

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Thomas Johann Seebecki elektroonikainstituut

Jälgimiselektronika „targale“ nõelale

Bakalaureusetöö

IB40LT

Üliõpilane: Marianne Nugis

Üliõpilaskood: 112601IAEB

Juhendaja: doktorant Asko Ristolainen

Kaasjuhendaja: dotsent Mihhail Pikkov

Tallinn 2014

Autorideklaratsioon

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teise autori tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Käesolevat tööd ei ole varem esitatud kaitsmisele kusagil mujal.

Kuupäev:

Autor:

Allkiri:

Lõputöö ülesanne

Lõputöö teema:

Jälgimiselektronika „targale” nõelale

Electronics for smart needle

Teema päritolu: TTÜ Biorobotika keskus

Lõputöö eesmärgid: Elektroonikasüsteemi disainimine, andurite testimine, mikrokontrolleri programmeerimine tagasiside andmiseks ning andmete edastuseks arvutisse, olemasolevate menetlusradioloogia objektiivsete hindamissüsteemide ülevaade.

Lahendatavad küsimused: Elektroonika disain peab olema võimalikult väike (kogu elektroonika võiks ära mahtuda 20 ml süstla korpusesse). Kirjutada selline kood, mis tagastaks andurite andmeid ning annaks andurite andmete alusel kasutajale tagasisidet, et oleks võimalik saadud andmeid säilitada.

Annotatsioon

Marianne Nugis (2014). Jälgimiselektronika „targale” nõielaale.

Bakalaureusetöö. Tallinna Tehnikaülikool. Infotehnoloogia teaduskond.

Thomas Johann Seebecki elektroonikainstituut. Käsikiri. Tallinn. 40 lk. 12 kasutatud allikat. 21 joonist. 2 tabelit. 1lisa.

Töö eesmärk on luua jälgimiselektronika „targale“ nõielaale, mis võimaldaks salvestada radioloogia residentide käelise tegevuse mustreid: nõielaale ja süstla kiirendusi, pöördnurki, nõielaale mõjuvaid jõude. Seade peaks võimaldama residentidel saada töö käigus otsest tagasisidet. Salvestatud andmeid peaks saama hiljem kasutada käelise töö efektiivsuse hindamiseks ning võrdlemiseks professionaalsete radioloogide töömustritega.

Töö käigus uuriti olemasolevate menetlusradioloogia hindamissüsteeme, valiti vajalikud andurid (rotatsiooniandurid, kiirendusandurid), programmeeriti mikrokontrollerit andmete edastamiseks arvutisse, disainiti elektroonikasüsteemi.

Töö tulemusena valmis disainitud ja loodud elektroonikasüsteem, mis jälgib kasutaja käelist tegevust.

Bakalaureusetöö on kirjutatud eesti keeles, sisaldades teksti 36 leheküljel, ühte lisa 4 leheküljel, kuut peatükki, 21 joonist, kahte tabelit.

Abstract

Marianne Nugis (2014). Electronics for smart needle. Bachelor thesis. Tallinn University of Technology. Faculty of Information technology. Thomas Johann Seebeck Department of Electronics. Manuscript. Tallinn. 40 pages. 12 used sources. 21 figures. 2 tabels. 1 appendix.

The goal of the thesis was to create and design electronics for smart needle which would be able to save radiology residents manual activity patterns: needle and syringe accelerations, angles of rotation, the forces exerted on the needle. The device should be capable of providing direct feedback to the residents during operation. The recorded data is later used as the measure of efficiency of the manual activity and compared to professional radiologists' work patterns.

During the work I examined the existing assessment systems of interventional radiology. I chose necessary sensors (rotation sensors, acceleration sensors), programmed the microcontroller to transfer the data to the computer and designed the electronic system.

The result of the work was a system of electronics which monitors the user's manual activity.

The bachelor thesis is written in Estonian, it contains of 36 pages of text, 4 pages of appendices, 6 chapters, 21 figures, 2 tables and 1 appendix.

Sisukord

Sissejuhatus	8
1 Menetlusradioloogia objektiivsete hindamissüsteemide ülevaade.....	9
1.1 Erinevad treeningvahendid menetlusradioloogias.....	9
1.1.1 Erinevad treeningvahendid – kasutusel olevad simulatsioonid.....	9
1.1.2 Ultraheliga teostatud menetlusradioloogia nõela simulatsioonid (<i>Imaging Guided Interventional Needle Simulation</i>).....	11
1.1.3 Virtuaalne simulaator menetlusradioloogias kasutatavates protseduurides (<i>A Computer Based Simulator for Ultrasound Guided Needle Insertion Procedures</i>).....	12
1.2 Biorobootika keskuse fantoomid.....	13
1.3 Töö teema aktuaalsus.....	14
2 Ülesannete püstitus elektroonikasüsteemile	15
2.1 Tehnilised nõuded elektroonikasüsteemile.....	15
2.1.1 Plaanitava mudeli disain	15
2.2 Konkreetsete andurite valimine ja tööpõhimõte	17
2.3 Katse painde- ja jõuanduri võimenduse valimine	18
2.3.1 Paindeanduri võimenduse valimine.....	18
2.3.2 Arvutuslik meetod.....	19
2.3.3 <i>Wheatstone'i</i> silla tasakaalu arvutus	19
2.3.4 <i>Wheatstone'i</i> silla tasakaalu katse ja paindeanduri töökatse.....	20
2.3.5 Jõuanduri võimenduse valimine	21
2.3.6 Jõuanduri töökatse.....	22
3 Elektroonikasüsteemi disain.....	24
3.1 Elektriskeem	24
3.2 Elektriskeemi kirjeldus.....	25
3.2.1 Instrumentaalvõimendi ja virtuaalne maa kirjeldus	25
3.3 Trükkplaat.....	25
3.3.1 Trükkplaadi disain	26
3.3.2 Komponentid.....	27
3.4 Trükkplaadi 3D mudel.....	28
4 Programmeerimine	29
4.1 Programmeerimise ülesannete püstitus	29
4.2 Tehnilised nõuded	29
4.3 Ülesannete lahendus	29

5 Elektroonikasüsteemi maksumus.....	31
6 Saadud tulemused	32
6.1 Elektroonikasüsteemi mehaanikadisain	33
Kokkuvõte.....	34
Summary	35
Kasutatud kirjandus.....	36
LISAD.....	37

Sissejuhatus

21. sajandi alguses pööratakse suurt tähelepanu meditsiinis kasutatavate tehnoloogiate arengule, mis tõstavad osutatava tervishoiuteenuse kvaliteeti. Pingsalt püütakse täiustada olemasolevaid ravimeetodeid ja luua uusi ning paremaid biomeditsiinilisi meetodeid. Tehnoloogia arengust tingituna täienevad erinevad menetlusradioloogia protseduurid, koos protseduuride arenguga ka nende treeningmeetodid. Protseduuride arenguga kaasas käimiseks peavad ka radioloogid pidevalt täiendama oma erialaseid oskusi. Radioloogia residentid omandavad viie aastase residentuuriõppe käigus lisaks teoreetilistele teadmistele ka praktilised oskused menetlusradioloogiast. Menetlusradioloogia sissejuhataval kursusel saavad radioloogia residentid esimese kogemuse patsiendiga töötamiseks, kus harjutatakse erinevaid minimaalse invasiivsusega protseduure. Sissejuhataval kursusel puutuvad nad esimest korda kokku nägemiskäelise praktilise töö keerukusega ja erinevate protseduuride iseärasustega. Hetkel hindab residentide käelist tegevust haiglas radioloog subjektiivselt. Residentide töö objektiivseks hindamiseks oleks vaja juhtmevaba seadet, nn „tarka“ nõela, mis võimaldaks salvestada residentide käelise liikumise mustreid (nõela ja süstla kiirendusi, pöördenurki, nõelale mõjuvaid paindejõude) võimaldamaks anda residentile otsest tagasisidet töö käigus. Salvestatud töomustreid oleks hiljem võimalik kasutada käelise töö efektiivsuse hindamiseks ning võrdlemiseks professionaalsete radioloogide töomustritega.

„Targa“ nõela põhiliseks funktsiooniks õppetöö käigus oleks jälgida radioloog-residentide käelist tegevust ning anda teada ohtlikest olukordadest ning salvestada informatsiooni nende käelise tegevuse kohta. Radioloogid saaksid kasutada „tarka“ nõela oma igapäevatoos ja operatsioonidel, lisaks aitaks see tuvastada nõelprotseduuridel erinevate koekihtide penetratsiooni ning parandada ka radioloogide töö kvaliteedikontrolli.

Bakalaureusetöö eesmärk on luua jälgimiselektroonika „targale“ nõelale, mis võimaldaks salvestada radioloogia residentide käelise tegevuse mustreid: nõela ja süstla kiirendusi, pöördenurki, nõelale mõjuvaid jõude. Seade peaks võimaldama residentidel saada töö käigus otsest tagasisidet. Salvestatuid andmeid peaks saama hiljem kasutada käelise töö efektiivsuse hindamiseks ning võrdlemiseks professionaalsete radioloogide töomustritega.

Järgnevates peatükkides täpsustatakse käesoleva bakalaureusetöö ülesannet ja luuakse sobiv jälgimiselektroonika „targale“ nõelale.

1 Menetlusradioloogia objektiivsete hindamissüsteemide ülevaade

1.1 Erinevad treeningvahendid menetlusradioloogias

Kaasaegsed meditsiinitehnoloogias kasutatavad simulatsioone mainiti esimest korda 1950. aastatel nõ teedrajavas töös kunstliku hingamissüsteemi arendamisel [1], millele järgnes arvutiga juhitud mannekeensimulatsioon arstitudengitele ning residentidele praktilise töö harjutamiseks (mille loojad olid Denson ja Abrahamson) [2]. 1990ndatel arendati välja objektiivne struktureeritud hindamisvahend (*objective structured assessment*), mille eesmärgiks oli hinnata kirurgia residentide tehnilisi oskusi [3].

1.1.1 Erinevad treeningvahendid – kasutusel olevad simulatsioonid

1) Loommudel (*animal model*). Loommudelitel on suurem anatoomiline korrektsus kui sünteetilistel mudelitel, mis annab eelise nii avatud kui ka minimaalsete protseduuride imiteerimise ees. Loommudelite kasutus on limiteeritud nende hinna, eetiliste probleemide, litsentside ja patoloogia puudustega.

Loommudeli plussid: kõrge tõhusus, terve protseduuri simuleerimine, potentsiaali hindamine.

Loommudeli miinused: kallis, eetilised probleemid, eriseadmete vajadus loomade elus hoidmiseks, anatoomiliste iseärasuste puudujäägid (kasvajate olemasolu).

2) Inimese organid (*human cadaver*). Inimese surnukeha vereringe mudel pakub soodsamaid tingimusi veresoonte sisesteks katseteks. Lisaks veresoonekonna mudelile saavad kirurgia tudengid harjutada erinevaid operatsioone. Joonisel 1.1 on esitatud inimese surnukeha [4].

Inimese organite plussid: kõrge treeningtulemus, kõrge testitavuse tulemus, täielik simulatsiooni protseduur.

Inimese laiba miinused: limiteeritud kättesaadavus, säilitamine, kõrge hind [5].



Joonis 1.1 Inimese surnukeha

3) Fantoom ehk tehisorgan. Tehisorgan on inimese organi mudel, mida kasutatakse erinevate protseduuride õpetamiseks. Näiteks radioloogia residendid õpivad neeru fantoomide peal punkteerimist ja nefrostoomi dreeni paigaldamist. Joonisel 1.2 on esitatud neeru fantoom [6].

Fantoomi plussid: kõrge treeningtulemus, kahjustatud tehisorgani osad saab kergesti asendada, erinevalt virtuaalsest keskkonnast saab tunnetada käelise tegevust.

Fantoomi miinused: tagasiside puudub, arst peab jälgima residendi tegevusi, organite osad on lihtsustatud (ultrahelipildi jaoks), sisestruktuuride puudumine organitel, limiteeritud võimalused ja kasutuskorrad.



Joonis 1.2 Neeru fantoom

4) VR (virtuaalne reaalsus) simulatsioon (*virtual reality*). VR (vt joonis 1.3 lk 11) on kommunikatsiooniliides, mis põhineb patsiendi organite imiteerimise kolmemõõtmelisel mudelil.

VR simulatsiooni plussid: korduvalt saab katseid teha, andmeid saab talletada, minimaalne seadistamise aeg, kõrge reaalsus, võimaldab läbi viia anatoomia seisukohalt täpsemaid protseduure kui tehisorganiga, tagasiside toimub simulatsiooni käigus, patsiendispetsiifilisus.

VR simulatsiooni miinused: hind, hooldus, kolmemõõtmelise mudeli kättesaadavus, käelise tunnetuse tagasiside puudub [7].



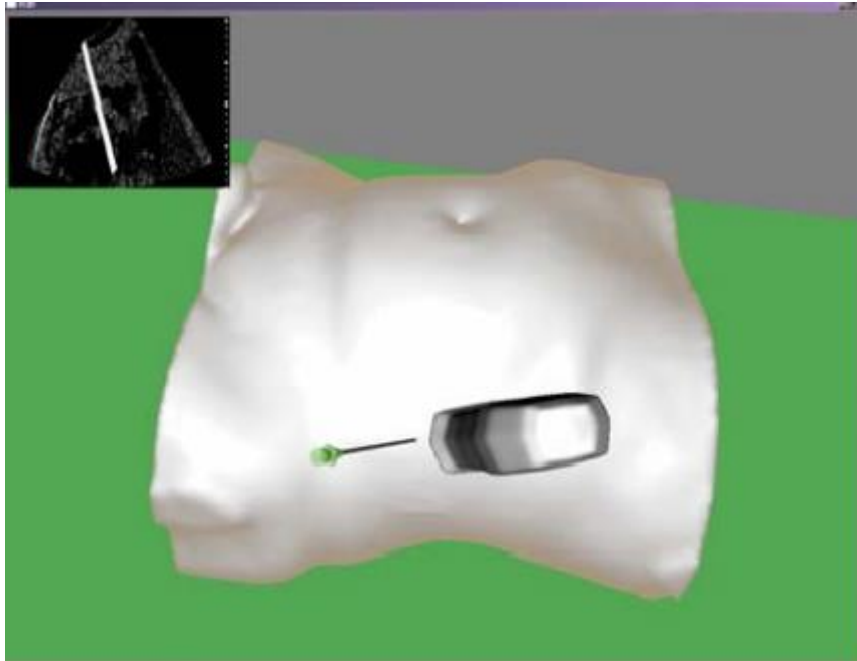
Joonis 1.3 Virtuaalne reaalsus

1.1.2 Ultraheliga teostatud menelusradioloogia nõela simulatsioonid (*Imaging Guided Interventional Needle Simulation*)

Menelusradioloogia kasutab minimaalse invasiivsusega meetodeid, mis põhjustavad vähem valu ja on lühema taastusajaga kui avatud operatsioonid. Menelusradioloogia protseduurid kasutavad röntgen-, ultraheli ja MRT pilti, et juhtida nõel patsiendis konkreetse anatoomia ning patoloogia seisukohalt õigesse ruumipunkti. Röntgen ja ultraheli on enim kasutatud uuringumeetodid.

Tavalised menelusradioloogia oskused, nagu punkteerimine nõuab kindlat väljaõpet. Selleks, et resident ei tekitaks patsiendile liigset valu ja tüsistusi, kasutatakse arvuti baasil simulatsioone.

Piltjuhitav nõela simulatsioon (vt joonis 1.4 lk 12) pakub efektiivset virtuaalset keskkonda, milles resident saab harjutada praktilisi oskusi [8].



Joonis 1.4 Virtuaalne patsient piltjuhitavas simulatsioonis

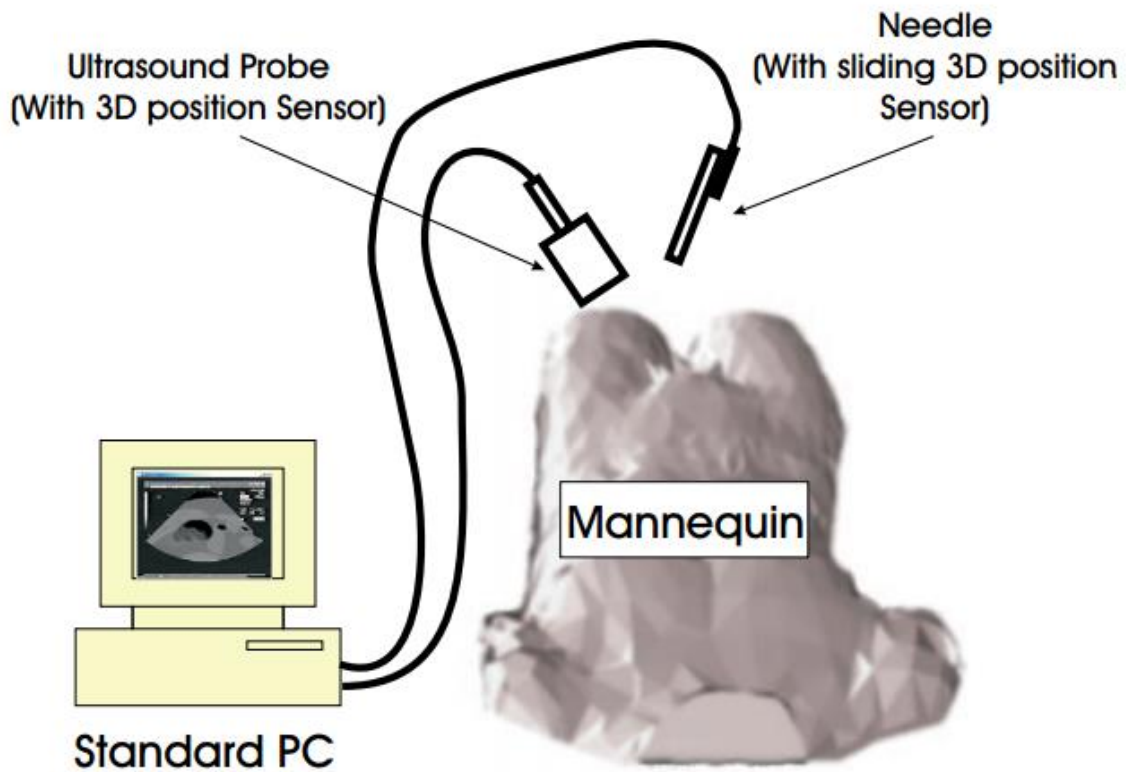
1.1.3 Virtuaalne simulaator menelusradioloogias kasutatavates protseduurides (*A Computer Based Simulator for Ultrasound Guided Needle Insertion Procedures*)

Menelusradioloogia üks põhioskustest on nõelaga punkteerimine ultrahelis. Simulatsioon põhineb graafilisele seadmele, mis pole jäigalt joondatud füüsilise mannekeeni pinnale. Magnetiline kolmemõõtmelise positsiooni andur lubab genereerida virtuaalset ultrahelipilti, mis põhineb ultraheli uuringu ja nõela asukohale.

Protseduuridel torgatakse nõel patsienti, kus ultraheli anduri ekraanil on näha reaajas kahemõõtmelist pilti.

Tarkvara töötleb virtuaalse ultraheli pildi spetsiaalsel ekraanil rekonstrueerimisel saadud andmete alusel. Pilti töödeldakse skaneerides iga pikslit ultraheli pildi osalt ja projekteeritakse iga punkti vastav punkt kolmemõõtmelisele põhinevale simulatsioonile. Andmed registreeritakse mannekeeni pinnalt kolme protseduuri etapiga (vt joonis 1.5 lk 13).

Esiteks, pinnalt on välja võetud mahulised andmed ja mannekeeni punkti kogum. Mahulise pinna andmed võetakse välja kahemõõtmelise kujuna, liikudes üle kogu mannekeeni pinna. Teise etapina registreeritakse jäiga joondumise kaks andmekogumit. Pärast andmete saamist sobivad andmed teisendatakse, kasutades erinevaid valemeid kolmemõõtmeliseks pildiks [9].



Joonis 1.5 Simulatsiooni tarkvara tööpõhimõte

1.2 Biorobotika keskuse fantoomid

Biorobotika keskus on osalenud kahes suures projektis - *SAFROS* ja *I-Sur*, mille raames loodi fantoome ehk tehisorganeid.

Arendati fantoomorganeid, mida saaks kasutada robotkirurgia süsteemide testimisel (*I-Sur* projekt). Eesmärk oli luua organifantoomid, mis imiteeriks reaalseid organeid ultrahelis, kompuutertomograafias ning mis oleks elastsusomadustelt sarnased pärisorganitega (vt joonis 1.6 lk 14).



Joonis 1.6 Biorobootika keskuse neeru fantoom

Sea kõhufantoom loodi selleks, et testida erinevaid seadmeid ja analüüsida robotoperatsioonide efektiivsust. Fantoomi loomise eesmärk oli vähendada loomade kasutamist katsetes, mida rakendas *SAFROS* projekt [10].

1.3 Töö teema aktuaalsus

Biorobootika keskuse loodud neerufantoomi kasutavad radioloogid residentide tööd hindamiseks. Radioloog peab ise juures olema, et hinnata residentide tööd erinevate protseduuride õppimisel fantoomi kasutades. Oleks vaja luua süsteem, mis kontrolliks residentide tööd protseduuride harjutamise ajal. Selleks, et oleks võimalik teha autonoomne hindamissüsteem, peab esmalt projekteerima ning valmistama mõõtevahendi – „targa“ nõela, millega saaks salvestada radioloogiatudengite tööd.

2 Ülesannete püstitus elektroonikasüsteemile

Jälgimiselektronika disainimise eesmärk „targale“ nõiela on anda radioloogidele tööriist, millega oleks võimalik objektiivselt hinnata residentide käelist tööd.

Disainitava elektroonikasüsteemi eesmärgiks on salvestada residentide käelise tegevuse mustreid. Samuti on vaja salvestada ja jälgida nõiela ja süstla kiirendusi, pöördenurki, nõiela mõjuvaid jõude ning paindeid. Disainitavas elektroonikas kasutatav mikrokontroller peab andma anduritelt saadud informatsiooni alusel tagasisidet kasutajale ohuolukorrast. Ohuteavitus peaks kasutajale edastama süstla korpusel indikaator-LED-id või heliallikas. Lisaks tagasiside andmisele on vajalik ka programmeerida mikrokontroller andmeid edastama arvutisse, et hiljem saaks salvestatuid töömustreid kasutada töö efektiivsuse hindamisel.

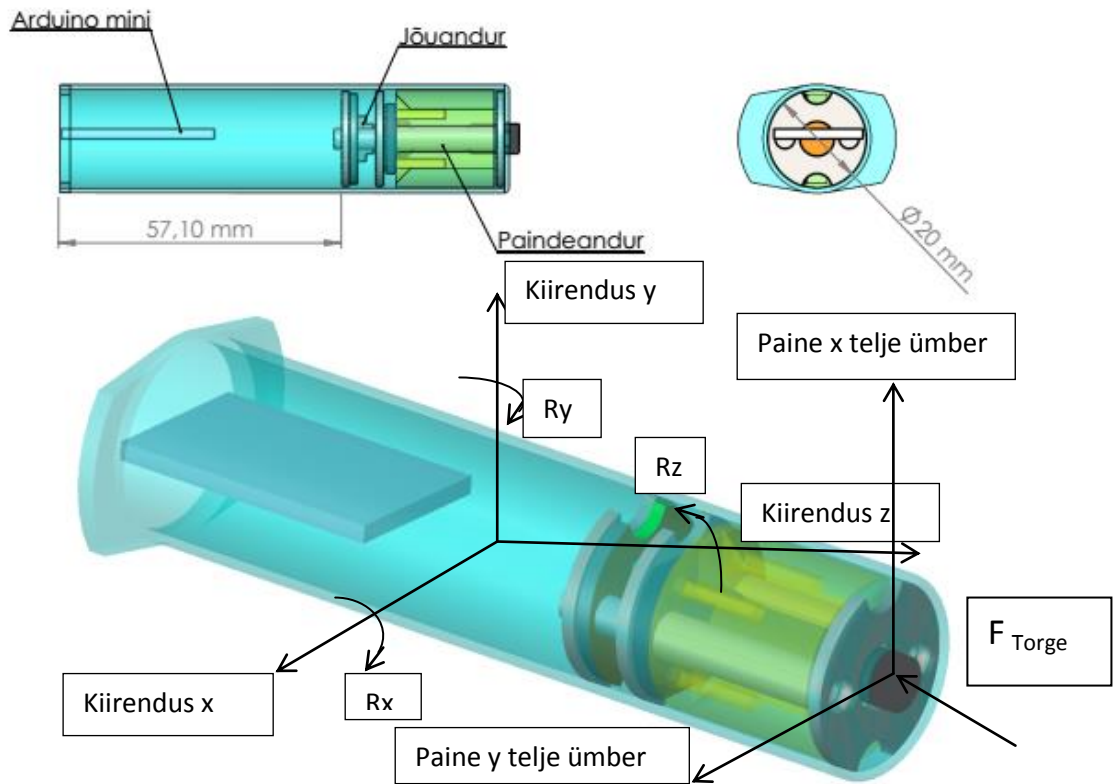
Mehaanikalahendus oli etteantud, millele pidi integreerima elektroonikakomponendid.

2.1 Tehnilised nõuded elektroonikasüsteemile

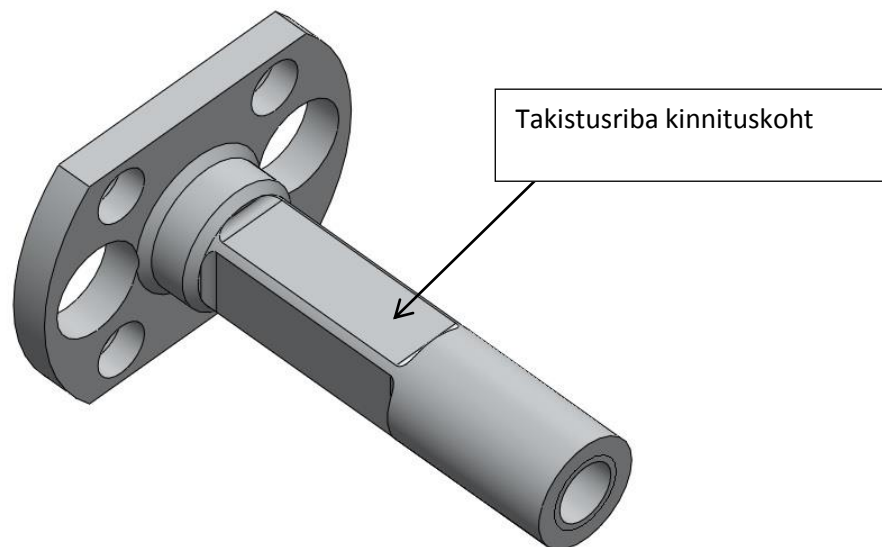
Esialgne nõue kogu elektroonikasüsteemile on, et see mahuks 20 ml süstla korpusesse. Seadmel peaks olema võimalus anda tagasisidet kasutajale töö käigus kas valgusallika või heliallika kaudu. Andurid tuleb valida sobilikud töö ülesannete edukaks täitmiseks. Tuleb valida sobiv rotatsiooniandur, kiirendusandur, paindesüsteemile vastav lahendus. Mikrokontrolleri programmeerimiseks on vaja leida sobib tarkvaraline arendusplatvorm.

2.1.1 Plaanitava mudeli disain

Mudeli pildilt joonisel 2.1 on näha anduritele plaanitud paigutust. Paindeanduri (vt joonis 2.1 lk 16) takistusribad (*strain gauge*) asuvad korpuse alumises otsas paindeelemendil (vt joonis 2.3 lk 16) paindemõõtmiseks. Jõuandur mõõdab nõiela torkejõudu ning asub paindemõõtmise anduri all. Korpuse ülemisse ossa peab ära mahtuma adapterplaat, mis ühendaks omavahel *Arduino Mini* arendusplaadi muu elektroonikasüsteemiga.



Joonis 2.1 "Targa" nõela mehaanikadisain



Joonis 2.2 Paindeanduri paindeelement

2.2 Konkreetsete andurite valimine ja tööpõhimõte

Tuleb valida konkreetset andurid: rotatsiooniandur, kiirendusandur, jõuandur, paindeanduri takistusribad.

Valitud andurid:

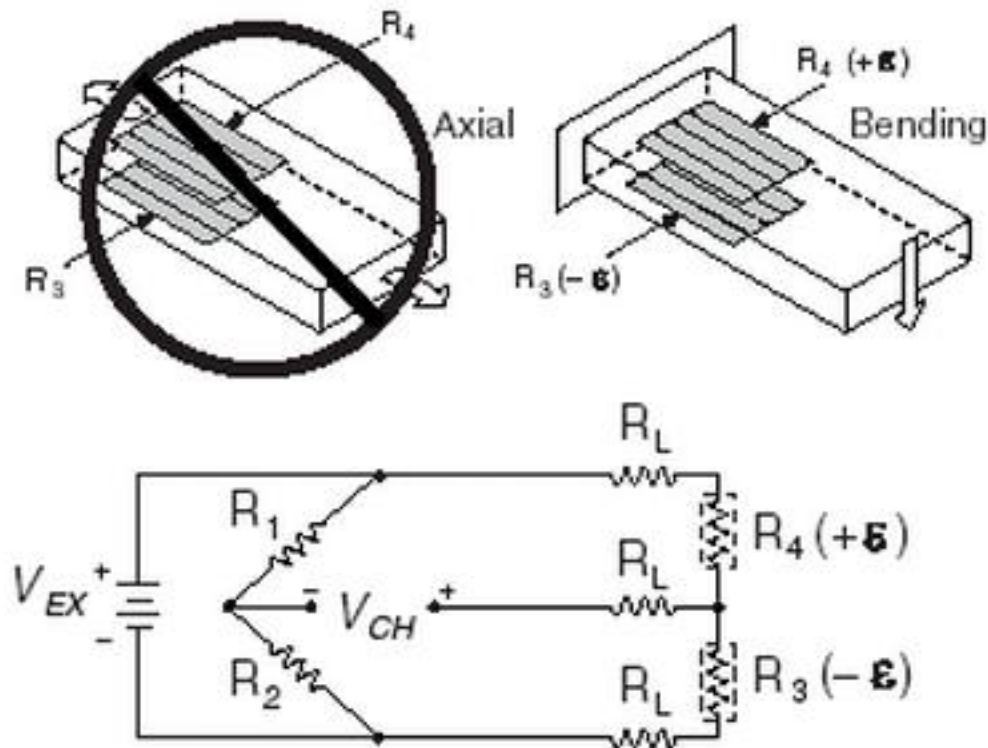
IMU 6DOF-MPU6050 – inertsiaalandur. Inertsiaalandur on mõõteandur, mille sisendosaad mõõdavad joonkiirust ja pöörlemiskiirust.

Valiku põhjendus: inertsiaalanduriks sai valitud IMU 6DOF – MPU6050, kuna anduri mõõtmed on väikesed, siis mahub süstla korpuse sisse. Üks element mõõdab kahte suurust korraga - kiirendusnurki ja pöördemomente. Resolutsioon oli sobilik. Andur ühildus *Arduino Miniga* (läbi I2C ühendusprotokoll).

FSS1500NSB – jõuandur - andur millega mõõdetakse jõu mõju suurust.

Valiku põhjendus: anduri mõõtmed olid sobivad, et kasutada seda andurit konkreetsetes töös. Anduri mõõtmispiirkond oli null grammist 1500 grammi. Mõõta oli vaja väikeseid jõudusid kuni 200 grammi.

Painde mõõtmiseks kasutatakse *Wheatstone*'i silla poolsilla tüüpi II, takistusriba on ühendatud *Wheatstone*'i silda. Füüsiliselt on nad liimitud varra kahele küljele. Poolsilla tüüp II mõõdab ainult paindejõudu. Painde mõõtmine toimub rakendava koormuse korral. Painde rakendamise korral muudetakse koormuse jõud pingeks, mille abil saab painde suurust ja suunda mõõta. Joonisel 2.3 lk 18 on takistid R1 ja R2 püsiva suurusega. Takisti R3 on aktiivse takistusriba (*strain gauge*) element, millega mõõdetakse suruvat jõudu/jõude. Takisti R4 on aktiivse takistusriba element, millega mõõdetakse pingutavat jõudu/jõude [11].



Joonis 2.3 *Wheatstone*'i silla poolsilla tüüp II

2.3 Katse painde- ja jõuanduri võimenduse valimine

Elektroonikasüsteemi trükkplaadil kasutati instrumentaalvõimendit INA333, millele on vaja kindla suurusega takistit (R_g), mis määrab *Wheatstone*'i silla ühe suuna painde võimenduse suuruse.

2.3.1 Paindeanduri võimenduse valimine

Paindeanduri võimenduse määramiseks disainiti ja valmistati katseplaat, mille peal olid järgmised komponendid: instrumentaalvõimendi INA333, virtuaalne maa TLE2426, kondensaator, R_g takisti ja kaks pistikut *Wheatstone*'i silla ühe painde poole takistusribade jaoks.

Katseplaadi valmistamise kirjeldus:

- 1) Skeemi disainimine;
- 2) Plaadi disainimine;
- 3) Skeemi väljaprintimine spetsiaalsele kilele;
- 4) Vaskplaadi ettevalmistamine;
- 5) Skeemi vaskplaadi peale triikimine;
- 6) Kile eemaldamine plaadilt;
- 7) Plaadi söövitamine raudkloriidi lahuses;
- 8) Plaadi puhastamine;
- 9) Komponentide peale jootmine.

Järgmiseks oli vaja leida, kui suur peaks olema R_g takisti väärtus. Alguses katsetati R_g väärtusega $1\text{ k}\Omega$, mis andis 100 kordse võimenduse. Eelnimetatud võimendustegur ei olnud piisav (paindekoormused 0-200 grammi).

2.3.2 Arvutuslik meetod

$$\frac{\text{väljund}}{\text{sisend}} = \text{võimendus}$$

$$\frac{5000\text{mV}}{180\text{mV}} = 27$$

Instrumentaalvõimendi INA333 saab väljund- ja sisendpinge vahet võimendada 27 korda.

Hiljem katsetati R_g väärtusega 100Ω , mis andis võimendusteguriks 1000. See R_g väärtus oli sobiv.

R_g takistuse arvutuse valem:

$$G = 1 + (100\text{k}\Omega/R_g)$$

G - võimendustegur

R_g - võimenduse määrav takisti

$$G = 1 + (100\text{k}\Omega/100,1\Omega) = 1000 \text{ Sobilik võimenduskordaja.}$$

Katse käigus selgus, et võimendi väljundi keskpunkt oli 4,38 V, aga peaks olema 2,5 V lähedal. Selleks, et saada võimendi väljundi keskpunkt 2,5 V, tuleb *Wheatstone*'i silla takistite õiged väärtused välja arvutada.

2.3.3 *Wheatstone*'i silla tasakaalu arvutus

$$\text{Sild on tasakaalus, kui } \frac{R_2}{R_1} = \frac{R_4}{R_3}.$$

Kui takisti R_3 ja takisti R_4 on täpselt paigas, aga takisti R_1 ja takisti R_2 on 1% täpsusega, siis võib juhtuda, et silla voltmeeter näitab 0,5 % valet pinget. Kui kõik takistid on 1% täpsusega, võib ka voltmeetri näit olla 1 % oodatavast erinev.

Ebatäpsuse korrigeerimiseks tuleb ühele takistitest joota paralleelselt teine takisti.

Algne *Wheatstone*'i silla takistus 220Ω , kui täpsus on 1% siis võimalik erinevus on $2,2\Omega$. Võimalik halvim silla konfiguratsioon on selline, kus takisti $R_1 = 222,2\Omega$ ja takisti $R_2 = 217,8\Omega$. Kui teine silla pool on täpselt tasakaalus, siis tuleb paralleelselt joota takisti takistile R_1 , et see takisti R_2 väärtuseni vähendada.

$$R_{\text{uus}} = \frac{R_{\text{paral. Reelm.}}}{R_{\text{eelm.}} - R_{\text{paral.}}} \quad R_{\text{uus}} = \frac{R_2 R_1}{R_1 - R_2}$$

$$R_{\text{uus}} = \frac{217,8\Omega 222,2\Omega}{222,2\Omega - 217,8\Omega} = 11\text{k}\Omega$$

Uus takisti, mida takistile R1 peale joota, peab olema 11 k Ω või suurem ehk sõltub sellest, kui palju takisti R1 takistist R2 tegelikult erineb.

Kui kasutada 0,1% täpsusega takistit ja 5 voldist pinget, siis on maksimaalne võimalik pingeviga 2,5 V.

$R_1 = R_2 = 300 \Omega$ (ideaalis).

Halvimal juhul takisti $R_1 = 300,3 \Omega$ ja $R_2 = 299,7 \Omega$ ning pinget on takistil R2,

$$U_{R_2} = \frac{R_2}{R_1 + R_2} 5V = \frac{299,7 \Omega}{600 \Omega} 5V = 2,4975 V$$

mis tähendab seda, et kõige halvemal juhul on silla poolt pingevahetus:

$$2,5V - 2,4975 V = 2,5mV$$

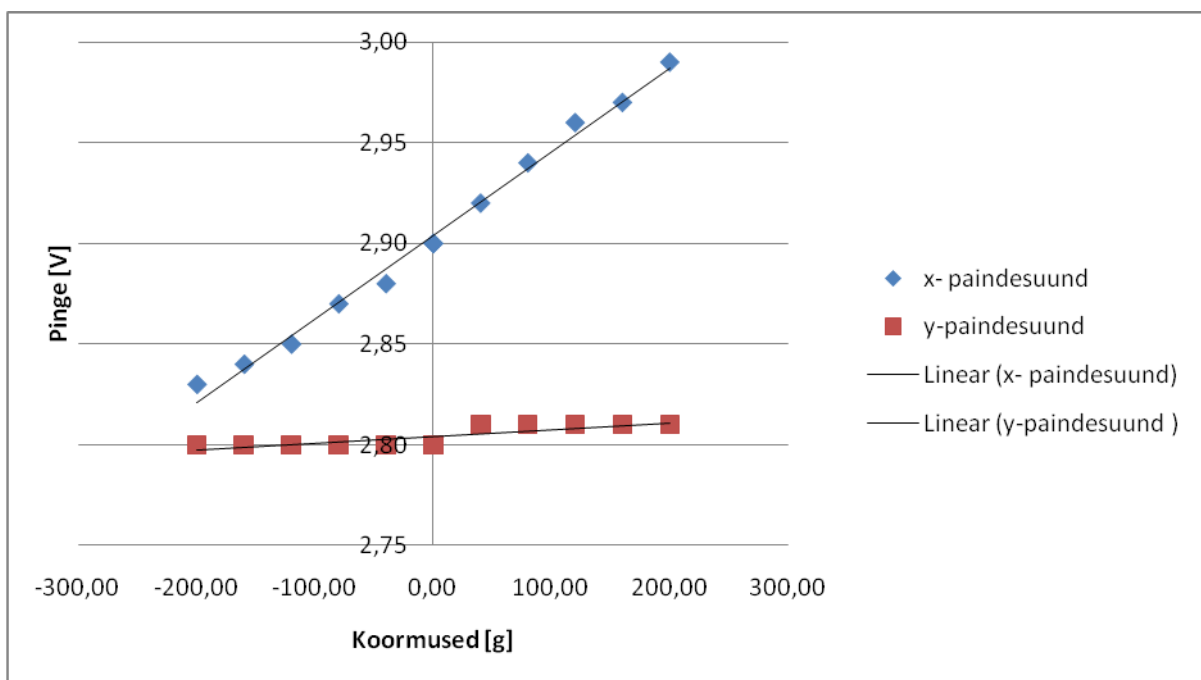
2.3.4 Wheatstone'i silla tasakaalu katse ja paindeanduri töökatse

Esmalt katsetati takisti R1 ja takisti R2 jaoks takistiteid väärtusega 300 Ω ning saadi võimendi väljundi keskpunktid (vt joonised 2.4 ja 2.5 lk 21).

Järgmiseks viidi läbi paindeanduri maksimaalse painde leidmiseks koormuskatse. Maksimaalne koormus, mida võis rakendada, oli enne katseliselt nõela painutamise leitud. Visuaalselt hinnati tehnilikul neerul, kui palju võib koormust rakendada, ilma, et nõel läheks ohtliku painde alla. Maksimaalne jõud, mida peeti ohutuks oli 215 grammi.

Koormustega katsed toimusid järgnevate raskustega. Esialgu ei olnud koormust, mistõttu lisati 20 g ja nii seni kuni koormus oli 200 grammi.

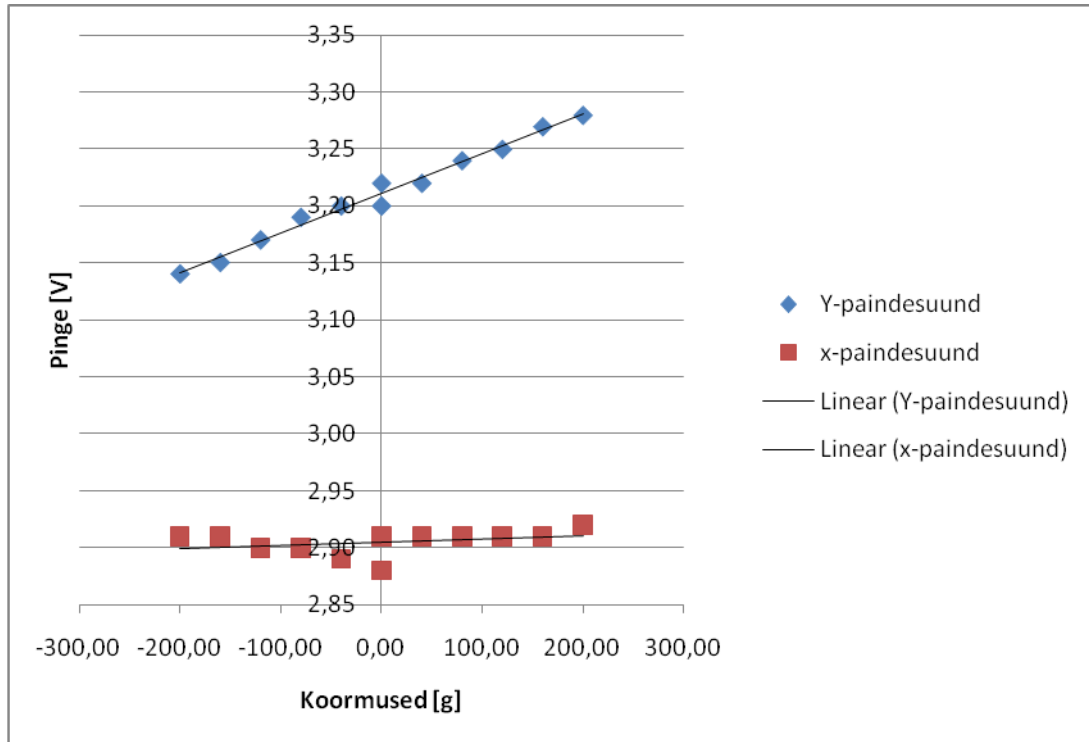
Joonisel 2.4 on esitatud x-telje paindesuuna graafik.



Joonis 2.4 x-telje paindesuund

Joonisel 2.4 (lk 20) võib järeldada, et kõrvalsuund y ei mõjuta oluliselt põhisuuna x -painesuuna väljundväärtusi. Võimendi väljundi keskpunkti väärtus x -painesuuna korral 0 voldi juures oli 2,9 V. (Painesuund x on esitatud joonisel 2.1 lk 16 – Paine x telje ümber).

Joonisel 2.5 on esitatud y -telje paindegraafik.



Joonis 2.5 y -telje paindesuund

Joonisel 2.5 lk 21 on võimalik järeldada, et kõrvalsuund x ei mõjuta oluliselt põhisuuna y paindesuuna väljundväärtusi. Võimendi väljundi keskpunkti väärtus y -painesuuna korral 0 voldi juures oli 3,20 V. (Painesuund y on esitatud joonisel 2.1 lk 16 – painde y telje ümber).

Joonistelt 2.4 ja 2.5 võib järeldada, et paindeandurid (takistusribade väärtus muutus lineaarselt) töötavad lineaarselt ja reageerivad erinevatele painde koormustele sõltumatult.

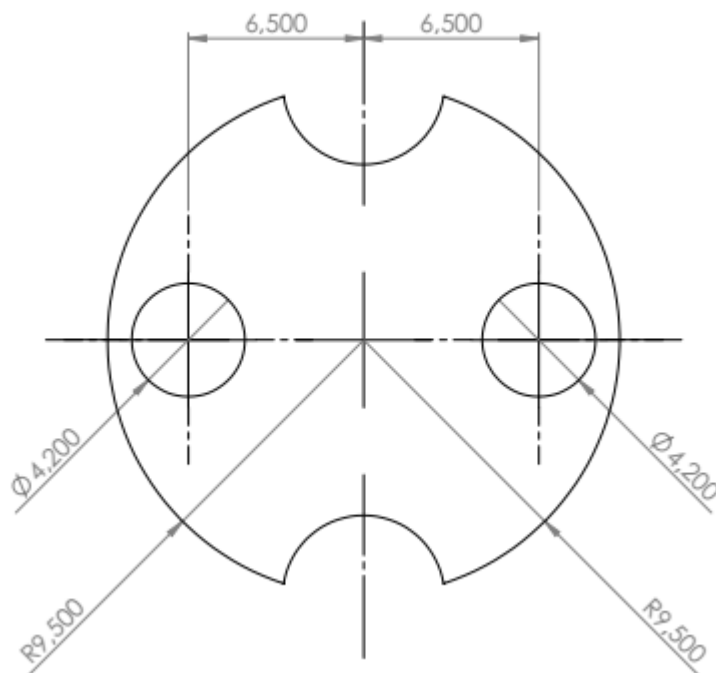
2.3.5 Jõuanduri võimenduse valimine

Jõuanduri võimenduse määramiseks disainiti ja valmistati katseplaat, mille peal olid järgmised komponendid: instrumentaalvõimendi INA333, virtuaalne maa TLE2426, kondensaator ja R_g takisti ja pistik jõuandurite väljundite jaoks. Katse käigus valmistati jõuandurile alusplaat (esitatud joonisel 2.1 lk 16).

Jõuanduri alusplaadi nõuded:

Jõuanduri alusplaat peab olema võimaikult väike ja mahtuma kinnitusplaadile.

Kinnitusplaadi andmed on esitatud joonisel 2.6 lk 22.



Joonis 2.6 Jõuanduri kinnitusplaat

Jõuanduri võimendutakisti arvutis:

kasutati valemit: $G = 1 + (100\text{k}\Omega/R_g)$

$$G = 1 + (100\text{k}\Omega/1,1 \text{ k}\Omega) = 100$$

Katse käigus selgus, et 100 kordne võimendus on sobilik ja võimendi väljundi kesk-väärtus tuli samuti sobilik 2,37 V.

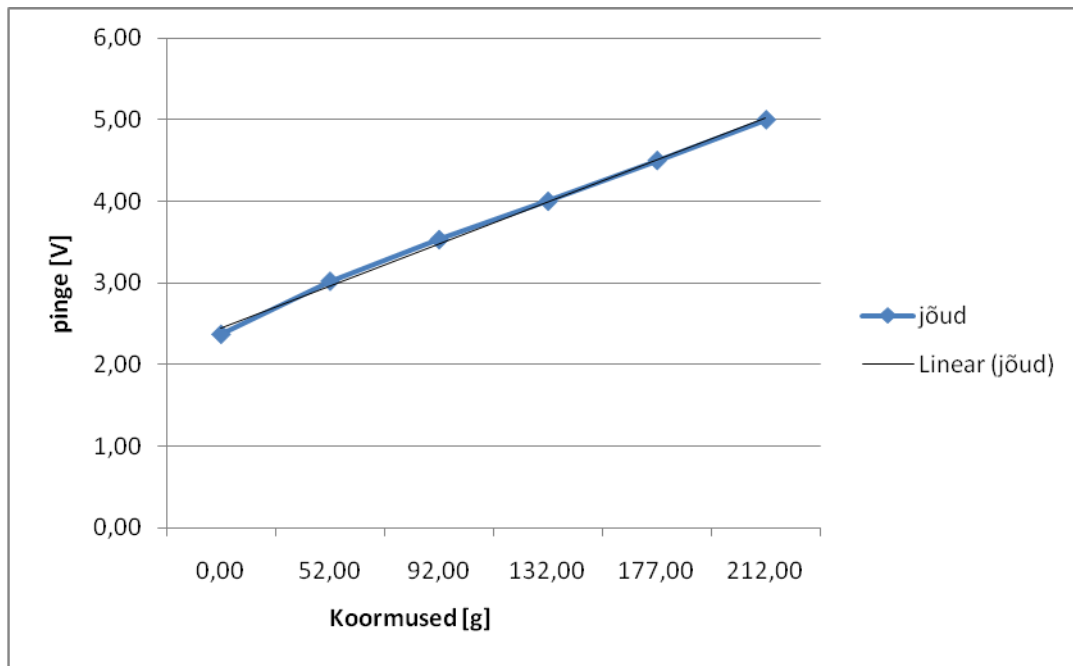
2.3.6 Jõuanduri töökatse

Jõuanduri töökatse käigus viidi läbi katse, milles koormati jõuandurit erinevate koormustega.

Koormustega katsed toimusid järgnevate raskustega:

Et esmakatsel ei olnud koormust, siis lisati 20 g kaupa koormusi kuni 212 grammini.

Joonisel 2.7 lk 23 on esitatud graafiliselt jõuandurile mõjuv jõud.



Joonis 2.7 Jõuandurile mõjuv jõud

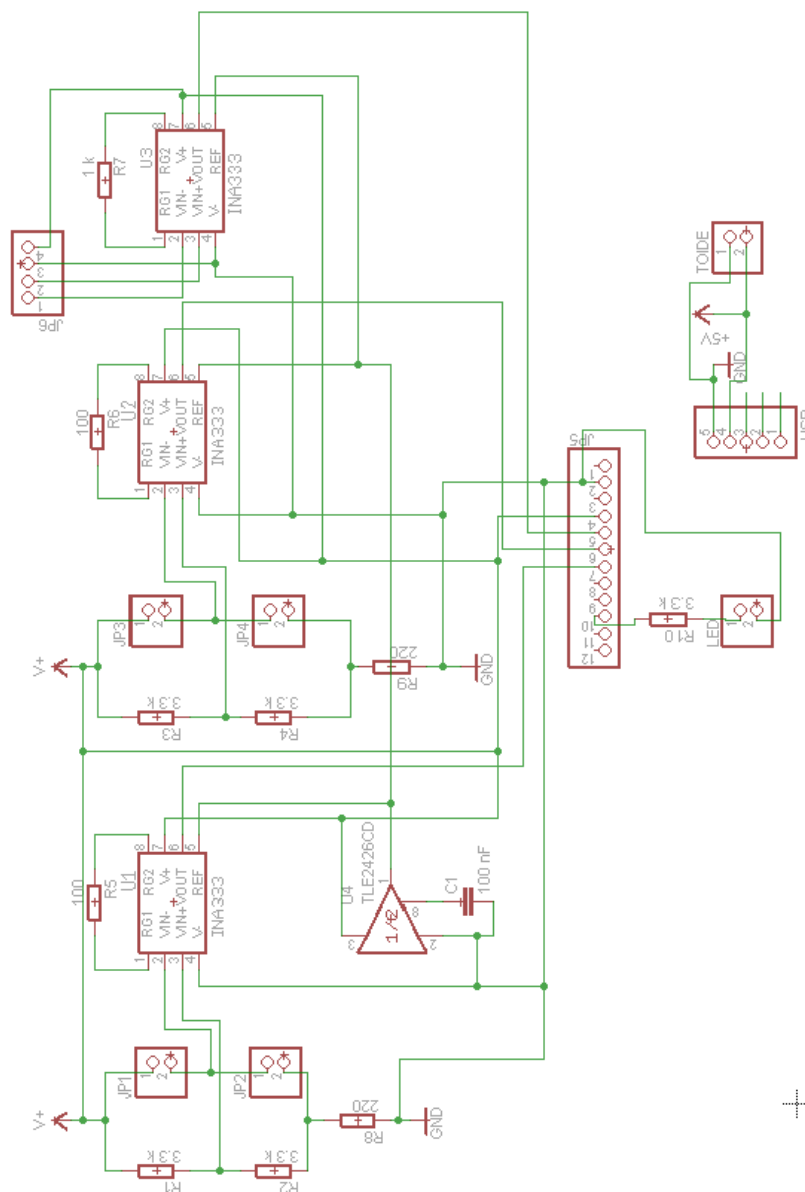
Jooniselt 2.7 võib järeldada, et jõuandur töötab linearselt ja reageerib erinevatele koormustele. Jõuanduri võimendi väljundi keskpunkti väärtus 0 voldi juures oli 2,37 V.

Jõuandurile mõjuva jõu suund on esitatud joonisel 2.1 lk 16 – F_{torge} .

3 Elektroonikasüsteemi disain

Bakalaureusetöö uurimuse raames oli vaja disainida adapterplaat, mis oma mõõtudel oleks võimalikult väike. Selleks valiti kõikideks kasutatavateks komponentideks SMD-komponendid. Elektriskeem disainiti programmis *Eagle 6.4.0 Light*.

3.1 Elektriskeem



Joonis 3.1 Elektriskeem

3.2 Elektriskeemi kirjeldus

Eagle 6.4.0 Light'is on elektriskeem disainitud põhimõttel, et trükiplaadi peale asetub *Arduino Mini*. JP5 on esitatud joonisel 3.1. *Arduino Mini* sisendid. U1 ja U2 on *Wheatstone*'i silla instrumentaalvõimendid ning JP1, JP2, JP3 ja JP4 on *Wheatstone*'i silla takistusribade ühenduspesad. U3 on jõuanduri instrumentaalvõimendi ja JP6 - jõuanduri väljundite ning sisendite pesa. Skeemi all paremas nurgas on näha *Arduino mini* USB pesa koos toite pesaga programmeerimise jaoks.

3.2.1 Instrumentaalvõimendi ja virtuaalne maa kirjeldus

Instrumentaalvõimendi INA333 on madala võimsusega täpsus mõõteriist. Väike suurus ja madal võimsus võimaldab INA333 laia kasutusala. Võimendusteguri suurus on ühest tuhandeni. INA333 võimenduse standardvalem: $G = 1 + (100k\Omega/Rg)$. Operatsioon toimib väga madalatel toitepingetel. Töövahemik on 1,8 V kuni 5 V-ni.

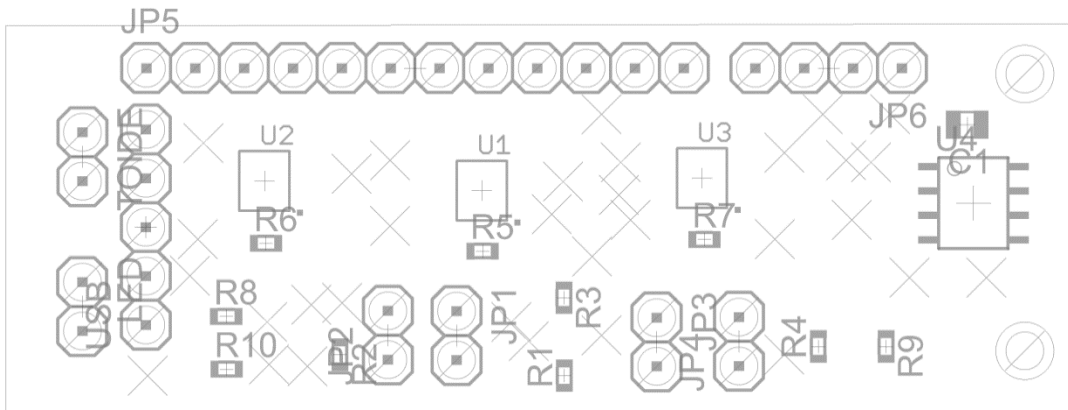
Põhjendus, miks kasutati INA333-e: et toitepinge ei ole üle 5 voldi, siis instrumentaalvõimendi INA333 on sobiv käesolevass töös, et jätta kogu toitepinge samaks (5 volti).

Virtuaalne maa TLE2426 väljundpinge: alati kaheks jagatud sisendpingega. TLE2426 annab madala impedantsiga väljundeid. Sisendpinge väärtused on 4V kuni 40V-ni.: TLE2426 kasutatakse selleks, et INA333 referentspinget alandada poole võrra ning et 0 voldise diferentsiaal sisendpinge puhul võimendi väljundi keskvärtus oleks ligilähedal 2,5 voldile.

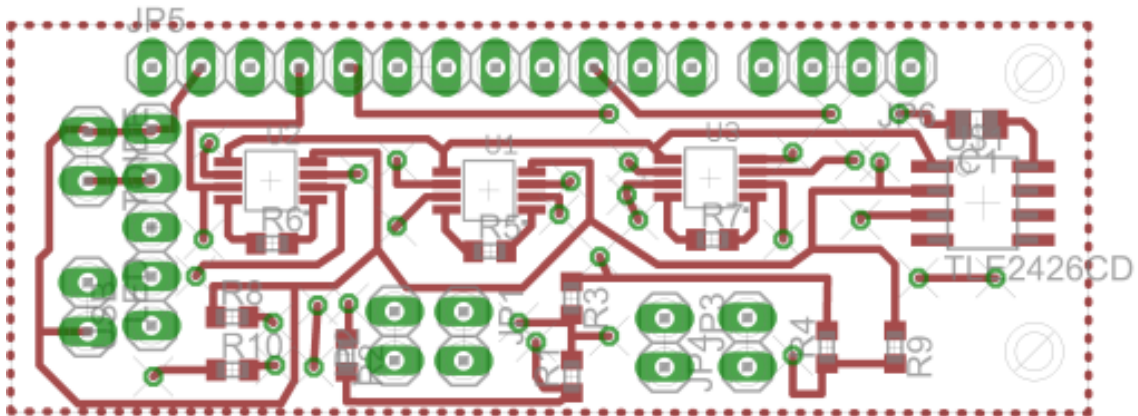
3.3 Trükkplaat

Trükkplaadi mõõtmed pidid olema võimalikult väikesed, et mahuks süstla korpuse sisse. Mõõtmed: 56,0 mm x 20,0 mm, paksus 1,6 mm. Joonisel 3.5 lk 27 on võimalik näha *Arduino Mini* asukohta. JP5 pikiriba on *Arduino Miniga* ühendamiseks (joonis 3.6 lk 28 illustreerib paremini *Arduino Mini* asukohta, *Arduino Mini* ühendatakse pikiribade peale). Trükkplaat on disainitud tarkvara *Eagle 6.4.0 Light* kasutades.

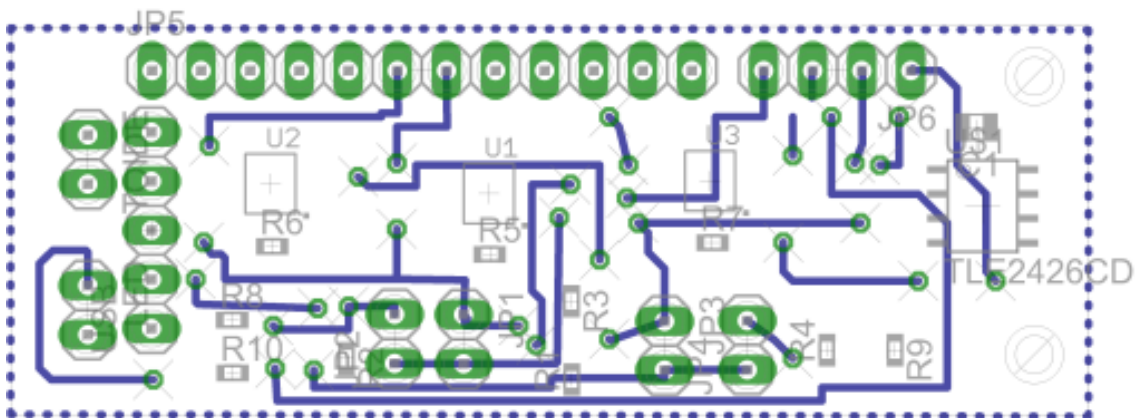
3.3.1 Trükkplaadi disain



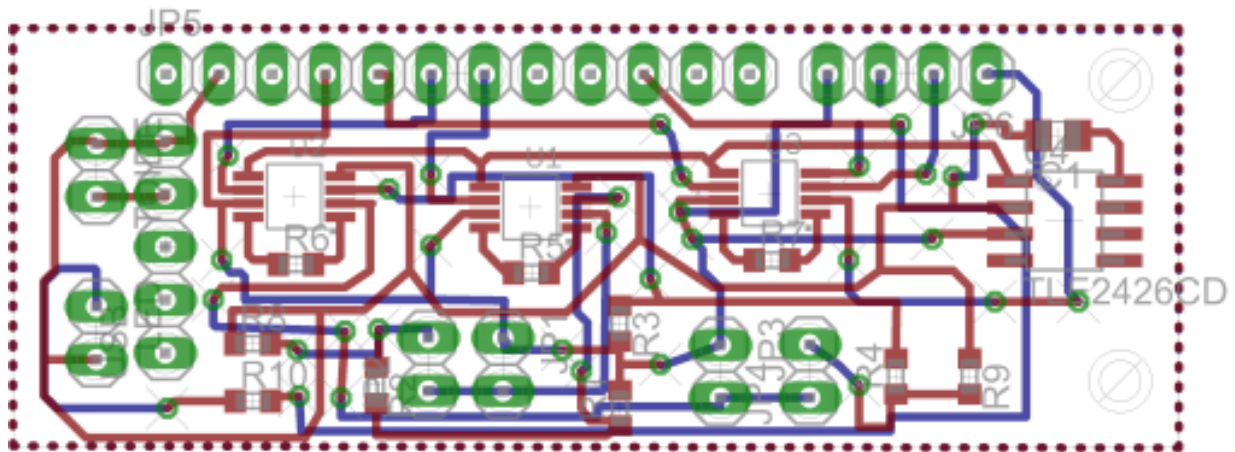
Joonis 3.2 Trükkplaadi komponentide paigutus



Joonis 3.3 Trükkplaadi ülemine kiht



Joonis 3.4 Trükkplaadi alumine kiht



Joonis 3.5 Trükkplaat

Trükkplaadi 3D mudeli loomiseks on kasutatud *SketchUp Make* 2014 programmi. Joonisel 3.6 lk 28 on võimalik näha trükkplaadi 3D mudelit.

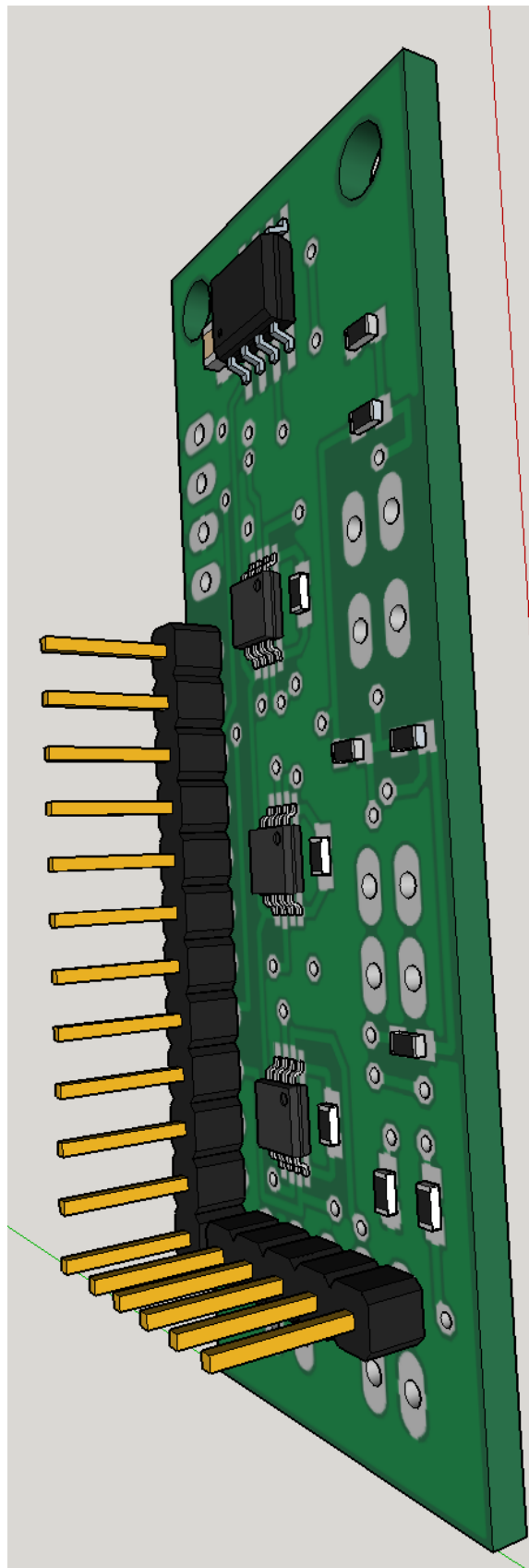
3.3.2 Komponentid

Trükkplaadil kasutatud komponentide nimekiri on esitatud tabelis 1.1.

Tabel 1.1 Trükkplaadi komponendid

Kogus	Väärtus	Seade	Korpus	Osad	Kirjeldus	Tootja
1		22-03-2121	2,54mm, 12pin	JP5	Piikriba	Molex Electronics
5		CES-102-01-T-S	2,54mm, 2pin	JP1, JP2, JP3, JP4, TOIDE	Piikriba	Sametec
1		CES-104-01-T-S	2,54mm, 4pin	JP6	Piikriba	Samtec
1		215297-5	2,54mm 5pin	USB	Piikriba	TE Connectivity
2	220	SMD-takisti	0603	R8,R9	Takisti	Bourns Electronics
1	1k	SMD-takisti	0603	R7	Takisti	Bourns Electronics
2	100	SMD-takisti	0603	R5,R6	Takisti	Bourns Electronics
1	100nF	SMD-kondensaator	0805	C1	Kondensaator	Kemet Electronics
1	red	L-934ID	3mm,red	LED	valgusdiod	Kingbright
5	3.3k, 1%	SMD-takisti	0603	R1,R2,R3,R4	Takisti	Bourns Electronics
3	INA333	INA333AIDGKT	MSOP8	U1,U2,U3	instrumentaalvõimendi	Texas Instruments
1	TLE2326CD	TLE2426CD	SO-8	U4	Virtuaalne maa (Texas Instruments "rail splitter" with noise-reduction)	Texas Instruments

3.4 Trükkplaadi 3D mudel



Joonis 3.6 Trükkplaadi 3D mudel

4 Programmeerimine

4.1 Programmeerimise ülesannete püstitus

Mikrokontrolleri programmi ülesandeks on anda tagasisidet residendile töö käigus ning edastada andurite väärtusi läbi COM pordi arvutisse, et käelisi töömustreid saaks kasutada töö efektiivsuse hindamisel. Loo algoritmi, mis annab teada ohtlikest olukordadest (liigne painde, jõud, pöördenurk, kiirendus).

4.2 Tehnilised nõuded

Arendusplatvorm peab mahtuma „targa” nõela süstla korpusesse.

4.3 Ülesannete lahendus

Kõigepealt kirjutati koodid katseplaatide testimiseks: painde- ja jõuanduri plaadid. Joonised 2.4, 2.5 ja 2.7 kirjeldavad nende andurite tööd.

Katseplaatide katsetamiseks kasutati Arduino Uno plaati. Arduino Uno kasutab mikrokontrollerit Atmega 328. Koodi kirjutamiseks on kasutatud *Arduino 1.5.0* programmi.

Programmijupi põhiosa - pöördenurkade, kiirenduste, painde- ja jõuandurite tulemuste väljastamine:

```
void loop() {  
  
  // väljastatakse pöördenurgad  
  Serial.print("ypr\t");  
  Serial.print(ypr[0] * 180/M_PI);  
  Serial.print("\t");  
  Serial.print(ypr[1] * 180/M_PI);  
  Serial.print("\t");  
  Serial.print(ypr[2] * 180/M_PI);  
  Serial.print("\t");  
  
  // väljastatakse kiirendused  
  Serial.print("areal\t");  
  Serial.print(aaReal.x);  
  Serial.print("\t");  
  Serial.print(aaReal.y);  
  Serial.print("\t");  
  Serial.print(aaReal.z);  
  Serial.print("\t");  
}
```

```

// loetakse paindejõudusid
sensorValue = analogRead(A2);
voltage = sensorValue * (5.0 / 1023.0);
sensorValue2 = analogRead(A1);
voltage2 = sensorValue2 * (5.0 / 1023.0);
//väljastatakse paindejõud
Serial.print(voltage);
Serial.print("\t");
Serial.print(voltage2);
Serial.print("\t");
//loetakse jõuanduri väärtusi
sensorValue = analogRead(A3);
voltage = sensorValue * (5.0 / 1023.0);
//väljastatakse jõuanduri väärtused
Serial.print(voltage);
Serial.print("\n");
} [12]

```

Kogu programmi kood on esitatud lisa 1 lk 37.

Lõplikku algoritmi, mis teavitaks ohuolukordades, ei jõutud lõpuni teha. Koodi arendatakse lähitulevikus edasi.

5 Elektroonikasüsteemi maksumus

Elektroonikasüsteemi komponentide maksumus on esitatud tabelis 5.1.

Tabel 5.1 Elektroonikasüsteemi komponentide hinnad

Kogus	Seade	Hind	Kogu hind
1	PIN HEADER, 12 pin	1,05	1,05
5	PIN HEADER, 2 pin	0,284	1,42
1	PIN HEADER, 4pin	0,54	0,54
1	PIN HEADER, 5 pin	0,671	0,671
2	SMD-takisti, 220	0,01	0,02
1	SMD-takisti, 1k	0,01	0,01
2	SMD-takisti, 100	0,01	0,02
1	SMD-kondensaator	0,016	0,016
1	LED	0,134	0,134
5	SMD-takisti, 3.3k 1%	0,02	0,1
3	INA333	4,45	13,35
1	TLE2426	1,45	1,45
1	PCB 56mm x 20 mm	22,12	22,12
1	Arduino mini 33mm x 18 mm	17,50	17,50
1	FSS1500NSB Force sensor	56,0	56,0
4	Strain gauge	7,80	31,20
1	IMU 6DOF – MPU6050	10,90	10,90

Trükkplaat telliti Kamitrast - hinnale lisandus töö alustamistasu 29 eurot.

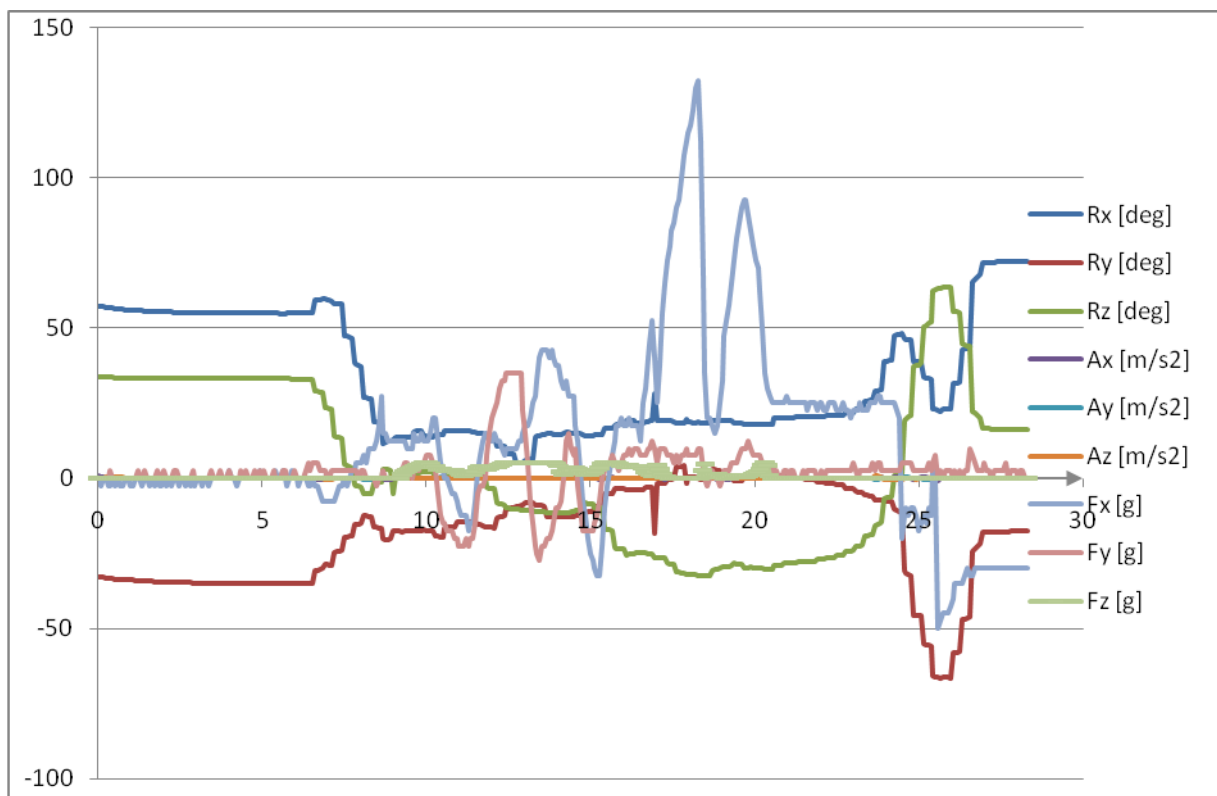
Ühe elektroonikasüsteemi maksumus kokku: 184,45 eurot.

6 Saadud tulemused

Valminud elektroonikasüsteemiga viidi läbi katse, et näha kuidas paindeandur, jõuandur, kiirendusandur ja pöördeandur koos töötavad.

Katse käigus torgati „targa“ nõela nõel neerufantoomi ja väljastati andurite tulemused *Arduino Mini* abil arvuti COM porti.

Joonisel 6.1 on esitatud katse tulemused.



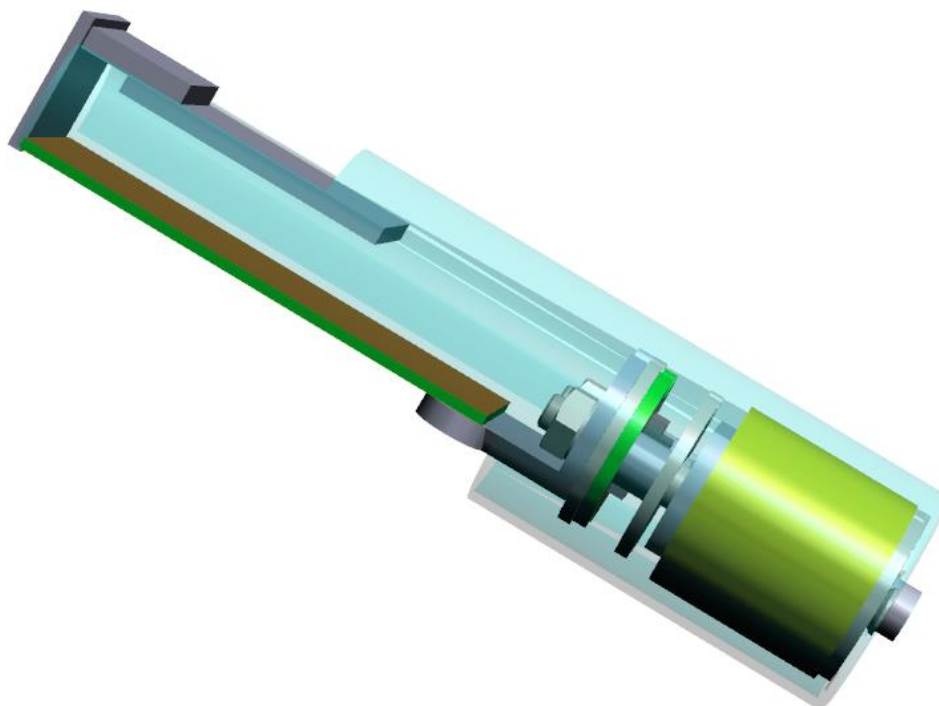
Joonis 6.1 Andurite töö tulemused

Jooniselt 6.1 võib järeldada, et kõik andurid töötasid nõuetekohaselt. Jooniselt 6.1 on näha, et paindejõu x suund muutub kõige rohkem, kiirendus jääb võrdseks peaaegu kõikides suundades. Tehtud katse on toodud näiteks, kuidas andurite andmed võivad muutuda, iga konkreetne katse tuleb erinev.

Horisontaalteljel on aeg (s). Rx, Ry, Rz tähistavad pöördenurga momenti. Ax, Ay, Az tähistavad kiirendusi, Fx - painet x telje ümber, Fy - painet y telje ümber, Fz - torke jõudu (joonisel 2.1 lk 16 on esitatud pöördenurkade, kiirenduste, painete ja jõu suunad).

6.1 Elektroonikasüsteemi mehaanikadisain

Joonisel 6.2 on esitatud töö käigus disainitud ja valmistatud elektroonikasüsteemile loodud „targa“ nõela süstla korpuse mehaanikamudel.



Joonis 6.2 Elektroonikasüsteemi mehaanikamudel

Alumine plaat (rohekat värvi) on töö käigus disainitud ja valmistatud trükkplaat. Trükkplaadi peal asub *Arduino Mini* plaat (suurem, hallikas plaat). Pöörde- ja kiirendusandur asub *Arduino Mini* plaadi peal (väiksem, hallikas plaat).

Kokkuvõte

Bakalaureusetöö eesmärk oli disainida ja valmistada jälgimiselektronika „targale“ nõelale.

Bakalaureusetöö esimeses osas uuriti erinevaid menetlusradioloogia objektiivseid hindamissüsteeme. Lühidalt on kirjeldatud biorobotika fantoome ja käsitletud töö teema aktuaalsuse küsimusi.

Bakalaureusetöö teises osas kirjeldatakse elektroonikasüsteemi ülesannete täitmise nõudeid ning põhjendatakse konkreetsete andurite valikut. Bakalaureusetöö käigus viidi läbi katsed, et leida sobivad võimendustegurid painde- ja jõuandurile. Lisaks on esitatud arvutusliku meetodiga võimendustegurite arvutused.

Bakalaureusetöö kolmandas osas on kirjeldatud elektroonikasüsteemi disaini, esitatud elektrikeem, trükkplaat, trükkplaadi 3D mudel, komponendid.

Bakalaureusetöö neljandas osas on kirjeldatud programmeerimise ülesannete püstitused ja loodud programmi kood, mis väljastab painde- ja jõuanduri väärtused, kiirendused, pöördenurgad.

Bakalaureusetöö viiendas osas kirjeldatakse elektroonikasüsteemi maksumust.

Bakalaureusetöö kuuendas osas tutvustatakse valmis elektroonikasüsteemiga läbi viidud katset, millest võib järeldada, et elektroonikasüsteemi andurid reageerivad erinevatele jõududele, kiirendusmomentidele ja pöördenurkadele.

Kokkuvõtteks võib bakalaureusetöö eesmärgid täidetuks tunnistada. Erinevaid katseid ja arvutusi sooritades jõuti töötava elektronika disainini ning loodi valmis elektroonikasüsteem. Edasiseks teadus- ja arendustöös tuleks luua algoritm, mis tuvastaks ohuolukorrad.

Summary

The goal of the bachelor's thesis was to design and create electronics for smart needle.

In second part, electronics assignment requirements are described and the particular choices of the sensors are explained. Tests are carried out to find suitable amplification factors for bending and force transducers.

In the third part, electronics design, electric circuit, printed circuit board, the 3D model of the printed circuit board, components are described.

In the fourth part, the programming tasks are described and a code is created, which gathers the deflection and force transducer values, accelerations and rotation angles.

In the fifth part, the cost of the electronic system is described.

In the sixth part a test with completed electronic system was carried out. From the test it can be concluded that the electronic system sensors responded to different forces, acceleration torques and angle of rotations.

In conclusion the bachelor's thesis objectives were met. The different test and calculations which were carried out proved that the electronics design worked and complete electronics system was created. However, there is room for further development to create algorithm which detects dangerous situations.

Kasutatud kirjandus

1. Lucas, B. G. B., Witcher, H.W. (1958). Artificial Respiration Comparison Between Manual and Intermittent Positive-Pressure Methods. - *British Medical Journal*, 887-889.
2. Denson, J.S., Abrahamson, S. A Computer – Controlled Patient Simulator. [WWW]
<http://jama.jamanetwork.com/article.aspx?articleid=345227>, (03.06.2014)
3. Chipma, J. G., Shmitz, C. C. (2009). Using Objective Structured Assessment of Technical Skills to Evaluate a Basic Skills Simulation Curriculum for First-Year Surgical Residents. – *Elsevier Inc.* 364-370.
4. Kamran, A., Keeling, A. N, Fakhry, M. (2010). Role of Virtual Reality Simulation in Teaching and Assessing Technical Skills Endovascular Intervention. 55-66.
5. Bertalan Mesko. Synthetic Human Cadavers in Medical Education. [WWW]
<http://sciencereview.com/2013/11/09/synthetic-human-cadavers-in-medical-education/>, (03.06.2014)
6. CIRS Kidney Training Phantom. InMed. [WWW]
<http://www.inmed.com.au/products/phantoms-test-objects/ultrasound/cirs-kidney-training-phantom/>, (03.06.2014)
7. Gould, D. (2010). Using simulation for interventional radiology training. – *The British of Radiology*, 546-553.
8. Bello, F., Bulpitt, D., Gould, A. (2009). ImaGINE-S Imaging Guided Interventional Needle Simulation. – *Proc. Eurographics*.
9. Magee, D., Kessel, D. (2005). A Computer Based Simulator for Ultrasound Guided Needle Insertion Procedures.
10. Ristolainen, A., Colucci, G., Kruusmaa, M. (2013). A Phantom Pig Abdomen as an Alternative for Testing Robotic Surgical Systems: Our Experience. - *Center for Biorobotics, Faculty of Information Technology, Tallinn University of Technology*, 359-367.
11. Transducer Techniques, Half-Bridge Type II. [WWW]
<http://www.transducertechniques.com/wheatstone-bridge.aspx> , (20.03.2014)
12. J. Rowberg. GitHub. [WWW]
<https://github.com/jrowberg/i2cdevlib>, (20.05.2014)

LISAD

Lisa 1

```
// I2C device class (I2Cdev) demonstration Arduino sketch for MPU6050 class using DMP
(MotionApps v2.0)
// 6/21/2012 by Jeff Rowberg <jeff@rowberg.net>
// Updates should (hopefully) always be available at https://github.com/jrowberg/i2cdevlib

#include "I2Cdev.h"
#include <math.h>
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

MPU6050 mpu(0x69); // <-- use for AD0 high

#define OUTPUT_READABLE_YAWPITCHROLL

#define LED_PIN 12 // (Arduino is 13, Teensy is 11, Teensy++ is 6)
bool blinkState = false;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorInt16 aa; // [x, y, z] accel sensor measurements
VectorInt16 aaReal; // [x, y, z] gravity-free accel sensor measurements
VectorInt16 aaWorld; // [x, y, z] world-frame accel sensor measurements
VectorFloat gravity; // [x, y, z] gravity vector
float euler[3]; // [psi, theta, phi] Euler angle container
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

// packet structure for InvenSense teapot demo
uint8_t teapotPacket[14] = { '$', 0x02, 0,0, 0,0, 0,0, 0,0, 0x00, 0x00, '\r', '\n' };

volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high
void dmpDataReady() {
    mpuInterrupt = true;
}

void setup() {
    // join I2C bus (I2Cdev library doesn't do this automatically)
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
        TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz)
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif

    pinMode (A2, INPUT); // paindeanduer
    pinMode (A1, INPUT);
    pinMode (A3, INPUT);

    Serial.begin(9600);
    while (!Serial); // wait for Leonardo enumeration, others continue immediately

    Serial.println(F("Initializing I2C devices..."));
    mpu.initialize();

    // verify connection
    Serial.println(F("Testing device connections..."));
    Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050
connection failed"));
}
```

```

// wait for ready
Serial.println(F("\nSend any character to begin DMP programming and demo: "));
while (Serial.available() && Serial.read()); // empty buffer
while (!Serial.available()); // wait for data
while (Serial.available() && Serial.read()); // empty buffer again

// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(28);
mpu.setYGyroOffset(7);
mpu.setZGyroOffset(19);
mpu.setXAccelOffset(-888);
mpu.setYAccelOffset(2506);
mpu.setZAccelOffset(1650); // 1688 factory default for my test chip

// make sure it worked (returns 0 if so)
if (devStatus == 0) {
    // turn on the DMP, now that it's ready
    Serial.println(F("Enabling DMP..."));
    mpu.setDMPEnabled(true);

    // enable Arduino interrupt detection
    Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
    attachInterrupt(0, dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();

    // set our DMP Ready flag so the main loop() function knows it's okay to use it
    Serial.println(F("DMP ready! Waiting for first interrupt..."));
    dmpReady = true;

    // get expected DMP packet size for later comparison
    packetSize = mpu.dmpGetFIFOPageSize();
} else {
    // ERROR!
    // 1 = initial memory load failed
    // 2 = DMP configuration updates failed
    // (if it's going to break, usually the code will be 1)
    Serial.print(F("DMP Initialization failed (code "));
    Serial.print(devStatus);
    Serial.println(F(")"));
}

// configure LED for output
pinMode(LED_PIN, OUTPUT);
}

void loop() {

    int sensorValue;
    int sensorValue2;
    float voltage;
    float voltage2;

    if (!dmpReady) return;

    // wait for MPU interrupt or extra packet(s) available
    while (!mpuInterrupt && fifoCount < packetSize) {
        // other program behavior stuff here
        // .
        // .
        // .
        // if you are really paranoid you can frequently test in between other
        // stuff to see if mpuInterrupt is true, and if so, "break;" from the
        // while() loop to immediately process the MPU data
        // .
        // .
        // .
    }

    // reset interrupt flag and get INT_STATUS byte
    mpuInterrupt = false;
    mpuIntStatus = mpu.getIntStatus();

    // get current FIFO count
    fifoCount = mpu.getFIFOCount();
}

```

```

// check for overflow (this should never happen unless our code is too inefficient)
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    // reset so we can continue cleanly
    mpu.resetFIFO();
    // Serial.println(F("FIFO overflow!"));

// otherwise, check for DMP data ready interrupt (this should happen frequently)
} else if (mpuIntStatus & 0x02) {
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    fifoCount -= packetSize;

#ifdef OUTPUT_READABLE_QUATERNION
    // display quaternion values in easy matrix form: w x y z
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    Serial.print("quat\t");
    Serial.print(q.w);
    Serial.print("\t");
    Serial.print(q.x);
    Serial.print("\t");
    Serial.print(q.y);
    Serial.print("\t");
    Serial.println(q.z);
#endif

#ifdef OUTPUT_READABLE_EULER
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetEuler(euler, &q);
    Serial.print("euler\t");
    Serial.print(euler[0] * 180/M_PI);
    Serial.print("\t");
    Serial.print(euler[1] * 180/M_PI);
    Serial.print("\t");
    Serial.println(euler[2] * 180/M_PI);
#endif

#ifdef OUTPUT_READABLE_YAWPITCHROLL
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    mpu.dmpGetQuaternion(&q, fifoBuffer);

    mpu.dmpGetAccel(&aa, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);

    //ComplementaryFilter((aaReal.x; aaReal.y; aaReal.z), ypr, 0, 0);

    Serial.print("ypr\t");
    Serial.print(ypr[0] * 180/M_PI);
    Serial.print("\t");
    Serial.print(ypr[1] * 180/M_PI);
    Serial.print("\t");
    Serial.print(ypr[2] * 180/M_PI);
    Serial.print("\t");

    Serial.print("areal\t");
    Serial.print(aaReal.x);
    Serial.print("\t");
    Serial.print(aaReal.y);
    Serial.print("\t");
    Serial.print(aaReal.z);
    Serial.print("\t");
    sensorValue = analogRead(A2);
    voltage = sensorValue * (5.0 / 1023.0);
    sensorValue2 = analogRead(A1);
    voltage2 = sensorValue2 * (5.0 / 1023.0);

    Serial.print(voltage);
    Serial.print("\t");
    Serial.print(voltage2);
    Serial.print("\t");

```

```

    sensorValue = analogRead(A3); //jōu
    voltage = sensorValue * (5.0 / 1023.0);

Serial.print(voltage);
Serial.print("\n");
#endif

#ifdef OUTPUT_READABLE_REALACCEL
    // display real acceleration, adjusted to remove gravity
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetAccel(&aa, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
    Serial.print("areal\t");
    Serial.print(aaReal.x);
    Serial.print("\t");
    Serial.print(aaReal.y);
    Serial.print("\t");
    Serial.println(aaReal.z);
#endif

#ifdef OUTPUT_READABLE_WORLDACCEL
    // display initial world-frame acceleration, adjusted to remove gravity
    // and rotated based on known orientation from quaternion
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetAccel(&aa, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
    mpu.dmpGetLinearAccelInWorld(&aaWorld, &aaReal, &q);
    Serial.print("aworld\t");
    Serial.print(aaWorld.x);
    Serial.print("\t");
    Serial.print(aaWorld.y);
    Serial.print("\t");
    Serial.println(aaWorld.z);
#endif

#ifdef OUTPUT_TEAPOT
    // display quaternion values in InvenSense Teapot demo format:
    teapotPacket[2] = fifoBuffer[0];
    teapotPacket[3] = fifoBuffer[1];
    teapotPacket[4] = fifoBuffer[4];
    teapotPacket[5] = fifoBuffer[5];
    teapotPacket[6] = fifoBuffer[8];
    teapotPacket[7] = fifoBuffer[9];
    teapotPacket[8] = fifoBuffer[12];
    teapotPacket[9] = fifoBuffer[13];
    Serial.write(teapotPacket, 14);
    teapotPacket[11]++; // packetCount, loops at 0xFF on purpose
#endif

    // blink LED to indicate activity
    blinkState = !blinkState;
    digitalWrite(LED_PIN, blinkState);
    delay(10);
}
}

```