

Petri Hartikainen

Dr16-veturin diagnostiikkajärjestelmän kehittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinöörityö

25.5.2013

Tekijä(t) Otsikko	Petri Hartikainen Dr16-veturin diagnostiikkajärjestelmän kehittäminen
Sivumäärä Aika	21 sivua 25.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Prosessiautomaatio
Ohjaaja(t)	Tuotantopäällikkö Jarmo Falck Lehtori Timo Tuominen
<p>Insinööriyön tavoitteena oli kehittää korvaava ratkaisu Dr16-veturin ohjausjärjestelmän käyttöliittymälle ja samalla parantaa diagnostiikkatietojen hyödyntämistä. Nykyisessä järjestelmässä käyttäjä vastaanottaa tiedot järjestelmästä kahden erillisen näytön kautta. Valvontapäänteen kautta käyttäjä saa mittaus- ja hälytystietoja sekä käyttäjä voi antaa pakko-ohjauksia takaisinpäin ohjausjärjestelmälle. Mittaristoyksiköllä esitetään dieselmoottorin tärkeimmät arvot. Uuden käyttöliittymän olisi tarkoitus sisältää valvontapäänteen ja mittaristoyksikön informaatio ja pakko-ohjauksien painonapit sekä mahdollistaa diagnostiikkatietojen tallentaminen.</p> <p>Työ alkoi tutustumisella ja muutoksien suunnittelulla veturin Selma 2 -ohjausjärjestelmään ja näyttöjen sarjaliikenneväylien ominaisuuksiin. Muutoksien toteutuksen jälkeen aloitettiin ohjelman kehittäminen kosketuspaneelinäytölle Labview-ohjelmointikielellä. Ohjelma sisältää tarvittavan protokollan käsittelyn, mittaus- ja hälytystietojen hallinnan sekä käyttöliittymän.</p> <p>Työn lopputuloksena saatiin käytettävyydeltään parempi käyttöliittymä ja järjestelmä, jonka toimintoja voidaan laajentaa tulevaisuudessa diagnostiikkatietojen tallentamiseen.</p>	
Avainsanat	Labview, Selma 2, Dr16

Author(s) Title	Petri Hartikainen Dr16-locomotive's Development of the Diagnostic System
Number of Pages Date	21 pages 25 May 2013
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Process Automation
Instructor(s)	Jarmo Falck, Production Manager Timo Tuominen, Senior Lecturer
<p>The aim of this thesis was to develop a replacement solution for the user interface of control system of Dr16-locomotive and at the same time improve the diagnostic data recovery. In the current system, the user receives information from the system through two separate displays. The user receives the measurement and alarm data and the user can give back forced controls to the control system. The main values of the diesel engine are presented through the instrumentation unit. The new interface would be intended to include information of the control terminal and the instrumentation unit, push buttons of the forced control and allow the recording of the diagnostic data.</p> <p>The project started by learning and designing changes in the Selma 2 control system of the locomotive and the characteristics of the serial buses of the displays. After the implementation of changes, development of program on the touch panel display was started with Labview programming language. The program includes the required protocol processing, measurement and alarm data management and user interface.</p> <p>The result of project was a better user interface in terms of usability and the system whose functions can be expanded in the future to record diagnostic data.</p>	
Keywords	Labview, Selma 2, Dr16

Sisällys

1	Johdanto	1
2	Dr16	2
3	Dr16-ohjausjärjestelmä	3
3.1	Yleiskuvaus	3
3.2	Selma 2	3
3.3	RS-232	4
3.4	USART86-CONT-tietoliikenneohjain	5
3.5	USART86-INT-virtasilmukkakortti	6
3.6	Valvontapääte	7
3.7	Mittaristoyksikkö	8
4	Labview	9
5	Dr16-ohjausjärjestelmän muutokset	11
5.1	USART86-CONT-tietoliikenneohjain	11
5.2	Jarrusylinterien painemittarin siirto näyttöön	12
5.3	Selma 2 -ohjelmamuutokset	13
6	Uusi käyttöliittymä	14
6.1	Yleiskuvaus	14
6.2	Ohjelma	16
6.3	Virheenhallinta	18
7	Kehitysmahdollisuuksia	19
7.1	Selma 2 -ohjelma	19
7.2	Uusi käyttöliittymä	20
8	Yhteenveto	20
	Lähteet	22

1 Johdanto

VR Group on kokonaan Suomen valtion omistama monipuolinen, ympäristöystävällinen ja vastuullisesti toimiva matkustuksen, logistiikan ja infrarakentamisen palveluyritys. VR Groupin liiketoiminnot koostuvat matkustajaliikenteen VR:stä, logistiikan VR Transpoin-tista ja infrarakentamisen VR Trackista. Liiketoimintoja tukevat junaliikennöinti- ja kun-nossapitoyksiköt sekä Venäjä ja kansainväliset toiminnot -yksikkö.

Hyvinkään konepaja on yksi VR Groupin kunnossapitoyksiköistä. Sen kaksi päätehtä-vää ovat rautatiekaluston peruskorjaukset ja vaihto-osatuotanto. Peruskorjauksessa veturi tai vaunu puretaan osiin, runko kunnostetaan ja maalataan sekä irto-osat kun-nostetaan tai korvataan uusilla. Vaihto-osatuotanto sisältää kunnossapitovariikoilla vaihdettavien komponenttien huoltoa, korjausta, uusien valmistusta tai tarvittaessa nii-den lähettämistä alihankkijoille huollettavaksi.

Dr16 on VR Groupin rautatieliikenteessä käyttämä dieselsähköveturi. Dr16-veturin käyttövoima sähköiselle voimansiirrolle tuotetaan dieselmoottorin pyörittämällä sähkö-generaattorilla. Tämä mahdollistaa myös sähköenergian syöttämisen matkustajavau-nuihin, minkä takia sitä voidaan käyttää monipuolisesti tavara- ja matkustajaliikentees-sä. Tämän opinnäytetyön aihe syntyi Hyvinkään konepajan elektroniikkaosastolla tar-peesta korvata uudella ratkaisulla Dr16-dieselsähköveturin kuljettajan valvontapääte ja mittaristoyksikkö, jotka ovat ehtineet käyttöikänsä loppupäähän sekä kärsivät varaosien puutteesta.

Tässä työssä esitellään Dr16-veturin perustekniikka lyhyesti ja työhön liittyviltä osin nykykuntoinen Dr16-veturin Selma 2 -ohjausjärjestelmä, johon liittyvät myös korvatta-vat valvontapääte ja mittaristoyksikkö. Tämän työn tavoitteena on kehittää ratkaisu niiden korvaamiseen kosketusnäytöllisellä paneeli-PC:llä ja samalla kehittää systeemiä siten, että veturin tuottamia diagnostiikkatietoja pystyttäisiin käyttämään jälkikäteen paremmin hyödyksi. Työn lopuksi esitellään vielä tämän työn ulkopuolelle jääneitä kehi-tysehdotuksia sekä kerrotaan työssä kehitetyn ratkaisun testauksesta saadut johtopää-tökset.

2 Dr16

Dr16 on yleiskäyttöinen tavara- ja henkilöjunien veturi. Veturi on varustettu keskiohjaamolla, johon kuuluu kaksi samanlaisilla toiminnoilla ja laitteilla varustettua ajopöytää. Käyttövoima tuotetaan dieselmootorilla, jonka mekaaninen energia muutetaan generaattorin avulla sähköenergiaksi sähköiselle voimansiirrolle. Veturin jokaisella neljällä akselilla on oma oikosulkumoottori ja veturin kokonaisteho on 1360 kW. Suurin sallittu nopeus on 140 km/h, ja monikäyttöön vetureita voidaan kytkeä enintään kolme. Dr16-vetureiden kehitys aloitettiin vuonna 1985 neljän protoveturin valmistuksella. Vuosina 1990–1992 valmistettiin 19 sarjaveturia, joista on nykyään käytössä 18 veturia. Kaikki protoveturit on poistettu käytöstä ja yksi sarjaveturi on hylätty vaurioitumisen johdosta. [1]



Kuva 1. Kaksi Dr16-veturia moniajossa.

3 Dr16-ohjausjärjestelmä

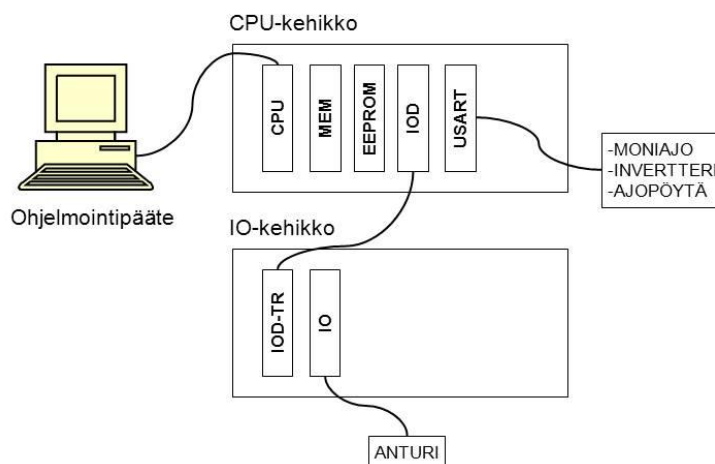
3.1 Yleiskuvaus

Dr16-ohjausjärjestelmän tehtävänä on inverttereille menevän momenttiohjeen muodostaminen, dieselin kierrosluvun säätäminen, paineilma- ja apukäyttölaitteiden ohjaus ja erilaiset veturin toimintojen valvonnat. [1]

Ohjausjärjestelmän päävastuu on ohjelmoitavalla Selma 2 -järjestelmällä. Selma 2 liittyy veturin muihin laitteisiin I/O-korttien ja sarjaliikennekanavien avulla. Mikroprosessoripohjainen ohjausjärjestelmä ja sarjaliikenteen hyödyntäminen laitteiden välisessä tiedonsiirrossa teki Dr16:sta uutena modernin veturin. Sarjaliikenteen kautta ollaan yhteydessä ajopöytien laitteisiin, inverttereihin ja moniajossa muihin vetureihin. [1; 2]

3.2 Selma 2

Selma 2 on Strömbergin 1980-luvulla kehittämä 16-bittiseen mikroprosessoritekniikkaan perustuva automaatiojärjestelmä. Sen toiminta perustuu 16-bittisiin mikroprosessorisiin 8086 ja apuna käytetään aritmetiikkaprosessoria 8087. Järjestelmän minimikokoonpano koostuu korttikehikosta, jännitelähteestä, prosessorikortista ja muistikortista. I/O-liityntöjen määrästä riippuen kokoonpanoa voidaan laajentaa yhden prosessorikortin ohjaamana neljään korttikehikkoon.



Kuva 2. Selman perusrakenne.

Järjestelmän tyypillisiä sovelluskohteita ovat esimerkiksi

- paperikoneiden, päällystyskoneiden ja valssainten tasavirta- tai vaihtovirtalinjakäyttöjen ohjaus
- laivojen kokonaisautomaatio sisältäen dieselvoimalan ja pääkoneiden ohjaukset, potkurisäädön ja koneistojen kunnonvalvonnan
- veturien diesel- ja ajomoottorien säätö sekä muut ohjaustoiminnot.

Selma 2:n merkittävin piirre oli aikanaan sen ohjelmointiperiaate. Ohjattavan prosessin tunteva henkilö pystyi ohjelmoimaan kohdejärjestelmän toiminnot lohkokaavio-ohjelmoinnilla, minkä takia perinteistä ohjelmointikoulutusta ei tarvittu. Tämän päivän automaatiojärjestelmien ohjelmointi tehdään hyvin samantyyppisellä lohkokaavio-ohjelmoinnilla. [2; 3]

3.3 RS-232

Electronic Industries Association (EIA) julkaisi RS-232-standardin vuonna 1969. Standardi kuvaa päätelaitteen (Data Terminal Equipment eli DTE) ja modeemin (Data Circuit Terminating Equipment eli DCE) välisen liitännän sarjadata siirtämiseksi. Se sisältää neljä osaa:

- sähköiset signaaliominaisuudet
- liitännän mekaaniset ominaisuudet
- liitäntäpiirien toimintakuvaukset
- standardiliitännät tavallisimmille tietoliikenne rakenteille.

RS-232 on alun perin yhdysvaltalainen standardi ja sitä vastaa Euroopassa CCITT:n suositus V.24. [6]

RS-232:n jännitelogiikka on käänteinen, positiivinen jännite vastaa loogista tilaa nollaa ja negatiivinen jännite ykköstä. Hyväksytyt jännitetasot ovat positiivisella puolella +5 ... +15 V ja negatiivisella puolella -5 ... -15 V. Väliin jäävä +5 ... -5 V alue on lähettävässä päässä määrittelemätön siirtymäalue, jonka logiikkatasoja ei ole määritetty. [6]

RS232:n vastaanottavan pään ainoana erona lähettyvään päähän on määrittelemätön siirtymäalue, joka on +3 ... -3 V. [6]

RS-232:n tyypillisiä nopeuksia ovat 2400, 4800, 9600 ja 19200 bps. Standardi määrittää korkeimmaksi nopeudeksi 20000 bps, mutta nykyaikaiset laitteet tukevat nopeuksia 115200 bps:ään asti. [6]

RS-232:n merkkikehys koostuu

- aloitusbitistä
- databiteistä (5, 6, 7 tai 8 bittiä)
- pariteettibitistä (parillinen, pariton tai pois käytöstä)
- lopetusbitistä (1, 1,5 tai 2 bittiä).

Pariteettibitti on virheentarkistus, joka asetetaan databittien ykkösten perusteella. Jos käytössä on parillinen pariteetti, se asetetaan siten, että ykkösiä on parillinen määrä ja parittomalla päinvastoin. [6]

3.4 USART86-CONT-tietoliikenneohjain

USART86-CONT-tietoliikenneohjain on Selman CPU-väylään asennettava älykäs kortti, jolla on neljä V.24-tyyppistä sarjaliikennekanavaa. Kortti kommunikoi prosessorikortin kanssa ja hoitaa itsenäisesti sarjaliikennekanavien tiedonsiirron. Sarjaliikennekanavat ovat optoerotettu kortin logiikasta. Kullekin kanavalle voidaan asettaa oma siirtonopeus (50–19200 bps). [4]

Standardiohjelmalla varustettu ohjainkortti ohjaa ja puskuroi tietoliikennettä Selman prosessorikortin ja sarjaliikennekanavien välillä. Ohjaimen ohjelmisto sisältää kahdeksan erilaista protokollaa eri käyttötarkoituksia varten. Jokaiselle kanavalle voidaan valita käyttötarkoitukseen sopiva protokolla Selman ohjelman käynnistysvaiheessa. Dr16:ssa käytetään Sami-protokollaa kaikilla käytössä olevilla kanavilla lukuun ottamatta valvontapäätte- ja mittaristoyksikkökanavia. [4]

Sami-protokollan sanoma sisältää osoitteen ja datan, jotka esitetään heksalukuja vastaavilla ascii-merkeillä (0 ... F). Esimerkiksi kokonaislukuosoite 255 on viestissä heksa-

lukuna FF. Data on 16-bittinen kahden komplementtiluku, joka on kokonaisluku. Tämän takia desimaaliluvut pitää kertoa Selman ohjelmassa kokonaisluvuksi. Sanomat ovat muotoa stx, a1, a0, d3, d2, d1, d0, cr, bcc.

Sanoman merkkien tarkoitus ja sisältö on

- stx = aloitusmerkki (02H)
- a0,a1 = osoite (00 ... FFH)
- d0, d1, d2, d3 = data (0000 ... FFFFH)
- cr = lopetusmerkki (0DH)
- bcc = tarkistussumma, $-(a0+a1+d0+d1+d2+d3)$.

Merkkien kehysasetuksina käytetään seitsemää databittiä, parillista pariteettia ja yhtä stop-bittiä. [4]

Valvontapäätteiden ja mittaristoyksiköiden tietoliikenneohjain on varustettu yksilöllisellä niitä palvelevalla ohjelmistolla. Selman ohjelmassa käytetään näiden kanavien kanssa kommunikoinnissa Sami-protokollan toimilohkoja. Standarditietoliikenneohjaimesta poiketen tässä ohjaimessa tietoja muokataan valvontapäätteelle ja mittaristoyksikölle sopivaan muotoon. [4]

3.5 USART86-INT-virtasilmukkakortti

20 mA virtasilmukkaa käytettiin siirtoyhteisissä 1960-luvulta 1980-luvun alkupuolelle. Sillä mahdollistettiin kustannustehokkaasti pitemmät häiriösietoiset siirtoyhteydet. EIA422 (joulukuu 1978) ja EIA485 (huhtikuu 1983) differentiaalitiedonsiirtostandardien myötä 20 mA virtasilmukan suosio laski nopeasti. Nykyisin digitaaliset siirtoyhteydet perustuvatkin useimmiten differentiaaliseen tiedonsiirtoon. [7]

Haluttaessa käyttää virtasilmukkaa USART86-CONT-tietoliikenneohjaimen yhteydessä voidaan V.24-liitännät sovittaa USART86-INT-virtasilmukkakortilla 20 mA:n virtasilmukaväyliksi. Kortilla on sovitukset neljälle kanavalle, jotka on erotettu galvaanisesti toisistaan. Dr16:ssa kaikissa sarjaliikenneyhteisissä käytetään virtasilmukkaa parantamaan toimintavarmuutta. [4]

3.6 Valvontapääte

Valvontapääteen tehtävä näyttää ohjausjärjestelmän mittausarvoja ja hälytyksiä sekä laajentaa kuljettajan mahdollisuuksia vaikuttamalla laitteiden toimintaan pakko-ohjausten avulla. Valvontapääte sisältää kuusirivisen tekstinäytön, 11 painonappia näytön toimintojen ohjaamiseen ja hälytyksien merkkiäänisummerin. Valvontapääteessä ei itsessään ole juurikaan älyä, vaan näytön toimintaa ohjataan tietoliikenneohjaimella. Tietoliikenneohjain muodostaa hälytys- ja mittaus tietojen sekä painonappien painallusten perusteella näytön tekstit, jotka lähetetään sarjaliikennekanavassa valvontapääteen näytölle. [5]



Kuva 3. Valvontapääteen kuva.

Hälytykset muodostetaan Selman ohjelmassa järjestelmästä kerätystä informaatiosta. Yleisemmin hälytystiedot muodostuvat mittausarvojen raja-arvoista tai muiden laitteiden tilatiedoista. Saapuneesta hälytyksestä saadaan merkkiääni summerilla. Merkkiääni poistuu, kun hälytys ilmoitetaan havaituksi "kuittaus"-painonapilla. Näyttöä voidaan käyttää muihin toimintoihin vasta hälytyksen kuittauksen jälkeen. [5]

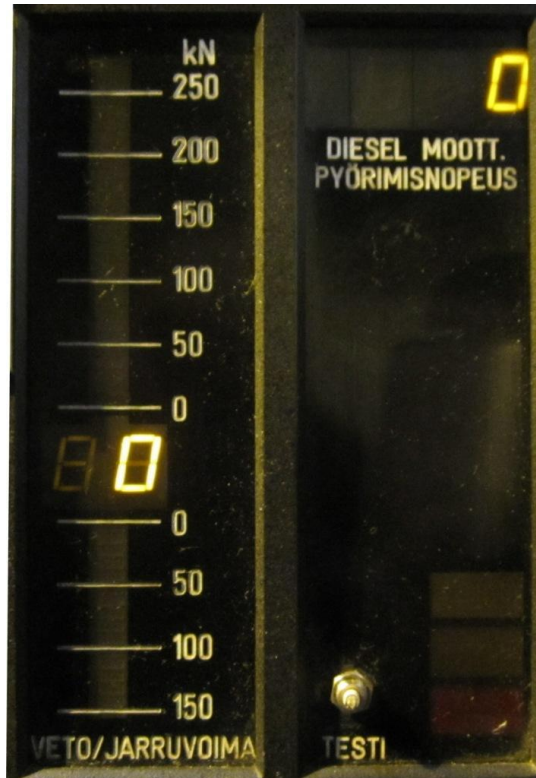
Näppäimistöllä voidaan valita viidestä mittausryhmästä yksi mittausryhmä kerrallaan esitettäväksi näytölle. Mittaustieto näytöllä koostuu kohteen nimestä ja mittausarvosta. Yhden mittausryhmän sisältämien arvojen määrä riippuu pääasiassa nimien pituuksista. Suurin osa ryhmistä sisältää kaksitoista mittausarvoa. [5]

Pakko-ohjauksilla voidaan ohjata joitakin suureita haluttuun tilaan. Tarvittaessa voidaan esimerkiksi poistaa lämpötila-anturien hälytysrajat anturin vikaantuessa. Pakko-ohjaukset on jaettu neljän ryhmän valikkoon, joita selataan näppäimistön avulla. Haluttu pakko-ohjaus valitaan ”pakko-ohjaus 1<>0”-painonapilla. [5]

Moniajossa apuvetureiden tiedot saadaan pakko-ohjauksilla näytölle. Eri vetureiden hälytys- ja mittaustietoja ei yksilöidä valvontapäätteellä, joten kuljettajan täytyy olla tietoinen, minkä veturin tiedot ovat esillä näytöllä. [5]

3.7 Mittaristoyksikkö

Mittaristoyksikön tehtävänä on esittää Selmalta tulevia tärkeitä mittaus- ja tilatietoja.



Kuva 4. Mittaristoyksikön kuva.

Analogimittareilla näytetään

- veto- ja sähköjarruvoima
- dieselin pyörimisnopeus
- vetovoimaportaiden määrä.

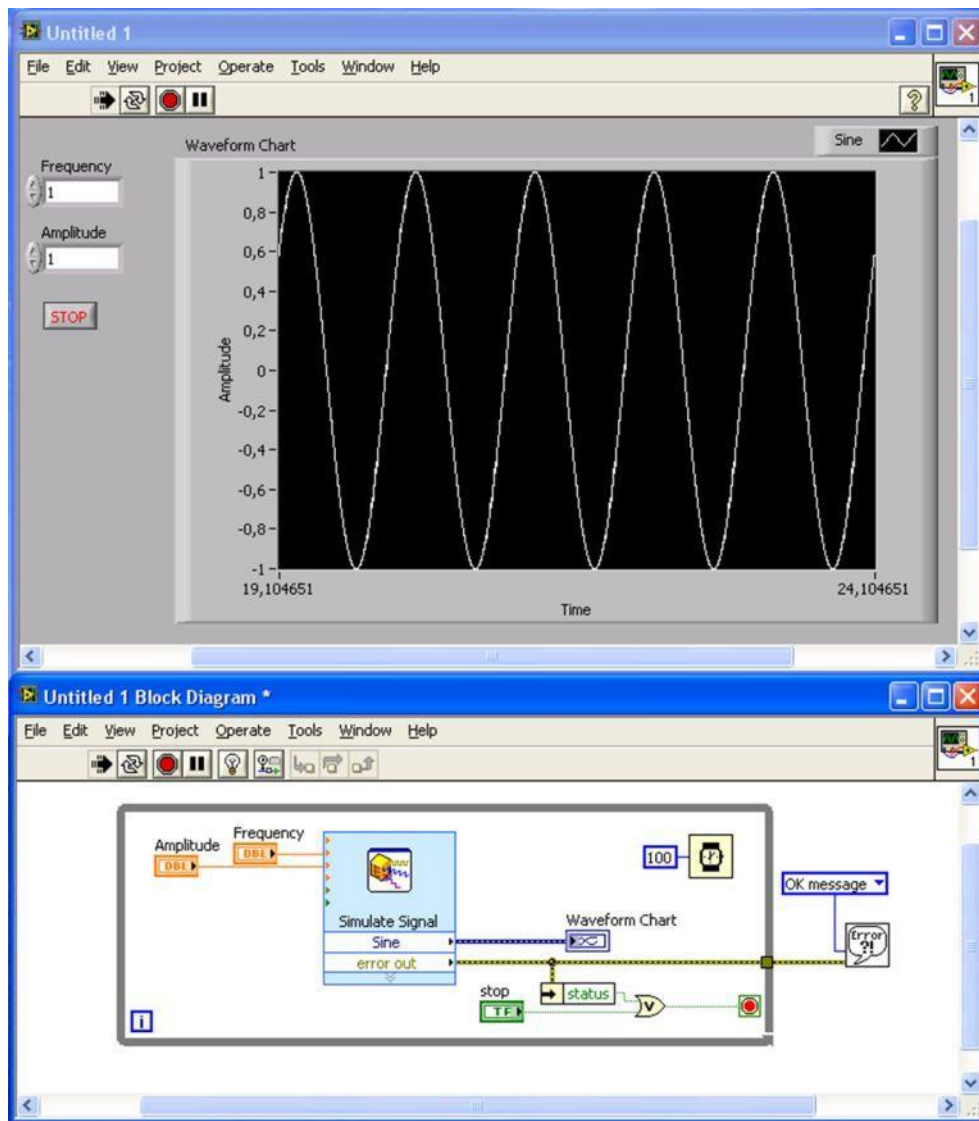
Led-merkkivaloilla näytetään tilatiedot

- käsijarrusta
- luistosta
- kytkijältä.

Testi-painonapilla voidaan testata mittaristoyksikön toimintakunto ja led-segmenttien toimivuus. [5]

4 Labview

Labview (Laboratory Virtual Instrument Engineering Workbench) on National Instrumentsin graafinen ohjelmointikieli. Ohjelmien tekoon käytetään tekstirivien sijaan funktioita kuvaavia ikoneita, jotka yhdistetään toisiinsa datajohdoilla. Ohjelman teko muistuttaaakin paljon lohko-ohjelmointia. Labview-sovelluskehitysympäristöä käytetään pääasiassa erilaisten mittaus- ja testausratkaisujen kehittämiseen. Labview pitää sisällään tekstiohjelmoinnista tutut ohjelmointirakenteet, joten se soveltuu myös muunlaisten sovellusohjelmien tekoon. Labview-ympäristöön liittyy vahvasti National Instrumentsin erilaiset fyysiset mittalaitteet ja sulautetut kehitysympäristöt, jotka yhdessä mahdollistavat nopean ja helpon kehitystyön ihmisille, joilla ei ole vahvaa osaamista ohjelmoinnista ja mittalaitteiden kehittämisestä. Myös monien muiden mittalaittevalmistajien tuotteille on saatavilla valmiita Labview-funktioita, ja tarvittaessa funktioita voi tehdä mittalaitteiden mukana saatavista DLL-kirjastoista. [8]



Kuva 5. Labview-ohjelman esimerkki.

Labview-ohjelmia kutsutaan virtuaali-instrumenteiksi (VI). VI sisältää kolme pääkomponenttia – etupaneeli-ikkuna, lohkokaavioikkuna ja ikoni/liitântätaulu (kuva 5). Etupaneeli on VI:n käyttöliittymä ja se sisältää ohjaukset ja tilanäytöt, jotka ovat VI:n interaktiivisia tuloja ja lähtöjä. Varsinainen ohjelmointi tehdään lohkokaavioikkunassa. Etupaneelin ohjaukset ja tilanäytöt näkyvät lohkokaavioikkunassa terminaaleina, jotka voidaan yhdistää datajohdoilla haluttuun kohtaan koodissa. Ikoniliitântätaulu sisältää VI:n ikonin muokkaukseen tarkoitetun editorin ja taulun, jolla voidaan muokata VI:n ulkopuolelle näkyviä liitântöjä. VI:tä, jota käytetään toisen VI:n sisällä, kutsutaan nimellä aliVI (sub-VI). AliVI:n etupaneelin ohjaukset ja tilanäytöt toimivat haluttaessa tuloina ja lähtöinä ulospäin, jotka määritellään liitântätaulussa. Ikonieditorilla voidaan tehdä VI:lle yksilöllinen ikoni, jolloin erilaiset aliVI:t on helpompi tunnistaa muun koodin joukosta. [8]

Ohjelman teossa on hankala ennakoida erilaisia virhetilanteita, joita voi tulla käyttäjän toimista tai satunnaisista häiriötilanteista esimerkiksi mittalaitteissa. Tästä syystä virheidenhallinta on oleellinen osa ohjelmointia. Labview-ohjelmissa virheenkäsittelyn voi hoitaa joko automaattisesti tai manuaalisesti. VI:t ja funktiot voivat palauttaa virheen kahdella tavalla – tyypillisesti funktiot käyttävät virheen numerokoodia ja VI:t käyttävät virheklusteria. Virheklusteri sisältää tilatiedon, virhekoodin ja virheen tapahtumispaikan.

Oletuksena Labview käsittelee ohjelman suorituksen aikana tulevat virheet automaattisesti. Virhetilanteessa ohjelman suoritus pysäytetään ja esiin hyppäävällä valintaikkunalla näytetään virheen numerokoodi ja tapahtumapaikka. Ohjelman pysähtyminen kesken suorituksen voi aiheuttaa ongelmia joissakin sovelluksissa, joten tällaisissa tapauksissa manuaalinen virheidenhallinta on parempi tapa. Samalla se on myös työläämpi, koska virheidenkäsittelyn toimenpiteet täytyy lisätä ohjelmakoodiin. Manuaalinen virheidenhallinta vaatii, että VI:t ketjutetaan yhteen virhetulojen ja -lähtöjen kautta datajohdoilla. Tällöin virhe ei keskeytä ohjelman suoritusta, vaan ohjelmakoodiin tehdyt virheenkäsittelytoiminnot hoitavat virhetilanteet ja esimerkiksi ohjelman toiminnot lopetetaan hallitusti ja sen jälkeen näytetään virhe. Satunnaishäiriöitä varten voidaan tehdä toiminto, joka poistaa yksittäisen virheen ja reagoi vasta toistuvaan virheeseen. Kuvassa 5 esitetyssä ohjelmassa on esimerkki yksinkertaisesta virheenkäsittelystä, joka pysäyttää ohjelman hallitusti ja esittää virheen tiedot valintaikkunalla. [8]

5 Dr16-ohjausjärjestelmän muutokset

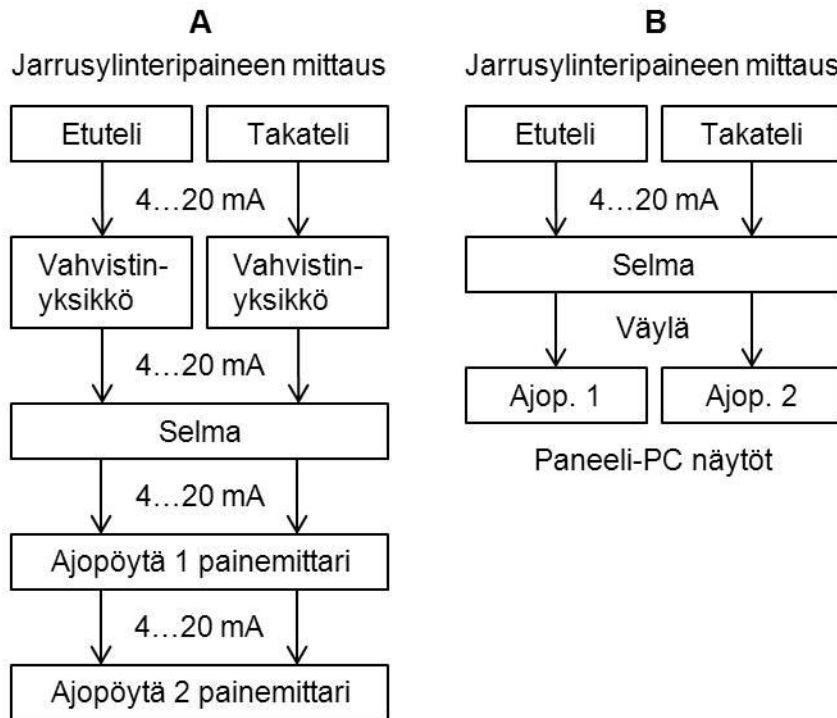
5.1 USART86-CONT-tietoliikenneohjain

Tietoliikenneohjaimen muutoksien kannalta keskeiset kysymykset liittyivät valvontapäätettä ja mittaristoyksikköä varten ohjelmoitujen ominaisuuksien aiheuttamiin joustavuusongelmiin sekä sarjaliikennekanavien määrään.

Selman ohjelman ja uuden näytön välinen kommunikointi haluttiin hoitaa suoraan ilman tietojen muokkausta tietoliikenneohjaimella, joten testaukset aloitettiin korvaamalla nykyinen kortti perusohjelmalla varustetulla kortilla. Protokollaksi valittiin Sami-protokolla, koska sitä käytetään kaikissa muissa veturin sarjaliikenneyhteyksissä. Muutoksien myötä kaikki tieto lähetetään yhteen sarjaliikennekanavaan nopeudella 19200 bps.

5.2 Jarrusylinterien painemittarin siirto näyttöön

Työn aikana testattiin mahdollisuutta korvata vanhat mekaaniset telien jarrupainemittarit uuteen käyttöliittymään sijoitetulla mittarilla. Kuvassa 6 on esitetty lohkokaaviotasolla jarrupaineen mittaus ja esitys nykyisellä tavalla kohdassa A. Kohdassa B on esitetty uuden käyttöliittymän mahdollistama mittaus ja esitys.



Kuva 6. Lohkokaaviokuva jarrusylinteripaineen mittauksesta ja esityksestä.

Nykyisessä mittauksessa painemittareiden näyttämät säädetään kohdalleen vahvistinyksikön säädöillä. Vahvistinyksikön virtasignaali kulkee sarjaan kytkettyjen Selman analogimittauksen ja painemittareiden lävitse, joten säätö vaikuttaa myös aina Selman mittausarvoon. Mittarin siirtoa uuteen näyttöön testattiin, koska mittarinäyttämän korjaus on työlästä ja siitä voi seurata yli 0.3 bar:n vääristymä Selman mittaustietoon, jolloin Selma antaa turhia Jarrut kiinni-hälytyksiä.

Kohdassa B anturi on liitetty suoraan Selman analogiatuloon. Paineantureiden näyttämä skaalataan Selmassa parametreilla oikeaksi ja mittausarvot siirretään näytön mittarille väylän avulla muiden mittaustietojen mukana. Tällöin Selman mittausarvo ja mittarin näyttämä ovat aina samat eikä painearvojen vääristymien kanssa tule ongelmia.

Testien aikana mittarin päivitysnopeus oli 350 ms, jolloin viisareissa oli havaittavissa pientä laahaamista. Sopiva päivitysnopeus voisi olla noin 200–250 ms, jolloin saavutettaisiin riittävä päivitysnopeus viisareiden tasaiseen liikkumiseen.

5.3 Selma 2 -ohjelmamuutokset

Tarvittavat Selman ohjelmamuutokset on testattu uuden näytön testien yhteydessä, mutta niiden teko rajattiin tämän työn ulkopuolelle, joten muutoksien peruserämuutoksista esitetään vain sanalliset esimerkit. Ohjelmistomuutoksien perussy on se, että sarjaliikennekortilla ei enää käsitellä mittaus- ja hälytystietoja ja kummankin ajopöydän näyttö toimii yksilönä, jolla ei ole tietoa esimerkiksi toisen näytön kautta annetusta pakko-ohjauksesta. Myös mahdollisen tiedonkeruun tekeminen vaatii ohjelmamuutoksia, joilla tunnistetaan Sami-protokollan sanoman osoitteen perusteella minkä veturin tiedosta on kyse.

Vanhassa järjestelmässä moniajossa apuvetureiden hälytykset lähetettiin samoihin osoitteisiin kuin johtoveturinkin. Uudella näytöllä apuvetureiden vikoja luetaan tietyin aikaväleihin, jos apuvetureissa on hälytyksiä päällä. Sarjaliikenteen lähetyslohkon ominaisuutena on käytetty tapaa, jossa arvo lähetetään muutoksesta tai joka 50. ohjelma-kierron. Tämä voi aiheuttaa ongelmia, jos useammassa veturissa on samanaikaisesti samat hälytykset päällä. Tästä syystä apuvetureiden hälytyksille on varattava omat osoitteet.

Hälytyksien tapaan myös eri vetureiden mittaustiedot on aikaisemmin lähetetty samaan osoitteeseen. Tiedonkeruuta varten jokaisen veturin mittaustiedoille täytyy antaa omat osoitteet. Tällä mahdollistetaan se, että oman veturin tietoja voidaan lähettää näytölle jatkuvasti. Aikaisemmin oman veturin tietojen syöttö on katkennut haluttaessa nähdä apuveturin tietoja.

6 Uusi käyttöliittymä

6.1 Yleiskuvaus

Uuden näytön valintaprosessi käynnistyi haluttujen ominaisuuksien selvittämisellä. Vaihtoehtoina oli korvata nykyiset näytöt vastaavatyypisillä näytöillä, jolloin muutostyöt muun järjestelmän osalta olisivat jääneet vähäisemmäksi. Tämänkaltaiset näytöt olisi joutunut teettämään tilaustyönä, joten niistä olisi tullut Dr16:n pienen sarjakoon takia kalliita ja samalla uudemman tekniikan mahdollisuudet olisivat jääneet hyödyntämättä.

Nykyään rautatiekalustossa käytetään käyttöjärjestelmän päälle tehtyjä ajopöytänäyttöjä, joten käyttöjärjestelmällä varustettu paneeli-PC oli myös tälle työlle sovelias lähtökohta. Nykyisessä järjestelmässä pakko-ohjauspainonappien tiedot kulkevat valvontapäänteen kautta, joten uuden näytön kohdalla päädyttiin kosketusnäyttöön, jolloin nykyiset painonapit voidaan tehdä käyttöliittymän ohjelmaan. Eräs ongelma nykyisessä järjestelmässä on pakko-ohjausvalikoiden selaus, koska jokaiselle pakko-ohjaukselle ei ole omaa painonappia. Kosketusnäyttö mahdollistaa kuljettajan kannalta helppokäyttöisemmän ratkaisun, koska jokaiselle pakko-ohjaukselle voidaan tehdä oma painonappi. Myös painonappien toimintojen muuttaminen ja niiden määrän muutos onnistuu tällä ratkaisulla pelkällä ohjelmamuutoksella.

Kosketusnäytön malliksi valittiin IEI UPC-V312-D525. Tämä kyseinen näyttö valittiin vain käyttöliittymän testausta varten, joten näytön valintaan vaikuttavia rautatiemääräyksiä ja tilavaatimuksia ei ole huomioitu tässä työssä.



Kuva 7. Kosketusnäyttö ja testiohjelma koeajolla.

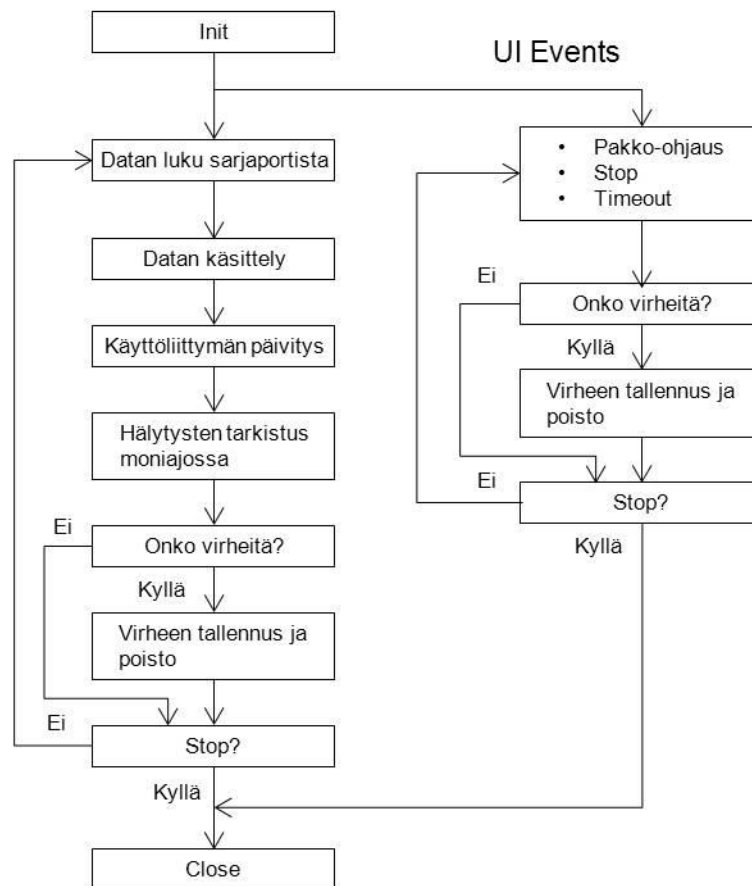
Käyttöliittymän ohjelman ohjelmointikieleksi valittiin Labview, josta työnantajan kautta saatiin käyttöön 2009 SP1 -versio. Käyttöliittymän kehittäminen olisi varmasti helppo tehdä esimerkiksi Wonderware InTouch -valvomo-ohjelmistolla, joka on tarkoitettu käyttöliittymien tekoon, mutta tällöin sarjaliikenteelle olisi pitänyt tehdä erillinen ajuri jollain muulla ohjelmointikielellä. Labview'lla voitiin toteuttaa samassa ohjelmassa käyttöliittymän teko ja sarjaliikenteen protokollan käsittely. Labview 2009 SP1 -versiolla tehtyjä sovelluksia voi käyttää Windowsin XP-, Vista- ja 7-käyttöjärjestelmissä. Tarvittaessa nykyinen ohjelma voidaan kääntää Windows 8 -käyttöjärjestelmälle uudemmalla Labview-versiolla.

6.2 Ohjelma

Ohjelman pää rakenne koostuu kahdesta rinnakkain suoritettavasta while-silmukasta. Ensimmäinen while-silmukka sisältää event-rakenteen, joka vastaanottaa käyttöliittymästä tulevia tapahtumia. Tällaisia tapahtumia ovat esimerkiksi painonapin painallukset. Event-rakenteen avulla painonapeille voidaan tehdä erilaisia toiminnallisuuksia. Sillä voidaan toteuttaa esimerkiksi sellainen toiminto, jossa painonappia täytyy painaa tietyn aikaa, ennen kuin painonapin haluttu toiminto tehdään. Event-rakenteella saadaan tapahtuma painonapin painalluksesta ja ylösnoususta. Jos tapahtumien välinen aika on ollut riittävän pitkä, voidaan toteuttaa haluttu toiminto. Event-rakenteella tällaisen toiminnon tekeminen vaatii vähemmän koneen prosessori-aikaa verrattuna tyyliin, jossa painalluksia seurattaisiin jatkuvasti pyörivässä while-silmukassa.

Toisessa while-silmukassa on varsinainen ohjelmakoodi, joka pyörii jatkuvasti silmukassa. Silmukassa käsitellään seuraavat asiat:

- Luetaan merkit sarjaportista.
- Käsitellään merkit Sami-protokollan mukaisesti.
- Sami-sanoman osoitteen perusteella arvo käsitellään hälytys- tai mittausarvona.
- Mittausarvo päivitetään käyttöliittymään, jos sille on osoitettu paikka ohjelman alustuksessa.
- Hälytysarvosta tarkistetaan, onko tilassa tapahtunut muutosta ja sen perusteella muokataan käyttöliittymän hälytyslistaa.
- Jos on apuvetureita ja niissä hälytyksiä päällä, tarkistetaan hälytyksien tila asetetun aikavälin mukaisesti.

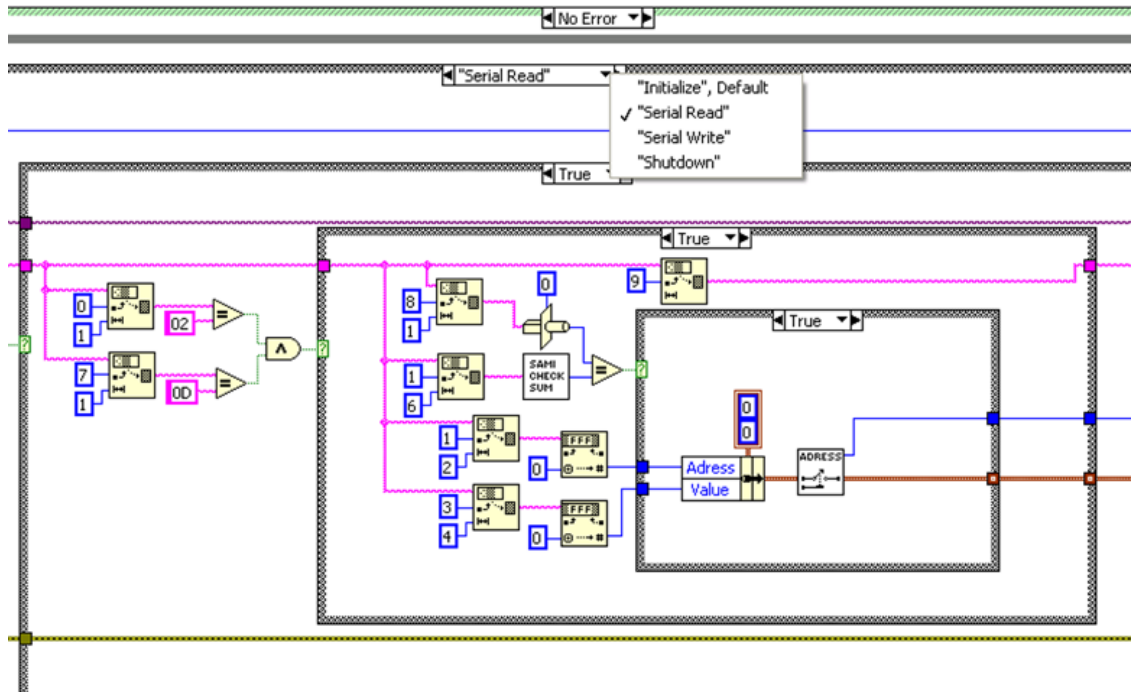


Kuva 8. Ohjelman lohkokaaavioesitys.

Käyttöliittymän mittausrvoja esittäviä komponentteja käytetään ohjelmakoodista *referencen* ja *property noden* avulla. Labview:ssa jokaisella käyttöliittymän komponentilla on yksilöllinen referenssinumero. *Property node* tarkoittaa komponentin yksilöllisiä funktioita, joilla voidaan esimerkiksi muuttaa komponentin ulkonäköä tai sen esittämää arvoa. *Referencen* avulla voidaan ohjelmassa yksilöidä, minkä käyttöliittymäkomponentin *property nodea* halutaan käyttää.

Tässä ohjelmassa käyttöliittymäkomponenttien referenssit järjestellään ohjelman alustuksessa taulukkoon mittausrvon osoitteen ja sitä esittävän käyttöliittymäkomponentin perusteella. Tämän taulukon avulla ohjelma ohjaa arvon oikeaan paikkaan käyttöliittymässä. Tällä tavalla voidaan välttää jäykkä yhteys ohjelmakoodin ja käyttöliittymän välillä ja mahdollistetaan saman ohjelmakoodin käytön erilaisten käyttöliittymien kanssa. Ohjelman alussa vain etsitään valitusta käyttöliittymästä halutut komponenttireferenssit taulukkoon, mutta ohjelmakoodi pysyy aina samana.

AliVI:den teossa on käytetty etäisesti olio-ohjelmoinnin luokkarakenteesta tuttua tyyliä, jossa VI sisältää joukon yhteen kuuluvia määrittelyjä. VI:n sisällä funktioita käsitellään case-rakenteella. VI:n julkisia muuttujia voi käsitellä funktioiden välityksellä. Kuvassa 9 on esitelty sarjaporttia käyttävän VI:n rakenne.



Kuva 9. Funktiorakenteen esimerkki.

Tällä tavalla ohjelman rakenteesta saatiin selkeä, VI:n sisältämiä muuttujia käsitellään vain funktioiden kautta sekä vältetään globaalien muuttujien käyttö. Koska ohjelmassa on kaksi rinnakkaista while-silmukkaa, on vaarana, että muuttujia käsitellään silmukoissa samanaikaisesti. VI:tä voidaan käyttää vain yhdessä silmukassa kerrallaan, joten muuttujan käsittely funktioilla estää samanaikaisen käytön rinnakkaisissa silmukoissa.

6.3 Virheenhallinta

Valvontapäätte ja mittaristoyksikkö eivät ole veturin kulun kannalta kriittisiä laitteita, joten korvaavan näytön toimintavakaudelle ja virhesietoisuudelle ei asetettu erityisiä vaatimuksia. Ohjelmaan tehtiin manuaalinen virrehallinta, joka tallentaa virheen tapahtumisajan ja tiedot tekstitiedostoon sekä poistaa virheen. Jos virhe on toistuva, käyttäjän täytyy uudelleen käynnistää ohjelma tai PC. Ohjelman teon ja testauksen aikana virheet liittyivät aina sarjaporttiin ja näissä tapauksissa virheet olivat satunnaisia, jolloin

niiden poistaminen oli riittävä toimenpide. Ohjelman kaatuminen esimerkiksi moniajossa, jolloin kuljettaja ei ole paikalla, ei vaadi uudelleenkäynnistystä välittömästi. Uudelleenkäynnistys tehdään vasta siinä vaiheessa, kun kuljettaja ottaa ajopöydän käyttöön. Näistä syistä tämän työn puitteissa ei lähdetty tekemään monipuolisempia toimintoja virheenkäsitteilyyn.

7 Kehitysmahdollisuuksia

7.1 Selma 2 -ohjelma

Selma 2 -ohjelman kehittämiselle suurimmat ongelmat asettaa veturien yhteensopiavuuden säilyttäminen moniajossa. Tämän takia ohjelmiston muokkauksessa on otettava huomioon asiat, jotka vaikuttavat moniajossa toisiin vetureihin. Myös ohjelmamuistin koko on rajallinen näin vanhassa järjestelmässä, mikä aiheuttaa omat rajoitteensa ohjelmiston kehittämiselle.

Tällä hetkellä veturin tiedonkeruu on toteutettu Selmalla, jota varten Selman muistista on varattu muistia ohjelmakoodille ja tiedonkeruun arvoille. Tiedonkeruun siirtämisellä pois Selmasta uudelleen näyttöpäätteelle on mahdollista vapauttaa tilaa muutoksille ja kokonaan uusille toiminnoille.

Nykyisin Selma lähettää koko mittaus-tietovektorin kerralla sekunnin välein, joka varaa 19200 bps:n nopeudella sarjaliikenneväylän n. 350 ms ajaksi. Tämän aikaikkunan sisällä väylälle ei voi lähettää muuta tietoa, joten esimerkiksi testatun jarrupainemittarin minimipäivitysnopeus on 350 ms. Mittaus-tietovektori sisältää muutosnopeudeltaan hyvin erilaisia arvoja. Taajuusmuuttajien välipiirin jännite voi vaihdella nopeasti, kun taas ulkolämpötilan vaihtelu on hidasta. Työn aikana mietityistä mittausarvojen lähetystavoista tehokkain vaihtoehto olisi tapa, jossa arvot lähetettäisiin ohjelmasta väylälle yksitellen ja vain arvon muutoksesta. Tällöin saataisiin tasattua väylän kuormitusta ja karsittua ylimääräisen tiedon lähetys minimiin. Ongelmaksi tämän ratkaisun kohdalla voi tulla Selman ohjelmamuisti, koska jokainen mittausarvo vaatisi oman sarjaliikennelohkon. Toinen ratkaisu voisi olla arvojen ryhmittely päivitysnopeuden mukaan pienempiin ryhmiin. Tässä vaihtoehdossa ohjelmamuisti ei tulisi ongelmaksi, koska ohjelma pysyy samankaltaisena, vain mittausarvoryhmien määrä kasvaisi.

7.2 Uusi käyttöliittymä

Ohjelman tekovaiheessa otettiin huomioon mahdollisuus lisätä hälytys- ja mittaustietojen tiedonkeruu. Labview mahdollistaa datan tallentamisen kahdella eri tavalla. Yksinkertaisimmillaan datan tallentaminen voidaan hoitaa tallentamalla dataa tekstitiedostoon, johon Labview'n perusversioissa on valmiit funktiot. Lisähankintana on myös saatavilla Database Connectivity Toolkit, jolla dataa voidaan tallentaa SQL-tietokantaan.

Jos ohjelmaan lisätään datan tiedonkeruu, tällöin tulee ajankohtaiseksi myös virnehallinnan parantaminen. Moniajossa näytön täytyy pysyä toimintatilassa ilman käyttäjän valvontaa, joten tämänhetkinen virnehallinta on riittämätön tällaisessa tilanteessa. Virnehallinta voisi koostua silloin kolmiportaisesta systeemistä:

- Virhe tallennetaan muistiin ja poistetaan.
- Jos sama virhe uusiutuu useamman kerran tietyn ajan kuluessa, vain viimeinen virhe tallennetaan tekstitiedostoon, poistetaan ja ohjelma alustetaan uudestaan.
- Jos sama virhe toistuu vielä tämänkin jälkeen, PC käynnistetään uudestaan.

Näillä toimenpiteillä pystyttäisiin parantamaan toimintavarmuutta ja estämään katkokset tiedonkeruun toiminnassa. Ajopöydän ollessa kuljettajan hallussa PC:n uudelleenkäynnistyksestä täytyisi informoida kuljettajaa, jotta estetään mahdolliset väärinkäsitykset näytön toiminnan kannalta.

8 Yhteenveto

Työn lähtökohtana oli kehittää ratkaisu Dr16-veturin diagnostiikkatietojen seuraamiseen kosketusnäytöllisellä paneeli-PC:llä. Työn ulkopuolelle rajattiin diagnostiikkatietojen varsinainen tallentaminen, mutta se on otettu huomioon työn aikaisissa ratkaisuisa. Työn lopullinen sisältö selvisi vasta Selma 2 -ohjausjärjestelmään perehtymisen jälkeen. Etenkin diagnostiikkatietojen keruu osoittautui mahdottomaksi nykyisessä järjestelmässä, joten työhön sisällytettiin myös Selman muutokset. Selman ohjelmamuutokset jätettiin tämän työn ulkopuolelle, ja tässä työssä ohjelmamuutoksista esitettiin vain sanalliset esimerkit.

Paneeli-PC:lle tehtiin ohjelma Labview-ohjelmointikielellä ja se sisältää käyttöliittymän, sarjaportin hallinnan sekä mittaus- ja hälytystietojen käsittelyn. Ohjelmakoodissa otettiin huomioon mahdollinen tiedonkeruun lisääminen jälkikäteen. Työhön käytetystä ajasta pääosa käytettiin ohjelman tekoon.

Työssä tehtyjen ratkaisujen ja näytön ohjelman testaaminen osoittautui hankalaksi, koska Dr16-veturit käyvät Etelä-Suomessa lähinnä poikkeustapauksissa. Työtä varten tehtiinkin Selman ohjelman perusteella simulaatio-ohjelma, joka vastaa Selman toimintaa yksittäisenä veturina sekä moniajoveturina. Ratkaisuja päästiin testaamaan työn aikana useampaan kertaan yksittäisellä veturilla, jolloin voitiin varmistua perusratkaisuista. Moniajota varten kehitettyjen ratkaisujen testaaminen jäi yhteen kertaan, mutta sillä saatiin varmistus simulaatio-ohjelman oikeasta toiminnasta ja tätä kautta myös kehitetyn ohjelman oikeanlaisesta toiminnasta.

Tämän työn tuloksena syntyneillä ratkaisuilla voidaan kehittää veturin diagnostiikkatietojen esittämistä ja keräämistä, jolloin niitä voidaan käyttää tulevaisuudessa kunnossapidon ja vianetsinnän tukemiseen ja tehostamiseen.

Lähteet

- 1 Dr16. Kuljettajan ohjekirja. 2004. VR Oy.
- 2 Selma 2 -automaatiojärjestelmä. 2005. Koulutusmateriaali. ABB Oy.
- 3 Selma. Yleisselosteet. 1990. ABB Strömberg.
- 4 Selma. Korttiselosteet 1. 1990. ABB Strömberg.
- 5 Selma. Korttiselosteet 2. 1990. ABB Strömberg.
- 6 Campbell, Joe. 1985. RS-232 liitäntä. Mänttä: Mäntän kirjapaino.
- 7 Current Loop Application Note. Verkkodokumentti. <http://www.pacontrol.com/download/Current%20Loop%20Tutorial.pdf>. Luettu 20.5.2013.
- 8 Labview Basics 1: Introduction Course Manual. 2008. National Instruments.