



Sähkökuorma-auton latausjärjestelmän päivittäminen

Tapio Reko

OPINNÄYTETYÖ
Toukokuu 2019

Sähkö- ja automaatiotekniikan koulutus
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan koulutus
Automaatiotekniikka

REKO, TAPIO:
Sähkökuorma-auton latausjärjestelmän päivittäminen

Opinnäytetyö 67 sivua, joista liitteitä 2 sivua
Toukokuu 2019

Opinnäytetyön tarkoituksena oli perehtyä sähköajoneuvojen latausmenetelmiin ja tutkia mahdollisuuksia valmiin sähkökäyttöisen jakeluauton latausjärjestelmän päivittämiseen. Opinnäytetyössä sivuttiin myös sähköisten jakeluautojen markkina-asemaa ja ominaisuuksia kaupunkiajossa.

Projektin tärkeimmäksi osaksi muodostui uuden latauksenhallintajärjestelmän liittäminen vanhan järjestelmän tilalle. Uusi latauksenhallintajärjestelmä vaati konfigurointia ajoneuvon ohjausyksikölle, ajotilan näytölle, akunhallintajärjestelmiin, sekä näitä yhdistäviin automaatioväyliin. Automaatioväylinä käytettiin CAN-Open- sekä SAE J1939 -väyliä. Laitteiden ja väylien konfigurointia ja ohjelmointia suoritettiin pääasiassa Multitool- ja Codesys-ohjelmistoilla. Tehtyjä toimintoja testattiin CANmoon- ja CANoe-ohjelmistoilla.

Uuden latauksenhallintajärjestelmän tarvitsema laitteisto saatiin toimimaan yhdessä ja todettiin latausjärjestelmän päivittämisen olevan mahdollinen ja harkitsemisen arvoinen kehitysmahdollisuus sähköajoneuvoille. Nopeampi lataus mahdollistaa paremman käytettävyyssasteen ajoneuville, sekä avaa vaihtoehtoja akuston kapasiteetille: mikäli ajoneuvoa voidaan ladata useammin ja tehokkaammin, voidaan akuston kokoa pienentää. Tämä mahdollistaa säästöt kalliita akustoja ostettaessa.

Asiasanat: sähköajoneuvot, automaatio, väylät (tietotekniikka), latauslaitteet

ABSTRACT

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

REKO, TAPIO:
Updating the charging system of an electric truck

Bachelor's thesis 67 pages, appendices 2 pages
May 2019

The purpose for this Bachelor's thesis was to get acquainted with electric vehicles' charging methods and to investigate possibilities for updating the charging system of an existing electric truck. The thesis also touches on the markets and features of electric trucks in urban driving.

Integrating the new charging equipment to an existing system became the most important part of the project. There were three existing devices that required configuration. The first device was an existing main controller of the truck. The second device was a touch display in the cabin of the truck. The third and final device was a new battery management system which replaced the existing battery management system. Devices were connected to each other with automation busses which also required configuration. CANOpen and SAE J1939 were used for the automation busses. The programming and the configuring of the devices and busses were mainly done with Multitool and Codesys programming tools. The changes were tested with CANmoon and CANoe software.

The equipment required by the new charge control unit started working together as intended and it was concluded that the update is both possible and worth considering for electric vehicles. Faster charging speed increases the usability of the vehicle, which opens up possibilities for the capacity of the batteries: It is possible to use fewer batteries if they can be charged faster and more often. This can be exploited to save money during acquisition of the devices.

Key words: electric vehicles, automation, communication busses, charging equipment

SISÄLLYS

1	JOHDANTO	7
2	SÄHKÖAJONEUVOJEN LATAUSTAVAT	8
2.1	Perinteinen AC-lataus	8
2.2	DC-pikalataus.....	9
3	KÄYTETTY LAITTEISTO.....	10
3.1	Epec 5050-02 -ohjausyksikkö	10
3.2	Epec 6107-12 -kosketusnäyttö.....	11
3.3	Epec 3509-CAN-adapteri	11
3.4	Vector Group VC-VCCU -latauksenhallintayksikkö	12
3.5	Vector Group VN1610-väylätesteri.....	13
3.6	Phoenix Contact Vehicle Connector -latauspistorasia.....	14
3.7	Orion BMS 2 -akunhallintajärjestelmä	17
4	KÄYTETYT OHJELMISTOT	18
4.1	Epec MultiTool 5.7	18
4.2	Codesys 3.5	18
4.3	Orion BMS 2 Control Application.....	21
4.4	CANmoon	21
4.5	CANoe	21
5	JÄRJESTELMÄN SUUNNITTELU	22
5.1	Näyttösuunnittelu	22
5.1.1	Aloituspäyttö	22
5.1.2	Laboratoriotestinäyttö	23
5.1.3	Latauksenhallintanäyttö	25
5.1.4	Ilmoituslista.....	26
5.1.5	Testaus.....	27
5.2	Väyläkommunikaation sovellussuunnittelu	27
5.2.1	Väylärakenne	27
5.2.2	6107-12-näytön väyläkonfigurointi.....	30
5.2.3	VC-VCCU	34
5.2.4	Akunhallintajärjestelmä.....	37
5.3	Toiminnallisuuden sovellussuunnittelu	38
5.3.1	VCCU:n toiminnallisuus	38
5.3.2	6107-12-näytön toiminnallisuus	39
5.3.3	5050-02-ohjausyksikön toiminnallisuus	39
6	TOTEUTUS	40
6.1	Kokonaiskytkentä	40

6.2	5050-01-ohjausyksikön kytkentä	42
6.3	CAN-adapterin kytkennät	42
6.4	VC-VCCU:n kytkennät	44
6.5	6107-12-kosketusnäytön kytkennät.....	45
7	SOVELLUKSEN TESTAUS SIMULOIMALLA.....	46
7.1	CANoe	46
7.2	CANmoon	52
7.3	Codesys	54
8	JÄRJESTELMÄN TESTAUS	57
8.1	Akunhallintajärjestelmän käyttöönotto.....	57
8.2	Latauspistokkeen toiminta.....	59
8.3	VCCU:n latausohjelman testaus	62
9	POHDINTA	63
	LÄHTEET.....	64
	LIITTEET	66
	Liite 1. Laitteiden kytkennät	67

LYHENTEET JA TERMIT

BMS	Battery Management System
CAN	Controller Area Network
CCS	Combined Charging System
EVSE	Electronic Vehicle Supply Equipment
LEZ	Low-Emission Zone
PDO	PHP Data Object
SOC	State of Charge
VC-VCCU	Vector Controller - Vehicle Charge Control Unit

1 JOHDANTO

Ympäristökysymykset CO₂-päästöistä sekä kaupunki-ilman laadusta ovat johtaneet yhä useampiin vähäpäästöisiin LEZ-alueisiin Euroopan kaupunkialueilla ja yhä tiukempiin päästörajoituksiin (Sadler Consultants Ltd 2018). Tämä rajoittaa kaupunkiajoon soveltuvien jakeluautojen määrää, mikä on suotuisaa vähäpäästöisten ja etenkin päästöttömien jakeluautojen markkinoille.

Sähkökäyttöisten jakeluautojen käyttö Euroopan tavarankäytössä onkin lisääntynyt ja todettu toimivaksi ratkaisuksi LEZ-alueilla esimerkiksi Norjan Trondheimissä (Marcin Foltyński 2014, 57). Sähköajoneuvot soveltuvat hyvin perinteiseen kaupunkiajoon, jossa kiihdytellään ja jarrutellaan paljon. Polttomoottoriajoneuvoissa kiihdyttäminen aiheuttaa piikkejä päästöarvoihin, mitä sähköajoneuvojen kanssa ei tapahdu. Jarruttaminenkin on sähköajoneuvoille edullista, koska jarrutettavalla liike-energialla voidaan ladata akkuja (Ahn, K., Fiori, C. & Rakha, H. 2016, 12).

Sähköajoneuvojen tekniikassa on kuitenkin vielä parannettavaa ja tekniikka kehittykin vauhdikkaasti. Tästä johtuen on tärkeää huomioida myös olemassa olevien ajoneuvojen järjestelmien päivittäminen uusien innovaatioiden mukaisiksi. Tällöin käytössä olevien sähköajoneuvojen elinkaarta voidaan pidentää, eikä ajoneuvokantaa ole välttämättä vaihdettava uusien teknologioiden saapuessa markkinoille.

Tämä opinnäytetyö pyrkiikin perehtymään sähköisten jakeluautojen lataustekniikoihin ja perinteisen AC-latausjärjestelmän päivittämiseen DC-pikalatauksen kanssa yhteensopivaksi. Projektia tehdään pääasiassa laboratorioympäristössä ja laboratoriolaitteilla, mutta projektissa pyritään huomioimaan mahdolliset jatkokehittelyt, jossa suunniteltua laitteistoa voitaisiin ottaa käyttöön oikeaan jakeluautoon. Alkuperäisenä jakeluautona käytettiin Tampereen ammattikorkeakoulun eTruck-projektin toteuttamaa sähkökäyttöistä jakeluautoa (Lauri Hietalahti 2017). Projektissa pyrittiin jäljittelemään kyseisen jakeluauton alkuperäistä ohjelmistoa ja varustelua, sekä pyrittiin selvittämään, millaisia muutoksia päivitys vaatisi vanhoille laitteille ja sovelluksille.

2 SÄHKÖAJONEUVOJEN LATAUSTAVAT

Projektissa perehdyttiin AC-lataukselle suunnitellun jakeluauton päivittämiseen yhteensopivaksi DC-pikalatauksen kanssa. Ennen varsinaisia muutostöitä selvitettiin lataustapojen eroja ja DC-latauksen lisäämisestä saatavia hyötyjä.

2.1 Perinteinen AC-lataus

Vaikka sähköajoneuvojen moottorit ovat AC-moottoreita, akusto varastoi vain tasavirtaa. Tästä johtuen AC-latauksessa tuotava virta viedään ensin ajoneuvon muuntimelle, joka muuttaa vaihtovirran tasavirraksi, jolla voidaan ladata akkua. (Schwitters 2013.)

Lataustehoa rajoittaakin käytännössä ajoneuvossa oleva muunnin. Ajoneuvoihin asennetut muuntimet on yleensä mitoitettu vain tavallisesta verkkovirrasta tapahtuvaa latausta varten (Schwitters 2013). Tavoitteena on ollut, että sopivan lataus- aseman voi asentaa minne vain, yleensä ajoneuvon omistajan tontille. AC-lataukseenkin on saatavissa tehokkaampia muuntimia, joita voidaan käyttää tehokkaampien AC-verkkojen kanssa, mutta tämä nostaa muuntimen hintaa ja painoa, mikä on ongelma ajoneuvoissa.

Tavalliseen yksivaiheiseen pistorasiaan kytkettävät latauspistokkeet saavuttavat noin 3,7 kW tehon. Kolmivaihepistorasian käyttö nostaa tehon noin 11 kW:iin. (Garo 2013, 2.) Tehorajat perustuvat käytettyihin sulakkeisiin. Yksivaiheisella pistorasialla ja 16 A sulakkeella tehomaksimi voidaan laskea

$$P = U * I = 230 \text{ V} * 16 \text{ A} = 3,68 \text{ kW}.$$

Kolmivaiheiselle pistorasialle vastaava laskenta on

$$P = \sqrt{3} * U * I = \sqrt{3} * 400 \text{ V} * 16 \text{ A} = 11,085 \text{ kW}.$$

2.2 DC-pikalataus

DC-latauksessa vaihtovirran muuntaminen tasavirraksi suoritetaan latausasemaan sisältyvällä muuntimella (Schwitters 2013). Tällöin ajoneuvon omaa muunninta ei tarvita, vaan akustoa voidaan ladata suoraan.

DC-lataus mahdollistaa tehokkaampien muuntimien käytön ja merkittävän lataus-tehon paranemisen, koska voidaan käyttää muuntimia, joiden asentaminen ajoneuvoon olisi käytännössä mahdotonta niiden koon ja painon takia. DC-aseman korkea hinta myös kompensoituu, kun samoja laitteita voidaan käyttää useiden ajoneuvojen toimesta, toisin kuin ajoneuvoihin kiinteästi asennettujen muuntimien kanssa.

DC-latauksella saavutetaan latausasemasta riippuen 50 kW - 120 kW latausteho (McDonald 2014).

DC-latauksen ongelmiin sisältyy pääasiassa varsinaisen latausaseman haasteet. Asemat ovat kalliita ja toistaiseksi harvakseltaan sijoitettuja (McDonald 2014). Suuri latausteho asettaa myös vaatimuksia syöttöverkolle, josta virtaa otetaan. Näistä syistä johtuen ajoneuvoissa halutaankin säilyttää yhteensopivuus myös AC-latauksen kanssa, jotta voidaan tarvittaessa käyttää yleisemmin saatavilla olevia latausasemia (Schwitters 2013).

3 KÄYTETTY LAITTEISTO

Käytettävien laitteiden valinnat oli pääasiassa tehty ennen opinnäytetyön aloittamista, mutta niissä oli pyritty tekemään tarkoitukseen sopivia päätöksiä. Laboratorio-olosuhteisiin suunniteltavassa laitteistossa pyrittiin jäljittelemään alkuperäisen jakeluauton nykyistä laitteistoa, koska työn näkökulmana oli vanhan järjestelmän päivittäminen. Joidenkin laitteiden kohdalla siirryttiin uudempiin versioihin kuin alkuperäisessä laitteistossa, mutta suuria vaihdoksia alkuperäisiin laitteistoihin pyrittiin välttämään.

3.1 Epec 5050-02 -ohjausyksikkö

Vanhassa jakeluautossa oli käytössä suomalaisen Epecin ohjausyksikkö. Uuden järjestelmän piti olla yhteensopiva vanhan ajoneuvosovelluksen kanssa, joten uusikin ohjausyksikkö oli valittu Epeciltä, vaihdettiin vain uudempaan laiteversioon. Käyttöön valittiin Epecin 5050-02 -ohjausyksikkö, jossa pyrittiin käyttämään vanhaa ohjelmakoodia mahdollisimman tehokkaasti, tehden muutoksia vain niiltä osin, kuin uudet laitteistot vaativat. Ohjausyksikön ohjelmointiin käytettiin Epecin MultiTool 5.7 -ohjelmistoa sekä Codesys 3.5 -ohjelmistoa. Kuvassa 1 on esitetty 5050-ohjausyksikkö.



KUVA 1. 5050-ohjausyksikkö (Epec Oy 2019, Epec 5050 Control Unit)

3.2 Epec 6107-12 -kosketusnäyttö

Jakeluauton ajotilan kosketusnäyttö valittiin myös alkuperäisen mallin mukaisesti Epeciltä. Epecin 6107-12 -kosketusnäyttöä käytettiin muun muassa latauksen hallinnan esittämiseen ajoneuvon ajotilassa. Näytön ohjelmointi suoritettiin ohjausyksikön tavoin Epecin MultiToolilla sekä Codesys-ohjelmistolla. Kuvassa 2 on esitetty 6107-kosketusnäyttö



KUVA 2. 6107-kosketusnäyttö (Epec Oy 2019, Technical Manual and Cabling Instructions)

3.3 Epec 3509-CAN-adapteri

Laitteiden välinen kommunikaatio suoritettiin pääasiassa CAN-väylää käyttäen. Erilliset laitteet, kuten ohjausyksikkö, näyttö, sekä lataushallintayksikkö, liitettiin toisiinsa Epecin 3509-CAN-väyläadapterin kautta. Adapterille tuotiin 24 V käyttöjännite. Kuvassa 3 on esitetty 3509-CAN-adapteri.



KUVA 3. 3509-CAN-adapteri (Epec Oy 2019, Accessories and Ordering codes)

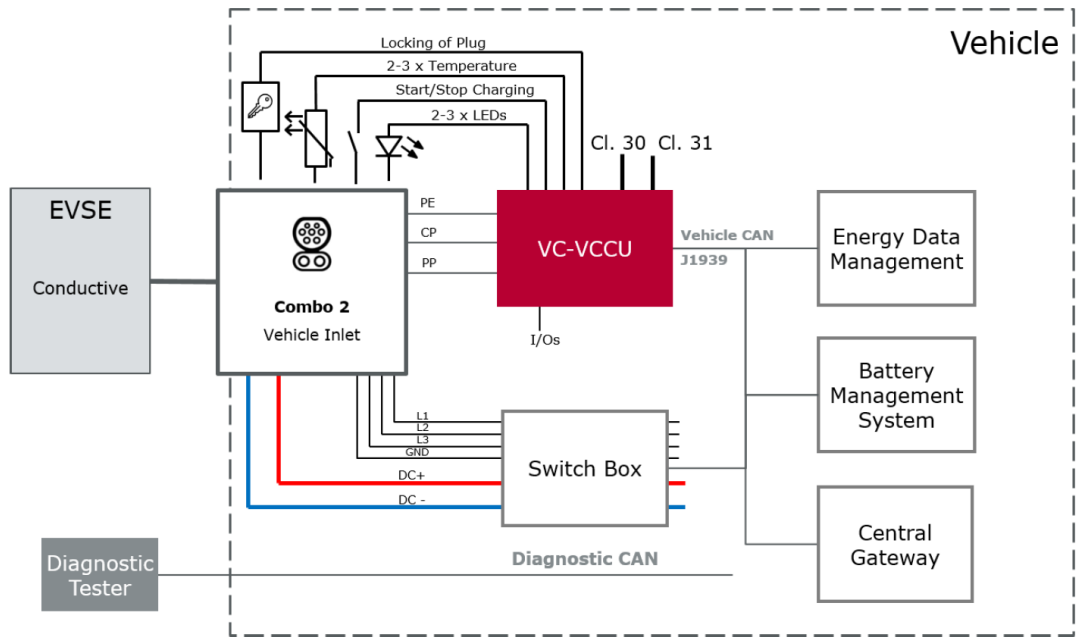
3.4 Vector Group VC-VCCU -latauksenhallintayksikkö

Merkittävin uusi lisäys vanhaan järjestelmään oli saksalaisen Vector Groupin VC-VCCU -latauksenhallintayksikkö (Vector Controller - Vehicle Charge Control Unit). Merkittävä osa työstä olikin perehtymistä latauksen hallinta yksikön toimintaan ja sen vaatimiin muutoksiin, joita oli vanhaan järjestelmään tehtävä. Kuvassa on esitetty VCCU-latauksenhallintayksikkö



KUVA 4. VCCU -latauksenhallintayksikkö (Bayer & Leserer 2018, 9)

Kuviossa 1 on esitetty VCCU:n periaatteellinen kytkentäkuva.



KUVIO 1. VC-VCCU:n teoreettinen kytkentäkuva (Bayer & Leserer 2018, 9)

Pääideana on, että latausasema EVSE kytkeytyy ajoneuvon Combo 2 -latauspistorasiaan, josta tasavirta-, vaihtovirta- ja nollajohtimet kytketään ajoneuvon akuston kytkinlaatikolle. Muut latauspistorasialta tulevat signaalit, viedään VCCU:lle, joka kommunikoi ajoneuvon akunhallintajärjestelmän kanssa ja ohjaa akkujen lataamista.

3.5 Vector Group VN1610-väylätesteri

Ohjausyksikön ja kosketusnäytön, sekä ohjausyksikön ja VCCU:n välisiä CAN-viestejä tarkkailtiin Vectorin VN1610-väylätesterillä. Testerit kytkettiin CAN-väylään CAN-väyläadapterin DE-9 sarjaportin kautta ja tietokoneeseen USB-liittimellä. Väylätesteri oli kaksikanavainen, eli sillä voitiin liittyä kahteen väylään samanaikaisesti. Kuvassa 5 on esitetty VN1610-väylätesteri.



KUVA 5. Vector VN1610-väylätesteri (Reko 2019)

Väylässä tapahtuva tietoliikenne saatiin avattua tietokoneelle CANmoon-ohjelmistoon ja CANoe-ohjelmistoon. Ohjelmistoilla voitiin testata signaaleja ja etsiä virheitä kytkettyjen laitteiden konfiguroinneista, mikäli yhteyksissä oli ongelmia. CANoe-ohjelmisto vaati Vectorin testerin, minkä takia päädyttiin käyttämään VN1610-testeriä. CANmoon-ohjelmisto oli yhteen sopiva useampien eri valmistajien testerien kanssa, joten se ei ollut rajoittava tekijä testerinä valittaessa.

3.6 Phoenix Contact Vehicle Connector -latauspistorasia

SFS-standardi 61851-1 sähköajoneuvojen latausjärjestelmistä määrittelee kolme vaihtoehtoista tapaa toteuttaa liityntä ajoneuvon ja latauspisteen välillä. Ensimmäisessä tavassa latauskaapeli on kiinteästi kiinni ajoneuvossa ja latauspistooli tuodaan latausaseman pistokkeeseen. Toinen mahdollinen tapa on kokonaan irroitettava latauskaapeli. Kolmas ja tässä projektissa käytetty tapa on latausasemalle kiinteästi liitetty latauskaapeli, joka tuodaan ajoneuvon latauspistorasiaan. (SFS EN 61851-1 2012, 14)

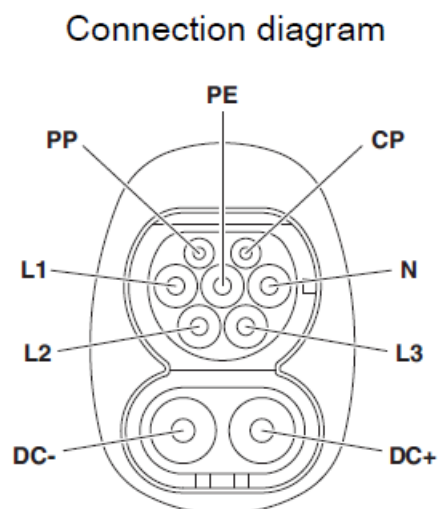
Latauspistorasiaksi valittiin Phoenix Contactin "Vehicle Connector test adapter - EV-T2MBIE12-3ACDC-INFRA - 1623492" -latauspistorasia. Kyseinen latauspistorasia on tarkoitettu nimenomaan laboratorio- ja testikäyttöön, mikä sopi projektin tarpeisiin hyvin.

Latauspistorasia on CCS-tyypin (Combined Charging System) mukainen latausyksikkö, joka mahdollistaa sekä AC- että DC-latauksen samalla ajoneuvoon kiinnitettävällä latauspistorasialla. Laite on tyyppiä 2 (Type 2) ja mallia Combo 2. Tyyppi 2 on liittimen eurooppalainen malli ja Combo 2 tarkoittaa, että pistorasia on yhteensopiva molempien AC- ja DC-latausten kanssa. (Phoenix Contact 2018, Vehicle Connector test adapter.)

Kuvassa 6 on esitetty latauspistorasia ja kuviossa 2 sen nastat.



KUVA 6. Phoenix Contact test adapter -latauspistorasia (Phoenix Contact 2018, Vehicle Connector test adapter)



KUVIO 2. CCS Type 2 -latauspistorasian nastat (Phoenix Contact 2018, Vehicle Connector test adapter)

Kuviosta 2 voidaan havaita, miten rasiän nastat on sijoitettu. Alimpana on tasavirran DC-nastat. Rasiän keskivaiheilta löytyy vaihtovirran vaiheet L1, L2 ja L3, sekä nollajohdin N. Yläosan keskimmäisenä nastana on maadoitus (PE), joka on käytössä molemmissa lataustavoissa. Maadoituksen jatkuvuuden valvonta on edellytyksenä lataussovellukselle (SFS EN 61851-1 2012, 18). Ylimpänä on kaksi nastaa, joita käytetään molempien lataustapojen latauksenhallintaan. Ensimmäinen on Proximity Pin (PP), jonka avulla tarkastellaan, onko latauspistooli paikallaan rasiassa. Toinen on Control Pilot (CP), jolla hallitaan käytettävää latausprofiilia, eli esimerkiksi millä teholla ladataan akun milläkin latausasteella (Bayer & Leserer 2018, 21).

Latauspistorasian ja latauspistoolin fyysinen rakenne tulee olla sellainen, että pistoolia kytkettäessä ensimmäisenä kontaktin saa maadoitusnasta, sitten vapaassa järjestyksessä CP-nastaa lukuun ottamatta muut nastat ja viimeisenä CP-nasta. Vastaavasti irroitettaessa pistoolia, ensimmäisenä katkeaa yhteys CP-nastalta ja viimeisenä maadoitusnastalta. (SFS EN 61851-1 2012, 23.) Tämä on toteutettu tekemällä liitinnastoista eri pituisia: PE-nasta on pisin ja CP-nasta lyhin, jolloin kontaktit tapahtuvat oikeassa järjestyksessä, kun pistooli kytketään latauspistorasiaan.

Latauspistorasian turvallisuudelta vaaditaan, että latauksen aikana käyttäjän ei ole mahdollista päästä koskemaan jännitteellisiin osiin. Jännitteellisten osien eristyksen ja suojauksen on lisäksi toteutettava niin, että niitä ei voi ilman työkaluja saada esille. (SFS EN 61851-1 2012, 19.) Laitteille on myös IP-vaatimukset IP55 ajossa ja IP44 latauksen aikana (SFS EN 61851-1 2012, 26), jotka laite täyttää (Phoenix Contact 2018, Vehicle Connector test adapter).

Latauspistorasiaan kuuluu myös Phoenix Contactin ”Locking - EV-T2M3S-E-LOCK12V - 1624129” -lukko, jolla varmistetaan pistoolin kiinnittyminen rasiaan latauksen ollessa käynnissä. Lukko on IEC 61851-1 -standardin vaatimusten mukainen (Phoenix Contact 2018, Locking). Standardi vaatii mekaanisen lukituksen latauspistoolille (SFS EN 61851-1 2012, 18).

3.7 Orion BMS 2 -akunhallintajärjestelmä

Akunhallintajärjestelmäksi valittiin Orionin BMS 2 -laitteisto. BMS 2 -laitteisto tukee J1939 CAN-väylää, samoin kuin aiemmin valitut 5050-ohjausyksikkö ja VCCU, joten se voidaan kytkeä niiden kanssa samaan väylään sen sijasta, että olisi tarvinnut ottaa käyttöön erillinen väylä. Kuvassa 7 on esitetty akunhallintajärjestelmä.



KUVA 7. Orion BMS 2 -akunhallintajärjestelmä (Ewert Energy Systems, Inc 2019)

4 KÄYTETYT OHJELMISTOT

Projektissa käytettiin useiden eri valmistajien laitteita, mikä asetti rajoitteita sovel-lussuunnittelulle. Laitteille löytyi valmistajiensa omat konfigurointisovellukset, joita käytettiin kunkin laitteen konfigurointiin.

4.1 Epec MultiTool 5.7

Epecin laitteiden ohjelmointi pyrittiin tekemään mahdollisimman pitkälti Epecin omalla MultiTool-ohjelmistolla. Ohjelmisto tarjoaa työkalut, joilla laitteiden ja nii-den lähettämien signaalien konfigurointi onnistuu varsin yksinkertaisesti. Asetus-ten määrittelyn jälkeen MultiToolilla tehty koodi voitiin kääntää Codesys-koodiksi, jota itse laitteet käyttivät. Näin saatiin väyläviestit luotua automaattisesti Co-desys-koodiin. Tällä yksinkertaistettiin laitteiden ohjelmointia, Codesys-ohjelmis-tolla tarvitsi vain ohjelmoida toiminnot, joiden mukaan laitteita ja niiden signaaleja käytettiin.

MultiToolista löytyy valmiit omat konfiguraatiopaketit Epecin omille laitteille, muille laitteille on käytettävä geneerisiä konfiguraatiopaketteja, kuten esimerkiksi VC-VCCU:n kanssa tehtiin.

4.2 Codesys 3.5

Epec:in laitteiden toiminnallisuus ohjelmoitiin Codesys 3.5 ohjelmistolla. Codesys on IEC 61131-3 -standardin mukainen ohjelmointityökalu teollisuusautomaatioon (Codesys Group 2017).

IEC 61131-3 standardi määrittelee viisi yleisimmin käytettyä ohjelmointikieltä au-tomaatiolle:

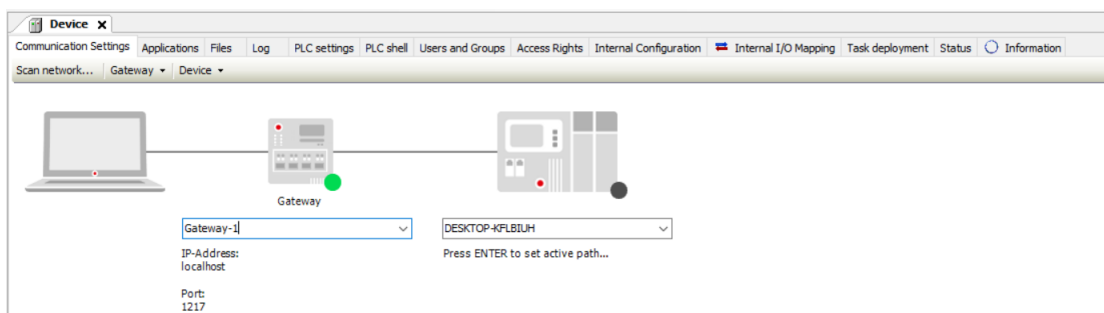
- tekstimuotoiset kielet
 - käskylista
 - rakenteinen teksti

- graafiset ohjelmointikielet
 - tikapuukaavio
 - toimilohkokaavio
 - sekvenssikaavio

(IEC 61131-3 2013, 9). Projektin Codesys-ohjelmointi toteutettiin rakenteisena tekstinä, eli Structured Text -muodossa.

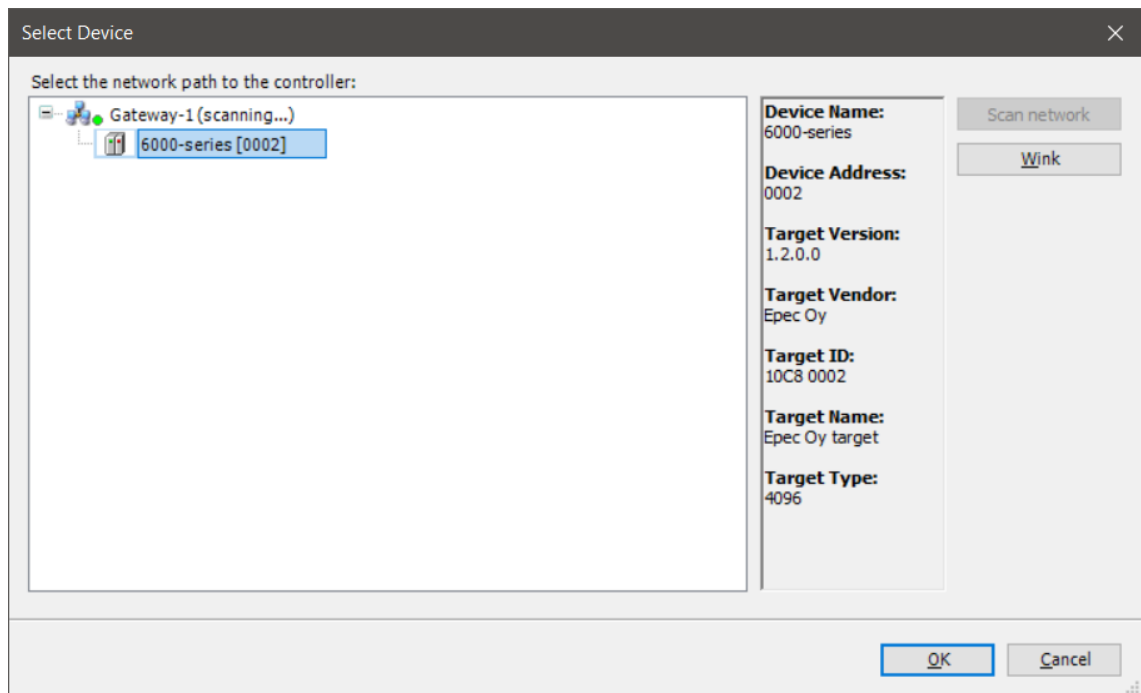
Codesys-ohjelmakoodin siirtäminen tietokoneelta laitteille vaati laitteiden yhdistämistä tietokoneelle. Tämä tapahtui liittämällä haluttu laite RJ45 ethernet-kaapelilla tietokoneeseen. Näytöltä saatiin kaapeli suoraan tietokoneelle, ohjausyksikölle liitettiin CAN-väyläadapterin ethernet-portin kautta. Fyysisen kytkennän lisäksi, oli muutettava liittyvän tietokoneen IP-osoitetta laitteiden mukaan. Näytölle IP-osoite oli 192.168.0.1 ja ohjausyksikölle 192.168.0.7. Maskina molemmissa 255.255.255.0, oletusyhdyskäytävää ei tarvinnut määritellä.

Kun laite oli kytketty, avattiin Codesysissä laitteen kommunikaatioasetukset, kuten kuvassa 8 on esitetty.



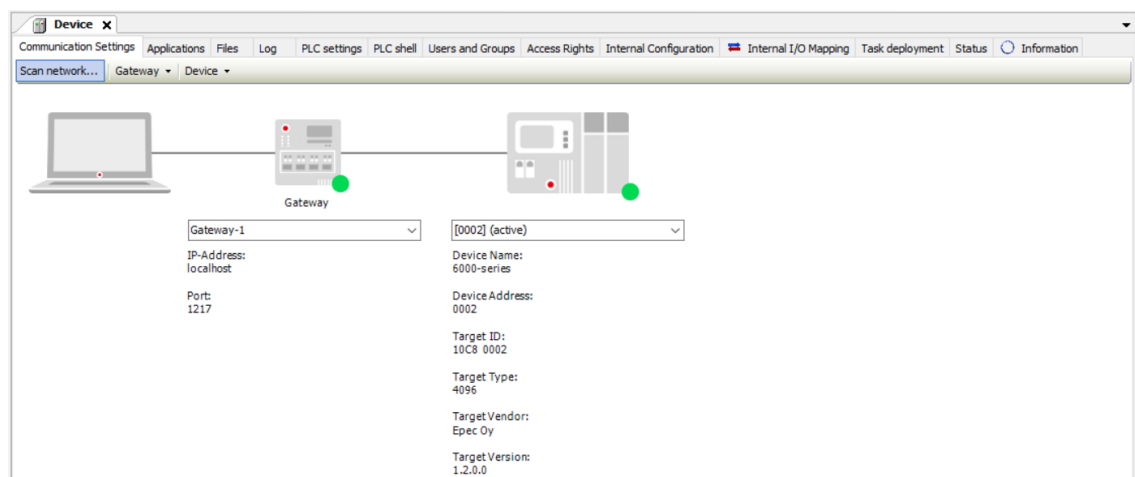
KUVA 8. Ei löydettyjä laitteita

Liitettyjä laitteita etsittiin "Scan network..."-painikkeella, jolloin aukesi kuvan 9 mukainen ikkuna, jossa näkyvät löydetyt laitteet, joihin voitiin yhdistää.



KUVA 9. Saatavilla olevien laitteiden etsiminen

Valittiin löydetty näyttö, jolloin sen määrittelyt kopioituivat kuvan 8 ikkunaan, kuten kuvassa 10 on esitetty.



KUVA 10. Valittu laite ja sen parametrit

Määrittelyn jälkeen voitiin yhdistää laitteelle työkalurivin tai valikkojen login-komennolla. Tämän jälkeen Codesys-koodiin määriteltäviä toimintoja voitiin testata ja signaalien ja muuttujien tiloja seurata ja muuttaa Codesysin sisällä.

4.3 Orion BMS 2 Control Application

Akunhallintajärjestelmän kommunikointi määriteltiin Orionin omalla ohjelmalla. Ohjelmalla määritettiin J1939-väylän siirtonopeus vastaamaan VCCU:n ja 5050-ohjausyksikön siirtonopeuksia. Ohjelmalla määriteltiin myös akunhallintayksikön lähettämät viestit, joita tarvittiin muilla laitteilla. Ohjelman toimintaa esitellään tarkemmin luvussa 5.2.4.

4.4 CANmoon

CANmoon-ohjelmistolla testattiin laitteiden välisiä CANOpen- ja J1939-väyläkommunikaatioita. Ohjelmistolla saatiin tarkasteltua väylässä tapahtuvaa viestiliikennettä. Väylään voitiin myös kirjoittaa viestejä yhteyden ja toimintojen testaamiseksi. Ohjelman toimintaa esitellään tarkemmin luvussa 7.2.

4.5 CANoe

CANoe-ohjelmistolla testattiin VCCU:n toimintoja. VCCU:lle oli toimittajalta valmis testisovellus, johon oli määritelty valmiiksi VCCU:n tarvitsema väyläliikenne. Testisovelluksessa oli myös kaksi valmista testinäyttöä, joista nähtiin VCCU:n signaalien tila. Ohjelman toimintaa esitellään tarkemmin luvussa 7.1.

5 JÄRJESTELMÄN SUUNNITTELU

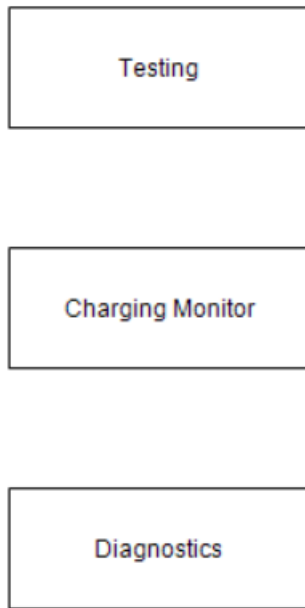
Järjestelmän suunnittelu koostui käyttöliittymäsuunnittelusta, kommunikaatio-suunnittelusta ja toiminnallisuuksien suunnittelusta. Käyttöliittymä eli näyttösuunnittelussa luonnosteltiin tarvittavat näyttöikkunat laitteiston testaamista ja käyttöä varten. Kommunikaatiosuunnittelussa suunniteltiin käytettävät väylät ja niissä kuljetettavat viestit. Toiminnallisuuteen kuuluu käytännössä kaikki muu, mikä ei suoranaisesti kuulu näyttöikkunoihin tai kommunikaatioon, esimerkiksi arvojen skaalaus ja oikeiden arvojen valitseminen eli väyläviesteille.

5.1 Näyttösuunnittelu

Näyttösuunnittelu suoritettiin Codesys-ohjelmistolla. Codesys-ohjelmistossa näyttöikkunat toteutetaan omina, varsinaisesta ohjelmakoodista erillisinä, objekteina. Näyttöelementeille valittiin varsinaisesta ohjelmakoodista muuttujat, joiden tiloihin näyttöelementit kiinnitettiin. Tämä tarkoitti, että esimerkiksi latauspistokkeelta tulevat lämpötilatiedot tuotiin väylästä näytön muuttujiin ja näitä muuttujia käytettiin näytön lämpötilamittarien mittaustietona.

5.1.1 Aloitusnäyttö

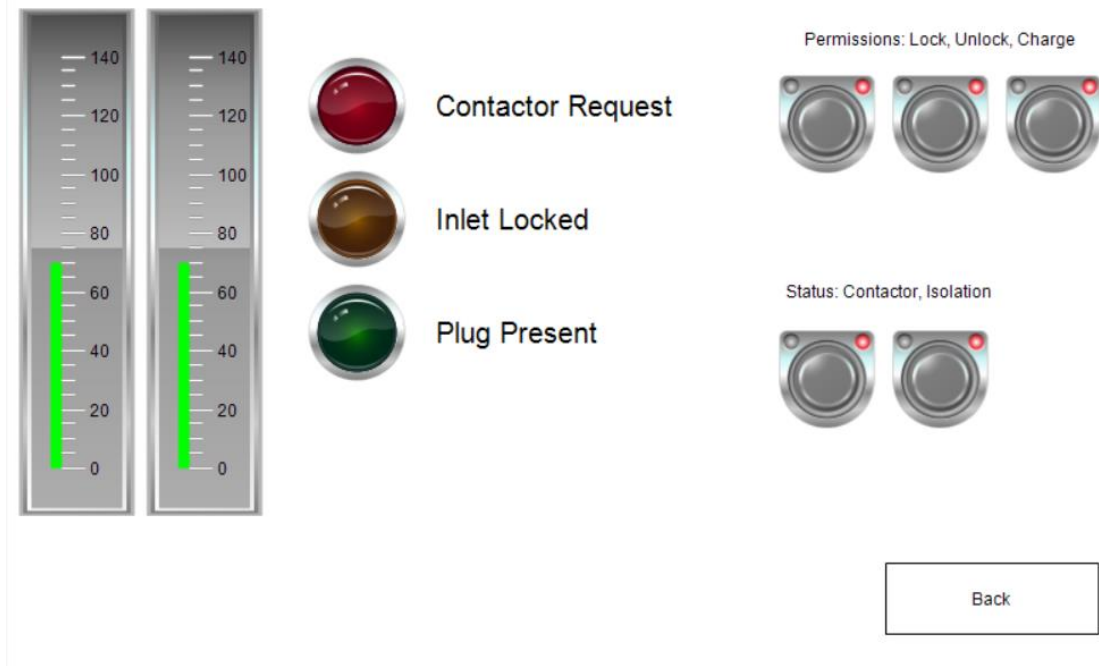
Näyttösuunnittelussa huomioitiin sekä laitteiden laboratoriotestaus että lopullinen latauksenhallinta. 6107-näytön käyttöliittymä jaettiin aloitusnäytön avulla kolmeen varsinaiseen näyttötilaan, joita käytettiin projektissa. Kuvassa 11 on esitetty aloitusnäytön sivunvaihtopainikkeet.



KUVA 11. Näyttöjen aloitusvalikko

5.1.2 Laboratoriotestinäyttö

Kuvan 11 ylin painike avaa VCCU:n laboratoriotestinäytön, jonka avulla voitiin ohjata ja testata VCCU:n toimintaa latauspistorasian kanssa. Tämä mahdollisti toimintojen testaamisen ilman jakeluauton muuta sovellusta. Laboratoriotestinäyttö on esitetty kuvassa 12.



KUVA 12. Laboratoriotestinäyttö

Näytön vasemmassa laidassa on AC- ja DC-latausliitinten lämpötilamittaukset, joilla voidaan varmistaa, että itse latauspistorasia ei kuumene liikaa.

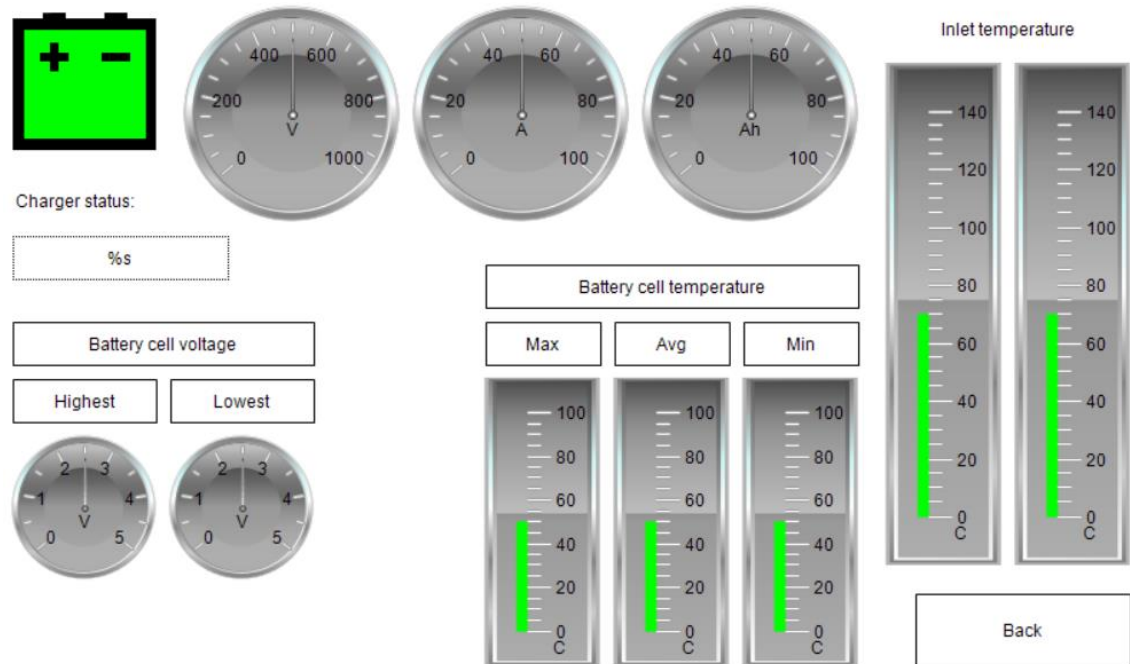
Lämpötilamittausten jälkeen on esitetty kolme tilatietoa. Alimpana on tieto latauspistoolin kytkeytymisestä latauspistorasiaan. Tämän yläpuolella on pistoolin lukitsemistieto. Ylimpänä on VCCU:n ohjelmasta tuleva pyynti kontaktorin ohjaamiselle, jolla lataaminen voidaan aloittaa.

Oikeassa laidassa on simulointipainikkeet laboratoriotestaukselle. Ylemmällä rivillä on kolme lupasignaalia, jotka oikeassa järjestelmässä tulevat muista soveluksista. Ensimmäisenä on lupatiedot latauspistoolin lukitukselle ja lukituksen poistamiselle. VCCU ei vahdi latausjännitteitä latauspistoolin lukitusta varten, vaan kyseinen toiminta on toteutettava itse. VCCU estää lukituksen poistamisen kesken latausohjelman, mutta ohjelman päätyttyä jännitteettömyyttä ei valvota (Bayer & Leserer 2018, 15). Kolmantena on lupasignaali latauksen aloittamiselle.

Alemmalla rivillä on vielä kaksi simulointipainiketta, joiden avulla voitiin simuloida kontaktorin ja eristysmittauksen tiloja. Näitä tarvittiin, kun testattiin toimintoja ilman oikeaa akunhallintajärjestelmää.

5.1.3 Latauksenhallintanäyttö

VCCU:n testinäytön lisäksi tehtiin näyttöluonnos varsinaista latausohjelmaa varten. Latauksenhallintanäyttö on avattavissa kuvan 11 toisesta painikkeesta (Charging monitor) painamalla. Näyttöluonnos on esitetty kuvassa 13.



KUVA 13. Näyttösuunnitelma akunhallintajärjestelmälle

Näytön vasemmassa yläkulmassa on akun tilaa kuvaava symboli, jonka pinnankorkeus ja väri muuttuu akun varauksen mukaan. Akkusymbolin oikealla puolella on kolme mittarinäyttöä, latauksen tilan seuraamista varten. Vasemmanpuolimmaisella esitetään akuston kokonaisjännite, keskimmaisella latausvirta ja oikeanpuolimmaisella akuston varaus ampeeritunteina.

Vasemmassa laidassa akkusymbolin alla esitetään laturin tila ja tämän alla akuston korkein ja matalin akkujännite. Vastaavasti keskellä alalaidassa on akuston lämpötilantarkkailun mittausarvot korkeimmasta ja matalimmasta lämpötilasta sekä kaikkien akkujen lämpötilan keskiarvo.

Viimeisenä oikeassa laidassa on vastaavat latauspistokkeen lämpötilamittaukset kuin VCCU:n laboratoriotestinäytössä kuvassa 12.

5.1.4 Ilmoituslista

Kolmas operointinäyttö on varattu ilmoituslistalle, joka aukeaa kuvan 11 aloitusnäytön alimmasta painikkeesta (Diagnostics). Kuvan 14 ilmoituslistalle kerätään akunhallintajärjestelmien varoitukset ja virheilmoitukset.

	Timestamp ▾	Message	Class
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Back

KUVA 14. Ilmoituslista

Ilmoituslistalle tuodaan tapahtuman aikaleima, tapahtuman nimi sekä tapahtuman tyyppi, joka on kaikissa tilanteissa virhetieto. Akunhallintajärjestelmän tilaviestien bitit lisättiin ilmoituslistan määrittelyyn kuten kuvassa 15.

ID	Observation type	Details	Deactivation	Class	Message
0	Digital	CANI_DTC_Status_1.0 = TRUE		Error	<DATE> <TIME> Discharge Limit Enforcement Fault
1	Digital	CANI_DTC_Status_1.1 = TRUE		Error	<DATE> <TIME> Charger Safety Relay Fault
2	Digital	CANI_DTC_Status_1.2 = TRUE		Error	<DATE> <TIME> Internal Hardware Fault
3	Digital	CANI_DTC_Status_1.3 = TRUE		Error	<DATE> <TIME> Internal Heatsink Thermistor Fault
4	Digital	CANI_DTC_Status_1.4 = TRUE		Error	<DATE> <TIME> Internal Software Fault
5	Digital	CANI_DTC_Status_1.5 = TRUE		Error	<DATE> <TIME> Highest Cell Voltage Too High Fault
6	Digital	CANI_DTC_Status_1.6 = TRUE		Error	<DATE> <TIME> Lowest Cell Voltage Too Low Fault
7	Digital	CANI_DTC_Status_1.7 = TRUE		Error	<DATE> <TIME> Pack Too Hot Fault
8	Digital	CANI_DTC_Status_2.0 = TRUE		Error	<DATE> <TIME> Internal Communication Fault
9	Digital	CANI_DTC_Status_2.1 = TRUE		Error	<DATE> <TIME> Cell Balancing Stuck Off Fault
10	Digital	CANI_DTC_Status_2.2 = TRUE		Error	<DATE> <TIME> Weak Cell Fault
11	Digital	CANI_DTC_Status_2.3 = TRUE		Error	<DATE> <TIME> Low Cell Voltage Fault
12	Digital	CANI_DTC_Status_2.4 = TRUE		Error	<DATE> <TIME> Open Wiring Fault
13	Digital	CANI_DTC_Status_2.5 = TRUE		Error	<DATE> <TIME> Current Sensor Fault
14	Digital	CANI_DTC_Status_2.6 = TRUE		Error	<DATE> <TIME> Highest Cell Voltage Over 5V Fault
15	Digital	CANI_DTC_Status_2.7 = TRUE		Error	<DATE> <TIME> Cell ASIC Fault
16	Digital	CANI_DTC_Status_2.8 = TRUE		Error	<DATE> <TIME> Weak Pack Fault
17	Digital	CANI_DTC_Status_2.9 = TRUE		Error	<DATE> <TIME> Fan Monitor Fault
18	Digital	CANI_DTC_Status_2.10 = TRUE		Error	<DATE> <TIME> Thermistor Fault
19	Digital	CANI_DTC_Status_2.11 = TRUE		Error	<DATE> <TIME> External Communication Fault
20	Digital	CANI_DTC_Status_2.12 = TRUE		Error	<DATE> <TIME> Redundant Power Supply Fault
21	Digital	CANI_DTC_Status_2.13 = TRUE		Error	<DATE> <TIME> High Voltage Isolation Supply
22	Digital	CANI_DTC_Status_2.14 = TRUE		Error	<DATE> <TIME> Input Power Supply Fault
23	Digital	CANI_DTC_Status_2.15 = TRUE		Error	<DATE> <TIME> Charge Limit Enforcement Fault

KUVA 15. Akunhallintajärjestelmän varoitukset ja hälytykset

Details-riville määriteltiin ehto, milloin hälytys tehdään, tässä tapauksessa luettiin hälytysviestin yksittäisiä bittejä. Mikäli ehto täyttyy, annetaan Message-rivin mukainen viesti kuvan 14 listalle.

5.1.5 Testaus

Näyttöjen testaus oli varsin yksinkertaista, kun laitteet oli kytketty. Tiedot näyttöelementeille saatiin joko oikeilta laitteilta tai kirjoittamalla testisignaaleja väylään CANmoon-ohjelmalla. Testausta käsitellään tarkemmin luvussa 7 ja Codesys-näyttöjen testausta luvussa 7.3.

5.2 Väyläkommunikaation sovellussuunnittelu

Väyläkommunikaatiota konfiguroitiin pääasiassa laitteiden valmistajien omilla ohjelmistoilla. Tavoitteena oli saada kaikki tarvittavat viestit kulkemaan laitteiden välillä.

5.2.1 Väylärakenne

Projektin määrittely MultiToolissa aloitettiin Network Editor -ikkunassa. Valikoista haettiin konfiguraatiopaketit projektin eri laitteille. Epecin omille laitteille löytyi tarkemmin konfiguroitavat paketit, joiden parametointi voitiin kääntää Codesys-koodiksi, mikä helpotti konfigurointia.

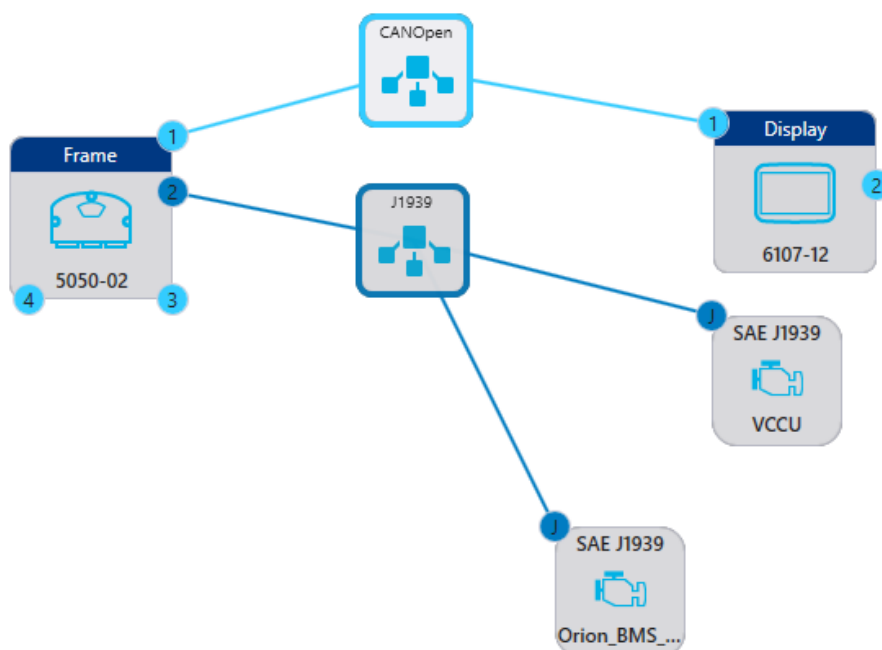
Valmistaja oli ilmoittanut VCCU:n kommunikaatiotavaksi J1939-väylän, joten VCCU toteutettiin MultiToolissa yleisenä SAE J1939 -laitteena. SAE J1939 -väylä on raskaille ajoneuvoille tarkoitettu CAN-väylään perustuva väylä (Kvaser 2019).

VCCU:n ja ohjausyksikön välille määriteltiin J1939-väylä. Väyläkonfiguraatio oli suoritettava ohjausyksikön määrittelyihin ja asetettava VCCU:n valmiiden parametrien mukaiseksi.

Akunhallintajärjestelmä Orion BMS 2 toteutettiin VCCU:n tavoin SAE J1939 -laitteena ja se kytkettiin samaan J1939-väylään 5050-ohjausyksikön ja VCCU:n kanssa.

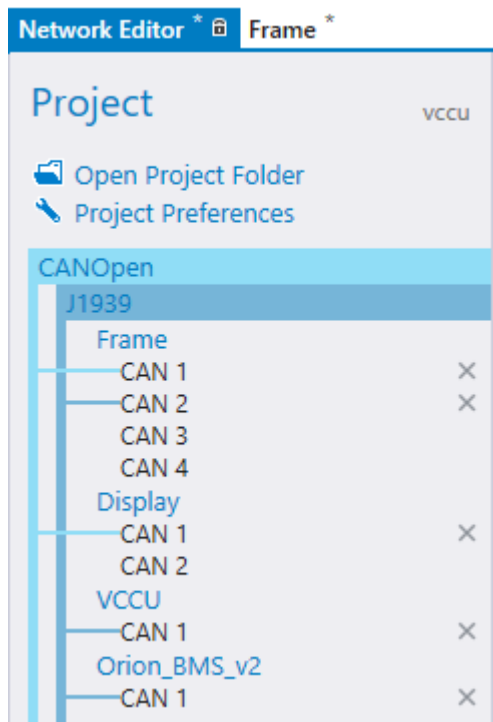
Kommunikaatio ohjausyksiköltä näytölle toteutettiin CANOpen-väylänä. CANOpen-väylän käyttäminen on yksinkertaisempaa kuin J1939-väylän käyttäminen MultiTool-ympäristössä.

Kuvassa 16 on esitetty lopullinen väyläkommunikaation rakenne laitteiden välillä.



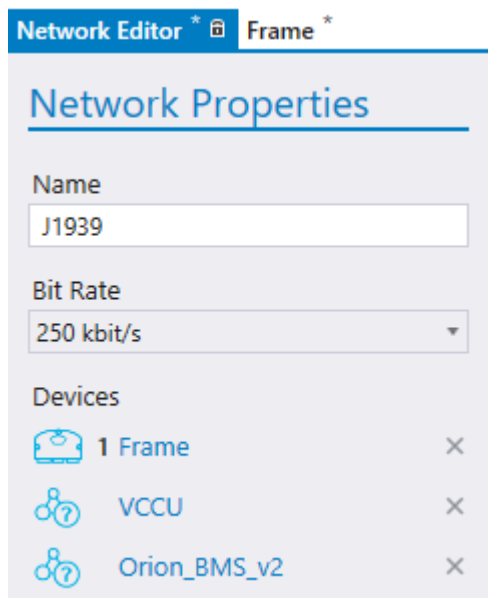
KUVA 16. Lopullinen väylätopologia MultiTool-projektissa

Projektin väylärakenne päivittyi myös kuvan 17 mukaiseen listaan.



KUVA 17. Laitteiden liitynnät väyliin MultiTool-projektissa

Valitsemalla kuvasta 16 väylämoduuli J1939, voitiin asettaa sen määrittäykset kuten kuvassa 18 on esitetty.



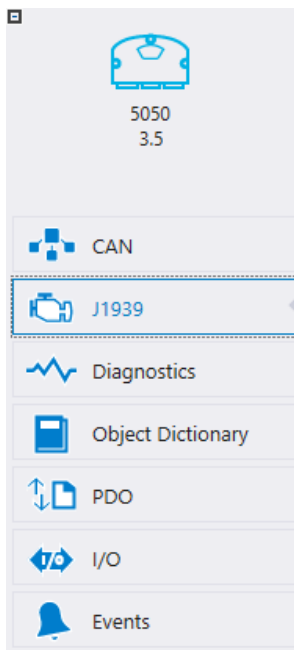
KUVA 18. J1939-väylän asetukset Multitoolissa

Väylän siirtonopeus asetettiin 250 kbits/s arvoon, joka oli VCCU:n toimittajan määrittelemä arvo. 5050-ohjausyksikkö ja BMS asetettiin samalle nopeudelle

VCCU:n kanssa, koska väylä ei toimi, mikäli sinne lähetetään erinopeuksisia viestejä samanaikaisesti.

5.2.2 6107-12-näytön väyläkonfigurointi

Kosketusnäytön ja 5050-ohjausyksikön välinen CANOpen-kommunikaatio määriteltiin MultiToolin CAN- ja PDO (PHP Data Object) -valikoista, jotka on esitetty kuvassa 19.



KUVA 19. Ohjausyksikön päävalikko MultiToolissa

Jokaiselle käytössä olleelle CAN-väylälle voitiin määrittellä omat parametrinsa. Kuvassa 20 on esitetty näytön ja ohjausyksikön välisen väylän määrittely.

The screenshot shows a software interface for configuring CAN bus parameters. At the top, there are four tabs labeled CAN 1, CAN 2, CAN 3, and CAN 4. The 'CAN 1' tab is selected. Below the tabs, the 'Properties' section includes a 'Send And Receive' section with checkboxes for 'Short ID Messages' and 'Extended ID Messages', both of which are currently unchecked. Below this is a 'Bit Rate' dropdown menu set to '250 kbit/s'. The 'Protocols' section is titled 'CANOpen' and contains a list of configuration parameters:

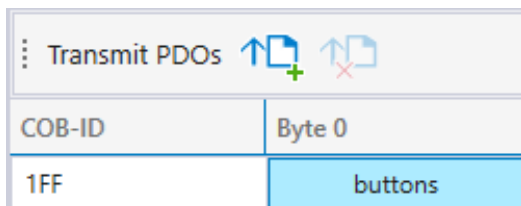
- CANOpen**: Initialize
- Application Node-ID**: 126
- Node-ID Offset for PDOs**: 32
- Heartbeat Interval**: 200 ms
- Device Profile**: 405
- Application Version String**: (empty field)
- NMT Protocol**: Slave, Master
- Monitoring Start Interval**: 1 000 ms
- Network Status**: Is a Condition for the G_SystemOk Variable
- Simultaneous Slave Configuration**: Enable

KUVA 20. CANOpen-väylän määrittely 5050-ohjausyksikölle

Ohjausyksikkö asetettiin Master-statukselle kaikkiin kommunikaatioihin ja väylien toiminnan tarkkailu asetettiin vaatimukseksi järjestelmän ok-tiedolle. Muille parametreille ei ollut laitteiden puolesta vaatimuksia, vaan ne valittiin itse.

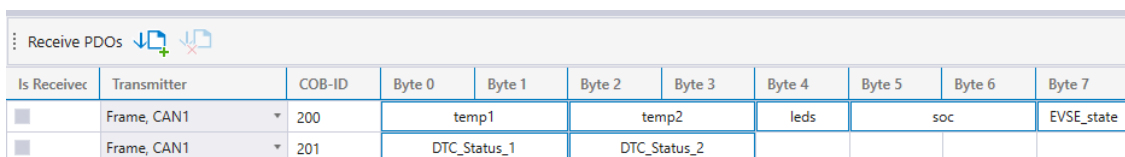
Vastaavat parametrit asetettiin myös näytön CANOpen-määrittelyihin, mutta näytölle asetettiin Slave-status, jotta ohjausyksikkö sai johtaa väylässä käytävää kommunikaatiota.

Kun CANOpen-väylän määrittely oli valmis, voitiin määrittellä väylässä käytettävät signaalit. Signaalien määrittely tehtiin näytön PDO-valikosta. Lähetettävät ja vastaanotettavat signaalit määriteltiin omille alueilleen. Näytölle määritellyt viestit on esitetty kuvissa 21 ja 22.



COB-ID	Byte 0
1FF	buttons

KUVA 21. Näytön lähettämät viestit



Is Receiver	Transmitter	COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<input type="checkbox"/>	Frame, CAN1	200	temp1	temp2	leds	soc	EVSE_state			
<input type="checkbox"/>	Frame, CAN1	201	DTC_Status_1	DTC_Status_2						

KUVA 22. Näytön vastaanottamat viestit

Valitsemalla jokin viesti, avautuu määrittelyikkuna, jossa asetettiin viestin nimi, tietotyyppi, kuvaus ja binääritietojen kanssa bittikohtaiset signaalit. Kuvassa 23 on esitetty led-ohjausten määrittely.

The image shows two configuration windows. The top window, titled "RPDO Properties", has a "COB-ID (hex)" field with the value "200" and a "DLC" field with a value of "8" and a small up/down arrow. Below the DLC field is a link labeled "Advanced...". The bottom window, titled "Variable Properties", has a "Name" field with "leds", a "DataType" dropdown menu set to "BYTE", and a "Description" field with "LED signals". Below these is a "Bits" section with a checked checkbox. Underneath, there are eight input fields for bit names, indexed from 0 to 7. Bits 0, 1, and 2 are named "led1", "led2", and "led3" respectively. Bits 3 through 7 are named "Bit3", "Bit4", "Bit5", "Bit6", and "Bit7".

Bit Index	Bit Name
0	led1
1	led2
2	led3
3	Bit3
4	Bit4
5	Bit5
6	Bit6
7	Bit7

KUVA 23. Vastaanotettavan PDO-viestin määrittely CANOpen-väylään

Viestin bitit 0-2 on määritelty ledeille, mutta loput on jätetty uudelleennimeämättä.

Samalla tavoin määriteltiin muut näytölle tuotavat signaalit. Määritellyt signaalit muodostivat automaattisesti laitteen muuttujalistan kuten kuvassa 24 on esitetty.

Variable	Size	
buttons	1	Remove
leds	1	Remove
EVSE_state	1	Remove
BMS1_warnings1	1	Remove
BMS1_warnings2	1	Remove
BMS1_alarms1	1	Remove
BMS1_alarms2	1	Remove
BMS2_warnings1	1	Remove
BMS2_warnings2	1	Remove
BMS2_alarms1	1	Remove
BMS2_alarms2	1	Remove
soc	2	Remove
temp1	2	Remove
temp2	2	Remove
DTC_Status_1	2	Remove
DTC_Status_2	2	Remove

KUVA 24. Näytön muuttujalista

Määritellyt viestit ilmestyvät kuvan 24 muuttujalistalle riippumatta siitä, onko niitä valittu lähetettäväksi tai luettavaksi vai ei. Viestien sininen tausta osoittaa, mitkä viesteistä on valittu joko lähetettäväksi tai luettavaksi. Listalla näkyvät valkoisella taustalla olevat viestit ovat vanhan akunhallintajärjestelmän viestejä, jotka jäivät pois lopullisesta toteutuksesta simuloitujen testausten jälkeen.

5.2.3 VC-VCCU

Sovellussuunnittelun kannalta VCCU oli kenties mutkikkain laite. Vaikka VCCU:n viesteistä ja signaaleista oli olemassa valmiit taulukot, niiden määrittely MultiTooliin osoittautui työlääksi, osittain signaalien määrän takia. MultiToolista löytyi omat valikkonsa J1939-kommunikaation määrittelyyn, kuten esitettiin aiemmassa kuvassa 19.


Kuten CANOpen-viestien PDO:iden kanssa, myös J1939-viestit määriteltiin omille alueilleen riippuen siitä, olivatko ne lähetettäviä vai vastaanotettavia viestejä. Kuvassa 25 on esitetty VCCU:n vastaanottama VCVCCU Requests -viestipaketti, joka koostuu sen alla esitetystä signaaleista.

Short Name / PG	COB-ID	Name	Start Bit	Length	Data Type
RQSTS : 0xFF20	0x18FF2082	VCVCCU_Requests		55	
		VCVCCU_LED0_Request	0	8	BYTE
		VCVCCU_LED1_Request	8	8	BYTE
		VCVCCU_LED2_Request	16	8	BYTE
		VCVCCU_HighSideOut0_Request	24	2	BYTE
		VCVCCU_HighSideOut1_Request	26	2	BYTE
		VCVCCU_HighSideOut2_Request	28	2	BYTE
		VCVCCU_HighSideOut3_Request	30	2	BYTE
		VCVCCU_HighSideOut4_Request	32	2	BYTE
		VCVCCU_RTC_TimerRequest	34	2	BYTE
		VCVCCU_PlugLock_MotorRequest	36	3	BYTE
		VCVCCU_RTC_TimerValue	40	8	BYTE
		VCVCCU_ControlPilot_ChargeMode	48	3	BYTE
		VCVCCU_ChargeUnit_Request	51	2	BYTE
		VCVCCU_Inlet_MotorRequest	53	2	BYTE

KUVA 25. VC-VCCU Requests -viestipaketti

Kyseinen viesti määriteltiin MultiToolissa valitsemalla rivi, jolloin aukeaa kuvan 26 mukainen määrittelyikkuna.

J1939 Message Properties

Name	VCVCCU_Requests
Short Name	RQSTS
Cycle Time (ms)	250
COB-ID	18FF2082h
Priority	6
Data Page	<input type="checkbox"/>
PDU Format (hex)	FF
PDU Specific (hex)	20
Destination	Broadcast (255)
Source	130
Variable alignment	

KUVA 26. VC-VCCU Requests -viestipaketin määrittelyosio

Viestin pitkä nimi (Name) luettiin ja kopioitiin valmistajan taulukosta, lyhyt nimi (Short Name) valittiin itse. Sykلياikaksi (Cycle Time) saatiin taulukoista arvo 250 millisekuntia. Myös prioriteetti (Priority) saatiin suoraan taulukoista. Kohteeksi (Destination) määriteltiin Broadcast (255), jotta viesti on väylän kaikkien laitteiden luettavissa.

COB-ID:n täsmäys taulukoiden ID:hen koostui useammasta parametrasta. Vectorin taulukot kertoivat Requests-viestin ID:n olevan 0x18FF2082 ja tämä arvo haettiin määrittellä MultiTooliin (Vector Group 2018, VC-VCCU CAN Messages). Arvo 18 tuli prioriteetin arvosta 6, MultiTool osaa itse valita protokollan mukaisen arvon prioriteetin perusteella. FF20 määriteltiin PDU-formaattina (PDU Format) ja täsmäteenä (PDU Specific), ja niiden arvot annettiin heksalukuina, kuten alkuperäisissä taulukoissakin. Viimeinen arvo 82 määriteltiin viestin lähteen (Source) avulla. Lähteeksi asetettiin heksalukua 82 vastaava kymmenjärjestelmän luku 130.

Kun Requests- viesti oli määritelty MultiTooliin, lisättiin sen sisälle tarvittavat signaalit. Signaalien nimi, aloitusbitti, pituus ja tyyppi luettiin ja kopioitiin valmistajan taulukosta (Vector Group 2018, VC-VCCU CAN Signals). Signaalien määrittelyt on esitetty kuvassa 25 Requests- viestin alla.

5.2.4 Akunhallintajärjestelmä

Акунhallintajärjestelmä toteutettiin MultiToolissa yleisenä SAE J1939 -laitteena samoin kuin VCCU.

Lähetettävät viestit määriteltiin Orionin BMS 2 Control Application -ohjelman avulla. Ohjelmalla määriteltiin käytettävät viestit kuvan 27 mukaisesta ikkunasta.

Enabled	ID	Length	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
<input checked="" type="checkbox"/>	0x1C00FFFE	8	Pack Current	IN USE	Pack Inst. Vol...	IN USE	Pack SOC	Relay State	IN USE	CRC Checksum
<input checked="" type="checkbox"/>	0x1C01FFFE	8	Pack DCL	IN USE	Pack CCL	IN USE	High Tempera...	Low Tempera...	Average Tem...	CRC Checksum
<input checked="" type="checkbox"/>	0x1C02FFFE	8	Pack Open Vo...	IN USE	Maximum Pac...	IN USE	Pack CCL KW	IN USE	Pack Amphours	IN USE
<input checked="" type="checkbox"/>	0x1C03FFFE	8	DTC Flags #1	IN USE	DTC Flags #2	IN USE	Low Cell Voltage	IN USE	High Cell Volt...	IN USE
<input type="checkbox"/>	0x000	0	Blank							
<input type="checkbox"/>	0x000	0	Blank							
<input type="checkbox"/>	0x000	0	Blank							
<input type="checkbox"/>	0x000	0	Blank							

Message Settings

Speed (ms): Receive/Transmit: CANBUS Interface: Extended ID: Pass-Thru Mesg: Keep-Alive Mesg:

Is-Charging MPI1 Active MPI2 Active MPI3 Active

Field Settings

Field Length (Bytes): Bit Order (First): Byte Order: Zero While Charging: Maximum Value: Minimum Value:

Multiply Value By: Then Divide By: Then Add: Signed Value:

KUVA 27. Orion BMS 2 Control Application -ohjelman CANBUS messages -konfigurointi-ikkuna

Orionin ohjeista löytyi listattuna mahdolliset käytettävät viestit, niiden selitykset, viestin pituudet ja skaalaukset. Jokaiselle viestille oli määriteltävissä, onko kyseessä lähetettävä vai vastaanotettava viesti ja kumpaa kahdesta BMS:n CAN-väylästä käytetään. Myös viestin ID:lle piti valita käytettäväksi pitkä nimi, joita käytetään J1939-kommunikaatiossa.

Viestin pituus oli asetettava ohjeiden mukaiseksi, riippuen onko viesti yhden vai kahden tavun pituinen. Mikäli kyseessä on kahden tavun pituinen tai pidempi viesti, on määriteltävä, kummin päin tavuja on tarkoitus tulkita: Vaihtoehtoina oli

Big Endian ja Little Endian, joista valittiin Little Endian, joka oli vakiona käytössä muilla käytetyillä J1939-väylän laitteilla (GeeksforGeeks 2019).

Viimeisenä voitiin vielä vaikuttaa viestien skaalaukseen, mitä ei tehty Orionin ohjelmistolla, vaan skaalat pidettiin vakioina ja korjaus tehtiin 5050-ohjausyksikön tai kosketusnäytön Codesys-koodiin. Skaalaaminen haluttiin suorittaa mahdollisimman myöhään, koska desimaalilukujen kanssa tarvitsisi käyttää tietomuotona reaalitylukuja, jotka vievät väylässä enemmän tilaa. Jättämällä skaalaaminen viimeiselle laitteelle, voidaan arvoa siirtää kokonaislukuna. Mikäli arvot skaalattaisiin ensin ja siirrettäisiin sitten kokonaislukuna, mittaustarkkuus heikkenisi, kun muodostuvat desimaalit tippuisivat kokonaisluvuista pois.

Akunhallintajärjestelmältä tulevat viestit määriteltiin 5050-ohjausyksikön vastaanotettaviin viesteihin, samoin kuin VCCU:n lähettämät oli määritelty.

5.3 Toiminnallisuuden sovellussuunnittelu

5.3.1 VCCU:n toiminnallisuus

VCCU:n varsinainen ohjelmakoodi oli toimitettu valmiina, eikä sitä lähdetty muuttamaan. VCCU:n valmiina toimitettu ohjelma sisälsi latauksenhallinnan ja lataussekvenssin.

DC-latauksen lisäksi VCCU hallitsi myös AC-latauksen, joten vanha AC-latauksenhallinta voitaisiin jättää pois alkuperäisistä suunnitelmista poiketen. Alun perin oli tarkoitus tuoda VCCU vanhan järjestelmän rinnalle, mutta havaittiin, että VCCU:lla voidaan korvata vanha latauksenhallintajärjestelmä kokonaisuudessaan.

5.3.2 6107-12-näytön toiminnallisuus

Näytölle ei tehty varsinaista ohjelmaa, poislukien näyttösivujen vaihdot, elementtien toiminnallisuus ja vikalista. Kaikki loogiset toiminnot tapahtuvat joko ohjausyksiköllä tai VCCU:lla. Näyttö toimii vain rajapintana muiden laitteiden ja operaattorin välillä. Väyläsignaalien konfigurointi saatiin käännettynä MultiToolilla tehdyistä määrittelyistä, joten niitä ei tarvinnut kirjoittaa itse uudestaan. Näytön Codesys-ohjelmaan lisättiin vain skaalauksia tiedoille, joita tuotiin näyttöelementeille.

5.3.3 5050-02-ohjausyksikön toiminnallisuus

Näytön tapaan väyläkommunikaation parametrit saatiin käännettyä MultiToolista. Koska näytön ja VCCU:n kommunikaatiot tehtiin eri väylillä, niiden väliseen kommunikaatioon piti määrittellä Codesys-ohjelmalla, mikä VCCU:lta tuotu muuttuja viedään mihinkin näytölle vietävään muuttujaan ja päinvastoin. Lisäksi VCCU:lle vietäviin signaaleihin alustettiin valmistajan taulukoiden mukaiset alkuarvot (Vector Group 2018, VC-VCCU CAN Signals).

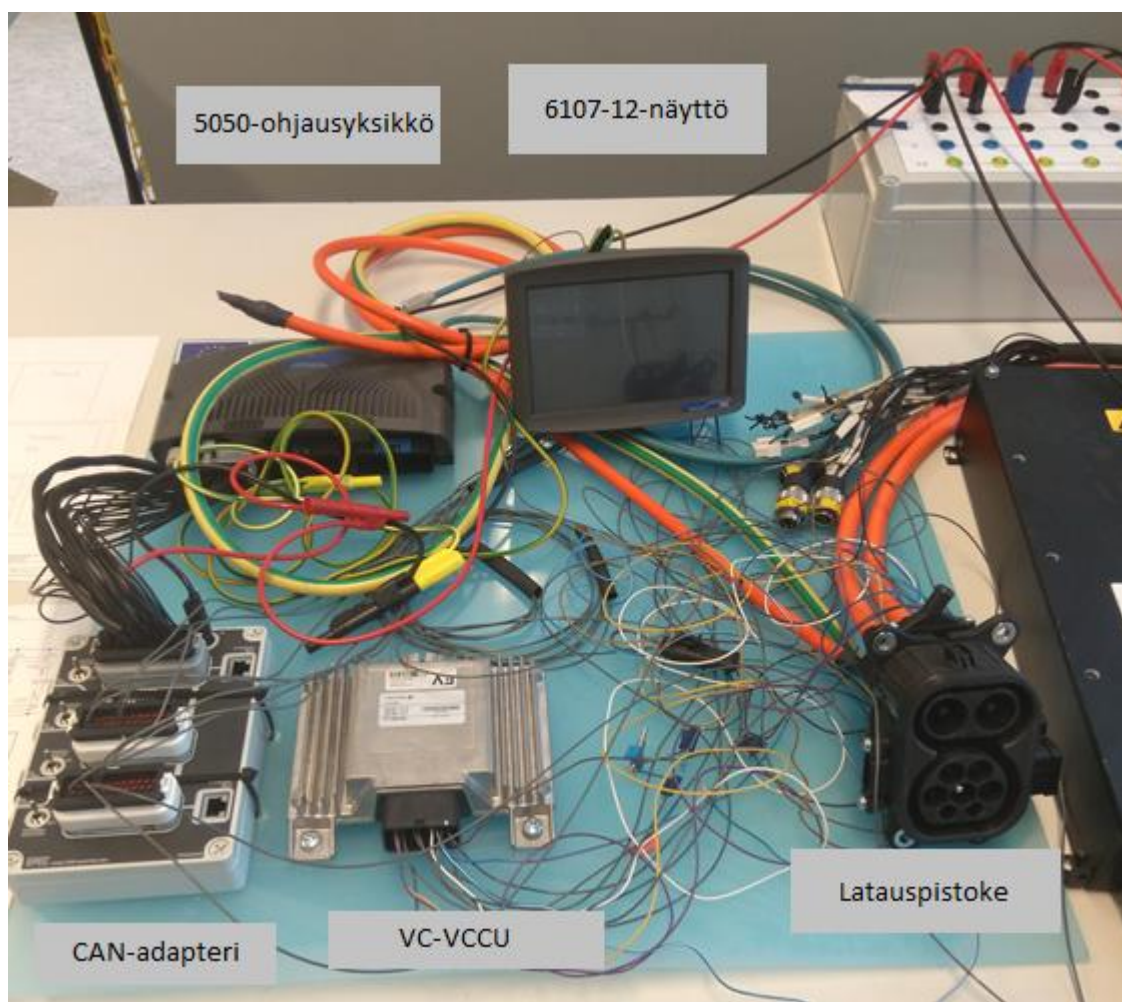
Myös akunhallintajärjestelmältä tulevat viestit skaalattiin ja tuotiin omiksi muuttujikseen 5050:n Codesys-koodissa. Akunhallintajärjestelmän ja VCCU:n välistä kommunikaatiota ei tehty suoraan, vaikka laitteet olivat samassa väylässä, vaan viestit kulkevat 5050:n kautta, joka varmistaa, että oikeat viestit viedään oikeassa muodossa oikeaan paikkaan.

6 TOTEUTUS

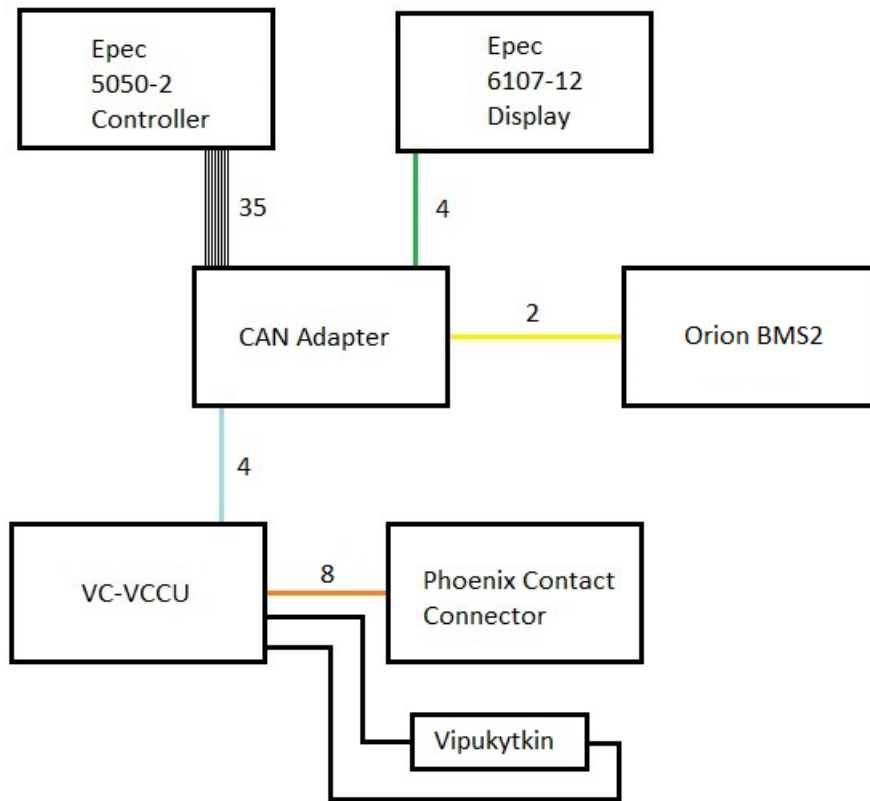
Kun suunnittelu oli valmista ja laitteiden toiminta ymmärretty, suoritettiin laitteiden lopullinen kytkentä laborioritestausta varten. Laitteet kiinnitettiin alustalevyyn, jotta laitteiston siirtely olisi mahdollista.

6.1 Kokonaiskytkentä

Kokonaiskytkentä on esitetty kuvassa 28 ja sitä selkeytetään kuviossa 3, jossa on esitetty laitteiden kytkennät toisiinsa.

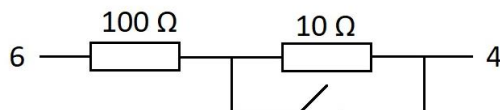


KUVA 28. Laborioritestaustilasto (Reko 2019)



KUVIO 3. Laborioritetauksen kytkentäkaavio

Kuvioon on merkitty laitteiden väliset johdinmäärät, signaalien tarkemmat selitykset on esitetty liitteessä 1. Laitekytkentöjen signaalit. Lukon asennontarkkailu toteutettiin erillisellä kuvion 4 mukaisella vipukytkimellä latauspistokkeen oman tarkkailun sijasta, koska lukolta saatava signaali ei ollut yhteensopiva VCCU:n kanssa. Tällöin saatiin simuloitua lataustilannetta laboratoriossa. Oikeassa järjestelmässä kyseiset kytkennät tehtäisiin latauspistokkeen asentotiedon plus- ja miinusjohtimiin.



KUVIO 4. Latauspistokkeen lukon tilatiedon vipukytkin

Johdinnumeroinnit 6 ja 4 ovat samat kuin liitteessä 1. Laitekytkentöjen signaalit.

Liitteen 1 VCCU:n johtimista nähdään myös, että SS_GND-maadoitusnastaan tuodaan kolme johdinta, ja että 5050-ohjausyksikkö, VCCU ja BMS on kytketty samaan CAN-väylään.

Kaaviota voidaan verrata kuvion 1 teoreettiseen kytkentään ja havaitaan pääperiaatteiden yhdenmukaisuus: Latauspistokkeelta tuodaan signaalit VCCU:lle, joka on J1939-väyläyhteydessä 5050-ohjausyksikön kanssa CAN-väyläadapterin kautta.

6.2 5050-01-ohjausyksikön kytkentä

Kuvassa 29 on esitetty 5050-ohjausyksikkö, sen ensimmäisistä AMP 35 -liittimistä tuodaan johtimet CAN-adapterille.



KUVA 29. 5050-02-ohjausyksikön AMP 35 -liityntä (Reko 2019)

6.3 CAN-adapterin kytkennät

CAN-adapterille tuodaan 5050-ohjausyksikön ensimmäisen AMP 35 -liittimen kaikki johtimet. Adapterilla näihin kytketään muilta laitteilta tuotavat signaalit. AMP 35 -liitynnät CAN-adapterille on esitetty kuvassa 30.



KUVA 30. CAN-adapterin AMP 35 -liittynät (Reko 2019)

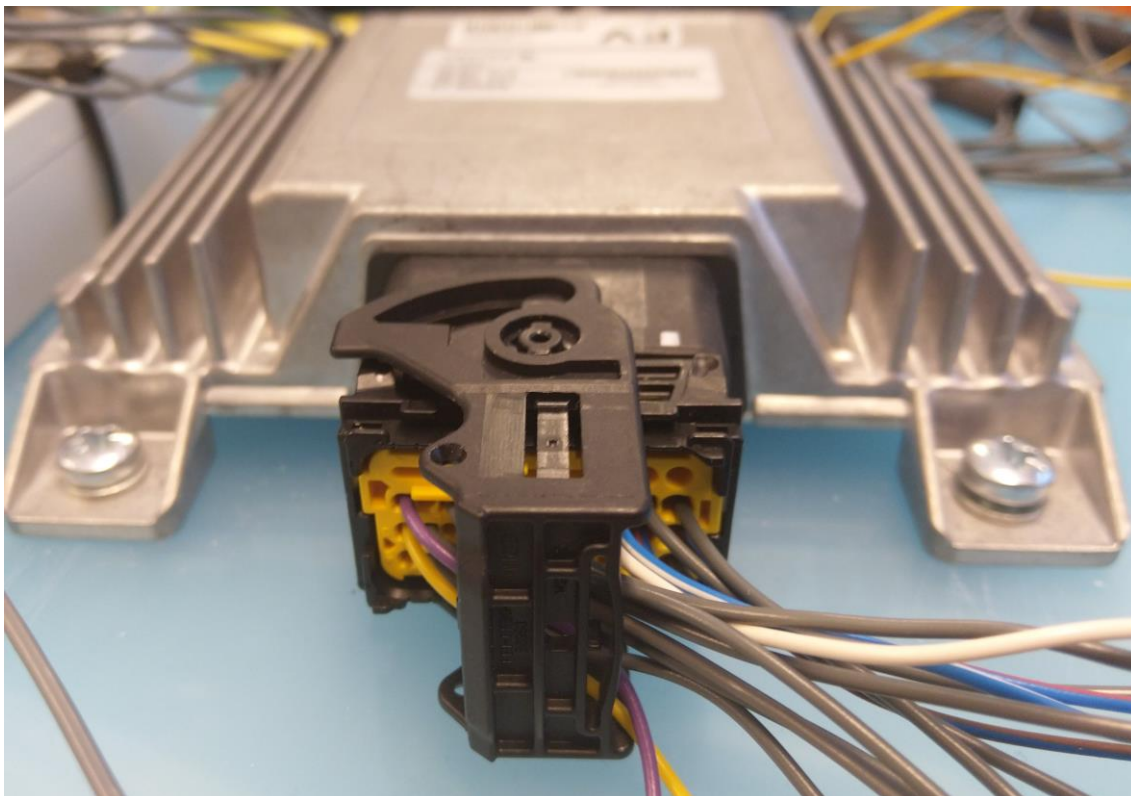
5050-ohjausyksiköltä tuotavia signaaleja saadaan käytettyä useammalle laitteelle, esimerkiksi syöttöjännitettä ja maadoitusnastaa käytetään VC-VCCU:lle ja 6107-näytölle. Eri laitteiden CAN-väyläkommunikaatiot käydään kuitenkin omilla kanavillaan, koska VC-VCCU:n kommunikaatio on J1939-kommunikaatiota ja

5050-ohjausyksikön ja 6107-näytön välinen kommunikaatio käydään CANOpen-kommunikaationa.

Kuvasta 30 havaitaan myös, että CAN-adapterilta on lisäksi kytketty CAN-väylien 120 Ω päätevastukset käyttöön. Adapterin oikeassa laidassa on myös havaittavissa RJ45-liittimet ethernet-liityntää varten, joita käytettiin sovelluksen siirrossa ohjausyksikölle.

6.4 VC-VCCU:n kytkennät

VCCU:lle tulevat johtimet tuotiin Molexin CMC 36 -liittimellä, kuten kuvassa 31 on esitetty.



KUVA 31. VC-VCCU:n CMC 36 -liityntä (Reko 2019)

6.5 6107-12-kosketusnäytön kytkennät

Ohjausyksiköltä tuotavat signaalit tuotiin näytölle AMP 23 -liittimellä. Ethernet-liityntä sovelluksen siirtoa varten tuotiin M12-liittimeen. Molemmat liittimet on esitetty kuvassa 32.



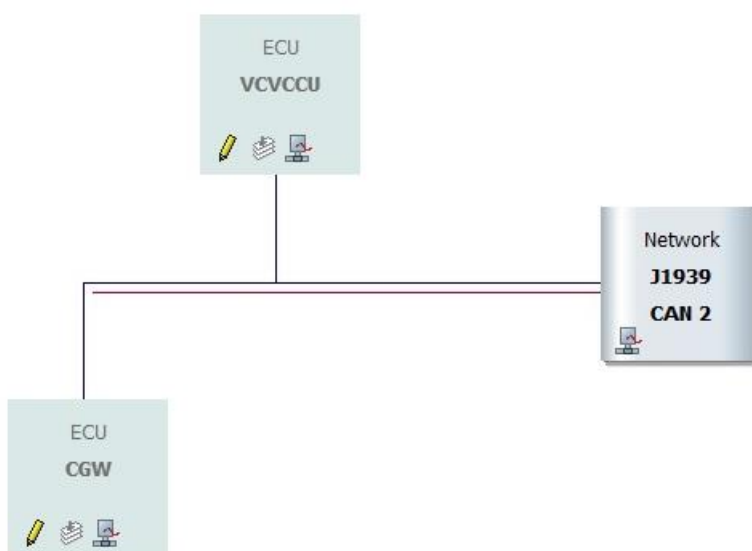
KUVA 32. Kosketusnäytön liitynnät AMP 23 ja M12 -liittimillä (Reko 2019)

7 SOVELLUKSEN TESTAUS SIMULOIMALLA

Sovellusta testattiin käyttäen samanaikaisesti CANoe-, Codesys-, ja CANmoon-ohjelmistoja. Testaus suoritettiin simuloimalla vanhaa akunhallintajärjestelmää, Orionin BMS 2 -akunhallintajärjestelmä otettiin käyttöön vasta lopulliseen kytkentään.

7.1 CANoe

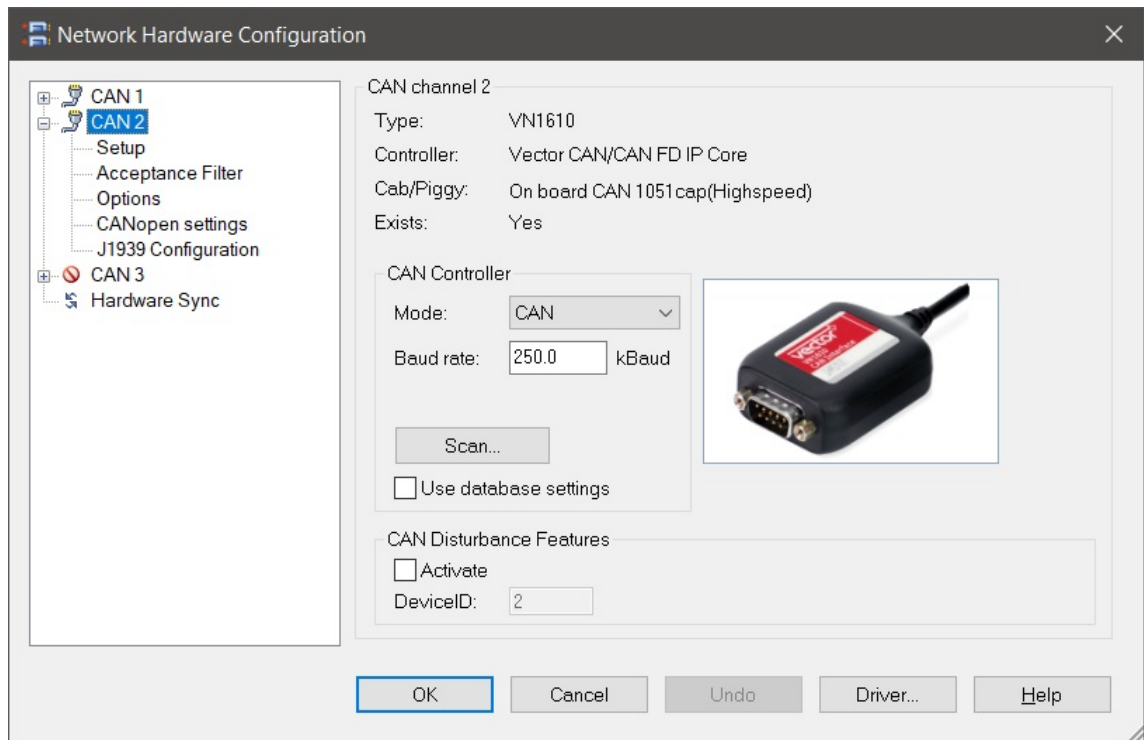
CANoe-testiympäristön käyttöönotto vaati hieman konfigurointia. Kuvassa 33 on esitetty CANoe-testisovelluksen toiminnallisen väylän rakenne.



KUVA 33. VCCU-testisovelluksen väylärakenne CANoe-ohjelmistossa

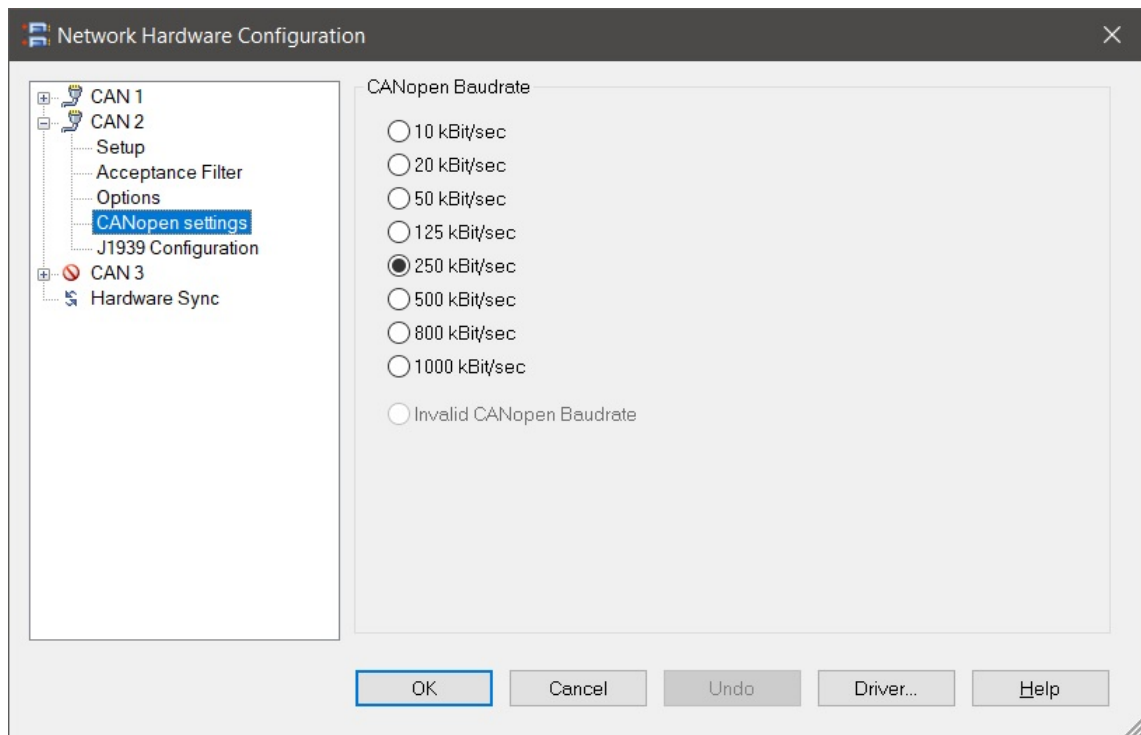
Testisovellusta on mahdollista käyttää fyysisten laitteiden kanssa, tai simuloiden niitä. Kuvan 33 tilanteessa sekä VCCU että ohjausyksikkö on valittu oikeiksi laitteiksi, mikä käy ilmi niiden väristä. Laitteiden välinen J1939-väylä sen sijaan on valittu tuotavaksi CANoe:n kautta, jotta siellä liikkuvaa liikennettä voidaan tarkkailla ja ohjata.

Ennen kuin testaaminen voidaan aloittaa, piti konfiguroida väylätesteri toimimaan testiovelluksen kanssa. Käytännössä konfigurointi vaatii Baud rate -arvon eli siirtonopeuden asettamista väyläliikenteeseen sopivaksi, eli VCCU:n tapauksessa arvoon 250,0 kBaud. Kuvassa 34 on esitetty väylätesterin konfiguroinnin pääikkuna.

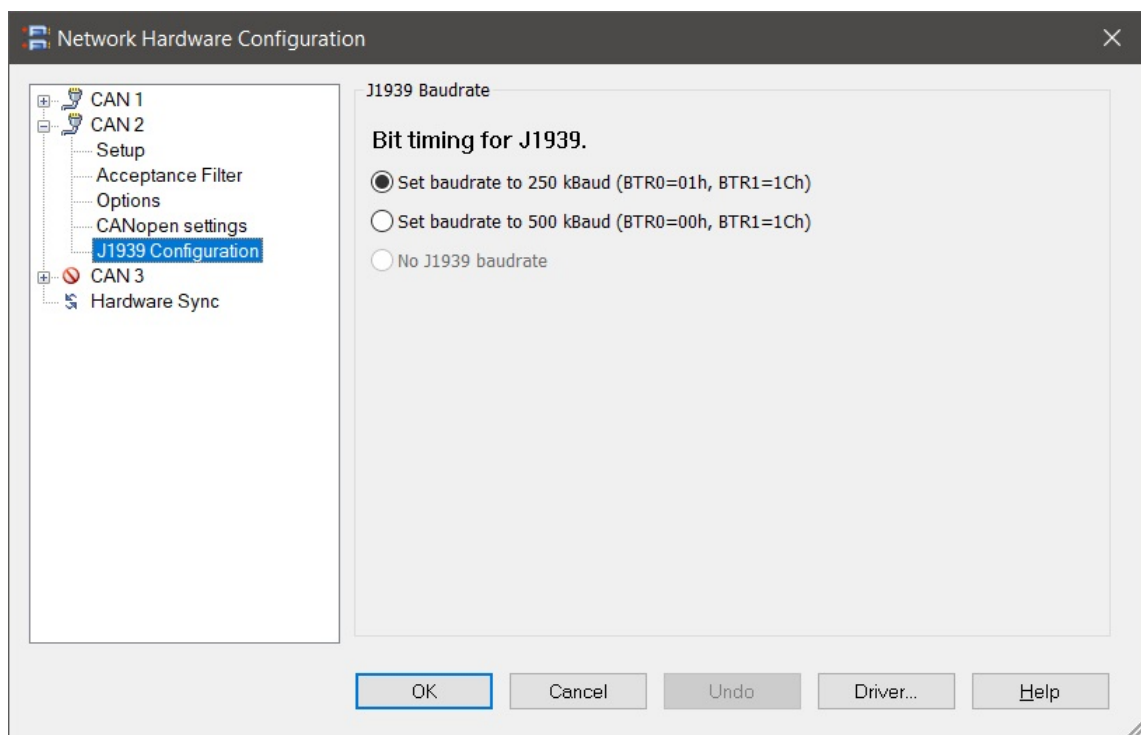


KUVA 34. Väylätesterin konfigurointi CANoe-ohjelmistossa

Pääsivun lisäksi siirtonopeus piti tarkistaa myös CANopen- ja J1939-väyläasetuksista, kuten kuvissa 35 ja 36 on esitetty.

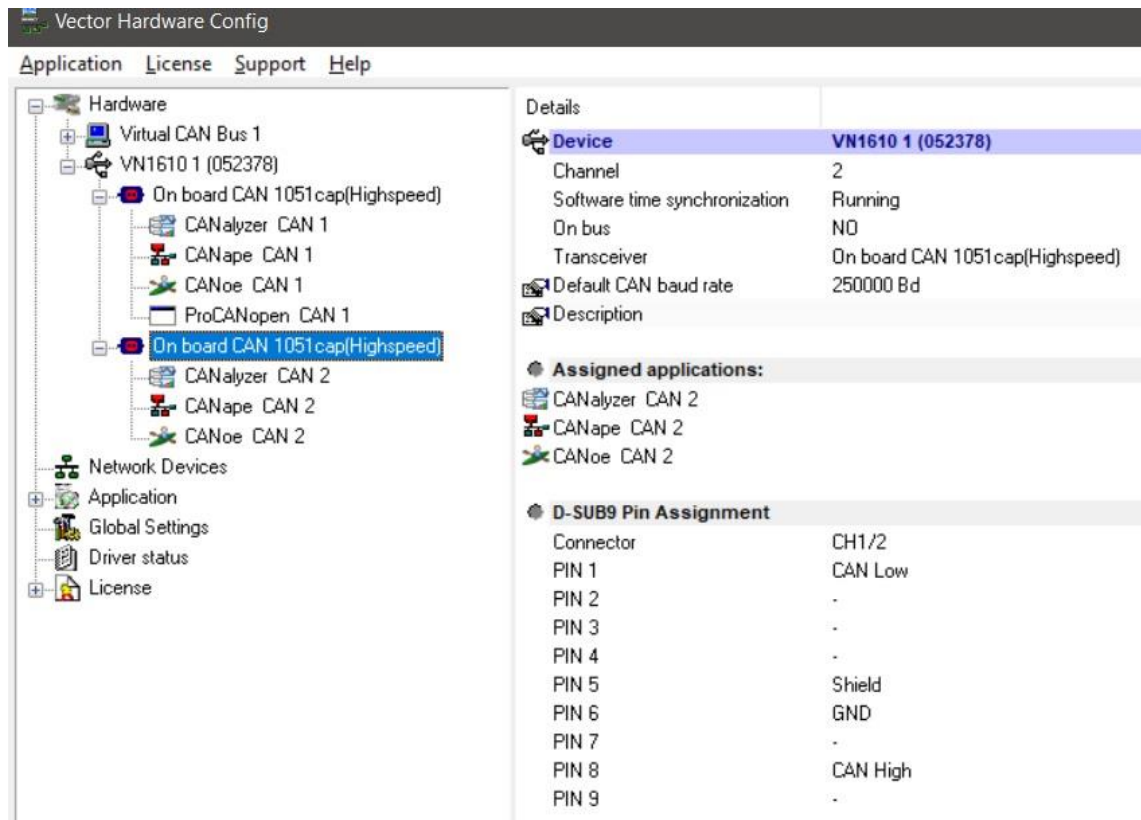


KUVA 35. Väylätesterin CANopen-konfigurointi CANoe-ohjelmistossa



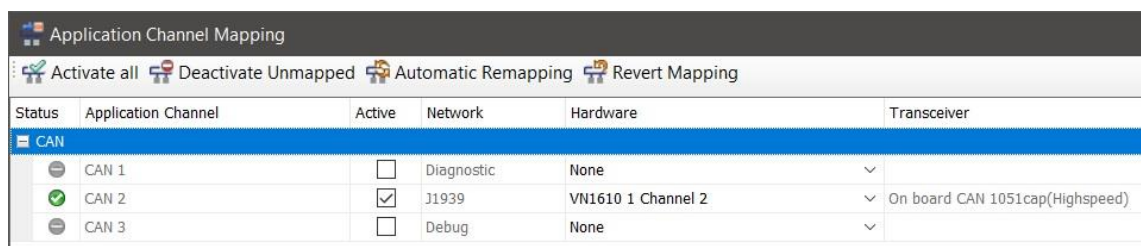
KUVA 36. Väylätesterin J1939-konfigurointi CANoe-ohjelmistossa

Konfigurointi-ikkunoiden "Driver..."-painikkeella saatiin auki kuvan 37 ikkuna, jossa oli nähtävissä väylätesterin tiedot.



KUVA 37. Väylätesterin tiedot CANoe-ohjelmistossa

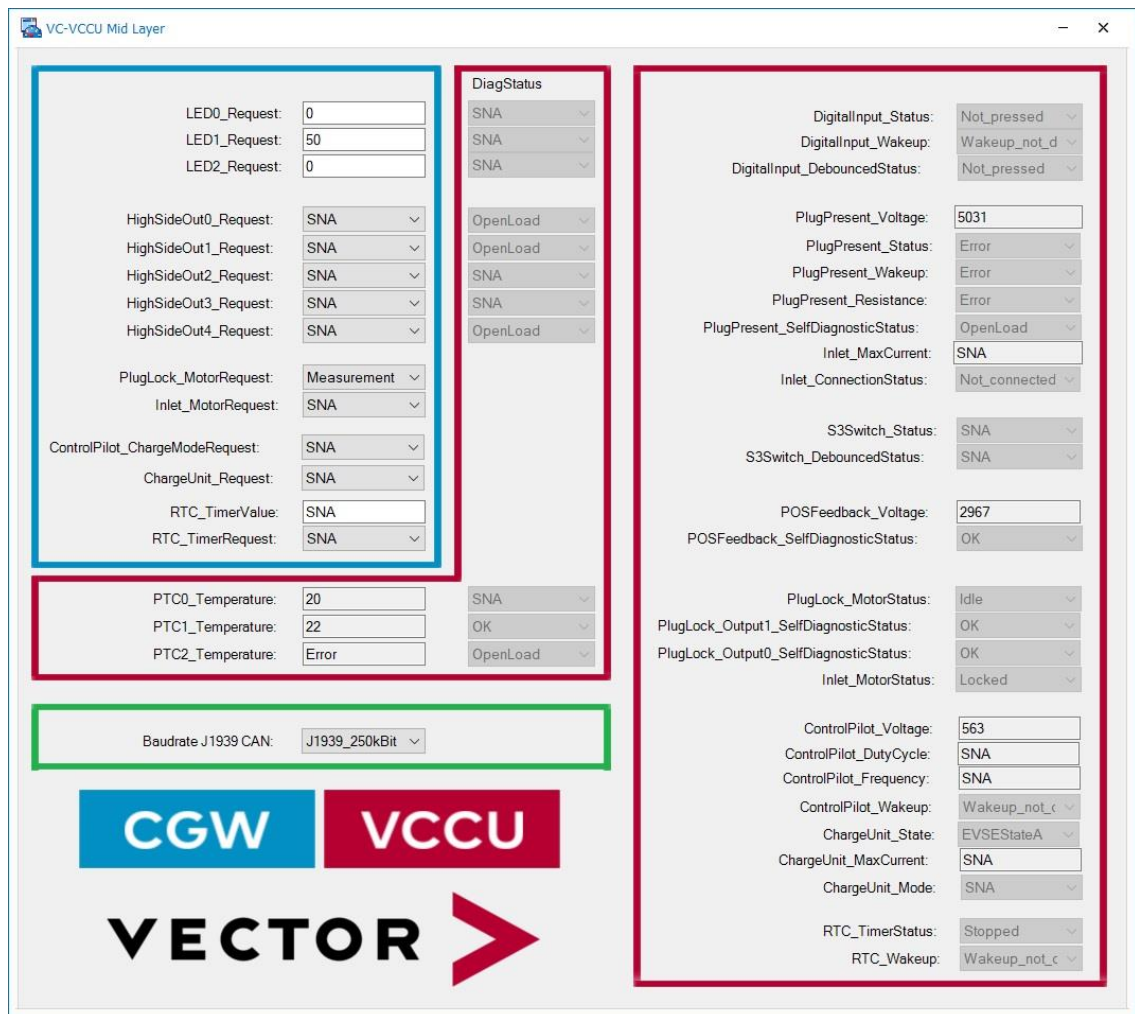
Konfiguroinnin jälkeen väylätesteri määriteltiin käytettäväksi VCCU:n toiminnallisen väylän kanssa. Kuvassa 38 on nähtävissä VCCU:n kolme väylää, joista toisessa tapahtuu VCCU:n kommunikaatio ohjausyksikön kanssa.



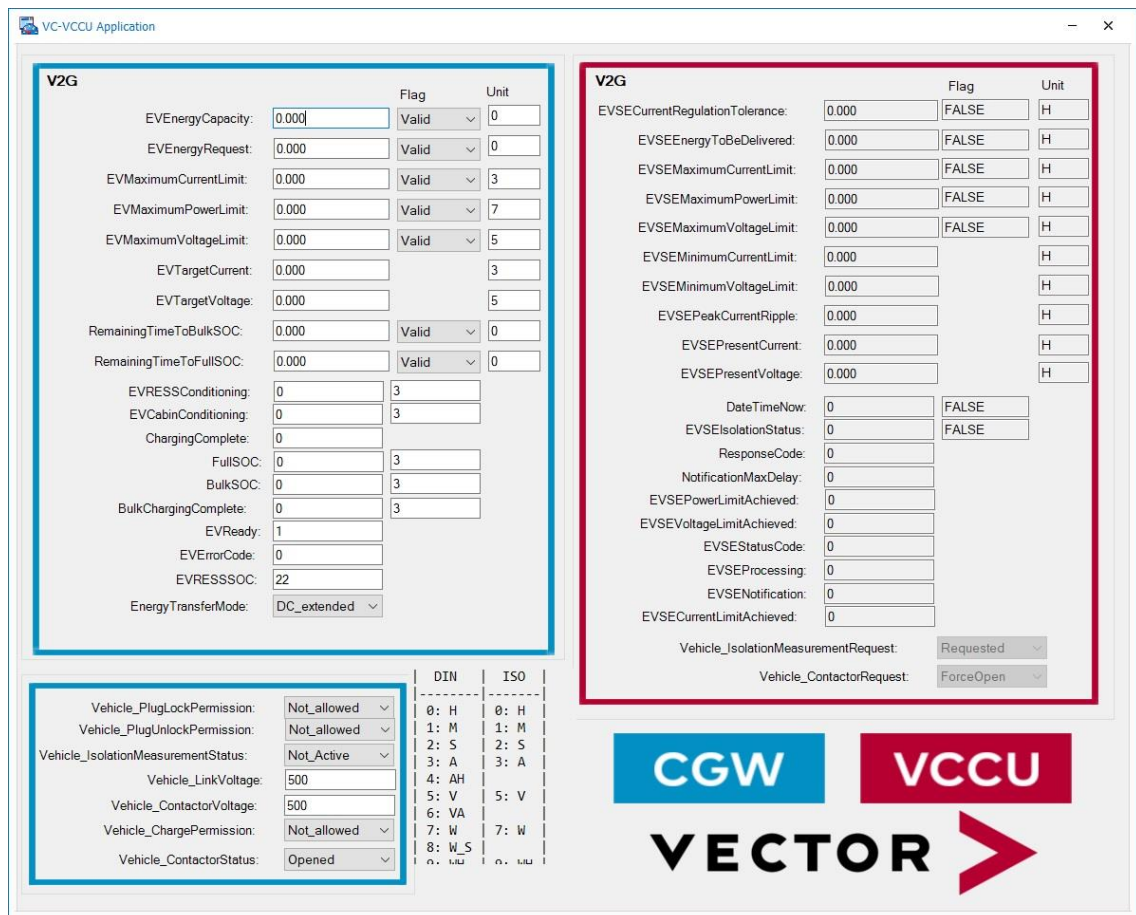
KUVA 38. Väylätesterin valitseminen väylälle CANoe-ohjelmistossa

Väylän tarkkailuun valittiin konfiguroidun VN1610-testerin toinen kanava, joka oli kytketty haluttuun väylään. Tämän jälkeen väylä asetettiin aktiiviseksi, jolloin ohjelma varaa testerin käyttöönsä.

Testerin asettamisen jälkeen CANoe-tarkkailu asetettiin päälle, ja tarkkailtiin kuvien 39 ja 40 testinäyttöjä, ja niillä näkyviä arvoja.



KUVA 39. VCCU:n testinäyttö 1 CANoe-ohjelmistossa



KUVA 40. VCCU:n testinäyttö 2 CANoe-ohjelmistossa

Kuvasta 39 kiinnitetään huomiota toisen ledin ohjausarvoon, sekä kahteen ensimmäiseen lämpötilamittaukseen, jotka tulevat latauspistokkeen antureilta. Lediohjaus on ohjaus fyysiselle merkkiledille, jonka voisi kytkeä VCCU:n lähtöön. Merkkiledin ohjausta ei tule sekoittaa kosketusnäytön ledeihin, joiden ohjaukset asetetaan ohjausyksiköltä.

Testinäyttöjä ei saatu suoraan ajoneuvon 6107-kosketusnäytölle, joten niitä ei käsitellä sen tarkemmin.

Väylässä liikkuvia signaaleja tarkasteltiin myös tarkkailuikkunasta, jossa voitiin lukea suoraan väylään määriteltyjä viestejä ja niiden sisältämiä signaaleja. Kuvassa 41 on esitetty tarkkailuikkunassa liikkuvia viestejä ja avattuna Requests-viestin signaalit.

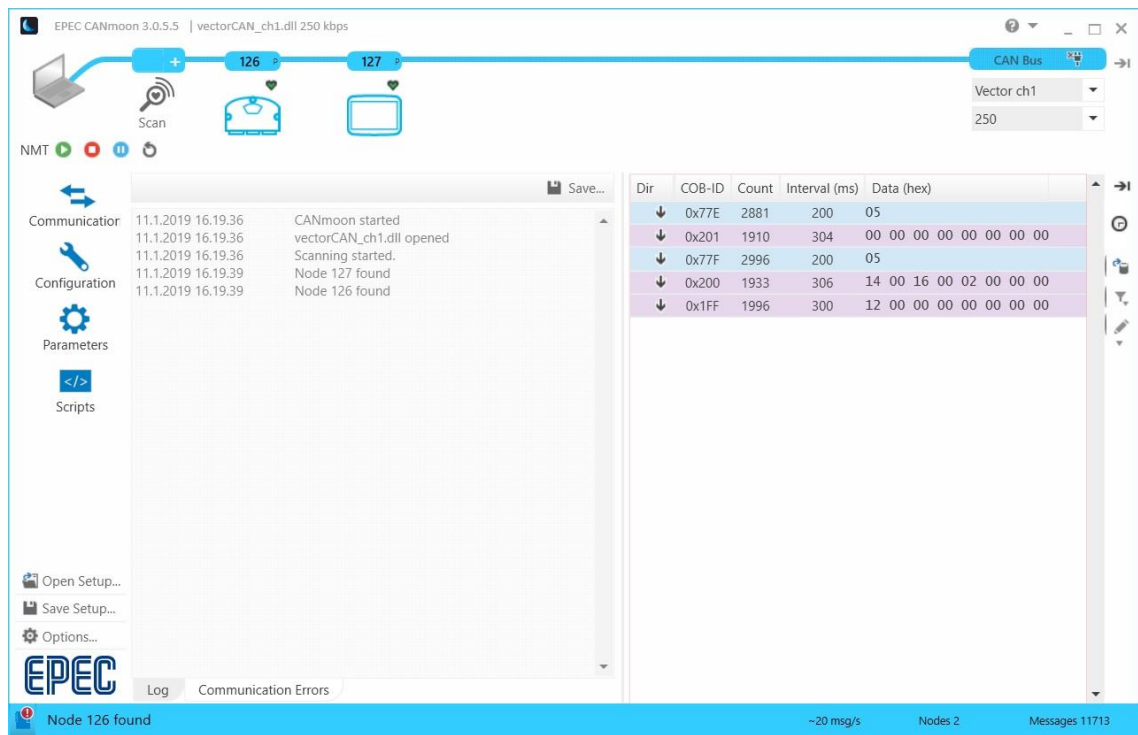
Time	Chn	ID	Name	Event Type	Dir	D...	D...	Data
5543.175677	CAN 2	18EEFF80x		CAN Frame	Rx	8	8	00 00 00 00 00 00 00 00
5544.276721	CAN 2	18FF1080x	VCVCCU_PTC0	CAN Frame	Rx	7	7	FF FF 3B 04 AA 00 FF
5544.277733	CAN 2	18FF1180x	VCVCCU_PTC1	CAN Frame	Rx	7	7	FF FF 3C 04 AC 00 F8
5544.278745	CAN 2	18FF1280x	VCVCCU_PTC2	CAN Frame	Rx	7	7	FF FF 97 06 FE FF FB
5544.279309	CAN 2	18FF1380x	VCVCCU_InletStatus	CAN Frame	Rx	8	8	97 08 B3 02 00 90 FF F0
5544.275649	CAN 2	18FF1480x	VCVCCU_ControlPilotStatus	CAN Frame	Rx	7	7	FF FF FF 33 7F 1C FF
5544.279853	CAN 2	18FF1580x	VCVCCU_OutputStatus	CAN Frame	Rx	7	7	BB FF FB FF 00 00 C7
5544.080301	CAN 2	18FF1680x	VCVCCU_RTCStatus	CAN Frame	Rx	1	1	CC
5544.095465	CAN 2	18FF1780x	VCVCCU_ChargeToVehicle	CAN Frame	Rx	1	1	F9
5544.268216	CAN 2	18FF1880x	VCVCCU_InternalVoltageStatus	CAN Frame	Rx	8	8	CD 2E 08 00 37 13 B1 58
5544.268780	CAN 2	18FF1980x	VCVCCU_InternalVoltageRawValues	CAN Frame	Rx	8	8	52 01 01 00 54 02 94 03
5544.269368	CAN 2	18FF1A80x	VCVCCU_InletStatus2	CAN Frame	Rx	8	8	A7 13 FF FF FF FF FF FF
5544.276189	CAN 2	18FF2082x	VCVCCU_Requests	CAN Frame	Rx	7	7	00 64 00 FF 3F FF 7F
			VCVCCU_LED0_Request	0.0000 %	0	Request for the duty cycle of the LED0		
			VCVCCU_LED1_Request	50.0000 %	64	Request for the duty cycle of the LED1		
			VCVCCU_LED2_Request	0.0000 %	0	Request for the duty cycle of the LED2		
			VCVCCU_HighSideOut0_Request	3	3	Request for high side output 0		
			VCVCCU_HighSideOut1_Request	3	3	Request for high side output 1		
			VCVCCU_HighSideOut2_Request	3	3	Request for high side output 2		
			VCVCCU_HighSideOut3_Request	3	3	Request for high side output 3		
			VCVCCU_HighSideOut4_Request	3	3	Request for high side output 4		
			VCVCCU_RTC_TimerRequest	3	3	RTC timer request		
			VCVCCU_PlugLock_MotorRequest	3	3	Request for the plug lock motor movement		
			VCVCCU_RTC_TimerValue	255 min	FF	RTC timer value		
			VCVCCU_ControlPilot_ChargeModeRequest	7	7	Value of the resistor connected to the Control Pilot which defines the mode used		
			VCVCCU_ChargeUnit_Request	3	3	Charging mode control of CP		
			VCVCCU_Inlet_MotorRequest	3	3	Requests to control the inlet motor		
5544.275112	CAN 2	18FF2182x	VCVCCU_ChargeFromVehicle	CAN Frame	Rx	6	6	F4 01 F4 01 00 00
5544.200351	CAN 2	18FF3082x	VCVCCU_V2G_VehicleStatus	CAN Frame	Rx	7	7	C0 0F C7 0D 00 00 16
5544.199255	CAN 2	18FF3182x	VCVCCU_V2G_RemainingTimeToFullSOC	CAN Frame	Rx	5	5	00 00 00 03 30
5544.198247	CAN 2	18FF3282x	VCVCCU_V2G_RemainingTimeToBulkSOC	CAN Frame	Rx	5	5	00 00 00 03 30
5544.197239	CAN 2	18FF3382x	VCVCCU_V2G_EVTargetVoltage	CAN Frame	Rx	5	5	26 02 00 01 05
5544.196243	CAN 2	18FF3482x	VCVCCU_V2G_EVTargetCurrent	CAN Frame	Rx	5	5	0A 00 00 01 03
5544.278205	CAN 2	18FF3582x	VCVCCU_V2G_EVEnergyCapacity	CAN Frame	Rx	5	5	30 00 00 00 03
5544.245156	CAN 2	18FF3682x	VCVCCU_V2G_EVEnergyRequest	CAN Frame	Rx	5	5	30 00 00 00 03
5544.246156	CAN 2	18FF3782x	VCVCCU_V2G_EVMaximumCurrentLimit	CAN Frame	Rx	5	5	13 00 14 00 01
5544.247156	CAN 2	18FF3882x	VCVCCU_V2G_EVMaximumPowerLimit	CAN Frame	Rx	5	5	17 00 B0 36 01
5544.195243	CAN 2	18FF3982x	VCVCCU_V2G_EVMaximumVoltageLimit	CAN Frame	Rx	5	5	15 00 58 02 01
5544.277197	CAN 2	18FF4082x	VCVCCU_V2G_DepartureTime	CAN Frame	Rx	5	5	00 00 00 00 03
5544.060548	CAN 2	18FF5080x	VCVCCU_V2G_EVSEStatus	CAN Frame	Rx	8	8	80 00 C0 E0 00 00 00 00
5543.960318	CAN 2	18FF5180x	VCVCCU_V2G_EVSECurrentRegulationTolerance	CAN Frame	Rx	5	5	C0 00 00 00 00

KUVA 41. Signaalien seuranta CANoe-ohjelmistossa

Kuvasta 41 voidaan havaita toisen ledin ohjauksesta vastaava arvo kuin kuvan 39 näytössä.

7.2 CANmoon

CANoe-ohjelmiston lisäksi väylää tarkasteltiin CANmoon-ohjelmistolla. CANmoonilla tutkittiin ensiksi kosketusnäytön ja ohjausyksikön välistä liikennettä. Kuvasta 42 voidaan havaita ohjausyksiköltä näytölle tulevat viestit, joiden COB-ID:t ovat 200 ja 201, sekä näytöltä ohjausyksikölle tulevat painiketiedot, jotka ovat COB-ID:llä 1FF.

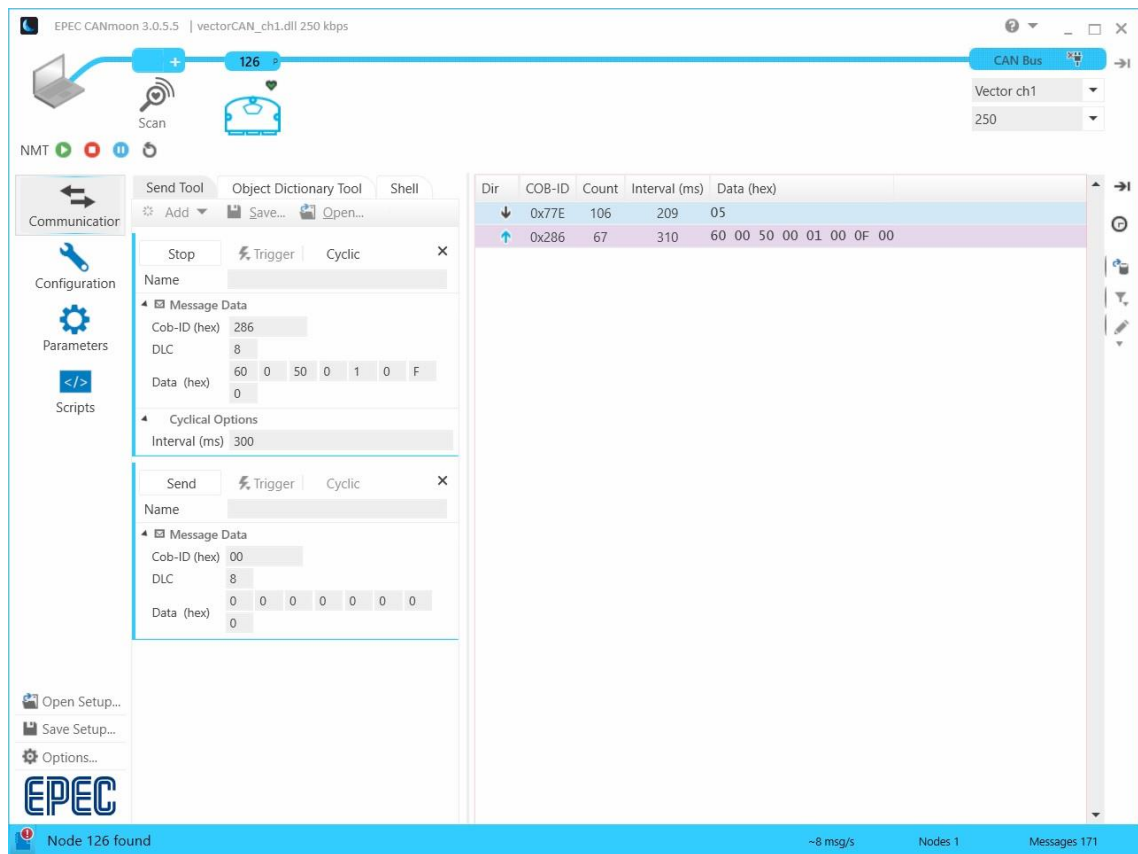


KUVA 42. Näytön ja ohjausyksikön välinen kommunikaatio CANmoon-ohjelmistossa

CANmoon esittää viestin sisällön heksalukuina, niin että vähiten merkitsevä tavu on vasemmalla ja eniten merkitsevä tavu oikealla, eli käytettiin Big Endian -määrittelyä (GeeksforGeeks 2019).

Kuvan 42 COB-ID 200 viestistä voidaan nähdä latauspistokkeen lämpötilamittausten arvot 20 °C ja 22 °C, sekä ohjaustieto ledille 2. Kosketusnäytön toinen ledi on varattu latauspistokkeen lukitustiedolle, jonka tila oli simuloitu kytkimellä. Lämpötiloja voidaan verrata kuvan 39 arvoihin ja todeta arvojen oikeellisuus.

CANmoonilla voitiin myös simuloida akunhallintajärjestelmää lähettämällä väylään haluttuja viestejä. Testitarkoituksiin valittiin COB-ID 286 viesti, joka sisälsi muun muassa akun varauksen tason. Kuvassa 43 on esitettyä CANmoonin lähetystyökalun ikkuna, jolla lähetetään syklisesti viestiä väylään.

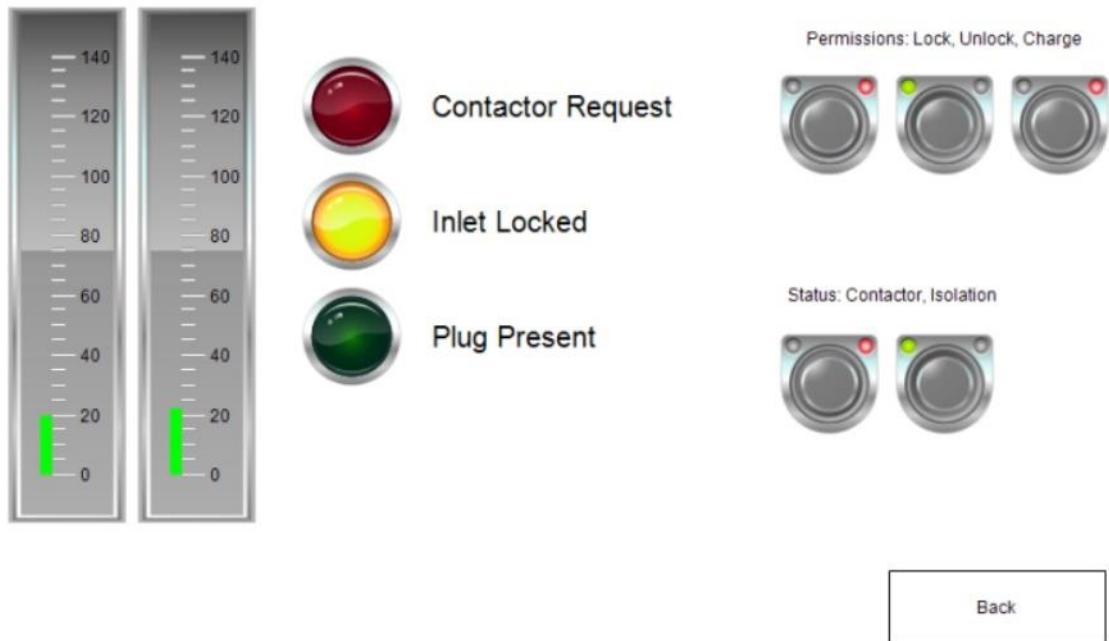


KUVA 43. Akunhallintajärjestelmän simulointi CANmoon-ohjelmistossa

Viesti on koottu vanhan akunhallintajärjestelmän mukaisesti: ensimmäisillä kahdella tavulla määritellään akun varauksen taso, seuraavilla kahdella tavulla asetettiin latausvirran pyynti, neljännellä ja viidennellä tavulla valittiin latausjännitteen pyynti ja viimeisillä kahdella tavulla rajoitettiin purkuvirran maksimi. Arvot syötettiin heksalukuina kuten näytön testaamisen yhteydessä kerrottiin. Akun taso oli siis 96 %, latausvirta 80 A, jännite 1 V ja purkuvirta 15 A. Arvot valittiin mielivaltaisesti, samalla tutkien heksalukujen esitystapaa CANmoonissa.

7.3 Codesys

Tietoliikennettä tarkasteltiin myös Codesysin puolella. Kuvassa 44 on esitetty kosketusnäytön testi-ikkuna testausten aikana.



KUVA 44. Kosketusnäytön testi-ikkuna Codesys-ohjelmistossa

Kuvasta 44 voidaan havaita lämpötilamittaukset 20 °C ja 22 °C sekä latauspistokkeen lukitustieto, joka on esitetty keltaisena ledinä. Lukon avauslupa ja pistokkeen eristys on myös asetettu aktiivisiksi näytöstä.

Codesysillä voitiin myös tarkastella laitteilla olevia arvoja. Kuvassa 45 on esitetty ohjausyksiköltä näytölle lähtevät arvot.

OD1_CANO x		
Device.Application.OD1_CANO		
Expression	Type	Value
CANO_Display_temp1	WORD	20
CANO_Display_temp2	WORD	22
CANO_Display_soc	INT	96
CANO_Display_leds	BYTE	2
CANO_Display_EVSE_state	BYTE	0
CANO_Display_BMS1_warnings1	BYTE	0
CANO_Display_BMS1_warnings2	BYTE	0
CANO_Display_BMS1_alarms1	BYTE	0
CANO_Display_BMS1_alarms2	BYTE	0
CANO_Display_BMS2_warnings1	BYTE	0
CANO_Display_BMS2_warnings2	BYTE	0
CANO_Display_BMS2_alarms1	BYTE	0
CANO_Display_BMS2_alarms2	BYTE	0

KUVA 45. Ohjausyksiköltä näytölle lähtevät viestit Codesys-ohjelmistossa

Kuvasta 45 voidaan havaita, että akun varauksen tilaa kuvaava muuttuja ”soc” (State of Charge) on automaattisesti muuttunut heksaluvusta kymmenjärjestelmän luvuksi, kun se on luettu CANopen-väylästä.

Kuten aiemmin on mainittu, Multitoolilla määritellyt muuttujat generoituivat automaattisesti Codesys-ohjelmaan, mutta niiden käyttö piti määrittellä itse. Kuvassa 46 on esitetty osa J1939-väylään lähetettävistä viesteistä.

```

3 // can2 rx (VCCU, J1939)
4
5 ● temperature_PTC0[20] := J1939_CAN2.In.VCCU.PTC0_VCVCCU_PTC0_Temperature[170] - 150; //offset -150
6 ● temperature_PTC1[22] := J1939_CAN2.In.VCCU.PTC1_VCVCCU_PTC1_Temperature[172] - 150;
7
8 // can2 tx (VCCU, J1939)
9
10 // VCVCCU_ChargeFromVehicle
11 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_ChargePermission[0] := BOOL_TO_BYTE(CANI_Display_charge[FALSE]);
12 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_ContactorStatus[0] := BOOL_TO_BYTE(CANI_Display_contactor[FALSE]);
13 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_ContactorVoltage[500] := 500;
14 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_IsolationMeasurementStatus[1] := BOOL_TO_BYTE(CANI_Display_isolation[TRUE]);
15 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_LinkVoltage[500] := 500;
16 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_PlugLockPermission[0] := BOOL_TO_BYTE(CANI_Display_lock[FALSE]);
17 ● J1939_CAN2.Out.Default.CFV_VCVCCU_Vehicle_PlugUnlockPermission[1] := BOOL_TO_BYTE(CANI_Display_unlock[TRUE]);

```

KUVA 46. Ohjausyksiköltä J1939-väylään lähteviä viestejä Codesys-ohjelmistossa

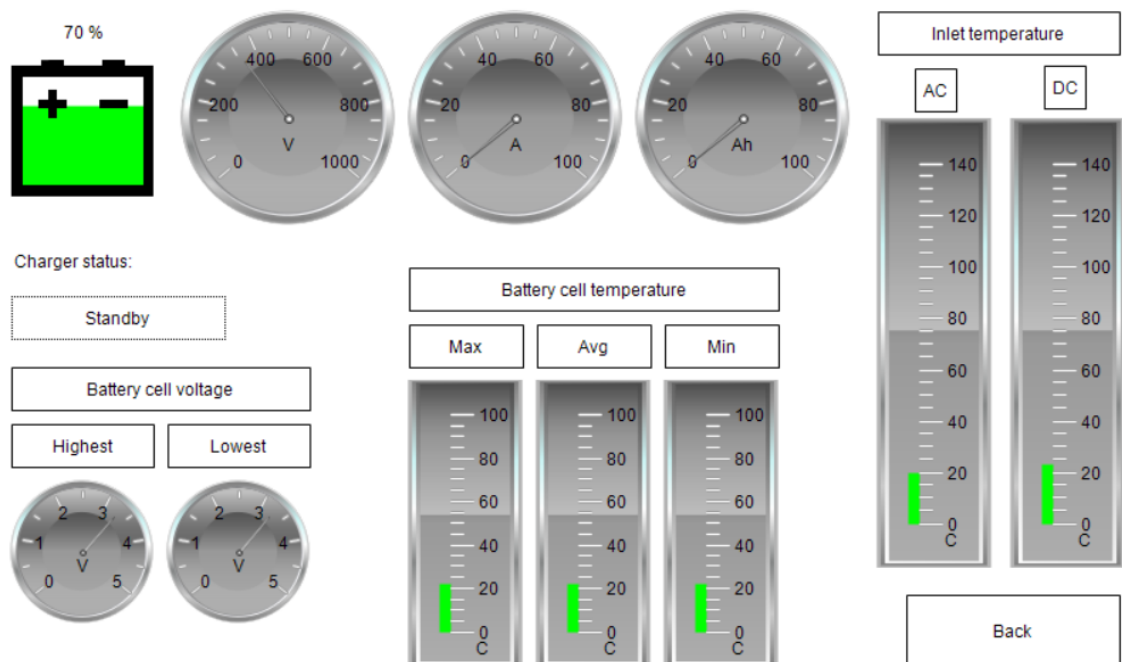
Kuvasta 46 voidaan huomata, että lämpötilamittauksiin pitää huomioida 150 asteen korjaus, joka oli määritetty valmistajan manuaaleissa (Vector Group 2018, VC-VCCU CAN Signals, 6). Kuvasta on havaittavissa myös pistokkeen eristykseen ja lukon avaamisen signaalit kosketusnäytöltä VCCU:lle, kuten kuvassa 44 oli asetettu.

8 JÄRJESTELMÄN TESTAUS

Järjestelmän testaus suoritettiin oikean akuston ja latausaseman kanssa. Akunhallintajärjestelmältä saatiin tuotua näytölle aiemmin puuttuneet tiedot. Kuten muukin laitteiden välinen kommunikaatio, tämäkin toteutettiin 5050-ohjausyksikön kautta.

8.1 Akunhallintajärjestelmän käyttöönotto

Latauksenhallintanäyttöä editoitiin kytkemällä akunhallintajärjestelmän mittaukset näyttökomponentteihin. Kuvassa 47 on esitetty latauksenhallintanäyttö, kun mittaukset ovat käytössä.



KUVA 47. Latauksenhallintanäyttö, kun kytkeydytään akunhallintajärjestelmään

Kuvasta 47 voidaan havaita päivittyneet mittausravot akuston lataustasolle, akuston kokonaisjännitteelle, yksittäisten akkujen korkeimmalle ja matalimmalle jännitteelle, sekä akkujen lämpötiloille.

Käyttöönoton helpottamiseksi akunhallintajärjestelmän tilatiedot tuotiin myös omaksi näytökseen. Osa tilatiedoista oli määritelty kiinteästi, osa oli valittavissa Orion BMS 2 Control Application -ohjelmistolla. Kuvassa 48 on esitetty akunhallintajärjestelmän tilatietonäyttö, kun laitteisto on aktiivisena.

Discharge relay enabled
Charge relay enabled
Charger safety enabled
DTC error active
Multi-purpose input #1 signal status
Always-on signal status
Is-ready signal status
Is-charging signal status
Multi-purpose input #2 signal status
Multi-purpose input #3 signal status
RESERVED
MP0#2 Isolation OK status
Multi-purpose output #3 signal status
Multi-purpose output #4 signal status
Multi-purpose enable signal status
Multi-purpose output #1 signal status

Back

KUVA 48. Akunhallintajärjestelmän aktiiviset tilatiedot

Tilatiedot olivat binäärisiä, eivätkä välttämättä suoranaisesti hyviä tai huonoja tiloja, joten 1-tilassa olevat signaalit esitettiin keltaisella taustalla ja 0-tilassa olevat valkoisella.

Tilatietojen lisäksi koestettiin myös ilmoituslista. Kuvan 48 perusteella eristyksen tila ei ole kunnossa, ja DTC error active -tila on päällä. Tämä havaitaan myös ilmoituslistalta kuvassa 49.

	Timestamp	Message	Class
0	24.04.2019 16:41:49	High Voltage Isolation Supply	Error

Back

KUVA 49. Aktiivinen vikailmoitus ilmoituslistalla

8.2 Latauspistokkeen toiminta

Järjestelmän testauksissa saatiin myös koestettua toimintoja oikean latauspistoolin kanssa. Testaukset suoritettiin Tampereen Kaupunginliikenteen (TKL) Nekalan varikolla. Testauksissa käytettiin puolalaisen Ekoenergytukan valmistamaa latausasemaa, joka on esitetty kuvassa 50.



KUVA 50. Käytetty latausasema (Reko 2019)

Kuvassa 51 on esitetty tarkemmin latausaseman käyttöpaneeli.



KUVA 51. Latausaseman käyttöpaneeli (Reko 2019)

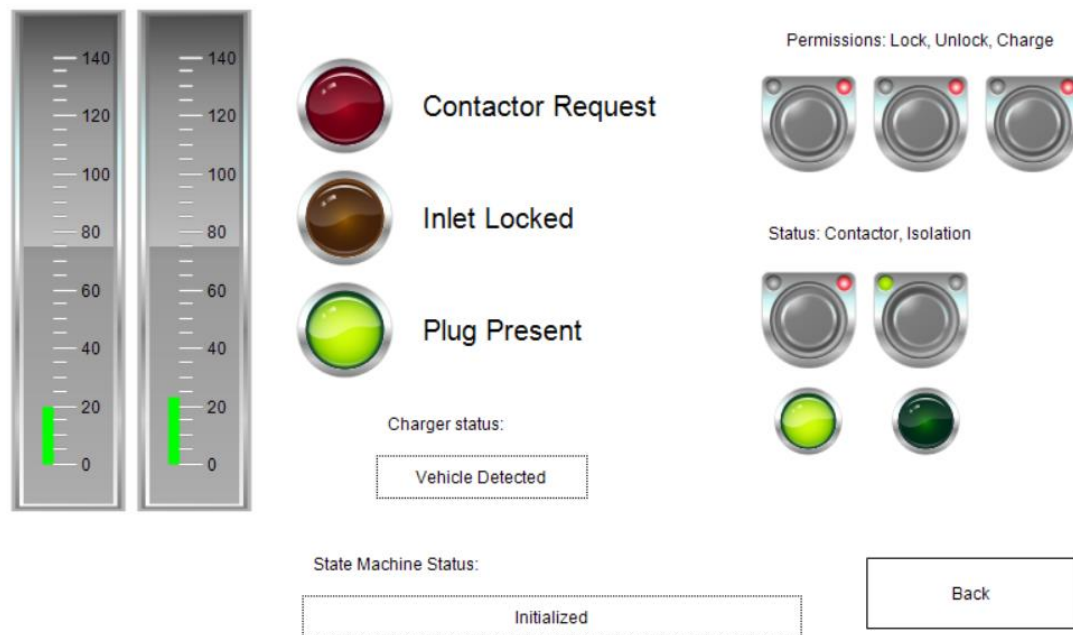
Kuvan 51 paneelista voidaan havaita ylärivin tilatiedot, sekä alariviltä käynnistyspainikkeet ja hätäseisäkytkin.

Latausasemalta tuotiin latauspistooli projektissa käytettyyn latauspistokkeeseen. Molemmat sattuvat olemaan Phoenix Contactin valmistamia, vaikka mikä tahansa muukin 2-tyyppin CCS-pistooli olisi kelvannut.



KUVA 52. Latauspistooli latauspistokkeessa (Reko 2019)

Kun latauspistooli asetettiin latauspistokkeeseen, saatiin järjestelmään tieto, että laitteisto on havainnut latauspistoolin. Kuvassa 53 on esitetty päivitetty versio kuvan 6 laborioriotestinäytöstä, jolla testiä seurattiin ja ohjattiin.



KUVA 53. Laborioriotestinäyttö, kun kytkeydytään latausasemaan

Kuvasta 53 voidaan havaita, että ohjelma on havainnut kytketyn latauspistoolin.

Latauspistokkeen ja VCCU:n kautta saatiin myös tieto latausaseman tilasta (Charger status). Latausaseman tilatieto tulee latauspistokkeen Control Pilot -kommunikaation kautta. Tieto oli alun perin esitettyä vain kuvan 13 näytössä, mutta se on lisätty kuvan 53 näyttöön testaamisen helpottamiseksi. Tämän lisäksi näytölle tuotiin myös VCCU:n sekvenssin askeltieto (State Machine Status).

Kuvan 53 näyttöön on myös lisätty akunhallintajärjestelmältä saatavat tiedot kontaktorien ja eristyksen oikeista tiloista, vaikka niitä vielä tässä vaiheessa asetettiin näytöstä manuaalisesti. Voidaan havaita, että akuston kontaktori on BMS:llä aktiivisena, mutta eristysmittaus ei ole.

8.3 VCCU:n latausohjelman testaus

Lopulta testattiin myös varsinaisen latausohjelman toimintaa latausaseman kanssa. VCCU:n latausohjelma oli toimitettu valmiina, mutta VCCU tarvitsi ulkoisia mittauksia ja ohjauksia sen suorittamiseen. Testattava olikin, onko kaikki tarpeelliset tiedot saatu johdettua VCCU:lle.

Kuvassa 53 oli esitettyä tila pistoolin kytkemisen jälkeen, tästä edettiin antamalla lukitsemislupa, jolloin pistooli lukittui pistokkeeseen. Lukituksen tilatieto asetettiin luvussa 6.1 ja kuviossa 4 esitellyllä vipukytkimellä, koska latauspistokkeen lukon tilatieto ei ollut yhteensopiva VCCU:n kanssa. Tämän jälkeen VCCU:n sekvenssi aloittaa kommunikoinnin latausaseman kanssa control pilot -kommunikaatiolla.

Kun VCCU:n sekvenssi pääsee kaapelitestiaskeleeseen (Cable check) annetaan kuvan 53 näytöstä latauslupa (Charge). Tämän jälkeen sekvenssi jatkaa esilataukseen (Pre-charge), jossa eristysmittaus siirtyy latausaseman tarkasteltavaksi ja BMS:n kontaktori suljetaan. Sekvenssi jatkaa latausaskleeseen (Current demand) ja jää lataamaan, kunnes siltä poistetaan latauslupa. Lataus lopetetaan myös akuston täytyessä tai akuston lämpötilojen noustessa liian korkealle.

9 POHDINTA

Latausjärjestelmän päivittäminen osoittautui ohjelmoinnin kannalta realistiseksi projektiksi. Pääasiassa työ oli laitteiden konfigurointia toimimaan yhdessä, mikä ei aina ollut yksinkertaista, kun eri laitteilla oli käytössä omat konfigurointityökäytäntönsä. Valituilla laitteilla konfigurointi saatiin kuitenkin tehtyä, eikä jouduttu tilanteeseen, jossa esimerkiksi valmistajan vanha ohjelmakoodi estäisi muutoksia. Tässä auttoi se, että pääohjelma oli tehty Epec:in laitteille ja oli siten helposti muokattavissa. Merkittävämpiä haasteita voisi aiheuttaa tarvittavien laitteiden asentaminen ajoneuvoon, mitä tässä työssä ei tehty.

VCCU:n väyläliikenteen kanssa oli aluksi merkittäviä ongelmia. Väylää kuitenkin saatiin testattua CANmoon-ohjelmalla, jonka avulla saatiin selvitettyä, että VCCU lähetti CAN-väylään väärää dataa väärillä kommunikaatioparametreilla. Lopulta selvisikin, että toimitetussa VCCU:n versiossa oli ohjelmavirhe. VCCU lähetettiin valmistajalle ohjelmistopäivitystä varten. Päivityksen jälkeen kommunikaatio toimi kuten piti.

Seuraavat työvaiheet tämän projektin jälkeen olisi laitteistojen asentaminen ajoneuvoon ja näyttöikkunoiden parantelu. Kun järjestelmää pääsisi käyttämään, voisi tehdä havaintoja, millaista informaatiota oikeasti latauksenhallinnasta oikeasti tarvitaan ja miten sitä tulisi esittää niin, että tärkeimmät asiat ovat parhaiten esillä. Akunhallintajärjestelmältä olisi mahdollista tuoda useitakin eri arvoja, mutta on kyseenalaista, onko niistä hyötyä operaattorille. Liiallinen informaatiomäärä voi olla haitallista, mikäli tärkeät tiedot hukkuvat muiden tietojen sekaan.

Henkilökohtaisten oppimistavoitteiden näkökulmasta projekti onnistui varsin hyvin perehdyttämään CAN-väyliin, käytettyihin ohjelmistoihin sekä väylätestereiden hallintaan ja väylän tarkkailuun. Projektissa oli useita eri ohjelmistoja, joita oli käytettävä ristiin, jotta saatiin toteutettua kokonaisuus, jossa kommunikaatio kulki laitteiden välillä oikein. Lisäksi projektissa tuli tutustuttua sähköajoneuvojen ja latausjärjestelmien toimintaan sekä sähköajoneuvojen tilanteeseen Euroopan alueella.

LÄHTEET

- Ahn, K., Fiori, C. & Rakha, H. 2016. Power-based electric vehicle consumption model: Model development and validation. Tulostettu 3.1.2019. https://www.researchgate.net/profile/Hesham_Rakha/publication/294112610_Power-based_electric_vehicle_energy_consumption_model_Model_development_and_validation/links/5a0089cd0f7e9b62a150ba39/Power-based-electric-vehicle-energy-consumption-model-Model-development-and-validation.pdf
- Bayer, F. & Leserer, J. 2017. VC-VCCU Technical Reference Version 1.1. Vector Informatik GmbH.
- Codesys Group. 2017. Codesys - the comprehensive software suite for automation technology. Tulostettu 4.1.2019. <https://www.codesys.com/the-system.html>
- Epec Oy. 2019. Accessories and Ordering codes. Tulostettu 25.4.2019. http://epec.planeetta.com/Public/Manuals/EPEC_Programming_And_Libraries/projecttopics/topic000918.htm
- Epec Oy. 2019. Epec 5050 Control Unit. Tulostettu 25.4.2019. <https://epec.fi/products/control-system-products-5050/>
- Epec Oy. 2019. Technical Manual and Cabling Instructions. Tulostettu 25.4.2019. http://epec.planeetta.com/Public/Technical_Documents/6107/MAN000596/6107_webhelp_man000596.htm
- Ewert Energy Systems, Inc. 2019. Orion BMS 2. Tulostettu 25.4.2019. <https://www.orionbms.com/products/orion-bms-standard/>
- Foltyński, M. 2014. Electric fleets in urban logistics. Tulostettu 26.12.2018. <https://www.sciencedirect.com/science/article/pii/S1877042814054494>
- Garo. 2013. Latausasemat. Garo Finland Oy. Tulostettu 2.1.2019. https://www.garo.fi/fileadmin/garofi/Kataloger/AU/Latausasemat_05-13.pdf
- GeeksforGeeks. 2019. Little and Big Endian Mystery. Tulostettu 3.5.2019. <https://www.geeksforgeeks.org/little-and-big-endian-mystery/>
- Hietalahti, L. 2017. eTruck - sähköinen 16t jakeluauto. Tulostettu 2.1.2019. <http://etruck.blogs.tamk.fi/2017/01/26/etruck-sahkoinen-jakeluauto/>
- IEC 61131-3. 2013 Programmable controllers - Part 3: Programming languages. Preview. Tulostettu 4.1.2019. https://webstore.iec.ch/preview/info_iec61131-3%7Bed3.0%7Db.pdf
- Kvaser. 2019. J1939 Introduction. Tulostettu 3.5.2019. <https://www.kvaser.com/about-can/higher-layer-protocols/j1939-introduction/>

McDonald, Z. 2014. A Simple Guide To DC Fast Charging. FleetCarma, a Gettab Company. Tulostettu 2.1.2019. <https://www.fleetcarma.com/dc-fast-charging-guide/>

Phoenix Contact. Locking - EV-T2M3S-E-LOCK12V - 1624129. Tekniset tiedot. Tulostettu 20.3.2018.

<https://www.phoenixcontact.com/online/portal/fi?uri=pxc-oc-itemdetail:pid=1624129&library=fien&tab=1>

Phoenix Contact. Vehicle Connector test adapter - EV-T2MBIE12-3ACDC-INFRA - 1623492. Tekniset tiedot. Tulostettu 20.3.2018.

<https://www.phoenixcontact.com/online/portal/fi?uri=pxc-oc-itemdetail:pid=1623492&library=fien&tab=1>

Sadler Consultants Ltd. Urban Access Regulations In Europe: What are Low Emission Zones. Tulostettu 2.1.2019. <http://urbanaccessregulations.eu/low-emission-zones-main/what-are-low-emission-zones>

Schwitters, C. 2013. AC versus DC charging - what is the difference. Plug In America. Tulostettu 2.1.2019. <https://pluginamerica.org/ac-versus-dc-charging-what-difference/>

SFS EN 61851-1. 2012. Electric vehicle conductive charging system - Part 1: General requirements (IEC 61851-1:2010). Helsinki: Suomen Standardoimisliitto SFS ry. Luettu 21.8.2018. Vaatii käyttöoikeuden. <https://online.sfs.fi/fi/index.html.stx>

Vector Group. 2018. Technical Reference VC-VCCU CAN Messages.

Vector Group. 2018. Technical Reference VC-VCCU CAN Signals.

LIITTEET

Liite 1. Laitteiden kytkennät

CAN Adapterin liitin	EPEC 5050	VCCU	VCCU description	VCCU Assignment	Orion BMS2	EPEC 6107	Latauspiste	Johdin	Vipukytkin
1	Logic GND	GND	ECU GND	Terminal 31				13	
4	CAN2H	CAN1_H	High Speed CAN	J1939 CAN	CANH			21	
16	CAN2L	CAN1_L	High Speed CAN	J1939 CAN	CANL			20	
24	Logic Power Supply	VCC_KL30	KL30	Terminal 30				24	
1	Logic GND					PWR_GND		22	
5	CAN1H					CAN1_H		8	
17	CAN1L					CAN1_L		7	
24	Logic Power Supply					PWR_IN		23	
		PTC0	Analog IN PTC	PTC AC			AC Temp +	Violetti	
		CP	Control Pilot	Control Pilot			Control Pilot	Valkoinen	
		PP	Proximity Pin	Proximity Pin			Proximity Pin	Valkoinen-sininen	
		PTC1	Analog IN PT1000	PT1000 DC+			DC Temp +	Keltainen	
		FB_OUT0	Full Bridge 12V	Plug Lock 1			Lock control +	Sininen-punainen	
		FB_OUT1	Full Bridge 12V	Plug Lock 2			Lock control -	Sininen-ruskea	
		POS_FB	Lock Position	Position Feedback				4	Lock position +
		SS_GND	Analog Ground	GND AC Temp			AC Temp -	Violetti-ruskea	
		SS_GND	Analog Ground	GND DC Temp			DC Temp -	Keltainen-ruskea	
		SS_GND	Analog Ground	GND Lock Position				6	Lock Position -