Kui magnet liigub magnetvälja andurist mööda, registreeritakse vastav magnetvälja muutus. Näiteks kui hooratta nurkkiirus on $2\pi \text{ rad/s}$, siis väljastab magnetandur kuusteist elektriimpulssi sekundis. Selline magnetite arv sai valitud, kuna siis on impulsside arv piisav ning mitte liiga suur. Kuna andmete uuendamine ja kuvamine toimub kümme korda sekundis, on vaja impulssi umbes iga 100 ms tagant. Hooratta liikumiskiirus on vahemikus 1 - 30 rad/s, mis tähendab, et impulsside vahe on 10 ms - 300 ms. Suurema hulga magnetite kasutamine ei ole vajalik, kuna see tekitaks lihtsalt üleliigseid andmeid.

Muutuvat magnetvälja mõõdab Halli efektil põhinev magnetvälja andur. Anduri tekitatud impulse loeb mikrokontroller ning arvutab nurkkiiruse ω ning nurkkiirenduse α vastavalt valemitele 3 ja 4. Saadud andmed saadetakse FTDI FT232RL mikrokiipi, mis saadab need omakorda üle USB-ühenduse tahvelarvutisse. Treeningut alustades peab treenija valima Androidi rakendusest, milliseid hoorattad on treeningseadme külge kinnitatud. Rakendus arvutab automaatselt inertsimomendi,

kasutades valemit 3 ning arvutab hooratta energia E , kasutades valemit 1.

Nurkkiiruse mõõtmiseks on lisaks toodule ka teisi võimalusi. Võib kasutada teljega ühendatud generaatorit ja mõõta sellelt pinget või kasutada optilist andurit, kus kiir katkestatakse ühe täispöörde jooksul teatud arv kordi. Generaatori puuduseks on see, et seda ei saa lihtsalt külge monteerida juba toodetud treeningseadmetele. Optilise anduri puuduseks on see, et diodi ning fotosensori vahele võib sattuda võõrkehi (st prügi, ämblikud jms), mis katkestavad kiire ning põhjustavad kiiruse arvutamises vigu. Diodi ning fotosensori vahelist ala tuleks alati kontrollida ning vajadusel puhastada. Magnetvälja andurit kasutades puuduvad liikuvad osad, mis üksteisega kokku puutuks ning selle tööd ei häiri ka väikesed võõrkehad, mis võivad sattuda anduri ja magnetite vahele, sest magnetväli läbib neid ilma probleemideta. [14]

Kuna alati ei pruugi olla treeningseadme lähedal pistikupesa, et süsteem saaks töötada võrgutoitelt, peab saama treenida ka akutoitelt. Aku säästmiseks on läheb süsteem magamisolekusse, kui keegi ei ole teatud aja jooksul seda kasutanud. Magamisolekust ärkab süsteem nii hooratta liikumise kui ka treeningseadme enda puudutamise peale, et treenija ei peaks süsteemi eraldi sisse lülitama, kui soovib seda kasutama hakata. Puututuse tuvastamiseks on kasutuses Atmeli QTouch mikrokiip, mis võimaldab muuta kogu treeningseadme kere mahtuvuslikuks puuteanduriks. Olenevalt seadistatud tundlikusest, võimaldab mikrokiip tuvastada raami puudust ning ka seda kui treenija seadmele lähedale läheb. [6]

2.2. Nõuded loodavale mõõtesüsteemile

Järgnevalt on lühidalt toodud nõuded loodavale mõõtesüsteemile.

2.2.1. Nõuded riistvarale

- Magnetvälja muutumist tuleb tuvastada kuni kahe sentimeetri kauguselt.
- Süsteem peab tuvastama metallist osade puudutust.
- Andmehõivemooduli tööolekut peab saama määrata ekraanist sõltumatult.
- Süsteem peab akutoitelt töötama kuni kaks tundi.
- Süsteem peab olema lihtsasti paigaldatav.
- Süsteem peab olema lihtne ning odav toota.

2.2.2. Nõuded tarkvarale

- Kasutaja peab saama valida treeninguks kasutatavaid hoorattaid. Süsteem arvutab ja salvestab koguinertsimomendi vastavalt valemile 2. Kuna inertsimomendi sisestamine on tülikas ja ketastele pole inertsimomendi peale kirjutatud, siis esitatakse kasutajale hooratuste valik koos piltidega, et muuta valimine lihtsamaks.
- Graafikul kuvatakse hooratta energia tulpdiagrammina (kulutatud energia on sellest suurem, kuna inimene on suhteliselt madala kasuteguriga soojusmasin). Tulpdiagrammilt on mugav visuaalselt hinnata, kuidas erinevad tõmbed üksteisest ning kas iga tõmbe jaoks kulutatakse sama palju energiat.
- Saab valida mitut tulpa treening-graafikul näidata. Kasutaja saab valida, kas soovib näha vähem tulpasid, et seda oleks mugavam kaugemalt lugeda või vastupidi.
- On võimalik valida, kas graafikul kuvatakse teatud arv viimaste tõmmete tulpi koos keskmisega või kogu treeningu info. Lisaks viimastele tehtud tõmmetele saab kasutada vaadata, kuidas energiakulu muutus kogu treeningu vältel.
- Tahvelarvuti peab suutma andmehõivemoodulist tulevat infot töödelda ning graafikut uuendada iga 0.1 sekundi järelt st sagedusega 10 Hz. See on piisav aeg, et graafiku uuemine tunduks inimesilmale sujuv.
- Süsteem peab magamisolekust väljuma puudutuse või hooratta liikumise peale, et kasutaja ei peaks süsteemi eraldi sisse lülitama, kui ta soovib treeningut alustada.
- Kui teatud aja jooksul, mida peab saama programmist muuta, pole puudutust või magnetvälja muutumist tuvastatud, läheb süsteem magamisolekusse, et säästa akut ja ekraani.
- Magamisolekus on aku säästmiseks tahvelarvuti ekraan välja lülitatud või heledus viidud miinimumi.
- Kui aku saab liiga tühjaks lülitub süsteem kohe välja.
- Tarkvara peab saama võrguühenduse olemasolul mugavalt uuendada, olemata sellest kus süsteem asub.

2.3. ATMega328P mikrokontrolleri ülevaade

ATMEGA328P põhiandmed. [15]

Tööpinge:	5 V
Voolutarve:	30 mA
Digitaalseid sisend/väljund	14
Analoogsisendeid:	8
Flash mälu:	32 kB (millest 2 kB kasutab alglaadur)
SRAM:	2 kB
EEPROM	1 kB
Taktsagedus	16 MHz

Antud mikrokontrolleril on madal energiatarve ning piisav hulk sisendeid ja väljundeid, et ühendada kõik lisaseadmed. Kõige olulisem on, et saab kasutada kahte välist katkestust, et ühendada puuteandur ning magnetvälja andur. Mikrokontrolleril on ka erineva energiatarbimisega magamisolekud, mis võimaldavad akut säästa, kui treeningut ei toimu. [15]

Suhtlus tahvelarvutiga toimub üle UART TTL jadaühenduse, mis kasutab digitaalseid viike 0 (Rx) ja 1 (Tx). Need on omakorda ühendatud FTDI FT232RL kontrolleri, mis võimaldab kasutada jadaühendust üle USB-liidese, kasutades FTDI draivereid. Viimane tekitab virtuaalse COM-pordi, millest saab vastavat tarkvara kasutades andmeid lugeda. Mikrokontrolleri põhiülesanne on tõlgendada andmeid, mis saabuvad magnetandurist ning puutesensorist. Lisaks tegeleb see andur-LED-ide juhtimisega, mis näitavad süsteemi tööolekut. Mikrokontroller arvutab ratta nurkkiirust ning nurkkiirendust ning saadab need üle FTDI kontrolleri arvutisse. Mikrokontrolleri ühendus anduritega on toodud lisa 1.

2.4. Honeywell SS460S magnetvälja anduri ülevaade

SS460S on väikeste mõõtmetega ja tundlik andur, mis kasutab Halli efekti magnetvälja tajumiseks. Rakendades andurile erineva suunaga magnetvälja, on väljund vastavalt kas kõrge või madal. [5]

Mõned Honeywell SS460S võimalikud kasutusalaad: [5]

- pöörlemiskiiruse mõõtmine,
- mootorite juhtimine,
- ventilaatorite juhtimine,

- tahhomeeter,
- asendiandur.

Halli efekti avastas dr Edwin Hall juba 1879. aastal, kui see on kasutust leidnud alles viimase kolmekümne aasta jooksul. Põhjuseks oli see, et Halli efekti tekitatud pinge nn Halli pinge on väga väike ning 19. sajandi lõpus ei olnud võimalik seda lihtsasti ja odavalt võimendada. Esimene praktiline kasutusala sellele andurile oli 1950-ndatel mikrolaineahju võimsuse sensor. [8]

Halli efektiks nimetatakse seda, et kui elektrijuht, mida läbib vool, on asetatud magnetvälja, siis tekib elektrijuhis pinge, mis on risti nii voolu kui ka magnetvälja suunaga. SS460S-is on elektrijuhiks kaks räniplaati, mis on paigutatud nii, et nad asuksid üksteisele võimalikult lähedal. Ühele plaatidest rakendatakse ergutusvool horisontaalsuunas ning teisele vertikaalsuunas. Nende väljundid ühendatakse kokku, et muuta võimalikult väikeseks räni piesotakistusliku efekti (plaadi takistuse sõltumise sellele mõjuvast jõust) mõju. Magnetvälja olemasolul tekib pinge, mida võimendatakse diferentsiaalvõimendiga. [8]

Võimendamine on vajalik, kuna tekkiv pinge on väike. Näiteks kui $B = 3 \text{ mT}$ (soovituslik magnetvälja tugevus magnetvälja anduri ümberlülitamiseks), $d = 1 \text{ mm}$ (keskmine räniplaadi paksus), $i = 0.03 \text{ mA}$ (vool räniplaadis) ning $n = 5 \cdot 10^{15} / \text{m}^3$ (vabade laengukandjate arv ränis) siis Halli pinge on: [9]

$$U(H) = \frac{-i \cdot B}{d \cdot n \cdot e} = \frac{-0.03 \text{ mA} \cdot 3 \text{ mT}}{1 \text{ mm} \cdot 5 \cdot 10^{15} / \text{m}^3 \cdot 1.6 \cdot 10^{-19} \text{ C}} = -0.11 \text{ V} \quad (5)$$

Võimendi väljundisse on ühendatud Schmitti trigger, mis võrdleb tekkinud pinget eelseatud pingega V_{ref} . Kui pinge on V_{ref} -ist suurem, lülitub trigger sisse, ning pinge langemisel alla V_{ref} -i lülitub trigger välja. Schmitti triggeri kasutamine põhjustab hüstereesi: sisse- ja väljalülitamine ei toimu mitte sama pinge juures, vaid on vajalik, et sisselülitamiseks oleks pinge teatud võrra suurem V_{ref} -ist ning väljalülitamisel väiksem. [8]

SS460S-il on suur lubatud toitepinge vahemik (3 – 24 V), väike voolutarve (keskmine 3.5 mA, maksimaalselt 8 mA) ning lülitamine toimub juba 3 mT tugevuse magnetväljaga, mis võimaldab kasutada suuremat kaugust anduri ja magneti vahel või väiksemaid (nõrgema magnetväljaga) magneteid. Võrreldes teiste sarnaste anduritega, on SS460S-il väiksem voolutarve ning suurem tundlikus sama hinna juures. [5]

Veel eeliseid. [5]

- Kiire ümberlülituse aeg (maksimaalselt 1.5 μ s), võrreldes teiste magnetvälja anduritega.
- Ümberlülitus toimub magnetvälja polaarsuse vahetumisel – võimaldab täpset kiiruse mõõtmist.
- Sisseehitatud valepolaarsuse kaitse – kaitseb andurit valetpidi pingestamise eest

Magnetandur on otse ühendatud mikrokontrolleri üldotstarbeliste sisend-väljund viikudega. Selline ühendusviis on võimalik, kuna andur tarbib maksimaalselt 6 mA ning väljundist võib tarbida kuni 40 mA. Lisaks võimaldab see andurit vajadusel tarkvaraliselt välja lülitada, et vähendada voolutarvet.

Magnetvälja andur on ühendatud mikrokontrolleri jalaga, mille külge on võimalik ühendada väliste katkestuste kuulamine. See võimaldab hoida süsteemi magamisolekus ning tulla sealt välja alles siis kui hooratas liikuma hakkab, vähendades jõudeolekus energiatarvet. Puuduseks on see, et magnetvälja anduris peab kogu aeg vool sees olema, sest vastasel juhul ei saaks andur mikrokontrollerile hooratta liikumise peale signaali anda. Magnetvälja anduri ühendus mikrokontrolleriga on toodud lisas 1.

Treeningseadme hoorattaga sama võlli külge (st hooratta ning sensor-ketta nurkkiirused on võrdse ja see võimaldab mõõta hooratta nurkkiirust võlli nurkkiiruse kaudu) on paigaldatud ketas, millesse on paigutatud magnetid vahelduva polaarsustega. Iga kord kui andurist möödub magnet, muutub anduri väljund: põhjapoolus seab anduri väljundi kõrgeks, lõunapoolus madalaks [5]. Mikrokontroller registreerib väljundi muutuse, mõõtes aega, mis kulub kõrgest olekust madalasse minekuks ja vastupidi. Näiteks kui väljund muutub iga sekundi tagant, on ratta nurkkiirus (kasutades valemit 4) $\omega = 2\pi/16$ rad/s.

2.5. Atmel AT42QT1010 puuteanduri ülevaade

AT42GT1010 on andur, mis suudab tuvastada nii objektide lähedust kui ka puudutust. See on ideaalne lihtsa puuteanduri loomiseks.. Andur suudab tekitada lähedust detekteeriva välja, mis töötab ka läbi mitme sentimeetri paksuste dielektrikute, näiteks klaasi, plastiku, kivi või puidu. [6]

AT42GT1010 võimaldab muuta erinevad metallist objektid anduriteks, mis reageerivad nii suure mahtuvusega objekti nt inimkeha lähedusele kui ka puudutusele. Lisaks suudab antud andur ennast kalibreerida vastavalt objekti mahtuvusele, mille külge ta ühendatud on. Kuna puuteandur on loodud asendama mehaanilisi lüliteid või nuppe, sisaldab AT42GT1010 nii riistvara kui ka signaalitöötluse funktsioone, mis tagavad anduri stabiilse töö erinevates tingimustes.[6]

AT42QT1010 puuteandur on ühendatud sama moodi üldotstarbeliste sisend-väljund viikudega nagu magnetvälja andur, kuna voolutarve on madal (alla 1 mA). See võimaldab ka andurit tarkvaraliselt välja lülitada. Anduri väljund on ühendatud mikrokontrolleri jalaga, mille külge saab ühendada väliste katkestuste kuulamise. See võimaldab süsteemi magamisolekust puudutuse peale äratada. Soovituslik väärtus kondensaatori Cs jaoks on 47 nF [6], lisaks on ühendatud kondensaator toite ja maa viikude vahele (100 nF), et häireid vähendada. Anduri täpne ühendus mikrokontrolleriga on toodud lisas 1. On oluline, et metallpind, mida kasutatakse puuteandurina ei oleks maandatud, kuna see ei võimaldaks puuteanduril töötada.

Anduri ülesandeks on registreerida puudutust ning äratada mikrokontroller magamisolekust. See toimub siis, kui ekraan on läinud madala energiatarbimisega olekusse. Inimene, kes soovib seadet kasutada, puudutab seda ning tahvelarvuti ja andmehõivemoodul lülitavad ennast automaatselt sisse.

2.6. Tahvelarvuti A13 ülevaade

A13 on piisava hinna ja võimsuse suhtega, et jooksutada loodud rakendust ning töödelda ja kuvada saabunud andmeid piisava kiirusega. Lisaks on olemas WiFi ning USB-lisaseadmete tugi, mida saab kasutada tarkvara uuendamiseks ning andmehõivemooduli ühendamiseks.

Omadused: [10]

Ekraani suurus:	7"
Ekraani tehnoloogia:	TFT LCD mahtuvustundlik puuteekraan
Ekraani resolutsioon	800 x 480 pikslit
Aku mahtuvus:	2000 mAh
Muutmälu (RAM):	512 MB
Protsessor:	Cortex A8, 1.2 GHz

Tahvelarvutiga on ühendatud kaks kaablit. Esimene on USB-ühendus andmehõivemooduliga, mille kaudu saab viimane toidet ning saadetakse ratta pöörlemise andmed tahvelarvutisse. Teine on toite- ja laadimiskaabel tahvelarvuti jaoks. Viimase puudumisel suudab süsteem töötada akutoitelt kuni kaks tundi.

Tahvelarvutil töötab kasutajatarkvara, mis teisendab andmehõivemoodulist saadud andmed kasutajale arusaadavaks. Treeningseadmel tehtud tõmbeid kuvatakse tulpdiaagrammil, mis näitab kulutatud energiat. Lisaks näidatakse keskmist ning ajalugu.

3. Andmete lugemine, töötlemine ning esitamine

3.1. Nurkkiiruse leidmine

Andmehõivemooduli programmi põhiprotsess on lõputus tsüklis, kus toimub ratta pöörlemiskiiruse saatmine, aja arvestamine ning magamisolekusse minemine, kui ühe minuti jooksul ei ole puutevõi magnetvälja andurist signaali tulnud. Programmi tööd juhitakse kahe välise katkestusega: esimene käivitab katkestusrutiini siis, kui magnetsensori väljund muutub, teine siis kui puuteanduri väljund muutub.

Nurkkiiruse arvutamiseks kutsutakse iga kord, kui magnetanduri signaal muutub, välja järgnev funktsioon:

```
void hall() {  
    if(SLEEP) {wakeUp(0); }  
    calcData(true);  
    before = millis();  
    int2=0;  
    if (int1>0) { SEND=true;  
}
```

Kood 1: Nurkkiiruse arvutamine

Kõigepealt kontrollitakse, kas mikrokontroller on magamisolekus ning vajadusel tuuakse ta sealt välja. Siis arvutatakse nurkkiirus, milleks kasutatakse muutujat *before*. Seal hoitakse viimase magnetanduri tekitatud katkestuse ajahetke. Lahutades hetkeajast *before*-i, saab kätte katkestusetevahelise aja Δt ning kasutades valemit 3 saab arvutada nurkkiiruse. *int1* loendab magnetanduri katkestusete arvu, *int2* puuteanduri katkestuste arvu. Kui *int1* on suurem ühest, alustatakse andmete saatmist tahvelarvutisse.

3.2. Nurkkiiruse filtreerimine

Ratta nurkkiiruse arvutamiseks on vaja magnetvälja andurist tulevat signaali filtreerida, kuna hüstereesi tõttu ei ole ratta ühtlase pöörlemiskiiruse juures ajavahe katkestuste vahel konstantne. Näiteks kui ratas pöörleb kiirusega 1 rad/s, peaksid katkestused tekkima iga $2\pi / 16 = 0.39$ sekundi tagant. Hüstereseis tõttu on ajavahe vaheldumisi 0.25 ja 0.51 sekundit.

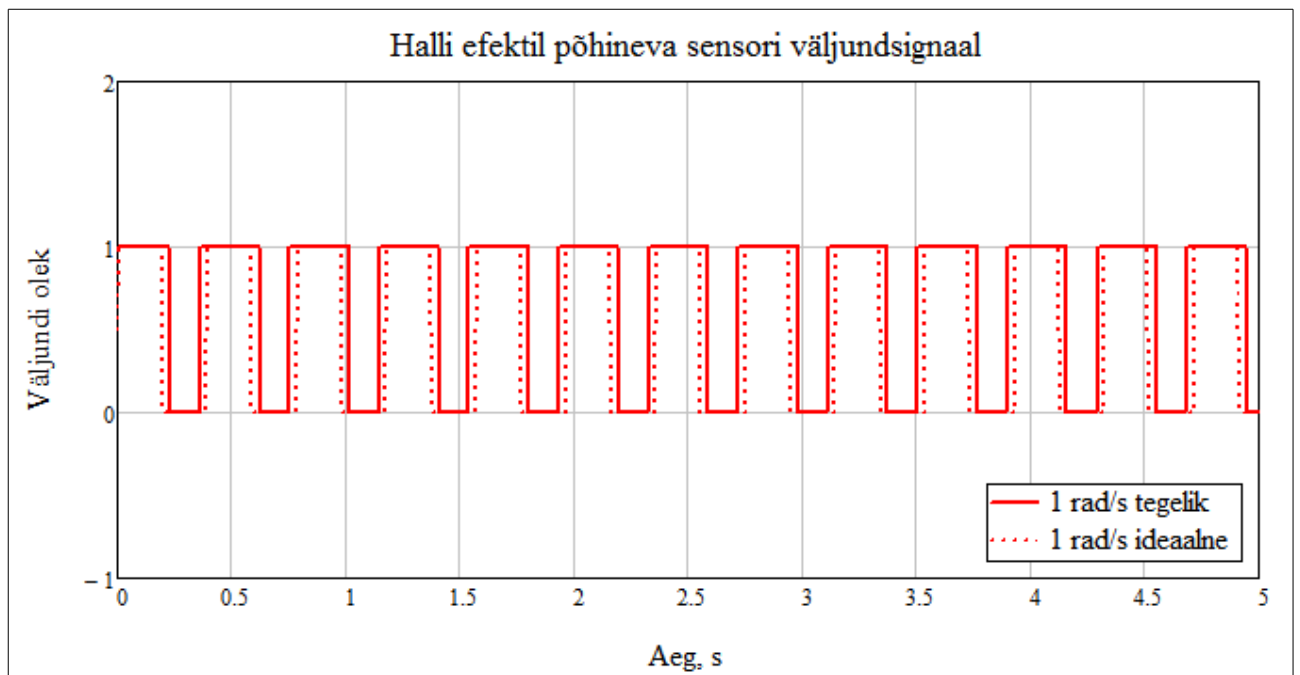
Selle efekti eemaldamiseks saab kasutada vastavast FIR-filtrit. Kuna hüstereesist põhjustatud lugemite nn kõikumine on seotud kahe elemendiga ja kahele elemendile $N = 2$ vastavate

sõltumatute Fourier' teisenduse elementide ehk sageduste arv on $N/2 + 1 = 2$, siis peab ka loodava filtri pikkus olema 2 elementi [13]. Kuna kõik muud muutused tuleb loodavast filtrist läbi lasta ja eemaldada ainult hüstereesiga seotud efekt, siis loodava filtri kordajate vektor H sagedusesituses on järgmine [13]:

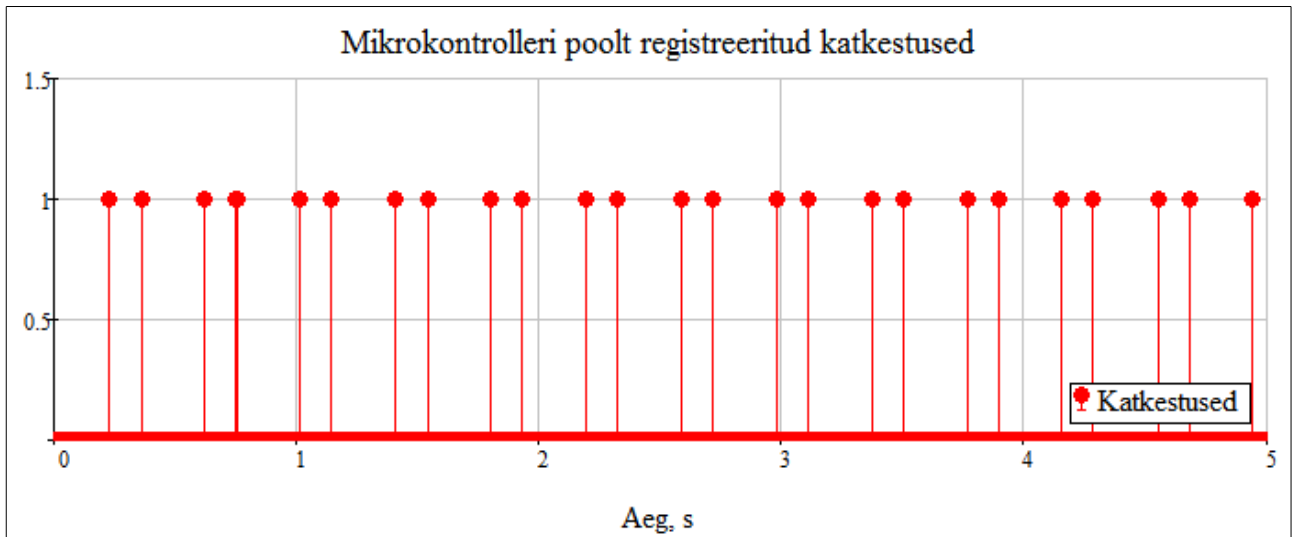
$$H = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (6)$$

See tähendab, et kõik sagedused kahe elemendi lõikes korrutatakse nulliga ja alaliskomponent korrutatakse ühega st ei muuda üldse. Otsitava FIR-filtri saab kui viia filtri kordajad H sagedusesitusest tagasi aegsesse ehk $h = CI \cdot IDFT(H)$. Siin on CI normeerimiskordaja, mis on valitud nii, et vektori h summa oleks 1. Saame [13]:

$$h = CI \cdot IDFT \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \quad (7)$$



Joonis 1: Magnetvälja anduri tegelik väljundsignaal võrreldes ideaalse signaaliga



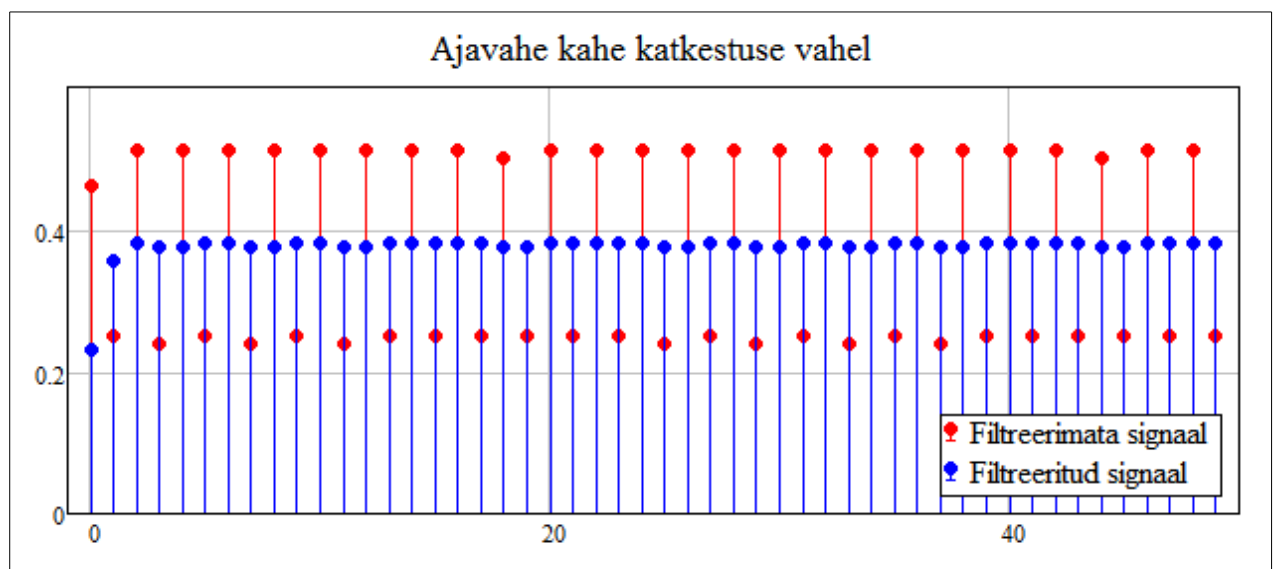
Joonis 2: Mikrokontrolleri poolt registreeritud katkestused, kui ratta pöörlemiskiirus on 10 rad/s

Jooniselt 2 on näha, et ajavahe katkestuste vahel pole konstantne, kuigi ratas liigub ühtlase kiirusega. Signaali filtreerimiseks on kasutusel FIR-filtri, kus võetakse sidum reaalarvuliste lugemite x ning filtri h vahel. Sidum on matemaatiliselt defineeritud:

$$y(n) = \sum_{j=0}^M h_j \cdot x_{n-j} \quad (8)$$

Siin on $y(n)$ filtreeritud väljundsignaali element indeksiga n ning M on vektori h elementide arv. [13]

Näiteks kui signaal on $x = \{0.46, 0.25, 0.51, 0.24, 0.51\}$ ja filter $h = \{0.5, 0.5\}$, siis nende sidum on $y = \{0.23, 0.36, 0.38, 0.375, 0.375\}$.



Joonis 3: Signaali vektor enne ja pärast filtreerimist

Joonisel 3 on näha, et peale FIR-filtri rakendamist on hüsterees eemaldatud.

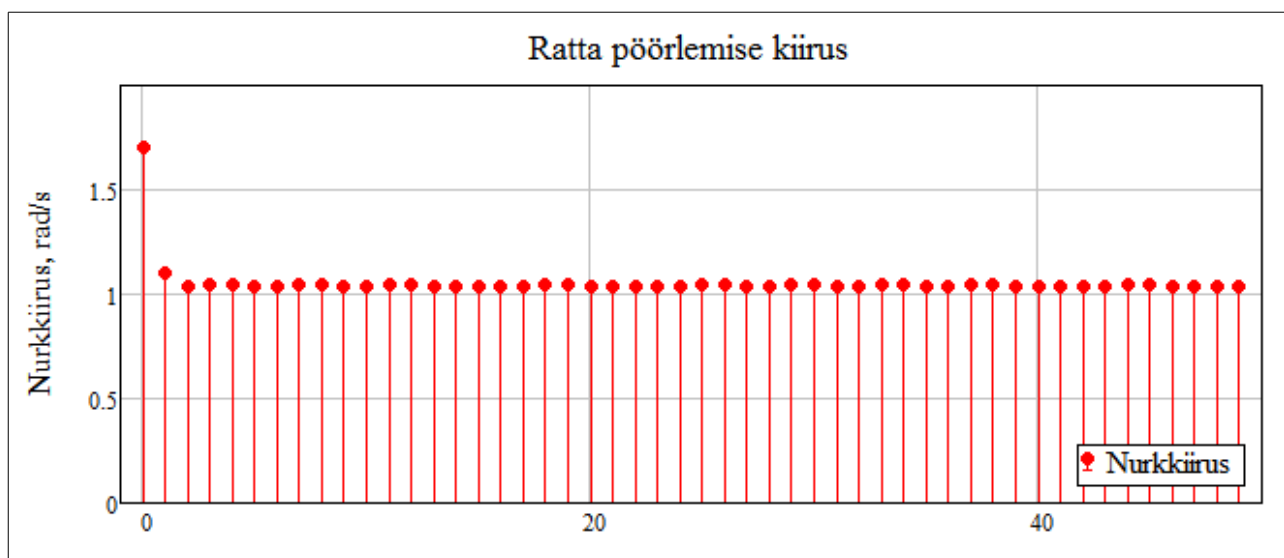
Tarkvaraliselt saab FIR-filtrit realiseerida nii, et üks alamprogramm korrutab ja summeerib ja teine siis nihutab elemente ($n-j$). Sidumi arvutamine definitsiooni järgi pole otseselt realiseeritav, sest elementide arv n pole teada, kuna see sõltub ajast kui kaua programm töötab, mis pole aga ette teada. Seega arvutatakse tarkvaraliselt sidumit järgmiselt:

```
float FIRFilter(float x){  
    float yn = (x + x1) / 2.00;  
    x1 = x;  
    return yn;  
}
```

Kood 2: Alamprogramm sidumi võtmiseks

Siin hoidutakse korrutamisest, kuna kaks korrutamist saab asendada ühe jagamistehtega, mis nõuab vähem arvutusaega. Eelmist aega $x1$ hoitakse mälus, iga kord kui arvutatakse uus väljundelement $y(n)$, asendatakse vastav eelmine ajamoment $x1$ praeguse ajamomendiga x . Sidumi arvutamine pole küll realiseeritud kahe eraldi alamprogrammiga, aga kuna alati on vaja summeerida ainult kahte elementi, siis tuleb nihutada ainult ühte elementi.

Ratta pöörlemiskiiruse arvutamiseks kasutatakse valemit 3. Kui $\Delta t = \{0.23, 0.36, 0.38, 0.375, 0.375\}$, siis $\omega = \{1.71, 1.11, 1.03, 1.05, 1.05\}$.

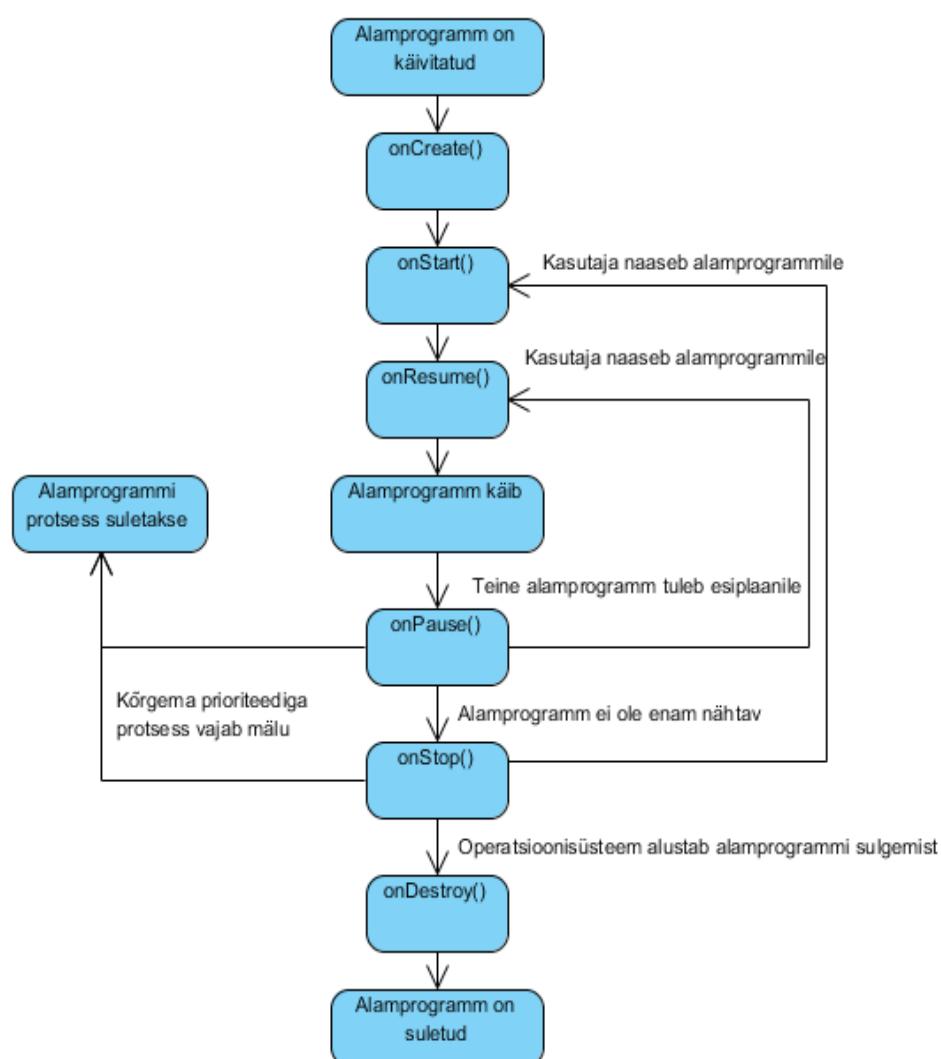


Joonis 4: Ratta pöörlemiskiirus

3.3. Androidi rakendus

Rakendus on kirjutatud Javas, kasutades integreeritud programmeerimiskeskonda Eclipse, millele on paigaldatud ADT (Android Development Tools, Androidi arendustööriistad) pistikprogramm. Viimane lisab Eclipse-ile erinevaid tööriistu, mis muudavad tarkvaraarenduse mugavamaks. Näiteks on seal mallid erinevate Androidi rakenduste lihtsamaks loomiseks ning graafiline tööriist kasutajaliidese loomiseks, mis lubab valida erinevate seadmete, ekraanisuuruste ning teemade (ingl *theme*) vahel. [7]

Alamprogramm Androidi rakenduse mõistes on üks konkreetne asi, mis suhtleb kasutajaga (näiteks seadete menüü). Klass Activity loob akna, kuhu on võimalik kasutajaliidese elemente panna. Tavaliselt töötavad alamprogrammid täisekraanil, aga nad võivad olla ka väiksemate akendena teiste tegevuste peal või sees. Klass Activity hoolitseb kogu alamprogrammi elutsükli eest: alates sellest, kui alamprogramm luuakse, kutsudes välja meetodi onCreate() kuni alamprogrammi sulgemiseni, kui kutsutakse välja meetod onDestroy(). [14]



Joonis 5: Alamprogrammi elutsükkel

Rakendus koosneb neljast alamprogrammist.

- SplashScreen – siin kuvatakse käivitamisel rakenduse logo.
- SettingsActivity – siin saab kasutaja programmi seadistada.
- DeviceListActivity – siin kontrollitakse, kas andmehõivemoodul on ühendatud.
- ActivityGraph – siin kuvatakse graafikul iga tehtud tõmme.

3.3.1. Alamprogramm SettingsActivity

SettingsActivity alamprogrammis kuvatakse kasutajale seadete muutmise kuva, millel saab valida kuna seade magamisrežiimi läheb, mitu tõmmet ekraanil kuvatakse ning milliseid hoorattaid hetkel treenimiseks kasutatakse. Ekraanitõmmis on toodud lisas 2

Ekraani ülaosas on nupp salvestamiseks. Seda vajutades kõigepealt kontrollitakse, et oleks valitud vähemalt üks hooratas, vastasel juhul teavitatakse kasutajat ja palutakse vähemalt üks ratas valida. Lisaks kontrollitakse, et magamisolekusse mineku aeg oleks nullist suurem ning et tegu oleks numbriga. Lõpuks arvutatakse inertsimoment, kasutades valemit 2 ning antakse see koos magamisolekusse mineku aja, graafikul kuvatavate tõmmete arvuga üle alamprogrammile ActivityGraph.

3.3.2. Alamprogramm DeviceListActivity

Alamprogrammis DeviceListActivity otsitakse seadmega ühendatud FTDI seadet, vaadatakse kas on olemas draiver seadme kasutamiseks ning lõpuks antakse seadme UsbSerialDriver objekt ActivityGraph alamprogrammile üle.

Ühendades Androidi tahvelarvutiga õige seadme (andmehõivemooduli), käivitatakse automaatselt ActivityGraph alamprogramm. Kui seadet pole ühendatud, teatatakse, et värskendamine on lõppenud ning ei leitud ühtegi seadet. Ühendades tahvelarvutiga seadme, millel puudub draiver, kuvatakse infoks, et UsbSerialDriver pole saadaval. Leitud seadmete nimekirja värskendatakse iga viie sekundi tagant.

3.3.3. Alamprogramm ActivityGraph

ActivityGraph on rakenduse põhiline alamprogramm, mida kuvatakse suurema osa programmi

käimise ajast. Seal kuvatakse iga tõmbe kohta tulp koos kulutatud energiaga, lisaks näidatakse tõmmete keskmist. Pikk vajutus ekraanil kuvab ajaloo, kus on näha kõik tõmbed ning keskmine. Lisaks on ülaosas nupp seadete muutmiseks ning treeningu kestvust näitav kell.

Iga kord kui saabuavad andmehõivemoodulist uued andmed, kutsutakse välja meetod *void OnNewData(final byte[] data)*, mis paneb saabunud juppidest kokku andmevektori ning kutsub välja meetodi *void updateGraph(byte[] data)*. Viimane uuendab graafikut, kasutades klassi *BarGraph*. Energia arvutatakse valemi 1 järgi.

Kui teatud aja möödudes, mida saab seadetest määrata, pole ühtegi tõmmet tehtud, läheb rakendus magamisrežiimi. Kui kasutataval tahvelarvutil on võimalik ekraani heledust muuta kogu ulatuses (0-100%), lülitatakse taustavalgus selleks ajaks välja. Teistel seadmetel viiakse taustavalgus miinimumi ning kuvatakse logo ja kella. Kui andmehõivemoodulist hakkavad saabuma uued andmed, puhastatakse eelmise treeningu ajalugu ning kuvatakse uue treeningu andmeid. Kuvatavate tulpdiaagrammide ekraanitõmmised on toodud lisades 3 ja 4.

3.3.4. Teek *BarGraph*

Lisaks neljale alamprogrammile on kasutuses teek *BarGraph*, mida kasutatakse tulpdiaagrammi joonistamiseks. *BarGraph*-is on eraldi isendiväljad tõmmete energia ja ajaloo jaoks. Iga uue tulba joonistamiseks tuleb kõigepealt uuendada andmeid, kasutades meetodit *addValue()* ning peale seda tuleb välja kutsuda funktsioon *update()*.

Värksendamisfunktsioon tegeleb nii treeningu tulpdiaagrammi kuvamisega kui ka ajaloo näitamisega. Seda, kumba näidatakse saab valida, muutes *boolean showHistory* väärtust. Lisaks saab valida, kas kuvatakse keskmise väärtuse joont, muutes *boolean showMean* väärtust.

Update() funktsiooni välja kutsudes joonistatakse kõigepealt tulbad, siis tekst tulpade kohale ning peale seda vajadusel joon, mis tähistab keskmist väärtust.

4. Tulemused ja analüüs

Selles peatükis antakse ülevaade erinevatest probleemidest, mis tekkisid manussüsteemi luues ning kuidas need lahendatud said.

Andmehõivemoodulis olev magnetvälja andur ei tohi olla magnetitest liiga kaugel, sest vastasel juhul ei registreerita kõiki magneteid, mis andurist mööda lähevad. See põhjustab ratta nurkkiiruse arvutamise juures ebatäpsusi. Testimise käigus selgus, et optimaalne kaugus magnetite ja anduri vahel on üks sentimeeter.

Anduri töötamiseks ei tohi olla treeningseadme raam maandatud ega ühendatud toite maaga. Samas peab olema toide maandatud. Kui puuteanduri ning treeningseadme raami vahel pole piisavalt head elektrilist ühendust, siis tekib anduris vigane signaal (palju puute tuvastamisi, mis seadme iga natukese aja tagant üles äratavad). Selle lahendamiseks on andmehõivemooduli tarkvaras kontroll, mis ühendab puuteanduri küljest katkestuse lahti, kui katkestuse tuleb lühikese aja jooksul mitu korda.

Tahvelarvuti ning andmehõivemooduli vahel ei teki FTDI ühendus alati esimesel korral, sest Androidi rakendus ei pruugi kohe õigeid draivereid leida. Selle parandamiseks otsitakse rakenduses ühendatud FTDI seadet iga sekundi tagant, kuni operatsioonisüsteem on õige draiveri leidnud.

Tagamaks, et manussüsteem suudaks akutoitelt piisavalt kaua töötada, tuleb energiatarbimine puhkeolekus viia võimalikult madalale. Süsteemil on kolm neli erineva energiatarbimisega olekut. Nende täpsem kirjeldus on toodud lisas 5.

- **Tavaline töö.** Tahvelarvuti ekraan on sisse lülitatud, andmehõivemoodulil on toide ning andmevahetus on aktiivne. Energiatarve on umbes 250 mA.
- **Magamine.** Tahvelarvuti ekraan on välja lülitatud ning süsteemi saab äratada ratta liikumise või puutega. Energiatarve on umbes 60 mA.
- **Sügav magamine.** Tahvelarvuti ekraan ning andmehõivemoodul on välja lülitatud. Energiatarve umbes 5 mA.
- **Välja lülitatud.** Tahvelarvuti on välja lülitatud, energiatarve puudub.

Kasutajatarkvara automaatseks uuendamiseks on see üles laetud Google Play-sse. Rakendustepoodi uue tarkvaraversiooni laadimisel uueneb tahvelarvuti rakendus automaatselt umbes ööpäeva jooksul.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli projekteerida andmehõivemooduli riist- ja tarkvara ning rakendus Androidi tahvelarvutile. Lõppeesmärgiks oli kuvada treenijale kulutatud energia infot. Selleks tuli kõigepealt mõõta hooratta nurkkiirus, kasutades Halli efektil põhinevat magnetvälja andurit ning hoorattaga samale völli kinnitatud ketas, millesse oli paigaldatud magnetid. Mikrokontroller luges anduri poolt saadud impulsse ning arvutas nende põhjal nurkkiiruse ja nurkkiiruse ning saatis need FTDI FT232RL mikrokiipi, mis saatis andmed omakorda üle USB-ühenduse tahvelarvutisse. Tahvelarvutis töötas Androidi rakendus, mis kuvas peale treeningseadmele kinnitatud hooratate valikut kulutatud energia tulpdiagrammina.

Esiteks tuli kirjeldada süsteemile esitatud nõuded. Siis tuli valida andmehõivemooduli jaoks sobiv riistvara, mis võimaldaks registreerida magnetvälja muutumist ning puudutust. Lisaks pidi moodul suhtlema tahvelarvutiga üle virtuaalse COM-pordi.

Andmehõivemooduli tarkvara peatükk käsitles nii selle üldist ülesehitust kui ka magnetvälja andurist tuleva signaali filtreerimist. Viimane oli vajalik, kuna andurist tulev signaal ei olnud hüstereesi tõttu ühtlane. Filtreerimise juures oli kirjeldatud kasutatav filter ning näidatud, kuidas erineb tegelik signaal ideaalsest. Lisaks oli välja toodud, kuidas signaal muutus filtreerimise käigus.

Androidi rakenduse peatükk kirjeldas kasutajatarkvara loomist. Alustuseks tuli leida teegid, mis võimaldaks luua ühendust üle FTDI ning joonistada tulpdiagramme. Edasi oli vaja luua tegevused, mis tegelevad seadete menüü kuvamise, õige FTDI draiveri otsimise ning tulpdiagrammi kuvamisega. Kuna ei õnnestunud leida graafikute loomise teeki, mis võimaldaks mugavalt tulpdiagramme luua, tuli vastav teek ise kirjutada.

Viimases osas kirjeldati erinevaid probleeme, mis tekkisid süsteemi luues ning pakuti neile lahendusi.

Androidi rakendust on võimalik edasi arendada, lisades sinna serveri ning veebirakenduse, kuhu salvestatakse kasutaja tehtud treeningud. See võimaldab treeneril ning treenijal näha, kuidas on tulemused aja jooksul muutunud ning kas areng on toimunud positiivses või negatiivses suunas.

Android OS based metering system for inertial flywheel exercise machines

Antti Jaaniste

Summary

The aim of the present project is to design software and hardware for a practical data acquisition system. The basic principle of operation is the following. The resistance for the trainee is determined by the moment of inertia of the used flywheel. By knowing the angular speed of the flywheel, energy E can be calculated. For calculating the rotational speed of the flywheel special wheel with 16 magnets installed inside and Hall-effect magnetic sensor are used. The sensor measures changing magnetic field and converts it to impulses. Microcontroller reads those impulses and calculates rotational speed and acceleration. Then it sends data to FTDI FT232RL microchip which transfers those numbers to Android tablet via USB. The Android software reads data and calculates energy of the flywheel by using user inputted data of selected flywheels.

The system should be autonomous, suitable for many different situations and also should have low power consumption. The project starts with theoretical study on embedded systems and description of Android operating system.

The second part focuses on designing the project. Firstly, it describes the system requirements. Next, the different hardware components are described with their features highlighted. In addition, there is description of how different components were used and how they were connected to the microcontroller.

The third part part describes the software used in microcontroller and tablet. It shows how the software works and what was needed to take into consideration. There is also description for FIR-filter used in microcontroller to reduce the effects of hysteresis. For the tablet software the different activities and the library BarGraph are described.

Last part focuses on different problems that came forth while designing the system and how they were solved.

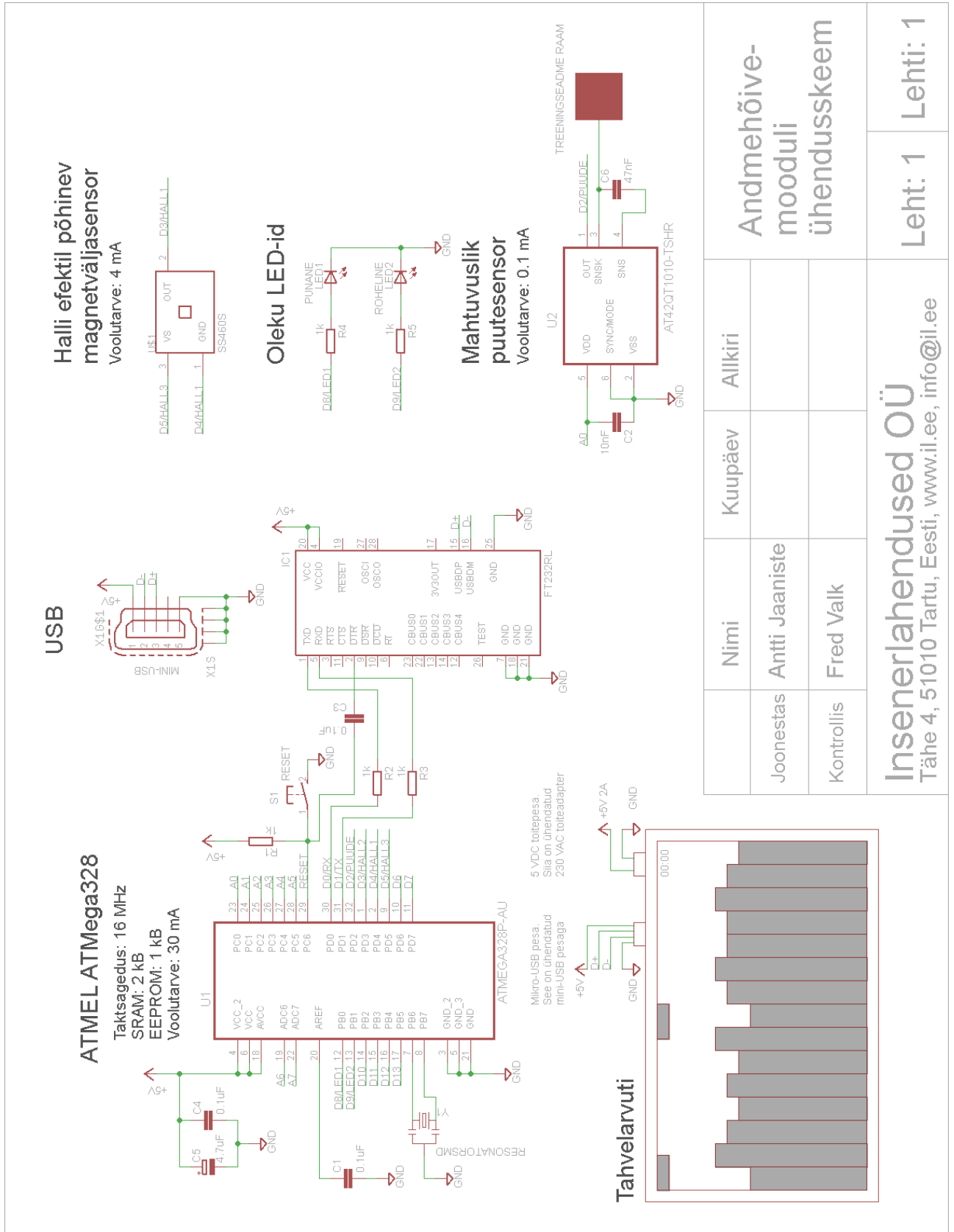
Viidatud allikad

1. S. Heath, *Embedded Systems Design, Second edition* (Newnes, Oxford, 2003)
2. B. Elgin, „Google Buys Android for Its Mobile Arsenal,“ WebCite (2005),
<http://www.webcitation.org/5wk7sIvVb>
3. M. Butler, „Android: Changing the Mobile Landscape,“ in *IEEE Pervasive Computing* (2011), pp. 4-7
4. E. Burnette, *Hello, Android - Introducing Googles Mobile Development Platform* (The Pragmatic Bookshelf, Dallas, 2008)
5. „1st Level Hall-Effect Sensors : SS400,“ Honeywell | Sensing and Control,
http://sensing.honeywell.com/product%20page?pr_id=143710
6. „Atmel AT42QT1010 Single-key Qtouch® Touch Sensor IC datasheet,“ Atmel,
http://www.atmel.com/Images/Atmel-9541-AT42-QTouch-BSW-AT42QT1010_Datasheet.pdf
7. „Developer Tools,“ Android Developers, <http://developer.android.com/tools/index.html>
8. „Hall effect sensing and application,“ Honeywell, <http://sensing.honeywell.com/honeywell-sensing-sensors-magneto-resistive-hall-effect-applications-005715-2-en.pdf>
9. Richard Fitzpatrick „The Hall effect,“
<http://farside.ph.utexas.edu/teaching/302/lectures/node74.html>
10. „Blue 7" Android 4.2 tablet,“ eBay, <http://www.ebay.com/itm/New-Blue-7-Inch-Android-4-2-Tablet-PC-8GB-DUAL-Camera-WiFi-3G-512MB-A13-Case-/271362363704>
11. F. Vahid, T. Givargis, „Embedded system design: a unified hardware/software approach“ (1999), <http://dsp-book.narod.ru/ESDUA.pdf>
12. „Activity Reference,“ Android Developers,
<http://developer.android.com/reference/android/app/Activity.html>
13. L. Tan, *Digital signal processing. Fundamentals and applications* (Academic Press, Elsevier, 2008)

14. „Angular velocity masurement,“ <http://www.slideshare.net/ChiragSolanki333/angular-velocity-measurement>
15. „Atmel ATmega328P microcontroller,“ Atmel, http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf

Lisad

Lisa 1. Andmehõivemooduli ühendusskeem



Lisa 2. Androidi rakendus. Seadete menüü


Save

Sleep after (s)
15


Number of pulls
How many pulls are shown on the history chart
2

Select wheel

PVC inertia 0,5
Thickness 12 mm, weight 1,45 kg



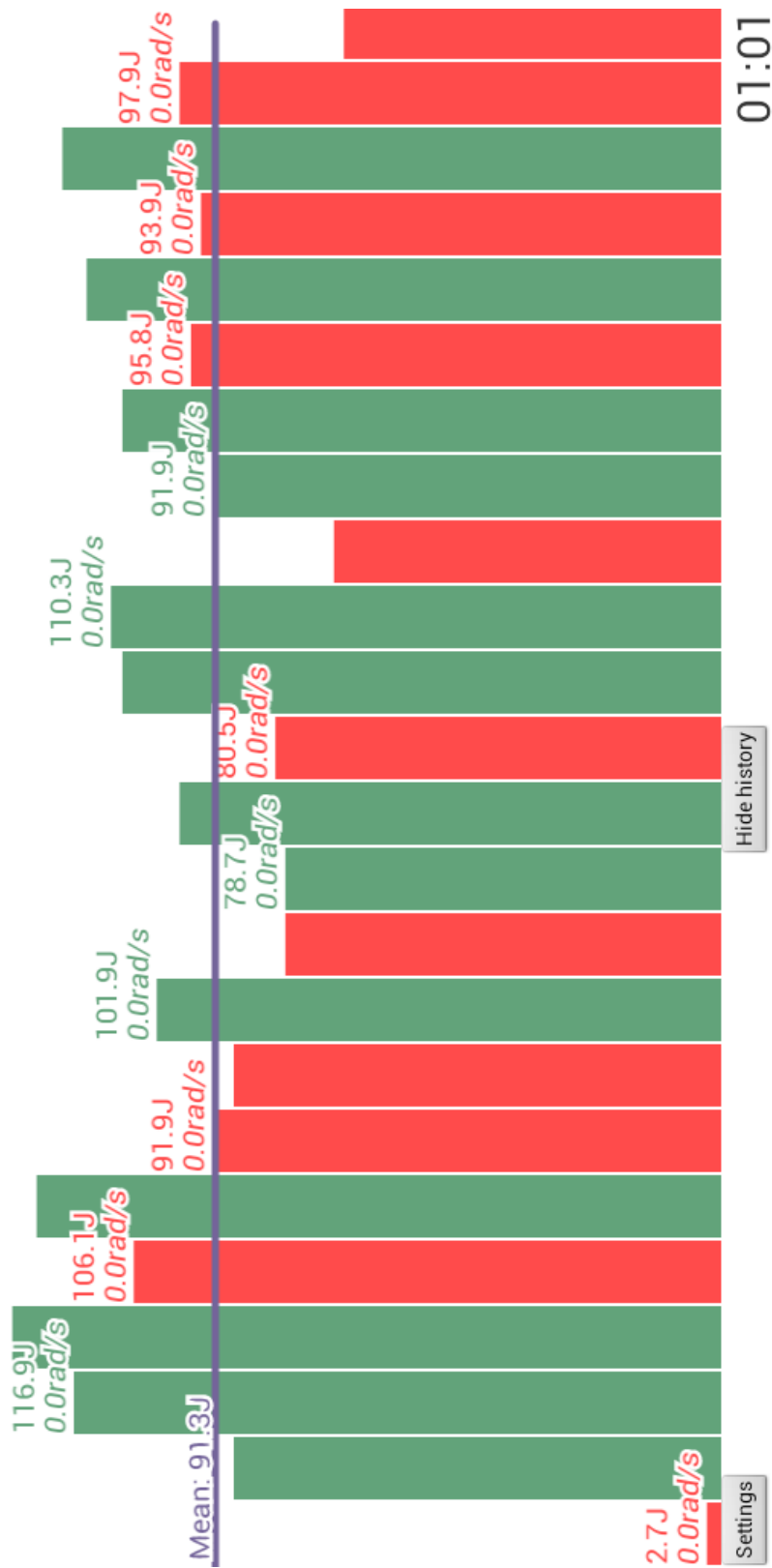
PVC inertia 1
Thickness 20 mm, weight 2,38 kg



Lisa 3. Androidi rakendus. Tehtud tömmete graafik



Lisa 4. Androidi rakendus. Tehtud tömmete ajalugu



Lisa 5. Manussüsteemi erinevad olekud.

Nr	Olek	Andmehõivemoodul	Tahvelarvuti	Kommentaariid
1	Tavaline töö	Töötab, tarbib umbes 30 mA	Töötab, tarbib umbes 200 mA	Andmevahetus on aktiivne, ekraan on sisse lülitatud, läheb teise olekusse, kui teatud aeg ei ole ratta liikumist või puudet tuvastatud
2	Magamine	Magab, andurid töötavad, tarbib umbes 10 mA	Töötab, ekraan on välja lülitatud, tarbib umbes 50 mA	Andmevahetus ei ole aktiivne, liigub esimesse olekusse ratta liikumise või puute peale. Kui teatud aeg ei ole ratta liikumist või puudet tuvastatud, läheb kolmandasse olekusse
3	Sügav magamine	Välja lülitatud, andurid ei tööta, voolu ei tarbi	Magab, tarbib umbes 5 mA	Esimesse olekusse saab liikuda ainult tahvelarvuti sisselülitamisnuppu vajutades
4	Välja lülitatud	Välja lülitatud, voolu ei tarbi	Välja lülitatud, voolu ei tarbi	Esimesse olekusse saab liikuda ainult tahvelarvuti sisselülitamisnuppu vajutades

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Antti Jaaniste

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
“Android operatsioonisüsteemil põhinev mõõtesüsteem treeningseadmetele“,
mille juhendaja on Fred Valk
 - 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 29.05.2014