

# Tiedonsiirtoväylä sähköajoneu- voon

Jukka Povelainen

OPINNÄYTETYÖ  
Elokuu 2021

Sähkötekniikka  
Älykkäät koneet

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkötekniikka  
Älykkäät koneet

POVELAINEN, JUKKA:  
Tiedonsiirtoväylä sähköajoneuvoon

Opinnäytetyö 43 sivua, joista liitteitä 4 sivua  
Elokuu 2021

---

Opinnäytetyön aiheena oli suunnitella ja rakentaa ajoneuvokäyttöön tiedonsiirtoväylä ja ajotietokone. Ajoneuvona oli sähkömopo, josta mitattiin erilaisia järjestelmän toimintaan liittyviä tietoja, jotka esitettiin erillisessä näytössä ajoneuvon kuljettajalle.

Aluksi perehdyttiin aiheeseen liittyvään teoriaan. Teoria sisältää fysiikan ilmiöt erilaisten mittaus- ja tiedonsiirtotekniikoiden taustalla. Muutamia väyläyhteyskäytäntöjä vertailtiin keskenään ja niistä valittiin tähän projektiin sopivin. Työssä sivuttiin hieman myös sitä, minkälaisia ominaisuuksia tulevaisuuden laitteistot tulevat tarvitsemaan.

Laitteiston vaatimukset määriteltiin ja valittiin sopivat komponentit tarvittavien vaatimusten täyttämiseksi. Tiedonsiirron alttius ulkoapäin tuleville häiriöille otettiin huomioon laitteiston osien valinnassa. Seuraavaksi suunniteltiin laitteiston kokoonpano ja ohjelmisto. Ohjelmiston suunnittelussa pyrittiin yksinkertaisuuteen ja mahdollisimman varmaan toimintaan. Lopuksi rakennettiin järjestelmä ja testattiin järjestelmän toiminta.

Työn lopputuloksena oli toimiva kolmen laitteen välinen tiedonsiirtoväylä. Laitteisto toimi suunnitellusti. Projekti onnistui kohtuullisen hyvin, ilman suurempia vastoinkäymisiä. Järjestelmää voisi jatkossa vielä kehittää parantamalla häiriönsuojausta ja näyttämällä näytössä erilaisia laskennallisia arvoja, kuten vaikkapa jäljellä oleva ajomatka.

---

Asiasanat: väylä, mikrokontrolleri, ajoneuvo, ajotietokone

## **ABSTRACT**

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

POVELAINEN, JUKKA:  
Data Bus for an Electric Vehicle

Bachelor's thesis 43 pages, appendices 4 pages  
August 2021

---

The objective of the thesis was to design and create a data bus with a driving computer for vehicle use. The vehicle was an electric moped, which was measured in various ways and the information was presented for the driver in a separate display.

At first, the theories of relevant topics were researched, including physical phenomena behind various measuring and data transferring technologies. A few data bus protocols were compared and the most compatible was chosen for this project. The features of the future technologies were briefly introduced.

The requirements of the hardware were determined, and the suitable components were chosen. The data transmission vulnerability for external interference was considered when choosing the components. Next, the topology and software were developed. The software was created with simplicity and reliability in mind. Finally, the system was built and tested.

The outcome was a working data bus with three connected devices. The system functioned as planned. The project succeeded rather well, without any bigger drawbacks. The system could be improved in future by shielding the signal from interference even better. And showing different calculated values in the display, for example the remaining driving range.

## SISÄLLYS

1	JOHDANTO .....	6
2	TEORIA .....	7
	2.1 Tiedonsiirtoväylä .....	7
	2.2 Erilaiset yhteyskäytännöt .....	7
	2.3 Häiriönpoisto .....	8
	2.4 Kehittyneet ominaisuudet .....	10
	2.5 Mikrokontrollerit .....	12
	2.6 Hall-efekti .....	13
	2.7 Litium-ioni akkutekniikassa .....	14
	2.8 Lämpötilan mittaus .....	14
3	VÄYLÄT AJONEUVOISSA .....	15
4	JÄRJESTELMÄN MÄÄRITTELY .....	16
	4.1 Tiedonsiirtonopeus .....	16
	4.2 Toimintavarmuus .....	16
	4.3 Akuston jännite, virran kulutus ja lämpötila .....	16
	4.4 Ajonopeuden mittaus .....	17
	4.5 Valojen ohjaus .....	18
	4.6 Näyttö .....	18
	4.7 Suojalaitteet .....	18
5	SUUNNITTELU .....	20
	5.1 Ajoneuvo, johon väylä suunnitellaan .....	20
	5.2 Yhteyskäytäntö .....	21
	5.3 Topologia .....	23
	5.4 Solmut .....	24
	5.5 Johtimet .....	24
	5.6 Näyttö .....	27
	5.7 Muut laitteet .....	27
	5.8 Kytkenät .....	28
	5.9 Ohjelmisto .....	31
6	RAKENTAMINEN .....	32
7	POHDINTA .....	36
	LÄHTEET .....	38
	LIITTEET .....	40
	Liite 1. Arduino 1:n ohjelmakoodi .....	40
	Liite 2. Arduino 2:n ohjelmakoodi .....	42

**ERITYISSANASTO**

Väylä	tapa siirtää tietoa
Protokolla	yhteyskäytäntö
CAN	Controller Area Network
I2C	Inter-Integrated Circuit
Bluetooth	langaton tiedonsiirtotapa
Mikrokontrolleri	mikroprosessori ja joitain lisälohkoja
IOT	Internet of Things, esineiden internet
LCD	Liquid Crystal Display

## 1 JOHDANTO

Nykypäivän autot ja työkoneet ovat täynnä älykkyyttä. Harva liikkuva kone toimii ilman elektronisia mittaus- ja säätölaitteita. Kaikki tämä mittaaminen ja säätäminen vaativat jonkin keinon siirtää tietoa toimilaitteiden välillä. Perinteisesti tuota tietoa on siirretty kaapelilla kahden pisteen välillä. Nykyaikaisessa henkilöautossa on satoja toisiinsa yhteydessä olevia laitteita. Ei ole järkevää vetää kaapelia jokaisen laitteen välille erikseen, tähän ongelmaan vastaa ratkaisu nimeltään väylä. Kaikki laitteet liittyvät tuohon samaan väylään, vähentäen tarvittavien johdinten määrää merkittävästi.

Tämän opinnäytetyön tavoitteena oli tutustua väylätekniikkaan yleisesti, väylien ominaisuuksiin erityisesti ajoneuvokäytössä ja valmistaa olemassa olevaan sähköllä liikkuvaan ajoneuvoon tiedonsiirtoväylä ihan alusta asti. Rakennettavan väylän tarkoitus on siirtää tietoa ajoneuvon nopeudesta, akuston tilasta ja virran kulutuksesta. Väylää käytetään myös ajovalon ohjaukseen.

Opinnäytetyön lopputuloksena on kirjallinen tietopaketti liikkuvien koneiden väylästä, sekä modernilla väylätekniikalla täydennetty 70-luvun mopo.

## 2 TEORIA

Käsitellään aluksi teoreettisesti projektiin liittyvien laitteiden toimintaa ja niihin liittyviä fysiikan ilmiöitä.

### 2.1 Tiedonsiirtoväylä

Väylä on tietokoneen sisäisten komponenttien tai erillisten tietokoneiden väliseen tiedonsiirtoon tarkoitettu järjestelmä. Tieto liikkuu sähköjohtimissa jännitetason vaihtelun muodostavina bitteinä, ykkösinä ja nollina. On olemassa myös langattomia tai valokaapeleihin perustuvia väyliä, joita käytetään vain jossain erityistapauksissa.

Suurimmat syyt väylän käyttöön ovat johtimien määrän väheneminen ja mahdollisuus liittää suuria määriä laitteita samaan tiedonsiirtokanavaan. Tavallisessa henkilöautossa voi olla satoja toisiinsa yhdistettyjä kytkimiä, antureita, näyttöjä ja ohjainlaitteita. Useampi laite voi tarvita jonkin tietyn anturin mittaamaa tietoa samanaikaisesti. Käyttämällä tiedonsiirtoväylää, kyseinen vaatimus voidaan toteuttaa.

Väylän tiedonsiirtonopeus, toimintavarmuus ja yhdistettävien laitteiden lukumäärä vaihtelee suuresti erilaisten väyläprotokollien välillä.

### 2.2 Erilaiset yhteyskäytännöt

Erilaisia väyliä käytetään tietokoneiden sisäiseen tietoliikenteeseen ja myös yhdistämään ulkoisia laitteita itse tietokoneeseen. Tietokoneen sisällä prosessori, keskusmuisti ja emolevy ovat yhteydessä nopeilla ja lyhyillä välimatkoille tarkoitetuilla yhteyskäytännöillä. Tietokoneeseen liitettävät ulkoiset laitteet, esimerkiksi tulostin tai ulkoinen kovalevy liitetään vähän pidemmällä kaapeleilla, jolloin yhteyskäytännöltä vaaditaan erilaisia ominaisuuksia. Bittien sekoittumisen estämiseksi joudutaan käyttämään korkeampaa jännitettä ja hitaampia nopeuksia.

Perinteisesti tietokoneen lisälaitteet on liitetty RS-232 (Recommended Standard) tai RS-485 yhteyskäytännöillä. Nykyisin vallalla on USB (Universal Serial Bus).

Tietokoneen sisäisessä liikenteessä käytetään rinnakkais- ja sarjaliikennestandarddeja. Rinnakkaisliikenteessä johtimia on enemmän, mutta sillä päästään myös suurempiin nopeuksiin. Sarjaliikennekäytäntöjä on esimerkiksi SPI, PCIe, I2C ja SATA.

### **2.3 Häiriönpoisto**

Liikkuvilla koneissa väylät joutuvat alttiiksi ympäristön olosuhteille. Ulkona saattaa olla pakkasta, vettä roiskuu, suolaa ja öljyä tunkeutuu joka paikkaan. Kaikkien liittimien, kaapeleiden ja läpivientien on oltava sellaisia, että ne kestävät kulutusta ja edellä mainittuja olosuhteiden rasituksia.

Johdinten lähistöllä olevat sähkömoottorit, polttomoottorin sytytyslaitteiston kipinäinti ja muut mahdolliset sähkömagneettisia häiriöitä aiheuttavat laitteet voivat tuottaa ongelmia tiedonsiirtoon. Käytettyjen kaapeleiden on oltava häiriöitä sietäviä. Erilaisia suojaustapoja voi olla esim. kaapeleiden kierteisyys, kaapelin ympärillä oleva suojakerros tai ohjelmallinen häiriönpoisto.

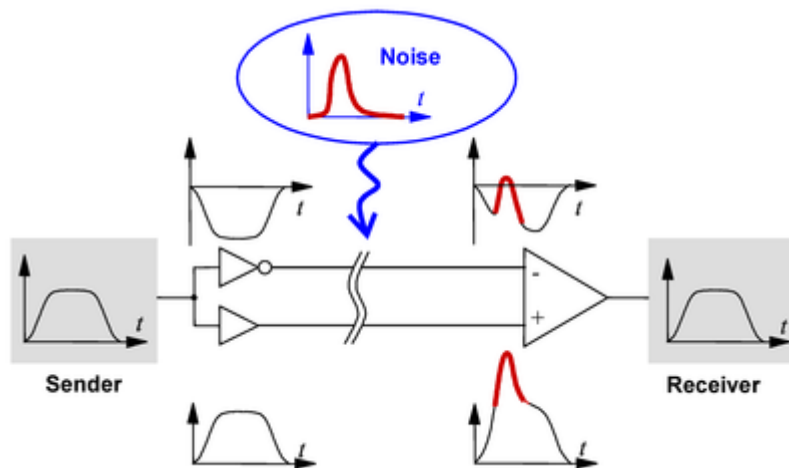
Johtimissa ilmeneviä häiriöitä on muun muassa kapasitiivisia ja induktiivisia. Kapasitiiviset häiriöt ovat sähkökentän aiheuttamia ja niitä siirtyy yleensä vierekkäin olevista johtimista toisiinsa. Helpoimmin tällaisia häiriöitä voi vähentää siirtämällä johtimia kauemmas toisistaan. Induktiivinen häiriö taas syntyy magneettikentän aiheuttamana. Magneettikentän muutos tapahtuu, kun johtimen lähellä on liikkuva magneetti tai lähellä olevassa johtimessa kulkevan virran suuruus muuttuu. Tässäkin tapauksessa helpointa on siirtää johtimia kauemmas toisistaan.

Signaalia voi suojata myös muilla keinoin. On saatavilla kaapeleita, joissa on johtimia suojaava metallinen kuori, joka pitää ulkoapäin tulevat häiriöt pois johtimista. Kierteisiä johtimia käytettäessä saadaan vähennettyä johdinparista toiseen aiheutuvia häiriöitä ja ylikuulumista. Johdinparit kannattaa muodostaa siten,



että signaalia kuljettava johdin kiertyy jonkin tasaista jännitetasoa pitävän johtimen kanssa, esimerkiksi maa- tai käyttöjännitejohdin. Jos kaksi datajohdinta kierretään toisiinsa, se saattaa aiheuttaa lisää ongelmia niiden vähentämisen sijaan. (Twisted pair, Wikipedia 2021)

Häiriöitä voi poistaa myös ohjelmallisesti. Voidaan käyttää esimerkiksi niin kutsuttua differentiaalista signaalia. Sama signaali kuljetetaan kahta johdinta pitkin, signaalien ollessa toistensa peilikuvia. Mahdolliset häiriöt ilmenevät samanlaisina kummassakin signaalijohtimessa. Viestiä vastaanottavassa päässä johtimien jännitetasot vähennetään toisistaan, jolloin häiriöt poistuvat (kuva 1). Tähän tekniikkaan perustuu hyvin yleisesti käytetty CAN-väylän häiriönpoisto.

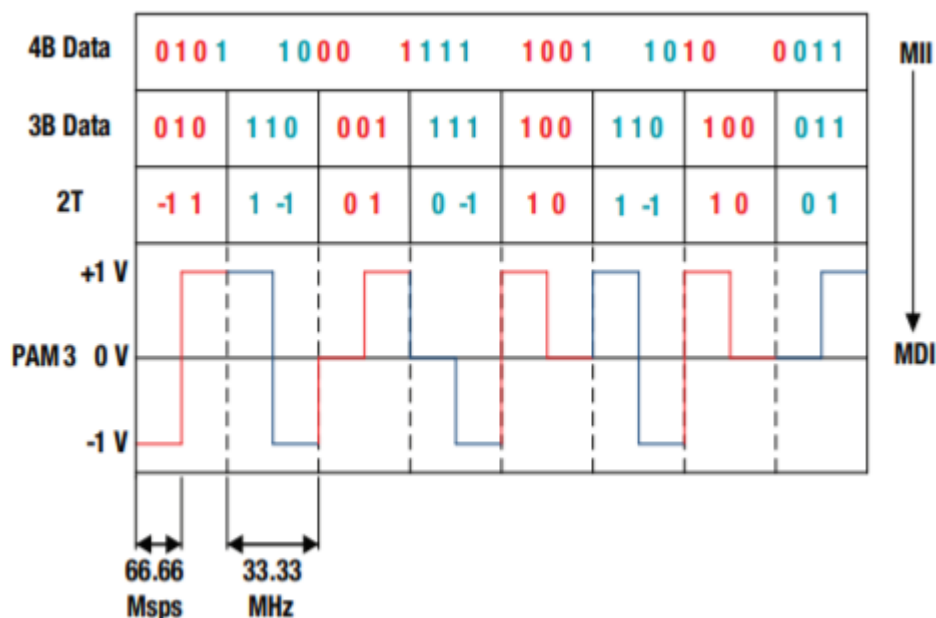


Kuva 1. Differentiaalinen signaali. (Differential signaling, Wikipedia 2021).

## 2.4 Kehittyneet ominaisuudet

Nykypäivän tiedonsiirrolta vaaditaan yhä suurempia nopeuksia ja suurempaa toimintavarmuutta. Häiriönsieto kasvaa, kun käytetään vaikka optista kaapelia sähköä johtavien kaapeleiden sijaan. Valokuituun kun ei pääse vaikuttamaan mitkään ulkoiset häiriöt.

Sähköä johtavissa protokollissa on alettu käyttämään useampaa kuin kahta jännitetasoa tiedonsiirron nopeuden nostamiseksi. 100BASE-T1 protokolla käyttää kolmea jännitetasoa; -1, 0 ja 1 voltia. Jokainen jännitetason muutos vastaa kolmea bittiä kuvan 2 mukaisesti. Tällä keinolla saadaan ahdettua enemmän dataa samaan kellotaajuuteen. Menetelmä on nimeltään Pulse Amplitude Modulation.



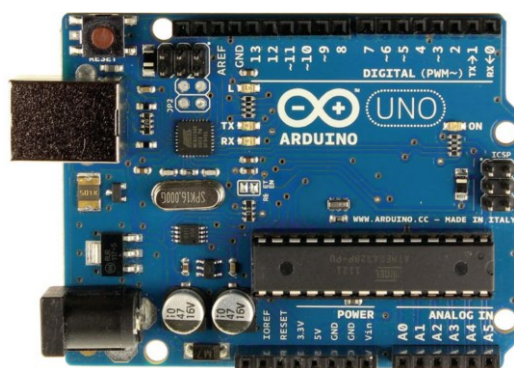
Kuva 2. Pulse Amplitude Modulation, eri jännitetasojen muuttaminen biteiksi. (100BASE-T1, Ti.com 2018)

Laitteet, jotka eivät pysy kiinteästi paikallaan, mutta jotka liittyvät osaksi ajoneuvon järjestelmää, siirtävät tietoa usein langattomasti. Tällaisia laitteita ovat muun muassa matkapuhelimet, jotka voivat siirtää musiikkia sisältävää dataa auton viihdejärjestelmään Bluetooth-tiedonsiirtoprotokollan avulla. Joskus tulevaisuudessa IOT:n yleistyessä, myös ajoneuvot todennäköisesti juttelevat keskenään. Ajoneuvot voivat ilmoittaa toisilleen sijaintinsa ja nopeutensa ja niillä tiedoilla esi-

merkiksi hätäjarrutukset voidaan aloittaa tarvittaessa automaattisesti. Näitä toimintoja varten on käytettävä jotain langatonta protokollaa, joskin Bluetooth on tässä tapauksessa riittämätön ratkaisu. Ehkäpä matkapuhelinverkon 5G tuo yleistyessään mahdollisuuden autojenkin verkottumiseen.

## 2.5 Mikrokontrollerit

Mikrokontrollerit, eli mikropiirit ovat pieniä komponentteja, joita käytetään sulautetuissa järjestelmissä. (Wikipedia: Mikrokontrolleri 2021). Mikropiiri sisältää mikroprosessorin lisäksi joitain muisti- ja liityntälohkoja. Elektroniikkaharrastajille on tarjolla kaupallisia tuotteita, joilla on helppo opetella mikrokontrollerien käyttöä. Tunnetuin tällainen harrastuskäyttöön tarkoitettu laite on Arduino (kuva 3). Arduino-kehitysalustassa on valmiina mikrokontrollerien kanssa yleisesti käytettyjä lisäkomponentteja, kuten oskillaattori, DA-muunnin sekä eri enkoodereita lisälaitteiden liittämistä varten.

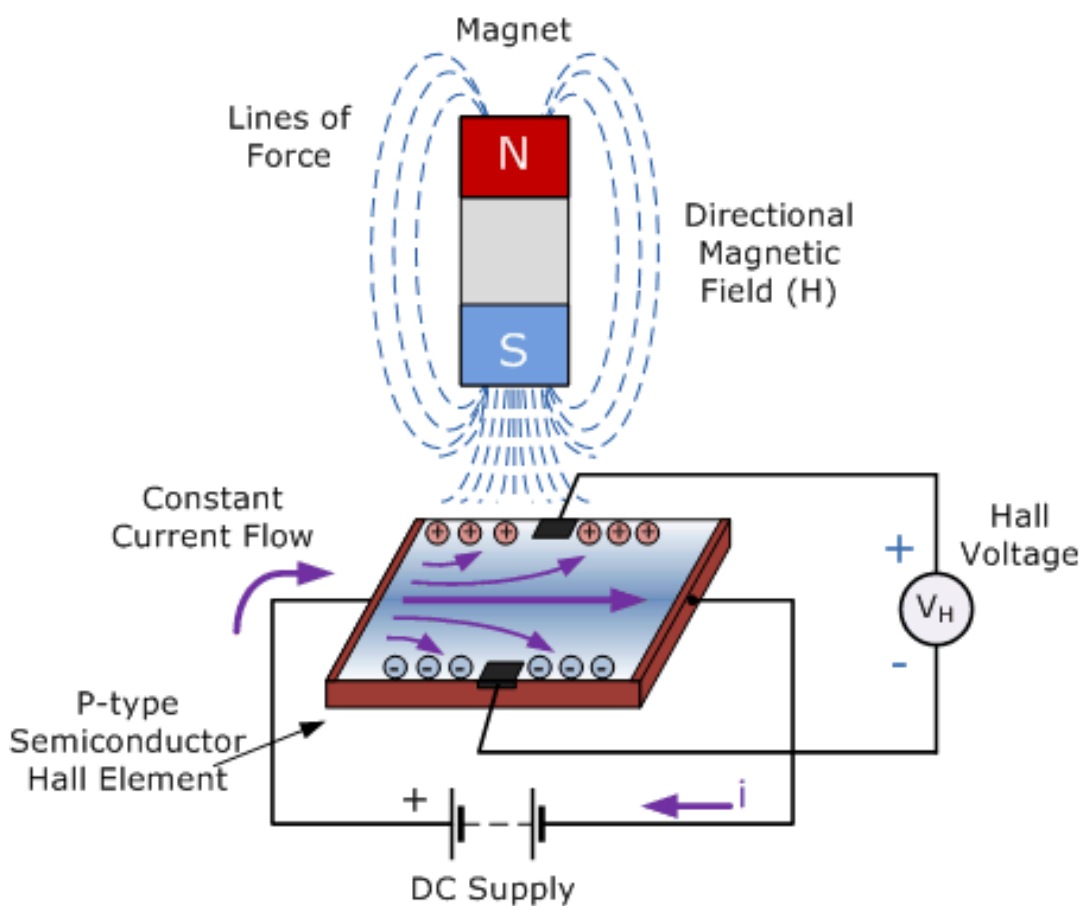


Kuva 3. Arduino Uno -mikrokontrolleri. (elfadistrec.fi 2021)

Arduinoja on erikokoisia, eri kokoonpanoilla ja eri mikropiireillä toteutettuna. Ohjelmointi tapahtuu helposti tietokoneella Arduino IDE -ohjelmalla. Ohjelma voidaan syöttää mikrokontrolleriin USB-kaapelin kautta.

## 2.6 Hall-efekti

Hall-ilmion keksi Edwin Hall, vuonna 1879 (Wikipedia: Hall effect, 2021). Kyseessä on ilmiö, jossa magneettikenttä vaikuttaa elektronien kulkuun. Elektronit pakkautuvat kappaleen toiseen reunaan aiheuttaen jännite-eron kappaleen reunojen välille. Magneettikentän voimakkuutta voidaan mitata tämän jännitteen perusteella. Hall-efektiä käytetään yleisesti mittaamaan magneettisten kappaleiden läheisyyttä tai sähkövirtaa johtimessa. Tässä projektissa kyseistä efektiä käytetään sähkömoottorin pyörintänopeuden mittaamiseen sekä akun virran mittaamiseen.



Kuva 4. Hall-ilmiö puolijohteessa. (Electronics Tutorials 2021)

Hall-ilmioon perustuva anturi ei tarvitse fyysistä kosketusta mitattavaan kohteeseen, joten se on hyvä sähkömoottorin asennon tunnistamiseen. Hall-anturi tunnistaa roottorin magneettikentän ja antaa tiedon moottorin ohjainlaitteelle, joka osaa tämän perusteella ohjata staattorin käämien virtaa halutun vääntömomentin aikaansaamiseksi.

## 2.7 Litium-ioni akkutekniikassa

Projektin kohteena olevassa mopossa sähköenergian varastointi on toteutettu litium-ioni-akustolla. Litium on kevyttä ja energiatiheydensä vuoksi soveltuvaa ajoneuvokäyttöön. Li-ion-akun purku- ja latausvirta on perinteiseen lyijyakuun verrattuna moninkertainen. (Cummins.com, 2019)

Yksi ongelma litiumiin perustuvissa akuissa on sen elektrolyytti. Se on neste-mäistä ja helposti syttyvää. Akuston lämmön hallinta ja akkukennojen vikaantumiset on otettava huomioon.

Projektin aihiona toimivassa sähkömopossa on 13S5P-kokoonpanolla oleva kierätysosista rakennettu li-ion akusto (13 kennoa kytketty sarjaan ja viisi rinnan). Yhden kennon nimellisjännitteen ollessa 3,7 voltia, akuston kokonaisjännitteeksi tulee tällöin 48,1V.

## 2.8 Lämpötilan mittaus

Akuston lämpötilan mittaukseen voidaan käyttää LM35 -lämpötila-anturia. LM35-komponentin toiminta perustuu transistorin kannan ja emitterin väliseen jännite-eroon.

Transistorin jännite-eroon vaikuttaa sen läpi kulkevan virran suuruus ja ympäröivä lämpötila. LM35-piirin lämpötilaa mittaavassa elementissä on kaksi transistoria, joihin ohjataan vakioidut erisuuruiset virrat. Emitterien välille muodostuu jännite, joka on suoraan verrannollinen transistoreihin vaikuttavaan lämpötilaan. Tällä menetelmällä toteutettu lämpötilan mittaus on edullinen ja suhteellisen tarkka. (Texas Instruments, Diode-based temperature measurement 2019).

LM35 on helppo liittää mikrokontrolleriin ja mittaus on yksinkertaista. Lämpötila-alue on -55 ja 150 celsiusasteen välillä, mikä riittää projektiin hyvin. Puolen asteen tarkkuuskin on aivan riittävä.

### 3 VÄYLÄT AJONEUVOISSA

Ajoneuvoissa käytetään tiedonsiirtoväyliä johtosekamelskan vähentämiseksi. Nykyaikaisen auton yleisin väyläprotokolla on CAN (Controller Area Network). Jotain ei-niin-kriittisiä laitteita voidaan ohjata hitaamman LIN-aliverkon kautta. (Local Interconnected Network). Syy LIN-väylän käyttöön CAN-väylän rinnalla on erityisesti hinta. LIN-väylään liitettävät laitteet ovat edullisempia valmistaa. Henkilöautossa LIN-väylään yleensä liitetyt laitteet ovat koritekniikan toiminnot, kuten ovien lukot, ikkunoiden nosto ja lasku, istuimien säätö ynnä muut kii-reettömät toimenpiteet. Edelleen kaikki ajamiseen ja turvallisuuteen liittyvä data liikkuu nopeammassa CAN-väylässä.

Väylässä liikkuva informaatio tulee tulevaisuudessa vain lisääntymään ja nopeutta vaaditaan entistä enemmän. Kuljettajaa avustavat järjestelmät (ADAS) käsittelevät monesti videokuvaa useasta kamerasta ja erilaisten tutkien tiedot vielä lisäksi. Vaatimukset väylän tiedonsiirtonopeudelle kasvavat todella suuriksi. Tähän ongelmaan on kehitelty uusia yhteyskäytäntöjä, kuten esimerkiksi CAN FD (Controller Area Network Flexible Data-Rate). Siinä tuupataan viestikehykseen enemmän bittejä ja jopa 8 megabitin nopeudella (vrt. CAN 1mbps).

Yksi lupaavimmista suuren nopeuden väyläprotokollista on Ethernet. Tämä käytäntö on yleisesti käytössä internetin tiedon kuljettamisessa. Uusimmilla datan pakkaustekniikoilla ethernetillä päästään jopa 10 gigabittiin sekunnissa. Ethernetistä on kehitetty omaa ajoneuvokäyttöön soveltuvaa versiota. (einfochips.com 2020).

Internet-yhteyskäytännön sulauttaminen osaksi ajoneuvon sisäistä tiedonsiirtoa antaisi myös mahdollisuuden yhdistyä vaivattomasti ajoneuvon ulkopuoliseen verkkoon.

## **4 JÄRJESTELMÄN MÄÄRITTELY**

Tavoitteena on rakentaa liikkuvan koneen tiedonsiirtojärjestelmä. Tietoja kerätään akusta, moottorista ja kytkimien asennoista. Akusta mitattavat tiedot ovat jännite, virta ja lämpötila. Moottorista mitataan kierrosnopeus. Ajovalon vaihtamiseen käytetään vaihtokytkintä, jonka asentotieto siirretään väylässä näytölle.

### **4.1 Tiedonsiirtonopeus**

Väylässä liikkuva data ei tässä projektissa sisällä mitään mopon ajettavuuden tai turvallisuuden kannalta kriittistä sisältöä. Jarrut toimivat puhtaasti mekaanisesti ja moottorin ohjaus on väylästä erillinen järjestelmä. Tiedonsiirtonopeutta ei siis tässä tapauksessa tarvitse ottaa huomioon väyläprotokollan valinnassa. Hitainkin tiedonsiirtotapa on riittävä.

### **4.2 Toimintavarmuus**

Toimintavarmuus ja häiriönsieto tässä projektissa on saatava sille tasolle, että järjestelmä on toimintakuntoinen. Satunnaisia toimintahäiriöitä tai virheellisiä mitausarvoja voidaan sietää jonkin verran, jos ne eivät haittaa laitteiston toimintaa yleisesti. Täydellisen jumiutumisen tapahtuessa, pääkytkintä kääntämällä voidaan järjestelmä käynnistää uudestaan.

Tiedonsiirron toimintavarmuutta voidaan parantaa kaapeleiden valinnalla, yhteyskäytännön valinnalla, kaapeleiden sijoittelulla sekä tarvittaessa aktiivisilla häiriöitä poistavilla järjestelmillä.

### **4.3 Akuston jännite, virran kulutus ja lämpötila**

Akussa jäljellä oleva varaus voidaan päätellä akun jännitteestä. Litium-ioniakun yksittäisen kennon (tyyppi: 18650) nimellisjännite on 3,7V. Täysi kenno on 4,2V ja tyhjä 3V. Akustossa kennoja on kytketty sarjaan 13, jolloin täyden akkupaketin jännite on 54,6 voltia ja tyhjän noin 39 voltia. Mitataan akun jännite ja lasketaan



sen perusteella hetkellinen varaustila. Akun varaustila ilmoitetaan näytössä helposti ymmärrettävänä prosenttilukuna. Mitä suurempaa virtaa akusta otetaan, sitä enemmän jännite alenee johtimien ja akun sisäisen resistanssin vuoksi. Tästä syystä akun varaustila näyttäytyy ajettaessa pienempänä kuin paikallaan pysyessä. Tätä virran aiheuttamaa jännitteenalenemaa voitaisiin kompensoida laskennallisesti, mutta toisaalta myös tieto jännitteen hetkellisestä arvosta antaa informaatiota järjestelmän suorituskyvystä.

Virran kulutus mitataan akuston positiiviselta johtimelta tarkoitukseen sopivalla Hall-efektiin perustuvalla anturilla. Anturille kytketään viiden voltin käyttöjännite ja maa. Anturilta tuleva signaalijohdin kytketään Arduino 2:n sisääntuloon.

Akuston lämpötila mitataan yksinkertaisella LM35-lämpötila-anturilla.

#### 4.4 Ajonopeuden mittaus

Moottorinohjainlaite tarvitsee toimiakseen tiedon moottorin asennosta. Tuo tieto tuotetaan moottorin sisälle sijoitetuilla hall-antureilla, jotka havaitsevat roottorin kestopagneettien ohitukset. Signaalien taajuuden perusteella voidaan suoraan laskea moottorin kierrosluku.

Moottorin staattorissa on neljä napaparia ja hall-antureita on kolme kappaletta. Yksi jokaista staattorin vaihetta kohti. Roottorin pyörähtäessä yhden kierroksen, hall-anturin jännite nousee ja laskee neljä kertaa. Kaavalla 1 saadaan siis selville moottorin kierrosnopeus.

$$N = \frac{\text{signaalien lukumäärä} \times 60 \text{ sekuntia}}{\text{napaparien määrä} \times \text{kulunut aika sekunteina}}$$

Kaava 1. Kierrosta minuutissa -laskukaava

Mopon ajonopeus voidaan laskea kaavan 2 mukaan, kun tiedetään moottorin kierrosluku, kokonaisvälyssuhde (eturatas 9, takaratas 42 hammasta,  $i=0,214$ ) ja takarenkaan ulkohalkaisija (60 cm).

$$v = \frac{N \times 0,214 \times \pi \times 0,6m \times 60 \text{ min}}{1000m}$$

Kaava 2. Nopeuden laskenta. (km/h)

#### 4.5 Valojen ohjaus

Toteutetaan valojen kytkentä siten, että lähivalot ovat automaattisesti toiminnassa, kun pääkatkaisija on kytketty johtavaan tilaan. Oikealla peukalolla käytetään erillistä vaihtokytkintä, jolla valitaan lyhyet tai pitkät ajovalot. Valokytkimen tilasta menee signaali umpiossa olevalle ohjainlaitteelle, joka taas vaihtaa jännitteen kulkemaan releen toisen asennon kautta kaukovalolle. Ajovalon vallitseva tila esitetään väylään liitetyllä näytöllä.

#### 4.6 Näyttö

Näyttöön kootaan kaikki kerätyt tiedot esitettäväksi ajoneuvon kuljettajalle. Näyttö liitetään väylään omalla osoitteella, omana laitteenaan. Näytön tiedot on kyettävä lukemaan myös auringonvalossa ja siihen pitää mahtua kaikki esitettävät tiedot kerralla. Esimerkiksi ajovalon tila voidaan esittää symbolisesti.

#### 4.7 Suojalaitteet

Oikosulun varalta järjestelmässä on ensimmäisenä pääsulake akulta lähtevässä positiivisessa johtimessa. Sen jälkeen on pääkytkin, joka tarjoaa 48 voltin jännitteen suoraan moottorinohjaimelle ja DC-DC-muuntimelle. Muunnin alentaa jännitettä 12 volttiin, jolloin voidaan käyttää mopon alkuperäisiä sähkölaitteita. Mik-

rokontrollerit, joita käytetään väylän ohjainlaitteina, tarvitsevat käyttöjännitteeseen 7 - 20 volttia. Mikrokontrollerit voidaan siten liittää suoraan tuohon 12 voltin jännitteeseen.

Laitteistossa esiintyvät jännitteet ovat niin matalia, ettei niille altistuminen aiheuta välitöntä hengenvaaraa. Alle 120 voltin tasajännitteet ovat niin kutsuttuja pienoisjännitteitä, joita on luvallista rakentaa ja muokata ilman rajoituksia. (Sähkötyöt- ja urakointi, Tukes.fi 2021)

## 5 SUUNNITTELU

### 5.1 Ajoneuvo, johon väylä suunnitellaan

Kohteena on Tunturi Trial, vuosimallia 1979. Mopossa on alun perin ollut itävaltalaisvalmisteinen Puch-merkkinen kaksitahtimoottori, nelivaihteisella vaihteistolla. Vuoden 2020 aikana moposta poistettiin polttomoottori vaihteistoinen ja tilalle asennettiin kiinalaisvalmisteinen kestromagneettitahtimoottori. Muutettu laite on esitelty kuvassa 5. Moottori tarvitsee toimiakseen tälle moottorityypille ja teholle sopivan ohjainlaitteen.

Akusto on koottu kierrätysosista, kannettavien tietokoneiden ja akkutoimisten työkalujen akuista. Akusto koostuu kahdesta moduulista, joista molemmat ovat nimellisjännitteeltään 48 voltia. Moduulit on kytketty rinnan, jolloin voimalinjan lopullinen jännite on edelleen 48 voltia. Näin valmistettu akku ei ole ominaisuuksiltaan ajoneuvokäyttöön erityisen hyvä. Kannettavien tietokoneiden akuissa on käytetty yleensä akkukennoja, joiden virranantokyky ei ole järin suuri. Ajoneuvokäytössä tuota virtaa kuitenkin kaivattaisiin, ja huipputeholla ajettaessa suuri virta aiheuttaa kennon jännitteen alenemisen. Toisinaan jännite alenee niin paljon, että moottorin ohjainlaite katkaisee vedon. Akusto täytyisi rakentaa uudelleen ominaisuuksiltaan mahdollisimman identtisillä ja suuremmalle virralle tarkoitetuilla kennoilla.



Kuva 5. Tunturi-mopo sähkökonversion jälkeen.

Polttonestetankkia ei mopossa enää tarvita, muuten kuin linjakkaan ulkomuodon säilyttämiseksi. Sähkösäiliön ohjainlaite on piilossa tankin alla.

## 5.2 Yhteyskäytäntö

Soveltuvia vaihtoehtoja yhteyskäytännöksi on UART, SPI, CAN ja I2C.

UART (Universal Asynchronous Receiver Transmitter) ei tässä tapauksessa käy, koska väylässä on useita laitteita ja jokaiselle täytyisi vetää oma yhteyskaapeliinsa.

SPI (Serial Peripheral Interface) on periaatteeltaan sopiva, mutta kaksi dataliikenteelle tarkoitettua johdinta on liikaa. Tavoitteena on kuitenkin mahdollisimman yksinkertainen järjestelmä.

CAN (Controller Area Network) on ajoneuvokäyttöön tarkoitettu, tähän työhön täysin soveltuva protokolla. CAN-väylä tarvitsisi kuitenkin arduinojen yhteydessä käytettäväksi erityisiä lisälaitteita jokaiselle solmukohtalle, jolloin järjestelmän

monimutkaisuus kasvaisi heti kertaluokkaa suuremmaksi. Laitteiston yksinkertaisuus tähtäimessä, hylätään CAN.

Valitaan toteutustavaksi I2C (IIC, Inter-Integrated Circuit). Protokollassa on jännite- ja maajohtimen lisäksi vain kaksi johdinta, data- ja kellopulssi johtimet. I2C on 1982 Philips Semiconductorin kehittämä sarjaliikenneyhteys. (Wikipedia: I2C, 2021)

I2C-yhteyskäytännössä voi olla kahdenlaisia laitteita, ohjaavia ja ohjattavia. Ohjaavia laitteita voi olla useampia kuin yksi ja rooleja voi myös vaihtaa tarvittaessa. Rooli määräytyy lähetettävän viestin perusteella.

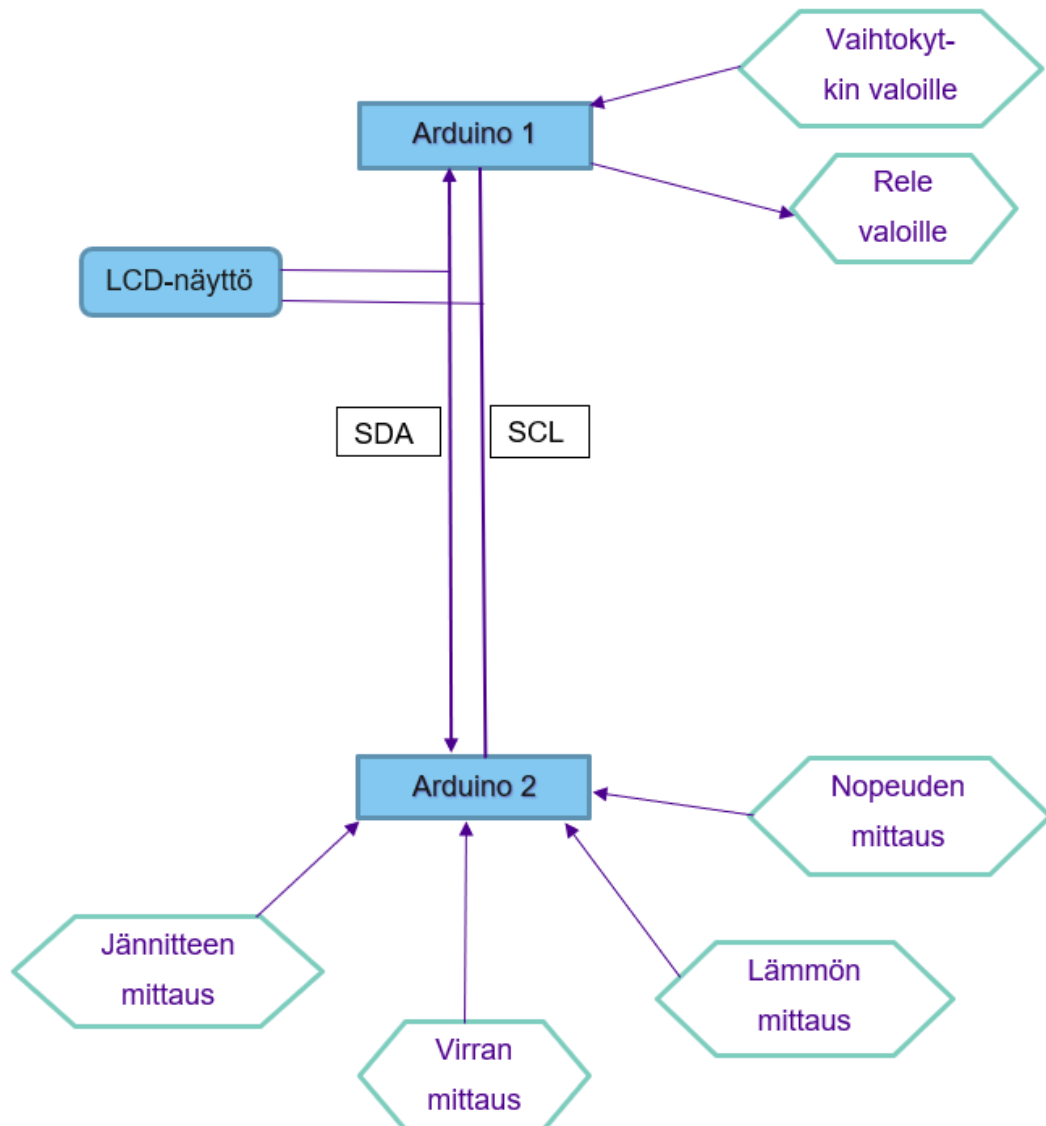
Johtimia on kaksi, SCL (kellopulssi) ja SDA (data). Molemmat linjat tarvitsevat ylösvetovastuksen, jolla johtimen jännite nostetaan viiteen volttiin. Bittejä siirretään vetämällä jännite alas (0-bitti) tai jätetään vetämättä (1-bitti). Jokaisella väylään liitetyllä laitteella on uniikki seitsemänbittinen osoite. Seitsemän bittiä rajoittaa liitettävien laitteiden määrän 128:aan. Toinen laitteiden määrä rajoittava tekijä on väylän kapasitanssi. Liian pitkät johtimet tai liikaa laitteita nostaa kapasitanssia, jolloin jännitteen vaihtelu hidastuu ja viestien välittyminen on epävarmaa. (Wikipedia: I2C, 2021).

Viestiliikenne on synkronista ja tyypillisesti sitä käytetään piirilevyn sisäiseen komponenttien väliseen tiedonsiirtoon. Tässä projektissa väylän pituus on tyypillistä tilannetta pidempi ja johtimet kulkevat sähkömoottorin läheisyydessä, joten tiedonsiirron varmuudessa saattaa ilmetä ongelmia.

### 5.3 Topologia

Alla olevassa vuokaaviossa on esitetty väylän fyysinen rakenne. Kahden arduino-mikrokontrollerin välillä on I2C-väylä, joka koostuu data- ja kellopulssi-johtimista. Kaavioon ei ole piirretty jännitejohtimia. Näyttö liitetään väylään omana laitteenaan.

Arduino 1 ohjaa ajovalon relettä vaihtokytkimen tilan perusteella ja ilmoittaa väylää pitkin valo-ohjauksen tilan LCD-näytölle. Arduino 2 mittaa tietoja akustosta ja moottorista, lähettämien tiedot väylää pitkin näytölle. Näyttö vastaanottaa väylästä tietoja ja esittää ne ajoneuvon kuljettajalle. Kuvassa 6 kuvataan laitteiston asettelua ja kuvassa 7 on hahmoteltu laitteiden ja johtimien sijoittelua ajoneuvoon.



Kuva 6. Väylän toimilaitteiden asettelu ja toiminnot





Kuva 7. Väylän suunniteltu reitti ja toimilaitteiden sijainnit.

#### 5.4 Solmut

Solmuiksi kutsutaan väylään liitettäviä laitteita. Projektissa käytetään solmuina Arduino UNO -mikrokontrollereita, yleisen saatavuutensa, helpon käytön ja edullisuutensa vuoksi. UNOssa on ATmega328P-mikroprosessori ja käyttöjännitteeksi voidaan syöttää 12 voltia, joka mopossa on valmiiksi saatavilla. Arduinossa on valmiina liitäntä I2C-väylää varten. UNOon liitettävät laitteet ja itse I2C-väylä ovat 0-5 voltin jännitealueella toimivia.

#### 5.5 Johtimet

Käytetään suojattua nelijohtimista kaapelia. Johtimia ympäröivässä kuoressa on kuparinen verkko, joka toimii kuten Faradayn häkki, estäen sähkömagneettista säteilyä muodostamasta häiriötä sisempänä kulkeviin johtimiin (Shielded cable, Wikipedia.org 2021). Suojajohdin yhdistetään molemmista päistään sähköjärjestelmän maapotentiaaliin.



Data- ja kellopulssijohtimet tarvitsevat ylösvetovastukset. I2C-protokollassa viiden voltin ylempi jännitetaso luodaan passiivisesti ylösvetovastuksen avulla. Johdin on siis yhdistetty vastuksen kautta viiden voltin jännitteeseen. Kun väylään halutaan alempi jännitetaso, nolla voltia, johdin vedetään aktiivisesti maapotentiaaliin. Arduinossa on sisäänrakennettuna 20kΩ ylösvetovastus, mutta tässä tapauksessa tuo resistanssin arvo saattaa olla liian suuri. Texas Instruments tarjoaa ohjeet sopivan vastusarvon laskemiseksi:

$R_p$ =Ylösvetovastus

$V_{cc}$ =Käyttöjännite

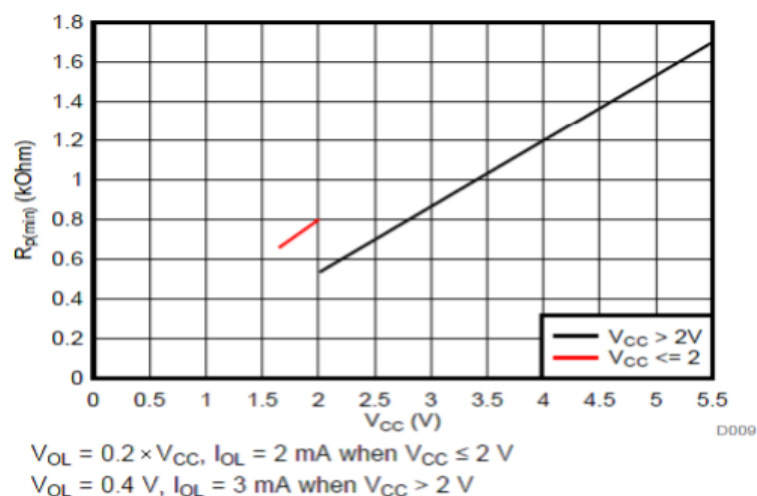
$V_{OL}$ =alajännite (ei ole ihan 0, koska alavetovaiheessa kulkee hieman virtaa)

$I_{OL}$ =alajännitteellä kulkeva virta (Texas Instrumentsin arvio)

$$R_{p\min} = \frac{(V_{cc} - V_{OL})}{I_{OL}} = \frac{5V - 0,4V}{0,3mA} = 1,533k\Omega$$

Kaava 2. Ylösvetovastuksen minimiarvon laskenta. (I2C pullup resistor calculation, Ti.com 2015)

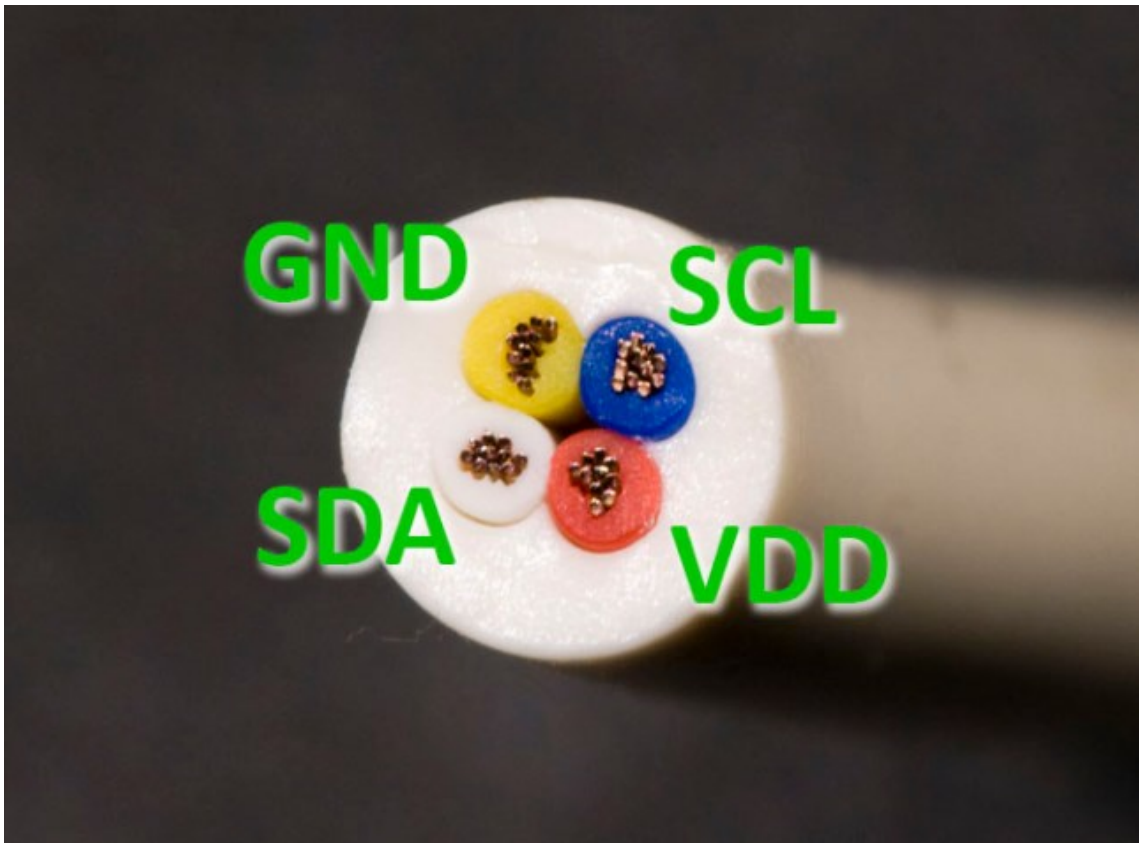
Kuvasta 8 voidaan varmistaa laskenta. Ylösvetovastus täytyy olla vähintään 1,5kΩ viiden voltin käyttöjännitteellä. Jos vastuksen arvo on tätä pienempi, arduino ei välttämättä jaksaa vetää signaalijohtimen jännitettä nolaksi.



Kuva 8. Vastuksen miniarvo käyttöjännitteen funktiona. (I2C pullup resistor calculation, Ti.com 2015)

Johtimien kapasitanssi saattaa aiheuttaa viivettä signaalin nouseviin reunoihin, mikä voi aiheuttaa ongelmia datan siirtymisessä. Tästä syystä ylösvetovastuksen mitoitus mahdollisimman pieneksi parantaa toimintavarmuutta.

Käytettäessä vierekkäisiä johtimia kaapelin sisässä, signaalien ylikuuluminen saattaa muodostua ongelmaksi. Ylösvetovastuksen valinnalla voi vaikuttaa signaalin reunojen jyrkkyyteen, mutta suurempi häiriötä vaimentava vaikutus saadaan valitsemalla johtimet oikein kaapelin sisässä. Kuvassa 9, SDA (datalinja) ja SCL (kellolinja) on valittu siten, että ne ovat mahdollisimman kaukana toisistaan. Tällöin niissä tapahtuvat jännitteenvaihtelut vaikuttavat toisiinsa merkittävästi vähemmän.



Kuva 9. Johtimien asettelu kaapelin sisällä. (axotron.se, 2016)

## 5.6 Näyttö

Valittiin näytöksi yksinkertainen nestekidenäyttö, jossa on kaksi riviä ja molemmille mahtuu 16 merkkiä. Jokaiseen merkkiin sisältyy 40 pikseliä. Pikseleitä tummentamalla saadaan muodostettua erilaisia symboleja. Symboleja voi muodostaa myös itse, määräämällä tietyt pikselit tummiksi ja loput vaaleiksi. Näytössä on valmiina lisäpiiri helppoa I2C-väylään liittämistä varten.

## 5.7 Muut laitteet

Ajovalo laitetaan toimimaan siten, että lyhyet ovat automaattisesti päällä, kun päävirta on kytketty. Kun valitaan vaihtokytkimestä kaukovalot, ohjataan releen kautta jännite kaukovalon napaan. Releenä käytetään viiden voltin ohjauksella toimivaa 10 ampeerin relettä.

Virran mittaus tapahtuu akuston positiiviselta johtimelta. Johtimen ympärille asetetaan hall-efektiin perustuva YHDC:n valmistama anturi, jonka ulostulosignaali on  $2,5 \pm 0,625V$ . Kun virtaa ei kulje, ulostulo on 2,5 voltia. Anturin maksimiarvolla, 50A, ulostulo on 3,125V. Virran kulkiessa toiseen suuntaan, jännite pienenee 2,5 voltista. Tässä konstruktiossa virran kulkusuunta on vakio.

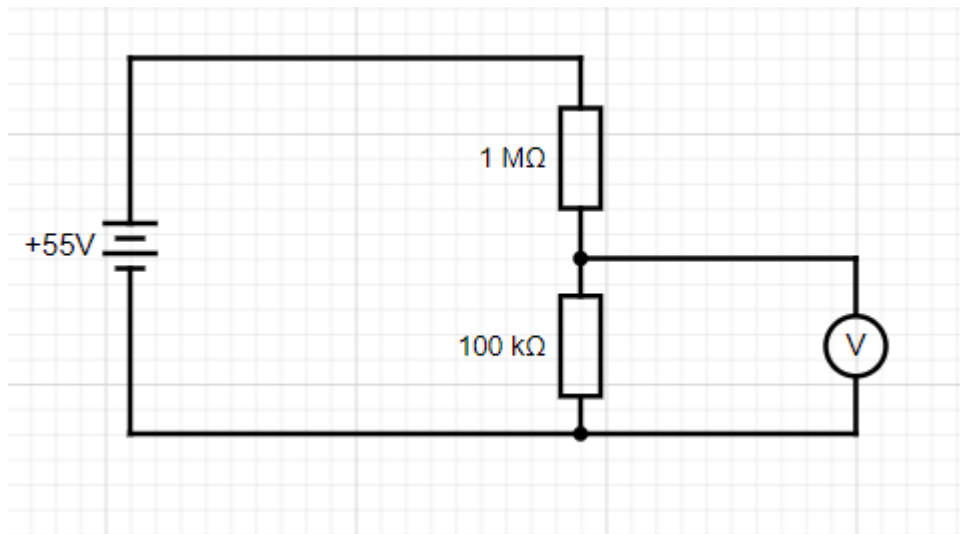
Lämpöanturi on LM35, jonka jännite muuttuu lämpötilan mukaan. Anturin toiminta on tarkemmin selitetty valmistajan datalehdellä, kts. lähteet.

Akuston jännite mitataan jännitteenjakopiiriin avulla. Mikrokontrolleri ei siedä suuria jännitteitä, joten mittausjännite täytyy skaalata pienemmäksi vastuksien avulla. Akuston ollessa täyteen ladattu, sen jännite on 54,6 voltia. Arduino kykenee mittaamaan jännitteitä viiteen volttiin asti, joten mittauspisteen jännite täytyy laskea vastaamaan mittauskykyä. Valitaan toiseksi vastukseksi 100kΩ. Tällöin suuremman vastuksen on oltava kaavan x perusteella 1MΩ.

$$R1 = \frac{R2 * V_{cc}}{V_{ulos}} - R2 = \frac{100k\Omega * 55V}{5V} - 100k\Omega = 1M\Omega$$

Kaava 3. Vastuksen mitoitus jännitejakopiiriin. (Wikipedia: jännitteenjakosääntö)

Muodostetaan vastuksista kuvan 10 mukainen piiri. Akuston suurimman jännitteen ollessa 55 voltia, mittauspisteen jännite on tällä kytkennällä maksimissaan viisi voltia.



Kuva 10. Jännitteenjakokytkentä. (circuit-diagram.org)

## 5.8 Kytkennät

Kohtuullisen pitkien johtimien aiheuttaman kapasitiivisen kuorman vuoksi lisäämään lämpöanturin yhteyteen valmistajan suosituksen mukaiset (kuva 11) häiriöpoistokomponentit. Tarkoituksena on parantaa lämpömittauksen luotettavuutta.

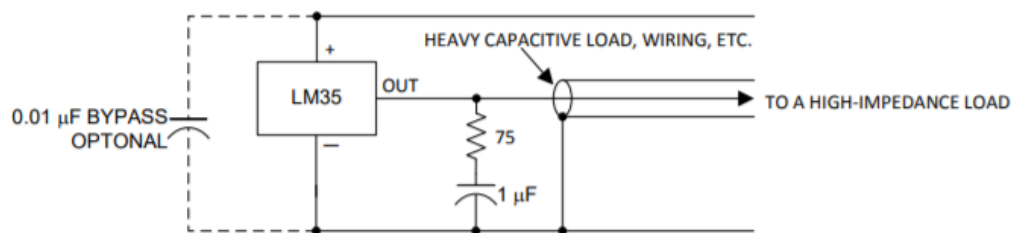
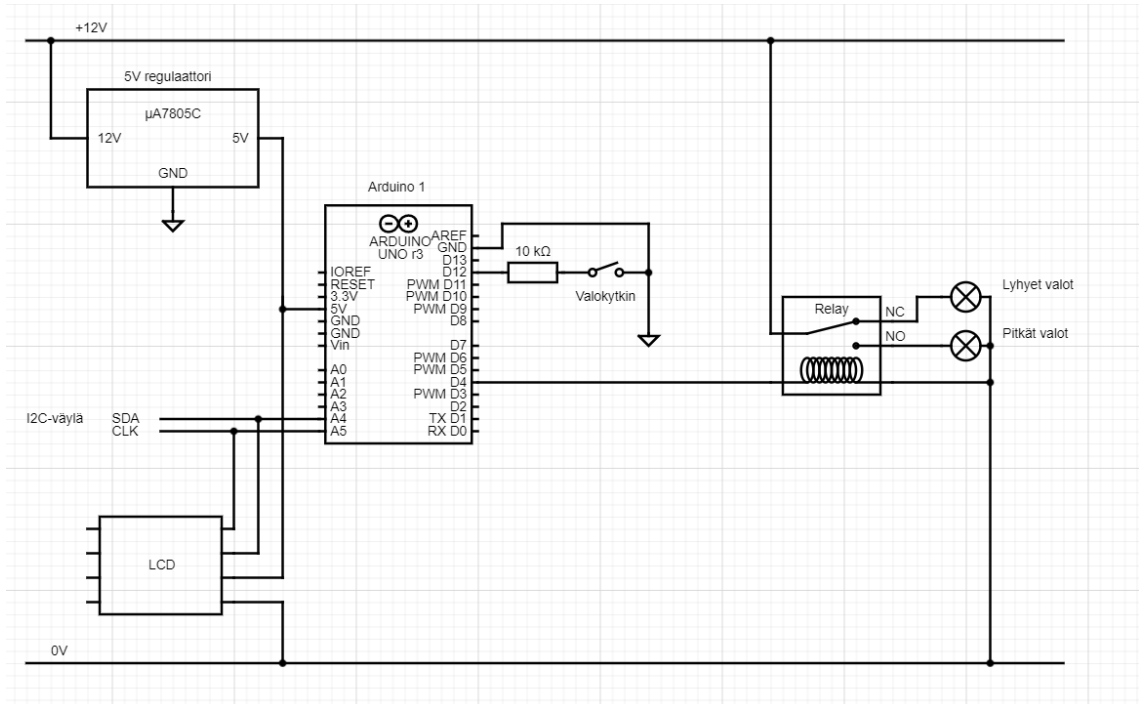


Figure 13. LM35 with R-C Damper

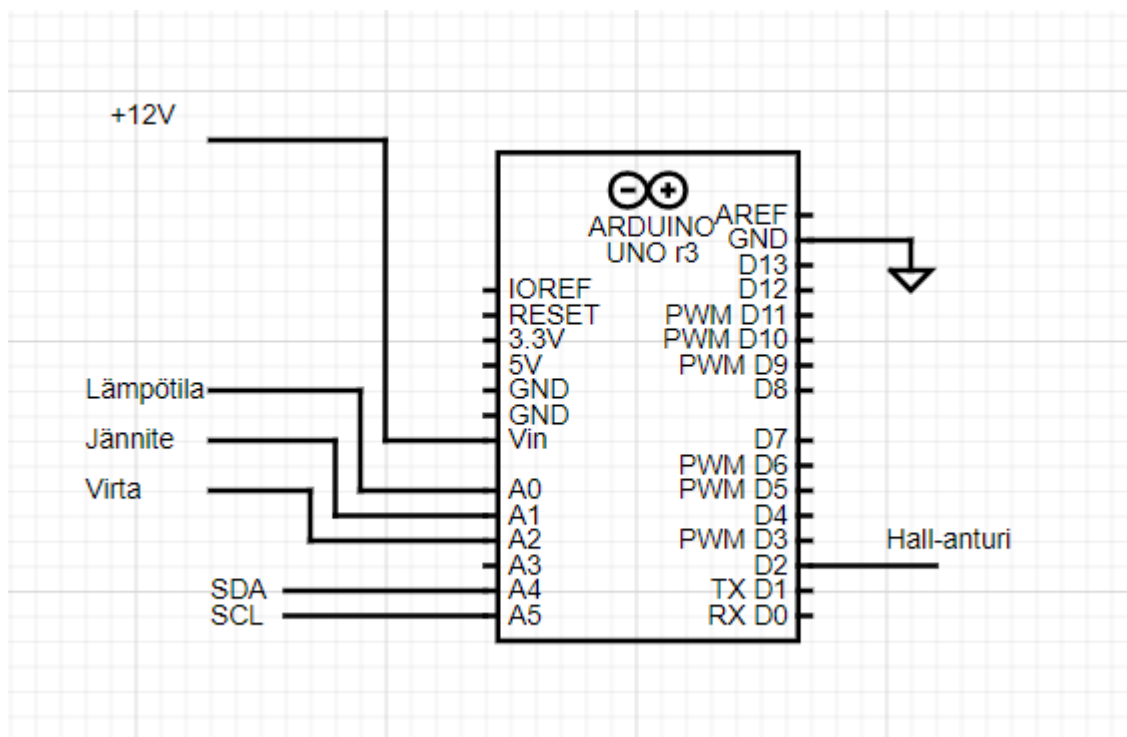
Kuva 11. Lämpöanturin häiriöpoistokytkentä (LM35 datalehti, Ti.com 2017)

Arduino 1 sijoitetaan lähelle ajoneuvon valaisinta. Jännitteen alentaminen 12 voltista viiteen volttiin toteutetaan ulkoisen regulaattorin avulla, koska arduinon oma regulaattori saattaisi ylikuormittua. Kytettävää kuormitusta tulee mikropiirin omasta kulutuksesta, näytöstä, releestä ja kytkimestä. Kuvassa 12 on havainnollistettu kytkennät.



Kuva 12. Arduino 1:n kytkennät (circuit-diagram.org)

Kuvassa 13 on vastaavasti Arduino 2:n kytkennät. Tämä laite saa pärjätä omalla regulaattorillaan, kuormituksen ollessa huomattavasti pienempää. Tällöin jännitteeksi voidaan syöttää suoraan 12 volttia. Kuvasta on selvyys vuoksi piirretty mittalaitteille ainoastaan signaalijohtimet. Jokaiseen menee kuitenkin näiden lisäksi myös käyttöjännite ja maa. Liitin D2 yhdistetään suoraan moottorin ja moottorinohjainlaitteen väliseen hall-anturin signaalijohtimeen. Tästä johtimesta luetaan signaalit roottorin liikkeistä.



Kuva 13. Arduino 2:n kytkennät (circuit-diagram.org)

## 5.9 Ohjelmisto

Ohjelmisto jakautuu kahteen osaan: Arduino 1 ja Arduino 2.

### Arduino 1

- Ohjaa ajovalon relettä valokytkimen asennon perusteella
- Pyytää Arduino 2:lta mittaustietoja ja tallentaa ne matriisiin
- Lähettää näyttölaitteelle kuljettajalle ilmaistavat tiedot

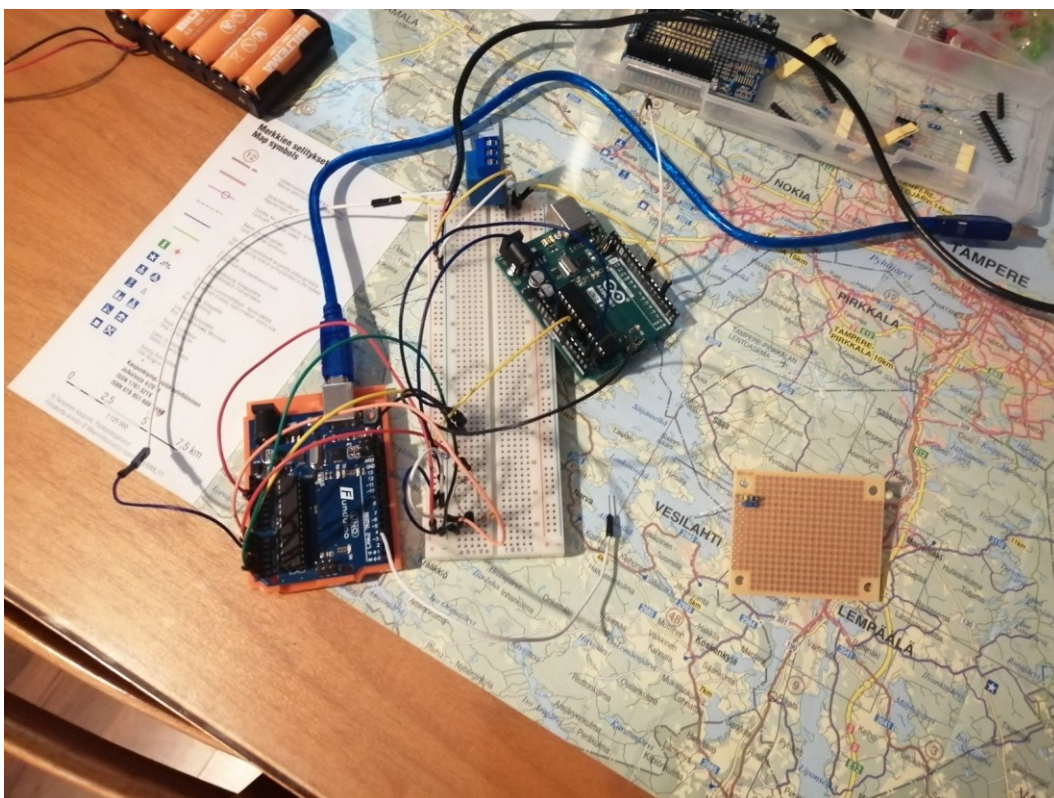
### Arduino 2

- Kerää tietoja neljältä anturilta
- Laskee mittausarvot käyttökelpoiseen muotoon
- Muodostaa luvuista matriisin
- Pyydettyessä lähettää matriisin sisällön ohjaavalle laitteelle

Ohjelmointi toteutettiin Arduino IDE -ohjelmalla. Käytettävä ohjelmointikieli on C++. Molempien mikrokontrollereiden koodit ovat liitteinä 1 ja 2.

## 6 RAKENTAMINEN

Rakentaminen aloitettiin pöydällä sisätiloissa. Kytkentöjen toimivuus testattiin ennen kuin laitteisto rakennettiin kiinni ajoneuvoon. Kuvassa 14 on osa järjestelmän komponenteista testausvaiheessa.



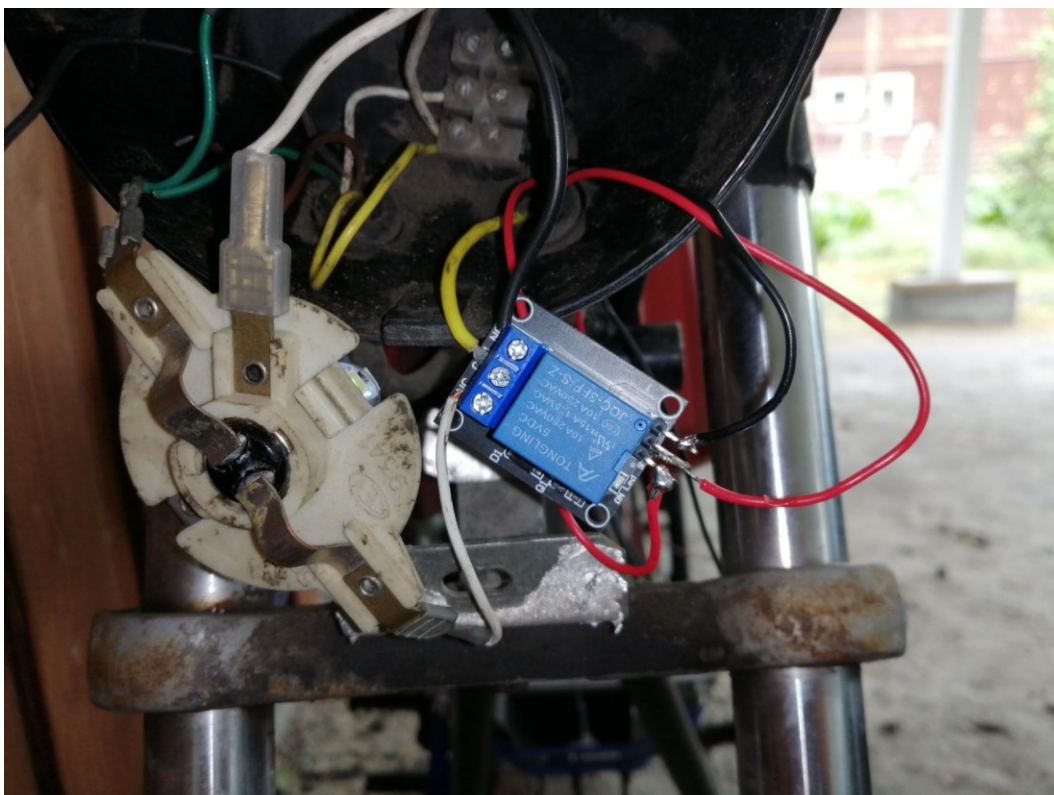
Kuva 14. Kytkentöjen testausta ennen asennusta.

Kuvassa 15 on lämpötila-anturin kytkentä, lisättynä häiriöitä vaimentavilla kondensaattoreilla. Mittausten luotettavuus parani tällä kytkennällä huomattavasti. Kuvissa 16, 17 ja 18 esitetään laitteiden sijoittelua ajoneuvossa. Kuvassa 19 on LCD-näyttö ja siinä näkyvät mittaustiedot. Näytön oikeassa reunassa on ajovalon tilan osoittava symboli. Kun valonsäteet osoittavat suoraan sivulle, on pitkät ajovalot päällä. Säteiden osoittaessa alaviistoon, kyse on lyhyistä ajovaloista. Kuva 20 on otettu kun laitteisto on saatu kasattua ja suoritetaan koeajoa.

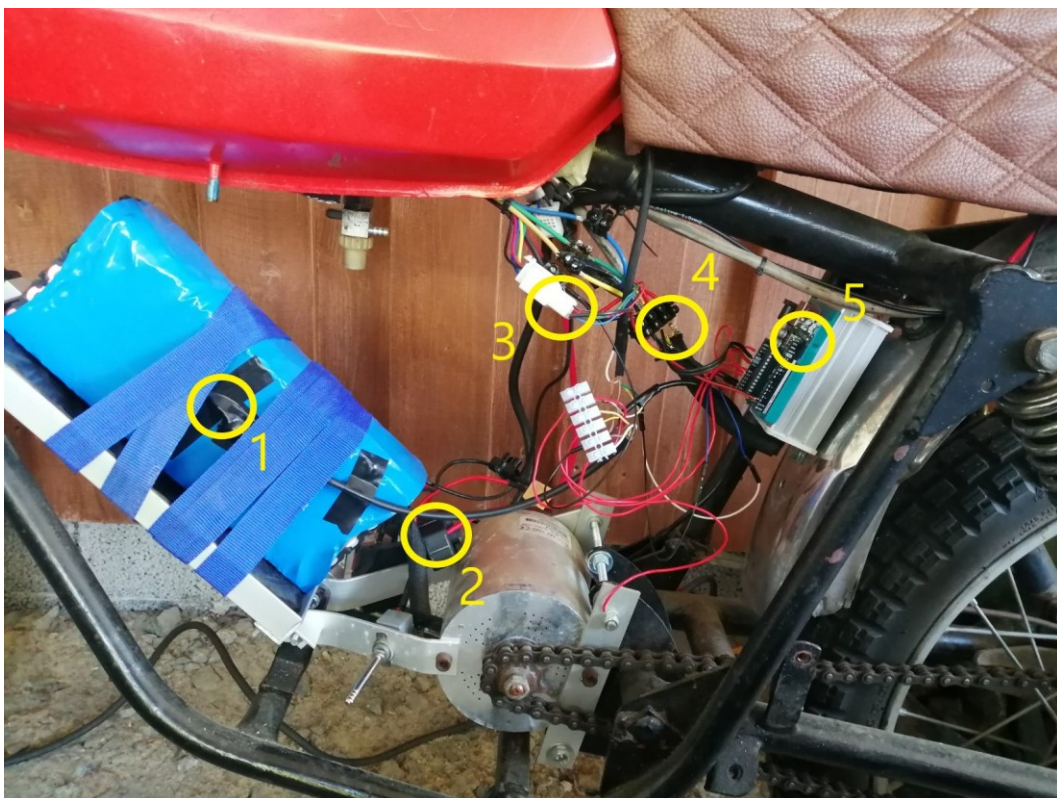




Kuva 15. Lämpötila-anturin kytkentä.



Kuva 16. Ajovaloa ohjaava rele sijoitettiin valoumpion sisään.

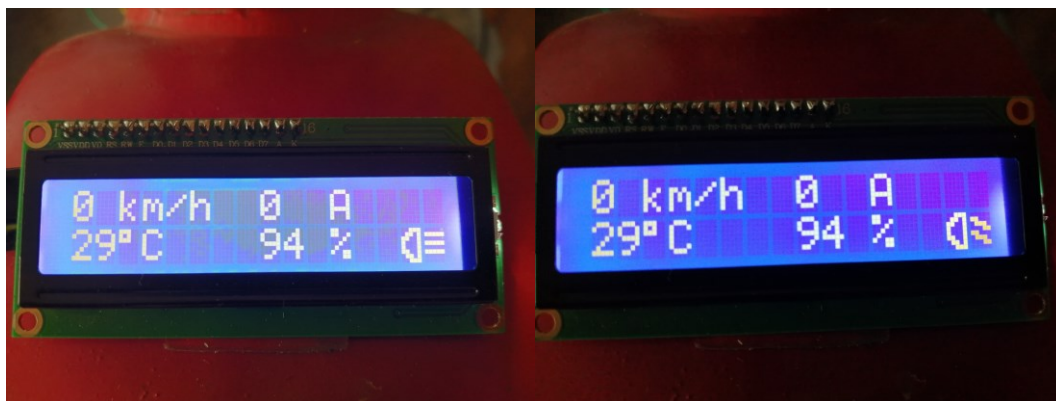


Kuva 17. Mittauspisteiden sijoittelu. (1) Lämpötila (2) Virta (3) Hall-anturi (4) Jännitemittaus (5) Arduino 2



Kuva 18. Väylän toinen pää. (1) Rele umpion sisällä (2) Arduino 1 (3) LCD-näyttö





Kuva 19. Näytöllä esitetään mittausarvot. Valosymboli muuttuu kytkimen asennon perusteella.



Kuva 20. Ajotilanne.

## 7 POHDINTA

Projektin tavoitteisiin päästiin kohtalaisen hyvin. Kaikki suunnitellut toiminnallisuudet saatiin mukaan, eikä mitään ylitsepääsemätöntä estettä ilmaantunut. Tiedonsiirto on jokseenkin luotettavaa ja lopputulokseen täytyy olla tyytyväinen.

Yksi niistä syistä, miksi ajoneuvoissa käytetään väyliä, on johdinten määrän väheneminen. Tässä tapauksessa näin ei kuitenkaan käynyt, vaan projektin lopussa mopon sähköisten johdinten määrä oli moninkertainen aloitushetkeen nähden. Väylätekniikan käyttö nykyaikaisissa liikkuvissa koneissa on kuitenkin hyvin perusteltua, väylään liitettävien laitteiden suuren lukumäärän takia.

Joitain kehityskohteita projektin valmistuttua nousi esiin. Häiriönsuojausta voisi entisestään parantaa hetkittäisten mittausarvojen epävarmuuksien poistamiseksi. Tiedonsiirtonopeuden laskeminen vähentäisi yksittäisten bittien väärin tulkintaa. Väylän ylösvetovastuksina käytettiin mikropiireissä valmiina olleita vastuksia. Jos käytettäisiin suunnitteluvaiheessa laskettua ylösvedon vastusarvoa, signaalin nouseva reuna olisi jyrkempi ja häiriöitä oletettavasti saataisiin vähennettyä.

Näytössä esitettäviä tietoja voisi kehittää laskemalla esimerkiksi jäljellä olevan toimintamatkan akun varauksen ja hetkellisen virrankulutuksen perusteella. Myös virrankulutuksen kasvaessa tapahtuva jännitteenalenema olisi hyvä ottaa huomioon ilmoitettaessa akuston varausprosenttia. Akuston sisäisen resistanssin voisi laskea kuormittamalla sitä jollain tietyllä tasaisella virralla ja mittaamalla sitä vastaava alentunut jännite. Akuston varaustilan laskennassa jännitteenaleneman voisi tällöin kompensoida pois.

Yksi signaalin laatua parantava muutos olisi lisätä kondensaattorit jokaisen väylään liitetyn laitteen virransyötön lähelle. Kondensaattorien puuttuessa äkkinäiset muutokset laitteen virrankulutuksessa väistämättä aiheuttavat häiriöitä lähellä kulkeviin signaali- ja kellopulsseihin.

Projektista jäi hyvin kokemusta sulautetun järjestelmän suunnittelusta ja toteutuksesta. Jatkossa samankaltaisten projektien toteutus onnistuu matalammalla kynnyksellä.

## LÄHTEET

Arduino language reference, Arduino.cc

<https://www.arduino.cc/reference/en/>

Arduino communication protocols tutorial, Arduino.cc

<https://www.deviceplus.com/arduino/arduino-communication-protocols-tutorial/>

Twisted pair, Wikipedia.org

[https://en.wikipedia.org/wiki/Twisted\\_pair](https://en.wikipedia.org/wiki/Twisted_pair)

Differential signaling, Wikipedia.org

[https://en.wikipedia.org/wiki/Differential\\_signaling](https://en.wikipedia.org/wiki/Differential_signaling)

Porter, D 2018: Ethernet, the evolution on automotive networking, Texas Instruments

[https://www.ti.com/lit/wp/szzy009/szzy009.pdf?ts=1626429961612&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/wp/szzy009/szzy009.pdf?ts=1626429961612&ref_url=https%253A%252F%252Fwww.google.com%252F)

Mikrokontrolleri, Wikipedia.org

<https://fi.wikipedia.org/wiki/Mikrokontrolleri>

Hall-effect, Wikipedia.org

[https://en.wikipedia.org/wiki/Hall\\_effect](https://en.wikipedia.org/wiki/Hall_effect)

Hall-effect, Electronics Tutorials

<https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html>

I2C, Wikipedia.org

<https://en.wikipedia.org/wiki/I%C2%B2C>

Jännitteen- ja virranjakosääntö, Wikipedia.org

[https://fi.wikipedia.org/wiki/J%C3%A4nnitteen- ja\\_virranjakos%C3%A4%C3%A4nt%C3%B6](https://fi.wikipedia.org/wiki/J%C3%A4nnitteen- ja_virranjakos%C3%A4%C3%A4nt%C3%B6)

Diode-based temperature measurement, Texas Instruments

[https://www.ti.com/lit/an/sboa277a/sboa277a.pdf?ts=1621700617063&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/sboa277a/sboa277a.pdf?ts=1621700617063&ref_url=https%253A%252F%252Fwww.google.com%252F)

Arora, R. 2015. I2C Pullup resistor calculation, Texas Instruments

<https://www.ti.com/lit/an/slva689/slva689.pdf>

Magnusson, P. 2016. Crosstalk problems when running I2C signals in a cable.

<http://axotron.se/blog/crosstalk-problems-when-running-i2c-signals-in-a-cable/>

Sähkötyöt- ja urakointi, Tukes.fi

<https://tukes.fi/sahko/sahkotyot-ja-urakointi/>

LM35 Precision centigrade temperature sensors datasheet, Texas Instruments

<https://www.ti.com/lit/ds/symlink/lm35.pdf>

Spot the difference: Lithium ion versus lead acid battery electric technology,  
Cummins Inc. 2019

<https://www.cummins.com/news/2019/06/17/spot-difference-lithium-ion-versus-lead-acid-battery-electric-technology>

## LIITTEET

### Liite 1. Arduino 1:n ohjelmakoodi

1 (2)

/\*Tämä koodi on Arduino 1:stä varten. Arduino 1 on määräävä laite, joka kysyy arduino 2:lta tiedot sen mittauksista ja lähettää tiedot kootusti LCD-näytölle. Lisäksi arduino 1 ohjaa valorelettä vaihtokytkimen tilan perusteella. \*/

```
//sisällytetään tarvittavat kirjastot:
```

```
#include <Wire.h> //I2C-tietoliikenne
```

```
#include <LiquidCrystal_I2C.h> //LCD-näyttö
```

```
//määritellään muuttujat ja portit
```

```
int kaukovalokytkin=12;
```

```
int kaukovaloTila=0;
```

```
int kaukovalo=4;
```

```
//määritellään datamatriisi, johon tallennetaan väylästä tuodut tiedot
```

```
int data[4]; //0=lämpö,1=jännite,2=virta,3=nopeus
```

```
//määritellään näytön osoite ja koko:
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

```
//luodaan symbolit pitkille ja lyhyille valoille
```

```
byte lyhyet[8] = {
```

```
  B00000,
```

```
  B11000,
```

```
  B01110,
```

```
  B00011,
```

```
  B11000,
```

```
  B01110,
```

```
  B00011,
```

```
  B00000
```

```
};
```

```
byte lamppu[8] = {
```

```
  B00111,
```

```
  B01101,
```

```
  B11001,
```

```
  B10001,
```

```
  B10001,
```

```
  B11001,
```

```
  B01101,
```

```
  B00111
```

```
};
```

```
byte pitkat[8] = {
```

```
  B00000,
```

```
  B11111,
```

```
  B00000,
```

```
  B11111,
```

```
  B00000,
```

```
  B11111,
```

```
  B00000,
```

```
  B00000
```

```
};
```



```

void setup() { //alustetaan tarvittavat toiminnot
  pinMode(kaukovalo,OUTPUT);
  pinMode(kaukovalokytkin,INPUT_PULLUP);
  Wire.begin(); //liitytään väylään ilman osoitetta, masterina
  lcd.init(); //näytön alustus
  lcd.clear(); //näytön tyhjennys
  lcd.backlight(); //varmistetaan taustavalo
}

void loop() { //varsinainen ohjelmaluuppi
  Wire.requestFrom(2,4); //pyydetään väylän kakkoslaitteelta neljä tavua
  for(int i=0;i<4;i++){
    data[i]=Wire.read(); //tallennetaan tuleva data matriisiin
  }

  //lähetetään LCD-näytölle kirjoitettavat tiedot
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(data[3]); //nopeus km/h
  lcd.setCursor(2,0);
  lcd.print("km/h");
  lcd.setCursor(8,0);
  lcd.print(data[2]); //virta ampeereina
  lcd.setCursor(11,0);
  lcd.print("A");
  lcd.setCursor(0,1);
  lcd.print(data[0]); //akun lämpötila
  lcd.setCursor(2,1);
  lcd.print((char)223); //asteen merkki
  lcd.print("C");
  lcd.setCursor(8,1);
  lcd.print(data[1]); //akun varaus prosentteina
  lcd.setCursor(11,1);
  lcd.print("%");

  //ajovalon tila ilmoitetaan omalla kahden merkin symbolilla. kaukovalokytkimen
  //ollessa kytkettynä (maadoitettuna), annetaan valoa ohjaavalle releelle signaali
  lcd.createChar(1,lamppu);
  lcd.setCursor(14,1);
  lcd.write(1);
  kaukovaloTila=digitalRead(kaukovalokytkin);
  if(kaukovaloTila==false){
    lcd.createChar(2,pitkat); //luodaan merkki
    lcd.setCursor(15,1); //määritetään paikka
    lcd.write(2); //kirjoitetaan luotu merkki
    digitalWrite(kaukovalo,HIGH); //valoreleelle signaali
  }
  else {
    lcd.createChar(3,lyhyet);
    lcd.setCursor(15,1);
    lcd.write(3);
    digitalWrite(kaukovalo,LOW);
  }
  delay(1000); //odotetaan sekunti
}

```

## Liite 2. Arduino 2:n ohjelmakoodi

1 (2)

```

/* Tämä koodi on arduino 2 -laitetta varten. Laitte kerää tiedot neljästä anturista, pyydetty-
essä lähettää
tiedot kootusti yhdessä matriisissa väylää pitkin ohjaavalle laitteelle. */

// lisätään tarvittavat kirjastot
#include <Wire.h>

//esitellään muuttujat
const int T_mittaus=A0;
const int V_mittaus=A1;
const int A_mittaus=A2;
const byte interruptPin=2;
volatile unsigned long dTime=0,vanhaAika=0,aika=0;
int rpm=0; //kierrosluvun laskentaan
float valitys=2.48; //eturatas 9, takaratas 42, renkaan halkaisija 60cm
int data[4]; //matriisi helppoon tiedonsiirtoon, 0=lampo,1=jännite,2=virta,3=nopeus
unsigned long laskuri=0; //laskee keskeytyspulssien määrää

//alustetaan ohjelman toiminnot
void setup() {
  pinMode(T_mittaus,INPUT);
  pinMode(V_mittaus,INPUT);
  pinMode(A_mittaus,INPUT);
  pinMode(interruptPin,INPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin),pulssi,RISING); //nousevan reunan
  //tullessa keskeytyspalvelu suorittaa funktion: pulssi
  Wire.begin(2); //liitytään väylään osoitteella kaksi
  Wire.onRequest(tapahtuma); //kyselyn tullessa suoritettava funktio
}

//määritellään tapahtuma-niminen funktio
void tapahtuma() {
  for (int i=0; i<4; i++){ //neljä kertaa suoritettava luuppi
    Wire.write(data[i]); //lähetetään datamatriisin sisältö
  }
}

//määritellään pulssi-niminen funktio
void pulssi(){
  laskuri++; //lisätään laskurin arvoa yhdellä
}

//pääohjelman luuppi
void loop() {
  //lasketaan moottorin kierrosluku
  noInterrupts(); //ei keskeytyksiä laskennan aikana
  aika=millis(); //laskennan ajan hetki
  dTime=aika-vanhaAika; //kulunut aika edellisestä laskennasta
  vanhaAika=aika;
  rpm=(laskuri/4*60000)/dTime; //kierrosluku
  interrupts();
  laskuri=0; //keskeytyslaskurin nollaus
}

```

2 (2)

```
//otetaan lämpötilasta viiden mittauksen keskiarvo ja tallennetaan tieto matriisiin soluun 0
int Tka=0; //uusi muuttuja keskiarvon laskentaan
for(int x=0;x<5;x++){
    Tka=Tka+(analogRead(T_mittaus)/2);
}
data[0]=Tka/5;

//luetaan jännite, muutetaan prosenteiksi täydestä varauksesta ja tallennetaan se
//matriisiin soluun 1
data[1]=analogRead(V_mittaus)/10;

//luetaan virta ja tallennetaan se matriisiin soluun 2
float apuluku=((analogRead(A_mittaus)*5.0/1023.0)-2.52)*31.25; //lasketaan anturin
//jännitteestä virran arvo
int virta=round(apuluku); //muunnos kokonaisluvuksi
data[2]=virta;

//lasketaan nopeus ja tallennetaan se matriisiin soluun 3
data[3]=rpm*valitys*60000;

delay(1000); //odotetaan sekunti ennen uutta kierrosta
}
```