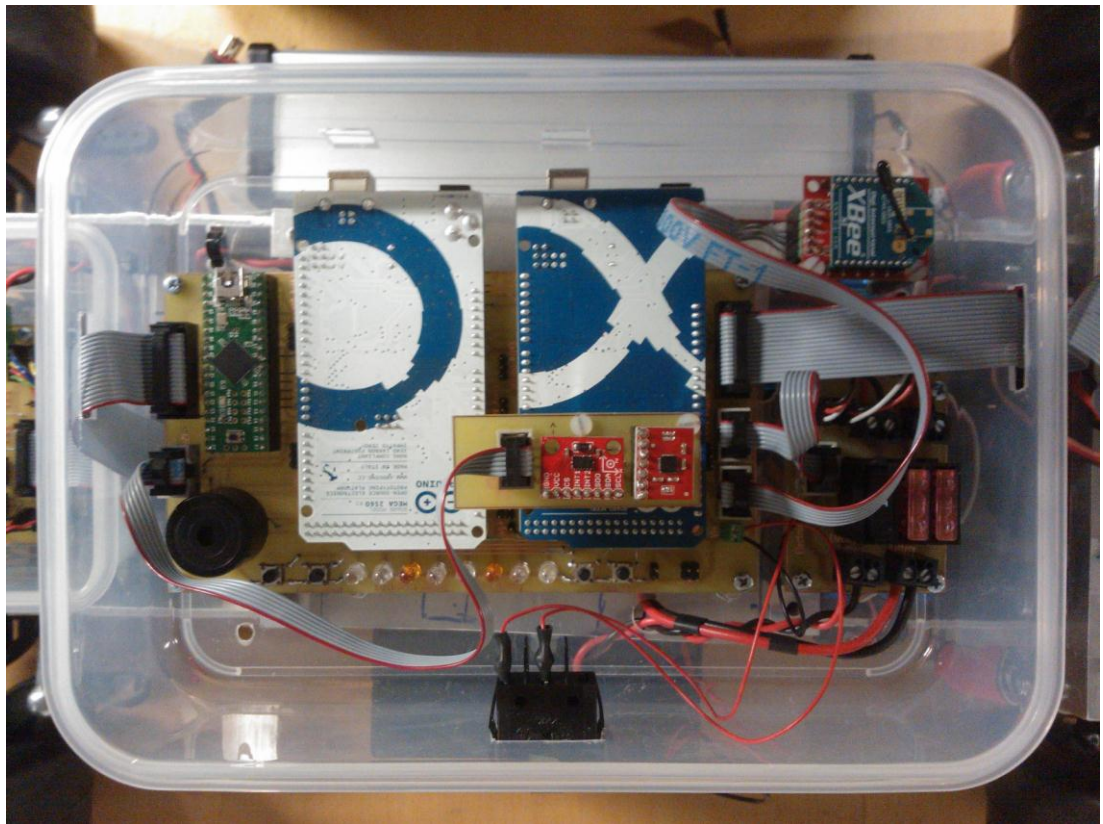


Liite: 5

Mekatroniikka



Markus Storsjö 69604J

Palautettu 21.12.2012

Sisällysluettelo

Sisällysluettelo	2
1. Lähtökohdat	3
1.1. Komponenttien valinta.....	3
1.2. Testipenkki.....	5
1.3. Tehtävänjako	5
2. Mekatroniikan rakenne	6
2.1. Elektroniikan layout.....	6
2.2. Sähköjärjestelmän layout	7
3. Piirilevyt	9
3.1. Virranjakolevy	9
3.2. Pääelektroniikkalevy	10
3.3. I ² C-levy	11
3.4. Akselimoduulilevyt.....	11
3.5. Muutoksia kytkentäkaavioihin.....	12
3.6. Pinmapping	13
4. Koodia.....	16
4.1. Ohjelman rakenne.....	17
4.2. Kommunikaatio	17
4.2.1. Protokolla.....	17
4.2.2. Protokollan speksit.....	18
5. Yhteenveto.....	21

5.1. Tulokset	21
5.2. Tunnetut ongelmat	21
5.3. Jatkokehitystä	21

1. Lähtökohdat

Tavoitteena oli jatkokehittää viime vuoden RC-autoa siten, että siitä voisi saada mitausdataa tallennettua, ja että autoon olisi mahdollista implementoida ABS- ja ESP-toiminnallisuutta. Parannettavina kohtina oli vielä kommunikaatio ja johdatus.

1.1. Komponenttien valinta

Tarkoituksena oli pääosin käyttää viime vuoden auton komponentteja. Uusina komponentteina piti kuitenkin hankkia kiihtyvyysanturi ja gyro. Vuoden 2011 RC-auton langaton kauko-ohjaus perustui Bluetoothiin. Suurin ongelma siinä on kantama, joka on kymmenien metrien luokkaa. Pääteltiin siksi hankkia uuden auton tiedonsiirrolle xBee-moduulit, joiden kantama on noin 100 m. xBee on suosittu tekniikka tällaiseen käyttötarkoitukseen, ja jos tulevaisuudessa ilmenisi tarvetta vielä suuremmalle kantamalle, löytyy kalliimpiakin moduuleja, joilla on suurempi kantama.

Mietittiin aluksi myös millä elektroniikalla autoa kannattaisi ohjata. Tiesimme, että tavallisten mikrokontrollereiden laskentateho ei riitä oikeiden ABS- ja ESP-algoritmien pyörittämiseen. Tehokkaammat laitteet kuitenkin ylittäisivät meidän budjettia reilusti, joten päädyttiin käyttää mikrokontrollereita. Käytettäviä algoritmeja pitäisi vain yksinkertistaa sen verran, että mikrokontrollerit pystyvät käsittelemään niitä. Mikrokontrollerivalinta päättyi Arduinoon, osittain koska viime vuoden autosta oli yksi Arduino Mega 2560 käytettävissä, mutta myös ennen kaikkea siksi, että Arduino-ohjelmointiin on aloittelijoidenkin helppo tutustua, joten tulevien vuosien jatkokehitystä silmälläpitäen Arduino olisi hyvä valinta.

Viime vuoden autossa oli Arduinon lisäksi yksi Teensy 2.0++. Arveltiin, että uuden autojen uusien lisäominaisuuksien takia, siihen tarvittaisiin vielä jokin kontrolleri, laskentatehoa lisäämään. Kontrollereiden määrä ja tehtävänjako ei ollut itsestään selvä, mutta ratkaisu joka valittiin oli se, että autossa olisi ”pääkontrollerina” yksi Arduino 2560, joka kommunikoisi tietokoneen kanssa, ja ohjaisi suurimman osan toiminnoista. Toinen Arduino 2560 toimisi ”jarrukontrollerina”, joka ottaisi sisäänsä anturitie-

toja sekä jarrukäskyjä, ja näiden perusteella laskisi ABS- tai ESP-algoritmeilla, miten jokaista pyörää kannattaisi jarruttaa. Teensylle jäi nopeusmittarin rooli. Korkeissa nopeuksissa voisi muuten olla riskinä, että enkooderipulssien käsittely veisi liikaa laskentatehoa itse pääohjelmasta. Tämä ongelma vältetään, kun pulssien käsittelylle on oma kontrolleri, joka ei tee muuta, ja joka lähettää valmiin nopeusdatan muille kontrollereille.

Huomattiin että viime vuoden autossa käytetyt harjalliset DC-moottorit eivät vaikuttaneet täysin terveiltä. Niitä ei juurikaan saatu liikahtamaan, kun virtalähteellä yritettiin pyörittää niitä 5 A virralla. Tilattiin siksi uudet moottorit, vuonna 2011 speksatut harjattomat DC-moottorit.

Koska DC-moottorit vievät suuret virrat, on varmintä kytkeä elektroniikka omaan akkuun, etteivät moottoreiden virtapiikit häiritse elektroniikan toimintaa. Moottorit on tarkoitettu 7,4 V jännitteelle, mikä saadaan 2-kennoisesta LiPo-akusta. Arduinolle kelpaa 7-12 V, mutta sen 5 V regulaattori on vain stabiili, jos sen sisääntulojännite on suurempi kuin 7 V. Valittiin siksi varmuuden vuoksi elektroniikan akun jännitteeksi 11,1 V (3 LiPo-kennoa). Servot puolestaan tarvitsevat 6 V. Tämä jännite voidaan saada regulaattoreiden kautta, joko 11,1 V:sta tai 7,4 V:sta. Pitkän etsinnän jälkeen löytyi regulaattori, joka pystyy laskemaan jännitteen 7,4 V:sta 6 V:iin. Tämä mahdollisti sen, että auton kaikki suuret virrat (moottorit ja servot), voidaan ottaa samasta akusta. Silloin akuissa on selkeä tehtäväjako, toinen on iso ”voimasähkön” akku, ja toinen on pieni, ”elektroniikkasähkön” akku. Elektroniikka-akkuun kelpaa silloin pienemmät kennot, joilla on pienempi purkautumisvirta.

Valittujen regulaattoreiden haittapuoli on suurin sallittu virta. 6 V regulaattorit eivät pysty antamaan enemmän kuin 1 A virtaa, mikä jarruservoille riittää, mutta ei isolle ohjausservolle, joka otettiin vuoden 2011 RC-autosta. Tilannetta ei helpottanut, että ohjausmekanismi oli ensimmäisen asennuksen jälkeen erittäin jäykkä. Ohjausservon ongelma pyrittiin ratkaisemaan siten, että ohjausservon regulaattori poistettiin, ja tilalle laitettiin 3 A sulake, jolloin servo saisi suoraan 7,4 V. Tämä toimi hetken, vaikka sulakkeen elinikä oli lyhyt. Lopulta servo kärähti lopullisesti, jolloin hankittiin tilalle 7,4 V:lla toimiva servo. Tämän jälkeen ei enää ollut ongelmia ohjausservon kanssa.

Taulukossa 1 näkyy kaikki autossa käytetyt elektroniikkakomponentit.

Taulukko 1. Autossa käytetyt komponentit

Pääkontrolleri, jarrukontrolleri	Arduino Mega 2560
Enkoodereita lukeva kontrolleri	Teensy 2.0++
Ohjausservo	Savöx SV-0235MG
Jarruservo, kytkinservo	Springrc SM-S4306B
Enkooderi	CUI AMT 102-V
Moottorihjain	LRP80450 SPX Brushless Bullet Reverse
Moottori	LRP50430 LRP Vector K4 Brushless Motor 6.5T
xBee-moduuli	XBee 1mW Wire Antenna - Series 1
Kiihtyvyysanturi	ADXL345
Gyro	ITG-3200
6 V regulaattori	STMICROELECTRONICS - LF60CV

1.2. Testipenkki

Keväällä rakennettiin mekatroniikalle testipenkki. Käytännössä tämä tarkoitti ison plexilevyn, johon kaikki komponentit kiinnitettiin. Alkuperäisen tarkoituksena oli tässä testipenkissä liittää kaikki komponentit yhteen, ja varmistaa koko järjestelmän toimivuus ennen kuin auton lopullista sähköjärjestelmää aletaan suunnitella, eli ”Hardware in the Loop”-menetelmän soveltamista.

Todettiin kuitenkin, että kaikkien tarvittavien kytkentöjen tekeminen hyppylangoilla on haastavaa ja aikaa vievää, sillä kytkentöjä on paljon. Sen lisäksi riski on suuri että langat irtoavat tai kytketään väärin, joten lopuksi testipenkkiä ei käytetty muuhun kuin komponenttien testaamiseen yksitellen.

1.3. Tehtävänjako

Viime vuoden RC-autoprojektin projektipäällikkö, Otso Saarentaus (AS), halusi myös osallistua uuden auton kehittämiseen 3 opintopisteen suorituksella. Otson teh-

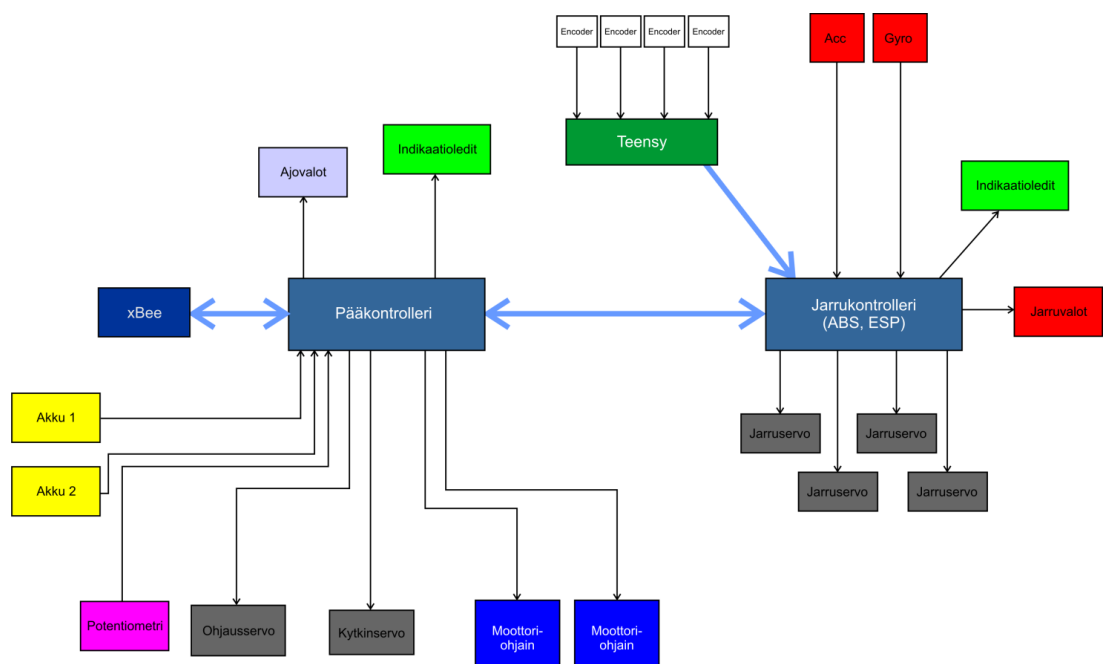
täväksi valittiin ABS- ja ESP-algoritmien implementointi, eli käytännössä jarrukontrollerin ohjelmointi.

Autoa ohjataan sarjaporttidataa tietokoneelta. Käytännössä tämä vaatii jonkinlaisen graafisen käyttöliittymän. Käyttöliittymän kehittäminen todettiin liian työllistäväksi al-lekirjoittaneelle tai Otsolle, joten tämä tehtävä annettiin toiselle AS-opsikelijalle, Val-ter Sandströmille. Valter valitsi käyttöliittymän perustaksi Matlabin, koska hänellä oli enemmän kokemusta Matlabista kuin esim. Labview:stä, ja Labview:n käyttö on hu-huttu olevan hankalaa.

2. Mekatroniikan rakenne

2.1. Elektroniikan layout

Kuvassa 1 näkyy auton komponentit, ja niiden välinen kommunikointi. Teensy lukee siis enkoodereita, ja välittää tiedon eteenpäin jarrukontrollerille. Jarrukontrolleri lu-kee gyron ja kiihtyvyysanturin I²C-väylän kautta, sekä vastaanottaa jarrutuskäskyjä pääkontrollerilta, ja näiden perusteella ohjaa sitten neljä jarruservoa. Pääkontrolleri vastaanottaa enkooderi-, kiihtyvyys- ja gyrodatan jarrukontrollerilta, ja edelleenvälit-tää sen tietokoneelle. Pääkontrollerin vastuulla on myös ohjata kaikki muut toiminnot autossa.



Kuva 1. Auton elektroniikkakomponentit

2.2. Sähköjärjestelmän layout

Sähköjärjestelmän suunnittelun tavoitteena oli saada mahdollisimman selkeä johdotus ja selkeät liittimet. Toinen tavoite oli että auton modulaarisuus (etumoduuli, takamoduuli ja runko) myös näkyisi sähköjärjestelmässä.

Tärkein väline sähköjohtojen minimoimisessa on piirilevyjen käyttö. Piirilevyjen avulla voidaan liittää suuri määrä kontrolleripinnejä toisiinsa ilman irtojohtoja. Piirilevyihin on myös helppo tehdä kunnollisia liittimiä, ja liitinten viereen voi vielä saada tekstit, jossa lukee, mikä liitin on kyseessä. Tavoitteena oli tehdä niin selkeät liitännät, että kuka tahansa voisi purkaa auton, ja pistää johdot takaisin.

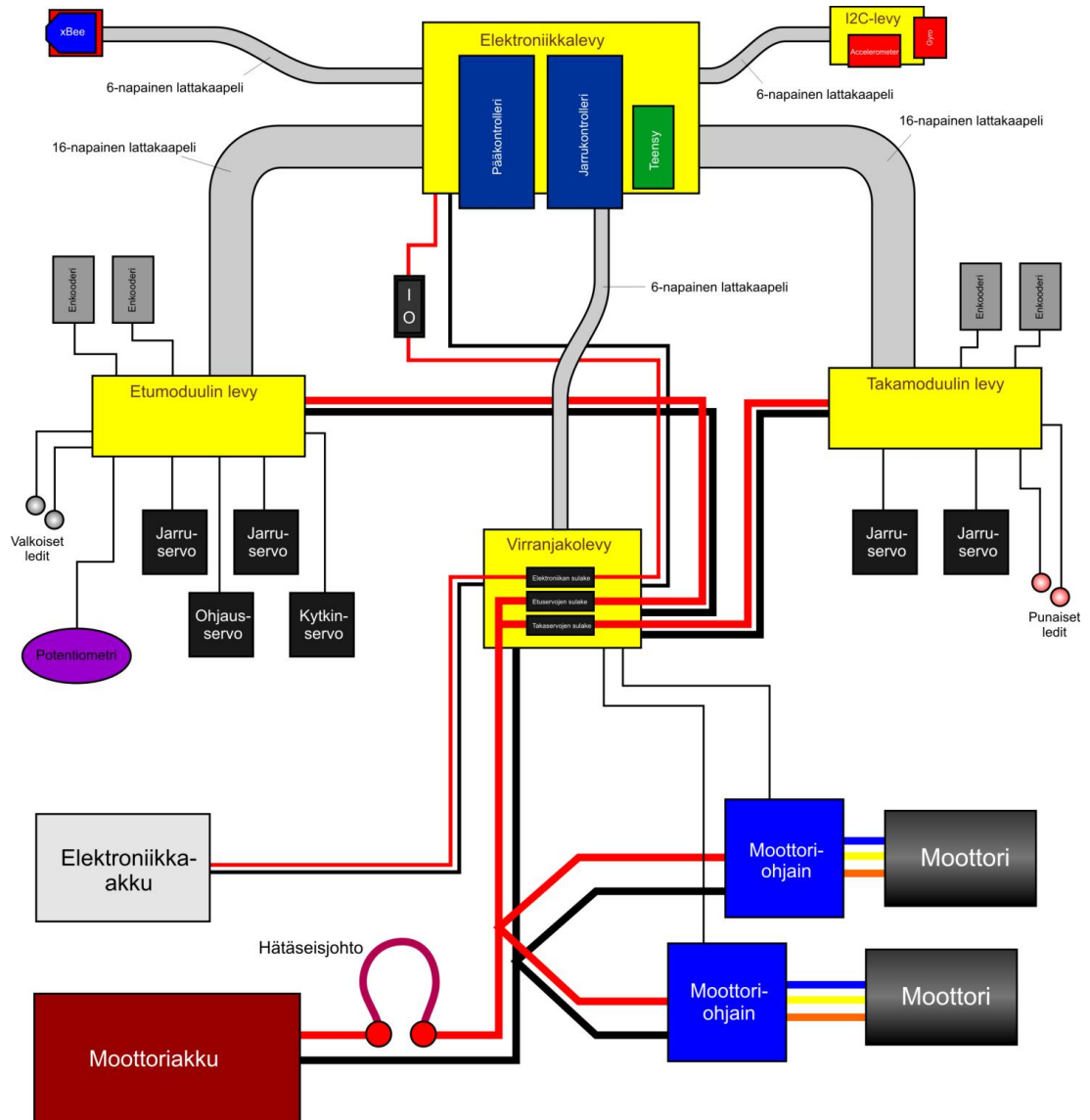
Komponenttien (esim. servojen) omat johdot ovat aika lyhyet. Kaikkien komponenttien liittäminen yhteen ja samaan elektroniikkalevyyn auton keskellä on siksi vaikeaa. Sen lisäksi auton modulaarisuus kärsii, jos suuri määrä johtoja pitää irrottaa kun etu- tai takamoduuli irrotetaan. Tämän takia päätettiin tehdä omat jakolevyt etu- ja takamoduuliin, johon kyseisen moduulin komponentit kytketään. Moduulin jakolevy liitetään sitten pääelektroniikkalevyyn yhdellä lattakaapelilla, sekä järeämmällä pari-kaapelilla moottoriakusta, jotta saadaan servoilta niiden tarvitsemat suuremmat virrat.

Servojen ja elektroniikan virta menee erillisen virranjakolevyn kautta. Tässä levyssä on sulakkeet servoilta ja elektroniikalle. Molempien akkujen maa yhdistetään virranjakolevyssä, mikä tarvitaan esim. jotta servoja olisi mahdollista ohjata. Akkujen jännitteiden mittausta tapahtuu vielä tässä samassa levyssä. Alun perin suunniteltiin että virranjakolevy olisi auton rungon sisällä, moottoreiden ja akkujen lähellä, ja sen takia virranjakolevy suunniteltiin omaksi levyksi, jotta sitä pystyisi sijoittamaan vapaasti. Oletettiin myös, että tämä piirilevy on se joka on lähimpänä moottoriohjaimia, sen takia sijoitettiin myös moottoriohjainten liittimet virranjakolevyyn. Virranjakolevy on yhdistetty pääelektroniikkalevyyn elektroniikan virransyöttöjohdoilla sekä kommunikaation lattakaapelilla.

Moottoreiden suuren virrantarpeen takia ne on kytketty suoraan akkuun, ilman sulakkeita. Moottoriakun plusjohto menee banaani-liitinten kautta, jotka toimivat hä-

täseis/päävirtakatkaisijana. Akkujohtojen haaroitus moottoreille ja servoille tehtiin abikoliittimillä, jotka paketoitiin kutistesukkien sisälle.

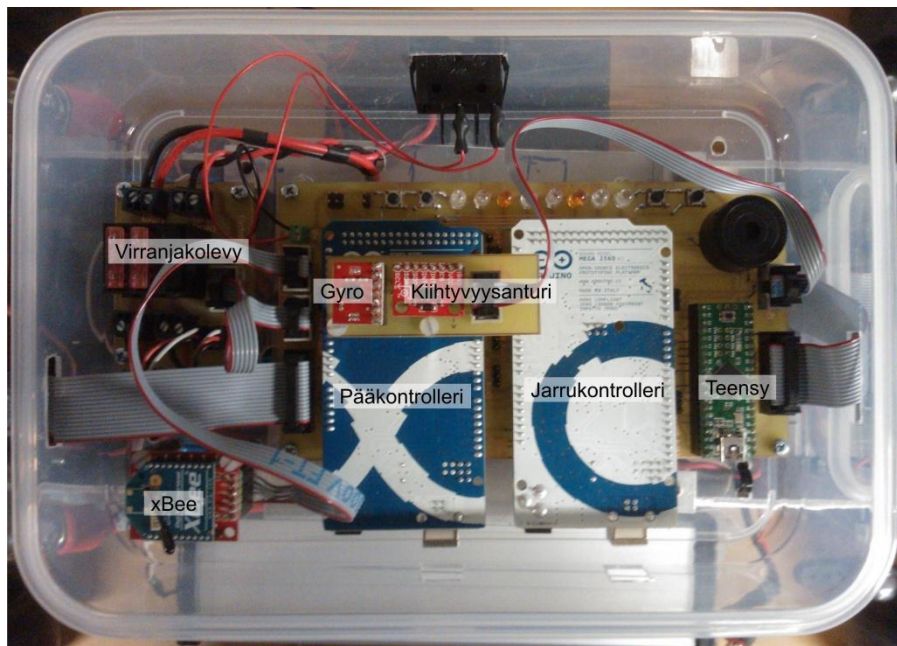
Kuvassa 2 näkyy yksinkertaistettu kytkentäkaavio auton sähköjärjestelmästä.



Kuva 2. RC-auton yksinkertaistettu kytkentäkaavio

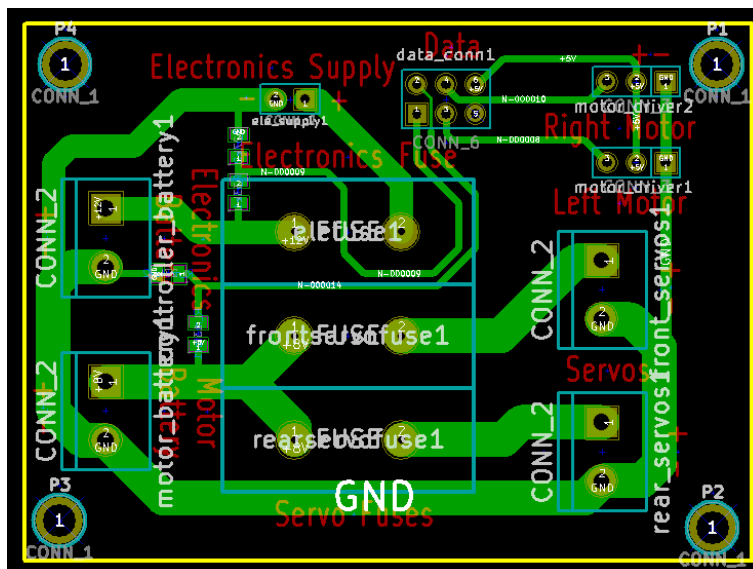
3. Piirilevyt

Piirilevyt suunniteltiin KiCadissa ja valmistettiin koneensuunnittelun laboratoriossa., ja niille löydettiin sopivankokoiset ja tiiviit suojaboksit läpinäkyvästä muovista. Ku-
vassa 3 näkyy pääelektroniikkaboksin sisältämät komponentit.



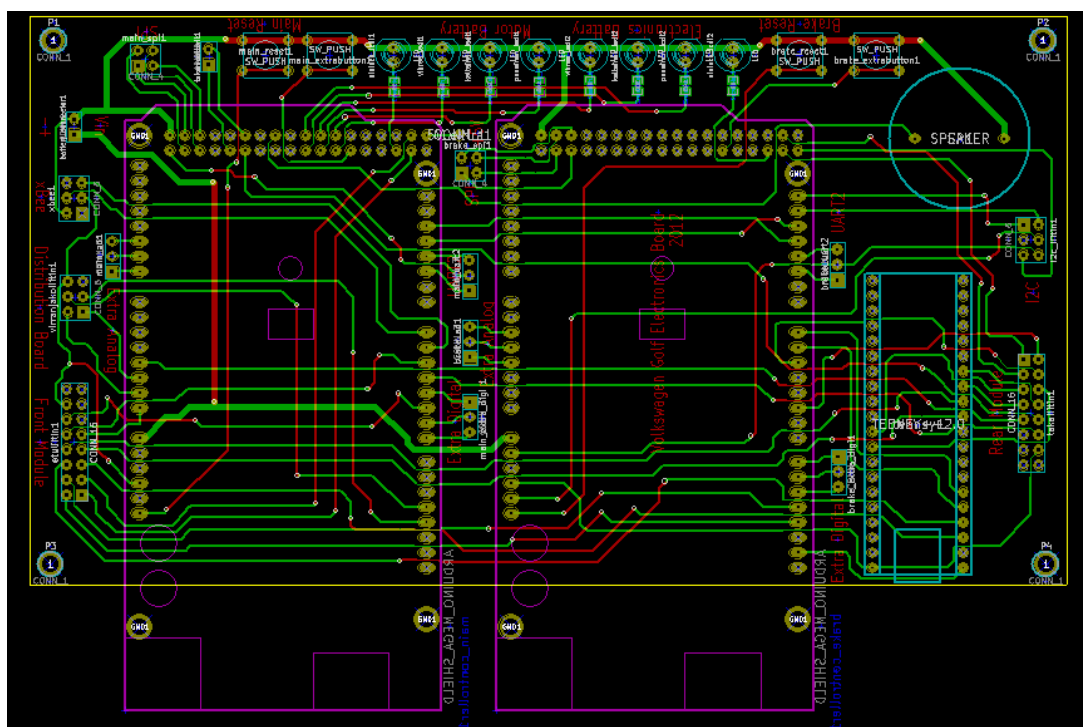
Kuva 3. Pääelektroniikkaboksin sisältämät komponentit

3.1. Virranjakolevy



Kuva 4. Virranjakopiirilevy

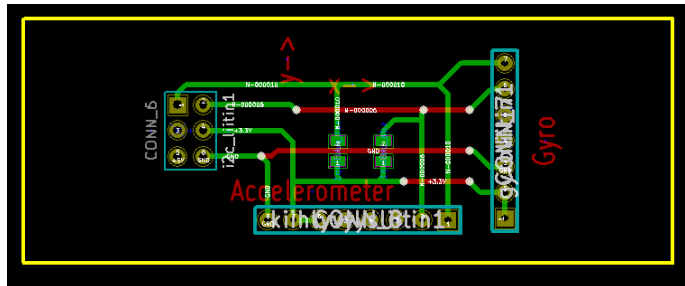
3.2. Pääelektroniikkalevy



Pääelektroniikkalevyssä (kuva 5) kaikki kolme kontrolleria on kytketty yhteen ilman irtojohtoja. Levyssä on myös summeri ("töötti"), kontrollerien reset-napit sekä ledejä. Jokaiselle Arduinolle on oma sininen heartbeatledi, joka vaihtaa tilansa jokaisella ohjelmasyklillä. Ledeistä nähdään, pyöriikö ohjelma, ja pyörivätkö eri kontrollereiden ohjelmat samalla taajuudella. Akkujen jännitteet näkyvät vihreistä, keltaisista ja punaisista ledeistä. Vihreä tarkoittaa että jännite on korkeampi kuin kyseisen akun nimellisjännite, punainen taas että jännite on alajänniterajan alla, ja keltainen että jännite on siinä välissä.

Piirilevyn suunnittelussa on pyritty ottamaan tulevaisuuden muutoksia ja tarpeita huomioon. Jokaiselle Arduinolle löytyy ylimääräisinä 3 digitaalipinniä, 3 analogipinniä, yksi sarjaportti, yksi SPI-liitin ja yksi painonappi. Löytyy myös pinnit takamoottoreiden signaaleille, mikäli autoa varustettaisiin nelivedolla tulevaisuudessa.

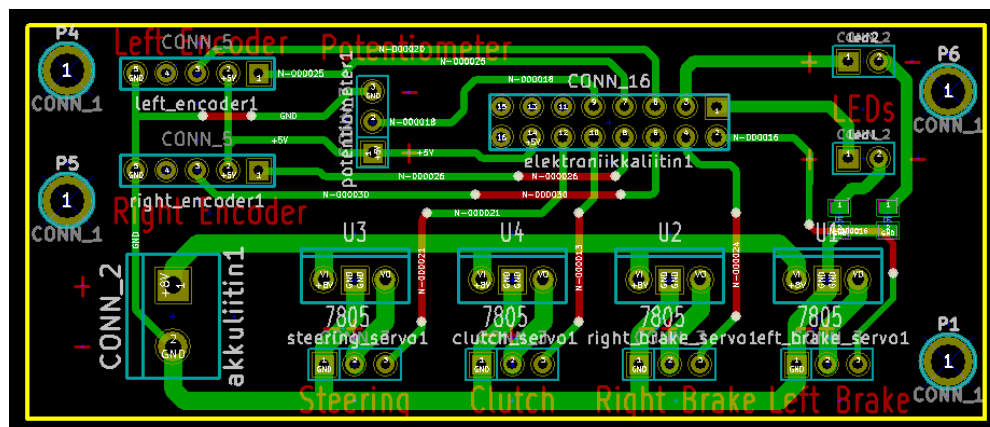
3.3. I²C-levy



Kuva 6. I²C-piirilevy

I²C-väylällä toimivat anturit, gyro ja kiihtyvyysanturi, päätettiin sijoittaa omaan piirilevyynsä (kuva 6). Tällöin niiden sijoittaminen onnistuu vapaammin, ja jos anturit vaihtuvat, tai uusia lisätään, riittää että vain tämä levy uudistetaan. Nykyiset anturit toimivat 3,3 V:lla, mutta mikäli uudet anturit tarvitsisivat 5 V, sitäkin saa I²C-levyn lattaakaapelista. Piirilevyssä on myös väylän 1,8 kΩ pull-up vastukset 3,3 V:iin.

3.4. Akselimoduulilevyt



Kuva 7. Akselimoduulien piirilevy

Akselimoduulien piirilevyt (kuva 7) toimivat lähinnä reitittimenä, joka yhdistää komponenttien signaalijohdot elektronikkalevyyn menevään lattaakaapeliin.

Servojen tarvitsema 6 V saadaan regulaattoreista, ja koska yhden regulaattorin antama virta on rajallinen, tarvitaan yksi regulaattori per servo, joten regulaattoreiden sijoittaminen on siten luontevinta servojen lähellä, akselimoduulilevyssä. Etu- ja taka-akselin piirilevyt ovat identtiset, joten mikäli taakse asennettaisiin samanlainen vetävä moduuli kuin edessä, uutta piirilevyä ei tarvitse tehdä.

Etumoduulin levyyn piti jälkeempäin tehdä jo aikaisemmin mainittu muutos, eli korvata ohjausservon regulaattori sulakkeella, jotta ohjausservo saisi riittävästi virtaa. Sulakkeen sopiva koko on noin 3 A.

Täytyy muistaa, että akselimoduulilevy saa sähköistä maata ainoastaan servojen virransyöttöjohdon kautta. Eli jos komponentteja otetaan irti testausta varten, akselimoduulilevyn pitää olla kytkettynä myös virranjakolevyn kautta elektroniikkalevyyn, jotta akselimoduulin komponentit toimisivat.

3.5. Muutoksia kytkentäkaavioihin

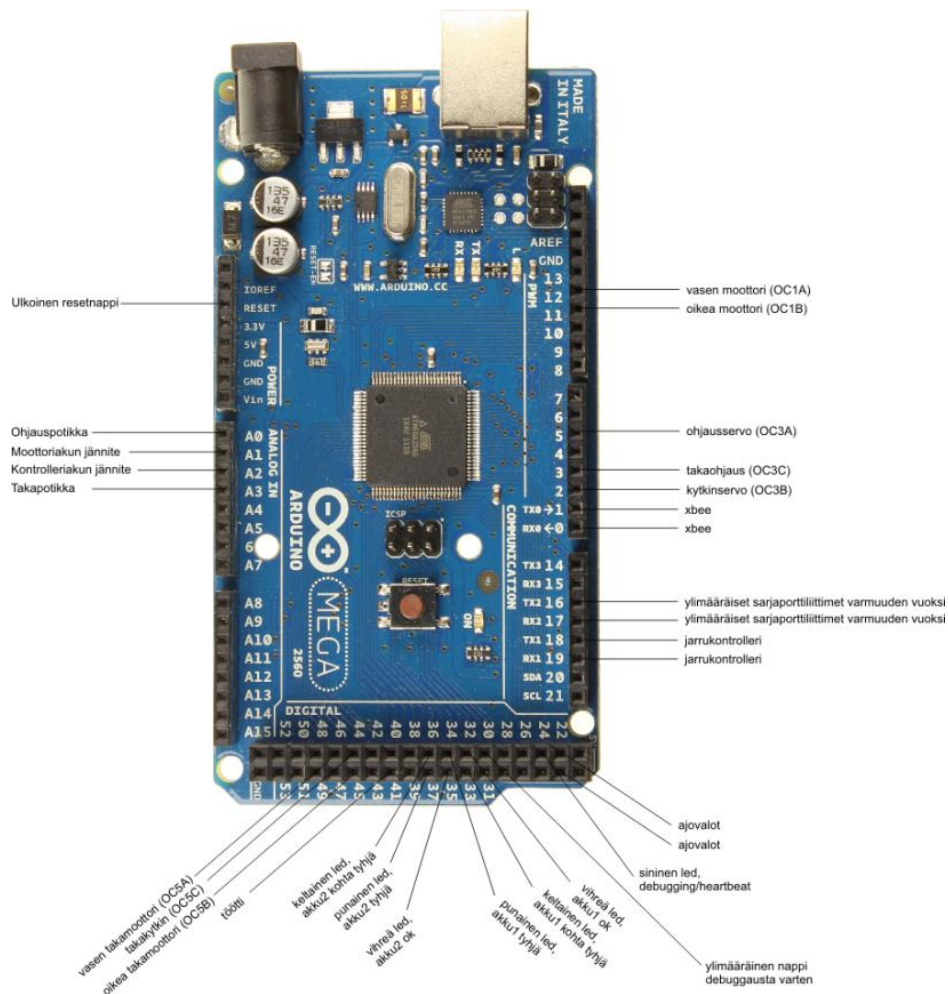
Kun päästiin testaamaan valmiita piirilevyjä, huomattiin muutama pieni vika. Ne on lueteltu tässä, jotta tulevaisuudessa tiedetään, miksi piirilevyjen kytkentäkaaviot eivät päde noissa kohdissa.

Teensy oli alun perin kytketty jarrukontrollerin uart0:aan. Silloin oli ongelmana, että jarrukontrolleri ei suostunut käynnistymään, niin kauan kuin Teensy lähetti dataa, eli koko ajan. Ongelmaa yritettiin kiertää siten, että Teensyn ohjelma käynnistyisi vasta muutaman sekunnin kuluttua kun se on saanut virtaa. Silloin saadaan kaikki kontrollerit heräämään, mutta jarrukontrollerin ohjelmoiminen ei ole mahdollista niin kauan kuin Teensy on kytkettynä ja lähettää dataa. Ongelma ratkaistiin kytkemällä Teensy jarrukontrollerin uart4:ään juottamalla pari hyppylankaa elektroniikkalevyn alapuolelle..

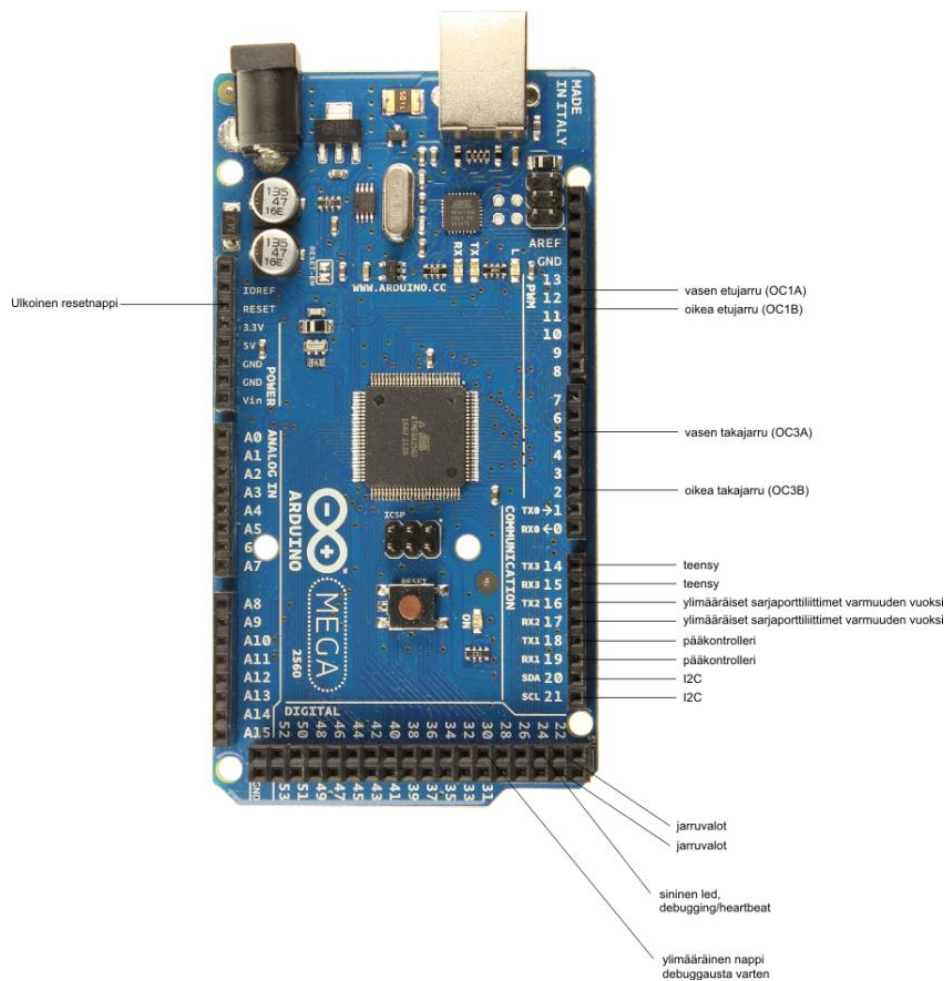
Moottoriohjaimille ei löydetty mitään datasheetiä, joka olisi sanonut mitä signaaliliittimeen oikeasti pitäisi kytkeä. Oletettiin, että punainen johto olisi 5 V input, mutta se olikin 5 V output. Kun moottoriohjaimet oli kytketty alkuperäisen kytkentäkaavion mukaan, moottoriohjainten kautta meni sähköä moottoriakusta kontrollereille. Es-tääksemme tätä, yhteys kontrollerien 5 V ulostuloon katkaistiin virranjakolevyssä.

3.6. Pinmapping

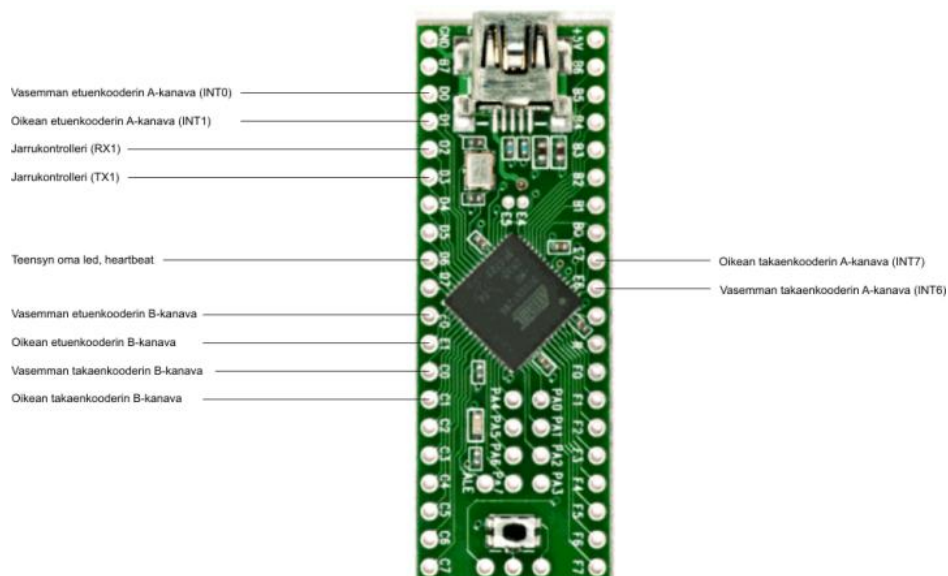
Tulevaisuuden vianetsintää varten näkyvät seuraavista kuvista, mitkä kontrolleripinnit on kytketty mihin, ja taulukkoihin on koottu lattaakaapeleiden johtojen järjestys.



Kuva 8. Pääkontrollerissa käytetyt pinnit



Kuva 9. Jarrukontrollerissa käytetyt pinnit



Kuva 10. Teensyssä käytetyt pinnit

Taulukko 2. xBeen lattakaapelin johdot

1	-
2	GND
3	+5 V
4	RX
5	TX
6	-

Taulukko 3. Virranjakolevyn lattakaapelin johdot

1	Elektroniikka-akun jännite (skaalattu 0-5V)
2	Moottoriakun jännite (skaalattu 0-5V)
3	Vasemman moottorin PWM-signaali
4	Oikean moottorin PWM-signaali
5	-
6	+5 V

Taulukko 4. Akselimoduulilevyn lattakaapelin johdot

1	Ledi (ajovalo tai jarruvalo)
2	Vasen jarru
3	Ledi (ajovalo tai jarruvalo)
4	Oikea jarru
5	Vasemman enkooderin A-kanava

6	Oikean enkooderin A-kanava
7	Vasemman enkooderin B-kanava
8	Oikean enkooderin B-kanava
9	Ohjauspottiometri
10	Kytin
11	-
12	Ohjausservo
13	-
14	+5 V
15	-
16	-

Taulukko 5. I2C-levyn lattaakapelin johdot

1	Serial Clock
2	Serial Data
3	-
4	+3,3 V
5	+5 V
6	GND

4. Koodia

Arduinoa ohjelmoitaessa löytyy paljon helppoja valmiita funktioita. Nämä eivät kuitenkaan ole järkeviä, joten RC-auton ohjelmien ohjelmissa niitä on mahdollisimman paljon pyritty välttämään. Esimerkkinä tästä on että Arduinon digitalWrite() on korvattu suoralla sijoituksella porttiin, esim. PORTA |= B00000001;

Ohjelman ajastamiseen sekä PWM-signaalien tuottamiseen tarvitaan timereita. Niiden alustamiseen pitää asettaa tiettyjä rekistereitä, mikä on manuaalisesti vähän haastavaa. Tässä tapauksessa käytettiin CodeVision-ohjelmiston (josta löytyy toimiva ilmaisversio) koodigeneraattori luomaan kontrollereille oikeat timerasetukset meidän tarpeisiin.

4.1. Ohjelman rakenne

Kaikkien kontrollereiden ohjelman rakenne on samanlainen. timer0 aiheuttaa keskeytyksen joka millisekunti. Joka kymmenes keskeytys nostetaan lippu, että pääohjelmaa pitäisi suorittaa. Ohjelman while-looppi pyörii jatkuvasti, ja ei tee muuta kuin että se tarkistaa koko ajan, onko ohjelman suorittamisen lippu nostettu, ja kutsuu sitten pääohjelman funktiota. Tällä tavalla varmistutaan siitä, että sykli aika on vakio. Sykli aika on myös helposti muutettavissa, esim. testausvaiheessa.

Arduinoissa timer1 ja timer3 on käytetty servojen ja moottoriohjainten PWM:n luomiseen.

4.2. Kommunikaatio

Laitteet kommunikoivat sarjaportin kautta. Jotta kommunikaatio olisi nopea, bittinopeus (baud rate) on nostettu vakioarvosta arvoon 38400. Mikäli tätä muutetaan, pitää muistaa ohjelmoida se muutos myös molempiin xBee-moduuleihin.

Sarjaportilla voidaan vain lähettää yksi tavu kerralla. Tämän takia tarvitaan jonkinlainen protokolla, jos halutaan lähettää useampia tavuja peräkkäin. RC-autossa on käytetty yksinkertaistettu versio peltoroboteissa käytetystä protokollasta.

4.2.1. Protokolla

Jotta vastaanotettua viestiä voisi tulkita oikein, siinä pitää olla alkumerkki. Jos alkumerkki olisi vain yksi tavu, tämä tavu olisi varattu, ja sellaista numeroarvoa ei saisi lähettää viestin sisällä. Sen takia käytetään kaksitavuista aloitusmerkkiä. Määritellään että aloitusmerkki on 0xFF seurattuna toisella tavulla, joka toimii viestityypin tunnisteena. Jotta 0xFF voitaisiin lähettää viestin sisällä numeroarvona, määritellään vielä, että viestin sisällä lisätään aina ylimääräinen 0x00 0xFF:n jälkeen.

Sarjaportin lukemisen logiikka on siis seuraavanlainen: Jokaisen vastaanotetun tavun kohdalla tarkistetaan onko se 0xFF. Jos se on, tarkistetaan mikä on seuraava

tavu. Mikäli seuraava tavu on 0x00, tiedetään että vastaanotettu tavu on data-arvo 0xFF. Jos 0xFF:n jälkeinen tavu on jokin muu, kyseessä on uuden viestin alku, ja viestityyppi määrittyy 0xFF:n jälkeisestä tavusta.

4.2.2. Protokollan speksit

Taulukossa 6 näkyy autossa käytetty protokolla kokonaisuudessaan.

Taulukko 6. RC-autossa käytetty protokolla

MSG_ID	Lähettiläs	Vastaanottaja	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7	Tavu 8	Tavu 9	Tavu 10	Tavu 11	Tavu 12	Tavu 13	Tavu 14	Tavu 15	Tavu 16	Tavu 17	Tavu 18	Tavu 19
0x10	Teensy	Jarrukontrolleri	Vasemman etupyörän nopeus	Oikean etupyörän nopeus	Vasemman takapyörän nopeus	Oikean takapyörän nopeus															
0x20	Jarrukontrolleri	Pääkontrolleri	Vasemman etupyörän nopeus	Oikean etupyörän nopeus	Vasemman takapyörän nopeus	Oikean takapyörän nopeus	x-kihtyvyyshigh byte	x-kihtyvyysslow byte	y-kihtyvyyshigh byte	y-kihtyvyysslow byte	z-kihtyvyyshigh byte	z-kihtyvyysslow byte	x-gyro high byte	x-gyro low byte	y-gyro high byte	y-gyro low byte	z-gyro high byte	z-gyro low byte			
0x30	Pääkontrolleri	Tietokone	Vasemman etupyörän nopeus	Oikean etupyörän nopeus	Vasemman takapyörän nopeus	Oikean takapyörän nopeus	x-kihtyvyyshigh byte	x-kihtyvyysslow byte	y-kihtyvyyshigh byte	y-kihtyvyysslow byte	z-kihtyvyyshigh byte	z-kihtyvyysslow byte	x-gyro high byte	x-gyro low byte	y-gyro high byte	y-gyro low byte	z-gyro high byte	z-gyro low byte	Etu- pyörän suunta	Moot- tori- akun jännite	Kont- rolleri- akun jännite
0x31	Pääkontrolleri	Jarrukontrolleri	Toivottu jarrutusvoima	Etu- pyörän suunta																	
0x41	Tietokone	Pääkontrolleri	Throttle ("kaasupolkimen asento")	Toivottu ohjaus-suunta	Toivottu jarrutus-voima	Kyt- kimen asento	Ajosu- unta														
0x42	Tietokone	Pääkontrolleri	Parametri numero "key"	Uusi arvo, tavu 1	Uusi arvo, tavu 2																
0x32	PC->Pääkontrolleri	Jarrukontrolleri	Parametri numero "key"	Uusi arvo, tavu 1 "value"	Uusi arvo, tavu 2 "value"																

Jotta kommunikaatio olisi mahdollisimman yksinkertainen ja nopea, pyrittiin käyttämään ainoastaan yksitavuisia muuttujia. Ainut kaksitavuiset muuttujat ovat anturilukemat, ja jarrukontrollerin säätöparametrit. Enkooderit on asetettu antamaan 512 pulssia per kierros, ja mikäli Teensy päivittää nopeuden joka 10 ms, yksi tavu riittää myös pyöränopeuden laskemiseen, vaikka moottori pyörisikin maksimikierroksilla.

Seuraavaan taulukkoon on koottu eri muuttujien tulkinnat.

Taulukko 7. Lähetettävien muuttujien tulkinta

Pyörien nopeus	Enkooderipulsseja yhdessä Teensysyklissä (10 ms)
Throttle	0 = vapaa, 255 = kaasu pohjassa
Ajosuunta	0 = eteenpäin, 1 = taakse
Ohjaussuunta	1 = vasemmalle, 128 = suoraan, 255 = oikealle
Jarrutusvoima	0 = ei jarrutusta, 255 = täysjarrutus
Kytkimen asento	0 = kytkin vetää, 1 = voimansiirto irtikytketty
Etupyörien suunta	Potentiometrin lukema, 8-bittisenä
Moottoriakun jännite	0 = 0 V, 255 = 8,4 V
Elektroniikka-akun jännite	0 = 0 V, 255 = 12,6 V

ABS- ja ESP-algoritmit vaativat paljon parametrivirtelyä, jotta ne saisi toimimaan. Sen takia on implementoitu toiminnallisuus, että uusia parametriarvoja voi lähettää tietokoneen käyttöliittymästä. Pääkontrolleri ottaa vastaan nekin viestit tietokoneelta, ja lähettää uudet parametrit eteenpäin jarrukontrollerille. Tietokone voi myös pyytää että jarrukontrolleri ilmoittaa parametrinsa, jolloin pääkontrolleri välittää pyynnön eteenpäin jarrukontrollerille. Jarrukontrolleri lähettää sitten parametrinsa pääkontrollerille, joka välittää ne eteenpäin tietokoneelle. Taulukossa 8 näkyy jarrukontrollerin kaikki parametrit.

Taulukko 8. Jarrukontrollerin parametrit

Mikrokontrolleri	Parametri	Id	Säätöarvot
Jarrukontrolleri	etuakselin vasemman jarruservon nolla asento	10	500 - 2500 (2 bytes)
Jarrukontrolleri	etuakselin oikean jarruservon nolla asento	11	500 - 2500 (2 bytes)
Jarrukontrolleri	takaakselin vasemman jarruservon nolla asento	12	500 - 2500 (2 bytes)
Jarrukontrolleri	takaakselin oikean jarruservon nolla asento	13	500 - 2500 (2 bytes)
Jarrukontrolleri	etuakselin vasemman jarruservon max asento	14	500 - 2500 (2 bytes)

Jarrukontrolleri	etuakselin oikean jarruservon max asento	15	500 - 2500 (2bytes)
Jarrukontrolleri	takaakselin vasemman jarruservon max asento	16	500 - 2500 (2 bytes)
Jarrukontrolleri	takaakselin oikean jarruservon max asento	17	500 - 2500 (2 bytes)
Jarrukontrolleri	abs on/off	20	1/0
Jarrukontrolleri	abs low threshold	21	0 - 255
Jarrukontrolleri	abs middle threshold	22	0 - 255
Jarrukontrolleri	abs high threshold	23	0 - 255
Jarrukontrolleri	viimeisen vaiheen pituus jarruvoiman vaiheittaisessa nostossa (1 = 10ms, 2 = 20ms ..)	24	0 - 255
Jarrukontrolleri	viimeisen vaiheen pituus jarruvoiman vakioarvon pitämisessä (1 = 10ms, 2 = 20ms ..)	25	0 - 255
Jarrukontrolleri	Ensimmäisen vaiheen kulmakerroin jarruvoimaa laskehtaessa	26	0 - 255
Jarrukontrolleri	Kolmannen vaiheen kulmakerroin jarruvoima nostettaessa	27	0 - 255
Jarrukontrolleri	Viimeisen vaiheen kulmakerroin jarruvoimaa nostettaessa (liittyy parametriin 24)	28	0 - 255
Jarrukontrolleri	ESP on/off	40	1/0
Jarrukontrolleri	ESP herkkyys kulmanopeussäädössä	41	0 - 255
Jarrukontrolleri	ESP herkkyys sortokulmasäädössä	42	0 - 255
Jarrukontrolleri	ESP jarruvoiman kerroin (muut paitsi säädettävä pyörä) (brake_pwr / esp_brake_div) * esp_brake_multi	43	0 - 255
Jarrukontrolleri	ESP jarruvoiman jako (muut paitsi säädettävä pyörä) (brake_pwr / esp_brake_div) * esp_brake_multi	44	0 - 255
Jarrukontrolleri	ESP kulmanopeussäätö, PD säätimen vahvistus P	45	0 - 255
Jarrukontrolleri	ESP kulmanopeussäätö, PD säätimen derivaatta D	46	0 - 255
Jarrukontrolleri	ESP sortokulmasäätö, PD säätimen vahvistus P	47	0 - 255
Jarrukontrolleri	ESP sortokulmasäätö, PD säätimen derivaatta D	48	0 - 255
Jarrukontrolleri	ESP sortokulmasäädön ja kulmanopeussäädön raja-arvo	49	0 - 255
Jarrukontrolleri	pyörän dynaaminen vierintäsäde	50	0 - 255
Jarrukontrolleri	akseliväli (m)	51	value(uint16)/100000,max 65535 (0,65m)
Jarrukontrolleri	ajoneuvon massa (kg)	52	value(uint16)/1000, max 65535 (65,535kg)
Jarrukontrolleri	etäisyys etu-akseliin painopisteestä (l1)	53	value(uint16)/100000,max 65535 (0,65m)

Jarrukontrolleri	etäisyys taka-akseliin painopisteestä (l2)	54	value(uint16)/100000,max 65535 (0,65m)
Jarrukontrolleri	Etu-akselin kaartojäykkyys C1	55	uint16, max 65535
Jarrukontrolleri	Taka-akselin kaartojäykkyys C2	56	uint16, max 65535

5. Yhteenveto

5.1. Tulokset

RC-auton mekatronikka on siinä mielessä onnistunut, että kaikki komponentit toimivat, ja ne on saatu siistiksi paketuksi. Elektroniikka seuraa samaa modulaarisuutta kuin loput autosta, ja johdotus on selkeä. Myös liittimet ovat selkeät, jotta melkein kuka tahansa voi purkaa komponentit ja laittaa ne takaisin oikein ilman suurempaa perehdyttämistä.

Testaamiseen jäi liian vähän aikaa, joten kokonaisuutta ei ehditty testata niin paljon kuin olisi pitänyt. Piirilevyjen juotosten suuren määrän takia, mukaan mahtui myös pieni määrä huonoja tai täysin unohdettuja juotoksia, minkä takia testauksessa huomattiin että jokin signaali ei pääse perille. Juotosviat oli kuitenkin helppo korjata.

5.2. Tunnetut ongelmat

Vaikuttaa siltä, että moottoreista tulee plusjännitettä runkoon. Voi olla että se on kyseisen moottorityypin ominaisuus. Jos jokin komponentti yhdistää rungon ja sähköisen maan, syntyy tällöin oikosulku, silloin kun moottoreita pyöritetään. Tämän takia pitää rungosta eristää etumoduulin enkooderit sekä potentiometri.

xBee on kytketty pääkontrollerin uart0:aan, eli sama sarjaportti kuin kontrollerin USB-liitin. Kun uutta ohjelmaa ladataan kontrolleriin, mikään ei saa olla kytkettynä uart0:aan. Tämän takia pitää aina irrottaa xBee:n johto, ennen kuin yrittää ladata uutta ohjelmaa pääkontrolleriin.

5.3. Jatkokehitystä

Suurin ongelma autossa on se, että kun moottoreita kuormitetaan, akut katkaisevat tosi herkästi niiden virran, ja auto pysähtyy. Hypoteesi on se, että akkujen jännite

tippuu liian alhaalle, ja akun suojakortti katkaisee virran suojatakseen akkua. Pitäisi siinä tapauksessa tutkia, miksi jännite tippuu niin alhaalle.

Moottoriakusta menevät johdot kannattaisi vaihtaa paksumpiin. Jos vertaa moottoriohjainten omiin johtoihin, akusta menevä johto on tosi kapea, kun otetaan huomioon, että siinä kulkee kahden moottorin virta. Hätäseisäkytkimen voisi vaihtaa oikeaan kytkimeen, mikäli sellainen kytkin löytyisi, joka kestäisi moottoreiden virran.

Kun moottoreita on saatu toimimaan luotettavasti, kannattaisi pääkontrolleriin implementoida PID-säädin moottoreiden nopeuksille. Nopeussäädin helpottaisi ajamista, ja se voisi myös kompensoida mahdollisia kalibrointieroja moottoriohjainten välillä. Kytkinservo olisi myös hyvä vaihtaa vahvempaan, jotta kytkintä voisi ohjata, vaikka siinä olisi parempaa vetoa antavat jäykemmät jouset.