Hydrauliikalla toimivan hybridiajoneuvon automaatio

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Sähkö- ja automaatiotekniikan opinnäytetyö

Valkeakoski, Sähkö- ja automaatiotekniikka

Syksy, 2019

Jere Lehtonen

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Sähkö- ja automaatiotekniikka
Valkeakoski

Tekijä Jere Lehtonen **Vuosi** 2019

Työn nimi Hydrauliikalla toimivan hybridiajoneuvon automaatio

Työn ohjaaja/t Mika Oinonen

TIIVISTELMÄ

Automaatioratkaisut helpottavat käyttäjiä erilaisten laitteiden sekä ajoneuvojen käytössä. Opinnäytetyössä yhdistyy automaatio sekä hydrauliikka, jota käytetään suunnittelemalla ja toteuttamalla hydrauliikalla toimiva hybridiajoneuvo. Opinnäytetyössä tutkittiin hydrauliikalla toimivan hybridiajoneuvon toimintaa, kuten mitkä eri vaikuttajat on otettava huomioon, jotta hybridiominaisuus käynnistyy järjestelmässä. Akunlataus ominaisuus aktivoituu vain kun ajoneuvo on liikkeellä, kierrosnopeus on yli 1200 rpm, akun paine on alle 210bar ja jarrua painetaan. Mitä suuremmaksi paine kasvoi akussa, jarrutusvoimakkuus kasvoi myös. Täten proportionaaliventtiili sulkeutuu lineaarisesti paineen kohotessa. Toiminta perustuu hydraulisen paineen siirtymiseen ajoneuvon voimansiirtoon, jolloin tuleva hydraulinen paine edesauttaa kiihdyttäessä. Myös jarrutuksessa kyseinen ominaisuus suoriutuu, mutta edesauttaen jarrutusta.

Akunpurkauksen ehtoina olivat, että kaasupoljinta on painettu, kytkintä ja jarrua ei ole painettu, kierrosnopeus on 1200 rpm ja että paineakussa on painetta yli 80bar. Kyseiset parametrin on säädetty turvallisuuden vuoksi, jotta testaus on turvallista ajoneuvoa ajaessa. CAN-väylän kommunikointi perustuu ohjelmoitavan logiikan ja näytön välille. Työssä on käytetty IFM valmistajan tuotteita. Väylän kommunikointi on toteutettu verkkomuuttuja listoilla. Lähetettäville ja vastaanotettaville kirjastoille on määritetty omat kommunikointi objektin tunnisteet, jolloin logiikka ja näyttö lähettävät ja vastaanottavat viestejä.

Avainsanat Automaatio, hybridi, ohjelmointi

Sivut 26 sivua, joista liitteitä 0 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Electrical and Automation Engineering
Valkeakoski

Author Jere Lehtonen **Year** 2019

Subject Automation of a hydraulic hybrid vehicle

Supervisors Mika Oinonen

ABSTRACT

Automation solutions ease users with different devices and vehicle applications. In this project, automation and hydraulics were merge, as concerns by planning and executing a hybrid vehicle powered by hydraulics. In this thesis we look closer at the different stages of the project and the implementation of a vehicle function, including which different factors need must be considered in order for the hybrid starts in the system. The pressure accumulator loading feature activates only when vehicle is in motion, rounds per minute is over 1200 rpm, pressure accumulator is less than 210 bar and the brake is applied. The higher pressure increased; braking force increased also. Hereby proportional control valve closes linearly when pressure increases. Operation is based on the transfer of hydraulic pressure in to the vehicle's transmission, whereby the incoming hydraulic pressure helps vehicle to accelerate. Also, in braking, this feature performs, but facilitates braking.

The conditions for the pressure accumulator discharge were that the accelerator pedal was depressed, the clutch and brake were not pressed, the speed was over 1200 rpm and the pressure accumulators' pressure was over 80 bar. These parameters have been adjusted for safety to ensure that the testing is safe while driving the vehicle. CAN bus communication is based on programmable logic controller and display. The products of the IFM manufacturer have been used in the work. Bus communication is implemented with network variable lists. The send and receive libraries have their own identifiers for the communication object, so that the logic and display send and receive messages.

Keywords Automation, hybrid, programming

Pages 26 pages including appendices 0 pages

SISÄLLYS

1 JOHDANTO

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa Dynaset Oy:lle automaatio-ohjelma hydrauliikalla toimiva hybridiominaisuus Volkswagen Transporteriin. Ohjelmointi suoritettiin käyttäen CODESYS 2.3V -ohjelmointiympäristöä.

Opinnäytetyön tarkoituksena on valmistaa ohjelma, jota käytetään Volkswagen Transporter –hybridiajoneuvossa. Projektin tuotos on suunniteltu valmistuvan huhtikuussa 2019, jolloin se on tarkoitus esitellä Saksan messuille Baumaan esiteltäväksi. Projektin päävaiheisiin kuuluu ohjelmointi, ohjelman testaus testialustalla ja ohjelman testaus ajoneuvossa.

Projektissa alueekseni kuuluu ohjelman tekeminen ohjausjärjestelmään sekä näytölle. Näille kummallekin oli suunniteltava ja toteutettava ohjelmisto. Ensimmäiseksi toteutetaan ohjelman toiminta lohkokaaviolla, jotta ohjelmien toiminta on tiedossa. Ohjelmien tekeminen myös helpottuu, kun lähtötiedot ovat kunnossa. Tärkeimmäksi kohdaksi työssä osoittautuu ohjelmoitavan logiikan ohjelmointi, koska ohjaus toteutetaan tällä. Näytön ohjelmointiin tarvittavat kriteerit ovat eri parametriarvojen näyttäminen käyttäjälle, jotta käyttäjä on myös tietoinen ohjelman toiminnasta ja tarvittaessa pysäyttämään ja käynnistämään ohjelman.

Haasteeksi projektissa osoittautuu uuden ohjelmiston omaksuminen. Aikaisempaa käyttöä CODESYS 2.3V -ohjelmointiympäristöstä ei minulla ollut. Kuitenkin en pitänyt sitä ylitsepääsemättömänä haasteena, koska ohjelmointiohjelmistot ovat jotenkuten samankaltaisia, jokaisessa tietenkin erilaisia eri ominaisuuksia.

2 PROJEKTI

Projektin suunnitelmana oli suunnitella ja valmistaa hydrauliikalla toimiva hybridiajoneuvo. Projektissa vastuualueekseni projektiin kuului suunnitella ja toteuttaa ohjelma, joka määrittää, milloin paineakusta tulevaa painetta käytetään akun lataamiseen ja myös käyttöön. Autosta kerätään tarvittavaa infoa, kuten kierrosluku, nopeus, paineanturin painetieto, kaasupolkimen asento, kytkin- ja jarrutieto. Kuvassa yksi on esitetty auto asennusvaiheessa.

Kuva 1. Volkswagen Transporter asennuksessa.

Työn tilaajana on Dynaset Oy. Dynaset Oy on maailman johtaviin hydrauligeneraattoreiden, -korkeapainepesureiden ja -kompressoreiden valmistaja. Dynaset-hydrauliikkalaitteet muuttavat liikkuvan työkoneen hydraulisen voiman sähköksi, korkeapainevedeksi, paineilmaksi, magneetiksi sekä myös tärinäksi. Dynaset tuotteita on käytössä ympäri maailmaa sadoissa erilaisissa tuotesovelluksissa. (Dynaset Oy. 2019).

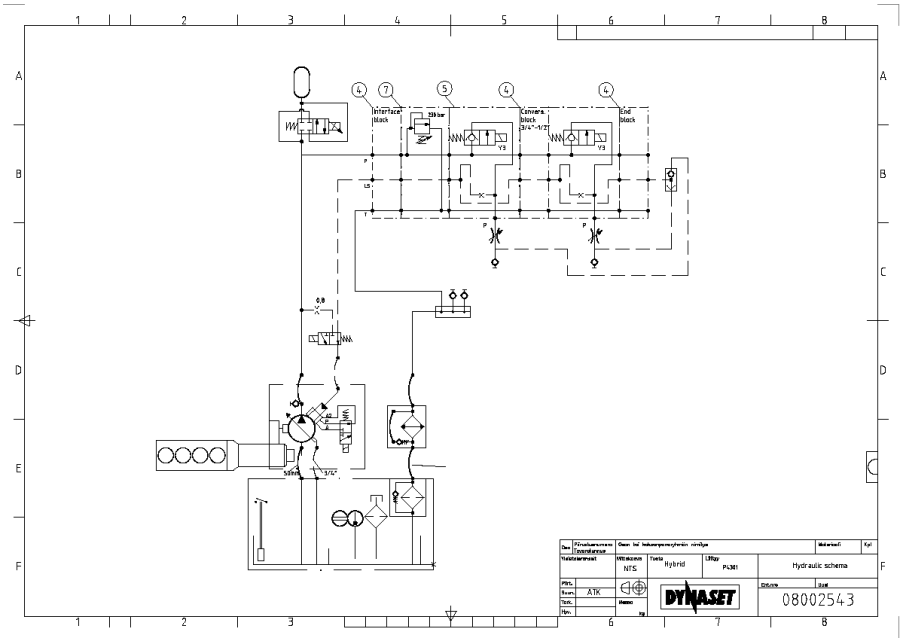
Dynasetin teknologia lisää erilaisten työkoneiden tuottavuutta, pienistä ajoneuvoista suuriin työkoneisiin. Teknologia perustuu työkoneen hydraulijärjestelmän hyödyntämiseen voimanlähteenä. Hydraulilaitteet toimivat tällöin ilman ylimääräisiä moottoreita, määräaikaishuoltoja ja päästöjä, tehden hydrauliikkalaitteista kaikkein ekologisimman vaihtoehdon. (Dynaset Oy, 2019).

3 HYBRIDIAJONEUVOTEKNIIKKA

Hybriditeknologia yhdistää ajoneuvossa polttomoottorin sekä sähkömoottorin. Hybriditeknologian tarkoitus on, että se kerää polttomoottorilta energiaa sähkömoottoriin. Normaaleissa autoissa jarrutusenergia menee hukkaan, kun taas hybridiajoneuvoissa jarrutusenergia kertyy sähkömoottorin käytettäväksi.

Hybridiajoneuvot sisältävät kaksi moottoria: bensiinikäyttöinen polttomoottori ja sähkömoottori. Useimmissa hybridiajoneuvoissa voidaan itse määrittää, milloin hybridiominaisuutta käytetään. Esimerkiksi kaupunkiajossa ja ylämäissä hybridiominaisuus on erinomainen. Tällöin suorituskyky ei hidastu ja polttoainekulutus pienenee huomattavasti. Autovalmistajilla on runsas määrä erilaisia variaatioita hybridiominaisuudelle esimerkiksi eco-tila, jolloin käytetään vain sähkömoottoria ja sport-tila, jolloin molemmat moottorit käyttävät maksimitehoa kiihdyttäessä. (Toyota, n.d.)

Projektimme tarkoitus on valmistaa Volgsvagen Transporterista hybridiajoneuvo, mutta sähkömoottorin sijaan käytämme hydrauliikkaa tuottamaan ylimääräisen tehon ajamisessa. Ajatusmalli on samanlainen kuin aikaisemmin kerrotussa sähkömoottorilla toimivassa hybridiajoneuvoissa, mutta esimerkiksi latauksessa syntyvää öljypainetta varastoidaan paineakulle, jota käytetään auton kiihdytyksessä. Kuvassa kaksi on esitetty hydrauliikkakaavio järjestelmästä.



Kuva 2. Hydrauliikkakaavio.

Projektissa käytetty hydraulijärjestelmä on avoin järjestelmä. Tämä tarkoittaa sitä, että öljy imetään pumpulla järjestelmään öljysäiliöstä, jonne se palaa takaisin toimilaitteilta. Tässä järjestelmässä ideana on, että säätötilavuuksinen hydraulipumppu ajetaan negatiiviselle kulmalle, jolloin se alkaa toimimaan moottorina. Toisin sanoin tämä tarkoittaa sitä, että jarruttaessa pumppu pumppaa paineakkuun painetta ja kiihdyttäessä paine puretaan pumppuun, jolloin pumppua käytetään moottorin tavoin. Tämä hoidetaan proportionaalisella suuntaventtiilillä.

Hydrauliikkapumpun tehtävänä on muuttaa moottorilla tuotettu mekaanisesti tuotettu energia hydrauliseksi energiaksi, paineeksi ja tilavuusvirraksi. Hydrauliikkapumpulle on asetettu erilaisia vaatimuksia, kuten käytetty neste, painealue, pyörimisnopeus ja tilavuusvirrantuotto. Hydrauliikkajärjestelmään kuuluu erilaisia paine-, säätö- ja vuotolinjoja. Paineenrajoitusventtiili rajoittaa hydrauliikan painetta järjestelmässä, joka on säädetty 210bar. Hydraulijärjestelmään kuuluu yksi virtaussuuntapumppu, jota käytetään myös moottorina. Ohjausventtileitä järjestelmässä on käytetty enemmän, näillä säädetään öljynvirtausta ja suuntaa.

4 PLC-OHJELMOINTI

Ohjelmointisovelluksena käytettiin codesys V2.3. Codesys on saksalaisen 3S-Smart Software Solutions automaatioyrityksen tuote. Codesys on suunniteltu logiikkaohjelmointiin ja käytettäväksi logiikoita sisältävässä automaatiossa. Codesys täyttää IEC 61131 standardin, joka on International Electrotechnical Comissionin laatima standardi ohjelmoitavasta logiikasta. (Codesys, 2011)

Kyseistä versiota käytettiin projektissa, koska 2.3 versiossa on mahdollista ohjelmoida käyttäen CR0403 ohjausjärjestelmää. Ohjausjärjestelmästä kerron lisää luvussa 5.1. Ohjelmoitavina komponentteina toimivat IFM valmistajan ohjausjärjestelmä CR0403 sekä näyttöpaneeli CR0451.

4.1 Codesys

Codesys on 3S-Smart Software Solutions yrityksen kehittelemä ohjelmointisovellus. Codesys on suunniteltu käytettäväksi logiikkaohjelmointiin ja myös kaikkiin logiikoita sisältävässä automaatiossa. Pääasiassa ohjelmaa käytetään liikkuvien työkoneiden ohjelmoinnissa, joten ohjelmisto kyseisessä projektissa sopii täydellisesti käytettäväksi. (Suhonen, 2011)

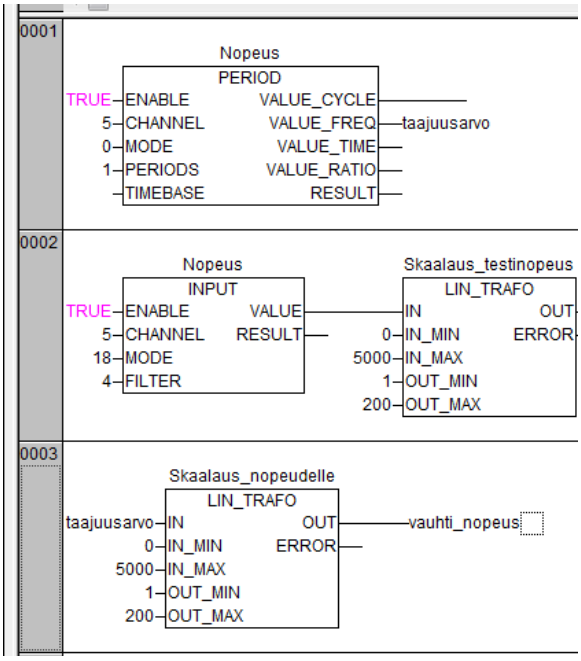
4.2 Ohjelmointikielet

Codesysissä on valittavana erilaisia ohjelmointikieliä. Ohjelmoija pystyy valitsemaan jo ennestään tutun ohjelmointikielen. Codesys ohjelmassa jokainen eri aliohjelma voi olla tehtynä eri ohjelmointikielellä ja kokonaisuus toimii kuitenkin ongelmitta. Erilaisia ohjelmointikieliä on:

- Structured Text (ST)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Ínstruction List (IL)
- Sequential Function Chart (SFC)
- Continous Functrion Chart (CFC)

Käytössäni oli kaksi ohjelmointikieltä: Structured text ja function block diagram. Structured text on C-ohjelmointikielen kaltainen ohjelmointityyli. Kyseinen ohjelmointikieli oli paras vaihtoehto, koska ohjelmointikielellä on mahdollista tehdä yksinkertaisempaa koodia kuin muilla ohjelmointikielillä. Projektissa käytin ST-kieltä kaikissa aliohjelmissa, paitsi sisääntulojen ja ulosmenojen aliohjelmissa.

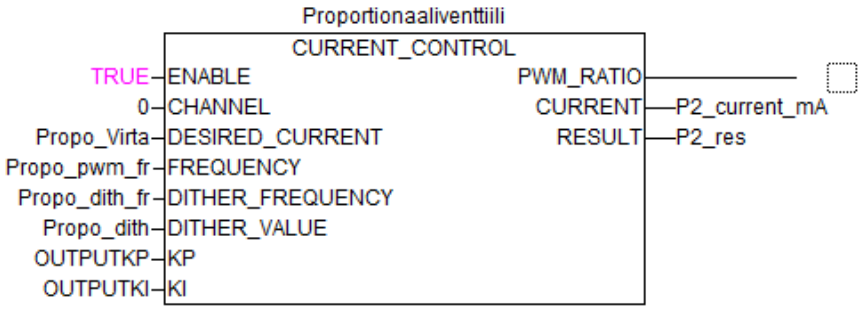
Function block diagram on erittäin havainnollinen, helpompi lukea ja käsittää kuin structured text ohjelmointikieli. Valmiiksi määritettyjä toimilohkoja oli erittäin paljon saatavilla. Tässä projektissa käytin ainoastaan sisäänmeno-, ulostulo- ja skaalaustoimilohkoja. Kuvassa kolme nähdään ohjelmassa olevien sisäänmenojen toimilohkot.



Kuva 3. Projektin sisäänmenot käyttäen toimintalohkoja.

Toimintalohkoille oli määritettävä sisäänmeno eli minkälaista arvoa toimilohkoista tulee ja mikä on filtteröintitaajuus. Työssä käytimme kaikissa toimilohkoissa 10Hz filtteröintitaajuutta, koska se oli suositeltava filtteröintitaajuus kyseisille toimintalohkoille.

Projektiin valitsimme teollisuusproportionaaliventtiilin D1FBE01KK0NKW3 kaksireunakytkennällä. Maksimipaineenkesto kyseiselle proportionaaliventtiilille on 210bar, joka sopi täydellisesti valitsemaamme paineakkuun. Proportionaaliventtiilin kelavirta täysin auki on 2,2A joten ohjelmassa määritettiin, että esimerkiksi akun purkautuessa proportionaaliventtiilin kelavirraksi asetetaan 2,2A. Tällöin proportionaaliventtiili on täysin auki, jolloin öljy virtaa paineakusta tuottaen tehoa auton kiihdytykseen.



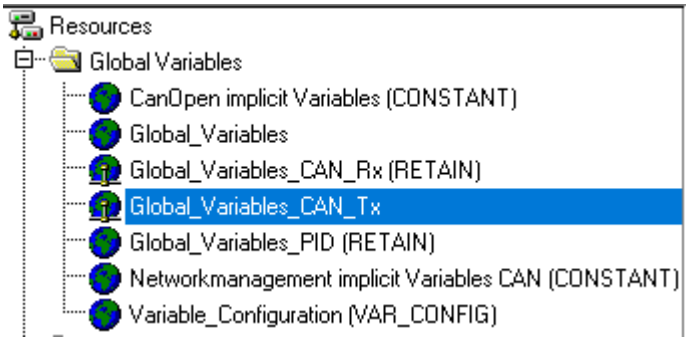
Kuva 4. Proportionaaliventtiilin ohjauksen toimilohko.

Kuten kuvassa 4 huomataan, toimilohkon sisäänmenoissa oleva DESIRED_CURRENT määrittää proportionaaliventtiilin kelavirran. Akun purkaessa määritimme kelavirraksi 2,2A, jolloin proportionaaliventtiili on täysin auki. Akkua ladatessa ilmeni, että mitä korkeammaksi paine kasvaa paineakussa sitä enemmän jarrutus korkean öljynpaineen takia kasvoi. Tämän takia ohjelmassa akun lataukselle määritettiin, että mitä korkeammaksi paine kasvaa paineakussa, sitä pienemmäksi proportionaaliventtiilin kelavirta pienenee.

4.3 **CAN-väylä**

CAN-väylä mahdollistaa eri laitteiden kommunikoinnin. Ifm valmistajan ohjausjärjestelmät ja näyttöpaneelit kommunikoivat CAN-kommunikoinnilla. Väylään yhdistettyjä laitteita voi olla esimerkiksi antureita tai muita tiedonkeruulaitteita. CAN-väylä on suunniteltu nopeaksi tiedonsiirroksi. CAN-väylässä liikkuvat tietomäärät ovat pieniä, joten suurien tiedonsiirtojen kohdalla olisi syytä harkita jotain muuta menetelmää kuin CAN-väylää. Väylän fyysinen muoto on kierretty parikaapeli. Kaapelin tulee olla laadukas ja suojattu, sekä liittimien laatuun tulee kiinnittää huomiota, jotta yhteys säilyy. Väylän molemmissa päissä on 120 ohmin päätevastukset, jotta väylän liikenne ei heijasta signaalia. Päätevastusten tehtävänä on estää signaalin heijastuminen takaisin. Signaalin takaisinheijastuminen on yleinen ongelma korkealaatuisilla signaaleilla. Tiedonsiirtoa väylällä ei välttämättä pysäytä sähkömagneettinen säteily, koska sähkömagneettinen säteilyn aiheuttama jännitteen muutos vaikuttaa molempiin johtimiin, jolloin niiden välinen jännite-ero pysyy samana. (Suhonen, 2011)

CAN-väylän ansiosta ohjausjärjestelmän ohjelma pystyy keskustelemaan näytön ohjelman avulla. Jotta kyseinen keskustelu olisi mahdollista, ohjelman gloobaleissa muuttujissa sijaitsee globaali muuttujalista, jossa voidaan valita, lähettääkö se muuttujien arvoja näytölle tai tuleeko näytöltä saadut tiedot, kuten käynnistys ominaisuus näytöltä ohjausjärjestelmään.



Kuva 5. CAN-Väylän muuttujille tarkoitettu muuttujalista. (Codesys, 2018.)

Kuten kuvassa 16 huomataan, CAN-väylälle on kaksi erilaista muuttujalistaa: CAN_Tx ja CAN_Rx. CAN_Tx:n sijaitsevat muuttujat lähettävät niihin sijoitettujen muuttujien tiedon, kun taas CAN_Rx:n sijoitetut muuttujat vastaanottavat näytöltä sijoitettujen muuttujien arvoja, kuten käynnistyksen.

Ehto, jotta CAN-väylä kommunikoi kahden eri ohjelman välillä on, että muuttujat on sijoitettu tismalleen samassa järjestyksessä molemmissa ohjelmissa. Jos muuttujat on sijoitettu eri järjestyksessä ohjelmien välillä, tällöin näytölle saadut arvot eivät näytä oikeita arvoja kuin ohjausjärjestelmässä. Ongelmaksi testauksessa ilmeni juuri kyseinen asia. Muuttujat näyttivät aivan eri lukemia näytöllä kuin mitä ohjausjärjestelmästä. Tämän takia ohjelmia muuttaessa piti aina varmistaa, että CAN-väyliin sijoitetut muuttujat ovat tismalleen samassa järjestyksessä.

4.4 Ohjelman määrittäminen

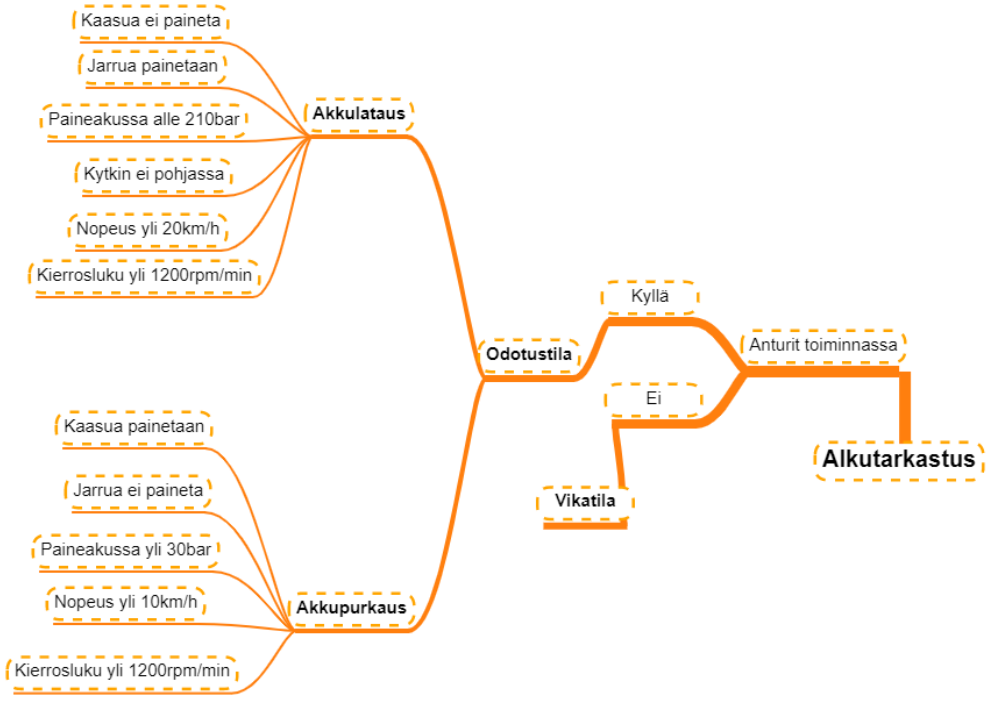
Ennen kuin itse ohjelmaa päästiin aloittamaan, oli ensin määritettävä projektiin I/O lista. Mitä kaikkea eri sisäänmenoja ja ulostuloja tarvitaan kyseiseen projektiin. Käytimme sisäänmenevinä muuttujina:

- Nopeus
- Jarru
- Kytkin
- Kaasupolkimen asento
- Start/Stop
- Kierrosluku.

Ja ulostuloissa on käytössä kolme eri muuttujaa: Paineakku, LS0-linja ja LS-linja.

Kun tarvittavat sisään -ja ulostulot oltiin määritetty, aloitin suunnittelemaan lohkokaaviota ohjelman toiminnasta. Suunnitelmana oli, että kun käyttäjä kytkee hybriditoiminnon päälle ajoneuvossa, ohjelma suorittaa alkutarkastuksen, jossa tarkastetaan antureiden toiminta. Jos anturi on vioittunut tai ei lähetä signaalia, menisi ohjelma heti vikatilaan, josta tulisi virheilmoitus näytölle.

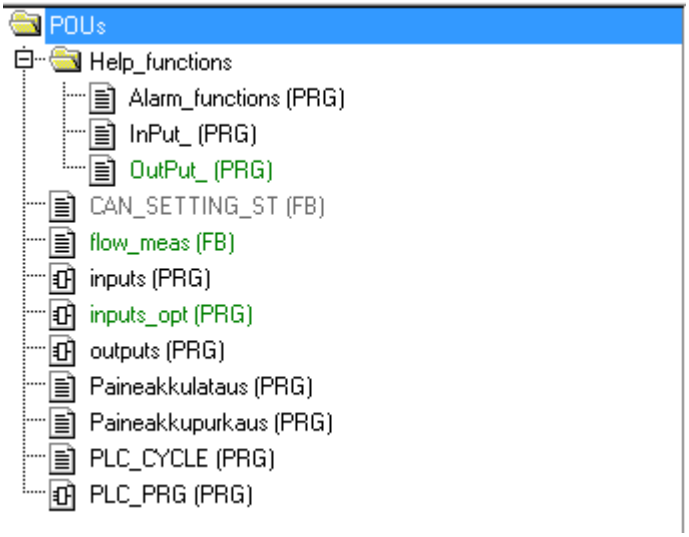
Kun anturit on tarkastettu toimivaksi, siirryttäisiin ns. odotustilaan. Odotustilasta siirryttäisiin joko akun latauskykliin tai akun purkauskykliin. Kuvassa 6 on selkeytetty kokonaisuudessaan ohjelman toiminta.



Kuva 6. Lohkokaavio.

4.5 **Ohjelma**

Kun sain määritettyä ohjelman toiminnan paperille, aloitin itse ohjelmoinnin. Käytin ohjelman pohjana aikaisempaa projektia, koska projektissa oli tarvittavat kirjastot CAN-väylälle ja logiikalle. Näin säästettiin aikaa. Aluksi luotiin tarvittavat aliohjelmat rakennepuuhun, Program organization unit: tiin (POU), joka on käytössä kuvassa 7 . Tarvittavia aliohjelmia pystyi lisäämään tarpeen mukaan.



Kuva 7. Ohjelman rakennepuu.

Hälytykselle, paineakun lataukselle ja purkaukselle tein omat aliohjelmat, joita pääohjelma (PLC_PRG) kutsuu. Aikaisemmassa projektissa oli valmiiksi konfiguroitu CAN-väylän asetukset, jolloin niitä ei tarvittu luoda uudelleen. Alla olevissa otsikoissa kerrotaan tarkemmin aliohjelmien toiminnasta.

4.5.1 Akkulataus

”Paineakkulataus”-aliohjelma määrittää, milloin paineakkuun varastoidaan painetta. Yksinkertaisimmillaan paineakkua ladataan, kun ajoneuvon jarrua painetaan. Paineakun lataus keskeytyy, kun ajoneuvon kierrosnopeus tippuu alle 1200 rpm. Paineakussa on yli haluttu virtauspaine (210 bar), kaasua painetaan ja jos kytkintä painetaan. Aliohjelmassa on myös käytetty ramp_int-toimintoa, jonka avulla proportionaaliventtiili siirtyy pienellä viiveellä latausasentoon. Samaa toimintoa käytettiin myös akun purkamisessa purkaus asentoon. Ohjelmaan lisättiin myös ominaisuus akunlataukselle. Mitä korkeammaksi paine nousee paineakussa, sitä pienemmäksi proportionaaliventtiili avautuu, jolloin paineen lataus pienenee.

4.5.2 Akkupurkaus

”Paineakkupurkaus”-aliohjelma määrittää, milloin paineakusta käytetään painetta. Kun akusta käytetään painetta, nopeuden on oltava yli kymmenen kilometriä tunnissa. Paineakussa on oltava enemmän kuin 30bar painetta, kaasupoljinta on painettava ja kuten aikaisemmassa aliohjelmassa kytkintä tai jarrua ei saa olla painettuna. Ohjelma keskeyttää heti latauksen sekä purkauksen, jos kytkintä tai jarrua painetaan.

4.5.3 PLC_cycle

Kun tarvittavat aliohjelmat oltiin saatu tehtyä, piti ohjelmalle määrittää, milloin se siirtyy aliohjelmasta toiseen. Tämän ominaisuuden täyttää PLC_cycle aliohjelma. Aliohjelmaan määritettiin case rakenteella, mihinkä eri tilaan ohjelmassa siirrytään. Nämä eri tilat ovat esitetty kuvassa 8. Aluksi luotiin tarvittavat syklitilat: Virheilmoitus, tarkastus, odotustila, paineakunpurkaus, paineakunlataus ja viimeiseksi odotustila. Virheilmoituksen sattuessa ohjelma siirtyy suoraan siihen määritettyyn syklitilaan. Kun virheilmoitusta ei tapahdu, ohjelmassa ollaan yleensä odotustilassa. Kun latauksen tai käytön muuttujat käyvät toteen, siirrytään syklitilasta toiseen. Kun ohjelmaa ei olla käynnistetty, syklitila on tällöin odotustila.

```
CASE Cyclestate OF

    100: (*Virheilmoitus*)

    0: (*Tarkastus*)


    1: (*Odotustila*)


    2: (*Paineakkupurkaus*)


    3: (*Paineakkulataus*)


    4: (*Ohjelmaa ei suoriteta, eli stop on päällä*)


    END_CASE
```

Kuva 8. PLC_cycle aliohjelman eri syklitilat.

5 TYÖSSÄ KÄYTETTÄVÄT KOMPONENTIT

On käytettävä tarvittavia komponentteja, jotta automaatio-ohjelma saadaan käyttöön ajoneuvossa. Tärkeimpinä komponentteina ovat CR0403 ohjausjärjestelmä, CR0451 grafiikkanäyttö ja VDO pedal interface. Ohjausjärjestelmällä pystytään käyttämään automaatio-ohjelmaa ja tarvittavia sisäänmenoja ja ulostuloja. CR0451 grafiikkanäytöllä voidaan autosta saadut tiedot näyttää näytöllä, jolloin auton käyttäjä pystyy seuraamaan hybridi ohjelman kulkua. Käytössä oli myös monia erilaisia komponentteja, mutta edellä mainitut osat olivat projektin pääkomponentit.

5.1 **CR0403**

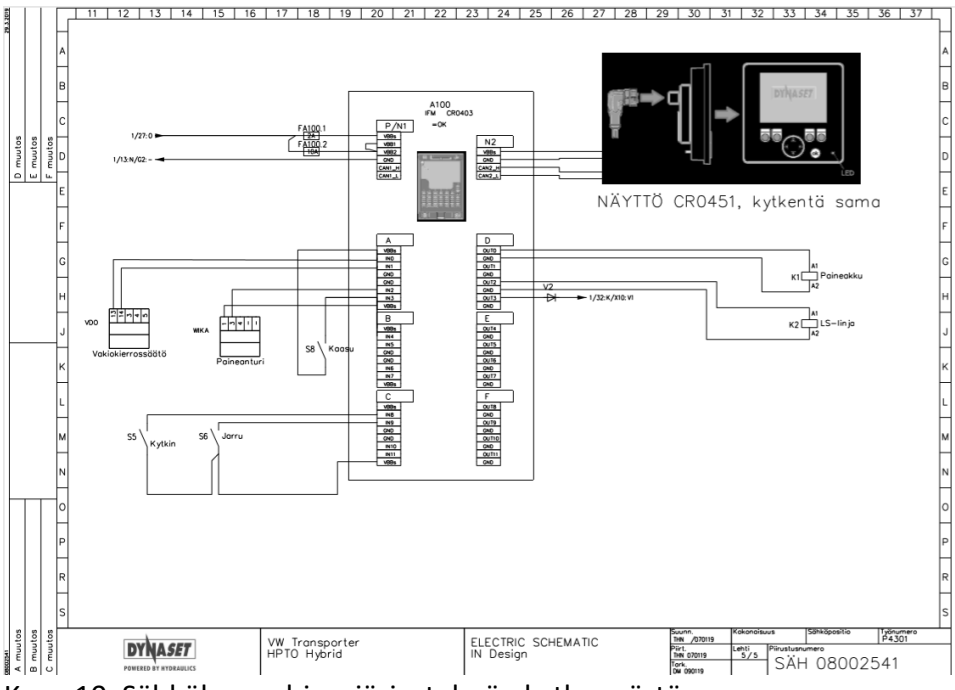
CR0403 on ohjelmoitava ohjausjärjestelmä. Se soveltuu parhaiten ajoneuvoihin ja muihin liikkuviin järjestelmiin. Ohjausjärjestelmää on mahdollista ohjelmoida vain codesys 2.3 versiolla. Tuloja ohjausjärjestelmässä on yhteensä 12, joista ensimmäisestä neljästä sisäänmenosta IN0 – IN3 voidaan tarkastella positiivista ja negatiivista sensorien signaalia, jännitettä, virtaa ja taajuutta.

Nopeuden ja kierrosluvun signaali oli mitattu taajuutena, kun taas kaasupolkimen asentoa mitattiin jänniteviestinä. Ohjausjärjestelmän sisäänmenoista IN4 – IN7 pystyttiin tarkastella binärituloa ja resistanssi arvoa. Lopuista sisäänmenoista IN8 – IN11 pystytään tarkastamaan vain binärituloa (ON/OFF). Näihin sisäänmenoihin sopi täydellisesti jarru ja kytkin, koska niistä haluttiin vain tarkastella milloin käyttäjä painaa kyseisiä polkimia autossa. (Ifm, 2015a)

Kuva 9. CR0403 ohjausjärjestelmä.

Kuten kuvassa 9 voidaan huomata, ohjausjärjestelmän kannesta voidaan nähdä, missä kukin sisäänmeno ja ulostulo sijaitsee. Tämän ansiosta asennus helpottui huomattavasti ja tarvittavat korjaukset olivat selkeästi korjattavissa. Kuvassa 10 esitetään ohjausjärjestelmän kytkentä.

Ulosmenoja ohjausjärjestelmässä on myös 12. Ulostuloina ohjelmassa oli yhteensä kolme: paineakulle, LS-linjalle ja LS0-linjalle.



Kuva 10. Sähkökuva ohjausjärjestelmän kytkennästä.

Näytössä sekä ohjausjärjestelmässä on sisäänrakennettu LED-valo, jolla voidaan tarkkailla komponenttien tiloja. Ledin väri sekä eri nopeudella vilkkuminen ilmaisee laitteen tilan. Kun vihreä valo vilkkuu 2 Hz nopeudella, laitteen sisälle ladattu ohjelma on päällä. Vihreän valon vilkkuessa 5 Hz nopeudella tarkoittaa, että ohjelma on ladattu laitteesseen, mutta se on stop tilassa. Kyseisestä tilasta päästään start tilaan käynnistämällä laite uudelleen tai suorittamalla Codesyssin kautta toiminto run, jolloin ohjelma käynnistyy ilman käynnistämättä laitetta uudelleen.

5.2 **CR0451**

CR0451 on ohjelmoitava grafiikkanäyttö liikkuvien työkoneiden ohjaukseen. Näyttö on myös ifm:n valmistama ja helposti yhdistettävissä CR0403 ohjausjärjestelmään. Selkeä värinäyttö mahdollistaa visualisoinnin siten, että tarvittavat tiedot nähdään näytöllä ja niitä on ajon aikana helppo tarkastella. Näytössä on viisi painonappia ja keinukytkin kursorin ohjaukseen. Näyttöön yhdistetään M12 liitäntäkaapeli, joka on yhteydessä ohjausjärjestelmään. Grafiikkanäytön resoluutio on 320 x 240. Kyseinen resoluutio sopii mainiosti kyseiseen projektiin. Kuvassa 11 esitetään grafiikkanäyttö. (Ifm, 2015)

Kuva 11. CR0451 Ohjelmoitava grafiikkanäyttö.

Tiedonsiirto suoritetaan käyttämällä CAN-väylää. Näyttö ja ohjausjärjestelmä keskustelevat tällöin ilman erillisiä komponentteja ja ohjelmia käyttäen.

Näytön ohjelmaan olisi ollut myös paljon erilaisia mahdollisuuksia, kuten sivujen vaihto, johon olisi mahdollista sijoittaa erilaisia tietoja, kuten latauksen ja purkausen määriä ja kuinka kauan kyseiset toiminnat ovat olleet käytössä, mutta projektin kiireellisyyden takia näytön ohjelmaksi tehtiin vain tarvittavat määritykset, joita ovat kaasupolkimen asento, kierrosluku, nopeus, sekä akun lataustila ja purkaustila.

5.3 **VDO Pedal Interface**

VDO kierrosnopeussäädin (Pedal Interface) on elektroninen kontrolleri, jonka toimintoihin kuuluu vakionopeudensäätö sekä nopeudenrajoitin. Kuvassa 12 esitetään kierrosnopeudensäätimen ulkoasu. Kierrosnopeussäädin soveltuu parhaiten isoihin ajoneuvoihin kuten rekkoihin sekä pakettiautoihin. Kierrosnopeussäätimestä on olemassa kolme eri versiota standard, enchanced -ja Premium malli. Kierrosnopeussäädin on turvallinen vaihtoehto isoihin ajoneuvoihin nopeuden rajoittimen ansiosta. Kierrosnopeussäätimen ansiosta polttoaineen kulutus myös pienenee. Kierrosnopeussäädin asennetaan kaasupolkimen ja moottorin ohjausyksikön väliin. Kierrosnopeussäätimelle on myös oma ohjelmisto, jolla voidaan itse määrittää esimerkiksi nopeudenrajoittimen maksiminopeus. (Autoalarm, 2015)

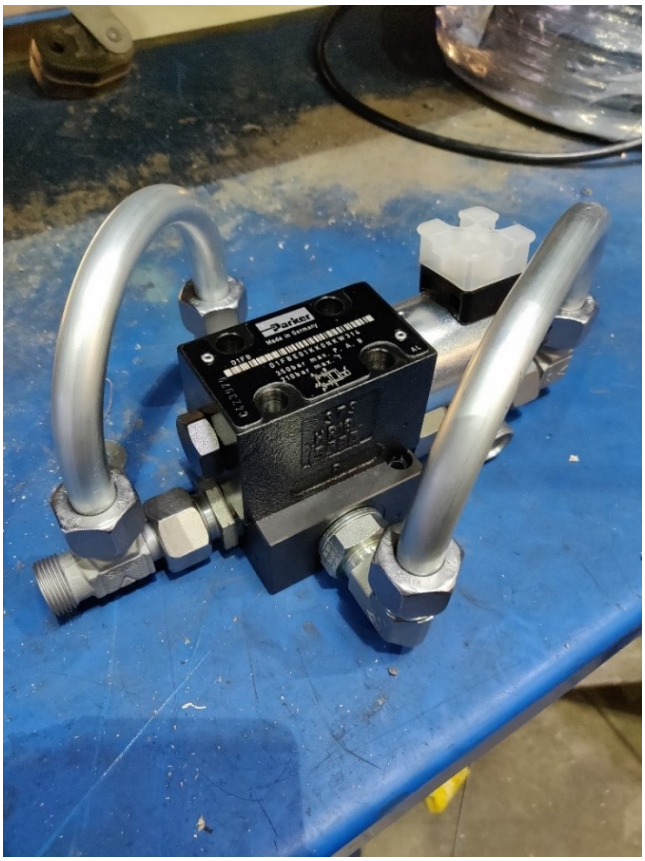
Kuva 12. VDO kierrosnopeussäädin. (Autoalarm. N.d).

5.4 **Hydrauliikkakomponentit**

Automaatio-ohjelman toteuttamiseen aikaisemmin mainitut komponentit ovat pääasiassa tärkeimmät osat, mutta itse ajoneuvoon kuului myös toimivuuden kannalta enemmän osia, kuten esimerkiksi proportionaaliventtiili sekä paineakku. Ilman kyseisiä osia ei öljynpainetta saataisi varastoiduttua taikka käytettyä.

Projektin paineakuksi valittiin 15L ja 210bar paineakku. Paineakku sijoitettiin auton pohjaan suuren koon vuoksi. Paineakkua testatessa huomattiin, että varastoitunutta painetta ei saatu ladattua paineakulle suuria määriä. Tällöin purkaus vaiheessa paineakku siirtyi täydestä tyhjäksi nopeasti. Isompi paineakku tälläisessä hydraulisessa hybridiajoneuvossa olisi suositeltavaa, mutta mitä isompaa paineakkua valitaan, sen enemmän tilaa se vie autosta. Kyseiseen projektiin valittu paineakku sopi hyvin, koska projektin tarkoituksena oli myös nähdä, onko hydrauliikalla mahdollista toteuttaa hybridiajoneuvoa.

Proportionaaliventtiili mahdollistaa tilavuusvirran suunnan ohjauksen, tilavuusvirran määrän ohjauksen ja järjestelmän paineen ohjauksen. Proportionaaliventtiilin etuna on, että niiden ohjaamia suureita eli joko tilavuusvirtaa tai painetta voidaan ohjata portaattomasti. Tavallisissa venttiileissä on yleisesti käytössä ON/OFF-tyyppistä kahden tai useamman tilan välistä ohjausta. Kuvassa 13 esitetään proportionaaliventtiilin ulkoasu.

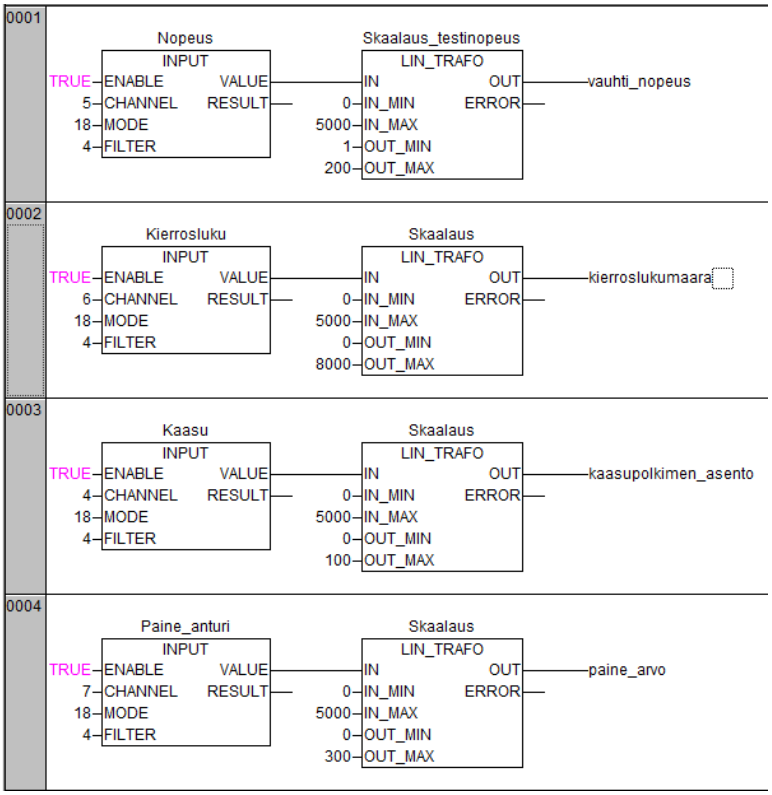


Kuva 13. Työssä käytetty proportionaaliventtiili.

6 OHJELMAN TESTAUS

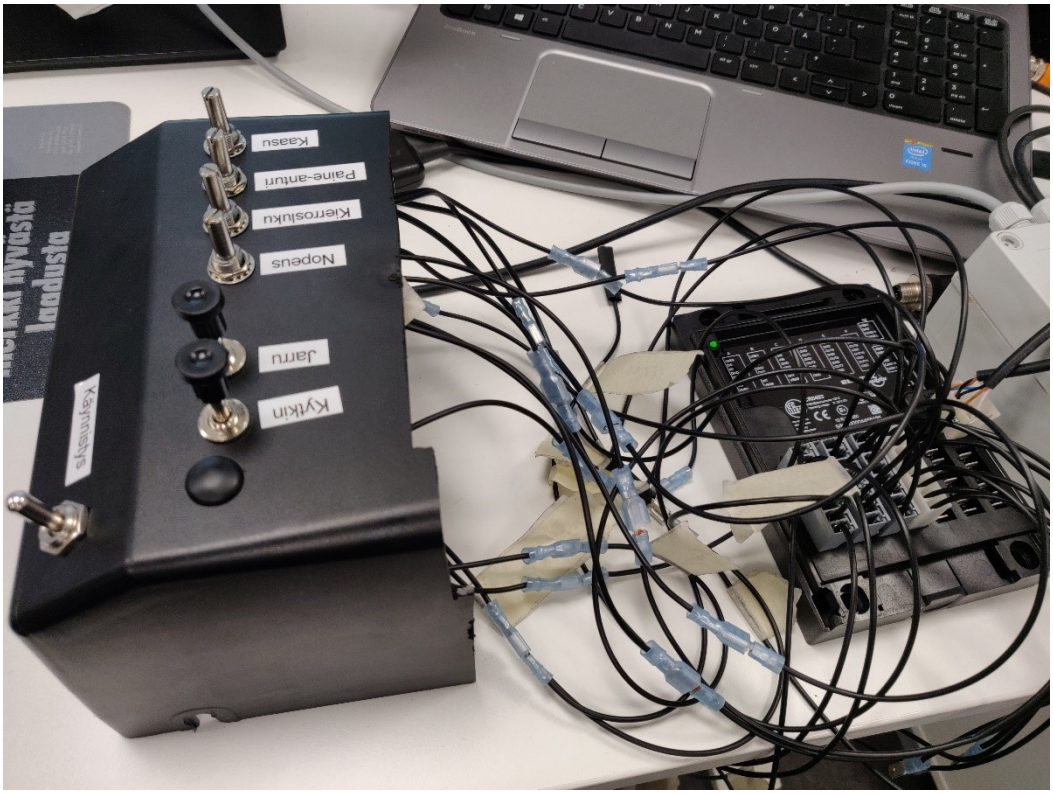
Kun ohjelma oltiin saatu valmiiksi, siirryttiin seuraavaksi sen testaamiseen. Jotta ohjelmaa pystyttiin testaamaan, tarvitsi sille rakentaa ns. Testialusta. Testialustan tarkoituksena oli muuttaa sisäänmenojen arvoja ja nähdä, täyttyykö akun lataamiseen ja purkamiseen täytetyt ehdot. Testialustassa käytimme viiden kilo-ohmin potentiometrejä. Jarrulle, kytkimelle ja käynnistys muuttujalle käytettiin ON/OFF-kytkimiä.

Sisäänmenoja tarvitsi muuttaa, jotta ohjausjärjestelmä pystyi lukemaan potentiometrin arvoa.



Kuva 14. Sisäänmenot testialustalle.

Kuten kuvassa 14 voidaan huomata, jokaisen sisäänmenon attribuutit ovat samanlaiset, ainoastaan sisäänmenokanavat ovat jokaiselle muuttujalle eri paikassa. MODE osioon ollaan määritetty, että sisäänmenosta luetaan resistanssia ja filtteröinnissä ollaan käytetty siihen suositeltavaa arvoa (10Hz). Jokaisessa sisäänmenossa on käytetty skaalaus toimilohkoa, jolloin lukema saadaan näyttämään samalta kuin ajoneuvossa käytetyt lukemat.



Kuva 15. Testialusta yhdistettynä ohjausjärjestelmään.

Ohjelman testaus käyttäen kuvan 15 testialustaa oli mielestäni erittäin järkevää, koska tällöin saatiin ohjelmaa muutettua, jotta se toimii halutulla tavalla. Ohjelma simuloitiin tietokoneella ja simuloinnissa säädettiin potentiometrejä ja katsottiin, että jokainen toiminto ohjelmassa toimii halutulla tavalla.

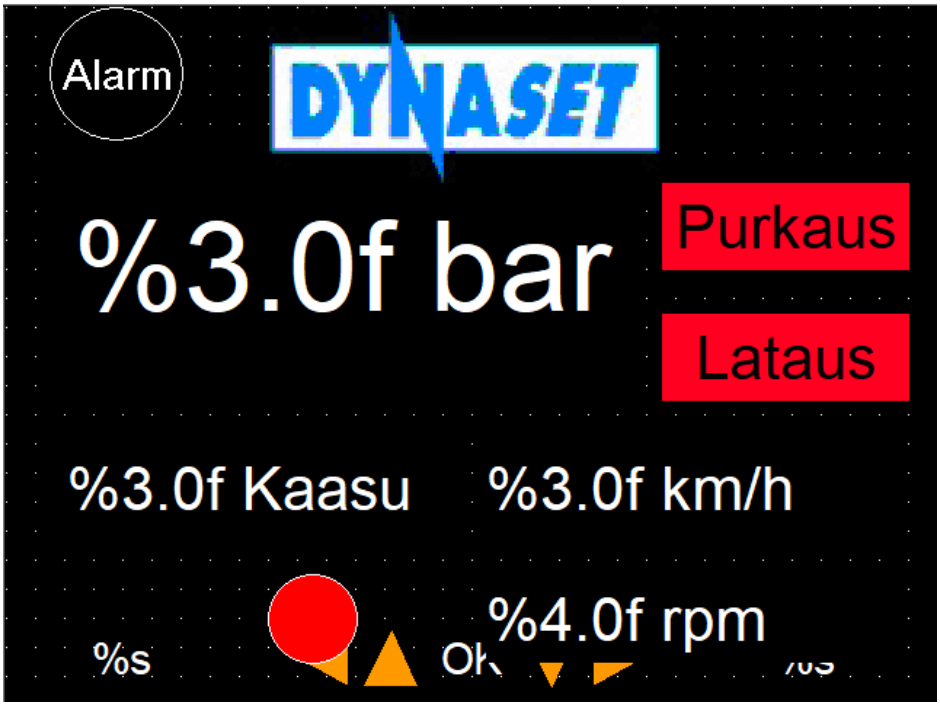
Testausvaiheessa ilmeni myös kytkentävirheitä testialustalla, jolloin aikaa kului odotettua enemmän. Kyseiset kytkentävirheet saatiin kuitenkin selvitettyä ja korjattua. Myös skaalausten minimi- ja maksimiarvoja muutettiin, koska ongelmaksi ilmeni, että jos potentiometrin arvoa muutettiin maksimiarvoksi (5kΩ). Tällöin arvo simulointivaiheessa muuttui takaisin nollaksi, koska se ylitti skaalauksessa maksimiarvoksi määritetyn arvon.

Ohjelma ladattiin ohjausjärjestelmään Ifm:n omalla sovelluksella. Sovelluksessa määritettiin, mitä ohjausjärjestelmää käytettiin ja ladattiin ohjelma kyseiseen ohjausjärjestelmään. Ohjelmassa aluksi tunnistettiin ohjausjärjestelmä, jota käytetään (tässä tapauksessa cr0403). Kun ohjelma tunnistaa ohjausjärjestelmän, ladataan siihen ohjelma. Kyseisessä ohjelmassa myös voidaan tarkastella, mikä ohjelma ohjausjärjestelmään on ladattu. Sillä saadaan myös todettua, että haluttu ohjelma on ladattu onnistuneesti ohjausjärjestelmään.

7 OHJELMA LCD-NÄYTÖSSÄ

Seuraavaksi vaiheeksi projektissa oli grafiikkanäytön lisääminen. Näyttöä varten luotiin oma ohjelma, jossa määritettiin näytölle visualisointi ja painonapeille tarvittavat komennot. Käynnistys siirrettiin näytön painonapeille ja tarvittavat muuttujat vaihdettiin Codesyssin CAN-väylän globaaleihin muuttujiin. Luvussa 7.1 käydään läpi, miksi näin tehtiin.

Visualisoinnissa muuttujat sijoitettiin näytölle siten, että kuljettajan on helppo havaita tiedot ja milloin akkua puretaan tai akkua ladataan. Näytöllä sijaitseva punainen ympyrä kertoo käynnistyksen tilan, painettaessa se siirtyy aktiiviseksi, jolloin väriksi muuttuu virheä. Myös purkaus ja lataus muuttuvat vihreäksi, kun jompikumpi niistä toteutuu. Kuvassa 16 on esitetty grafiikkanäytön ulkoasu.



Kuva 16. Grafiikkanäytön valikko.

Näytöllä sijaitsevat arvot näkyvät kokonaislukuina, koska kyseisien arvojen tarkka lukema ei ollut näytöllä tarpeellinen. Näyttöön on sijoitettu kaikki tarvittavat tiedot, jolla saadaan selville missä tilassa hybridiohjelmaa suoritetaan. Näyttöön on mahdollista ohjelmoida monta eri sivua ja valikkoa, mutta kyseisessä projektissa sitä ei nähty tarpeelliseksi.

Kun näytölle saatiin tehtyä ohjelma, testattiin sen toimivuus. Näyttö yhdistettiin ohjausjärjestelmään ja näytölle ladattiin ohjelma aikaisemmin kerrotulla Ifm:n sovelluksella. Testauksen alussa huomattiin, että skaalaukset eivät olleet tarpeeksi isot, joten tämä korjattiin heti testauksen alkuvaiheessa. Kuvassa 18 nähdään kytkentä testialustaan yhdistettynä.

Kuva 18. Näyttö yhdistettynä ohjausjärjestelmään.

Testialustaa käyttäen ohjelman toimivuus todettiin toimivaksi. Näytöllä arvot muuttuivat potentiometrejä säätämällä hienosti ja jokainen eri toiminto suoriutui näytölle. Tämä ei kuitenkaan tarkoittanut sitä, että ohjelma toimisi ajoneuvossa heti samanlaisesti, koska sisäänmenot piti muuttaa, jotta ohjelma pystyy lukemaan tarvittavat arvot. Myös ajoneuvosta saadut arvot eivät välttämättä käyttäydy samanlaisesti kuin potentiometrien vastusarvot.

8 JÄRJESTELMÄN TESTAUS AJONEUVOSSA

Projektin edetessä ajoneuvoa päästiin tarkastelemaan, jolloin asentajia neuvottiin johdotuksessa ja VDO:n asentamisessa. Isoksi ongelmaksi heti asennusvaiheessa ilmeni, että mistä kanavasta kaasupolkimen asentoon tarvittava jännitesignaali saataisiin. VDO:n datalehdissä oli kerrottu kaasupolkimen asennon minimi -ja maksimiarvot, mutta sille annettua lähtöä ei. Tällöin kävimme jokaisen sisääntulon kaasupolkimelle ja katsoimme missä kanavassa jännite suurenee ja pienenee. Oikean kanavan löydettyä yhdistettiin ohjausjärjestelmästä kaasupolkimen asennoksi määritetty sisääntulo vakionopeudensäätöön. Ohjelmassa kyseinen sisääntulo skaalattiin, jotta kaasupolkimen asento saadaan oikein näkyväksi näytöllä.

Seurvaaksi ongelmaksi ilmeni nopeuden ja kierosluvun signaali. Ohjausjärjestelmä ei pystynyt lukemaan pienen jännitevaihtelun (0-5V) takia kyseisiä arvoja. Ohjausjärjestelmä lukee parhaiten 0-12V jännitealuetta. Ratkaisuksi löydettiin aluksi optoerotinkytkentä, mutta pitkien toimitusaikojen takia ongelmaa ei pystytty ratkaisemaan kyseisellä kytkennällä. Tätä projektia aikaisemmissa projekteissa tämä kyseinen ongelma ratkaistiin lisäämällä vastus sisääntulon ja jännitelähteen väliin, joten samaa ratkaisua testattiin myös kyseisessä projektissa. Jotta oikea vastusarvo löydettiin, kytkentään sijoitettiin potentiometri, jonka vastusarvoa muutettiin, kunnes arvot näkyivät ohjauskeskuksen sisääntuloissa. Arvot tulivat näkyviin 1,5kΩ vastusarvolla, joten potentiometri korvattiin kyseisellä vastuksella.

Arvojen näkyessä ohjauskeskuksella ja näytöllä, päästiin ensimmäiselle koeajolle. Emme kuitenkaan ensimmäisellä koeajolla testanneet ohjelman toimivuutta, vaan tarkastelimme ohjelman muuttujien arvojen vaihtelua. Koeajon ansiosta korjasimme skaalausvirheitä ja mitattujen arvojen mittausvälejä. Esimerkiksi nopeus sisääntulo mittasi kymmenlukuina. Nopeus ja kierrosluku oli saatava näkyville siten, että arvo muuttuisi yhden kokonaisluvun verran eikä kymmenen kokonaisluvun välein.

Kun tarvittavat muutokset tehtiin näytön ja ohjausjärjestelmän ohjelmiin, käytiin uudella koeajolla, tällä kertaa testaamaan, toimiiko ohjelma halutulla tavalla. Koeajossa havaittiin, että näin ei tapahtunut. Ohjelmassa oli proportionaaliventtiilin toimilohkossa virheellisesti määritettyjä osioita, jota en havainnut kuin muutamien päivien jälkeen. Ennen kyseisen virheen havainnollistamista kävimme auton hybridipuolen kytkennän moneen kertaan läpi ja tarkistin ohjelman toimivuuden useaan kertaan läpi. Vian havainnollistamiseen kuitenkin kesti oma aikansa.

Proportionaaliventtiilin toimilohkovian havainnollistamisen jälkeen kävimme koeajolla testaamassa ohjelman toimivuutta, jolloin havaittiin ohjelman toimivuus toimivaksi. Tämän jälkeen testasimme huolella

jokaisen eri ominaisuuden toimivuuksia, tulisiko minkäänlaisia eri virhetiloja.

Teimme myös erilaisia mittauksia, kuinka hybridiominaisuus nopeuttaa tasaisella ajamisella sekä kiihdyttämisessä. Tasaisella ajamisella hybridiominaisuus oli n.100m matkalla melkein puolitoista sekuntia nopeampi. Tähän kuitenkin vaikutti, että kyseisellä koeajolla proportionaaliventtiilin kela ei ollut akun purkauksessa täysin auki. Proportionaaliventtiili oli tässä vaiheessa 68% auki. En kuitenkaan kyseisellä hetkellä halunnut turvallisuussyistä proportionaaliventtiiliä täysin auki ensimmäisessä koeajossa, jossa ohjelma toimi halutulla tavalla.

Kun proportionaaliventtiilin asetti täysin auki purkauksessa, huomasi todella helposti, että hybridiominaisuus auttoi kiihdyttäessä huomattavasti. Latausvaiheessa havaittiin myös, että paineen kasvaessa paineakulle, jarruttaminen kasvoi huomattavasti. Tästä syystä tehtiin ohjelmaan muutos, joka pienentää proportionaaliventtiilin kelavirtaa paineen kasvaessa. Tällöin paineakun ollessa täynnä jarruttaminen on luontevaa.

9 JATKOKEHITYS

Projektin kiireellisyyden vuoksi ohjelma todettiin toimivaksi ja käytettäväksi sellaisenaan. Kuitenkin ohjelmaa voisi kehittää, jolloin siitä tulisi paljon monipuolisempi kuin nykyinen käytössä oleva ohjelma. Esimerkiksi näytölle voitaisiin lisätä uusia valikkoja, jossa näkyisi kuinka useasti akkua on ladattu ja käytetty purkaukseen. Kuten hybridiautoissa, voisi myös kyseisessä projektissa olla erilaisia vaihtoehtoja purkauksen ”voimakkuudesta” nimikkeillä kuten eco, sport ja standardi.

Esimerkiksi Sport tilassa proportionaaliventtiili aukeaisi kokonaan heti kaasua painettaessa ja latausvaiheessa. Tällöin kiihdytys paranisi normaalista versiosta. Huonoksi puoleksi ilmenisi paineakun pieni tilavuus, jolloin kiihdytys vaiheessa akku tyhjenisi todella nopeasti, mutta ladattaessa se myös täyttyisi nopeammin, koska proportionaaliventtiili aukeaisi kokonaan. Käyttäjä voisi näytöllä valita itselleen sopivan toimintatilan.

Eco tila toimisi n.20% voimakkuudella, jolloin proportionaaliventtiili aukeasi huomattavasti vähemmän kuin sport tilassa, mutta paineakku ei tyhjenisi niin nopeasti. Kyseinen toimintatila olisi polttoaineen kulutuksen kannalta mainio ratkaisu, mutta kiihdyttämiseen siitä ei olisi niin isoa vaikutusta kuin sport tai normaalissa tilassa. Kuvassa 19 on esitetty minkälainen esimerkiksi toimintatilojen vaihto voisi mahdollisesti olla.

Kuva 19. Esimerkki erilaisista toimintatiloista.

Ohjelmaan olisi myös mahdollista saada kerättyä lataus- ja purkauskertoja, kuinka useasti auton käynnistyksen ja sammutuksen aikana akkua on ladattu ja purettu ja paljonko polttoainetta ollaan säästetty yhden matkan aikana ja kokonaisuudessaan. Myös ohjelman visuaalisuutta voitaisiin parantaa, jotta käyttäjälle selkeytyisi paremmin, milloin akkua ladataan ja milloin taas puretaan.

Viimeisimmillä koeajoilla ohjelma toimi mainiosti eikä pitkillä koeajomatkoilla havaittu minkäänlaisia virheitä. Koeajojen jälkeen kuvasimme hybridiominaisuutta, jota käytettiin saksan messuilla.

10 POHDINTA JA JOHTOPÄÄTÖKSET

Projektille asetetut tavoitteet toteutuivat. Hybridiominaisuus toimi halutulla tavalla ja oli valmiina ennen huhtikuun messuja. Näyttö ja ohjausjärjestelmä kommunikoivat keskenään, sekä näytöllä esitetyt parametrit antavat tarkkaa tulosta auton nopeudesta sekä kierrosluvusta, joka osoittautui projektissa hankalimmaksi osuudeksi.

Projekti oli kokonaisuudessaan todella opettavainen ja mielenkiintoinen. Harmillisesti projektin visuaalisia puolia ei projektin kiireellisyyden takia päästy parantamaan sekä kehittää ohjelman toimivuutta.

Kyseinen projekti ei välttämättä ole tuotteeksi sovelias, mutta se antaa käytännön kuvaa, että tarvittaessa asiakkaan tarpeiden mukaan voimme kehittää niitä heille. Mielestäni kyseinen projekti tukee tätä näkymää erittäin hyvin.

Aikaisempi kokemus Codesys 2.3V -ohjelmointiympäristöstä oli todella vähäinen. Tämä työ palveli erinomaisesti tarkoitustaan ja se opetti erittäin paljon codesys ohjelman ohjelmoinnista. Codesys-ohjelmointiin tutustuminen ja sen käyttäminen oli erittäin mielekästä.

Muiden valmistajien ohjelmistoista minulla oli kokemusta, mutta ei Codesyssistä. Kun ohjelmaan pääsi käsiksi ja sain neuvoa ohjelman toiminnasta, alkoi se vähitellen tuntumaan paljon monipuolisemmalta kuin aikaisemmat käytössäni olevat ohjelmistot. Codesys on mielestäni erittäin hyvä ohjelmointityökalu ja sen monipuolisuus ohjelmoinnissa vakuutti minut. Ohjelmassa on erittäin paljon erilaisia ominaisuuksia, jotka helpottavat ja myös nopeuttavat ohjelmointia erilaisissa projekteissa.

Projekti ei valmistunut alkuperäisten aikataulujen mukaan, mutta valmistui kuitenkin ennen saksan messuja, jossa tarkoituksena oli esitellä projektia. Työkoneiden älykkyyttä sekä ohjelmistojen osuutta lisätään, jotta tuotteiden kilpailukyky kasvaa.

LÄHTEET

Autoalarm. (2015). Vakionopeudensäätimet. Haettu 9.4.2019 osoitteesa https://www.autoalarm.fi/VDO-Pedal-Interface-2

Autoalarm. (2015).Vakionopeudensäätimet. Haettu 9.4.2019 osoitteesta https://www.autoalarm.fi/VDO-Pedal-Interface-2

Codesys. (2011). Haettu 9.4.2019 osoitteesta https://www.codesys.com/

Ifm. (2015a). CR0403. Haettu 15.03.2019 osoitteesta https://www.ifm.com/fi/fi/product/CR0403

Ifm. (2015). CR0451. Haettu 15.3.2019 osoitteesta https://www.ifm.com/fi/fi/product/cr0451

Parker. (2019). Proportionaaliventtiili. Haettu 9.4.2019 osoitteesta https://www.parker.com/Literature/Industrial-Systems-Division-Europe/Catalogues/Industrial%20Valves%20UK/03/D1FB%20UK.pdf

Suhonen, L. (2011). *IFM-Näytön sovittaminen CAN-väylällä dieselmoottoriin codesys-kehitysympäristön avulla.* Opinnäytetyö. Kone- ja tuotantotekniikka. Tampereen ammattikorkeakoulu. Haettu 1.3.2019 osoitteesta https://www.theseus.fi/handle/10024/33867

Toyota. (n.d.). Mikä on hybridi. Haettu 9.4.2019 osoitteesta https://www.toyota.fi/hybrid/mika-on-hybridi.json

Liite 1

LIITTEEN OTSIKKO