

TARTU ÜLIKOOL

Loodus- ja tehnoloogiateaduskond

Tehnoloogiainstituut

Sten-Oliver Salumaa

**NUTISEADMEGA JUHITAV PEEGELKAAMERA
LIIKUMIST KONTROLLIV SEADE**

Bakalaureusetöö (12 EAP)

Juhendaja: BSc Artur Abels

Tartu 2015

Sisukord

Tähised, lühendid ja definitsioonid	5
1. Sissejuhatus	6
1.1. Teema tutvustus.....	6
1.2. Töö eesmärk	7
2. Valdonna ülevaade	8
2.1. Turul olevad ettevõtted.....	8
2.1.1. Dynamic Perception	8
2.1.2. Emotimo	8
2.1.3. Syrp.....	8
3. Seadme üldine kirjeldus	9
4. Seadme üldise arhitektuuri projekteerimine.....	10
4.1. Mehaanika	10
4.1.1. Mootorid.....	10
4.1.2. Disain.....	10
4.2. Elektroonika	11
4.2.1. Aku	11
4.2.2. Pingeregulaator.....	11
4.2.3. Mikrokontroller	11
4.2.4. Bluetooth moodul	11
4.2.5. Lülitid liikumise katkestamiseks	11
4.3. Tarkvara.....	12
4.3.1. Telefonipoolne andmevahetus.....	12
4.3.2. Telefonipoolne kasutajasõbralikkus	12
4.3.3. Alguskoha seadmise võimalus	12

4.3.4. Samm-mootorite juhtimine.....	12
4.3.5. Liikumise täpsus ja kiirus.....	13
5. Mehaanika disain.....	14
5.1. Alumine moodul koos relssidega.	14
5.2. Relssidel veerev moodul.....	16
5.3. Vertikaalse pööramise moodul.....	17
6. Elektroonika disain.....	19
6.1. Mootorite kontrollkiibid.....	19
6.1.1. Sammu sooritamine.....	19
6.1.2. Voolupiiraja.....	20
6.2. Seadme liikumise piiramine.....	21
6.3. Suhtlus nutiseadmega.....	21
6.4. Trükkplaadi disain.....	21
7. Mikroprotsessori programm.....	23
7.1. Andmevahetus.....	23
7.2. Liikumine.....	25
7.2.1. Lineaarse liikumise kalkulatsioonid.....	26
7.2.2. Horisontaalteljelise pööramise kalkulatsioonid.....	26
7.2.3. Vertikaalteljelise pööramise kalkulatsioonid.....	27
7.2.4. Katkestuse sisu.....	27
8. Android operatsioonisüsteemil põhinev nutiseadme rakendus.....	29
8.1. Arenduskeskkond Android Studio.....	29
8.2. Bluetoothi seadistamine.....	30
8.3. Andmeplokkide seadistamine.....	30
8.4. Andmete edastamine.....	31
9. Kokkuvõte.....	33

10. Summary.....	34
11. Viidatud allikad	35
Lisad	36

Tähised, lühendid ja definitsioonid

Mikrokontroller (ingl microcontroller unit) – Ühe kiibi sisse ehitatud arvuti [1]

Kõrge ja madal olek – käesolevas töös käsitletud kui kiipide juhtimiseks kasutatav loogiline „0“ ja loogiline „1“. TTL-loogika puhul jäävad need sõltuvalt sisend- või väljundsignaalist vastavalt 0-0,8 V ja 2-5 V vahele. [2]

AVR – Atmeli poolt väljatöötatud mikroprotsessorite sari

UART (ingl Universal Asynchronous serial Receiver and Transmitter) – mikrokiip või selle osa, mis ühendub jadaliidese abil väliste seadmetega ning on võimeline nendega andmeid vahetama

USB (ingl Universal Serial Bus) - universaalne järjestiksiin [1]

Bluetooth – mobiilse raadioside spetsifikatsioon, mis on eelkõige levinud mobiiltelefonides ja sülearvutites [1]

PWM (ingl pulse width modulation) – modulatsiooni liik, kus väljundsignaali reguleeritakse väljundpinge impulsside laiusega

Time-lapse – videokunsti liik, kus pannakse üheks videoks kokku palju fotosid, mis on intervallidega pildistatud

API (ingl Application Programming Interface) - arvuti või mõne muu seadme operatsioonisüsteemiga või rakendusprogrammiga määratud reeglistik, mille alusel rakendusprogramm kasutab operatsioonisüsteemi või teise rakendusprogrammi teenuseid [1]

SDK (ingl Software Development Kit) - arendustarkvara programmipakett, mis võimaldab programmeerijal luua rakendusi konkreetsele platvormile

DC-DC muundur – seade, mis muudab alalisvoolu pinget

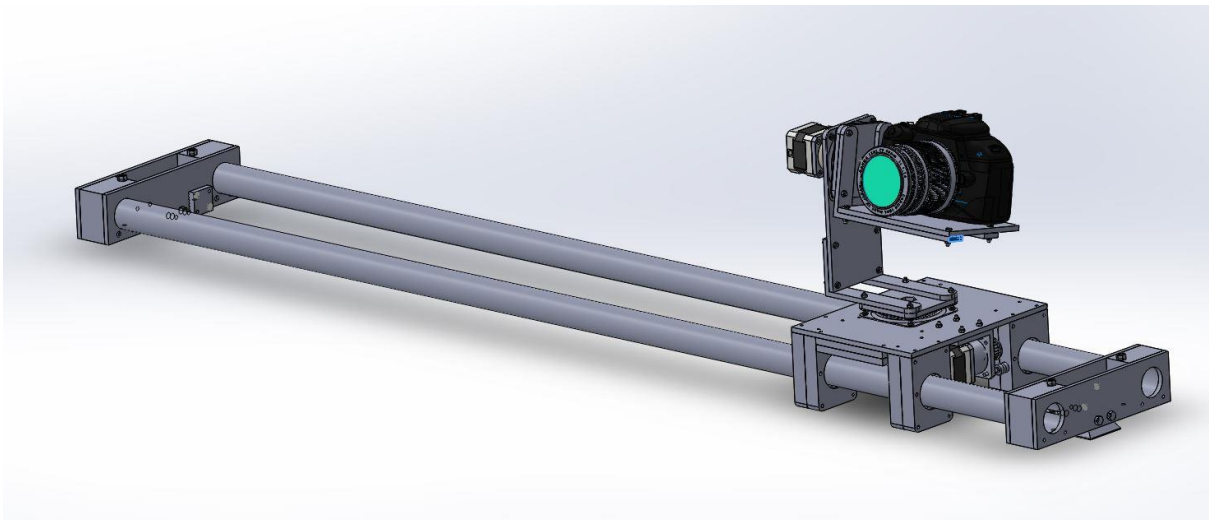
1. Sissejuhatus

1.1. Teema tutvustus

Time-lapse ehk madala võttesagedusega fotograafia on videožanr, kus mängitakse ette suur hulk sarivõttena tehtud fotosid. Mainitud video konstrueeritakse nõnda, et filmiks pannakse kokku fotod, mis on pika ajaperioodi jooksul tehtud ühest kohast. Fotode järjestikusel ettemängimisel on näha pildistatud paiga kiiret muutust ajas, näiteks pilvede liikumist või päikese loojumist.

Time-lapse fotograafia on tänapäeval muutumas üha populaarsemaks. Üheks selle põhjuseks on digifotograafia levik, mis on vähendanud *time-lapse* videote tegemisele kuluvat aega ja raha. Ühtlasi on odavaks muutunud kaugpäästikud, mida saab seadistada täitma sellist automaatprogrammi, mis sunnib kaamerat kindlate ajaintervallide järel pilti tegema. Lisaks on olemas digikaameratele laaditav kolmandate osapoolte tarkvara, mis suudab täita samu funktsioone nagu professionaalsed kaugpäästikud [3]. Selliste lisaseadmetega saab kaamera panna iseseisvalt pildistama.

Käesoleva töö raames valmib seade, mis võimaldab teha *time-lapse* videoid, mille käigus liigub ka kaamera (Joonis 1).



Joonis 1. Lõpliku toote üldine välimus

Kaamera saab liikuda relssidel edasi-tagasi ning pöörata end horisontaalselt ja vertikaalselt. Kaamera liikumine on võimalik eelseadistada nutitelefon rakendusega.

Kaamerat aeglaselt liikuma panevaid seadmeid on müügil vähe ning enamasti on need kallid ja vahel ka keerulised kasutada. Kui teha professionaalne tehnika kättesaadavamaks, on võimalik tõsta üldist *time-lapse* videote kvaliteeti ja selle alaga tegelevate inimeste arvu. Seadme saaks ehitada seni saadaval olevatest eksemplaridest kasutajasõbralikumaks ja soodsamaks, asendades masina juhtpuldi riistvara nutitefoniga. Nõnda on võimalik telefoniga ära seadistada kaamera liikumine ning saata andmed kaameraliigutajale, mis kõik ülejäänud teeb.

1.2. Töö eesmärk

Käesoleva töö eesmärgiks on valmistada kaamerat liigutav seade *time-lapse* fotograafia võimaluste laiendamiseks ning nende rakendamiseks reaalsetes olukordades. Projekti edukaks läbiviimiseks on tarvis põhjalikult tegeleda kolme valdkonnaga: mehaanika, elektroonika ning tarkvara. Tööga tegeledes asetatakse suurt rõhku modulaarsusele, seadme ohutusele ja detailide tootmise lihtsusele. Samuti on tähtis, et projekti lõpuks valmiks toode, mida on võimalik reprodutseerida mõistliku aja jooksul.

Töö lõpuks valmiv seade peab olema mobiilne ehk toimima akutoitel ning vastu pidama vähemalt kahetunnisele võttele. Seadme elektrivoolu katkemine ei tohi ohustada selle opereerijat ega kasutatavat tehnikat.

Kui toote intensiivse kasutamise tõttu peaks tulevikus mõni komponent vahetamist vajama, on tähtis, et kasutaja saaks seda lihtsate vahendite ja töövõtetega teha. Valmiv seade peab vastu pidama sagedasele transpordile ning kasutamisele.

Töö lõpuks valmiv seade peab olema juhitav telefonirakenduse kaudu, mis põhineb Android operatsioonisüsteemil.

2. Valdonna ülevaade

2.1. Turul olevad ettevõtted

Praegu on turul mõned ettevõtted, mis tegelevad *time-lapse* fotograafia abiseadmete valmistamisega.

2.1.1. Dynamic Perception

Ettevõtte, mis keskendub *time-lapse* fotograafia tarbeks liikumismehhanismide tootmise ja turustamisega. Ettevõtte müüb seadmeid, mis algkonfiguratsioonis suudavad maksimaalselt liikuda kahes teljes [4]. Võimalik on olemasolevat konfiguratsiooni täiendada ka kolmanda mootoriga, mille tulemusena võimaldatakse kaameral liikuda lineaarselt ning pöörata vertikaalselt ning horisontaalselt.

2.1.2. Emotimo

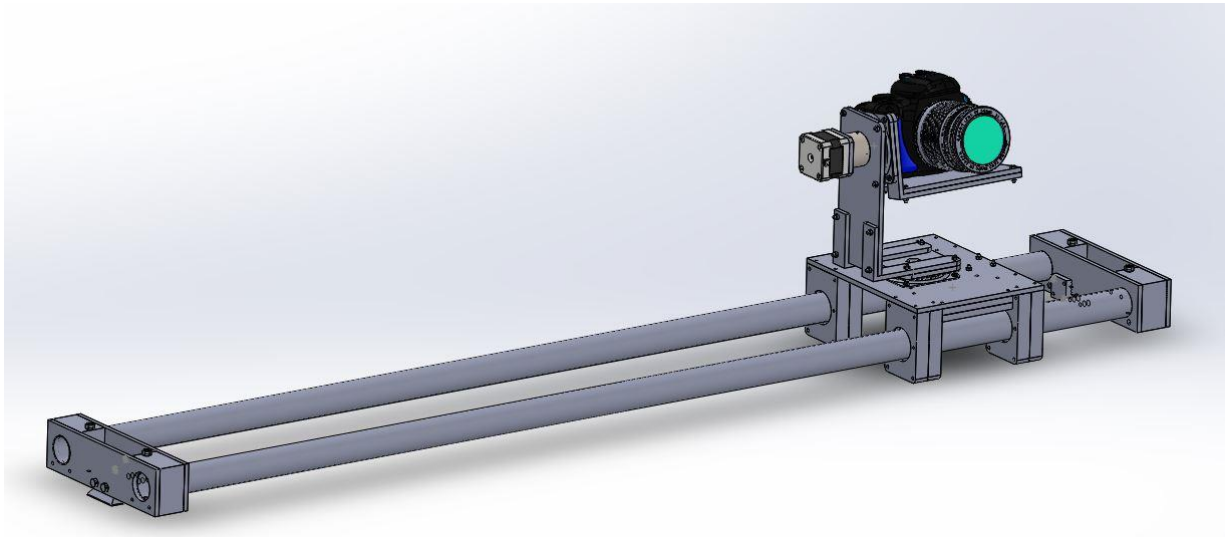
Nimetatud ettevõtte toodab sarnase eesmärgiga tooteid nagu Dynamic Perception [5]. Emotimo tootevalikus leidub ka seadmeid, mille liikumist saab kontrollida juhtmevabalt spetsiaalse puldiga. Firma müüb ka valmiskomplekte, mis võimaldavad kaamera liigutamist kolmes teljes.

2.1.3. Syrp

Syrpi toote Genie erilisus seisneb selles, et kogu süsteem paikneb ühes keskses kompaktses seadmes, mis on väga mitmekülgne ja laialt kasutatav [6]. Seade võimaldab liikuda küll kahes teljes (lineaarne liikumine ning horisontaalne pööramine), kuid jätab selle kasutajale palju võimalusi, mis vahenditega seda teha. Näiteks on võimalik lineaarne liikumine saavutada tavalise nõõri abil, mille seadmest peaks läbi vedama. Samuti on Genie ühilduv enamike turul olevate kaamerastatiividega.

3. Seadme üldine kirjeldus

Käesoleva töö raames valmis seade, mis võimaldab teha *time-lapse* videoid nõnda, et kaamera liigub samal ajal maksimaalselt kolmes teljes korraga.



Joonis 2. Valminud seadme 3D-joonis

Kogu kaamerat toetav moodul veereb kahe toru peal. Mooduli alumised detailid on kinnitatud nõnda, et torud jooksevad nende seest läbi. Kaamera pööramiseks horisontaal- ja vertikaaltelje suhtes on kasutatud tugilaagreid, millele kinnituvad kaamerat toetavad struktuurid.

Kõiki seadme liigutusi realiseerivad samm-mootorid, mida juhitakse läbi keskse elektroonikaplaadi. Elektroonikaplaadi külge on ühendatud kogu süsteemi elektriga varustav aku ning väline juhtmevaba suhtlust võimaldav Bluetooth moodul.

Seadme juhtimine toimub läbi nutitelefoni rakenduse, mis saadab andmed juhtmevabalt eelpool nimetatud Bluetooth moodulile.

4. Seadme üldise arhitektuuri projekteerimine

Seadme kavandamise käigus tuli vastu võtta mitmeid disaini puudutavaid otsuseid, mille tulemusena valmis käesolev töö. Järgnevalt kirjeldatakse nende tegemise põhilisi põhjuseid.

4.1. Mehaanika

4.1.1. Mootorid

Kuna käesoleva projekti eesmärgiks on teha seade, mis liigutaks sellel olevat pooleteisekilogrammist koormist väga aeglaselt (minimaalse kiirusega 0,25 m/h või 5 °/h pöörde puhul), sobivad seda ülesannet kõige paremini täitma samm-mootorid. Nende abil on võimalik liikuda täpselt nii pikk maa, kui vaja. Kuigi mootoritel puudub tagasiside süsteem, saab nendega teha täpseid liigutusi. Selleks, et mootor sammu vahele ei jäta, peab selle kontrollkiibi väljundvoolu reguleerima piisavalt kõrgele tasemele.

Kuna kogu seade peab töötama ka akutoitel, siis on parem, kui saaks mootorite võimsustarvet hoida võimalikult madalana. Samas peab nende väändemoment olema piisavalt tugev, et mitte samme vahele jätta. Samm-mootorite väändemomendi piisavalt kõrgel tasemel hoidmine vajab üsna palju energiat. Selleks, et mootorite võimsustarvet vähendada ning väändemomenti hoida piisavalt suurena, on mõistlik kasutada käigukastiga samm-mootoreid.

4.1.2. Disain

Time-lapse videoid tehes on tähtis sujuv kaamera liikumine, mistõttu peab kogu seade olema tugev ja mitte painduma olenemata kuni 1,5-kilogrammise kaamera ja 2-kilogrammise liikurmooduli asukohast relssidel. Sellise jäikuse saavutamiseks kasutatakse töös süsinikkiust torusid, mida toimivad relssidena, kus peal liigub kaameraalune moodul.

Kuna aparaadid, mis seadmepool liikuma hakkavad, on tihti väga väärtuslikud, peab seade olema piisavalt turvaline ka keerulistes oludes. Peamine moodul paikneb ümber süsiniktorude ega saa nende küljest ära kukkuda.

Selleks, et terviksüsteemi asetust ja kõrgust suures ulatuses reguleerida saaks ning see kogu aja stabiilsena püsiks, toetub alus teiste tootjate kvaliteetsetele kaamerastatiividele.

4.2. Elektroonika

4.2.1. Aku

Seadet liikuma panevad samm-mootorid võtavad üsna palju voolu, kuid samas on tähtis, et sellega saaks küllalt pikkasid võtteid teha. Minimaalseks piiriks on määratud kaks tundi kestvat liikumist. Liitium-polümeer akud on ühed kõige energiatihedamad ning seega aitavad ka seadme kaalu madalana hoida [7]. Projekti sai valitud 3-elementiline 11,1-voldine aku.

4.2.2. Pingeregulaator

Kuna seadme kontroll-loogikale ei sobi toiteks aku pinge, tuleb see madalamaks teha. Mobiilsel seadmel on tähtis väike voolutarve ning seetõttu on trükkplaadil kasutatud pinge alandamiseks ligikaudu 90% efektiivsusega DC-DC muundurit [8].

4.2.3. Mikrokontroller

Mikrokontrollerite osas on palju valikuid, aga valik langes ATmega32U4 kasuks. Protsessor sai valitud seetõttu, et

- sellel piisavalt taimereid, sealhulgas PWM-i võimalus
- sellel on antud projekti jaoks piisavalt sisend-väljund jalgasid
- selle programmeerimine USB kaudu on lihtne ja kiire
- töö tegijal on eelnev kogemus AVR-arhitektuuriga

4.2.4. Bluetooth moodul

Selleks, et säilitada trükkplaadi modulaarsus ja vähendada keerukust, võeti projektis Bluetoothi abil suhtlemiseks kasutusele väline HC-06 moodul. See on laialt kasutusel ning kasutajate poolt positiivset tagasisidet saanud seade, mis suudab Bluetooth protokolliga abil suhelda ning vastuvõetud andmed väljastada kujul, mis on UARTile arusaadav.

4.2.5. Lülitid liikumise katkestamiseks

Kuna seade ei tohi kahjustuste vältimiseks liiga suurtes ulatustes liikuda, on mootoritele vaja anda ka tagasisidet, kui need piirpositsioonini jõudma peaksid. Selleks on mikroprotsessoril

sisendid, mida muudavad füüsilised lülited iga piirjuhu korral. Nende andmete põhjal on võimalik otsustada, millise mootori peaks seiskama.

4.3. Tarkvara

4.3.1. Telefonipoolne andmevahetus

Mobiiltelefon peab suutma korraga kogu *time-lapse* võtte jaoks vajalikud andmed ära saata. Selleks kogutakse kõik kasutaja poolt sisestatud andmed lokaalselt kokku, et need üle Androidi Bluetooth ühenduse vastuvõtjamoodulisse edastada. Tööd lihtsustab Androidi põhjalikult dokumenteeritud Bluetoothi API.

4.3.2. Telefonipoolne kasutajasõbralikkus

Üks kogu töö eesmärke on teha *time-lapse*-seadmete kasutamine lihtsamaks. Kuna põhiline seadme juhtimine käib telefonirakenduse kaudu, siis on oluline, et kasutajale oleks see võimalikult intuitiivne ja silmasõbralik.

Rakendusesisesed alamprogrammid on rangelt nende funktsioonide kaupa osadeks jagatud, et kasutaja peaks keskenduma ühele asjale korraga.

Selleks, et juhtmevaba ühenduse probleemide tõttu rakendus ei sulguks, on implementeeritud erinevate vigade haldamine ja nendele vastavalt reageerimine. Samuti on valminud rakendus võimeline hoiatama ja tõkestama kasutajat, kui ta soovib Bluetooth moodulile saata andmeid, mis mikrokontrolleri loogikale või riistvarale vastuvõetamatud on. Nendeks on näiteks liiga suur andmete hulk või liiga suur liikumiskiirus.

4.3.3. Alguskoha seadmise võimalus

Selleks, et kasutaja saaks võtet alustada sellisest positsioonist nagu parasjagu vaja, on rakenduses enne *time-lapse*'i alustamist võimalik liigutada kaamera positsiooni. See laiendab seadme kasutusvaldkonda ka tavalise filmimiseni, kus on vaja sujuvat liikumist, aga kiiremini kui *time-lapse* videotel.

4.3.4. Samm-mootorite juhtimine

Kuna samm-mootori kontrollkiip vajab ühe sammu tegemiseks signaali tõusvat fronti, saab seda hõlpsasti genereerida taimerite katkestuste sees. Kasutataval mikrokontrolleril

ATmega32U4 on kokku neli taimerit, millest kolme katkestusi kasutatakse tõusvate frontide genereerimiseks vastavatel samm-mootorite kontrollkiipide jalgadel.

4.3.5. Liikumise täpsus ja kiirus

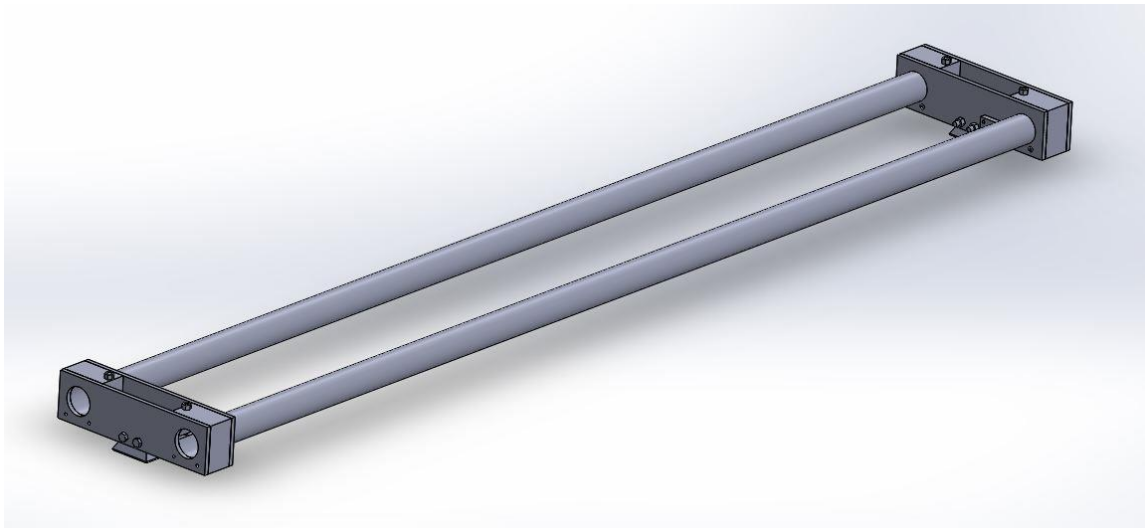
Võimalikult aeglase liikumiskiiruse saavutamiseks on kasutusel mootorite kontrollkiipide *microstepping* režiim, mis jagab iga mootori täissammu (ilma käigukastita 1,8 kraadi) kuueteistkümneks väiksemaks sammuks. Sellest režiimist kirjutatakse lähemalt kuuendas peatükis.

5. Mehaanika disain

Käesoleva seadme planeerimisel oli tähtis, et kogu projekt oleks modulaarne ning taastoodetav. Seadme kavandamisel on enamuse mehaanilisi komponente disainitud nõnda, et neid saaks 3D-printeriga toota. Mõnede suuremate tasapinnaliste detailide puhul on arvestatud seda, et neid saaks laseriga lõigata.

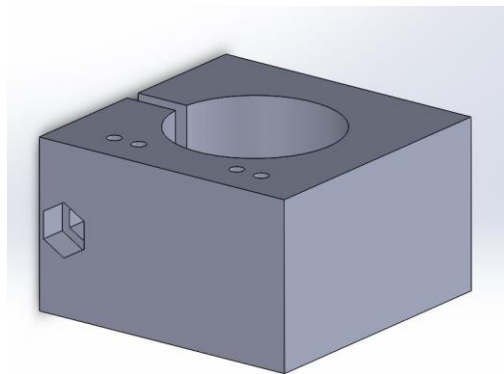
5.1. Alumine moodul koos relssidega.

Relssideks on kasutatud kahte 30 mm x 1 mm x 1200 mm süsinikkiust toru (Joonis 3). Süsinikkiust torud on kerged ning vastupidavad väändumisele ja paindumisele. Seetõttu sobivad need põhilise struktuurse tugevuse tagamiseks.



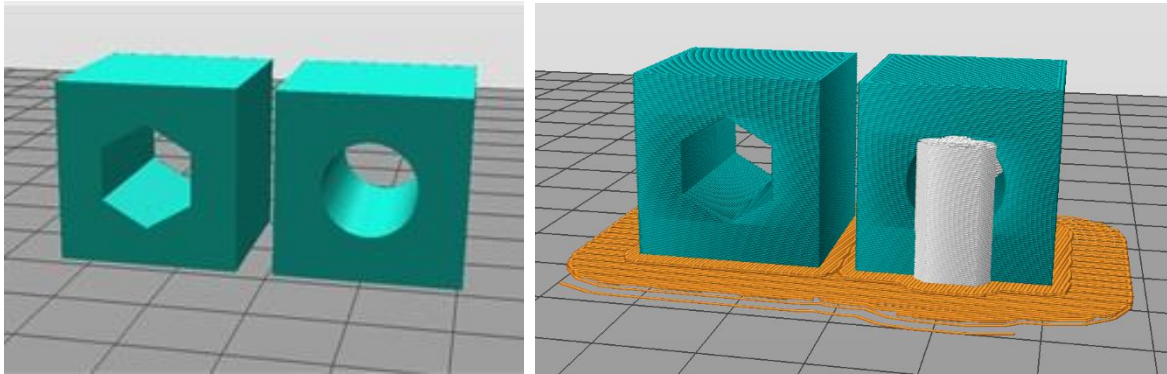
Joonis 3. Seadme mehaanika alumine moodul.

Selleks, et torud ei liiguks, kinnitatakse need spetsiaalsete detailide ja poltide abil otstest (Joonis 4).



Joonis 4. Torukinnitusdetail

3D-printeriga töötamisel peab arvestama, et see pole võimeline mõnda struktuuri korralikult printima. Kui proovida printida ringikujulise ristlõikega avausega detaili, peab arvestama, et olenevalt detaili asetusest võib printer proovida sellele juurde tekitada tugistruktuure (Joonis 5). Seevastu kuuskandi kujuga avause ülemiste servade printimisel tugistruktuure vaja pole. Kuuskandikujulisi avausi kohtab siin töös ka teiste 3D-prinditud detailide puhul.

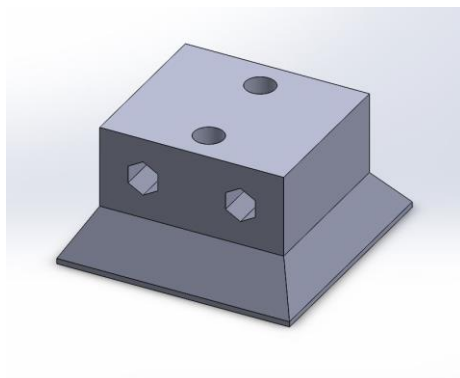


Joonis 5. Illustratsioon tugistruktuuridest. Vasakul on kahe näidisdetaili 3D-mudelid ning paremal simulatsioonid lõplikest 3D-prinditud detailidest. Kuvatõmmis tarkvarast Zortrax Z-Suite.

3D-prinditud detaile saab hõlpsasti kinnitada ka tavaliste puidukruvidega. Käesolevas töös on kõik selliste kruvide jaoks mõeldud augud 0,5 mm väiksema diameetriga kui kruvi enda laius. Tänu sellele saab kruvi keere end plastiku sisse kinnitada.

Torusid kinnitavaid detaile seovad omavahel POM-plastikust väljalõigatud detailid. Nimetatud komponendid on välja lõigatud laserlõikuri Full Spectrum Laser MLE-40 abil. POM ei kahjusta aurustumisel laseri läätse, aga selle lõikamisel peab meeles pidama, et sellest eralduvad kantserogeensed osakesed ning kandma peaks hingamisteid, silmi ja nahka kaitsvat varustust ning olema hästiventileeritud ruumis.

Seadme asukoha seadistamiseks maapinna suhtes kasutatakse kahte kaamerastatiivi, mille kinnitamiseks on kummaski seadme otsas spetsiaalne detail (Joonis 6).

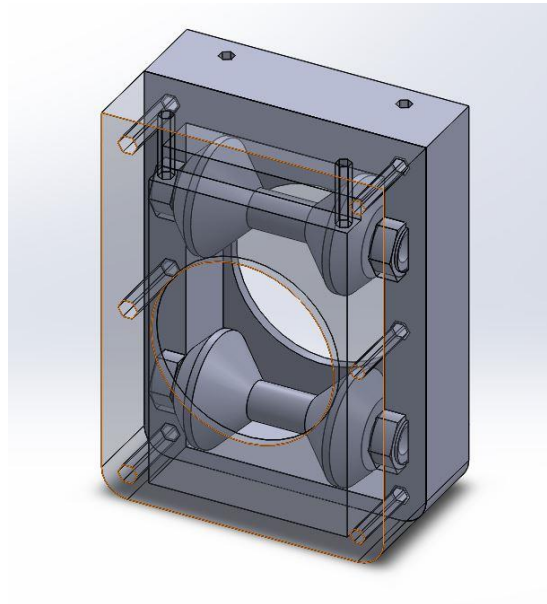


Joonis 6. Statiivi kinnitamise detail

5.2. Relssidel veerev moodul

Põhiline kandev roll on siin neljal alammoodulil, milles igaüks on kaks laagritele toetuvat rullikut (Joonis 7). Seadme mitmekülgsuse ja turvalisuse tagamiseks on rullikuid kaks. Liigse hõõrdumise vältimiseks on rullikud disainitud torusid puutuma kahes punktis.

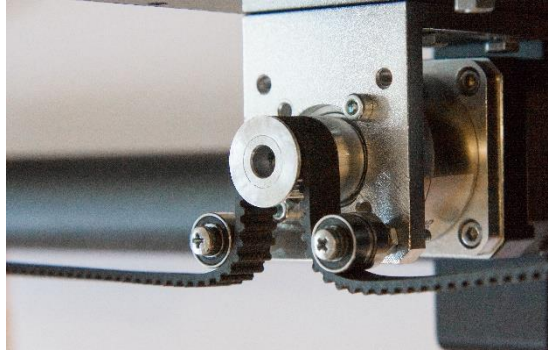
Rullikuid hoidev kest ümbritseb ka kogu sellest läbi minevat toru ning välistab torudest eemaldumise võimaluse. Samuti kaitseb kest rullikuid välise mustuse ja tolmu eest. Käesolevas töös valmivat seadet on võimalik kinnitada ka tagurpidi, nii et kaamerat hoidev kinnitus ripub maapinna poole. Nõnda saab videotegija vajadusel teha ka võtteid, kus kaamera liigub mingi tegevuse või objekti kohal.



Joonis 7. Rullikud koos ümbrisega

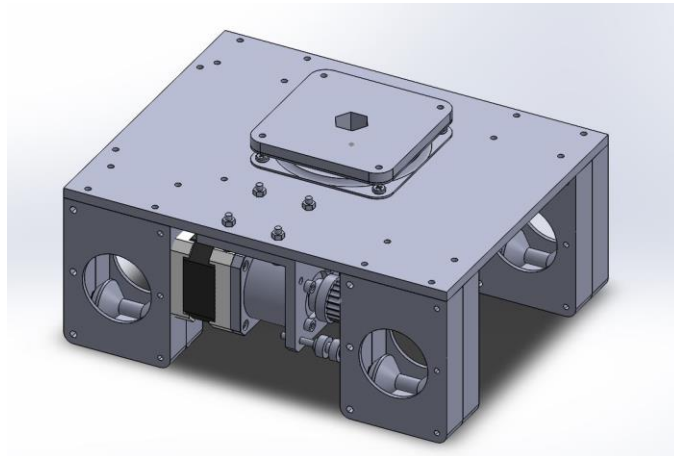
Veereva mooduli külge on kinnitatud kaks mootorit. Ühe abil toimub edasi-tagasi ehk lineaarne liikumine relssidel ning teine mootor vastutab horisontaalteljelise pööramise eest.

Lineaarne liikumine toimub seadme otstesse kinnitatud hammasrihma abil. Samm-mootori võllile kinnitatud hammasrattas pöörleb vastu rihma ning tänu sellele liigub kogu moodul edasi. Selleks, et vältida rihma libisemist, kasutatakse rihma kokkupuuteala suurendamiseks hammasrattaga rihmapinguteid, mis sunnivad hammasrihma igal ajal kokku puutuma vähemalt poole hammasratta välispinnaga (Joonis 8).



Joonis 8. Rihmaratas ja rihm koos pingutitega

Horisontaaltasandil pööramiseks on kasutatud tugilaagrit. Selle alumine pool on kinnitatud veereva mooduli külge ning ülemine pool pöörleb selle suhtes vabalt (Joonis 9). Mootori võll on vertikaalteljes vabalt liikuvalt kinnitatud laagrisüsteemi ülemise osa külge. Tänu sellele langeb kogu ülemise poole raskus laagrikuulidele ning mootor tegeleb ainult pööramisega. Kui seadme kasutaja ei soovi vertikaalteljes pöörämist kasutada, on võimalik vertikaalmooduli asemele panna kaamerastatiivi pea ning selle külge kinnitada kaamera ise. Sellisel juhul on seade kompaktsem ning võtab vähem voolu.



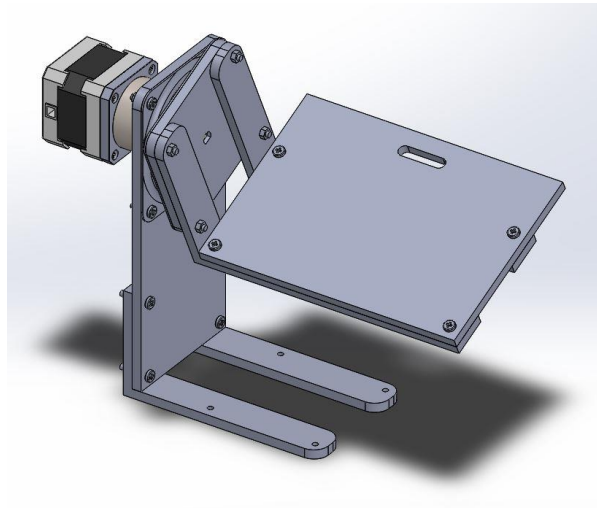
Joonis 9. Veerev moodul

5.3. Vertikaalse pööramise moodul

Vertikaalse pööramise mooduli püstine osa on veerevast moodulist alumiiniumlehtede abil kaugemale viidud selleks, et sinna peale hiljem asetatud kaamera raskus langeks võimalikult täpselt veereva mooduli tasakaalupunkti keskele (Joonis 10). See ühtlustab koormust rullikutele ning pikendab seeläbi seadme eluiga. Kaamera vertikaalses suunas pööramine toimub

analoogselt horisontaalsega. Kasutatud on sama põhimõttega võlliühendust ning sama tugilaagrit.

Plaat, kuhu kinnitub peegelkaamera, on ühes otsas pikemaks tehtud, sest on arvestatud objektiivi kaaluga. Ideaalsel juhul on kaamera ja objektiivi massikeske samal teljel vertikaalselt pöörava mootori võllile väiksem koormus ja selle liigutamiseks võib kasutada väiksemat voolutugevust.



Joonis 10. Vertikaalse pööramise moodul

6. Elektroonika disain

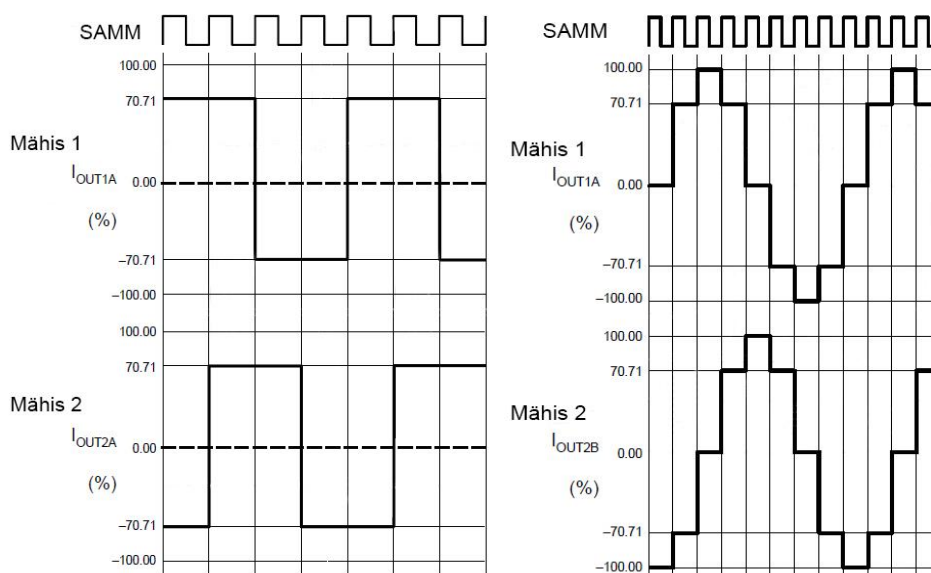
6.1. Mootorite kontrollkiibid

Samm-mootorite juhtimiseks on valitud Allegro A4982 kiip. Selle põhilised omadused on:

- PWM-iga juhitud voolupiiraja
- Sammurežiimid: täissamm, poolsamm, neljandiksamm, kuuteistkümnendiksamm
- Kontrollitav mootoritüüp: bipolaarne samm-mootor
- Maksimaalne väljundvool: 2 A
- Loogika toitepinge vahemik: 3 V kuni 5,5 V
- Sammu suuna muutmise võimalus sisendi abil

6.1.1. Sammu sooritamine

Allegro A4982 võimaldab valida nelja erineva sammurežiimi vahel: täissamm, poolsamm, neljandiksamm ja kuuteistkümnendiksamm ehk mikrosamm. Täissammu režiimi korral liigub mootori võll ühe STEP-sisendisse saadud tõusva signaalifrondi korral kõige rohkem ning mikrosammu korral kõige vähem.



Joonis 11. Täissammu (vasakul) ja poolsammu (paremal) režiimide erinevate voolunivoode illustatsioon [9]. Mõlema joonisel oleva sammurežiimi puhul on käsil liikumine päripäeva. Joonis 1/16 sammurežiimi kohta asub töö lõpus lisade sektsioonis Lisa 2 all.

Sisuliselt erinevad sammurežiimid selle poolest, et mitu erinevat voolutugevusnivood suudab kontrollkiip samm-mootori mähistesse anda (Joonis 10). Täissammu režiimi korral on erinevate voolutugevuste kombinatsioone kahe mähise peale kokku neli, mikrosammu korral 64. [9]

Käesolevas töös on igal seadme liikumise teljel kasutusel mikrosammu režiim. Tänu sellele on iga samm väike ning vähendab seadmele kinnitatud kaamera värisemist. Samuti suurendab see videovõtte sujuvust, sest iga kaadri vaheline liikumise hulk on väike. Selleks, et määrata sammurežiimi, on samm-mootori kontrollkiipide sisendid MS1 ja MS2 ühendatud mikrokontrolleri sisend-väljund jalgadega, mille signaale saab tarkvaraliselt reguleerida.

Reaalne samm-mootori samm sooritatakse siis, kui kontrollkiip saab enda STEP-sisendisse tõusva pingefrondi. Toimiva sammu suund on määratud sisendi DIR väärtusega. Kui DIR on kõrge, siis liigub mootori võll päripäeva, vastasel juhul vastupäeva.

6.1.2. Voolupiiraja

Samm-mootori kontrollkiibil on neli klemmi, mis ühendatakse mootori kahe mähisega. Mootori mähistesse lastakse vastavalt hetke sammupositsioonile sisse teatud tugevusega vool (Lisa 2). Kuna mootori mähiste puhul on sisuliselt tegu induktiivpoolidega, ei ole voolutugevuse tõus hetkeline. Voolutugevus tõuseb kuni kontrollkiibi poolt lubatud väärtuseni, siis lülitatakse voolu juurdevool kontrollkiibi poolt välja ning voolutugevusel lastakse teatud aja jooksul langeda. Kui see aeg on läbi, suunatakse mootori mähistesse uuesti vool sisse. Seetõttu toimib samm-mootori kontrollkiip kui DC-DC muundur, mis induktiivpoolina kasutab mootori mähist.

Samm-mootorite kontrollkiipide voolupiirajat kontrollib V_{ref} signaal. Mida suurem on pinge kontrollkiibi V_{ref} jalal, seda suurem on maksimaalne lubatav voolutugevus mootori mähises. Väljundvoolu piiramisväärtust saab arvutada järgneva valemi abil:

$$I_{TripMax} = V_{Ref}/(8 * R_s) \quad [9]$$

Sealjuures $I_{TripMax}$ on maksimaalne voolutugevus, mida lastakse ühe samm-mootori ühest mähisest läbi ning R_s on mähistest tuleva voolu mõõtmise otstarbega takistite väärtus. Antud juhul on takistite väärtus $0,1\Omega$.

Kui samm-mootorit läbiv voolutugevus jõuab käsilolevale mikrosammule seatud piirväärtuseni (Lisa 2), katkestab kontrollkiip voolu edastamise ning laseb mõnda aega mootori mähisest voolul väljuda. Kasutusel oleva kontrollkiibi puhul saab selle aja välja arvutada järgmise valemiga (tulemus mikrosekundites):

$$t_{OFF} \approx R_{OSC} / 825$$

Antud kontrollkiibi väljundvoolu piirajat on võimalik muuta mikrokontrollerist pärineva PWM-signaaliga. Kontrollkiibi sisend V_{ref} on võrdne PWM-signaali täituvusprotsendi ning mikrokontrolleri toitepinge korrutisega.

6.2. Seadme liikumise piiramine

Selleks, et vältida seadme mehaanilisi kahjustusi ja mootorite toitejuhtmete väänamisi, on veerevale moodulile lisatud piirjuhtudele reageerivad lülitid. Kui veerev moodul peaks jõudma piirasendini, millest edasi liikuda ei tohi, keelatakse vastavas suunas liikuva mootori edasised sammud. Piirjuhtudele reageerivaid lüliteid saaks kasutada ka seadme nullpositsiooni määramiseks, aga seda pole käesolevas töös tehtud, kuna kasutajale on jäetud võimalus rakendusega ise alguspunkt defineerida.

6.3. Suhtlus nutiseadmega

Seadme juhtmevabaks suhtluseks on kasutusele võetud eraldiseisev HC-06 Bluetoothi moodul. Mooduli ja mikrokontrolleri vaheline suhtlus käib läbi UARTi, mille jaoks on plaadil TX ja RX rajad.

6.4. Trükkplaadi disain

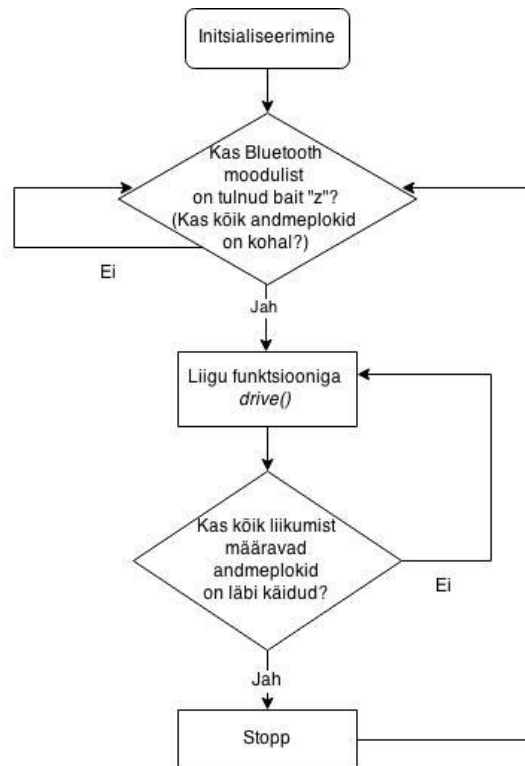
Trükkplaadi disainides on silmas peetud hilisemat kooslust koos seadme mehaanilise osaga. Selleks on kõik mootorite mähiste ning voluklemmid viidud ühele trükkplaadi küljele (Lisa 1). Nõnda ei jää nähtavale kohale ka elektroonika detaile.

Muuhulgas on trükkplaadi disaini koostamisel silmas peetud erinevaid volutugevusi, mis sellel olevaid vaseradasid läbima hakkavad. Mootoreid ja nende kontrollkiipe varustavad rajad on tehtud 1 millimeetri laiuste voluradadega ning signaalirajad on enamasti 0,3-millimeetrised. Voolurajad samm-mootorite kontrollkiipidest mootorite ühendusklemmideni on piisavalt laiad, et vältida lisatakistusest tulenevat soojust. Samuti on need rajad hoitud võimalikult lühikestena, et hoida nendest pärinev suurte pingemuutuste tõttu tekkiv müra võimalikult väikesena.

Trükkplaadi alumisel küljel on kõik rajad tehtud võimalikult lühikesed, et vältida alumise maa-ala liigset killustumist. Tänu sellele on kogu elektroonika vähem vastuvõtlik segavatele elektromagnetlainetele ning suudab paremini ära hajutada soojust, mis skeemi töö jooksul tekib.

7. Mikroprotsessori programm

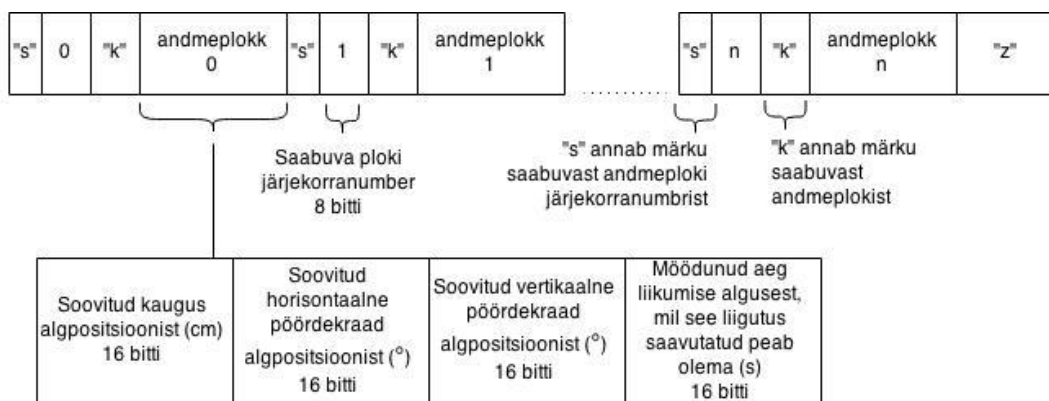
Mikroprotsessori enamus koodi on jagatud kahte suurde plokki: andmevahetus ja seadme liigutamine (Joonis 12).



Joonis 12. Seadme peamise loogika plokk skeem

7.1. Andmevahetus

Andmepakettide edastamiseks ja nende identifitseerimiseks nutiseadmelt on välja töötatud protokoll. Kui kõik liikumispunktid on nutiseadme rakenduses ära määratud ning kasutaja vajutab nuppu „Start time-lapse“, saadetakse üle Bluetoothi järgnev baidijada:



Joonis 13. Võtte alustamiseks saadetav baidijada

Kasutajal on pärast seadme sisselülitamist ning enne võtte algust võimalik seadistada kaamera positsiooni. Pärast rakenduse ekraanil noolenupu vajutamist ning ka vajutuse lõpetamist saadetakse seadme poole teele järgneva struktuuriga baidijada:

"m"	Mootori indeks n 8 bitti	Kas mootori n taimer peaks sisse või välja lülitama? 8 bitti	Liikumiskiirus (% mootori n maksimaalsest kiirusest) 8 bitti	Mootori n suund ("1" liigutab päripäeva, "0" vastupäeva) 8 bitti
-----	--------------------------------	---	--	---

Joonis 14. Seadme algpositsiooni muutmist käivitav baidijada

Nende andmete põhjal pannakse vastav samm-mootor vajaliku kiirusega samme tegema ning enne nutiseadme poolt tuleva peatava käsu saabumist mootor seisma ei jää. Mootorite liigutamise koodi põhimõtteid tutvustatakse hiljem.

Kõigi andmete vastuvõtu ja nendele reageerimisega tegeleb funktsioon *receiveAllData()*.

```
void receiveAllData(){
    uint8_t firstByte = 0;

    while(firstByte != 'z'){           //'start' käsk ei saabunud
        firstByte = USARTgetLetter();
        uint8_t secondByte;
        if (firstByte == 'k'){         //'järgmisena saabub andmepakett liikumise kohta
            keyframes[keyFrameReadIndex] = readKeyframe();
        }else if(firstByte == 's'){    //'järgmisena saabub järgmisena saabuva andmepaketi järjekorranumber
            secondByte = USARTgetLetter();
            keyFrameReadIndex = secondByte;
        }else if(firstByte == 'm'){   //'muudetakse liikumise alustamise stardipositsiooni
            uint32_t receivedMoveData = 0;
            receivedMoveData = USARTgetDoubleWord();
            moveDevice(receivedMoveData);
        }
    }
}
```

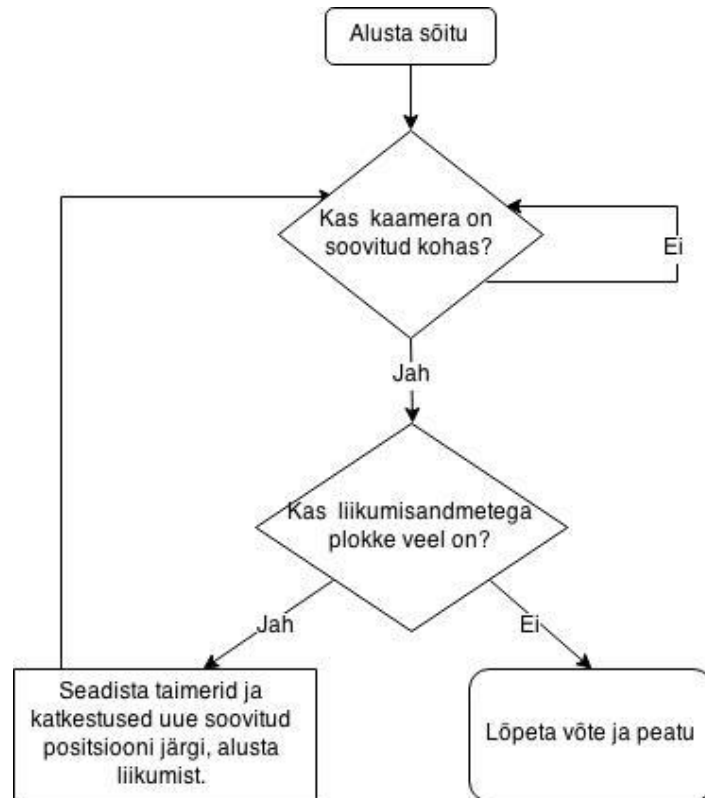
Joonis 15. C kood andmete vastuvõtmiseks

Kui funktsioon *receiveAllData()* peaks vastu võtma tähemärgi „z“, lõpetatakse *while*-tsükkel ning liigutakse edasi järgmise funktsioonini, mis alustab vastuvõetud andmetele tuginedes seadmel paiknevate mootorite liigutamist.

Kogu vastuvõetav informatsioon käib läbi protsessori UARTi RX-sisendi. Iga baidi puhul oodatakse, kuni protsessoris olev lipp RXC1 läheb kõrgeks. Kui nii juhtub, loetakse UARTi vastuvõturegistrist UDR1 väärtus kohalikku muutujasse. Viimane käik langetab automaatselt ka RXC1 lipu UCSR1A registrist.

7.2. Liikumine

Pärast andmete vastuvõtmist liigutakse koodi täitmises funktsioonini *drive()* (Joonis 16). Selles funktsioonis tegeletakse andmetele vastavate liigutuste realiseerimisega füüsilises maailmas.



Joonis 16. Funktsiooni *drive()* plokk skeem.

Iga andmekaadri puhul on teada seadme soovitud asukoht kolmes teljes algpositsiooni suhtes. Antud on kaugus algpositsioonist sentimeetrites ning horisontaalne ja vertikaalne pöördenurk kraadides. Samuti on programmi mälus kontsant, mis näitab, mitu mikrosammu peab tegema, et mootori võll teeks ühe täispöörde, ning konstant, mis näitab, kui suur on seadet edasi-tagasi liigutava mootori võllile kinnitatud hammasratta ümberringjoon.

Pärast kõikide katkestusi genereerivate taimerite seadistamist jälgib algoritm pidevalt ka hetke asukohta. Hetke asukoht samm-mootori sammude arvestuses salvestatakse kohalikesse märgiga täisarvmuutujatesse. Iga sammu puhul suurendatakse või vähendatakse seda muutujat vastavalt sammu suunale ehk mootorikontrolleri DIR-signaali tasemele. Kui hetke asukoht läheb üle vastuvõetud andmetes soovitud piiri, võetakse ette järgmine liikumisandmeid sisaldav plokk. Kui kaks järjestikust plokki ühe mootori kohta sisaldavad samu kaugusi, saab järeldada, et kasutaja on soovinud selle mootori liikumise seisata. Sellisel puhul keelatakse vastaval

taimeril katkestuste genereerimine, kuni saabub uus plokk, mis nõuab selles teljes asukoha muutmist.

Kui kõik plokid on otsas, võib võtte lõppenuks lugeda ning kõikide taimerite katkestused keelatakse, mille tõttu seade peatub.

7.2.1. Lineaarse liikumise kalkulatsioonid

Vastuvõetud andmete hulgast võetakse välja soovitud kaugus algpositsioonist ning selle ajahetk võtte algusest. Kõigepealt arvutatakse kaugus sentimeetrites kohalike konstantide abil ümber kauguseks sammudes. Kui on teada, et hammasratta ühe täispöörde tegemisel liigutakse edasi 4,8 cm, siis 48 cm läbimiseks on vaja teha 10 täispööret. Samuti on teada, et iga täispöörde tegemiseks on tarvis 16576 kuueteistkümnendiksammu. Arvutustele tuginedes saab konkreetse näite puhul teada, et algpositsioonist alates oleks vaja liikuda 165760 sammu. Sellest väärtusest lahutatakse seadme hetkeasukoht ning saadakse teada vajalik kaugus praegusest positsioonist. Kui eelnev tulemus oli negatiivne, siis muudetakse mootori suunda DIR-signaaliga ning iga järgneva sammu puhul vähendatakse kohalikku hetkekaugust sisaldavat muutujat.

Kui vajalik sammude arv on teada, arvutatakse vajalik taimeri võrdlusregistri väärtus. Taimer genereerib katkestuse iga kord, kui selle väärtuseni jõuab.

Kui soovitud katkestuse sagedus on väiksem kui 244 Hz, ei piisa nende haldamiseks taimeri sisemisest registrist. Selleks, et suurendada sammu sooritamise sageduste võimalusi, on kasutusel lisamuutujad, mida vajadusel iga katkestuse ajal suurendatakse ning samm sooritatakse alles siis, kui lisamuutuja jõuab vajaliku väärtuseni. Tänu katkestuste lisamuutujatele jääb *teoreetiline* sammusagedus umbes 500 kHz kuni 0,0037 Hz vahele. Lisamuutujad võetakse kasutusele ainult siis, kui taimeri enda loendurist ei piisa.

Edasi-tagasi liikumise puhul on kasutusel ATmega32U4 16-bitine taimer 1.

7.2.2. Horisontaalteljelise pööramise kalkulatsioonid

Kuna selles teljes on kasutusel samade omadustega taimer nagu lineaarses liikumises, on suur osa arvutamiskäike samad. Samuti on analoogne lisamuutujate ja DIR-signaalide tarvitamine. Ainuke erinevus tuleb sisse esialgsete vajalike sammude hulga arvutamisel.

Vastu võetud infos on soovitud positsioon kraadides. Kui on tarvis liigutada näiteks 10 kraadi, siis arvutatakse vajalike sammude arv välja täispöörde puhul teada olevast sammude arvust.

Nagu edasi-tagasi liikumise puhul, on ka horisontaalteljelise pööramise eest hoolitsevate katkestuste generaatoriks 16-bitine taimer.

7.2.3. Vertikaalteljelise pööramise kalkulatsioonid

Vertikaalse pööramise arvutused toimuvad analoogselt horisontaalse pööramise omadele. Erinev on aga see, et vertikaalse pööramise mootorit liigutavad katkestused genereeritakse 8-bitise taimeri poolt, sest protsessoril on pakkuda kokku vaid kaks 16-bitist taimerit. Selleks et antud taimeri katkestuste sagedus tuua samasugustesse piiridesse nagu 16-bitistel taimeritel, on vertikaalteljelise pööramise katkestustel kasutusel taktide eeljagamine. Kui protsessor genereerib 1024 takti, suureneb kõnealuse taimeri loendurregister TCNT0 ühe võrra. Tänu eeljagamisele kulub selle taimeri loenduri maksimaalse väärtuse saavutamiseks sama palju aega kui 16-bitiste taimerite puhul. Sellise käigu puhul väheneb aga kiirete sammusageduste poolne spekter. Kuna aga vertikaalteljeline pööramine polegi mõeldud väga kiireteks liigutusteks, pole sellel tähtsust.

7.2.4. Katkestuse sisu

Iga katkestuse puhul vaadatakse kõigepealt üle, kas kasutusel on lisamuutujad katkestuse lugemiseks (Joonis 17). Kui on, siis kontrollitakse selle vastavust nõutud loenduri väärtusele. Kui väärtus veel piisavalt suur pole, suurendatakse loendurit. Kui aga on, siis nullitakse loenduri väärtus ning sooritatakse üks samm. Selle jaoks saadetakse mootorikontrolleri kiibi STEP-jalale kõrge signaal, mis kohe uuesti nullitakse. Samuti kontrollitakse parasjagu käsil olevat liikumise suunda ning vastavalt sellele suurendatakse või vähendatakse kohalikku lineaarset kaugust algpositsioonist meeles pidavat muutujat ühe võrra.

Kui aga katkestuste loendamiseks pole lisamuutujat kasutusele võetud, sooritatakse mootori samm kohe ning märgitakse asukoha muutus kohalikku muutujas.

```

SIGNAL(TIMER1_COMPB_vect){
  if(distTimerTOCounterFlag){
    if(distTimerTOCounter < (distTimerTOCounterMax-1)){
      distTimerTOCounter++;
    }else{
      distTimerTOCounter = 0;
      PORTB = PORTB^(1<<PB6);
      PORTB = PORTB^(1<<PB6);
      // mis suunas liigume?
      if(distDirFlag){
        curDist++;
      }else{
        curDist --;
      }
    }
  }
}
else{
  PORTB = PORTB^(1<<PB6);
  PORTB = PORTB^(1<<PB6);
  // mis suunas liigume?
  if(distDirFlag){
    curDist++;
  }else{
    curDist --;
  }
}
}
}

```

Joonis 17. Edasi-tagasi liikumise eest vastutava taimeri katkestus

8. Android operatsioonisüsteemil põhinev nutiseadme rakendus

8.1. Arenduskeskkond Android Studio

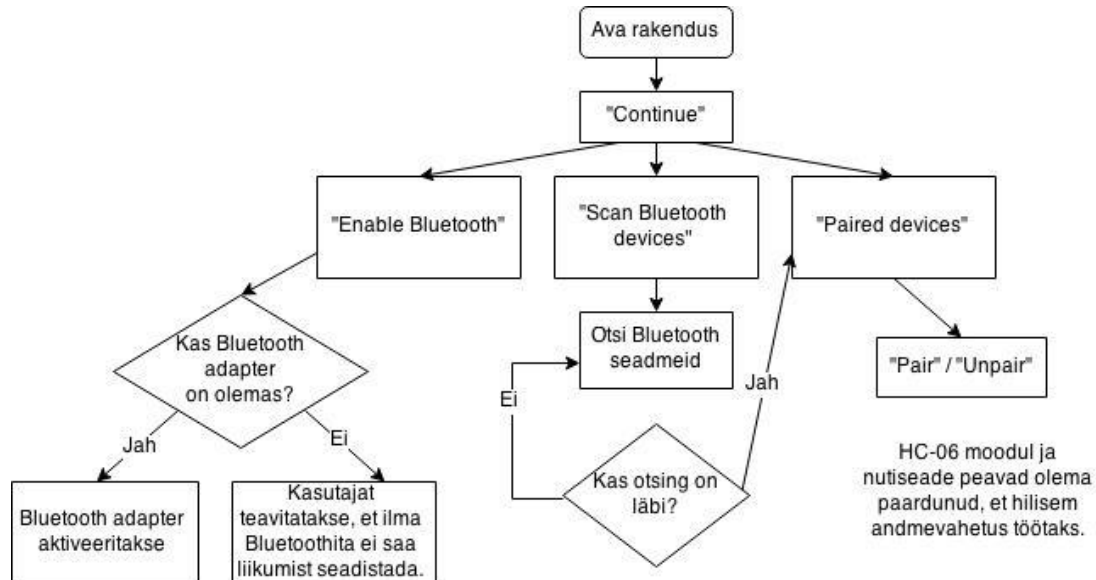
Kogu nutirakendus on programmeeritud Android Studios. See on Androidi rakenduste arendamise ametlik keskkond ning põhineb IntelliJ IDEA platvormil, mis on üks levinumaid Java arendamiseks loodud tööriistu [10]. See võimaldab loodud projektidele lihtsalt lisada uusi klasse, alamprogramme ning kujundusmalle.

Android Studio hõlmab endas suurte võimalustega kasutajaliidese modifitseerimise tööriistu. Kujundusaknas saab lisada kõikvõimalikke kasutajaliidese objekte: nuppe, tekstivälju, liugureid ja palju muud. Kõik loodud elemendid saab Java ning Androidi API abil siduda ning vajalikke ülesandeid täitma panna. Samuti saab neid objekte suurel määral vastavate vaadete XML-failides modifitseerida ja ka väliste pildifailide abil kujundada.

Android Studiosse on sisse ehitatud ka Androidi operatsioonisüsteemil põhinev emulaator rakenduste testimiseks. Selleks, et koodi testida, ei pea tingimata omama reaalselt Androidiga seadet vaid võib kasutada ka emulaatorit, mis võimaldab koodi käivitamisplatvormiks valida paljude erinevate tuntud telefonimudelite vahel. Käesoleva töö käigus kasutati emulaatori võimalusi vähe, sest rakenduse üks tähtsamaid osi oli Bluetooth ühendus ning emuleerida seda ei saa.

8.2. Bluetoothi seadistamine

Programmi Bluetoothi seadistamise vaates saab kasutaja ära seadistada tähtsamad Bluetoothi atribuudid (Joonis 18).

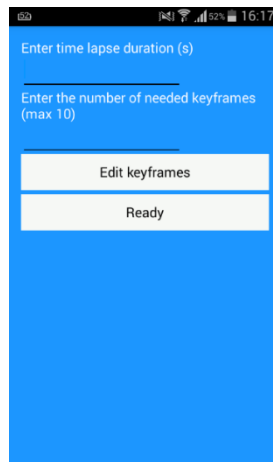


Joonis 18. Androidi rakenduses Bluetoothi põhiliste atribuutide määramise skeem.

Bluetoothi seadistamisprogrammi kirjutamisel on kasutatud koodinäiteid Lorensius Londa blogist [11].

8.3. Andmeplokkide seadistamine

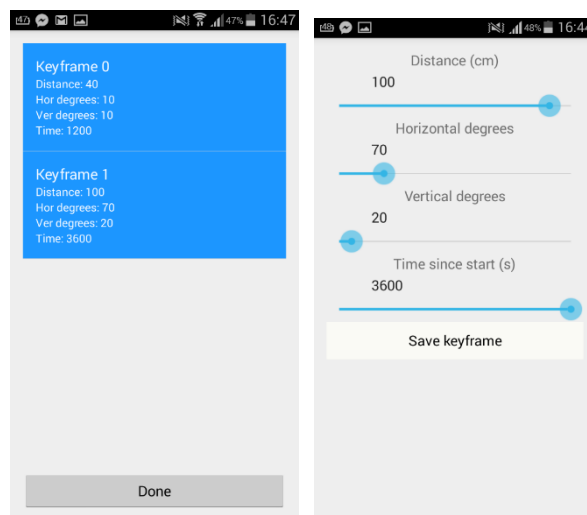
Kogu liikumiseks vajalik teave edastatakse kaamerat liigutavasse seadmesse baidijadana (Joonis 13). Selles olevate andmeplokkide sisu saab kasutaja seadistada. Selle jaoks peab kasutaja eelnevalt määrama, mitu andmepaketti kokku defineeritakse ning kui pikka võtet soovitakse teha (Joonis 19).



Joonis 19. Võtte esialgse info seadistamise vaade

Kasutaja sisestatava info baasil algseadistatakse andmeplokid, mida kasutaja muuta saab. Andmeplokkidest ülevaate saamiseks kasutatakse Androidi SDK *ListView* klassi (Joonis 20).

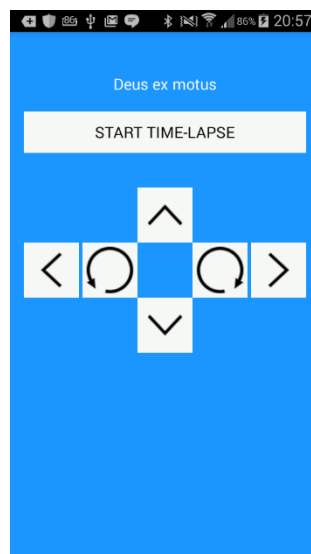
Iga andmeploki kirjelduse peale on võimalik vajutada, et pääseda ligi selle väärtuste seadistamisele. Väärtusi on võimalik muuta liugurite abil, aga kui vajutada näpuga väärtuse peale, saab täpsema väärtuse trükkida ka klaviatuuri abil. (Joonis 20)



Joonis 20. Andmeplokkide ülevaade ja nende muutmine

8.4. Andmete edastamine

Kasutaja käivitab liikumise, kui vajutab rakenduse lõppekraanil nupule „Start time-lapse“. Enne seda on kasutajal nuppude abil võimalik muuta reaajas seadme algpositsiooni (Joonis 21).



Joonis 21. Rakenduse lõppvaade enne liikumise alustamist.

Üle Bluetoothi andmete saatmiseks on vaja luua ühendus HC-06 mooduliga. Seda saab teha Androidi teegi klasside *BluetoothAdapter* ja *BluetoothSocket* erinevate meetodite abil, millest kõige viimane on *connect()*. Bluetoothiga suhtlemiseks on vaja luua uus lõim, millesse saab meetodiga *write(byte[])* andmeid saata.

Käesoleva töö nutiseadme rakendus saadab peamist liikumist puudutavad andmed meetodiga *sendBtData(ArrayList<keyFrame> keyFrameList)* (Joonis 22).

```
private void sendBtData(ArrayList<keyFrame> keyFrameList) {
    for(int i = 0; i<keyFrameList.size(); i++){
        byte[] dataToTransfer = new byte[9];
        //notify board that the i-th keyFrame is coming
        sendBtOrderNr(i);

        dataToTransfer[0] = (byte) 0x6B; // send char 'k'

        dataToTransfer[1] = (byte) ((keyFrameList.get(i).mDistance >> 8) & 0xFF); //higher byte
        dataToTransfer[2] = (byte) (keyFrameList.get(i).mDistance & 0xFF); //lower byte
        dataToTransfer[3] = (byte) ((keyFrameList.get(i).mHorDeg >> 8) & 0xFF); //higher byte
        dataToTransfer[4] = (byte) (keyFrameList.get(i).mHorDeg & 0xFF); //lower byte
        dataToTransfer[5] = (byte) ((keyFrameList.get(i).mVerDeg >> 8) & 0xFF); //higher byte
        dataToTransfer[6] = (byte) (keyFrameList.get(i).mVerDeg & 0xFF); //lower byte
        dataToTransfer[7] = (byte) ((keyFrameList.get(i).mTime >> 8) & 0xFF); //higher byte
        dataToTransfer[8] = (byte) (keyFrameList.get(i).mTime & 0xFF); //lower byte

        try {
            conBT.write(dataToTransfer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Joonis 22. Andmete valmispanek Bluetoothi saatmiseks

9. Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli konstrueerida mobiilne seade, mille abil oleks võimalik teha *time-lapse* videoid, mille kestel kaamera liiguks aeglaselt kolmes teljes. Peale selle oli üheks tähtsamaks nõudmiseks seadme juhtsüsteemi arendamine Android operatsioonisüsteemiga nutiseadmele.

Töö käigus tuli põhjalikult tegeleda konstrueeritava seadme mehaanika, elektroonika ning tarkvaraga. Lisaks pidi arendama Android operatsioonisüsteemile rakenduse, mis suudaks eelpoolmainitud seadmega juhtmevabalt suhelda.

Töö raames valminud mehaanikalahenduste disainimisel peeti silmas, et komponente saaks osta, tellida, lõigata või 3D-printeriga toota. Samuti pandi rõhku sellele, et mehaanikakomponentide kulumisel oleks neid võimalik lihtsalt vahetada. Lisaks eelnevale prooviti kogu mehaanika disainimisel hoida esteetilist joont.

Elektroonikalahenduse väljatöötamisel pöörati rõhku selle töökindlusele. Tähtis oli, et kõikide voolujuhtide ristlõike pindala oleks vastavuses sellest läbi liikuva vooluga. Nutiseadmega suhtlemise tarbeks kasutati välist Bluetooth moodulit.

Mikroprotsessorile laetud koodi testiti selle loomisel põhjalikult, et vältida seadme ootamatut käitumist, mis võiks sellel paiknevat kaamerat kahjustada. Kirjutatud juhtseadme programm juhib kolme mootori liikumist ning ka nende liikumise lõpetamist Bluetoothi kaudu vastuvõetud andmete alusel.

Androidil põhineva nutiseadme rakenduse arendamise käigus pöörati tähelepanu sellele, et see oleks kasutajasõbralik. Nuppude arv hoiti minimaalsena ning rõhutati erinevate vaadete loogilisele järgnevusele. Lisaks oli tähtis, et kasutaja saaks välise Bluetooth mooduliga rakendusesiseselt ühenduse luua. Samuti loodi kasutajale võimalus enne kaamera liikuma panemist selle algset asukohta reaajas mugavalt nutiseadme ekraanilt muuta.

Töö alguses seatud eesmärk täideti ning valmistati kõikidele algsetele nõuetele vastav seade, mida on kasutatud *time-lapse* videote filmimisel. Seadme omahind tuli ligikaudu neli korda väiksem praegu saadaval olevate variantide turuhinnast.

10. Summary

SMARTPHONE-CONTROLLED MOTION CONTROL DEVICE FOR MAKING TIME-LAPSE VIDEOS

The purpose of this project was to design and construct a mobile motion control device that would make time-lapse videos more dynamic. The device had to be capable of moving in three axes all at once. Moreover, the device was designed to be controlled by a smartphone running on Android operating system.

During this project tasks regarding mechanical design, electronics design and software development were dealt with.

All mechanical solutions were designed to be replaceable but reliable. Many components for this project had to be 3D-printed so they were designed keeping printability in mind. Also, the overall appearance of the finished product was kept as aesthetic as possible.

The design of electrical solutions placed strong emphasis on reliability. All the traces on the printed circuit board were designed to produce as little heat and noise as possible. For communication with a smartphone an external Bluetooth module was used.

The program that controlled the motion control device was tested thoroughly to prevent possible damages to the connected camera and the device itself. All of the motion of the device is directed by the received data from the Bluetooth module.

The Android application was designed to be user-friendly. User can perform all the needed tasks to initialize and start a controlled motion within the application.

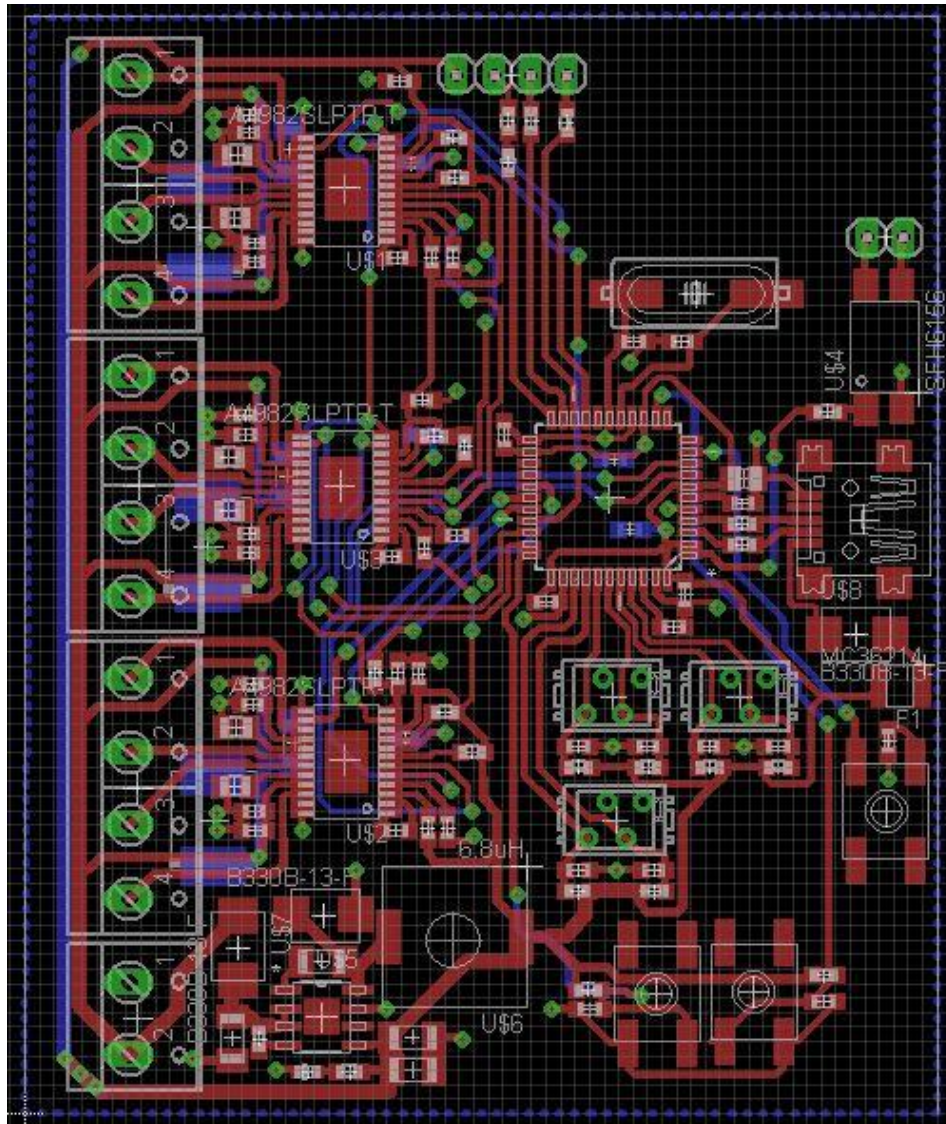
All of the goals set at the beginning of this project were accomplished and the device has already been used to film motion controlled time-lapse sequences. The final net cost of the device turned out to be about a quarter of the price of similar products currently on the market.

11. Viidatud allikad

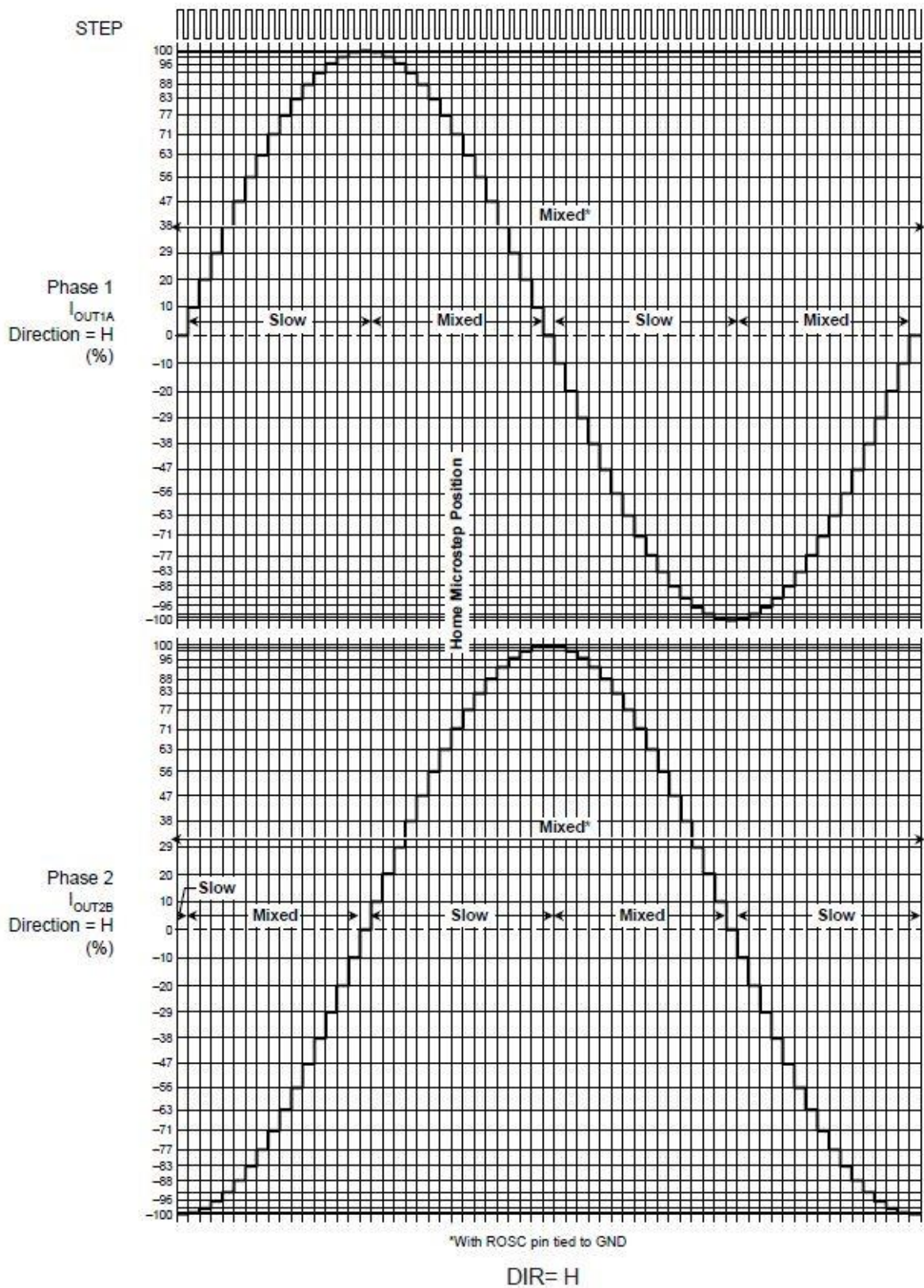
1. Canoni peegelkaameratele mõeldud tasuta lisatarkvara kodulehekülg, vaadatud 19.05.2015.
<http://www.magiclantern.fm/>
2. IT ja sidetehnika seletav sõnaraamat, vaadatud 19.05.15.
<http://www.vallaste.ee/>
3. Elektroonikahuvilistele pühendatud veebileht SparkFun, vaadatud 19.05.15.
<https://learn.sparkfun.com/tutorials/logic-levels>.
4. Dynamic Perception veebipood, vaadatud 19.05.15.
<http://www.dynamicperception.com/category-s/1823.htm?searching=Y&sort=2&cat=1823&show=30&page=1>.
5. Emotimo ettevõtte kohta korduma kippuvad küsimused, vaadatud 19.05.15.
<http://emotimo.com/emotimo-faqs/>.
6. Genie tooteleht, vaadatud 19.05.15. <https://syrp.co.nz/products/genie>.
7. Analog Devices ADP2302ARDZ-3.3 andmeleht, vaadatud 19.05.15.
<http://www.farnell.com/datasheets/1624274.pdf>.
8. A. Microsystems, Allegro A4982 samm-mootori kontrolleri andmeleht (2011)
9. Graafik erinevate akutüüpide energiatihedustest, vaadatud 19.05.15.
http://www.mpoweruk.com/images/energy_density.gif.
10. Android Studio arenduskeskkonna ülevaade, vaadatud 19.05.15.
<http://developer.android.com/tools/studio/index.html>.
11. Androidi rakenduste arendaja Lorensius Londa blogipostitus Bluetoothi arenduse teemadel, vaadatud 19.05.15. <http://www.londatiga.net/it/programming/android/how-to-programmatically-scan-or-discover-android-bluetooth-device/>

Lisad

Lisa 1 – seadme trükkplaadi trasseering



Lisa 2 – 1/16 sammurežiimi erinevate voolunivoode võrdlus ühe võlli täispöörde jooksul. [9]



Lisa 3 – CD

CD sisu:

- Androidi programmi kogu lähtekood koos projektifailidega
- Protsessorile laetud programmi lähtekood koos projektifailidega
- Trükkplaadi disaini skeem, trasseering ja projektifailid
- Bakalaureusetöö pdf-versioon

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Sten-Oliver Salumaa,

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

Tartus, **22.05.2015**