Juha Järvenpää

SATAKUNNAN AMMATTIKORKEAKOULUN SÄHKÖAUTON SÄHKÖJÄRJESTELMÄN OHJAUKSEN PÄIVITYS

Automaatiotekniikan koulutusohjelma
2014

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

SATAKUNNAN AMMATTIKORKEAKOULUN SÄHKÖAUTON SÄHKÖJÄRJESTELMÄN OHJAUKSEN PÄIVITYS

Järvenpää, Juha
Satakunnan ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma
Joulukuu 2014
Ohjaaja: Asmala, Hannu
Sivumäärä: 30
Liitteitä: 3

Asiasanat: sulautettu järjestelmä, arduino, raspberry pi, auton ohjausjärjestelmä

Opinnäytetyön tarkoituksena oli korvata Satakunnan ammattikorkeakoulun Volkswagen Kleinbus sähköauton sähköjärjestelmän ohjaus sulautetulla järjestelmällä. Autossa oli Beckhoff:n logiikalla toteutettu ohjaus. Vanha ohjausjärjestelmä oli paljon tehoa kuluttava ja hidas käynnistymään ja sammumaan.

Ohjausjärjestelmä ohjaa muun muassa valoja, moottorin ja akkukotelon tuulettimia sekä mittaa lämpötiloja ja nopeutta. Moottorin ohjaukseen ja lataukseen on omat järjestelmänsä. Kosketusnäytöltä voidaan kytkeä päälle esimerkiksi stereot, invertteri ja alustavalot. Alustavalojen väri ja valon voimakkuus voidaan asettaa kosketusnäytöltä. Näytöllä näytetään nopeus, lämpötilat ja perinteiset symbolit valoille ja muille laitteille.

Työssä käytettiin Arduino Mega ADK:ta ohjaamaan järjestelmää ja lukemaan antureita. Kosketusnäyttöä ohjaa Raspberry Pi pienoistietokone. Arduinon päälle kytkentärimoihin kiinni suunniteltiin kytkentäpiirilevy, jossa on ruuviliittimet tuleville signaaleille ja lähteville ohjauksille sekä tarvittavat jännitetasomuutokset. Kytkentäpiirilevy tilattiin erillisosina ja koottiin juottamalla ammattikorkeakoulun tiloissa. Uusi ohjausjärjestelmä asennettiin sähköautoon ja testattiin sen toimivuus.

Työn toteutuksen pääosat olivat kytkentäpiirilevyn suunnittelu ja kokoaminen, Arduinon ohjelmoiminen C-kielellä, Raspberry Pi:n ohjelmoiminen Java-kielellä ja asennus autoon sekä testaus. Järjestelmä on pyritty tekemään siten, että sen muokkaaminen on mahdollista myöhemmin. Esimerkiksi kosketusnäytön grafiikat ovat yhdessä kansiossa png-muodossa ja vaihdettavissa mihin tahansa samassa muodossa olevaan samankokoiseen kuvaan.

THE UPDATE OF SATAKUNTA UNIVERSITY OF APPLIED SCIENCES ELECTRIC VEHICLE ELECTRICAL SYSTEM CONTROLLER

Järvenpää, Juha
Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences
Degree Programme in Automation
December 2014
Supervisor: Asmala, Hannu
Number of pages: 30
Appendices: 3

Keywords: embedded system, Arduino, raspberry pi, car electrical system

---

The purpose of this thesis was to replace the electrical system of Volkswagen Kleinbus electric car with embedded system. The old system fitted in car was done with Beckhoff logic. The old system consumed a lot of power and was slow to start and shut down.

The control system controls for example lights and motor and battery casing fans. It also measures temperatures and vehicle speed. There are independent systems for controlling motor and charging. Stereos, inverter and chassis lights for example can be turned on from touch screen. The color and brightness of the chassis lights are adjustable through touch screen. Speed, temperatures and traditional symbols for lights and other apparatus are shown in the touch screen.

Arduino Mega ADK was used for controlling outputs and reading input data from sensors. Raspberry Pi, a small computer, controls the touch screen. Circuit board with screw connectors for inputs and outputs and necessary voltage level transforming was designed so that it can be fitted on top of Arduino using its pin headers. The individual components for circuit board were ordered and the circuit board was soldered in Satakunta University of Applied Sciences. The new controlling system was installed in the electric car and tested.

The main parts of the work were design and assembly of the circuit board, writing programming code for Arduino with C-language and raspberry pi with Java-language, installment in car and testing. The system is designed so that it can be modified with relative ease later on. For example the graphics on the touch screen are in one folder in png-form and interchangeable with any picture in the same form and size.

SISÄLLYS

1 JOHDANTO

Satakunnan ammattikorkeakoulussa on tehty oppilastyönä sähköautokonversio vuosimallin 1973 Volkswagen Kleinbus T2:lle. Auto oli takamoottorinen ja sähköjärjestelmältään yksinkertainen, joten konversio oli helppo tehdä. Auton valintaan vaikutti myös huomiota herättävä ikoninen ulkomuoto. Lisäksi pakettiautomaisessa korissa oli runsaasti tilaa akustolle ja ohjausjärjestelmille. (Myntti 2014, 6; Taberman 2014, 6) Sähköauto on kuvassa 1.1.

Kuva 1.1. Satakunnan ammattikorkeakoulun sähköauto.

Myntin mukaan (2014, 11) auton sähkömoottori on Kostov Motors:n valmistama K11” Alpha. Teho siinä on 45 kW ja vääntöä maksimissaan 85 Nm. Sähkömoottori on kytketty auton alkuperäisen vaihdelaatikon kautta taka-akselille, jolloin vaihdevalitsimella voidaan vaikuttaa välityksiin ja siten kiihtyvyyteen ja huippunopeuteen. Sähkömoottori näkyy kuvassa 1.2, jossa sähköauton konepelti on auki. Kuvassa 1.3 on auton takasisätilat ja toinen televisioista.

Kuva 1.2. Satakunnan ammattikorkeakoulun sähköauton takaosa.

Kuva 1.3. Satakunnan ammattikorkeakoulun sähköauton takasisätilat.

Auton alkuperäiset ajoneuvosähköt korvattiin ensin Beckhoff:n logiikkaohjauksella, jossa kosketusnäytöltä voitiin seurata ajoneuvon tilaa. Ohjelmoitava logiikka ohjasi

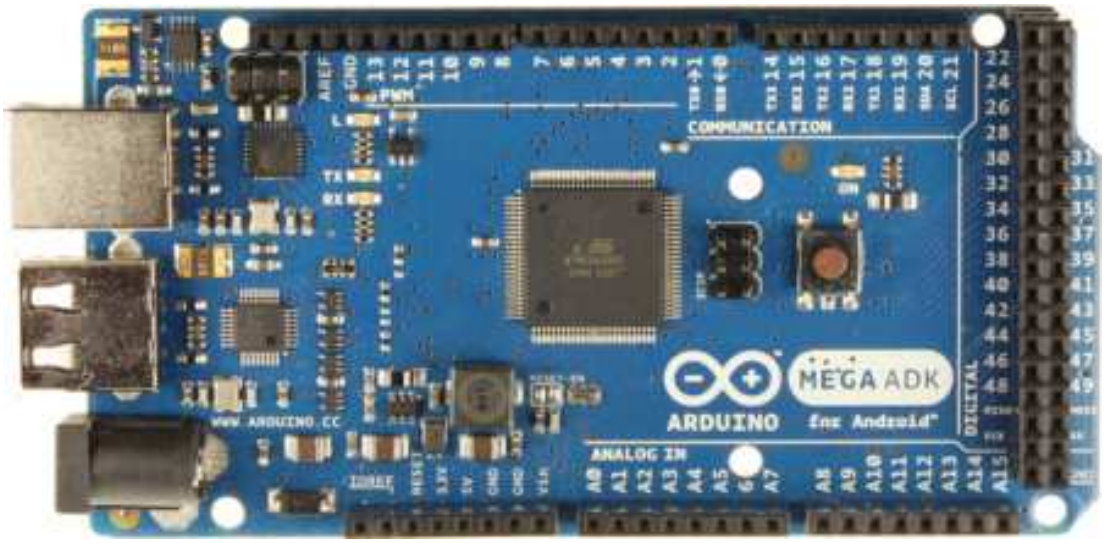
releitä, joihin on kytketty auton ohjattavat sähköjärjestelmän komponentit. (Taberman 2014, 18)

Opinnäytetyön tarkoituksena on muuttaa Beckhoff:n logiikkaohjauksella toimiva sähköjärjestelmän ohjaus toimimaan sulautetulla järjestelmällä. Siten saadaan ohjausjärjestelmän tehonkulutusta vähennettyä ja toimintaa nopeutettua. Beckhoff:n järjestelmä oli hidas käynnistymään ja sammumaan. Lisäksi kosketusnäytön kosketusherkkyys ei ollut kovin hyvä ja grafiikka oli yksinkertaista.

2 KOMPONENTTIEN VALINTA

2.1 Arduino

Auton sähköjärjestelmää valittiin ohjaamaan Arduino Mega ADK mikrokontrolleriohjattu kehitysalusta. Kuvassa 2.1 on Arduino Mega ADK. Kehitysalustassa on ATmega 2560 mikrokontrolleri, joka toimii 16 MHz taajuudella. Nopeus riittää hyvin autossa oleviin yksinkertaisiin ohjauksiin ja antureiden lukemiseen.



Kuva 2.1. Arduino Mega ADK. (Arduino www-sivut)

Ohjelmointiin käytetään erillistä Arduino-ohjelmaa, jonka ohjelmointikielenä on C. Mikrokontrollereiden ohjelmointi C-kielellä on perinteisesti haastavaa aloittelijoille, koska tarvittavien rekisterien asetukset ovat hankalia määrittää. Arduino käyttää kirjastoja, joiden avulla hankalatkin asetukset voidaan määrittää hyvin helposti muutamalla komennolla. Arduino on laajasti käytössä, joten esimerkkejä ja sovelluksia löytyy runsaasti. Arduinoa voidaan käyttää myös kaupallisiin sovelluksiin. (Arduino www-sivut 2014)

Mega ADK-malli valittiin, koska siinä on runsaasti sisään- ja ulostuloja ja USB-portti latausohjaimen tietojen lukemiseen. Arduino käynnistyy nopeasti, jolloin vilkut, valot sekä muut ohjaukset ja sisääntulot ovat heti käytettävissä. Arduinon päälle on suunniteltu I/O-kortti, jossa oleviin ruuviliittimiin on kytketty sisään- ja ulostulot.

Arduino lukee esimerkiksi nopeus- ja lämpötila-antureita sekä ON/OFF -kytkimiä. Ulostulot voivat ohjata releitä tai jopa suoraan laitteita.

2.2 Raspberry Pi

Arduinon suorituskyky ei riitä kosketusnäytön ohjaamiseen, joten sitä varten on valittu Raspberry Pi –pienoistietokone. Kuvassa 2.2 näkyvä Raspberry Pi on luottokortin kokoinen tietokone, jossa toimii linux käyttöjärjestelmä. Suorittimena on 700 MHz taajuudella toimiva ARM. Suorittimen nopeus on 1990-luvun lopun tietokoneiden tasolla. Näytönohjain on suorittimeen nähden tehokas ja pystyy suorittamaan teräväpiirtovideota. (Raspberry Pi www-sivut 2014)

Kuva 2.2. Raspberry Pi. (Raspberry Pi www-sivut)

Raspberry Pi:n erona perinteisiin tietokoneisiin on, että siinä on sisään- ja ulostulopinnejä, joiden kautta voidaan lukea ON/OFF-tietoa ja ohjata ulkopuolisia laitteita. Näiden pinnien kautta Raspberry Pi keskustelee Arduinon kanssa.

Raspberry Pi käyttää SD-korttia muistinaan. Sinne varastoidaan kaikki tiedot käyttöjärjestelmä lukien. Muistikorttia voidaan lukea ja muuttaa tietokoneella. Käyttöjärjestelmän lataamiseen muistikortille käytettiin Win32DiskImager – ohjelmaa. Muistikortin tietoja voidaan muokata Paragon ExtFS for Windows – ohjelman avulla.

Ohjelmointikieleksi Raspberry Pi:hin valittiin java. Ohjelmointiin käytettiin Netbeans-ohjelmaa. Käyttöliittymä on toteutettu Java Swing:llä. Java Swing on nykyisin korvattu JavaFX:llä, mutta kosketusnäyttö ei suostunut näyttämään JavaFX:n ohjelmia. Sen vuoksi jouduttiin käyttämään vanhentunutta tekniikkaa. JavaFX:n ei ole rakennettu ominaisuutta, jolla voitaisiin lähettää kuvaa USB-liitintä käyttäen. Kuva tulee ainoastaan hdmi-liittimestä. Järjestelmään valitussa näytössä kuvasignaalin siirto oli mahdollista ainoastaan USB-liitintä käyttämällä. JavaFX:n etuina olisivat olleet vähemmän prosessoritehoa vaativat animaatiot, huomattavasti yksinkertaisemmat käskyt ja parempi ohjelmointiympäristö.

2.3 Kosketusnäyttö

Kosketusnäyttöjen etsiminen osoittautui hankalaksi, koska linux vaatii kosketukselle omat ajurinsa. Raspberry Pi:n linux-jakelut ovat erikoisversioita ja linux-ajurit eivät välttämättä toimi niissä. Etsimisessä painotettiin valmistajan mainintaa Raspberry Pi yhteensopivuudesta. Kosketusnäytöksi valittiin kuvassa 2.3 näkyvä kapasitiivinen 10,1 tuumainen Mimo Magic Touch monitori. Monitorin resoluutio on 1024 x 600 ja kirkkaus 200 cd/m^2. Monitorin kuva ja kosketustiedot kulkevat USB 2.0 liitännän kautta. Siinä on lisäksi toinen USB-liitin, jonka avulla voidaan syöttää lisävirtaa. Kosketusnäyttö on koteloitu mustaan muovikuoreen.

Kuva 2.3. Mimo Magic Touch. (Mimo www-sivut)

Kosketusnäytöksi oli tarkoitus hankkia Chalkboard Electronics:n 10-tuumainen malli, mutta heidän varastonsa olivat loppuneet ja saatavuudesta ei ollut tietoa. Chalkboard Electronics:n kosketusnäyttö olisi ollut huomattavasti halvempi ja sen luvattiin toimivan javaFX:n kanssa.

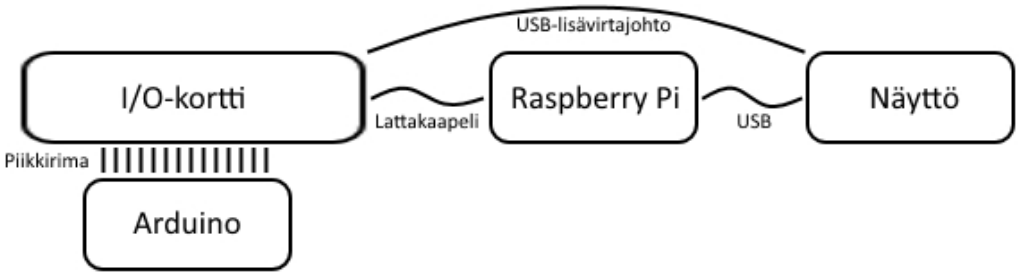
2.4 Muut

Piirilevy hankittiin Eurocircuit:lta, kosketusnäyttö Mimo:lta ja muut komponentit ELFA Distrelec:ltä. Auton laitteiden käyttöjännite on 12V, mutta Arduino ja Raspberry Pi toimivat 5V jännitteellä. Siksi oli hankittava jännitemuunnin. Muuntimeksi valittiin 15W tehoinen malli, jonka ulostulo on hienosäädettävissä viiden voltin molemmin puolin. Muunnin kestää myös akun latausjännitteen, joka tyypillisesti on hieman alle 14V. Siinä on ruuviliittimet, joiden avulla se voidaan liittää I/O-kortille. Jännitemuuntimet tyypillisesti menettävät antotehoaan lämpötilan noustessa, mikä on pyritty ottamaan huomioon mitoituksessa.

Ulostuloja ohjaamaan on valittu mosfet-komponentit. Alustavalojen PWM-säätöä (jännitepulssien pituuden säätöä) varten on valittu n-kanava MOSFETIT ja muihin ulostuloihin on p-kanava MOSFETIT. MOSFETTIEN ohjausjännitettä korottamaan on valittu edulliset n-kanava MOSFETIT. Vastusten fyysiseksi kooksi on suunnittelun ja juotosten helpottamiseksi otettu 1206. Vastusten koko on siten 3,2mm x 1,6mm. Suuren vastuksen alta voi tarpeen mukaan vetää jopa kaksi johdinvetoa piirilevyllä. Sisään- ja ulostuloliittimet ovat ruuviliittimiä, joihin mahtuvat paksutkin johtimet. I/O-kortti on yhteydessä Arduinoon piikkirimojen avulla. Koska Arduinon pinnalla on korkeita liittimiä, tuli piikkirimojen olla riittävän pitkiä.

3 SUUNNITTELU JA TOTEUTUS

3.1 Järjestelmä yleisesti



Kuva 3.1. Järjestelmän kaaviokuva

Kuvassa 3.2 on esitetty järjestelmä yleisesti. I/O-kortti on Arduinon päällä ja kiinnitetty siihen piikkirimalla. I/O-kortin ja Raspberry Pi:n välillä on lattakaapeli. Kosketusnäyttö on USB-kaapelilla kiinni Raspberry Pi:ssä. Kosketusnäytön lisävirtajohto on kiinni I/O-kortin USB-liittimessä.

Järjestelmästä on pyritty tekemään yksinkertainen käytöltään. Valot ja muut auton laitteet ovat käyttökunnossa heti virran päälle kytkemisen jälkeen. Kosketusnäytöllä näkyvät informaatiot, kuten nopeus tulevat näkyviin hetken kuluttua. Käyttäjän pitää vain kytkeä virta yhdestä kytkimestä päälle, niin se on käyttökunnossa. Auton hallintalaitteet ovat perinteiset, jolloin kuka tahansa voi ajaa autoa. Kaikki tarvittava informaatio on nähtävissä kosketusnäytöltä.

Arduino Mega ADK on järjestelmän tärkein osa, koska se lukee sisään tulevat signaalit, käsittelee ne ja ohjaa tarvittavia ulostuloja. Suurin osa signaaleista on päälle/pois tietoja. Lämpötilat tulevat jännitetietona ja ne muutetaan mikrokontrollerin sisäisellä AD-muuntimella 10-bittiseksi tiedoksi. Nopeus on pulssimuotoista, jolloin nopeus saadaan laskemalla pulssien lukumäärä tietyssä ajassa. Arduino laskee pulssit yhteen ja lähettää ne Raspberry Pi:lle, jossa ne muutetaan nopeudeksi liukulukulaskennalla. Raspberry Pi:ssä on prosessorissa oma liukulukulaskentayksikkö, jolloin niiden laskenta ei vie prosessoriaikaa muilta toiminnoilta.

Arduinon päällä on piikkirimojen avulla kiinnitettynä I/O-kortti. I/O-kortilla on ruuviliittimet tuleville ja lähteville signaaleille ja tarvittavat jännitemuutokset. Lisäksi siinä on lattakaapeliliitin Raspberry Pi liikennöintiä varten, USB-liittimet Raspberry Pi:n ja näytön virransyöttöä varten sekä varalla R9-liitin akkujen latausyksikön kanssa kommunikointiin. Ensisijaisesti liikennöinti latausohjaimen kanssa on suunniteltu toimivan Arduinon USB-portin kanssa. Arduino Mega ADK-malli pystyy toimimaan myös USB-master laitteena toisin kuin tavallinen Mega.

Raspberry Pi ja kosketusnäyttö saavat virtansa I/O-kortilta. Tarkoitus oli ohjata niiden virransyöttöä Arduinolla. Kosketusnäyttö sai kytkennästä kuitenkin niin paljon häiriötä, että ohjaus poistettiin varmuuden vuoksi. Kytkentää muutettiin niin, että Raspberry Pi ja kosketusnäyttö käynnistyvät samaan aikaan kuin Arduinokin. Raspberry Pi:lle on asennettu Raspbian linux-versio, jonka käynnistymistä on nopeutettu.

Kosketusnäytön käyttöliittymä on tehty Java Swing – kielellä ja se käynnistyy automaattisesti virran kytkemisen jälkeen noin 44 sekunnissa. Kosketusnäytöltä voidaan ohjata esimerkiksi sisävalot ja radio päälle sekä lukea auton nopeus ja moottorin sekä akkulaatikon lämpötilat.

Arduino I/O-kortteineen sekä Raspberry Pi on sijoitettu etupenkkien alle. Kosketusnäyttö sijaitsee kojetaulussa. I/O-kortti on kytketty ohjaamaan autossa valmiiksi olleita releitä, mutta käytetyt mosfetit pystyvät ohjaamaan myös suoraan laitteita. Releet kuluttavat useita kymmeniä watteja ollessaan kytkettynä. Mosfetit kuluttavat vain kymmeniä milliwatteja. Arduinolta ohjattavat ja luettavat signaalijohdot kulkevat johtokouruja pitkin auton eri osiin.

3.2 Anturit

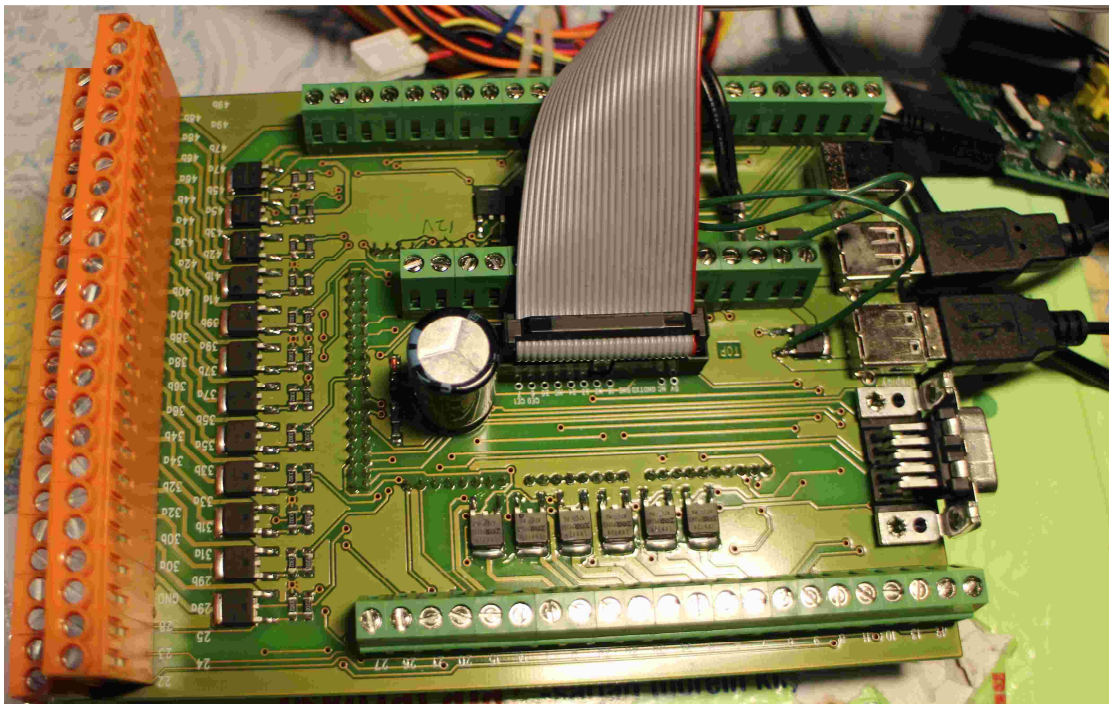
Lämpötila mitataan resistiivisellä NTC-anturilla (negatiivisen lämpötilakertoimen omaava anturi). Anturin toinen pää on kytketty maihin ja toinen pää Arduinon sisääntuloon. Sisääntulossa on lisäksi 33 kΩ ylösvetovastus, jolloin syntyy jännitteen-

jakokytkentä. Jännite sisääntulossa vaihtelee NTC-anturin vastuksen mukaan. NTC-anturin vastus vähenee lämpötilan noustessa, jolloin jännite Arduinolle vähenee.

Nopeus mitataan Hall-anturilla. Arduinolla on sisääntulossa ylösvetovastus, joka pitää sisääntulon käyttöjännitteessä. Kun Hall-anturin ohi kulkee magneetti, anturi avaa hetkeksi ulostulonsa maihin. Arduinon sisääntulo laskee silloin myös nollaan volttiin. Näitä pulsseja mitataan keskeytysohjatusti, jotta yksikään pulsseista ei mene ohitse. Joka kerta, kun Arduino havaitsee pulssin laskevan reunan, niin se suorittaa keskeytysohjelman, joka keskeyttää ohjelman muun toiminnan hetkeksi. Keskeytysohjelmassa lasketaan pulssien lukumäärää.

3.3 I/O-kortti

I/O-kortti on 125mm leveä ja 170mm pitkä, jolloin se mahtuu sähköautossa samaan tilaan, jossa Beckhoffin logiikka oli. Kuvassa 3.1 näkyy I/O-kortti testausvaiheessa. Ruuviliittimet vievät paljon tilaa, jonka vuoksi niitä on sijoitettu eri puolille levyä. Keskellä olevissa ruuviliittimissä on käyttöjännitteet 12V, 5V ja GND eli signaalimaa. Signaalimaa on yhdistetty auton maapotentiaaliin. Kuvassa 3.1 vasemmalla olevat kaksikerroksiset ruuviliittimet ovat pääosin ulostuloja ja molemmissa reunoissa olevat sisääntuloja. Kuvassa oikealla ovat USB-liittimet. Keskellä oleva lattakaapeli on liikennöintijohto Arduinon ja Raspberry Pi:n välillä.



Kuva 3.2. I/O-kortti.

Sivuilla oleville ruuviliittimille jää autossa asennustilaa noin 20mm. MOSFETIT on pyritty sijoittamaan lähelle riviliitintään, jotta uloslähtevät signaalireitit olisivat mahdollisimman lyhyet. Komponentit on sijoitettu pääosin suoriin riveihin, jotta niiden tunnistaminen levyllä vianetsintätilanteessa on helppoa. Kuvassa 3.1 on piirilevyn yläpuoli. Piirilevyn toisella puolella on vielä enemmän komponentteja. Vedot (johtimet), joissa kulkee paljon virtaa, on pyritty tekemään mahdollisimman paksuiksi. Muut vedot on tehty ohuemmiksi, jotta niiden reititys on helpompaa. Piirilevyn suunnittelussa on käytetty PADS-ohjelmistoa.

Alussa kaikki MOSFETIT oli suunniteltu olevan n-kanava tyylisiä, joita ohjattiin 5V signaalilla. Auton releitä ohjataan pätkimällä 12V tulopuolta eikä maapuolta, johon n-kanava MOSFETIT on suunniteltu. N-kanava MOSFETIT toimivat myös tulopuolella, mutta niiden virran päästökyky ei aivan riittänyt releiden keloille. Releet vaativat 300mA virtaa kytkeytyäkseen.

Piirilevyn MOSFETIT vaihdettiin p-kanava toimisiksi PWM-ohjausta lukuun ottamatta. P-kanava MOSFET vaatii vähintään ohjattavan jännitteen verran kannalleen, jotta se sulkeutuu. Alkuperäisesti ohjausjännite oli suoraan Arduinolta tuleva 5V. Siksi jouduttiin tekemään pienen n-kanava MOSFETIN avulla jännitteen nosto. Kyt-

kentä tehtiin piirilevyllä olleiden vastusten juotostäplien päälle sijoittaen komponentit hieman päällekkäin. MOSFETEILTÄ tulee kaksi ulostuloa ruuviliittimille. Ruuviliittimeen, jonka nimessä on a, on kytketty 12V (esimerkiksi 50a). Toiseen ruuviliittimeen, jonka nimessä on b, on kytketty varsinainen signaali.

Kytkentä ruuviliittimille oli tehty joustavaksi, joten niihin ei tarvinnut tehdä muutoksia. Jokaiselta ulostulon MOSFETILTA tulee kaksi johdinta ruuviliittimelle. Toiseen ruuviliittimeen kytketään 12V ja toiseen ohjattava johdin. Sisääntuloissa on yksinkertainen jännitteenjakokytkentä vastuksilla. Jännite muunnetaan 12 voltista viiteen volttiin, jonka Arduino pystyy lukemaan. Vastukset rajoittavat samalla sisään tulevaa virtaa.

Piirilevylle on suunniteltu myös mahdollisuus äänisignaalin sisään tuomiseksi ja muuttamiseksi ohjaamaan alustavaloja musiikin tahtiin. Kytkentä koostuu suojadiodista ja kaistanpäästösuotimesta. Kaistanpäästösuotimen taajuus on noin 62 Hz ja kaista on pyritty tekemään kapeaksi.

Piirilevyt tilattiin Puolalaiselta yritykseltä nimeltään Eurocircuit. Piirilevyt tulivat noin viikossa. Kiinalaisilta toimittajilta piirilevyjä olisi saanut huomattavasti edullisemmin, mutta toimitusajat olisivat olleet yli kuukauden mittaisia. Suomalaiselta Prinel Oy:llä toimitus olisi ollut erittäin nopea, mutta hinta oli suuri. Piirilevyyn painettiin silkkipainolla selitteet ruuviliittimille, jotta kytkentä olisi helpompaa. Juotteenestopinnoitteeksi valittiin perinteinen vihreä, piirilevyn paksuudeksi 1,6mm ja kuparin paksuudeksi 16µm.

3.4 Auton sähköjärjestelmän ohjaus Arduinolla

Arduinon koodi on kirjoitettu C-kielellä käyttäen Arduinon valmiita kirjastoja. Koodi on kirjoitettu Arduinon omalla ohjelmalla, jolla se myös ladataan USB-kaapelin kautta mikrokontrolleriin. Valmiit kirjastot nopeuttavat huomattavasti koodin kirjoittamista, koska yhdellä rivillä koodia voidaan tehdä useita kymmeniä rivejä vaativa toiminto.

Perinteisesti koodin kirjoittamisen yhteydessä joutuu lukemaan runsaasti mikrokontrollerin valmistajan datakirjaa. Ne ovat yleisesti erittäin sekavia ja siten virheiden mahdollisuus kirjoitettavassa koodissa on suuri. Helposti jokin rekisterimerkintä jää tekemättä tai valitaan väärä yhdistelmä, jolloin ohjelma ei tee haluttua toimintoa. Arduinon kirjastojen komennot on tehty erittäin yksinkertaisiksi ja niihin löytyy hyvät ohjeet ja esimerkkejä internetistä. Koska Arduinon käyttäjämäärä on suuri, on keskustelupalstoilla ratkaisuja ja esimerkkejä useimpiin ongelmatilanteisiin virallisten sivujen lisäksi.

Koodi on pyritty tekemään yksinkertaiseksi ja loogiseksi, jolloin uuden lisääminen ja virheiden hakeminen on mahdollisimman helppoa. Koodin alussa on koodiversio, tekijä ja viimeisimmät muutokset. Sen jälkeen on sijoitettu useimmin muutettavat vakiot ja niiden perään muut vakiot ja muuttujat. Vakioita ovat esimerkiksi vilkkutaajuus. Kaikille ulos- ja sisääntuloille on annettu selkeä nimi ja yhdistetty se varsinaiseen tunnukseen, jota ohjataan. Varsinaisessa ohjelmassa viitataan tähän nimeen, jolloin ohjelma osaa yhdistää sen oikeaan tunnukseen. Siten voidaan koodin alussa yhtä tunnusta muuttamalla muuttaa se kaikkialle koodiin. Jos näin ei tehtäisi, voisi joku kohta jäädä muuttamatta, jolloin virheen hakeminen voi olla vaikeaa.

Noin puolet Arduinon I/O-pinneistä on järjestelmässä sisääntuloja ja puolet ulostuloja. Arduino lukee sisääntuloja pollaamalla, eli käymällä järjestelmällisesti kaikki sisääntulot läpi. Järjestelmässä ei ole nopeusanturia lukuun ottamatta niin aikakriittisiä tapahtumia, että keskeytyksiä tarvittaisiin.

Koska ohjelma ei saa jäädä odottamaan paikalleen mihinkään vaiheeseen, niin ei voida käyttää yksinkertaista viive-funktiota. Viiveet on tehtävä laskureilla, jotta ne eivät vie turhaan prosessoriaikaa. Auton hallintalaitteiden painaminen ei saa jättää ohjelmaa jumittamaan painamisen ajaksi. Esimerkiksi vilkkua käytettäessä ohjelman pitää pystyä lukemaan jarrupolkimen painallus ja sytyttää jarruvalot.

Lämpötila-anturikytkennän antama jännitetieto muutetaan AD-muuntimella (analogisesta digitaaliseksi muuntamalla) 10 bittiseksi digitaalitiedoksi. 0-5V tulolla saadaan maksimissaan 1024 eri arvoa. Anturilta tulevan jännitetiedon vaihteluväli on 0,1V – 4,9V. AD-muuntimen lämpötilan lukutarkkuus on noin 0,2 $^{\circ}$C. Anturin datalehdessä

ilmoitettu tarkkuus on 1,5 %. Datalehden kaavan mukaan laskettu tarkkuus huonelämpötilassa on noin 0,2 °C. Antureissa oli kuitenkin keskinäisiä selviä eroja, joten käyttöön valittiin lähimpänä toisiaan olevat anturit.

Lämpötila lasketaan lineaarisesti interpoloiden ohjelmaan tehdystä taulukosta. Taulukon arvot on laskettu lämpötila-anturin datalehden antamista resistanssiarvoista määrätyissä lämpötiloissa. Taulukossa on 32 arvoa, jolloin AD-muuntimelta saatu arvo voidaan jakaa 32:lla. Ohjelma hakee taulukosta kyseisen lämpötilan ja seuraavan lämpötilan. Sen jälkeen estimoidaan lämpötila kyseiseltä väliltä olettaen lämpötilan käyttäytyen lineaarisesti. Todellisuudessa lämpötilan käyttäytyminen resistanssin suhteen NTC-vastuksilla ei ole lineaarinen, mutta virhe on olemattoman pieni verrattuna muihin virheisiin.

Summautuvien virheiden ja selkeyden vuoksi lämpötila esitetään asteen tarkkuudella. Antureiden jännitearvot luetaan AD-muuntimelta jokainen erikseen. AD-muuntimen suorittama muunnos kestää useamman ohjelmasyklin, jolloin sen valmistumista ei voida jäädä odottamaan. Jollei uutta tulosta ole tullut, niin käytetään vanhoja arvoja. Lämpötila lasketaan uuden ja vanhan arvon keskiarvona, jolloin muutosnopeus hidastuu hieman, mutta virhemittaukset eivät aiheuta suurta heiluntaa lämpötilassa.

Lämpötilat muutetaan lukuarvosta tekstiksi, jotta ne voidaan lähettää Raspberry Pi:lle. Lämpötilalle on varattu kahdeksan bittiä ja nopeudelle kaksitoista bittiä. Vaikka lämpötila-arvo ei vaatisi kahdeksaa bittiä, niin se muutetaan 8 bittiseksi lisäämällä nollia luvun eteen. Yhtenäisen viestin pituuden ansiosta tiedetään aina, mistä kohtaa viestiä voidaan kukin arvo lukea.

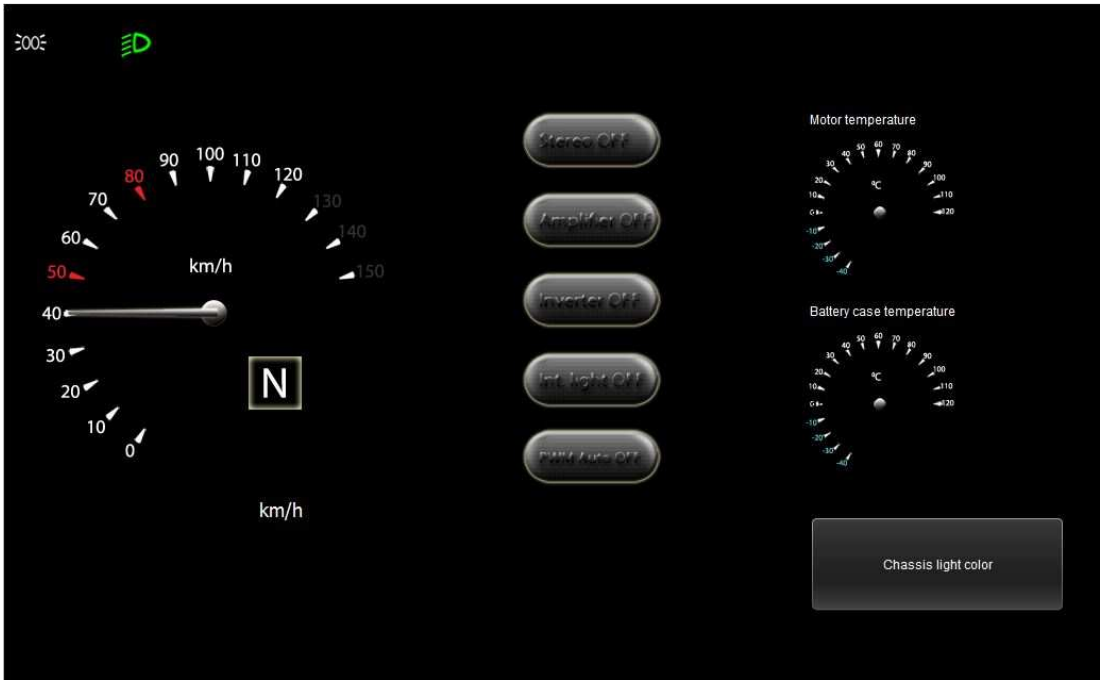
Nopeusanturilta tulee ON/OFF pulssitietoa. Joka kerran kun magneetti ohittaa Hall-anturin, niin Arduino saa pulssin, jonka se tunnistaa laskevasta reunasta. Nopeusanturin sisääntulo on normaalitilanteessa vedetty käyttöjännitteeseen ylösvetovastuksella. Hall-anturi kytkee linjan maahan, kun sen lähellä on riittävän suuri magneettikenttä.

Nopeusanturin pulssitiedot lasketaan keskeytysohjelmassa. Keskeytysohjelma ei tee mitään muuta kuin laskee pulsseja yhteen. Varsinainen laskenta suoritetaan muualla

koodissa ohjelman normaalin suorituksen aikana. Keskeytysohjelman aikana ohjelma lopettaa muun suorittamisen ja suorittaa vain keskeytysohjelmaa. Keskeytysohjelman on oltava mahdollisimman nopea, jotta muun ohjelman suoritus ei häiriinny. Arduino laskee vain pulssimäärän tietyssä ajassa ja muuttaa nopeuden lähetettävään muotoon. Varsinainen nopeuden laskenta suoritetaan Raspberry Pi:ssä, koska se on huomattavasti parempi suorittamaan liukulukulaskentaa.

3.5 Käyttöliittymä Raspberry Pi:llä

Käyttöliittymä on pyritty tekemään mahdollisimman selkeäksi käyttää ja katsoa. Kuvassa 3.2 on Netbeans-ohjelmasta otettu kuvakaappaus kosketusnäytön käyttöliittymästä. Kuvasta puuttuvat lämpötilojen viisarinäyttöjen viisarit ja digitaalinäytöt. Parkkivalojen ja lähivalojen symbolit ovat näkyvissä. Vasen osa ruudusta on varattu nopeusmittarille. Nopeus näytetään viisarinäytöllä ja digitaalisena. Viisarinäyttö näyttää maksimissaan 150 km/h, mutta digitaalinäytöllä näkyvät suuremmatkin luvut. Suurin nopeus, jolla autoa voi Suomessa ajaa, on 120 km/h, joten sitä suuremmat olevat luvut ovat harmaampia. Havainnollisuuden vuoksi 50 km/h ja 80 km/h ovat punaiset. Nopeusmittarissa näkyy myös valittu vaihde.



Kuva 3.3. Kosketusnäytön käyttöliittymä.

Nopeus voidaan näyttää tarkasti, mutta nopeuden tarkkuus riippuu vahvasti nopeusanturista. Nopeuden tarkkuus riippuu siitä kuinka monta kertaa sekunnissa magneetti kulkee anturin ohitse. Jos käytetään yhtä magneettia ja se ohittaa anturin kerran renkaan kierroksessa, niin nopeuden tarkkuus on vain noin 5 km/h. Tarkkuutta voidaan parantaa asentamalla useita magneetteja tai asentamalla anturi nopeammin pyörivään akseliin.

Nopeus tulee pulssimääränä viimeisen kahden sekunnin aikana. Nopeus muutetaan kilometreiksi tunnissa ja ilmoitetaan kolmen viimeisen mittauksen keskiarvona. Viisarinäytön viisaria kierretään oikeaan kohtaan näytössä. Viisarin liike olisi ollut helppo toteuttaa JavaFX:ssä animaatiolla, mutta Swingissä se joudutaan toteuttamaan monimutkaisemmin. Viisari joudutaan mallintamaan uudelleen, jotta se voidaan piirtää eri kulmaan.

Keskeltä näyttöä voidaan ohjata stereot, vahvistimet, invertteri, sisävalo ja alustavalon automaattitila päälle. Painikkeet on tehty photoshop-ohjelmalla ja ne muuttuvat harmaasta vihreiksi, kun ne kytketään päälle.

Näytön oikeassa laidassa on akkukotelon lämpötilan ja moottorin ulkolämpötilan näytöt. Molemmissa on samankaltainen viisarinäyttö ja digitaalinäyttö kuin nopeusmittarissa. Viisarinäytön negatiiviset lämpötilat ovat sinertäviä ja muut valkoisia. Viisarinäyttö näyttää arvot -40 $^{\circ}$C – 120 $^{\circ}$C. Digitaalinäytöltä voidaan lukea yli asteikon menevät arvot.

Näytön oikean laidan alaosassa on alustavalojen värin säätö. Painike avaa ohjelman, jossa on useita mahdollisuuksia värin valintaan. Väri voidaan valita esimerkiksi värikartalta tai liukukytkinten avulla säätää RGB-väriarvo. Kun sopiva väri on valittu, muuttuvat painike ja alustavalot valitun väriseksi.

Näytön yläosassa näkyy esimerkiksi valojen ja tuulilasinpyyhkimien päällä olotiedot perinteisinä symboleina. Varoitusvalojen symbolit ovat näytön alareunassa. Ne tunnistaa punaisista väreistä. Ohjelma lukee kaikki symbolit ja viisarinäytöt yhdestä kuvakansiosta. Kuvat voidaan vaihtaa mihin tahansa samassa png-muodossa olevaan samankokoiseen kuvaan.

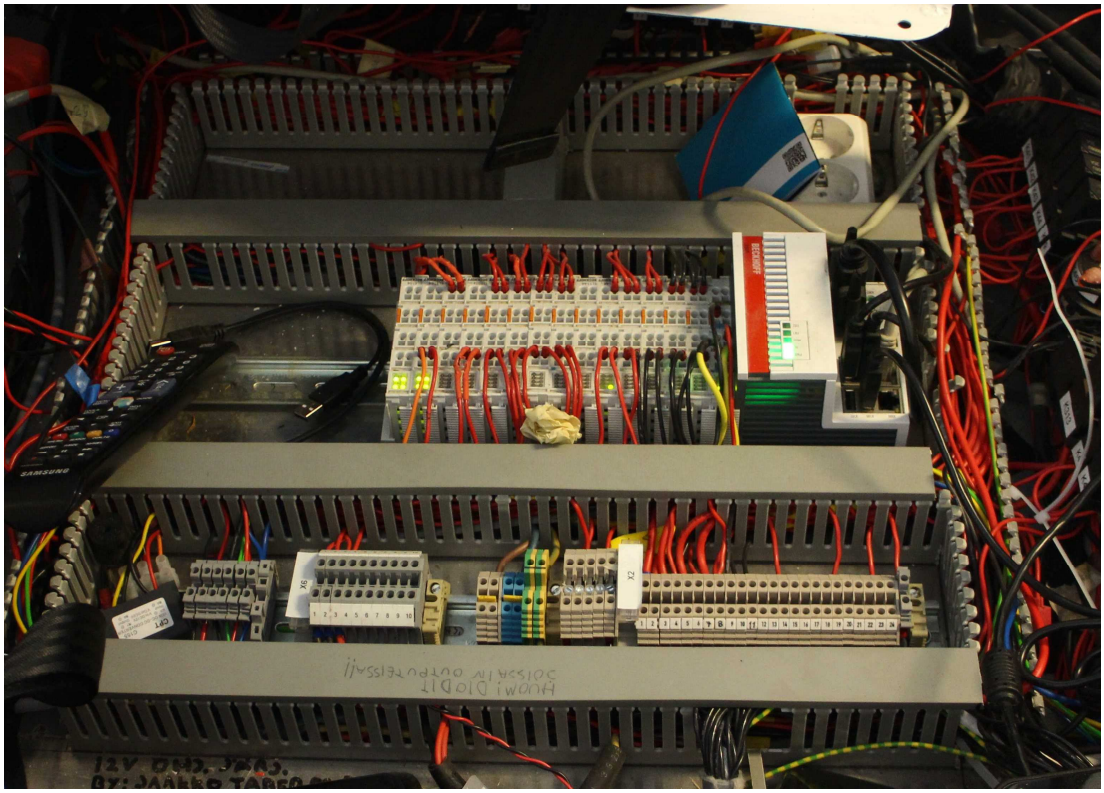
Raspberry Pi:n käynnistymistä on pyritty nopeuttamaan lukuisilla seikoilla. SD-muistikortiksi on valittu 10-luokan nopea muistikortti. Raspberry Pi:n prosessoria ja näytönohjainta on lievästi ylikellotettu. Käynnistymisen ajalta tulostus ruudulle on estetty ja joitain käyttämättömiä toimintoja on estetty.

4 ASENNUS, TESTAUS JA KEHITYSEHDOTUKSET

4.1 Asennus ja testaus

Kuvassa 4.1 on sähköauton alkuperäinen logiikkaohjaus. Logiikka sijaitsee apukuljettajan penkin alapuolella. Sähköauton ajomoottori hajosi juuri ennen kuin oli tarkoitus aloittaa järjestelmän asennus. Hajonnut moottori ei kuitenkaan haitannut asennusta ja testaustyötä. Nopeusanturin testaus ja moottorin ulkolämpötila-anturin asennus olivat ainoat, jotka hajonnut moottori esti.

Piirilevy kalustettiin Satakunnan ammattikorkeakoulun tiloissa, koska siellä oli juotoskaasujen suodatuslaitteisto. Juotoskaasujen hengittäminen ei ole terveellistä ja käytetty lyijyllinen tina on myrkyllistä (Brindley 2005, 108). Juotostina oli 0,7mm paksua fluxia (juoksutetta) sisältävää lyijyllistä tina-lyijy seosta.

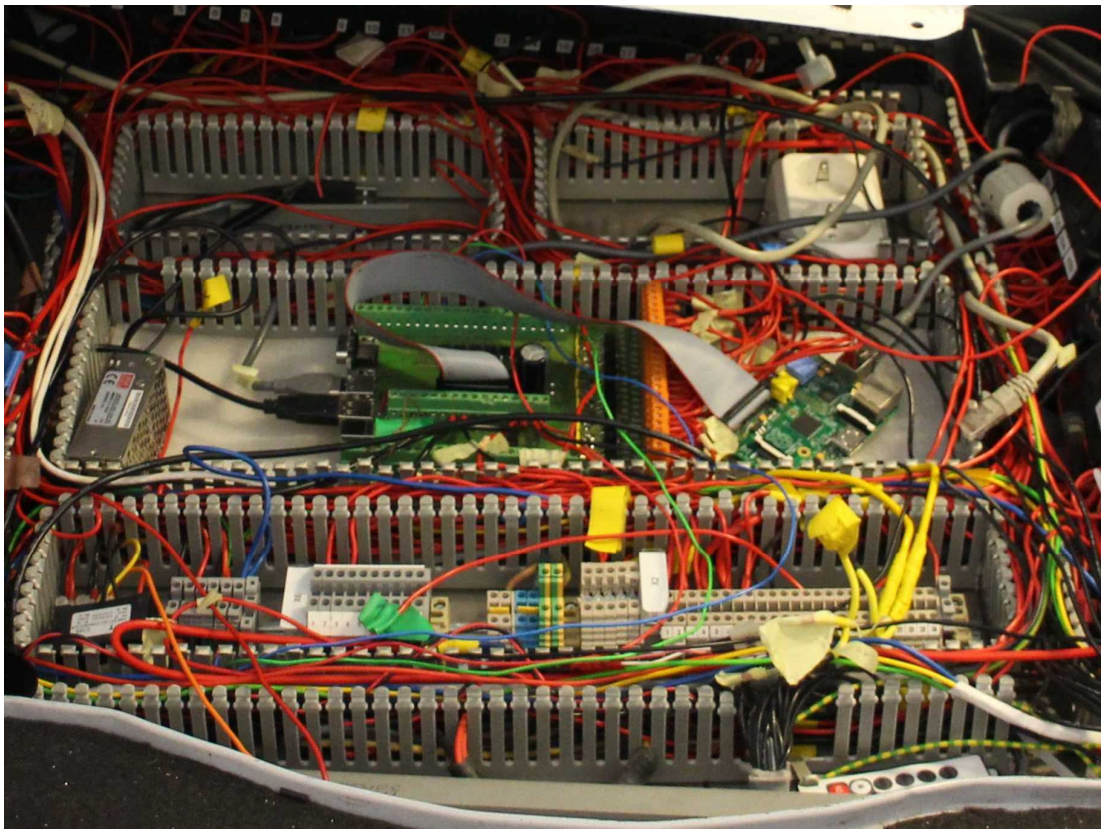


Kuva 4.1. Alkuperäinen logiikkaohjaus.

Piirilevyn toimintaa testattiin käyttämällä tietokoneen virtalähdettä ja ohjaamalla 5W autolamppua. Kaikkien ulos- ja sisääntulojen toiminta testattiin ennen asennusta sähköautoon.

Ennen varsinaista asennusta kokeiltiin mosfettien toimintaa ja havaittiin, että niiden virranläpäisykyky ei riittänyt ohjaamaan sähköauton releitä käytetyllä ohjausjännitteellä. Uudet mosfetit tilattiin ja asennettiin piirilevylle. Niiden toiminta testattiin ja todettiin toimiviksi.

Kuvassa 4.2 keskivaiheilla näkyy uusi ohjausjärjestelmä. Kuvassa vasemmalla on jännitemuunnin, keskellä I/O-kortti ja oikealla Raspberry Pi. Arduino on I/O-kortin alla. Kuva on otettu testausvaiheessa, jolloin johtimia ei ole vielä siistitty.



Kuva 4.2. Uusi ohjausjärjestelmä.

Autossa olleesta Beckhoff:n logiikasta irrotettiin johtimet ja nimettiin jokainen maalarinteipillä. Kunkin kortin datalehti haettiin internetistä, pääteltiin kortin funktio ja nimettiin johtimet Taberman (2004) opinnäytetyn liitteiden avulla. Kortteja oli neljää erilaista: käyttöjännitekortti, digitaalinen sisääntulokortti, analoginen sisääntulokortti ja ulostulokortti.

DIN-kisko poistettiin ja maadoitetun aluslevyn pinta pinnoitettiin teipillä. Myöhemmin päälle lisättiin vaahtomuovia, koska teippi ei kestänyt teräviä komponenttien jalkoja. Suora kontakti maadoitettuun levyyn vaurioittaa helposti jännitteistä laitetta riippuen siitä mikä sähköä johtava kohta osuu levyyn. Arduino I/O-kortteineen asennettiin samaan kohtaan, jossa Beckhoff:n logiikka oli ollut. Raspberry Pi ja 5V jännitemuuntaja sijoitettiin niiden viereen.

Osa alkuperäisistä johtimista olivat liian lyhyitä. Niiden vaihtaminen olisi edellyttänyt uudelleen johdottamista kojetaululta etupenkkien alle, joten niitä pidennettiin. Pidennys suoritettiin kuorimalla noin 10 mm johtojen päistä, asettamalla päät osittain toistensa sisään ja juottamalla toisiinsa kiinni. Suoja oikosuluilta saatiin suojaamalla liitoskohta kutistesukalla. Kuumailmapuhaltimen puutteesta johtuen kutistesukka lämmitettiin kolvin kärjellä nopeasti sivellen. Puristusliittimiä ei ollut välittömästi saatavilla, joten uusien johtojen pää kierrettiin ja laitettiin ruuviliittimeen kiinni.

Akkukotelon lämpötila-anturin johto vedettiin kuljettajan penkin takaa 12V akun koteloon. Anturi kiinnitettiin kotelon seinään teipillä. Moottorin ulkolämpötila-anturi vedettiin moottorin läheisyyteen. Anturi voidaan kiinnittää moottorin ulkopintaan moottorin vaihdon yhteydessä.

Sähköauton matkustamon kauko-ohjainkeskus aiheutti usean mosfetin vaurioitumisen. Keskuksesta voitiin ohjata muun muassa sisävalo, stereot, invertteri ja alustavalot päälle. Keskuksen oli tarkoitus toimia logiikkaojauksen rinnalla. Silloin kyseiset ohjaukset voitaisiin kytkeä päälle joko keskuksesta tai kosketusnäytöltä logiikan kautta. Keskuksen kytkinten toiset asennot olivat kytketty kaikki yhteen ja maihin. Sen pääkytkin ei vaikuttanut toimintaan mitenkään.

Kun kytkimet olivat pois päältä ja siten yhdessä maihin ja logiikasta ohjattiin rele päälle, niin 12V kulki suoraan maihin ja rikkoi mosfetin. Nopeampi sulake olisi estänyt mosfettien rikkoutumisen. Oikosulussa mosfet toimi sulakkeen tavoin ja sulki yhteyden maahan hajotessaan. Sähkökuviin oli piirretty diodit suojaamaan lähtöjä, mutta todellisuudessa niitä ei ollut kytketty suunnitellusti. Jos diodit olisi kytketty sähkökuvien mukaisesti, kytkentä olisi toiminut.

Ongelma korjattiin poistamalla kytkinten toinen asento kokonaan. Keskuksen pääkytkimen johto poistettiin, jolloin kytkin ei tee mitään. Ohjauskytkimet toimivat silloin käytettäessä eivätkä aiheuta häiriötä logiikkaohjauksen kanssa. Koska ohjaukset on kytketty rinnan, niin ohjaus joko kauko-ohjauskeskuksesta tai kosketusnäytöltä kytkee ohjattavan releen päälle.

Yhden oikosulkutilanteen aiheutti rikkoutunut USB-liitin Raspberry Pi:ssä. Toisen USB-liittimen muovinen keskiosa oli irronnut, jolloin liittimen metallihakaset osuivat sen kuoreen aiheuttaen oikosulun. Raspberry Pi:n ja kosketusnäytön virransyöttö oli kytketty yhteen ohjattavan mosfetin jälkeen. Niiden virta syötettiin jännitemuuntajan kautta. Koska jännitemuuntajan tehonsyöttökyky on rajattu, ei oikosulku vaurioittanut laitteita. Rikkoutuneen USB-liittimen metallihakaset taivutettiin pois kontaktista metallikuoresta ja kosketusnäyttö kytkettiin ehjään USB-liittimeen.

Kosketusnäyttöä varten hankittiin USB-jatkokaapelit. Jatkokaapelien kanssa USB-johdon pituus on noin kolme metriä. Niiden kanssa kuitenkin havaittiin, että kosketusnäyttö lakkaa hetken kuluttua toimimasta. Ilmeisesti USB-kaapeliin indusoituu niin paljon häiriötä, että se sekoittaa kosketusnäytön toiminnan. Toiminta palautuu normaaliksi vasta virran katkaisun jälkeen. Sama häiriö ilmeni, kun kosketusnäytön USB-kaapelin kytki kulkemaan I/O-kortin liitinten läpi. Vaikka datajohtojen pituus piirilevyllä on vain muutaman kymmenen milliä, niin näyttö vain vilkkui.

Alustavaloja tutkittaessa huomattiin, että sinisen värin paluujohto oli oikosulussa maahan kokoajan. Koska paluujohto maahan kytkettäessä ledit syttyvät normaalisti, ei se vaurioita valoja. RGB-värisävyt ovat kuitenkin sekaisin, koska siniset ledit ovat aina päällä, kun alustavalojen päävirtakytkin on päällä. Vika paikallistettiin oikean puolen ledinauhaan, jossa kiinnitysteippi, joka toimii myös suojana, oli huonosti pai-

koitettu. Se oli noin millin väärässä kohtaa jättäen kupariset liitoskohdat paljaiksi. Kun ledinauha oli kiinnitettynä auton runkoon, niin sinisen värin paluujohto oli kokoajan maissa.

Molemmilla sivuilla olevat ledinauhat otettiin irti ja niiden pohjaan laitettiin kaksipuoleinen teippi. Nauhat kiinnitettiin takaisin paikalleen ja testattiin toimivuus. Edessä ja takana olevat nauhat on kiinnitetty maalattuun pintaan, joka ei ole sähköä johtava, jolloin ongelmaa ei ole. Alustavalojen alkuperäinen kauko-ohjaimella toimiva säädin ohitettiin ja johtimet kytkettiin suoraan I/O-korttiin. Valoja voidaan ohjata kosketusnäytön avulla.

4.2 Kehitysehdotukset

Levylle on tehty kaistanpäästösuodin, johon syötetään sisään audiosignaalia Arduinolle. Audiosignaalia lukemalla voidaan alustavaloja vilkuttaa musiikin tahtiin. Aina kun signaalitaso sisääntulossa nousee tietyn tason yli, niin valot syttyvät. Vaihtoehtoisesti signaalin voimakkuuden mukaan voidaan muuttaa väriä tai kirkkautta. Signaalia luetaan AD-muuntimella.

Latausohjainta voidaan lukea USB-portin kautta. Arduino Mega ADK:ssa on USB-portti, joka voi toimia orjana tai isäntälaitteena. Latausohjaimen viestiliikennettä on siten ainakin teoriassa mahdollista lukea. Esimerkiksi akuston lämpötilat ja varaustasot voidaan lukea ja näyttää kosketusnäytöllä.

I/O-kortilla olevat ulostulot pystyvät ohjaamaan suuriakin virtoja, joten niillä voidaan ohjata laitteita suoraan. Releet voidaan silloin jättää pois välistä. Mosfet ei kuluta juuri ollenkaan virtaa, mutta jokainen rele vie 1,4 W tehoa jatkuvasti päällä ollessaan. Tällä hetkellä kaikki virta tulee yhden sulakkeen kautta. Jos laitteita ohjataan suoraan, on käytettävä niiden omilta sulakkeilta tulevaa jännitettä ja kytkettävä se kyseisen ulostulon a-liittimeen.

Kosketusnäytön grafiikat voidaan muuttaa halutun laisiksi. Jos halutaan muuttaa jonkin symbolin grafiikka, riittää pelkkä kuvatiedoston ylikirjoitus kuvakansiossa. Jos halutaan muuttaa symboleiden paikkaa tai kokoa, niin silloin käytetään Netbeans-ohjelmaa. Näyttö on hieman liian iso vanhan näytön paikalle, joten aukkoa on suurennettava. Koska pitkiä USB-johtoja ei voitu käyttää, niin näytön USB-johto on vedettävä suorempaa reittiä, mikä vaatii johdon suojaamista.

Nopeusanturi pitää kytkeä johonkin pyörivään kappaleeseen. Mitä enemmän magneetteja saadaan kulkemaan Hall-anturin ohitse jokaisella kierroksella, sitä parempi tarkkuus nopeudelle saadaan. Yhdellä anturilla tarkkuus on vain noin 5 km/h. Moottorin lämpötila-anturi on myös kiinnittämättä, koska moottori tullaan vaihtamaan.

Tuulilasin pyyhkimen hitaan asennon kytkin ratissa on rikki. Se ei anna lainkaan signaalia. Raspberry Pi:tä ja jännitemuuntajaa ei ole kiinnitetty lainkaan. Arduino ja I/O-kortti pysyvät paikallaan tulo- ja lähtöjohtojen avulla. 24V jännitesyöttöä ei tarvita mihinkään. Sen voi poistaa. Akkukotelon lämpötila-anturi voidaan sijoittaa helposti muualle. Se on kiinnitetty teipillä kotelon seinään.

5 LOPPUPÄÄTELMÄT

Työn ansiosta linux-osaaminen vahvistui huomattavasti. Käyttöliittymän tekeminen toi uutta osaamista Java-kieleen. Aiemmasta melko laajasta Java-osaamisesta oli työn tekemiseen paljon hyötyä. Kuitenkin käyttöliittymän tekeminen, symboleiden piirtäminen mukaan lukien, vei useita viikkoja. C-koodi, jota Arduinossa käytetään, vei vain noin yhden työpäivän verran. Arduinon koodikirjastot nopeuttavat koodin kirjoittamista huomattavasti. Olen tehnyt päätyökseni suurimmaksi osaksi C-kielellä koodaamista suoraan Atmelin mikrokontrollereille parin vuoden ajan. Arduinon koodin pituus on samaa luokkaa käyttöliittymän koodin kanssa. I/O-kortin piirilevyn suunnittelu sujui ongelmitta aiemman kokemuksen vuoksi.

Testien mukaan Arduinon ohjelma toimii halutusti. Arduino ohjaa auton järjestelmiä ja lukee tietoa antureilta. Kosketusnäyttö on selkeä ja toimii riittävän nopeasti. Arduino käynnistyy välittömästi ja kosketusnäytön käynnistysaika on siedettävä. Koska kosketusnäytön kiinnittäminen auton kojetauluun vaatii aukon suurentamista, jäi se tekemättä, sillä aukon tekeminen esteettisesti vaatii enemmän taitoa kuin opinnäytetyön tekijältä löytyy. Lisäksi USB-johto on suojattava.

LÄHTEET

Taberman, J. 2014. Satakunnan ammattikorkeakoulun sähköautoprojektin 12 voltin järjestelmä. AMK-opinnäytetyö. Satakunnan ammattikorkeakoulu.

Myntti, M. 2014. Satakunnan ammattikorkeakoulun sähköautoprojektin moottori, moottoriohjain ja latauksenohjaus. AMK-opinnäytetyö. Satakunnan ammattikorkeakoulu.

Arduino www-sivut. Viitattu 15.9.2014. http://www.arduino.cc

Raspberry Pi www-sivut. Viitattu 15.9.2014. http://www.raspberrypi.org/

Mimo www-sivut. Viitattu 25.9.2014. http://www.mimomonitors.com

Brindley, K. 2005. Starting electronics construction. Newnes.

LIITE 1

RASPBERRY PI:N MUISTIKORTIN ASENNUSOHJE

Tarvittavat tiedostot ja ohjelmat ovat saatavissa erikseen. Kansiossa olevat tiedostot siirretään muistikortille samoihin kansioihin korvaten mahdolliset aiemmat versiot.

1. Alusta SD-muistikortti tietokoneella
2. Asenna wheezy-raspbian linux
3. Siirrä ja korvaa tiedostot muistikortilla kansiosta: RaspberryPi-Mimo-add before first boot
4. Kytke Raspberry Pi HDMI-kaapelilla mihin tahansa näyttöön. USB-näyttö ei käy, koska HDMI on oletusarvoisesti käytössä ensimmäisen käynnistyksen yhteydessä. Muuta asetukset: memory split: 256/256Mb, extend memorycard, boot to desktop. Memory split antaa näytönohjaimelle lisämuistia, Extend memorycard vapauttaa koko muistikortin kapasiteetin käyttöön. Boot to desktop käynnistää linuxin graafisen käyttöliittymän. Uudelleenkäynnistyksen jälkeen käytetään Mimo USB-näyttöä.
5. Siirrä tietokoneella muistikortille java kansiosta: RaspberryPi-Java. Siirrä java Raspberry Pi:n kotikansioon: /home/pi
6. Käynnistä Raspberry Pi. Käynnistä työpöydältä löytyvä konsoli.
7. Luo konsolilla uusi kansio javalle komennolla: **sudo mkdir -p -v /opt/java**
8. Pura java: **tar xvzf ~/jdk-8u6-linux-arm-vfp-hflt.tar.gz**
9. Siirrä java uuteen kansioon: **sudo mv -v ~/jdk1.8.6 /opt/java**
10. Poista alkuperäinen pakattu javatiedosto: **rm ~/jdk-8u6-linux-arm-vfp-hflt.tar.gz**
11. Kerrotaan javalle mihin JVM on asennettu: **sudo update-alternatives --install "/usr/bin/java" "java" "/opt/java/jdk1.8.6/bin/java" 1**
12. Kerrotaan linuxille mitä JDK:ta käytetään: **sudo update-alternatives --set java /opt/java/jdk1.8.6/bin/java**
13. Testaa onnistuiko javan asennus: **java –version**. Version pitäisi olla 1.8.6
14. Siirrä ja korvaa tietokoneella tiedostot muistikortilla kansiosta: RaspberryPi-other-add after first boot. Tiedostot enabloivat UART-liikennöinnin ja asettavat java-home kansiot.
15. Siirrä varsinainen ohjelma PI_0_1.jar kansioon /home/pi. Suositeltavaa on siirtää koko PI_0_1 –kansio kansioon /home/pi, jolloin ohjelman koko osoite on oikein: /home/pi/PI_0_1/dist/PI_0_1.jar

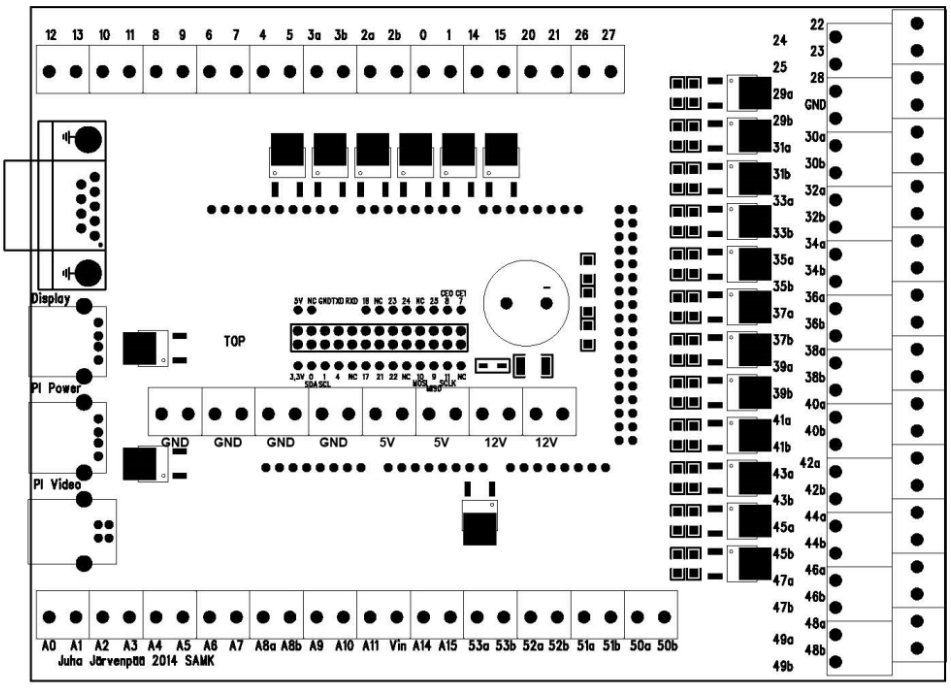
16. Tiedosto lightdm.conf on paras muuttaa suoraan Rapberry Pi:n konsolilla. Anna komento: **sudo nano /etc/lightdm/lightdm.conf.** Lisää kohtaan: **[SeatDefaults]** seuraava rivi: **xserver-command=X -s 0 dpms –nocursor.** Tämä poistaa näytönsäästäjän ja hiiren.

17. Siirrä tietokoneella muistikortille tiedosto: **.profile** kansiosta: RaspberryPi last files/home/pi. Tiedosto korvaa .profile –tiedoston /home/pi –kansiosta. Tiedosto saa aikaan automaattikäynnistyksen kokoruututilassa. Jos kosketusnäytön ruutu välkkyy ja näyttää vain mustaa, poista rivi: **startx .profile** –tiedostosta.

18. Jos halutaan nopeuttaa käynnistymistä, niin siirretään muistikortille tietokoneella tiedostot kansiosta: RaspberryPi-faster-boot

LIITE 2

I/O PIIRILEVY



LIITE 3

SISÄÄN- JA ULOSTULOT

Ulostuloissa a on kytketty 12V jännitteeseen ja b releelle.

Nro:	**Nimike Input**	**Arduino Pin**
1	Äänimerkki	27
2	Suuntamerkki,vasen	28
3	Suuntamerkki,oikea	A6
4	Valokytkin kaukoajovalot	A7
5	Tuulilasin pyyhin asento 1	23
6	Tuulilasin pyyhin asento 2	24
7	Pesuri	25
8	Valokytkin parkkivalot	26
9	Valokytkin lähiajovalot	8
10	Valokytkin Päiväajovalot	9
11	Hätävilkut päällä-tieto	10
12	Eteenpäin	11
13	”Vapaa” vaihde	12
14	Peruutus	13
15	Hätäseis	14
16	Johto seinässä	15
17	Etuovien aukiolotieto	A14
18	Jarrun toimintahäiriö	A15
19	Jarrun tilatieto	A11
20	Moottorinohjaimen toimintahäiriö	22
21	Nopeusanturi (takometri)	21
22	Moottorin lämpötila (sisä)	A0
23	Moottorin lämpötila (pinta)	A1
24	Akkulaatikon lämpötila	A2
25	Varalla	A3
26	Audio in	A4
27	Varalla	20
28	Varalla	0
29	Varalla	1
30	Varalla analogitulo	A9
31	"OFF"-painike	A10
32	Start-painike	A5

Nro:	**Nimike Output**	**Arduino Pin**
1	Äänimerkki	30
2	Varoitusääni sisälle	52
3	Pysäköintivalot	51
4	Lähiajovalot	32
5	Kaukoajovalot	49
6	Päiväajovalot	48
7	Suuntamerkki,vasen	47
8	Suuntamerkki,oikea	46
9	Takavalot+rekisterikilven valo	45
10	Jarruvalot	44
11	Peruutusvalo	43
12	Sisävalo,matkustamo	42
13	Tuulilasin pyyhin asento 1	41
14	Tuulilasin pyyhin asento 2	40
15	Pesuri	39
16	Kiertovesipumppu+puhallin	38
17	Audiovahvistimien heräte	37
18	Varalla	36
19	Varalla	50
20	Varalla	35
21	Varalla	53
22	Ajonesto (ohjelmallinen)	29
23	Invertterin ohjaus	31
24	Akkulaatikon puhallin	34
25	Moottoripuhallin	33
26	PWM ulostulo red	7
27	PWM ulostulo green	6
28	PWM ulostulo blue	5
29	PWM ulostulo	4
30	Varalla PWM ulostulo	3
31	Varalla PWM ulostulo	2
32	Varalla ulostulo	A8