



TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND

ELEKTROENERGEETIKA JA MEHHATROONIKA
INSTITUUT

INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMI
PARANDAMINE JA PROGRAMMEERIMINE.

IMPROVEMENT AND PROGRAMMING OF A HUMAN ARM SIMULATING
SYSTEM.

BAKALAUREUSETÖÖ

Üliõpilane: Aleksandr Kotšetov

Üliõpilaskood: 164609MAHB

Juhendaja: Lauri Kütt

Tallinn 2019

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” 201.....

Autor:

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“.....” 201.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”.....201... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Elektroenergeetika ja mehhatroonika instituut

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Aleksandr Kotšetov, 164609MAHB
Õppekava, peeriala: MAHB02/13 - Mehhatroonika
Juhendaja(d): programmijuht, professor Lauri Kütt, Telefon: 620 3806
Konsultandid:

Lõputöö teema:

(eesti keeles) Inimkäät järgiva robotkäe-süsteemi parandamine ja programmeerimine.

(inglise keeles) Improvement and programming of a human arm simulating system.

Lõputöö põhieesmärgid:

1. Uurida olemasolevat robotkäe komplekti ja luua programm selle juhtimiseks
2. Uurida robotkäe saatja/inimkäe järgija mehaanilist ülesehitust ja kujundada parandatud sensorikoost
3. Luua programm robotkäe manipuleerimiseks ja inimkäe liikumise ülekandmiseks täiturkomplektil

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Roboti mehaaniliste osade kokkupanek ja luua programm selle juhtimiseks	10.03
2.	Mehaaniliste osade uurimine ja puuduste parandamine nende olemasolul	24.03
3.	Programmi koostamine robotkäe juhtimiseks	21.04
4.	Lõputöö kirjutamine ja vormistamise lõpetamine	12.05

Töö keel: Eesti kee

Lõputöö esitamise tähtaeg: " 21" mai 2019a

Üliõpilane: Aleksandr Kotšetov "1" märts 2019a

/allkiri/

Juhendaja: Lauri Kütt "1 " märts 2019a

/allkir

SISUKORD

EESSÕNA	6
SISSEJUHATUS	7
1 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMIT.....	8
1.1 Juhtmevaba mehaanilise kinda komplekti ülevaade	8
1.2 Lisavahendite ülevaade	9
1.2.1 Bluetooth	9
1.2.2 Güroskoop - akseleromeeter	10
1.2.3 Potentisomeetrid.....	11
1.2.4 Sensorplaadi skeem.....	12
1.3 Olemasoleva robotkäte komplekti ülevaade	12
1.3.1 Bluetooth	14
1.4 Mootorid ülevaade.....	15
2 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMI PARANDAMINE.....	16
2.1 Robotkäte saatja/inimese järgija mehaanilise ülesehituse parandamine	16
2.1 Mehaaniline juhtmevaba kinda parandamine	16
2.2 Kinemaatika arvutamine	18
2.3 Manipulaatori mehaaniline käsi parandamine	21
3 JUHT PROGRAMMI KUJUNDAMINE ROBOTKÄE MANIPULEERIMISEKS JA INIMKÄE LIIKUMISE ÜLEKANDMISEKS TÄITURKOMPLETEKILE	26
3.1 Esimene programm mehaanilise kinda juhtmevaba juhtimise jaoks	26
3.2 Teine programm manipulaatori mehaaniline käe jaoks.....	30
3.3 Kasutusjuhend	32
4 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMI EDASIARENDUSED	34
4.1 Mehaanilise käe manipulaatori edasiarendamine	34
4.1.1 Mikrokontroller	34
4.1.2 Andmevahetussüsteem	34
4.1.3 Andurid.....	35
4.2 Juhtmevaba mehaanilise kinda edasine arendamine	36
KOKKUVÕTE	37
SUMMARY	39
KASUTATUD KIRJANDUSE LOETELU	41

LISA 1 Poldid ja erinevad detailideks kinnitused	45
LISA 2 Arduino Uno juhtplaadi skemaatiline diagramm ja kinnastega juhtpaneeli skemaatiline diagramm	46
LISA 3 Mikrokontrolleri andurite ühendusskeem	48
LISA 4 Bluetooth mooduli programm	49
LISA 6 Programm andmete saamiseks	50
LISA 7 Kalmani filtrit programm	52
LISA 8 Programm andmete saatmiseks	53
LISA 9 Matlab programm kinemaatika jaoks	54

EESSÕNA

Töö teema kerkis üles, kuna Hiinast saabunuds mehaanilise käe manipulaator ja juhtmevaba andmeedastusega juhtimiskinnas saadeti ilma tarkvara. Töö oli välja pakutud Tallinna Tehnikaülikooli poolt. Töö esimeseks etapiks oli uurida mehaanilise käe manipulaatori olemasolevat komplekti ja käe juhtmise mehhanismi. Lisaks tuli praktiliselt nullist luua juhtprogramm robotkäe komplekti juhtimiseks. Teine etapp oli komplekti mehaaniliste omaduste uurimine ja nende parandamine. Kolmas etapp on programmi loomine mehaanilise käe manipulaatori juhtimiseks juhtmevaba juhtkindaga. Lõputöö teostus ja baasandmete kogumine viidi läbi Tallinna Tehnikaülikooli Elektroenergeetika ja mehhatroonika instituut Instituudis.

Tallinna Tehnikaülikooli professor Lauri Kütt oli lõputöö juhendajaks ja töö valmis tema juhtimise all. Tahaksin teda tänada järeleaitajat hea juhendamise ja hea töökeskkonna kasutamise eest.

SISSEJUHATUS

Tallinna Tehnikaülikoolis pakkus selle töö juhendaja võimalust uurida, luua tarkvara ja paremaks teha juhtvaba mehaaniline kinnas ja mehaanilise käe manipulaator. Hiinast tellitud käe-sensoril ja mehaanilisel käemanipulaatoril puudus tarkvara ja juhtplokid. Tallinna Tehnikaülikooli saabusid ainult detailid komplekti jaoks.

Töö alguses kirjeldatakse esimest tutvumist nende komponentidega ja tegevusi, mida nendega edasi tuleb sooritada. Kõigi komplektis olevate andurite ja detaile kirjeldus, üldvaate üldplaneering ja esmakordne kokkupanek ilma juhendita. Uuritava süsteemi, koos operaatori sisendiga võib nimetada on küberfüüsiliseks süsteemiks. Töö praktilises osas kirjeldatakse nende mehhanismide parandamist, uute detailide loomist ja detailide kinemaatika arvutamist. Samuti kahe koodi kirjeldamine, mis on üleslaetud kahele mikrokontrollerile mehaanilise käe manipulaatori juhtimiseks. Viimases peatükis arutletakse olulisi komponente, mida tuleb veel välja arendada, või kuidas olemasolevaid komponente parandada, tagamaks nende stabiilsem ja täpsem töö.

1 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMIT

1.1 Juhtmevaba mehaanilise kinda komplekti ülevaade

Enne antud lõputöö teemaga alustamist oli teada, et Hiinast tellitud juhtmevaba mehaanilise kinda ja mehaaniline käe manipuleerija komplekt olid tarkvarata, juhtimisseadmeteta ja kokkupaneku juhendita. Sellest olid määratud ka esmased ülesanded, et uurida robotiseeritud käesüsteemi olemasolevat komplekti. Eesmärgiga parandada detailide puudused.

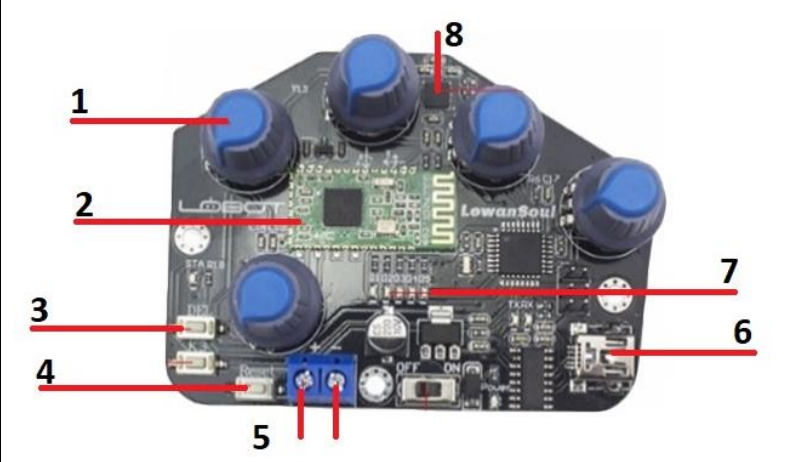
Esimene minu poolt uuritud komplekt koosnes mehaanilisest kindast juhtmevaba andmevahetusega, mida nimetatakse Lewansoulikks.

Mehaanilise juhtmevaba kinda parameetrid: [1]

- 1) Mikrokontrolleri mudel: Atmega 328P (Arduino Uno ühilduv)
- 2) Aku: 7,4V 2tk Liitiumpolümeerakud (sisseehitatud)
- 3) Mõõtmed: 147mm x 113mm
- 4) Kaal: 0.26kg

See kontrolleri ei näe välja nagu tavaline Arduino Uno, kuna kinda arendajad tegid selle toote jaoks spetsiaalselt uue plaadi (vt. Joonis 1.1). See sisaldab endas lisaks standardsetele Arduino Uno liidestele ka uusi lisaseandmeid (vt. Table 1.1). [1]

Table 1.1 Sensorite trükkplaat robotkäte juhtimiseks [1]

	<ol style="list-style-type: none">1) Potentsiomeetrid2) Bluetooth moodul3) Kaks funktsionaalset nuppu4) Reset nupp5) Power6) USB-liides7) LED indikaatorid8) Güroskoopiline kiirendusandur
--	---

Joonis 1.1 Sensorite trükkplaat robotkäte juhtimiseks [1].

Sensoriplaadile omased lisaseadmed: [1]

- 1) Potentsiomeetrid
- 2) Bluetooth moodul
- 3) Kaks funktsionaalset nuppu
- 4) Nupp ON/OFF
- 5) LED indikaatorid 5 tükki
- 6) GÜroskoopiline kiirendusandur

Potentsiomeetreid kasutatakse manipulaatori eeskujuks oleva käe asendi määramiseks. Bluetooth moodulit kasutatakse siin andmevahetuseks. Kahte funktsionaalset nuppu on võimalus programmeerida erinevate tegevuste jaoks. ON/OFF nuppu kasutatakse toite väljalülitamiseks, säästmaks toiteallika laetust. LED indikaatoreid kasutatakse töörežiimide määramiseks. GÜroskoopiline kiirendusandur võimaldab määrata objekti asukohta, liikumist ruumis ja nurkkiirust pöörlemise ajal. [1][3][5]

Kuigi saadetud komplektis pidi sisalduma ka lähtekood, laadija, USB-kaabel ja skeemid, kuid ükski neist ei tulnud pakis kaasa.. [1]

1.2 Lisavahendite ülevaade

Järgnevalt on esitatud ülevaade sensoriplaati sisseehitatud anduritel ja lisaseadmete spetsifikatsioonid.

1.2.1 Bluetooth

Bluetoothi moodul on selle versioon HC-08 (vt. Joonis 1.2).



Joonis 1.2 Bluetooth module HC-08 [29].

See seade põhineb Bluetoothi protokollit spetsifikatsioonil V4.0, samuti on see seade ultra-väikese energiatarbimisega BLE (Bluetooth Low Energy). See tehnoloogia tagab, et moodulit kasutatakse ainult andmete saatmise ajal. See annab võimalust oluliselt säästa energiat, mis annab parema võimaluse kasutada seda moodulit aku toitega toodetel. Samuti suureks plussiks võrreldes Bluetoothi mooduliga HC-05 on automaatne ühendus teise Bluetooth V4.0 mooduliga. Selle mooduliga pole vaja kasutada oma AT-käskudes IP-d moodulite ühendamisel. See on väga mugav ja säästab programmeerijatele palju aega. Siiski on võimalik programmeerida Bluetoothi tööd ka AT-käskude abil. Bluetooth ühendamisel teise Bluetoothiga on kasutusel valgusdiood STA (olekunäidik) (vt. Joonis 1.3), mis põleb ühendamisel ning vilgub otsimisel. [1][3][4]



Joonis 1.3 Bluetooth olekuindikaator [1].

Omadused: [3]

- 1) Raadiosignaali sagedus: 2.40 - 2.48 GHz
- 2) Suurus: 27 x 13 X 1,8 mm
- 3) Side protokoll: Bluetooth v4.0 BLE
- 4) Tegevuse kaugus: 80 m
- 5) Kontrolleri: CC2540 Texas Instruments
- 6) Tööpinge: 3.3 V

1.2.2 GÜROSKOOP - AKSELEROMEETER

Güroskoop – akseleromeetrina kasutatakse mudelit MPU6050 (vt. Joonis 1.4). See moodul on 3-teljeline güroskoop ja kiirendusandur. See andur võimaldab määrata objekti asukohta, selle liikumist ruumis ja nurkkiirust pöörlemise ajal. [5]



Joonis 1.4 Güroskoop – akseleromeeter [30].

Omadused: [5]

- 1) Mikroskeemi tüüp: MPU-6050
- 2) Toitepinge: 3,5V kuni 6V
- 3) Güroskoobi tundlikkus: ± 250 500 1000 2000 ° / s
- 4) Aktseleerandi vahemik: ± 2 ± 4 ± 8 ± 16 g
- 5) Side liides: I2C
- 6) Suurus: 15x20 mm.
- 7) Kaal: 5 g

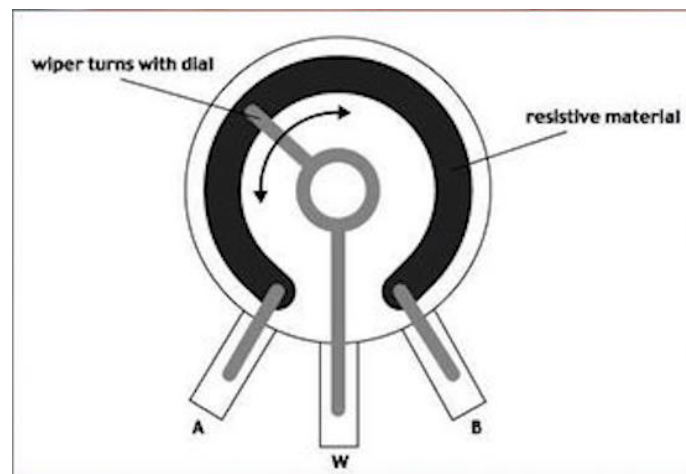
1.2.3 Potentsiomeetrid

Samuti kasutatakse sellel trükkplaadil potentsiomeetreid (vt. Joonis 1.5). Potentsiomeeter on muutuva takistusega komponent, milles takistus muutub potentsiomeetri võlli pööramisel.



Joonis 1.5 Potentsiomeetrid [1].

Need potentsiomeetrid on ühendatud mikrokontrolleri analoogsisenditega. Analoogsisendid võivad mõõta pinget 0 voldist kuni 5 voldini. Peale pinget mõõtmist saadud analoogpinge digitaliseeritakse, et sellega saaks edasi töötada. Atmega328P digitaal-analoogmuunduril on resolutsioon 10 bitti. See tähendab, et kui analoog-digitaalmuunduri sisendile rakendatakse pinget 0 voldist kuni 5 voldini, siis ADC (Analoog-digitaalmuundur) väljundis saame pingele vastava väärtuse arvulise väärtuse, mis on vahemikus 0 kuni 1023. Neid väärtusi kasutatakse hiljem kontrolleri programmi koostamisel. [6]



Joonis 1.6 Potentsiomeeter skeem [31].

Kõigil potentsiomeetritel on olemas kolm klemmi. Takistust tekitava omavahel materjalile on ühendatud äärmised klemmid ning keskmise klemmiga on ühendatud liugur, mis liikudes mööda resistiivset pinda muudab takistust. Keskmisele klemmile ühendatakse mikrokontrolleri analoogsisend, külgmised klemmid ühendatakse GND ja 5 voldise toitega (vt. Joonis 1.6). [6]

1.2.4 Sensorplaadi skeem

Internetiotsinuga oli selgitatud juhtplaadi selektriskeem, mis on esitatud allpool täiendavate materjalide. Need skeemid näitavad ära, mis lisaseade kuhu ühendatakse. Sealhulgas ka selle, millised klemmid on Bluetooth-mooduliga ühendatud jpm.

1.3 Olemasoleva robotkäe komplekti ülevaade

Teine komplekt, mis on antud lõputöö aluseks, sisaldas detaile mehaanilise käsivarre manipulaatori koostamiseks nimega 5DOF Bionic (vt. Joonis 1.7). Käsi pidi koos saabuma käsiraamatu ja paigaldusjuhendiga, kuid seda polnud seal, ja seetõttu oli esimeseks ülesandeks leida juhised selle mehaanilise käe kokkupanemiseks. Kuna Internetis dokumentatsiooni otsimine ei andnud

tulemust, on käe manipulaator kokku monteeritud toote fotode alusel, mis olid lisatud veebilehel, kust telliti selle mehaanilise käe manipulaatori komplekti. Lisaks prooviti ühendust võtta tarnijaga, et saada nende käest selle toote koostamisjuhised, kuid nad ei vastanud. Piltide alusel õnnestus lõpuks edukalt antud mehaaniline manipulaatorit komplekteerida [2].

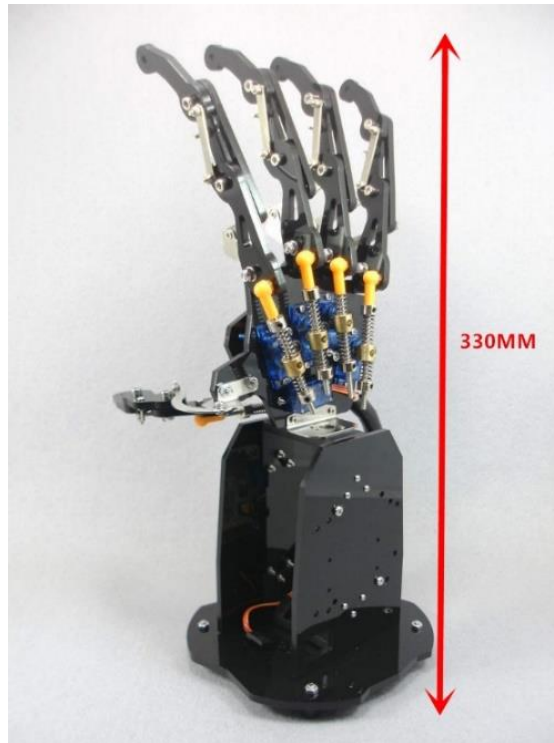


Joonis 1.7 Mehaaniline käe manipulaatori osad [2].

Komplekt sisaldab: [2]

- 1) 5 servomootorit
- 2) Detailid mustast akrüülist
- 3) Poldid, mutrid, kruvid jne.

Manipulaatori alusega on mehaaniline käsi 200 mm pikkune. Alusega oli selle manipulaatori kõrgus 330 mm (vt. Joonis 1.8). [2]



Joonis 1.8 Mehaanilise käe manipulaator [2].

Samuti veebilehel, kust olid tellitud antud mehaanilise käe manipulaator, tarnijad soovivad osta selle roboti jaoks spetsiaalse trükiplaadi, mis on valmistatud Arduino Uno baasil. Kuna see pole saadaval, kasutati tavalist Arduino Uno arendusmoodulit. Kuna komplektis ei sisaldu midagi muud, peale käe komponentide, siis selle manipulaatori jaoks tuli tellida lisaks erinevaid andureid ja lisaseadmeid, sealhulgas näiteks.

Näiteks:

- 1) Bluetooth moodul HC-08
- 2) Potentsiomeetrid

1.3.1 Bluetooth

Nendest komponentidest võib lugeda üleval peatükis 1.2. Kahe mikrokontrolleri omavaheliseks ühendamiseks on vaja Bluetooth-moodulit HC-08, et nad saaksid omavahel andmeid edastada. Potentsiomeetreid selles manipulaatori mudelis kasutatakse servomootorite kontrollimiseks ja mehaanilise käe manipulaatori mehaaniliste osade liikumise kontrollimiseks. Kuid isegi mehaanilise käe manipulaatori mehaaniliste osade liikumise kontrollimisel tekkisid probleemid. Kui telliti uusi Bluetooth-moduleid antud manipulaatori jaoks, polnud mooduli mudel näidatud. Saabumisel selgus, et HC-05 Bluetooth-versioon ei olnud sobiv, sest Bluetoothi spetsifikatsiooni v2.0-sideprotokoll ei võimalda ühendada versiooniga 4.0. Ka AT-käsud Bluetooth-seadmete

programmeerimiseks olid erinevate versioonide tõttu väga erinevad. HC-05 moodulit oli keerulisem programmeerida kui HC-08, seega oli vaja kiirelt leida Bluetooth-moodul mudeliga HC-08, kuna seda kasutatakse ka juhtmevaba mehaanilise kinda juhtimisseadmes. Seetõttu oli mingi aeg töö peatunud, sest ühelt mikrokontrollerilt teisele ei olnud võimalik andmeid edastada. [2][3][6]

1.4 Mootorid ülevaade

Antud komplektis sisaldus ka 5 Tower Pro SG90 servomootorit (vt. Joonis 1.9), mida kasutatakse peamiselt väikeste lihtmehhanismide juhtimiseks. Antud mootoritega mehaanilise käe manipulaatoriga saab tõsta kuni 200 grammi kaaluvaid esemeid. [1][7]



Joonis 1.9 Tower Pro SG90 [7].

Tehnilised andmed: [7]

- 1) Tööpinge: от 3V до 7.2V (Volt)
- 2) Seadme mõõtmed: 22.6 x 21.8 x 11.4 mm
- 3) Kaal: 13 grammi
- 4) Kasutamise temperatuur: -30 kuni +60 C kraadi
- 5) Pööramisnurk: 0 kuni 180 kraadi
- 6) Väändemoment: 1,8 kg / sm (4,8 V)

Juhtmete tähistus SG90:

- 1) Pruun: GND
- 2) Punane: VCC 4.8-7.2V
- 3) Oranž: SIGNAL

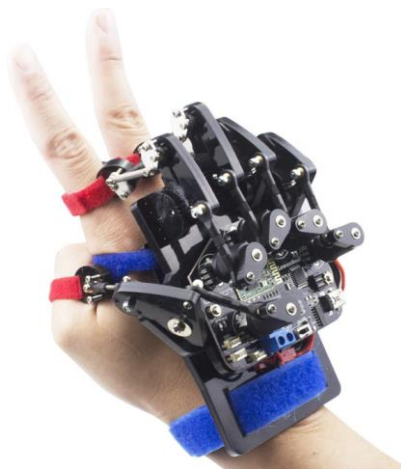
2 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMI PARANDAMINE

2.1 Robotkäe saatja/inimese järgija mehaanilise ülesehituse parandamine

Pärast juhtmevaba mehaanilise kinda ja mehaanilise käe manipulaatori valimist, oli ülesandeks parandada juhtmevaba mehaanilise kinda ja mehaanilise käe manipulaatorit. Pärast kahe komplekti valiku pikka analüüsi otsustati alustada juhtmevaba mehaanilise kinda parandamisest.

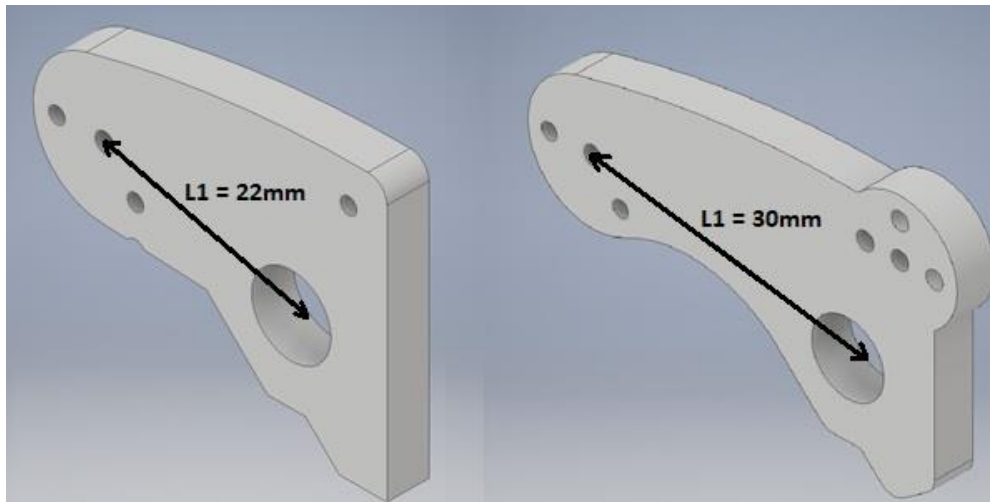
2.1 Mehaaniline juhtmevaba kinda parandamine

Juhtmevabal mehaanilisel kindal on 5 potentsiomeetrit (vt. Joonis 2.1). Nendel potentsiomeetritel on liigendid, mis võimaldavad sõrmede liigutamise viia üle potentsiomeetrite völli pöörlemiseks. Esialgsel kujul olid need liigendid väikesed ja ebamugavad. Täiskasvanu inimese käe liigutamine ja rusikasse surumine oli piiratud. Seetõttu oli otsustatud neid kinnitusi parandada. [1]



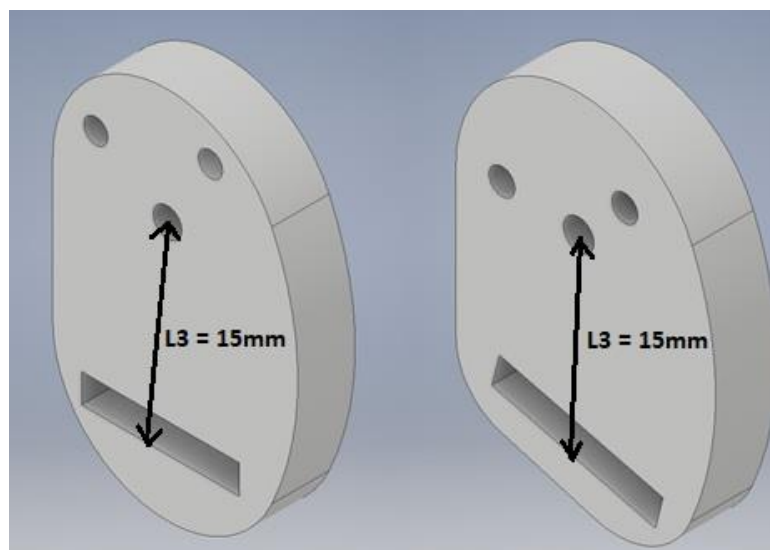
Joonis 2.1 Juhtmevaba mehaanilise kinda ülevaade [1].

Esialgselt oli otsustatud teha vanade detailide taasloomine 3D modelleerimise keskkonnas, kasutades Autodesk Inventori programmi. See programm oli valitud, sest programmi kasutati eelnevalt ülikoolis ja selle programmi jaoks on olemas tasuta tudengilitsents. Pärast detailide mõõtmist, mis oli tingitud detailide ja koostude käsiraamatut puudumisega, algas 3D mudeli loomine. Töötati välja kaks uut detaili mudelit, mis lahendasid kinnitusprobleeme. [8]



Joonis 2.2 Ülemised detailid sõrmede ühendamiseks kindaga.

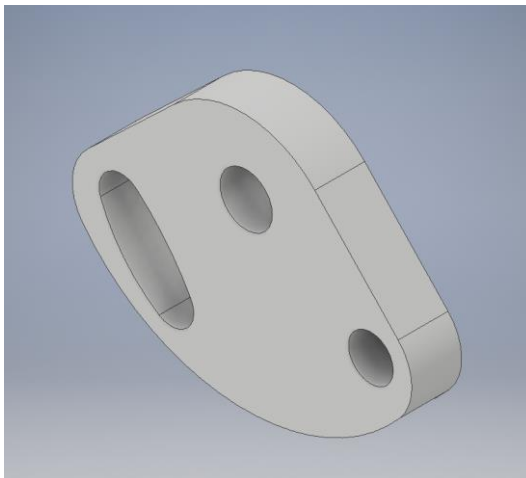
Pildil vasakul on näha originaal osa ja paremal on parandatud osa (vt. Joonis 2.2). Nagu näeme, on detail muutunud pikemaks, et saada rohkem vabadust sõrme painutamiseks. Eelmine versioon ei võimaldanud painutada sõrmi piisavalt hästi. Loodud detail andis võimaluse meil hoida kõik vanad avad ja lisada uusi, millega oleks võimalik seadistamine erinevate suurustega käte jaoks.



Joonis 2.3 Alumised osad sõrmede ühendamiseks kindaga.

See detail on mõeldud sõrmede kinnitamiseks juhtmevaba mehaanilise kinda külge. Läbi augu pannakse takjapael, mis fikseerib inimese sõrme. Pildil vasakul on originaal detail ja paremal on parandatud (vt. Joonis 2.3). Parandatud osas oli muudetud avasid, ühendamismehhanismi töökäigu suurendamiseks. Pööramisnurk on uue detailiga 50 kraadi. Samuti oli muudetud alumise osa kuju ja alumise ava asukohta, et lüli ei tekitaks kinda kasutamisel sõrmedele valu.

Kõigil sõrmedel, välja arvatud põidlal, lahendasid uued kinnitusdetailid sõrmede liikumiskiirapääsiooni suurendamise ülesande. Seetõttu otsustati ka põidla jaoks teha uus kinnitus (vt. Joonis 2.4).



Joonis 2.4 Kinnitus põidla fikseerimiseks.

Kõik detailid kinnitatakse poltide abil ja valmisid igati edukalt 3D-printeril

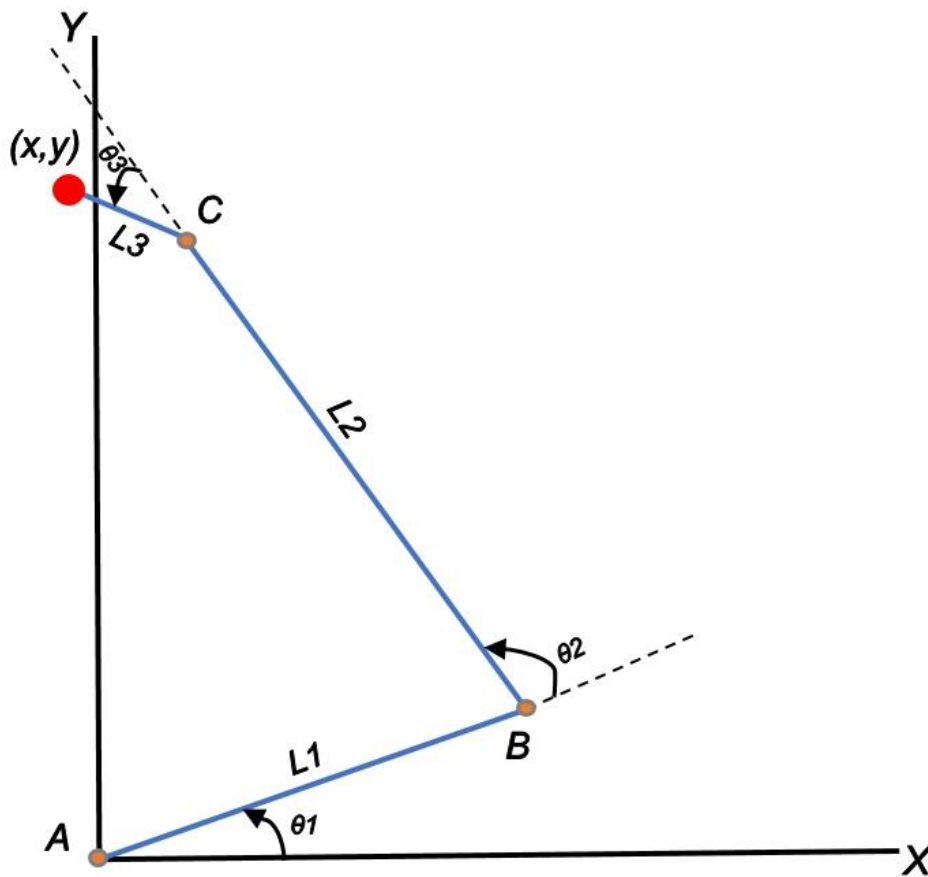
2.2 Kinemaatika arvutamine

Lõputöö üheks tähtsaks osaks oli kinemaatika arvutamine. Kinemaatika on mehaanika osa, kus kehade liikumist uuritakse, ilma liikumise põhjustaja uurimist. Kinemaatika vastab küsimusele - kuidas keha liigub. Kinemaatika ei tekita küsimusi, miks või mis oli sellise liikumise põhjus. [9]

Mehaanilise juhtkinda konstruktsioonis oli 6 lüli, mis kinnitavad sõrme kindaga. Lülisid ühendavad potentsiomeetreid selle kinda otse effektoriga ja otsas asub inimese sõrm. Kuna meie eesmärk on teha paremad kinnitused sõrmede paremaks paindumiseks, otsustati muuta esimest lüli ja kolmandat lüli. Peamiselt otse efektori positsiooni mõjutas esimene liiges. Kolmas lüli muutus, sest esialgsel kujul olid need kinnitused väikesed ja ebamugavad.

Juhtmevaba mehaanilise kinda kinemaatika arvutamiseks kasutati otsest kinemaatikat, kuna kavandatav juhtmevaba mehaanilise juhtkinnituse kinemaatiline mudel määrab roboti otse efektori asukoha ja orientatsiooni selle ühenduse nurkade põhjal. Otseste kinemaatilise analüüsi alguses määratakse koordinaatteljed. [10][11][12]

Esimeseks sammuks oli kinemaatilise skeemi koostamine (vt. Joonis 2.5). See tegevus aitas hoolikalt arutada roboti füüsilist konfiguratsiooni, vältides vigaseid oletusi.



Joonis 2.5 Kinda andurite kinemaatiline skeem.

Juhtmevaba mehaaniline kinnas võib töötada ainult ühel tasandil ja igal sõrmehoidjal on kolm painduvat liigest. Esimene liiges L1 on kinnitatud alusele ja pööratud nurgale θ_1 , teine liiges L2 on kinnitatud esimese liigendi otsale ja pööratud selle suhtes nurgale θ_2 , samuti kolmas ühendus on kinnitatud teise liigendi otsale ja pööratud selle suhtes nurgale θ_3 . Kinemaatika otsene eesmärk on leida tööorgani (x, y) koordinaadid, mis on antud $L1, L2, L3, \theta_1, \theta_2, \theta_3$. [12]

$\theta_1, \theta_2, \theta_3$ - tähistab nurga ühendused.

$L1, L2, L3$ - tähistab lüli pikkust (liigendi)

Lahendus: [12]

Meil on kolm loendamise süsteemi - esimene on seotud öla kinnituspunktiga $L1 - A$, teine küünarnuki kinnituspunktiga $L2 - B$ ja kolmas koordinaatide algusega küünarnuki kinnituspunktis - C. Leiame teise süsteemi nihke esimese suhtes (loendamise süsteemi A punkti B koordinaadid):

$$XA = L1 * \cos(\theta_1)$$

$$YA = L1 * \sin(\theta_1)$$

Joonisel on näha, et süsteemis A liiges on pööratud L2 õla suhtes $\theta_1 + \theta_2$:

$$x_1 = L2 * \cos(\theta_1 + \theta_2)$$

$$y_1 = L2 * \sin(\theta_1 + \theta_2)$$

Ja samuti joonisel on näha, et A süsteemis liiges L3 samuti pööratud ka õla suhtes

$\theta_1 + \theta_2 + \theta_3$:

$$x_2 = L3 * \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y_2 = L3 * \sin(\theta_1 + \theta_2 + \theta_3)$$

Järelikult:

$$x = XA + x_1 + x_2 = L1 * \cos(\theta_1) + L2 * \cos(\theta_1 + \theta_2) + L3 * \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y = XA + y_1 + y_2 = L1 * \sin(\theta_1) + L2 * \sin(\theta_1 + \theta_2) + L3 * \sin(\theta_1 + \theta_2 + \theta_3)$$

Arvutused viidi läbi maksimaalse pöördenurga all, et näha, kus otse effektor asub, kui sõrmed on täielikult kokku painutatud (vt. Table 2.1 ja Table 2.2).

Vanade detailide väärtused (x,y) saame:

Table 2.1 Kinda varasemate detailide väärtused.

L1	L2	L3	θ_1	θ_2	θ_3
22 mm	43 mm	15 mm	35°	110°	40°

$$x = XA + x_1 + x_2 = -32.1451$$

$$y = XA + y_1 + y_2 = 35.9751$$

Uute detailidega saadakse väärtused (x, y), kui:

Table 2.2 Kinda detailide praegused tähendused.

L1	L2	L3	θ_1	θ_2	θ_3
30 mm	45 mm	15 mm	35°	105°	40°

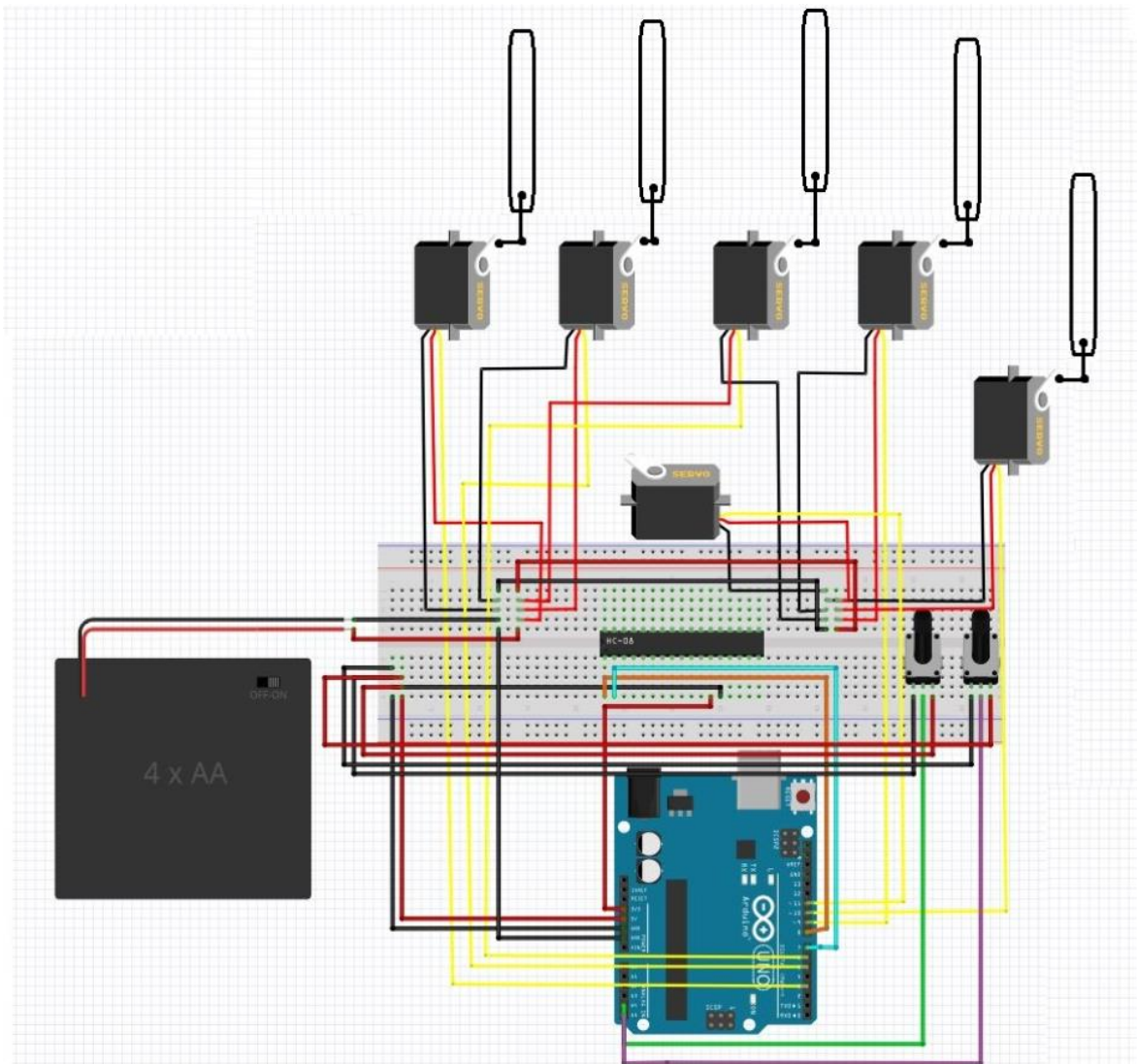
$$x = XA + x_1 + x_2 = -24.8974$$

$$y = XA + y_1 + y_2 = 46.1327$$

Tuleb märkida, et uute detailidega roboti otsefaktor asub allpool ja õla aluse läheduses. See teeb mugavamaks sõrmede painutamise, juhtmevaba mehaanilise kinda kasutamisel, mis oli esialgu väikeseks probleemiks selle juhtmevaba mehaanilise kinnas. Arvutamiseks oli kasutatud MATLAB programm. Täiendavates materjalides on võimalik leida arvutuste kood. [13]

2.3 Manipulaatori mehaaniline käsi parandamine

Järgmiseks, üks tähtsamaid samme, oli mehaanilise käe manipulaatori parandamine, kuna komplektis ei olnud midagi peale detaile ja servomootorite. Kõigepealt oli vaja leida mikrokontroller, mis võiks seda mehaanilist kätt juhtida. Nagu eespool oli mainitud, oli kasutatud Arduino Uno mehaanilise käe juhtimiseks. Pärast seda, kui oli valitud mikrokontroller, tehti skeem erinevate mootorite ja andurite ühendamiseks mikrokontrolleriga (vt. Joonis 2.6). Täpsema skeemi võib leida lisa täiendavates materjalides. [14]



Joonis 2.6 Mikrokontrolleri andurite ühendusskeem [14].

Nagu näha on selles skeemis kaks potentsiomeetrit. Neid kasutatakse servomootorite ja demo-programmide kontrollimiseks. Samuti kasutakse ka 5 Tower Pro SG90 servomootorit. Neid kasutatakse mehaanilise käe sõrmede juhtimiseks. Ühendatud on ka Bluetoothi moodul HC-08 andmete edastamiseks mikrokontrollerite ja teise K-POWER M1500 (vt. Joonis 2.7) servomootori vahel.



Joonis 2.7 K-POWER M1500 mikroservomootor [32].

Tehnilised omadused: [15]

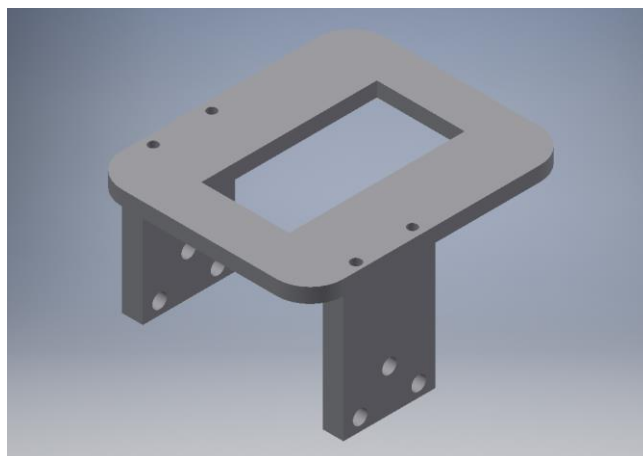
- 1) Suurus: 40*20*37.5 mm
- 2) Tööpinge: 4.8-6 V
- 3) Väändemoment: 15.5kg-cm/4.8 V; 16.3kg-cm/6.0 V
- 4) Kaal: 58 g
- 5) Pööramisnurk: 0 kuni 180 kraadi
- 6) Töötemperatuur: -10 kuni 60 C

Käe paremaks kasutamiseks olid vajalikud täiendavad kaks servomootorit kuna, ülesandeks oli täiendada süsteemi selliselt, et käsi saaks pöörata ja painutada, lisades universaalsust. Mootorite lisamisel tekkis probleem, sest pwm väljundid piisaval arvul puudusid. Arduino Uno arendussüsteemil oli neid vaid 6, kuid 7 tk oli kokku servomootoreid. Selleks prooviti seda mikrokontrollerit asendada Arduino Mega 2560 mikrokontrolleriga. Kuid siis sai probleemiks mikrokontrolleri Bluetooth-mooduli ühendamisel tühik, mis on kirjeldatud peatükkides 3.1 ja 4.2. Järelikult oli võimalik kasutada ainult 6 servomootorit.

Samuti näitab skeem, et servomootorite jaoks on täiendav toiteallikas. Pole soovitatav ühendada võimsaid servomootoreid otse plaadiga, sest need tarbivad voolu, mis kokku ületab Arduino toiteallika piiri. Kõige sagedamini on ülekoormuse ja vale servo toite sümptomid servomootori liikumisel tekkivad tõmblemine ja ebameeldiv heli. Toiteallika jaoks on parem kasutada väliseid allikaid, ühendades kokku toiteallikate ja Arduino mooduli maad, et ka Arduino saaks signaali. [16]

Järgmiseks etappiks pärast skeemi koostamist oli uute osade loomine Autodesk Inventori 3D-modelleerimiskeskonnas. Uued detailid olid vajalikud mehhanismi jaoks, mis võimaldaksid mehaanilise käe manipulaatori pöörlemist ja painutamist. Pärast selle konstruktsiooni analüüsi ja mõõtmist otsustati kasutada kahte servomootorit, mis oleksid kinnitatud selle konstruktsiooniga ja võimaldaksid manipulaatori mehaanilist kätt pöörata ja painutada. Kuna käsi ise kaalus üsna palju, ei suutnud Tower Pro SG90 servomootorid seda kätt pöörata ja painutada. Kuna nende pöördemoment oli 1,8 kg / cm (4,8 V), võeti kasutusele K-POWER M1500 servomootorid, millel on suurem pöördemoment, näiteks 15,5 kg / cm / 4,8 V; 16,3kg-cm / 6,0. Esialgu kasutati K-POWER M1500 asemel Parallax Continuous Rotation servomootoreid. Need servomootorid said pöörata 180 kraadi asemel 360 kraadi. Pärast kokkusobimatuse tuvastamist leiti vajalikud K-POWER M1500 mootorid. Pärast töötati välja 6 detaili, mis olid hoolikalt kinnitatud antud konstruktsiooniga mehaanilise käe manipulaatori juhtimiseks. [7][15][17]

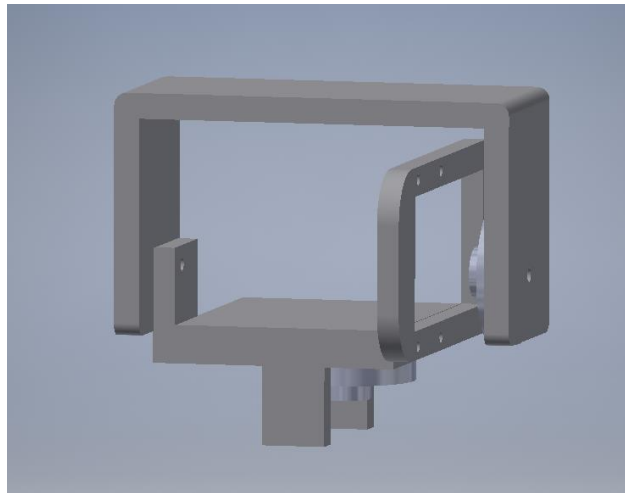
Esimesed detailid, mis olid tujundatud, olid servomootori kinnitused (vt. Joonis 2.8). Välja pakuti detail servomootori paigaldamiseks ja detail alusele paigaldamiseks. Samuti oli tehtud käsitsi 4 ava, kasutades servomootori fikseerimiseks. Servomootor on paigaldatud sellesse kinnitussele horisontaalselt, sellega tagatakse mehaanilise käe manipulaatori pöörlemine. Peaaegu kõik detailid kinnitati poldiga, välja arvatud servomootori kinnitus, mis kinnitati detailile, et see ühenduks alusega. Seal kasutati kruvisid.



Joonis 2.8 Servomootorite kinnitus, mis ühendab alusega.

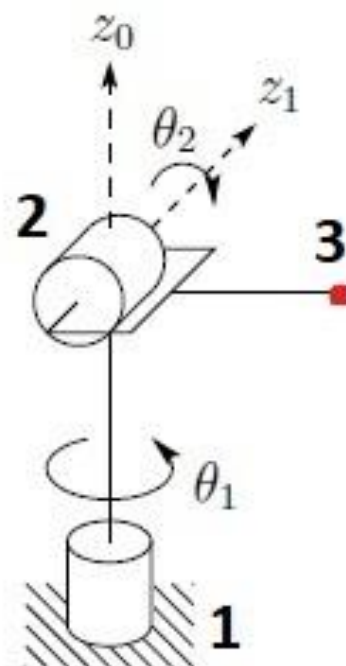
Järgmiseks etappiks oli luua servomootorile kinnitus teise servomootori jaoks, selleks et mehaanilise käe manipulaatorit oleks võimalik painutada (vt. Joonis 2.9). Järgmiseks oli valmistatud servomootorile kinnitus - 4 ava, kasutades servomootori fikseerimiseks, et servomootor saaks paikneda vertikaalselt mehaanilise käe manipulaatori painutamiseks. Pärast oli tehtud aluse tugi

teise servomootori alumisele servomootorile paigaldamiseks. Pärast oli mõeldud välja kinnitus, mis võimaldaks käe painutamist, kuid selleks oli vaja teha veel kaks auku trelli abil. Lisaks oli tehtud veel üks detail selle kinnituse fikseerimiseks. Peaaegu kõik detailid olid kinnitatud poltidega, välja arvatud servomootori kinnitus, mis oli kinnitatud alumise servomootori alusele. Seal kasutati kruvisid.



Joonis 2.9 Kinnitus, mis võimaldab käe painutamist.

Samuti meil on võimalus kinemaatika arvutamiseks kasutati Denavit-Hartenberg parameetreid. Kinemaatiline mudel määrab roboti otsaefektori asukoha ja orientatsiooni selle ühenduse nurkade põhjal. Seejärel on vaja koostada kinemaatiline skeem (vt. Joonis 2.10). See tegevus aitab hoolikalt arutada roboti füüsilist konfiguratsiooni, vältides vigaseid oletusi. [10][28]



Joonis 2.10 Kinemaatiline skeem 2 [28].

1 – keha

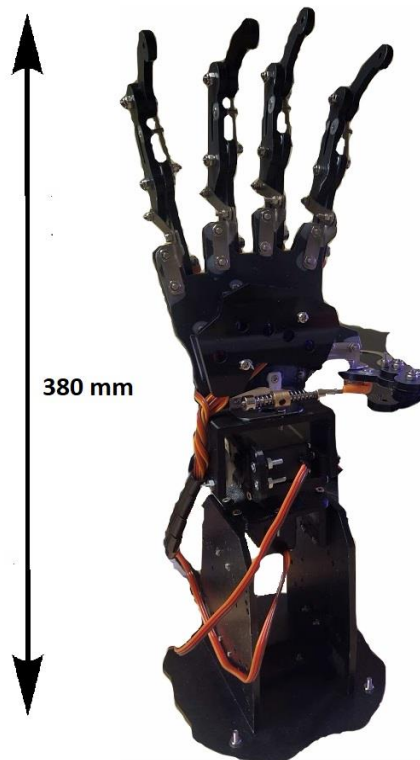
2 – õlg

3 – roboti otsaefektor

θ_1, θ_2 - esitab nurga ühendused.

Kuna teine mootor ei ole kaasatud mikrokontrolleri tõttu, sellepärast kinemaatika ei ole arvatud.

Pärast detailide trükkimist ja kokkupanekut oli mehaanilise käe manipulaatoril võimalus pöörduda ja painduda, kuid käe kaal ja kõrgus muutusid. Alusega on selle manipulaatori kõrgus 380 mm (vt. Joonis 2.11).



Joonis 2.11 Manipulaatori mehaaniline käsi parandatud.

3 JUHT PROGRAMMI KUJUNDAMINE ROBOTKÄE MANIPULEERIMISEKS JA INIMKÄE LIIKUMISE ÜLEKANDMISEKS TÄITURKOMPLEKTILE

3.1 Esimene programm mehaanilise kinda juhtmevaba juhtimise jaoks

Programmi loomise esimese etapina tuli luua programm juhtmevaba mehaanilise kinda jaoks. Programm on kirjutatud Arduino IDE arenduskeskkonnas. Alguses oli meil eesmärk ühendada omavahel kaks Bluetooth-moodulit HC-08. Selleks oli vaja aktiveerida AT käsud. AT-käske kasutatakse mooduli seadistuste rakendamiseks. Parameetrite muutmist saab teha ainult enne ühenduse loomist teiste Bluetooth-seadmetega. AT-käskude aktiveerimiseks oli vaja luua kood, mis võimaldab Arduino Uno's juhtida uut jada-andmeliidese ühendust ja sooritada andmevahetust seeriamonitori ja Bluetooth-mooduli vahel. Teiste sõnadega, vajalik on veel ühte jadaporti andmete edastamiseks Bluetoothi kaudu, koodis aktiveerime teegi "<SoftwareSerial.h>" ja seejärel kirjutame koodi sisse, et TX ja RX kontaktid on ühendatud 11 ja 12 väljunditega. Seejärel loome uue jadapordi. Samuti peab seadistama Bluetooth-mooduli asteastmelise side kiiruse ja peamise seeriapordi kiiruse, mis on 9600 b/s. Pärast ülaltoodut avame Arduino IDE jadaliidese monitori ja selles aknas on ka võimalus teha muudatusi. Kui sisestada käsk AT + RX ja kuvatakse Bluetoothi mooduli omadused, siis programm on tehtud õigesti (vt. Joonis 3.1). Täiendavates materjalides on näha selle Bluetooth-mooduli koodi. [3]

 COM7

```
Enter AT commands:  
Name:HC-08  
Role:Master  
Baud:9600,NONE  
Addr:C8,DF,84,FF,6D,29  
PIN :000000
```

Joonis 3.1 AT+RX käskude näited.

AT-käsud, mida me vajame: [3]

- 1) AT - Testkäsk
- 2) AT+RX – Peamiste parameetre kontroll
- 3) AT+ROLE=x - Rolli valimine peamine / sekundaarne. Tähhendused x: M, S
- 4) AT+CLEAR - Bluetooth aadressi puhastus
- 5) AT+DEFAULT – Tehaseseadetele tagastus (vaikimisi)
- 6) AT+LUUID=? - Otsingu ID läbivaatus, samas on võimalik seda ka muuta, kirjutades midagi "?" asemel.

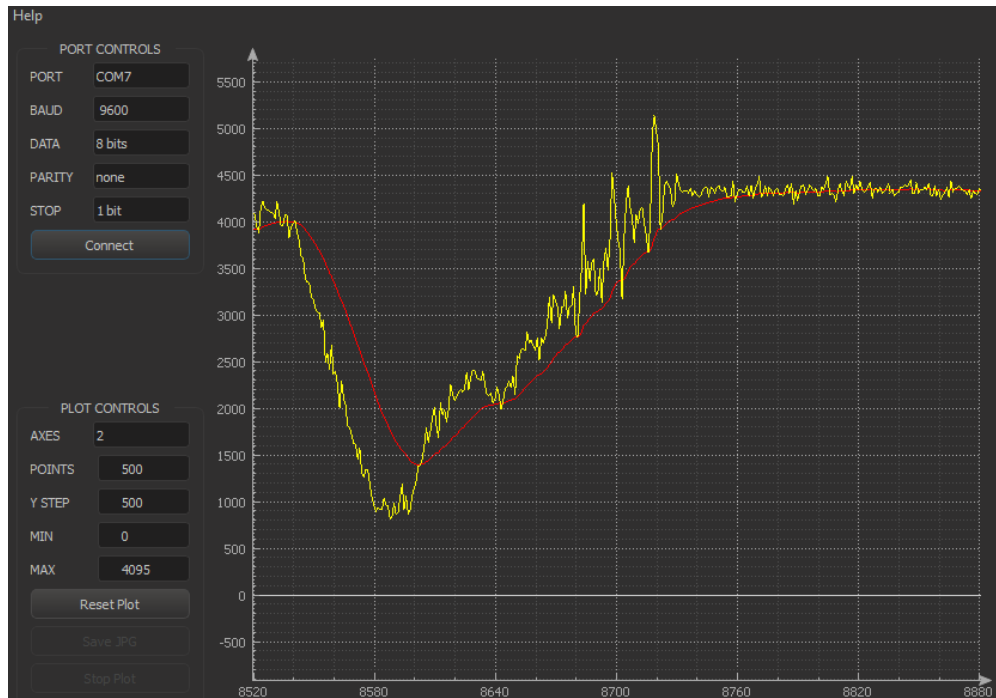
Kahe Bluetooth-mooduli ühendamiseks piisab, kui üks neist on master. Kui Bluetooth-moodulite vahel on ühendus, siis peamine Bluetooth mäletab teiste Bluetooth-moodulite unikaalset aadressi ja tulevikus ühendab ainult temaga. Selleks, et ühenduda teise Bluetooth-mooduliga peab tühjendama teise Bluetooth-mooduli unikaalse aadressi. Seda saab teha kolmanda käsu abil. Kui tahame, et Bluetooth-moodul ühenduks ainult ühe Bluetooth-mooduliga, peame tegema ühise otsingu ID. Vaikimisi on kõikidel moodulitel samasugused parameetrid. [3][18]

Järgmiseks etapiks oli teha MPU6050 güro-akseleromeetri väljundile sobiv filter (vt. Joonis 3.2). Kõigi selliste akseleromeetrite, güroskoopide probleem - nad on väga mürarikka väljundiga.. Nad peavad olema täpselt kalibreeritud ja filtreerima vastuvõetud andmeid. Nende väärtuste kalibreerimiseks kasutati väga lihtsustatud versioonis Kalmani filtrit. Maatriksid ja mitmetasandilised võrrandid siin puuduvad, oli kasutatud ainult üks valem: [19]

$$M_n = k * A_n + (1 - k) * M_{n-1}$$

Kus:[19] M_n - praeguse arvutuse tulemuse määr
 k - stabiliseerumisühik
 A_n - Käesoleva mõõtmise algtähendus
 M_{n-1} - eelmise arvutamise tulemuse tähendus

Nagu näha kõik, mis on vajalik, on ainult stabiliseerimisteguri valimine. See peab olema suurem kui 0 ja väiksem kui 1. Mida väiksem on koefitsient, seda rohkem andmed silutakse, kuid samal ajal stabiliseerimisaeg suureneb. [19]

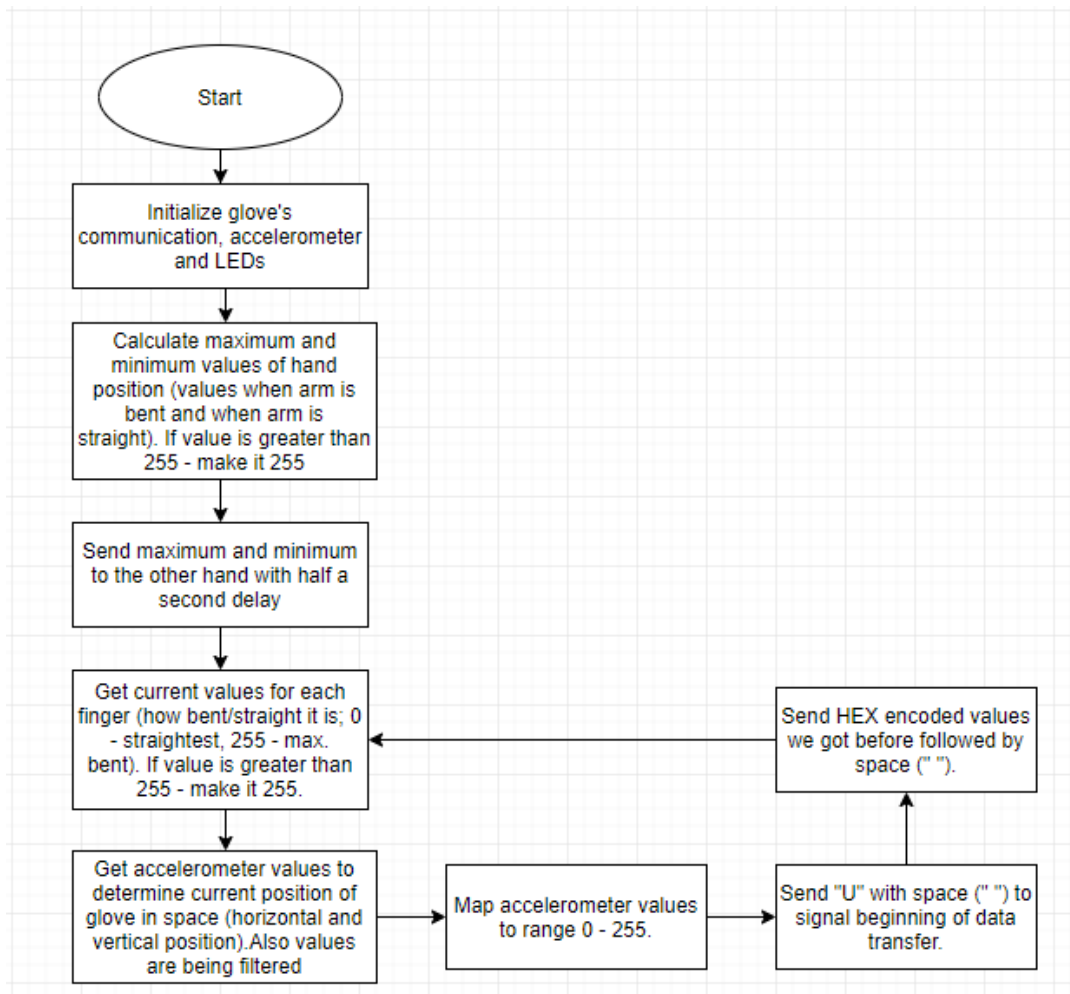


Joonis 3.2 Kalmani filtrit.

Punane joon on anduri filtreeritud tähendused ja kollane joon ei ole anduri filtreeritud tähendused. Nagu võime näha, lähevad filtreeritud väärtused sujuvamas, ilma järskude hüpeteta. Lisamaterjalides näete selle mudeli koodi MPU6050. Tulevikus kasutatakse seda koodi põhiprogrammis funktsioonina. [27]

Kolmeks etapiks oli lugeda tähendused potentsiomeetritest. Selleks on vaja kasutada funktsiooni `analogRead()`. See funktsioon loeb määratud analoogsisendi väärtise. Kuna mehaanilise kinda juhtmevaba jaoks on 5 tükki, oli vaja arvutada kõigi 5 väärtusest. Lugemisel tekkis problem, potentsiomeetrite ühendamise analoogsisendite tõttu. Esimesed neli potentsiomeetrit ühendati nullist kolmandasse sisendisse, aga neljas ühendati kuuendasse analoogsisendisse. Lisaks toodud materjalides on esitatud potentsiomeetritest suuruste lugemise kood. [20]

Neljandaks etapiks oli ühendada kõik väikesed alamprogrammid üheks suureks juhtmevaba mehaanilise kinda programmiks. Selle programmiga on see võimeline edastama ja lugema vajalikke väärtusi, et juhtida mehaanilise käe manipulaatorit. Selles oligi esimese juhtmevaba mehaanilise kinda juhtimise programmi olemus.



Joonis 3.3 Juhtmevaba mehaanilise kinda algoritm [34].

Nagu kinda algoritmi tabelist on näha (vt. Joonis 3.3), alustame käivitumisel aktiveerides side kindaga, akseleromeetriga ja LEDidega. Järgnevalt arvutame käe asendi maksimaalse ja minimaalse sisendite väärtused, kui käsi on painutatud ja kui on sirge. Seda tehakse selleks, et seadistada asendeid kasutaja käe mugavuseks, et neid muutuseid servomootoritele korrektselt edasi anda, sest inimeste käed võivad olla erineva suurusega. Siis saadetakse mehaanilise käe manipulaatori maksimaalsed ja minimaalsed piirid poole sekundi viivitusega, kuna viivitamata ei saa arvutada kõiki suurusi. Peale seda viiakse läbi igalt sõrmelt saadavate väärtuste lugemine. 0 tähendusega on sõrm sirges asendis, 255 on painutatud asendis. Kui tähendus on suurem kui 255, piiratakse see 255 väärtusele, kuna saadud suurused on salvestatud baitides ja bait on täisarv vahemikus 0 ... 255. Pärast seda saame MPU6050 anduri väärtused, et määrata kinda praegune horisontaalne ja vertikaalne asend ruumis. Samal tähendused filtreeritakse stabiilsemate tulemuste saamiseks. Järgnevalt olid tehtud mõõtmised andurilt saadud tähenduste suhtes vahemikus 0 kuni 255, kuna saadud suurused on salvestatud ka baitides. Seehärel tuleb "U" saatmine tühikuga (« »), et teatada andmeedastuse algusest. Pärast läheb kodeeritud tähenduste saatmine kuueteistkümnendvormis,

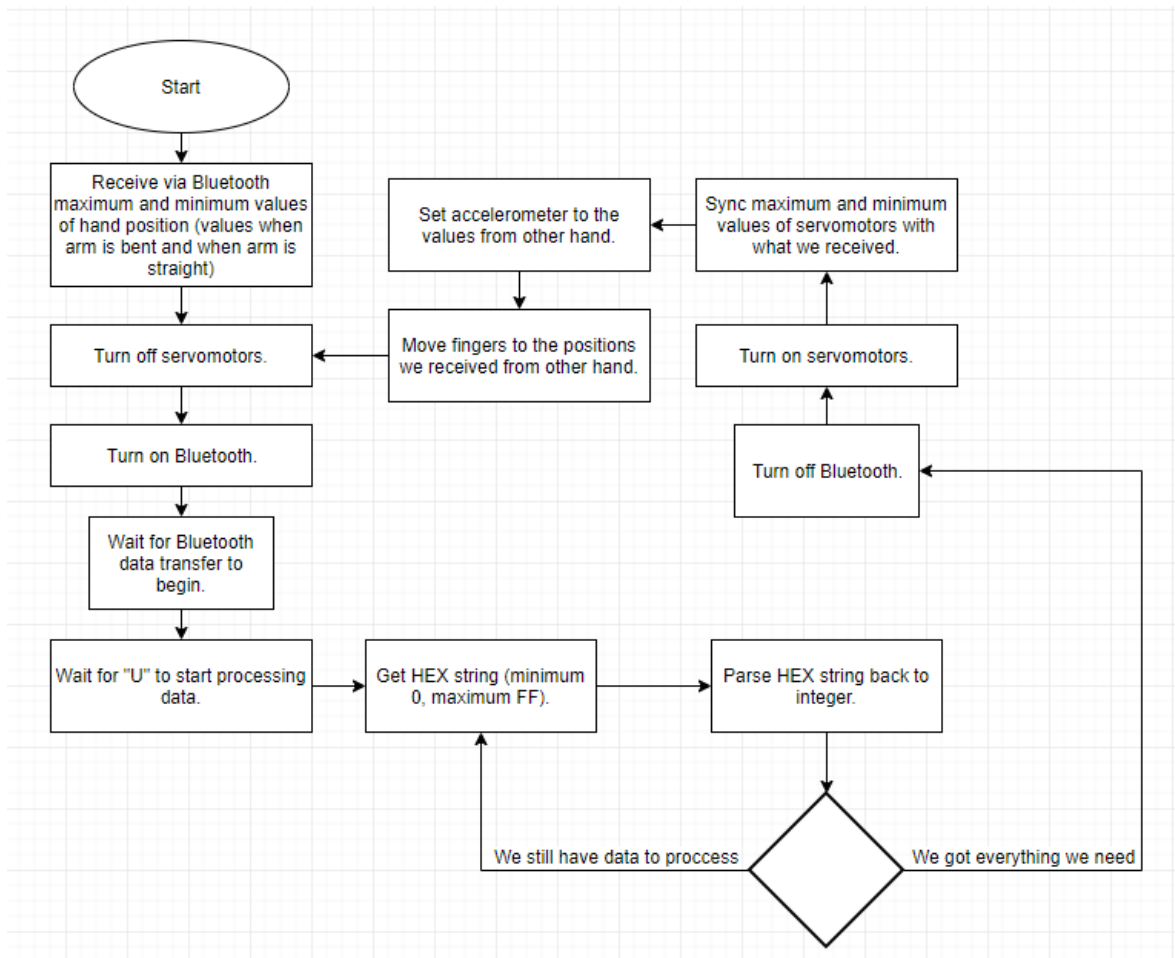
millele järgneb tühik. Need tühikud on vajalikud mehaanilise käsivarrega manipulaatori tähenduse täpsemaks lugemiseks. Pärast viimast toimingut program tuleb tagasi iga sõrme praeguste väärtuste saamiseks. Programm alustab tsüklit uuesti. Lisamaterjalides näete mehaanilise kinda juhtmevaba juhtimiskoodi.

3.2 Teine programm manipulaatori mehaaniline käe jaoks

Kõige esimeseks etapiks mehaanilise käe manipulaatori programmi loomisel oli servomootorite kontrollimine, selleks ühendati kaks potentsiomeetrit ja oli kirjutatud programm, mis kasutab potentsiomeetri tähenduste suhet vahemikus 0 kuni 255 nurgal 90 kuni 180 kraadi ning seda infot edastati servomootorile. Seega oli tehtud mootori kontroll. Servomootorite juhtimiseks kasutati raamatukogu «<VarSpeedServo.h>». Selle teegi toetatud funktsioonid võimaldavad juhtida ja reguleerida servomootorite kiirust, samuti kasutab see kohandatult omakorda standartset teeki «<Servo.h>» . [21]

Teiseks etapiks oli kahe Bluetooth-mooduli HC-08 ühendamine omavahel. Seda protsessi kirjeldatakse eespool. Bluetoothi juhtmooduli juhtmevaba mehaanilise kinda juhtimisel on kaks erinevust. Bluetooth-mooduli roll tuleb valida sekundaarseks ja seejärel registreerida ühendatud TX- ja RX-kontaktid 7 ja 8, et nad saaksid omavahel ühendada. Ülejäänud juhtudel on kõik toimingud samad, mis on mehaanilise juhtmevaba juhtimise kinda Bluetooth-moodulis. [3]

Pärast ülaltoodud toiminguid peab alamprogrammide ühendama põhiprogrammiga, mis jätkab mehaanilise käe manipulaatori juhtimist. Selle põhiprogrammi abil saab mehaanilise käe manipulaator saada väärtuseid juhtmevabast mehaanilise kindast, teisendada väärtuseid ja edastada need servomootoritesse, et juhtida mehaanilist kätt.



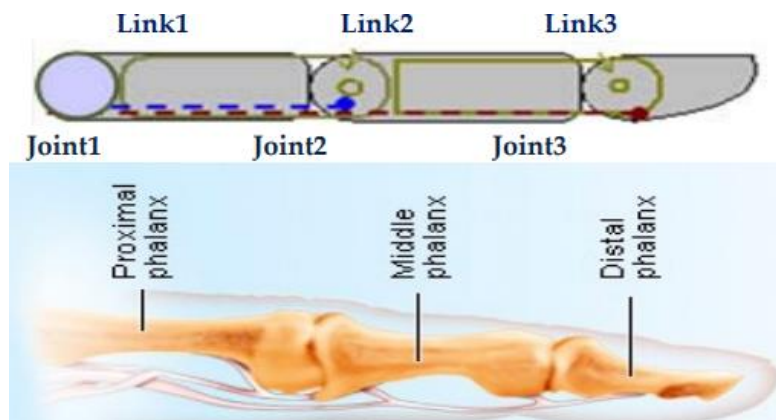
Joonis 3.4 Mehaanilise käe manipulaatori algoritm [34].

Nagu me võime näha (vt. Joonis 3.4), ootab mehaanilise käe manipulaator algusest peale, et kätte saaks maksimaalse ja minimaalse käe positsiooni Bluetooth-mooduli kaudu, kui käsi on painutatud see on MAX ja kui käsi on sirge siis MIN. Pärast lülitatakse servomootor välja ja lülitakse sisse Bluetooth-moodul. See on tehtud selleks, et servomootorid ei tekitaks endale kahju. Uuring näitas, et nad võivad kahjustuda aktiivse jadapordi käivitamise tõttu. Seega, kui Bluetooth-moodul töötab, mootorid lülitakse välja. Pärast ootab mikrokontroller andmeedastuse algust, ootab, et "U" tähendus hakkaks andmete töötlemist alustama, ja seejärel läheb ridade kättesaamine kuuteistkümnend vormingus, 0 on minimaalne ja FF on maksimaalne. Sellele järgneb kuuteistkümnend vormingu muundamine täisarvuks. Kui töötlemiseks on veel andmeid, siis jätkub kuuteistkümnend vormis ridade vastuvõtt, kui meil on kõik, mida vajame, siis Bluetooth-moodul lülitakse välja ja seejärel toimub servomootorite sisse lülitamine. Sellele järgneb alguses saadud servomootoritelt maksimaalsete ja minimaalsete tähenduste sünkroniseerimine. Samuti tuleb seadistada servomootori akseleraatori väärtused, mille saime. Järgmine samm on andmete edastamine servomootoritesse ja nihkumine positsioonile, mida saime teisest käest. Pärast viimast toimingut lülitatakse servomootorid välja ja Bluetooth-moodul on sisse lülitatud. Programm hakkab

töötama tsüklis. Täiendavates materjalides on toodud ka mehaanilise käe manipulaatori koodi näited.

3.3 Kasutusjuhend

Enne juhtmevaba mehaanilise kinda ja mehaanilise käe manipulaatori kasutamist peate esmalt programmikoodi mikrokontrolleritesse alla laadima. Pärast seda, peate oma käele panna juhtmevaba mehaanilise kinda ja selle hästi fikseerima. Punased kinnistused (lõppefaktor) peavad paiknema lülidel 1 (vt. Joonis 3.5).



Joonis 3.5 Skemaatiline näpu vaade [11].

Pärast seda, kui kinnas on käes, on vaja ühendada toitumist mehaanilise käepideme manipulaatoriga. Mehaanilise käe manipulaator läheb kohe ooterežiimi, et saada teavet teisest Bluetooth-moodulist. Seejärel anname juhtmevaba mehaanilisele kindale toite. On kaks võimalust. Esimene võimalus on kasutada käe juhtmevaba juhtseadmega komplektis olevaid akusid, teine on käe ühendamine arvutiga USB-pordi abil. Kasutades jadapordi monitori saame jälgida, kuidas väärtused muutuvad. Samuti saate ühendada ka mehaanilise käe manipulaatori. Kui toiteallikas on ühendatud ja põlevad viis indikaatorit, programm hakkab tööle.



Joonis 3.6 Led indikaatorid [1]

Niipea, kui LED-tuled põlevad (vt. Joonis 3.6), on käe painutamiseks teil on 5 sekundit. Seda tehakse väärtuste lugemiseks maksimaalse kumeruse juures. Seejärel kustuvad need 5 indikaatorit ja on jälle viis sekundit, et käsi sirgeks ajada, et arvutada minimaalsed väärtusid, mida saab teie käega kasutada. Seejärel vilguvad LED-indikaatorid üks kord ja algab toimingu edastamine teisele.

Alguseks edastab see väärtused maksimaalsele, seejärel minimaalsele väärtusele ja seejärel saadab need väärtused pidevalt mehaanilisele käele ning manipulaator paindab sõrmed ja liigub juba sõrmede ja käe asendi järgi. Seda kätt saab kasutada näiteks lihsate ülesannete jaoks, ilma suure nõudmistest liigeste liikumise täpseta, tööstuses jne.

4 INIMKÄTT JÄRGIVA ROBOTKÄE-SÜSTEEMI EDASIARENDUSED

Mehaanilise käe manipulaatori juures on hetkel võimalik kahte aspekti parandada. Üks neist on ühendatud mikrokontrolleriga, teine täiendavate anduritega mehaanilise käe sõrmedel. Juhtmevaba mehaanilise kindale saab lisada näiteks Wifi mooduli pikema rakendusvahemaa ja andmeedastuskiiruse jaoks. Samuti võib täiendada programmi.

4.1 Mehaanilise käe manipulaatori edasiarendamine

4.1.1 Mikrokontroller

Kõige olulisemaks aspektiks on Arduino Mega 2560 arendusmoodulil täiendava jadapordi initsialiseerimine. See on vajalik selleks, et oleks võimalik juhtida veel ühega servomootorit käe painutamiseks. Ka tulevikus on võimalik vähendada mudeli plaadi mõõtmeid ning parem oleks teha oma kompaktseks ja mugavaks. Plaat võiks sisaldada vähem komponente ja väiksemat plaadi suurust, kus kuvatakse kõik vajaminevad pin-id, et saaks hiljem mootorid, andurid jne ühendada juhtida mehaanilise käega. Arduino Mega 2560 mikrokontrolleri arendusmoodulil on rakendatud ATmega2560 mikrokontroller (vt. Joonis 4.1). [22]



Joonis 4.1 ATmega2560 [22].

4.1.2 Andmevahetussüsteem

Samuti sobiks uude arendusplaati sisseehitatud Bluetooth-moodul või kiirem andmeedastusmoodul. Näiteks, ESP8266 (vt. Joonis 4.2). See on populaarne Hiina mikrokontroller Espressifi, mille peamiseks eeliseks on sisseehitatud WiFi-liides ja ühilduvus Arduino-ga. Ühilduvus tähendab võimet kirjutada programme ja alla laadida neid Arduino IDE kaudu. On teada, et Wi-Fi on parem kui Bluetooth, kuna vaateväljas on Wi-Fi vahemik umbes 300 meetrit, Bluetooth-seadmetel on vaid 30 meetrit. WiFi-ühendust kasutatakse ka traadita kohalikus (WLAN) võrgus.

Sellises võrgus võib olla suur hulk erinevaid seadmeid ja Bluetoothiga saate luua ainult kahe seadme vahelise piiratud võrgu. Üks peamisi eeliseid on andmeedastuskiirus. Võrdlemiseks: kuni 0.8 Mbit / s Bluetooth-iga ja kuni 108 Mbit / s Wi-Fi-ga. [23][24]



Joonis 4.2 ESP8266 moodul [33].

4.1.3 Andurid

Samuti saate mehaanilise käe sõrmedele paigaldada igale sõrmele 5 resistentset rõhuandurit (vt. Joonis 4.3). Need andurid võimaldavad mõõta rõhku ja jõudu. Enamikul anduritel on kas ümmargune või ristkülikukujuline mõõtetsoon. Ruudukujulise mõõtetsooniga andurid on head, kui vajate suurt mõõtetsooni. Neid andureid saab kasutada õhukeste objektide vastutõstmiseks. [25]



Joonis 4.3 FSR402 — resisttiivne rõhuandur [25].

Selleks, et mehaaniline käsi manipulaator saaks tõsta raskemaid esemeid, tuleb servomootorid välja vahetada võimsamate vastu, mis võimaldavad saavutada suure kandevõime.

4.2 Juhtmevaba mehaanilise kinda edasine arendamine

Kinda edasisel arendamisel saab andmete edastamiseks kasutada kiiremat moodulit, näiteks ESP8266 (vt. Joonis 4.2). Sellest on kirjutatud eelnevalt. Väärtuste täpsemaks mõõtmiseks on võimalik asendada potentsiomeetrid viie painutusanduriga (vt. Joonis 4.4). Mehaanilise painutamise kohta info saamiseks kasutatakse spetsiaalseid andureid õhukese pika takistusriba kujul. Selline andur muudab takistust sõltuvalt painutamise suurusest, st see muudab mehaanilise struktuurimuutuse elektriliseks takistuseks. Mida suurem on painutus, seda suurem on takistus. Teiste sõnadega on paindlikud andurid potentsiomeetritega sarnased komponendid, mis töötavad koostises muutuva pingejaguri abil. [23][24][26]



Joonis 4.4 Painumise andur [26].

Kasutades neid andureid, saate vältida probleeme lülidega, mis on seotud juhtmevaba mehaanilise kinda potentsiomeetrite ja otse efektori ühendamise, kuna on võimalus neid mitte kasutada. See andur on juhtimiseks ühendatud analoog sisendiga, nii et potentsiomeetri tuleb lahti ühendada ja seejärel on võimalik uued andureid külge ühendada. Võimalusel platvorm ümber ehitada, millele plaat koos mikrokontrolleriga on kinnitatud. Samasuguseis andureid võib paigaldada näiteks inimese enda sõrmele.

KOKKUVÕTE

Koostatud bakalaureusetöö põhines Tallinna Tehnikaülikooli elektrotehnika ja mehhatroonika instituudi poolt välja pakutud projektil. Bakalaureusetöö eesmärgiks oli mehaanilise käe manipuleerimine juhtmevaba mehaanilise kinda abil. Töö eesmärgini liikumise käigus tuli juhtmevaba mehaaniline kinnas kokku panna, muuta osa detaile sujuvama töö jaoks ja kirjutada programmikood.

Bakalaureusetöö esimeses osas kirjeldati juhtmevaba mehaanilise kinda komplekti ja mehaanilise käe manipulaatori komplekti. Tutvustati toodete komplektis saanud detaile, sh kooste detailid ja andurid.

Bakalaureusetöö teine osa esitab mehaanika-teemalise arenduse, selles on esitatud parandused antud komplektide jaoks. Näiteks juhtmevaba mehaanilise kinda puhul tuli välja töötada uued detailid, mis ühendavad inimese sõrme ja kinnast, et oleks võimalik käe surumine rusikasse. Samuti on arvatud otsene kinemaatika, et kontrollida kui madalale laskub viimane lüli mööda Y telge ja palju lüli nihkub mööda X telge. Manipulaatorile on välja töötatud uus mootorite kinnitus, mis võimaldab manipulaatori pööramist ja paindumist.

Kolmandas osas kirjeldatakse programmide kirjutamist kahe mikrokontrolleri jaoks, mis töötavad juhtmevaba mehaanilise kindaga ja mehaanilise käe manipulaatoriga. Lisaks on kirjeldatud kuidas filtreerida andurite väärtuseid, kui need pole stabiilsed. Samas osas asub ka juhend, mis kirjeldab, kuidas kasutada juhtmevaba mehaanilist kinnast ja mehaanilise käe manipulaatorit.

Neljandas osas on esitatud probleemid ja perspektiivid, mis tuleks tulevikus lahendada. Peale probleemide lahendamist võib lugeda juhtmevaba mehaanilise juhtimise kinnast ja mehaanilise käe manipulaatorit täiesti valminuks edasise kasutuse jaoks. Praegusel momendil mehaanilise käe manipulaator ei võimalda viimase liigendi painutamist. Selle lahendamiseks tuleb võtta ette täiendavad tegevused mikrokontrolleriga.

Vaatamata palju keerukamate ja täpsemate juhtmevaba mehaanilise kinda konstruktsioonide olemasolule, on antud komplektidel praktilise kasutuse potentsiaal olemas. Kuna mehaanilise käe manipulaator ja juhtmevaba mehaaniline kinnas pole piisavalt täpsed, siis on võimalik antud

komplektide kasutus ülesannete puhul, kus pole nõutud suur täpsus. Antud komplekte kasutatakse tulevikus Tallinna Tehnikaülikoolis robotkäe ja juhtmevaba manipuleerimise demonstreerimiseks.

SUMMARY

Consisted of four different parts, the idea of this thesis was formed by Tallinn University of Technology. The aim of this graduate work was to construct devices, improve the details and write a program for mechanical glove with wireless control for managing a mechanical glove manipulator

In the first part of this thesis, was described whole kit consisted of wireless control glove and a mechanical manipulation glove. For example, tools, details, sensors etc. Also, features were described, for full understanding of working process.

The second part made in the field of mechanics. Because of new improvements for available components, such as, developing new details for wireless control glove, which connect human finger and a glove for making a proper fist. Further, direct kinematics was calculated, to check how much the final effector descended along the Y-axis and moved along the X-axis. For the mechanical arm manipulator a new mount with motors was developed, it allowed the manipulator to rotate and bend.

The third part describes the process of making a program for two microcontrollers, which were working with a wireless control glove and a mechanical arm manipulator. In addition, there is a description of sensors data filtering, if their values are unstable. In the same section you can find a manual for using the wireless control glove and the mechanical arm manipulator.

In the fourth part it was described future opportunities for improving used sets. The thesis task could almost be considered solved, because most of the assigned tasks were completed. However, some unexpected issues appeared which could be resolved only in future perspectives. If solution is found the mechanical arm manipulator and the wireless control glove will be fully prepared for usage. At this moment, the mechanical arm manipulator, cannot be bent. As a solution microcontroller need to be changed. More on that described in further sections.

Besides the fact that, there are a lot of difficult and accurate constructions of wireless control glove and mechanical arm manipulator, these sets could be used for potential practice usage. Since mechanical arm manipulator and wireless control glove are not accurate enough, there is an opportunity to use these kits without high requirements. For example, moving objects in industrial

usage. Used sets could be further used in Tallinn University of Technology for demonstrating wireless technology of manipulating and its use in mechanic.

KASUTATUD KIRJANDUSE LOETELU

1. DIY Wearable Mechanical Glove Wireless Somatosensory Controller Exoskeleton RC Robot Hand Palm Finger Remote Control Robot palm
<https://ru.aliexpress.com/item/DIY-Wearable-Mechanical-Glove-Wireless-Somatosensory-Controller-Exoskeleton-RC-Robot-Hand-Palm-Finger-Remote-Control-Robot/32868941495.html?spm=a2g0v.10010108.1000016.1.4d2040a8Fj4W85&isOrigTitle=true>
(03.03.2019)
2. 5DOF Bionic robot claw/ gripper/ manipulator / holder/robot/car accessories/DIY
<https://ru.aliexpress.com/item/5DOF-Bionic-robot-claw-gripper-manipulator-holder-robot-car-accessories-DIY/32738664616.html?spm=a2g0v.10010108.1000016.1.778118dbigaOhg&isOrigTitle=true>
(03.03.2019)
3. Модуль HC-08 Bluetooth 4.0 BLE, чип CC2540, master+slave режимы
<https://compacttool.ru/viewtovar.php?id=1659>
(05.03.2019)
4. Bluetooth HC-08
<https://arduino.ua/prod1595-bluetooth-hc-08>
(10.03.2019)
5. Гироскоп + акселерометр GY-521 (MPU-6050)
<http://roboshop.spb.ru/GY-521-MPU-6050>
(22.03.2019)
6. Потенциометры и аналоговые пины
<https://www.youtube.com/watch?v=ypH3W8r41Cw&list=PL66U7r2iLGRZvrDyLKKYkbqpidvtdpPRE>
(22.03.2019)
7. SG90, Сервопривод [TowerPro]
<http://roboparts.ru/products/sg90-servoprivod-9g>
(22.03.2019)
8. Autodesk inventor
<https://www.autodesk.eu/>
(22.03.2019)

9. Кинематика
<http://ru.solverbook.com/spravochnik/mexanika/kinematika/>
(30.03.2019)
10. A novel exoskeleton robotic system for hand rehabilitation – Conceptualization to prototyping
<https://www.sciencedirect.com/science/article/pii/S0208521614000047#sec0055>
(03.04.2019)
11. A Low-Cost Smart Glove for Hand Functions Evaluation
<https://pdfs.semanticscholar.org/20c3/1875b5291fc7cf5b143b39435ac1210cf372.pdf>
(03.04.2019)
12. Кинематика: прямая и обратная задачи
<http://robocraft.ru/blog/mechanics/756.html>
(03.04.2019)
13. MATLAB Online
<https://matlab.mathworks.com/>
(03.04.2019)
14. Fritzing is an open-source initiative[2] to develop amateur or hobby CAD software for the design of electronics hardware
<http://fritzing.org/home/>
(03.04.2019)
15. K-POWER M1500
<https://www.tme.eu/en/details/kp-m1500/servo-motors/k-power/m1500/>
(15.04.2019)
16. Сервоприводы Ардуино SG90, MG995, MG996: схема подключения и управление
<https://arduinomaster.ru/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie/>
(15.04.2019)
17. Parallax Continuous Rotation Servo
<https://www.parallax.com/product/900-00008>
(15.04.2019)
18. Bluetooth HC-08. Master and Slave role
<https://www.youtube.com/watch?v=KYRJBPEYWy0&list=PL66U7r2iLGRZvrDyLKKYkbqpidvtdpPRE&index=9>
(17.04.2019)

19. ФИЛЬТР КАЛМАНА
http://avrproject.ru/publ/poleznaja_informacija/filtr_kalmana_bascom_avr/4-1-0-164
(17.04.2019)
20. Функция analogRead()
<http://arduino.ru/Reference/AnalogRead>
(24.04.2019)
21. VarSpeedServo.h
<https://github.com/netlabtoolkit/VarSpeedServo>
(24.04.2019)
22. ATmega2560
<https://www.microchip.com/wwwproducts/en/ATmega2560>
(01.05.2019)
23. Wi-Fi и Bluetooth в чем разница?
<https://myhelpit.ru/index.php?id=55>
(03.05.2019)
24. Прошивка esp8266 через Arduino IDE
<https://arduinomaster.ru/platy-arduino/proshivka-esp8266-через-arduino-ide/>
(03.05.2019)
25. FSR402 — резистивный датчик давления
<https://voltiq.ru/shop/fsr402-pressure-sensitive-resistor/>
(03.05.2019)
26. Arduino и датчик изгиба
http://digitrode.ru/computing-devices/mcu_cpu/716-arduino-i-datchik-izgiba.html
(03.05.2019)
27. Serial Port Plotter
https://github.com/CieNTi/serial_port_plotter
(10.05.2019)
28. Прямая и обратная кинематика
<http://l1va.ru/ru/forward-inverse-kinematics/>
(10.05.2019)
29. Bluetooth модуль HC-08
<https://www.mini-tech.com.ua/bluetooth-modul-hc-08>
(10.05.2019)

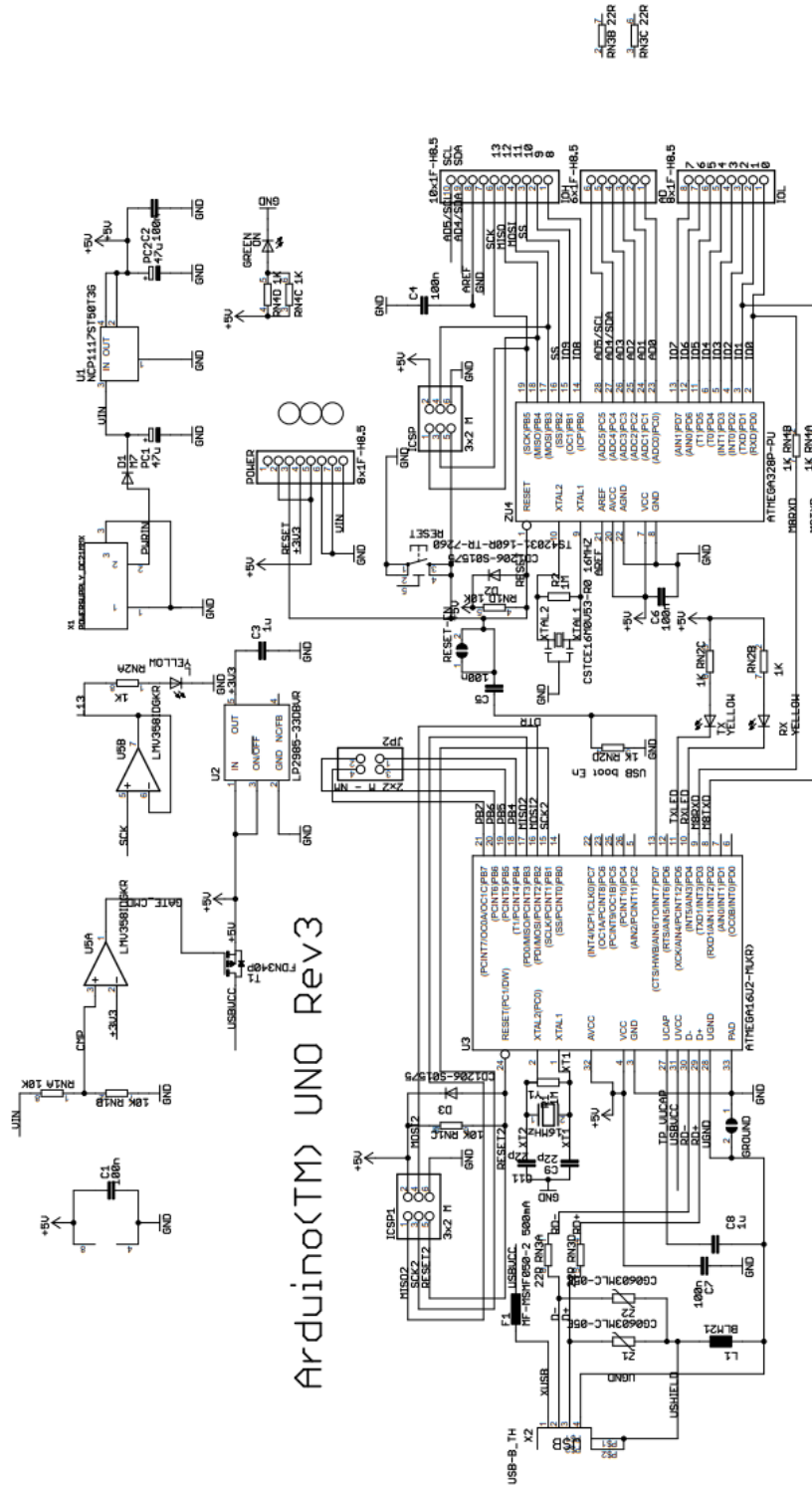
30. 3-Axis Gyro/Accelerometer IC - MPU-6050
<https://www.sparkfun.com/products/10937>
(15.05.2019)
31. Arduino boards: Using your laptop in the laboratory
http://faculty.jsd.claremont.edu/jmilton/Math_Lab_tool/Labs/Lab12.pdf
(15.05.2019)
32. KP-M1500
<http://www.rcdrive.ru/unit.php?unit=22229>
(15.05.2019)
33. WiFi ESP8266 в проектах Arduino
<https://arduinomaster.ru/platy-arduino/arduino-esp8266/>
(15.05.2019)
34. Diagram
<https://www.draw.io/>
(18.05.2019)

LISA 1 Poldid ja erinevad detailideks kinnitused

Accessories package (copper pillars & screws)							
JQ1002-06	L connecting piece		1	JQ1002-07	Front pull rod	 A:1 B:4	5
JQ1002-08	Lower pull rod	 A:1 B:4	5	JQ1002-09	Fixed support		5
JQ1002-10	Lower rod support	 A:1 B:4	5	JQ1002-11	M2*12 screw		8
JQ1002-12	M2*10 screw		20	JQ1002-13	M2*6 screw		6
JQ1002-14	M3*14 screw		16	JQ1002-15	5 mm copper bearing		16
JQ1002-16	M3 pad		20	JQ1002-17	M3 non slip nut		18
JQ1002-18	Cross recessed head		15	JQ1002-19	M2 pad		25
JQ1002-20	M2 non slip nut		20	JQ1002-21	M3*15 pillar		2
JQ1002-22	M3*10 screw		15	JQ1002-23	M3*9 screw		6
JQ1002-24	Fixed sleeve		10	JQ1002-25	Long spring		5
JQ1002-26	Short spring		5	JQ1002-27	Brass sliding sleeve		5
JQ1002-28	M2*6 screw		6	JQ1002-29	Swing arm		5
JQ1002-30	M2*4 screw		6	JQ1002-31	Pull rod		5
JQ1002-32	Spherical parts		5				

Joonis L1.1 Poldid ja erinevad detailideks kinnitused [2].

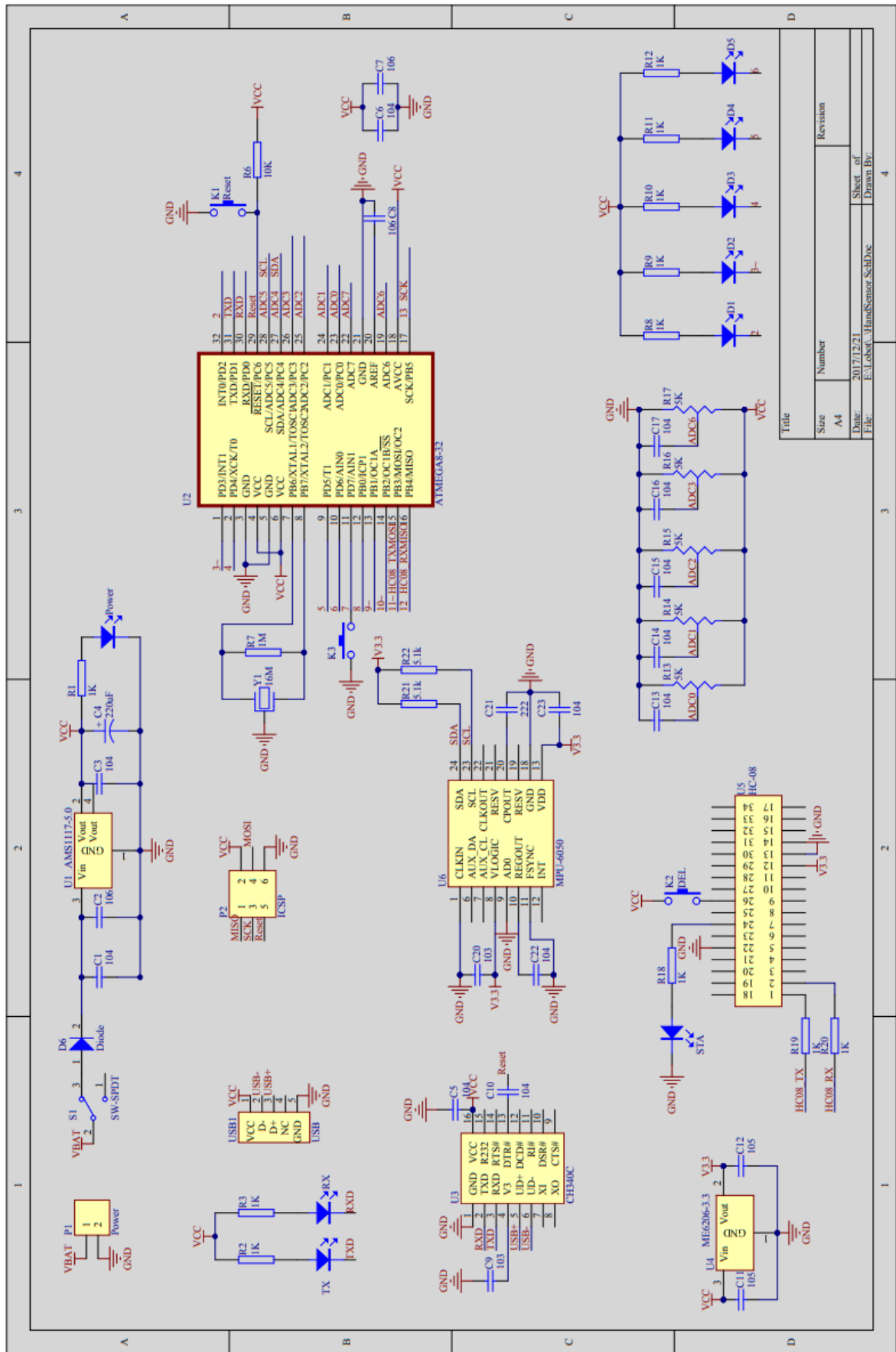
LISA 2 Arduino Uno juhtplaadi skemaatiline diagramm ja kinnastega juhtpaneeli skemaatiline diagramm



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

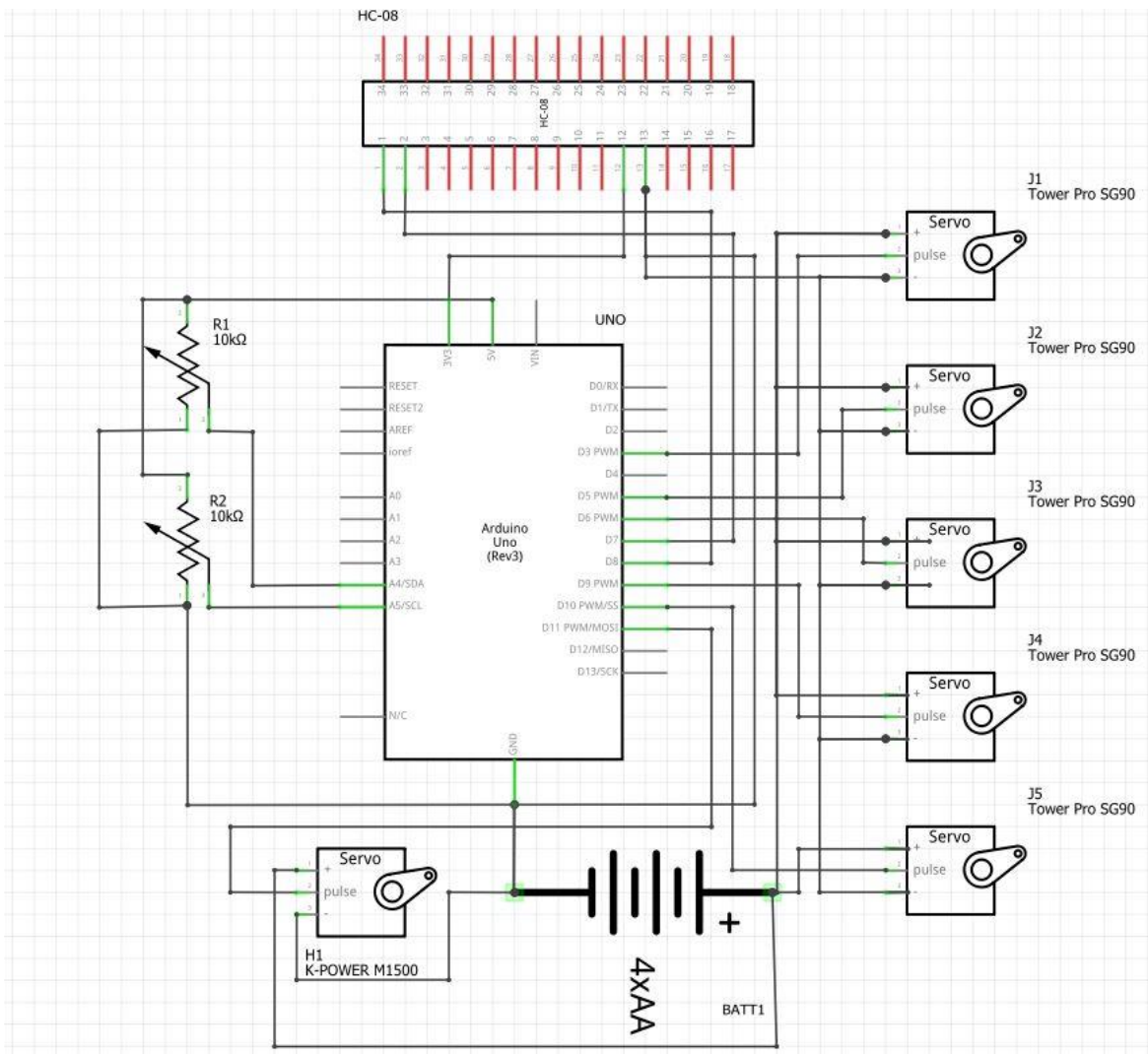
Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Joonis L2.1 Arduino Uno juhtplaadi skemaatiline diagramm [1].



Joonis L2.2 Kinnastega juhtpaneeli skemaatiline diagramm [1].

LISA 3 Mikrokontrolleri andurite ühendusskeem



Joonis L3.1 Mikrokontrolleri andurite ühendusskeem [14].

LISA 4 Bluetooth mooduli programm

```
#include <SoftwareSerial.h> // initsialiseerimine uus raamatukogu

SoftwareSerial BTSerial(8, 7); // TX | RX initsialiseerimine uus järjestikühenduse

void setup()
{
  Serial.begin(9600); // Määrab järjestikühenduse ja määrab andmeedastuskiiruse
  Serial.println("Enter AT commands:"); // see kasutatakse arusaamiseks kui on vaja alustada
  kirjutada AT komandid
  BTSerial.begin(9600); // Määrab järjestikühenduse ja määrab andmeedastuskiiruse Bluetooth
  mooduliseks
}

void loop()
{
  if (BTSerial.available()) // kui BTSerial jõudsin andmed Bluetooth mooduli kaudu

  Serial.write(BTSerial.read()); // kuvatakse kõik saadud väärtused

  if (Serial.available()) // kui Serial jõudsin andmed järjestikühenduse kaudu

  BTSerial.write(Serial.read()); // kuvatakse kõik saadud väärtused
}
```

LISA 6 Programm andmete saamiseks

```
void ReceiveData_MAX()
{
  BTSerial.begin(9600); // Määrab järjestikühenduse ja määrab andmeedastuskiiruse Bluetooth
  mooduliseks

  if (BTSerial.available() > 0) // kui BTSerial jõudsin andmed Bluetooth mooduli kaudu
  {
    val = BTSerial.read(); // Kirjutage väärtus muutujas

    if (val == 0x55) // kui esimese saanud täht on U, siis toimub tegevus
    {
      command_start = true;
    }
    else if (command_start == true)
    {
      if (index == 5) // kui index võrdub 5, siis kuvab kõik Serial pordi, lülitakse välja ja tehakse ülekäik
        järgmisse case.
      {
        index = 0;
        command_start = false;
        Serial.println("MAX");
        Serial.println(max_list[0]);
        Serial.println(max_list[1]);
        Serial.println(max_list[2]);
        Serial.println(max_list[3]);
        Serial.println(max_list[4]);
        i_step++;
      }
      else if (val != ' ') // kui ei võrdu tühikuga, siis alustatakse andmete töötlust
      {
        fullHex[fullHexIndex] = val; // võrdsustame muutuja saadud arvuga, alates algusest
        fullHexIndex = 0
      }
    }
  }
}
```

```

fullHexIndex++; // suurendame väärtust, et liikuda massiivi järgmisse ritta
if (fullHexIndex == 2 || val == '0') { // peale teise rea kirjutamist massiivis, saame, et
fullHexIndex == 2, alustatakse andmete töötlust
    max_list[index] = (int)strtol(fullHex, 0, 16); // massiivi rea võrdsustamine saadud muutuja
väärtusega ja tagasi kanda see väärtus int

    index++; // suurendame väärtust järgmise massiivi rea jaoks
    fullHexIndex = 0; // võrdsustame fullHexIndex nulliga, et olems võimalik uuesti lugeda andmed

    fullHex[0] = '\0'; // nullterminaator kasutatakse rea lõpu tähistamiseks
    fullHex[1] = '\0'; // nullterminaator kasutatakse rea lõpu tähistamiseks
}
}
else if (fullHexIndex != 0) // kui esineb tühik, siis FullHexIndeks on igal juhul võrdne nulliga ja
tühik jäetakse vahele. Kui ei ole võrdne nulliga, siis algab andmete töötlus
{
    max_list[index] = (int)strtol(fullHex, 0, 16); // massiivi rea võrdsustamine saadud muutuja
väärtusega ja tagasi kanda see väärtus int

    index++; // suurendame väärtust järgmise massiivi rea jaoks
    fullHexIndex = 0; // võrdsustame fullHexIndex nulliga, et olems võimalik uuesti lugeda andmed

    fullHex[0] = '\0'; // nullterminaator kasutatakse rea lõpu tähistamiseks
    fullHex[1] = '\0'; // nullterminaator kasutatakse rea lõpu tähistamiseks
}
}
}
}

```

LISA 7 Kalmani filtrit programm

```
void filter()
{
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // Väärtuste saamine güro-aktseleraatori
kohe
    An = ax; // Käesoleva mõõtmise algväärtus
    Mn = k * An + (1 - k) * MnP; // praeguse arvutuse tulemuse määr
    MnP = Mn; // eelmise arvutamise tulemuse väärtus
    return (Mn); // tagastamine väärtust
}
```

LISA 8 Programm andmete saatmiseks

```
void sendData_MIN()
{
  for (int i = 0; i < 5; i++) // massivi lugemine
  {
    min_list_list[i] = (char)min_list[i]; // teisendatakse väärtused char tüübis
  }

  BTSerial.write(0x55); // saadetakse väärtus U kontrollimiseks
  BTSerial.print(" "); // saadetakse tühik

  for (int i = 0; i < 5; i++) // massivi lugemine
  {
    BTSerial.print(min_list_list[i], HEX); // teisendatakse väärtused HEX binaarsüsteemis
    BTSerial.print(" "); // saadetakse tühik
  }
}
```

LISA 9 Matlab programm kinemaatika jaoks

```
// nurgad
fi1=35;
fi2=110;
fi3=40;

// liigese pikkus
L1=22;
L2=43;
L3=15;

// arvutamise valemid
XA=L1*cosd(fi1);
YA=L1*sind(fi1);

x1=L2*cosd(fi1+fi2);
y1=L2*sind(fi1+fi2);

x2=L3*cosd(fi1+fi2+fi3);
y2=L3*sind(fi1+fi2+fi3);

x=XA+x1+x2
y=YA+y1+y2
```