SAVONIA

OPINNÄYTETYÖ - **AMMATTIKORKEAKOULUTUTKINTO**
TEKNIIKAN JA LIIKENTEEN ALA

AJOSIMULAATTORIN ELEKTRONIIKAN SOVELTAMINEN JA ASENNUS

TEKIJÄ/T: Arto Pirinen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Elektroniikan koulutusohjelma	
Työn tekijä(t) Arto Jalmari Pirinen	
Työn nimi Ajosimulaattorin elektroniikan soveltaminen ja asennus	
Päiväys 12.3.2014	Sivumäärä/Liitteet 23
Ohjaaja(t) yliopettaja Väinö Maksimainen	
Toimeksiantaja/Yhteistyökumppani(t) Lekasteel Oy	

Tiivistelmä

Tämän opinnäytetyön tarkoituksena oli suunnitella ohjauslaitteiden toteutusta alustavaan prototyyppiin ajosimulaattoria varten. Alustavaan selvitykseen kuului kustannusarvion tekeminen sekä hankittavat laitteet sekä kehitysalustan valinta. Kustannusarvion perusteella valittiin kehitysalustaksi Arduino Mega 2560, sekä prototyypin kehitystä tukeva Logitech G25 -ratti/poljinjärjestelmä. Peliohjain hankittiin sillä perusteella, että auton oma ratti saadaan kiinnitettyä peliohjaimeen. Keskeisenä ominaisuutena kehitysalustassa oli AD-signaalien käsittely, sekä analogisen jännitteen muuntaminen digitaaliseen muotoon.

Työ toteutettiin Tuusniemeläiselle Lekasteel yritykselle, joka aiemmin on keskittynyt metallipajatoimintaan sekä ajoneuvojen huoltotöihin, mutta laajentanut lähivuosina simulaattorien kehitykseen ja prototyyppeihin. Aikaisempana projektina heillä on ollut moottoripyörän dynamometri, joka on jatkokehitetty toimivaksi moottoripyöräsimulaattoriksi.

Opinnäytetyön tuloksena ohjauslaitteiden kytkennät saatiin toimiviksi ja havainnoitua Arduino Mega 2560 -kehitysalusalla, joka pohjautuu Atmelin Atmega2560- mikroprosessoriin. Analogisten signaalien käsittelystä siirryttiin digitaaliporttien ylösvetovastusten hyödyntämiseen kehitysvaiheessa. Lisäksi tietokoneelle tehtiin käyttöliittymä, jolla seurataan ohjauslaitteiden tiloja. Myöskin kehitysalustalle tehtiin ohjelma, joka kerää ohjauslaitteiden asennot ja lähettää ne eteenpäin.

Avainsanat
Simulaattori, jännite, AD-muunnin, Arduino, ylösvetovastus

julkinen

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Electronic Engineering			
Author(s) Arto Jalmari Pirinen			
Title of Thesis Embedding and installing electronics for a driving simulator			
Date	March 12, 2014	Pages/Appendices	23
Supervisor(s) Mr. Väinö Maksimainen, Principal Lecturer			
Client Organization /Partners Lekasteel Ltd.			

Abstract

The purpose of this final year project was to design implementation of the controlling devices for a driving simulator prototype. The preliminary study included a cost estimation, obtaining the required equipment and choosing the development board.

The work was comissioned for a metal company from Tuusniemi called Lekasteel. Lekasteel has earlier focused on machining and metal workshop but has expanded to development of simulators and different prototypes. Previous prototypes made by Lekasteel include a motorbike dynamometer which was further developed to a working motorbike simulator. Arduino Mega 2560 was chosen to be the development board based on the cost estimation and Logitech G27 game controller was purchased because of its supportive role to the project. The reason for purchasing G27 was that the original steering wheel of the car could be fixed to the G27 with an adapter. Essential properties in the development board were the capability to handle analog and digital signals and the conversion of analog voltage to digital form.

As a result of this project the wiring of controlling devices was working and these signals were observed with Arduino Mega 2560 which is based on Atmel's Atmega2560 embedded processor. Then, analog signal observing was changed to usage of digital pull-up resistors. After that, a user interface was created in computer for observing the control devices of the project. A program was made for the development board which observes the statuses of the control devices and sends those statuses to the computer.

Keywords
Simulation, voltage, AD -converter, Arduino, pull-up resistor, development board

public

ESIPUHE

Tein opinnäytetyöni vuosina 2013-2014. Työn ohella sain kokemusta sulautetun järjestelmän suunnittelusta ja sen tuomista haasteista. Kokemusta tuli niin ohjelmoinnista kuin laitteistostakin.

Haluan kiittää Lekasteelin Eero Lehtoa opinnäytetyöni mahdollistamisesta, Lekasteelin osaavaa henkilökuntaa ja opinnäytetyöni ohjaajaa yliopettaja Väinö Maksimaista sekä kaikkia muita tukijoukkoja, jotka ovat olleet mukana tukemassa opinnäytetyöni valmistumista.

Kuopiossa 12.3.2014

Arto Pirinen

SISÄLTÖ

LYHENTEET JA KÄSITTEET

AD-muunnin, ADC	Komponentti, joka muuntaa digitaalisen signaalin analogiseksi (ADC, analog-to-digital converter)
Simulaattori	Laite tai ohjelma, jolla jäljitellään todellista maailmaa
Jännite	Kahden pisteen potentiaaliero
Reaaliaikakäyttöjärjestelmä	Ohjelma tai käyttöjärjestelmä, joka toimii reaaliajassa
I/O-pinni	Input/output (sisään-/ulostulo) pinni, digitaalinen tai analoginen
Kellotaajuus	Ilmaisee kuinka monta tilanvaihdosta prosessori eli suoritin voi suorittaa sekunnissa
Flash-muisti	Puolijohdemuisti, voidaan sähköisesti tyhjentää ja uudelleen ohjelmoida
Mikrokontrolleri	Kehitysalustan prosessori
Kanttiaalto	Signaaliaalto, digitaalisena arvot ovat 0 tai 1
Arduino Mega 2560 R3	Työssä käytetty kehitysalusta varustettuna Atmega 2560 prosessorilla
Atmega 2560	Atmelin kehittämä mikrokontrolleri
Mega	Työssä käytetty Arduino Mega 2560 R3
USB	Universal Serial Bus, sarjaväyläarkitehtuuri, tekniikka jota käytetään laajalti kytkettäessä oheislaitteita tietokoneeseen
Sarjaliikenne	Sarjaliikenteen kautta tapahtuva kommunikointi kahden laitteen välillä
Protokolla	Tietyt säännöt, joiden mukaan toimitaan. Ohjelmoinnissa tarkoittaa tiettyjä parametreja joiden mukaan funktio tai ohjelma toimii.
Vastus	Elektroniikassa käytettävä passiivinen komponentti, rajoittaa virran kulkua virtapiirissä
Rattipoljinpaketti	Peliohjain, Logitech G27, koostuu ratista, polkimista ja vaihteistosta
Ylösvetovastus	Vastus, joka ajaa digitaalisen pinnin tilan 1:seen, kun pinniin ei ole kytketty katkaisijaa tai signaalia
Arduino	Työssä käytetty Arduinon oma ohjelmointiympäristö

1 JOHDANTO

Ajo-opetuksessa ja pelikäytössä realististen simulaattorien käyttö on yleistymässä jatkuvasti. Simulaatiolla voidaan jäljitellä oikeata tilannetta turvallisessa, tapaturmavapaassa ympäristössä. Simulaattorissa yleensä pyritään luomaan todentuntuinen tilanne esimerkiksi autoiluun tai moottoripyöräilyyn liittyen. Ohjauslaitteistossa pyritään luomaan oikean ajoneuvon kaltainen ympäristö.

Työ tehtiin Lekasteel Oy:lle. Lekasteel on alunperin metalli- ja huoltopaja, mutta sittemmin laajentanut elektroniikkaan, dynamometreihin sekä simulaattoriprototyyppeihin. Aiempaa simulaattorikokemusta Lekasteelillä on moottoripyöräsimulaattorissa, joka on jo tuotteistamisasteella. Työn lähtökohtana oli aloittaa projekti autosimulaattorin rakentamisesta, ja ensimmäinen vaihe on työssä tehty elektroniikan soveltaminen ja testaus tulevan simulaattorin autoaihion ohjauslaitteista.

Elektroniikan käyttö kohdistui ohjauslaitteiden havainnointiin. Ohjauslaitteisiin kuuluu auton tavallisimmat ohjauslaitteet: valoviiksi, päävalokatkaisija, pyyhkijäviiksi sekä käsijarru. Kaikki ohjauslaitteet saatiin havainnoitua Arduinolla ja näiden ohjauslaitteiden tilat luettua tietokoneella.

2 SUUNNITTELU

Lähtökohtana suunnittelulle oli kehittää reaaliaikainen mittaus simulaattorin hallintalaitteista, jotka olisivat mahdollisimman pitkälle auton omia ohjauslaitteita. Tavoitteena oli kehittää toimiva prototyyppi, joka voisi toimia tulevan simulaattorin pohjana. Auton ohjauslaitteistoon suunniteltiin johdettavaksi jännite, jonka vaihtelua kytkimillä voitaisiin mitata. Koko järjestelmä koostuu ohjaustlaitteista, kehitysalustasta, tietokoneesta sekä ohjelmistoista.

2.1 Reaaliaikajärjestelmän ominaisuudet

Sulautettu reaaliaikakäyttöjärjestelmä on simulaattorissa käytettävä ohjaus- ja mittausjärjestelmä. Järjestelmä koostuu signaaleja havannoivasta kehitysalustasta, kehitysalustan ohjelmasta ja tietokoneen puolella toimivasta käyttöliittymästä. Tietokoneen puolella käyttöliittymä on Windowsin konsolisovellus. Kommunikointi tietokoneen ja kehitysalustan välillä tapahtuu sarjaliikenteellä. Vaatimuksena ohjelmistolle oli riittävä reaaliaikaisuus, jottei ohjauslaitteiden havainnointiin tulisi kohtuutonta viivettä.

2.2 Käyttölaitteet

Alustavana ideana oli käyttää auton omia ohjauslaitteita simulaattorin prototyypin ohjaukseen. Ohjauslaitteisiin kuuluivat valojen pääkatkaisija, valo- sekä pyyhkijäviiksi, ratti ja käsijarru. Ratin kääntämisen osalta päädyttiin kuitenkin rattipoljinpaketin hankintaan, koska sen käyttö on prototyypin ohjelmallisen toteutuksen kannalta järkevämpi tulevaisuudessa ja se yksinkertaistaa laitteistoa huomattavasti alustavaa kehitystä varten. Laitteiston simulointiin Lekasteelille hankittiin peliohjain, Logitechin G27 rattipoljinpaketti. Hankinta tehtiin sillä perusteella, että lopullisen simulaation kannalta on helpompaa toteuttaa auton liikuttamiseen liittyvät toiminnot valmiilla paketilla.

Valinta kohdistui malliin G27, koska peliohjaimeen on mahdollista sovitinkappaleella sovittaa oma ratti, mikä tuo todentuntuisuutta ajamiseen. Kuitenkin haluttiin säilyttää jotain toiminnallisuutta auton omista laitteista, joten vilkkujen, valojen ja pyyhkijöiden käyttö säilytettiin auton laitteilla. Rattipoljinpakettiin, toisin sanoen peliohjaimeen, suunniteltiin adapteri, jotta peliohjaimen pienen ratin tilalle saataisiin aidomman tuntuinen ja isompi auton alkuperäinen ratti.

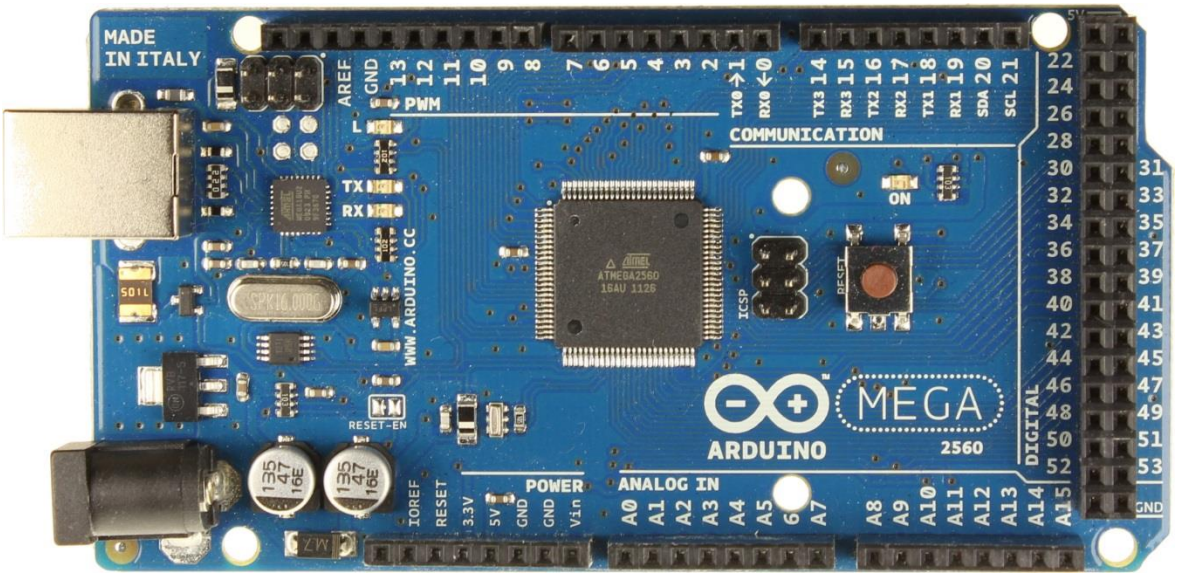
3 LAITTEISTO JA TOTEUTUS

3.1 Arduino ja Arduino Mega 2560 R3

Arduinolla on useita erilaisia kehitysalustoja avoimeen käyttöön. Erilaisten alustojen avulla on mahdollista toteuttaa eri vaativuusasteen prototyyppejä, aina yksinkertaisesta mittarista monimutkaiseen anturointiin. Erillistä ohjelmointiyhteyttä ei tarvita, koska useissa malleissa on USB-sarjamuunnin. Sarjaliikenne on toteutettu USB:n kautta, joten kortin lisäksi perusohjelmointiin tarvitaan vain USB-kaapeli. Arduinon ideana onkin helppo lähestyttävyys sulautetun elektroniikan ohjelmointiin. Arduinon verkkosivuilta löytyy kattavat tietokannat aina esimerkeistä teoriaan.

Käytettäväksi kehitysalustaksi valitsin mallin Arduino Mega 2560 R3, koska Arduinon ohjelmointi oli jo ennestään tuttua. Kortti on avoimeen lähdekoodiin perustuva mikrokontrollerialusta. Korttiin liittyy myös kehitysympäristö kortin ohjelmointia varten. Korttia voidaan käyttää erilaisten antureiden tai analogisten suureiden mittaukseen. Suurin osa korteista valmistetaan Italiassa, SmartProjects:lla.

Mega 2560 perustuu Atmega2560 mikrokontrolleriin. Megassa on 54 digitaalista I/O-pinniä ylösvetovastuksilla, 16 analogista sisääntuloa, neljä UARTia, 16 MHz kristallioskillaattori, USB liitäntä, virtaliitäntä, ICSP-liitäntä ja reset- nappi. Kortilla on myös Atmega16U2 USB-to-serialmuunnin. Tietokoneen ja alustan liikennöintiin tarvitaan vain USB-yhteys. USB-portista voidaan myös ottaa käyttöjännite. (Arduino 2013.)



Kuva 1. Arduino Mega 2560 R3 (Arduino 2013)

Seuraava lista on kooste Mega 2560 R3:n keskeisistä ominaisuuksista (Arduino 2013):

- Mikrokonttolleri ATmega 2560
- Käyttöjännite 5 volttia
- Digitaalisia I/O-pinnejä 54 kpl ylösvetovastuksilla
- Analogiset sisääntulot 16 kpl (ADC)
- Kellotaajuus 16 MHz
- Flash muistia 256 KB
- USB- sarjaliikenne

3.2 Mittaukset Megalla

3.2.1 Signaalien keruu

Auton ohjauslaitteiden käyttöjännitteeksi oletettiin 12 volttia. Megan sisääntulovirran maksimiarvo on 40 mA ja tämän takia mittauksiin jouduin mitoittamaan etuvastukset rajoittamaan sisääntulovirtaa estämään Megan vaurioitumisen. Alustavasti mitoitin vastuksen kymmenykseen maksimivirrasta, eli 4mA. Vastuksen arvo on 5500 ohmia. Tarvittava vastus laskettiin Ohmin lakia noudattaen kaavalla 1:

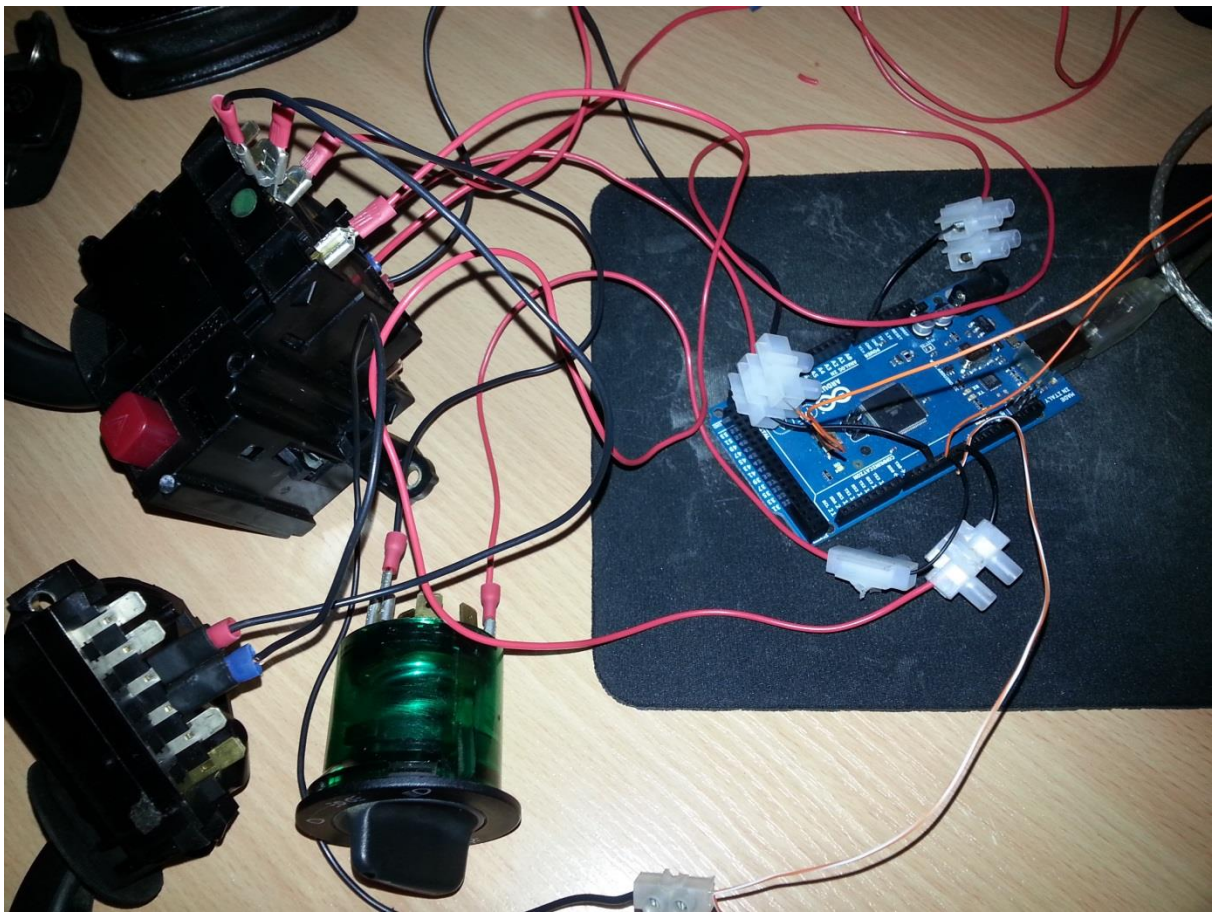
$$U = R * I => R = \frac{U}{I} \quad (1)$$

Kaavassa R on vastus, U on jännite ja I on virta.

Aluksi mittaukset tehtiin analogisesti syöttämällä jännitettä ohjauslaitteille. Kun ohjauslaite kytkettiin johtavaan tilaan, jännite mitattiin ADC:llä ja tulokset käsiteltiin Megan ohjelmalla. Mittauksia tehdessä kuitenkin ilmeni, että kun useita kytkentöjä tehtiin ADC:lle, syntyi häiriöitä kytkentöihin. Häiriöt vaikuttivat merkittävästi jännitteiden mittaustuloksiin. Erilaisista kytkennöistä huolimatta häiriöt jatkuivat. Siitä johtuen haettiin vaihtoehtoista ratkaisua ja päädyttiin siihen, että ohjauslaitteita voitaisiin käyttää katkaisijoina Megan ylösvetovastusten avulla. Näin ollen jännitettä ei tarvittaisi ollenkaan.

3.2.2 Megan ylösvetovastukset

Megassa on digitaalisissa I/O-pinneissä käytettävissä ylösvetovastukset. Ylösvetovastusten tehtävänä on pitää digitaalisen pinnin looginen tila arvossa 1, kun signaalia ei ole kytketty. Tällä varmistetaan, että pinni ei jää kellumaan ja tuota satunnaisia arvoja häiritsemään mittauksia. Kun pinniin kytketään signaali tai katkaisija kytketään päälle, pinnin looginen tila laskee nollaan. Tätä ominaisuutta hyödynnettiin mittauksissa. Kun katkaisija kytketään päälle, esim. vilkku vasemmalle, pinnin tila laskee nollaan ja se voidaan havaita ohjelmallisesti. Kaikki ohjauslaitteet ovat kytkettynä näihin digitaalisiin pinneihin, ja mittauksissa hyödynnetään ylösvetovastuksia. Megassa ylösvetovastukset ovat 20-50 kOhm. (Arduino 2013.)



Kuva 2. Testikytkentöjä Megalla (valokuva Arto Pirinen)

Kuvassa 2 nähdään kaikki ohjauslaitteet käsijarrun painiketta lukuunottamatta kytkettynä Megaan. Kytkennöissä hyödynnetään Megan ylösvetovastuksia. Näin ollen kytkennöissä voitiin hyödyntää ohjauslaitteiden katkaisijana toimimista, ja saatiin luotettavia ja häiriöttömiä mittauksia aikaiseksi. Lopullisessa toteutuksessa hyödynnettiin Megan ylösvetovastuksia onnistuneesti.

3.3 ATmega 2560

Atmega 2560:ssa on monipuolinen ja vähän virtaa kuluttava RISC-arkitehtuuriin perustuva mikrokontrolleri. Ulkoisiin ominaisuuksiin kuuluu erilaiset ajastimet ja laskurit, reaaliaikalaskuri, kahdeksan PWM moduloitua kanavaa, 16-kanavainen 10-bitin ADC, SPI-liitäntä, 4 USART ja bittiorientoitu 2-wire sarjaliikenne ja monia muita. (Atmel 2013.)

3.4 Auton ohjauslaitteet

Simulaattorin fyysisenä ympäristönä toimii Saab 9000 -henkilöauton ohjaamo. Ohjaamoon tullaan sijoittamaan prototyypin ohjauslaitteet peliohjaimineen. Suunnittelun mukaan peliohjaimelle tehtiin adapteri, johon Saabin alkuperäinen rattirunkorakenne saatiin asennettua. Käyttöaste auton omille ohjauslaitteille säilyi siis varsin korkeana. Rattirungossa on kiinni valo- ja pyyhkijäviiksi. Rattia käännettäessä vilkut menevät pois päältä tietyllä liikkeellä kuten oikeassakin autossa. Ohjauslaitteisiin liittyi melko paljon taustatyötä kytkentöjen selvittämiseen ja näiden kytkentöjen testaamiseen. Kytkennöistä ei ollut mitään varsinaista opasta, koska on kyse niin vanhasta autosta. Kytkentöjä ja laitteiden tietoja joutuikin selvittämään melko

lailla eri lähteistä, mutta lopuksi kuitenkin löytyi osia korjausoppaasta, jossa oli virheenetsintää varten viiksien kytkennät. Päävalokatkaisijan kytkennät oli helpompi selvittää, koska päävalokatkaisija toimii vain päälle/pois -katkaisijana valoille.

Kuva 3. Auton ohjauslaitteet paikoillaan (valokuva Arto Pirinen)

Kuvassa 3 nähdään auton valoviiksi ja pyyhkijäviiksi kiinnitettynä rattirunkoon. Kuvassa alhaalla näkyy peliohjaimen rattiosa, johon ohjausakseli on kiinnitetty. Ohjausakselin kiinnitys ja adapteri esitellään myöhemmin.

Kuva 4. Auton rattirunko (valokuva Arto Pirinen)

Kuvassa 4 näkyy auton rattirunko ilman päävalokatkaisijaa ja käsijarrun katkaisijaa. Laitteet koostuvat ratista, ohjainviiksistä, ohjausrungosta sekä ohjausakselista. Rattia käännettäessä ratin juuressa sijaitseva kieleke ohjaa vilkkujen toimintaa katkaisemalla vilkun toiminnan tietyllä käännöksellä. Käyttämällä omaa rattia vältyttiin siltä, että käännös olisi täytynyt anturoida ja simuloida elektronisesti tässä vaiheessa prototyyppiä. Ohjausakseli välittää voiman ohjausniveleen, joka edelleen välittää voiman adapterille. Ohjausnivelen ja peliohjaimen välille tehtiin adapteri, jotta käännös saatiin välitettyä peliohjaimelle.

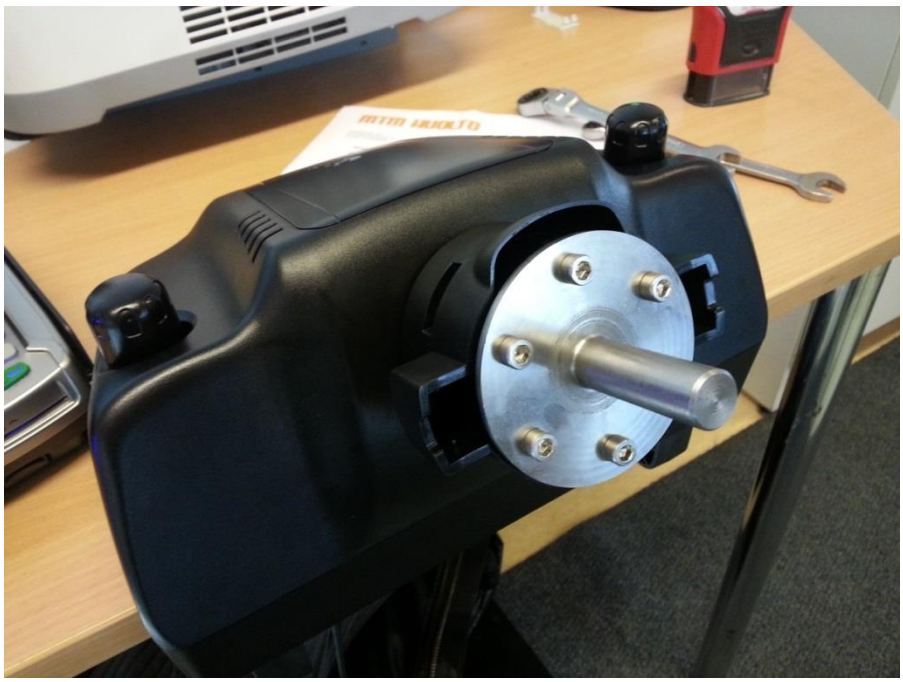
Kuva 5. Ohjausakselin pään kulmanivel (valokuva Arto Pirinen)

Kuvassa 5 nähdään ohjausnivel. Nivel kiertyy eri kulmissa, ja mahdollistaa ohjausakselin päästä saatavan kiertoliikkeen välittymisen toiselle akselille suorana kääntöliikkeenä.

3.5 Peliohjain

Logitechin G27 sisältää vastusmoottorit (Force Feedback). Vastusmoottorit antavat käytön aikana vastusta ohjausliikkeelle, jotta tulisi tuntuma oikean auton ajamisesta. Peliohjaimen ratissa on voimapalaute: kaksoismoottorien tärinäpalaute simuloi tarkasti todellista ajotilannetta. (Logitech 2013.)

Peliohjaimeen täytyi suunnitella adapteri jotta saataisiin ohjausakselin kiertoliike välitettyä peliohjaimelle. Adapterin liittäminen vaati alkuperäisen ratin poistamista peliohjaimesta, koska adapteri kiinnitettiin ratin kiinnikkeillä.



Kuva 6. Adapteri kiinnitettynä (valokuva Arto Pirinen)

Kuvassa 6 nähdään adapteri kiinnitettynä peliohjaimeen. Käyttämällä ohjausrungon omia kiinnityksiä autossa vältyttiin siltä, että peliohjaimen muoviselle rungolle tulisi kohtuuttomasti painoa. Varsinaista simulaatiota tehdessä ratin ominaisuudet tulevat käyttöön. Peliohjaimesta saadaan poimittua informaatio kiertoliikkeestä suoraan ilman erillistä anturointia. Peliohjaimen aiheuttama vastus tuo myös todentuntuisuutta simulaatioon, koska oikeatakin autoa ajettaessa kääntöliikkeistä aiheutuu vastusta rattiin. Adapteri valmistettiin Lekasteelin tiloissa.

3.6 AD-muunnin

AD- muunnin muuntaa muuntimelle tulevan analogisen signaalin digitaaliseksi signaaliksi, joka jäljittelee analogista signaalia. Tässä tapauksessa se on jännite. Yleisimpiä ovat perättäin arvioivat (successive approximation) A/D-muuntimet. Muuntimen toiminto perustuu kahden eri analogisen signaalin vertailuun komparaattorissa. Siksi kyseisessä A/D-muuntimessa on myös D/A-muunnin, joka tuottaa referenssisignaalin A/D-muuntimelle. Muuntimessa on myös ohjauspiiri ja rekisteri. Rekisteri säilyttää ohjausyksiköltä saatuja tietoja. Rekisteri välittää tiedon D/A-muuntimelle, joka edelleen antaa signaalin komparaattorille. Ohjauspiirille tulee kellosignaali, analoginen signaali ja aloituskomento. Kellosignaali jaksottaa muuntimen toimintaa, jotta muuntajan toiminta olisi ajoitettu mikroprosessorin muuhun toimintaan nähden. Komparaattorin periaatteellinen toiminta on seuraavanlainen,

$V_{DA} < V_A$ Komparaatorin ulostulo on 1
$V_{DA} = V_A$ Komparaattorin ulostulo on 0
$V_{DA} > V_A$ Komparaattorin ulostulo on 0

V_{DA} on D/A-muuntimelta saatu analoginen signaali. V_A on mitattu jännite.

Atmega 2560 sisältää referenssijännitteen, joko 2,56 V tai 1,1 V. Megan A/D-muunnin on asetettu toimimaan alueella 0-5 V, alueen digitaalinen arvo on 0-1023. (Hughes 2005, Atmel 2013, Arduino 2013.)

4 OHJELMISTO

4.1 Tietokoneen ohjelma

Tietokoneen ja Megan välinen kommunikointi tapahtuu sarjaliikenteellä käyttäen USB- porttia. C++ -kielen sarjaliikenne perustuu I/O-tiedostoon kirjoittamiseen, josta luetaan ja johon kirjoitetaan sarjaliikenneprotokollan mukaisesti informaatiota. C++ -kieli sisältää Windowsin omia referenssikirjastoja sarjaliikennekommunikaatioon liittyen, ja kirjottamani ohjelma pohjautuu kommunikoinnin osalta näihin kirjastoihin.

Ohjelmien ominaisuutena on reaaliaikaisuus, jotta simulaattorin hallintalaitteisiin ei syntyisi kohtuuttomasti viivettä. Reaaliaikaisuudella haettiin sitä, että ohjainlaitteiden luennassa ja tulkinnassa ei syntyisi tietokoneen päässä turhaa viivettä. Tietokoneen ohjelman ohjelmointiympäristönä toimi Visual Studio 2010.

Ohjelmointikielenä tietokoneen puolella toimi tavallisen C:n sijasta C++, koska C++ sisältää enemmän käytettäviä ominaisuuksia kuin peruskieli C ja ohjelmointityökalut soveltuivat tähän projektiin normaalia C -kieltä paremmin. C++ -kieltä en juurikaan ollut käyttänyt, joten jouduin opettelemaan kielen perusteet ja työkalut. C++ tarjoaa myöskin laajemman tuen sarjaliikennettä varten, sekä mahdollisuuden virtaviivaisempaan ohjelmointiin.

Kun Mega antaa tiedon tietystä ohjauslaitteen kynnysjännitteen ylityksestä, tietokoneen ohjelma tulkitsee sarjaliikenteen kautta tulleen tiedon. Sarjaliikenteen kautta tuleva data on ASCII-muotoista. Saatu 1 Megalta on tietokoneen ohjelmassa numero 49. Tietokoneen ohjelma reagoi näihin tietoihin kertomalla käyttäjälleen tietoa siitä, mitä laitetta on käytetty.

4.1.1 Sarjaliikenteen ominaisuudet

Lähetettäessä tietoa sarjamuotoisesti käytetään useimmiten asynkronista sarjaliikenneyhteyttä. Asynkronisessa sarjaliikenteessä ei lähetetä tahdistussignaaleja, vaan tieto otetaan vastaan sovitun protokollan mukaisesti. USB- sarjaliikenteessä laitteistoja on yhdistetty yhden väylätyypin alle, USB eli Universal Serial Bus. (Maksimainen, 2008.)

4.1.2 Sarjaliikenne C++ -ohjelmassa

Ensiksi avataan portti kommunikaatiota varten. Tämä tapahtuu käyttämällä C++ -referenssikirjastoa käyttäen, hyödyntäen CreateFile -funktiota sarjaliikenneportin avaamiseen. Tälloin luodaan Input/Output -tiedosto, johon isäntä sekä orja pystyvät kirjoittamaan tietoja ja kommunikoimaan keskenään.

```
HANDLE hComm;
hComm = CreateFile( gszPort,
                    GENERIC_READ | GENERIC_WRITE, // read/write enabled
                    0, //zero, cannot share COM- port
                    0, //zero, no security
                    OPEN_EXISTING,
                    FILE_FLAG_OVERLAPPED,
                    0);
```

Yllä olevassa koodissa avataan kahva, HANDLE hComm, jolla käsitellään CreateFile -funktiota. CreateFile -funtion sisällä on erinäisiä parametreja tiedoston luontiin, kuten avataanko uusi tiedosto vai käytetäänkö entistä. FILE_FLAG_OVERLAPPED tarkoittaa sitä, että tiedoston lukuun ja kirjoitukseen voidaan käyttää useita säikeitä samanaikaisesti. Se tukee ohjelman reaaliaikaisuutta, kun ei jouduta luomaan jonoa lukuun ja kirjoitukseen. Synkronointi tosin täytyy ottaa huomioon silti; säikeen täytyy odottaa että operaation tulos on saatavilla. Tekemässäni ohjelmassa ei kuitenkaan ollut tarvetta monisäikeiseen luentaan, vaan lukuoperaatio voitiin suorittaa normaaleilla parametreilla. (Windows, 2013.)

```
HANDLE hSerial;
hSerial =::CreateFileA("COM7",
            GENERIC_READ | GENERIC_WRITE,
            0,
            0,
            CREATE_NEW,
            //OPEN_EXISTING,
            FILE_ATTRIBUTE_NORMAL,
            0);
```

Yllä olevassa koodissa on kirjoittamani ohjelman I/O –tiedoston luonti. Tiedostoatribuutteina on normaali luenta ja kirjoitus, sekä kahvaa avattaessa luodaan aina uusi tiedosto.

Ohjelmaan sisältyy myös virheiden tarkastus. Ohjelma tarkastaa virheet sen varalta, että kehitysalusta ei ole kytketty tai kehitysalustan tilaa ei voida hakea.

Alla olevassa koodissa käsitellään mahdollinen sarjaliikenneportin virhetila. Jos sarjaliikenneportin tilaa ei voida hakea, tai se ei ole parametrien mukainen, käyttäjälle ilmoitetaan virheestä.

```
if(!GetCommState(hSerial, &dcbSerialParams)){
            std::cout<<"Error getting state"<<std::endl;
            return 0;
```

4.1.3 Tiedon käsittely tietokoneella

Megan lähettämät signaalit luetaan silmukassa seuraavanlaisella lauseella:

```
if((sTable[i] == sLeft) && (conLeft == 0) )
        {
                    std::cout<<"\nLeft signal on"<<::std::endl;
                    conLeft = 1;
        }
```

Yllä olevassa koodissa luetaan taulukkoon tallennetun muuttujan arvo ja verrataan toiseen muuttujaan. Muuttujalla on myös aina vasta-arvo, jota käytetään ohjauslaitteen tilan vertailussa. Tiloja verrataan sen takia, että saadaan tietää missä asennossa ohjauslaite on. Tieto ohjauslaitteen asennosta tulostetaan konsoliin cout-funktiolla, josta käyttäjä näkee ohjauslaitteen tilan, yllä olevassa esimerkissä vasemman vilkun kytkeytymisestä päälle. Vastaavasti käyttäjälle kerrrotaan, kun ohjauslaite, esim. vasen vilkku, kytkeytyy pois päältä. Vastaavanlainen koodi käsittelee kaikkien ohjauslaitteiden tiloja. Ainoana poikkeuksena on päävalokatkaisijaan ja valoviikseen liittyvä pitkien valojen toiminta. Pitkiä valoja varten tehtiin laskuritoiminto, jotta saatiin pitkät valot toimimaan kuten ne Saabin valoviiksessä oikeastikin toimisivat: yhdellä painalluksella pitkät valot päälle, toisella pois.

4.2 Arduino Mega:n ohjelmisto

4.2.1 Megan alustava ohjelma

Megan ohjelma oli alunperin jännitemittari. Jännitearvo käsitellään digitaaliseksi muutettuna Megalla, arvot vaihtelevat välillä 0-1023, vastaavasti jännite on 0-5 V. Vaihtelu tapahtuu alla olevan kaavan 2 mukaan. Kynnysarvo asetetaan halutun mittausjännitteen mukaan. Kommunikaatioprotokollat ovat Arduinon omaa käsialaa, ja kirjoittamani ohjelma hyödyntää näitä protokollia sarjaliikenteessä. Kynnysjännitteen ylittyessä Mega lähettää kyseisen ohjauslaitteen aiheuttamasta jännitteestä tiedon tietokoneelle sarjaliikennettä hyväksi käyttäen. Tämä tieto sitten käsitellään tietokoneella.

$$jännite = mitattu\ ADC\ arvo * \frac{5}{1024} \quad (2)$$

Alla on esimerkki Megan jännitteitä lukevasta ohjelmasta. Ohjelma havaitsee sen, kun kynnysarvo ylittyy, ja lähettää tiedon sarjaliikennettä hyväksikäyttäen tietokoneelle.

```
int sLeft = analogRead(A0);

if(sLeft >275)
 {
   //Serial.print("Signal to left");
   Serial.print(1);
   delay(250);}
```

4.2.2 Megan lopullinen ohjelma

Myöhemmin AD-muuntimen käytössä ilmeni ongelmia, joten ohjelma täytyi muokata lukemaan Megan digitaalisia portteja ja käyttämään ylösvetovastuksia. Ohjauslaitteet saatiin näin luettua katkaisijoina, mikä selkeytti ohjelman toimintaa huomattavasti.

```
void setup()
{
  //serial init at 9600 bps
 Serial.begin(9600);
pinMode(2, INPUT_PULLUP); //Left signal switch
}
```

Yllä olevassa koodissa on esimerkkinä ohjelman alustus. Setup() -funktio on Arduinon referenssifunktio, jossa ohjelman pääparametrit säädetään, avataan sarjaliikenne haluttaessa ja esitellään pinnien käyttökohteet. Tässä tapauksessa valitaan digitaalipinni 2, ja otetaan käyttöön pinnin sisääntulon ylösvetovastus. Kommenttina on, että pinniä käytetään vasemman vilkun luentaan.

```
int sLeft = digitalRead(2);

if(sLeft == LOW)
  {
  Serial.println(1);
  delay(50);
  }
```

Seuraavana esimerkkinä on signaalin luenta ja sen käsittely. Ensin asetetaan muuttuja digitaalipinnin luentaa varten. Digitaalipinni luetaan digitalRead(n) -funktiolla. If -lauseella vertaillaan digitaalipinnin tilaa. Jos tila on LOW, eli pinni on johtavassa tilassa, lähetetään siitä tieto sarjaliikennettä pitkin tietokoneelle. Lähetyksessä on myös pieni viive, jotta tietokone ja Mega ennättävät käsittelemään toiminnot varmasti ennen seuraavaa luentaa.

```
if(sLeft == HIGH )
{
  if((sLeft == LOW) || (sRight == LOW))
  {
  }
  else
  {
  Serial.println(7);
  delay(50);
  }}
```

Yllä olevassa koodissa on digitaalipinnien käsittely, kun pinni ei ole johtavassa tilassa. Kun pinni ei ole johtavassa tilassa, sen luenta ja lähetys on jatkuvaa. Näin saadaan varmistettua, että ohjauslaitteen pois päältä kytkeminen havaitaan ohjelmassa varmasti. If -lause ensin varmistaa että pinnin tila on looginen 1, ja jos näin on, edetään seuraavaan if -lauseeseen, jossa verrataan, onko toinen vilkku päällä tai vasemman vilkun tila muuttunut. Jos näin on, ei tehdä mitään. Jos ehdot eivät täyty, eli molemmat vilkut ovat pois päältä, ja varsinkin vasen vilkku, lähetetään tieto tilasta tietokoneelle. Megan ohjelmistossa pinnien luenta ja tiedon lähetys toimii kaikille signaaleille ja ohjauslaitteille vastaavasti.

5 TULOKSET

5.1 Alustavat tulokset

Alustavina tuloksina saatiin testikytkentä, jolla todennettiin että autosta otetut ohjauslaitteet toimivat tarpeen mukaan. Kommunikointi Megan ja tietokoneen välillä toimi, ja ohjauslaitteiden jänniteinformaatiota voitiin lukea käytännössä. Ohjelmistot vaativat vielä tässä vaiheessa säätöä päällekkäisyyksien ja datan käsittelyn suhteen. Ongelmaksi muodostui signaalien lukumäärä, ADC ei kyennyt mittaamaan jokaista signaalia ilman merkittäviä häiriöitä. Tästä johtuen tehtiin koekytkennät digitaalisilla sisääntuloilla hyödyntäen Megan omia ylösvetovastuksia. Tulokset olivat lupaavia.

5.2 Lopulliset tulokset

Lopullisena tuloksena syntyi toimivat kytkennät reaaliaikamittausten rinnalle. Testikytkentöjen perusteella Megan digitaaliset pinnit tulivat ratkaisevaksi osaksi järjestelmää, ja ratkaisivat analogisista mittauksista aiheutuneet ongelmat. Analogisista mittauksista luovuttiin kokonaan, ja siirryttiin käyttämään onnistuneesti Megan digitaalipinnien ominaisuuksia. Arduinon ja tietokoneen ohjelmat kommunikoivat keskenään moitteetta käyttäen sarjaliikennettä, ja ohjauslaitteiden tilat saatiin luettua häiriöttömästi ja luotettavasti.

6 POHDINTA

Tässä opinnäytetyössä toteutettiin toimiva sulautettu järjestelmä auton ohjauslaitteiden mittaamiselle tietokoneen ja Arduino Mega 2560:n väliseen sarjaliikennekommunikaatioon perustuen. Lopullinen järjestelmä koostuu Arduinon ja tietokoneen ohjelmista, laitteiden kytkentöjen selvityksestä sekä testaamisesta, ja lopullisen järjestelmän testaamisesta työpöytäolosuhteissa Lekasteelin käyttöön. Työn ohella tehtiin myös prototyyppivaiheessa mukana olevaan peliohjaimeen adapteri auton rattirunkoa varten.

Insinöörityössäni pääsin työskentelemään sulautetun järjestelmän kaikkien komponenttien, sekä ohjelmiston että laitteiston kanssa. Työ tarjosi myös haasteita molemmin puolin, ja oli tästä syystä varsin opettavainen. Oman lisänsä toi myös C++- kieleen perehtyminen ja sen käyttäminen työssä. Haastavinta oli ADC:n häiriökäyttäytyminen, sekä ohjauslaitteiden kytkentöjen selvittäminen testikytkentöjä varten. Työ oli hyvä oppimiskokemus vaihtoehtoisten ratkaisujen etsinnässä sekä hyödynti koulutukseni kaikki osa-alueet aina vianetsinnästä sulautetun järjestelmän toteuttamiseen. Työ havainnollistaa hyvin Arduino Megan monikäyttöisyyttä sulautetuissa järjestelmissä ja Arduinon käyttäminen sulautetun järjestelmän mittaus- sekä sensorikeskuksena tarjoaa kattavat mahdollisuudet erilaisiin prototyyppiprojekteihin.

LÄHTEET

ARDUINO 2013 Mega 2560 R3 *Arduinon verkkosivut* [Verkkodokumentti] [viitattu 16.6.2013]

Saatavissa: http://arduino.cc/en/Main/ArduinoBoardMega2560 22.3

ATMEL 2013. ATMega 2560. *Atmelin verkkosivut ATmega2560 alustasta* [Verkkodokumentti] [viitattu 16.6.2013]

Saatavissa: http://www.atmel.com/devices/atmega2560.aspx

ATMEL 2012. ATMega 2560. *ATmega 2560:n tekninen dokumentti, datasheet* [PDF- dokumentti] [viitattu 30.7.2013, 25.10.2013]

Saatavissa: http://www.atmel.com/Images/doc2549.pdf

WINDOWS 2013. Win32 -sarjaliikenneprotokolla. *Windowsin verkkosivu* [Verkkoartikkeli] 1995 [viitattu 16.6.2013]

Saatavissa: http://msdn.microsoft.com/en-us/library/ms810467.aspx

LOGITECH 2013. G27 -peliohjaimen ominaisuudet. *Logitechin verkkosivu* [viitattu 25.10.2013]

Saatavissa: http://gaming.logitech.com/en-us/product/g27-racing-wheel

HUGHES, Edward. 2005. Electrical and electronic technology, ninth edition, [viitattu 25.10.2013]

Saatavissa: Savonia ammattikorkeakoulun TeKu kirjasto

MAKSIMAINEN, Väinö. 2008. RS232- sarjaliikenneraportti. *Kurssimateriaali* [PDF- fokumentti] 2008 [viitattu 18.11.2013]

MAKSIMAINEN, Väinö. 2008. USB- johdanto. *Kurssimateriaali* [PDF- dokumentti] 2008 [viitattu 18.11.2013]