



Scott Drive-väyläohjaus ajoneuvosovelluksissa

Sami Vähäpesola

Opinnäytetyö, AMK

Tammikuu 2024

Sähkö- ja automaatiotekniikan tutkinto-ohjelma (AMK)

Vähäpesola, Sami

Scott Drive-väyläohjaus ajoneuvosovelluksissa

Jyväskylä: Jyväskylän ammattikorkeakoulu. Tammikuu 2024, 56 sivua.

Sähkö- ja automaatiotekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Työ toteutettiin Lielahden autokeskus Oy:n toimeksiantona. Työn päätavoitteena oli selvittää, kuinka Scott Drive -ohjainta voidaan hyödyntää CAN-väylän avulla sähkömoottorin ohjauksessa ja miten tämä teknologia voidaan integroida monipuolisesti erilaisiin ajoneuvosovelluksiin. Tämä pyrkimys johti käytännönläheisen ohjeistuksen luomiseen, jonka avulla voidaan helpottaa Scott Drive -ohjaimen käyttöönottoa eri ajoneuvoissa.

Työn monivaiheinen lähestymistapa käsitti ohjaimen ja tarvittavien laitteiden suunnittelun, asennuksen, väylän suunnittelun ja rakentamisen, sekä ohjaimen hallintaan liittyvän ohjelmoinnin Codesys-ohjelmistoa hyödyntäen. Teoriaosuudessa keskityttiin perusteellisesti CAN-väylän käsitteisiin, kerroksiin ja erityisesti CANOpen-protokollaan, mikä antoi vahvan teoreettisen pohjan käytännön toteutukselle.

Varsinainen väyläohjauksen käyttöönotto suoritettiin sähkölaboratoriossa, missä ohjaimen, moottorin ja muiden laitteiden asennus ja analoginen koekäyttö varmisti normaalin toiminnan. Onnistuneiden testien jälkeen siirryttiin väylän suunnitteluun laitteiden välille, samoin kuin ohjaimeksi suunnitellun näytön ja väylän ohjelmointiin. Väyläohjaus integroitiin ajoneuvokäyttöön tarkoitetulla kosketusnäytöllä, joka mahdollisti ohjaimen helpon hallinnan näytölle luoduilla säätimillä. Kosketusnäytölle luotiin myös ohjelma, joka mahdollisti ohjaimen lähettämien parametrien vastaanoton.

Työn kaikki vaiheet, mukaan lukien suunnittelu, asennus ja ohjelmointi, dokumentoitiin tarkasti. Näistä tiedoista koostettiin kattava kirjallinen esitys, jonka avulla Scott Drive -ohjainta voidaan hyödyntää laajasti erilaisissa ajoneuvosovelluksissa. Kirjallisessa esityksessä korostettiin helppolukuista ja ymmärrettävää kokonaisuutta, jonka avulla ohjaimen käyttöönottoon ja hyödyntämiseen ei vaadita syvällistä perehtymistä CAN-väylän ominaisuuksiin. Tämä dokumentaatio toimii arvokkaana resurssina niille, jotka haluavat hyödyntää Scott Drive -ohjainta tehokkaasti ja vaivattomasti sähköajoneuvojen ohjauksessa.

Avainsanat (asiasanat)

CAN-väylä, CANOpen, väyläohjaus, sähköajoneuvot

Vähäpesola, Sami

Scott Drive bus-control in vehicle applications

Jyväskylä: JAMK University of Applied Sciences, January 2024, 56 pages.

Degree Programme in Electric and automation Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The work was carried out within the framework of Lielahden autokeskus Oy. The main objective of the project was to investigate how the Scott Drive controller can be utilized through the CAN bus for electric motor control and how this technology can be integrated into various vehicle applications. This endeavor led to the creation of a practical guide aimed at facilitating the implementation of the Scott Drive controller in different vehicles.

The multi-phase approach of the project encompassed the design, installation, bus planning, and construction of the controller and necessary devices. Additionally, programming for controller management was conducted using Codesys software. The theoretical part focused extensively on the concepts of the CAN bus, its layers, and particularly the CANOpen protocol, providing a strong theoretical foundation for the practical implementation.

The actual deployment of bus control took place in an electrical laboratory, where the installation of the controller, motor, and other devices, along with analog testing, ensured normal functionality. After successful testing, attention shifted to bus planning between devices, as well as the programming of the display designed as a controller. Bus control was integrated using a touchscreen designed for vehicle use, enabling easy management of the controller through created controls on the screen. Additionally, a program was developed for the touchscreen to facilitate the reception of parameters sent by the controller.

All stages of the work, including design, installation, and programming, were meticulously documented. These details were compiled into a comprehensive written presentation, providing a valuable resource for utilizing the Scott Drive controller in a wide range of vehicle applications. The written presentation emphasized a user-friendly and understandable approach, eliminating the need for in-depth familiarity with the features of the CAN bus. This documentation serves as a valuable tool for those seeking to integrate the Scott Drive controller efficiently and seamlessly into electric vehicle control systems.

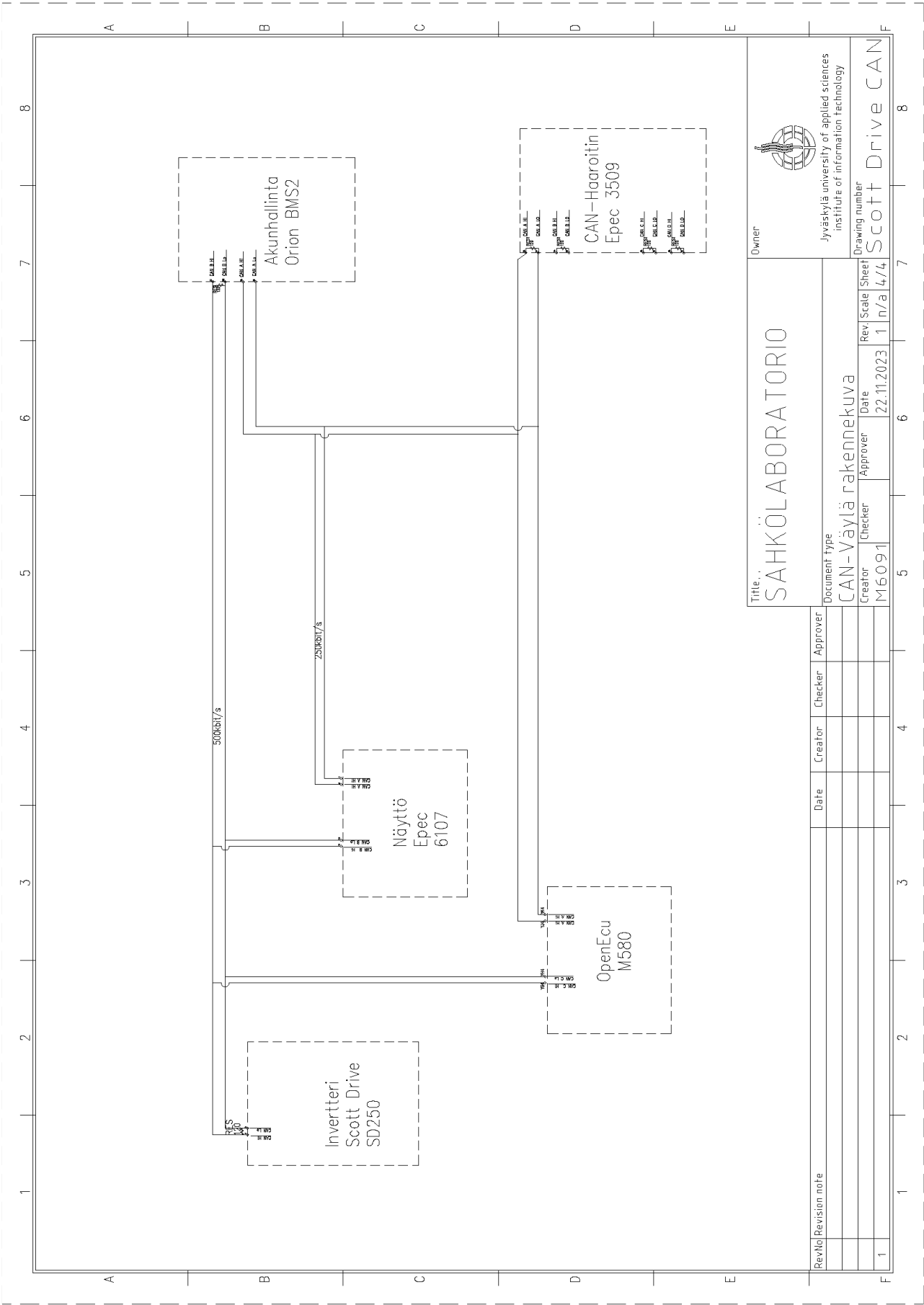
Keywords/tags (subjects)

CAN-bus, CANOpen. bus control, electric vehicles

Sisältö

KÄYTETYT TERMIT JA LYHENTEET	9
1 Johdanto	11
1.1 Tausta, tavoitteet sekä rajaukset	11
1.2 Käytetyt menetelmät ja prosessi	12
1.3 Lielahden autokeskus Oy.....	14
2 Opinnäytetyössä käytettävä laitteisto ja ohjelmistot.....	15
2.1 Scott Drive SD 250.....	15
2.2 Emrax 268.....	16
2.3 Epec 6107	17
2.4 Orion BMS 2	17
2.5 OpenECU M580.....	18
2.6 Epec MultiTool Creator	19
2.7 Codesys 3.5.....	19
2.8 SEV-View.....	20
2.9 PCAN View.....	20
2.10 Simulink.....	21
3 CAN-väylä.....	22
3.1 CAN-väylän toiminta	22
3.1.1 Fyysinen Kerros.....	23
3.1.2 Sidos-, verkko- ja kuljetuskerrokset.....	25
3.1.3 Sovelluskerros.....	25
3.2 CAN-Protokolla	25
3.3 CAN-viestikehys.....	26
3.3.1 Scott Drive CAN-viestit.....	29
4 Työn toteutus	31
4.1 Laitteiston rakentaminen	31
4.2 Väylän rakentaminen	34
4.3 Parametrien vastaanotto näyttöön.....	37
4.4 Moottorin ohjaus	40
4.4.1 CAN-väylä ohjauksen käyttöönotto.....	41
4.4.2 Ohjauksen ohjelmointi CODESYS ohjelmistolla	44

5	Pohdinta ja lopputulokset.....	48
	Lähteet	50
	Liite 1. Program- ja Engine Mode State (Scott Drive n.d. 74.)	54
	Liite 2. CAN-ohjaus viestit (Scott Drive n.d. 22-24.).....	55
	Liite 3. 12 V ja 24 V kytkennät.....	56
	Liite 4. High Voltage kytkennät	57
	Liite 5. CAN-väylä	58



Kuvio 1 Scott Drive SD250.....	15
Kuvio 2 Emrax 268 moottori	16
Kuvio 3 Emrax 268 HV-kytkennät	16
Kuvio 4 Epec 6107 näyttö.....	17
Kuvio 5 Orion BMS 2	17
Kuvio 6 OpenEcu m580.....	18
Kuvio 7 Epec MultiTool Creator	19
Kuvio 8 Codesys logo.....	19
Kuvio 9 SEV-View päänäyttö	20
Kuvio 10 PCAN View.....	20
Kuvio 11 Simulink.....	21
Kuvio 12 OSI-mallin kerrokset.....	23
Kuvio 13 CAN-väylä periaatekuva23	
Kuvio 14 CAN-väylä jännitteet	24
Kuvio 15 CAN-väylän häiriönsuodatus.....	24
Kuvio 16 CANopen-viestikehys	27
Kuvio 17 CAN-viestin muoto Orion BMS.....	27
Kuvio 18 COB-ID selite	28
Kuvio 19 Orion BMS2 CAN-viestien määrittely.....	28
Kuvio 20 Parametripyyntö	29
Kuvio 21 CAN-väylän kautta saatavat parametrit.....	30
Kuvio 22 Laitteiston käyttöpaneeli	31
Kuvio 23 Scott Drive-liitännät	32
Kuvio 24 Scott Drive HV-liitännät.....	32
Kuvio 25 CAN-liitin ohjaimessa	34
Kuvio 26 CAN-väylä fyysisesti.....	34
Kuvio 27 Multitool Creator CAN-verkko	35
Kuvio 28 Multitool PDO viestit näytölle.....	36
Kuvio 29 Scott Drive CAN-viestit Multitool Creatorissa.....	36
Kuvio 30 Hahmotelma viestirakenteesta.....	37
Kuvio 31 Parametrisivu valintänäyttö.....	38
Kuvio 32 Parametrinäyttö	39
Kuvio 33 Parametrinäyttö parametrit.....	39

Kuvio 34 Päänäyttö	40
Kuvio 35 SEV View Communication setting valikko	40
Kuvio 36 CAN-väylä ohjauksen konfigurointi viestikehys	41
Kuvio 37 CAN-väylä ohjauksen vastauskehys	41
Kuvio 38 CAN-väylä digitaalisen ohjauksen ohjauskehys	42
Kuvio 39 CAN-väylä digitaalisen ohjauksen vastauskehys	42
Kuvio 40 CAN-ohjaus analogisten parametrien ohjauskehys	42
Kuvio 41 CAN-ohjaus analogisten parametrien vastauskehys	43
Kuvio 42 CAN-väylä muuttujat CODESYS	44
Kuvio 43 CAN-viestikehys muuttujilla	44
Kuvio 44 CAN-ohjauksen käyttöönotto LD	45
Kuvio 45 Kosketusnäytön päänäyttö muuttujanimillä	45
Kuvio 46 CAN-control kaasuvivun asento	46
Kuvio 47 CAN-ohjaus pääohjelma CODESYS	46
Kuvio 48 CAN-väylä ohjaus pois käytöstä	47

KÄYTETYT TERMIT JA LYHENTEET

BMS	Battery management system akunhallintajärjestelmä, valvoo ja hallitsee akuston yksittäisiä kennoja
CAN	Communication Area Network erityisesti ajoneuvoissa käytetty tiedonsiirto väylä
COB-ID	Communication Object Identifier CANOpen protokollan 11-tavuisen kehyksen tunnus
ECU	Electric Control Unit ajoneuvon moottorinhallinta yksikkö, tyypillisesti polttomoottoriajoneuvoissa
HV	High Voltage Korkea jännite, työssä viitataan 355V järjestelmään
IGBT	Insulated-Gate Bipolar Transistor Suuritehoisiin sovelluksiin käytetty transistori
ISO	International Standardization Organization Kansainvälinen standardointi järjestö
LD	Ladder-ohjelmointi Yksi PLC-ohjelmoinnissa käytettävistä ohjelmointikielistä
LSB	Least Significant Bit Tietokoneen tapa käsitellä lukuarvoa, jossa luku tulkitaan oikealta vasemmalle
LV	Low Voltage Matala jännite, työssä viitataan 12/24V järjestelmään
MSB	Most Significant BIT Tietokoneen tapa käsitellä lukuarvoa, jossa luku tulkitaan oikealta vasemmalle
OSI	Open System Interconnection ISO järjestön kehittäämä malli kommunikaatioprotokollien vertailuun

PDO	Process Data Object CANOpen protokollan mukainen viestikehys
PLC	Programmable Logical Controller automaatioprosessien ohjaamiseen tarkoitettu tietokone
ST	Structured Text yksi PLC-ohjelmoinnissa käytetyistä ohjelmointikielistä
VCU	Vehicle Control Unit sähköinen ajoneuvon hallinta yksikkö, vastaa ECUa

1 Johdanto

1.1 Tausta, tavoitteet sekä rajaukset

Liikenteen sähköistyminen on pinnalla oleva trendi, globaalit ilmastotoimenpiteet ja tavoitteet todennäköisesti tulevat lähitulevaisuudessa voimistamaan tätä entisestään. Fossiilisten polttoaineiden kasvavat kustannukset ohjaavat ihmisiä muiden vaihtoehtojen pariin. Vuoden 2023 tilastojen valossa, Suomen henkilöautokannasta 7,1 % käytti vaihtoehtoista käyttövoimaa ja merkittävin kasvu on tapahtunut täyssähköisten autojen määrässä, vaikka toistaiseksi erilaisia hybridimoottoria autoja on enemmän. Vaihtoehtoisia käyttövoimia hyödyntävien ajoneuvojen osuuden on enustettu nousevan nopeasti tulevina vuosikymmeninä. Sähköistä käyttövoimaa käyttävät ajoneuvot lisäävät ympärilleen uudenlaista liiketoimintamalleja, kuten latausinfrastruktuurin, huoltojen ja valmistuksen saralla. (Liikenteen palvelujen muutostrendit 2023.) Suomessa tieliikenteen päästöt muodostavat yli 90 % liikenteen päästöistä, tästä luvusta henkilöautot synnyttävät yli puolet. Vuoteen 2030 mennessä tavoitellaan liikenteen päästöjen puolittamista ja nollapäästöisen liikenteen tavoite on asetettu 2045. Suomessa nopeasti tapahtunut kehitys liikenteen sähköistymisessä ei tue pelkästään päästötavoitteita, vaan antaa mahdollisuuden olla edelläkävijä sähköisen liikenteen kehittämisessä. Kehityskohteina ovat muun muassa akku- ja latausteknologia, uusiutuvan energian tuotanto ja kiertotalouden hyödyntäminen. (Suomi sähköisen liikenteen edelläkävijä 2030 2023.)

Tämän opinnäytetyön toimeksiantaja on Lielahden autokeskus Oy. Opinnäytetyön tarkoitus on suunnitella ja toteuttaa sähköinen voimalinja käyttäen Scott Drive-ohjainta ja tehdä laitteeseen toimiva ohjaus CAN-väylää hyödyntäen laboratorio-olosuhteissa. Käytännössä työn kohteena oleva sähköinen voimalinja tullaan asentamaan myöhemmässä vaiheessa veneeseen, mutta tämän opinnäytetyön puitteissa varsinaista ajoneuvoasennusta ei toteutettu. Opinnäytetyön kirjallisen tuotoksen tavoitteena on toimia helposti ymmärrettävänä ohjekirjana Scott Drive ohjaimen käytöstä riippumatta siitä, mihin ajoneuvoon sähköistä voimalinjaa ollaan toteuttamassa. Opinnäytetyössä keskitytään Scott Drive-ohjaimen CAN-väylä ohjaukseen, tämän vuoksi opinnäytetyössä ei käsitellä teknisiä ajoneuvokohtaisia yksityiskohtia eikä käytännön asennusta ajoneuvoihin, vaan tarkoitus on tuottaa tietoa, miten ohjaimen CAN-väylää voidaan hyödyntää ajoneuvosovelluksissa. Tavoitteena on saada käytännönläheinen ja helposti lähestyttävä ohjeistus, jota pystyy hyödyntämään ilman syvällistä CAN-väylän periaatteiden tuntemista. Kirjallisessa

osassa ei käsitellä tekniikan perusteita, vaan se on tarkoitettu lähinnä tietopaketiaksi aiemmin ajoneuvotekniikkaan perehtyneille henkilöille. Työssä esiin tulevia lukumuunnoksia ei käsitellä tarkemmin, lukumuunnokset voidaan toteuttaa ohjelmoijalaskimella, jollainen löytyy Windows käyttöjärjestelmästä, lisäksi ilmaisia lukumuuntimia löytyy verkosta.

Työ toteutettiin rakentamalla voimalinja irrallisena käytössä olevaan sähkölaboratorioon. Ensimmäisessä vaiheessa rakennettiin ohjaimen, moottorin ja muiden laitteiden välille tarvittava 12 voltin sekä korkeajännite johdotus. Ohjaimella ja moottorille rakennettiin erillinen jäähdytysjärjestelmä. Sähköinen voimansiirto koekäytettiin laboratoriossa analogisella ohjauksella useita kertoja, jolloin varmistettiin sen toiminta. Toisessa vaiheessa rakennettiin CAN-väylä laitteiden välille. Tuleva ajoneuvo asennus ei sisällä A/D-muuntimia, tämän vuoksi CAN-ohjaus toteutettiin luomalla Codesys-ohjelmistolla kosketusnäytölle ohjauspaneeli, jolla ohjainta voidaan hallita CAN-väylää pitkin.

1.2 Käytetyt menetelmät ja prosessi

Opinnäytetyössä käytettiin tutkimuksellisen kehittämistyön menetelmää. Kuten Ojasalo, Moilanen & Ritalahti (2009, 22–23) kirjassaan kuvaa, voidaan kehittämistyö nähdä prosessina, jossa vaiheet seuraavat toisiaan. Yksinkertaistetusti prosessi koostuu suunnittelu-, toteutus- sekä arviointivaiheista. Kehittämistyön prosessia voidaan kuvata tarkemmin seuraavanlaisilla vaiheilla:

- Tunnistaminen ja tavoitteiden asettaminen
- Kehittämistyön kohteeseen tutustuminen
- Kehittämistyön tarkempi määrittely ja rajausten teko
- Tietoperustan hankkiminen ja menetelmän päättäminen
- Kehittämistyön toteuttaminen
- Työn ja tulosten arviointi

Kehittämistyön lähtökohtana voidaan pitää kehitettävän kohteen tunnistamista ja siihen liittyvien tavoitteiden asettamista. Kehitettävään kohteeseen tarkempaan tutustumiseen kuuluu olennaisena osana tiedon hankkiminen ja taustalla olevaan teoriaan sekä aiempiin tutkimuksiin perehtyminen. (Ojasalo, Moilanen & Ritalahti 2009, 22–23) Opinnäytetyön kehittämiskohteen, väyläohjauksen toteuttaminen Scott Drive ohjaimeen, päättämisen jälkeen aloitin tutustumaan aiheeseen perehtymällä itse ohjaimen käyttöohjekirjan lisäksi aiemmin aihepiiristä tehtyihin tutkimuksiin,

erityisesti CAN-väylä ohjauksesta tehtyihin opinnäytetöihin. Kuvatuista prosessin vaiheista ensimmäistä ja viimeistä lukuun ottamatta vaiheet tapahtuivat päällekkäin ja limittäin toisiinsa nähden. Aiheeseen tarkemmin perehtyessä alkoi tietoperustan luominen ja tarkennuksia tavoitteisiin ja rajauksiin muodostui työn suorituksen aikana.

Kirjallisuudessa kuvatuista (Ojasalo, Moilanen & Ritalahti 2009, 36) lähestymistavoista tämän opinnäytetyön tekemistä kuvaa parhaiten konstruktivinen tutkimus. Konstruktivisen tutkimuksen määrittely pitää sisällään tuottaa jokin ratkaisu käytännön ongelmaan. Lopputuloksena syntyy jokin uusi konkreettinen aikaansaannos. Konstruktivinen menetelmän kohde on aina jokin konkreettinen asia. Konstruktiviselle tutkimukselle kuvaavaa on myös jatkuva käytännön ja teorian välinen yhteys. (Ojasalo, Moilanen & Ritalahti 2009, 38) Työn kohde oli konkreettinen laite, johon käytännössä toteutettiin väylä ja väylän ohjelmointi. Sen lisäksi tuloksena on ohjekirja siihen, miten vastaavaa laitetta voidaan soveltaa muissa yhteyksissä.

Viimeinen vaihe kehittämistyön prosessissa on arviointi. Loppuarvioinnissa tarkastellaan miten työssä onnistumista. Arvioinnin pohjalla on tavoitteiden ja lopputulosten kuvaaminen ja tunnistaminen. Arviointia tapahtuu luonnollisesti prosessin kuluessa, jolloin sen tavoitteena on ohjata työtä, loppuarvioinnissa tarkastellaan koko kehittämisprosessia. Lopputuloksen arvioinnissa voidaan käyttää erilaisia määritelmiä, kuten esimerkiksi sovellettavuutta, yksinkertaisuutta tai toistettavuutta. (Ojasalo, Moilanen & Ritalahti 2009, 47) Lyhyesti arvioiden, työlle asetettu tavoite väyläohjauksen toteuttamisesta saavutettiin, myös kirjallinen osuus sisältää keskeisiä asioita, joita voidaan hyödyntää muissa toteutuksissa.

1.3 Lielahden autokeskus Oy

Lielahden Autokeskus Oy on perustettu vuonna 1987, perustajana toimi Mikko Leppälahti. Yrityksen päätoimiala on kuorma-autojen myynti sekä huoltotoiminta. Yritys harjoittaa myös raskaankaluston tuontia pääosin Euroopan alueelta (Truck.fi Yritys n.d.) Vuonna 2021 yritys laajensi toimintaansa aloittamalla sähkökuorma-autojen konversiotehtaan. Lielahden autokeskus järjesti oman rekrytointikoulutuksen, jonka tavoitteena oli kouluttaa sähkökuorma-auto asentajia yrityksen palvelukseen. Kuorma-autojen sähkökonversiot toteutetaan Keuruulla yrityksen toimipisteessä, jossa aikaisemmin toimi huolto- ja korjauspalvelut. Toukokuussa 2022 yrityksen toimesta valmistui ensimmäinen asiakkaalle tilattu konvertoitu sähkökuorma-auto. Tämä kyseinen ajoneuvo tuli KWH Logistics Oy:lle käyttöön Äänekoskelle biotuotetehtaan tehdasalueen sisäiseen liikennöintiin. (Truck.fi Sähköajoneuvot n.d.)

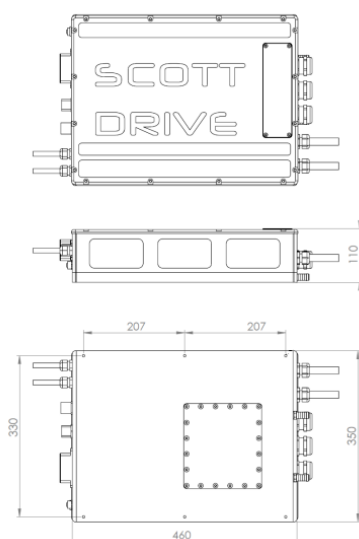
2 Opinnäytetyössä käytettävä laitteisto ja ohjelmistot

Tässä kappaleessa käsittelen tiivistetysti keskeisimmät työssä käytettävät laitteet ja niiden keskeisimmät tekniset ominaisuudet. Ohjelmistojen osalta käyn läpi tärkeimmät työn tekemisessä käytetyt ohjelmistot yleisellä tasolla. Käyttöjärjestämä tai siihen liittyvät ohjelmistot ovat rajattu opinnäytetyön ulkopuolelle.

2.1 Scott Drive SD 250

Scott Drive SD250-moottoriohjain on tarkoitettu 3-vaihe moottorisovelluksiin aina 300kVA tehoon asti. Sillä voidaan käyttää sekä kestopagneetti- että induktiomootoreiden ohjaukseen. Scott Driven ohjaimet ovat suunniteltu helposti käyttöönotettaviksi niin nykyisiin ajoneuvoihin kuin itsevalmistettuihin ajoneuvosovelluksiin. Scott Drive-ohjain ylittää 700 ampeerin ja 425 voltin jatkuviin tehollisarvoihin. (Scott Drive SD250 n.d.)

Scott Drive-ohjain sisältää kiinteän sisäänrakennetun esilatauspiirin, jolloin esilatausta ei tarvitse erikseen suunnitella. Komponentit ovat valikoitu ja suunniteltu siten, että ohjain sietää suuriakin lämpötilanvaihteluita eikä ole altis tärinälle. Ohjaimessa on sisäänrakennettuna lämpöanturit, jotka suojaavat IGBT:tä vaurioilta. Ohjain on nestejäähdytteinen ja siinä on lämpöanturit sekä jäähdytysnesteelle että moottorin lämpötilalle. Ohjaimessa on yhdeksän digitaalista sisääntuloa erilaisille katkaisijoille ja neljä 12 voltin ohjattavaa lähtöä. (Scott Drive n.d., 6.)



Kuvio 1 Scott Drive SD250 (Scott Drive n.d.)

2.3 Epec 6107

Epec 6107 on erityisesti liikkuviin työkoneisiin tarkoitettu 7 ” kosketusnäyttö. Näytössä on 32 bittinen prosessori ja 4–16 Gigatavua Flash muistia. Näyttö on suunniteltu yhteensopivaksi Codesys 3.5 ohjelmiston kanssa. Näyttö pohjautuu CANOpen-protokollaan, mutta näyttö tukee lisäksi J1939, ModBus sekä NMEA2000 protokollia. Näytössä on viisi I/O-liitäntää: kaksi digitaalista sisääntuloa, kaksi digitaalista lähtöä sekä analoginen sisääntulo. Näytössä on myös kaksi erillistä CAN-väylää. Näytön käyttöjännite on 12–24 voltia. (Epec 6107 Display 2019.)



Kuvio 4 Epec 6107 näyttö (Epec 6107 Display 2019.)

2.4 Orion BMS 2

Orion BMS 2 on akunhallintajärjestelmä, joka on mallista riippuen tarkoitettu 24–180 kennolle laitetta kohti. Laitteessa on mahdollisuus myös kahdelle etälaajennus moduulille, joiden avulla akun kokonaisjännitettä on mahdollista kasvattaa aina 800 volttiin saakka. Laite tukee useita eri latausprotokollia, mukaan luettuna pikalatauksen. Käyttöjännite voi olla 12–24 voltia. Laitteessa on kaksi eri CAN-väylää ja neljä käyttäjän ohjelmoitavissa olevaa digitaalista lähtöä. (Orion BMS 2 n.d.)



Kuvio 5 Orion BMS 2 (Orion BMS 2 n.d.)

2.5 OpenECU M580

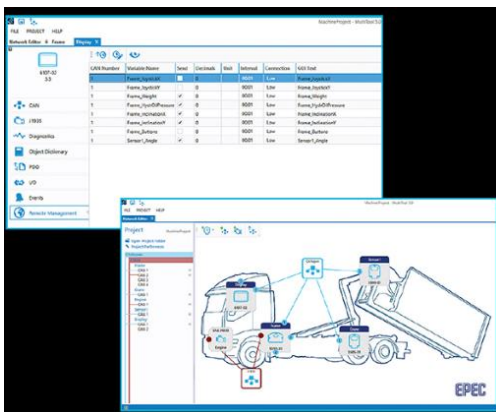
OpenECU M580 on tarkoitettu ajoneuvonhallintayksiköksi 24 voltin käyttöjännitettä käyttäviin sähkö- ja hybridi ajoneuvoihin. Se on suunniteltu erityisesti käytettäväksi hyötyajoneuvoissa sekä maatalouskoneissa. Yksikössä on sisäänrakennettuna latauksenhallinta järjestelmä, erilliselle latauksenhallintayksikölle ei ole tarvetta. Yksikössä on neljä CAN-väylää ja 112 ohjelmoitavaa I/O liitäntää, joten yksikköä voidaan käyttää laaja-alaisesti monissa erilaisissa sovelluksissa. Yksikön ohjelmointi toteutetaan MATLAB/Simulink ohjelmistolla. (Introducing m580 n.d.)



Kuvio 6 OpenEcu m580 (Introducing m580 n.d.)

2.6 Epec MultiTool Creator

Epec on suomalainen teknologia yritys, joka toimittaa ohjelmia ja järjestelmiä erityisesti hyötyajoneuvoihin ja koneisiin (Tuote- ja ohjelmistokehitys n.d.). MultiTool Creator on CAN-väylän määrittämiseen tehty ohjelmisto, joka mahdollistaa nopean ja virheettömän ohjelmoinnin. Ohjelmistolla voidaan määrittää I/O käyttöliittymä, CAN-väylän asetukset ja protokolla. Ohjelmisto luo valmiin mallikoodin käytettäväksi Codesys ohjelmistoympäristössä. (Software Development Environment n.d.)



Kuvio 7 Epec MultiTool Creator (Software Development Environment n.d.)

2.7 Codesys 3.5

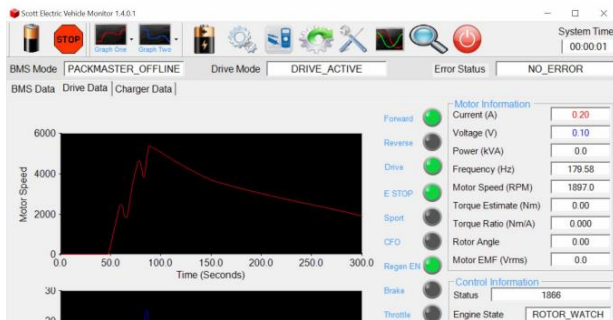
Codesys on automaatio-ohjelmisto, jolla voidaan tehdä ohjelmia kaikilla PLC ohjelmointikielillä. Ohjelmiston on kehittänyt alun perin 3S-Smart Software Solutions GmbH vuonna 1994, yrityksen nimi vaihtui CODESYS GmbH:ksi vuonna 2020. Codesys on valmistajasta riippumaton ohjelmisto, jota voidaan soveltaa useiden eri valmistajien PLC ohjelmoinnissa. Codesys on yksi laajimmin käytetyistä automaatio-ohjelmistoista. (The company n.d.)



Kuvio 8 Codesys logo (The company n.d.)

2.8 SEV-View

SEV-View, eli Scott Electric Vehicle Viewer on Scott Drive-yrityksen ohjaimiin tarkoitettu ohjelmisto. Laitevalmistaja toimittaa ohjelmiston pyynnöstä. Ohjelmisto on tarkoitettu ohjaimen asetusten sekä parametrien määrittämiseen. Lisäksi se sisältää diagnostiikan ja reaaliaikaisen datan seurannan. (Scott Drive n.d.,30.)



Kuvio 9 SEV-View päänäyttö (Scott Drive n.d.,30.)

2.9 PCAN View

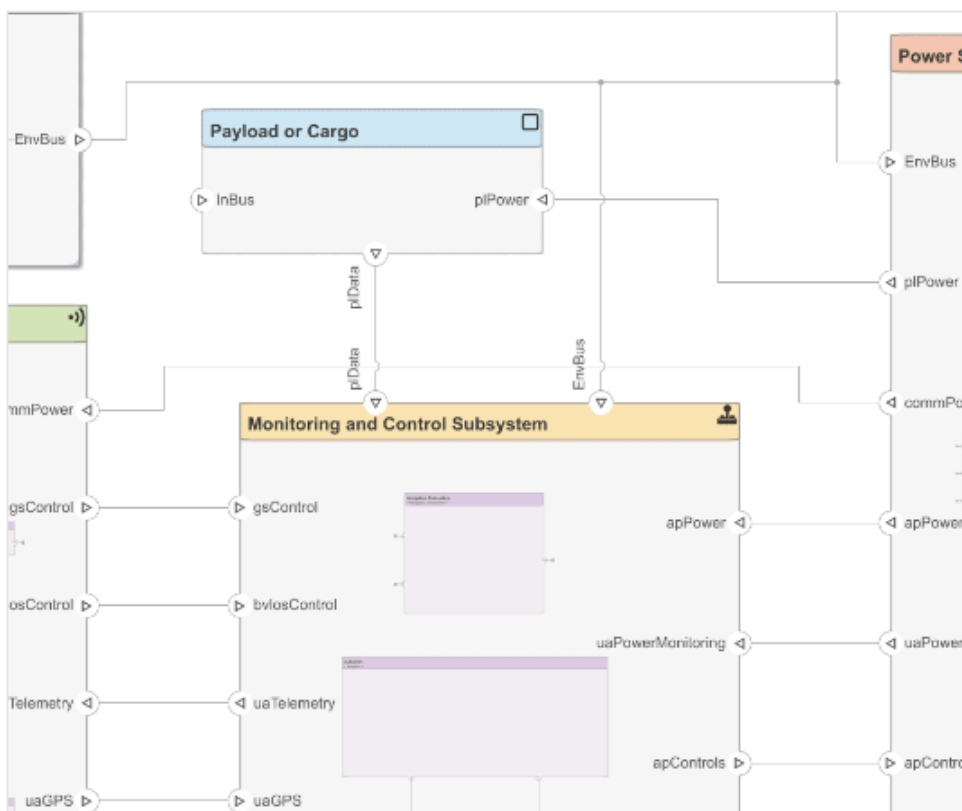
PCAN View on Windows käyttöjärjestelmälle tehty yksinkertainen CAN-väylä monitorointi ohjelma, jolla voi lukea, lähettää ja tallentaa CAN-väylän liikennettä. Ohjelma näyttää myös mahdolliset väylän virheilmoitukset. Ohjelma on tarkoitettu PCAN-lukijoiden ohjelmistoksi (PCAN View n.d.)



Kuvio 10 PCAN View (PCAN View n.d.)

2.10 Simulink

Simulink on MathWorks-yrityksen tekemä ohjelmisto. Se on blokkidiagrammi suunnittelu ympäristö, jolla voidaan suunnitella monialaisia malleja. Ohjelmistolla voidaan simuloida ohjelmistoa ennen sen siirtämistä laitteistoon ja sitä pystyy käyttämään ilman varsinaisen koodin kirjoittamista. Ohjelmistossa luotua mallia voidaan skaalata erilaisiin sovelluksiin ja se kykenee simuloimaan useampia malleja samalla kertaa samassa ympäristössä. (Simulink is for Model-Based Design n.d.)



Kuvio 11 Simulink (Simulink is for Model-Based Design n.d.)

3 CAN-väylä

Vuonna 1986 Robert Bosch GmbH esitteli Controller Area Network (CAN) -sarjaviestijärjestelmän. CAN on yksi kaikkien aikojen menestyksekkäimmistä verkkoprotokollista ja sitä käytetään nykyään käytännössä jokaisessa valmistetussa autossa. Autojen lisäksi CAN-väylää käytetään laajalti eri ajoneuvoissa, kuten junissa ja laivoissa. CAN-väylää käytetään myös teollisuuden ohjauksessa.

CAN väylän kehitys alkoi 1980-luvun alussa Boschin insinöörien toimesta. CAN-väylän kehitys aloitettiin vastaamaan autoteollisuuden vaatimuksia toimintojen lisäämisestä. Merkittävin tavoite kehittämiselle oli uusien toimintojen lisääminen ja tarvittavien johdotusten ja johdinsarjojen väheneminen oli kehityksen sivutuote. Boschin insinööreistä Kiencke aloitti uuden sarjaviestijärjestelmän kehittämisen vuonna 1983. Myös Mercedes-Benzin insinöörit osallistuivat järjestelmän määrittelyvaiheeseen. Järjestelmä esiteltiin vuonna 1986. CAN-protokolla standardisoitiin 1990-luvun alussa ISO 11898 -standardissa. (History of CAN technology n.d.)

3.1 CAN-väylän toiminta

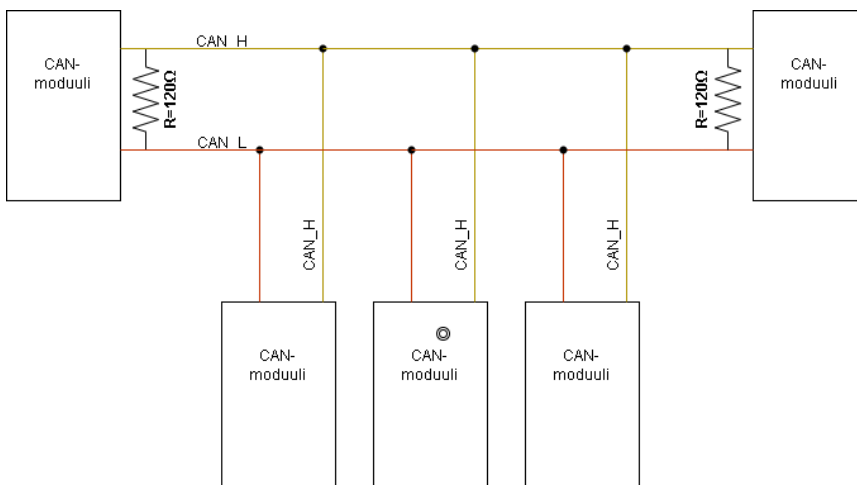
CAN-väylässä on useita kerroksia, jotka mahdollistavat tehokkaan viestinnän laitteiden välillä. Tyypillisesti ne luokitellaan OSI-mallin mukaisesti (kuvio 12), johon kuuluu seitsemän kerrosta. (Reif & Dietche 2014, 1252.) OSI-mallin merkitys väylässä perustuu sen kykyyn toimia yleisenä viitekehyksenä. Malli jakaa verkkokommunikaation diskreetteihin kerroksiin, mikä helpottaa väyläprotokollien ja -järjestelmien suunnittelua, kehittämistä, vianmäärittystä ja ylläpitoa. Se tarjoaa standardoidun tavan keskustella ja ymmärtää väyläkommunikaatiota. Tämä standardointi yksinkertaistaa kommunikaatiota eri valmistajien ja teknologioiden välillä. Koska kerroksille on määritetty selkeät rajat ja vastuut, malli edistää yhteensopivuutta. Itsenäisesti kehitetyt laitteet ja ohjelmistot voivat viestiä tehokkaasti, jos ne noudattavat mallin määrittämiä. Malli helpottaa vianetsintää, koska mallin pohjalta voidaan määrittää missä kerroksessa ongelma sijaitsee. (The OSI Model: Breaking Down Its Seven Layers n.d.) Seuraavissa kappaleissa kuvaan kerroksia ja niiden toimintaa pääpiirteittäin. Toteutetun työn kannalta keskeisimmät kerrokset ovat fyysinen kerros sekä sovelluskerros, näiden osalta kuvaus on tarkempaa.



Kuvio 12 OSI-mallin kerrokset (Muokattu Reif & Dietche 2014, 1254.)

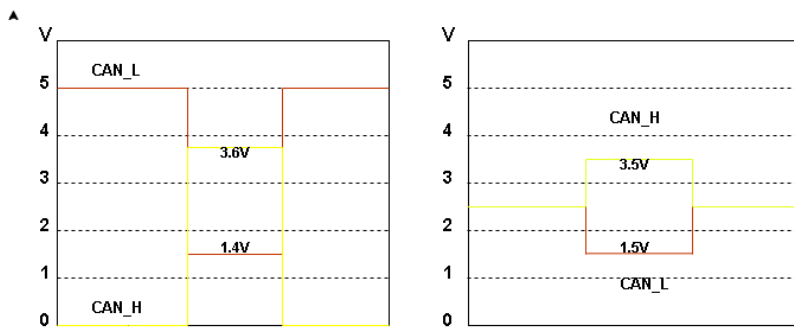
3.1.1 Fyysinen Kerros

CAN-väylässä kaapeli kulkee jokaisen laitteen kautta ja väylän molemmat päät päätetään pääte-
vastuksella (Alanen 2000, 6). Väyläkaapeli muodostuu kahdesta johtimesta, CAN_H ja CAN_L. Pää-
tevastus on välttämätön väylän molemmissa päissä, kulkevien viestien muutokset aiheuttavat hei-
jastuksia, jotka johtavat virheellisiin viesteihin. Väylä on kaksisuuntainen, joten väylän molemmat
päätt täytyy varustaa päätevastuksella. Jokaista moduulia ei tarvitse siis varustaa vastuksella. Peri-
aate on esitetty kuviossa 13.(CAN-bus topology and termination, 2023.) Väylässä vastuksena käy-
tetään 120 Ω vastusta (The CAN Bus Protocol tutorial).

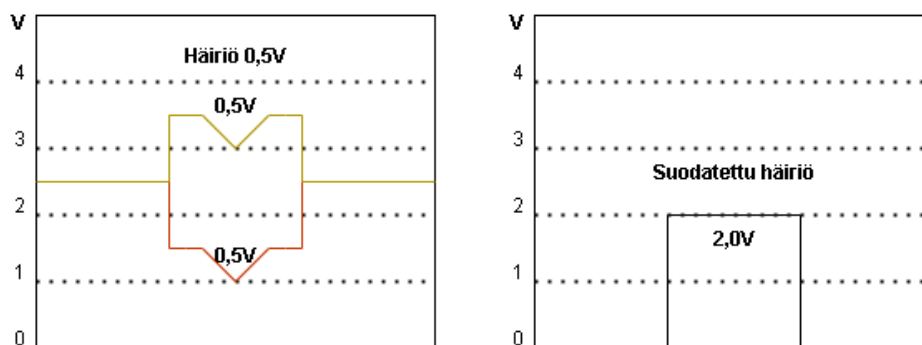


Kuvio 13 CAN-väylä periaatekuva (Muokattu CAN-bus topology and termination, 2023.)

Väyläkaapelina yleisesti käytetään kierrettyä parikaapelia, vaikka olosuhteista riippuen on mahdollista käyttää myös kiertämätöntä. Parikaapelin käytöllä saavutetaan symmetrinen tiedonsiirto, jossa viestin bitit lähetetään molempia johtimia pitkin eri jännitteellä. Tämä vähentää väylän alttiutta häiriöille, koska häiriöt vaikuttavat molempiin johtimiin ja häiriöt voidaan tämän erotuksen avulla suodattaa pois. Väyläjohtimissa käytetty jännite riippuu väylän nopeudesta. CAN-väylän tiedonsiirtonopeus on tyypillisesti välillä 125 kbit/s – 1 Mbit/s. 125 kbit/s ja sen alle on hidas väylä ja tästä 1Mbit/s asti lasketaan nopeaksi väyläksi. Viestinsiirrossa väylässä käytetään kahta tilaa: dominanttia, jolloin bitti on nolla sekä resessiivistä, jolloin bitti on yksi. Hitaissa väylissä CAN_H nimellijännite on 0 voltia resessiivisessä tilassa ja CAN_L nimellijännite on 5 voltia. Dominantissa tilassa CAN_L jännite laskee 1,4 volttiin ja CAN_H jännite nousee 3,6 volttiin. Nopeassa väylässä nimellijännite on 2,5 voltia. Resessiivisessä tilassa sekä CAN_H että CAN_L jännite on 2,5 voltia. Dominantissa tilassa CAN_H jännite on 3,5 voltia ja CAN_L 1,5 voltia. Jännitteet ovat esitettynä kuviossa 14 (Reif & Dietche 2014, 1260–1261.) Edellä kerrottu väylän häiriön suodatus toteutetaan käytännössä kuvion 15 esittämällä tavalla.



Kuvio 14 CAN-väylä jännitteet (Muokattu Reif & Dietche 2014, 1261.)



Kuvio 15 CAN-väylän häiriönsuodatus (Muokattu Reif & Dietche 2014, 1261.)

3.1.2 Sidos-, verkko- ja kuljetuskerrokset

Sidoskerros (Data-link layer) vastaa tiedon siirtämisestä moduuleiden välillä. Tieto järjestellään sidoskerroksessa kehyksiksi. (Reif & Dietche 2014, 1254. Se koodaa, dekodaa sekä järjestää yksittäiset bitit ja valmistelee ne datapaketeiksi. Tämä tarkoittaa, että se luo datapaketteja ja muuttaa suuremmat datapaketit pienemmiksi yksiköiksi. Data siirretään sitten sidoskerroksesta verkkokerrokseen, joka on sen yläpuolella oleva kerros. Verkkokerroksessa datapaketille annetaan osoite ja se lähetetään eteenpäin. Sidoskerros varmistaa, että kehykset siirretään virheettää ja tekee näin ollen suojaamattomista järjestelmäyhteyksistä turvallisia. (What is the data link layer in the OSI model? 2023.)

3.1.3 Sovelluskerros

Sovelluskerros, joka on OSI-mallin seitsemäs ja korkein kerros, toimii rajapintana väylän ja sovellusten välillä. Tässä kerroksessa määritellään protokollat, jotka ohjaavat kommunikointia sovellusten ja palveluiden välillä. Eri protokollat määräävät, miten data muotoillaan, siirretään ja tulkitaan. Sovelluskerroksessa sijaitsee käyttöliittymät, joilla vuorovaikutus väylän kanssa tapahtuu. Sovelluskerros vastaa datan vaihdosta väylässä olevien eri laitteiden sovellusten välillä. Usein myös eri turvamekanismit, kuten datan salaaminen, toteutetaan sovelluskerroksessa. (The OSI Model: Breaking Down Its Seven Layers n.d.)

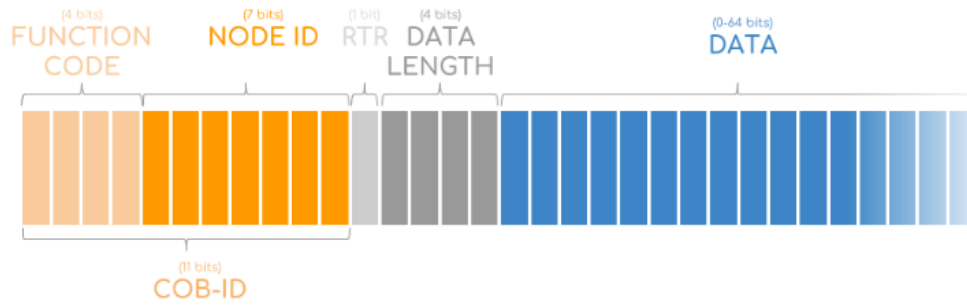
3.2 CAN-Protokolla

CAN-protokolla määrittää miten väylässä olevat laitteet vaihtavat tietoa toistensa kanssa. Protokolla koostuu joukosta sääntöjä viestien lähettämiseen ja vastaanottamiseen väylässä. Se määrittelee, miten data siirretään laitteiden välillä. Vaikka CAN-väylä kehitettiin alun perin autoalan tarpeisiin, sen arkkitehtuuri ja edut ovat johtaneet siihen, että useat muut alat ovat ottaneet protokollan käyttöön. Jokainen CAN-protokollaa käyttävä elektroninen laite (tai solmu) on yhdistetty toisiinsa väylän kautta, mikä mahdollistaa viestien siirron. CAN-protokolla ei välttämättä noudata isäntä-orja-arkkitehtuuria, mikä tarkoittaa, että jokaisella solmulla on oikeus lukea ja kirjoittaa viestejä väylään. Kun solmu on valmis lähettämään dataa, se tarkistaa väylän saatavuuden ja lähettää CAN-kehysen väylään. Kehys on rakenne, joka kuljettaa bittien tai tavujen muodostaman viestin väylän sisällä. On olemassa kahdenlaisia protokollia: osoite- tai viestipohjaisia. Osoitepohjaisessa protokollassa datapaketeissa on laitteen osoite, jolle viesti on tarkoitettu.

Viestipohjaisessa protokollassa jokainen viesti tunnistetaan ennalta määrätyn tunnuksen avulla eikä osoitteen perusteella. CAN-lähetetty kehys on tyypillisesti viestipohjainen protokolla. Viesti on tietopaketti, joka kantaa informaatiota. CAN-viesti koostuu 11 tavusta dataa, jotka on järjestetty tiettyyn rakenteeseen, jota kutsutaan kehykseksi. Kussakin tavussa kuljetettu tieto on määritetty CAN-protokollassa. Kaikki CAN-protokollaa käyttävät solmut saavat kehyksen ja solmun ID:n perusteella tehdä päätös, hyväksytäänkö viesti vai ei. Jos useat solmut lähettävät viestin samanaikaisesti, korkeimman prioriteetin omaava solmu saa väyläyhteyden käyttöönsä. Alempiin prioriteettisolmuihin on odotettava, kunnes väylä on vapaa. (Parikh n.d.) Yleisimmin käytetyt CAN-protokollat ovat CANOpen, OBD2 sekä J1939. OBD2-protokolla on suunniteltu pääasiassa vikadiagnostiikkaan ja reaaliaikaisen tiedon seurantaan. J1939-protokolla on suunnattu raskaan kaluston väyläprotokollaksi, kuten kuorma-autojen ja erilaisten työkoneiden. CANOpen-protokollaa käytetään ajoneuvojen lisäksi myös teollisuusautomaatiossa ja lääketieteellisissä sovelluksissa. (CANOpen tutorial – Simple intro 2022.)

3.3 CAN-viestikehys

Aiemmissa kappaleissa on viitattu usein viestikehykseen. Eri CAN-protokollien viestikehykset poikkeavat toisistaan rakenteeltaan ja sisällöltään. Tässä luvussa keskityn käsittelemään CANOpen-viestikehyksen rakennetta yksityiskohtaisemmin ja rajaan muiden protokollien käsittelyn aiheen ulkopuolelle. CANOpen on yksi yleisimmin käytettyjä protokollia. Scott Drive käyttää CAN-viesteissä protokollaa, joka muistuttaa CANOpen-protokollan viestien rakennetta, mutta Scott Drive-ohjaimen viestit eivät kuitenkaan ole CANOpen-protokollan mukaisia. Yhtymäkohtia kuitenkin löytyy, joten käsittelen CANOpen-viestikehyksen lisäksi ohjaimen viestejä soveltuvin osin. CANOpen-viestien hahmottamiseksi niitä on syytä tarkastella CANOpen-protokollan viestikehyksen näkökulmasta. Viestien rakennetta käsitellään useissa lähteissä, kuten Siemensin julkaisemassa CANOpen tutorial (2019) oppaassa sekä kattavasti Alanen & Scholliers (2004, 81–137.) suomeksi koostamana VTT:n sovellusohjeessa. Kuviossa 16 on havainnollistettu CANOpen-viestikehyksen rakenne. COB-ID on heksadesimaali muodossa, binäärimuotoon muutettaessa on iso virheen mahdollisuus, mikäli ID:n tulkitsee desimaalina.



Kuvio 16 CANopen-viestikehys (CANopen tutorial – Simple intro, 2022.)

Kuviossa 17 nähdään 11 bittinen COB-ID, eli Communication Object Identifier. Scott Drive-käyttöohjekirja määrittää valmiiksi käytettävät viestit ja niiden muodon, joten myös käyttäjä, jolla ei ole tuntemusta CAN-protokollan käytöstä, pystyy luoda laitteeseen CAN-viestit. Kuviossa 6 esitellään Scott Drive käyttöohjekirjan mukainen viestikehys perusmuodossaan. Esitelty viesti on akunhallintajärjestelmän lähettämä viesti, jonka Scott Drive vastaanottaa. Viestien määrittely toteutetaan niissä laitteissa, joilta Scott Drive ottaa viestejä vastaan.

ID	Length	Byte 0	Byte 1	Byte 2	Byte 3	Byte4-7
0x350	8	Pack CCL (MSB)	Pack CCL (LSB)	Pack DCL (MSB)	Pack DCL (LSB)	Blank

Kuvio 17 CAN-viestin muoto Orion BMS (Scott Drive n.d.)

Kuvion 18 esimerkissä COB-ID on 0x350, joka on binäärimuotoon muutettuna 011 0101 0000. Viestikehysten tunnisteessa neljä ensimmäistä bittiä määrittelevät viestin funktion. CANopen-protokollan osalta funktiot ovat esiteltynä kuviossa 7. Kuvion pohjalta voidaan todeta funktion 0110 olevan TPDO 2, eli Transmit Process Data Object.

Object	Index in OD	Identifier (COB-ID)		COB-ID (hex)
		Function code	Node ID	
NMT function	-	0000	0000000	0 _h
Synchronization	1005 _h -1007 _h	0001	0000000	80 _h
Emergency	1014 _h , 1015 _h	0001	xxxxxxx	80 _h + node ID (81 _h – FF _h)
T_PDO1	1800 _h	0011	xxxxxxx	180 _h + node ID (181 _h – 1FF _h)
R_PDO1	1400 _h	0100	xxxxxxx	200 _h + node ID (201 _h – 27F _h)
T_PDO2	1801 _h	0101	xxxxxxx	280 _h + node ID (281 _h – 2FF _h)
T_PDO2	1401 _h	0110	xxxxxxx	300 _h + node ID (301 _h – 37F _h)
T_PDO3	1802 _h	0111	xxxxxxx	380 _h + node ID (381 _h – 3FF _h)
R_PDO3	1402 _h	1000	xxxxxxx	400 _h + node ID (401 _h – 47F _h)
T_PDO4	1803 _h	1001	xxxxxxx	480 _h + node ID (481 _h – 4FF _h)
T_PDO4	1403 _h	1010	xxxxxxx	500 _h + node ID (501 _h – 57F _h)
T_SDO	-	1011	xxxxxxx	580 _h + node ID (581 _h – 5FF _h)
R_SDO	-	1100	xxxxxxx	600 _h + node ID (601 _h – 67F _h)
NMT Error Control		1110	xxxxxxx	700 _h + node ID (701 _h – 77F _h)

Kuvio 18 COB-ID selite (CANopen tutorial 2019, 21.)

Lähetettävä viesti määritellään akunhallintajärjestelmän ohjelmiston kautta. Kuviossa 19 on Orion BMS 2-hallintaohjelmistolla tehtynä aikaisemmin esitelty viesti. Käyttäjä voi luoda itse tarvitsemansa viestit, ohjelmasta löytyy valmiiksi listattuna viestien datakenttien sisällöt.

Kuvio 19 Orion BMS2 CAN-viestien määrittely

3.3.1 Scott Drive CAN-viestit

Tässä kappaleessa käyn läpi Scott Drive-ohjaimen CAN-viestit. Viestejä tullaan käsittelemään tarkemmin kappaleessa 5.3. Tässä kappaleessa havainnollistan viestien rakenteen ja niiden eron CANopen-protokollaan. Kuviossa 9 on esitettyä CAN-viesti, jolla voidaan pyytää ohjainta lähettämään haluttuja parametrejä. Aikaisemman taulukon (kuvio 20) pohjalta voidaan päätellä viestin olevan funktioltaan TDPO3. Vielä tämän perusteella tarkasteltuna yhtäläisyys CANopen-protokollaan on havaittavissa.

Frame Identifier	Byte 0	Byte 1
0x399	Parameter Page [1..15]	Controller Node Address

Kuvio 20 Parametripyyntö (Scott Drive n.d., 19.)

Kun Scott Drive vastaanottaa oikeanlaisen CAN-viestin, se vastaa COB-ID:llä 0x399. Kuviossa 21 on esitettyä ohjaimen antama vastaus. Ensimmäinen tavu (Byte 0) on vastauksessa lähettävän, eli Scott Driven, Node-ID. Toinen tavu (Byte 1, joka kuviossa virheellisesti nimetty Byte2) sisältää pyydetyin parametrivun numeron (Scott Drive n.d., 20). Jäljelle jäävät kuusi tavua sisältävät parametrit, jokainen parametri on pituudeltaan kaksi tavua, muuttujatyypinä on WORD. Ensimmäisen parametrivun tilatiedot ovat esitettyä liitteessä 1.

Data Type	Byte 2	Bytes 3:4	Bytes 5:6	Bytes 7:8
Drive Status	1	Program Mode	Engine State	Switch State
Error Status	2	Error Code	Error Value	Mode Timer
Motor Data	3	Motor RPM	Motor Current (×10)	Motor Voltage (×10)
Analog Data	4	Throttle Check (×100)	Throttle Level (×100)	Brake Pressure (×100)
HV Bus Data	5	Frequency (×10)	Bus Current (×10)	Bus Voltage (×10)
Temp Data	6	Internal Temp (×100)	Motor Temp 1 (×100)	Motor Temp 2 (×100)
IGBT Temp	7	Leg A Temp (×100)	Leg B Temp (×100)	Leg C Temp (×100)
Phase Current	8	Phase A Current (×10)	Phase B Current (×10)	Phase C Current (×10)
Phase Voltage	9	Phase A Voltage (×10)	Phase B Voltage (×10)	Phase C Voltage (×10)
Aux Current	10	Aux 1 Current	Aux 2 Current	Aux 3 Current
FOC Id	11	Id Reference (×10)	Id (×10)	Vd (×10)
FOC Iq	12	Iq Reference (×10)	Iq (×10)	Vq (×10)
FOC Vs	13	Vs (×10)	Phase Speed	0
Angle Info	14	Slip (ACIM Only)	FW Angle	Phase Angle
PWM Data	15	Phase A PWM	Phase B PWM	Phase C PWM

Kuvio 21 CAN-väylän kautta saatavat parametrit (Scott Drive n.d., 20.)

Tässä tulee esiin merkittävä ero Scott Drive-ohjaimen käyttämän viestiprotokollan ja CANopenin välillä. CANopen-protokollassa COB-ID:t tulee olla yksilöllisiä, jotta voidaan välttyä kommunikaatio-ongelmilta väylässä. Viestien yksilöllisyys on perusta väylän tehokkaalle ja virheettömälle kommunikaatiolle. (CANopen Guide (2023): Simplified Intro to Network Protocols 2022) Scott Drive käyttää samaa COB-ID:tä sekä vastaanotettavissa että lähetettävissä viesteissä. Tämä aiheuttaa haasteita muiden laitteiden kanssa, jotka toimivat CANopen-protokollan mukaisesti.

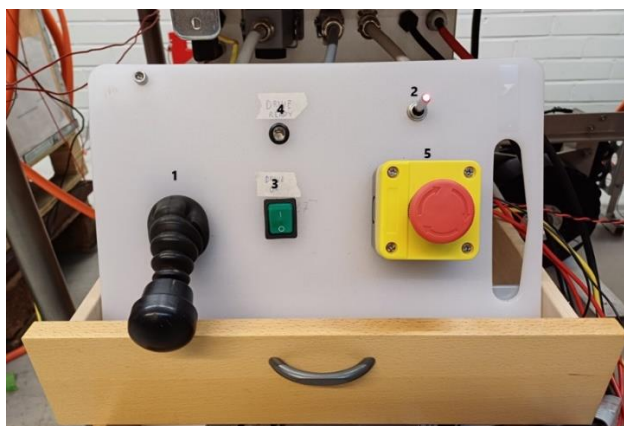
4 Työn toteutus

Työ toteutettiin TKI-keskuksen sähkölaboratoriossa. Työn ensisijainen tavoite oli selvittää, miten Scott Drive-ohjainta voidaan käyttää CAN-väylä ohjauksella. Tämän lisäksi työssä selvitettiin, miten saadaan ohjaimen lähettämät tilatiedot näytölle luettaviksi arvoiksi käyttäjän nähtäville. Ohjaimessa on CAN-väylä tuki ja ohjaus on mahdollista toteuttaa kosketusnäytön kautta ohjelmallisesti. Ohjaimen tilatietojen tuomiseen näytölle joutui käyttämään kiertotietä, koska ohjain ei noudata CANopen-protokollaa.

4.1 Laitteiston rakentaminen

Työn toteutus aloitettiin siirtämällä Scott Drive ja Emrax sähkölaboratorioon. Kytkennot rakennettiin lopullista asennusta vastaaviksi. Tähän poikkeuksena on akusto, joka on ajoneuvon osalta vielä rakennusvaiheessa. Työssä käytettiin sähkölaboratorion akustoa, jossa jännite on 355 voltia. Tämä vastaa jännitteeltään ajoneuvoon tulevaa akustoa. Laitteisto rakennettiin ja koekäytettiin analogisesti, jotta voidaan varmistua oikeasta toiminnasta. Laitteistoon valmistettiin oma käyttöpaneeli. Käyttöpaneeli on esiteltynä kuviossa 22. Käyttöpaneelin toiminnot ovat seuraavat:

- 1) Kaasukahva
- 2) Virtakytkin
- 3) Drive-käyttökytkin
- 4) Drive ready merkkivalo
- 5) Hätkatkaisin



Kuvio 22 Laitteiston käyttöpaneeli

Scott Drive-ohjaimen käyttöjännite on 12 voltia ja ohjaimen käyttöjännite otettiin HV-akustolta muuntajan avulla. Laboratorion muu laitteisto toimii 24 voltin jännitteellä, joka tuotetaan 24 voltin laboratoriovirtalähteellä, eikä tätä voitu hyödyntää ohjaimessa. Kuvioissa 23 sekä 24 on kuvattuna ohjaimeen tulevat liitännät. Kytkentäkaavio 12/24 voltin järjestelmälle on liitteessä 3 ja korkeajännite kytkennöille liitteessä 4.



Kuvio 23 Scott Drive-liitännät



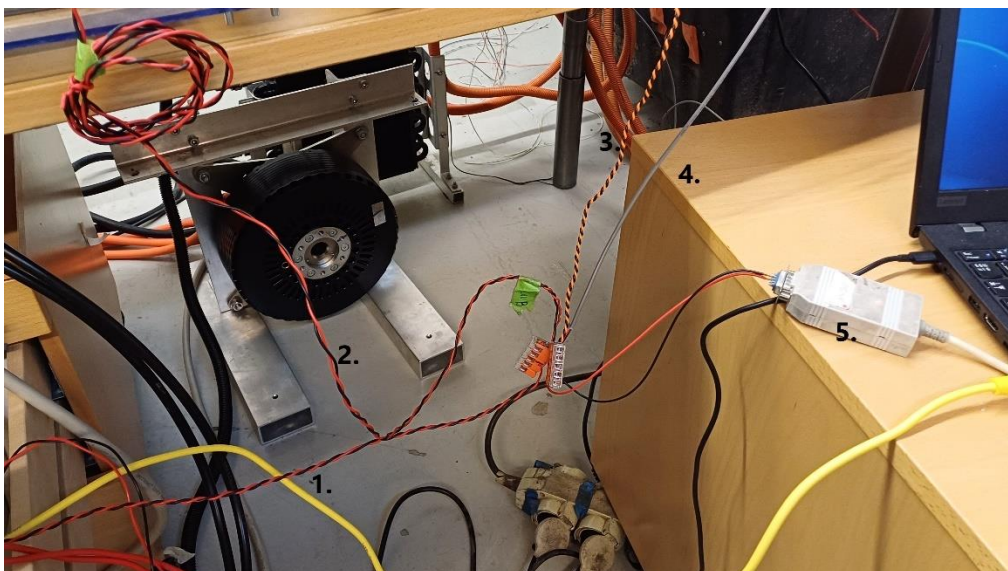
Kuvio 24 Scott Drive HV-liitännät

4.2 Väylän rakentaminen

Scott Drive-ohjaimen väylä toteutetaan 5-napaisella M12 liittimellä. Valmista kaapelia ei ollut saatavilla, tätä varten hankittiin erillinen liitin, johon tehtiin johdotus itse. Kuvissa 25 ja 26 on esitettyä CAN-kytkentä ohjaimeen sekä fyysisesti väylän rakenne. Kaavakuvio väylän rakenteesta on liitteessä 5.

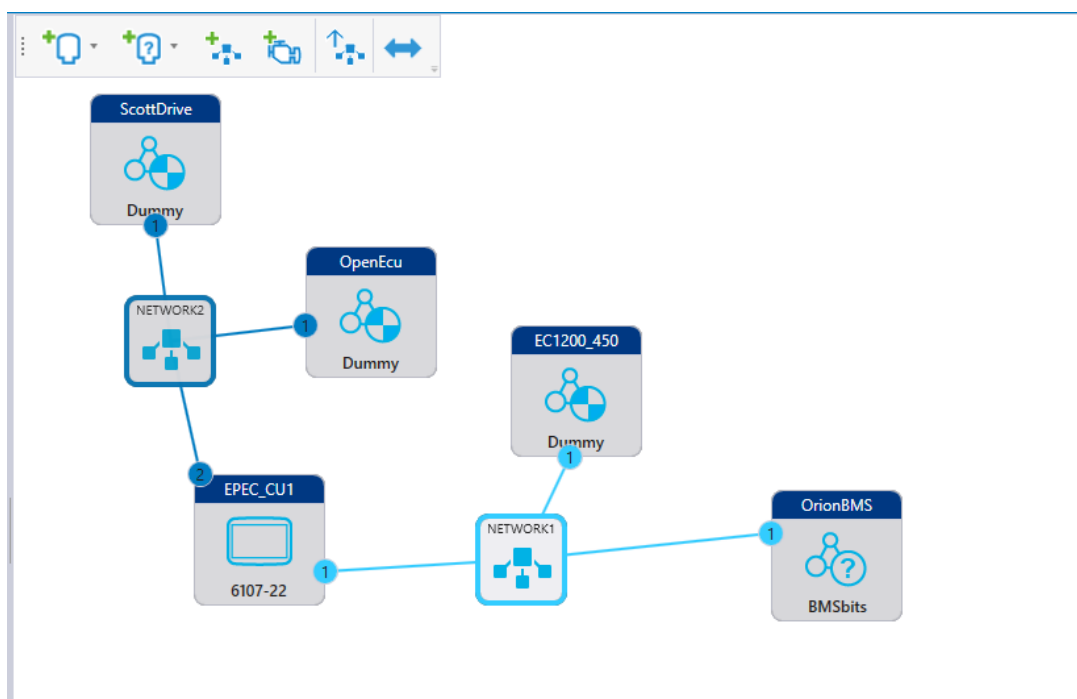


Kuvio 25 CAN-liitin ohjaimessa



Kuvio 26 CAN-väylä fyysisesti. 1. Ohjaimen väylä 2. Näytön väylä 3. VCU väylä 4. BMS väylä 5. PCAN-lukija

Ohjelmistossa väylä rakennettiin Epec Multitool Creator-ohjelmistolla. Kuviossa 27 on kuvattuna Multitool Creator näytöstä väyläverkon rakenne. Koska Scott Drive-ohjain käyttää väylänopeutta 500 kbit/s, toteutettiin väylä hyödyntämällä laitteiden muita väyläportteja. Kuvioista 22 nähdään, että toteutus on rakennettu näytön ympärille, network 1 on laboratorion käytössä ja network 2 on tehty ohjainta varten. Huomioitavaa on, ettei tässä verkon esityksessä BMS ole kytkettynä network 2, vaikka BMS CAN 2 on fyysisesti samassa väylässä. Tämä johtuu siitä, että network 2 osalta näyttö ei tarvitse viestejä suoraan BMS:ltä, vaan tarvittavat viestit tulevat näyttöön network 1 kautta. Johtuen ohjaimen CAN-protokollan eroista verrattuna CANopen-protokollaan, liitettiin väylään myös VCU käsittelemään parametrintiviestejä.



Kuvio 27 Multitool Creator CAN-verkko

Näytön CAN 2 väylään tehdyt viestit ovat nähtävissä kuviossa 28. Viestit 0x399 sekä 0x201 ovat tarkoitettu parametrien keräämiseen ja 0x39B ohjausviestille. 0x39B viesti näkyy harmaana, koska sen parametrit ovat asetettu ohjelmistossa Scott Drive-ohjaimelle. Lisäksi 0x201 viesti näkyy harmaana vastaavasta syystä, viestin sisältö parametroitu OpenEculle. Kuviossa 29 on esitettyä Multitool ohjelmistolla luodut CAN-viestit ohjaimelle.

Network Editor EPEC_CU1

6107 3.5

CAN 1 CAN 2

Variable Size

Variable	Size	
ParamData1	2	Remove
ParamData2	2	Remove
ParamData3	2	Remove
OpenEcu_ParamData1	2	Remove
OpenEcu_ParamData2	2	Remove
OpenEcu_ParamData3	2	Remove
ScottDrive_ThrottleLevel	2	Remove
ScottDrive_BrakePress	2	Remove
ParamPage	1	Remove
SDNodeAdr	1	Remove
PageNbr	1	Remove
OpenEcu_PageNbr	1	Remove
OpenEcu_EPEC_CU1_SDNodeAdr	1	Remove
OpenEcu_SDNodeAdr	1	Remove
ScottDrive_NodeAddr	1	Remove
ScottDrive_MsgType	1	Remove
ScottDrive_DirectionCtrl	1	Remove

Transmit PDOs

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
399	ParamPage	SDNodeAdr						
39B	ScottDrive_No	ScottDrive_Ms	ScottDrive_ThrottleLevel		ScottDrive_BrakePress		ScottDrive_Dir	

Receive PDOs

Is Receiv	Transmitter	COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<input checked="" type="checkbox"/>	OpenEcu	201	SDNodeAdr	PageNbr	ParamData1		ParamData2		ParamData3	

Kuvio 28 Multitool PDO viestit näytölle

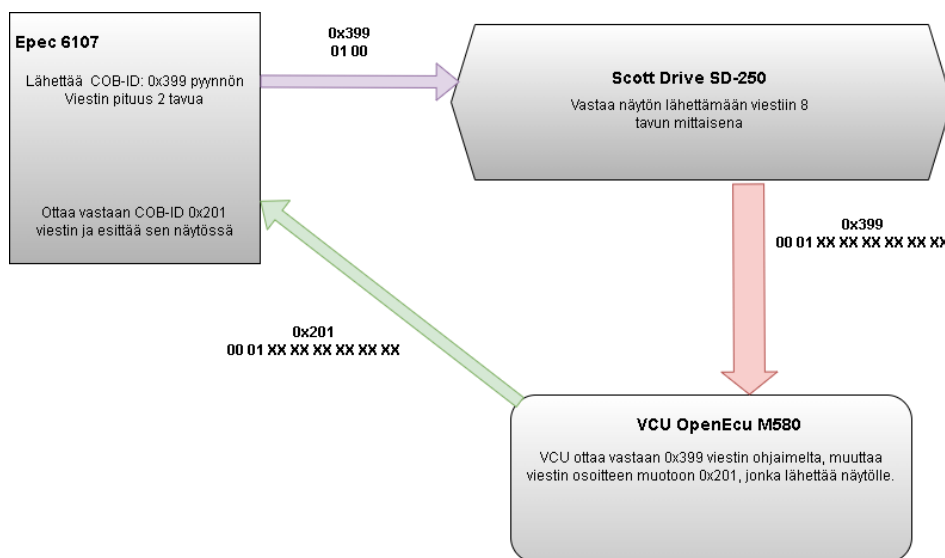
Receive PDOs

Is Receiv	Transmitter	COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<input checked="" type="checkbox"/>	Not Configured	350	PackCCL	PackDcl						
<input checked="" type="checkbox"/>	EPEC_CU1, CAN2	39B	NodeAddr	MsgType	ThrottleLevel		BrakePress		DirectionCtrl	
<input type="checkbox"/>	EPEC_CU1	399	ParamPage	SDNodeAdr						
<input type="checkbox"/>	OpenEcu	201	SDNodeAdr	PageNbr	ParamData1		ParamData2		ParamData3	

Kuvio 29 Scott Drive CAN-viestit Multitool Creatorissa

4.3 Parametrien vastaanotto näyttöön

Kuten aikaisemmin kappaleessa 3.1.1 mainittu, Scott Drive-ohjaimen käyttämä CAN-viestimuoto aiheuttaa haasteita käyttäessä CANopen-laitteistoa. Näytön tulisi lähettää COB-ID 0x399 viestinä tavut 0 sekä 1 ja ottaa vastaan tavut 2–7. Käytännössä tämä ei onnistu, Multitool noudattaa CANopen protokollaa, eikä lähtevä ja vastaanotettava viesti voi olla samalla COB-ID:llä. Mikäli viesti asetetaan näyttöön TPDO:ksi, se lähettää jatkuvasti viestin kahdeksan tavun mittaisena, ohjain ei kykene vastaamaan kahta tavua pidempään viestiin. Jos viesti asetetaan vaatimuksen mukaan kahden tavun mittaiseksi, ohjelma ei tällöin luo vastaanotettaville viesteille muistipaikkaa. Mikäli viesti taas asetetaan RPDO:ksi, näyttö ei pysty lähettämään väylään viestiä kyseisellä COB-ID:llä. Tämän vuoksi väylään liitettiin lisäksi VCU. Kuviossa 30 on hahmotelma viestirakenteesta, jolla ongelma saadaan kierrettyä. Toteutuksessa näyttö lähettää 0x399 osoitteella kaksi tavua ohjaimelle. Ohjain vastaa 0x399 osoitteella kahdeksan tavun mittaisen viestin, jonka VCU ottaa vastaan, muuttaa viestin COB-ID:n 0x201, joka lähtee näytölle. Näyttöön on määritelty COB-ID 0x201 vastaanotettavaksi ja näyttö pystyy esittämään halutut parametrit. Parametrit sisältöineen ovat esiteltynä kappaleessa 3.3.1



Kuvio 30 Hahmotelma viestirakenteesta

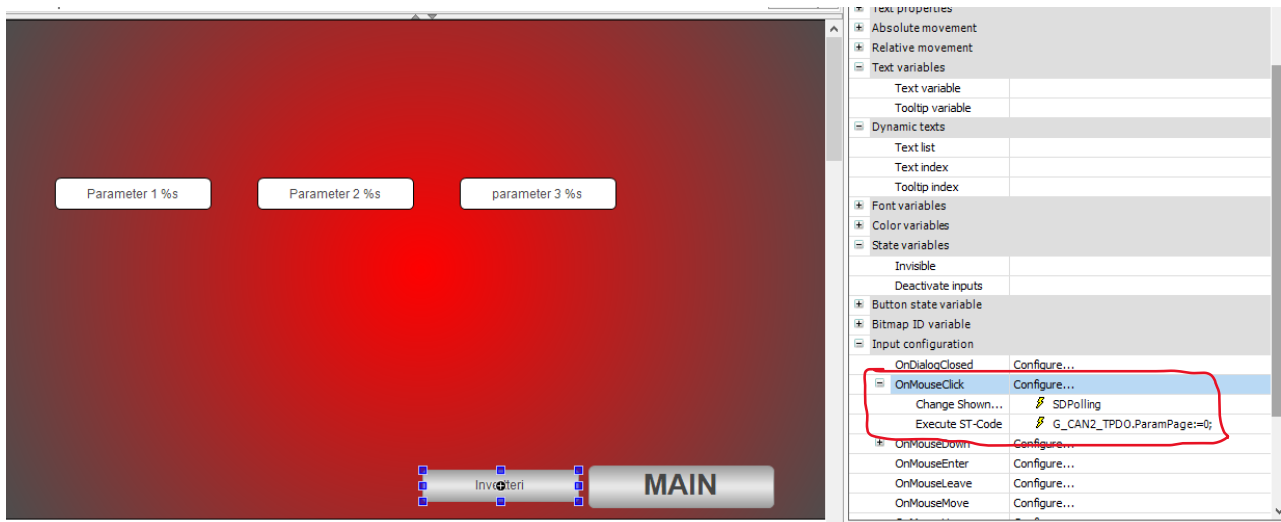
Koska kaikki parametrit ovat saman COB-ID:n takana, niiden saaminen yhteen näytönäkymään olisi haastavaa viestin päällekkäisyyksien vuoksi. Jokaiselle parametrisivulle luotiin oma visualisatio Codesys-ohjelmistolla, tällä välttyttiin virheiltä, joita päällekkäisyys voi aiheuttaa.

Valintanäyttö toteutettiin tekemällä jokaiselle parametrisivulle oma visualisaationsa. Parametrien pyyntö tapahtuu viestillä COB-ID 0x399, jossa ensimmäinen tavu on pyydetyn parametrin sivunumero ja toinen tavu on Scott Drive-ohjaimen solmun osoite. Kuviossa 26 on parametrien valintanäyttö, jossa on painike jokaiselle ohjaimen parametrisivulle. Kuviossa on esimerkkinä esitetty toteutus ohjaimen ensimmäisen parametrisivun, Drive status, osalta. Drive status-painiketta painaessa, ohjelma vaihtaa visualisoinnin kuvion 31 mukaiseen näyttöön, samalla ohjelma suorittaa lyhyen ST-koodin, jolla COB-ID 0x399 ensimmäinen tavu vaihdetaan arvoon 1, eli ohjaimelta pyydetään ensimmäisen parametrisivun arvot.

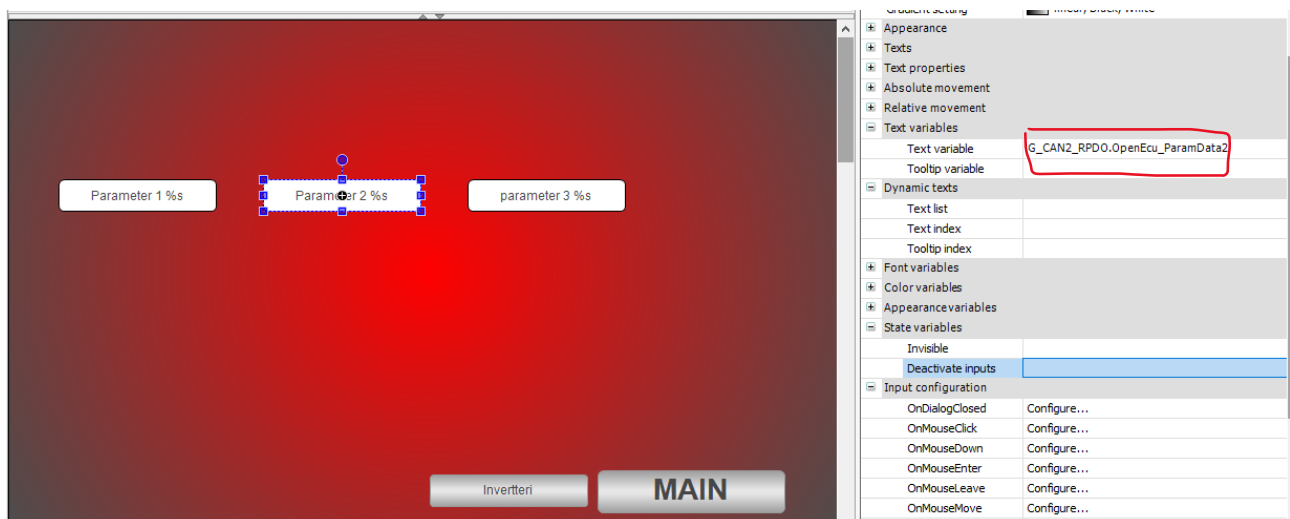


Kuvio 31 Parametrisivu valintanäyttö

Parametri sivut ovat käytännössä keskenään identtiset, koska valintanäyttö toteuttaa parametrisivun pyytämisen ohjaimelta. Kuviossa 32 on nähtävillä, miten invertteri painike palaa takaisin parametrisivun valintanäyttöön, lisäksi se vaihtaa ensimmäisen tavun arvoksi nollan. Kuviossa 33 on esitelty, miten itse parametrin arvo on tuotu näyttöön.



Kuvio 32 Parametrinäyttö



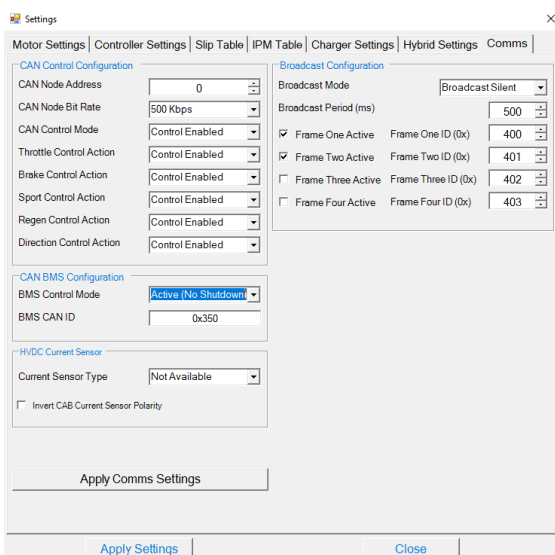
Kuvio 33 Parametrinäyttö parametrit

4.4 Moottorin ohjaus

Työssä ohjainta hallitaan kosketusnäytöllä olevalla liukuohjaimella. Kosketusnäytön ohjelmointi on tehty Codesys 3.5-ohjelmistolla, päänäyttö on esitetty kuviossa 34. Kuvioon on lyhyesti kirjoitettu kuvaus näytöllä esiintyvistä elementeistä sekä painikkeista. Scott Drive-ohjainta hallinnoidaan SEV View-ohjelmistolla, jonka valmistaja toimittaa pyynnöstä. CAN-väylä ohjauksen toteuttamisen ensimmäinen vaihe on vaihtaa laitteen asetuksista CAN-ohjaus enabled-tilaan (Kuvio 35)



Kuvio 34 Päänäyttö



Kuvio 35 SEV View Communication setting valikko

4.4.1 CAN-väylä ohjauksen käyttöönotto

Ohjauksen käyttöönottamiseksi ohjaimelle tulee lähettää kolme viestiä. Viestien tarkempi sisältö on kuvattuna liitteessä 2. Viestien toimivuus varmistettiin PCAN-lukijalla sekä PCAN-view ohjelmistolla.

Ohjaimelle lähetetään kuvion 36 mukainen viesti, johon ohjain vastaa kuviossa 37 esitetyllä viestillä. Vastausviesti eroaa ohjausviestistä viimeisen tavun osalta, joka osoittaa ohjaimen hyväksyvän CAN-ohjauksen käyttöönoton. Valmistajan ohjeen mukaan tämä viesti tulee lähettää ohjaimelle jokaisen käynnistyksen yhteydessä (Scott Drive n.d. 24). Viestien tarkempaa muodostamista käsitellään kappaleessa 5.4.2.

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	01	01	01	01	01	01	00
	Osoite	Viesti tyyppi	Jarru ohjaus	Kaasu ohjaus	Regen ohjaus	Sport ohjaus	Suunta ohjaus	Tyhjä

Kuvio 36 CAN-väylä ohjauksen konfigurointi viestikehys

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	01	01	01	01	01	01	01
	Osoite	Viesti tyyppi	Jarru ohjaus	Kaasu ohjaus	Regen ohjaus	Sport ohjaus	Suunta ohjaus	CAN ohjaus

Kuvio 37 CAN-väylä ohjauksen vastauskehys

Toinen ohjaimen tarvitsema viesti on digitaalisten parametrien ohjaus. Ensimmäisessä viestissä määritettiin suuntaohjauksen käyttöönotto, toisessa viestissä määritetään suunta. Digitaalisten parametrien ohjausviesti on kuviossa 38. Huomioitavaa esitetyssä viestissä on, että tavut 2 ja 3 voisivat olla myös 00, viestityyppi tavussa 1 sekä suunta tavussa 4 ovat merkityksellisiä. Vastavasti kuin edellisessä viestissä, vastauskehys on identtinen, mutta viimeinen tavu on 01 mikäli viesti on hyväksytty. Vastausviestin rakenne kuvattu kuviossa 39.

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	03	01	01	01	00	00	00
	Osoite	Viesti tyyppi	Regen ohjaus	Sport ohjaus	Suunta ohjaus	Tyhjä	Tyhjä	Tyhjä

Kuvio 38 CAN-väylä digitaalisen ohjauksen ohjauskehys

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	03	01	01	01	00	00	01
	Osoite	Viesti tyyppi	Regen ohjaus	Sport ohjaus	Suunta ohjaus	Tyhjä	Tyhjä	CAN ohjaus

Kuvio 39 CAN-väylä digitaalisen ohjauksen vastauskehys

Kaksi edellä esitettyä viestiä lähetetään näytöltä painamalla CAN-ohjaus-painiketta. Näytön ohjelma käsitellään kappaleessa 5.4.2. Näiden viestien lähettäminen luonnollisesti kytkee analogisen ohjauksen mahdollisuuden pois käytöstä, ohjainta ei voida tässä vaiheessa enää ohjata käyttöpaneelin (Kuvio 22, 1) ohjainkahvalla.

Ohjainta voidaan tämän jälkeen hallita kosketusnäytöltä painamalla Ohjauksen käyttöönotto-painiketta (Kuvio 34, 4) ja ohjaamalla näytön liukukytkimellä ohjaimen moottorille välittämää nopeustietoa. Ohjausviesti on esitetty kuviossa 40 ja vastausviesti kuviossa 41.

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	02	00 00 - 10 27		00 00 - 10 27		00	00
	Osoite	Viesti tyyppi	Ohjaimen asento		Jarrun asento		Tyhjä	Tyhjä

Kuvio 40 CAN-ohjaus analogisten parametrien ohjauskehys

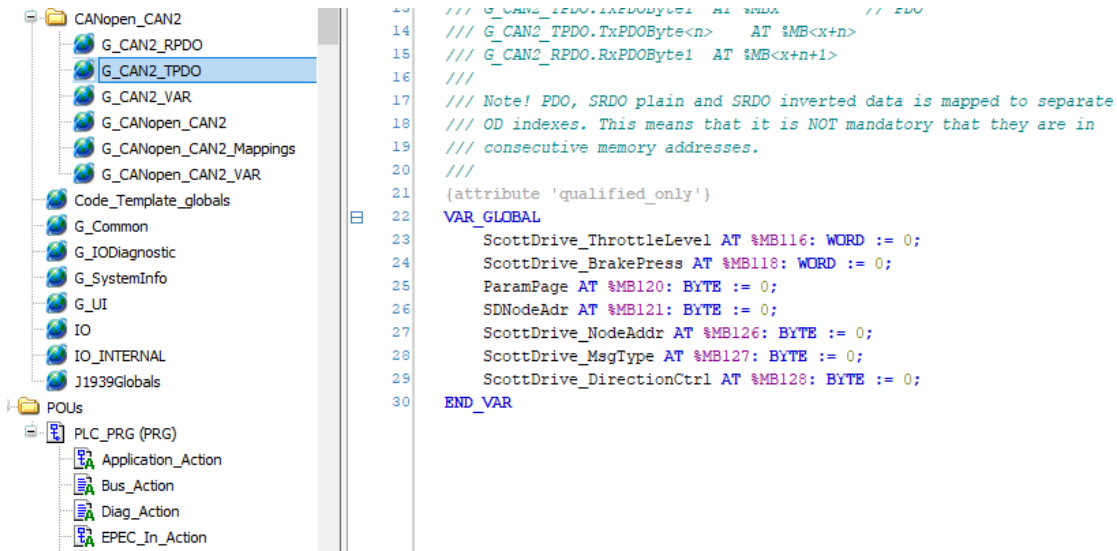
COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	00	02	00 00 - 10 27		00 00 - 10 27		00	01
	Osoite	Viesti tyyppi	Ohjaimen asento		Jarrun asento		Tyhjä	CAN ohjaus

Kuvio 41 CAN-ohjaus analogisten parametrien vastauskehys

Liikusäätimen arvo on välillä 0–10000, joka on $0-100\% * 100$. Tämä määritellään valmistajan ohjeessa (Liite 2). Viestiä tarkemmin tutkiessa huomataan maksimi arvon olevan 10 27, joka heksadesimaalista desimaaliksi muunnettuna on 4135. Ohjain käyttää viesteissä nousevaa tavujärjestystä, eli viestin sisältö tulee tulkita käänteisessä järjestyksessä eli 27 10, joka on heksadesimaalista desimaaliksi muutettuna 10 000. Tätä kutsutaan muun muassa Intel tai little endian tavujärjestykseksi. Otetaan esimerkkinä luku 0x12345678. Laskevalla eli MSB lukujärjestyksellä, big endian, joka voidaan ajatella normaalina tapana kirjoittaa luku, tämä luku olisi muodossa 12 34 56 78. Jokainen kahden numeron sarja edustaa yhtä tavua. Nousevassa LSB, eli little endian, lukujärjestyksessä tämä sama luku ilmoitettaisiin muodossa 78 56 34 12. (Jorquera 2020) Näytössä oleva kierroslukumittari on ohjaimen moottorilta pyytämä kierrosluku, joka on skaalattuna moottorin datalehden mukaiseen 0–4500 rpm. Tässä vaiheessa kierroslukua ei siis lueta moottorilta tai ohjaimelta.

4.4.2 Ohjauksen ohjelmointi CODESYS ohjelmistolla

Näytön ohjelman pohja luotiin Multitool Creator-ohjelmistolla, joka on esiteltyinä kappaleessa 2.6. Kuviossa 42 näkyy Multitool Creator-ohjelman luomat CAN-väylään tarvittavat muuttujat.



Kuvio 42 CAN-väylä muuttujat CODESYS

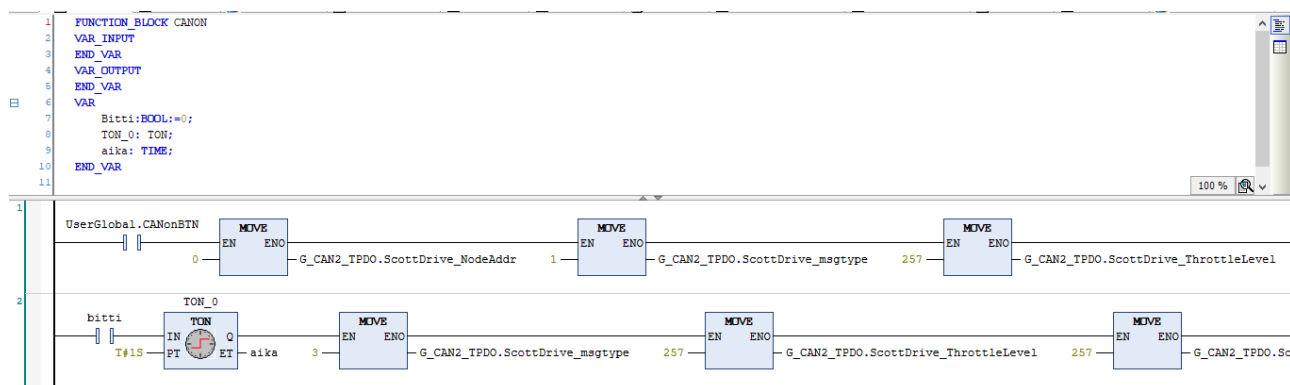
Kuviossa 43 on viestikehys esitettyä aiemman kuvion muuttujilla. CAN-väylän ohjauskehyksessä ja vastauskehyksessä oli tavujen 2 ja 3 sekä tavujen 4 ja 5 arvoina joko 00 tai 01, mutta kuviosta voidaan huomata muuttujatyypin olevan kahden tavun mittainen, jolloin ei voida lähettää arvoa 1 tavulle. Tavoiteltu arvo on heksadesimaalina 0101, joka desimaalina on 257.

COB-ID	Tavu 0	Tavu 1	Tavu 2	Tavu 3	Tavu 4	Tavu 5	Tavu 6	Tavu 7
0x39B	%MB126	%MB127	%MB116		%MB118		%MB128	
	ScottDrive NodeAddr	ScottDrive MsgType	ScottDrive ThrottleLevel		ScottDrive BrakePress		ScottDrive DirectionCtrl	

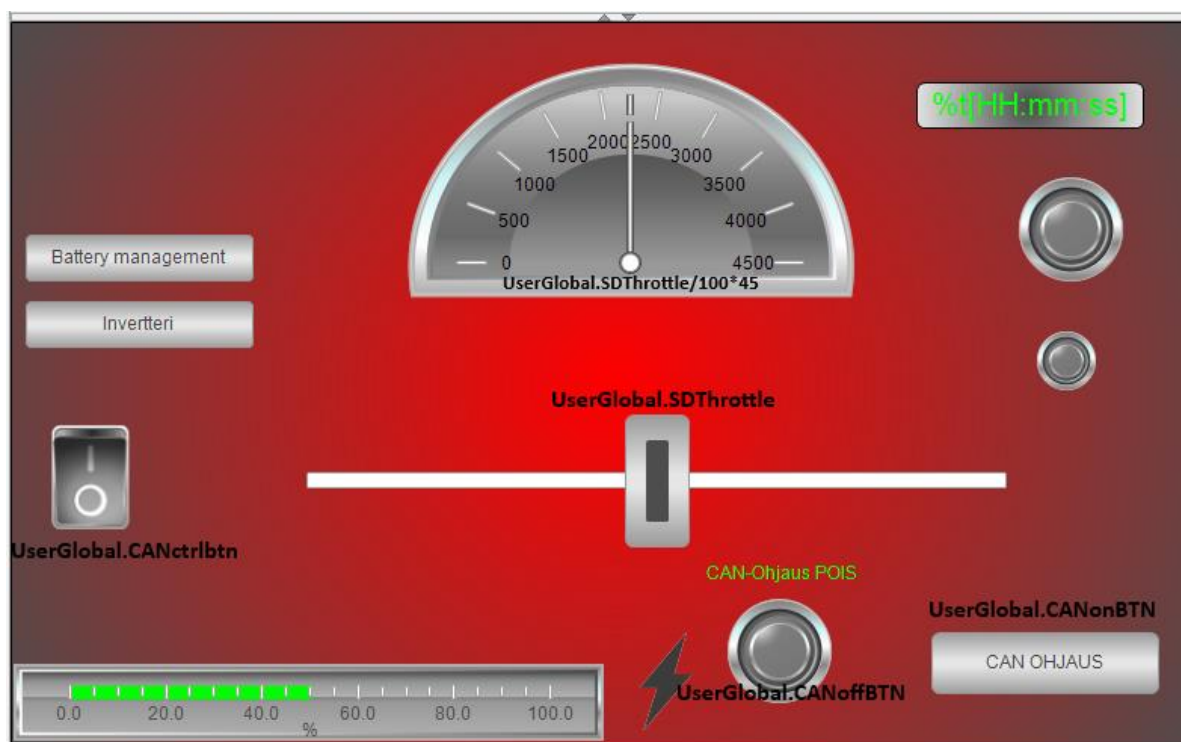
Kuvio 43 CAN-viestikehys muuttujilla

Ohjelmointi suoritettiin käyttämällä LD-ohjelmointia. Kuviossa 44 esimerkki miten CAN-ohjauksen käyttöönotto tapahtuu. Painettaessa CANonBTN-painiketta (Kuvio 45), vaihdetaan muuttujien arvot. Muuttujat ovat määriteltynä TPDO:ksi, näyttö lähettää viestiä käynnistyksestä lähtien, tavujen arvoja muutetaan käyttämällä MOVE lohkoa siirtämällä haluttu arvo muuttujaan. Kuvion

ulkopuolelle jää muuttuja listataan lisätty BOOL tyyppinen bitti-muuttuja, joka saa arvon 1 ensimmäisen viestin muutoksen päätyttyä. Tätä muuttujaa käytetään ajastimessa, jolla saavutetaan pieni viive ensimmäisen ja toisen viestin lähetyksen välillä. Toisen viestin jälkeen bitti-muuttuja saa jälleen arvon 0.

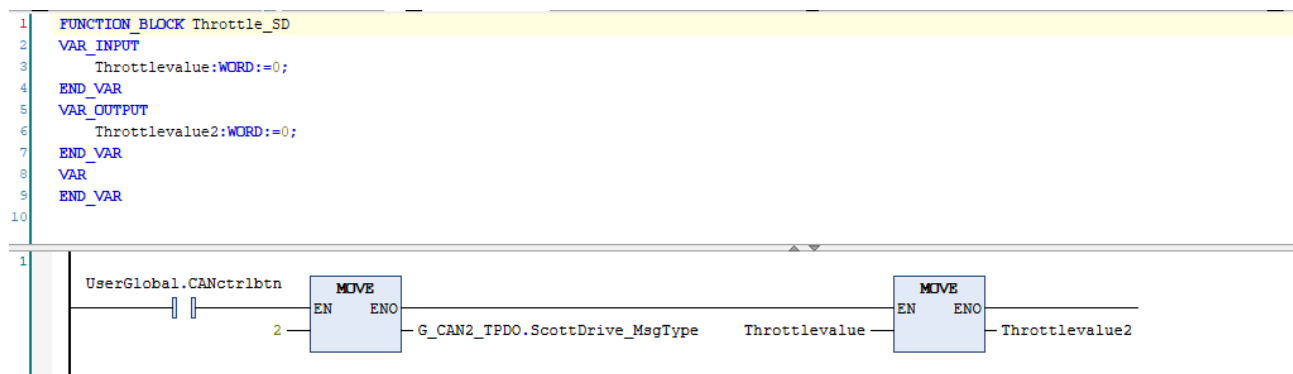


Kuvio 44 CAN-ohjauksen käyttöönotto LD

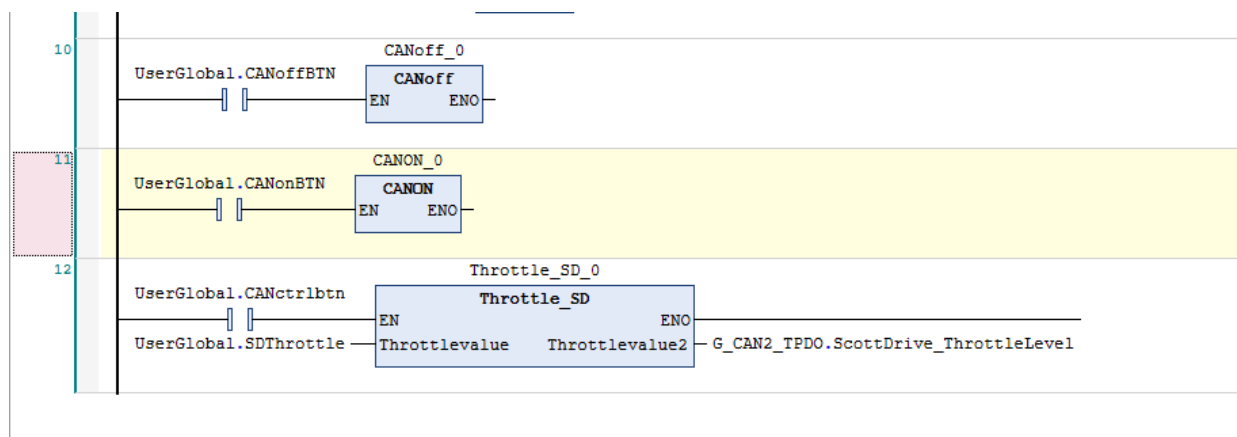


Kuvio 45 Kosketusnäytön päänäyttö muuttujananimillä

Scott Drive on tämän jälkeen valmis ohjattavaksi CAN-väylän kautta. Näytöltä painetaan CANctrlbtn-muuttujaan sidottua painiketta, jonka jälkeen liukukytkimellä voidaan hallita ohjaimen moottorille välittämää kaasuvivun asentoa. Kuviossa 46 on LD:llä toteutettu kuvaus liukukytkimestä. Painikkeen painallus lähettää arvon 2 tavuun 2. Liukusäädin on toteutettu käyttämällä funktioblokin sisäisiä muuttujia. Muuttujat ovat määritelty VAR_INPUT ja VAR_OUTPUT muuttujiksi. Muuttujan arvot liitetään funktioblokkiin vasta pääohjelmassa (kuvio 47). Samaa funktioblokkia voidaan käyttää tarvittaessa muualla ohjelmassa, kun varsinaisia muuttujia ei ole sidottu blokkiin.



Kuvio 46 CAN-control kaasuvivun asento



Kuvio 47 CAN-ohjaus pääohjelma CODESYS

5 Pohdinta ja lopputulokset

Opinnäytetyön tavoitteena oli toteuttaa Scott Drive-ohjaimen väyläohjaus CAN-väylää hyödyntäen. Teoriatiedon ja aikaisempiin aihetta, CAN-väylä ohjausta, käsitteleviin opinnäytetöihin perehtymisen jälkeen työ aloitettiin rakentamalla venetoteutukseen tarkoitettu laitteisto sähkölaboratorioon ja suorittamalla järjestelmän koeajo analogista ohjausta käyttäen. Tällä varmistettiin ohjaimen ja moottorin toimivuus ennen väyläohjauksen toteuttamista. Väyläohjauksen suunnittelu aloitettiin perehtymällä ohjainvalmistajan käyttöohjekirjaan ja CAN-väylän teoriaan tarkemmin. Osoittautui, ettei laitevalmistaja käytä standardinmukaista CAN-protokollaa, tietopohjaan tarkasteltavaksi valittiin CANOpen-protokolla sen yhtäläisyyksien vuoksi ohjaimen protokollan kanssa, lisäksi muut väylässä käytössä olleet laitteet perustuvat CANOpen-protokollalle. Opinnäytetyön rajaaminen osoittautui haastavaksi. Työhön sisältyi useita kokonaisuuksia suunnittelusta, asennuksesta sekä ohjelmoinnista, jokaisesta osa-alueesta olisi voinut tehdä erillisen opinnäytetyön, tein rajoja erityisesti laitteiden esittelyyn ja teknisiin yksityiskohtiin liittyen. Yksi rajoituksen peruste oli tavoiteltu kohdeyleisö, jolta osin aiheen käsittelyä rajattiin teknisten perustiedon omaaville, koska opinnäytetyön laajuuden puitteissa jokaisen käsitellyn asian yksityiskohtainen käsittely ei olisi tarkoituksenmukaista. Tällaisesti kenties selkein esimerkki oli lukumuunnosten rajaaminen ulkopuolelle, koska lukumuunnokset voi toteuttaa käyttämällä laskuria, enkä tältä osin pitänyt mielekkäänä käydä läpi lukumuutosten teoriaa.

Opinnäytetyölle asetetut tavoitteet saavutettiin väyläohjauksen toteutuksen osalta. Työn toteutuksessa haastetta tuotti viiveet yhteydenpidossa laitevalmistajan kanssa. Ohjaimen ohjelmisto ei ollut ajan tasalla, eikä vanhalla ohjelmalla CAN-väylän käyttöönotto onnistunut. Kun laitevalmistaja toimitti ohjaimen päivitystiedostot, työ saatiin nopeasti päätökseen. Tarvittava ohjauslogiikka ja ohjelma oli toteutettu Codesys-ohjelmistolla näyttöön työn aikaisessa vaiheessa, sen muokkaaminen ja käyttöönotto tapahtui sujuvasti ohjaimen laitepäivityksen jälkeen. Ohjaimen ja moottorin kierrosnopeuden hallinta toimi kosketusnäytöltä oletetulla tavalla. Itse näytön visuaalisuuteen ei työssä kiinnitetty suurta huomioita, ohjauslogiikka sekä parametrien nouto toimivat, vaikka visualisointiin olisi ollut mahdollista panostaa enemmän. Kirjallisessa työssä pyrin havainnollistamaan lähestymistapaan, jossa koetin tuoda selkeästi esiin sekä fyysisen osuuden laitteiden asennuksesta ja johdotuksista että ohjelmistopuolen käytännön toteutuksesta. Tämän vuoksi käytin runsaasti kuvia havainnollistaakseni käytännön esimerkkejä sekä miten nämä käytännössä voi toteuttaa. Työn toteutuksessa yksi rajoite oli soveltuvan A/D-muuntimen puute, joka tässä työssä korvattiin

kosketusnäytön kautta toteutetulla ohjelmalla ja ohjauksella. Käytännössä A/D-muuntimen avulla olisi mahdollista muuttaa erillisen kaasupolkimen tai -kahvan antama signaali CAN-väylä viestiksi, joka olisi perinteisempi tapa ajoneuvosovelluksissa. Työn aiheena kuitenkin oli toteuttaa väyläohjaus, näiltä osin asetetut tavoitteet saavutettiin. Jatkossa kehittämistä voisi keskittää erilaisten ulkoisten hallintalaitteiden lisäämiseen väyläohjaukseen.

Asiaa voidaan tarkastella globaalilla tasolla tavoitteena hillitä päästöjä, mutta myös paikallisella tasolla lähipäästöjen ja kaupunkialueiden ilmanlaadun kautta, eivätkä tavoitteet ole keskenään ristiriidassa. Sähköautojen hankintakustannukset ovat merkittävät yksilön kannalta, kiertotalouden kautta pohtien, ajoneuvojen konvertointi polttomoottorikäyttöisestä sähkökäyttöiseksi on kustannuksiltaan matalampi. Vaikka tämä ei välttämättä massatuotantoon soveltuva toimintamalli ole, on konvertoinnilla mahdollisuuksia ottaa oma roolinsa liikenteen sähköistämisessä ja tätä kautta päästöjen vähentämisessä. Kyseessä on varsin tuore aihealue, joten kustannustaso on vielä korkealla eikä osaamista aiheeseen löydy samassa mittakaavassa kuin perinteisemmästä ajoneuvojen teknisestä toteutuksesta, mutta teknologian kehittyessä ja yleistyessä hinnat voivat laskea ja tulevat taloudellisesti mahdolliseksi suuremmalle joukolle. Tämä avaa mahdollistuksia uudennaisille liiketoimintamalleilla, kuten myynti-, huolto-, ja asiantuntijapalveluille.

Työn kokonaisuutta arvioidessa koen saavuttaneeni niin työlle asetetut tavoitteet kuin myös henkilökohtaiset oppimistavoitteet. Automaatiotekniikan opinnoissa väylätekniikka on olennainen osa kokonaisuutta, mutta CAN-väylä poikkeaa merkittävästi automaatiotekniikassa käytetystä väylätekniikasta. Omana tavoitteena, näin autoharrastajana, oli ymmärtää CAN-väylää ja sen antamia mahdollisuuksia. Tämän työn toteuttaminen antoi paljon uutta tietoa ja ymmärrystä, niin ammatillisesti, kuin myös omiin harrastuksiin.

Lähteet

Alanen, J. & Scholliers, J. 2004. EN 13149 -standardin sovellusohje. 2004. Helsinki: FITS-julkaisuja 2004. Viitattu 20.11.2023. http://virtual.vtt.fi/virtual/proj6/fits/julkaisut/hanke1/fits33_2004.pdf.

Alanen, J. 2000. CAN - ajoneuvojen ja koneidensisäinen paikallisväylä. Verkossa julkaistu dokumentti. VTT AUTOMAATIO. Viitattu 20.11.2023. <https://docplayer.fi/10821742-Can-ajoneuvojen-ja-koneiden-sisainen-paikallisvayla.html>.

CAN-bus Topology and Termination. 2023. Artikkel National Instruments yrityksen sivustolla. Viitattu 30.10.2023. <https://www.ni.com/docs/en-US/bundle/sbrio-9627-feature/page/can-bus-topology-termination.html>.

CANopen Guide (2023): Simplified Intro to Network Protocols. 2022. Viitattu 30.10.2023. <https://www.autopi.io/blog/canopen-simple-introduction-explained/>.

CANopen tutorial – Simple intro. 2022. Artikkel CSS electronics yrityksen verkkosivustolla. Viitattu 20.11.2023. <https://www.csselectronics.com/pages/canopen-tutorial-simple-intro>.

CANopen tutorial. 2019. Verkojulkaisu Siemens Industry Online Support sivustolla. Viitattu 20.11.2023. https://cache.industry.siemens.com/dl/files/771/109479771/att_993267/v1/109479771_CANopen_Tutorial_V20_en.pdf.

Controller Area Network (CAN) Overview. 2023. Artikkel National Instruments yrityksen sivustolla. Viitattu 30.10.2023. <https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/controller-area-network--can--overview.html>.

Emrax 268. N.d. Emrax yrityksen sivusto. Viitattu 31.12.2022. <https://emrax.com/e-motors/emrax-268/>.

Epec 6107 Display. 2019. Tekninen käsikirja Epec yrityksen sivustolla. Viitattu 1.1.2024. https://extranet.epec.fi/Public/Technical_Documents/6107/MAN000596/6107_web-help_man000596.htm.

Gustafsson, R & Blomqvist, N. 2016. Options handling using external devices in forklift trucks. Opinnäytetyö. <https://www.diva-portal.org/smash/get/diva2:907584/FULLTEXT01.pdf>

History of CAN technology. N.d. Artikkel CAN in Automation yhdistyksen verkkosivustolla. Viitattu 11.12.2023. <https://www.can-cia.org/can-knowledge/can/can-history/>.

Introducing m580. N.d. Tuote-esittely sivu OpenEcu yrityksen verkkosivustolla- Viitattu 2.1.2024. <https://openecu.com/product/m580/>.

Introducing Scott Drive. N.d. Artikkel Scott Drive yrityksen verkkosivustolla. Viitattu 10.12.2023. <https://scottdrive.co.nz/>.

Jorquera, Z. 2020. What Is Little-Endian And Big-Endian Byte Ordering? Artikkel section.io verkkosivustolla. Viitattu 28.12.2023. <https://www.section.io/engineering-education/what-is-little-endian-and-big-endian/>.

Liikenteen palvelujen muutostrendit. 2023. Artikkel Liikenne- ja viestintä viraston verkkosivuilla. Viitattu 8.12.2023. <https://tieto.traficom.fi/fi/tilastot/liikenteen-palvelujen-muutostrendit>.

Ojasalo, K., Moilanen, T. & Ritalahti, J. 2009. Kehittämistyön menetelmät. Uudenlaista osaamista liiketoimintaan. Helsinki: WSOYpro Oy.

Truck.fi Yritys. N.d. Yritysesittely Lielahden autokeskus Oy:n sivustolla. Viitattu 9.12.2023. <https://lielahdenautokeskus.fi/yritys/>

Trucks.fi Sähköajoneuvot. N.d. Artikkel Lielahden autokeskus Oy:n sivustolla. Viitattu 9.12.2023. <https://lielahdenautokeskus.fi/sahkoajoneuvot/>

Orion BMS 2. N.d. Tuote-esittely Orion yrityksen verkkosivustolla. Viitattu 1.1.2024.
<https://www.orionbms.com/products/orion-bms-standard/>.

Parikh, B. N.d. CAN protocol: Understanding the controller area network. Artikkele Engineers garage verkkosivustolla. Viitattu 30.12.2023. <https://www.engineersgarage.com/can-protocol-understanding-the-controller-area-network-protocol/>.

PCAN View. N.d. Ohjelmisto esittely PEAK-system yrityksen verkkosivustolla. Viitattu 22.1.2024.
<https://www.peak-system.com/PCAN-View.242.0.html?&L=1>

Reif, K. & Dietche, K-H. 2014. Automotive Handbook. 9.p. Karlsruhe: Robert Bosch GmbH.

ScottDrive User Manual. N.d. 1.7.p. ScottDrive SD-sarjan invertterin käyttöohje. Valmistajalta saatavissa, ei julkisessa levityksessä.

Simulink is for Model-Based Design. N.d. Ohjelmistoesittely MathWorks yrityksen verkkosivustolla. Viitattu 22.1.2024. <https://se.mathworks.com/products/simulink.html>.

Software Development Environment. N.d. Epec yrityksen verkkosivusto. Viitattu 20.1.2024.
<https://epec.fi/epec-oy-products/application-development-environment/>

Suomi sähköisen liikenteen edelläkävijä vuonna 2030. 2023. Artikkele Teknologiateollisuuden sähköisen liikenteen verkkosivustolla. Viitattu 9.12.2023. <https://emobility.teknologiateollisuus.fi/fi/suomi-sahkoisen-liikenteen-edellakavija-vuonna-2030>.

Technical Data and Manual for EMRAX Motors/Generators. 2018. Emrax moottorin käyttöohjekirja yrityksen sivustolla. Viitattu 31.12.2023. [https://emrax.com/wp-content/uploads/2017/10/user manual for emrax motors.pdf](https://emrax.com/wp-content/uploads/2017/10/user_manual_for_emrax_motors.pdf).

The CAN Bus Protocol tutorial. N.d. Artikkele Kvaser yrityksen verkkosivustolla. Viitattu 20.12.2023-
<https://www.kvaser.com/can-protocol-tutorial/>.

The company. N.D. Yritysesittely Codesys yrityksen verkkosivustolla. Viitattu 21.1.2024.
<https://www.codesys.com/company.html>.

The OSI Model: Breaking Down Its Seven Layers. N.d. Artikkele Radware verkkosivustolla. Viitattu 31.12.2023. <https://www.radware.com/cyberpedia/application-security/the-osi-model-breaking-down-its-seven-layers/>.

Tuote- ja ohjelmistokehitys. N.d. Epec yrityksen verkkosivusto. Viitattu 20.1.2024.
<https://ura.epec.fi/departments/tuote-ja-ohjelmistokehitys>.

What is the data link layer in the OSI model? 2023. Artikkele Ionos verkkosivustolla. Viitattu 30.12.2023. <https://www.ionos.com/digitalguide/server/know-how/data-link-layer/>.

Liite 1. Program- ja Engine Mode State (Scott Drive n.d. 74.)

Value	Name
0	DRIVE_OFFLINE
1	DRIVE_INITIALISATION
2	MEASURE_CURRENT_OFFSETS
3	MEASURE_RESOLVER_OFFSETS
4	MEASURE_VOLTAGE_OFFSETS
5	UPDATE_FLASH_DATA
6	DRIVE_STANDBY
7	DRIVE_CHARGE
8	DRIVE_ACTIVE
9	MEASURE_CHARGER_OFFSETS
10	IP_CHARGER_MODE
11	AUTOTUNING_MODE
12	SYSTEM_ERROR
13	FIRMWARE_UPDATE
14	SETTINGS_UPDATE
15	SENSOR_TUNING_MODE
20	CONTROLLER_SAFE_MODE

Program Mode States

Value	Name
0	ENGINE_IDLE
1	ROTOR_BRAKE
2	ROTOR_DRIVE
3	ROTOR_WATCH
4	ROTOR_IDLE
5	DRIVE_EXIT

Engine Mode States

Liite 2. CAN-ohjaus viestit (Scott Drive n.d. 22-24.)

Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Message Type	Control Mode Configuration: Type=1
Byte 2	Brake Control	Analogue Input Control:0, CAN Control: 1
Byte 3	Drive Control	Analogue Input Control:0, CAN Control: 1
Byte 4	Regen Control	Digital Input Control:0, CAN Control: 1
Byte 5	Sport Control	Digital Input Control:0, CAN Control: 1
Byte 6	Direction Control	Digital Input Control:0, CAN Control: 1
Byte 7	Unused	Leave zero

CAN-väylä ohjauksen konfigurointikehys

Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Command Type	Control Mode Configuration: Type=1
Byte 2	Brake Control	Analogue Input Control:0, CAN Control: 1
Byte 3	Drive Control	Analogue Input Control:0, CAN Control: 1
Byte 4	Regen Control	Digital Input Control:0, CAN Control: 1
Byte 5	Sport Control	Digital Input Control:0, CAN Control: 1
Byte 6	Direction Control	Digital Input Control:0, CAN Control: 1
Byte 7	CAN Control Mode	Disabled:0, Enabled:1

CAN-väylä ohjauksen vastauskehys

Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Message Type	Analogue Parameter Values: Type=2
Byte 2	Throttle Level Lo	Throttle Level Scaled by 100 i.e 53.45%=5345DEC=0x14E1
Byte 3	Throttle Level Hi	
Byte 4	Brake Pressure Lo	Brake Pressure Scaled by 100 i.e 25.50%=9F6DEC=0x9F6
Byte 5	Brake Pressure Hi	
Byte 6	Unused	
Byte 7	Unused	

CAN-väylä analogisten arvojen ohjauskehys

Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Message Type	Analogue Parameter Values: Type=2
Byte 2-3	Throttle Level	Throttle Level Scaled by 100
Byte 4-5	Brake Pressure	Brake Pressure Scaled by 100
Byte 6	Unused	0
Byte 7	CAN Control Mode	Disabled:0, Enabled:1

CAN-väylä analogisten arvojen vastauskehys

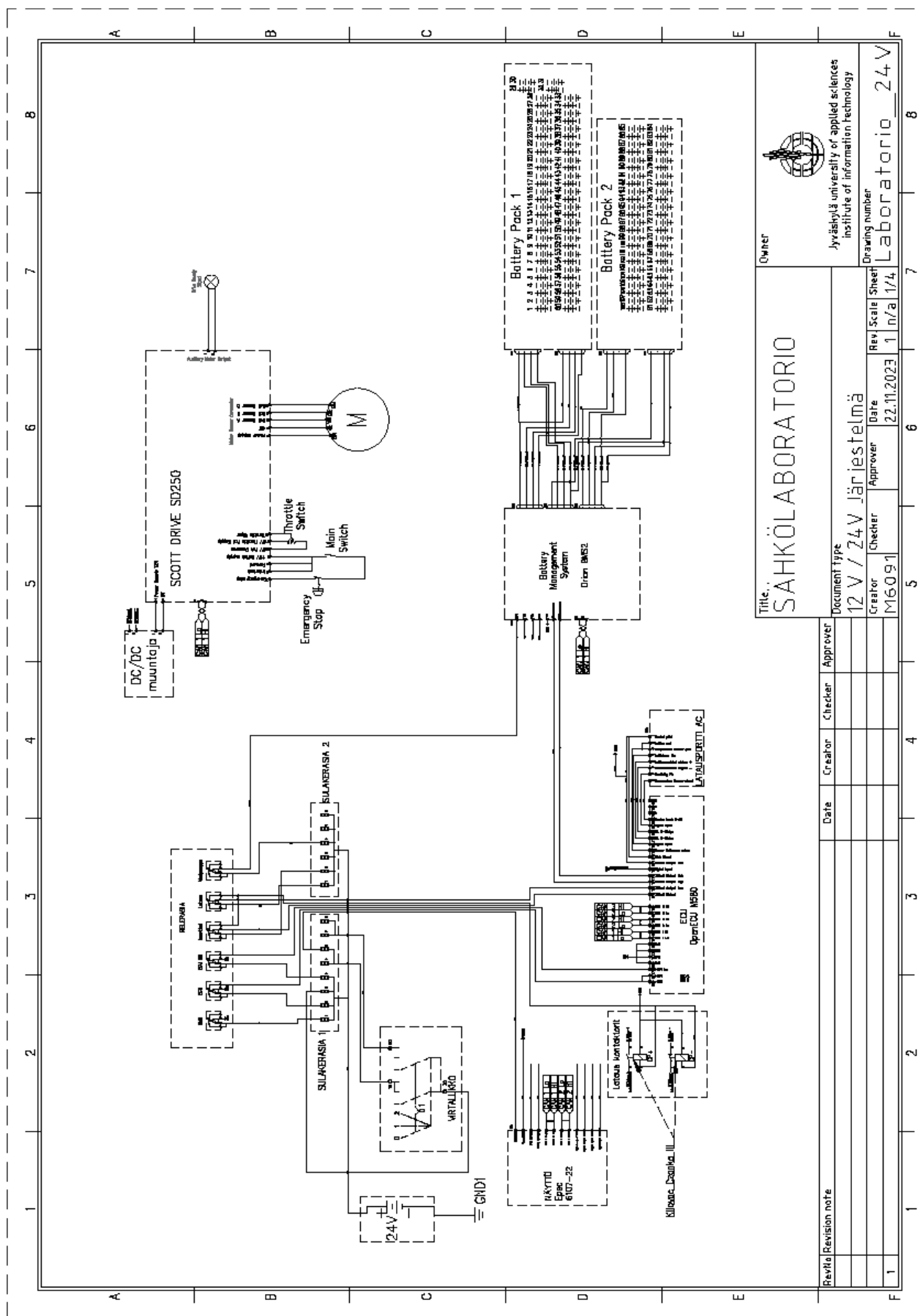
Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Message Type	Digital Parameter Values: Type=3
Byte 2	Regen Control	Regen Disable:0 Regen Enabled:1
Byte 3	Sport Control	Sport Mode OFF:0 Sport Mode ON:1
Byte 4	Direction Control	Neutral:0, Forward:1, Reverse:2
Byte 5	Unused	
Byte 6	Unused	
Byte 7	Unused	

CAN-väylä digitaalisten arvojen ohjauskehys

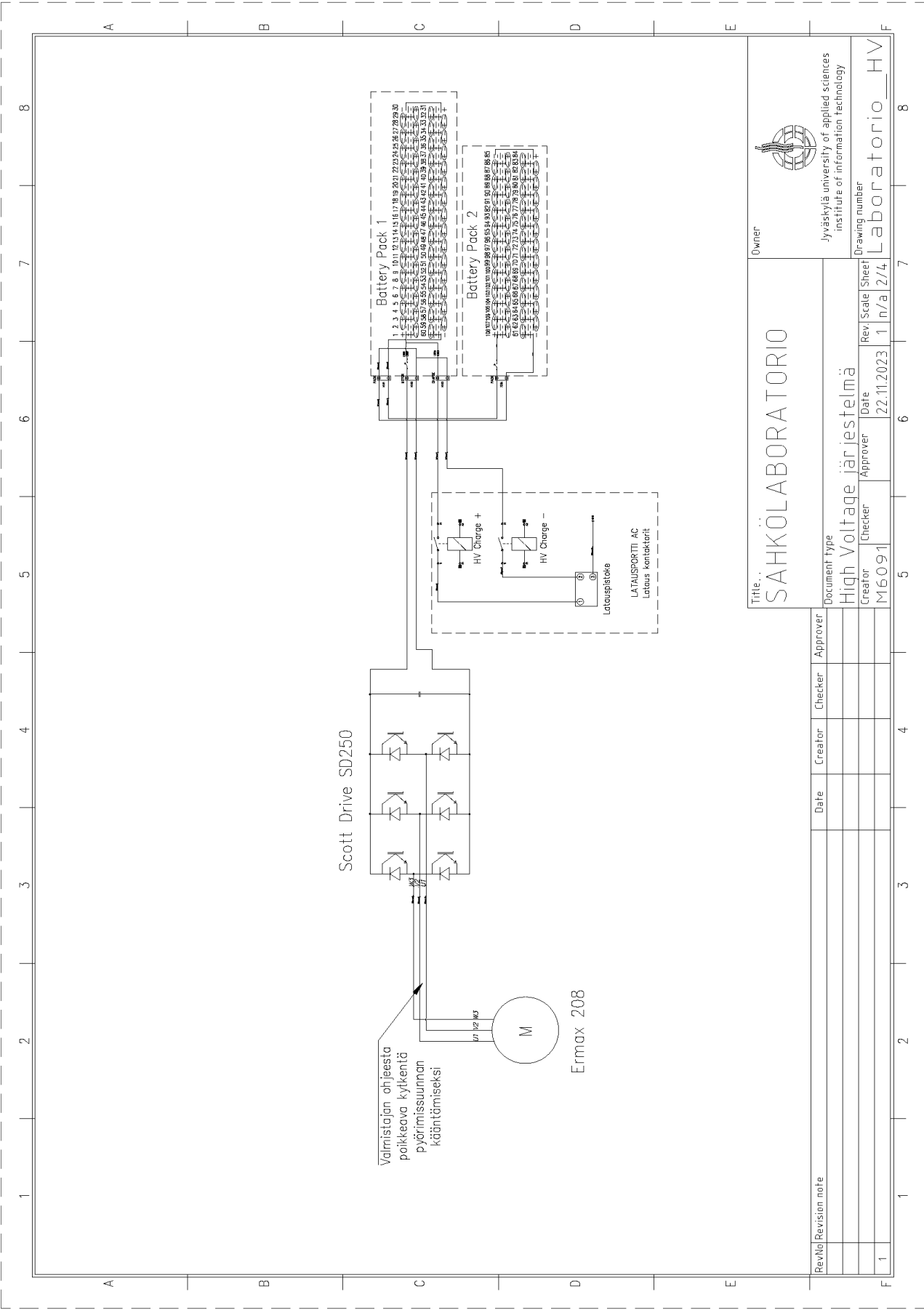
Field	Value	Description
Message ID	0x39B	CAN Mailbox ID
Byte 0	Address	CAN Node Address
Byte 1	Message Type	Digital Parameter Values: Type=3
Byte 2	Regen Control	Regen: Disable:0 Regen Enabled:1
Byte 3	Sport Control	Sport Mode OFF:0 Sport Mode ON:1
Byte 4	Direction Control	Neutral:0, Forward:1, Reverse:2
Byte 5	0	
Byte 6	0	
Byte 7	CAN Control Mode	Disabled:0, Enabled:1

CAN-väylä digitaalisten arvojen vastauskehys

Liite 3. 12 V ja 24 V kytkennät



Liite 4. High Voltage kytkennät



Liite 5. CAN-väylä

