

Markus Niiranen

HARVESTERIPÄÄN SAHAN KETJUNOPEUDEN SÄÄTÖ



Insinööri (AMK)

Tietotekniikka

Kevät 2016



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tekijä: Niiranen Markus

Työn nimi: Harvesteripään sahan ketjunopeuden säätö

Tutkintonimike: Insinööri (AMK), tietotekniikka

Asiasanat: käyttöliittymä, PLC, kosketusnäyttö, harvesteri, CX-One, CAN-väylä, J1939

Tämän opinnäytetyön tilasi Nivalassa sijaitseva, metsäkoneita valmistava ProfiPro Oy, ja työn välittäjänä toimi Kajaanin Ammattikorkeakoulu. Yrityksellä oli tarve helpottaa huoltohenkilön suorittamaa työtä, harvesteripään sahan ketjunopeuden säätämiseksi metsäkoneessa. Työn tavoite on tuoda säätötoiminnot koneen kosketusnäytölle, sekä toimia ohjeistuksena vastaavia uudistuksia laadittaessa.

Pääpaino työssä oli käyttöliittymän suunnittelussa ja toteuttamisessa, ohjelmoitavan logiikan ohjelman ymmärtämisessä, sekä niiden ohjelmointiin tarvittavien työkalujen käyttämisessä. ProfiPro Oy:n valmistamissa metsäkoneissa käytetään Omronin valmistamaa ohjelmoitavaa logiikkaa ja kosketusnäyttöä, ja niitä ohjelmoidaan CX-One -ohjelmistolla. Lisäksi työssä tutustutaan metsäkoneen tiedonsiirtoon CAN -väylän ja siinä käytettävän raskaan kaluston käyttämän SAE J1939 -standardin osalta, joka oli keskeisessä roolissa tuloksia simuloitaessa.

Työn tuloksena syntyi metsäkoneen käyttöliittymään uusi näyttösivu, jonka avulla koneen huoltohenkilö pystyy toteuttamaan tarvittavat säädöt harvesteripään sahanopeuden säätämiseksi, ilman erillistä tietokonetta ja ohjelmistoa. Näyttösivulle pääsy on rajattu vain huoltohenkilöiden käyttöön. Lisäksi syntyi myös opastava dokumentti, jonka avulla vastaavia muutoksia metsäkoneen käyttöliittymään saadaan aikaan vaivattomasti CX-One -ohjelmistoa käyttäen, ja muutoksia voidaan simuloida testiympäristössä. Työn tulokset tulevat tuotantoon ja niillä on tarkoitus säästää huolto- ja suunnitteluhenkilöstön resursseja.

ABSTRACT

Author: Niiranen Markus

Title of the Publication: Adjustment of saw chain speed in harvester head

Degree Title: Bachelor of Engineering, Information Technology Engineering

Keywords: CX-One, CAN-bus, harvester, J1939, PLC, user interface, touch screen

This thesis was commissioned by a Finnish forestry machine manufacturer ProfiPro Oy which is located in Nivala, and it was passed on to me by Kajaani University of Applied Sciences. The company had the need to facilitate the adjustment of the chain speed of the saw in the harvester head which is a task performed by the maintenance personnel. The goal was to add the needed controlling functions to the touch screen user interface of the machine. The objective was also for the thesis to act as a guide when designing similar reforms.

The main focus of the work was on designing the user interface and its implementation. It was also required to understand the programmable logic and the use of the necessary programming tools for PLC and touch screen. Forestry machines manufactured by ProfiPro Oy use programmable logic and a touch screen manufactured by Omron. These can be programmed with CX-One software. In addition, the work introduced the CAN bus and SAE J1939 -standard, used by heavy machinery for the data transmission in harvesters, and therefore in this thesis for simulating the results.

The work resulted a new user interface screen in the existing system that allows the maintenance personnel of the machine to be able to use the necessary controlling functions to adjust the saw speed of the harvester head without a separate computer and software. The use of the new screen is limited to service personnel. In addition, the work brought about a guiding document which allows similar changes to be made more fluently to the forestry machine's user interface, by using CX-One software. This helps to simulate the changes in the test environment. These results will be updated in the interfaces of forestry machines, and will thus save maintenance resources, as well as those of engineering personnel.

LYHENNELUETTELO

ACK	Acknowledgement
CAN	Control Area Network
CRC Field	Cyclic Redundancy Check Field
DLC	Data Length Code
DM	Data Memory
EMC	Electro Magnetic Compatibility
ID	Identifier
IDE	Identifier Extension
I/O	Input/Output
kBit/s	kilobittiä sekunnissa
kb/s	kilobittiä sekunnissa
LD	Ladder Logic
Mb/s	Megabittiä sekunnissa
PDU	Protocol Data Unit
PGN	Parameter Group Number
PLC	Programmable Logic Controller
RTR	Remote Transmission Request
SAE	Society of Automotive Engineers
SPN	Suspect Parameter Number
SRR	Substitute Remote Request

UINT	Unsigned Integer
USB	Universal Serial Bus

Sisällysluettelo

1 JOHDANTO	1
1.1 Työn taustoja	1
1.2 Sahanopeuden määrittäminen	3
2 TIEDONSIIRTO METSÄKONEESSA	4
2.1 CAN-väylä	4
2.1.1 CAN-väylän rakenne	4
2.1.2 CAN-viestin rakenne	6
2.2 SAE J1939 -standardi	9
3 OHJELMOITAVA LOGIIKKA, NÄYTTÖ JA NIIDEN OHJELMOINTI	12
3.1 Omron CJ2M-logiikka	12
3.2 Omron NS12-kosketusnäyttö	13
3.3 Omron CX-One -ohjelmisto	14
4 TYÖN VALMISTELU	16
5 TYÖN SUORITUS	20
5.1 Ohjelmoitavan logiikan ohjelma	20
5.2 Kosketusnäytön käyttöliittymän toteuttaminen	21
6 TESTAUS	33
7 YHTEENVETO	39
LÄHTEET	40

1 JOHDANTO

1.1 Työn taustoja

Aihe tähän insinööriyöhön tuli ProfiPro Oy:ltä, ja välittäjänä toimi Kajaanin Ammattikorkeakoulu. Työn päätavoitteena oli modernisoida olemassa oleva metsäkoneen hallintajärjestelmä harvesteripään sahan ketjunopeuden osalta, helpottamaan koneen huoltohenkilön työtä. Lisäksi työn on tarkoitus toimia ohjeena vastaavien hallintajärjestelmän muutosten toteuttamisessa.

ProfiPro Oy on Nivalassa sijaitseva, vuonna 2009 perustettu, pohjois-pohjanmaalainen metsäkoneita valmistava yritys, joka on jatkanut Profi-merkkisten metsäkoneiden valmistusta ja markkinointia [1]. Kuvassa 1 esitellään ProfiPro Oy:n valmistama Profi 54 -metsäkone.



Kuva 1. Profi 54.

ProfiPro Oy:n metsäkoneissa voidaan käyttää eri valmistajien harvesteripäätä, esimerkiksi Kesla tai Keto, joiden sahan ketjunopeus ja parametrit vaihtelevat. Lisäksi samassa harvesteripäässä voidaan käyttää erilaisia sahoja, ja sahan hydraulikkamoottorin tilavuus tai laipan pituus vaikuttavat myös käytettävään ketjunopeuteen. Tähän mennessä huoltohenkilön on sahanopeutta kalibroitaessa pitänyt etsiä koneen hallintaohjelmasta tietyt muistipaikat, joihin tallentuneita parametrien arvoja muuttamalla sahan ketjunopeus muuttuu. Tällainen etsiminen on aikaa vievää ja sitoo huoltohenkilön resursseja turhaan. Työssä pyritään helpottamaan kouran vaihtamisen tai huollon seurauksena tarvittavaa sahanopeuden kalibrointia tuomalla parametrien muuttaminen koneen kosketusnäytön käyttöliittymään.

Työssä käsitellään metsäkoneessa käytettävää tiedonsiirtotekniikkaa sekä koneen logiikan ja käyttöliittymän ohjelmointia.

1.2 Sahanopeuden määrittäminen

Harvesterikourassa käytetään hydraulilla pyöritettävää sahaa. Sahan ketjunopeutta säädetään säätämällä hydraulikapumpulta tulevaa painetta. Paine riippuu siitä, pyöriikö kone työkierroksilla vai tyhjäkäynnillä, sekä kukin metsäkoneenkuljettaja voi käyttää eri työkierroksia työskennellessään.

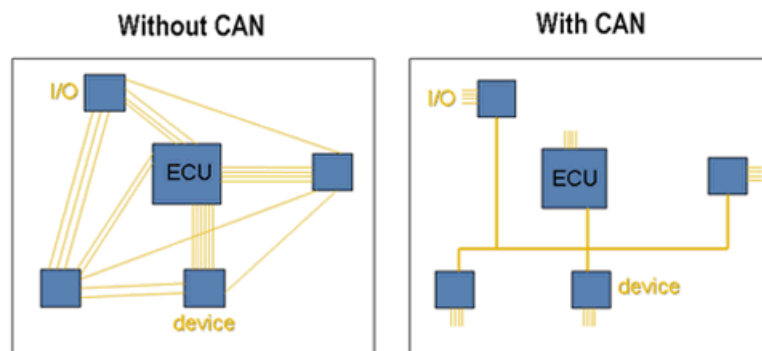
Sahan ketjunopeuden säätämisen tarkoitus on kompensoida paineen kasvamista työkierroksilla ja saada sahan ketjunopeus työkierroksilla lähelle tyhjäkäyntierroksien nopeutta. Itse sahan pyörimisnopeutta mitataan kalibroitaessa takometrillä suoraan sahalta. Sahan ketjunopeuteen vaikuttavat parametrit muodostavat kuvaajan, joka esittää ketjunopeuden muuttumista. Toinen säätöarvoista osoittaa kuvaajan alkupisteen ja toinen on kuvaajan kulmakerroin.

2 TIEDONSIIRTO METSÄKONEESSA

ProfiPron valmistamissa metsäkoneissa käytetään raskaalle kalustolle kehitetyn SAE J1939 -standardin mukaista protokollaa CAN-väylässä tapahtuvalle tiedonsiirrolle. CAN-väylä yhdistää koneissa muun muassa moottorin, koneen ohjauksen, työlaitteiden ja myös hytin säätöjen ohjausyksiköt toisiinsa.

2.1 CAN-väylä

Ajoneuvoissa johdotusten lisääntyminen johti jo 1980-luvulla sarjaväylätekniikan kehittämiseen. Vuonna 1986 Robert Bosch GmbH julkisti CAN-protokollan ensimmäisen version. CAN-väylä on sittemmin laajennut ajoneuvojen ja työkoneiden lisäksi tehdas- ja rakennusautomaatiokäyttöön [2]. CAN-väylää hyödyntämällä ajoneuvon johdotuksia saadaan huomattavasti vähennettyä, kuten kuvassa 2 on havaittavissa.

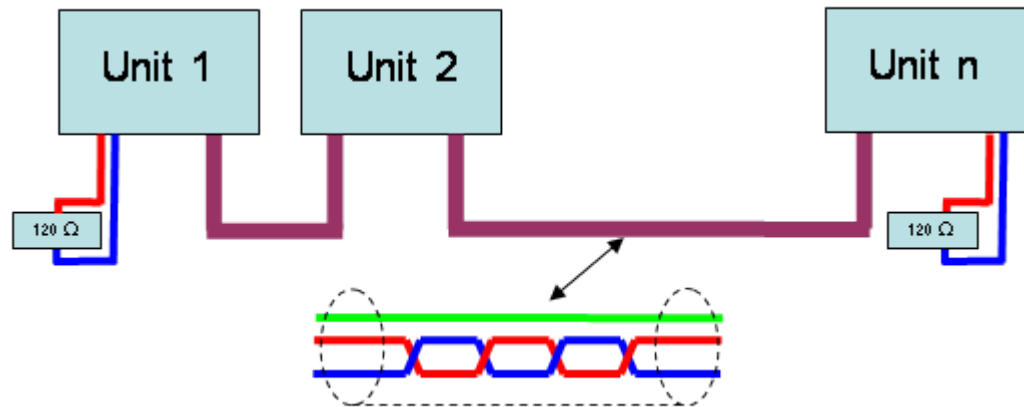


Kuva 2. CAN-väylä vähentää merkittävästi johdotuksen tarvetta ajoneuvoissa [3].

2.1.1 CAN-väylän rakenne

CAN-väylä on sarjaväyläjärjestelmä, joka yhdistää älykkäistä yksiköistä (solmut) muodostuvan hajautetun järjestelmän. Kuvassa 3 on nähtävissä CAN-järjestelmän yleinen rakenne. Järjestelmässä jokainen solmu voi lähettää viestejä väylään ja vastaanottaa väylästä tulevia viestejä. Se, saako solmu

lähettää viestejä väylään, riippuu viestin prioriteetista, ja se, vastaanotetaanko viesti väylästä, määräytyy viestin tunnisteesta. Prioriteetista ja tunnisteesta on kerrottu tarkemmin viestikehyksien yhteydessä.



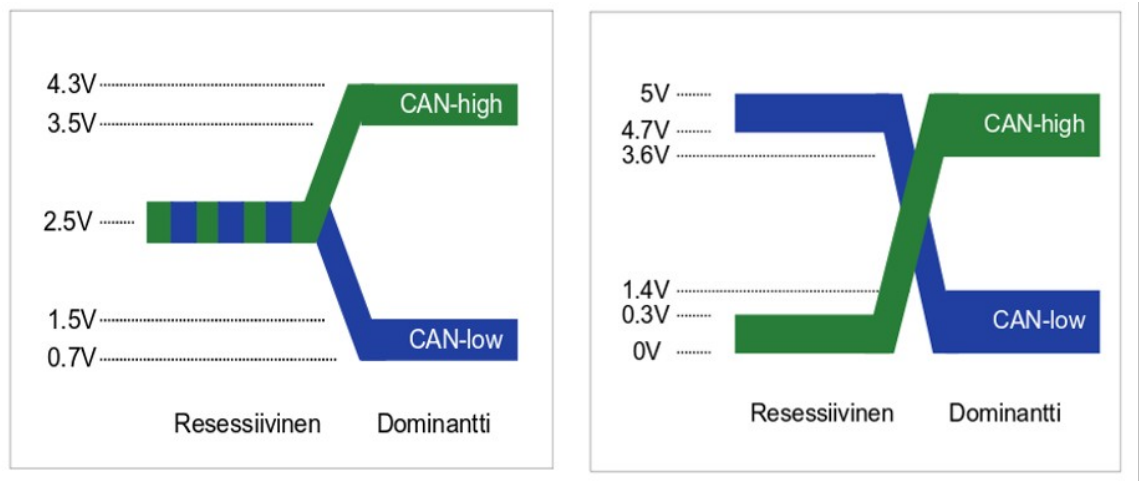
Kuva 3. CAN-järjestelmän rakenne [5].

Eri solmut yhdistyvät toisiinsa yleensä kierretyllä parikaapelilla. Näin saadaan parannettua väylän luotettavuutta ja minimoitua häiriöt.

CAN-väylästä on kolme eri versiota: High Speed CAN, Low Speed CAN (Fault tolerant CAN) sekä Single-wire CAN. High Speed CAN ja Low Speed CAN eroavat toisistaan nimensä mukaisesti tiedonsiirtonopeudessa, mutta myös vikasietoisuudeltaan. Johtimet on nimetty "CAN high" ja "CAN low", ja ne voivat olla joko resessiivisessä eli väistyvässä tilassa (looginen 1) tai dominantissa eli määräävässä tilassa (looginen 0). High Speed CAN-väylä on resessiivisessä tilassa silloin, kun molemmat, CAN high ja CAN low, ovat noin 2,5 voltia, ja dominantissa tilassa, kun CAN high on 3,5–4,3 voltia ja CAN low 0,7–1,5 voltia, jolloin differentiaalijännite on vähintään kaksi voltia. Tämä mahdollistaa nopeamman tiedonsiirron, maksimissaan noin 1 Mb/s. Low Speed CAN-väylä on resessiivisessä tilassa silloin, kun CAN low on 4,7–5,0 voltia ja CAN high on 0,0–0,3 voltia, jolloin differentiaalijännite on vähintään 4,4 voltia. Dominantissa tilassa Low Speed CAN-väylä on, kun CAN low on 0,0–1,4 voltia ja CAN high on 3,6–5,0 voltia, jolloin differentiaalijännite on vähintään 2,2 voltia. Low Speed CAN-väylän tiedonsiirtonopeus on maksimissaan noin 125 kb/s. Jännitetasot on esitetty kuvassa 4.

High Speed CAN-väylän johtimet yhdistyvät päätevastuksin (120 ohmia) molemmista päistään, ja niiden välinen resistanssi tulisi olla noin 60 ohmia. Näin vältetään signaalin heijastumista. Häiriöitä voidaan vähentää suorittamalla päättäminen kahdella 60 ohmin vastuksella ja lisäämällä häiriönpoistokondensaattori päätevastuksien välistä maihin [3].

Low Speed CAN-väylä on kuitenkin vikasietoisempi, koska sitä voidaan käyttää, vaikka toinen johtimista katkeaisi. Vikasietoisuuden ansiosta sitä käytetäänkin kohteissa, joissa ei vaadita suurta tiedonsiirtonopeutta mutta johtimet joutuvat mekaaniseen rasitukseen, esimerkiksi taipumaan. Single-wire CAN koostuu vain yhdestä johtimesta, ja sen viestit tulkitaan jännitetason mukaan. Single-wire CAN-väylän maksiminopeus on 41,6 kb/s.



Kuva 4. High Speed CAN- ja Low Speed CAN-jännitetasot [3].

2.1.2 CAN-viestin rakenne

Väylällä liikkuvat viestit sisältävät viestikehyksen. Viestikehystä on kaksi eri muotoa, CAN 2.0A ja CAN 2.0B, jotka eroavat toisistaan ID-bittien määrällä. CAN 2.0A käyttää 11 ID-bittiä ja laajennettu CAN 2.0B käyttää 29 (11+18) ID-bittiä. Viestikehyksiä on neljä erilaista:

- Datakehys (Data Frame)
- RTR-kehys (Remote Frame)
- Virhekehys (Error Frame)
- Viivekehys (Overload Frame)

Datakehys on yleisin lähetettävistä kehyksistä. Sitä käyttämällä voidaan välittää tietoja eri solmujen välillä. Esimerkiksi anturi voi lähettää tietoja toimilaitteelle.

RTR-kehystä käytetään, kun solmu haluaa toisen solmun lähettävän tietyn viestin, esimerkiksi toimilaitte pyytää anturilta tiedon. RTR-kehyksessä ei lähetetä datakenttää, vaan pyydetty tieto määräytyy kehyksen tunnisteiden mukaan. Kehys tunnustetaan RTR-bitin resessiivisestä tilasta.

Virhekehystä käytetään, kun solmu havaitsee virheen viestissä. Se lähettää siitä tiedon muille solmuille käyttäen virhekehystä. Tämän seurauksena virheellisen viestin lähettäjä yrittää lähettää viestin uudelleen. Virhekehys rikkoo CAN-viestin kehysääntöjä lähettämällä kuusi dominanttia bittiä ja lisäksi kahdeksan resessiivistä erotusbittiä.

Viivekehys tarkoitus on pitää väylää varattuna viestin käsittelyn ajan. Sitä ei enää juurikaan käytetä, koska viestien käsittelyyn käytettävät prosessorit ovat kehittyneet riittävän nopeiksi. Eri kehyksien rakenteet ovat kuvassa 5 esitetyn kaltaisia.



Kuva 5. CAN2.0A- ja CAN2.0B-viestikehyksien sisältämät kentät. [4]

CAN-väylän kehys koostuu seuraavasti:

- Aloitusbitti: Kertoo uuden viestin alkamisen ja synkronoi asemat.
- Haltuunottokenttä: Sisältää viestin tunnusteen (CAN 2.0A 11-bittinen ja CAN 2.0B 29-bittinen), joka kertoo asemalle, kuuluuko viesti ottaa vastaan vai ei. Lisäksi tunniste kertoo viestin prioriteetin. Pienimmällä ID:llä on korkein prioriteetti, eli kilpavarauksilanteessa korkeimmalla prioriteetilla varustettu viesti säilyy muuttumattomana. Lisäksi haltuunottokentässä lähetetään RTR-bitti, joka kertoo, onko kyseessä datakehys (bitti dominantti) vai RTR-kehys (bitti resessiivinen). RTR-bitti lähetetään tunnistekentän jälkeen CAN 2.0A -muotoisissa viesteissä ja CAN 2.0B -muotoisissa viesteissä tunnistekentän laajennuksen jälkeen.
- Ohjauskenttä: Kentän IDE-bitti kertoo, onko kyseessä normaali vai laajennettu kehys. Lisäksi ohjauskenttä sisältää tulevaisuuden laajennoksia varten varatun bitin. Varattu bitti pitää olla aina dominantti. DLC-bitit kertovat viestin koon neljällä bitillä.
- Datakenttä: Kenttä sisältää lähetettävän datan, jonka pituus on määritetty ohjauskentässä. Data lähetetään väylään eniten merkitsevä bitti ensimmäisenä.
- Tarkistuskenttä: Sisältää tarkistussumman, jonka avulla valvotaan viestin virheettömyyttä. Tarkistuskentän lopussa on erotusbitti, joka lähetetään resessiivisenä.
- Kuittauskenttä: Kentässä asemat kuittaavat eheän viestin vastaanoton lähettämällä kuittauskentässä dominantin tilan. Kuittauskentän lopussa on erotusbitti.
- Lopetuskenttä: Kentässä laite kertoo viestin loppumisesta seitsemällä resessiivisellä bitillä.
- Lopetuskentän jälkeen lähetetään vielä kolme bittiä erottamaan kehykset toisistaan

CAN-väylässä käytetään virheiden havaitsemiseen bit stuffing -menetelmää, jossa aina viiden samanarvoisen perättäisen bitin esiintymisen jälkeen lähettäjä lisää niiden perään vastakkaisen bitin. Vastaanottaja huomioi tämän ja poistaa ylimääräisen bitin viestin saadessaan.

2.2 SAE J1939 -standardi

J1939 on joukko autoalan standardisointijärjestön luomia standardeja, joita käytetään CAN-väylää hyödyntävissä raskaissa ajoneuvoissa, kuten kuorma-autoissa, busseissa ja metsäkoneissa. Niissä määritellään muun muassa fyysinen, data-, verkko- ja sovelluskerros, diagnostiikkaliittymän tyyppi ja parametriryhmät.

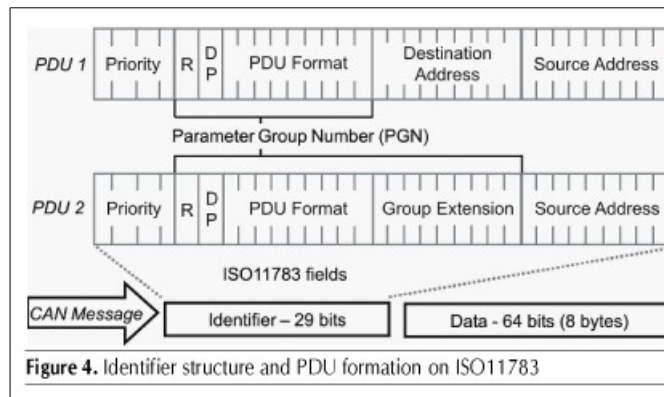
Standardissa on määritetty käytettäväksi fyysisessä kerroksessa suojattua ja kierrettyä parikaapelia, jonka maksimipituus on 40 metriä. Tiedonsiirtonopeus saa olla maksimissaan 250 kb/s, ja väylään saa liittää enintään 30 solmua.

Suurin osa standardissa määritetyistä viesteistä on tarkoitettu lähetettäväksi väylään ilman vastaanottajan osoitetta. Tämän ansiosta väylän jokainen solmu voi käyttää viestin sisältämää tietoa hyväkseen ja se lisäksi helpottaa uusien solmujen väylään liittämistä. J1939-standardin mukaiseen viestin tunnisteeseen voidaan kumminkin sisällyttää vastaanottavan solmun osoite, jos halutaan lähettää tietoa tietylle solmulle [6].

Datakerroksesta standardissa on määritetty käytettäväksi CAN 2.0B -muotoa, eli viestin tunniste on 29-bittinen. Tunnisteen kolme ensimmäistä bittiä kertovat viestin prioriteetin. Neljäs bitti on varattu tulevaisuuden tarpeita varten, ja sen arvo tulee asettaa nolllaksi. Viidentenä lähetetään Data Page -bitti, jolla voidaan laajentaa lähetettävien parametriryhmien määrää. Sen jälkeen tulee PDU format -tavu, jota seuraa PDU specific -tavu. Lopuksi lähetetään Source Address -tavu.

J1939-standardin mukaiset viestit on järjestetty ryhmiä (PDU), jotka koostuvat tunnisteesta ja 8 datatavusta. Ryhmiä on kaksi eri muotoa ja muoto ilmoitetaan

PDU format -tavulla, joka on esitetty kuvassa 6. Jos PDU format -tavu on alle 240, niin käytössä on PDU1 ja sitä käytetään, kun halutaan lähettää viesti tiettyyn osoitteeseen eli tietylle solmulle. Osoite kerrotaan PDU specific -tavussa. Jos PDU format -tavu on yli 239, niin käytössä on PDU2 ja sitä käytetään, kun halutaan kertoa lisää parametriryhmiä PDU specific -tavussa.



Kuva 6. Protokollataryhmät. [7]

Parametriryhmät (PGN) koostuvat varatusta ja Data Page -bitistä sekä PDU format- ja PDU specific -tavusta, jotka yhdistetään 18-bittiseksi arvoksi. Parametriryhmiä on määritetty standardissa kattavasti, ja ne kuvaavat millaista tietoa viestissä välitetään, esimerkiksi, että onko kyseessä moottorin kierroslukutieto tai tieto väännöstä. Esimerkki parametriryhmästä numero 61442 (0x00F002) nähtävillä kuvassa 7. [8]

PGN 61442		Electronic Transmission Controller 1		ETC1
Transmission Repetition Rate:	10 ms			
Data Length:	8			
Extended Data Page:	0			
Data Page:	0			
PDU Format:	240			
PDU Specific:	2	PGN Supporting Information:		
Default Priority:	3			
Parameter Group Number:	61442 (0x00F002)			
Start Position	Length	Parameter Name	SPN	
1.1	2 bits	Transmission Driveline Engaged	560	
1.3	2 bits	Transmission Torque Converter Lockup Engaged	573	
1.5	2 bits	Transmission Shift In Process	574	
1.7	2 bits	Transmission Torque Converter Lockup Transition in Process	4816	
2-3	2 bytes	Transmission Output Shaft Speed	191	
4	1 byte	Percent Clutch Slip	522	
5.1	2 bits	Engine Momentary Overspeed Enable	606	
5.3	2 bits	Progressive Shift Disable	607	
5.5	2 bits	Momentary Engine Maximum Power Enable	5015	
6-7	2 bytes	Transmission Input Shaft Speed	161	
8	1 byte	Source Address of Controlling Device for Transmission Control	1482	

Kuva 7. J1939-standardin mukainen PGN [9].

Kuvasta selviää, että viesti toistetaan 10 millisekunnin välein ja siinä on 8 tavua tietoa. Se kertoo myös, minkä tavun milläkin bitillä kukin tieto ilmoitetaan. Esimerkiksi tässä tapauksessa viidennen tavun ensimmäisellä ja toisella bitillä kerrotaan, onko moottorin hetkellinen ylinopeus sallittu. SPN on numero, jolla voidaan selvittää tarkemmat tiedot parametrasta.

3 OHJELMOITAVA LOGIIKKA, KOSKETUSNÄYTTÖ JA NIIDEN OHJELMOINTI

ProfiPro Oy:n valmistamissa metsäkoneissa niiden toimintaa ohjataan pääosin ohjelmoitavan logiikan avulla. Ohjelmoitavaa logiikkaa sekä muita toimintoja käytetään kosketusnäytössä toimivalla käyttöliittymällä. Koska metsäkoneet työskentelevät hyvin äärimmäisissä olosuhteissa, vaaditaan sekä ohjelmoitavalta logiikalta että kosketusnäytöltä äärimmäistä suorituskykyä, luotettavuutta ja fyysistä kestävyyttä. ProfiPro Oy:n koneissa käytetään Omronin valmistamia ohjelmoitavia logiikoita ja kosketusnäyttöjä. Käytössä olevat Omronin tuotteet ovat hyvin yhteensopivia niin vanhempien kuin uudempienkin tuotteiden osalta, ja eri tuotteet toimivat toistensa kanssa, joten niitä on helppo päivittää tulevaisuudessakin.

3.1 Omron CJ2M -logiikka

ProfiPro Oy:n metsäkoneissa käytetään Omronin valmistamaa CJ2M-mallista ohjelmoitavaa logiikkaa, joka on esitelty kuvassa 8, ja johon on lisätty kaksi Omronin valmistamaa CORT21-mallista CAN-väyläohjainta, joiden avulla logiikka yhdistyy CAN-väylään. CJ2M on laajennuksiensa ansiosta hyvin monipuolinen logiikka. Siihen voi liittää jopa 40 lisäkorttia tarpeen mukaan, ja nämä kortit voidaan synkronoida. Siinä on 20 kiloaskeleen ohjelmointimuisti ja sen sisäiseen datamuistiin mahtuu 32 kilosanaa, mutta datamuisti on laajennettavissa 160-kilosanaiseksi. LD-ohjeen toteutusaika on vain 0,04 mikrosekuntia ja se käyttää vain vähän virtaa. Logiikalla voidaan ohjata 2560 I/O-bittia. Siinä on USB-liitäntä, jonka kautta sitä on helppo ohjelmoida. Logiikka yhdistyy näyttöön Ethernet-kaapelilla, ja siinä on myös rs-232 sarjaliitäntä. Ohjelmoitava logiikka toimii laajalla käyttölämpötila- ja kosteusalueella sekä korkeusalueella. Se kestää hyvin EMC-häiriöitä, tärinää sekä iskuja, ja täyttää EMC zone B -luokituksen, cULus- ja EC-direktiivit. CJ2M -logiikkaa voidaan ohjelmoida neljällä eri kielellä, tikapuu-, sekvenssifunktiokaavioidin, rakenteisella tekstillä ja ohjelistoin. [10.]



Kuva 8. Omron CJ2M -ohjelmoitava logiikka. [10.]

3.2 Omron NS12 -kosketusnäyttö

ProfiPro Oy:n valmistamissa metsäkoneissa käytetään Omron NS12 -mallista kosketusnäyttöä, joka on esitelty kuvassa 9. Siihen asennetun käyttöliittymän avulla koneenkuljettaja voi hallita lähes kaikkia koneen säätöjä. Myös huoltohenkilö voi näytön avulla suorittaa huoltoja, vianetsintää ja säätää laitteita.

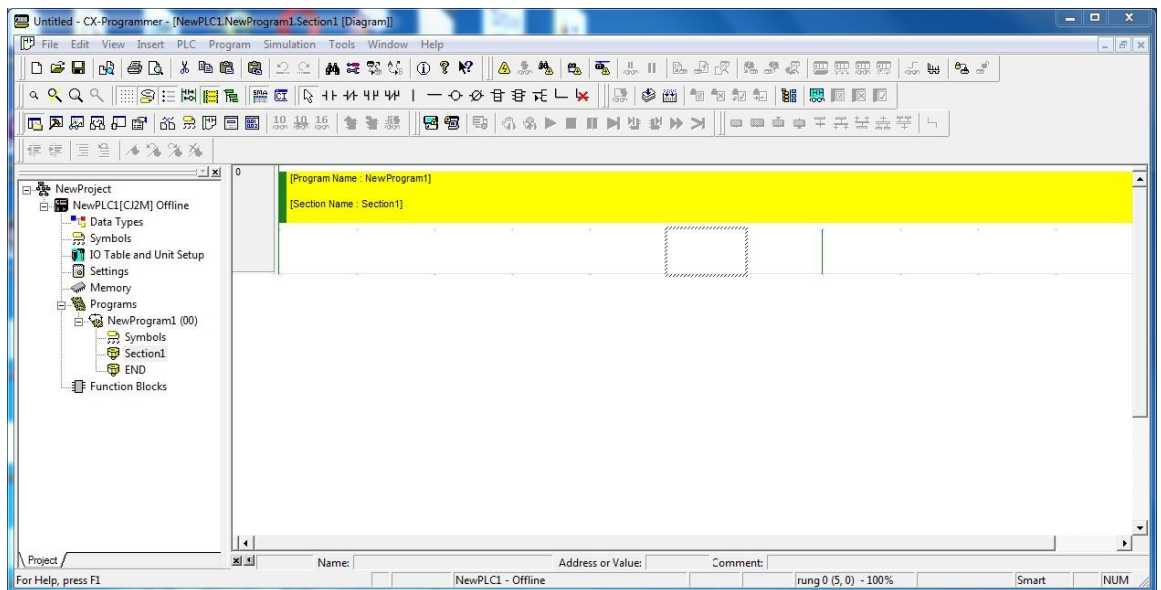


Kuva 9. Omron NS12 -kosketusnäyttö. [11.]

3.3 Omron CX-One -ohjelmisto

Edellä esitellyn ohjelmoitavan logiikan ohjelmointiin ja kosketusnäytön käyttöliittymän luomiseen löytyvät työkalut CX-One-ohjelmistopaketesta. Työssä ohjelmistopaketesta käytetään CX-Programmeria logiikan ohjelmointiin ja CX-Designeria käyttöliittymän luomiseen. Nämä ohjelmat ovat keskenään täysin yhteensopivia, mikä helpottaa huomattavasti ohjelmien tekoa.

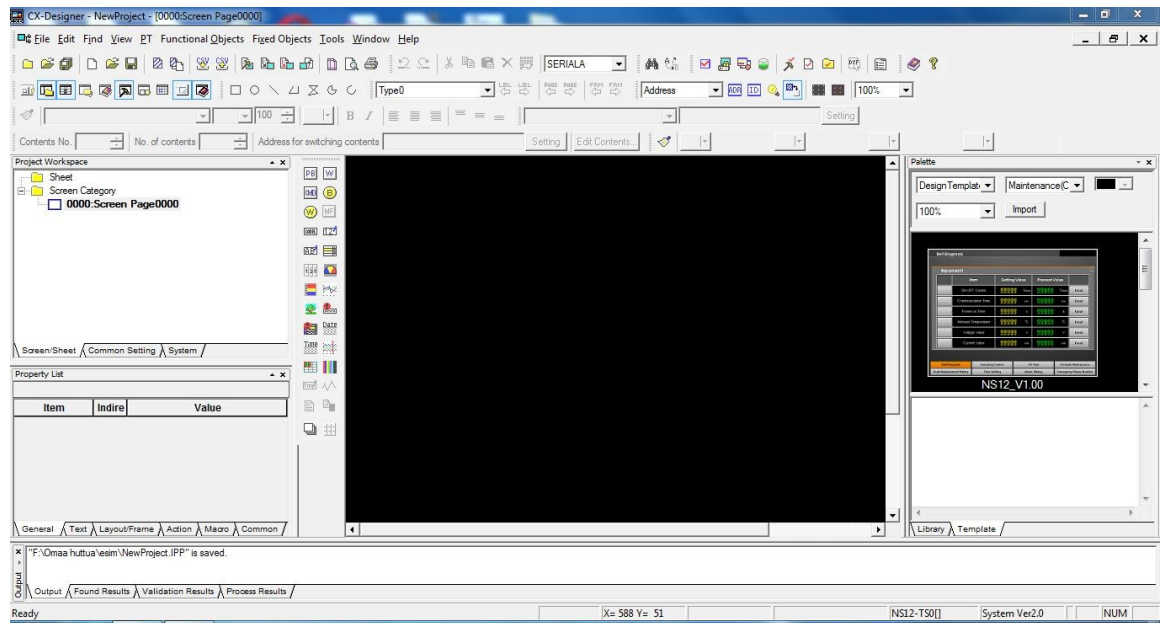
CX-Programmer -ohjelma on ohjelmointiympäristö, jolla voidaan tehdä ohjelmia Omron-logiikoille aina mikrologiikoista suurimpiin Duplex-järjestelmiin asti ja se tukee PLC-tuotesarjaa viimeisen 20 vuoden ajalta. Se mahdollistaa myös vanhemmilla ohjelmointiohjelmistoilla luotujen ohjelmien muuntamisen nykyiselle ohjelmoitavalle logiikalle sopivaksi. Kuvassa 10 on esitelty CX-Programmerissa avautuva näkymä. [11.]



Kuva 10. CX-Programmerin aloitusikkuna.

CX-Designer on skaalattava käyttöpääteohjelmisto, ja sillä voidaan luoda käyttöliittymiä näyttösiivuihin 5,7–12,1 tuuman NS-sarjan ohjelmoitaville päätteille. CX-Designerilla voidaan tarkistaa luodun sovelluksen toimivuus tietokoneella ilman, että tarvitsee olla kytkettyneenä päätteeseen. CX-Designerin avulla voi tehokkaasti luoda näyttösiivuja, simuloida ja ottaa käyttöön

projekti. CX-Designerista löytyy noin tuhat toiminnallista objektiä, joissa on grafiikka ja kehittyneet toiminnot. Kuvassa 11 on esitelty CX-Designer -projektin aloitusnäkyä. [11.]



Kuva 11. CX-Designerin aloitusikkuna.

4 TYÖN VALMISTELU

Työ aloitettiin Nivalassa ProfiPro Oy:n tiloissa aloituskokouksella, jossa päätettiin ja käytiin läpi opinnäytetyön keskeiset seikat. Lisäksi sovittiin laitetoimituksista koululle. Mukana oli koulun ja ProfiPro Oy:n edustajien lisäksi myös edustaja Omronilta.

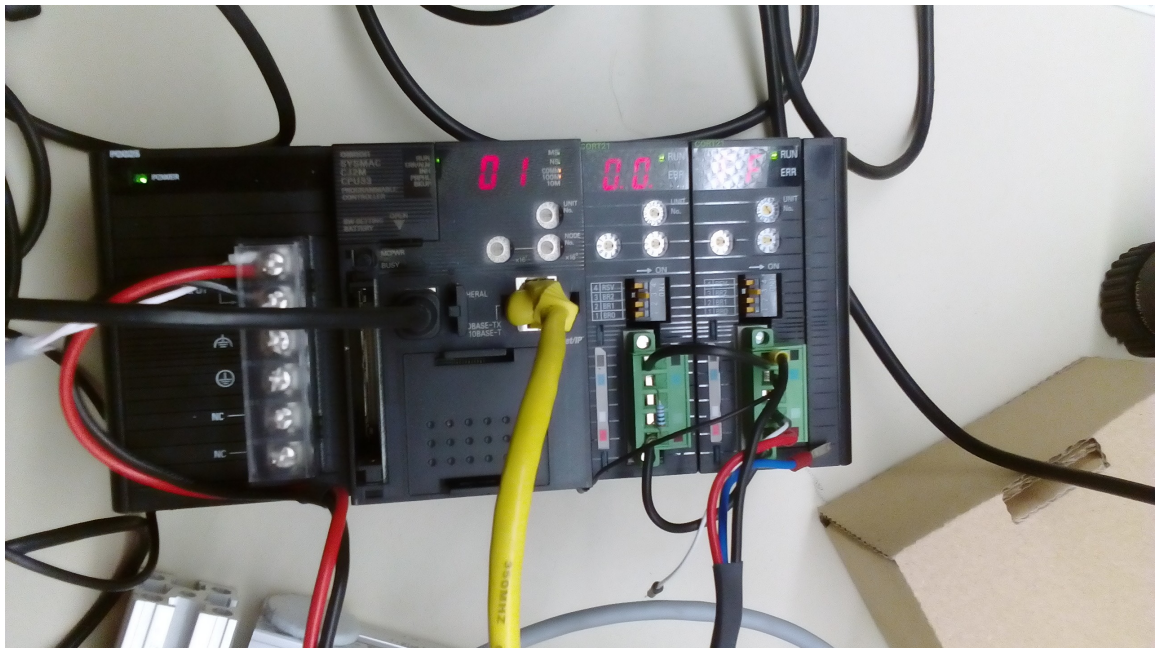
Ennen laitteiston saapumista voitiin aloittaa tutustuminen käytettäviin tiedonsiirtotapoihin sekä käytettäviin laitteisiin ja ohjelmiin. Tiedonsiirron osalta tämä tarkoitti tutustumista CAN-väylän spesifikaatioon ja SAE J1939 -standardiin eri lähteistä, laitteiden osalta logiikan ja näytön manuaalien tutkimista, ja ohjelmiston osalta CX-One-oppaiden sekä opastusvideoiden tarkastelua.

Laitteiston saavuttua aloitettiin ohjelmiston asennus ja kehitysympäristön pystyttäminen. CX-One -ohjelmisto asennettiin kuin mikä tahansa Windows-pohjainen ohjelma. Omron CJ2M -logiikkaan liitettiin kaksi CORT21-mallista CAN-väyläohjainta ja päätelevy. CORT21-väyläohjaimista määritettiin tiedonsiirtonopeudeksi J1939-standardin mukainen 250 kb/s:ssa dip-kytkimillä sekä valittiin kortin järjestysnumero UNIT No. -valintakiekolla. Väyläohjaimiin kytkettiin myös 24 voltin käyttöjännite, sekä CAN high- että CAN low -johtimet, joihin pystyttiin PCAN-USB-laitetta käyttämällä lähettämään CAN-viestejä, joilla voitaisiin simuloida metsäkoneen tuottamia signaaleja. PCAN-USB-laite on esitelty kuvassa 12. PCAN-USB-laitetta varten tietokoneelle ladattiin ja asennettiin peak-system-yrityksen kotisivulta [12] PcanView-ohjelma, joka sisälsi kaiken tarvittavan laitteen käyttöä varten.



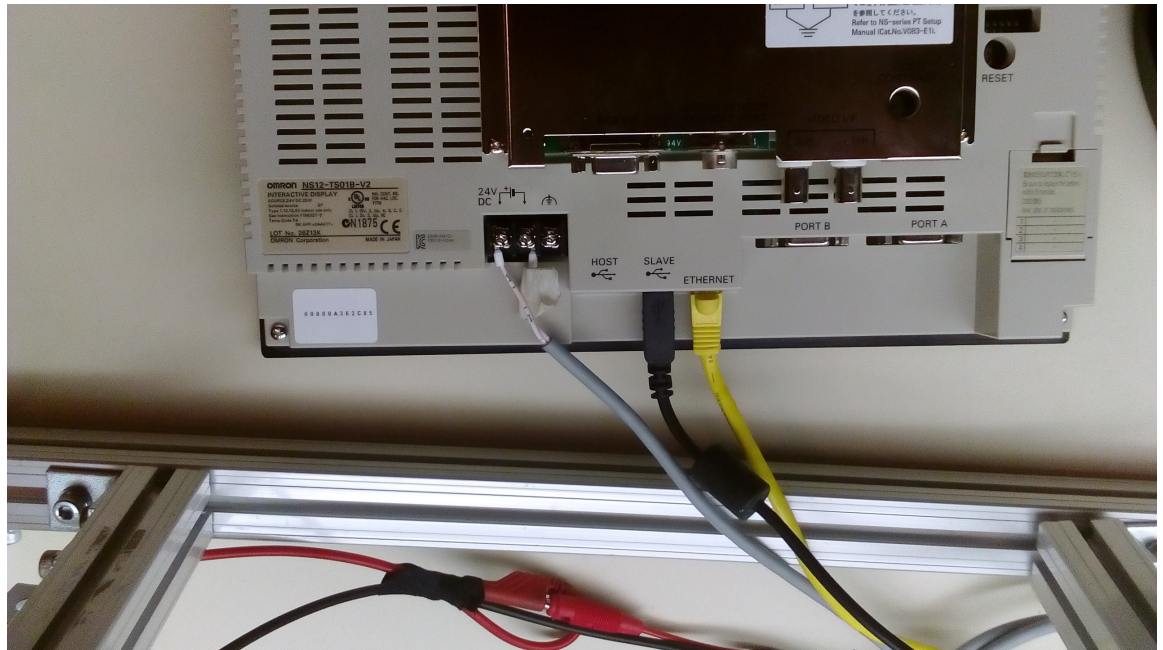
Kuva 12. PCAN-USB-adapteri. [12]

Itse ohjelmitavaan logiikkaan tarvitsi kytkeä vain 24 voltin käyttöjännite, USB-kaapeli ohjelmointia varten ja Ethernet-kaapeli, jolla logiikka kommunikoi näytön kanssa. Logiikan kytkennät on havainnollistettu kuvassa 13.



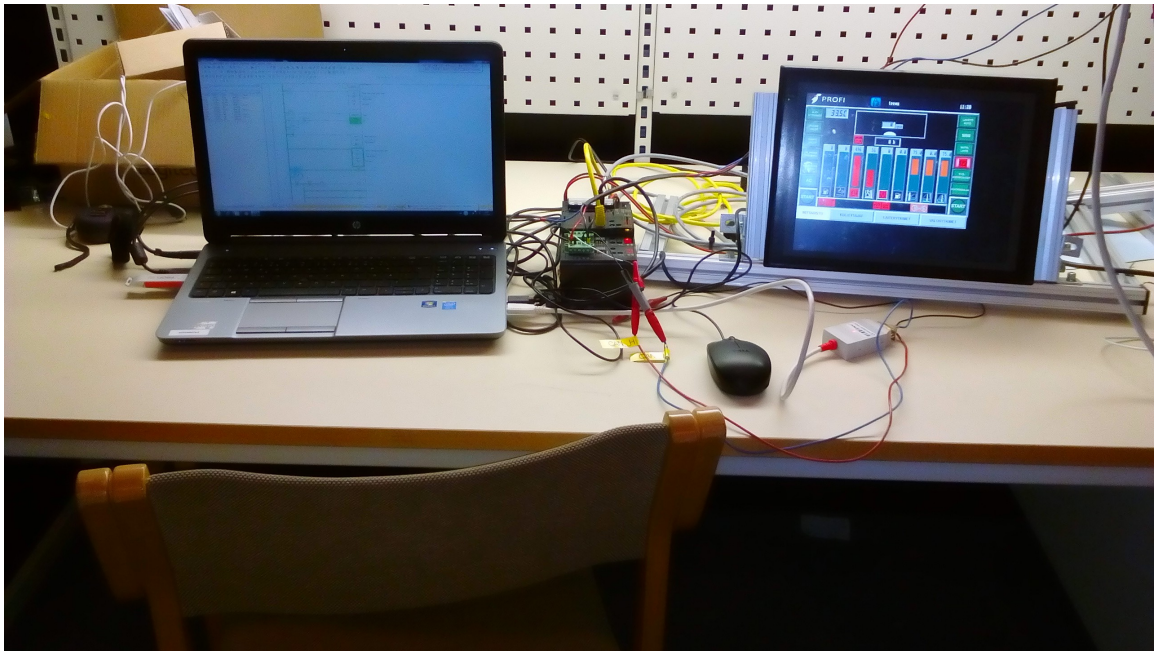
Kuva 13. Omron CJ2M -logiikka kytkettynä.

NS-12 -kosketusnäytön kytkentä oli tehty hyvin yksinkertaiseksi ja se on esitelty kuvassa 14. Näyttöön tarvitsi kytkeä vain 24 voltin käyttöjännite, USB-kaapeli ohjelmointia varten ja logikalta tuleva Ethernet-kaapeli laitteiden välistä kommunikointia varten.



Kuva 14. Omron NS-12 -kosketusnäytön kytkennät.

Käyttöjännitteet kaikkiin laitteisiin saatiin laboratoriopöydän säädettävästä jännitelähteestä. Ohjelmointiin käytettiin koulun kannettavaa tietokonetta, johon CX-One-ohjelmisto sekä PcanView-ohjelma asennettiin. Kuvassa 15 on esitelty valmiiksi saatettu kehitysympäristö.



Kuva 15. Käyttövalmis kehitysympäristö.

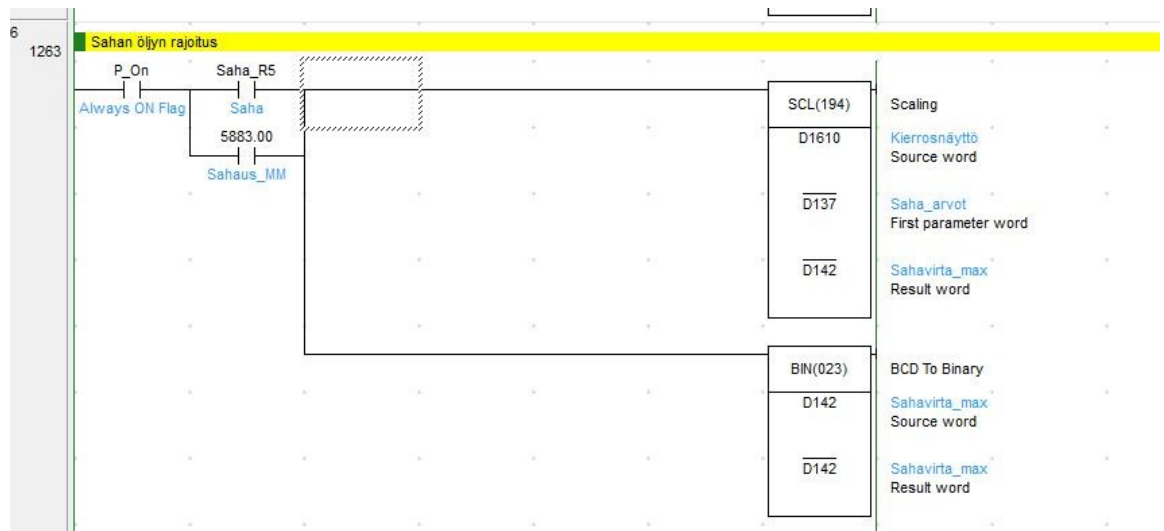
Kehitysympäristön avulla voitiin kehittää ja testata sekä käyttöliittymän että logiikan toimintaa lähes todellisuutta vastaavassa ympäristössä. Lisäksi metsäkoneen sisäistä toimintaa voitiin simuloida CAN-viestein.

5 TYÖN SUORITUS

Aloittamiseen vaadittavien valmisteluiden jälkeen ryhdyttiin tutustumaan jo olemassa oleviin ohjelmiin, niin kosketusnäytön kuin ohjelmoitavan logiikankin osalta. Logiikan ohjelmasta pyrittiin löytämään tapa, jolla käsitellä sahanopeuteen vaikuttavia parametreja, ja näytön käyttöliittymästä pyrittiin löytämään paikka, minne parametrien muuttamiselle voitaisiin toteuttaa käyttöliittymä, sekä tutustumaan käyttöliittymässä käytettävään toteutustyyliin.

5.1 Ohjelmoitavan logiikan ohjelma

ProfiPro Oy:n kanssa pidetyssä palaverissa oli selvitetty bufferimuistipaikat, joita muuttamalla sahanopeutta oli tähän mennessä säädetty. Logiikan ohjelmaan tutustuessi kävi ilmi, että CAN-väylää pitkin tulevat tarvittavat pumpun parametritiedot tallennetaan logiikan buffereihin ja siirretään datamuistiin automaattisesti. Ohjelmasta löytyikin jo valmiina sahan öljyn rajoitukseen liittyvien parametrien skaalaus- ja datamuistiin tallennusfunktiot, joita on selvennetty kuvassa 16. Näistä funktioista saatiin selville datamuistipaikat D138 ja D139, joita käyttöliittymän toteuttamisessa tarvitaan parametrien muuttamiseksi. Lisäksi käyttöliittymään pystyttiin suorittamaan tarvittava parametrien muuttaminen yksinkertaisemmin, joten itse ohjelmoitavan logiikan ohjelman muutoksille ei ollut tarvetta.



Kuva 16. Sahanopeusparametrien skaalaus- ja tallennusfunktiot logiikkaohjelmassa.

Kuvasta käy ilmi, kuinka sahaustoiminto suoritetaan metsäkoneessa. Ensimmäinen ehto on, että koneessa on virrat päällä. Toinen ehto on, että sahauspainiketta on painettu tai mittalaitteen automaattinen sahaus on kytketty. Tämän jälkeen siirrytään sahan nopeuden skaalaukseen, jossa kierrosten lukumäärä skaalataan sahanopeusparametreilla ja lopputulos tallennetaan Sahavirta_max-muuttujaan. Lopuksi Sahavirta_max-muuttuja muutetaan vielä binaari-muotoon, jonka jälkeen se on valmis lähetettäväksi työpumpun ohjausyksikölle.

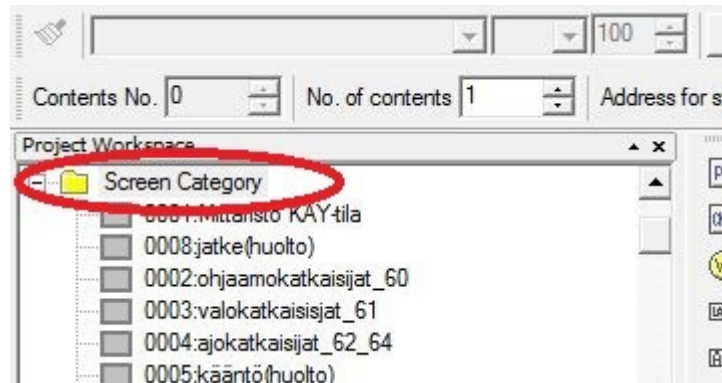
5.2 Kosketusnäytön käyttöliittymän toteuttaminen

Kun tarvittavien parametrien datamuistipaikat oli selvitetty, voitiin aloittaa käyttöliittymän lisäysten suunnittelu. Päätettiin, että sahanopeusparametreille tehdään oma näyttösivu.

Näyttösivun luonti

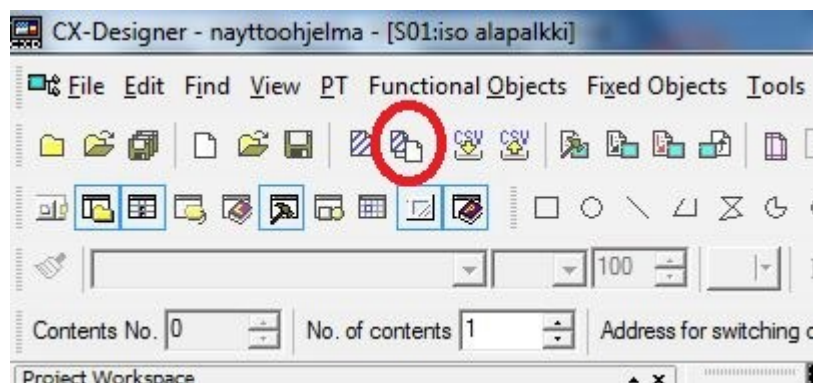
Näyttösivun luonti aloitettiin avaamalla ProfiPro Oy:ltä saatu, metsäkoneissa käytössä oleva näyttöohjelma CX-Designer-ohjelmassa. Project Workspace

-ikkunasta hiiren oikealla painikkeella Screen Category -kansion (kuva 17), päältä valikko, josta valittiin New Screen -vaihtoehto.



Kuva 17. Project Workspace -ikkuna.

Avautuvaan ikkunaan syötettiin luotavalle sivulle kuvaava nimi, tässä tapauksessa Sahanopeuden säätö. Uusi sivu löytyi viimeisenä Project Workspace -ikkunasta. Hiiren oikeaa painiketta painamalla luodun sivun päällä ja valitsemalla Screen/Sheet Property päästiin sivun asetuksiin, jossa Background/Others -välilehdeltä valittiin Background Color -alasvetovalikosta musta taustaväri sivulle, jotta painikkeet ja mittarit erottuisivat taustasta hyvin. Asetuksissa voidaan tehdä määrytyksiä esimerkiksi sivun koolle, mutta muut asetukset olivat jo oletuksena oikein. Sivulle lisättiin muillakin sivuilla käytettävä yläpalkkikerros painamalla työkalupalkista Apply Sheet -painiketta (kuva 18), jonka jälkeen avautuvasta ikkunasta etsittiin äsken luotu sivu ja merkittiin listasta Sheet No. 0 käyttöön.



Kuva 18. Sheet-kerroksen valintapainike.

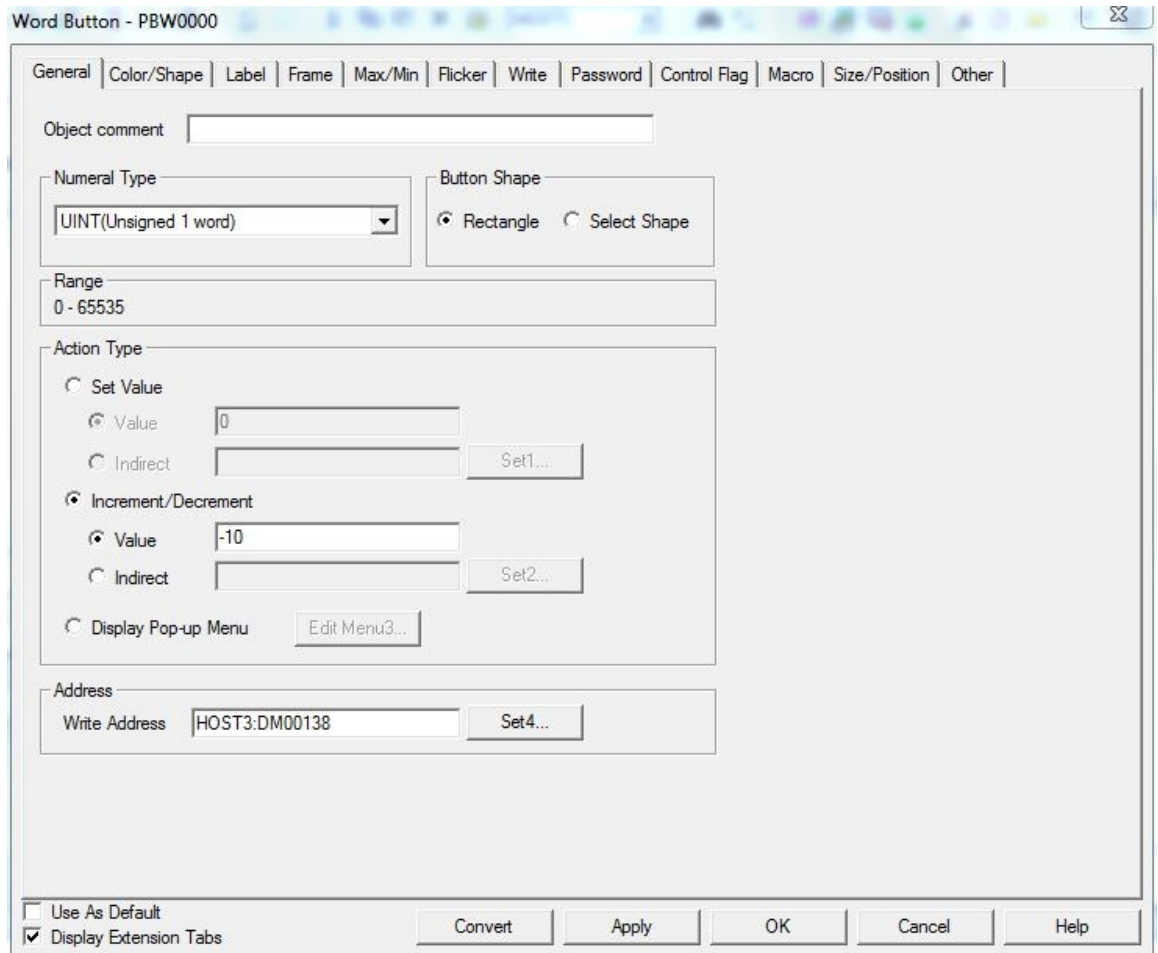
Painikkeen luonti

Kun näyttösivun pohja oli saatettu valmiiksi, voitiin sivulle alkaa luomaan toimintoja. Palaverissa oli sovittu, että molempaa sahanopeutta säätävää parametria tulisi pystyä muuttamaan erikseen, ja niitä säädettäisiin yhden ja kymmenen yksikön välein. Tähän tarkoitukseen soveltui parhaiten Functional Object -työkaluriviltä löytyvä Word Button -painike (kuva 19).



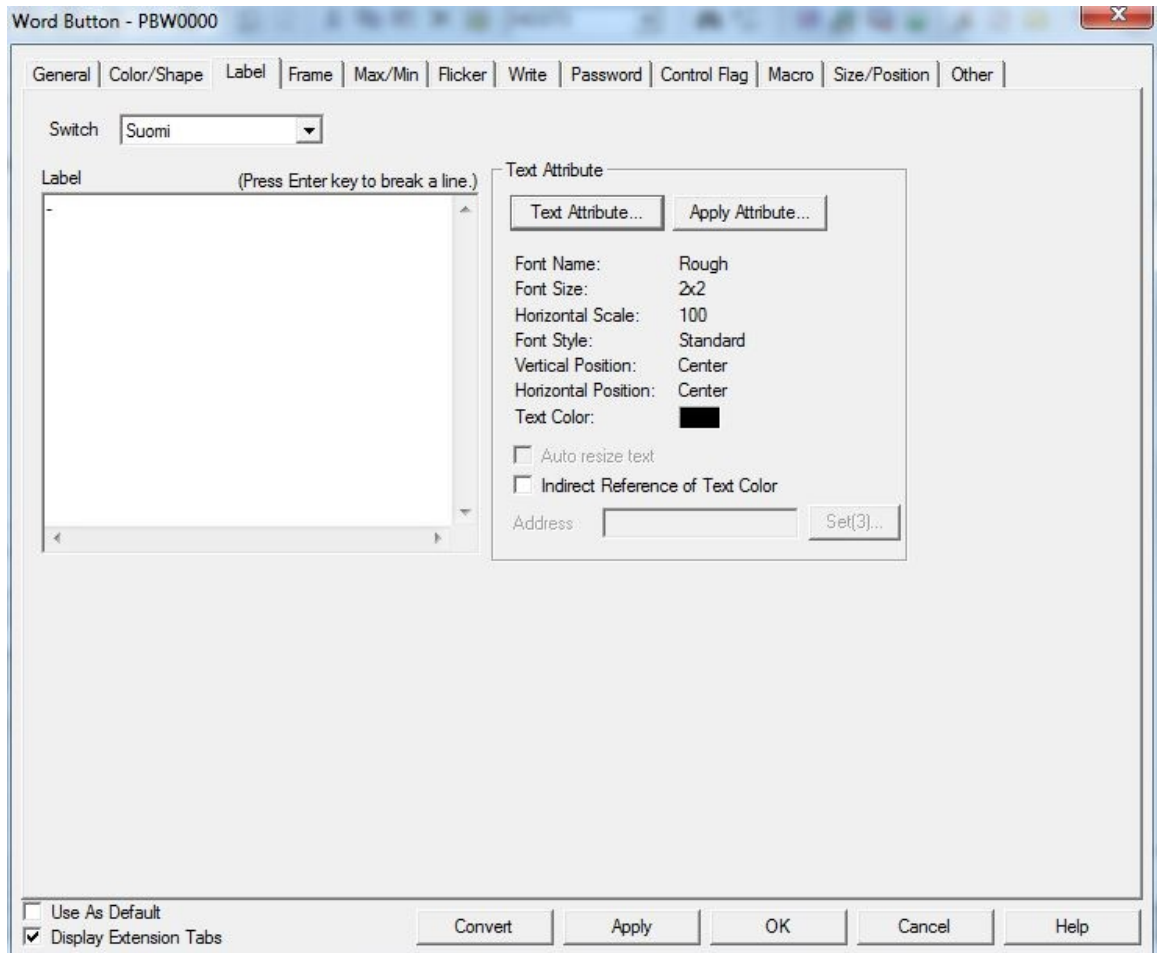
Kuva 19. Word Button -painikkeen valinta.

Word Buttonin valinnan ja paikoittamisen jälkeen päästiin sen asetuksiin valitsemalla hiiren oikean näppäimen avaamasta valikosta Property... -valinta. Ensimmäisen painikkeen tarkoitus oli muuttaa D138-muistipaikan arvoa kymmenellä alaspäin, joten General-välilehdeltä valittiin sen Numeral Type -tyypiksi UINT, koska arvot ovat positiivisia kokonaislukuja, ja Button Shape -muodoksi Rectangle, jotta painikkeesta tulisi nelikulmainen. Action type -tyypiksi valittiin Increment/Decrement ja arvoksi -10, joka muuttaa tämänhetkistä arvoa syötetyn arvon verran. Write Address -kenttään osoitteeksi tuli HOST3:DM00138. General-välilehti valintoineen on nähtävissä kuvassa 20.



Kuva 20. Word Button -painikkeen toimintoasetukset.

Label-välilehdeltä määritetään painikkeen teksti. Koska ensimmäisen painikkeen tehtävä on vähentää arvoa, sen tekstiksi Label-kenttään kirjoitetaan miinusmerkki. Text Attribute ... -painikkeella päästiin tekemään tarkemmat määrittymiset tekstin tyypille, koolle sekä asemoinnille, jotta ne olisivat yhdenmukaisia muun ohjelman kanssa. Tekstiasetukset on selvitetty kuvassa 21.



Kuva 21. Käyttöliittymän tekstiasetukset.

Frame-välilehdellä valittiin painikkeen reunusasetukset. Three-dimensional Frame -asetus ja Draw Border -asetus merkattiin käyttöön, jotta painike korostuu painettavaksi. Palaverissa oli sovittu parametrien minimi- ja maksimiarvot. Max/Min välilehdeltä asetettiin Maximum Limit -arvoksi 2500 ja Minimum Limit -arvoksi 2000.

Vastaavanlainen painike tehtiin parametrin pienentämiseksi yhdellä yksiköllä, mutta painikkeesta tehtiin selkeyden vuoksi hieman pienempikokoinen. Tämä oli yksinkertaista CX-Designerin mahdollistaman toimintojen kopioinnin ansiosta. Samoin tehtiin myös painikkeet parametrin kasvattamiseksi, vaihtamalla vain Increment/Decrement arvoksi yksi ja kymmenen. Myös toisen parametriarvon muutospainikkeet tehtiin kopioimalla aiemmat painikkeet ja vaihtamalla Write Address -kenttään osoitteeksi HOST3:DM00139.

Mittarit ja numeronäytöt

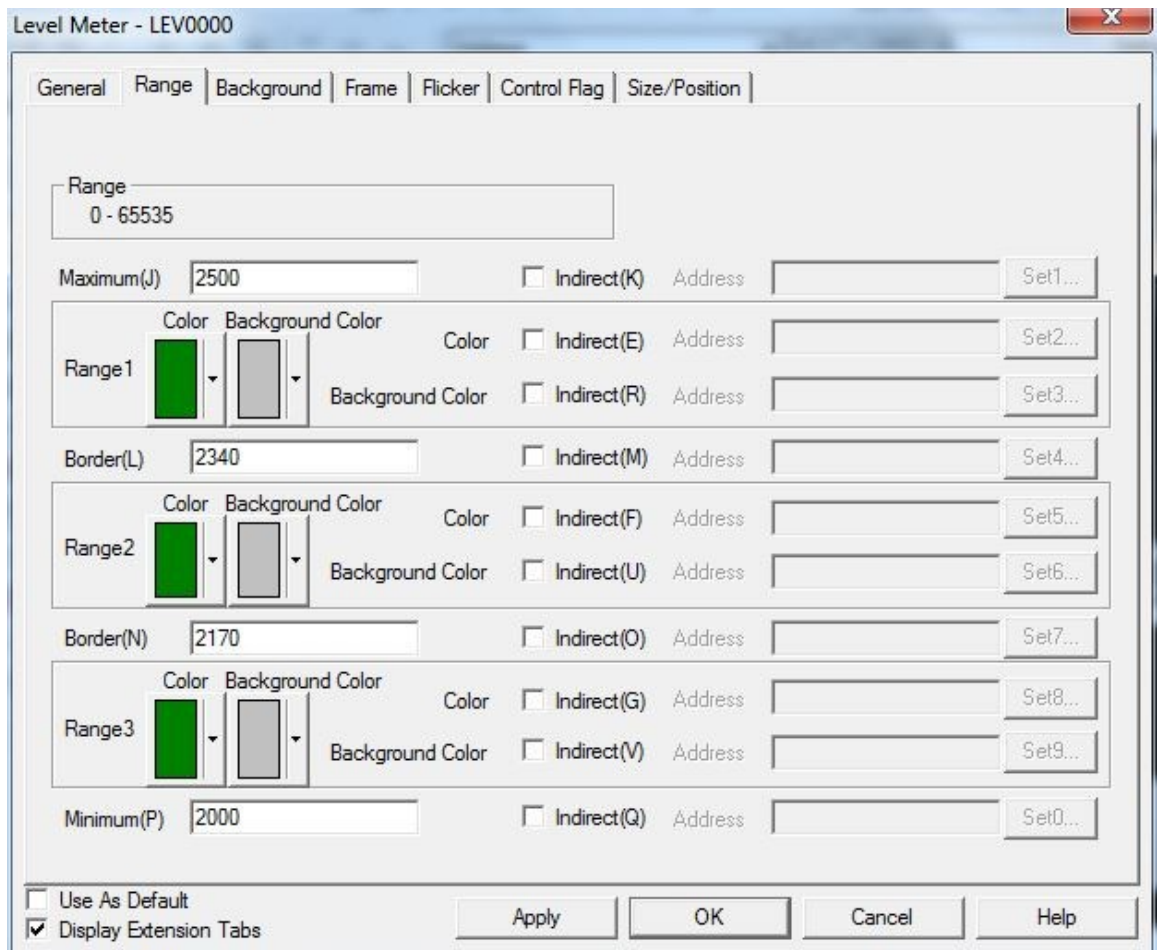
Parametrien arvot ja niiden avulla saatava Maksimi sahavirta -arvo oli syytä saattaa käyttäjän tietoon, ja siihen tarkoitukseen päätettiin käyttää sekä visuaalista tasomittaria että arvot numeerisessa muodossa esittävää numeronäyttöä. Näihin tarkoituksiin soveltui parhaiten Functional Object -työkaluriviltä löytyvä Numeral DisplayInput -näyttö ja Level Meter -mittari (kuva 22).



Kuva 22. Numeronäytön ja tasomittarin sijainti työkalupalkissa.

Ensimmäisen numeronäytön tarkoitus oli esittää datamuistipaikan 138 arvo numeerisessa muodossa käyttäjälle. Näyttö luotiin ja paikoitettiin samaan tapaan kuin painikkeet aiemmin. Luodun näytön asetuksista General -välilehdellä muutettiin Storage Type -tyyppi UINT-muotoiseksi ja Format-valinnassa valittiin näytön näyttämä neljännumeroiseksi kokonaisluvuksi sekä osoitteeksi Address-kenttään HOST3:DM00138. Koska numeronäytön avulla voidaan myös syöttää arvoja ohjelmaan, niin Max/Min-välilehdellä määrättiin rajat syötettäville arvoille, jotta ohjelma antaisi virheilmoituksen ja hylkäisi syötteen, jos syötetty arvo ei ole rajojen sisällä. Maksimiarvoksi määritettiin 2500 ja minimiarvoksi 2000. Samoin toimittiin datamuistipaikan 139 arvon kanssa, vaihtaen Address-kenttään HOST3:DM00139 ja maksimiarvoksi 1800 ja minimiarvoksi 1400. Myös edellä olevien parametrien muuttamisella aikaan saatu Maksimi sahavirta -arvo haluttiin käyttäjän tietoon, ja se luotiin samalla tavalla, muuttaen Address-kenttään HOST3:DM00142. Lisäksi Maksimi sahavirta -arvo muuttuu vain parametrien vaikutuksesta, joten siitä jätettiin arvon syöttämismahdollisuus pois valitsemalla asetuskunan Control Flag -välilehdeltä Input-kohtaan Disable.

Tasomittari luotiin ja paikoitettiin edelleen samaan tapaan kuin painikkeet aiemmin. Ensimmäisen mittarin tarkoitus oli esittää datamuistipaikan 138 arvo visuaalisesti käyttäjälle. Mittarin asetuksista General-välilehdeltä muutettiin Display Direction -valikosta Right, jotta mittari arvon kasvaessa täyttyisi vasemmalta oikealle. Välilehdellä poistettiin myös valinta Scale-kohdasta, koska skaalaukselle ei ollut tarvetta, sekä muutettiin Storage Type -tyyppi UINT-muotoiseksi. Address-kenttään osoitteeksi muutettiin HOST3:DM00138. Range-välilehdellä mittari oli valmiiksi jaettu kolmeen osaan, jos haluttaisiin käyttää erivärisiä arvoalueita. Tälle ei nähty tarvetta, joten arvoalue jaettiin kolmeen osaan, mutta kaikille alueille annettiin sama väri. Ensimmäisen tasomittarin Range-välilehden asetukset on esitelty kuvassa 23.



Kuva 23. Mittarin arvoalueiden esitys.

Frame-välilehdeltä poistettiin valinta Draw Border -kohdasta, jotta mittari sulautuu taustaan paremmin. Samoin toimittiin datamuistipaikan 139 arvoa

osoittavan mittarin kanssa, vaihtaen Address-kenttään HOST3:DM00139 ja Maksimi sahavirta -arvoa osoittavan mittarin kanssa vaihtaen Address-kenttään HOST3:DM00142.

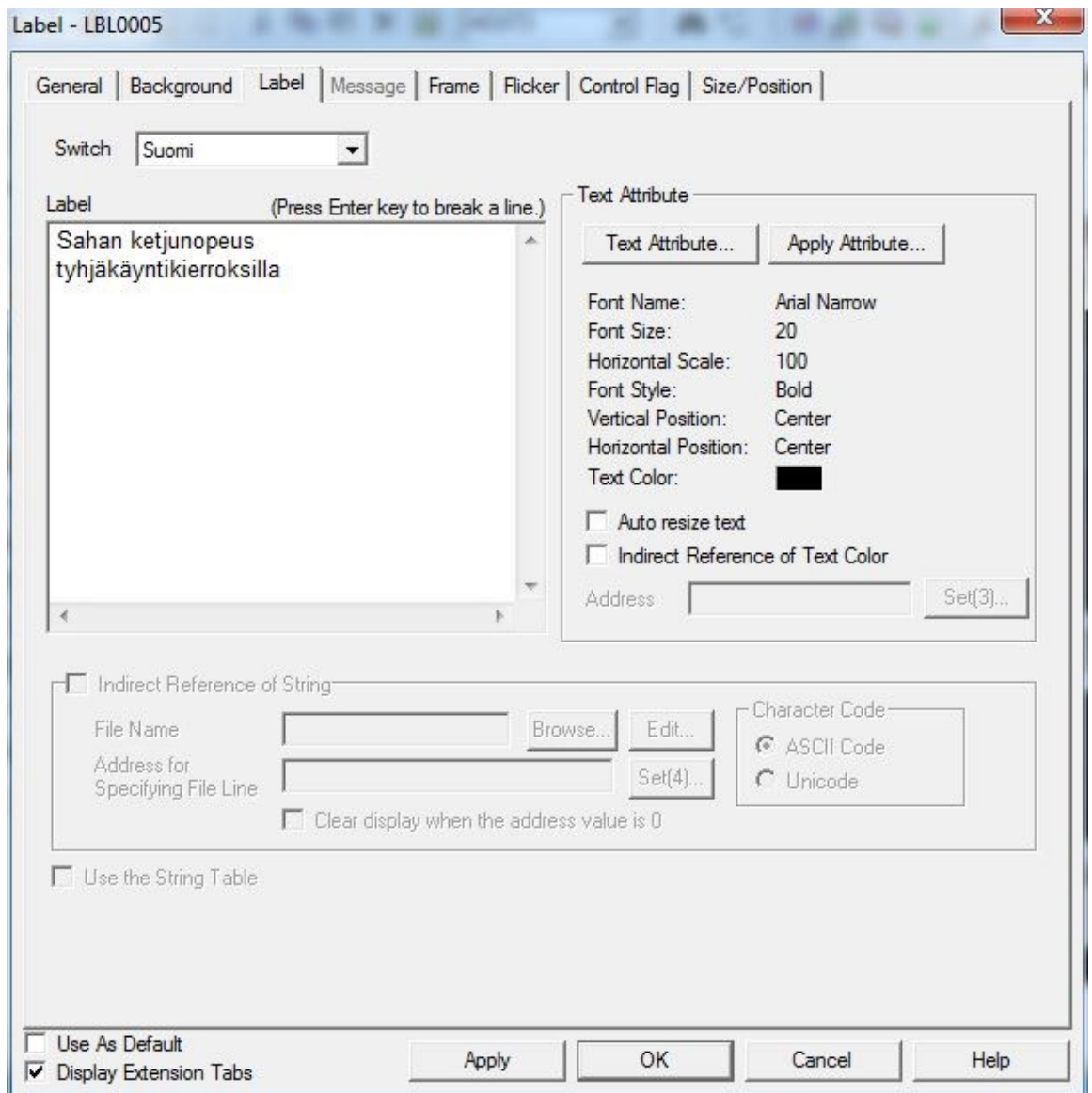
Säätötoimintojen otsikot

Käyttöliittymän selkeyttämiseksi eri toiminnoille oli luotava kuvaavat otsikkokehykset. Tähän tarkoitukseen Functional Object -työkaluriviltä löytyi Label-kehystyökalu (kuva 24).



Kuva 24. Kehystyökalun valinta -painike.

Otsikkokehys luotiin ja paikoitettiin samaan tapaan kuin aiemmat toiminnot. Kehyksen asetuksista Label-välilehdellä määritettiin Label-kenttään kullekin toiminnolle kuvaava nimi ja Text Attribute... -painike painamalla tarkemmat tekstiä koskevat asetukset. Asetukset ovat näkyvissä kuvassa 25 ja ovat samat kaikissa Label-kehyksissä.



Kuva 25. Kehyksen tekstiasetukset.

Lopuksi kehyksien teksteille tehtiin lokalisointi. Sovittiin, että tekstit käännetään suomeksi, ruotsiksi ja englanniksi. Lokalisointi onnistui valitsemalla hiiren oikealla painikkeella otsikkokehysten päältä avatusta valikosta Edit Properties -vaihtoehto. Edit Properties -ikkunassa pystyttiin syöttämään eri kielille omat tekstinsä.

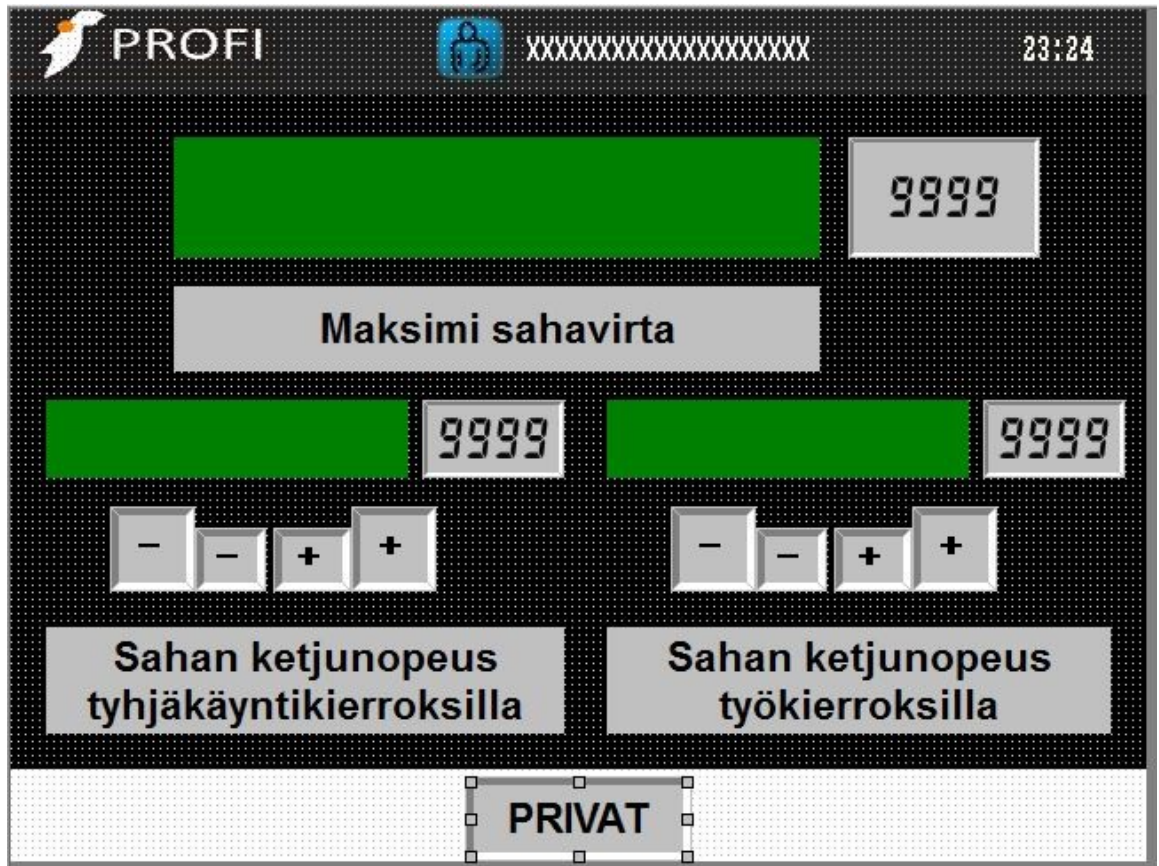
Sahanopeuden säätö -näytösivulle pääsyä tuli rajoittaa huoltohenkilöihin väärinkäytösten välttämiseksi, joten siirtyminen näytösivulle päätettiin tehdä PRIVAT-sivulle, koska sivulle pääsy on suojattu salasanalla. Näytösivulle lisättiin poistumispainike, joka vie takaisin PRIVAT-sivulle. Se tapahtui valitsemalla

Functional Object -työkaluriviltä Command Button -työkalu (kuva 26) ja paikoittamalla se näyttösivun alalaitaan.



Kuva 26. Command Button -työkalun valinta.

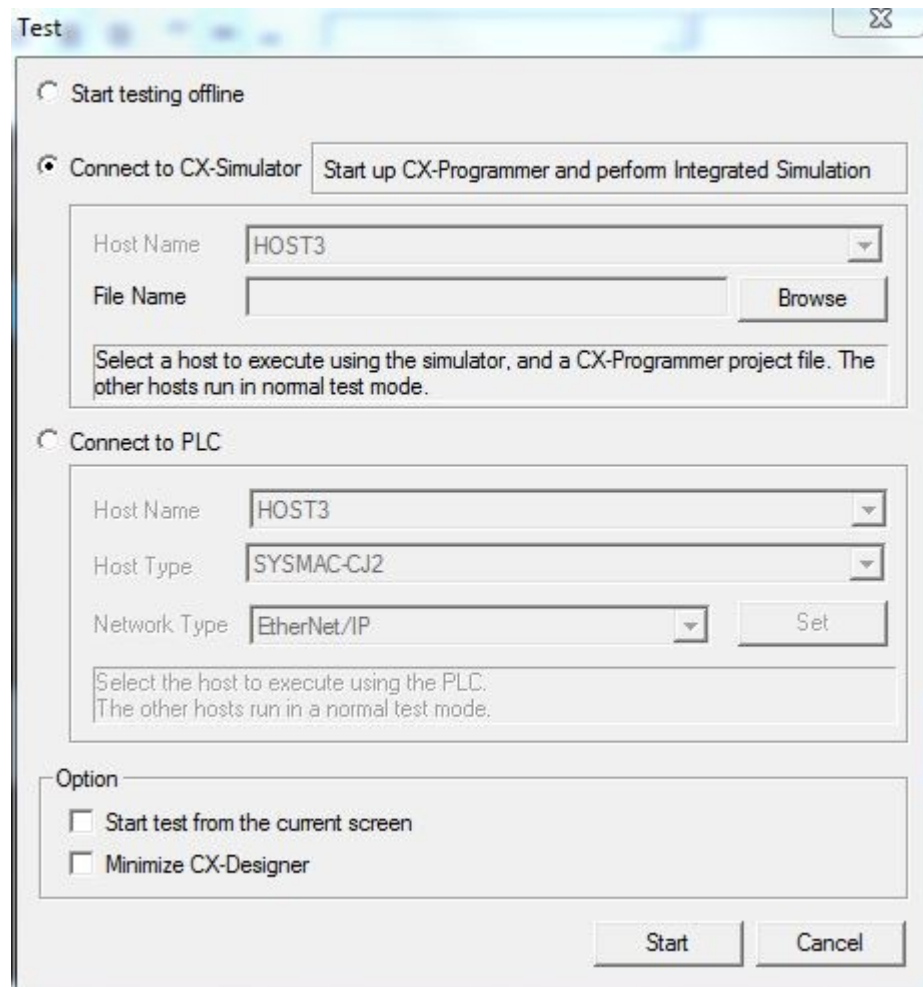
Command Button -painikkeen asetuksista General-välilehdeltä valittiin paluusivuksi Specified Screen -kohdassa Select 1 -napilla sivu 0049:PRIVAT ja Button Shape -kohdassa Rectangle. Label-välilehdellä Label-tekstikenttään kirjoitettiin PRIVAT ja muutoin tekstiasetukset olivat samoin kuin otsikkokehystä luotaessa. Frame-välilehdellä painikkeelle valittiin jälleen Three-dimensional Frame ja Draw Border -valinnat käyttöön, jotta se erottuisi painettavaksi painikkeeksi. Lopputuloksena saatiin valmis näyttösivu, jolta löytyvät kaikki tavoitteena olleet toiminnot, ja se on esitelty kuvassa 27.



Kuva 27. Valmis näyttösivu.

PRIVAT-sivulle päästiin valitsemalla se Project Workspace -ikkunasta. Sivulle luotiin Command Button -painike johtamaan käyttäjä Sahanopeuden säätö -sivulle samalla tavoin kuin edellä kuvattu PRIVAT-sivulle palauttava painike, vaihtamalla vain Specified Screen -kohdassa sivuksi 0056:Sahanopeuden säätö ja Label-tekstikenttään teksti SAHANOPEUDEN SÄÄTÖ.

CX-Designer-ohjelmassa oli mahdollista ajaa simulaatiota kolmessa eri moodissa valitsemalla Tools-valikosta Test-valinta, ja avautuvasta Test-ikkunasta, joka on esitelty kuvassa 28, valitsemalla haluttu moodi.



Kuva 28. Testaus-moodin valintaikkuna.

Offline-moodissa oli mahdollista testata CX-Designer-ohjelman sisällä toimintoja, ja tätä toimintoa käytettiin esimerkiksi selvittäessä, muuttavatko painikkeet haluttuja arvoja. Connect to CX-Simulator -moodissa pystyttiin yhdistämään käyttöliittymäohjelma logiikkaohjelmaan ja testaamaan, että integraatio CX-Designer- ja CX-Programmer -ohjelman välillä toteutuu, ja halutut arvot muuttuvat myös logiikkaohjelman muistissa, mutta tämäkin tapahtui vain ajettaessa ohjelmia tietokoneella. Kolmas moodi eli Connect to PLC -moodi mahdollisti realistisen käyttöliittymän simulaation todellisen ohjelmoitavan logiikan kanssa.

6 TESTAUS

Valmiita ohjelmia testattiin kehitysympäristössä lataamalla käyttöliittymäohjelma CX-Designer-ohjelman avulla kosketusnäyttöön ja logiikkaohjelma ohjelmoitavaan logiikkaan CX-Programmer-ohjelman avulla. CX-Designer-ohjelmassa tämä tapahtui valitsemalla PT -valikosta Transfer-alavalikko, josta valittiin Quick Transfer(To PT)... -valinta. Auenneesta ikkunasta tarkistettiin, että USB näkyy Transfer Setting (Common) -listassa ja painettiin Yes-nappia. Tämän jälkeen ohjelma kysyi salasanaa. Salasanan syötön jälkeen kysytään vielä varmistus, joka kuitataan OK-napilla, jonka jälkeen siirron edistyminen näkyy sekä tietokoneen näytöllä että kosketusnäytössä. Siirron valmistumisesta tulee jälleen ilmoitus, joka kuitataan ja jonka jälkeen kosketusnäyttö käynnistyy uudelleen. Vielä kerran tulee ilmoitus siirron onnistumisesta, joka kuitataan OK-napilla.

CX-Programmer-ohjelmassa ohjelman lataus onnistui seuraavasti: ennen latauksen aloittamista valittiin PLC -valikosta Work Online -valinta ja hyväksyttiin avautuva ilmoitus. Siirron aloittamiseksi PLC-valikosta valitaan Transfer-alavalikko, josta valitaan To PLC... -valinta. Tämän jälkeen valitaan siirrettävät kohteet ja painetaan OK-nappia. Seuraa kaksi varmistusikkunaa, jotka kuitataan OK-napilla. Sitten avautuu edistymisruutu tietokoneen näytölle, ja siirron valmistumisesta ilmoitetaan, jonka jälkeen vielä kysytään varmistus. Molemmat kuitataan painamalla OK-nappia.

Sen jälkeen tarvitsi simuloida metsäkoneen logiikkaohjelmasta virta-avaimen ”virrat päällä” -asentoa logiikkaohjelmasta, joka on ehtona monelle toiminnolle, ja myös näytön käyttämiselle. Etsittiin ohjelmasta Virtapäällä-niminen kontakti, ja hiiren oikealla painikkeella kontaktin päältä avautuvasta valikosta valittiin Set-alavalikko, josta On-valinta. Nyt myös näyttö tuli aktiiviseksi.

Maksimi sahavirta -arvo riippuu myös moottorin kierrosten määrästä, joten oli syytä simuloida myös moottorin kierroslukutietoa logiikalle ja tähän tarvittiin PCAN-USB-laitetta. PcanView-ohjelma oli yksinkertainen käyttää: aluksi aloitusikkunasta varmistettiin oikea laite ja valittiin käyttöön Bit rate

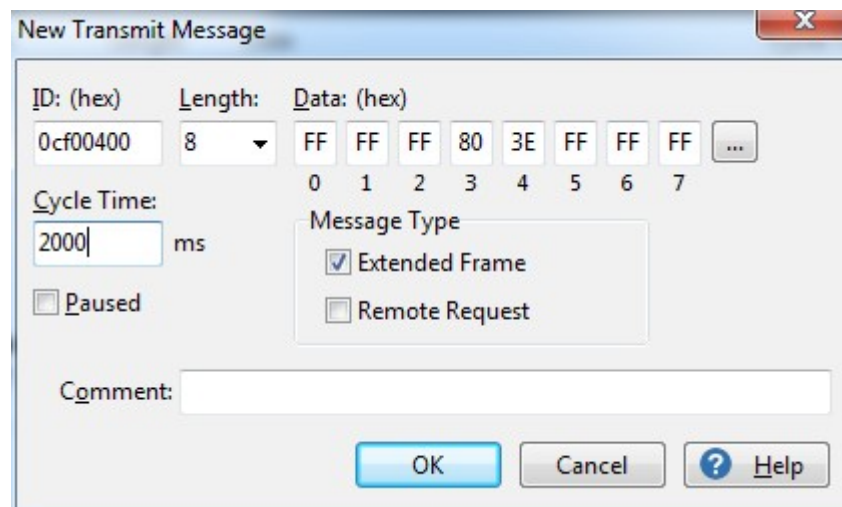
-alaszvetovalikosta SAE J1939-standardin mukainen väylänopeus 250 kBit/s ja Filter settings -kohdasta Extended eli 29 -bittinen ID-kehys, ja hyväksyttiin valinnat painamalla OK. Tämän jälkeen avautui ohjelman pääikkuna, jossa näkyivät sekä vastaanotetut että lähetetyt CAN-viestit. Valitsemalla New Message päästiin luomaan uusi CAN-viesti. Moottorin PGN-numero, viestin prioriteetti ja Source Address selvitettiin ProfiPro Oy:ltä saadusta spesifikaatiosta ja näistä voitiin luoda viestin tunnistekenttä. Kuvassa 29 on esitelty moottorin PGN- ja SPN-taulut.

PGN 61444 (R) Electronic Engine Controller 1			- EEC1
Engine related parameters			
Transmission Repetition Rate:	engine speed dependent		
Data Length:	8		
Extended Data Page:	0		
Data Page:	0		
PDU Format:	240		
PDU Specific:	4	PGN Supporting Information:	
Default Priority:	3		
Parameter Group Number:	61444	(0xF004)	
Start Position	Length	Parameter Name	SPN
1.1	4 bits	Engine Torque Mode	899
1.5	4 bits	Actual Engine - Percent Torque High Resolution	4154
2	1 byte	Driver's Demand Engine - Percent Torque	512
3	1 byte	Actual Engine - Percent Torque	513
4-5	2 bytes	Engine Speed	190
6	1 byte	Source Address of Controlling Device for Engine Control	1483
7.1	4 bits	Engine Starter Mode	1675
8	1 byte	Engine Demand – Percent Torque	2432
SPN 190 Engine Speed			
Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.			
Data Length:	2 bytes		
Resolution:	0.125 rpm/bit, 0 offset		
Data Range:	0 to 8,031.875 rpm	Operational Range: same as data range	
Type:	Measured		
Supporting information:			
PGN	61444		

Kuva 29. Moottorin PGN- ja SPN-taulut [13].

Tunnistekenttä alkaa viestin prioriteetilla, joka tässä tapauksessa oli 3, ja External datapage -bitti ja Datapage -bitti määritettiin nolliksi. PGN-numero oli F004H, joten siitä saatiin PDU Format -tavuiksi F0H ja PDU Specific -tavuiksi 04H. Source Address -arvo oli spesifikaatiossa 0, joten Source Address -tavuiksi laitettiin 00H. Binaarina tunnistekentän arvoksi saatiin siis 011 00 1111 0000 0000 0100 0000 0000, ja katkaisemalla tämä oikein saatiin binaarina 0 1100 1111

0000 0000 0100 0000 0000 eli heksadesimaalina 0CF00400H, joka on siis lähetettävän viestin tunnustekenttä. Viestidatan pituudeksi valittiin Length-kohtaan 8. Spesifikaatiosta oli selvitetty, että viestidatan neljäs ja viides tavu ilmoittavat moottorin kierrosluvun, ja se ilmaistaan 0,125 kierrosta/minuutissa/bitti. Viestidatalla ei haluttu muuttaa kierrosluvun lisäksi muita arvoja, joten muut tavut asetettiin arvoon FH. Aluksi kierrosluvuksi tavoiteltiin 870 kierrosta minuutissa, joka vastaa metsäkoneen tyhjäkäyntikierroksia. Tällöin laskettiin desimaaliarvo jakamalla kierrosarvo arvolla 0,125. Tuloksena saatu luku 6960 muutettiin heksadesimaaliksi, ja saatiin 1B30H, joka sitten sijoitettiin viestidatan neljänneksi ja viidenneksi tavuiksi siten, että neljäs tavu sai arvon 30H ja viides tavu arvon 1BH, koska tavut lähetetään käänteisessä järjestyksessä väylään. Viestidataksi saatiin siis FFFFFFF301BFFFFFFFH, ja tämä syötettiin Data: (hex) -kenttien arvoksi. Cycle Time -kentässä pystyttiin määrittämään, kuinka usein viesti lähetetään väylään, ja tähän tapaukseen väliksi valittiin 2000 millisekuntia. Kuvassa 30 on esitetty eräs luotu viesti, jossa simuloidaan moottorin kierrosluvuksi 2000 kierrosta minuutissa.



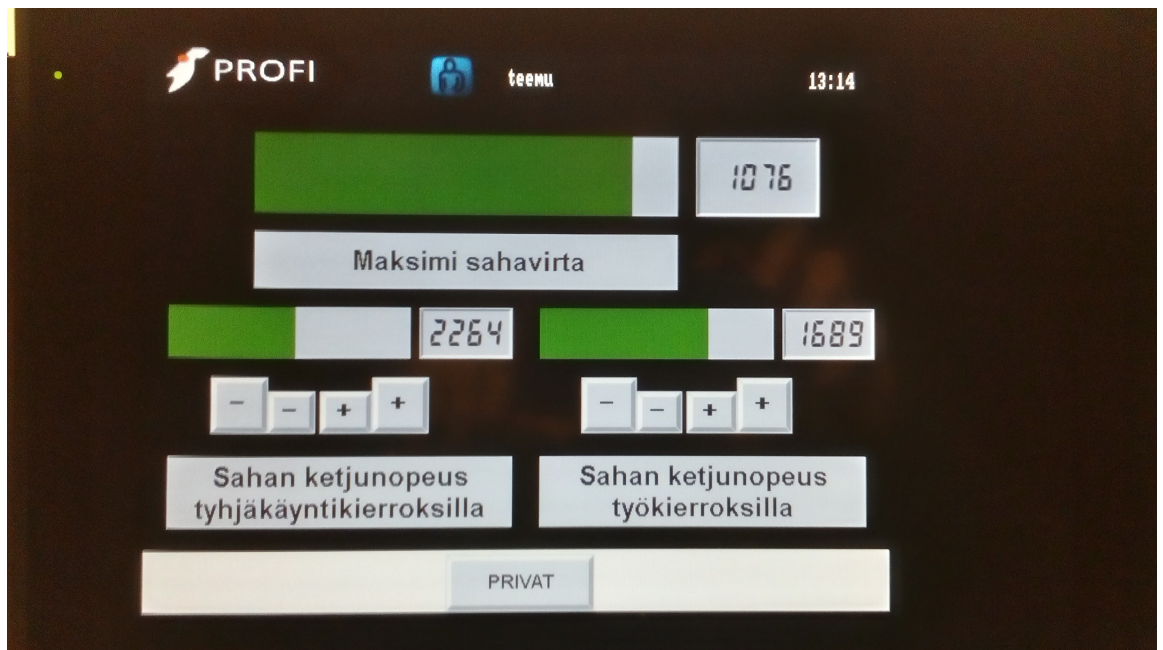
Kuva 30. Luotu SAE J1939 -standardin mukainen kierroslukua simuloiva CAN-viesti.

OK-napin painamisen jälkeen ohjelma alkoi lähettää luotua viestiä CAN-väylään, ja onnistunut viestin luonti ja lähetys voitiin todeta seuraamalla näytöstä kierrosmittaria, joka on esitelty kuvassa 31.



Kuva 31. Käyttöliittymän kierrosmittari, johon on simuloitu kierroksia CAN-väylää pitkin.

Ohjelmaa testattiin useilla eri kierrosluvuilla välillä 870–2000 kierrosta minuutissa. Eri kierrosluvuilla käytiin läpi säätöparametrien eri arvot, ja seurattiin niiden vaikutusta Maksimi sahavirta -arvoon sekä testattiin, että jokaisen painikkeen toiminta on toivotunlainen. Sekä maksimi sahavirta -arvon että painikkeiden todettiin toimivan aiotulla tavalla. Kuvissa 32 ja 33 on esimerkki säädetyistä Maksimi sahavirta -arvosta ilman kierroksia ja simuloituilla kierroksilla, joista havaitaan kierrosten vaikutus Maksimi sahavirta -arvon säädössä.



Kuva 32. Maksimi sahavirta ilman simuloituja kierroksia.



Kuva 33. Maksimi sahavirta, kun simuloitu kierrosluku on 1800 kierrosta minuutissa.

Testeissä testattiin myös, ettei parametreille voi antaa kiellettyjä arvoja, sekä numeronäyttöjen ja tasomittareiden lukemien oikeellisuutta. Siirtymäpainikkeet

todettiin myös toimivan odotetusti ja vievän käyttäjän tarkoitetulle sivulle. Kokonaisuudessaan todettiin ohjelman tekevän sen, mikä oli tavoitteeksi asetettukin.

7 YHTEENVETO

Insinööriyön tavoitteena oli saada helpotettua huoltohenkilön suorittamaa harvesteripään sahanopeuden säätämistä, ja ainakin simulaationa suoritettussa testauksessa tässä onnistuttiinkin mainiosti. Aikataulu meni hieman tiukille loppua kohden, joten ajan puutteen ja pitkien välimatkojen vuoksi aikaansaatus sahanopeuden hallintasovellusta ei päästy testaamaan oikeassa harvesterissa. Työn tuloksena saatua ohjelmaa voidaan kuitenkin myöhemmin hyödyntää käytännössä osana metsäkoneen käyttöliittymää ja se tulee säästämään aikaa ja vapauttamaan muitakin resursseja metsäkoneen huoltotoiminnassa.

Toisena tavoitteena oli saada aikaan dokumentti, jonka avulla pystytään ymmärtämään ja toteuttamaan sahanopeuden säätämisen kaltaisia muutoksia ja lisäyksiä harvesterin hallintaohjelmiin. Myös tämä tavoite saavutettiin onnistuneesti. Tämä dokumentti toimii ohjeena, kun suunnitellaan käyttöliittymää toteutettavaksi CX-One-ohjelmistolla.

Työn avulla saavutetut tiedot ja taidot monipuolistavat osaamistani, koska niitä voidaan hyödyntää jatkossakin esimerkiksi teollisuusautomaatio tai ajoneuvoalalla, jossa ne ovat laajalti käytössä.

Työ oli haastavampi kuin aloittaessa osasi odottaa, ja se vaati laajaa tutustumista muun muassa CAN-väylätekniikkaan ja siihen liittyviin standardeihin sekä ohjelmoitavan logiikan ohjelmointiin, jotta työstä voitiin suoriutua mallikkaasti. Työn suorittamista helpotti yhteistyö sekä ProfiPro Oy:n edustajan että toisien samaan aihepiiriin tehtävien opinnäytetöiden tekijöiden kanssa. Koulun henkilökunta oli myös suureksi avuksi.

LÄHTEET

[1] profipro.fi, Yritys. Verkkosivu. Viitattu 20.3.2016.

<http://profipro.fi/index.php/fi/yritys>

[2] ni.com, kuva. Verkkosivu. Viitattu 20.3.2016. <http://www.ni.com/white-paper/2732/en/>

[3] Saha 2005, CAN-väylä. Verkkodokumentti. Viitattu 18.3.2016.

<http://www.canopen.fi/artikkelit/CAN.pdf>

[4] kvaser.com, kuva. Verkkosivu. Viitattu 23.3.2016.

<https://www.kvaser.com/can-protocol-tutorial/>

[5] elmomc.com, kuva. Verkkosivu. Viitattu 20.3.2016.

http://www.elmomc.com/members/NetHelp/ImagesExt/image381_0.png

[6] kvaser.com, SAE J1939 Introduction. Verkkosivu. Viitattu 20.3.2016.

<https://www.kvaser.com/about-can/higher-layer-protocols/j1939-introduction/>

[7] scielo.br, kuva. Verkkosivu. Viitattu 3.4.2016. http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-43662010001100015

[8] kvaser.com, SAE J1939 Standards overview. Verkkosivu. Viitattu 20.3.2016.

<https://www.kvaser.com/about-can/higher-layer-protocols/j1939-standards-overview/>

[9] blog.frogslayer.com, kuva. Verkkosivu. Viitattu 30.3.2016.

<https://blog.frogslayer.com/a-primer-on-can/>

[10] CJ2M datalehti. Verkkojulkaisu. Viitattu 1.4.2016.

https://industrial.omron.fi/fi/products/catalogue/automation_systems/programmable_logic_controllers/modular_plc_series/cj2m/default.html

- [11] industrial.omron.fi, Automaatio-ohjelmisto. Verkkosivu. Viitattu 3.4.2016.
https://industrial.omron.fi/fi/products/catalogue/automation_systems/software/programming/cx-one/default.html
- [12] peak-system.com, PCAN-USB -laitteen ohjelmisto ja kuva. Verkkosivu. Viitattu 20.4.2016. <http://www.peak-system.com/PCAN-USB.199.0.html?L=1>
- [13] Sokka, J. 2009. CAN-väylätekniikka ajoneuvokäytössä. Theseus.fi. Viitattu 20.4.2016. <http://urn.fi/URN:NBN:fi:amk-201003063134>