

Sähkökuorma-auton ohjausjärjestelmän ohjelmointi

Alexi Siirtola

OPINNÄYTETYÖ
Joulukuu 2021

Sähkö- ja automaatiotekniikan koulutus
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikka
Automaatiotekniikka

SIIRTOLA, ALEKSI:
Sähkökuorma-auton ohjausjärjestelmän ohjelmointi

Opinnäytetyö 49 sivua, joista liitteitä 0 sivua
Joulukuu 2021

Opinnäytetyö tehtiin Tampereen ammattikorkeakoululle osana eRetrofit-hanketta. Opinnäytetyössä suunniteltiin ja toteutettiin dieselkuorma-autosta sähkökuorma-autoksi muunnettavan ajoneuvon sulautetun ohjausjärjestelmän ohjelmointi. Opinnäytetyön tavoitteena oli tuottaa toimiva ohjelma, jolla pystytään ohjaamaan ajoneuvon sulautetun ohjausjärjestelmän perustoimintoja. Tehtyä ohjelmaa käytetään eRetrofit-hankkeessa sähkökuorma-auton ohjausyksikön ja muiden toimilaitteiden toimintaa ohjaavana ohjelmana.

Ajoneuvon ohjausyksikkönä käytettiin opinnäytetyössä Pi Innovon M580-ohjausyksikköä. Ohjausyksikön ohjelmointi tapahtui mallipohjaisesti Simulinkilla hyödyntäen ohjausyksikön valmistajan luomaa lisäkirjastoa, joka on tarkoitettu ohjausyksikön ohjelmointiin ja konfigurointiin. Simulinkin lisäksi työn aikana käytettiin muiden ohjausjärjestelmän toimilaitteiden ohjelmointi- ja konfigurointityökaluina Vector CANdb++ Editoria, Pi Snoopia sekä CODESYSiä.

Opinnäytetyön tuloksena oli toimiva ohjelma, jolla pystytään työn aikana suoritettujen testien perusteella ohjaamaan sähkökuorma-autolle muuntotyön aikana rakennetun sulautetun ohjausjärjestelmän perustoimintoja. Ohjelman toimintaa testattiin sen kehitystyön aikana jokaisen ohjelmaan lisätyn ominaisuuden osalta.

Opinnäytetyössä tuotetun ohjelman jatkokehitys tulee koostumaan ohjelman käyttöönoton yhteydessä suoritettavista toimenpiteistä. Toimenpiteitä tulevat olemaan esimerkiksi ohjelman käyttämien arvojen kalibrointi ja ohjelman toiminnan testaaminen oikealla kalustolla.

Asiasanat: sähköajoneuvo, mallipohjainen ohjelmointi, ohjausjärjestelmä

ABSTRACT

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

SIIRTOLA, ALEKSI:
Programming the Control System of an Electric Truck

Bachelor's thesis 49 pages, appendices 0 pages
December 2021

The purpose of this bachelor's thesis was to create a control program for the embedded control system of an electric truck. The thesis was a part of the project eRetrofit by Tampere University of Applied Sciences, in which a decommissioned diesel truck was converted into an electric truck to study and develop the conversion process. The objective of the thesis was to produce a working program capable of controlling the vehicle's embedded system's basic functions. The resulting program will be used in the project eRetrofit as the control program of the electric truck's electronic control unit (ECU).

The electronic control unit used during the thesis was Pi Innovo M580. The ECU was programmed in model-based form using Simulink, which was equipped with an additional OpenECU function block library, which provided the function blocks used to program and configure the ECU.

Based on the tests conducted during the thesis work, the result of the thesis was a working program, which can be used to control the embedded system installed on the vehicle during the conversion process. The program was tested during the development process after every added feature.

The future development of the program created during the thesis will include the commissioning and further testing of the program. The future development will include calibrating the variable values used in the program and testing the program's functionality with the finished embedded control system.

Key words: electric vehicle, model-based design, control system

SISÄLLYS

1	JOHDANTO	6
2	STARA eRETROFIT	7
3	TEORIA	8
	3.1 Sulautettu järjestelmä.....	8
	3.2 Ajoneuvon elektroninen ohjausyksikkö	9
	3.3 SAE J1939 CAN -kenttäväylä	10
	3.4 CCS-latauspistokejärjestelmä	11
	3.5 Latausaseman ja ajoneuvon välinen kommunikaatio	12
4	KÄYTETTY LAITTEISTO	15
	4.1 Pi Innovo OpenECU M580 -ohjausyksikkö	15
	4.2 Epec 6107 -kosketusnäyttö.....	16
	4.3 Danfoss EC-C1200-450 -moottorinohjausyksikkö.....	18
	4.4 Orion BMS 2 -akustonhallintajärjestelmä	19
	4.5 Sevcon HLVP10 -invertteri.....	19
	4.6 Phoenix Contact Vehicle Connector -latauspistorasia.....	20
	4.7 Omron G2R-1-SNI 120VAC (S) -tehorele	21
5	KÄYTETYT OHJELMISTOT	23
	5.1 Simulink R2018b -ohjelmointiympäristö	23
	5.2 Vector CANdb++ Editor -tietokantaohjelmisto.....	24
	5.3 Pi Snoop -diagnostiikkatyökalu	25
	5.4 CODESYS-ohjelmistoalusta.....	26
6	OHJELMAN SUUNNITTELU	28
	6.1 Väyläsuunnittelu.....	28
	6.2 Ohjelman jako alajärjestelmiin.....	29
	6.2.1 Latauksenohjaus	29
	6.2.2 Moottorinohjaus	30
	6.2.3 Ohjaustehostimen ja kompressorin invertterien ohjaus	32
	6.2.4 Lepotila.....	33
7	TOTEUTETTU OHJELMA	35
	7.1 Latauksenohjaus	38
	7.2 Moottorinohjaus.....	40
	7.3 Invertterien ohjaus.....	42
	7.4 Lepotila	43
8	OHJELMAN TESTAUS.....	45
9	POHDINTA	46
	LÄHTEET.....	48

LYHENTEET JA TERMIT

AC	Vaihtovirta
BMS	Battery Management System
CAN	Controller Area Network
CCP	CAN Calibration Protocol
CCS	Combined Charging System
DBC	Database CAN
DC	Tasavirta
ECU	Electronic Control Unit
H-silta	Elektroninen kytkentä, jota voidaan käyttää esimerkiksi tasavirtamoottorin ohjaukseen kahteen eri suuntaan
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
I/O	Tulo- ja lähtöviestit
kbps	kilobittiä per sekunti
Mbps	Megabittiä per sekunti
PT1000	Vastuslämpötila-anturi, jonka nimellisvastus on 1000 ohmia 0 °C lämpötilassa
PWM	Pulssinleveysmodulaatio
SAE	Society of Automotive Engineers
SOC	State of Charge
VCCU	Vehicle Charging Control Unit
VCU	Vehicle Control Unit

1 JOHDANTO

Opinnäytetyö tehtiin osana Tampereen ammattikorkeakoulun Helsingin kaupungin liikelaitos Staran ja Forum Virium Helsingin kanssa yhteistyössä toteuttamaa eRetrofit-hanketta, jonka tavoitteena on tutkia ja kehittää olemassa olevan ajoneuvokaluston sähköistämistä. Opinnäytetyön tarkoituksena oli tuottaa eRetrofit-hankkeessa dieselkuorma-autosta sähkökuorma-autoksi muunnettavan ajoneuvon sulautetun ohjausjärjestelmän toimintaa ohjaava ohjelma. Työn tavoitteena oli suunnitella ja toteuttaa toimiva ohjelma, jolla pystytään ohjaamaan hankkeessa tuotettavan sähkökuorma-auton ohjausjärjestelmän perustoimintoja.

Opinnäytetyössä ohjelmoitiin ajoneuvon sulautetun ohjausjärjestelmän ohjausyksikkönä toimivaa OpenECU M580-ohjausyksikköä, joka on suunniteltu käytettäväksi ajoneuvon ohjausyksikkönä ja ajoneuvon latauksenohjausyksikkönä kaikkein vaativimmissa sähkö- ja hybridihyötyajoneuvojen 24 V ohjausjärjestelmissä. Ohjausyksikkö kommunikoi ohjausjärjestelmään kuuluvien toimilaitteiden ja ajoneuvon oman kenttäväylän kanssa käyttäen SAE J1939 -standardin mukaista CAN-väylää. Ohjausyksikön ohjelmoiminen tapahtui MathWorksin kehittämällä graafisella ohjelmointiympäristöllä Simulinkilla. Ohjelmointi tapahtui toimilohkokaaviomuodossa hyödyntäen Simulinkin omia toimilohkoja sekä OpenECU-ohjausyksikön ohjelmointiin ja konfigurointiin luodun lisäkirjaston omia toimilohkoja. Opinnäytetyössä toteutetulla ohjelmalla ohjataan sähkökuorma-auton moottorinohjausta, latauksenohjausta, sekä kaikkia muunto-työssä ajoneuvon lisättäviä toimilaitteita.

Opinnäytetyön raportti koostuu työssä käytettyjen laitteiden ja ohjelmistojen esittelystä, työssä tuotetun ohjelman suunnittelusta, suunnitellun ohjelman toteutuksesta sekä toteutetun ohjelman testauksesta.

2 STARA eRETROFIT

Stara eRetrofit on Helsingin kaupungin liikelaitos Staran ja Forum Virium Helsingin Tampereen ammattikorkeakoulun kanssa yhteistyössä toteuttama hanke, jonka tarkoituksena on selvittää ajoneuvokaluston sähköistämisen vaikutukset operatiiviselle toiminnalle, todentaa raskaan ajoneuvon sähköistämisen kokonaiskustannukset elinkaaren aikana ja selvittää mahdollisuudet nopeuttaa sähköiseen ajoneuvokalustoon siirtymistä. Tampereen ammattikorkeakoulu toimii hankkeessa muutostyön teknisenä toteuttajana. Hankkeessa toteutettavassa muutostyössä hyödynnetään noin seitsemän vuotta käytössä ollutta kuorma-autoa, joka on varusteltu tavanomaisin työlaittein. Ajoneuvon voimalinja on hankkeen aikana vaihdettu sähköiseksi ja sen ohjausta varten on tämän opinnäytetyön aikana tehty tarvittava ohjelmointityö. Muunnettu ajoneuvo tullaan ottamaan normaaliin päivittäiseen käyttöön. (hel.fi).

Raskaan liikenteen sähköistäminen tulee kestävämpään pidempään kuin henkilöautojen sähköistäminen, mikä johtuu muun muassa ajoneuvokannan ja työkoneiden pitkästä käyttöiästä, uuden sähköisen kaluston kalleudesta sekä kaluston huonosta saatavuudesta. Kaluston luonnollisesta uusiutumisesta johtuva muutos tulee näiden syiden seurauksena olemaan hidasta, mikä on ongelma erityisesti Helsingin kaupungin Hiilineutraali Helsinki 2035 -ohjelman näkökulmasta, koska liikenteen sähköistymisellä on suuri rooli ohjelman tavoitteiden täyttymisessä. Muutosta fossiilisten polttoaineiden ajoneuvoista sähköisiin ajoneuvoihin voitaisiin nopeuttaa muuntamalla nykyisiä käytössä olevia fossiilisten polttoaineiden ajoneuvoja sähkökäyttöisiksi. (hel.fi)

3 TEORIA

Opinnäytetyössä toteutettu ohjelma tehtiin ohjaamaan elektronista ohjausyksikköä, joka toimii ajoneuvon sulautetun ohjausjärjestelmän ohjauslaitteena.

Työssä toteutetun sulautetun järjestelmän toimilaitteet kommunikoivat keskenään SAE J1939 CAN -kenttäväylien välityksellä. Työssä perehdyttiin myös sähköajoneuvon lataussekvenssiin ja ajoneuvon lataamiseen CCS-latauspistorasian välityksellä.

3.1 Sulautettu järjestelmä

Sulautetulla järjestelmällä voidaan määritellä tarkoitettavaksi laitetta, joka sisältää elektroniikkaa, mekaniikkaa sekä laitteen toimintaa ohjaavaa ohjelmistoa. Sulautetuksi järjestelmäksi luokiteltava laite on suunniteltu tietyn ennalta määritellyn toiminnon suorittamiseen, sitä ei ole tarkoitettu käyttäjän itsenäisesti ohjelmoitavaksi ja sen odotetaan toimivan vähäisellä tai kokonaan ilman ihmisen vuorovaikutusta. Sulautetuille järjestelmille tyypillisiä ominaisuuksia ovat myös niiden reaktiivinen toiminta ja vahvasti rajoitettu toiminnallisuus. (Jiménez, Palomera & Couvertier, 2013.)

Sulautetun järjestelmän tietokone tai muu ohjelmistoa pyörittävä ohjausyksikkö on normaalisti omistettu tietyn tehtävän hoitamiseksi. Sulautettua järjestelmää ohjaa tyypillisesti vain yksi ohjelma, joka on pysyvästi käynnissä järjestelmän sisältävän laitteen ollessa käytössä. Sulautetun järjestelmän etuna on se, että järjestelmän toiminnallisuuden määrittelee sitä ohjaava ohjelmisto, eikä sen sisältämä laitteisto. (Catsoulis, 2005.)

Ajoneuvoissa yhä useammat varusteet ovat kehittymässä mekaanisista järjestelmistä sähköisiksi. Sulautettu järjestelmä on joustavuutensa ansiosta ajoneuvon sähköisten järjestelmien ydin. Tyypillisessä modernissa ajoneuvossa sulautetuilla järjestelmillä voidaan ohjata esimerkiksi turvatyynyjen, lukkiutumattoman jarrutuksen, mukautuvan vakionopeudensäätimen, luistoneston, automaatt-

tisen pysäköinnin, navigaatiojärjestelmän sekä ilmastoinnin toimintaa. Ajoneuvoissa sulautettuja järjestelmiä ohjataan yleensä runkoon kiinnitettävillä elektronisilla ohjausyksiköillä. (WatElectronics, 2021.)

3.2 Ajoneuvon elektroninen ohjausyksikkö

Elektroninen ohjausyksikkö (ECU) on ajoneuvon runkoon kiinnitettävä mikrokontrollerin sisältävä laite, jota käytetään ajoneuvon erilaisten sähköisten järjestelmien ohjaamiseen. Tyypillisessä modernissa ajoneuvossa ohjausyksiköillä saatetaan ohjata esimerkiksi moottorin, ohjaustehostimen sekä turvatoimintojen, kuten ilmatyynyjen tai automaattisen hätäjarrituksen toimintaa. Ohjausyksikön tyypillisesti sisältämä mikrokontrolleri pyörittää omaa ohjelmistoaan ja tarvitsee toimiakseen virtalähteen ja datayhteyksiä. (Aptiv, 2020.)

Ohjausyksikkö vastaanottaa toimintatarkoituksestaan riippuen tietoa ajoneuvon eri osissa olevilta antureilta tai muilta mittauslaitteilta ja ohjaa saamansa tiedon perusteella ajoneuvon toimilaitteita ohjausyksikölle ladatun ohjelman määrittelemällä tavalla. Esimerkiksi automaattisesta hätäjarruksesta vastaava ohjausyksikkö saattaisi vastaanottaa tietoa ajoneuvon keulassa olevalta tutkalta, mitä käyttämällä ohjausyksikön ohjelma voisi tunnistaa ajoneuvon lähestyvän edessä olevaa estettä liian nopeasti ja aktivoida jarrituksen estääkseen törmäyksen. (Aptiv, 2020.)

Polttomoottoriajoneuvoissa yksi tärkeimmistä elektronisen ohjausyksikön käyttökohteista on moottorinohjausyksikkö. Moottorinohjausyksikkö ohjaa tyypillisesti moottoriin ruiskutettavan polttoaineseoksen säätöä, sytytyksen ajoitusta, tyhjäkäynnin säätöä sekä muuttuvaa venttiilienajoitusta. Ohjausyksikön tavoitteena on edellä mainittuja toimintoja ohjaamalla optimoida moottorin käyttö mahdollisimman tehokkaaksi, mikä tekee moottorista sekä ekologisemmän että ekonomisemmän. (ECU Testing, 2021.)

Sähkömoottoriajoneuvoissa moottorinohjausyksikön toiminta poikkeaa polttomoottoriajoneuvoista huomattavasti, sillä edellä mainittujen toimintojen sijaan ohjausyksikön tarvitsee huolehtia vain moottorin tuottamasta tehosta ja sen

pyörimisnopeudesta, sekä mahdollisesta regeneraatiosta. Sähköajoneuvoissa moottorinohjausyksikkönä toimii ECU:n sijaan yleensä moottoria ohjaava invertteri. Sähköajoneuvoissa moottorinohjauksen lisäksi perinteisistä ajoneuvoista poikkeava käyttötarkoitus elektroniselle ohjausyksikölle on ajoneuvon latauksen ohjaus. Latausta ohjaava ohjausyksikkö välittää tietoa ajoneuvon akustohallintajärjestelmän ja latausaseman välillä, sekä ohjaa latausprosessia tarvittaessa. (Aptiv, 2020.)

3.3 SAE J1939 CAN -kenttäväylä

J1939 on ajoneuvoalan standardisointijärjestö SAE:n (Society of Automotive Engineers) määrittelemä CAN-väylän standardisetti, joka on kehitetty käytettäväksi raskaan käytön ajoneuvoissa, kuten kuorma- ja linja-autoissa. J1939 mahdollistaa sarjaliikennekommunikaation sähköisten ohjausyksiköiden välillä kaikentyyppisissä raskaissa ajoneuvoissa. Standardia käyttävien väylien tiedonsiirtonopeus on joko 250 tai 500 kbps. (Voss, 2008.)

CAN-väylän tärkeimmät edut kenttäväylänä ovat vähäisempi tarvittavien johtojen määrä, erittäin luotettava kommunikaatio, helppo toteutettavuus ja parempi huollettavuus, mitkä yhdessä tuottavat ajoneuvon paremman suorituskyvyn sekä auttavat pienentämään tuotantokuluja. J1939-pohjaisia kommunikaatioprotokollia käytetään esimerkiksi dieselmoottoriratkaisuissa, kuorma-autojen ja linja-autojen sisäisissä verkoissa, maa- ja metsätalouskoneissa sekä sotilasajoneuvoissa. (Voss, 2008.)

CAN DBC-tiedosto (CAN-tietokanta) on tekstitiedosto, joka sisältää informaatiota CAN-väylän raakan datan tulkitsemiseen fyysisiin arvoihin. DBC-tiedosto sisältää dekodeaus säännöt, joita hyödyntämällä raaka data muunnetaan fyysisiksi parametreiksi ja niiden arvoiksi. (ccselectronics.com, 2021.)

3.4 CCS-latauspistokejärjestelmä

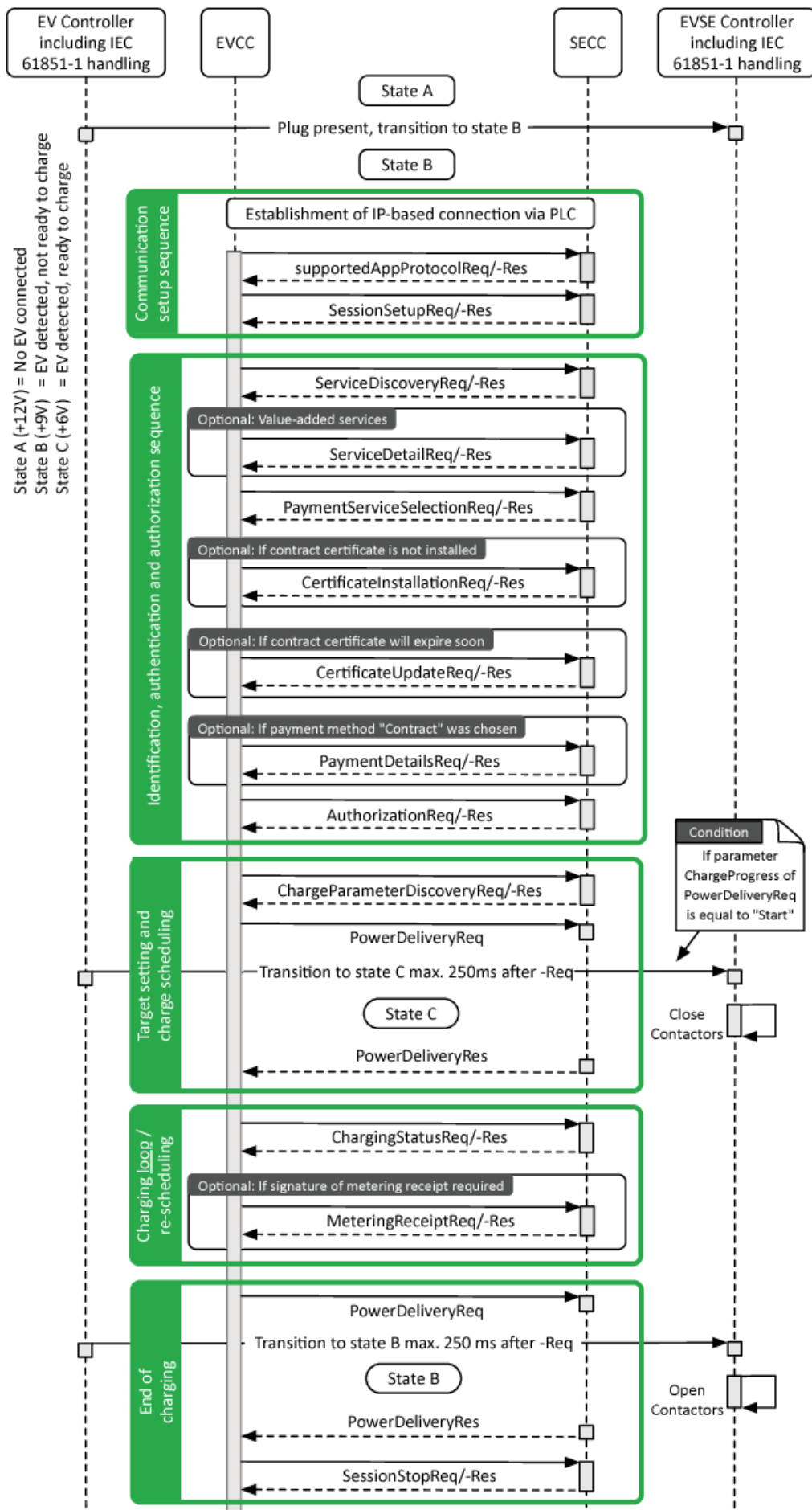
Combined Charging System (CCS) on yhdistetty latauspistokejärjestelmä sähköajoneuvojen lataamiseen vaihtovirralla (AC) ja tasavirralla (DC). CCS-latauspistokejärjestelmällä varustetun sähköajoneuvon latausliittimeen voidaan kytkeä sekä AC- että DC-latauspistokkeet, minkä ansiosta ajoneuvossa tarvitaan vain yksi latausliitin. Perinteistä vaihtovirtalatausta voidaan käyttää, kun ajoneuvoa ladataan esimerkiksi yön yli autotallissa, kun taas DC-pikalatauksella mahdollistetaan ajoneuvon lataaminen muutamissa minuuteissa esimerkiksi huoltoasemalla. Combined Charging System on standardoitu IEC 62196-3:n ja SAE J1772:n mukaan. (Phoenix Contact, 2021.) CCS-latauspistokejärjestelmä laajentaa tämänhetkisiä alueellisia ratkaisuja kohti yhtä globaalia integroitua järjestelmää. CCS edustaa tulevaisuuden nopeaa lataamista ja maksimoi sähköajoneuvojen integrointia tulevaisuuden älykkäisiin sähköverkkoihin. (Kübel, 2015.)

Standardien mukaisesti ajoneuvon CCS-liittimet tulee varustaa sähkömekaanisella lukituslaitteella, joka lukitsee latauspistoolin kiinni ajoneuvon latauspistokeeseen latauksen ajaksi sivulta tai suoraan pistokkeessa olevalla salvalla. Lukitussalpa on suunniteltu kestämaan suuria vetovoimia, mikä estää latauspistoolin irrottamisen kesken latauksen. Toinen latausprosessin turvallisuutta lisäävä tekijä on lämpötilan valvonta. Jos järjestelmä ylikuumenee esimerkiksi korkeiden ulkolämpötilojen tai ylikuormituksen vuoksi, PT1000-vastusanturit havaitsevat korkeat lämpötilat ja latausasema voi siten ylikuumentapauksessa keskeyttää latauksen tai laskea lataustehoa. IEC 62196:n mukaan latauslämpötila ei saa ylittää 105 °C. (Phoenix Contact, 2021.)

3.5 Latausaseman ja ajoneuvon välinen kommunikaatio

Sähköajoneuvon latausprosessi etenee ISO 15518 -standardin mukaisesti tiettyjä vaihteita noudattaen. Vaiheet voidaan jakaa neljään päävaiheeseen, jotka ovat latausession initialisointi, eristyksen tarkkailu ja esilataus, energian siirto sekä latauksen lopetus ja latausasemasta irrottautuminen. Kommunikaatio latausaseman ja ajoneuvon välillä tapahtuu latauspistokkeessa sijaitsevan Control Pilot -pinnin välityksellä PWM-signaalina. Control Pilot -pinnin ja latausaseman välinen jännite muuttuu myös lataussekvenssin aikana tilasta B tilaan C. (Kübel, 2015.) Lataussekvenssin vaiheet päävaiheeseen ja Control Pilotin tiloihin on esitelty kuvassa 1.

Control Pilot -pinnin jännite on tilassa A, eli ennen latauspistokkeen kiinnittämistä latauspistorasiaan +12 V. Latauspistokkeen kiinnittämisen jälkeen pinnin jännite siirtyy tilaan B, jossa sen jännite on +9 V. Latauksen initialisoinnin ja latauksen aloittamisen välissä pinnin jännite siirtyy tilaan C, jossa sen jännite on +6 V. Yksinkertaistetusti Control Pilot -pinnin jännite on tilassa A, kun latauspistoketta ei ole havaittu, tilassa B, kun latauspistoke on havaittu, mutta ajoneuvo ei ole valmis lataamaan, ja tilassa C, kun latauspistoke on havaittu ja ajoneuvo on valmis lataamaan. (Mültin, 2018.)



KUVA 1. Viestien järjestys latausession aikana. (Mültin, 2018.)

Työssä käytetyn ohjausyksikön oman toimilohkokirjaston sisäänrakennetut lataussekvenssin viestit poikkesivat hieman kuvassa 1 esitetyistä viesteistä. Kaikki lataussekvenssin viestit koostuvat kahdesta osasta, jotka ovat ajoneuvon lataus-
 asemalle lähettämä pyyntöviesti (-Req) ja latausaseman ajoneuville lähettämä
 vastausviesti (-Res). Pyyntöviestin numero- tai kirjaintunnuksen perään lisätään
 a ja vastausviestin tunnuksen perään lisätään puolestaan b. Esimerkiksi ajoneu-
 von latausasemalle lähettämä lataussekvenssin valmistelun pyyntöviesti Session-
 SetupRes on siten tunnukseltaan 1a ja latausaseman ajoneuville lähettämä vas-
 tausviesti SessionSetupReq on tunnukseltaan 1b. Työssä käytetyt lataussek-
 venssin viestit on esitelty tunnuksineen ja Control Pilot -pinnin tiloineen taulu-
 kossa 1.

TAULUKKO 1. Lataussekvenssin viestit ja niiden tunnuksiset

Viesti	Tunnus	CP-tila
SupportedAppProtocol	0	B
SessionSetup	1	B
ServiceDiscovery	2	B
ServicePaymentSelection	X	B
ContractAuthentication	Y	B
ChargeParameterDiscovery	3	B
CableCheck	4	C
Pre-Charge	5	C
PowerDelivery (latauksen alussa)	6	C
CurrentDemand	7	C
PowerDelivery (latauksen lopussa)	8	C
WeldingDetection	9	B
SessionStop	10	B

4 KÄYTETTY LAITTEISTO

Opinnäytetyössä käsitelty ajoneuvon sulautettu ohjausjärjestelmä koostui järjestelmää ohjaavasta elektronisesta ohjausyksiköstä sekä järjestelmän osana toimivista toimilaitteista. Työssä käsiteltäviä toimilaitteita olivat ajoneuvon kuljettajan operointipaneelina toimiva Epec 6107 -kosketusnäyttö, ajoneuvon sähkömoottorin ohjausyksikkönä käytettävä Danfoss EC-C1200-450 -invertteri, Orion BMS 2 -akustonhallintajärjestelmä, Sevcon HVLP10 -invertterit, Phoenix Contact Vehicle Connector -latauspistorasia sekä Omron G2R-1-SNI 120VAC (S) -teherele.

4.1 Pi Innovo OpenECU M580 -ohjausyksikkö

OpenECU M580 on Pi Innovon valmistama elektroninen ohjausyksikkö, joka on suunniteltu käytettäväksi ajoneuvon ohjausyksikkönä (VCU) ja ajoneuvon latauksenohjausyksikkönä (VCCU) kaikkein vaativimmissa sähkö- ja hybridihyötyajoneuvojen 24 V ohjausjärjestelmissä. Ohjausyksiköllä on 4 erillistä CAN-väylää, mikä mahdollistaa kommunikoinnin useiden muiden ajoneuvon ohjaukseen tarvittavien laitteiden kanssa. CAN-kanavien lisäksi sillä on 112 I/O-pinniä, joita voidaan käyttää esimerkiksi erilaisten antureiden jänniteviestien lukemiseen sekä jänniteohjausviestien lähettämiseen toimilaitteille, joita ei ohjata CAN-väylien kautta. Tavallisten I/O-pinnien lisäksi ohjausyksiköllä on myös esimerkiksi useita H-siltapinnejä, jotka mahdollistavat toimilaitteiden ohjauksen 24 V jänniteviesteillä, sekä integroidun CCS-latausjärjestelmän rajapinnan kommunikointipinnit, jotka mahdollistavat standardien mukaisen kommunikoinnin sähköajoneuvon latausaseman kanssa. Ohjausyksikön käyttöjännitteen voimakkuus on 24 V, mikä mahdollistaa käyttöjännitteen helpon syötön 24 V jännitettä käyttävän ajoneuvon pääkäyttöjännitteestä. (Pi Innovo, 2021.) Kuvassa 2 on esitetty valokuva työssä käytetystä M580-ohjausyksiköstä.



KUVA 2. OpenECU M580 -ohjausyksikkö

Ohjausyksikön ohjelmointi toteutettiin Simulinkilla hyödyntäen Pi Innovon omaa OpenECU-lisäkirjastoa, joka sisältää kaikki ohjausyksikön ja sen CAN-väylien konfigurointiin, sisään- ja ulostulopinnien ohjaukseen, CAN-viestiliikenteen ohjaukseen sekä lataussekvenssin kommunikaation tarvittavat toimilohkot. Lisäkirjaston toimilohkoja käytettiin yhdessä Simulinkin omien toimilohkojen kanssa mallipohjaisen ohjelman luomiseksi. Toimilohkoilla toteutettu valmis mallipohjainen ohjelma käännettiin lopuksi C-koodiksi Simulink Coder -lisäosalla. C-koodipohjainen ohjelma syötettiin ohjausyksikölle CAN-väylää käyttäen, minkä jälkeen ohjausyksikkö toimi uudelleen käynnistämisen jälkeen sille syötetyn ohjelman mukaisesti. (Pi Innovo, 2021.)

4.2 Epec 6107 -kosketusnäyttö

Epec 6107 on suomalaisen Ponsse Groupiin kuuluvan Epec Oy:n valmistama kosketusnäyttö, joka on tarkoitettu käytettäväksi vaativimmissa raskaissa ajoneuvoissa, kuten kaivinkoneissa, maatalouskoneissa ja pyöräkuormaajissa. Kosketusnäyttö on käyttäjän avoimesti ohjelmoitavissa CODESYS 3.5:n välityk-

sellä, joten se soveltuu erinomaisesti käytettäväksi operointipaneelina liikkuvissa koneissa. Näyttö sisältää kaksi CAN-väylän rajapintaa, jotka sallivat näytön yhdistämisen korkeamman tason tietojärjestelmiin. (Epec, 2021.)

Kosketusnäyttöä käytettiin opinnäytetyössä sähkökuorma-auton kuljettajan operointipaneelina, joka sijoitettiin sähkökuorma-auton hytin kojelaudalle. Näytön lopullisena käyttötarkoituksena on ajodatan esittäminen ajoneuvon kuljettajalle, mahdollisten ominaisuuksien, kuten sähkömoottorin regeneraation ohjaaminen, sekä ajoneuvon diagnostiikan lukeminen ajoneuvon huoltojen ja kehitystyön yhteydessä. Opinnäytetyön tutkimustyön aikana kosketusnäyttöä käytettiin lisäksi monipuolisesti ajoneuvon kehitykseen ja testaukseen. Näyttöä käytettiin erityisesti moottorinohjauksen demonstraatioon, HVLP10-inverttereiden ohjaamiseen päälle ja pois niiden testien aikana, sekä vaihteen valitsemiseen ennen vaihdetiedon lukemista ajoneuvon omasta CAN-väylästä. Kosketusnäyttö kommunikoi ohjausyksikön kanssa CAN-väylän välityksellä.



KUVA 3. Epec 6107 -kosketusnäyttö (epec.fi, 2021)

4.3 Danfoss EC-C1200-450 -moottorinohjausyksikkö

Danfoss EC-C1200-450 on sähkömoottorin ohjaukseen tarkoitettu invertteri, joka on suunniteltu erityisesti ohjaamaan hybridi- ja sähkövoiman virtausta ajoneuvo-, kone- ja merenkäyntisovelluksissa. Invertteri kestää tukevan muotoilunsa ansiosta hyvin korkean tason mekaanista tärinää ja iskuja, minkä ansiosta se sopii hyvin käytettäväksi raskaissa työkoneissa. (danfoss.com, 2021.) Invertteri tukee SAE J1939 -kommunikaatioprotokollaa nopeuksilla 100, 125, 250, 500 ja 1000 kbps. Sen ohjaaminen tapahtuu CAN-viestien välityksellä hyödyntäen valmistajan omaa DBC-tietokantaa. (Danfoss, 2019.)

Työssä invertteriä käytettiin ajoneuvon sähkömoottorin ohjausyksikkönä. M580-ohjausyksikkö ohjaa invertteriä lähettämällä sille CAN-väylää pitkin valmistajan DBC-tietokannan mukaisia viestejä. Invertteriä voidaan ohjata monilla eri tiloilla, joita ovat moottorin pyörimisnopeuden ohjaus, moottorin väännön ohjaus, moottorin tehon ohjaus sekä invertterin lähtötaajuuden ohjaus. (Danfoss, 2019.) Moottorin ohjaustilana käytettiin tässä työssä moottorin tehon ohjausta, sillä sen todettiin olevan paras tapa saada sähkömoottorin toiminta vastaamaan lähimminkin ajoneuvon alkuperäisen dieselmootorin toimintaa.



KUVA 4. Danfoss EC-C1200-450-sähkömoottorinohjain (danfoss.com, 2021)

4.4 Orion BMS 2 -akustonhallintajärjestelmä

Orion BMS 2 on Ewert Energy Systems Inc:n tuottama akustonhallintajärjestelmä, jolla pystytään hallitsemaan jopa 180 sarjassa olevaa akkua. Järjestelmä tarkkailee jokaisen sarjassa olevan akun jännitettä ja lataustilaa, hallitsee akkujen jännite-, virta- ja lämpötilarajoja sekä tasapainottaa akuston akkujen lataustasoja. Järjestelmä on yhteensopiva lähes kaikkien litiumioniakkujen kanssa. Järjestelmä sisältää kaksi ohjelmoitavaa CAN-väylän rajapintaa, jotka tukevat SAE J1939 -protokollaa. Järjestelmän yleisiä käyttösovelluksia ovat sähköajoneuvot, hybridi-ajoneuvot, aurinko- ja tuulienergian varastointi sekä tutkimus- ja laboratoriotestaus. (orionbms.com, 2019.)

Työssä käytettiin kahta Orion BMS 2 -akustonhallintajärjestelmää kahden erillisen akkupaketin hallintaan. Akkupakettien hallinta toteutettiin Master-Slave -rakenteella, jossa Master-järjestelmää ohjataan M580-ohjausyksiköllä ja Slave-järjestelmää ohjataan Master-järjestelmällä. M580-ohjausyksikkö kommunikoi Master-järjestelmän kanssa CAN-väylän välityksellä laitevalmistajan oman DBC-tietokannan mukaisia viestejä hyödyntäen. Tärkeimmät akustolta saatavat tiedot ovat akuston lataustila (SOC), akuston maksimijänniteraja, akuston purkuvirtaraja (DCL), akuston latausvirtaraja (CCL) sekä akuston lataustehoraja (CCL KW).

4.5 Sevcon HLVP10 -invertteri

HVLP10 on Sevconin valmistama ilmajäähdytteinen invertteri, joka on suunniteltu ohjaamaan kolmivaiheisia vaihtovirtamoottoreita akulla tai generaattorilla toimivissa sovelluksissa. Invertteri on suunniteltu käytettäväksi monipuolisissa sovelluksissa apumoottorien ohjaukseen esimerkiksi maataloudessa, kevyissä kuorma-autoissa sekä linja-autoissa. Invertteriä ohjataan valmistajan DBC-tietokannan mukaisilla SAE J1939 -protokollan CAN-viesteillä. (hsct.nl, 2015.)

Opinnäytetyössä käytettiin kahta HVLP10-invertteriä ajoneuvon ohjaustehostimen sekä sen hydrauliiikan kompressorin ohjaamiseen. Inverttereitä ohjataan M580-ohjausyksikön lähettämällä CAN-viesteillä. Viesteillä asetetaan niille

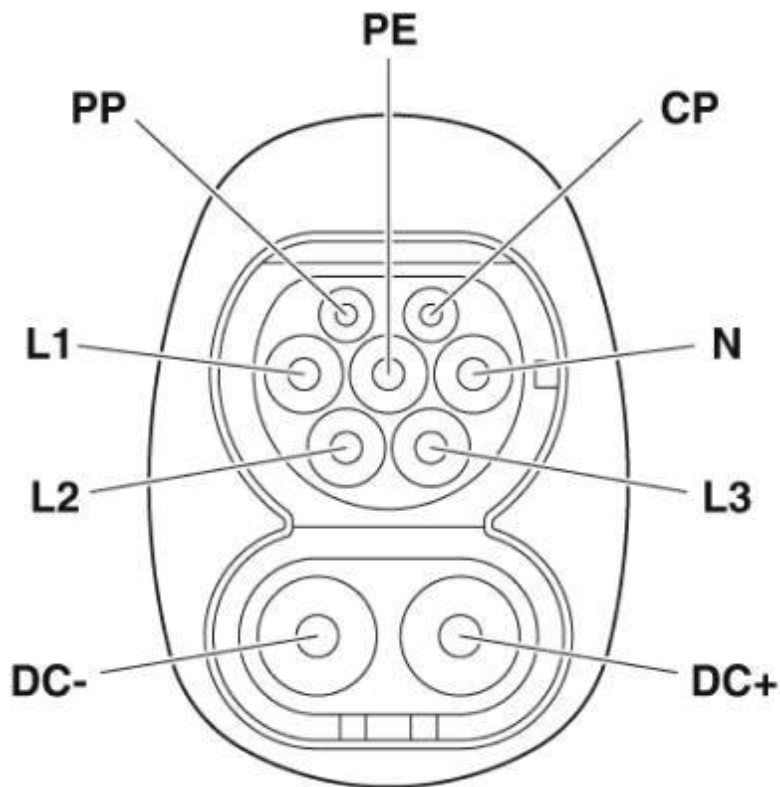
kompressorin ja ohjaustehostimen vaatimat raja-arvot parametreille, kuten pyörimisnopeudelle ja maksimivirroille, ja ne ohjataan ajoneuvon ollessa käynnissä haluttuihin tiloihin lähettämällä valmistajan määrittelemiä Control Wordjeja.



KUVA 5. Sevcon HVLP-10 -invertteri (hsct.nl, 2015)

4.6 Phoenix Contact Vehicle Connector -latauspistorasia

Phoenix Contactin valmistama Vehicle Connector on laboratoriotesteihin tarkoitettu CCS-latausjärjestelmää tukeva latauspistorasia, jota käytetään työssä ajoneuvon latauksenohjausjärjestelmän testaamiseen. Latauspistorasia on varustettu latauspistoolin latauksen ajaksi paikalleen lukitsevalla lukitusmoottorilla, jota ohjataan $\pm 24V$ jänniteviesteillä. Latauspistorasian pinnit on esitetty kuvassa 6.



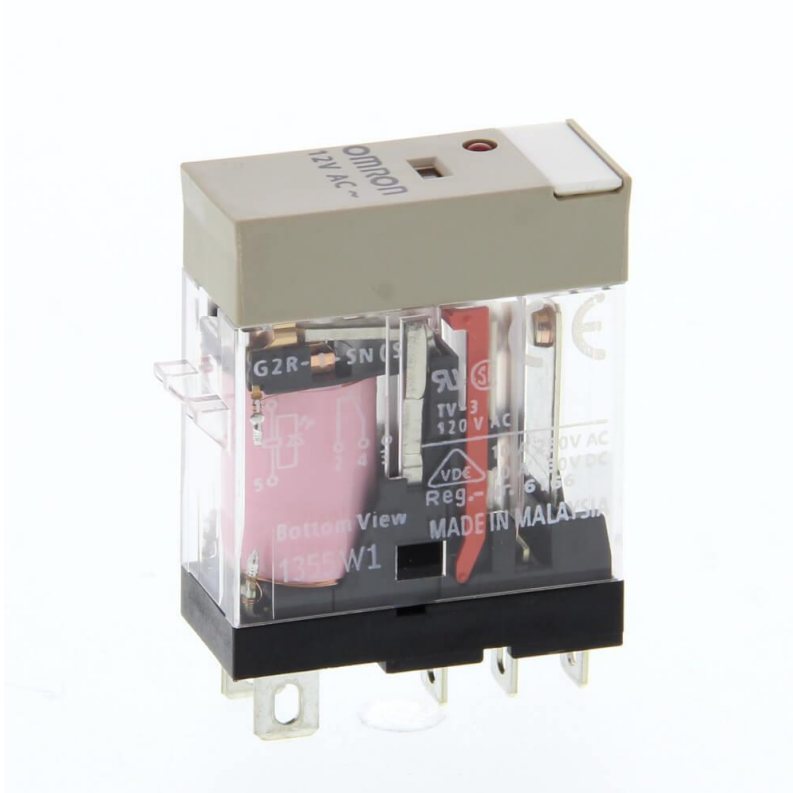
KUVA 6. Latauspistorasian pinnit (phoenixcontact.com, 2021)

Latauspistorasia sisältää CCS-latausstandardin mukaisesti latauspinnit sekä vaihtovirtalataukselle (L1, L2, L3, N) että tasavirtapikalataukselle (DC-, DC+). Muut pistorasian pinnit ovat suojamaadoituspinni (PE), Proximity Pin (PP) sekä Control Pilot (CP). Proximity Pin on tarkoitettu latauspistoolin kunnollisen kiinnityksen toteamiseksi latauksenohjaussovelluksessa ja Control Pilot on tarkoitettu ajoneuvon latauksenohjausyksikön ja latausaseman väliseen kommunikaatioon. (phoenixcontact.com, 2021.)

4.7 Omron G2R-1-SNI 120VAC (S) -tehorele

Omron G2R-1-SNI 120VAC (S) on Omronin Industrial Automationin valmistama tehorele, jota käytettiin työssä Control Pilotin jännitteen muuttamiseen latausprosessin aikana. Tehoreleen avulla M580-ohjausyksikön ja latauspistorasian välinen Control Pilot -piiri ohjataan kulkemaan eri vastusten kautta Control Pilotin jännitetilan siirtyessä lataussekvenssin aikana tilasta B tilaan C. Piirin ohjaa-

minen matalamman resistanssin vastusten kautta laskee Control Pilotilta luettavaa jännitettä, minkä ansiosta latausasema tulkitsee sen olevan tilassa C, mikä mahdollistaa lataussekvenssin siirtymisen eteenpäin.



KUVA 7. Omron G2R-1-SN 120VAC (S) -teherele (industrial.omron.fi)

5 KÄYTETYT OHJELMISTOT

Opinnäytetyön aikana suoritettuun ohjelmointityöhön käytettiin useita eri ohjelmistoja. Työn ensisijaisena ohjelmointityökaluna käytettiin MathWorksin kehittämää Simulink-ohjelmointiympäristöä, jota käytettiin ohjausjärjestelmän elektronisen ohjausyksikön ohjelmointiin. Työssä käytettiin lisäksi DBC-tietokantojen käsittelyyn Vector CANdb++ Editor -tietokantaohjelmistoa, testien aikaiseen diagnostiikkaan Pi Snoop -työkalua ja ajoneuvon kuljettajan operointipaneelin ohjelmointiin CODESYS-ohjelmistoa.

5.1 Simulink R2018b -ohjelmointiympäristö

Simulink on MathWorksin kehittämä MATLAB-pohjainen graafinen ohjelmointiympäristö, joka on tarkoitettu mallipohjaiseen ohjelmointiin (MBD) sekä mallien simulointiin. Simulink-mallit luodaan ensisijaisesti käyttämällä ohjelmointiympäristön lohkokaaviotyökalua ja sen mukautettavia lohkokaaviokirjastoja. Ohjelmointiympäristö sisältää laajan valikoiman ennalta määriteltyjä toimilohkoja, joiden avulla käyttäjä pystyy luomaan malleja nopeasti. Ennalta määriteltyjen toimilohkojen lisäksi käyttäjä pystyy lataamaan ohjelmointiympäristöön erilaisia lisäkirjastoja, joiden toimilohkot ovat yhteensopivia ennalta määriteltyjen toimilohkojen kanssa. Simulointiominaisuuden ansiosta käyttäjä pystyy testaamaan luomiaan malleja ilman tarvetta rakentaa fyysisiä malleja laboratoriossa, mikä mahdollistaa nopean kokeilupohjaisen testaamisen mallin ohjelmoimisen lomassa. Koska Simulink on integroitu MATLABIN kanssa, sen data on helposti siirrettävissä ohjelmien välillä. (mathworks.com, 2021.)

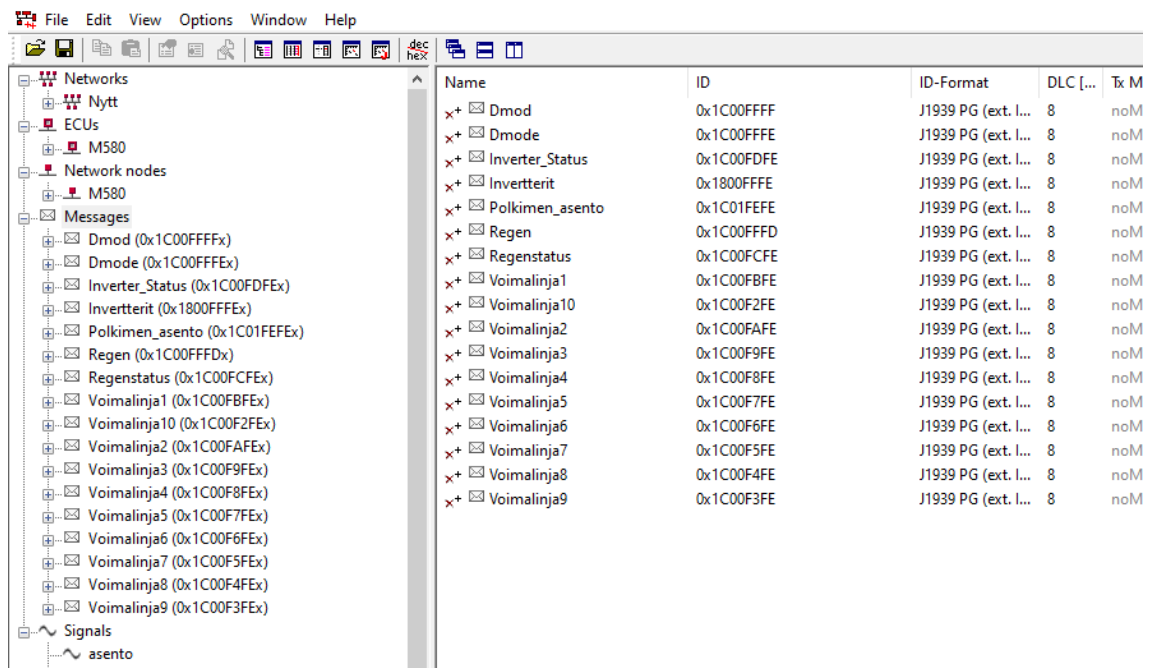
Simulink Coder on Simulinkin lisäosa, jonka avulla Simulinkilla luoduista malleista pystytään generoimaan C- ja C++-koodia. Lisätyökalulla generoitua koodia pystytään käyttämään reaaliaikaisessa simulaatiossa ja testaamisessa tai se voidaan ajaa sisään sulautetun järjestelmän ohjausyksikölle järjestelmää hallitsevaksi ohjelmaksi. (mathworks.com, 2021.)

Tässä opinnäytetyössä Simulinkia käytettiin pääasiallisena ohjelmointityökaluna. M580-ohjausyksikön ohjelmointi tapahtui toimilohkokaaviomuodossa hyödyntäen Simulinkin omia toimilohkoja sekä Pi Innovon OpenECU-lisäkirjaston omia toimilohkoja. Ohjausyksikölle syötettävän C-koodin generointi mallipohjaisesta ohjelmasta tapahtui Simulink Coderilla. Koodin generointi vaati lisäksi aktiivisen Wind River (Diab) C compiler -lisenssin. Mallista generoitu C-koodi ajettiin sisään ohjausyksikölle CAN-väylän välistyksellä PiSnoop-ohjelmalla.

5.2 Vector CANdb++ Editor -tietokantaohjelmisto

CANdb++ Editor on Vectorin DBC-tietokantojen luomiseen, vertailuun ja hallintaan kehittämä ohjelmisto. Ohjelmiston toiminnallisuus on räätälöity tukemaan SAE J1939 CAN -kommunikaatioprotokollaa. Ohjelmiston Editor-versio on tarkoitettu ensisijaisesti olemassa olevien DBC-tietokantojen analysoimiseen ja soveltamiseen, mutta pienempien tietokantojen luominen on mahdollista myös kyseisellä versiolla. (vector.com, 2021.)

Tässä opinnäytetyössä Vector CANdb++ Editoria käytettiin työssä luotavan ajoneuvon sulautetun ohjausjärjestelmän eri toimilaitteiden valmistajien luomien DBC-tietokantojen hallintaan sekä Epec 6107 -kosketusnäyttöä ohjaavan viestiliikenteen ja ajoneuvon omasta CAN-väylästä luettavien viestien DBC-tietokantojen luomiseen. Toimilaitteiden valmistajien luomia DBC-tietokantoja käytettiin moottorinohjausjärjestelmän, akustonhallintajärjestelmän sekä ohjausvahvistimen ja hydrauliiikan kompressorin invertterien ohjaamiseen. Kuvassa 8 on esitelty näyttöleike eräästä opinnäytetyössä luodusta DBC-tietokannasta.

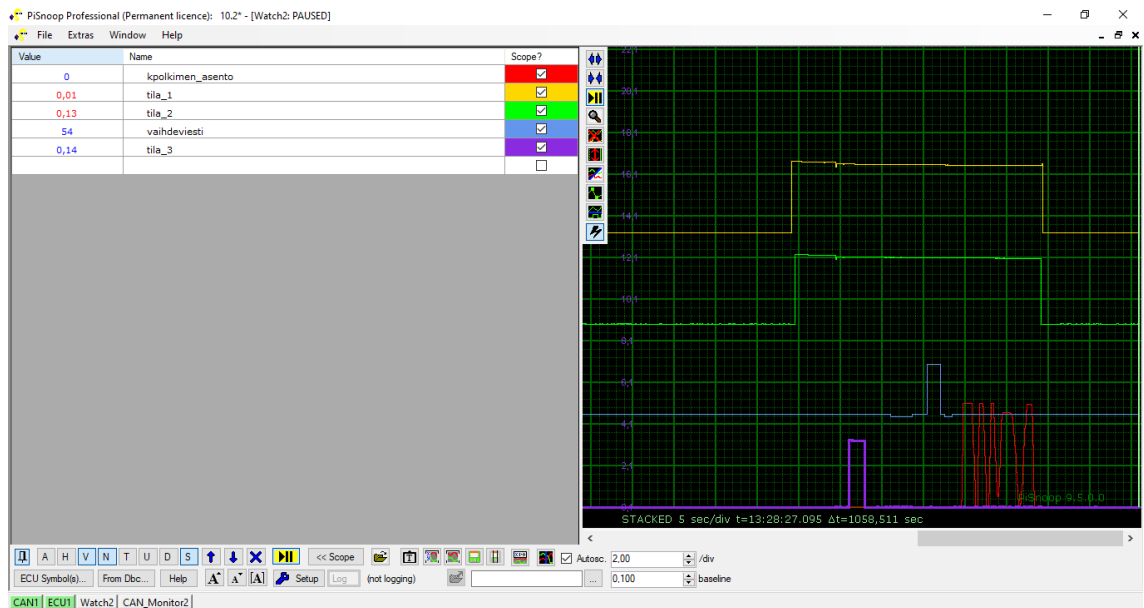


KUVA 8. Näyttöleike Vector CANdb++ Editorilla luodusta DBC-tietokannasta

5.3 Pi Snoop -diagnostiikkatyökalu

Pi Snoop on Pi Innovon kehittämä työkalu, jota voidaan käyttää sulautettujen järjestelmien kalibrointiin, diagnostiikkaan sekä datankeräämiseen. Pi Snoop sallii käyttäjän olla vuorovaikutuksessa tutkittavan ohjelman kanssa ohjelman ollessa käynnissä sulautetussa järjestelmässä tai elektronisessa ohjausyksikössä. CAN-väylän välityksellä työkalu pääsee käsiksi elektronisen ohjausyksikön muuttujiin ja muistiin reaaliaikaisesti. Pi Snoopia voidaan käyttää myös elektronisen ohjausyksikön uudelleenohjelmointiin joko lataamalla ohjausyksikölle kokonaan uusi ohjelma, tai osia siitä. (openecu.com, 2021.)

Tässä opinnäytetyössä Pi Snoopia käytettiin opinnäytetyössä tuotetun ohjelman lataamiseen sisään M580-ohjausyksikölle CAN-väylän välityksellä sekä datan lukemiseen ajoneuvon sulautetun järjestelmän CAN-väylistä. Datat lukemista käytettiin työn aikana erityisesti testattavien ominaisuuksien toiminnan toteamiseksi sekä toimintojen simuloimiseksi. Kuvassa 9 on esitetty näyttöleike Pi Snoopista ohjelman testauksen aikana.

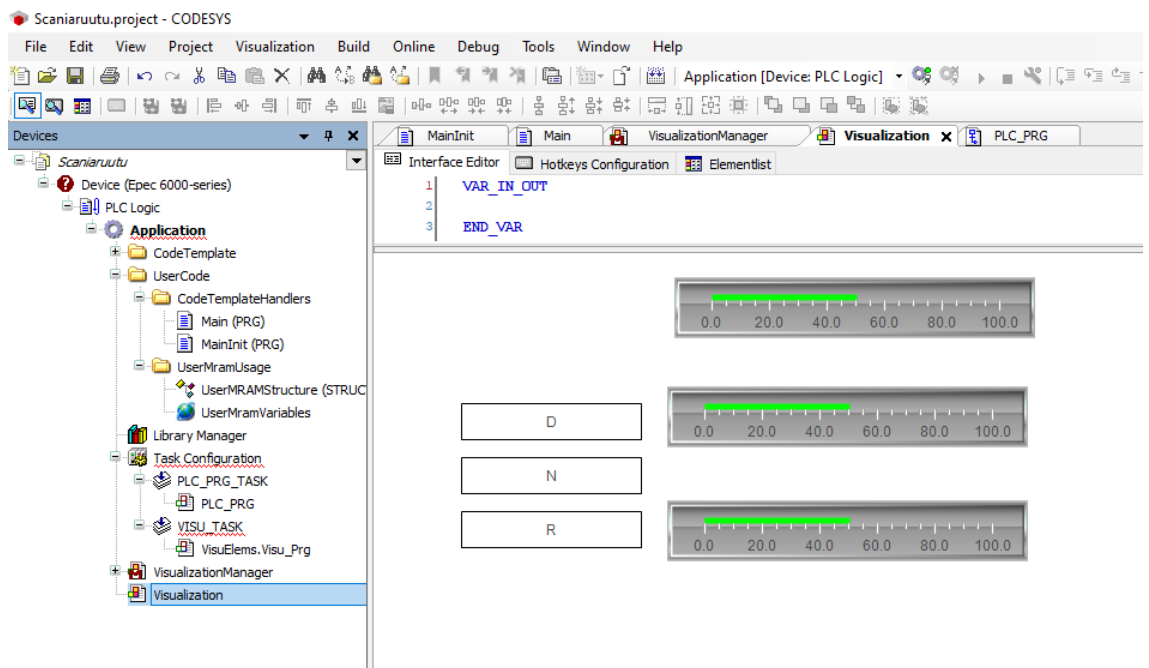


KUVA 9. Näyttöleike Pi Snoopista ohjelman testauksen aikana

5.4 CODESYS-ohjelmistoalusta

CODESYS on CODESYS Groupin kehittämä ohjelmistoalusta, joka on tarkoitettu käytettäväksi teollisuusautomaatioteknologiassa. Alustan ydin on CODESYS Development System -ohjelmointityökalu, joka mahdollistaa automaatiosovellusten käytännönläheisen ja helpon ohjelmoinnin ja konfiguroinnin. Laittevalmistajat käyttävät CODESYSiä omien ohjelmoitavien tai konfiguroitavien automaatiolaitteidensa toteuttamiseen, minkä ansiosta CODESYSin varaan toteutettuja laitteita löytyy maailmasta tuhansia erilaisia. (codesys.com, 2021.)

Tässä opinnäytetyössä CODESYSiä käytettiin Epec 6107 -kosketusnäytön ohjelmointiin. CODESYSin visualisointityökalulla luotiin näytölle ajoneuvon kuljettajan operointipaneeli sekä operointipaneelin diagnostiikkasivut. Kuvassa 10 on esitetty CODESYSillä testivaiheessa luodusta käyttöliittymästä.



KUVA 10. Näyttöleike CODESYSillä testivaiheessa luodusta käyttöliittymästä

6 OHJELMAN SUUNNITTELU

Opinnäytetyössä tuotetun ohjelman suunnittelu koostui ohjausjärjestelmällä käytössä olleiden CAN-kenttäväylien väyläsuunnittelusta sekä varsinaisen tuotetun ohjelman suunnittelusta. Väyläsuunnittelussa ohjausjärjestelmän toimilaitteet jaettiin käytössä olleiden CAN-väylien kesken mahdollisimman tasaisesti siten, että kenttäväylille järjestelmän toimilaitteiden keskinäisestä kommunikaatiosta syntyvät kuormitukset ovat mahdollisimman tasaisia. Ohjelman suunnittelussa ohjelma jaettiin ensin eri toimintoja ohjaaviin alajärjestelmiin, minkä jälkeen suunniteltiin erikseen kunkin alajärjestelmän tarkempi toiminta.

6.1 Väyläsuunnittelu

M580-ohjausyksiköllä on käytössä 4 SAE J1939 -protokollan CAN-väylää, jotka jaettiin mahdollisimman tasaisesti väyliä käyttäville toimilaitteille mahdollisimman vapaan viestiliikenteen takaamiseksi. Lähtökohtaisesti CAN-väylien perusnopeutena käytetään 500 kbps. Koska ajoneuvon oman CAN-väylän nopeudeksi on määritetty 250 kbps, ajoneuvon CAN-väylän ja ohjausyksikön välisen viestiliikenteen väylän nopeuden tuli olla perusnopeudesta poiketen 250 kbps. Työn aikana HVLP10-inverttereitä ohjaavan viestiliikenteen todettiin myös kuormittavan viestejä kuljettavaa CAN-väylää liikaa väylän nopeuden ollessa 500 kbps, joten invertterien väylän nopeudeksi nostettiin J1939-protokollasta poiketen 1 Mbps, koska sekä M580-ohjausyksikkö että HVLP10-invertterit tukevat kyseistä väylän nopeutta.

Koska sekä ajoneuvon oma CAN-väylä että HVLP10-invertterit tarvitsevat edellä mainituista syistä CAN-väylien perusnopeudesta poikkeavat viestiliikenteen nopeudet, varattiin niille väyläsuunnittelussa omat CAN-väylät. HVLP10-inverttereille varattiin CAN-väylä 2, jonka nopeudeksi asetettiin 1 Mbps ja ajoneuvon oman CAN-väylän kanssa tapahtuvalle viestiliikenteelle varattiin CAN-väylä 4, jonka nopeudeksi asetettiin 250 kbps. Loput CAN-väyliä käyttävät toimilaitteet ja niiden väylät on esitetty taulukossa 2.

TAULUKKO 2. Toimilaitteiden jako CAN-väyliin

CAN-väylä	Toimilaite
1 (500 kbps)	Danfoss EC-C1200-450 Motor Inverter
1 (500 kbps)	Epec 6107 -kosketusnäyttö
2 (1 Mbps)	Sevcon HLVP10 -invertteri 1
2 (1 Mbps)	Sevcon HLVP10 -invertteri 2
3 (500 kbps)	Orion BMS 2 (Master)
4 (250 kbps)	Ajoneuvon oma CAN-väylä

6.2 Ohjelman jako alajärjestelmiin

Toteutettava ohjelma jaettiin suunnitteluvaiheessa pienempiin kokonaisuuksiin, jotka toteutetaan Simulink-mallissa omina alajärjestelminään mallin selkeyttämiseksi. Alajärjestelmät, joihin malli jaetaan voivat välittää signaaleja toistensa välillä, mutta ne toimivat silti pääasiassa itsenäisinä kokonaisuuksina. Ohjelman sisältämiä konfiguraatiolohkoja varten luodaan myös niiden oma alajärjestelmä mallin aloitussivun selkeyttämisen vuoksi.

6.2.1 Latauksenohjaus

Sähkökuorma-auton latauksenohjausta hallitseva alajärjestelmä sisältää projektin aikana aikaisemmin valmiiksi luodun mallin, joka ohjaa latauksen aloittamiseen ja lopettamiseen liittyviä toimintoja sekä latauksen aikaista kommunikatiosekvenssiä latausaseman kanssa. Latauksenohjausalajärjestelmään ei opinäytetyön aikana juuri tehdä sen pääasialliseen toiminnallisuuteen vaikuttavia muutoksia, mutta alajärjestelmän siirtyessä osaksi koko ajoneuvon ohjausjärjestelmän kokonaisuutta siihen lisätään joitakin latauksen turvallisuuteen liittyviä ehtoja. Kuorma-auton virtalukolta tulevien asentoviestien tulee asettaa latauksenohjaussekvenssin muuttuja "ChargePermission" tilaan 0, mikä keskeyttää käynnissä olevan latauksen ja estää uuden latauksen aloittamisen. Vastaavasti kuorma-auton latauspistokkeelta tulevan Proximity Pinnin tilaviestin 1 tulee estää kuorma-auton asettaminen ajotilaan. Näiden ehtojen sisällyttäminen latauk-

senohjauksen alajärjestelmään poissulkee tilanteen, jossa kuorma-auto olisi samanaikaisesti latauksessa ja ajotilassa. Alajärjestelmään sisällytetään lisäksi Orion BMS 2 -akustohallintajärjestelmien ja M580-ohjausyksikön välistä CAN-viestiliikennettä ohjaavat lohkot, koska alkuperäinen latauksenohjausmalli on tehty eri akustohallintajärjestelmää käyttävälle testijärjestelmälle.

Latauksenohjausta hallitseva alajärjestelmä jakautuu kahteen osaan, joista toinen ohjaa latauksen aloittamiseen ja lopettamiseen liittyviä toimenpiteitä, kuten lataustilaan siirtymistä sekä latauspistorasian lukitusmoottorin ohjausta, ja toinen, alajärjestelmän sisäinen alajärjestelmä ohjaa ajoneuvon lataussekvenssiä.

6.2.2 Moottorinohjaus

Ohjelman moottorinohjausalajärjestelmään sisällytetään kaikki ajoneuvon moottorin ohjaukseen ja muut ajoneuvon ajotilan hallintaan liittyvät toiminnot. Alajärjestelmässä luetaan tietoja ajoneuvon omasta CAN-väylästä sekä ajoneuvon virtalukon asento, joka saadaan luettua virtalukolta jänniteviestinä. Lisäksi alajärjestelmään tuodaan ohjelman latauksenohjausalajärjestelmältä tieto latauspistorasian Proximity Pinnin tilasta. Alajärjestelmään tuodaan myös ajoneuvon ohjaamossa sijaitsevalta Epec 6107 -kosketusnäytöltä CAN-viestin välityksellä sähkömoottorin regeneraatiota päälle ja pois ohjaava muuttujan tila, jota ajoneuvon kuljettaja pystyy muuttamaan kosketusnäytöltä käsin. Näytöltä ohjattavan muuttujan avulla ajoneuvon kuljettaja voi siten kytkeä sähkömoottorin regeneraation pois päältä tilanteissa, joissa sitä ei haluta käyttää. Moottorinohjausalajärjestelmän tulossignaalit ja niiden lähteet on esitetty taulukossa 3.

TAULUKKO 3. Moottorinohjauksen tulossignaalit ja niiden lähteet

Tuloviesti	Lähde	Viestin tyyppi
Virtalukon asento	Virtalukko	Jänniteviesti
Proximity Pin status	Latauksenohjaus	Binääriviesti
Kaasupolkimen asento	Ajoneuvon CAN-väylä	CAN-viesti
Vaihdekepin asento	Ajoneuvon CAN-väylä	CAN-viesti
Regeneraation status	Epec 6107 -kosketusnäyttö	CAN-viesti

Ajoneuvon CAN-väylästä moottorinohjaukseen luettavat CAN-viestit ovat ajoneuvon kaasupolkimen ja vaihdekepin asentoviestit, jotka saadaan väylälle luodun DBC-tietokannan avulla käännettyä 8-bittisistä binääriluvuista ohjelmaan desimaaliluvuiksi. Testaamisen aikana vaihdekepin asentoviestin arvon todettiin olevan asennossa N 78, asennossa R 82 ja asennossa D 49–54, joten vaihdekepin tila saadaan tulkittua asentoviestistä ohjelmaan yksinkertaisesti vertailuoperaattoreita käyttämällä. Kaasupolkimen asentoviestin todettiin vaihtelevan kaasupolkimen ääriasentojen välillä lineaarisesti 0 ja 255 välillä. Asentoviestiä halutaan ohjelmassa käyttää moottorinohjausyksikölle lähetettävänä ohjausviestinä vaihteluvälillä 0–100 %, joten CAN-väylästä saatavaa asentoviestiä tulee käsitellä ohjelmassa laskennallisesti. Kaasupolkimen asennosta poistetaan saturaatiolla ensimmäiset 10 % tasaisemman ajokokemuksen tuottamiseksi, joten ohjelmassa käsiteltävä vaihteluväli on 25–255. Kaavassa 1 on esitetty ohjelmaan tehtävä laskutoimitus, jolla kaasupolkimen saturoitu asentoviesti saadaan vaihteluvälille 0–100 %.

$$y = \frac{x - 25}{2,3} \quad (1)$$

jossa y on moottorinohjausyksikölle välitettävä ohjausviesti ja x on ajoneuvon CAN-väylästä luettava kaasupolkimen asentoviesti.

Ajoneuvon CAN-väylästä luettavaa vaihdekepin asentoviestiä käytetään ohjelmassa sähkömoottorin pyörimissuunnan määrittelyyn. Viestin arvon ollessa vaihteluväliltä 49–54 (vaihdekepin asento D) moottorin ohjataan pyöri-
vään eteenpäin ja arvon ollessa 82 (vaihdekepin asento R) moottorin ohjataan pyöri-
vään taaksepäin. Viestin arvon ollessa 78 (vaihdekepin asento N) ajoneuvon
sähkömoottorin teho ajetaan alas, jolloin moottori vastustaa pyörimistään hyvin
vähän. Tämän ansiosta ajoneuvo käyttäytyy kuten neutraalilla vaihteella oleva
polttomoottoriajoneuvo.

Sähköajoneuvoilla on polttomoottoriajoneuvoista poiketen teoriassa mahdollista saavuttaa ajoneuvon maksiminopeus myös peruuttaessa, koska erillisen peruutusvaihteen sijaan peruuttaminen tapahtuu vain vaihtamalla moottorin pyörimis-
suunta. Tasaisemman ja turvallisemman ajokokemuksen mahdollistamiseksi

taaksepäin pyöriessä moottorinohjausyksikölle välitettävää kaasupolkimen asentoviestiä kuitenkin rajoitetaan siten, että moottoria ohjataan vain murto-osalla sen maksimitehosta. Työn aikana ohjausviestin taaksepäin pyörimisen kertoimeksi määriteltiin 0,1.

Ohjelman halutaan siirtyvän ajotilaan, kun kaikki ohjelmassa käytettävän Drive Permission -parametrin ehdot täyttyvät. Tämänhetkisessä toteutuksessa parametrin ehtoina on, että ajoneuvon latauspistorasian tulee olla tyhjä, eli latauspistooli ei ole paikoillaan, ja ajoneuvon virtalukko tulee olla käännettynä ajoasentoon. Mikäli yksikin Drive Permission -parametrin ehdoista ei toteudu, ajoneuvon moottorinohjausta ei aktivoida. Vastaavasti, jos parametrin ehdot lakkaavat toteutumasta moottorinohjauksen ollessa aktiivisena, moottori sammutetaan. Parametrin ehtojen ohjelmointi halutaan ohjelmassa toteuttaa siten, että sen ehtoja on myöhemmässä kehitysvaiheessa yksinkertaista muokata tai muuttaa.

Moottorin ajokontaktorien ohjaaminen päälle tulee suorittaa ohjelmassa kolmessa vaiheessa. Miinusnavan ajokontaktori suljetaan, kun ajoneuvo siirtyy ajotilaan kuljettajan kääntäessä virtalukon tilaan 3. Miinusnavan sulkeutumisen jälkeen ohjelman tulee odottaa 3 sekuntia, jonka jälkeen moottorin esilatauskontaktori suljetaan 3 sekunniksi. Esilatauskontaktori avataan, kun esilataus on suoritettu, minkä jälkeen ohjelma odottaa jälleen 3 sekuntia, minkä jälkeen ohjelma sulkee plusnavan ajokontaktorin. Sekä miinus- että plusnapainen ajokontaktori avataan, kun ajoneuvon virtalukko käännetään asennosta 2 asentoon 1, eli siirryttäessä ajotilasta valmiustilaan.

6.2.3 Ohjaustehostimen ja kompressorin invertterien ohjaus

Ohjaustehostimen ja kompressorin HVLP10-invertterien ohjaukselle eritellään ohjelmassa oma alajärjestelmä. Ohjaustehostimen invertterin ohjaus toteutetaan ohjelmassa yksinkertaisesti kaksiasento-ohjauksella, jossa invertteri ohjataan päälle aina, kun ajoneuvo on ajotilassa ja pois päältä, kun ajoneuvo siirtyy pois ajotilasta. Kompressorin invertterin ohjaus toteutetaan niin ikään kaksiasento-ohjauksella, mutta sen ohjauksessa on otettava huomioon ajoneuvon tilan lisäksi kompressorin pneumaattinen paine. Invertteri tulee ohjata ajoneuvon

tilasta huolimatta pois päältä, kun sen paine saavuttaa ylärajansa. Invertteri ohjataan takaisin päälle, kun paine saavuttaa taas alarajansa. Kompressorin ohjauksessa seurataan lisäksi sen lämpötila-anturia. Kompressorin puhallin ohjataan erillisellä jänniteviestillä ohjattavalla releellä päälle lämpötilan saavutettua ylärajansa ja pois päältä lämpötilan saavutettua määritellyn alarajan.

HVLP10-invertterien ohjauksessa inverttereille tulee myös asettaa raja-arvot niiden toimintaan vaikuttaville parametreille. Raja-arvot on määritelty hankkeelle aiemmin tehdyn opinnäytetyön aikana. Parametrit ja niiden arvot ohjaustehostinta ja kompressoria ohjaaville inverttereille on esitetty taulukossa 4.

TAULUKKO 4. Invertterien parametrit arvoineen

Parametri	Arvo kompressorin invertterille	Arvo ohjaustehostimen invertterille
Torque Traction Limit	800	800
Torque Request	1600	1600
Torque Regeneration Limit	0	0
Speed Limit (Forward)	1450	1440
Speed Limit (Reverse)	0	0
DC Link maksimipurkuvirta	21	13
DC Link maksimilatausvirta	0	0

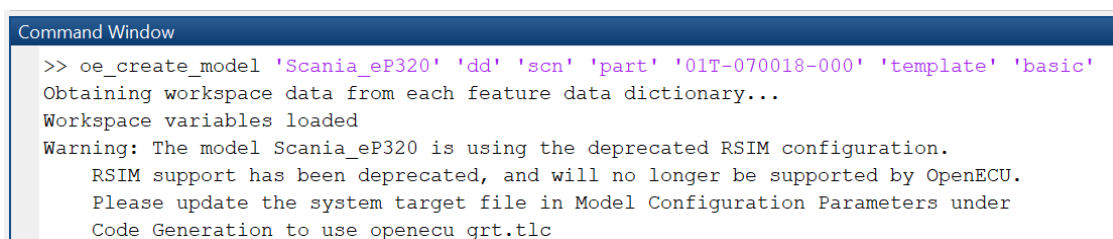
6.2.4 Lepotila

Ohjelman lepotila-alajärjestelmän tarkoituksena on katkaista käyttöjännite ajoneuvon toimilaitteilta tilanteessa, jossa kuljettaja jättää epähuomiossa päävirtakytkimen päälle ajoneuvosta poistuessaan. Lepotilan perimmäisenä tarkoituksena on siten estää ajoneuvon akuston latausta kulumasta auton ollessa poissa käytöstä. Lepotila kytketään päälle, kun ajoneuvon virtalukko on ollut käännettynä asentoon 0 yhtäjaksoisesti 20 minuutin ajan, mutta ajoneuvon päävirtakytkintä ei ole käännetty pois päältä.

Lepotilan toteutuksessa käytetään hyödyksi erillistä relettä ja M580-ohjausyksikön sisäistä Power Supply Hold -toimintoa, joka pitää käytössä ollessaan ohjausyksikön käynnissä herätejännitteen puuttuessa. Releen avulla ajoneuvon toimilaitteiden 24 V syöttöjännitteen syöttö laitteille katkaistaan, kun lepotilan ehtojen täyttymisestä on kulunut 20 minuuttia. Syöttöjännitteen katkaisemisen jälkeen ohjelma odottaa vielä 10 minuuttia, ennen kuin se sammuttaa ohjausyksikön poistamalla Power Supply Hold -toiminnon pois käytöstä. Ajoneuvo poistuu lepotilasta, kun päävirtakytkin käännetään pois päältä ja takaisin päälle, tai kun virtalukko käännetään asentoon 1, mikä syöttää herätejännitteen ohjausyksikölle.

7 TOTEUTETTU OHJELMA

Ohjelman toteuttaminen aloitettiin luomalla ohjelman Simulink-malli, mikä tapahtui syöttämällä MATLABin komentoikkunaan OpenECU-lisäkirjastoon kuuluva komento `oe_create_model`. Komennon perään syötettiin lisäksi mallin luomiseen tarvittavat parametrit, joita olivat mallin nimi, mallin Data Dictionaryn nimi, käytettävän ohjausyksikön osanumero sekä mallin luomiseen käytettävä pohja. Mallin luomiseen käytetty komentorivi on esitetty kuvassa 11.



```
Command Window
>> oe_create_model 'Scania_eP320' 'dd' 'scn' 'part' '01T-070018-000' 'template' 'basic'
Obtaining workspace data from each feature data dictionary...
Workspace variables loaded
Warning: The model Scania_eP320 is using the deprecated RSIM configuration.
        RSIM support has been deprecated, and will no longer be supported by OpenECU.
        Please update the system target file in Model Configuration Parameters under
        Code Generation to use openecu_grt.tlc
```

KUVA 11. Ohjelman mallin luominen MATLABin komentoikkunassa

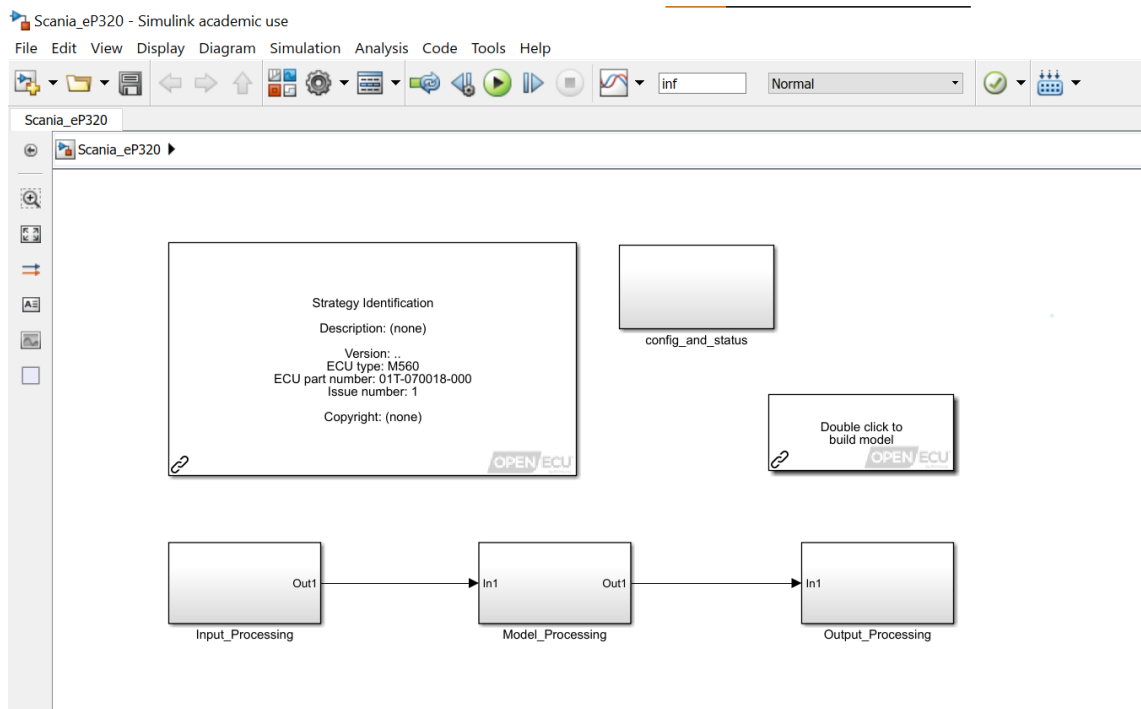
Mallin Data Dictionary (kuvassa 11 'dd') on mallin luomisen yhteydessä luotava tekstilista, johon listataan kaikki mallissa käytettävät signaalit. Signaalinen määrittely Data Dictionaryssä mahdollistaa signaalien arvojen seuraamisen Pi Snoopilla sekä arvojen pitämisen sallituissa rajoissa. Data Dictionaryssä voidaan signaalien lisäksi määrittellä ohjelmassa käytettäviä vakioarvoisia muuttujia. Data Dictionaryyn syötetään signaalien nimi, mahdollinen vakioarvo, mahdollinen yksikkö, signaalin tyyppi, tarkkuus, minimi- ja maksimiarvo sekä mahdollinen kommentti, joka selittää mikä signaalin käyttötarkoitus on. Data Dictionary luodaan tekstitiedostoon, mutta sen käsittely on yksinkertaisinta Excelillä, koska signaalien parametrit on listassa eritelty tabulaattorilla. Kuvassa 12 on esitetty näyttöleike työssä käytetystä Data Dictionarystä.

10	Name	Value	Units	Type	Accuracy	Min	Max	DeclFile	DefnFile	Description
11	msg_sampletime	0.001	s	real_T	0.001	0	1			Sample time for V2G communication messages
12	client_task_state			real_T	1	0	10			Client Task State
13	a_if_chip_status			real_T	1	0	10			PLC Interface Chip Status
14	a_slac_state			real_T	1	0	10			SLAC state
15	proximity		V	real_T	0.01	0	5			Voltage measured from the proximity pin
16	proximity_bool		state	BOOL	1	0	1			State of the proximity pin, if 1 proximity pin detected
17	pt1000_voltage		V	real_T	0.01	0	5			Voltage measured from Pt1000
18	pt1000_temp		degC	real_T	0.01	0	120			Pt1000 voltage converted to temperature
19	pt1000_temp_limit	90	degC	real_T	0.01	0	120			Pt1000 temperature limit
20	pt1000_temp_valid		state	BOOL	1	0	1			True when temperature below 90
21	CP_freq		Hz	real_T	1	0	10000			Control pilot frequency
22	CP_duty_cycle			real_T	0.01	0	1			Control pilot duty cycle
23	CP_timed_out		state	BOOL	1	0	1			True when CP has timed out
24	CP_pin_state		state	BOOL	1	0	1			Control pilot pin state
25	release_button		state	BOOL	1	0	1			True when release button is pressed
26	hbridge_mode			real_T	1	0	5			Driving mode of the Hbridge controlling the inlet motor
27	inlet_motor_voltage		V	real_T	0.01	0	5			Voltage measured from the inlet motor
28	inlet_motor_locked		state	BOOL	1	0	1			True when inlet motor is locked
29	resistor_relay		state	BOOL	1	0	1			True when resistor relay is on

KUVA 12. Näyttöleike työssä käytetystä Data Dictionarystä

Data Dictionaryn lisäksi mallin luomisen yhteydessä luotiin mallin Units File, johon listataan kaikki signaalien käyttämät yksiköt. Units File on Data Dictionaryn tapaan tekstitiedosto. Mallin Units File ja Data Dictionary tulee pitää toistensa kanssa ajan tasalla, sillä mikäli Data Dictionaryssä määritellään signaalille yksikkö, jota ei ole lueteltu Units Filessa, mallin kääntäminen C-koodiksi ei onnistu.

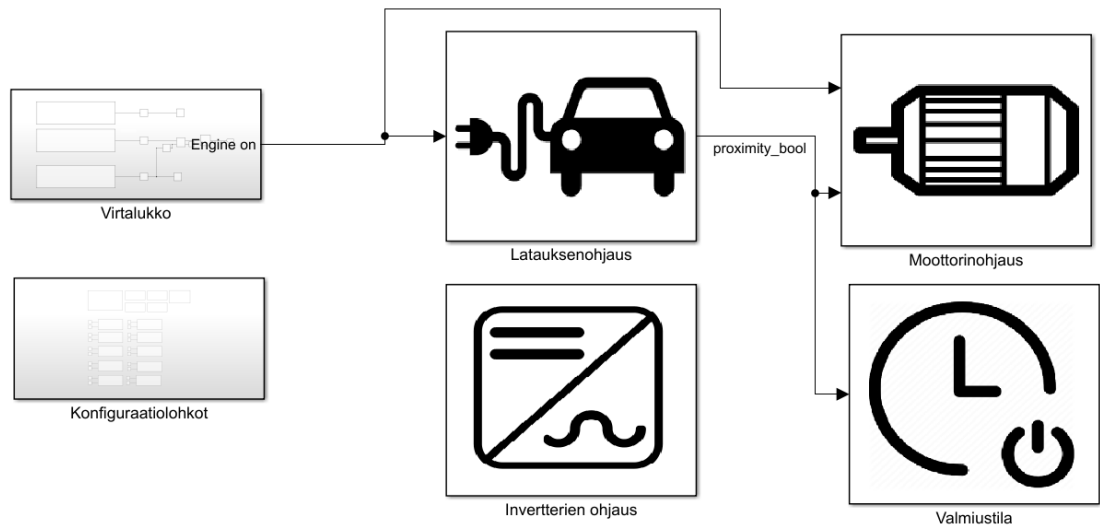
Mallin luominen työssä käytetyllä Basic-pohjalla luo Simulink-mallin, joka sisältää ohjausyksikön identifiointilohkon, konfigurointialajärjestelmän sekä kolme tyhjää mallialajärjestelmää. Identifiointilohkon lisäksi Basic-pohjan sisältämistä lohkoista säilytetään CAN-väylän konfigurointilohko sekä CCP-konfigurointilohko. Muut pohjan sisältämistä lohkoista poistettiin tarpeettomina. Basic-pohjalla luotu malli on esitetty kuvassa 13.



KUVA 13. Ohjelman alkuperäinen Basic-malli

Ohjausyksikön ja sen CAN-väylien konfiguraatiolohkot sijoitettiin ohjelmassa omaan alajärjestelmäänsä. Konfiguraatiolohkojen alajärjestelmä sisältää ohjausyksikön identifiointilohkon, ohjausyksikön neljän CAN-väylän konfigurointilohkot, CCP-konfigurointilohkon sekä 10 muuta ohjausyksikön konfigurointiin käytettävää lohkoa. Ohjelman väyläsuunnittelun mukaisesti väylät 1 ja 3 (ohjelmassa A ja C) määriteltiin nopeudelle 500 kbps, väylä 2 (ohjelmassa B) nopeudelle 1 Mbps ja väylä 4 (ohjelmassa D) nopeudelle 250 kbps. CCP-konfiguraatiolohkossa CCP:n määriteltiin olevan käytössä ja käyttävän väylää 1. CCP (CAN Configuration Protocol) tulee pitää ohjelmaa kehittäessä aina aktivoituna, sillä mikäli ohjausyksikölle ladataan ohjelma, jossa CCP ei ole aktivoituna, ohjausyksikköön ei saa enää yhteyttä CCP:n välityksellä. Tällöin ohjausyksikkö joudutaan palauttamaan tehdasasetuksiinsa CCP-yhteyden uudelleen muodostamiseksi.

Konfiguraatiolohkojen alajärjestelmän lisäksi ohjelmalle luotiin virtalukolta luettavia jänniteviestejä tulkitseva alajärjestelmä sekä ohjelman suunnitteluvaiheessa määritellyt 4 alajärjestelmää, joihin ohjelman eri toiminnot jaetaan. Ohjelman mallin ylin taso ja sen sisältämät pääalajärjestelmät on esitetty kuvassa 14.



KUVA 14. Ohjelman ylimmän tason alajärjestelmät

7.1 Latauksenohjaus

Ohjelman suunnittelun mukaisesti latauksenohjauksen alajärjestelmä sisältää projektin aikana aiemmin luodun latauksen aloittamiseen ja lopettamiseen liittyviä toimintoja sekä latauksen aikaista kommunikaatiosekvenssiä ohjaavan mallin. Latausaseman kanssa tapahtuvan kommunikaation sekvenssiä ohjaava osuus on sijoitettu omaan alajärjestelmäänsä mallin selkeyttämiseksi.

Ohjelma aloittaa latauksen aloittamiseen liittyvien toimintojen suorittamisen, kun ohjelma havaitsee latauspistorasian Proximity Pinnin jännitteen muutoksen. Latauspistorasialta luettavan jänniteviestin arvon laskiessa alle 4,2 voltin latauspistokkeen todetaan olevan paikoillaan latauspistorasiassa. Jänniteviestin toiminnan varmistamiseksi jänniteviestin arvon tulee myös olla yli 1 voltia. Ehtojen toteutuessa ohjelman proximity_bool-signaali asettuu tilaan 1 ja signaali välittää siten tiedon latauspistokkeen läsnäolosta ohjelman toiminnoille, jotka kyseistä tietoa tarvitsevat. Proximity_bool-signaalin asetuttua tilaan 1 ohjelma odottaa kaksi sekuntia ennen latauspistorasian lukitusmoottorin sulkemista. Lukitusmoottori suljetaan ohjaamalla moottoria ohjaavan H-sillan lähtö tilaan 2 (ajo eteenpäin) sekunnin ajaksi, minkä jälkeen H-silta palautetaan tilaan 0. H-siltaa ohjataan ohjelmassa vain sekunnin pulsseilla, koska työn aikana latauspistok-

keen lukitusmoottorin havaittiin ylikuumentuvan jatkuvan ohjauksen seurauksena. Latauksen loputtua lukitusmoottori avataan ohjaamalla H-silta sekunnin ajaksi tilaan 3 (ajo taaksepäin).

Proximity Pinnin tilan ja lukitusmoottorin ohjaamisen lisäksi alajärjestelmä hallitsee ohjelman ChargePermission-muuttujan tilaa. ChargePermission on ohjelmassa muuttuja, joka määrittelee, onko lataus mahdollista aloittaa ja täytyykö käynnissä oleva lataus mahdollisesti keskeyttää. ChargePermission-muuttujan ehtoina käytetään ajoneuvon virtalukon asentotietoa, lukitusmoottorin lukitustilaa sekä latauspistorasian lämpötilaa. ChargePermission-muuttujan todentuminen vaatii virtalukon asentoa 0, lukitusmoottorin lukitusta sekä latauspistorasian lämpötilan ylärajan alittumista. Mikäli jokin ehdoista ei toteudu, ChargePermission pysyy tilassa false, mikä estää latauksen aloittamisen ja keskeyttää käynnissä olevan latauksen.

Lataussekvenssin ohjaus sisältää kappaleessa 3.5 esitetyn latausaseman ja ajoneuvon välisen kommunikaatiosekvenssin ohjauksen. Sekvenssin ohjaus aloitetaan, kun proximity_bool- ja ChargePermission-muuttujat ovat molemmat tilassa 1. Sekvenssin alussa ohjelma perustaa yhteyden ohjausyksikön ja latausaseman välille, jonka jälkeen se lähtee käymään sekvenssin vaiheita järjestyksessä läpi. Sekvenssin ohjaaminen toteutetaan OpenECU-lisäkirjaston Vehicle to Grid Message -toimilohkoilla. Toimilohkon parametreista valitaan haluttu viesti, minkä jälkeen toimilohko päivittää itselleen kyseisen viestin vaatimat tulo- ja lähtöviestit.

Taulukossa 1 esitellyt sekvenssin vaiheet 0a-Yb etenevät ohjelmassa itsenäisesti, eivätkä niiden toimilohkot tarvitse muita tuloviestejä kuin lähetyskäsken, joka lähetetään ohjelmassa edellisen vaiheen saapumisilmoituksella. Vaiheen 3a (ChargeParameterDiscoveryReq) toimilohkolle syötetään latauksessa käytettävät parametrit. Akustoon liittyvät parametrit, kuten maksimijännite ja -virta, syötetään toimilohkolle CAN-viestien välityksellä ajoneuvon akustoja ohjaavalta Orion BMS 2 -akustonhallintajärjestelmältä. Vaiheen 3b (ChargeParameterDiscoveryRes) saapumisviestin vastaanottamisen jälkeen ohjelma ohjaa Control Pilotin jännitettävä säätävän piirin Omron G2R-1-SNI tehoreleellä tilaan C, mikä mahdollistaa lataussekvenssin siirtymisen vaiheeseen 4a (CableCheckReq).

Sekvenssi etenee tämän jälkeen automaattisesti latauksen aloittamiseen asti. Latausta aloittaessa ohjelma tarkistaa, että ChargePermission-muuttuja on tilassa 1, ja akuston lataustaso ei ole jo halutussa arvossa. Latauksen aloitettua sekvenssi pitää latauksen käynnissä, kunnes ChargePermission-muuttujan arvo muuttuu, tai lataus on valmis. Latauksen loputtua ohjelma ohjaa tehoreleen avulla Control Pilotin jännitteensäätöpiirin takaisin tilaan B.

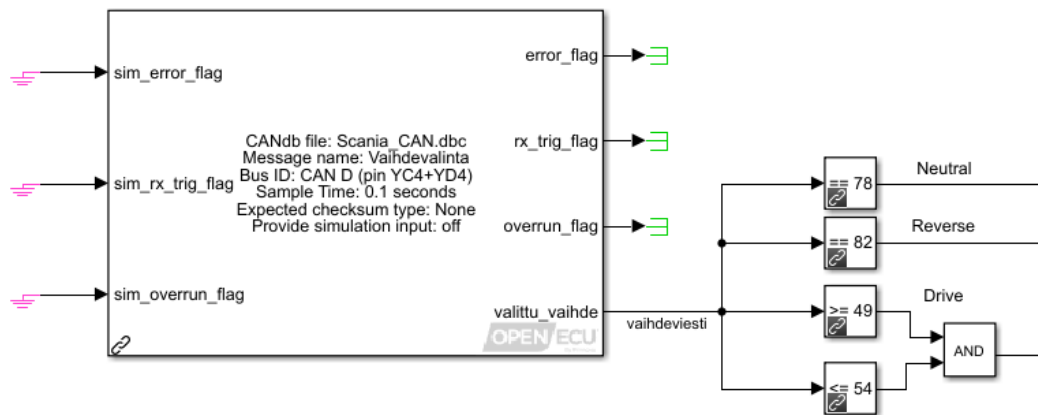
7.2 Moottorinohjaus

Ohjelman suunnittelun mukaisesti moottorinohjausjärjestelmä sisältää kaikki ajoneuvon sähkömoottorin ohjaamiseen liittyvät toiminnot. Kaikki moottorin ohjaamiseen liittyvät toiminnot vaativat toimiakseen DrivePermission-muuttujan tilan 1, jonka ehtoina on proximity_bool-muuttujan tila 0 ja virtalukolta saatavan EngineOn-muuttujan tila 1. EngineOn-muuttuja asetetaan tilaan 1, kun ajoneuvon virtalukko käännetään ääriasentoonsa 3 ja pysyy tilassa 1, kunnes virtalukko käännetään pois asennosta 2. EngineOn-muuttujan asettaminen tilaan 1 täytyy ohjelmassa tehdä hyödyntäen pitopiiriä, koska virtalukko pysyy asennossa 3 vain, kun virta-avainta käännetään lukossa aktiivisesti ääriasentoon ja palautuu aktiivisen kääntämisen loppumisen jälkeen asentoon 2.

Moottorinohjausyksikkö aktivoidaan ohjelmassa, kun EngineOn-muuttuja on tilassa 1. Ohjausyksikkönä toimiva Danfoss EC-C1200-450-invertteri aktivoidaan asettamalla CAN-väylän välityksellä sen DBC-tietokannan mukaisen cmd_enable-viestin arvoksi 1. Ajoneuvon vaihdekepin ollessa asennossa D tai R sähkömoottori käynnistetään asettamalla invertterin DBC-tietokannan mukaisen cmd_mot_run-viestin arvoksi 1. Lisäksi invertteri asetetaan käyttämään sähkömoottorin ohjaamiseen teho-ohjausta asettamalla invertterin DBC-tietokannan mukaisen cmd_mot_ctrl_mode-viestin arvoksi 4.

Ajoneuvon CAN-väylästä luettavaa vaihdekepin asentoviestiä käytetään ohjelmassa sähkömoottorin ohjaussuunnan määrittämiseen. CAN-viestin arvot luettiin ajoneuvon CAN-väylästä eri vaihdekepin asennoilla, minkä avulla määriteltiin vaihdeviestin arvon olevan asennossa N 78, asennossa R 82 ja asennossa D 49–54. Vaihdekepin asento saadaan siten tulkittua ohjelmassa vertaamalla

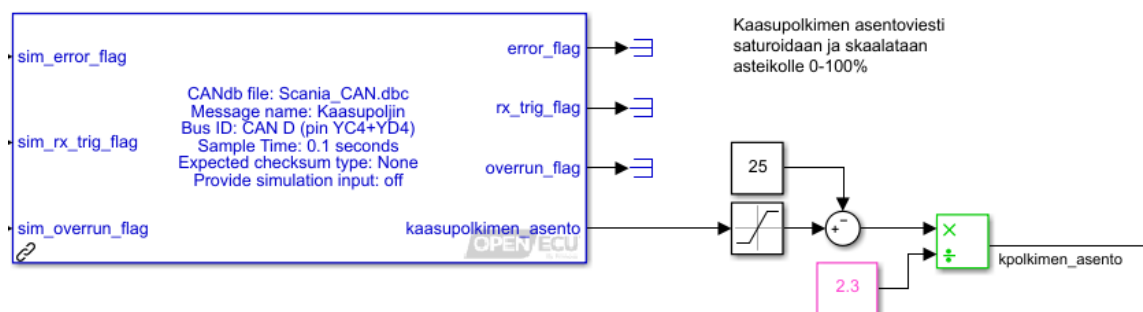
vaihdeviestin arvoa tiedettyihin asentojen arvoihin. Viestin vertailu on esitetty kuvassa 15.



KUVA 15. Vaihdekepin arvojen vertailu ohjelmassa

Vaihdekepin ollessa asennossa D tai R kaasupolkimen asentoviesti ohjataan moottorinohjausyksikölle. Vaihdekepin ollessa asennossa N kaasupolkimen asentoviestiä ei välitetä moottorinohjausyksikölle, vaan moottorinohjausyksikölle lähetettävä signaali cmd_mot_run asetetaan tilaan 0. Asennossa R invertterille välitettävä kaasupolkimen asentoviesti kerrotaan kertoimella 0,1 turvallisemman peruuttamisen mahdollistamiseksi.

Ajoneuvon CAN-väylästä luettava kaasupolkimen asentoviesti käsitellään ohjelmassa kaavan 1 mukaisesti ennen sen välittämistä eteenpäin. Laskennalliseen käsittelyyn käytetyt toimilohkot on esitetty kuvassa 16.



KUVA 16. Kaasupolkimen asentoviestin käsittely

Kaasupolkimen käsitelty asentoviesti välitetään moottorinohjausyksikölle, kun ajoneuvon vaihdekeppi on asennossa D tai R. Asentoviestin käsittelemätöntä arvoa käytetään lisäksi sähkömoottorin regeneraation mahdolliseen aktivointiin. Regeneraatiota ohjaava ohjataan arvoon 1, kun Epec 6107 -kosketusnäytöltä luettavan RegenerationStatus-muuttujan arvo on 1 ja kaasupolkimen asentoviestin käsittelemätön arvo on 0, eli kaasupoljinta ei paineta. Opinnäytetyössä toteutetussa ohjelmassa regeneraatiota ohjaavaa muuttujaa ei käytetä varsinaiseen regeneraation aktivointiin, vaan ominaisuus luotiin mahdollisen tulevaisuudessa toteutettavan regeneraation ohjauksen pohjaksi.

7.3 Invertterien ohjaus

Ohjelman suunnittelun mukaisesti ohjaustehostimen ja kompressorin HVLP10-invertterien ohjaus toteutettiin ohjelmassa omassa alajärjestelmässään. Ohjaustehostimen invertteri ohjataan ohjelmassa päälle, kun ajoneuvo siirtyy ajotilaan ja pois päältä, kun ajoneuvo siirtyy pois ajotilasta. Kompressorin invertteri ohjataan vastaavasti päälle, kun ajoneuvo siirtyy ajotilaan ja pois päältä, kun ajoneuvo siirtyy pois ajotilasta, mutta kompressorin invertterin ohjauksessa otetaan lisäksi huomioon kompressorin paine ja lämpötila.

Ohjaustehostimen invertteri ohjataan ajotilaan siirryttäessä päälle lähettämällä sille käynnistysviesti CAN-väylän välityksellä. CAN-viestin lähettämisessä käytetään hyödyksi laitteen valmistajan DBC-tietokantaa. DBC-tietokannan CAN-viesteillä asetettiin myös invertterille parametrit. Taulukon 4 mukaisesti invertterin Torque Traction Limit -parametrin arvoksi asetettiin 800, Torque Request -parametrin arvoksi 1600, Torque Regeneration Limit -parametrin arvoksi 0, Speed Limit (Forward) -parametrin arvoksi 1440, Speed Limit (Reverse) -parametrin arvoksi 0, sekä DC Linkin maksimipurkuvirraksi 13 ja maksimilatausvirraksi 0. Invertteri sammutetaan ajoneuvon siirryessä pois ajotilasta lähettämällä sille CAN-väylän välityksellä sammutusviesti.

Kompressorin invertteri ohjataan ajotilaan siirryttäessä päälle lähettämällä sille CAN-väylän välityksellä käynnistysviesti, kuten ohjaustehostimen invertterille. Käynnistysviestin lähetykselle on kuitenkin ajoneuvon ajotilan lisäksi ehtona

myös kompressorin paineen sallituissa rajoissa oleva arvo. Tieto kompressorin paineen arvosta saadaan paineanturilta ohjausyksikölle jänniteviestinä. Ylärajan ylityttyä kompressorin invertteri ohjataan pois päältä, kunnes paineanturilta saatava arvo saavuttaa sille asetetun alarajan. Mikäli ajoneuvo on alarajan saavuttamisen jälkeen edelleen ajotilassa, kompressorin invertteri ohjataan jälleen päälle. Samantapainen ohjaus tehtiin ohjelmaan myös digitaalisella jännitelähtöviestillä ohjattavalle kompressorin puhaltimelle. Puhallin ohjataan päälle, kun lämpötila-anturilta saatava jänniteviesti saavuttaa sille asetetun ylärajan ja pois päältä, kun lämpötila saavuttaa sille asetetun alarajan. Kompressorin invertterille asetettiin myös DBC-tietokannan viesteillä parametrit. Taulukon 4 mukaisesti invertterin Torque Traction Limit -parametrin arvoksi asetettiin 800, Torque Request -parametrin arvoksi 1600, Torque Regeneration Limit -parametrin arvoksi 0, Speed Limit (Forward) -parametrin arvoksi 1450, Speed Limit (Reverse) -parametrin arvoksi 0, sekä DC Linkin maksimipurkuvirraksi 21 ja maksimilatausvirraksi 0. Invertteri sammutetaan paineanturin arvosta huolimatta ajoneuvon siirtyessä pois ajotilasta lähettämällä sille CAN-väylän välityksellä sammutusviesti.

7.4 Lepotila

Ohjelman suunnittelun mukaisesti lepotila-alajärjestelmän tarkoituksena on katkaista käyttöjännite ajoneuvon toimilaitteilta, kun ajoneuvon virtalukko on käännetty asentoon 0, mutta päävirtakytkin on jätetty päälle. Virtalukon tilaa tarkkailaan ohjelmassa ohjausyksikön Ignition-pinnillä, jolle syötetään jänniteviesti ajoneuvon virtalukon asennosta 1. Kun Ignition-pinnin jännite muuttuu arvoon 0, ohjelma käynnistää ajastintoimilohkon, jonka tavoiteajaksi on asetettu 20 minuuttia. Kun ajastintoimilohko saavuttaa tavoiteaikansa, se ohjaa toimilaitteiden käyttöjännitteen syöttöpiiriin kytketyn releen pois päältä katkaisten käyttöjännitteen syötön ajoneuvon ohjausjärjestelmän toimilaitteille. Samalla ajastintoimilohko käynnistää alajärjestelmän toisen ajastintoimilohkon, jonka tavoiteajaksi on asetettu 10 minuuttia. Toinen ajastintoimilohko deaktivoi tavoiteajan saavuttuaan ohjausyksikön sisäisen Power Supply Hold -toiminnon. Kun toiminto

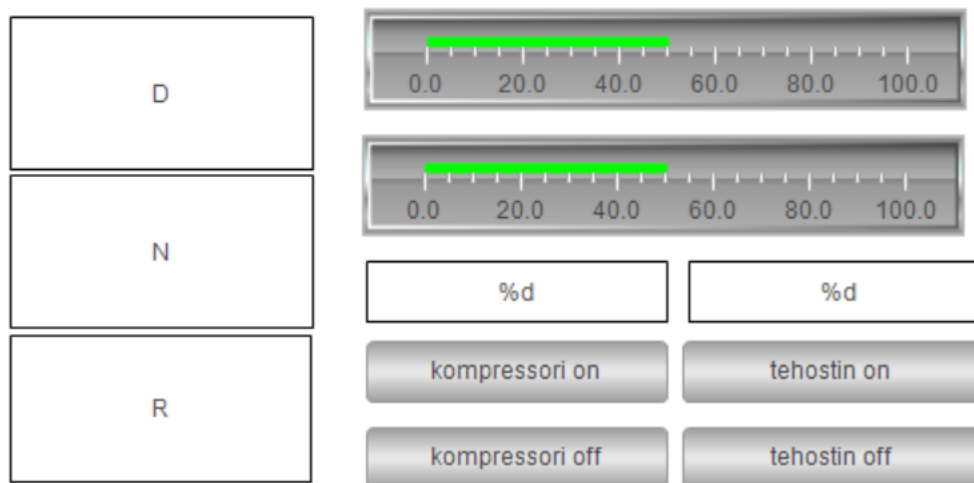
poistetaan käytöstä, se ei enää pidä ohjausyksikköä käynnissä, kun Ignition-pinnillä ei ole jännitettä, mikä johtaa ohjausyksikön sammumiseen ja ajoneuvon ohjausjärjestelmän siirtymisen lepotilaan.

Ohjausyksikkö käynnistyy uudelleen joko kääntämällä ajoneuvon virtalukko asentoon 1, kääntämällä ajoneuvon päävirtakytkin pois päältä ja takaisin päälle tai kytkemällä latauspistoke ajoneuvon latauspistorasiaan. Virtalukon kääntäminen asentoon 1 syöttää virtalukolta jännitteen ohjausyksikön Ignition-pinnille, mikä herättää ohjausyksikön. Päävirtakytkimen kääntäminen pois päältä katkaisee käyttöjännitteen syötön ohjausyksikölle, minkä seurauksena ohjausyksikkö käynnistää ohjelmansa uudelleen nollaten lepotila-alajärjestelmän suorittamat toiminnot. Ohjausyksiköllä on myös käytettävissä Pilot Wake -herätetoiminto, jonka avulla ohjausyksikkö saadaan heräämään, kun latauspistorasian Control Pilot -pinni saa jännitesignaalin latauspistokkeesta. Pilot Wake -herätetoiminto tulee toimiakseen nollata, kun ohjausyksikkö käynnistyy, joten toiminto nollataan ohjelmassa ohjausyksikön sisäisellä Clear Pilot Wake -viestillä aina ohjelman käynnistyessä.

8 OHJELMAN TESTAUS

Ohjelman toimintaa testattiin työn aikana jatkuvasti monissa eri vaiheissa. Testeissä käytettiin ohjelman signaalien seuraamiseen Pi Snoopia, jolla pystyttiin seuraamaan reaaliaikaisesti ohjelman Data Dictionaryssä määriteltujen signaalien arvoja, sekä ohjausjärjestelmän CAN-väylissä liikkuvien viestien arvoja. Pi Snoopilla seurattiin testien aikana esimerkiksi kaasupolkimen asentoviestin saturoinnin ja skaalauksen onnistumista. Pi Snoopia käytettiin testaamisen lisäksi myös ajoneuvon omassa CAN-väylässä liikkuvien viestien arvojen tarkkailuun niiden käyttämiseksi ohjelmassa. Pi Snoopilla selvitettiin esimerkiksi vaihdekepin asentoviestin arvo vaihdekepin eri asennoissa, minkä ansiosta kyseisiä arvoja pystyttiin käyttämään ohjelmaa tehdessä.

Pi Snoopin lisäksi ohjausjärjestelmän testaamiseen käytettiin Epec 6107 -kosketusnäyttöä. Kosketusnäytön avulla pystyttiin simuloimaan esimerkiksi moottorinohjauksen toimimista työn vaiheessa, jossa varsinainen moottorinohjausyksikkö ei ollut vielä käytössä. Kosketusnäyttöä käytettiin työn aikana myös HVLP-10-invertterien koeajamiseen, sillä sen todettiin olevan paras väline invertterien ohjaamiseen päälle ja pois koeajojen aikana. Kosketusnäytölle luotiin CODESYSiä käyttämällä käyttöliittymä, jolla HVLP-10-invertterejä ohjattiin. Testeihin luotu käyttöliittymä on esitetty kuvassa 17.



KUVA 17. HVLP10-invertterien testaamiseen käytetty käyttöliittymä

9 POHDINTA

Opinnäytetyön tavoitteena oli luoda dieselkuorma-autosta sähkökuorma-autoksi muunnettavalle ajoneuvon prototyypille toimiva ohjausjärjestelmää ohjaava ohjelma. Työn aikana ohjelma suunniteltiin ja toteutettiin jakaen se neljään pienempään kokonaisuuteen, jotka olivat latauksenohjaus, moottorinohjaus, ohjaustehostimen ja kompressorin invertterien ohjaus sekä lepotila. Ohjelmointityön aikana ohjelmaan uutena lisättyjä ominaisuuksia testattiin käyttämällä hyödyksi Pi Snoop -työkalua. Ohjausyksikön ohjelmointi suoritettiin MathWorksin Simulink -ohjelmistolla käyttäen hyödyksi ohjausyksikön valmistajan omaa ohjausyksikön ohjelmointiin ja konfigurointiin tarkoitettua lisäkirjastoa.

Opinnäytetyölle asetettu tavoite saavutettiin. Työn aikana onnistuttiin tuottamaan ohjelma, jolla pystytään työn aikana suoritettujen testien perusteella ohjaamaan sähkökuorma-autolle muuntotyön aikana rakennetun sulautetun ohjausjärjestelmän perustoimintoja.

Tämän projektin seuraavat työvaiheet ovat sähkökuorma-auton sulautetun järjestelmän käyttöönottoon liittyviä toimenpiteitä. Tässä opinnäytetyössä luotua ohjelmaa on hyvin todennäköisesti tarpeellista muokata ohjausjärjestelmän käyttöönoton aikana ilmenevistä syistä. Myös monia ohjelman ominaisuuksia, kuten moottorinohjausta on hyvin tarpeellista testata varsinaisella kalustolla pelkän simuloinnin varassa olevan testauksen lisäksi. Ohjelman moottorinohjausajajärjestelmään on myös lisättävä ohjelman lähettämiä CAN-viestejä, joilla syötetään ajoneuvon CAN-väylään arvoja parametreille, joita ajoneuvon anturit yrittävät lukea ajoneuvosta muuntotyön aikana poistetuilta osilta. Näitä parametreja ovat esimerkiksi ajoneuvon alkuperäisen moottorin öljynpaine. Syöttämällä parametreille arvot pystytään estämään ajoneuvon meneminen vikatilaan.

Opinnäytetyölle toi haastavuutta vertailukohteiden ja lähdemateriaalin puute. Polttomoottorikuorma-autojen muuntaminen sähkökuorma-autoksi ei ole maailmanlaajuisesti vielä kovin yleistä, mikä vaikeutti luotettavan lähdemateriaalin löytämistä ohjelman suunnittelun tueksi. Lisähaastetta työlle toi lisäksi hank-

keessa käytettävien laitteiden, kuten ajoneuvon sähkömoottorin saapumisaikataulu, minkä seurauksena monia työssä luodun ohjelman ominaisuuksia ei päästy työn aikana testaamaan oikealla kalustolla. Haasteista huolimatta eRetrofit-hankkeen, jonka osana opinnäytetyö suoritettiin, merkitys voi olla tulevaisuudessa merkittävä. Olemassa olevan ajoneuvokaluston sähköistäminen hankkeessa kehitettävällä tavalla voisi mahdollisesti auttaa Helsingin lisäksi myös muita kaupunkeja sekä yrityksiä kohti hiilineutraaliuden saavuttamista karSIMalla ajoneuvokaluston sähköistamisestä koituvia kustannuksia.

LÄHTEET

Helsingin kaupunki. 2020. Stara eRetrofit. Luettu 10.11.2021.

<https://www.hel.fi/helsinki/fi/kaupunki-ja-hallinto/strategia-ja-talous/innovaatiora-hasto/hankkeet/stara+eretrofit>

Jimenez, M., Palomera, R. & Couvertier, I. 2013 Introduction to embedded systems. New York: Springer.

Catsoulis, J. 2005. Designing embedded hardware. Sebastopol: O'Reilly Media

WatElectronics. 2021. Embedded Systems Role in Automobiles with Applications. Luettu 16.11.2021. <https://www.watelectronics.com/importance-of-embedded-systems-in-automobiles-with-applications/>

Aptiv. 2020. What Is an Electronic Control Unit? Luettu 16.11.2021.

<https://www.aptiv.com/en/insights/article/what-is-an-electronic-control-unit>

ECU Testing LTD. 2021. ECU Explained. Luettu 16.11.2021. <https://www.ecu-testing.com/categories/ecu-explained/>

Voss, W. 2008. A Comprehensive Guide to J1939. Greenfield: Copperhill Media Corporation.

CSS Electronics. 2021. CAN DBC File Explained – A Simple Intro. Luettu

21.11.2021. <https://www.csselectronics.com/pages/can-dbc-file-database-intro>

Kübel, M. 2015. Design Guide for Combined Charging System. Luettu

10.11.2021 http://tesla.o.auroraobjects.eu/Design_Guide_Combined_Charging_System_V3_1_1.pdf

Phoenix Contact. 2021. Combined Charging System (CCS). Luettu 16.11.2021.

<https://www.phoenixcontact.com/online/port>

https://www.phoenixcontact.com/online/portal/fi?1dmy&urile=wcm%3apath%3a/fifi/web/main/products/subcategory_pages/Charging_cables_and_charging_sockets_P-29-03/3f34965c-f842-4adc-a9f6-28126dc0a51a/3f34965c-f842-4adc-a9f6-28126dc0a51a

Mültin, M. 2018. ISO 15118 as the Enabler of Vehicle-to-Grid Applications. Lu-

ettu 20.11.2021. <https://www.semanticscholar.org/paper/ISO-15118-as-the-Enabler-of-Vehicle-to-Grid-M%C3%BCltin/54c170d4989dfc14b6ee94848750cd3097d3d3e0>

OpenECU. 2021. Introducing M580. Luettu 20.11.2021.

<https://www.openecu.com/product/m580/>

Epec. 2021. Epec 6107 Display Unit. Luettu 20.11.2021.

<https://epec.fi/products/display-6107/>

Danfoss. 2019. EC-C1200-450 MOTOR INVERTER – Communication manual.

Danfoss. 2021. Electric converters. Luettu 20.11.2021. <https://www.danfoss.com/en/products/dps/electric-converters-and-machines/electric-converters-and-machines/electric-converters/#tab-overview>

Orion BMS. 2021. Orion BMS 2. Luettu 20.11.2021. <https://www.orionbms.com/products/orion-bms-standard/>

Orion BMS. 2021. Orion BMS 2 Specifications. Luettu 20.11.2021. http://www.orionbms.com/downloads/documents/orionbms2_specifications.pdf

Sevcon. 2015. HVLP-10/20 Applications Reference Manual. Luettu 20.11.2021. <https://hsct.nl/wp-content/uploads/2015/03/HVLP-Product-Manual-v1.1.pdf>

Phoenix Contact. 2021. Vehicle Connector test adapter. Luettu 20.11.2021. <https://www.phoenixcontact.com/online/portal/fi?uri=pxc-oc-itemdetail:pid=1410506&library=fifi&tab=1>

Omron Industrial Automation. 2021. G2R-1-SN 120VAC (S). Luettu 20.11.2021. <https://industrial.omron.fi/fi/products/G2R-1-SN-120VAC-S>

MathWorks. 2021. Simulink. Luettu 20.11.2021. <https://mathworks.com/products/simulink.html>

MathWorks. 2021. Simulink Coder. Luettu 20.11.2021. <https://mathworks.com/products/simulink-coder.html>

Vector. 2021. CANdb++. Luettu 20.11.2021. <https://www.vector.com/int/en/products/products-a-z/software/candb/>

OpenECU. 2021. Pi Snoop. Luettu 20.5.2021. <https://www.openecu.com/product/pisnoop/>

CODESYS Group. 2021. CODESYS. Luettu 20.11.2021. <https://www.codesys.com/the-system.html>