

Sebawe

Kaksipyöräinen henkilökuljetin

Lauri Läärä

Olli-Pekka Karjalainen

Opinnäytetyö
Toukokuu 2014
Kone- ja tuotantotekniikka
Kone- ja laiteautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Kone- ja tuotantotekniikka
Kone- ja laiteautomaatio

Lauri Läärä & Olli-Pekka Karjalainen:
Sebawe
Kaksipyöräinen henkilökuljetin

Opinnäytetyö 70 sivua, joista liitteitä 11 sivua
Toukokuu 2014

Sebawe eli self-balancing two wheeler on kaksipyöräinen yhden ihmisen henkilökuljetin. Tämä on projektin kirjallinen osuus työstä, jonka tavoitteena oli tutkia käytännön kautta laitteen suunnittelu, rakennus ja toiminta. Käytännön osuus toteutettiin kahden opiskelijan yhteistyöprojektina, koska projekti oli erittäin laaja. Tavoitteena oli rakentaa kaupallista Segwayta vastaava laite, mutta halvemmalla. Pienempinä kokonaisuuksina tavoitteet olivat mekaniikan, elektroniikan ja ohjelmiston saattaminen toimivaksi kokonaisuudeksi.

Laite myös rakennettiin täysin kahden oppilaan voimin. Laitteen rakenteessa käytettiin hyvin vähän valmiita komponentteja. Suurin osa niin mekaanisista kuin sähköisistäkin osista suunniteltiin, valmistettiin ja ohjelmoitiin itse. Yllättäen projektin suurimmat ongelmat liittyivät juuri niihin osiin, joita ei valmistettu itse. Osa komponenteista suunniteltiin itse ja tilattiin alihankintana ajan säästämiseksi.

Laitteesta tuli hyvä kokonaisuus, joka onnistui suunnitellulla tavalla. Suuret ongelmat saatiin ratkaistua, koska suunnitelmat olivat toteutettu modulaarisella tavalla ja mahdolliset ongelmat oli otettu huomioon jo ennen niiden havaitsemista. Projektista opittiin paljon ja erittäin laajalla skaalalla asioita. Parempaan lopputulokseen olisi päästy, jos komponenttien toimituksissa ei olisi ollut ongelmia. Alkuperäiset tavoitteet Segwaytä vastaavasta ja halvemmasta laitteesta kuitenkin saavutettiin ja projektia voidaan pitää onnistuneena.

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in Mechanical and Production Engineering
Machine Automation

Lauri Läärä & Olli-Pekka Karjalainen:
Sebawe
Two Wheeled Person Transporter

Bachelor's thesis 70 pages, appendices 11 pages
May 2014

Sebawe is an abbreviation for “self-balancing two-wheeler”, which is a one-person transporter. Our goal was to research, design and build a device of this type. This is the written report of our final year thesis project. Because of the size of this project and the great amount of work involved, the project was executed by two students. Our main objective was to build a device similar to the commercially-produced Segway, but cheaper. Smaller sets of objectives were to make the mechanical build, electronics and software work as a single functional unit.

The device was built entirely by two students, the writers of this report. Only a few ready-made components were used. Most of the mechanical and electronic parts were designed, assembled and programmed by ourselves. To our surprise, the biggest problems in the project were caused by those parts which we did not make. To save time, a few components were designed by us but purchased from subcontractors.

The device became a successful unified whole which worked as planned. Major problems were solved, since the plans were implemented in a modular way and problematic issues were noticed and resolved at the planning stage. Through this project, we learned a lot about a wide range of issues. A better outcome would have been achieved if the supply of components had not been so problematic. The original objective of making a cheaper version of the Segway was accomplished and the project was successful.

SISÄLLYS

1	JOHDANTO	7
2	SEBAWE	8
2.1	Laitteen yleinen käyttö	8
2.2	Kaupalliset versiot.....	10
2.3	Tavoitteet ja turvallisuus	11
3	MEKANIikka.....	13
3.1	Runko	13
3.2	Akselit ja laakerit	14
3.3	Moottorien kiinnitys.....	15
3.4	Rungon ja akselin vaatimukset	16
3.5	Kotelointi.....	17
4	OHJAUSKAHVA	18
4.1	Kahvan 3D-rakenne ja vaatimukset	18
4.2	Ohjaimet	19
4.3	3D-Tulostus	20
5	ELEKTRONIIKKA.....	21
5.1	Yleinen rakenne	21
5.2	Sähköinen mitoitus.....	23
5.3	Akusto ja laturi.....	25
5.4	Anturit	27
5.4.1	ADXL345-Kiihtyvyyssanturi	29
5.4.2	L3G4200D-gyro	31
5.4.3	TMP36GZ -lämpötila-anturit	32
5.5	Moottorinohjaimet	34
5.6	Langaton data.....	37
5.7	Langaton näyttö	39
6	OHJELMA	41
6.1	I2C-väylä	41
6.2	RS232-Väylä.....	41
6.3	Arduino 0.....	42
6.4	Arduino 1	46
6.5	Arduino 2.....	48
6.6	Arduino 3.....	49
6.7	Datan monitorointiohjelma (PC).....	52
7	LOPPUTULOS	56
	LÄHTEET.....	59

LIITTEET.....	60
Liite 1. Akuston liitin sähkökuva.....	60
Liite 2. Moottorinohjain sähkökuva.....	61
Liite 3. Arduino 0 sähkökuva.....	62
Liite 4. Arduino 1 sähkökuva.....	63
Liite 5. Arduino 2 sähkökuva.....	64
Liite 6. Arduino 3 sähkökuva.....	65
Liite 7. Akselin CAD-kuva.....	66
Liite 8. Laipan CAD-kuva.....	67
Liite 9. Kahvan tulostusdokumentti.....	68
Liite 10. Kannen tulostusdokumentti.....	69
Liite 11. Kannakkeiden tulostusdokumentti.....	70

LYHENTEET JA TERMIT

ADC	Analog to Digital Converter, analogi-digitaali-muunnin
AI	Analog Input, analooginen sisäänmeno (ADC).
AO	Analog Output, analooginen lähtö (PWM).
Asynkroninen	Sarjaliikennemuoto, jossa on yksi signaali kumpaankin suuntaan.
Baud	Baudinopeus, elektronisen signaalin muutosnopeus per. sekunti. Tässä työssä sama asia kuin bittinopeus.
DI	Digital Input, digitaalinen sisäänmeno.
DO	Digital Output, digitaalinenlähtö.
EMI	Electro Magnetic Interference, sähkömagneettinenhäiriö
Hila	Mosfetin hila (eng. Gate) pinni, ohjauspinni.
IMU	InertialMeasurementUnit, yhdistelmä liikeantureita.
Lähde	Mosfetin lähde (eng. Source) pinni, ns.sisäänmeno.
PWM	Pulse Width Modulation, pulssinleveysmodulaatio.
Sebawe	Self-Balancing Two Wheeler, laitteen nimi.
STL	Stereo lithography, 3D-tulostimen vaatima tiedostomuoto
Synkroninen	Sarjaliikennemuoto, joka toimii vain toiseen suuntaan yhdellä signaalilla.

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli tutkia kaksipyöräisen henkilökuljettimen rakentamiseen vaadittavan tekniikan toimintaa. Maailmalla vastaavia kaupallisia laitteita on jo yleisessä käytössä. Tämä työ kartoittaa laitteeseen vaadittavan tekniikan ja esittelee yhden ratkaisumallin kaikkiin projektissa ilmenneisiin ongelmiin. Työ on tuotoksellinen työ, jonka aikana rakennettu laite jäi Tampereen Ammattikorkeakoulun käyttöön. Laitteen rakennus ja suunnittelu suoritettiin kahden opiskelijan yhteisenä projektina. Projekti aloitettiin suunnittelusta, josta edettiin sähköjen rakentamiseen ja sitten mekaniikan rakentamiseen. Viimeisenä vaiheena oli mikroprosessorien ohjelmointi ja projektin viimeistely.

Suunnittelussa pyrittiin yksinkertaisuuteen ja nopeaan kokoonpantavuuteen. Suunnitelmat mekaanisesta ja sähköisestä rakenteesta saatiin nopeasti kasaan, mutta mekaanisten osien tilauksen myöhästyminen oli ensimmäinen projektin ongelmista. Sähköiset komponentit saatiin ajallaan ja projektin kasaaminen aloitettiin lopulta sähköjen puolelta. Kun mekaniikan kokoonpanoon päästiin lopulta siirtymään oli sähkötekniinen puoli jo lähes valmis. Kaikeksi onneksi mekaanisetkin suunnitelmat oli tehty hyvin ja kokoonpano oli nopea. Tilausten myöhästymisistä johtuen projektin loppupuolella oli hyvin kiire rakentamisen suhteen. Kiirettä ei auttanut se, että erä viallisia transistoreita vei useita päiviä hukkaan. Kun ohjelmointiin päästiin, niin projekti oli rakentajista riippumattomista syistä noin kuukauden myöhässä. Ajanpuutteen vuoksi monista alkuperäisistä ideoista luovuttiin ja projektin laajuutta päätettiin karsia.

Monista ongelmista huolimatta projekti saatiin päätökseen ja tämä kirjallinen opinnäytetyö kertoo tarkemmin kaikista vaiheista. Projektin valintaan oltiin tyytyväisiä, koska sen kautta päästiin kerryttämään paljon uutta osaamista varsinkin elektroniikan ja 3D-tulostuksen puolelta. Projekti oli erittäin opettavainen ja toimi myös erittäin hyvänä harjoituksena laitteen suunnittelusta ja proto-rakentamisesta. Jos laite myöhemmin tarvitsee huoltotoimenpiteitä suosittelomme kaikkia, jotka laitetta aikovat huoltaa lukemaan tämän opinnäytetyön huolellisesti läpi.

2 SEBAWE

2.1 Laitteen yleinen käyttö

Sebawe eli Self balancing two wheeler on kahdella pyörällä kulkeva yhden ihmisen kulkuväline. Se pysyy mikroprosessorin avulla pystyssä ja käyttäjä pystyy ohjaamaan sitä omalla kehollaan. Eteen- ja taaksepäin kulkemista kontrolloidaan kallistamalla omaa kehoa eteen tai taakse, kääntymistä ohjataan ohjauskahvassa olevalla sauvaohjaimella. Laite huolehtii pystyssä pysymisestä, kunhan käyttäjän liikkeet ovat riittävän rauhallisia, eikä käyttäjä yritä itse tasapainoilla paljoa. Tämä johtaa muuten helposti liikkeiden kertautumiseen, kun käyttäjä ja laite reagoivat molemmat kallistuksiin. Paras tilanne on siis se, että käyttäjä seisoo mahdollisimman lähellä 90 asteen kulmaa sebawen seisontatasoon nähden. Tällöin toiminta on optimaalista.

Laitteen käyttö aloitetaan kääntämällä "ARMED / OFF" kytkin asentoon "OFF". Myös päävirtakytkimen täytyy olla asennossa "OFF". Tämän jälkeen laitteen pohjassa oleva akuston liitin voidaan kytkeä. Liittimen kytkemisen jälkeen päävirtakytkin voidaan kääntää "ON" asentoon, jolloin PITÄÄ kuulua käynnistysääni. Jos käynnistysääntä ei kuulu, niin käyttöä ei saa jatkaa. Käynnistysäänen puuttuminen on seurausta jostain häiriöstä, jolloin laitteen turvallinen käyttö ei välttämättä ole mahdollista. Käynnistysäänen kuulumisen jälkeen voidaan kääntää "ARMED / OFF" kytkin asentoon "ARMED". Tällöin moottorinohjaimet käynnistyvät. Laite on nyt valmis ajettavaksi. Ajaminen aloitetaan nostamalla laite suoraan (vaakatasoon) ja painamalla ohjauskahvan keskellä olevaa vihreää nappia. Laite päästää käynnistysäänen ja aloittaa aktiivisen tasapainottamisen. Tämän jälkeen laitteen päälle voi varovasti astua. Laitteen päällä pitää pyrkiä seisomaan mahdollisimman keskellä seisontatasoa. Laitteen ollessa ajossa ohjauskahvan kolme merkkivaloa ilmaisevat akun varaustasoa. Kun varaustaso on laskenut punaiselle laite alkaa pitää merkkiääntä merkiksi erittäin alhaisesta akkujen varaustasosta. Ajaminen EI OLE TURVALLISTA enää kun merkkiääni kuuluu. Ääni varoittaa siitä, että jännite on vaarallisen alhainen, jolloin tasapainotus ei enää välttämättä toimi. Ajaminen tulee tällöin lopettaa ja akut kytkeä lataukseen.

Jos ajamisesta pidetään lyhyt tauko, niin laitteen paikallaan pysyminen voidaan varmistaa kääntämällä kytkintä "ARMED / OFF" asentoon "OFF". Tällöin moottorinohjaimet sammuvat, mutta muuten laite jää päälle. Kun ajaminen lopetetaan kokonaan on virrat syytä katkaista pääkytkimestä, tällöin akusto ei tyhjene. Kun pääkytkin on asennossa "OFF" voidaan laite kallistaa varovasti etupuskurilleen ja irrottaa akuston liitin. Laturi voidaan näin kiinnittää akuston liitimeen ja valita laturista auton kuva (12V), jonka jälkeen laturi hoitaa automaattisesti latauksen (6-10h). Laturiin syttyy merkkivalo, kun akut ovat latautuneet täyteen. Kuvassa yksi laite on valmiina ajoon.



KUVA 1. Sebawe valmiina ajoon.

Muuta laitteen käytössä huomioitavaa on se, että se ei ole kaupallinen tuote ja sen käytössä on aina hyvä noudattaa suurta varovaisuutta. Hyvin epätasaiset tai vetiset ajoalustat eivät ole suositeltavia. Laitetta ei ole eristetty roiskevettä vastaan. Laitteen kantaminen tulee aina suorittaa pohjasta, eikä missään nimessä ohjauskahvasta. Kaikki mekaaniset iskut ja pudotukset on syytä pitää hyvin pieninä.

2.2 Kaupalliset versiot

Tunnetuin kaupallinen kaksipyöräinen yhden ihmisen kulkuväline on Pohjois-Amerikkalainen Segway. Sen rakenne on päällisin puolin hyvin samankaltainen kuin sebawe:ssa, mutta tekniikka eroaa varmasti paljonkin. Tekniikasta ei tietysti ole täysin tarkkaa tietoa, koska laite on kaupallinen tuote ja tarkka tekniikka on valmistajan salaisuus. Se tiedetään kuitenkin, että Segway rakentuu litiumioni-akuston ja harjattomien moottoreiden ympärille. Segwayn huippunopeus on n. 20 km/h. Tasaipainoituksesta kerrotaan, että Segway käyttää gyroja ja nestemäisiä kallistusantureita. Sebawessa käytämme gyroja ja kiihtyvyyssantureita.

Myös muutamia muita kaupallisia tuotteita on, mutta hyvin vähän ja lähinnä marginaalisia. Aika ei kenties ole vielä valmis tämän kaltaisille tuotteille, jotta ne voisivat yleistyä enemmän. Tämä on varmasti myös syy siihen, miksi markkinoille ei ole suurta tunkua. Tästä myös johtui koko idea lähteä tutkimaan kyseisen kaltaisia laitteita, koska niitä ei selvästikään ole vielä kehitetty loppuun asti. Suomessa ei saa käyttää Segway:tä tai mitään muutakaan vastaavaa laitetta tieliikenteessä. Lainsäädäntö ei tosin ole aivan yksiselitteinen laitteiden osalta. Laki vaatii moottoriajoneuvojen kuten Segwayn rekisteröimistä, jotta sillä saisi liikkua tieliikennelain määrittelemillä alueilla. Segwaytä ei kuitenkaan voi rekisteröidä nykyisen lainsäädännön puitteissa. Harmaan alueen lakiin aiheuttaa kuitenkin se, että yksikään virallinen taho ei ole vielä ottanut kantaa koko asiaan kumpaankaan suuntaan. Epävirallisesti on myös tiedossa, että Segwayta on käytössä useampia kappaleita Suomessa, joihin viranomaiset eivät ole puuttuneet millään tavalla. Useat muut Euroopan maat ovat nykyisin hyväksyneet Segwayn käytön kirjoittamalla lakeihin sopivat pykälät niitä varten. Kävimme katsomassa Segway vuokrauspalvelua Tallinnassa, sieltä löytyy muutamia liikkeitä. Kuvassa kaksi on yksi näistä vuokrapaikoista.



KUVA 2. Segway vuokrauspalvelu Tallinassa.

2.3 Tavoitteet ja turvallisuus

Tavoitteet voisi jakaa tarkemmin kolmeen eri osa-alueeseen. Ensimmäisenä oli tutkia kahdella pyörällä tasapainottelevan laitteen tekniikkaa ja sitä kautta miettiä, että miten sellaisen saisi kuljettamaan ihmistä. Toisena oli suunnitella ja valmistaa toimiva laite tehtyjen tutkimusten ja olettamusten kautta. Kolmas ja tärkein tavoite oli laitteen suunnittelun kautta tehdä siitä turvallinen ja helppokäyttöinen.

Ajatusprosessi lähti moottoreista ja akuista, sähköturvallisuus ohjasi valintoja käytännössä kokonaan. Suunnitelmamme perustuivat suojajännitteen antaman rajan sisään. Tämä poissulki käytännössä isot servomoottorit. Toinen erittäin tärkeä tekijä oli akusto ja sen turvallisuus, joka poissulki listalta isot harjattomat moottorit. Laite jää koulun käyttöön ja emme voi vaikuttaa ihmisten käyttövirheisiin, mutta voimme ennakoida suunnittelussa mahdollisten vahinkojen laajuuden. Esimerkiksi harjattomat moottorit ja litiumpolymeeri- tai litiumioniakut olisivat olleet erinomainen valinta, mutta hyvissä puolissa piilee juurikin niiden vaarallisuus. Purkuvirrat näissä akuissa

ovat suuria. Li-Po:t pystyvät antamaan täyden purkuvirran heti ja ylläpitämään sitä lähes loppuun asti. Mutta jos laitteeseen tulisi oikosulku, seuraukset voisivat olla katastrofaaliset. Se ei pelkästään sulattaisi johtimia ja aiheuttaisi palamisvaaran, vaan potentiaalinen räjähtämisvaarakin on olemassa. Niitä ei myöskään saa ikinä ajaa loppuun asti ja lataus on myös paljon tarkempaa, kuin esimerkiksi lyijyakuissa. On hyvä myös mainita, että jos Li-Po-akku kokee mekaanisen vaurion, joka vahingoittaa sen sisäistä rakennetta aiheuttaen oikosulun on seurauksena herkästi räjähdysmäinen tulipalo. Tämä johtuu litiumin alhaisesta itsesyttymislämpötilasta (179°C). Oikosulku voi nostaa johtimien lämpötilan nopeasti useisiin satoihin asteisiin. Näiden rajojen sisällä valitsimme turvalliset lyijyhyytelöakut ja Boshin planeettavaihteelliset tasavirtamoottorit. Yhden moottorin vaihteiston jälkeinen maksimi vääntömomentti on 80Nm ja maksimi kierrosnopeus 400 kierrosta minuutissa.

Laitteen käytön kannalta tärkein turvallisuustekijä oli suunnitella toimiva kuolleen miehen kytkin. Juotimme kahden neodyymimagneetin väliin johdon ja kun magneetit ovat paikallaan seisonatasoon kiinnitettyjen kontaktien päällä, mosfet:ien ohjauksen virtapiiri on aktiivinen. Jos toinen tai molemmat magneetit irtoavat, virtapiiri katkeaa. On tärkeää huomata, että kuolleen miehen kytkin vaikuttaa vain ja ainoastaan ohjaussignaalin kulkuun. Laite on tästä huolimatta jännitteellinen ja toimintavalmis. Jos käyttäjä on jostain syystä irrotanut kuolleen miehen kytkimen, on syytä kääntää laitteen takana oleva ARMED-kytkin OFF-asentoon, ennen kuin asettaa kuvassa kolme esitetyn kytkimen takaisin paikoilleen. Vaihtoehtoisesti laitteen voi myös deaktivoida ohjaukskahvan painonapista, ennen kuin liittää magneetit takaisin kontaktipinnoille.



KUVA 3. Kuolleen miehen kytkin.

3 MEKANIikka

3.1 Runko

Laitteen runko tehtiin kokonaan alumiiniprofiilista, sen modulaarisuuden ja helpon käsiteltävyyden takia. Profiilin kooksi valittiin 30x30mm ja se tilattiin valmiiksi mittoihin sahattuna. Kuvassa 4 olevat runkopalat kiinnitettiin toisiinsa kuvassa 5 esitetyillä kulmapaloilla, eikä yhtään hitsisaumaa tarvinnut kasausvaiheessa tehdä. Ohjauskahva tehtiin ohuemmasta materiaalista, jotta siitä saatiin kevyemmän ja siromman näköinen. Materiaaliksi valittiin halkaisijaltaan 19mm teräsputki.

Alumiiniprofiili ja kiinniketarvikkeet tilattiin EasyConveyor Oy:ltä ja teräsputki haettiin koulun metallipajalta. Profiilin toimituksessa oli merkittäviä ongelmia, tilaus lähetettiin Easy Conveyor:lle Joulukuussa 2013 ja jotkut tilatuista osista puuttuivat vielä projektin päätöspäivänäkin. Ensimmäinen lähetys vastaanotettiin maaliskuun puolivälissä, joka osoittautui vääräksi. Toinen tilaus saatiin huhtikuun alussa, joka osoittautui puutteelliseksi. Tästä huolimatta rungon kasaus aloitettiin viimeisimmän tilauksen saavuttua, vaikka alkuperäistä suunnitelmaa jouduttiin muuttamaan.



KUVA 4. Runkopalkit



KUVA 5. Kulmaliitos

3.2 Akselit ja laakerit

Akselit suunniteltiin itse ja tilattiin ammattikoulu Iisakistamittoihin koneistettuina. Laakerit katsoimme Vuorenmaa Oy:n sivuilta, FK Bearinggroupin laakeriluettelosta. Tarvitsimme valmiin laakeriyksikön, joka oli helppo kiinnittää alumiiniprofiiliin pulttiliitoksella, ilman koneistusta tai hitsausta. Löysimme listalta yhden laakeriyksikön joka sopi tarkoitukseen. Tästä mallista tilasimme isoimman, koska fyysisellä koolla ei ollut merkitystä, sisätilaa rungossa oli reilusti. Koneistettu akseli ja laakeriyksiköt ovat esitetty kuvassa kuusi. Laakeriksi siis valikoitui Nachin UP-006 laakeriyksikkö, sisähalkaisijaltaan 30mm. Valmistajan mukaan yhden laakeriyksikön dynaaminen kantavuusluku C on 13200 kN. Staattinen kantavuusluku C_0 on 8300 kN. Laakerin kiinnitys akseliin tehtiin laakeriyksikön mukana tulleella epäkeskokiristimellä.

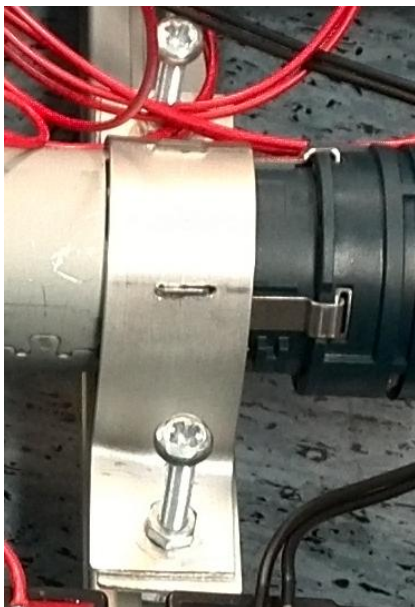


KUVA 6. Laakerit ja koneistettu akseli

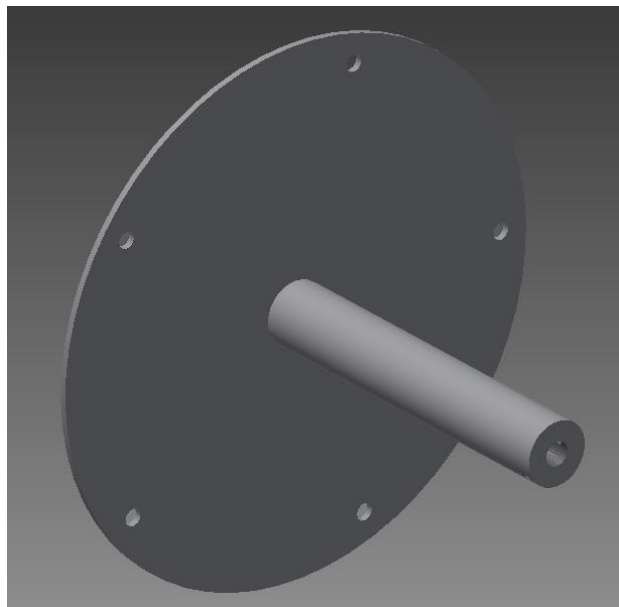
3.3 Moottorien kiinnitys

Moottorit kiinnitettiin runkoon alumiinisilla pannoilla, jotka muokattiin 30x2mm alumiinilistasta moottorin omiin olakkeisiin sopiviksi. Pannat taas kiinnitettiin alumiinirunkoon pulttiliitoksin. Kuvassa seitsemän on kuvattuna moottorien kiinnike. Moottorien päässä on ½ tuuman UNF ulkokierre, joten akselin päähän koneistettiin vastaava sisäkierre. Samaan päähän akselia on porattu ja kierteytetty M6 läpireikä, minkä avulla saa puristusliitoksen akselin molemmin puolin kahdella lukitusruuvilla. Moottorin akseliin ajettiin kiviterällä tasopinnat lukituspultteja varten ja lopuksi kierre sinetöitiin lukitteella.

Voimansiirto akselistä pyörille tapahtuu laserleikattujen laippojen avulla, jotka on kiinnitetty alumiinisen pyörän vanteeseen viidellä pulttiliitoksella. Materiaalivahvuutta kiinnityslaipalla on 3mm ja se on tehty kylmävalssatusta teräslevystä. Sen keskellä on 20mm kohdistusreikä ja koneistetusta akselistä löytyy samankokoinen olake, jotta se kohdistuisi suoraan keskelle. Akseli silloitettiin ensin toiselta puolelta vetelyn estämiseksi ja sitten hitsattiin kiinni a-mitaltaan noin 10mm pienahitsillä. Suunnittelimme kuvassa kahdeksan esitetyt laipat AutodeskInventorilla ja niiden laserleikkauksen toteutti HT-koneistus.



KUVA 7. Moottorin kiinnike



KUVA 8. Akseli ja laippa

3.4 Rungon ja akselin vaatimukset

Vaatimukset rungolle olivat helppo koottavuus, hyvä kestävyys ja modulaarisuus. Modulaarisuudella tässä tarkoitetaan alumiiniprofiilin monia eri käyttötapoja. Kun koko runko on kasattu siitä, liitokset ovat helppoja, nopeita ja suunnitelmia voi muuttaa tarvittaessa. Esimerkiksi vielä suunnitteluvaiheessa laitteen koteloinnista ei oltu päätetty, mutta tiedettiin ettei se ole este, koska runkoon oli helppo kiinnittää mitä tahansa materiaalia pulttiliitoksin. Tämä kyseinen profiili on myös paljon käytetty monissa automaatiosovellutuksissa. Profiilia varten on räätälöity erilaisia kiinnikkeitä, listoja, renkaita, laakereita, ja vaikka mitä.

Mekaanisesti rungon, akselien ja laakeroinnin tulisi kestää ihmisen paino, laitteen paino, sekä mahdollinen pieni tiputus esimerkiksi kynnyksen yli. Mitoittavan voiman arvioitiin olevan noin 50 kg laite ja maksimissaan 120 kg ihminen joka 100mm pudotuksella aiheuttaa 1633N suuruisen voiman.

Useimmissa tapauksissa akselin voi mitoittaa väsyttävän taivutusmomentin mukaan, ja vääntöä tarkastella yksinkertaisesti akselin suuruusluokan kautta. Meidän tapauksessa mitoitus tapahtui laakereiden valinnalla. Ensin etsittiin laakeri, joka soveltui tarkoitukseen parhaiten. Sen sisähalkaisija sattui olemaan 30mm, joten tämän jälkeen akselin halkaisija oli selvä. Tiedämme, että akselit ja laakerit ovat mitoitettu aika rajusti yli, mutta siitä ei ole haittaa. Painoa tuli ehkä muutama kilo lisää, mutta paksusta voimansiirtoakselista on meille suurta etua. Se antaa tukevan kiinnityksen moottorin omaan akseliin ja näin pystymme tekemään kahden akselin liitoksen pelkällä pulttiliitoksella ja lukitteella.

3.5 Kotelointi

Laitteen runko koteloitiin läpinäkyväksi, jotta ihmisten olisi helpompi tutkia mitä se pitää sisällään. Läpinäkyvänä sitä on myös hyvä käyttää oppimismateriaalina muille ja onhan sen sisällä oleva tekniikka siististi toteutettu ja miellyttävää katsella. Materiaaliksi valikoitui 3mm polykarbonaatti, eli ns. iskunkestävä muovi. Kyseisen materiaalin hyviin puoliin kuuluu hyvä iskunkestävyys ja helppo työstettävyys esimerkiksi lastuamalla.

Polykarbonaatti kestää kuitenkin erittäin huonosti liuotainaineita, ne voivat pinnan lisäksi pilata myös materiaalin iskunkestävyyden. Sen ominaisuudet lämpökäsittelyssä on myös heikot. Polykarbonaattia voi taittaa sen rakennetta murtamatta, mutta tämä ominaisuus katoaa pakkasella. Siksi onkin erittäin tärkeää, ettei laitteen koteloon kohdisteta mitään voimia, jos sitä on kuljetettu tai säilytetty kylmässä. Materiaali haettiin Etralta mittoihin sahattuna ja muokattiin sopivaksi porakoneen ja dremelin avulla. Kuvassa yhdeksän on nähtävillä laitteen vasen kylki sekä jyrsin, jolla muoviin työstettiin tarvittavat reiät.



KUVA 9. Koteloinnin työstövaihe

4 OHJAUSKAHVA

4.1 Kahvan 3D-rakenne ja vaatimukset

Ohjauskahvan muotoilua ja materiaalia mietittiin pitkään. Päädyttiin kuitenkin käyttämään hyödyksi koulun tarjoamaa 3D-tulostinta. Ensin hahmoteltiin paperille muutama luonnos mahdollisesta kahvasta ja siihen halutuista ominaisuuksista, sitten piirrettiin sama AutodeskInventorilla ja vietiin 3D-printterille työstettäväksi.

Kahva koostuu kahdesta osasta, rungosta ja kannesta. Runkoon kohdistuu mekaaniset rasitukset sekä pulttiliitokset ja sen sisälle on jätetty tilat komponenteille ja niiden johtimille. Kuvassa kymmenen esitetystä kannesta on paikka akun tilaa indikoiville ledeille, laitteen aktivointinapille ja tarkasti mitoitettu paikka sauvaohjaimelle. Kannen tarkalla mitoituksella saatiin mekaaninen rasitus siirrettyä sauvaohjaimen heikolta piirilevytä kannen runkoon. Pyöreän ohjaustangon kiinnittämiseksi runkoon tarvittiin myös asianmukaiset kiinnikkeet. Kuvassa 11 on kannakkeet kiinnitettynä ohjaustankoon.



KUVA 10. Juuri tulostettu kahva ja kansi



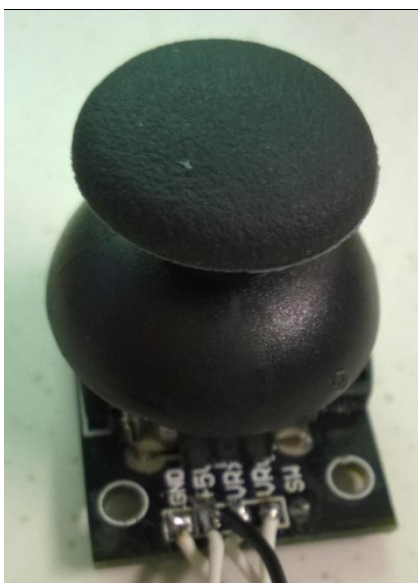
KUVA 11. Kiinnike kannakkeet

4.2 Ohjaimet

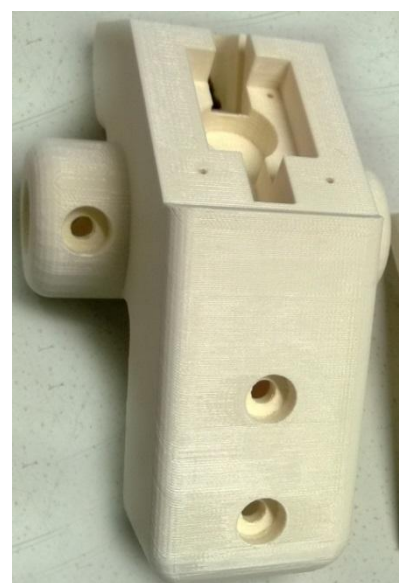
Ohjaimia kuvassa 12 esitettyyn kahvaan tuli kaksi, sauvaohjain kääntymistä varten ja jousikuormitteinen painonappi aktivointia sekä deaktivointia varten. Kuvassa 13 esitetty sauvaohjain on käytännössä sama, mitä on käytetty vanhemmissa Playstation pelikonsoleiden ohjaimissa. Se koostuu kahdesta potentiometrasta, suunnat x ja y. Potentiometreiltä saamme analogiset paluuarvot, jotka tuodaan mikroprosessorin 8bit ADC-muuntimen läpi ja saadaan lukemat väliltä 0 ja 1023. Eli lukema joka saadaan koskemattomasta sauvaohjaimesta on 512. Tästä se skaalataan selkeyden vuoksi niin, että keskelle jää 0 ja vasemmalle puolelle -512 ja oikealle +512. Y-suunta ei ole käytössä ollenkaan. Kuvassa 14 on esitettyä laitteen rungon sisäpuoli.



KUVA 12. Valmis kahva



KUVA 13. Lähikuva sauvaohjaimesta

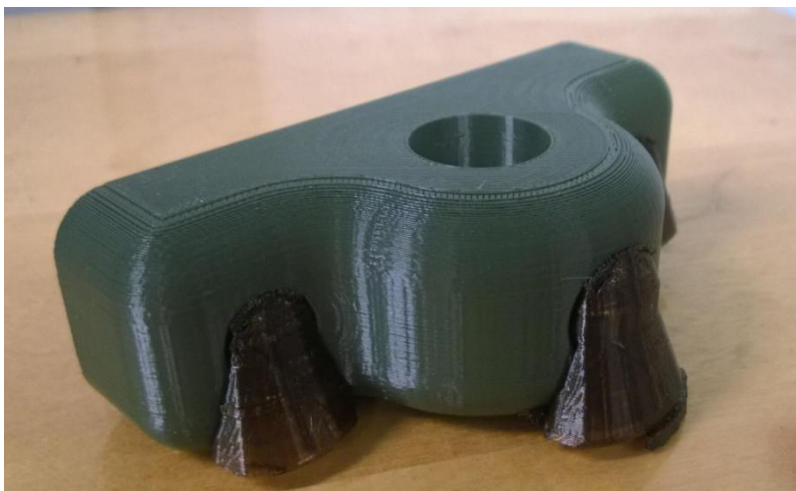


KUVA 14. Kahva ilman kantta

4.3 3D-Tulostus

3D-tulostusta varten tarvitaan vain idea, suunnitteluohjelma ja 3D-tulostin. Tulostimena toimi TAMK:n Dimension Elite printteri, suunnitteluohjelmisto oli AutodeskInventor 2013 ja idea muotoutui kahvaa tehdessä ja mittoja tietokoneella pyöritellessä. Suunnitteluohjelmalla aikaansaatu piirros talletettiin STL-muotoon, joka avattiin printterin omalla ohjelmistolla tulostuslabran tietokoneella. STL-muoto on 3D-tulostusta varten kehitetty tiedostomuoto, jota monet 3D-tulostimet vaativat ja se on alunperin Amerikkalaisen 3D Systemsin kehittämä. Tämän jälkeen printteriin asetettiin uusi puhdas pohjalevy ja kerrottiin tulostinohjelmalle mitkä kappaleet halutaan tulostaa ja miten päin. Sitten tarkastettiin vielä, että tuloste skaalautui oikein ja mitat olivat millimetreinä eikä tuumina. Tässä vaiheessa on myös tärkeää miettiä miten päin runko kannattaa tulostaa, jotta aikaa ja tukiainetta ei kulu turhaan. Pelkällä asettelulla on mahdollista voittaaunteja sekä euroja, tai hävitä niitä.

Runkokappaleemme oli iso ja siten siihen kului paljon ABS-muovia. Tämän takia päätimme kokeilla tulostimen tarjoamaa automaattista kennotusta ja näin säästää tulostuskustannuksissa. Kokonaistulostusaika kahvalle ja kannelle oli noin 38 tuntia. Myöhemmin parantelimme vielä kannen ominaisuuksia ja tulostimme sen toistamiseen. Lopputuloksesta tuli hyvä, ja rakenne on erittäin jäykkä, vaikka tiettyjä ennakkoluuloja ontelorakennetta kohtaan olikin. Tulostuksen jälkeen irrotimme kuvassa 15 esitetyn tukiaineen kolmen tunnin lipeäkylvyllä ja puhdistimme alustan seuraavaa käyttäjää varten.



KUVA 15. Kannakkeen tukiaine ennen lipeäkylpyä.

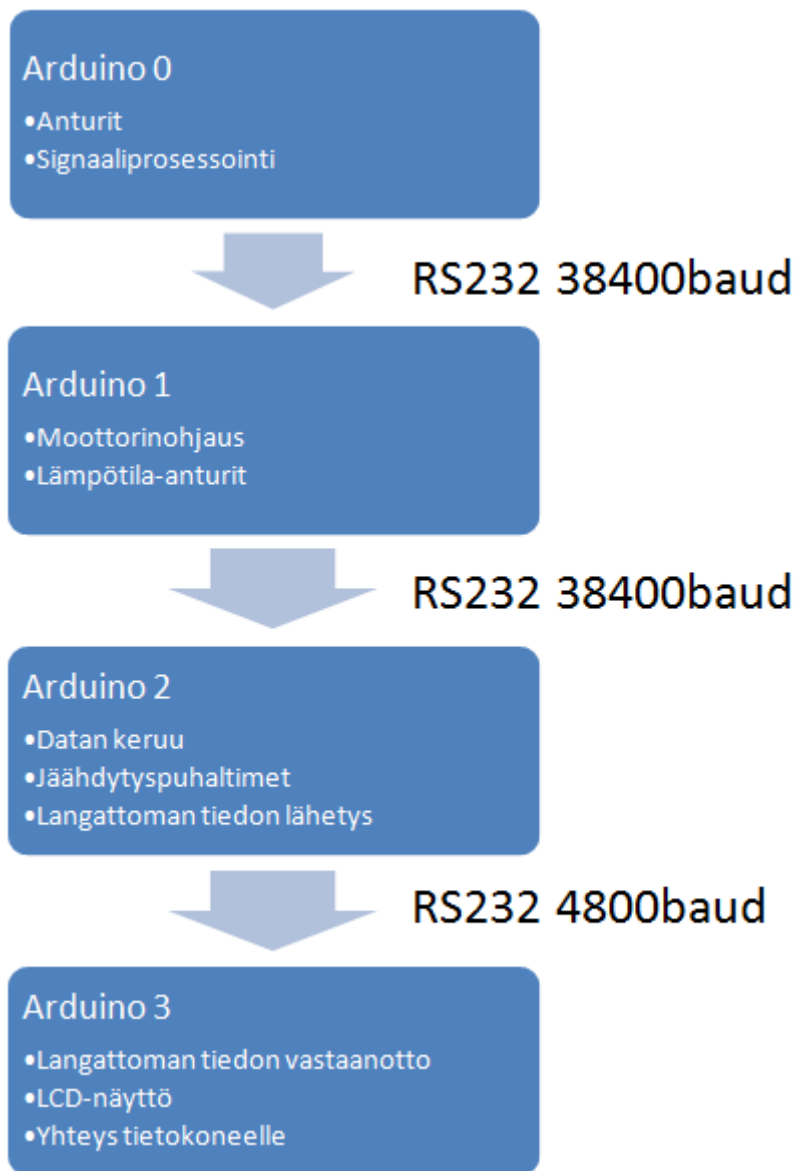
5 ELEKTRONIIKKA

5.1 Yleinen rakenne

Elektroniikassa päädyttiin ratkaisuun, jossa itse laitteessa on 4kpl arduinoUNOja eli ATMEGA328-mikroprosessoreita. Toiminnot päätettiin jakaa tasaisesti arduinojen kesken seuraavasti:

- Arduino 0: Antureiden luku ja signaalien käsittely.
- Arduino 1: Moottoreiden H-siltojen PWM-ohjaus
- Arduino 2: Datan lähetys langattomasti ja muut oheistoiminnot
- Arduino 3: Datan vastaanotto ja infonäytön ohjaus

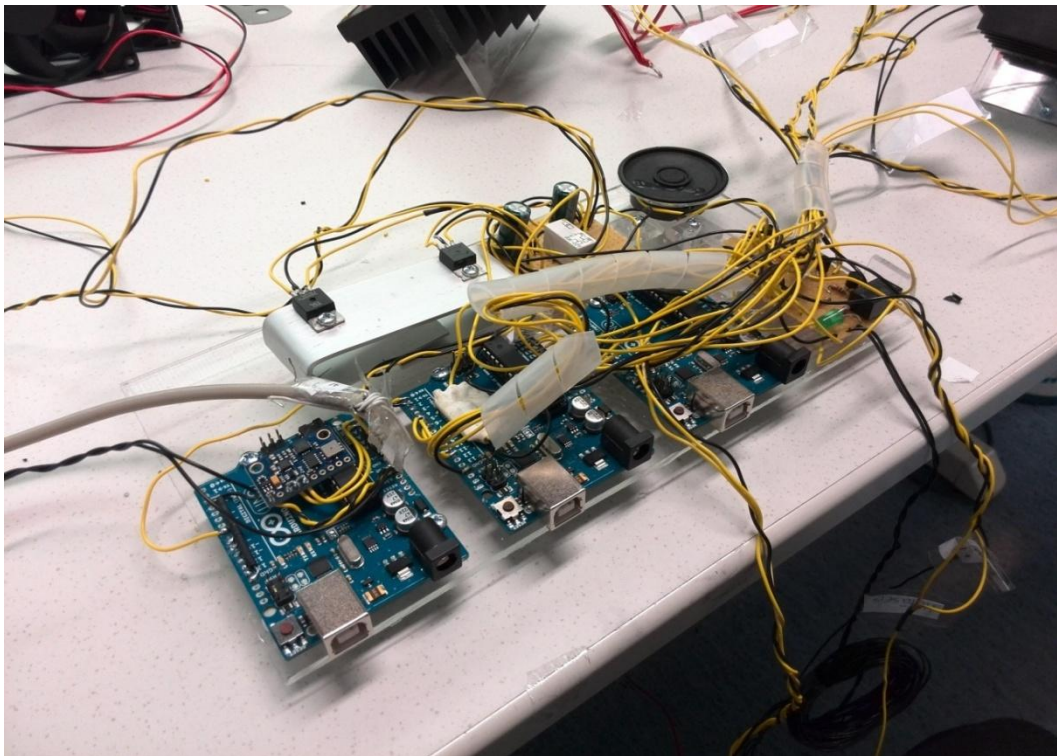
Monen mikroprosessorin käyttöä puolsi moni asia tässä projektissa. Suurin syy oli kuitenkin halu modulaariseen rakenteeseen, jossa toiminnot saadaan erotettua omiksi kokonaisuuksikseen. Modulaarinen rakenne on helpompi, jos tulee ongelmia ja niitä oletettiin tulevan. Koska arduinot keskustelevat väylän kautta oli niiden ohjelmointi myös mahdollista virtojen ollessa kytkettyinä. Kaikki muut arduinot on mahdollista päivittää virrat päällä, paitsi moottoreita ohjaava arduino 1. Virtojen päällä ollessa päivittäminen on nopeampaa, kuin virtojen jatkuva kytkeminen päälle ja pois testivaiheessa. Myös tämä oli tärkeä saavutettu etu usean mikroprosessorin rakenteessa. Kuvassa 16 on graafinen esitys elektroniikasta yleisesti.



KUVA 16. Prosessipuu toiminnasta.

Tämä rakenne mahdollistaa mahdollisimman nopean ohjelman kierron. Arduinojen välinen liikenne hoidetaan RS232-standardin mukaisella liikenteellä 38400baud nopeudella.

Ratkaisu myös paransi huomattavasti mahdollisuuksia luoda laitteesta turvallinen. Turvallisuus oli keskeinen syy siihen, että miksi moottorinohjaus haluttiin yhden erillisen Arduinon taakse. Dataa lähettävä langaton arduino taas on huomattavasti hitaampi ohjelmakierroltaan, kuin antureita lukeva. Periaatteessa olisi ollut mahdollista toteuttaa koko laite yhdellä Atmega 2560:lla tai vastaavalla enemmän I/O-linjoja tarjoavalla mikroprosessorilla, mutta tämän ratkaisun kanssa olisi jouduttu tekemään enemmän kompromisseja. 3x16Mhz tarjoaa enemmän laskutehoa kuin 1x16Mhz. Kuvassa 17 "emolevy" eli muoviplexi jolle on kiinnitetty suurin osa elektroniikasta.



KUVA 17. Pääelektroniikka "Emolevy"

5.2 Sähköinen mitoitus

Sähköinen mitoitus aloitettiin moottorien tarpeista. Moottoreita etsittäessä vaatimuksena oli kohtuullisen alhainen kierrosluku (0-300rpm) ja riittävälähtömomentti. Harjaton neodyymi-magnetoitu moottori olisi mahdollisesti ollut paras valinta nopeuden ja vääntömomentin suhteen, mutta sopivan kokoista moottoria ei löydetty. Servo ja vaihtovirtamoottorit olivat myös monella valmistajalla vääränlaisia tähän tarkoitukseen.

Tunnistimme myös heti sen faktan, että moottorin kuormitus tulee olemaan erittäin vaihtelevaa ja nykivää. Lopulta päädyimme Bosch GSR 18 VE-2-LI akkuporakoneen varaosamoottoriin ja planeettavaihteistoon. Kyseisen moottorin ja vaihteiston tekniset arvot olivat juuri sopivat, 0-400rpm ja 80Nm. Moottorien käyttöjännite on 18V ja maksimi teho n. 300W. (BOSH 2014.) Moottorien myös arveltiin kestävän hyvin voimakkaita suunnanvaihtoja ja rasituksia, koska ne kestävät myös ammattilaisten käyttöön tarkoitetussa akkuporakoneessa työmaaolosuhteissa. Kuvassa on moottorit ja vaihteistot asennettuina yhteen.



KUVA 18. Moottori, vaihteisto ja akseli kokoonpano

Annettujen reunaehtojen pohjalta (18V ja 600W) Lähdimme etsimään sopivia akkuja. Aluksi 18V:n käyttöjännite vaikutti hieman ongelmalliselta, koska useimmiten lyijyakkujen jännite on 12V tai 24V. Löysimme kuitenkin sopivan kokoisia 6V:n 12Ah:n lyijyakkuja jotka totesimme kelvollisiksi.

Kun moottorit ja akut oli valittu aloimme suunnitella moottorinohjaimia, jotka pystyisivät hallitsemaan hyvin 300W jatkuvaa tehoa. Kyseisten moottorinohjainten todettiin myös tarvitsevan erittäin korkea hetkellinen virranantokyky, koska moottorit tulee saada liikkeelle nopeasti tasapainottamista varten. Akustolta tulevat päävirtajohdot ovat suurimman kuormituksen alla. Johtimien pituudet ovat kuitenkin hyvin lyhyitä, joten jännitteen putoamista johtimessa on turha ottaa huomioon. 33A:n virran takia akkujen pääjohdot päätettiin tehdä kahdesta $1,5\text{mm}^2$ johtimesta. Johdinkokoon päädyttiin, koska Bosch on käyttänyt moottoreissaan $1,5\text{mm}^2$ johtoja. Tällöin tuplakokoinen johdotus akustolta on varmasti riittävästi. Muutoin sähköinen mitoitus oli helppoa, koska ainoastaan akuilta H-siltojen kautta moottoreille kulki suuria virtoja. Kaikki muut johdotukset olivat lähinnä signaalijohtoja tai muuten hyvin alhaisen tehon siirtäviä.

Lisäksilaite tarvitsi reguloidun 12V:n jännitelähteen tuulettimia ja arduinojen 5V:n regulaattoreita varten. Arduinoja voi periaatteessa syöttää täydellä 18V:n akkujännitteellä, mutta ladattujen akkujen jännite on lähes 22V:a ja siksi tämä ei ole suositeltavaa. Pienet 7805-sarjan regulaattorit lämpenevät huomattavasti, kun jännitepudotus regulaattorin yli on 17V:a. Suurempi jännitepudotus tarkoittaa samalla virralla suurempaa lämpökuormaa regulaattorille teholauseen $P=UI$ mukaisesti. Tämän takia elektroniikkaa (arduinot, puhaltimet, yms.) päätettiin ajaa kahden 7812-regulaattorin läpi. Näitä regulaattoreita jäähdytetään erillisillä alumiinisiileillä. Näin voitiin varmistua siitä, että mikään jännitteen regulointiin vaikuttava osa ei ylikuumene missään tilanteessa.

5.3 Akusto ja laturi

Akuston jännitevaatimus oli 18V moottoreiden määrittämänä. Varsinaista kapasiteettimäärittystä ei ollut. Suurempi kapasiteetti tarkoittaa tietysti suurempaa toimintamatkaa. Toinen ehto oli se, että akuston täytyy kestää n. 600W:n kuorma ilman, että jännite romahtaa täysin. Valitsimme CT-leader nimisen valmistajan CT12-6-akut, jotka ovat 6V:n ja 12Ah:n akkuja. Akuston kapasiteetiksi tuli siis 18V 24Ah. (CT-leader 2014.) Jos moottoreiden keskikuormitus on n. 50% (300W) voidaan arvioitu käyttöaika laskea seuraavasti:

$$300\text{W} / 18\text{V} = 16,67\text{A}$$

$$24\text{Ah} / 16,67\text{A} = 1,44\text{h}$$

Tämä laskelma ei ota huomioon akuston jännitteen laskua, mutta se toimii alustavana mitoitusperiaatteena hyvin. Noin puoleltoista tunnin käyttöaika kuulostaa hyvältä, ja tämän kokoinen akusto on vielä suhteellisen kompakti mitoiltaankin. Akuston painoksi tuli n.12kg. Akkujen maksimaalisen purkutehon ollessa n.600W tulee maksimi purkuvirraksi 33A. Tällöin yhteen akkuun kohdistuu 16A purkuvirta, jota voidaan pitää kohtuullisena 12Ah:n akulle. Kuvassa 19 on 6kpl CT12-6-akkuja.



KUVA 19. Akusto.

Suurempi toimimatka ja pienempi paino olisi saavutettu LiPo- tai Li-Ion akuilla, mutta ne olisivat vaatineet erillistä elektroniikkaa, jotta niiden käyttö olisi ollut turvallista. Huolimatta suojapiireistä, litium-akut ovat lähes poikkeuksetta vaarallisempia kuin “tavalliset” lyijyakut. Ajattelimme tässä laitteen turvallista huoltoa ja käyttöä myös myöhemmin, kun tuntematon henkilö huoltaa sitä. Kaikki nämä seikat puolsivat lyijyakkujen käyttöä.

Laturiksi haluttiin mahdollisimman helppokäyttöinen laturi ja niin päädyttiin tavalliseen auton akun automaattilaturiin. Ongelmana oli kuitenkin 18V:n käyttöjännite itse laitteessa ja laturin 12V:n jännite. Tämä ongelma päätettiin ratkaista tekemällä laitteeseen ja laturiin eri tavalla silloitetut liittimet. Itse laitteen liitin kytkee akuston kahteen rinnakkaiseen kolmen akun sarjaan (3 series 2 parallel kytkentä). Laturissa taas liitin kytkee akuston kolmeen rinnakkaiseen kahden akun sarjaan (2 series 3 parallel). Kuvassa 19 on esitetty akusto ja sähkökuva sen liittimistöä löytyy liitteestä yksi.

Liittimeksi akustolle vaadittiin liitin, joka kestää 33A virran ja josta löytyy riittävän monta kytkentänapaa silloittaa akusto laturiin ja itse laitteessaliitteen 1 kuvan mukaisesti. Valitsimme Molex-5559-20-liittimen, joka on monelle varmasti tuttu tietokoneen virtalähteen ja emolevyn välisenä liittimenä. Tämä 20-pinninen liitin pystyy siirtämään virtaa hieman laadusta riippuen 6-8A / pinni. Kuten sähkökuvasta käy ilmi, niin täysi 33A virta johdetaan neljän kytkentäpinnan läpi, eli jatkuva suurin mahdollinen virta on sallitun rajoissa. Yhden akkusarjan läpi kulkeva virta on n. 16A, ja se voidaan johtaa kahden kytkentäpinnan läpi. Laturin latausvirta on alle 5A, joten sen osalta ei tarvitse huolehtia liittimen virrankestosta.

5.4 Anturit

Aikaisemmista kokemuksista johtuen antureiksi päätettiin valita digitaaliset anturit. Analogisten antureiden lukeminen on hitaampaa ja ne ovat alttiimpia mm. EMI-häiriöille (ElectroMagneticInterference). Digitaalisten antureiden hyöty on myös se, että itse anturissa on jo mukana mikroprosessori, jolla on mahdollista esikäsitellä tieto. Monet digitaaliset anturit sisältävät nykypäivänä jo säädettävät yli- ja alipäästösuotimet, säädettävän näytteenottotaajuuden, mittausalueen säädön yms. Digitaalisten antureiden ainoa ehkä huonompi puoli on se, että ne käyttävät tiedonsiirrossa väylätekniikkaa (RS232, I2C yms.). Pohdinnan jälkeen päätettiin anturitekniikka toteuttaa seuraavalla tavalla:

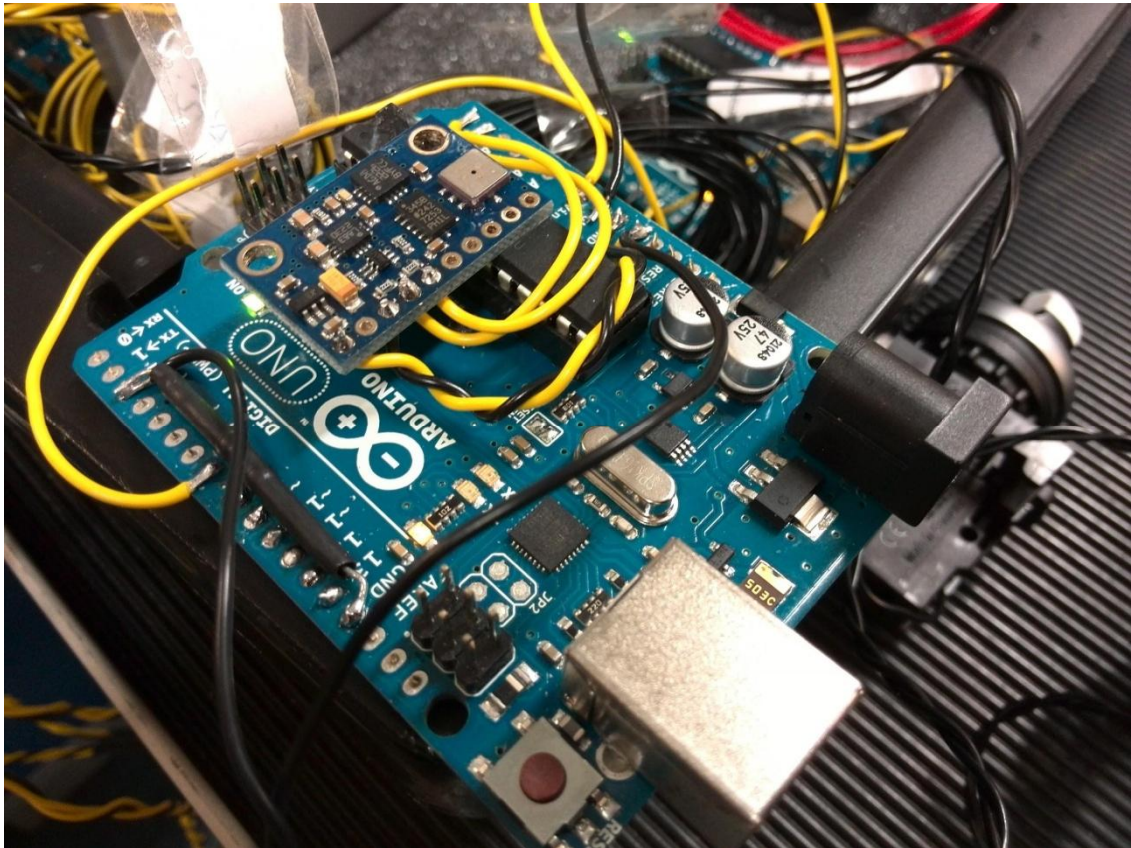
- GY-80 IMU (Inertial Measurement Unit)
 - ADXL345 3-akselinen kiihtyvyyssanturi
 - L3G4200D 3-akselinen gyro
 - (HMC5883L)
 - BMP085

-TMP36GZ (4kpl Atmega328:n ADC kautta)

- Atmega328:n oman ADC muuntimen kautta akuston jännite mitta

Tasapainotuksen pääanturina päätettiin käyttää GY-80 IMU-levyä, jolle on koottu ADXL345-kiihtyvyyssanturi ja L3G4200D-gyro. GY-80-levyltä löytyy myös HMC5883L-magnetometri ja BMP085-paineanturi, mutta niitä ei käytetty tällä kertaa. GY-80-IMU:lle on kerätty useampi I2C-väylää käyttävä anturi kompaktiin kokoon.

Täysin samaan päämäärään pääsisi myös käyttämällä erillisiä antureita, mutta mm. johdotustapustytään yksinkertaistamaan huomattavasti, kun kaikki viedään samalle levylle. Kuvassa alla on GY-80 IMU. I2C-väylän kommunikointi tapahtuu standardin mukaisesti kahdella johtimella; SCL (Clock) ja SDA (Data). Tietysti myös nastat +5V ja GND tarvitaan. Vertailukohtana analogiset kolmiakseliset anturit olisivat vaatineet vähintään kahdeksan johdinta.



KUVA 20. GY-80 IMU kiinnitettyä Atmega328 mikroprosessorin päälle.

5.4.1 ADXL345-Kiihtyvyyssanturi

ADXL345 on 3-akselinen digitaalinen kiihtyvyyssanturi. I2C-väylän kautta sen mittausalue on asetettavissa johonkin seuraavista alueista: +/-2g, +/-4g, +/-8g, +/-16g. 10-bittisellä tarkkuudella tuleva arvo on luettavissa ADXL345:n muistista kahdesta 16-bittisestä muistipaikasta. Käytännössä tämä tarkoittaa sitä, että kun lasketaan kallistuskulma painovoimasta (putoamiskiihtyvyys), pystytään saavuttamaan alle yhden asteen tarkkuus. ADXL345 on täysin digitaalinen anturi, mikä tarkoittaa käytännössä, että se sisältää oman ADC-muuntimen ja mikroprosessorin. Se ei myöskään ole suoraan toimintavalmis ilman tiettyjä toimenpiteitä. Toiminta on käynnistettävä erillisillä komennoilla väylän kautta ja samalla tehtävä asetukset. ADXL345:n alustukset ohjelmassamme näyttävät seuraavilta:

```
void ADXL345Initialize()
{
for(int x = 1; x < 4; x++)
{
Wire.beginTransmission(ADXL345_Address); // Begin transmission
Wire.write(0x2D); // write to register 0x2D on ADXL345
(power register)
if(x == 1) { Wire.write(0); } // Wake up
if(x == 3) { Wire.write(8); } // Measure
Wire.endTransmission(); // End transmission
}
}
```

ADXL345:n käynnistys vaatii sen, että muistiosoitteen 0x2D bitit 0 ja 3 kirjoitetaan tilaan "1", jolloin anturi ei mene lepotilaan ja käynnistää mittauksen. Muita bittejä ei ole pakko asettaa, jos tyydytään oletusarvoihin. Tämä tarkoittaa 100Hz:n näytteenottotaajuutta ja +/-2g skaalan osalta. Tämän jälkeen voidaan lähteä lukemaan anturin mittaustuloksia. Mittaustulosten luku tapahtuu tyypilliseen tapaan I2C-väylän protokollan mukaisesti. Luettavia lukuarvoja on kaksi jokaista akselia kohden, ns. "highbytes" ja "lowbytes". Nämä ilmaisut tarkoittavat 16-bittisen lukuarvon kahdeksaa suurinta ja kahdeksaa pienintä arvoa vastaavia bittejä. Luku on jaettu kahteen muistipaikkaan sen takia, että mikro-ohjaimienresoluutio on vielä nykyäänkin 8-bittiä,

vaikka tietokoneet toimivatkin jo 64-bittisillä järjestelmissä. Luettavia muistialueita on siis kaiken kaikkiaan 6kpl. Tämän jälkeen lukuarvot yhdistetään, jolloin saadaan kyseisen akselin kiihtyvyyssarvo. Yhdistäminen on yksinkertainen toimenpide, jonka voi tehdä "bitshift"-tyyppisesti helposti. Vastaava ohjelma on seuraavan näköinen:

```
void ADXL345Refresh()
{
  Wire.beginTransaction(ADXL345_Address); // adxl345 address 0x53
  Wire.write(0x32); // read from 0x32
  Wire.endTransmission();

  Wire.requestFrom(ADXL345_Address, 6); // request 6 bytes from ADXL345

  int x = 0;
  while(Wire.available()) // Check data availability
  {
    Abuffer[x] = Wire.read(); // receive 1 byte at one loop round
    x++;
  }
  Wire.endTransmission();

  Ax = (((int)Abuffer[1]) << 8) | Abuffer[0]; // Convert to accel integers
  Ay = (((int)Abuffer[3]) << 8) | Abuffer[2];
  Az = (((int)Abuffer[5]) << 8) | Abuffer[4];
}
```

Näihin kolmeen muuttujaan saadaan luettua kunkin akselin kiihtyvyyssarvo väliltä 0-2g. Tämän perusteella voidaan laskea painovoiman summavektori, josta saadaan laitteen kallistuskulma painovoimaa vasten selville. Tämä on yksi koko laitteen toiminnan tärkeimmistä elementeistä.

5.4.2 L3G4200D-gyro

Koska kiihtyvyyssanturi on hyvin herkkä myös tärähdyksille ja kaikille muille liikkeille, jotka aiheuttavat kiihtyvyyksiä sen apuna tarvitsee olla myös kolmiakselinen gyro. Kiihtyvyyssanturin arvoja lasketaan juoksevana keskiarvolaskentana n. 1-2 sekunnin ajanjaksolla. Gyrojen tehtävänä on täyttää tämä n. 1-2 sekunnin viive kiihtyvyyssantureiden reagoinnin välillä. Gyrojen tehtävä on myös pitää laitteen kulkusuunta suorana silloin, kun käyttäjä ei ohjaa käännöksiä. Teoriassahan laitteen kuuluisi mennä suoraan, jos moottoreille syötetään täsmälleen sama teho. Käytännössä tämä ei kuitenkaan toteudu, koska moottoreiden sisäiset kitkat, pyörien vierintävastukset ja kaikki muutkin todellisen maailman muuttujat estävät laitteen kulkemisen suoraan. Tämä ongelma korjataan tarkkailemalla gyron pystyakselin kiertoa ja säätämälle sen mukaan moottoreiden välinen tehosuhte sopivaksi, jotta laite kulkee suoraan. Tämä tarkoittaa myös sitä, että laite kulkee suoraan myös sivuttaissuunnassa kaltevalla pinnalla.

Myös gyroksi valittiin digitaalinen 3-akselinen gyro L3G4200D. Kyseinen gyro toimii I2C-väylässä TTL-tasoisilla signaaleilla (3.3V). Gyro on samalla tavalla kuin kiihtyvyyssanturikin ohjelmoitavissa väylänsä kautta, jopa hieman laajemmin kuin kiihtyvyyssanturi. Gyron valittavana skaalana on +/-250, +/-500 tai +/-2000dps. DPS on lyhenne sanoista Degree Per Second eli astetta sekuntia kohden. +/-250dps skaalalla mitattava maksimiarvo on siis 250 astetta pyörähdystä sekuntia kohden. Vastaavasti tarkkuus laskee, kun skaala kasvaa. Tämä johtuu siitä, että mittaustarkkuus ei muutu skaalan mukana. 16-bittisellä mittaustarkkuudella tämä tarkoittaa 250dps skaalalla 0.00875 astetta / sekuntia kiertymän havaitsemista. Kyseessä on siis erittäin tarkka ja herkkä gyro.

Tällaisella herkkyydellä varustetun anturin signaali voi sisältää paljon häiriöitä, ellei sen kaikkia sisäisiä ominaisuuksia käytetä juuri kulloiseenkin tilanteeseen sopivalla tavalla. Alla on esitettyalustuskomennot L3G4200D-gyrolle.

```
void L3G4200DInitialize()
```

```
{
```

```
for(int x = 32; x < 37; x++) // Loop addresses 0x20 --> 0x24 (Control registers 1 -> 5)
```

```
{
```

```

Wire.beginTransmission(L3G4200D_Address); // start transmission to device
Wire.write(x); // Write to control register
if(x == 32) { Wire.write(0b01001111); } // Enable x, y, z and turn off power down. ODR
200Hz, Cut-off 12.5Hz
·if(x == 33) { Wire.write(0b00011000); } // HPF SET 0.05Hz, "Reference signal for
filtering)
if(x == 34) { Wire.write(0b00000000); } // No interrupts
if(x == 35) { Wire.write(0b00010000); } // Scale: 250dps (0b00000000), 500dps
(0b00010000), 2000dps (0b00110000)
if(x == 36) { Wire.write(0b00010001); } // FIFO disable, HIGHPASS Enable
Wire.endTransmission(); // end transmission
}
}

```

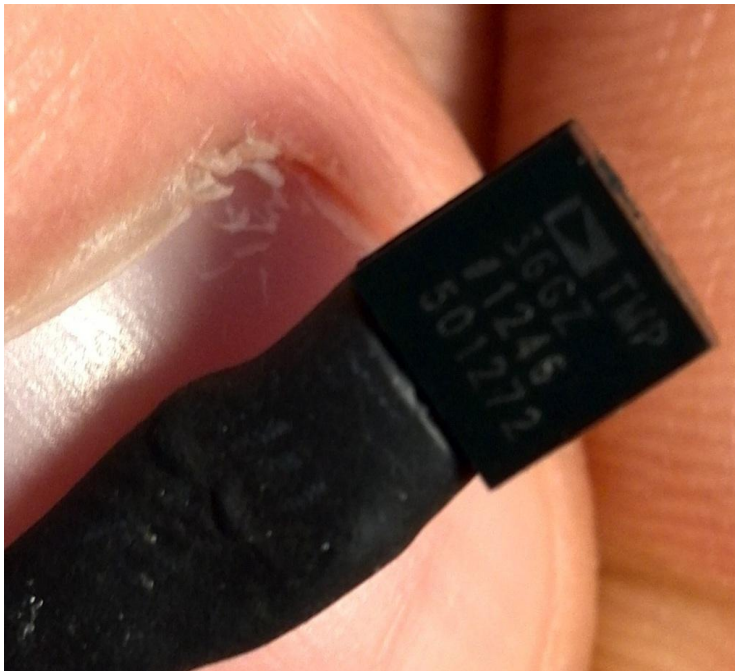
Aluksigyroa käsketään käynnistämään jokaisen akselin mittaus, sammuttamaan virransäästö ja asettamaan datanopeus arvoon 200Hz (ODR = Digital output data rate). Tämän jälkeen digitaalinen ylipäästösuodin asetetaan 0,05Hz:n kohdalle. Tämä on erittäin tärkeätä, koska tällä keinoin pystytään suodattamaan sähköiset yms. häiriöt lähes täydellisesti pois. Alle 0.05Hz alueella tapahtuvien kallistuksen muutoksien menettäminen signaalista on lähes merkityksetöntä. Sisäisiä keskeytyksiä ei tarvita, joten ne tulee kytkeä pois päästä ja skaalaksi asetetaan 500dps. Sen jälkeen FIFO muisti kytketään pois ja ylipäästösuodin laitetaan päälle.

Kuten huomataan L3G4200D-gyron säätömahdollisuudet ovat melko suuret, mutta kuitenkin yksinkertaiset. Käyttämällä varsinkin suotimia oikein saadaan anturilta jo valmiiksi hyvää signaalia, ja mikroprosessorilla suoritettava signaalinkäsittely saadaan pidettyä minimissään.

5.4.3 TMP36GZ -lämpötila-anturit

Laitteeseen päätettiin asentaa neljä lämpötila-anturia tarkkailemaan tärkeimpiä ja mahdollisesti kuumimpia kohtia, eli molempia moottoreita ja moottorinohjaimia. Koska tällä lämpötilatiedolla ei ole suurta tarkkuusvaatimusta tai suurta päivitysvälivaatimusta, niin päädyttiin yksinkertaisiin analogisiin antureihin. TMP36GZ-anturi on pieni ja antaa

lineaarisen jännitesignaalin lämpötilasta. Alla kuva 21, jossa sen kokoa voi verrata peukalon kynteen.



KUVA 21. TMP36GZ Anturi

TMP36GZ Lämpötila-anturi on erittäin helppo käyttää. Siinä on kolme liitännästä, jotka ovat +V, GND ja Signaali. Käyttöjännitteeksi käy mikä vain väliltä 2.7-5.5V. Jänniteensyötössä on kuitenkin huomioitava se, että anturin käyttämä virta on erittäin pieni. Tämä johtaa herkästi häiriöihin jos virtajohtimet ovat pitkät ympäristössä, missä on paljon sähkömagneettisia häiriöitä. Anturin lähtösignaali on lineaarinen jänniteviesti lämpötilaan verrattuna. +25C:n lämpötilassa anturi antaa ulos 0.75V ja lämpötilan muuttuessa asteen jännite muuttuu 10mV. (10mV / Cdeg). Kovin tarkka lämpötila-anturi TMP36GZ ei ole, mutta jos n. 1-2C-asteen tarkkuus välillä -40 - +125C riittää, niin se on oivallinen valinta. Asensimme molempiin moottorinohjaimiin ja moottoreihin kuhunkin yhden anturin. Tällöin kahta jäähdytyspuhallinta voidaan käyttää tarvittaessa, mutta niiden ei tarvitse pyöriä kokoajan. Lämpötila-anturit luettiin Atmega328-mikroprosessorin oman ADC-muuntimen kautta. Seuraavalla ohjelmanpätkällä suoritettiin vaadittavat laskut, jotta jänniteviesti saatiin tallennettua lämpötila-arvona muuttuinaan.

```
int temp1 = int((((analogRead(A1)*0.0048828)-0.75)/0.01)+25);
```

Sama toistettiin jokaiselle lämpötila-anturille eri AI pinnissä. Lämpötila tallennettiin kokonaislukumuuttuja muotoon, koska tarkempia desimaaleja ei tarvittu.

Muunnoskaava purettuna matemaattiseen muotoon on seuraavanlainen:

8-bit tarkkuus on 1024 ADC-porrasta.

X = Anturin arvo ohjelmassa.

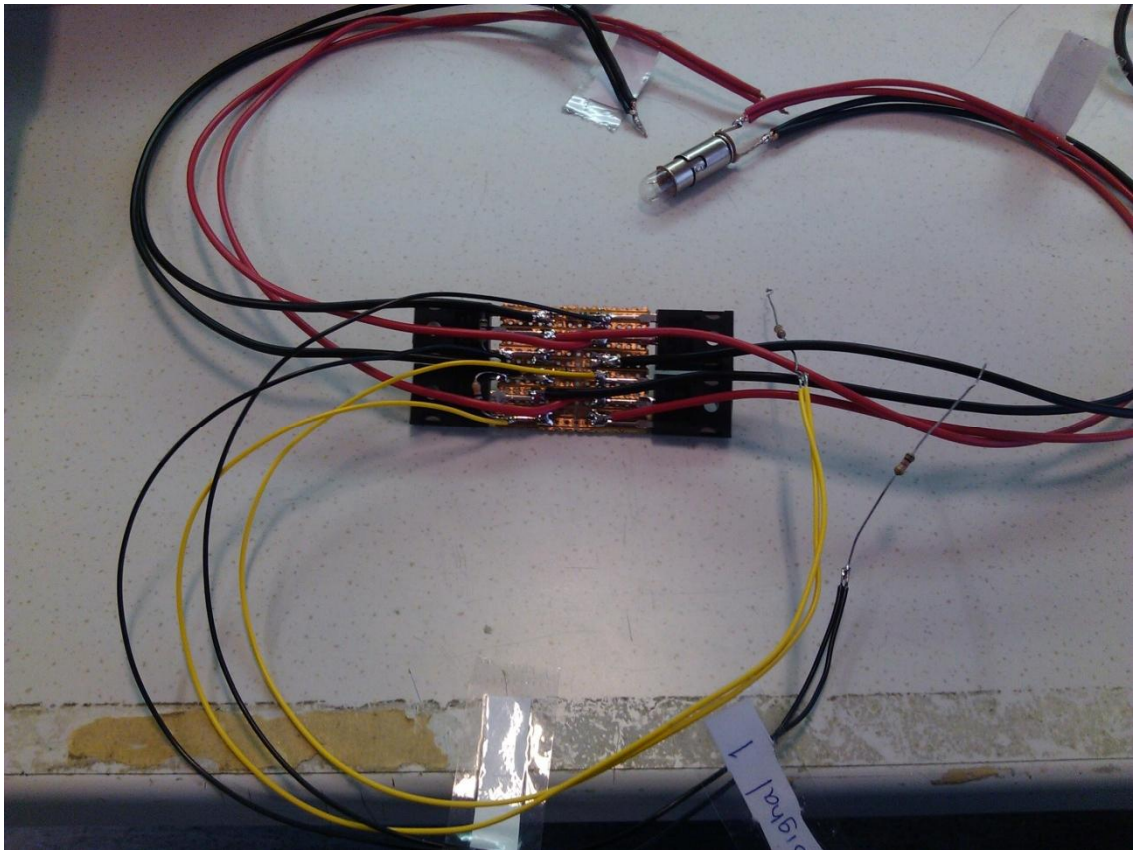
Y = Lämpötila

$$Y = (X * (5V / 1024) - 0.75V / 10mV) + 25$$

Käytännössä siis ensin muutetaan ADC-arvo jännitteeksi, josta vähennetään 0.75V ja jaetaan sitten 10mV:lla, johon täytyy vielä lisätä 25C arvo ja tulokseksi saadaan lämpötila. Tämän antureiden luvun ja muunnokseen tekee moottorinohjaimia käyttävä arduino, joka sitten lähettää tiedot RS232-väylää pitkin eteenpäin. Lämpötila-anturit kytkettiin tähän arduinoon, koska siinä oli eniten vapaita I/O pinnejä. Käytettäessä väylää mikroprosessorien välillä ei ole varsinaisesti merkitystä mihin mikroprosessoriin on kiinnitetty mitään.

5.5 Moottorinohjaimet

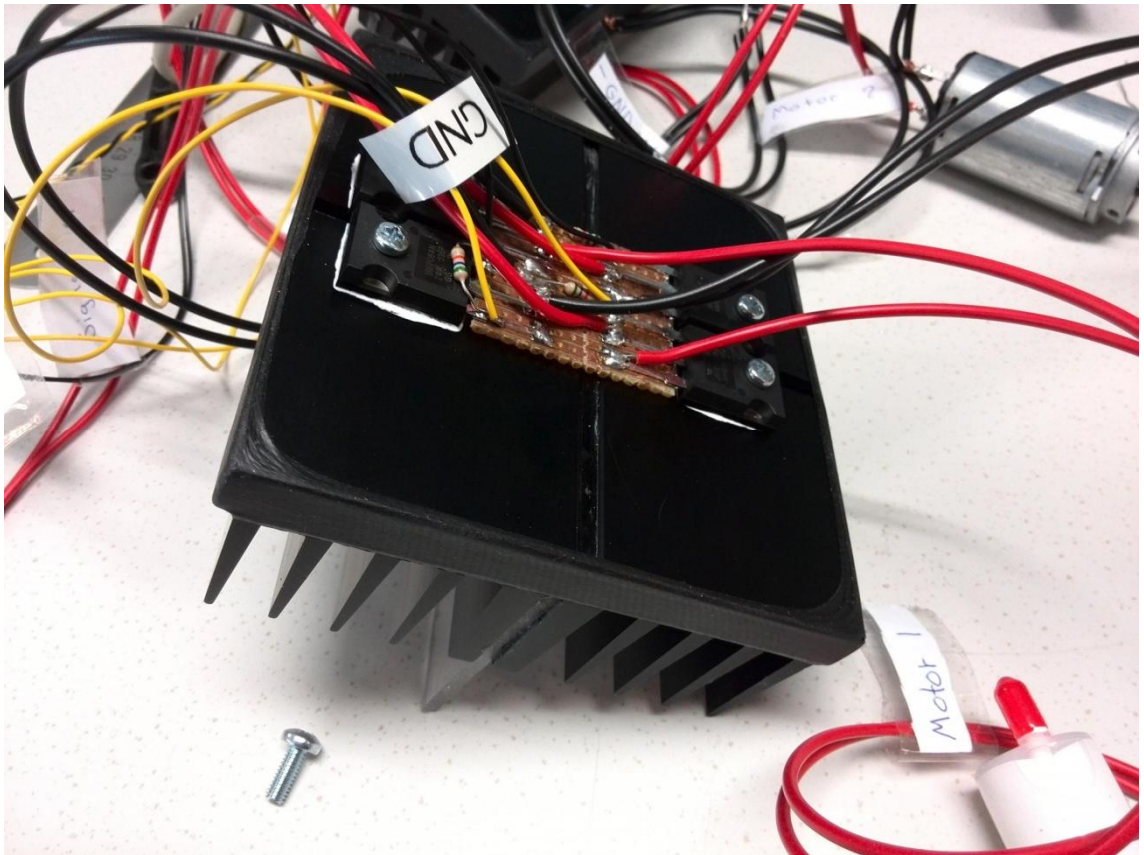
Moottorinohjaimet päätettiin ylimitoittaa rajusti johtuen kuormituksen luonteesta. Harjalliset melko tehokkaat sähkömoottorit voivat ottaa hetkellisesti melko korkeita piikkivirtoja, kun yrittävät kääntää koko laitteen ja päällä olevan ihmisen massaa takaisin tasapainoon. Moottorinohjaimien peruskytkentä on aivan tavallinen H-silta mosfeteillä toteutettuna. H-siltaan valittiin 4kpl IRFP064N-fettiä. Kyseiset fetit ovat erittäin järeitä suurien virtojen ohjaukseen soveltuvia puolijohteita. Niiden maksimi toiminta-arvot ovat 55V ja 110A. Myös lämmön kesto on 175C asti ja 200W lämpökuormaan. (IRF MOSFET 2014.) Testauksen aikana tapahtunut vahinko, jolloin moottorinohjaimien läpi ajettiin 2kW:n teho suoraan oikosulkuun. Ohjaimet eivät vahingoittuneet millään tavalla. Kuvassa 22 näkyy valmis moottorinohjain ilman jäähdytysiitä.



KUVA 22. Valmis H-silta ilman jäähdytysiiltä.

Fettien ohjaaminen päätettiin ratkaista yksinkertaisella transistorikytkennällä. Liitteessä 2 on esitetty transistoriohjaimen ja H-sillan piirikaavio, jossa jokainen transistori ohjaa omaa fettiparia niin, että virtaa voidaan ajaa moottorissa molempiin suuntiin, vaikka käytössä on yksipuolinen jännitelähde. Fettiparit aktivoituvat transistorien ollessa sulkutilassa. Koska fettejä käytetään vain kiinni- ja auki-tiloissa, ei sillassa esiinny periaatteessa juurikaan lämpöhäviöitä. Tämä siksi, että kyseisten fettien nimellinen ON-tilan resistanssi on vain 0.008ohm.

Moottorinohjaimia ohjataan mikroprosessorin generoimalla PWM-signaaliilla, jonka taajuus on n. 1 Khz. PWM-signaalin tarkkuutena on 8 bittiä eli 256 porrasta eteen ja taakse. Tämä tarkkuus on enemmän kuin riittävästi Sebawen tapauksessa. Moottorinohjaimien mosfet:ien pull-up vastukset päätettiin vielä releyttää ja ohjata releen toiminta hätä-seis ja ON/OFF kytkimen kautta. Tällöin hätä-seis katkaisee välittömästi moottoreille syötettävän virran, huolimatta PWM-signaalista. Kuvassa 23 on valmis H-silta jäähdytys-siiliin kiinnitettynä.



KUVA 23. Valmis mosfet H-silta.

Moottorinohjaimen ensimmäinen versio ei toiminut täydellisesti. Se oli suunniteltu hieman väärin. Ongelma ilmeni lopulta testivaiheessa, kun moottorinohjain kytkettiin varsinaiseen sebawe:n moottoriin. Pienemmillä moottoreilla ohjain oli toiminut hyvin. Ongelma johtui siitä, että kuorman molemmin puolin H-sillassa oli käytetty IRFP064N fettejä, jotka ovat N-kanavan mosfettejä. N-kanavan fet on johtavassa tilassa, kun HILA-LÄHDE jännite on yli tietyn raja-arvon. IRFP064N feteillä tämä rajajännite on n. 2-4V. Myös se, että moottorinohjaimia testattiin pienillä kuormilla johti erehdykseen. IRFP064N mosfettien resistanssi on valmistajan ilmoittamana n. 0.008 ohmia johtavassa tilassa. Tällöin jos kuorman resistanssi on yhden ohmin, niin kuorman yli vaikuttaa yli 99%:sti sama jännite kuin on syöttöjännite. Tästä seurausta on se, että ns. yläpuolen (kuorman positiivisen puolen) mosfetin LÄHDE pinnin jännite maahan verrattuna on lähes sama kuin akuston jännite. Näin ollen vaikka mosfetin hila olisi kiinni suoraan akuston positiivisessa navassa pysyy mosfet silti kiinni eli on johtamattomassa tilassa.

Ongelmaan on olemassa muutama yleinen ratkaisutapa, joista ensimmäinen on ns. ”bootstrap” kytkentä, jossa mosfetin hilan lähettyvillä on ylimääräinen kondensaattori. Tämä kondensaattori varataan toisen PWM-jakson aikana, ja sitten kytketään sarjaan jännitelähteen kanssa, jolloin hilan jännite saadaan nostettua yli syöttöjännitteen. Kyseinen kytkentä on kuitenkin tietyissä tapauksissa hieman hankala, koska kondensaattorin koko ja purkausajat yms. täytyy määrittää hyvin tarkasti mm. PWM-taajuuden suhteen. Hylkäsimme siis tämän idean ja päädyimme toiseen ratkaisuun, jossa piiriin tuodaan kiinteä jännitetuplain. Näin mosfettien hiloja voidaan tarvittaessa ajaa akkujännitettä korkeammalla jännitteellä ja varmistaa niiden pysyminen johtavassa tilassa. Koska moottorin resistanssi voi kuitenkin muuttua sen kommutaattorin asennon vaihtuessa, niin piti varmistaa myös ettei mosfettien hila-jännite nouse yli sallitun +/-20V:n. Jännitetuplain säädettiin syöttämään 36V jännitettä, joka sitten ajetaan vastusjännitejaon kautta mosfetin hilaan. Toiminta perustuu siihen, että jos mosfetin lähde on maassa, niin hilan jännite on $(56 / 124)\text{ohm} * 36\text{V}$ eli n. 16V:a riippumatta akkujännitteestä. Jos mosfetin lähde-jännite on lähes syöttöjännite, niin hilajännite on $(56 / 124)\text{ohm} * 36\text{V} + 18\text{V}$ eli n. 26V:a. Tällöin mosfet pysyy auki myös tässäkin tilanteessa koska HILA-LÄHDE jännite on n. 8V, joka ei riko mosfettia. Myös aikaisemmin käytetyt BC548 transistorit päätettiin vaihtaa BC547 malliin, jonka etuna on korkeampi jännitteenkesto, mikä oli tarpeen jännitetuplauksen vuoksi.

Tämä kytkentä ei toiminut alkuun, mutta lopulta syyllisiksi paljastuivat tehtaalla rikkinäisenä tullut erä BC547 transistoreita. Aikataulukkioiden vuoksi uusia BC547 transistoreita ei ehditty hankkia, mutta saimme konelaboratorion elektroniikkavarastosta korvaavan BC639 transistorin, joka sopi arvojen puolesta hyvin kytkentään. Kun vaihdoimme kytkentään toimivat transistorit se alkoi toimia välittömästi täysin oletetulla tavalla.

5.6 Langaton data

Koska laite on liikkuva ja se on akuilla toimiva päädyttiin tiedon monitoroinnissa langattomaan ratkaisuun. Kaapelin käyttö tiedonsiirrossa olisi ollut käytännönsyistä hieman hankalaa. Lähetettävän tiedon määrä oli myös melko pieni, joten todettiin yksinkertaisen ja tiedonsiirtonopeudeltaan melko hitaankin langattoman yhteyden riittävän hyvin.

Langattoman datalinkin rakennuksessa on syytä ottaa myös huomioon Suomen lainsäädäntö. On olemassa useampia lupavapaita taajuusalueita, joissa tietyn tehoisia radiolähtettä voi käyttää ilman erillisiä lupia. Valitsimme yleisimmän, eli 433-434Mhz kaistan, jolla saa lähettää jatkuvaa dataa pienellä lähetysteholla. Tarkempi erittely sallituista lähetystehoista ja asetuksista löytyy lähteestä neljä, joka on viestintäviraston määryys kyseisistä laitteista. Laitteen datalähete ei ole jatkuvaa ja sen lähetysteho on hyvin pieni (n.1-10mW) ja antennin vahvistus arviolta alle 3dB. Kantomatkatesteissä todettiin, että sisätiloissa lähete kantaa maksimissaan n. 50m ja läpäisee huonosti seiniä. Rakennuksesta ulospäin suuntautuvan radiolähteen teho on siis olematon. Avoimissa tiloissa voidaan odottaa maksimissaan n. 100-200m kantomatkaa. (Viestintävirasto 2014.)

Käytetty 434Mhz radiolähtetin & vastaanotinpari käyttää ASK (AmplitudeShift-Keying) muotoa radiolähteessä. Tämä on hyvin yksinkertainen kantoaallon amplitudia vaihteleva lähete. Tietoa päätettiin lähettää RS232 standardilla, joka ei varmasti ole paras mahdollinen ratkaisu langattoman datan lähetykseen, koska sen virheensieto on huonoa. Se on kuitenkin yksinkertainen ja riittävän luotettava lyhyillä etäisyyksillä. Maksimi nopeudeksi todettiin 4800 baud eli 9600 bittiä sekunnissa. Tällä nopeudelle ja lähetettävällä tiedonmäärällä päästiin n. 3Hz virkistystaajuuteen kaikissa arvoissa.

Antenneina käytettiin yksinkertaista sopivan mittaista johdon pätkää. Yhden elementin antennina käytettiin 1/2-, 1/4- tai 1/8-aallon antennia yleisesti. Suurin vahvistus on 1/2-aallon antennilla ja pienin 1/8-aallon antennilla. Kuitenkin 1/2-aallon antenni on huomattavasti pidempi kuin 1/8-aallon. Koska emme halunneet Sebaween pitkää piiska-antennia, johon ajaja voi sotkeentua, niin päätimme tehdä mahdollisimman lyhyen ja silti toimivan antennin. Antennin pituus 434Mhz taajuudelle laskettiin seuraavasti:

$$c = 299\,792\,458\text{m/s (valonnopeus)}$$

$$f = 434\text{Mhz (taajuus)}$$

$$A_p = \text{Antennin pituus}$$

$$A_p = (c / f) / 4$$

$$A_p = (299\,792\,458\text{m/s} / 434\text{Mhz}) / 4$$

$$A_p = 0,172\dots$$

Antennin pituudeksi tuli n. 17,2cm ja tämän todettiin olevan sopivan mittainen $\frac{1}{4}$ -aallon antenni. Tämä antenni oli vielä sopivan lyhyt piilotettavaksi laitteen takaosaan. Myös $\frac{1}{2}$ -aallon antennaa kokeiltiin, mutta yllättäen sen kantomatka sisätiloissa oli huonompi kuin $\frac{1}{4}$ -aallon. Tähän ei tarkalleen löydetty syytä, mutta syyksi epäiltiin liiallisia heijastuksia laboratoriotiloissa. Asiaa ei myöskään tutkittu sen enempää, koska tämän opinnäytetyön tarkoitus ei ollut paneutua radiotekniikkaan.

5.7 Langaton näyttö

Opinnäytetyön suunnitelmassa oli mainittu kohta, missä tavoitteena oli kehittää ja toteuttaa langattoman datan vastaanotto, käsittely ja lähetys tietokoneelle. Rakensimme akulla toimivan LCD-näytöllisen radiovastaanottimen, joka vastaanottaa RS-232-standardin mukaista dataa, käsittelee sen ja lähettää sen edelleen arduinon USB-sarjaporttiin. Tekemämme datan monitorointiohjelma lukee sarjaliikennettä tietokoneen USB-portista, johon vastaanotin on kytketty. Radiolinkki on 434MHz ja datan nopeus linkissä on 4800baud.

Mekaanisesti päätimme tehdä laitteesta yksinkertaisen, yksi 20x4 LCD-näyttö, käyttöjännite 9 voltin vaihdettavalta paristolta, radiovastaanotin $\frac{1}{4}$ aallonpituuden antennilla, virtakytkin, valintakytkin valikolle ja yksi arduino Uno ohjaamaan kaikkea. Kokosimme kaikki komponentit alumiiniseen laatikkoon, jonka tilasimme Partcosta. Langattomasta näytöstä on mahdollista seurata reaaliajassa laitteen käyttäytymistä, ilman tietokonettakin. Langattoman näytön voi asettaa näyttämään kolmea eri sivua. Laitteen pääsivu kertoo moottorien käyttöprosentit, kummankin puolen korkeimman lämpötilan, akkujännitteen ja mosfet:ien tilan. Vasemmassa alakulmassa näkyy jokaisella sivulla erikseen myös datalinkin toimintaa kuvaava animaatio. Lämpötilastamainittakoon sen verran, että laitteen neljästä lämpötilasensorista kaksi on vasemmalla ja kaksi oikealla puolella. Tarkemmin määriteltynä yksi pari kummankin mosfet:in jäähdytyssiilien keskellä ja toinen pari kummankin moottorin kylkeen liimattuina. Päänäytöllä näytetään siis kummankin puolen korkein lämpötila, joka määritetään yksinkertaisella ehtolauseella aina erikseen kummaltakin puolelta. Toiselle sivulle on koottu kaikki lämpötilasensorit ja niiden arvot. Kolmannella sivulla on gyroilta ja kiihtyvyyssantureilta lasketut kallistuskulmat. Kuvissa 24, 25 ja 26 on esitetty langattoman näytön eri sivut ja kuvassa 27 on laite kokonaisuudessaan.



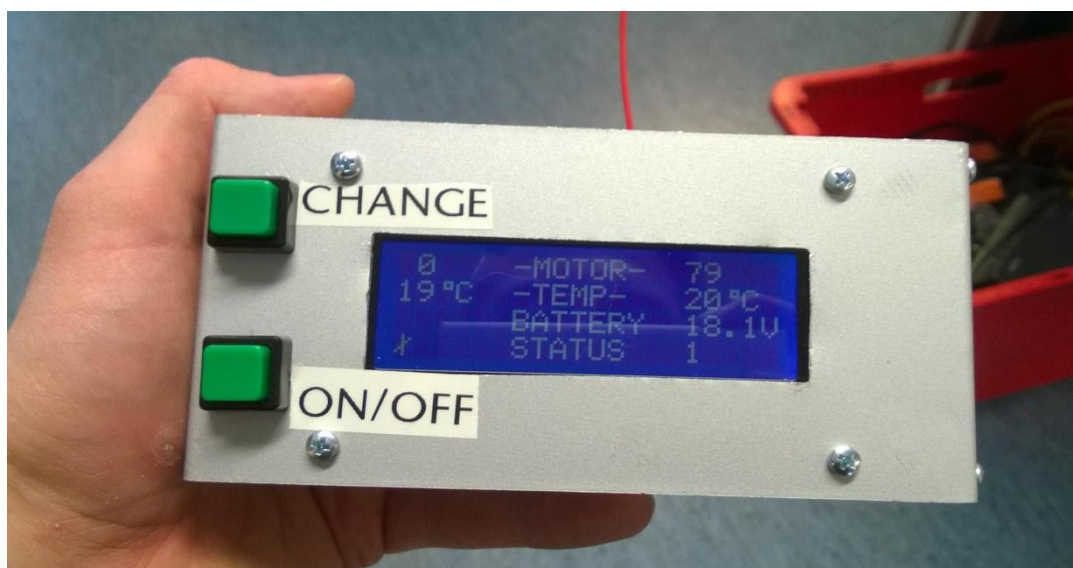
KUVA 24. Langattoman näytön etusivu.



KUVA 25. Näytön toinen sivu, laite on sammutettuna.



KUVA 26. Kolmas sivu, kallistuksen eri akseleiden suhteen.



KUVA 27. Langaton näyttöyksikkö koteloituna.

6 OHJELMA

6.1 I2C-väylä

I2C-Väylän on alunperin kehittänyt Philips vuonna 1982. Se on tarkoitettu matalan tason väyläksi erilaisille laitteille ja komponenteille tiedonsiirtoon. Philipsin keksintöä on alkanut käyttää myös lukematon määrä muita valmistajia, ja nykyään se on yksi yleisimmistä mikroprosessorien väylistä. Sen nopeus vaihtelee välillä 10-400kb/s yleisimmin. Nykyään on myös monta megabittiä sekunnissa siirtäviä versioita, mutta ne eivät ole kovin yleisiä alhaisten laskentatehon laitteissa. I2C-väylä on hyvin yksinkertainen kaksijohtiminen väylä. Johtimet ovat SDA (Data) ja SCL (Clock). Väylässä jokaisella laitteella on oma osoitteensa ja väylä on terminoitu ylös- ja alaspäin vastuksilla. Normaalisti väylä sisältää yhden master-laitteen ja monta slave-laitetta. Käyttämämme kytkentä on juuri tällainen, jossa mikroprosessori on master- ja anturit slave-laitteita.

Jokaisella anturilla on oma osoitteensa, josta ne vastaavat kun mikroprosessori niitä kutsuu. Muutoin anturit eivät lähetä väylään tietoa omatoimisesti. Antureissa on itsessään omia muistiosoitteita ja rekistereitä. Näihin rekistereihin mikroprosessori voi kirjoittaa asetusarvoja, tai lukea muistista antureiden mitta-arvoja. Kaupallisista tuotteista vastaavia I2C-väylän käyttöä löytyy mm. Nintendo Wii:n ohjaimet, näyttöjen ohjaustiedot, OLED-näytöt ja pienet näytöt yleisesti, tietokoneiden DDR muistit yms. I2C-väylän patentit ovat nykyään rauenneet, joten sen käyttö on kaikin puolin täysin vapaata. Väylän valinta käytettäväksi oli melko selvää, koska siihen oli saatavilla sopiva antureita halvalla. (I2C-väylä 2014.)

6.2 RS232-Väylä

RS-232 on yksi monista sarjaliikenteen standardeista. Se on ensi kerran esitelty vuonna 1962 ja on myös ensimmäisenä yleistynyt. Se on samalla myös yksi hitaimmista sarjaliikenteen standardeista. Juurikin hitauden ansiosta se sopii langattomaan viestiliikenteeseen jotenkin, sillä se sietää hieman virheitä joita syntyy radiolinkissä väistämättä. RS-232 ei sovellu kuitenkaan suurten datamäärien lähettämiseen

langattomasti, mitä enemmän dataa ja lähetysnopeutta lisätään, sitä epäluotettavammaksi datan vastaanotto menee. RS-232 on kuitenkin riittävän nopea ja sopii hyvin arduinojen väliseen langalliseen tiedonsiirtoon. Verraten I2C:llä voi väylän kellopulssin tiheys olla 100kHz, siinä missä me käytämme RS-232-väylässä 4800baud eli noin 4800Hz.

I2C-väylä ei myöskään sovellu järkevästi arduinojen väliseen liikenteeseen, koska se edellyttää master ja slave jaon jokaiselta väylän laitteelta. Muutenkin meille riittää, että data kulkee vain yhteen suuntaan, laiteosoitteita ei tarvita. Datan lähetys ja vastaanotto arduinojen välillä tapahtuu minimaalisella “kolmen kaapelin” RS-232-liikennöinnillä. Sitä käytetään silloin, kun täyden standardin ominaisuuksia ei tarvita. Käytännössä siis arduinolla on RX- ja TX-nastat, eli lähetys ja vastaanotto, kolmas kaapeli on ground. RS-232 mukaisen standardin datan siirto tapahtuu asynkronisesti tai synkronisesti bitti kerrallaan. Lähetystavu alkaa starttibitillä, jota seuraa kahdeksan databittiä ja se päättyy lopetusbittiin. Langattoman näytön RS-232-liikennettä pystytään lukemaan tietokoneen USB-portin kautta. Langattoman näytön arduinon USB-portti kytketään tietokoneen USB-porttiin, josta se tunnistautuu virtuaaliseksi COM-portiksi, eli sarjaliikenneportiksi. Erillistä adapteria ei tarvita.

Aikanaan RS-232 oli yleisin liikennöintitapa tietokoneissa käytettävien oheislaitteiden, kuten printterin, modeemin ja skannereiden liittämiseksi. Nopeuksien ja vaatimusten kasvaessa se jäi kuitenkin nopeasti USB:n jalkoihin, eikä sitä nykyään löydä enää normaalin tietokoneen liitännöistä. Jotkin teollisuuden oheislaitteet voivat kuitenkin käyttää vielä tätä standardia. (RS-232 väylä 2014.)

6.3 Arduino 0

Arduino 0:n pääsääntöisenä tehtävänä oli lukea I2C-anturit, käsitellä tieto ja lähettää moottoreille oikeat käskyt tasapainotilan saavuttamiseksi. Tämän arduinon ohjelman kiertonopeus haluttiin säilyttää suurena, eli +200Hz. Suurella nopeudella haluttiin varmistaa, ettei arvokkaita antureiden viestejä menetetä. Suuren nopeuden etuna on myös mahdollisuus laskea juoksevaa keskiarvoa anturisignaaleista. Riittävän nopeilla antureilla suoritettu juokseva keskiarvolaskenta on erinomainen tapa poistaa arvoista satunnaiset virheet. Ohjelma pyörii rajoittamattomalla nopeudella, joka on käytännössä

n. 200-250Hz:ä. Tämä tarkoittaa siis n.4-5ms päivitysväliä. Juokseva keskiarvo ja kulmalaskenta suoritettiin seuraavanlaisella ohjelmalla:

```
voidFilterValues()
{
  AxR = (AxR * 0.930) + (Ax * 0.070); // Smoothed accelerometer
  AyR = (AyR * 0.930) + (Ay * 0.070);
  AzR = (AzR * 0.930) + (Az * 0.070);

  GxR = (GxR * 0.97) + (Gx * 0.03);
  GyR = (GyR * 0.97) + (Gy * 0.03);
  GzR = (GzR * 0.97) + (Gz * 0.03);

  AngleX = atan(AxR/(sqrt((AyR*AyR)+(AzR*AzR))));
  AngleY = atan(AyR/(sqrt((AxR*AxR)+(AzR*AzR))));
  // AngleZ = atan((sqrt((AxR*AxR)+(AyR*AyR)))/AzR);
}
```

Jokaisella ohjelman kierroksella kiihtyvyysanturin uusista signaaleista otetaan talteen 2% ja säilytetään 98% vanhaa arvoa. Tällöin yksi virheellinen signaali voi aiheuttaa vain 2% poikkeaman lukuarvoihin. Gyrojen signaalit ovat parempilaatuisia, koska ne eivät kärsi tärinästä. Näin niiden vanhasta arvosta säilytetään 60% ja uutta lisätään 40%. Lopulta kiihtyvyysanturin keskiarvolasketuista akselikihtyvyyksistä lasketaan vektorilaskennan avulla kallistuskulmat. Käytännössä kaksipyöräisellä laitteella on tarpeen laskea vain yhden suunnan kallistuskulma.

Sen jälkeen kun arduino 0 on saanut suodatettua anturiarvot ja laskettua kallistuskulmat se summaa arvoon myös sauvaohjaimen arvon. Näin käyttäjä voi hallita eteen- ja taaksepäin kulkemista painopisteen siirrolla ja kierron (kääntymisen) sauvaohjaimella. Parhaaseen ratkaisuun päädyttiin säädöllä, jossa moottoreiden ohjaussignaali on suoraan kerrottu kallistuskulmasta ja gyrojen arvoista. Kallistuskulma käskii siis vaadittavaa moottorijännitettä, jotta laite saataisiin tasapainoon. Gyrot taas pyrkivät ainoastaan hidastamaan laitteen kiertonopeutta. Molemmat säädöt ovat siis proportionaliivypisiä. Gyrojen hidastava vaikutus kuitenkin mahdollistaa kulmalaskennan pysymisen mukana. Funktio, joka hoitelee vakautusta, on esitetty alla.

```

void Balance()
{
  AngValue = AngleX * 50;           // Angle signal multiplier
  GyrValue = GxR * 0.035;          // Rotation speed multiplier

  M1Temp = AngValue + GyrValue;     // Motor 1 power sum
  M2Temp = AngValue + GyrValue;     // Motor 2 power sum

  M1Temp = M1Temp + (-0.10 * JoyY); // Add joystick turn command
  M2Temp = M2Temp + (0.10 * JoyY);

  if(M1Temp > 150) { M1Temp = 150; } // Max power limits
  if(M1Temp < -150) { M1Temp = -150; }
  if(M2Temp > 150) { M2Temp = 150; }
  if(M2Temp < -150) { M2Temp = -150; }

  if (MainGain > 0.99 && (AngleX > 10 || AngleX < -10)) { BalanceMode = 0; } // Too
  much angle, stop. Maingain used in startup. Normaly 1.0

  if(M1Temp < 5 && M1Temp > 0) { M1Temp = M1Temp + 10; } // start boost
  if(M1Temp > -5 && M1Temp < 0) { M1Temp = M1Temp - 10; }

  if(M2Temp < 5 && M2Temp > 0) { M2Temp = M2Temp + 10; }
  if(M2Temp > -5 && M2Temp < 0) { M2Temp = M2Temp - 10; }

  ToM1 = MainGain * M1Temp; // To Motor 1 value
  ToM2 = MainGain * M2Temp; // To Motor 2 value

  // Serial.print(AngleX); Serial.print(" "); Serial.print(MainGain); Serial.print(" ");
  Serial.println(ToM1);
}

```

"AngValue" on kiihtyvyyssantureilta laskettava kulma suhteessa painovoiman suuntaan, jonka poikkeama nollassa kerrotaan sopivalla kertoimella. Kerroin määritettiin kokeilemalla ja se on myös riippuvainen ajajan painosta. Kerrointa nostettiin, kunnes laite alkoi oskilloida. Kertoimen nosto siis tarkoittaa laitteen tasapainoitusta, mutta liian

iso kerroin johtaa "overshoot"-tilaan, jossa korjaukset ovat liian isoja. Tämä ongelma ratkaistaan "GyrValue"-arvolla, joka on sivuttaissuuntaisen akselin gyron arvo kerrottuna tietyllä kertoimella. Mitä suurempi kerroin on, sitä enemmän gyro jarruttaa kiertonopeutta. Gyron liikenopeuden jarruttava vaikutus estää näin kiihtyvyyssantureiden ylikorjaamisen. Nämä kertoimet ovat riippuvaisia ihmisen painosta ja myös hieman ajajan tasapainosta laitteen päällä. Tämän takia arvot täytyi säätää kohdalleen käytännön testien kautta. Alapuolella taulokossa on esitetty arduino 0:n tärkeimmät signaalimuuttujat.

TAULUKKO 1. Arduino 0:n signaalimuuttujat.

Muuttuja	Arvoväli	Selite
GxR	n. -5000 ... 5000	Filtteröity gyrosignaali
AngleX	-1.00 ... 1.00	Laskettu kallistuskulma kiihtyvyyssantureilta.
AngValue	-50 ... 50	Kerrottu kallistuskulma
GyrValue	n. -175 ... 175	Kerrottu gyrosignaali.
JoyX / JoyY	-512 ... 512	Joystickin arvot.
M1-,M2Temp	-150 ... 150	Väliaikaismuuttujat.
MainGain	0 ... 1.00	Pehmokäynnistys alussa. Nousee 0 → 1 viiveellä.
ToM1,ToM2	-150 ... 150	Moottoreille lähtevät arvot.

Kun arduino 0 on saanut laskettua vaadittavat ohjausarvot moottoreille lähettää ne oman viestiliikenteemme mukaisesti arduino 1:lle, joka ohjaa moottoreita. Laitteen sisäiset viestit lähetetään 38400 baudin nopeudella yksi viesti 15 millisekunnin välein. Eri viestejä on kahdeksan kappaletta, joten näin ollen jokainen viesti toistuu 0,12 sekunnin välein. Tämä on riittävän pieni viive kaikkiin tilanteisiin nähden. Päätimme käyttää viestejä, jotka alkavat aina M-kirjaimella, jota seuraa osoite. Osoitteen jälkeen piste erottaa arvon ja viimeinen piste päättää viestin. Taulukossa alla on esitetty arduino 0:n kaikki mahdolliset viestit arduino 1:lle.

TAULUKKO 2. Arduino 0:n viestit.

Viestiesimerkki	Arvoväli	Osoite	Selite
M0.1.	0 ... 1	0	MOSFET ON/OFF
M1.255.	-255 ... 255	1	Moottori 1
M2.-125.	-255 ... 255	2	Moottori 2
M3.20.	-90 ... 90	3	Kallistuskulma X
M4.14.	-90 ... 90	4	Kallistuskulma Y
M5.124.	-512 ... 512	5	Joystick X
M6.-10.	-512 ... 512	6	Joystick Y
M7.1.	0 ... 1	7	Äänimerkki

6.4 Arduino 1

Arduino 1:n päätehtävänä on moottoreiden turvallinen ohjaus. Turvallisella ohjauksella tarkoitetaan tässä tapauksessa sitä, että moottoreita ei käskytetä pyörimään, elleivät arduino 0:n käskyt ole selviä ja oikealla arvoalueella. Arduino 1 hallitsee myös relettä, jonka ollessa auki ei moottoreiden käyttö ole mahdollista. Oikeassa käynnistyssekvenssissä arduino 1:n pitää siis ensin vastaanottaa käsky "M0.1.", jonka jälkeen se kytkee releen avulla moottorien käytön mahdolliseksi ja vasta sen jälkeen on mahdollista käskä moottoreiden pyöritys päälle. H-siltojen ohjaus haluttiin myös siksi erillisen mikroprosessorin ohjaamaksi, että H-siltojen toimintahäiriö voisi johtaa suurin oikosulkuvirtoihin. Toimihäiriön todennäköisyys on pienempi, kun H-siltoja ohjaavalla mikroprosessorilla ei ole ylimääräisiä tehtäviä. Turvareleen toinen maan puoleinen johdin on kierrätetty hätävetonarun ja moottorinohjainten käyttökytkimen kautta. Tämä tarkoittaa hätä-seis toimintoa, joka on riippumaton ohjainelektronikasta. Moottorien hätäpysäytys toimii siis kaikissa tilanteissa ja tapahtuu myös automaattisesti jännitteiden laskiessa.

Moottorinohjainten tärkein tehtävä on siis 4:n PWM-signaalin tuotto H-silloille. Neljä signaalia tulee siitä, että moottoreita on kaksi, joista molemmat ovat käskytettävissä eteen ja taakse. Jokaista toimintoa kohden on siis kaksi mosfettia, joita sama PWM-signaalilähtö ohjaa. Kun pyörimissuunta vaihtuu, niin vaihtuu mosfet-pari. Parien kytketyminen yhtäaikaisesti päälle täytyy estää ohjelmallisesti, koska yhtäaikainen

kytkeytyminen tarkoittaisi oikosulkua. Ohjelmakoodissa täytyi huomioida myös se, että transistorit ennen mosfettejä kääntävät signaalin päinvastaiseksi. Mikroprosessorin lähdön ollessa ylätilassa transistorit ohjaavat mosfetit kiinni. Eli ohjelmakoodillisesti tilanne, jossa moottorit ovat paikoillaan on se, jossa kaikki lähdöt ovat ylhäällä. Seuraava ohjelmakoodi ohjaa toista moottorinohjainta.

```
void Motor1(int Value)
{
  if(Value > SlowValueM1) { SlowValueM1 = SlowValueM1 + 1; delay(3); }
  if(Value < SlowValueM1) { SlowValueM1 = SlowValueM1 - 1; delay(1); }

  if(SlowValueM1 > 0 && SlowValueM1 < 256)
  {
    digitalWrite(6, HIGH);
    analogWrite(5, (255-SlowValueM1));
  }

  if(SlowValueM1 < 0 && SlowValueM1 > -256)
  {
    digitalWrite(5, HIGH);
    analogWrite(6, (255-(SlowValueM1*-1)));
  }

  if(SlowValueM1 == 0 || SlowValueM1 > 255 || SlowValueM1 < -255)
  {
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
  }
}
```

Ohjelma tutkii kumpaan suuntaan moottoria on tarkoitus pyörittää vertaamalla onko arvo yli vai ali nollan. Sen jälkeen vastakkainen mosfet-pari asetetaan EI-johtavaksi (lähtö ylätilaan). Tämän jälkeen halutun pyörimissuunnan lähtöön voidaan asettaa PWM-arvo. Alimmalla ehtolauseella asetetaan kaikki lähdöt ylätilaan (moottori seis) jos käskytetty arvo on nolla tai ulkona oikealta vaihteluväliltä.

Myös lämpötila-anturit kiinnitettiin arduino 1:een. Sen tehtävä ei kuitenkaan ole analysoida niiden arvoja, vaan ainoastaan hoitaa ADC-muunnos ja lähettää arvot eteenpäin. Arduino 1 lähettää samat tiedot eteenpäin jotka se vastaanottaa arduino 0:lta. Lisäksi se lähettää myös neljä lämpötila-arvoa ja yhden jännitelukeman. Lämpötila-arvot ovat yksi kummaltakin moottorilta ja yksi kummaltakin H-sillalta. Jännitearvo on akkujännite, joka mitataan arduinon oman ADC-muuntimen avulla. Arduino 1:n lisäykset väyläviesteihin ovat alla olevassa taulukossa.

TAULUKKO 3. Arduino 1:n lisäviestit.

Viestiesimerkki	Arvoväli	Osoite	Selite
M8.182.	100-250	8	Akkujännite
M9.210.	50-1000	9	Lämpötila 1
M10.250.	50-1000	10	Lämpötila 2
M11.240.	50-1000	11	Lämpötila 3
M12.250.	50-1000	12	Lämpötila 4

Lämpötila ja jännitearvot lähetetään kymmenkertaisina lukuina, jotta myös ensimmäinen desimaali saadaan tarkkuuteen mukaan. Esimerkiksi viesti "M8.182." tarkoittaa siis akkujännitettä 18,2V. Arvon lähettäminen suoraan desimaalimuodossa vaatisi vastaanottavan ohjekoodin ottavan huomioon myös desimaalimerkin. Tämä toisi ohjelmaan turhaa lisää pituutta, kun huomattavasti helpompi tapa on poistaa desimaalit kertomalla luku kymmenkertaiseksi ennen lähetystä.

6.5 Arduino 2

Arduino 2:n pääsääntöinen tehtävä on valvonta ja tiedonsiirto. Se toimii laitteen sisäisen väylän päätepisteenä, joka kerää kaiken väyläpitkin tulevan tiedon ja lähettää langattomasti eteenpäin kohti arduino 3:ea langattomasti. Arduino 3:n tehtäviin kuuluu myös jäähdytyspuhallinten ohjaus. Puhaltimia ohjataan yksinkertaisesti 5V:n releellä, joka kytkee puhaltimille jännitteeksi 12V 7812-regulaattorilta. Puhaltimet käynnistyvät jos yksikin lämpötila-anturi ylittää +35C. Kun lämpötila jälleen laskee alle +30C:n puhaltimet sammuvat.

Langaton tieto lähetetään hitaammin kuin sisäisen väylän tiedot. Tämä johtuu siitä, että käytetty langaton lähetin ja vastaanotin eivät ole yhteensopivia yli 9600baud:n tiedonsiirtonopeuksilla. Käytännössä parhaaksi nopeudeksi havaittiin 4800baud, jolloin tiedonsiirto oli vakaata ja mahdollisimman nopeaa. Suuremmilla nopeuksilla kantomatka ja häiriöt lisääntyivät liiaksi. Langattoman tiedonsiirron viestit eroavat myös hieman sisäisen väylän viesteistä, ne ovat muotoa "XXXM1.125". Kolme "X" merkkiä lisättiin viestin alkuun, koska viestin alun data oli usein virheellistä. Tämä johtuu siitä, että koska lähete ei ole jatkuvaa, niin muutaman ensimmäisen merkin bitit eivät usein mene perille oikein. Ongelma ratkeaa, kun viestin ensimmäiset merkit ovat ns. merkityksettömiä eli pois filteröitäviä merkkejä. Käytettäessä alkumerkkejä läheteessä ei havaittu mitään ongelmia varsinaisen viestin merkeissä.

6.6 Arduino 3

Langattoman näytön arduinon tehtävistä ensimmäinen oli vastaanottaa 433MHz radiolinkillä langaton viesti, prosessoida sen sisältämä data ja tallentaa se muuttujiin. Vastaanotettu sarjamuotoinen data sisälsi yhteensä 13 eri osoitetta ja niiden arvoa. Totesimme jo ensimmäisissä testeissä että vastaanotetun sarjaliikenteen ensimmäiset merkit katosivat radiolinkin virheisiin. Lisäsimme aina lähetetyn osoitteen eteen kolme X-kirjainta. Tällä kompensoimme aiheutuneita virheitä ja saimme osoitteen ja arvon linkin läpi luotettavasti. Vastaanotetusta datasta parsittiin hyödyttömät etumerkit pois sekä jäsennettiin osoite ja arvo muuttujiin myöhempää käyttöä varten. Arvojen jäsentämiseen käytettiin "switch case"-rakennetta, koska se on tavallista "if-else"-rakennetta nopeampi. Jokaisella funktion kierroksella ei käydä kaikkia arvoja läpi, vain se mikä tapaus kulloinkin on kyseessä. Jäsennysfunktio `parseInt()` etsii järjestyksessä seuraavan lukuarvon ja asettaa sen määritettyyn muuttujaan. `parseInt()` lopettaa lukemisen kun vastaan tulee jokin muu kuin numero, eli tässä tapauksessa piste. Oletetaan esimerkiksi, että vastaanotettava arvo on XXXM1.100. Ensiksi jäsennetään osoite 1, sitten jäsennetään arvo 100 ja lopuksi poistetaan mahdolliset virheet viimeisen pisteen jälkeen ja jatketaan seuraavaan tapaukseen.

```
voidParseData()
{
if(ArduinoDATA.available() > 5)      // Jos dataa vastaanotetaan, jatketaan funktiota
```

```

    {
    if(ArduinoDATA.find("M") > 0)      // Jatketaan jos löytyy osoitteen merkki M
    {
    intaddress = ArduinoDATA.parseInt(); // Asetetaan osoitte
    intvalue = ArduinoDATA.parseInt(); // Asetetaan osoitteen arvo
    intwaste = ArduinoDATA.read(); // Poistetaan virheet viimeisen pisteen jälkeen

    switch(address)      // funktion tarkastaa mitä osoitetta käsitellään
    {
    case 0: FETState = value; break;
    case 1: M1Value = value;
    LCD.setCursor(0,4); LCD.print("/"); break;
    case 2: M2Value = value; break;
    case 3: AngleX = value; break;
    case 4: AngleY = value;
    LCD.setCursor(0,4); LCD.print("/"); break;
    case 5: JoyX = value; break;
    case 6: JoyY = value; break;
    case 7: Horn = value; break;
    case 8: BatteryVoltage = float (value) / 10;
    LCD.setCursor(0,4); LCD.print("-"); break;
    case 9: Temperature1 = value / 10; break;
    case 10: Temperature2 = value / 10; break;
    case 11: Temperature3 = value / 10; LCD.setCursor(0,4);
    LCD.print(char(92)); break;
    case 12: Temperature4 = value / 10; break;
    // Tapauksien välissä olevat LCD-setcursor ja LCD-print komennoilla tuotetaan pieni
    animaatio vastaanotetun datan merkiksi.
    }
    }
    }
    }

```

Toinen tehtävä tällä arduinolla oli tarkastella ja karsia vastaanotetusta datasta virheet. Kun sarjaliikennettä tuodaan radiolinkin läpi tulee väistämättä virheitä ja ne näkyvät datan monitorointi-ohjelman histogrammeissa. Joten kun vastaanottotaajuus on kuitenkin riittävän suuri, 4800 baud, voidaan vastaanotetuista arvoista karsia selkeästi liian suuret muutokset pois. Suurimmat häiriöt havaitsimme lämpötila-arvoissa, joten pyöristimme niiden luettuja arvoja siten, että lämpötila saa muuttua vain yhden asteen suuntaansa yhden sekunnin aikana. Alla olevassa koodissa on päänäytön lämpötila-arvojen datankäsittely. Kuten jo aikaisemmin on mainittu, päänäytöllä näkyy vain kunkin moottorin puolen korkein lämpötila ja se on toteutettu seuraavasti.

```
voidDataHandle()
{
  HTempLeft = 0;           // Asetetaan muuttujat funktion alussa nolliin
  HTempRight = 0;

  if(Temperature1 >HTempLeft){ HTempLeft = Temperature1; }

  //Tarkastellaan anturia 1, jos se on suurempi kuin korkein lämpötila, asetetaan anturin
  //1 arvo vasempaan muuttujaan.

  if(Temperature2 >HTempLeft){ HTempLeft = Temperature2; }

  //Tarkastellaan anturia 2, jos se on suurempi kuin korkein lämpötila, asetetaan anturin
  //2 arvo vasempaan muuttujaan.

  if(Temperature3 >HTempRight){ HTempRight = Temperature3; }

  // Sama järjestely tehdään oikean puolen antureille
  if(Temperature4 >HTempRight){ HTempRight = Temperature4; }

  M1ValueMapped = map(M1Value, -255, 255, -100, 100);

  M2ValueMapped = map(M2Value, -255, 255, -100, 100);
```

```
// Tässä skaalataan moottorien arvot väliltä -255-255 välille -100%-100%, koska
// päänäytöstä on mukavampi tulkita moottorin käyttöprosentteja, kuin ohjauksen
// PWM:ää.
}
```

Kolmas tehtävä tällä arduinolla oli kirjoittaa vastaanotetut ja käsitellyt arvot 20x4-riviselle LCD-näytölle. Tähän samaan tehtävään kuului myös datan välitys arduinon omaan USB-sarjaporttiin myöhempää käyttöä varten. Sarjaporttiin kirjoituksesta ei kannata antaa esimerkkiä, koska se on hyvin samankaltainen switchcase-rakenne, kuin aikaisemmassa jäsennysfunktiossa.

LCD-näyttöä varten arduinon kirjastoista löytyy useimmin käytetyt merkit ja näytölle tulostaminen käy helposti. Tarvitsee vain kertoa osoittimen paikka ja haluttu teksti tai muuttuja. Jotain merkkejä kuitenkin kirjastosta ei löydy ja ne tarvitsee itse tehdä, kuten esimerkiksi aste-merkki. Yksi LCD-näytön ruutu koostuu 5x7 pikselistä, ja niistä luotu asteen merkki näyttää koodirivinä seuraavalta

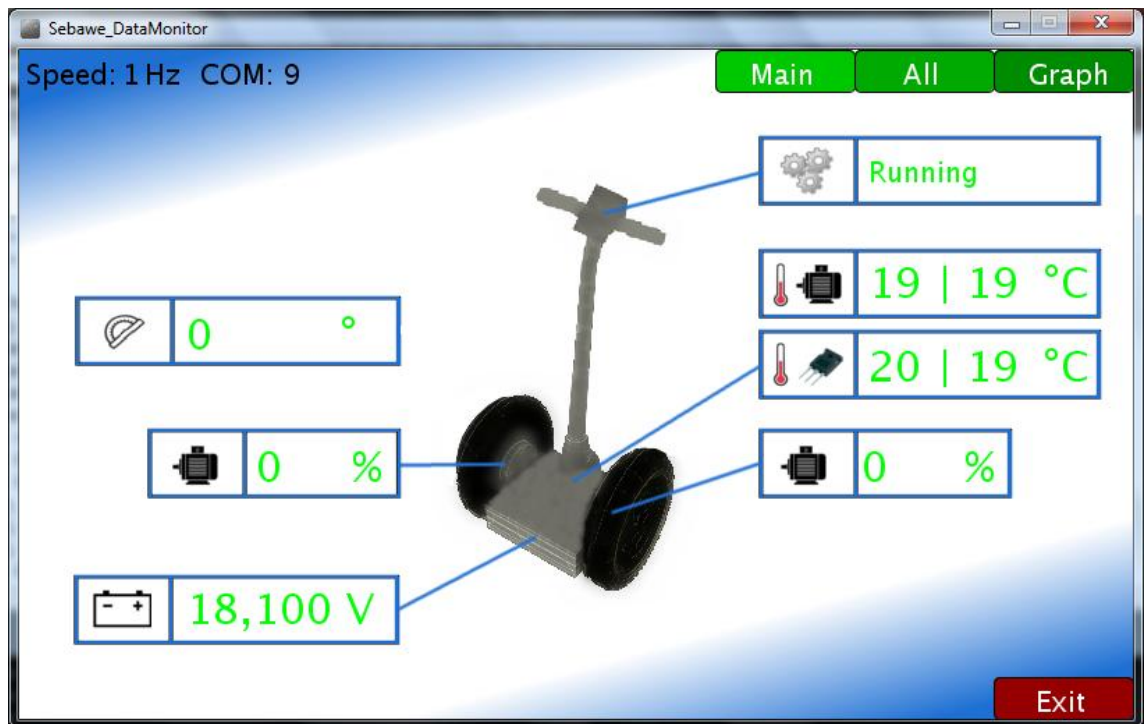
```
byte degree[8] = {
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000,
};
```

6.7 Datan monitorointiohjelma (PC)

Testausvaihetta ja miksei toki käyttöäkin varten päätettiin tehdä myös tietokoneohjelma, jolla voidaan tarkkailla laitteen arvoja. Tietokoneohjelma saa anturitiedon USB-väylän kautta. USB on mahdollista kiinnittää langattoman näytön USB-porttiin. Langaton näyttö jatkaa toimintaansa normaalisti, vaikka se samalla välittää tiedot myös PC-ohjelmalle. Ohjelma kirjoitettiin Processing kielellä Processing editorilla, joka toimii

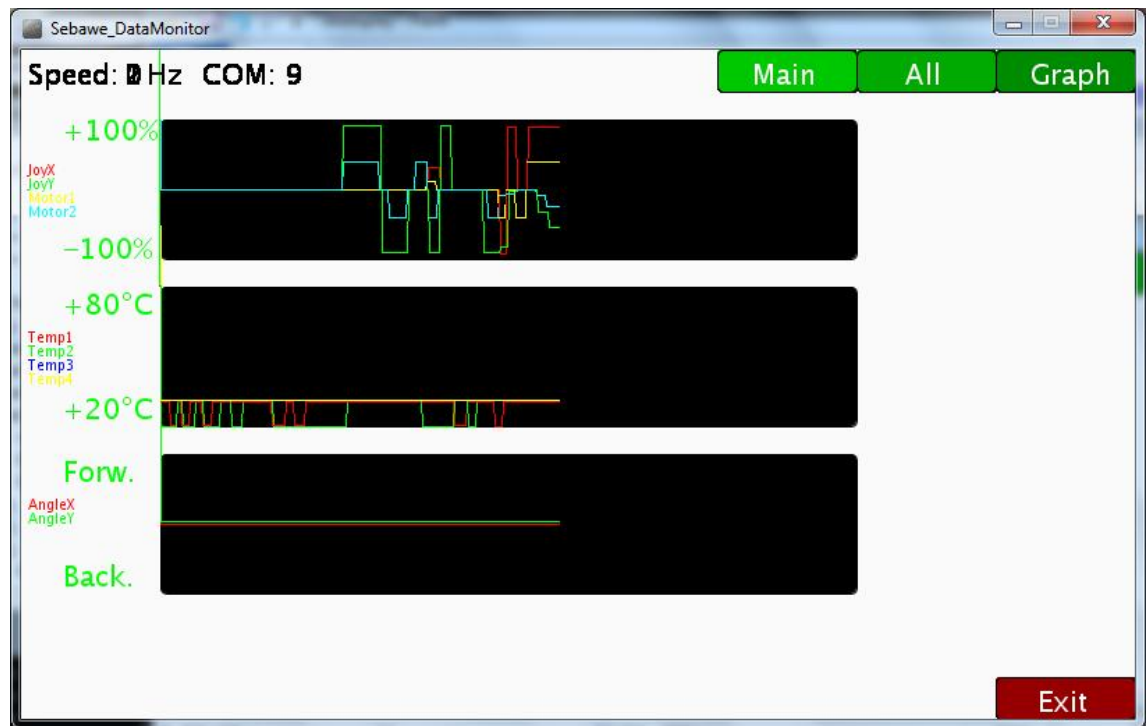
Javan päällä. Ohjelmointikieli olikin hyvin lähellä C++ / Java kieliä. Tarkoitus oli tehdä hyvin yksinkertainen ohjelma, jolla voidaan monitoroida suurinta osaa laitteen toiminnoista.

Kuvassa 28 on ohjelman päänäkymä, joka tulee näkyviin kun ohjelma on onnistunut hakemaan oikean COM-portin automaattisesti.



KUVA 28. PC-Ohjelman päänäkymä

Ohjelman visuaaliseen näkymään ei panostettu paljoa, koska ohjelma ei ollut tämän työn varsinainen aihe. Ohjelmalla kuitenkin helpotettiin laitteen säätöä huomattavasti, koska sen avulla saattoi havainnoida anturien arvoja reaaliaikaisesti ja langattomasti. Ohjelma tehtiin melko yksinkertaisella tavalla, jossa grafiikat luotiin suurimmaksi osaksi kuvankäsittelyohjelmalla nimeltä Gimp. Se on ilmainen open-source kuvankäsittelyohjelma. Näitä grafiikoita kutsuttiin ohjelmakoodista käskyillä ja piirrettiin oikeille paikoilleen. Tämän perusgrafiikan päälle pystyttiin sitten piirtämään kuvaajia ja tekstiä sekä numeroita sopivilla käskyillä. Ohjelmaan tehtiin kolme välilehteä, jotka ovat "Main", "All" ja "Graph". Main-lehdellä on yksinkertaistettu näkymä tärkeimmistä arvoista. All-lehdellä näkyy kaikki mahdolliset arvot ja Graph-lehti sisältää graafiset jatkuvasti päivittyvät käyrät suurimmasta osasta arvoja. Seuraavassa kuvassa näkyy hyvin, kuinka arvojen monitorointi graafisesta näkymästä on helppoa.



KUVA 29. Graafinen arvohistoria.

Ohjelma pyrittiin tekemään yksinkertaiseksi niin, että ainoa asetettava asia on COM-portti. Tämäkin päätettiin automatisoida niin, että ohjelma etsii automaattisesti oikean COM-portin tietokoneesta ja avaa sen. Oikean portin etsintä toteutettiin niin, että ohjelma pyytää aluksi windowsilta listan porteista, ja avaa sitten niistä ensimmäisen. Tämän jälkeen ohjelma tarkkailee liikennettä portista ja etsii siitä käyttämiämme viestejä. Jos viestejä ei löydy, niin ohjelma sulkee portin ja avaa taas seuraavan kuuntelua varten. Kun ohjelma löytää portista laitteen viestejä se poistuu etsintätilasta ja aloittaa normaalin toiminnan. Jos tietoliikenne katkeaa kesken toiminnan, niin ohjelma ei aloita etsintää uudelleen. Tämä päätettiin tehdä näin siksi, että langaton linkki saattaa aina hetkellisesti katketa ja tällöin porttien läpikäyminen uudelleen olisi turhaan. Seuraava ohjelmakoodi on funktio, jolla COM-portit scannataan läpi.

```
void ComScan()
{
    if(UpdateSpeed > 0) { ComInit = 1; }

    if(ComInit == 0)
    {
        int MaxPort = Serial.list().length - 1; //get number of ports available
        ComPort++;
    }
}
```

```
if(ComPort>MaxPort) { ComPort = 1; }

sPort = new Serial(this, Serial.list()[ComPort], 4800); // Open 1st port in list
delay(2000);
print("COM Scan:"); print(ComPort); print(" / "); println(MaxPort);

for(int x = 0; x < 20; x++)
{
  RefreshScreen(); // Draw screen
  ReadSerial(); // Read serial
  delay(10);
}

}
}
```

Ohjelman toiminnan kannalta on huomioitavaa, että se vaatii tietokoneelta Java-ohjelmistoa. Jos ohjelmistoa ei ole, niin ohjelma ei käynnisty ja windows antaa virhekoodin. Ohjelman toiminta on testattu vain Windows7 32- ja 64-bittisillä versioilla. Todennäköisesti se kuitenkin toimii ongelmitta kaikilla Windows XP tai uudemmilla käyttöjärjestelmillä.

7 LOPPUTULOS

Aloitimme projektin varsinaisen työvaiheen joulukuussa 2013, vaikka suunnittelutyö ajatustasolla oli alkanut jo lokakuussa Helsingin automaatiomessuilla. Joulukuun lopulla suunnittelimme rungon rakenteen ja teimme suurimmat tilaukset, kuten alumiiniprofiilit, kiinnikkeet, laakerit, moottorit ja koneistetut akselit. Kokemuksen perusteella tiesimme, että profiilin toimituksessa on ollut ongelmia, joten olimme ajoissa liikenteessä.

Kaikki muut tilatut osat saapuivat ajallaan paitsi alumiiniprofiili, jonka varaan koko rungon suunnitelmat rakentuivat. Profiilin toimituksessa oli merkittäviä ongelmia. Ensimmäisen paketin vastaanotimme maaliskuun puolivälissä, joka osoittautui vääräksi. Ja toisen paketin huhtikuun alussa, josta puuttui tilaamamme kiinnikkeet. Hieman alkuperäistä suunnitelmaa muuttamalla aloitimme rakentamisen viimeisen paketin saatuaamme. Näin suuresta myöhästymisestä seurasi se, että emme päässeet oikeisiin ongelmakohtiin käsiksi ennen huhtikuuta. Eli meille jäi noin kuukausi aikaa kohdata ja ratkoa kaikki tiellemme osuvat vastoinkäymiset ja se oli yksinkertaisesti liian vähän. Tämän myöhästyksen lisäksi suurinta murhetta meille on aiheuttanut lähestulkoon ainoa asia, jota emme ole tehneet itse, eli moottorit.

Ensimmäinen mekaaninen ongelma meinasi kaataa koko projektin onnistumisen. Saimme huomata, ettei moottorien planeettavaihteistossa ole vapaavaihdetta. Eli käytännössä tarkoittaa sitä, ettei vaihteiston akselia voinut pyörittää kuin moottorin päästä. Tässä vaiheessa meillä ei enää ollut aikaa tilata uusia moottoreita, saati sitten sovittaa runkoa ja suunnitelmia uudestaan, joten lähdimme purkamaan planeettavaihteistoja. Löysimme vaihteiston kärjestä pienet metallitapit, joiden tarkoitus oli toimia haittoina vapaan pyöryksen lukitsemiseksi. Poistimme tapit ja saimme näin muokattua moottorit käyttötarkoitukseen sopiviksi.

Toinen ongelma oli moottoreiden planeettavaihteiston mekaaninen vällys. Se ei ollut suuri vielä ennen haittojen poistoa, mutta moninkertaistui niiden poiston jälkeen. Vällys on erittäin ikävä asia ja kertautuu pienestä akselin liikkeestä suureksi pyörän liikkeeksi. Vaikka moottorit reagoivat erittäin nopeasti, on laitteen tasapainottelu silti epävarmaa. Vällys aiheuttaa yli 100 mm liikkeen renkaan päässä.



KUVA 30. Planeettavaihteiston vapaavaihteen haitat.

Tämän lisäksi moottorin akseli aiheutti murhetta. Alunperin suunnittelimme pitävämme sen paikallaan pyörän akselia vasten kahdella lukitusruuvilla, mutta se ei riittänyt ja ruuvit lähtivät aukeamaan kun akseliin kohdistui vääntöä. Paransimme suunnitelmaa ja koitimme porata läpireiän akselistä, mutta se osoittautui äärimmäisen kovaksi valetuksi teräkseksi, eivätkä terämme pystyneet lastuamaan sitä. Teimme kompromissin ja hioimme pienellä kivipäällä moottorin akselin molemmin puolin tasopinnat, joita vasten kiersimme lukitusruuvit niin lujaan kuin mahdollista. Lopuksi täytimme kaikki liitokset pikaliimalla. Muuten laitteen mekaaninen rakenne on erittäin luja ja kestävä. Mitään ongelmia ei ole ilmennyt laakereiden, pyörien eikä rungon kanssa.

Sähkösuunnitteluun käytimme reilusti aikaa. Pelkästäänkin siitä syystä, että jouduimme odottamaan runkopalkkien toimitusta liian pitkään. Kehitimme moottorinohjaimet muutaman prototyypin kautta huippuunsa, ja saimme niiden lämmöt pysymään niin tasaisina, että jäähdytyssiilien puhaltimet jäivät lähes turhiksi. Langaton datan lähetykset, datan monitorointiohjelma ja langaton seurantanäyttö toteutettiin lähes ensimmäisenä ja siitä oli suurta apua laitteen rakennuksen aikana. Pystyimme seuraamaan siitä rasiuskokeiden aikana mm. mosfet:ien lämpötiloja ja akkujännitteiden käyttäytymistä.

Sähköisessä toteutuksessa ilmeni kaksi ongelmaa, joista toinen oli pieni suunnitteluvirhe moottorinohjaimissa, joka korjattiin yhdessä päivässä ja toinen oli Bebekistä ostetut 20 kappaletta viallisia BC547 transistoreja. Jälkimmäinen ongelma olikin suhteessa vakavampi, koska sen löytämiseen kului kaksi kokonaista työpäivää. Kun nämä vastoinkäymiset otetaan huomioon projektin laajuudessa, voidaan todeta, sähkösuunnittelun ja toteutuksen olleen erittäin onnistunut.

Opimme myös paljon uutta 3D-tulostuksesta projektin myötä. Se oli käytännössä ainoa vaihtoehto tehdä siisti ja toimiva ohjainkahva. Koululle hankittu tulostin on laadukas ja se pystyy kokemuksen mukaan ainakin 0.5mm tarkkuuteen. ABS-muovista tulostetut kappaleet ovat erittäin kovia ja kestäviä. Näiden ominaisuuksien lisäksi tulostinohjelman automaattisella kennotuksella saatiin kappaleista kevyempiä ja merkittävästi edullisempia.

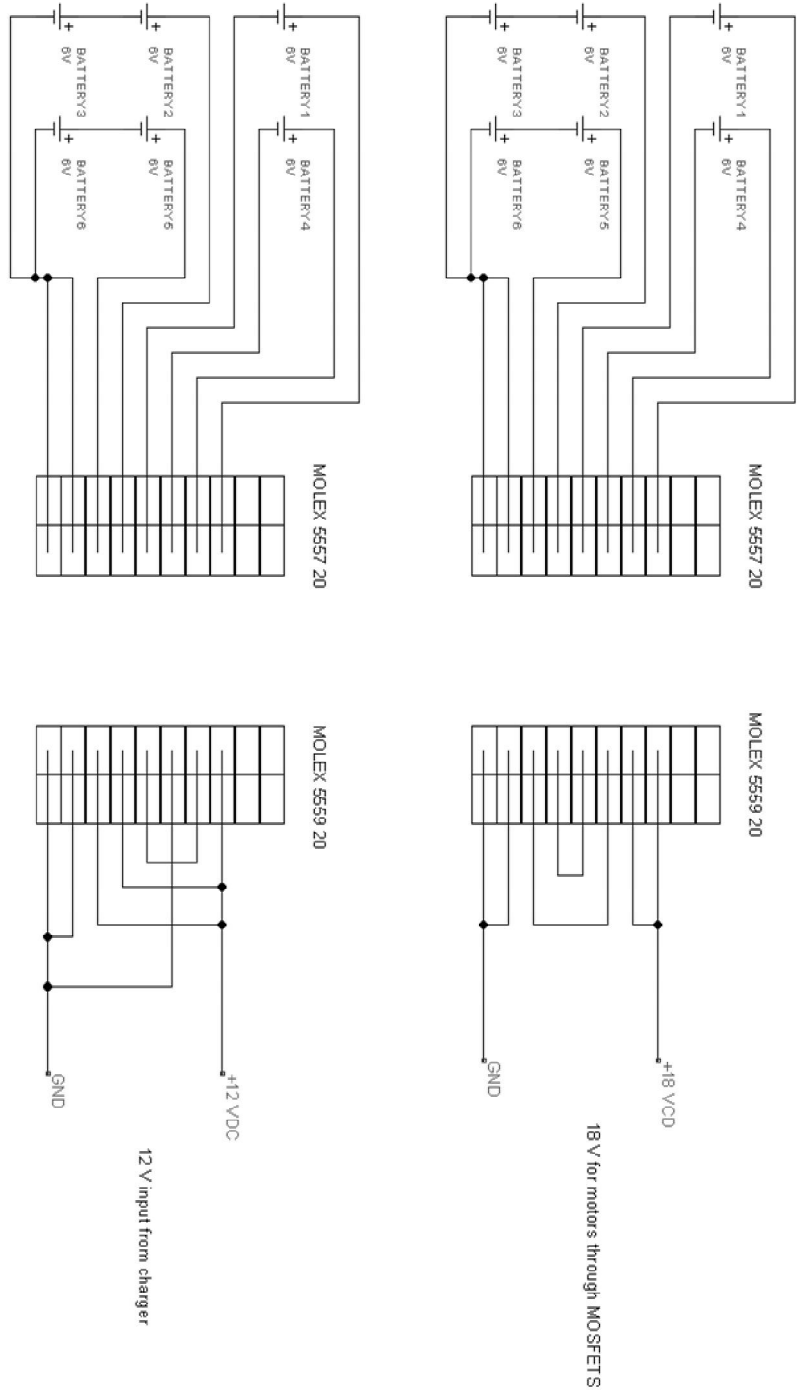
Laitetta voisi parantaa merkittävästi vaihtamalla nykyisten planeettavaihteellisten tasavirtamoottorien tilalle kulmavaihteelliset tasavirtamoottorit. Tarkoitukseen sopivia moottoreita käytetään paljon sähköllä toimivissa pyörätuoleissa. Niiden hyviin puoliin kuuluu myös integroidut laakeripesät ja kiinnityslaipat renkaita varten. Tällä tavoin päästäisiin eroon pahimmasta ongelmasta eli mekaanisesta välyksestä. Moottoreiden vaihdolla saataisiin myös tuotua painopistettä alemmaksi. Mitä alempana painopiste on renkaan akselin suhteen, sitä parempi tasapaino. Ideaalitulanteessa ihminen seisoo pyörintäakselin alapuolella ja etäisyyttä maahan kasvatetaan suuremmalla renkaalla.

LÄHTEET

1. Bosch akkuporakone, jonka moottoreita käytetty. Luettu 10.4.2014.
<http://www.bosch-professional.com/fi/fi/gsr-18-ve-2-li-34753-ocs-p/>
2. Akkujen esite valmistajalta. Luettu 3.3.2014.
http://www.partco.biz/verkkokauppa/datasheet/ctleader_esite.pdf
3. Mosfetti:en tekninen dokumentti. Luettu 3.3.2014.
<http://www.irf.com/product-info/datasheets/data/irfp064n.pdf>
4. Määräys luvasta vapaiden radiolähettimien yhteistajuuksista ja käytöstä. Luettu 4.3.2014.
<https://www.viestintavirasto.fi/attachments/Viestintavirasto15AF2013M.PDF>
5. I2C-Väylä yleisesti. Luettu 29.4.2014.
<http://www.i2c-bus.org/>
6. TietopakettiRS-232 väylästä. Luettu 10.4.2014.
<http://www.arcelect.com/rs232.htm>

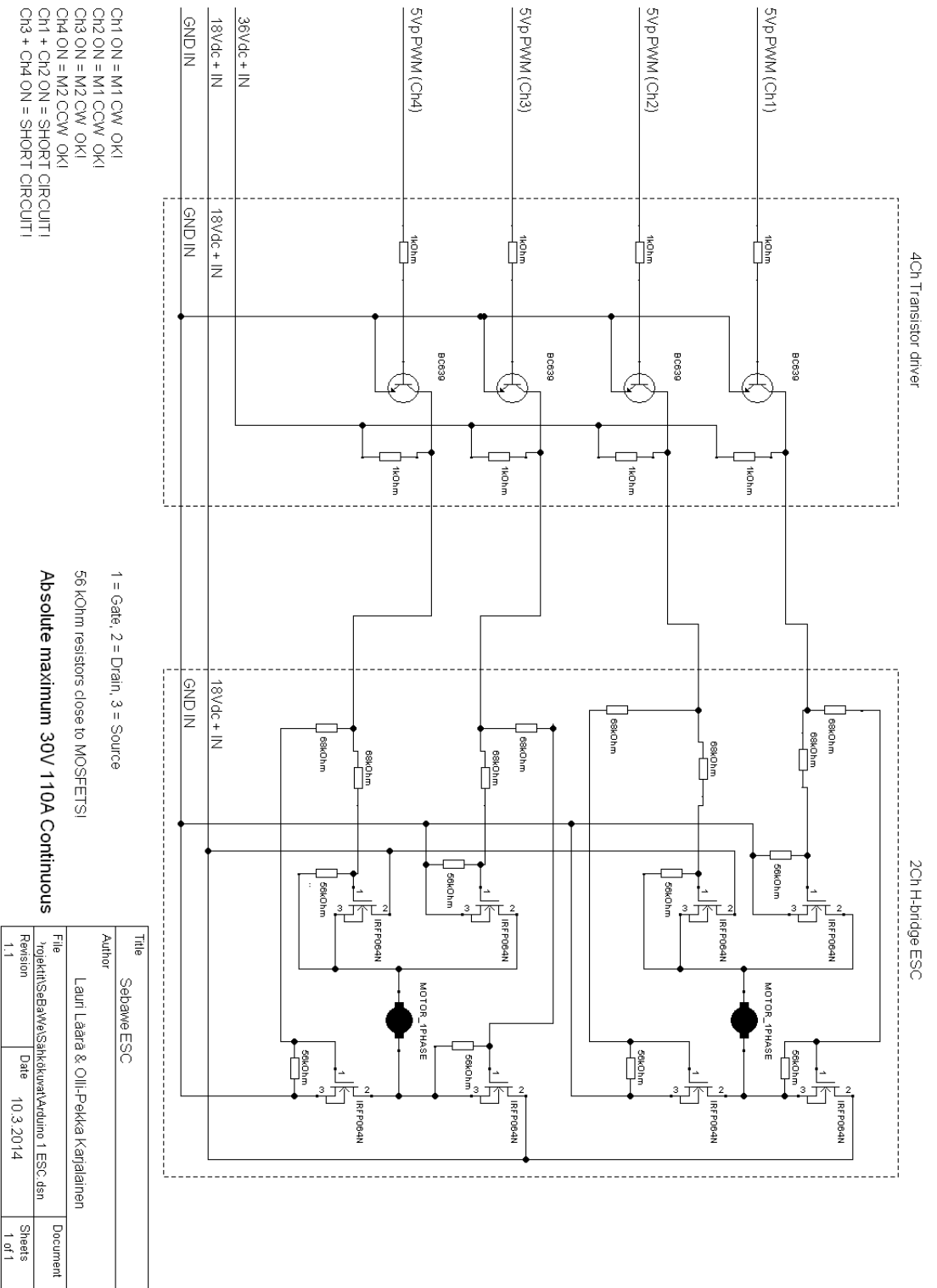
LIITTEET

Liite 1. Akuston liitin sähkökuva

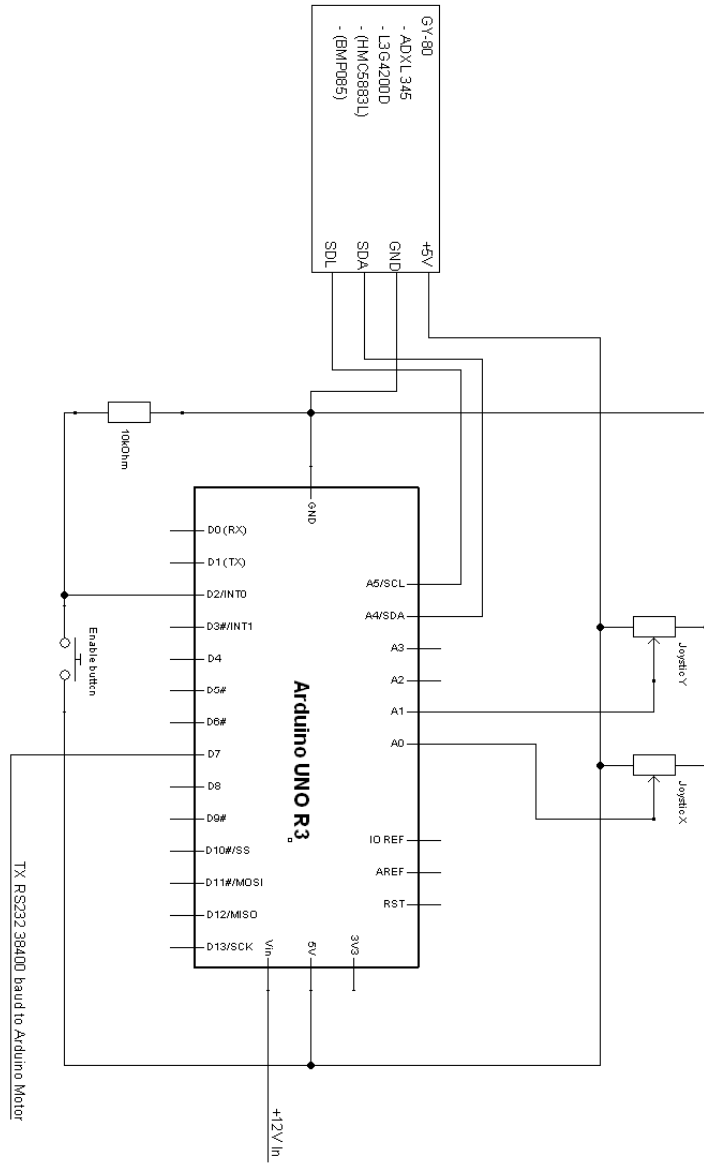


Title	Battery bridge	Document
Author	Olli-Pekka Karjalainen & Lauri Laara	Sheets
File	C:\Users\Pek\Desktop\Sebawa\Sillaus.dsn	1 of 1
Revision	Date	
1.0		

Liite 2. Moottorinohjain sähkökuva

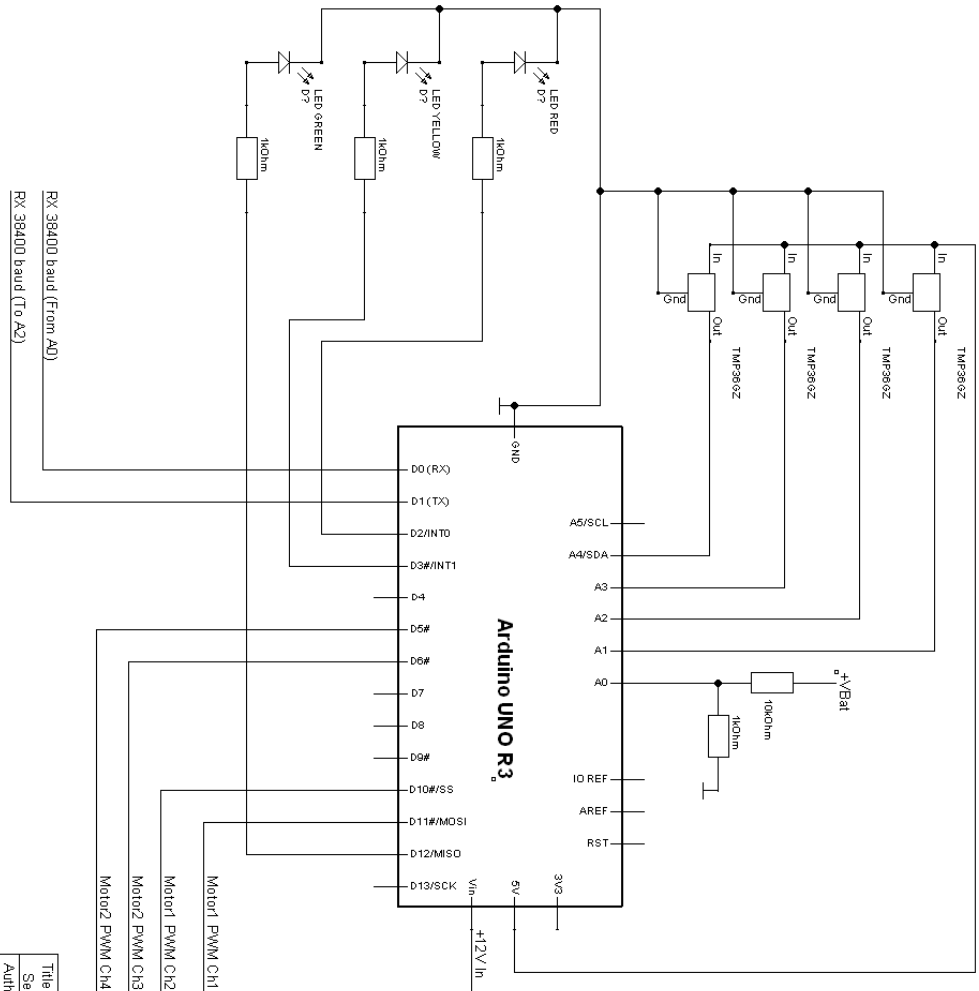


Liite 3. Arduino 0 sähkökuva



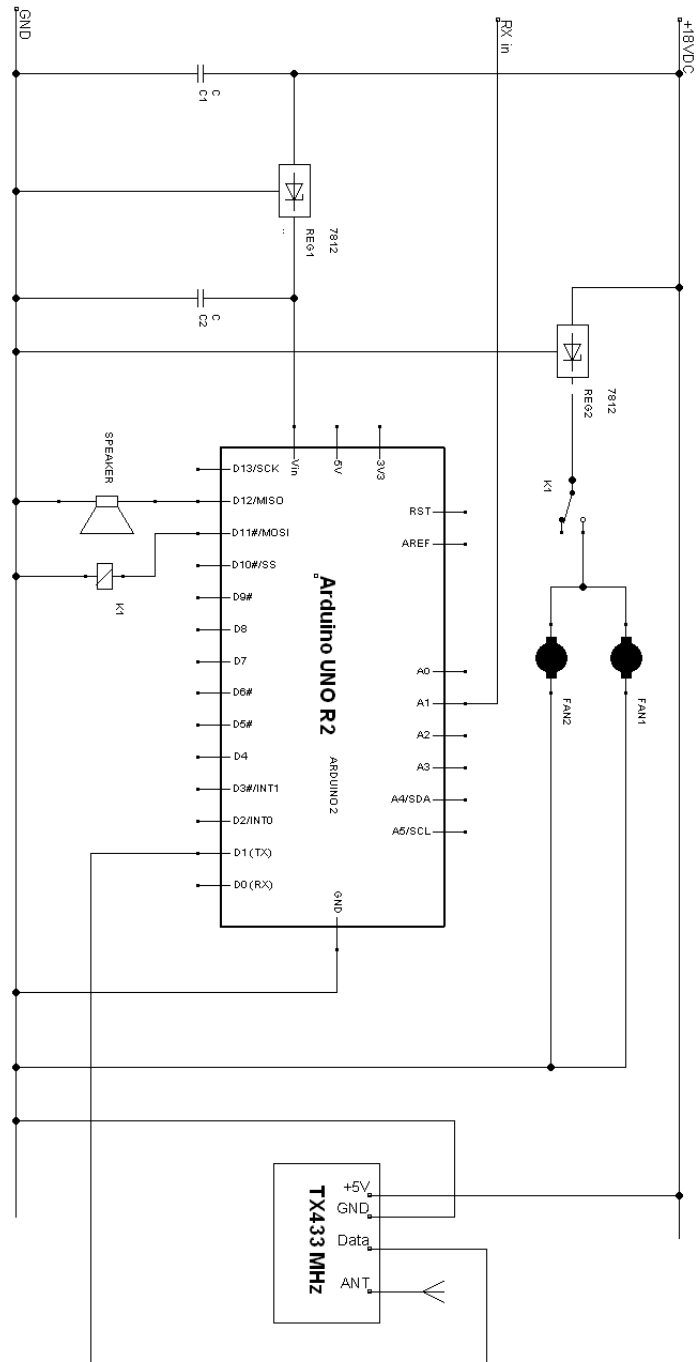
Title	Arduino Gyro board	
Author	Lauri Laara & Olli-Pekka Karjalainen	
File	projekti\SeBaWi\Sähkökuvat\Arduino 0 gyro.dsn	Document
Revision	Date	Sheets
1.0		1 of 1

Liite 4. Arduino 1 sähkökuva



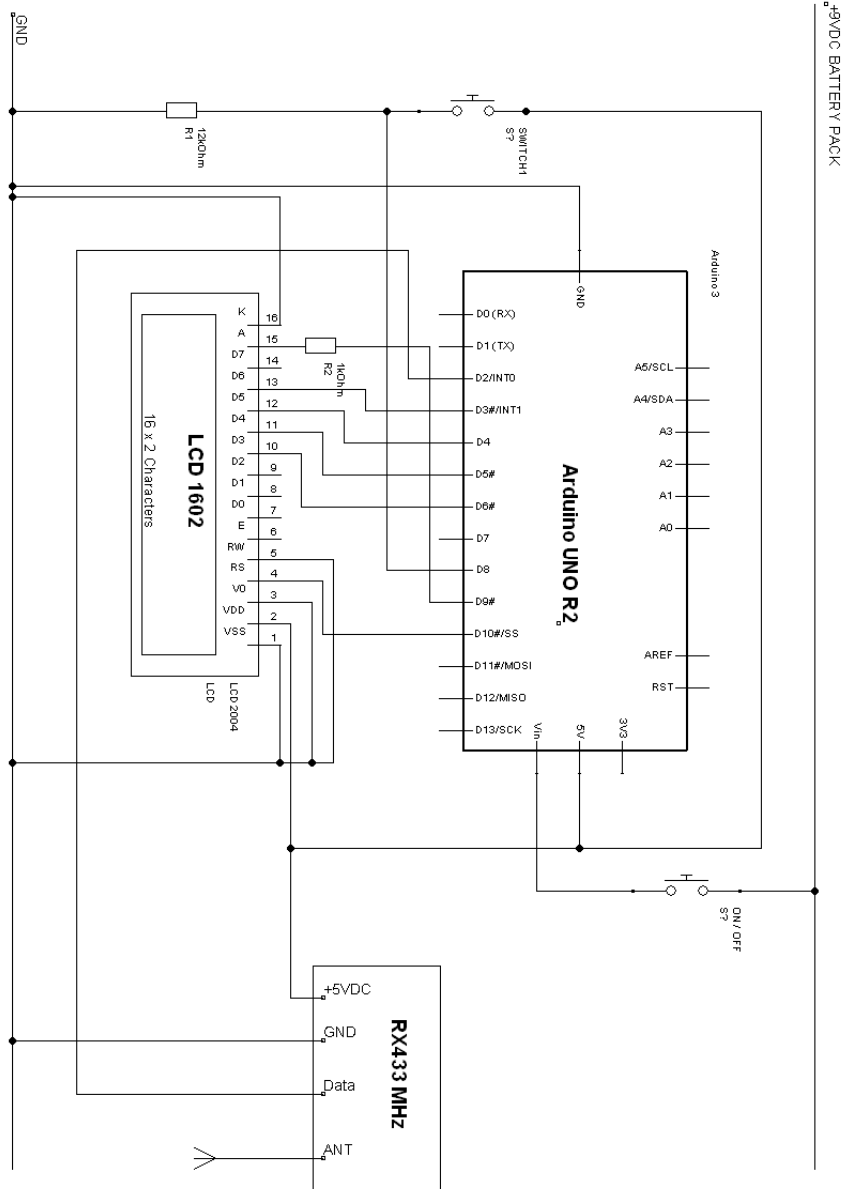
Title	Sebawe A1	
Author	Lauri Laara & Olli-Pekka Karjalainen	
File	projektit\SeBawe\SahkoKurat\Arduino 1 motor.dsn	
Revision	Date	Document
1.0		Sheets 1 of 1

Liite 5. Arduino 2 sähkökuva



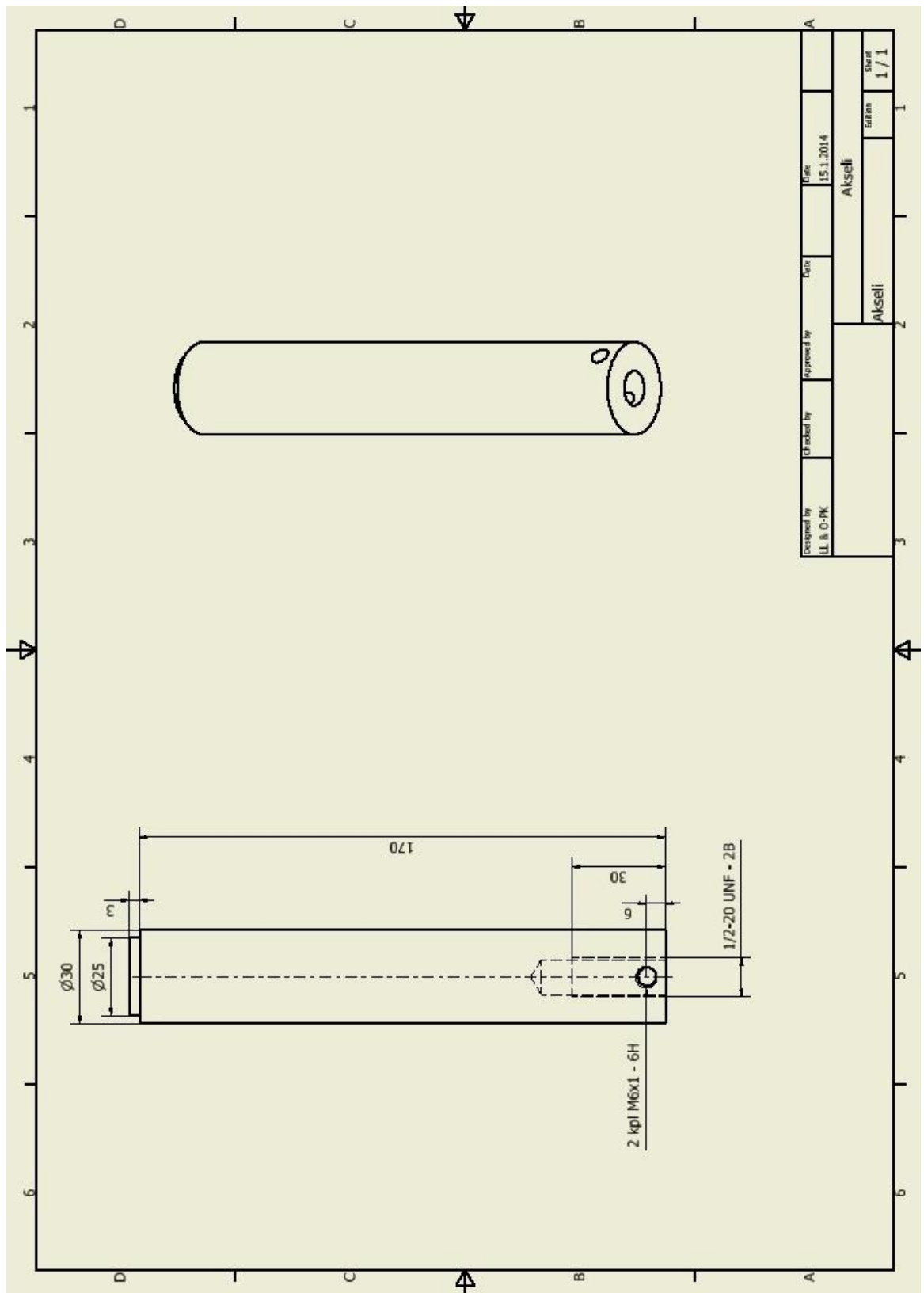
Title		Sähkö Arduino 2	
Author		Olli-Pekka Karjalainen & Lauri Lähtä	
File	ProjektinSäilytysSähkökuva\Arduino 2 data.dsn	Document	
Revision	1.0	Date	13.3.2014
		Sheets	1 of 1

Liite 6. Arduino 3 sähkökuva

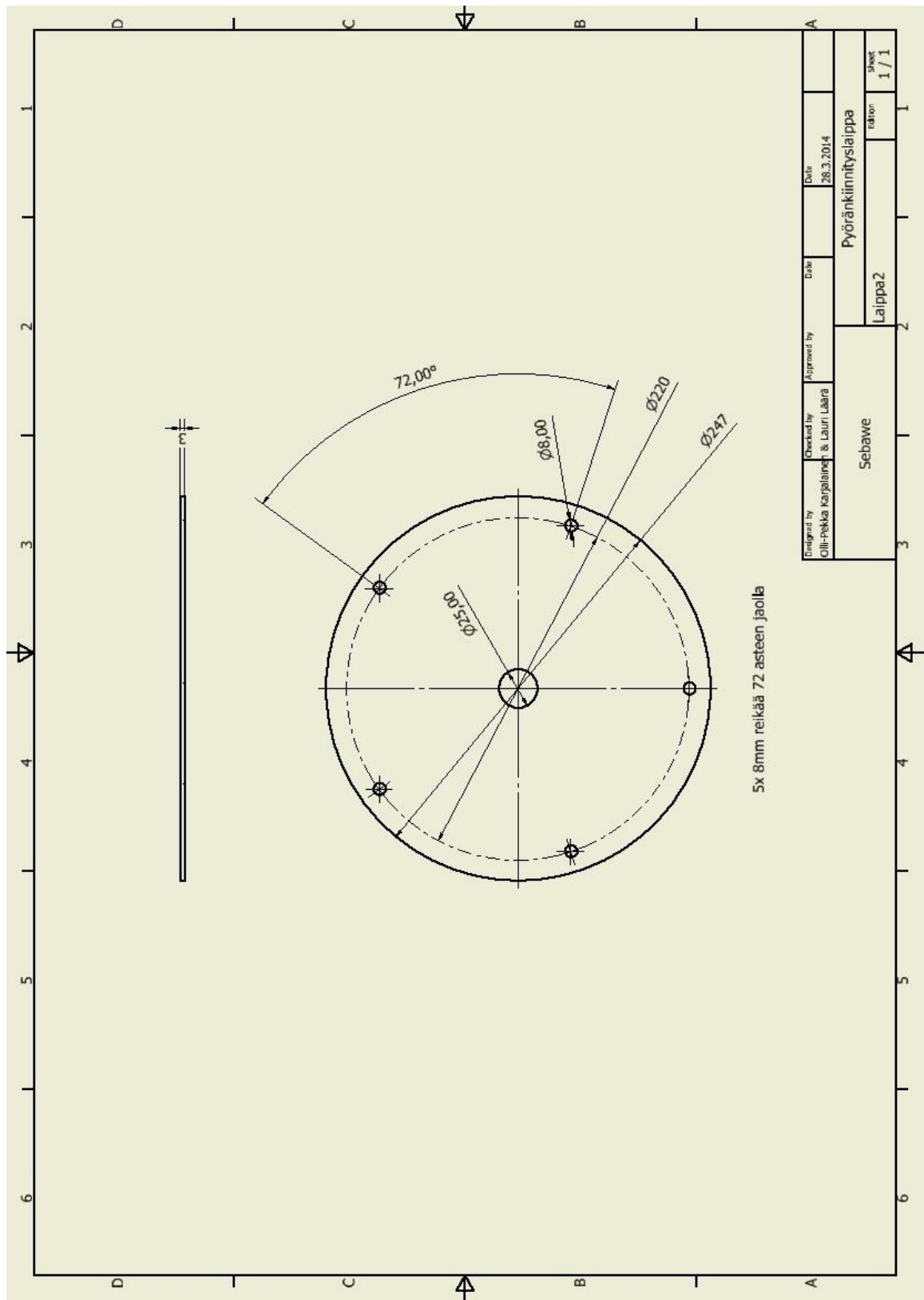


Title	Sehwise Arduino 3 Wireless receiver	
Author	Olli-Pekka Karjalainen & Lauri Laara	
File	3aivw/Sähkökuva/Arduino 3 wireless display.dsn	
Revision	Date	13.3.2014
1.0	Revision	1 of 1

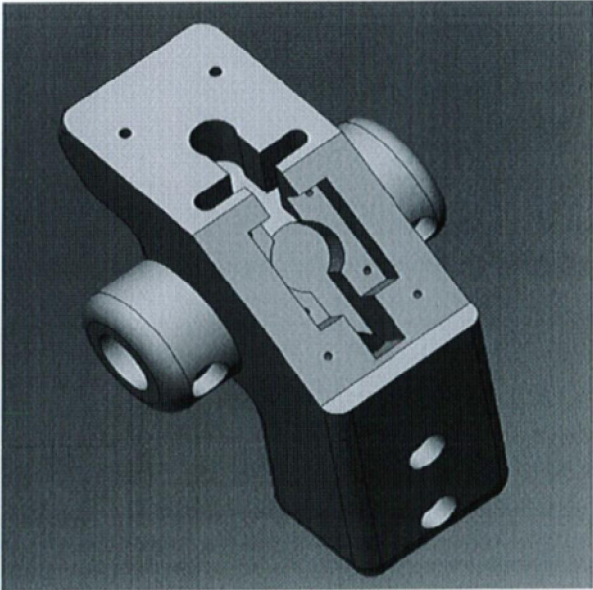
Liite 7. Akselin CAD-kuva



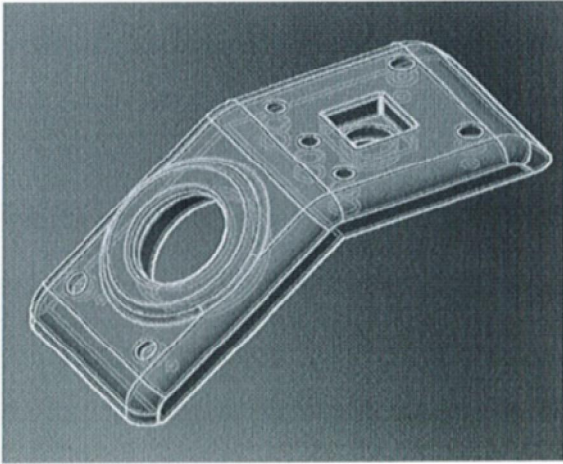
Liite 8. Laipan CAD-kuva



Liite 9. Kahvan tulostusdokumentti

Selostus 3D-tulostuksesta TAMKin tuotekehityslaboratoriossa (palautus sähköpostiosoitteeseen jani.katajisto@tamk.fi)	
Yhteyshenkilön nimi ja puh. nro.:	Olli-Pekka Karjalainen 0504001189
Opiskelija / henkilökunta? (O/H):	O
Koulutusohjelma:	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto:	Kone- ja laiteautomaatio
Kurssi, projekti, opinnäytetyö tms., johon tuloste liittyy:	Opinnäytetyö
Onko luottamuksellinen? (K/E)	E
Asiakasyritys /-taho (tarvittaessa):	
Kuvaus tulosteesta ja sen käyttötarkoituksesta:	Kahva
Tulostusasetukset:	
Resolution (mm):	0,178
Model interior:	SPARSE LOW DENSITY
Support fill:	SMART
Arvioitu materiaalin kulutus (cm ³):	112,92
Arvioitu tukiaineen kulutus (cm ³):	52,16
Arvioitu tulostusaika (tunteina):	31
Valokuva tulosteesta tai ruudunkaappaus mallista tähän:	
	

Liite 10. Kannen tulostusdokumentti

Selostus 3D-tulostuksesta TAMKin tuotekehityslaboratoriossa (palautus sähköpostiosoitteeseen jani.katajisto@tamk.fi)	
Yhteyshenkilön nimi ja puh. nro.:	Olli-Pekka Karjalainen 0504001189
Opiskelija / henkilökunta? (O/H):	O
Koulutusohjelma:	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto:	Kone- ja laiteautomaatio
Kurssi, projekti, opinnäytetyö tms., johon tuloste liittyy:	Opinnäytetyö
Onko luottamuksellinen? (K/E)	E
Asiakasyritys /-taho (tarvittaessa):	
Kuvaus tulosteesta ja sen käyttötarkoituksesta:	Kahvan kansi
Tulostusasetukset:	
Resolution (mm):	0,178
Model interior:	SPARSE LOW DENSITY
Support fill:	SMART
Arvioitu materiaalin kulutus (cm ³):	21,39
Arvioitu tukiaineen kulutus (cm ³):	13,48
Arvioitu tulostusaika (tunteina):	7,5
Valokuva tulosteesta tai ruudunkaappaus mallista tähän:	
	

Liite 11. Kannakkeiden tulostusdokumentti

Selostus 3D-tulostuksesta TAMKIn tuotekehityslaboratoriossa (palautus sähköpostiosoitteeseen jani.katajisto@tamk.fi)	
Yhteyshenkilön nimi ja puh. nro.:	Olli-Pekka Karjalainen 0504001189
Opiskelija / henkilökunta? (O/H):	O
Koulutusohjelma:	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto:	Kone- ja laiteautomaatio
Kurssi, projekti, opinnäytetyö tms., johon tuloste liittyy:	Opinnäytetyö
Onko luottamuksellinen? (K/E)	E
Asiakasyritys /-taho (tarvittaessa):	
Kuvaus tulosteesta ja sen käyttötarkoituksesta:	Kaksi kappaletta kannakkeita
Tulostusasetukset:	
Resolution (mm):	0,178
Model interior:	SPARSE HIGH DENSITY
Support fill:	SMART
Arvioitu materiaalin kulutus (cm ³):	113,48
Arvioitu tukiaineen kulutus (cm ³):	18,62
Arvioitu tulostusaika (tunteina):	16,36
Valokuva tulosteesta tai ruudunkaappaus mallista tähän:	
