



# **MITTAUS JA DATANKERÄYS LIKKUVASTA KOHTEESTA**

Topi Talvitie

Opinnäytetyö  
Tammikuu 2015  
Konetekniikka  
Älykkäät koneet

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Konetekniikka  
Älykkäät koneet

TOPI TALVITIE:

Mittaus ja datankeräys liikkuvasta kohteesta

Opinnäytetyö 56 sivua, joista liitteitä 16 sivua  
Tammikuu 2015

---

Tämä opinnäytetyö on tehty Tampereen Ammattikorkeakoululle, auto- ja kuljetustekniikan opetuksen käyttöön. Työn tavoitteena oli valmistaa kokoluokan 1:8 sähkökäyttöiseen radio-ohjattavaan autoon mittauslaitteisto, jolla voidaan tutkia ajoneuvon jousituksen liikkeitä reaaliajassa eri ajotilanteissa. Työssä haluttiin käyttää halpoja ja helposti saatavilla olevia komponentteja, jotta valmis mittauslaitteisto innostaisi opiskelijoita toteuttamaan omia projektejaan.

Instrumentoinnissa käytettiin sekä kaupallisia, että omavalmisteisia antureita, joille valmistettiin kiinnikkeet PLA muovista pikavalmistustekniikalla. Jousituksen liikkeet mitattiin lineaariantureilla ja auton nopeus vanteen pyörähdyksiä havaitsevalla infrapunasensorilla. Mittauslaitteiston keskusyksiköksi valittiin Arduino Uno R3 mikrokontrolleri. Arduinolle luotiin ohjelma, joka vastaanottaa sekä käsittelee antureilta tulevan mittausdatan. Datan käsittelyn jälkeen se kirjoitetaan reaaliajassa Arduinon sarjaporttiin tiedonsiirtoa varten. Tiedonsiirtoon tietokoneelle testattiin XBee radiomoduulein luotua yhteyttä. Testien tulokset olivat lupaavia, mutta kehitystyö jouduttiin keskeyttämään laitevian ja saatavuus ongelmien vuoksi. Valmis mittauslaitteisto ei toimi langattomasti, vaan lähettää mittausdataa usb-yhteyttä pitkin.

Työ osoittaa että mittaus ja datankeräys on mahdollista toteuttaa Arduinon pohjautuvalla laitteistolla. Laitteiston kellotaajuus riittää käsittelemään antureilta tulevat signaalit, sekä lähettämään mittausdatan radioyhteydellä reaaliajassa. Radio-ohjattavan auton nopeuden mittaus vanteen pyörimisliikettä havainnoimalla ei tuota tarkkaa tietoa nopeudesta, mikäli rengas pääsee luistamaan ajoalustalla. Havainnointi kiinteästä mittauspisteestä mahdollistaisi luotettavamman nopeuden mittaamisen. Tiedonsiirto langattoman lähiverkon yli, sekä datan jatkokäsittely kaupallisen mittausohjelmiston avulla on suositeltavaa.

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Mechanical Engineering  
Intelligent Machines

Topi Talvitie:  
Measurement and data acquisition from a moving object

Bachelor's thesis 56 pages, appendices 16 pages  
January 2015

---

This thesis was ordered by Tampere University of Applied Sciences, and will be used as a teaching device in the vehicle laboratory. The goal of this thesis was to design and build a data acquisition device to be used with a 1:8 scale radio controlled brushless powered car. The device would be able to measure the movements of the car's suspension in different driving situations. Furthermore, the device would be build using affordable and easy to acquire parts, so that the finished product would inspire students to engineer their own projects.

Both commercial and self-made sensors were used in making the data acquisition device. All the necessary parts to assemble the device were manufactured from PLA plastic filament with a 3d printer. Linear actuators were used to measure the movements of the suspension and an infrared sensor to measure the spinning of the car's tire. Arduino Uno R3 microcontroller board was used as data processing unit. A specialized program was made to read and process the signals from the sensors. XBee radio modules were tested for the method of data transfer, but had to be dismissed because of malfunctions and availability issues. The finished device does not operate wirelessly, but transfers measurement data via usb connection.

This thesis demonstrates that a data acquisition device is possible to build by using Arduino based technology. Clock speed of the Arduino is enough to process and transfer measurement data in real time. Further improvements should include speed measurement from a fixed measurement point. At the moment the device will not give accurate speed information if the car's tires slip on the ground. Another advisable improvement would be to use a WiFi connection to transfer the data and commercial data acquisition software for graphical presentation of the data.

---

Key words: mechanics, measurements, instrumentation, arduino

## SISÄLLYS

1	JOHDANTO.....	7
2	DATANKERÄYS .....	8
2.1	Kaupalliset järjestelmät.....	8
2.2	Anturit .....	9
2.3	Arduino .....	9
3	MITTAUSLAITTEISTO .....	12
3.1	Komponentit .....	12
3.1.1	Penny&Giles lineaarianturit.....	12
3.1.2	Infrapunasensori.....	13
3.1.3	Arduino Uno R3.....	14
3.1.4	XBee.....	15
4	KÄYTETYT TYÖVÄLINEET.....	17
4.1	Työkalut .....	17
4.2	Prenta Duo .....	17
4.3	Ohjelmistot.....	18
4.3.1	Arduino IDE.....	18
4.3.2	Autodesk Inventor Professional 2013 .....	19
4.3.3	Processing .....	20
4.3.4	X-CTU.....	21
5	INSTRUMENTOINTI .....	22
5.1	Kiinnikkeet.....	22
5.1.1	Lineaariantureiden kiinnitys.....	22
5.1.2	Korin kannakkeiden muutos .....	24
5.1.3	IR-sensorin kiinnitys .....	26
5.2	Arduinon kotelointi.....	27
5.3	Lineaarianturien kytkentä .....	30
5.4	IR-sensorin kytkentä .....	31
6	OHJELMA .....	35
6.1	Nopeuden määrittäminen .....	38
7	POHDINTA.....	39
	LÄHTEET.....	40
	LIITTEET .....	41
	Liite 1. Arduino Dimensions and Hole Patterns.....	41
	Liite 2. Anturin alakiinnike .....	42
	Liite 3. Anturin yläkiinnike .....	43
	Liite 4. Keulan kannake.....	44

Liite 5. Perän kannake .....	45
Liite 6. Arduinon kotelo .....	46
Liite 7. Infrapunaledin tekniset tiedot .....	47
Liite 8. Valovastuksen tekniset tiedot .....	50
Liite 10. IR-sensorin runko.....	53
Liite 11. SLS-095 lineaarianturin tekniset tiedot .....	54
Liite 12. Arduino ohjelma .....	56

**LYHENTEET JA TERMIT**

Shield	Arduinon päälle asennettava piirilevy, jolla pystytään lisäämään mikrokontrollerin ominaisuuksia.
LDR	Light Dependent Resistor; photoresistor; puolijohdemateriaalista valmistettu valovastus, jonka resistanssi laskee valon vaikutuksesta.
I/O-pinni	Digitaalisesti ohjattu nasta, joka pystytään asettamaan joko sisääntuloksi tai lähdöksi.
Serial Monitor	Ikkuna, joka näyttää sarjaportin kautta kulkevan datan.
IDE	Integrated Development Environment, ohjelmisto joka mahdollistaa suoraviivaisen ohjelmien luonnin.
IR	Infrapuna (engl. Infrared), valon aallonpituudet välillä 700 nm – 1 mm
WiFi	Wireless local area network (WLAN) -tuotteista yleisesti käytetty nimitys.
C++	Tärkeimpiä kaupallisesti käytettyjä ohjelmointikieliä.
RC	Radio Controlled, radio-ohjattu.

## 1 JOHDANTO

Tuotekehityksessä laitteen kunkin osan toiminnan tunteminen on ensiarvoisen tärkeää. Tämän opinnäytetyön tavoite on suunnitella ja toteuttaa instrumentointi auton iskunvaimennuksen käyttäytymistä kuvaavaan mittauslaitteistoon. Valmis laitteisto tulee ajoneuvotekniikan opetuksen käyttöön.

Mittauslaitteiston tulee kyetä keräämään dataa jousituksen liikkeistä, sekä RC-auton nopeudesta, ja lähettämään tämä data tietokoneelle tarkasteltavaksi. Opetuskäytössä olevan RC-auton on tarkoitus kuvata täysikokoisen ajoneuvon käyttäytymistä, jolloin instrumentointi ei saa oleellisesti muuttaa RC-auton ajo-ominaisuuksia. Mittauslaitteiston tulee olla kevyt, eikä se saa estää pääsyä niihin komponentteihin, joihin normaalin käytön ja huollon aikana tulee päästä käsiksi.

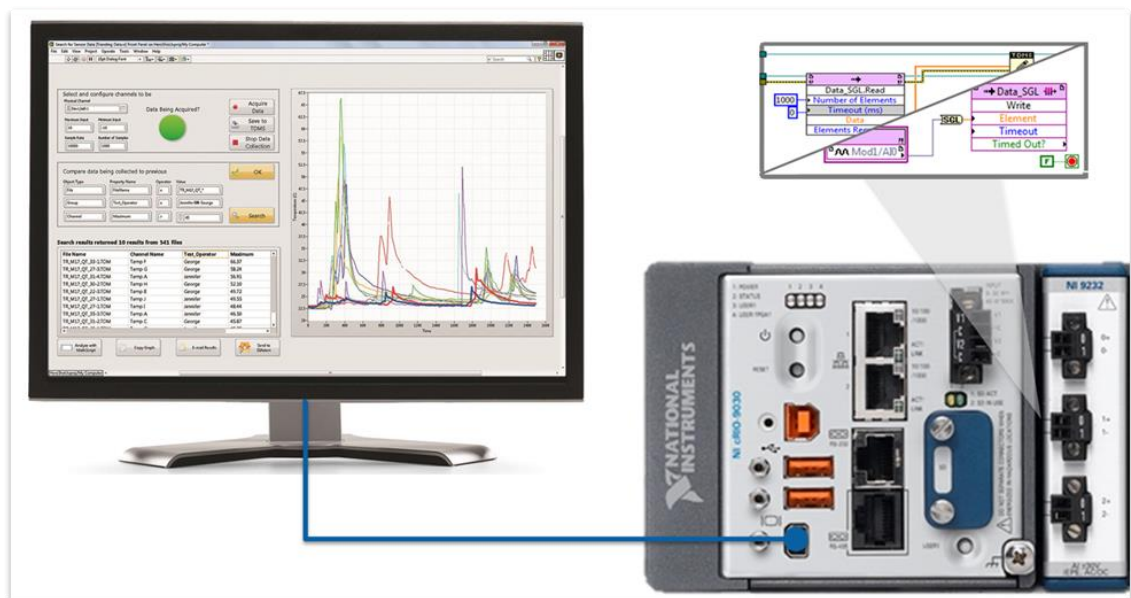
Tässä työssä esitellään myös ne ohjelmalliset perusteet, joiden mukaan mittauslaitteisto toimii. Työn piiriin ei kuitenkaan kuulu ohjelmoinnin yleinen teoria, ja lukijalta oletetaan kappaleen 6 OHJELMA kohdalla ohjelmoinnin perusteiden tuntemusta. Työn keskeisimmän sisällön muodostaa instrumentointi ja siihen liittyvien komponenttien valmistus.

## 2 DATANKERÄYS

### 2.1 Kaupalliset järjestelmät

Monet yritykset tarjoavat valmiita ratkaisuja mittaukseen ja instrumentointiin. Nämä järjestelmät ovat usein hyvin kattavia ja mahdollistavat monipuolisen ja nopean datankeräyksen ja -käsittelyn. Markkinat ovat suuntautuneet prosessin seurantaan ja ohjaukseen, sekä sulautettuihin järjestelmiin koneenrakennuksessa ja erityisesti ajoneuvotekniikassa.

Monipuolisen laitteistotarjonnan rinnalla on tarjolla kehittyneitä ohjelmistoja, jotka mahdollistavat mittausdatan käsittelyn ja tarkastelun kulloisenkin tilanteen vaatimalla tavalla. Yksi suurimmista laitteisto ja ohjelmisto valmistajista on National Instruments, jonka LabView ohjelmointiympäristöä käyttää maailmanlaajuisesti yli 6000 yliopistoa. (us.ni.com/academic/software, 2014.)



KUVA 1. LabView ja NI Rio

(<http://www.ni.com/cms/images/devzone/tut/phdksfgd3041570997824202028.png>)

Kaupallisesti järjestelmät ovat hyvin pitkälle kehitettyjä ja niiden komponentit ovat hyvin koteloituja, silloin kun niitä ei ole tarkoitettu sulautettuihin järjestelmiin jo valmistusvaiheessa liitettäväksi. Niin antureilta kuin dataloggereiltakin vaaditaan pitkää käyttöikää, usein haastavissa olosuhteissa. Vankat koteloinnit ja lujat, lukittavat liittimet

tuovat käyttövarmuutta, mutta tekevät samalla laitteista isokokoisia. Ajoneuvojen ja työkoneiden yhteydessä laitteiden koko ei kuitenkaan muodostu ongelmaksi. Tämän työn kannalta oleellista on kuitenkin keveys ja pieni koko. Lisäksi mittauslaitteistolta vaaditaan suhteellisen pienet määrät liitäntöjä kaupalliseen tarjontaan verrattuna.

## 2.2 Anturit

Anturi määritellään usein laitteeksi, joka vastaanottaa ja reagoi signaaliin tai ärsykkeeseen. Antureita on mahdollista löytää lähes jokaisesta laitteesta ja niillä suoritetaan mitä moninaisimpia tehtäviä. Oli kyseessä sitten prosessinohjaus tai ajoneuvon luis-tonestojärjestelmä, anturit ovat keskeisessä osassa toteutusta. Maailma on jaettavissa kahteen osaan, ihmisen luomaan ja luonnolliseen. Anturein suoritetuissa mittauksissa näiden kahden osan yhteistoiminta on kriittisessä roolissa. Esimerkiksi ulkoinen voima saa anturin sisältämässä materiaalissa aikaan sähkönjohtavuuden muutoksen, mikä havaitaan anturiin kytketyssä virtapiirissä. Näin laitteeseen vaikuttava voima on muutettu sähköiseen muotoon, jota pystytään käsittelemään elektronisesti.

Datankeräystä on suoritettu jo kauan ennen kuin anturitekniikka kehittyi nykyiselle tasolle, jopa ennen kuin varsinaisista antureista alettiin puhua. Koneenkäyttäjän silmä tarkkaili laitteen toimintaa ja kädet tekivät tarvittavia toimenpiteitä, jotta työ saatiin suoritettua. Testipilotit ja –ajajat käyttivät kokemustaan hyväksi arvioidessaan miten tehdyt muutokset vaikuttivat koneen toimintaan. Nykyään monet näistä toiminnoista on korvattu automaatiolla, jonka anturien tuottama mittausdata mahdollistaa. Ilmiöiden mittaaminen on hyödyllistä myös silloin kun se ei liity suoranaisesti jonkin toiminnon suorittamiseen. Muun muassa ajoneuvojen toiminnan tarkastelu on antureiden keräämän datan myötä helpottunut ja suoraviivaistunut. Teknisten muutosten sekä olosuhteiden vaikutus ajoneuvon toimintaan pystytään todentamaan anturein ja näin kehittämään entistä tehokkaammin toimivia kokonaisuuksia. (Fraden, 2004.)

## 2.3 Arduino

Arduino on mikrokontrolleri-elektroniikka-alusta ja ohjelmistoympäristö. Itse mikrokontrolleri koostuu piirilevystä joka sisältää prosessorin, muistia, digitaalisia ja analogi-

sia I/O-portteja sekä muuta kunkin Arduino-laitteen tarvitsemaa elektroniikkaa. (KUVA 2.) Alkujaan Arduino kehitettiin helposti lähestyttäväksi avoimeksi laitteistoksi, nimenomaan opiskelijoita ja muita innokkaita varten. Koska kyseessä on avoin laitteisto, on Arduinon, ja kaikkien sen pohjalta rakennettujen laitteiden, suunnitelmat ja käyttämiin vaaditut laiteajurit vapaasti saatavilla ja muunneltavissa. Arduino onkin laajentunut useiden yritysten hyödyntämäksi järjestelmäksi, joka tarjoaa laajan kirjon mikrokontrollereja ja Arduino-yhteensopivia lisäosia. Siinä missä yksi tuote tarjoaa täysin vapaasti muokattavaa prototyypialustaa, toinen on tarkoitettu aivan tietynlaisten projektien toteuttamiseen, kuten vaatteisiin ommeltavan järjestelmän keskusyksiköksi suunniteltu LilyPad Arduino. (Banzi, M. 2011. Getting Started with Arduino. Make:Books.)



KUVA 2. Arduino Uno R3 (<http://arduino.cc>)

Ohjelmistoltaan Arduino pohjautuu C-kieleen ja sitä pystyykin laajentamaan C++ -kirjastoilla. Eräs Arduinon eduista onkin toisaalta helposti lähestyttävä ohjelmointikieli, joka ei kuitenkaan rajoita edistyneempien käyttäjien mahdollisuuksia. Toinen ratkaiseva etu moniin muihin järjestelmiin on hinta. Arduinolla prototyyppien rakentaminen ja ideoiden testaus on halpaa eikä vaadi sitoutumista tietyn valmistajan järjestelmiin. Avoimeen lähdekoodiin pohjautuvana Arduinon ympärille kehittynyt yhteisö tarjoaa paljon valmiita ratkaisuja, mikä helpottaa käytön oppimista ja omaksumista.

Arduino-laitteiden eri mallit tarjoavat valinnanvaraa muistin, prosessointitehon, I/O-porttien määrän sekä muiden ominaisuuksien osalta. (TAULUKKO 1.) Näiden lisäksi on tarjolla eri tarkoituksiin räätälöityjä lisälaitteita, joita kutsutaan englanninkielisellä

termillä *shield*, vapaasti käännettynä: kilpi. Nimensä mukaisesti nämä lisäosat kiinnittyvät suoraan Arduino-laitteen päälle ja alle jäävän piirilevyn nastat kulkevat lisälaitteen läpi, jolloin niiden käyttö ei esty. Näillä lisälaitteilla Arduinoon saadaan lisättyä muun muassa langattomia ominaisuuksia (WiFi, GSM, radiolähetin-vastaanotin), ethernet-portti, moottorinohjaus, USB isäntä jne. (<http://arduino.cc>, 22.12.2014.)

TAULUKKO 1. Arduino-laitteita.

	Uno R3	Mega 2560	Micro	Nano
Mikrokontrolleri	ATmega328	Atmega2560	ATmega32u4	ATmega328
Käyttöjännite	5V	5V	5V	5V
Sisääntulojännite (suositus)	7-12V	7-12V	7-12V	7-12V
Sisääntulojännite (rajat)	6-20V	6-2V	6-20V	6-20V
Digitaalinen I/O	14 (joista 6 PWM)	54 (joista 15 PWM)	20 (joista 7 PWM)	14 (joista 6 PWM)
Analoginen sisääntulo	6	16	12	8
Virta (DC) / I/O pinni	40 mA	40 mA	40 mA	40 mA
Virta (DC) 3.3V pinnille	50 mA	50 mA	50 mA	-
Flash muisti	32 KB	256 KB	32 KB	32 KB
Kellotaajuus	16 MHz	16 MHz	16 MHz	16 MHz
Pituus	68.6 mm		48 mm	18.5 mm
Leveys	53.4 mm		18 mm	43 mm
Paino	25 g		13 g	

## 3 MITTAUSLAITTEISTO

### 3.1 Komponentit

Komponentteja valittaessa pyrittiin saavuttamaan mittauslaitteistolle mahdollisimman pieni koko. Auton liikkeet ovat hyvin pieniä joten anturien valinnassa tuli kiinnittää huomiota myös riittävään tarkkuuteen. Mittauslaitteiston keskusyksiköksi valittiin Arduino Uno R3 –mikrokontrolleri sen pienen koon, avoimen lähdekoodin ja kattavien ominaisuuksien vuoksi.

Mittauslaitteisto tulee opiskelijoiden käyttöön, joten sen haluttiin rakentuvan alustalle jota he voivat vapaasti muokata ja kehittää edelleen. Näin mittauslaitteiston opetukselista hyötyä pystytään kasvattamaan jatkossa ja opiskelijat saavat halutessaan tarkemman kuvan laitteiston toiminnasta. Komponenttien valinnassa kiinnitettiin mahdollisuuksien mukaan huomiota myös niiden saatavuuteen.

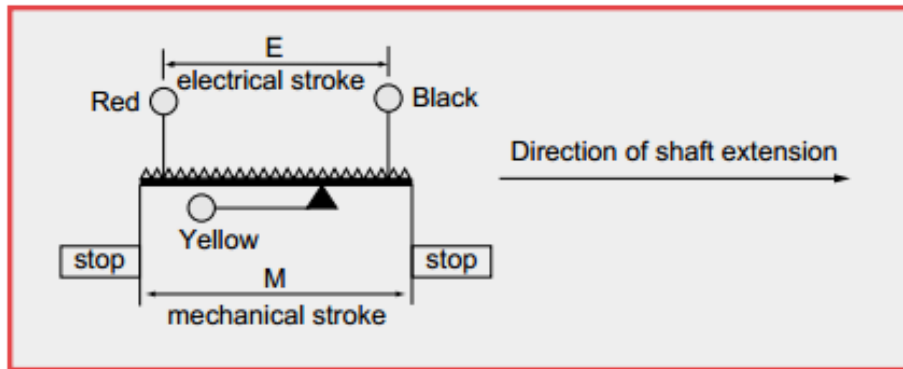
#### 3.1.1 Penny&Giles lineaarianturit

Iskunvaimentimien liikettä mittaamaan valittiin Penny&Giles SLS095 lineaarianturit (KUVA 3.). SLS095 on moottoriurheilun tarpeisiin kehitetty sarja erittäin pienikokoisia lineaariantureita, joiden tarkkuus on teoriassa ääretön. Mittauslaitteistossa tarkkuutta rajoittaa ainoastaan analogisen sisääntulon tarkkuus. Antureihin valittiin nivelletyt ja laakeroidut kiinnityspisteet, sekä varren tiiveysluokaksi IP66. Anturit tulevat sijaitsemaan renkaiden välittömässä läheisyydessä ajoradan elementeille alttiina, joten hyvin suojattu anturin varsi on välttämättömyys.



KUVA 3. SLS095-lineaarianturi. (<https://www.smart-home-products.de/media/images/popup/lindisp.JPG>)

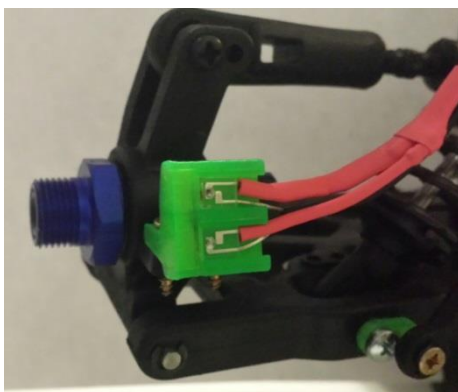
Sarjasta valittiin 20 mm elektronisella iskulla oleva yksilö (KUVA 4.). Mekaanisen iskun ollessa 22,5 mm jää kumpaankin päähän 1,25 mm mittauskelvotonta liikettä. Valintaan vaikutti erityisesti kiinnityspisteiden etäisyys, ja sitä kautta anturin kokonaispituus, jonka ei haluttu kasvavan liian suureksi. SLS095/20 kiinnityspisteiden etäisyys on 80 mm, anturin ollessa täysin sisäänvedetty.



KUVA 4. SLS095 periaatekuva ( Penny+Giles, Motorsport Sensors Brochure)

### 3.1.2 Infrapunasensori

RC-auton nopeuden määrittämiseksi päätettiin käyttää renkaan yhteyteen asennettua infrapunasensoria. Sensori koostuu infrapunaledistä ja infrapunalle herkästä valovastuksesta (LDR), jonka resistanssi laskee valon vaikutuksesta (KUVA 5.). Sensorin toiminta perustuu infrapunaledin säteilemään valoon, joka heijastuu vanteen sisäpintaan kiinnitystä heijastavasta pinnasta. Renkaan pyöriessä heijastava pinta kulkee sensorin ohi, laskien valovastuksen resistanssia.



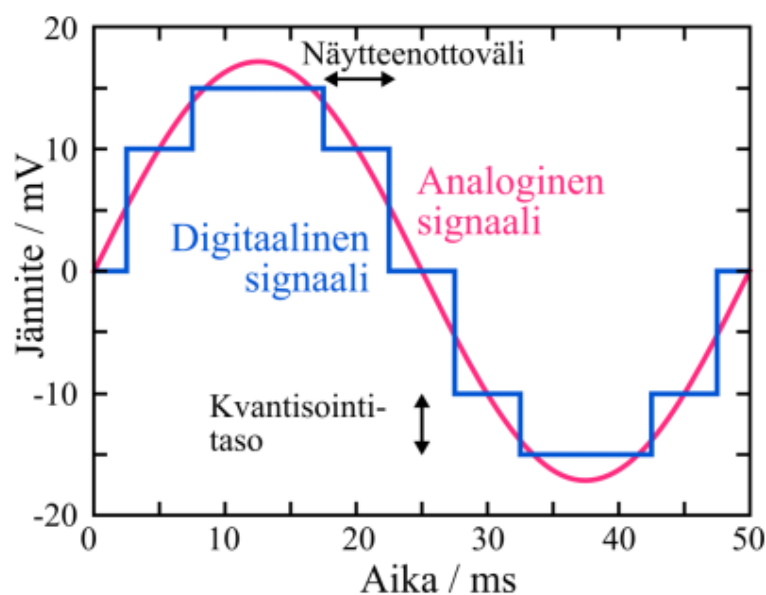
KUVA 5. IR-sensori.

Käytetyt komponentit ovat Sparkfunin valmistamia ja tyypiltään sivulle katsovia. Komponentit toimivat 940 nm valon aallonpituudella. Liitteissä 7 ja 8 on esitetty infrapunaledin ja valovastuksen tekniset tiedot. Sensorin runko valmistettiin 3D-tulostimella, jotta komponentit saatiin asemoitua vanteen suhteen oikein.

### 3.1.3 Arduino Uno R3

Mittauslaitteiston ytimeksi valittiin Arduino Uno R3 mikrokontrolleri. Tätä mikrokontrolleria käytetään TAMKin konetekniikan opiskelijoiden mekatroniikan koulutuksessa, joten se on opiskelijoille tuttu. Arduino Uno sisältää ATmega328 prosessorin, joka kykenee 16 MHz kellotaajuuteen, sekä 32 kilotavua muistia. Analogisia sisääntuloja Unossa on kuusi kappaletta ja digitaalisia I/O-pinnejä 14, joista kuuden yhteydessä voidaan käyttää pulssinleveysmodulaatiota (PWM, Pulse-Width Modulation). Tässä työssä tarvitaan neljä analogista sisääntuloa lukemaan kunkin lineaarianturin liike, joten liitännöitä on riittävästi. Digitaalisia liitännöitä voidaan haluttaessa hyödyntää esimerkiksi eri kytkimien yhteydessä. Kompaktin kokonsa vuoksi Arduino Uno soveltuu hyvin liikkuvan laitteiston keskusyksiköksi. (Arduino. 2014.)

Mittauslaitteiston kannalta merkittävimpiä Arduinon ominaisuuksia ovat sen AD-muuntimen ominaisuudet. Analogisten sisääntulojen antureilta saama jännitesignaali muutetaan AD-muuntimella digitaalseksi. (KUVA 6.)



KUVA 6. AD-muunnos periaatekuva. (Pulli, T. 2014)

AD-muuntimen resoluutio on 10 bittiä, mikä tarkoittaa 1024 jännitetasoa (0 – 1023). Referenssijännitteen ollessa oletusasetuksessaan viidessä voltissa, vastaa luku 0 nollaa voltia ja luku 1023 viittä voltia. Näytteistystaajuus on suurimmillaan noin 10 kHz eli 10 000 näytettä sekunnissa.

### 3.1.4 XBee

XBee on ZigBee protokollaa käyttävä 2.4 GHz radiolähetinvastaanotin, josta on tullut hyvin suosittu langattomien sensoriverkkojen parissa. Tässä työssä käyttämämme XBee radio on uudempaa Series 2 –mallia. Se poikkeaa aiemmasta Series 1 –mallista paremmalla kantamalla ja matalammalla virrankulutuksella. Series 1:stä poiketen Series 2 käyttää ZigBee'n omaa sensoriverkkojen luontiin optimoitua laiteohjelmistoa. (TAULUKKO 2.)

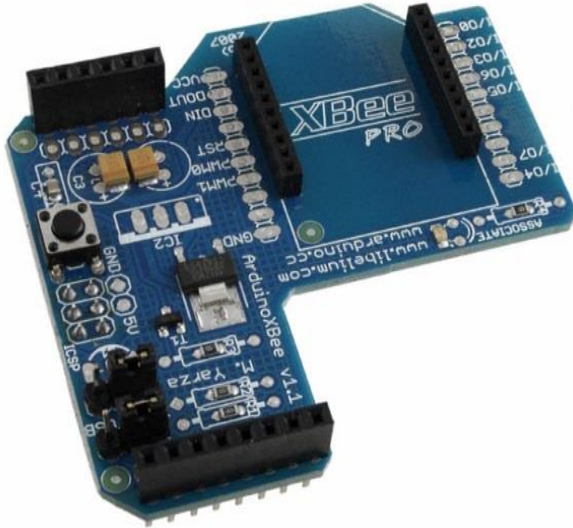
TAULUKKO 2. XBee vertailu.

Ominaisuus	Series 1	Series 2
Tyypillinen kantama sisätiloissa	30 metriä	40 metriä
Suurin kantama	100 metriä	120 metriä
Lähetys-/vastaanottovirta	45/50 mA	40/40 mA
Laiteohjelmisto	802.15.4 point-to-point	ZB ZigBee mesh
Tähtitopologia	Kyllä	Kyllä
Mesh-topologia	Ei	Kyllä
RF Datanopeus	250 Kbps	250 Kbps

XBee radioita valmistetaan usealla erityyppisellä antennilla. Käyttämässämme radioissa on PCB-antenni, joka on painettu suoraan piirilevyille. PCB-antennilla on samat edut ja heikkoudet kuin siruantennissa (Chip-antenna). Nämä antennityypit ovat kestävämpiä ja pienempiä kuin lanka-antenni, mutta niiden säteilykuvio on sydämenmuotoinen. Tällöin lähetysteho ei ole yhtä hyvä joka suuntaan. XBeetä on saatavilla myös U.FL ja RPSMA liittimillä, jolloin siihen voi yhdistää erillisen ulkoisen antennin.

Radio liitetään Arduinoon XBeetä varten suunnitellulla Arduino XBee Shield –laajennusosalla (KUVA 7.). Laajennusosan avulla Arduino pystyy käyttämään radioyh-

teyttä sarjaportin tapaan. Kun kahden XBee radiot on asetettu ja yhteys luotu, toimii varsinainen ohjelma kuten langallisella yhteydellä.



KUVA 7. Arduino XBee Shield.

(<http://media.digikey.com/Photos/Arduino/A000021.JPG>)

## 4 KÄYTETYT TYÖVÄLINEET

### 4.1 Työkalut

Laitteiston valmistamiseen ja asentamiseen käytettiin tavanomaisia työkaluja, kuten erikokoisia risti- ja talttapääruuvimeisseleitä, elektroniikkapihtejä ja työntömittaa (KUVA 8.). Lisäksi käytettiin juotoskolvia IR-sensorin ja liitososien valmistamiseen, sekä kuumailmapuhallinta johdotusten suojaamiseen kutistesukilla.

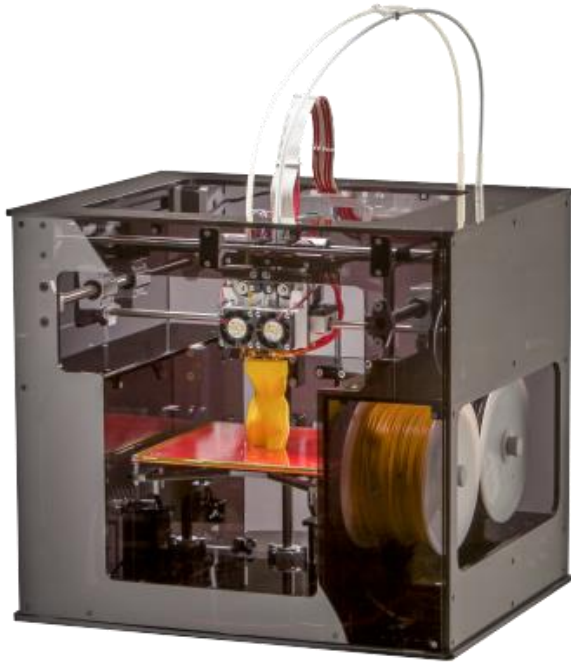


KUVA 8. Työkaluja

### 4.2 Prenta Duo

Mittauslaitteiston osien valmistamiseen käytettiin Prenta Duo 3D-tulostinta. (KUVA 9.) Tulostinta ohjattiin Repetier Host -ohjelmistolla ja kappaleet viipaloitiin Slic3r-ohjelmalla. Tulostimen suutinkoko on 0,5 mm ja siinä käytetään 1,75 mm filamenttia.

Tulostus lämpötilana käytettiin 195 – 210 °C. Kerrosvahvuutena käytettiin osasta riippuen 0,15 – 0,3 mm. Nämä arvot ovat kuitenkin järjestelmä- ja olosuhderiippuvaisia.

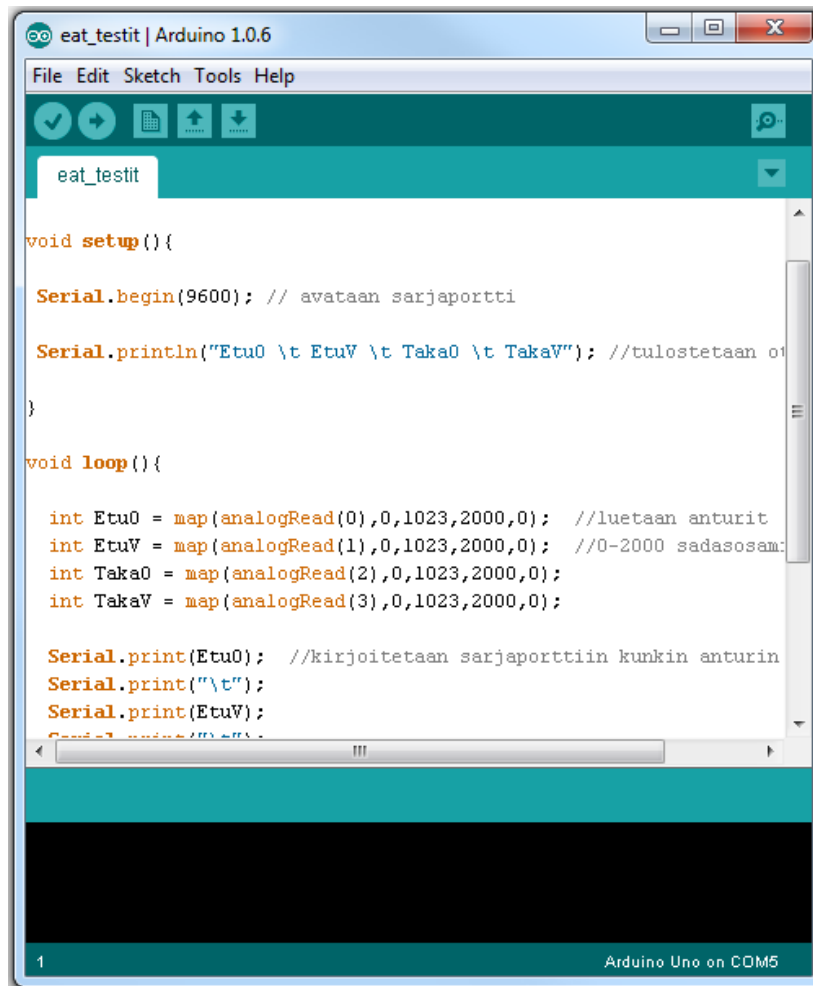


KUVA 9. Prenta Duo ([http://www.prenta.fi/images/duo/duo1\\_400.png](http://www.prenta.fi/images/duo/duo1_400.png))

### 4.3 Ohjelmistot

#### 4.3.1 Arduino IDE

Arduino IDE (Integrated Development Environment) on ohjelmointiympäristö jolla luodaan ja siirretään ohjelmia (*sketch*) Arduino mikrokontrollerille. (KUVA 10.) IDE:n kautta on myös mahdollista käyttää *Serial Monitoria* ja tarkkailla sen avulla Arduinon sarjaportin kautta lähettämää dataa. Arduinon käyttämä ohjelmointikieli on C++:n kaltaista ja ohjelmointikieltä voidaan laajentaa C++ kirjastoilla. Arduinon ohjelmointiympäristö pohjautuu Processing kieleen.



KUVA 10. Arduino IDE

Ohjelmointiympäristö on rakennettu siten että ohjelman luominen on johdonmukaista ja selkeää. Oikein kirjoitetut komennot värittyvät automaattisesti, joten tehdyt virheet huomaa välittömästi. Lisäksi käännohjelma ilmoittaa kaikki rivit joilla on jokin ohjelman suorittamisen päättävä virhe. (Karvinen, T. & K. 2009.)

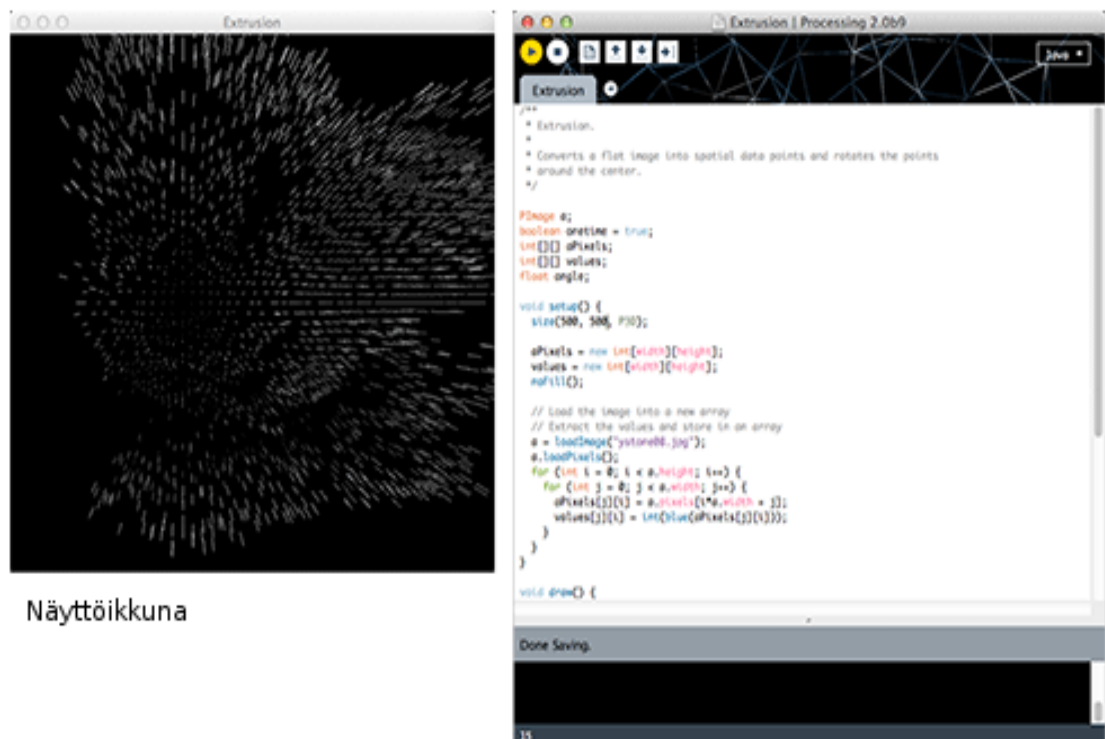
#### 4.3.2 Autodesk Inventor Professional 2013

Kaikki kappaleet mallinnettiin Autodesk Inventor Professional 2013 –ohjelman opiskelijaversiolla. Kappaleiden mallit tallennettiin sekä ipt- että .stl-muotoisina. Stl-tiedosto on kappaleen pintamalli, jota 3D-tulostimien hallintaan tarkoitettujen ohjelmien yleisesti lukevat. Mallinnusohjelman avulla pystyttiin mittauslaitteiston osia sovittamaan toisiinsa ennen niiden valmistamista.

Autodesk Inventor Professional on TAMKissa opetuskäytössä oleva ohjelmisto, jonka opiskelijat saavat ilmaiseksi käyttöönsä myös henkilökohtaiselle tietokoneelleen. Opiskelijat pystyvät hyödyntämään tämän työn aikana luotuja malleja, ja muokkaamaan niitä omiin projekteihinsa sopiviksi.

### 4.3.3 Processing

Processing on avoimen lähdekoodin ohjelmointikieli ja -ympäristö. (KUVA 11.) Se on ensisijaisesti rakennettu taiteiden käyttöön visuaalisuuteen pohjautuvien ohjelmien ja järjestelmien luomiseksi. Processingin sisältämät piirtotyökalut mahdollistavat visuaalisesti tietoa esittävien ohjelmien luonnin. Tiedon esittämisen lisäksi Processingin avulla pystytään luomaan visuaalisia käyttöliittymiä, muun muassa Arduinoa käyttäville järjestelmille. Tässä työssä Processingia käytettiin luomaan yksinkertainen ohjelma, joka lukee sarjaportilta Arduinon lähettämää dataa ja kirjoittaa sen tekstitiedostoon.

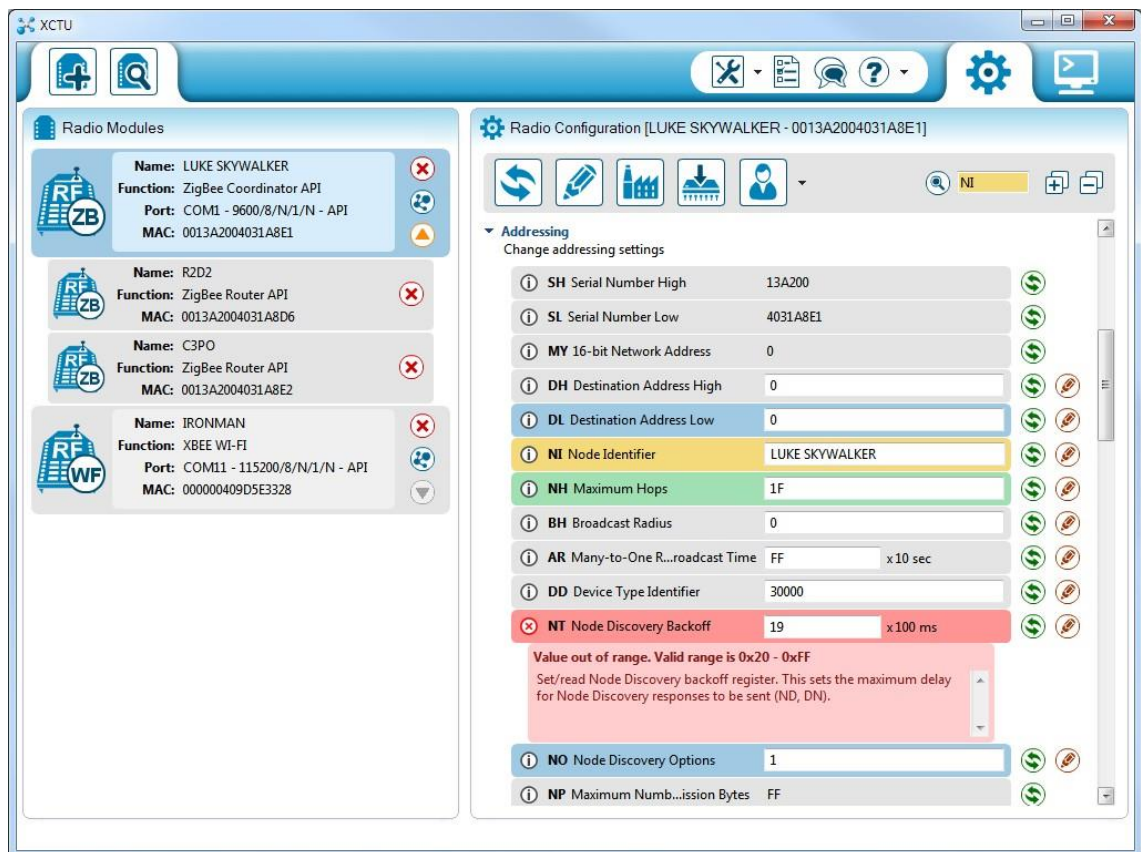


Näyttöikkuna

KUVA 11. Processing IDE (<https://processing.org/reference/environment/>)

### 4.3.4 X-CTU

XBee radiomoduulien asetusten säätämiseen käytettiin ilmaista X-CTU ohjelmaa. X-CTU on Digi Internationalin kehittämä ohjelma, jolla voidaan ohjelmoida ja testata Digi RF tuotteita, jollaisia tässä työssä käytetyt XBee radiomoduulit ovat. X-CTU hyödyntää selkeää graafista käyttöliittymää (KUVA 12.) jossa eri väreillä merkitään oletusasetuksista poikkeavat, tallentamattomat ja kelpaamattomat asetukset. Ohjelman avulla on myös mahdollista päivittää moduulin laiteohjelmisto ja tallentaa eri asetusprofileja myöhempää käyttöä varten, mikä helpottaa testausta kuhunkin käyttötarkoitukseen parhaiten soveltuvien asetusten löytämiseksi.



KUVA 12. X-CTU ([http://www.faludi.com/wp-content/uploads/2013/09/xctu\\_2.jpg](http://www.faludi.com/wp-content/uploads/2013/09/xctu_2.jpg))

## 5 INSTRUMENTOINTI

### 5.1 Kiinnikkeet

Mittauslaitteiston sijoittaminen kompaktiksi suunniteltuun radio-ohjattavaan autoon vaati pienikokoisten komponenttien lisäksi myös kiinnikkeiden huolellista suunnittelua. Auton rakenne haluttiin säilyttää muuttumattoman, joten kiinnityksiä suunniteltaessa hyödynnettiin mahdollisimman paljon auton rakenteita. Kaikki kiinnikkeet suunniteltiin Autodesk Inventor Professional 2013 –mallinnusohjelmalla ja valmistettiin pursotustekniikkaan perustuvalla Prenta Duo 3D-tulostimella.

Kiinnikkeiden materiaaliksi valittiin 3D-tulostimella pursotettu PLA-muovi. Osat on mahdollista valmistaa myös muita muoveja käyttäen, kuten ABS ja Polykarbonaatti. PLA sallii suuremmat vaihtelut tulostusolosuhteissa ilman että kappaleeseen syntyy epämuodostumia 3D-tulostettavat kappaleet mahdollistivat työnaikaisen kiinnikkeiden prototyyppien nopean valmistamisen, mikä edesauttoi vähäisen tilan hyödyntämistä.

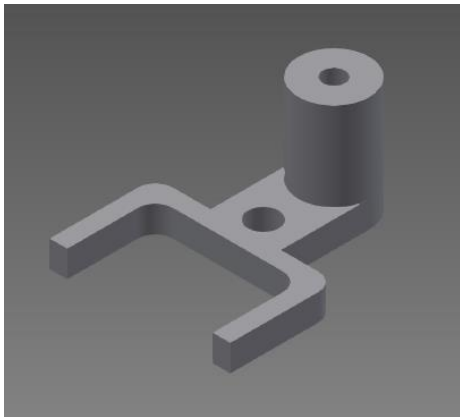
#### 5.1.1 Lineaariantureiden kiinnitys

Iskunvaimentimien liikettä mittaavat lineaarianturit haluttiin sijoittaa mahdollisimman tarkasti iskunvaimentimien rinnalle. Näin anturien liike vastaa suoraan iskunvaimentimien liikettä ja välttyään ylimääräisiltä ohjelmallisilta laskutoimituksilta. Sekä ylä- että alapään kiinnitys toteutettiin välikappaleilla joiden läpi iskunvaimentimen kiinnitys kulkee. Lineaarianturit kiinnitettiin ylä- ja alapäästään näihin välikappaleisiin siten, että anturin päissä olevat pallonivelet pystyvät edelleen toimimaan (KUVA 13.). Kuvasta on nähtävissä että vain pallonivelel sisempi osa on kosketuksissa kiinnikkeeseen.



KUVA 13. Anturin yläpään nivellys

Anturin yläpään kiinnitys nostettiin iskunvaimentimen kiinnityksen yläpuolelle, jotta varmistetaan anturin riittävä liikerata. Samalla saatiin myös kiinnike tukevasti omalla ruuvillaan kiinni iskunvaimentimen yläpään. Yläpään kiinnikkeeseen mallinnettiin jalat tukemaan kiinnike iskunvaimentimen yläosaan. (KUVA 14.) Nämä jalat ottavat vastaan mahdolliset kiinnitystä kiertävät voimat.



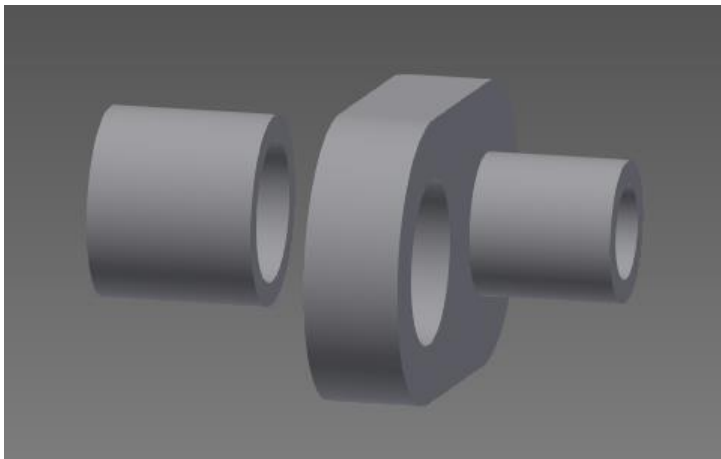
KUVA 14. Anturin yläpään kiinnike

Alapään kiinnitykselle tilaa oli niukemmin kuin yläpään kiinnityksellä. Iskunvaimentimen alapäälle on alatukivarressa kahdet kiinnitysreiät, joista toinen otettiin käyttöön tukemaan anturin alapään kiinnitystä. (KUVA 15.) Tämä ratkaisu mahdollisti hyvin tukevan kiinnityksen, mutta vei anturin alapään 4 mm sivuun iskunvaimentimen alapäästä, aiheuttaen täten lievän poikkeaman anturin ja iskunvaimentimen linjoissa. Poikkeama jää kuitenkin niin pieneksi, että anturi antaa luotettavan kuvan iskunvaimentajan liikkeistä, etenkin kun otetaan huomioon auton pyöräntuennan heitot ja välykset.



KUVA 15. Anturin alapään kiinnitys

Kiinnike mallinnettiin siten että sen pystyy valmistamaan 3D-tulostimella ilman tukimateriaalia. Koska kiinnike haluttiin lisäksi tukea iskunvaimentimen alapään kiinnitysreiän sisään, jouduttiin sen yhteydessä käyttämään erillistä holkkia anturin kiinnittämiseksi. (KUVA 16.)

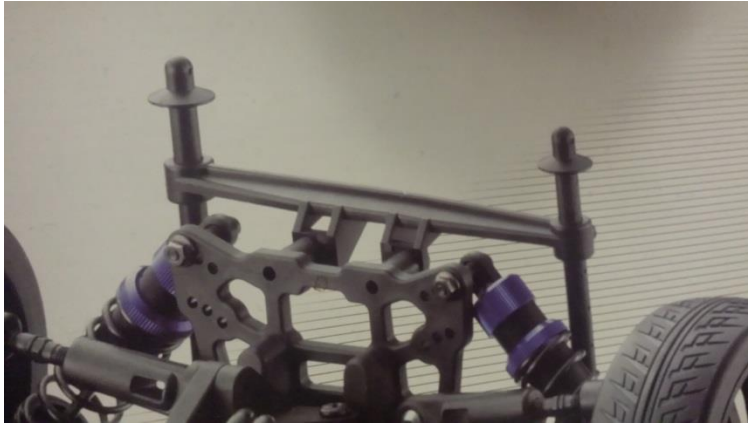


KUVA 16. Alapään kiinnike

### 5.1.2 Korin kannakkeiden muutos

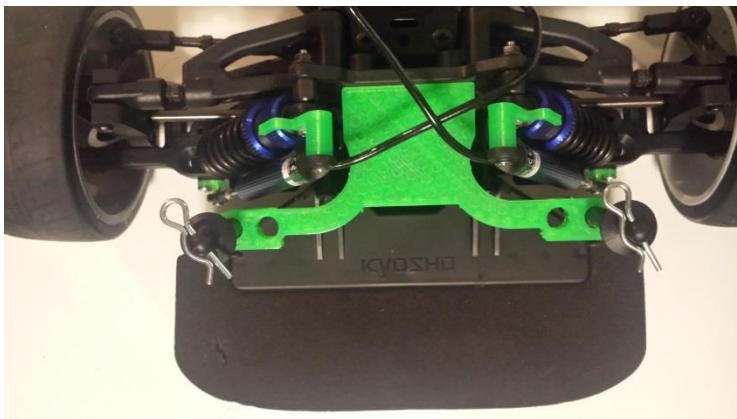
Radio-ohjattavan auton kevyt muovikori kiinnittyy runkoon muovitappien ja sokkien avulla (KUVA 17.). Näiden kiinnitysten sijaintia ei haluttu muuttaa, sillä kori on erityisesti renkaiden suhteen tarkasti mitoitettu. Lineaarianturit eivät kuitenkaan mahtuneet iskunvaimentimien rinnalle alkuperäisten kannakkeiden kanssa, mikä aiheutti kannakkeiden muutostyön. Alkuperäiset kannakkeet mitattiin tarkasti ja nämä mitat siirrettiin

mallinnusohjelmaan suunnittelutyön pohjaksi. Kannakkeille tehtiin anturien vaatimat muutokset siten, että kannakkeen kiinnitysreikien keskinäiset etäisyydet pysyivät muuttumattomina.



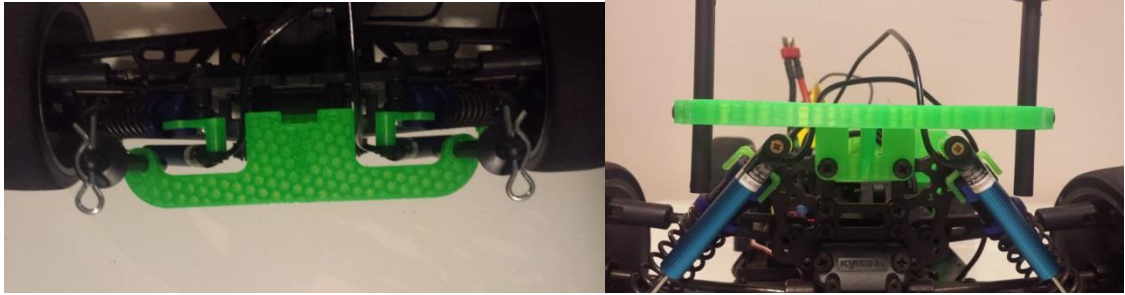
KUVA 17. Korin kiinnitys

Korin kannakkeet eivät saa estää lineaariantureihin tai iskunvaimentimiin kohdistuvia huoltotöitä. Keulan kannake sijaitsee jo alun perin iskunvaimentimien yläpään kiinnityksen alapuolella, joten pelkkä pituus suunnassa tehty tilan lisäys riitti keulan kohdalla (KUVA 18.).



KUVA 18. Keulan kannake

Auton keulalla on vaahtomuovi puskuri, jonka tarkoitus on ottaa vastaan mahdolliset törmäykset. Perässä tällaista puskuria ei ole vaan kori ottaa kannakkeidensa kautta vastaan perään kohdistuvat iskut. Koska perän kannaketta jouduttiin anturoinnin vuoksi korottamaan, kohdistuu törmäyksen aiheuttama voima kauemmas kannakkeen kiinnikkeiden linjasta. Tätä törmäystilanteessa lisääntyntä räsitystä kompensoitiin kannakkeen rakennetta vahvistamalla (KUVA 19.).



KUVA 19. Perän kannakointi

Kannakkeet tulostettiin 3D-tulostimella PLA-muovista joten rikkoutuneen tilalle pystytään valmistamaan nopeasti ja edullisesti uusi kappale.

### 5.1.3 IR-sensorin kiinnitys

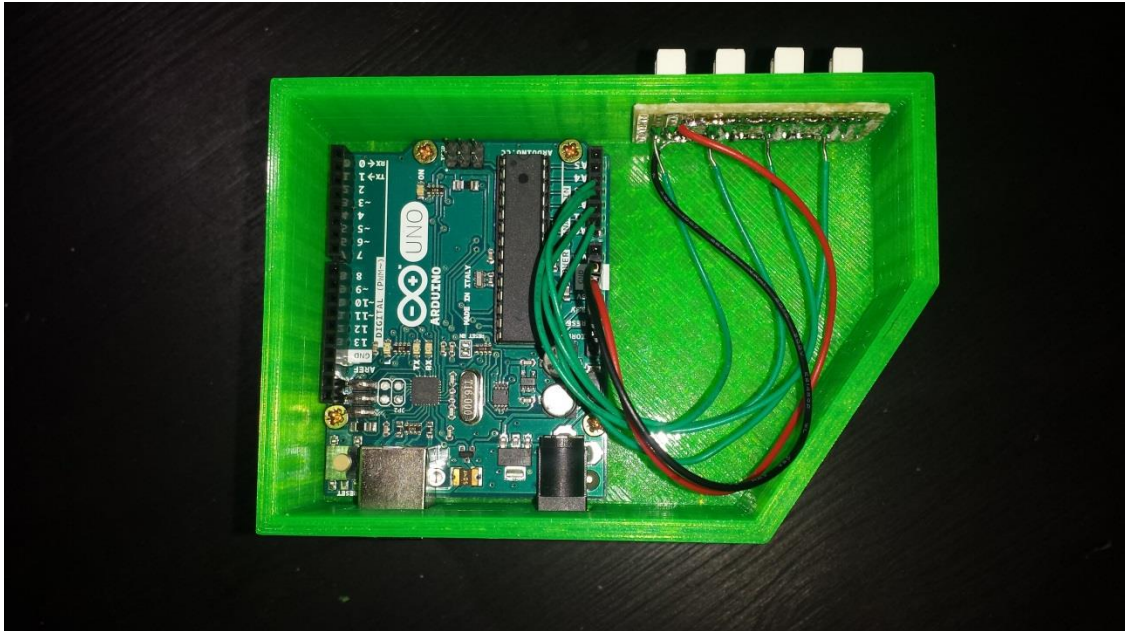
Infrapunasensori kiinnitettiin pyörännavan yhteydessä oleviin kiinnitysreikiin. Sensorin rungossa on infrapunakomponenttien muotoon sopivat hahlot, joihin komponentit voidaan liu'uttaa sivusta. Hahlot pitävät sekä säteilylähteen että sensorin suunnattuna kohti vanteen sisäpintaa. Rungon yläosaa kallistettiin kohti pyörännapaa, jotta sensori olisi yhdensuuntaisesti vanteen sisäreunan kanssa (KUVA 20.). Tällöin vanteen kaarevuus auttaa suuntaamaan infrapunaledin säteilemän valon kohti valovastusta.



KUVA 20. IR-sensorin kiinnike

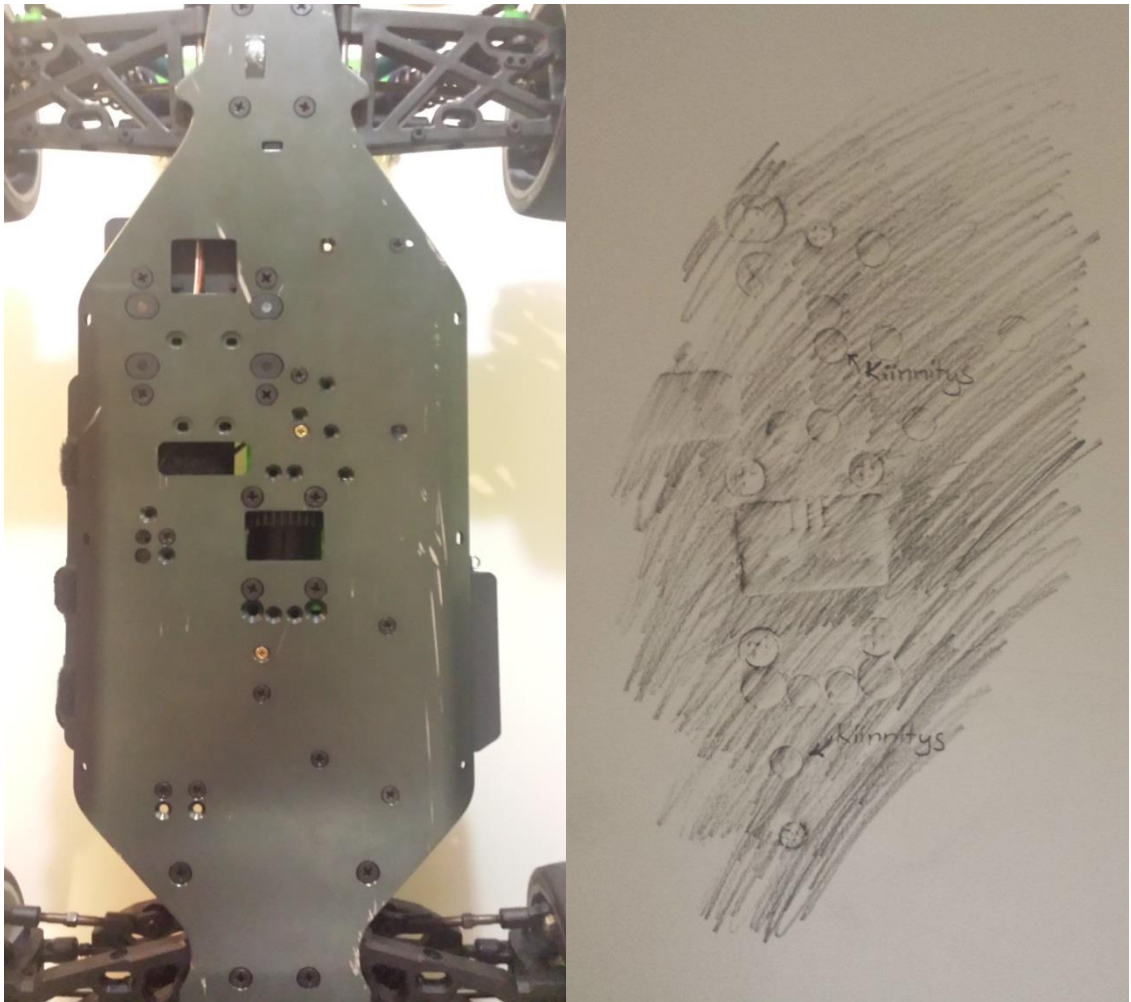
## 5.2 Arduinon kotelointi

Radio-ohjattavan auton keskelle suunniteltiin kotelo johon sijoitetaan Arduino Uno R3 mikrokontrolleri sekä anturien liittimet. (KUVA 21.) Lisäksi haluttiin paikka mittauslaitteiston virtalähteenä toimivalle paristolle. Paikka kotelolle löytyi moottorin ja jakovaihteiston yläpuolelta. Näin ollen kotelo ei estä pääsyä komponentteihin joihin normaalin käytön aikana tulee päästä käsiksi



KUVA 21. Kotelo

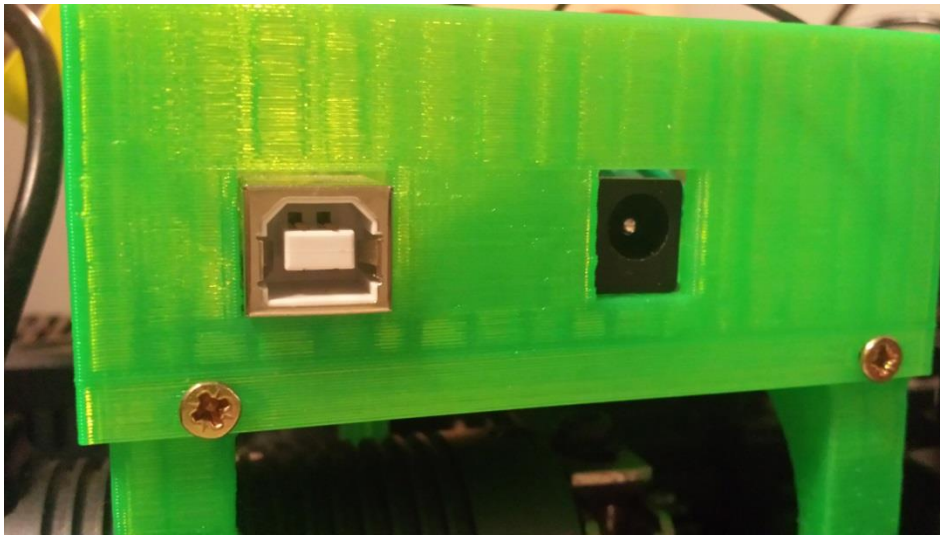
Kotelon suunnittelu lähti kiinnityspaikkojen kartoittamisesta. RC-auton pohjalevystä tehtiin lyijykynä jäljennös, josta työntömitalla mitattiin olemassa olevien ja käyttämättömien kiinnitysreikien sijainti. Tämän lisäksi mitattiin moottorin ja vaihteiston sijainnit sekä korkeudet. Pohjalevyssä oleva jakovaihteiston neliskanttinen aukko toimi sijaintien määrittämisessä mittauslaitteiston kiinnityspisteinä (KUVA 22.).



KUVA 22. Pohjalevy ja lyijykynäjäljennös

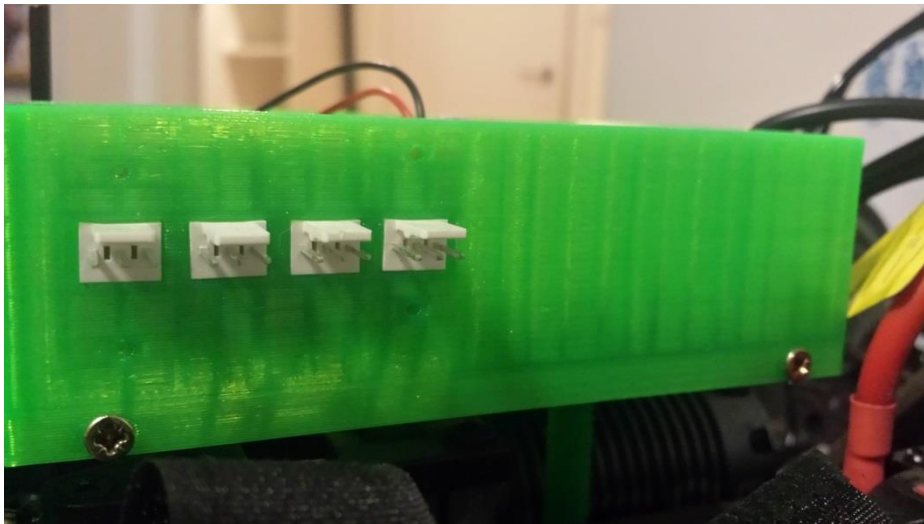
Kotelo tulisi koostumaan kolmesta osasta joista alimmaisina kiinnittää kotelon rc-auton pohjalevyyn ja nostaa sen irti muista komponenteista. Arduino kiinnittyy tähän kappaleeseen kiinnitysreikiensä kautta, joiden keskinäinen sijainti saatiin J. M. DeCristofaron piirustuksesta (Liite 1. Arduino Dimensions and Hole Patterns).

Ensimmäisen osan päälle tuleva seinäosa sisältää aukot anturien liittimille sekä Arduinon virta- ja USB-liittimille. Arduinon liittimiä varten tulevat aukot jätettiin väljiksi, jotta piirilevyn pystyy irrottamaan kotelosta riittävän helposti. (KUVA 23.)



KUVA 23. Arduinon liittimet

Anturien liittimien aukot mallinnettiin tiukoiksi jotta johtojen heilunta ei pääsisi rasittamaan liittimien juotoksia. (KUVA 24.)



KUVA 24. Anturien liittimet

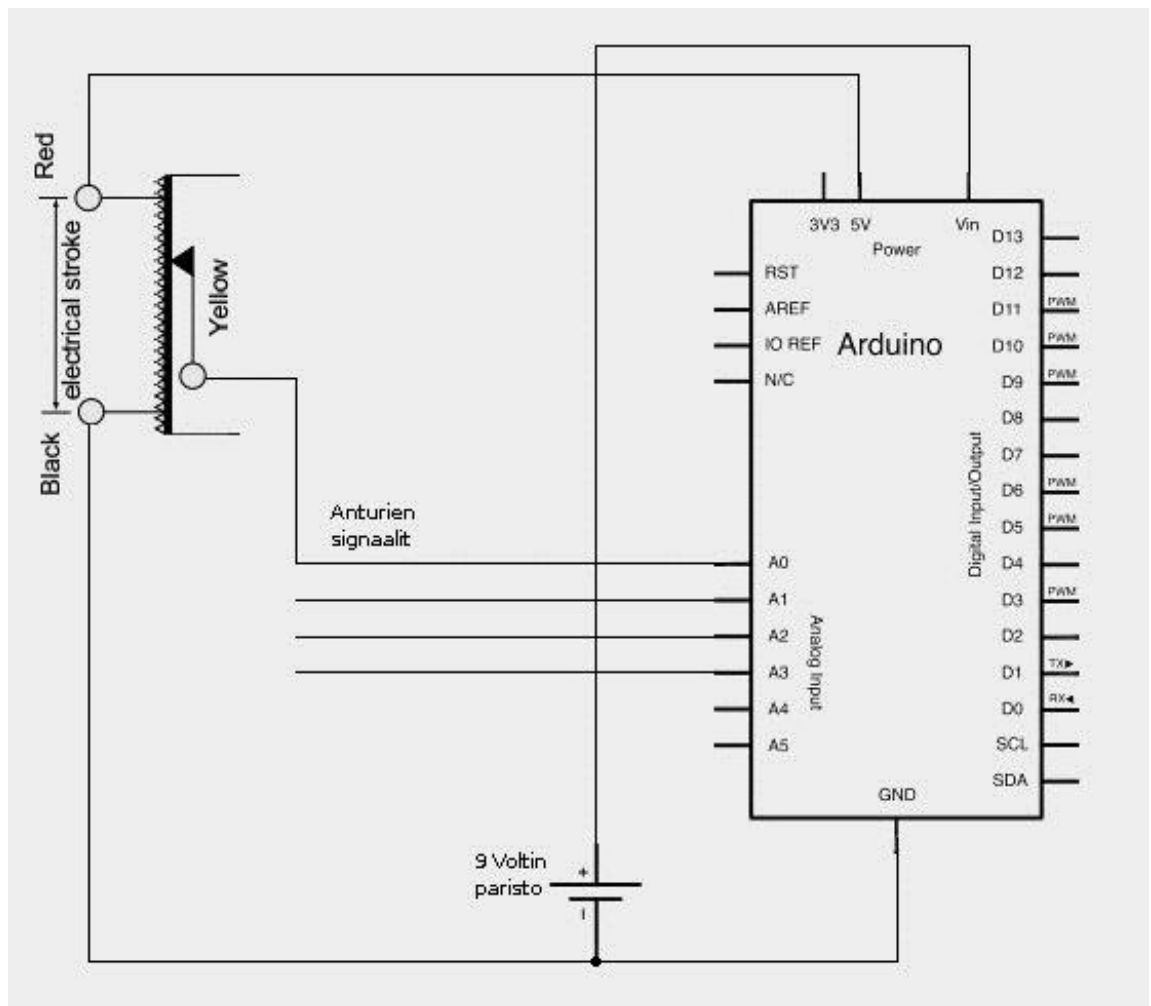
Kotelosta mallinnettiin riittävän suuri, jotta siihen mahtuu myös langattoman tiedonsiirron mahdollistava XBee lähetin-vastaanotin. XBee kiinnittyy Arduinon päälle erillisen shieldin avulla. Kotelon kolmas ja viimeinen osa on kansi joka estää lian ja vesiroiskeiden pääsyn koteloon. Kanteen on myös mahdollista integroida LCD-näyttö, jolla voidaan esittää mittauslaitteiston tilaa ja toimintaa, sekä mahdolliset virta- ja toimintakytkimet.

### 5.3 Linearianturien kytkentä

Linearianturit kytketään mikrokontrollerin analogisten sisääntulojen ja 5 voltin syötön väliin. (KUVA 25.) Linearianturi toimii potentiometrinä jonka vastus muuttuu varren ulkoneman mukaan. Analogista sisääntuloa lukemalla pystytään seuraavan kaavan avulla laskemaan anturin varren ulkonema 0,02 mm tarkkuudella.

$$\text{Varren ulkonema} = \frac{\text{Varren maksimi ulkonema}}{1023} \times \text{sisääntulon lukema}$$

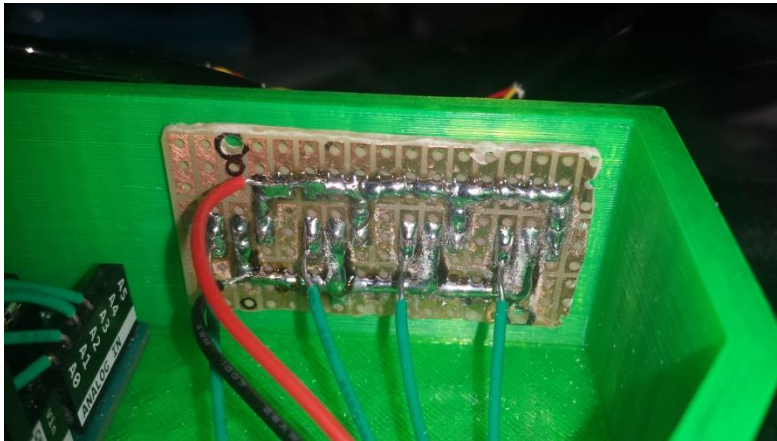
Saatu lukema kuvaa varren ulkonemaa sillä ajan hetkellä millä signaali on luettu.



KUVA 25. Anturien kytkennät

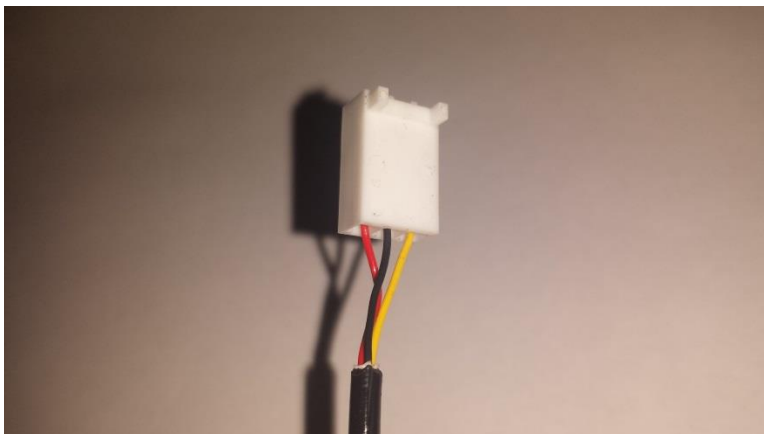
Kaikkien neljän anturin kytkemiseksi samaan jännitelähteeseen ja yhteiseen maadoitukseen, tehtiin erillinen kytkentälevy. (KUVA 26.) Tähän levyyn juotettiin jokaiselle anturille kolminapaiset KK-sarjan liitinpistukat. Kunkin liittimen yksi napa yhdistettiin

yhteen johtimeen joka vietiin Arduinon 5 voltin jännitelähtöön. Samoin yhdistettiin jokaisen liittimen toinen napa yhdeksi johtimeksi joka vietiin maadoitusnastaan. Kolmas napa käytettiin kunkin anturin signaalin viemiseksi omaan analogiseen sisääntuloonsa.



KUVA 26. Lineaarianturien kytkentälevy

Antureista lähtee valmiina kolminapainen johto, jonka päähän liitettiin liitinpistukkaa vastaava kolminapainen KK-sarjan pistokerunko. (KUVA 27.) Liitos anturien ja Arduinon välille toteutettiin erillisten liittimien avulla, suorien johtoliitosten sijaan, jotta anturit ovat irrotettavissa huoltojen ajaksi.



KUVA 27. Anturin liitin

#### 5.4 IR-sensorin kytkentä

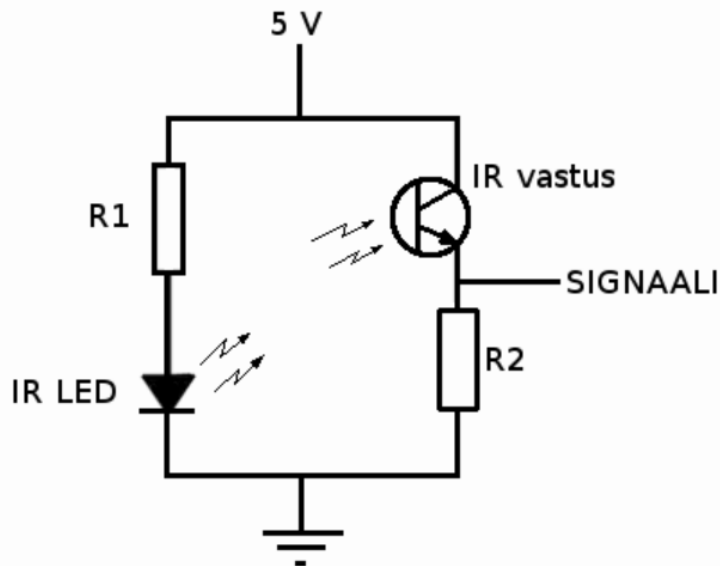
RC-auton nopeuden määrittämiseksi päätettiin käyttää vanteen sisäpuolelle pyörännävan yhteyteen sijoitettua infrapunasensoria. Sensori koostuu kahdesta puolijohdekomponentista, joista infrapunaledi säteilee infrapunavaloa ja infrapunavalolle herkkä valo-

vastus reagoi siihen osuvaan infrapunavaloon. Molemmat komponentit kytkettiin Arduinin 5 voltin jännitelähtöön. Lisäksi valovastus liitettiin jännitteenjako piirinä Arduinin digitaaliseen sisääntuloon. Koska komponentit käyttävät yhteisiä jännite- ja maajohtimia, pystyttiin liitäntä toteuttamaan samaa KK-sarjan kolminapaista pistoketta käyttäen kuin lineaarianturien kanssa.

Infrapunaledin kanssa käytetään 220  $\Omega$  etuvastusta, joka rajoittaa ledin kautta kulkevan virran alle 50 mA (KAAVA 1.).

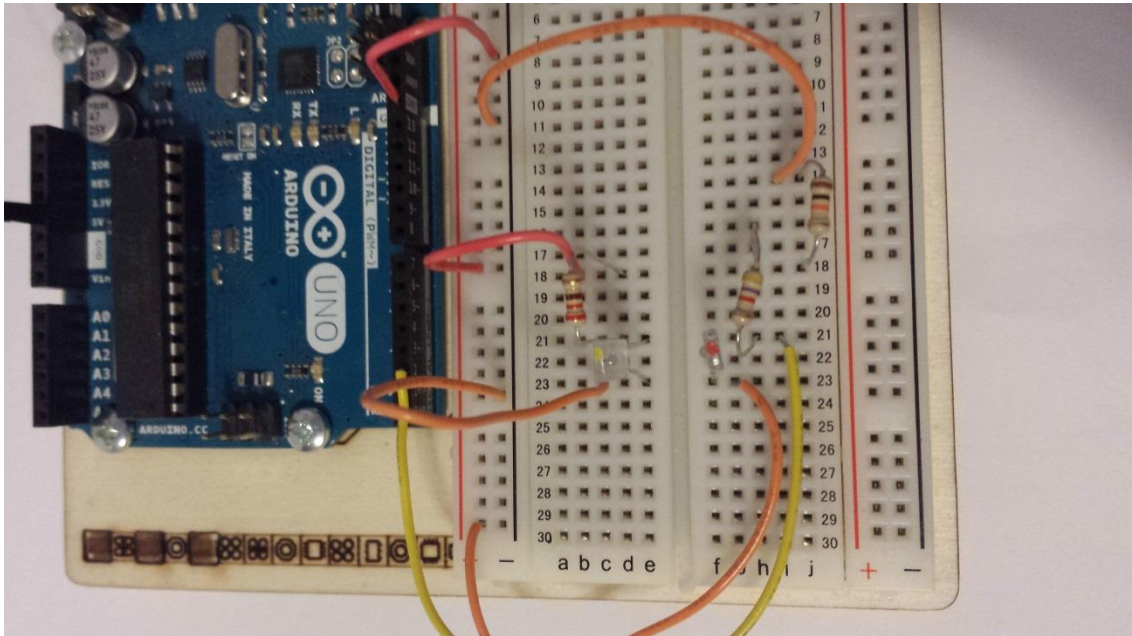
$$I = \frac{5V}{220\Omega} = 23\text{ mA} \quad (\text{KAAVA 1.})$$

Valovastus yhdessä 14,7 k $\Omega$  vastuksen kanssa muodostaa jännitteenjakopiirin, kun Arduinin I/O-pinnille vietään johdin vastusten välistä. Sensorin kytkentä kokonaisuudessaan on nähtävissä kuvassa 28.



KUVA 28. IR-sensorin kytkennät

Infrapunavalon heijastuessa valovastukseen kasvaa I/O-pinnillä näkyvä jännite. Valituilla vastuksilla pinnan jännite liikkui testauksessa (KUVA 29.) 0,5 voltin ja 4,5 voltin välillä, kun infrapunaledin ja valovastuksen välillä liikutettiin näköestettä. Mahdollinen ympäristön infrapunasäteily ja vanteen heijastuspinnalle tarttuva lika vaikuttaa jänniteväliin. Arduinin sisäisen logiikan kannalta merkittävä raja-arvo on 3,33 voltia, joten varmuus jännitevälin heilahduksille on riittävän suuri.



KUVA 29. IR-sensorin testaus

IR-sensorin tarvitsemat vastukset juotettiin johtimiin ja suojattiin kutistesukalla. Infra-punakomponenttien jännite- ja maajohtimet juotettiin yhteen ja myös nämä liitokset suojattiin kutistesukalla. Koska sensori sijaitsee kosteudelle alttiissa paikassa, päätettiin se suojata lakalla, jotta kosteus ei pääsisi vahingoittamaan sensoria kutistesukkien päähän mahdollisesti jääneistä suojaamattomilta alueilta. Valmis sensori kiinnitettynä paikoilleen on nähtävissä kuvassa 30.



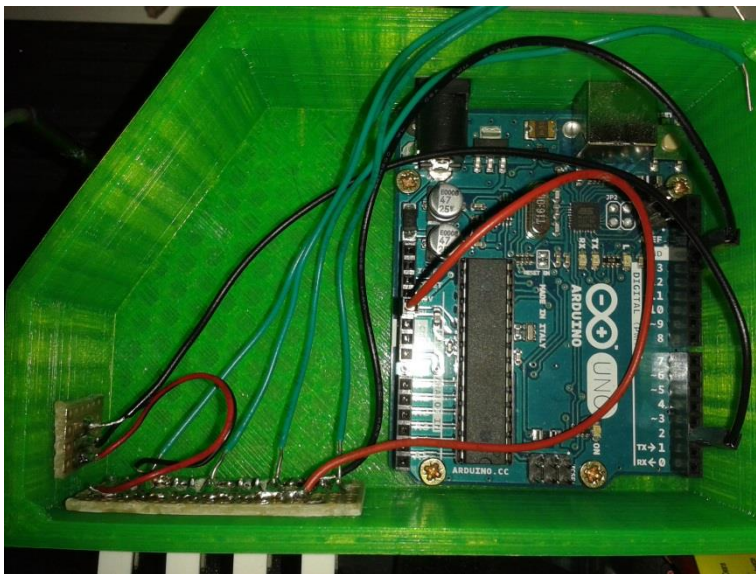
KUVA 30. IR-sensori asennettuna

Koteloon tehtiin tarvittavat muutokset IR-sensorin liitännöille. Kotelon etureunaan lisättiin aukko liittimelle (KUVA 31.), jona käytettiin samaa KK-sarjan liittintä kuin lineaariantureiden kanssa.



KUVA 31. IR-sensori liitettynä

Liitin sijoitettiin omalle kytkentälevylleen, johon 5 voltin jännite ja maa tuotiin hyppylangoilla lineaarianturien kytkentälevyltä (KUVA 32.). Kotelon tila saatiin hyödynnettyä, eikä sen mittoihin tarvinnut tehdä muutoksia.



KUVA 32. IR-sensorin kytkentälevy kotelossa.

## 6 OHJELMA

Tähän työhön ei sisällytetä kattavaa selostusta ohjelmoinnin perusteista tai Arduinon ohjelmoimisen erityispiirteistä. Tässä kappaleessa käydään läpi mittausten vaatiman ohjelman peruspiirteet sekä mahdolliset ongelmakohdat, siltä osin kuin niiden selvittäminen on katsottu aiheelliseksi.

Ohjelman rakennetta suunniteltaessa on hyvä miettiä millaisista osista lopullinen ohjelma koostuu. Mitä erillisiä toimintoja ohjelma sisältää ja mitä edellytyksiä kuhunkin vaiheeseen sisältyy. Mittauslaitteiston ohjelma koostuu kolmesta kokonaisuudesta; datan keräämisestä, sen käsittelystä haluttuun muotoon sekä käsitellyn datan esittämisestä käyttäjälle. Kun kukin kokonaisuus jaetaan vielä osiin, saadaan käytännöllisen kokoisia ohjelman pätkiä joiden työstäminen on selkeämpää kuin koko ohjelman kerralla. (Banzi, M. 2011.)

Mittausdataa kerätään neljältä lineaarianturilta, jotka on kytketty Arduinoon, kukin omaan Analog In -nastaansa. Arduino tunnistaa näiden nastojen jännitteen väliltä 0–5 voltia ja antaa nastalle vastaavan arvon välillä 0–1023. Kohdassa 5.3 esitettyä kaavaa voidaan käyttää suoraan anturia luettaessa.

$$\text{float Anturi1} = \text{analogRead}(0) * (20 / 1023);$$

Anturi1 on määritelty *float*-muuttujaksi, joka on desimaaliluku. Muuttujan arvo kertoo nyt suoraan anturin ulkoneman millimetreinä. Kaikki neljä anturia luetaan samalla tavalla ja jokaisen arvo tallennetaan omaan muuttujaansa. *Float*-muuttuja on kooltaan suuri suhteessa vain kokonaislukuja käyttäviin muuttujiin ja sitä käyttävät laskutoimitukset ovat paljon muistia vaativia.

Suoraan *Serial*-yhteyden ja usb-kaapelin välityksellä dataa siirrettäessä pystyttäisiin käyttämään *float*-muuttujia ja desimaalilukuja. Kun dataa halutaan lähettää langattomasti, voi datansiirtokyky nousta rajoittavaksi tekijäksi. Tällöin nastalta luettu arvo voidaan muuttaa myös `map()`-komennolla vastaamaan lineaarianturin mekaanista liikettä.

$$\text{map}(\text{analogRead}(0), 0, 1023, 0, 20)$$

Nyt ensimmäisestä analogisesta nastasta luettu arvo vastaa anturin liikettä (0–20 mm), mutta vain millimetrin tarkkuudella. Mittauslaitteiston esittämältä arvolta vaaditaan kuitenkin parempaa resoluutiota. Nastan lukema voidaan kartoittaa vastaamaan millimetrin kymmenes- tai sadasosia, ja käyttää kokonaislukuja näiden esittämiseen.

```
map(analogRead(0),0,1023,0,200) // 0–200 millimetrin kymmenesosaa
```

```
map(analogRead(0),0,1023,0,2000) // 0–2000 millimetrin sadasosaa
```

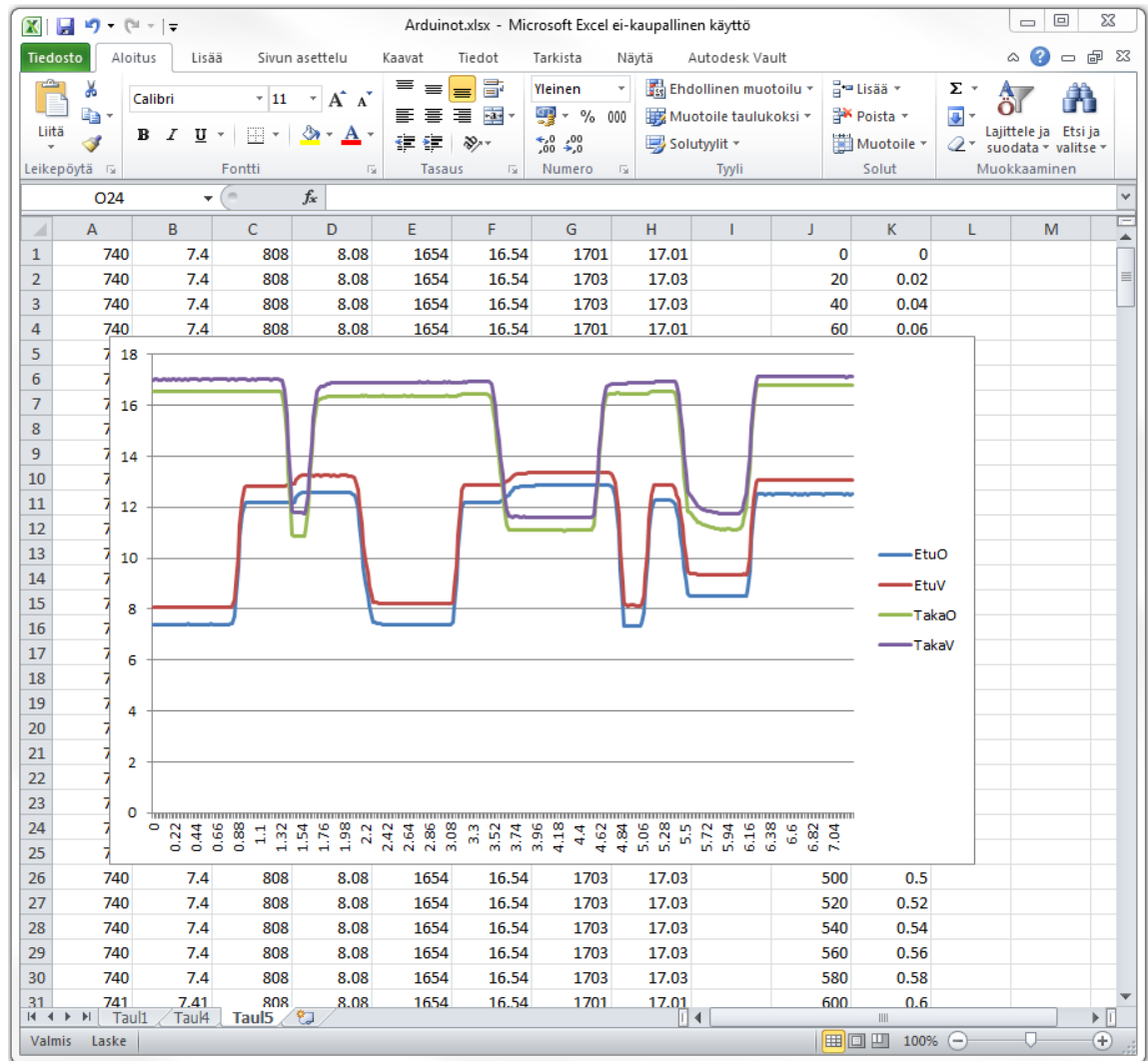
Jälkimmäisenä esitetty väli 0–2000 on tarkempi, mutta vaatii arvon tallentamisen *int* -muuttujaan. *Int* -muuttuja käyttää muistista kaksi tavua, kun *byte* -muuttuja, johon väli 0–200 mahtuu, tarvitsee nimensä mukaisesti yhden tavun. Mittausdataa reaaliajassa langattomasti siirrettäessä tulee tehdä kompromissi yhden sekunnin aikana lähetettävän datan määrän ja laadun välillä. (Banzi, M. 2011. Getting Started with Arduino. Make:Books)

Anturien arvot sisältävät muuttujat kirjoitetaan yksitellen sarjaporttiin *Serial.print()* -komennolla. Kunkin muuttujan välissä sarjaporttiin kirjoitetaan sarkain, jotta anturien arvot pystytään myöhemmin helposti erottamaan toisistaan. Rivin viimeistä muuttujaa kirjoitettaessa käytetään komentoa *Serial.println()*, joka sisältää rivinvaihdon.

```
Serial.print(Anturi1);  
Serial.print("\t");  
Serial.print(Anturi2);  
Serial.print("\t");  
Serial.print(Anturi3);  
Serial.print("\t");  
Serial.println(Anturi4);
```

Usb-liitännän kautta lähetettyä dataa pystytään tarkkailemaan sellaisenaan *Serial Monitorin* kautta, mutta datan tallentamiseksi esimerkiksi tekstitiedostoon tarvitaan ohjelma lukemaan sarjaporttia pitkin lähetettyä dataa. Processing ohjelmointiympäristössä luotiin Naveen Karuthedathin koodia käyttäen ohjelma, joka lukee sarjaporttia ennalta asetuin väliajoin. Lukemansa datan ohjelma tallentaa tekstitiedostoksi ohjelman sisällä

määritettyyn sijaintiin. Koska kunkin anturin arvo on erotettu toisistaan sarkaimella jo lähetettäessä, pystytään anturien lukemat siirtämään tästä tekstitiedostosta suoraan taulukkolaskentaohjelmaan jatkokäsittelyä varten. Koska ohjelma lukee anturien tiedot määritellyin väliajoin, voidaan iskunvaimentimien liike piirtää kuvaajana ajan suhteen. (KUVA 33.)



KUVA 33. Iskunvaimentajien liikkeen kuvaaja

Processing mahdollistaa ohjelmointikielenä myös graafiset esitykset. Sen avulla on mahdollista luoda mittauslaitteistolle graafinen käyttöliittymä, jonka kautta pystytään sekä esittämään mittausdataa, että antamaan komentoja laitteistolle. Tällaisen käyttöliittymän luontia ei kuitenkaan sisällytetty tähän työhön, mutta tämän mahdollisuuden hyödyntämistä jatkossa suositellaan. (Processing 2014 & Arduino 2014.)

## 6.1 Nopeuden määrittäminen

Nopeuden mittaamiseksi tulee ohjelman kyetä havaitsemaan renkaan jokainen kierros ja laskea kierrokseen kulunut aika. Kierrosten havaitsemiseen käytetään Arduinon digitaaliseen sisääntuloon D2 kytkettyä IR-sensoria. Arduino Unossa I/O-pinnit D2 ja D3 voivat toimia ISR-tilassa (Interrupt Service Routine), joka keskeyttää ohjelman ajon keskeytykseen liitetyn funktion suorittamisen ajaksi. Ohjelmassa pinni asetetaan ISR-tilaan *attachInterrupt()* -komennolla. Komennolla määritetään, kumpi toimintoon kykenevä I/O-pinni funktioon yhdistetään, mikä funktio toiminnossa ajetaan, sekä mikä neljästä eri toimintatilasta kytketään (TAULUKKO 3.).

TAULUKKO 3. Keskeytyksen toimintatilat

Toimintatila	Suoritetaan, kun
LOW	pinnin tila on LOW (jännite alle 3,3 V).
CHANGE	pinnin tila muuttuu (jännite siirtyy 3,3 V rajan toiselle puolelle).
RISING	pinnin tila muuttuu LOW:sta HIGH:ksi (jännite nousee 3,3 V rajan yli).
FALLING	pinnin tila muuttuu HIGH:sta LOW:ksi (jännite laskee 3,3 V rajan alle).

Keskeytyks on hyödyllinen silloin kun halutaan havaita kaikki jonkin sensorin tapahtumat ilman että yksittäistä sisääntuloa luetaan jatkuvasti. Tällöin laskentatehoa ja aikaa jää muiden toimintojen suorittamiseen. Kun keskeytykseen liitetty tapahtuma havaitaan, ajetaan yksinkertainen funktio kuten laskurimuuttujan arvon lisäys. Tällainen toimenpide on nopea suorittaa eikä häiritse ohjelman muuta toimintaa.

Nopeus lasketaan jakamalla renkaan kulkema matka siihen kuluneella ajalla. Keskeytyks-funktiolla kasvatetaan muuttujaa, jonka avulla määritellään koska nopeus lasketaan. Nopeus päätettiin laskea viiden kierroksen välein. Viidessä kierroksessa rc-auto on kulkenut 1,5 metrin matkan, olettaen ettei rengas ole luistanut ajoalustalla. Matkaan kuluneen ajan määrittämisessä käytetään hyväksi Arduinon sisäistä kelloa. Laskennan ajaksi keskeytykset otetaan pois käytöstä, jotta kaikki toisiinsa liittyvät toiminnot saadaan ajettua ongelmitta. Ohjelma on esitetty liitteessä 12.

## 7 POHDINTA

Työssä haluttiin toteuttaa mittauslaitteisto siten, että sen toimintaa ja rakennetta on helppo muuttaa tulevaisuudessa. Pyrkimyksenä oli lisäksi osoittaa, TAMKissa opetus-käytössäkin olevan, Arduino Uno:n antamat mahdollisuudet edullisten mutta tarvittaessa monimutkaisten laitteiden ja prototyyppien valmistuksessa. Vaikka tavoiteltu langattomuus jäikin saavuttamatta laitteiden vikaantuessa, voidaan muilta osin todeta nämä tavoitteet saavutetuiksi.

Instrumentointi itsessään on toimiva ja varsinkin iskunvaimentimien yhteydessä olevat lineaarianturit voidaan hyödyntää sellaisenaan myös mahdollisissa kaupallisiin järjestelmiin pohjautuvissa tulevissa versioissa. Valmistetut osat saatiin yhdistettyä olemassa oleviin rakenteisiin siten, että ne eivät estä rc-auton normaalia toimintaa.

Merkittävin jatkokehitys laitteistolle on toimivan langattoman tiedonsiirron luominen. Arduino antaa tähän hyviä mahdollisuuksia muun muassa WiFi Shieldin avulla. Työn-aikaisessa tutkimuksessa selvisi, että Arduino on mahdollista liittää jopa LabView-ohjelmistoon ulkoisena anturina. Tietojen vastaanotto ja niiden tallennus Processingilla luodulla ohjelmalla vaatii viitseliäisyyttä, eikä tarjoa samoja mahdollisuuksia mittausda-tan jälkikäsitteilyyn kuin kaupallinen ohjelmisto. Tämän lisäksi nopeuden mittaaminen ulkoisesta kiinteästä mittauspisteestä tulisi ottaa tarkasteltavaksi.

## LÄHTEET

Arduino. 2014. <http://arduino.cc/>.

Banzi, M. 2011. Getting Started with Arduino. 2. painos. Sebastopol: Make:Books. ISBN 978-1-449-309879.

Fraden, J. 2004. Handbook of Modern Sensors. Physics, Designs, and Applications. 3. painos. New York: Springer Science+Business Media. ISBN 0-387-00750-4.

Karvinen T. & K. 2009. Sulautetut. Opi rakentamaan robotteja ja muita sulautettuja järjestelmiä. Jyväskylä: Gummerus Kirjapaino Oy. ISBN 978-952-220-111-9.

National Instruments Corporation. 2014. <http://us.ni.com/academic/>. Luettu 31.1.2015.

Penny & Giles. 2012. Motorsport Brochure.  
<http://www.pennyandgiles.com/Products/Linear-Position-Sensors/Linear-Displacement-Sensor-SLS095.aspx>.

Processing. 2014. <https://processing.org/>.

Pulli, T. 2014. Anturit ja Arduino. Luento. ELEC-A4010 Sähköpaja. Aalto-yliopisto.

Liite 1. Arduino Dimensions and Hole Patterns

[http://www.adafruit.com/datasheets/arduino\\_hole\\_dimensions.pdf](http://www.adafruit.com/datasheets/arduino_hole_dimensions.pdf)

Arduino MEGA,  
MEGA 2560

0.05 [1.3]  
0.55 [14.0]  
2.10 [53.3]  
2.00 [50.8]  
4.00 [101.6]  
0.95 [24.1]  
0.60 [15.2]  
1.10 [27.9]  
0.20 [5.1]  
0.10 [2.5]

Arduino UNO,  
UNO SMD,  
Duemilanove,  
Diecimila, NG,  
most "full-size"  
clones.

0.05 [1.3]  
0.55 [14.0]  
2.10 [53.3]  
2.00 [50.8]  
2.70 [68.6]  
0.60 [15.2]  
1.10 [27.9]  
0.20 [5.1]  
0.10 [2.5]

This hole does not exist on pre-UNO boards

1 in. = [25.4mm]

**NOTES:**

**Dimensions:**  
Dimensions are given in inches, as well as [millimeters], in order to accommodate 98% of the planet.

**Arduino Hole Dimensions:**  
All Arduino mounting holes are 3.2mm – about 0.125in. – in diameter. They will accommodate M3-0.5 or UNC 4-40 screws and standoffs.

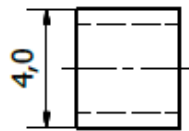
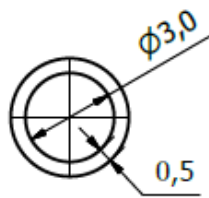
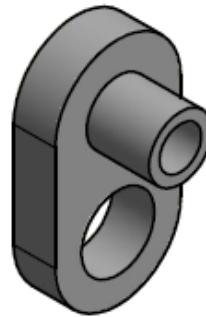
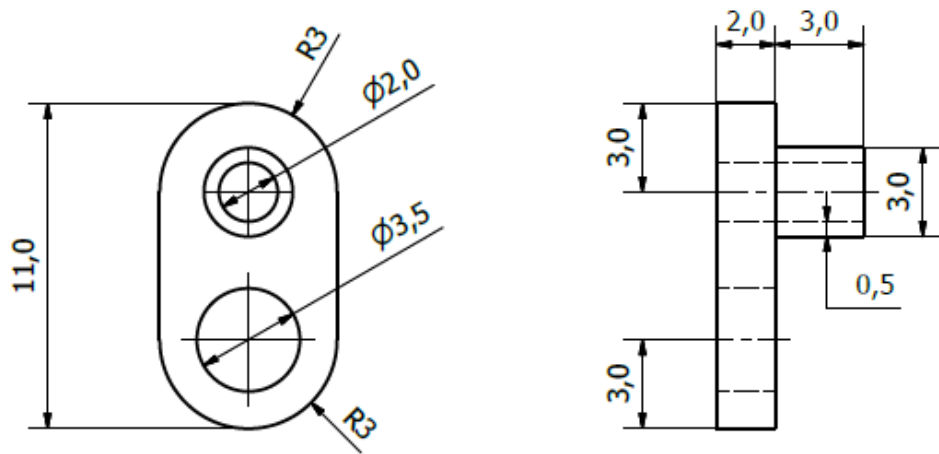
If you wish to tap threaded holes in a baseplate, you should use the following pre-tap drill sizes:  
M3-0.5    --    2.5mm  
4-40       --    #43 drill

**Using This Document:**  
You can use it simply as a reference to the physical dimensions or as a drilling guide. This document is drawn at a 1:1 scale, so you can print it out, tape it to the mounting surface and drill directly through the holes on the page. Use of a pre-drill punch is suggested.

**Printing:**  
To print at 1:1 scale, you should disable any "fit to page" or "page scaling" options. In Adobe Acrobat, select "Page Scaling: None" in the Print dialog box. Otherwise, the page will print about 6% smaller than full size.

TITLE: Arduino Dimensions and Hole Patterns		REVISION: 1-A
DESIGNED BY: J. M. DeCristoforo ("JohnGinger")	DATE: Feb. 24th, 2011	CC-BY-SA 3.0
SCALE: 1:1		

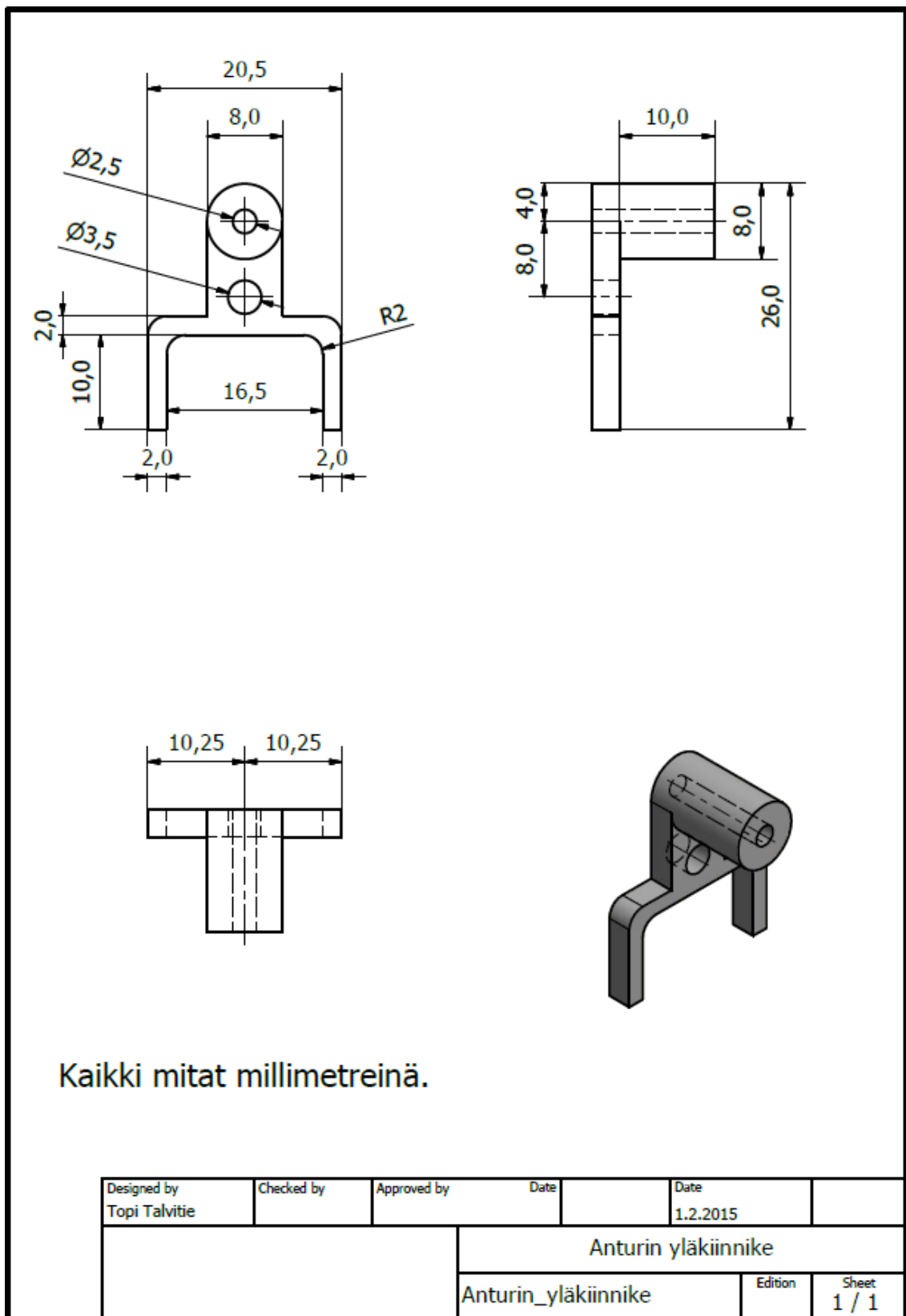
Liite 2. Anturin alakiinnike



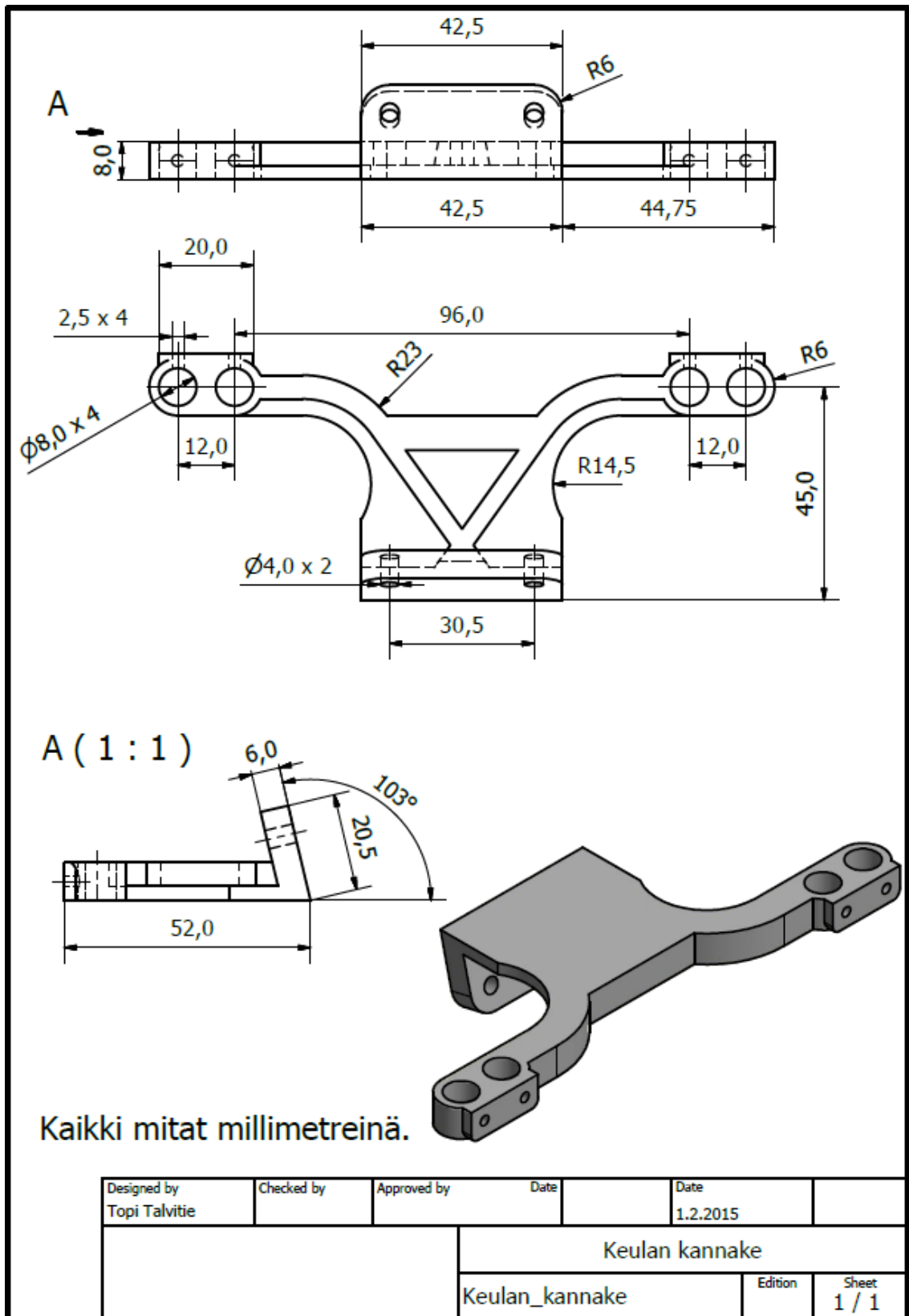
Kaikki mitat millimetreinä.

Designed by Topi Talvitie	Checked by	Approved by	Date	Date 1.2.2015	
			Anturin alakiinnike ja holkki		
			Anturi_alakiinnike	Edition	Sheet 1 / 1

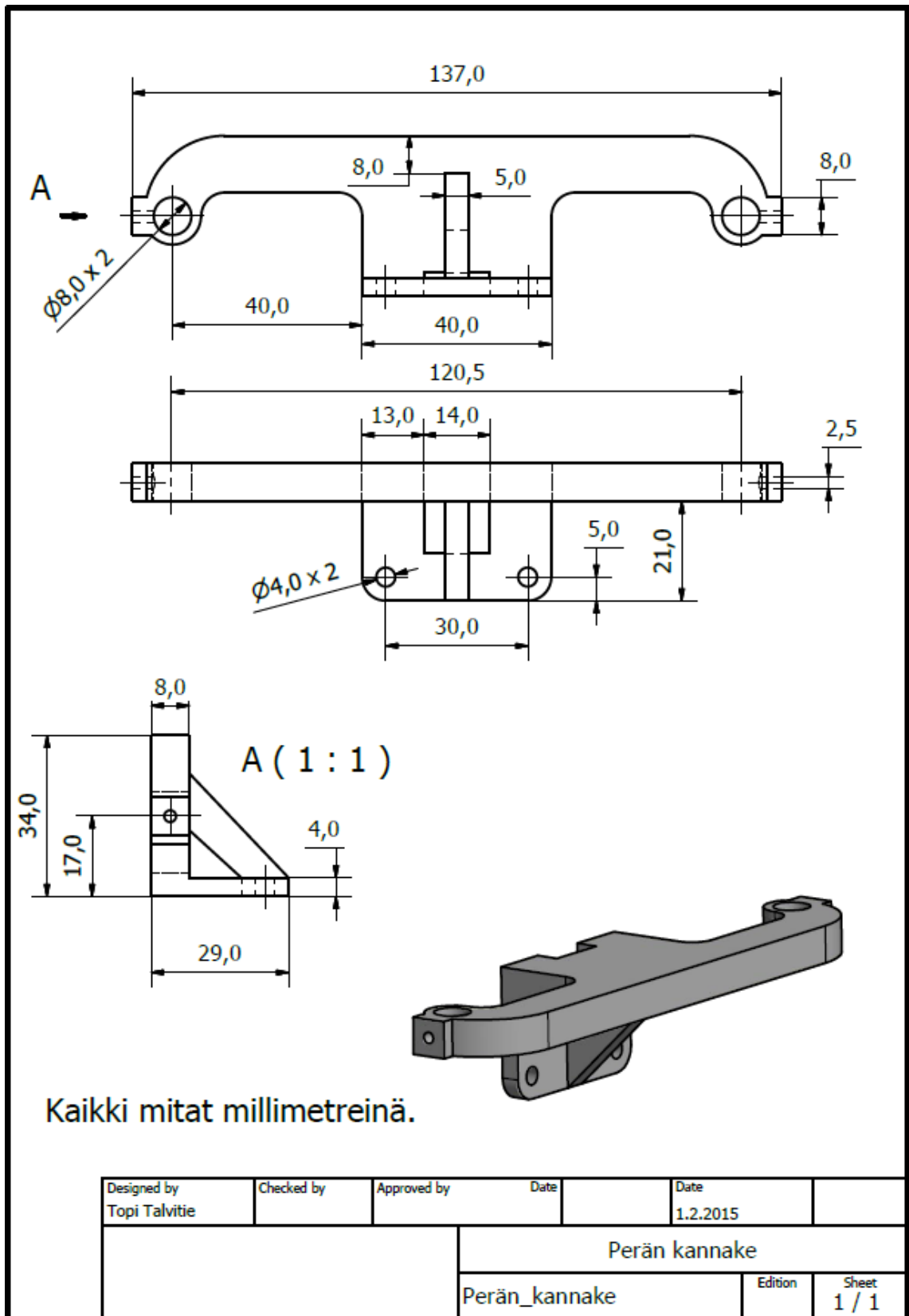
Liite 3. Anturin yläkiinnike



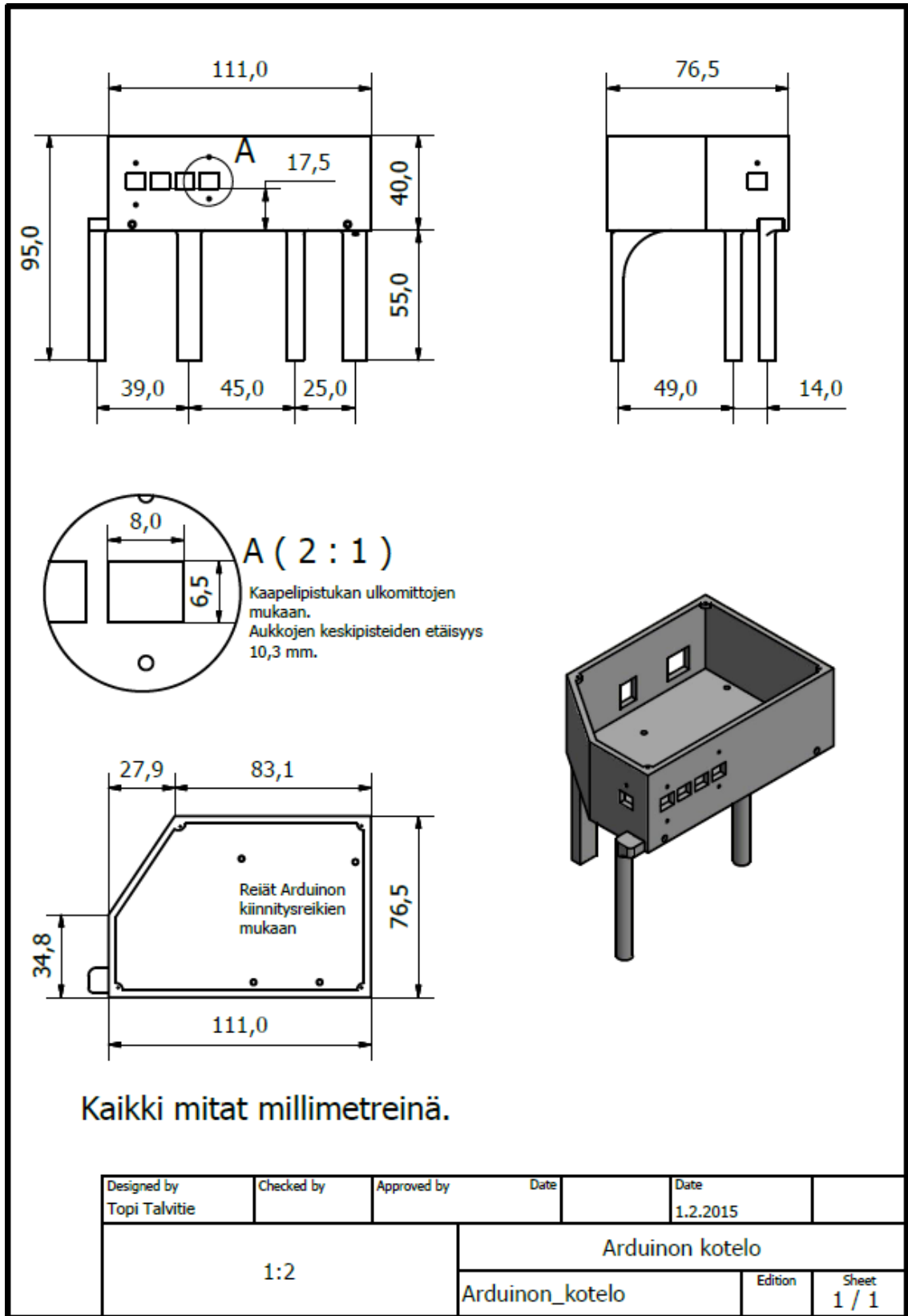
Liite 4. Keulan kannake



Liite 5. Perän kannake



Liite 6. Arduinon kotelo



<https://www.sparkfun.com/datasheets/Components/LTE-302.pdf>



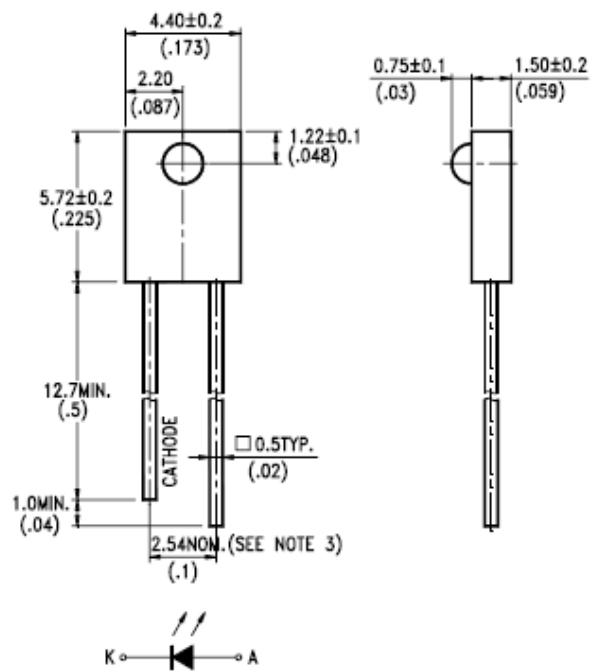
**LITE-ON ELECTRONICS, INC.**

Property of Lite-On Only

**FEATURES**

- \* SELECTED TO SPECIFIC ON-LINE INTENSITY AND RADIANT INTENSITY RANGES
- \* LOW COST MINIATURE PLASTIC SIDE LOOKING PACKAGE
- \* MECHANICALLY AND SPECTRALLY MATCHED TO THE LTR-301 SERIES OF PHOTOTRANSISTOR

**PACKAGE DIMENSIONS**



**NOTES:**

1. All dimensions are in millimeters (inches).
2. Tolerance is  $\pm 0.25\text{mm}(.010\text{'})$  unless otherwise noted.
3. Lead spacing is measured where the leads emerge from the package.
4. Specifications are subject to change without notice.


**ABSOLUTE MAXIMUM RATINGS AT TA=25°C**

PARAMETER	MAXIMUM RATING	UNIT
Power Dissipation	75	mW
Peak Forward Current (300pps, 10 $\mu$ s pulse)	1	A
Continuous Forward Current	50	mA
Reverse Voltage	5	V
Operating Temperature Range	-40°C to +85°C	
Storage Temperature Range	-55°C to +100°C	
Lead Soldering Temperature [1.6mm(.063") From Body]	260°C for 5 Seconds	

**ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25°C**

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITION	BIN NO.
Aperture Radiant Incidence	Ee	0.088		0.168	mW/cm <sup>2</sup>	I <sub>f</sub> = 20mA	BIN B
		0.112		0.204			BIN C
		0.136		0.240			BIN D
		0.160		0.288			BIN E
		0.192					BIN F
Radiant Intensity	I <sub>e</sub>	0.662		1.263	mW/sr	I <sub>f</sub> = 20mA	BIN B
		0.842		1.534			BIN C
		1.023		1.805			BIN D
		1.203		2.165			BIN E
		1.444					BIN F
Peak Emission Wavelength	$\lambda_{Peak}$		940		nm	I <sub>f</sub> = 20mA	
Spectral Line Half-Width	$\Delta \lambda$		50		nm	I <sub>f</sub> = 20mA	
Forward Voltage	V <sub>F</sub>		1.2	1.6	V	I <sub>f</sub> = 20mA	
Reverse Current	I <sub>R</sub>			100	$\mu$ A	V <sub>R</sub> = 5V	
Viewing Angle (See FIG.6)	2 $\theta_{1/2}$		40		deg.		



TYPICAL ELECTRICAL / OPTICAL CHARACTERISTICS CURVES

(25°C Ambient Temperature Unless Otherwise Noted)

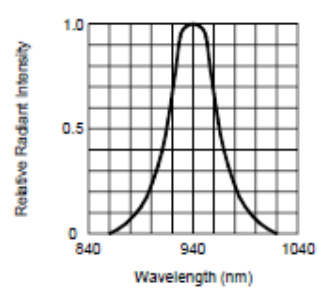


FIG.1 SPECTRAL DISTRIBUTION

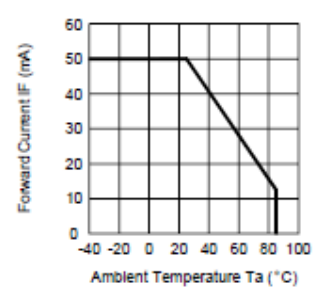


FIG.2 FORWARD CURRENT VS. AMBIENT TEMPERATURE

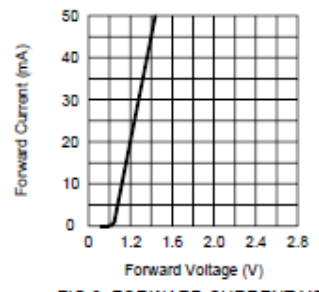


FIG.3 FORWARD CURRENT VS. FORWARD VOLTAGE

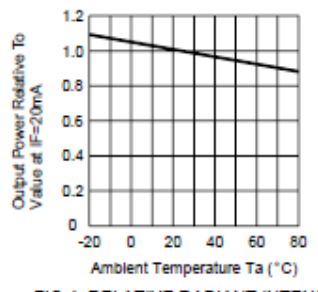


FIG.4 RELATIVE RADIANT INTENSITY VS. AMBIENT TEMPERATURE

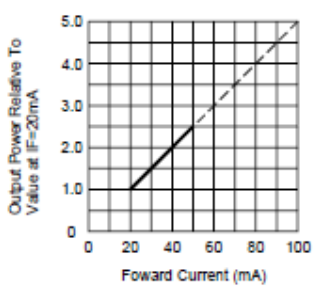


FIG.5 RELATIVE RADIANT INTENSITY VS. FORWARD CURRENT

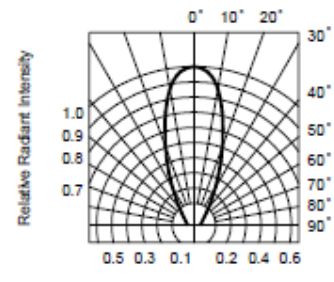


FIG.6 RADIATION DIAGRAM

<https://www.sparkfun.com/datasheets/Components/LTR-301.pdf>



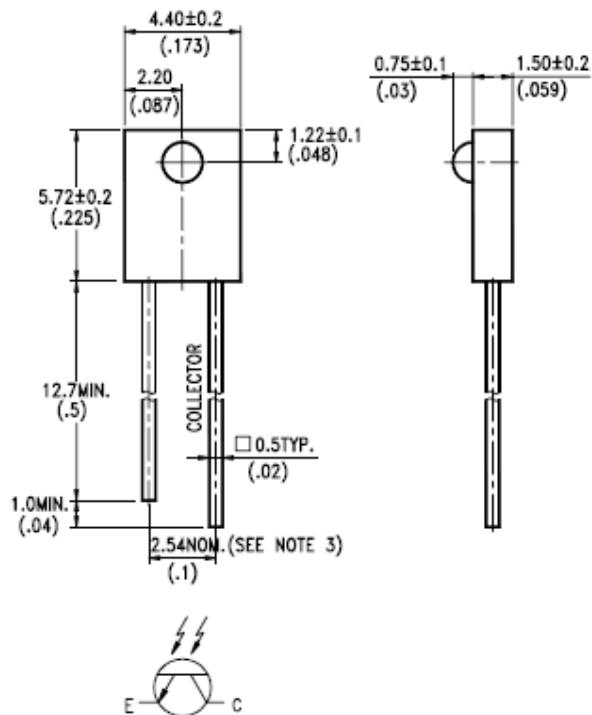
**LITE-ON ELECTRONICS, INC.**

Property of Lite-On Only

**FEATURES**

- \* WIDE RANGE OF COLLECTOR CURRENT
- \* LENSED FOR HIGH SENSITIVITY
- \* LOW COST PLASTIC SIDE LOOKING PACKAGE
- \* CLEAR TRANSPARENT COLOR PACKAGE

**PACKAGE DIMENSIONS**



**NOTES:**

1. All dimensions are in millimeters (inches).
2. Tolerance is  $\pm 0.25$ mm(.010") unless otherwise noted.
3. Lead spacing is measured where the leads emerge from the package.
4. Specifications are subject to change without notice.


**ABSOLUTE MAXIMUM RATINGS AT TA=25°C**

PARAMETER	MAXIMUM RATING	UNIT
Power Dissipation	100	mW
Collector-Emitter Voltage	30	V
Emitter-Collector Voltage	5	V
Operating Temperature Range	-40°C to +85°C	
Storage Temperature Range	-55°C to +100°C	
Lead Soldering Temperature [1.6mm(.063") From Body]	260°C for 5 Seconds	

**ELECTRICAL OPTICAL CHARACTERISTICS AT TA=25°C**

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITION	BIN NO.
Collector-Emitter Breakdown Voltage	$V_{(BR)CEO}$	30			V	$I_C = 1mA$ $E_e = 0mW/cm^2$	
Emitter-Collector Breakdown Voltage	$V_{(BR)ECO}$	5			V	$I_E = 100 \mu A$ $E_e = 0mW/cm^2$	
Collector Emitter Saturation Voltage	$V_{CE(SAT)}$			0.4	V	$I_C = 0.1mA$ $E_e = 1mW/cm^2$	
Rise Time	$T_r$		10		$\mu s$	$V_{CC} = 5V$ $I_C = 1mA$ $R_L = 1K\Omega$	
Fall Time	$T_f$		15		$\mu s$		
Collector Dark Current	$I_{CEO}$			100	nA	$V_{CE} = 10V$ $E_e = 0mW/cm^2$	
On State Collector Current	$I_{C(ON)}$	0.20		0.60	mA	$V_{CE} = 5V$ $E_e = 1mW/cm^2$ $\lambda = 940nm$	BIN A
		0.40		1.08			BIN B
		0.72		1.56			BIN C
		1.04		1.80			BIN D
		1.20		2.40			BIN E
		1.60					BIN F



TYPICAL ELECTRICAL / OPTICAL CHARACTERISTICS CURVES

(25°C Ambient Temperature Unless Otherwise Noted)

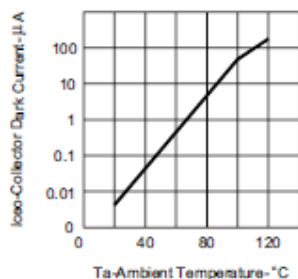


FIG.1 COLLECTOR DARK CURRENT VS AMBIENT TEMPERATURE

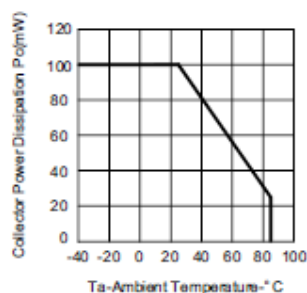


FIG.2 COLLECTOR POWER DISSIPATION VS AMBIENT TEMPERATURE

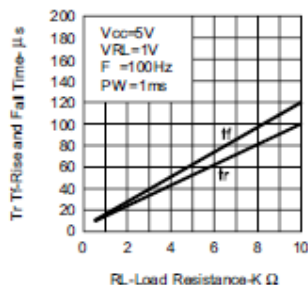


FIG.3 RISE AND FALL TIME VS LOAD RESISTANCE

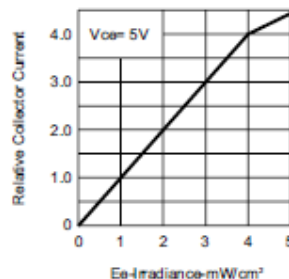


FIG.4 RELATIVE COLLECTOR CURRENT VS IRRADIANCE

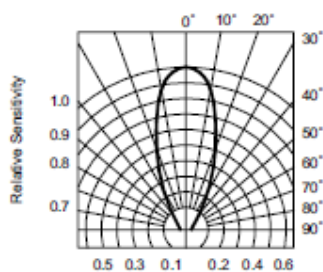
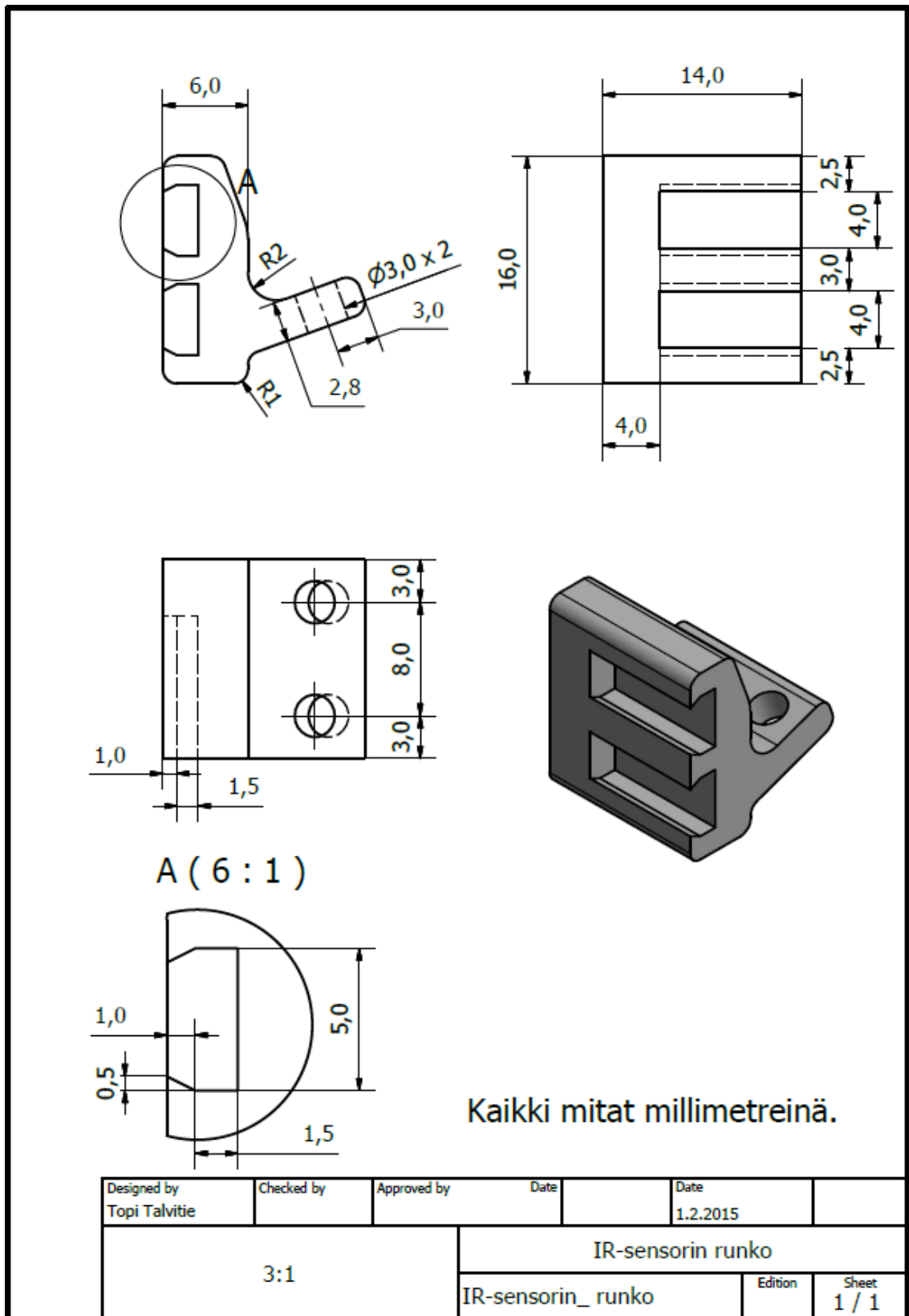


FIG.5 SENSITIVITY DIAGRAM

Liite 10. IR-sensorin runko



# SLS095 LINEAR DISPLACEMENT SENSOR

SLS095 is designed to provide maximum performance benefits within an extremely compact body diameter of 9.5mm, with stroke lengths from 10 to 100mm. The miniature size of this sensor makes it ideal for applications in robotics, animatronics, medical equipment and motorsport data acquisition.

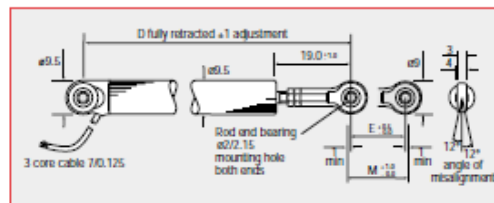
<b>PERFORMANCE</b>								
<b>Electrical stroke E</b>	<b>mm</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>	<b>75</b>	<b>100</b>
<b>Resistance ±10%</b>	<b>kΩ</b>	0.4†	0.8	1.2	1.6	2.0	3.0	4.0 †±15% for SLS 095/10
<b>Independent linearity</b>	<b>±%</b>	0.5	0.35	0.25	0.25	0.25	0.15	0.15
<b>Power dissipation at 20°C</b>	<b>W</b>	0.2	0.4	0.6	0.8	1.0	1.5	2.0
<b>Applied voltage maximum</b>	<b>Vdc</b>	8.9	17.9	26	40	44	67	74
<b>Resolution</b>		Virtually infinite						
<b>Hysteresis (repeatability)</b>		Less than 0.01mm						
<b>Operational temperature</b>	<b>°C</b>	-30 to +100						
<b>Output smoothness</b>		To MIL-R-39023 grade C 0.1%						
<b>Insulation resistance</b>		Greater than 100MΩ at 500Vdc						
<b>Operating mode</b>		Voltage divider only - see Circuit Recommendation below						
<b>Wiper circuit impedance</b>		Minimum of 100 x track resistance or 0.5MΩ (whichever is greater)						
<b>Operating force maximum</b>		300 in horizontal plane						
<b>sealed</b>	<b>gf</b>	100 in horizontal plane						
<b>unsealed</b>	<b>gf</b>	Typically greater than 100 million operations (50 x 10 <sup>6</sup> cycles) at 25mm stroke length						
<b>Life at 250mm per second</b>		200 million operations (100 x 10 <sup>6</sup> cycles) at ±0.5mm, 60Hz						
<b>Dither life</b>		IP50 standard - IP66 see options						
<b>Sealing</b>		20 million operations (10 x 10 <sup>6</sup> cycles)						
<b>Shaft seal life</b>		2.5						
<b>Shaft velocity maximum</b>	<b>m/s</b>	RTCA 160D 10Hz to 2kHz (random) @ 4.12g (rms) - all axes						
<b>Vibration</b>		40g 6mS half sine						
<b>Shock</b>								
<b>CIRCUIT RECOMMENDATION</b>		Hybrid track potentiometers feature a high wiper contact resistance, therefore operational checks should be carried out only in the voltage divider mode. Hybrid track potentiometers should be used only as voltage dividers, with a minimum wiper circuit impedance of 100 x track resistance or 0.5MΩ (whichever is greater). Operation with wiper circuits of lower impedance will degrade the output smoothness and affect the linearity.						
<b>OPTIONS</b>								
<b>IP 66 sealing</b>		Designed to accept integral shaft seal to give IP66 rating						
<b>Mounting</b>		Can be supplied with self aligning bearings or a plain body for use with body clamps or flange mounting kit.						
<b>ACCESSORIES</b>		Mounting kits ———— Body clamp kit - SA200841 Flange kit - SA200842						
<b>AVAILABILITY</b>		All standard configurations can be supplied rapidly from the factory - check with your local supplier for more details						
<b>ORDERING CODES</b>		<p style="text-align: center;">SLS095/...../...../...../.....</p> <p>Electrical stroke ———— Resistance ———— Sealing 50 = IP50, 66 = IP66                  Mounting option R = Self aligning bearing                  P = Plain</p>						

**4**

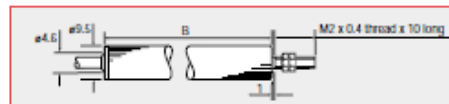
## DIMENSIONS AND MOUNTING OPTIONS

Note: drawings not to scale

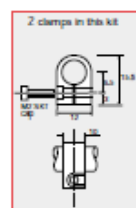
### SELF ALIGNING BEARING MOUNTING



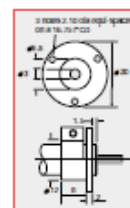
### PLAIN BODY MOUNTING



### MOUNTING OPTIONS



**Body clamp**  
SA200841

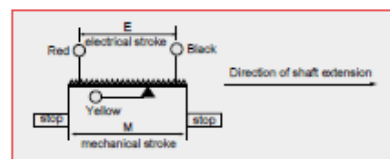


**Flange mounting**  
SA200842

<b>Electrical stroke E</b>	<b>mm</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>	<b>75</b>	<b>100</b>
<b>Mechanical stroke M</b>	<b>mm</b>	<b>12.5</b>	<b>22.5</b>	<b>32.5</b>	<b>42.5</b>	<b>52.5</b>	<b>77.5</b>	<b>102.5</b>
<b>Body length B</b>	<b>mm</b>	45.5	55.5	65.5	75.5	85.5	110.5	135.5
<b>Between centres D</b>		70	80	90	100	110	135	160
<b>Weight approximate (mounting option R)</b>	<b>g</b>	11	13	14.5	16	17.5	21.5	25.5

### ELECTRICAL CONNECTIONS

3 core cable: PUR sheathed 0.3m long with PTFE insulated 7/0.125 cores.



## Liite 12. Arduino ohjelma

```
volatile int luku = 0; //muuttuja johon renkaan pyörähdykset tallennetaan

int matka = 1500; //Auton kulkema matka millimetreinä,
//kun renkaat ovat pyörähtäneet if-lauseeseen asetetut kierrokset.
unsigned long kello = 0; //Tähän tallennetaan millis() lukema seuraavaa laskutoimitusta varten

unsigned long aika = 0; //Nopeuden laskennassa käytettävä kierrokseen kulunut aika

unsigned long nopeus = 0; //Muuttuja johon nopeus tallennetaan

void setup(){

  Serial.begin(9600);
  attachInterrupt(0, laskuri, RISING); //Liitetään digitaalinen pinni 2 (Interrupt 0) keskeytykseen
//ja käsketään ajaa funktio "laskuri" kun pinnin jännite nousee
//rajan yli.

  Serial.println("Etu0 \t EtuV \t Taka0 \t TakaV \t Nopeus"); //tulostetaan otsikkorivi
}

void loop()
{

  int Etu0 = map(analogRead(0),0,1023,2000,0); //luetaan anturit
  int EtuV = map(analogRead(1),0,1023,2000,0); //0-2000 sadasosamillia
  int Taka0 = map(analogRead(2),0,1023,2000,0);
  int TakaV = map(analogRead(3),0,1023,2000,0);

  if(luku == 5){ //Ehtoon asetettava luku on kierrosmäärä jonka mukaan nopeus lasketaan.
//Tämän tulee vastata matka-muuttujaan asetettua matkaa.
//Nopeuden laskenta ja seuraavaa laskentaa varten tehtävät nollaukset

    noInterrupts(); //Estetään keskeytykset laskennan aikana

    aika = millis() - kello; //Kierrokseen kulunut aika millisekunteina

    nopeus = matka / aika; //Nopeus m/s, laskennassa millimetrit jaetaan millisekunneilla

    kello = millis(); //Tallennetaan aika
    luku = 0; //ja nollataan laskurin luku.

    interrupts(); //Otetaan keskeytykset jälleen käyttöön
  }

  Serial.print(Etu0); //kirjoitetaan sarjaporttiin kunkin anturin lukema
  Serial.print("\t");
  Serial.print(EtuV);
  Serial.print("\t");
  Serial.print(Taka0);
  Serial.print("\t");
  Serial.print(TakaV);
  Serial.println(nopeus); //sekä nopeus

  delay(10); //odotetaan 10 millisekuntia. tätä säätämällä saadaan useampi
//otos sekunnissa
}

void laskuri() //Keskeytyksessä ajettava funktio
{
  luku = ++luku; //Lisätään lukuun yksi
}
```