OULUN AMMATTIKORKEAKOULU

Kimmo Suorsa

FPGA-EVALUOINTILEVYN SOVELTUVUUS LIIKKEENOHJAUKSEEN

FPGA-EVALUOINTILEVYN SOVELTUVUUS LIIKKEENOHJAUKSEEN

Kimmo Suorsa
Opinnäytetyö
Kevät 2014
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, langattomat laitteet

Tekijä: Kimmo Suorsa
Opinnäytetyön nimi: FPGA-evaluointilevyn soveltuvuus liikkeenohjaukseen
Työn ohjaajat: Pekka Raudaskoski, Hannu Siipola, Ensio Sieppi
Työn valmistumislukukausi ja -vuosi: Kevät 2014
Sivumäärä: 29 + 1 liite

Opinnäytetyön aiheena oli tutkia National Instrumentsin LabVIEW RIO -evaluointityökalun soveltuvuutta liikkeenohjaukseen. Tavoitteena oli tehdä LabVIEW'ta käyttäen toimiva ohjelma moottorinohjaamiseen ja optiona oli tutkia, miten siihen saisi integroitua erilaisia mittauksia. Työtä voidaan soveltaa JOT Automationin kehittämien laitteiden liikkeenohjaukseen.

Työn ensimmäinen osa suoritettiin etsimällä tietoa internet-lähteistä sekä keskustelemalla parametreista ja toteutuskeinoista työn ohjaajan kanssa. Työ muotoutui lopulta tutkimuksen ja keskusteluiden perusteella. Työ laajennettiin koskemaan muutamia työnantajan tarvitsemia mittauksia, jotta työn laajuus olisi tarpeeksi suuri. Tutkimisen jälkeen tehtiin LabVIEW'n avulla ohjelma, jolla selvitettiin moottorinohjauskomennot, ja lopuksi tehtiin ohjelmat liikkeenohjaukseen ja mittauksiin.

Työn tulokset esiteltiin työnantajalle, joka päättää, kannattaako kehitystä jatkaa kyseiseen suuntaan vai hyödynnetäänkö työstä vain tiettyjä osia. Työ sisältää liikkeenohjausohjelman evaluointityökalulle ja sekä jännitteen tarkkailu- ja FFT-ohjelman, joiden kehittäminen eteenpäin National Instrumentsin työkaluilla on helposti toteutettavissa.

Asiasanat: FPGA, LabVIEW, liikkeenohjaus, evaluointityökalu

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, option of Wireless Devices

---

Author: Kimmo Suorsa
Title of thesis: The suitability of motion control for FPGA-evaluation board
Supervisors: Pekka Raudaskoski, Hannu Siipola, Ensio Sieppi
Term and year when the thesis was submitted: Spring 2014
Pages: 29 + 1 appendix

The objective of this thesis is to research the suitability of National Instruments LabVIEW RIO Evaluation Kit for motion control. Goal was to make a functioning program with LabVIEW to control the movements of the motor and option was to research how to integrate various measurements to it. The work can be applied for motion control in various equipments developed by JOT Automation.

The first part of the thesis was executed by researching online information sources and by discussing about the parameters and methodology for the implementation with supervisor. The thesis eventually took shape based on the research and discussions. The thesis was expanded to include few measurements, so that the scope of the thesis was big enough. After the research LabVIEW-program was made in order to find out motor control commands. Finally LabVIEW-programs for motor control and measurements were made.

The result of the thesis was presented to the employer, who will make the decision, if it's worthwhile to continue development to this direction or to use only parts of the work. Thesis includes motion control program for evaluation kit and voltage monitoring- and FFT-programs, the development of which forward is easy to implement with National Instruments tools.

Keywords: FPGA, LabVIEW, Motion control, evaluation kit

ALKULAUSE

Tämä opinnäytetyö käsittelee liikkeenohjaamisen toteuttamista National Instrumentsin LabVIEW RIO -evaluointityökalun avulla. Työ tarjosi minulle mahdollisuuden näyttää osaamistani työelämässä ja laajentaa tietoisuuttani aiheesta.

Haluan kiittää Pekka Raudaskoskea, Hannu Siipolaa ja JOT Automationia, jotka antoivat mahdollisuuden tehdä opinnäytetyö JOT Automationille, sekä kaikkia JOT:n työntekijöitä, jotka tarjosivat apua ja mieluisan työympäristön.

Kiitos myös ohjaavalle opettajalle Ensio Siepille palautteista ja ohjeista työn aikana.

22.5.2014 Oulussa

Kimmo Suorsa

SISÄLLYS

SANASTO

CAN	Controller Area Network. CAN-väylä
FFT	Fast Fourier Transform. Fourier'n muunnos
FPGA	Field-Programmable Gate Array. Ohjelmoitava porttimatriisi
PLD	Programmable Logic Devices. Ohjelmoitava logiikkapiiri
PROM	Programmable Read Only Memory. Ohjelmoitava lukumuisti
sbRIO	Single Board RIO. National Instrumentsin single board RIO-levy
VI	Virtual Instrument. LabVIEW’n tuottama ohjelma

1 JOHDANTO

JOT Automation perustettiin vuonna 1988 tukemaan matkapuhelinalan kasvua ja tuomaan matkapuhelimia kaikkien saataville. JOT Automation toimii neljällä mantereella ja työllistää yli 300 ihmistä 13 eri maassa. Sen asiakasluettelo sisältää yrityksiä, joilla on jopa satoja tuotantolaitoksia ympäri maailmaa. Työn lopputulosta voidaan soveltaa JOT Automationin kehittämien laitteiden liikkeenohjaukseen, kuten JOT Automationin matkapuhelimien testauslaitteissa. (1.)

Opinnäytetyö työn aiheeksi valittiin FPGA-evaluointilevyn soveltuvuus liikkeenohjaukseen. Työn tavoitteena oli selvittää, voiko liikkeenohjausta toteuttaa National Instrumentsin LabVIEW RIO -evaluointityökalun avulla, ja jos voi niin miten. Ideana työlle oli, että jos liikkeenohjauksen saa toteutettua evaluointityökalun avulla, JOT Automation voisi hyödyntää tulevien tuotteiden suunnittelussa LabVIEW'n laajoja kirjastoja.

Liikkeenohjaus oli työn päätavoite, mutta jos aikaa jäisi jäljelle, olisi tavoitteina tutkia, miten liikkeenohjaukseen saisi yhdistettyä erilaisia mittauksia, joita JOT Automation tarvitsee tuotteissaan. Näin varmistettiin, että opinnäytetyö on laajuudeltaan ja vaatimuksiltaan sopiva. JOT Automationilla on olemassa liikkeenohjausjärjestelmä, mutta siinä ei hyödynnetä vielä FPGA:ta.

2 FPGA-PIIRI

2.1 Historia ja nykypäivä

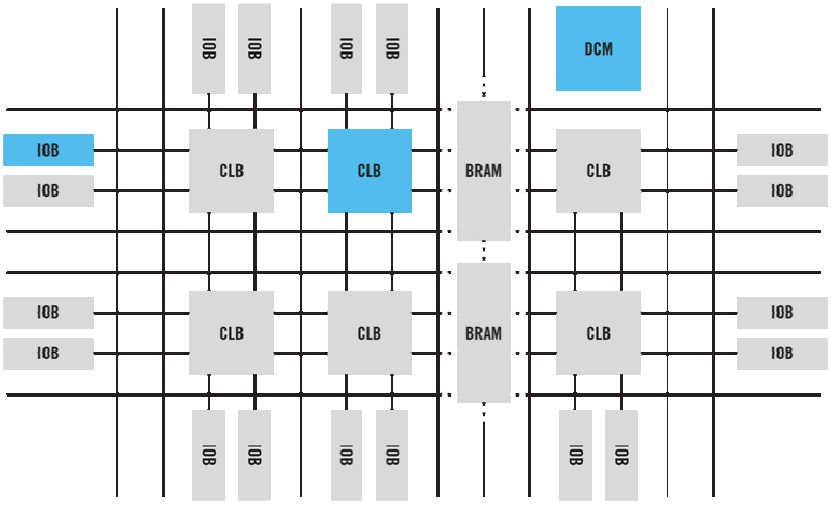
FPGA:t (Field-programmable gate array) kehittyivät PROM- ja PLD-piireistä. Molemmat voitiin ohjelmoida tehtaalla tai kentällä, mutta molemmissa ohjelmoitava logiikka oli kytketty kiinteästi logiikkaporttien välille. Ensimmäinen ohjelmoitava logiikkalaite oli PROM-muisti (Programmable Read Only Memory). PROM on haihtumatonta muistia, joka voidaan ohjelmoida tehtaalla tai käyttäjän toimesta kentällä (FPROM). Seuraava askel kehityksessä kohti FPGA-piirejä oli PLD-piirit (Programmable Logic Devices). On olemassa useita erityyppisiä PLD-piirejä, mutta yleisin toteuttaa joukon kiinteitä loogisia TAI-portteja, joita edeltää joukko loogisia JA-portteja. Kuten PROM, PLD-piirejä voidaan ohjelmoida tehtaalla tai käyttäjän toimesta kentällä. (2.)

Xillinx toi vuonna 1985 ensimmäisen kaupallisen FPGA-piirin markkinoille. Xillinxin FPGA perustui yhtiön omaan patentoituun LCA-teknologiaan (Logic Cell Array). Yhtiön tarjoama paketti koostui ohjelmoitavasta sirusta ja ohjelmistopaketista, jonka avulla siru voitaisiin ohjelmoida vastaamaan tarkkoja vaatimuksia. XC2064-piirissä oli ohjelmoitavat portit ja ohjelmoitavat yhteydet porttien välillä. (3.)

Koska FPGA on ohjelmoitavuutensa ansiosta ideaalinen tuote monille markkinoille, käytetään sitä nykypäivänä useilla eri sovellusalueilla avaruusteknologiasta lääketieteelliseen ja kaikkialla siltä väliltä (4).

2.2 FPGA:n rakenne

FPGA:t (kuva 1) ovat uudelleen ohjelmoitavia puolijohdekomponentteja, jotka koostuvat toisiinsa ohjelmoitavilla yhteyksillä kytketyistä konfiguroitavista logiikkaelementeistä rakentuvan matriisin ympärille (4).



KUVA 1. FPGA:n rakenne (4.)

FPGA:n logiikkaelementit voidaan ohjelmoida toteuttamaan monimutkaisia ja monimuotoisia funktioita tai yksinkertaisia logiikkaportteja. Nykyaikaisissa FPGA-piireissä logiikkaporttien määrä on jo miljoonaluokkaa. Useimmat FPGA-piirit sisältävät myös ei-ohjelmoitavia osia, kuten muistia, suodattimia ja erilaisia yksinkertaisia aritmeettisia operaatioita, joiden avulla saadaan parannettua piirin suorituskykyä ja pienennettyä tehonkulutusta. (4; 5.)

FPGA:ta käytetään laajalti useilla sovellusalueilla, koska FPGA:t ovat helposti uudelleen ohjelmoitavissa ja niillä voidaan teoriassa toteuttaa mikä tahansa digitaalipiirin toiminnallisuus. Tämän ansiosta FPGA:n avulla tuotteet pystytään tekemään nopeasti ja alhaisella kustannuksella. (4; 6.)

3 CAN-VÄYLÄ

3.1 Historia

CAN-väylä (Controller Area Network) on automaatioväylä, joka kehitettiin alun perin autoihin. Sen avulla pystytään mm. välittämään mittaus- ja ohjaustietoja laitteiden välillä. Väylä kehitettiin, jotta laitteet voivat kommunikoida toistensa kanssa ilman isäntäkonetta. (7.)

CAN-väylän kehittämisen aloitti Robert Bosch GmbH jo vuonna 1983 ja se esiteltiin ensimmäisen kerran vuonna 1986. Ensimmäiset CAN-ohjainsirut tulivat markkinoille vuonna 1987, 1990-luvun alussa Boschin CAN 2.0 version spesifikaatiot toimitettiin kansainväliseen standardointiin ja ISO-standardi 11898 CAN-protokollalle julkaistiin marraskuussa 1993. (7.)

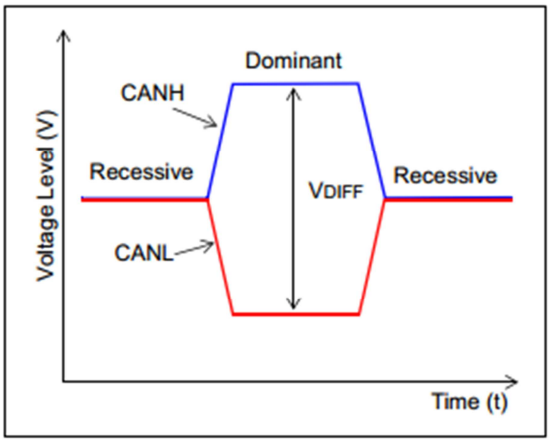
Nykypäivänä CAN-protokolla on jokaisessa uudessa autossa ja sen lisäksi sitä käytetään myös muissa kulkuneuvoissa (mm. junat ja laivat) sekä teollisissa ohjauslaitteissa (7). CAN-protokollan etuna on, että se on vakaa protokolla. Se on erittäin vikasietoinen ja luotettava, sekä tämän lisäksi se on matalakustannuksinen toteuttaa. (12.)

3.2 Fyysinen kerros

CAN-väylällä on useita fyysisiä kerroksia, kuten esimerkiksi High-Speed CAN ja Low-Speed/Fault-Tolerant CAN. Nämä fyysiset kerrokset luokittelevat tiettyjä piirteitä CAN-verkossa, kuten sähkötasot, signalointimalleja, kaapeli-impedanssin ja maksimibaudinopeudet. (8.)

CAN-väylälle on määritelty useita eri tiedonsiirtonopeuksia, korkeimmillaan tiedonsiirtonopeus on 1 Mbit/s ja matalimmillaan 10 kbit/s. Kaikkien moduulien on tuettava 20 kbit/s:n tiedonsiirtonopeutta (9). Kaapelin pituus riippuu käytettävästä tiedonsiirtonopeudesta. Käytettäessä korkeinta 1 Mbit/s tiedonsiirtonopeutta kaapelin maksimipituus on 40 m ja kaikki sitä pidemmät väylät toteutetaan laskemalla tiedonsiirtonopeutta. Hitaimmalla 10 kbit/s tiedonsiirtonopeudella kaapelin maksimipituus on 6 km. (10; 12.)

CAN määrittää kaksi loogista tilaa: peittyvä ja dominantti (kuva 2). ISO-11898-standardi määrittää jännite-eron esittämään peittyvää ja dominanttia tilaa. Dominantti vastaa loogista tilaa 0 ja peittyvä loogista tilaa 1 (10.).



KUVA 2. CAN-väylän tilat (10.)

3.3 CAN-kehykset

CAN-protokollassa on neljä erilaista kehystä: datakehys, remote-kehys, error-kehys ja overload-kehys. Datakehys hoitaa tiedonvälityksen, remote-kehys tiedustelun, error-kehys tiedonsiirtovirheistä ilmoittamisen ja viimeinen on overload-kehys, jota käytetään luomaan viive, kun joku solmuista ei ehdi käsittelemään sille tulevaa dataa. (11.)

3.3.1 Normaali datakehys

CAN-protokolla tukee kahta datakehysformaattia, jotka on määritetty Boschin version 2.0 spesifikaatioissa. Olennainen ero näiden kahden formaatin välillä on ID-kentän pituus. Normaalissa datakehysformaatissa (2.0A) ID-kentän pituus on 11 bittiä, kun taas laajennetussa datakehysformaatissa (2.0B) ID-kentän pituus on 29 bittiä. (12.)

Jokaisen datakehyksen (kuva 3) alussa on 1-bittinen viestikehyksen aloituskenttä (SOF). Se näyttää viestin alun ja sitä käytetään solmujen synkronoimiseen väylällä. Sen jälkeen on 11-bittinen ID eli tunnistekenttä, joka määrittää viestin prioriteetin väylällä. Jos useat solmut yrittävät lähettää viestiä väylällä yhtä aikaa, solmu, jolla on suurin prioriteetti, lähettää viestin automaattisesti.

Tunnistekentän jälkeen on RTR-kenttä, joka määrittää, onko kehys data- vai remote-kehys. RTR-bitti on dominantti, kun se on datakehys, ja peittyvä, kun se on remote-kehys. Seuraavana on IDE-bitti, joka määrää viestikehyksen tyypin. r0 on varattu bitti, joka ei ole tällä hetkellä käytössä, vaan on varattu tulevaisuutta varten. DLC-kenttä puolestaan kuvaa datakehyksen sisältämien datatavujen määrän, joka voi olla 0–8 tavua. Datakenttä sisältää itse viestin, joka lähetetään, ja se sisältää DLC-kentän ilmaiseman määrän datatavuja. 16-bittinen CRC-kenttä sisältää tarkistussumman, jota käytetään virheen tarkastukseen. Tarkistussumma lasketaan aloituskentän jälkeisistä kentistä tarkistussummaan asti. ACK-kenttä on viestin kuittauskenttä. Tämän jälkeen on vielä EOF, joka on datakehyksen lopetuskenttä. Viimeinen kenttä on IFS, joka määrittää peräkkäiset viestit erottavien bittien vähimmäismäärän. (11; 12.)

KUVA 3. Normaali datakehys (11.)

3.3.2 Laajennettu datakehys

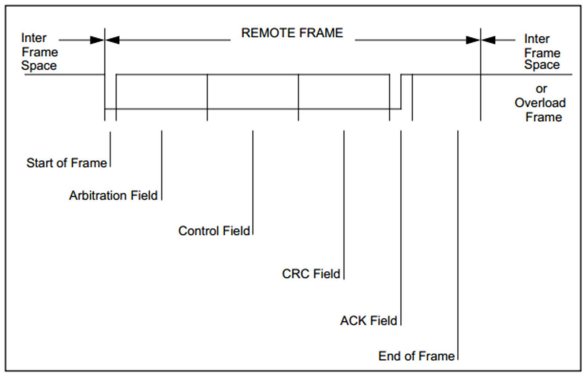
Laajennettu datakehys (kuva 4) on suurilta osin samanlainen kuin normaali datakehys. Siihen on lisätty vain pari kenttää. SRR-kenttä lähetetään aina peitettynä, jotta jos normaali ja laajennettu datakehys lähetetään samaan aikaan samalla pohja-ID:llä (ensimmäiset 11-bittiä), normaalilla datakehyksellä on ensisijainen lähetysoikeus. r1-kenttä on samanlainen kuin r0-kenttäja, ja sekin on varattu tulevaisuutta varten. (11.)

KUVA 4. Laajennettu datakehys (11.)

3.3.3 Remote-kehys

Remote-kehyksellä (kuva 5) ja datakehyksellä on kaksi eroa: remote-kehyksessä ei ole ollenkaan datakenttää ja se on merkitty remote-kehykseksi

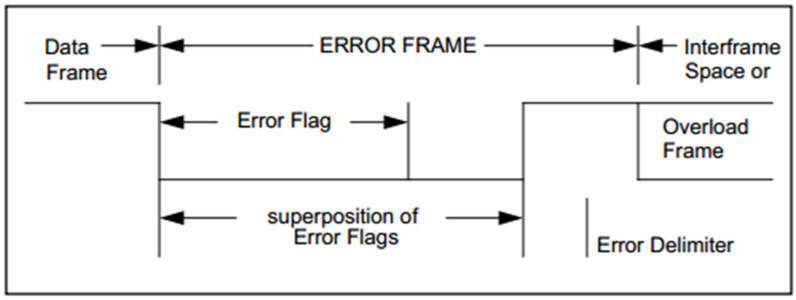
eli RTR-bitti on peittyvä. Remote-kehykstä käyttäen viestiä odottava verkon solmu voi käynnistää viestin lähetyksen lähettämällä remote-kehyksen, jonka tunniste on sama kuin viestillä, jonka se haluaa. (11.)



KUVA 5. Remote-kehys (13)

3.3.4 Error-kehys

Error-kehys eli virhekehys (kuva 6) lähetetään, kun viestissä huomataan virhe ja sen lähettäminen aiheuttaa sen, että kaikki muutkin verkon solmut huomaavat virheen. Tällöin lähettäjä yrittää automaattisesti uudelleenlähetystä. Error-kehys koostuu virhelipusta ja virhelipun erottimesta. Virhelippu koostuu kuudesta määräävästä bitistä ja virhelipun erotin koostuu kahdeksasta peittyvästä bitistä. (11.)

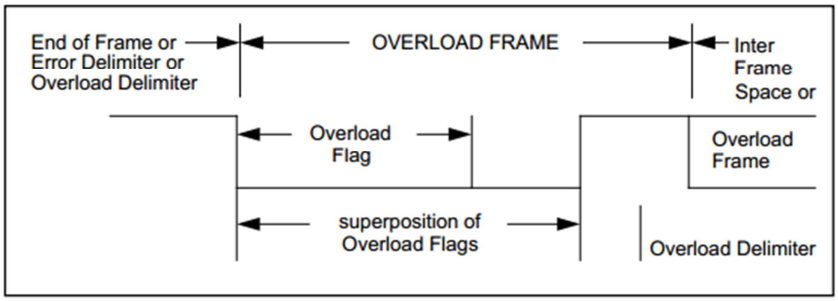


KUVA 6. Virhekehys (13.)

3.3.5 Overload-kehys

Overload-kehys eli ylikuormituskehys (kuva 7) on samantyyppinen kuin error-kehys. Siinä on virhekehyksen tavoin kaksi kenttää: ylikuormituslippu, joka

koostuu kuudesta määräävästä bitistä, ja ylikuormituserotin, joka koostuu kahdeksasta peittyvästä bitistä. (13.)



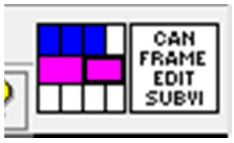
KUVA 7. Ylikuormituskehys (13.)

4 LABVIEW

LabVIEW on National Instrumentsin kehittämä ohjelmointiympäristö visuaaliselle ohjelmointikielelle. LabVIEW'ta käytetään yleisesti mittauksiin, laiteohjaukseen ja automaatioon. Sen ensimmäinen versio LabVIEW 1.0 julkaistiin lokakuussa 1986 Applen Macintoshille. (14; 15.)

LabVIEW'n käyttämää ohjelmointikieltä kutsutaan G-kieleksi, joka on datavuo-ohjelmointikieli. Datavuo-ohjelmointikielessä koodi suoritetaan vaiheittain eli jokainen koodin sisältämä komponentti suoritetaan, kun komponentin kaikki tarvittavat sisääntulot ovat saaneet tarvittavat tiedot. Tämän jälkeen komponentti tekee sille ohjelmoidun toimintonsa ja tulos siirretään komponentin ulostulosta seuraavalle komponentille. Toisin sanoen ohjelman suoritus määrittyy lohkokaavion rakenteen mukaan. (14.)

LabVIEW'lla luotuja ohjelmia kutsutaan Virtual Instrumenteiksi eli VI:ksi. Jokaisella LabVIEW VI:llä on kolme osaa: lohkokaavio (Block Diagram), etupaneeli (Front Panel) ja liitinruutu (Connector Pane). Ohjelman ohjelmointi tapahtuu pääosin lohkokaaviossa ja se sisältää myös lähdekoodin. Etupaneeli toimii puolestaan ohjelman käyttöliittymänä ja se sisältää kaikki ohjelmaan luodut ohjauskomponentit. Ohjauskomponentteihin syötetty data välitetään ohjelmaa ajettaessa lohkokaavioon. Liitinruutua (kuva 8), joka sijaitsee etupaneelin oikeassa yläkulmassa, käytetään määrittämään ohjelman lähdöt ja tulot, muita sitä kutsuvia ohjelmia varten. (14.)

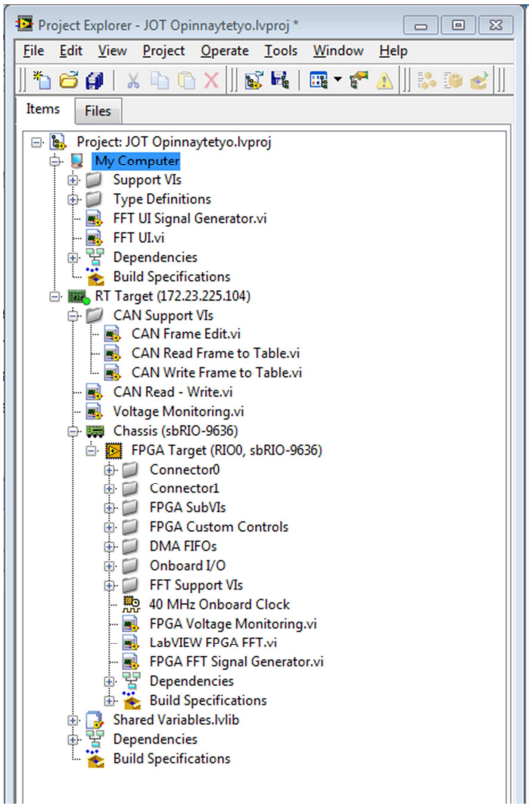


KUVA 8. CAN Frame Edit.vi:n liitinruutu, johon on määritetty ohjelman lähdöt ja tulot.

4.1 LabVIEW-projekti ja datatiedostot

LabVIEW'lla luodut ohjelmat (VI:t) tallennetaan sille määrättyyn projektiin (kuva 9). Projektin alla on hakemistoja, jonne tiedostot tallennetaan tiettyyn paik-

kaan riippuen siitä, mikä sitä käyttää. Kaikkien FPGA:lle ladattavien ohjelmien täytyy olla FPGA Target-hakemiston alla ja pääohjelmien joko RT Target- tai My Computer-hakemiston alla. Kaikkien aliohjelmien, joita ohjelma käyttää, täytyy olla myös saman hakemiston alla.



KUVA 9. LabVIEW-projekti.

Mikäli pääohjelma käyttää FPGA:ta, täytyy siihen määritelty FPGA-ohjelma ajaa ensin sbRIOlle, ennen kuin pääohjelmaa voi käyttää. LabVIEW'lla pystyy ajamaan vain yhtä FPGA:ta käyttävää ohjelmaa kerrallaan.

4.2 LabVIEW FPGA -moduuli

NI LabVIEW FPGA -moduuli laajentaa LabVIEW-kehitysalustan ja National Instrumentsin I/O RIO-laitteiston FPGA-piirien toimintaa. FPGA-moduulin avulla voidaan luoda mittaus- ja säätölaitteita ilman HDL-kieliä tai piiritason suunnittelua. (16.)

4.3 LabVIEW Real-Time -moduuli

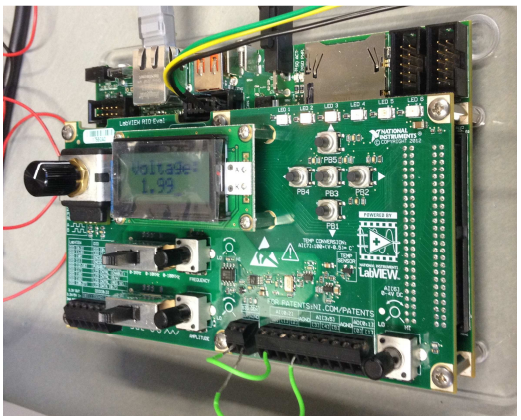
LabVIEW Real-Time -moduuli on lisäosa LabVIEW-kehitysympäristöön. Real-Time -moduulin avulla ohjelmia pystytään lataamaan ja suorittamaan sulautetuilla laitteilla. Se sisältää real-time käyttöjärjestelmän (RTOS), joka pyörii National Instrumentsin sulautetuilla laitteilla (17.). Suurin osa LabVIEW-sovelluksista pyörii yleiskäyttöisellä käyttöjärjestelmällä, kuten Windows tai Mac OS, mutta jotkin sovellukset vaativat deterministisen reaaliaikaisen suorituksen, mitä yleiskäyttöiset käyttöjärjestelmät eivät voi taata (18.).

4.4 NI LabVIEW RIO Evaluation Kit

Työssä käytetty National Instrumentsin LabVIEW RIO -evaluointityökalu (kuva 10) sisältää 90 päivän laajennetun kokeiluversion LabVIEW'sta, LabVIEW FPGA- ja LabVIEW Real-Time -moduulit, tytärlevyn sbRIOlle sekä NI sbRIO-9636 -alustan (19.).

National Instrumentsin sbRIO-9636 -levyllä on reaaliaikainen prosessori, Xillinxin Spartan-6 LX45 FPGA-piiri, 16 yksipuoleista ja 8 differentiaalista 16-bittistä analogiatulokanavaa, neljä 16-bittistä analogiatulokanavaa ja 28 digitaalista I/O linjaa. sbRIOn käyttölämpötila on –40–85 ℃ (20.).

Evaluointityökalu on erinomainen työkalu nopeaan FPGA-prototyypin tekemiseen, sillä siinä on kaikki tarvittavat osat ja sen avulla prototyypin teko onnistuu nopeasti.



KUVA 10. sbRIO Evaluation Kit.

5 LIIKKEEN OHJAAMINEN SBRIOLLA LABVIEW'N KAUTTA

5.1 Lähtökohdat

Opinnäytetyön tavoitteena oli tutkia, onnistuuko moottorien ohjaaminen LabVIEW'n kautta. Ensimmäinen vaihe oli tutkia sbRIOn tukemat protokollat, joilla moottoria voisi liikuttaa. Toinen vaihe oli tehdä LabVIEW-ohjelma, jolla moottorin ohjaus voidaan toteuttaa, ja lopuksi tutkia, millaisia erilaisia mittauksia FPGA:n avulla voisi toteuttaa.

5.2 Toteutusvaihe

Tutkimuksen aikana selvisi, että moottoriohjaus sbRIOlla voidaan toteuttaa ainoastaan CAN-protokollaa käyttäen. sbRIO-kehitysalusta käyttää NI-Embedded CAN -ajureita, joka on pelkistetty versio NI-CAN-ajureista, eikä siinä ole NI-CAN-ajurien kehittyneimpiä ominaisuuksia, kuten esimerkiksi suodatusta.

Tutkimisen jälkeen rakennettiin välikaapeli, jonka avulla voisi tutkia sbRIOlla JOT Automationin nykyisen moottorinohjauskortin ja -ohjelmiston välistä liikennettä ja selvittää, millaisilla CAN-viesteillä moottoria ohjataan. JOT Automationilla on käytössä oma CAN-protokolla, mutta sitä ei käsitellä työssä, koska sen tiedot ovat luottamuksellisia.

Seuraavaksi tehtiin ohjelma, jolla pystyi lukemaan moottoriohjauskortin ja ohjelmiston välistä liikennettä sbRIOlla. Kun moottorin ohjauskäskyt olivat selvillä, alkoi moottorinohjausohjelman tekeminen LabVIEW'lla. Tarkoituksena oli saada ohjelmaan samat toiminnallisuudet kuin JOT Automationin aiemmassa moottorinohjausohjelmassa ja kehittää niitä eteenpäin. Käyttöliittymän tekeminen ohjelmaan tapahtui samanaikaisesti, sillä LabVIEW'ssa lohkokaavio ja etupaneeli ovat yhteydessä toisiinsa.

Kun ohjelma oli valmis, sbRIOn CAN-väylä kytkettiin moottorinohjauskorttiin, johon moottori oli kytkettynä. Moottorin käyttämiseen CAN-väylän kautta tarvittiin väylältä CAN_H-, CAN_L- ja GND-signaalit. Käyttöliittymään tehtiin eri-

laisia ohjaimia, joilla pystytään vaihtamaan ja testaamaan ideaaliset parametrit moottorinohjaukselle.

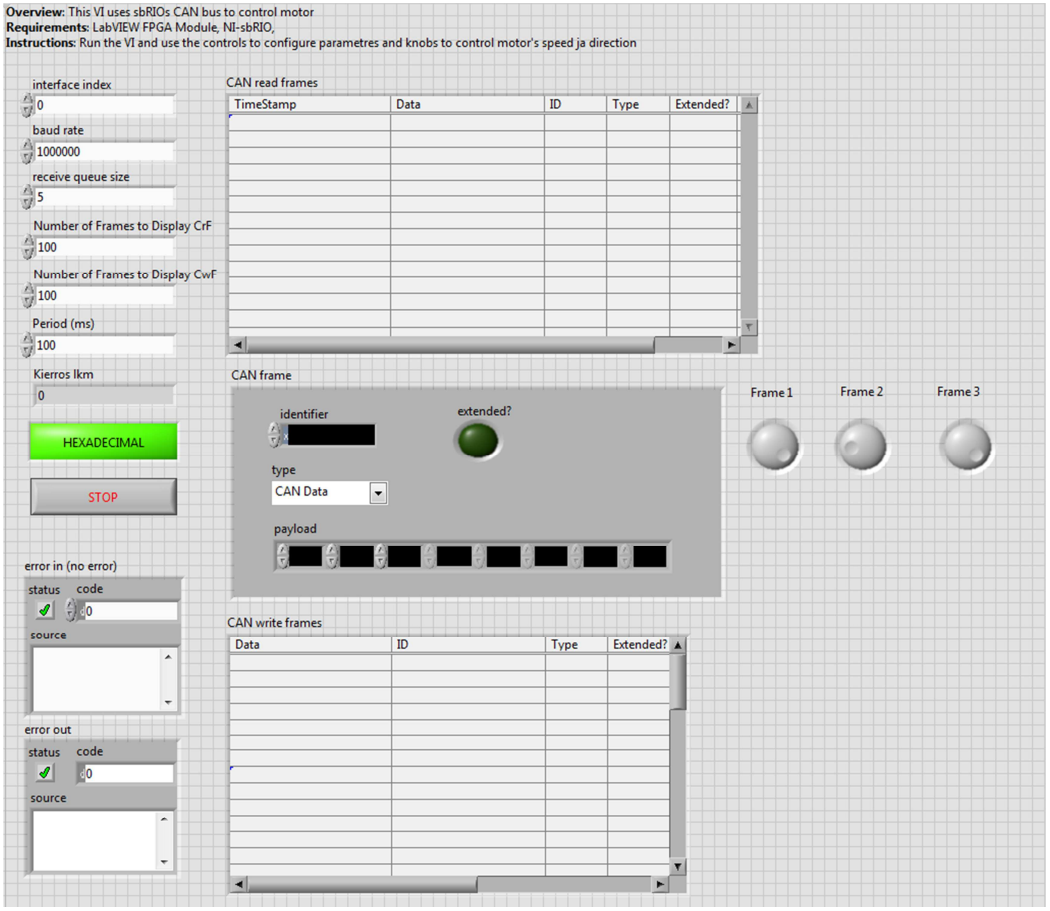
Käytettävissä olevan ajan puitteissa tehtiin lisäksi vielä kaksi ohjelmaa erilaisia mittauksia varten. Ensimmäinen ohjelma tarkkaili sbRIOn analogiseen lähtöön syötettyä jännitettä, ja jännitteen noustessa yli ennalta määrätyn kynnysarvon se sammutti molemmat ohjelmat globaalin muuttujan avulla. Ohjelma tulostaa jännitearvon sbRIOn tytärlevyn LCD-näytölle, josta sitä voi tarkkailla. Jännitteen tarkkailuohjelma tehtiin, koska sen toiminta periaate on sama kuin JOT Automationin tulevaisuudessa tarvitsemissa mittauksissa.

Toinen mittauksia varten tehty ohjelma tekee FFT-muunnoksen signaaligeneraattorista syötettävälle signaalille. Signaalin muotoa ja taajuutta ohjataan tytärlevyllä olevasta signaaligeneraattorista käsin. Molemmat mittausohjelmat tehtiin JOT Automationin tuleviin tarpeisiin.

5.3 CAN-ohjelman toiminta

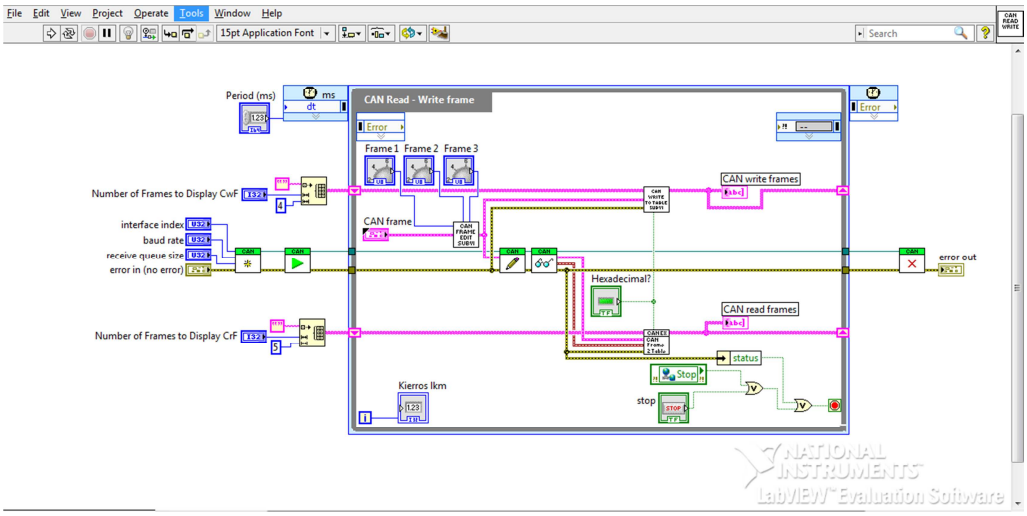
Tarkemmat tiedot CAN-ohjelmasta sekä muista ohjelmista löytyvät liitteestä 1, joka on saatavilla vain JOT Automation Oy:n työntekijöille. Ennen ohjelman suorittamista sille annetaan alkutiedot, joilla se toimii: Baud Rate, interface index, receive queue size, period ja number to frames display. Tämän jälkeen ohjelman voi suorittaa.

Moottoria ohjataan käyttöliittymässä (kuva 11) olevilla nupeilla, joista ensimmäinen määrittää moottorin pyörimissuunnan ja kaksi seuraavaa nopeuden. Ohjelma myös tulostaa lähetyt ja vastaanotetut CAN-viestit taulukoihin, joiden avulla voi seurata ohjelman toimintaa. CAN-viestit voidaan tulostaa taulukkoon joko heksadesimaalina tai desimaalina. Lähetyt CAN-viestit ohjaavat moottoria ja vastaanotetut CAN-viestit sisältävät moottorin lähettämät paikkatiedot.



KUVA 11. CAN-ohjelman käyttöliittymä.

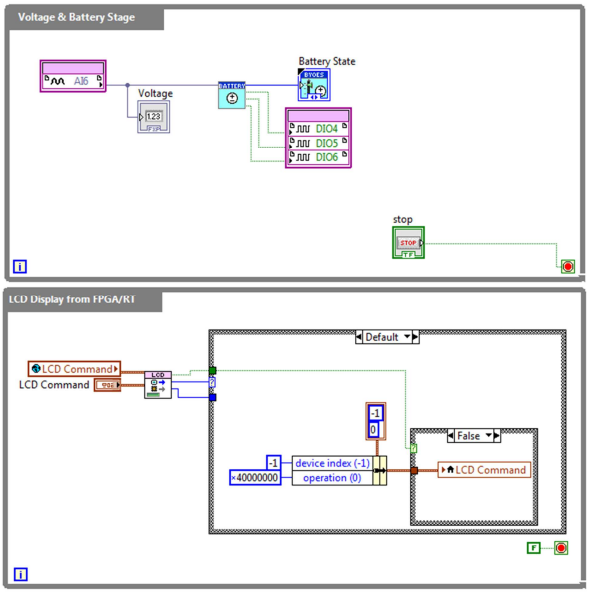
Lohkokaavion (kuva 12) puolella näkyy ohjelman toiminnallisuus. Ohjelma etenee ajettaessa vasemmalta oikealle. Alussa määritetään alkuarvot ja valmistetaan taulukot. Sen jälkeen ohjelma menee silmukkaan, jossa CAN-viestien lähetys ja lukeminen tapahtuvat. Ohjelma pyörii silmukan sisällä sille alussa määrätyin jaksonajoin, kunnes siinä tapahtuu virhe tai se pysäytetään. Jokaisella kierroksella ohjelma lähettää ja lukee yhden CAN-viestin ja sijoittaa ne taulukoihin. CAN-kehyksen muokkaus ja taulukoiden tulostaminen tapahtuu erillisissä aliohjelmissa.



KUVA 12. Lohkokaavio CAN Read – Write.vi:lle.

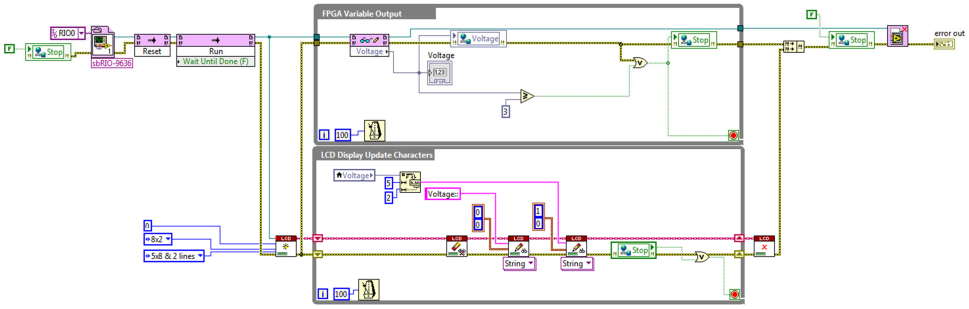
5.4 Jännitteen tarkkailuohjelman toiminta

Jännitteen tarkkailuohjelmaa käytettäessä täytyy ensin ajaa FPGA VI sbRIO-levylle, jotta isäntä-VI toimii. FPGA VI:ssä (kuva 13) määritettään analoginen lähtö ohjelmalle, jota se tarkkailee. Tämän lisäksi siinä alustetaan LCD-näyttö valmiiksi käyttöä varten.



KUVA 13. FPGA Voltage Monitoring.vi -ohjelmassa määritettään analogisen signaalin tulo ja alustetaan LCD-näyttö.

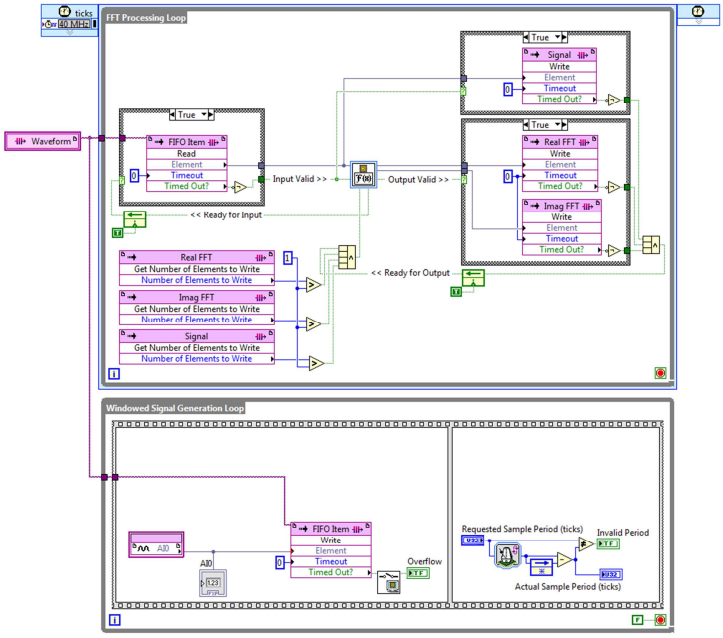
Tämän jälkeen ajetaan isäntä-VI (kuva 14), jonka alussa määritetään FPGA VI, jota se käyttää. Isäntäohjelmassa tehdään jännitteen vertailu kynnysarvoon ja jännitteen tulostus tytärkortin LCD-näytölle. Kun jännitteen arvo ylittää asetetun kynnysarvon, ohjelma sammuu automaattisesti.



KUVA 14. Voltage Monitoring.vi:n lohkokaavio

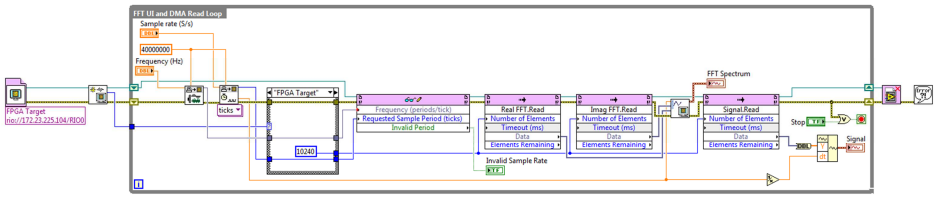
5.5 FFT-ohjelman toiminta

FFT-ohjelman käyttäminen toimii samalla lailla kuin jännitteen tarkkailu-ohjelma. Ennen isäntä-VI:n ajamista täytyy ajaa FPGA VI (kuva 15), jossa tapahtuu signaalin generointi ja FFT:n prosessointi.



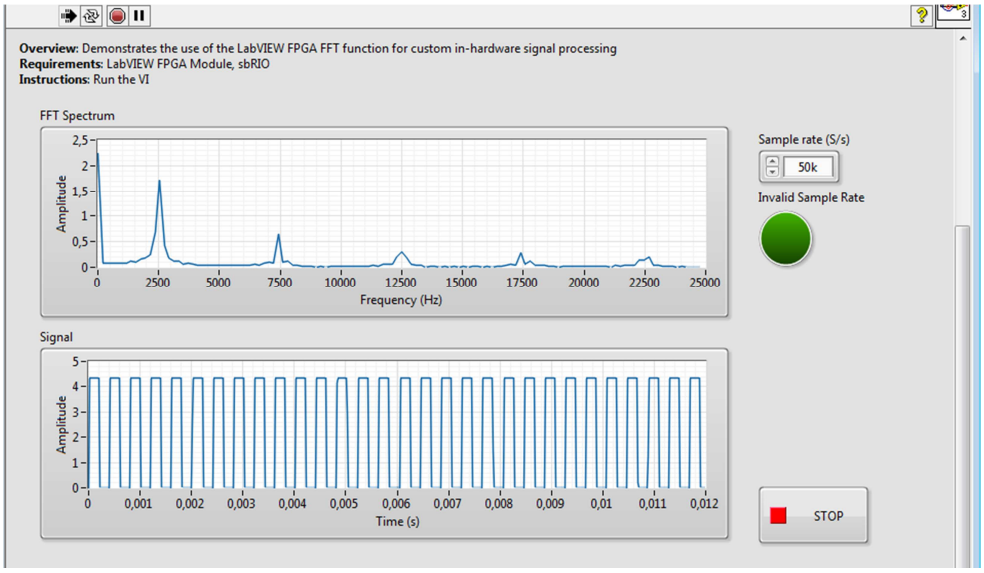
Kuva 15. FFT FPGA.vi:n lohkokaavio

Tämän jälkeen ajetaan isäntä-VI (Kuva 16), joka käsittelee haluttuun muotoon signaalin ja tulostaa sen kaavioon.



KUVA 16. FFT UI.vi:n lohkokaavio

Signaalin muotoa ja taajuutta muutetaan sbRIOn tytärlevyyn liitetystä signaaligeneraattorista, kaaviot päivittyvät reaaliaikaisesti etupaneeliin (kuva 17*).*



KUVA 17. FFT UI.vi toiminnassa 2,5 kHz:n kanttiaallolla

6 TULOKSET

Työn pääasiallinen tarkoitus oli tutkia, pystytäänkö liikkeenohjaaminen toteuttamaan National Instrumentsin LabVIEW RIO -evaluointityökalulla. Lisäksi optiona oli tutkia erilaisia mittauksia, joita evaluointi pakkauksen avulla voitaisiin toteuttaa. Kaikki työn alussa asetut tavoitteet saatiin täytettyä aikataulussa.

Moottorinohjausohjelman tärkeimpiä tuloksia oli, että saatiin selville, kuinka nopeasti silmukka voi pyöriä, niin että moottori pystyy suorittamaan sille annetut käskyt reaaliaikaisesti. Mittausohjelmien tekeminen puolestaan auttoi ymmärtämään FPGA:n mahdollisuuksia ja rajoituksia. Mittausohjelmat täyttivät moottorinohjausohjelman tavoin niille asetetut vaatimukset.

Työn kannalta mahdollisesti tärkein tulos oli, että JOT Automation sai lisätietoa FPGA:n toiminnasta ja paremman kuvan siitä, mitä sen avulla pystytään toteuttamaan. Näiden tietojen perusteella JOT Automation voi arvioida ja laskea, onko tämä kehityssuunta sellainen, johon kannattaa panostaa enemmän rahaa ja työvoimaa.

Tämän lisäksi ohjelmille tehtiin käyttöohje, joka on liitteenä työssä nimellä LabVIEW Guide for sbRIO Evaluation Kit, mutta se on saatavilla vain JOT Automation Oy:n henkilökunnalle. Käyttöohje helpottaa työtä eteenpäin jatkavan henkilön perehtymistä työhön.

7 YHTEENVETO

Työssä tutkittiin liikkeenohjaamista LabVIEW RIO -evaluointityökalulla. Työn edetessä opinnäytetyön laajennettiin koskemaan muutamia mittauksia, joita työnantaja tarvitsee. Tuloksena tein LabVIEW'lla liikkeenohjausohjelman, jännitteen tarkkailuohjelman ja FFT-ohjelman, jotka kaikki toimivat evaluointialustalla sekä käyttöohjeen ohjelmille seuraavaa käyttäjää varten.

Työn aikana jouduin aktiivisesti hakemaan tietoa minulle uusista protokollista ja tekniikoista. Itsenäisen tiedonhaun lisäksi sain myös apua ohjaavalta opettajalta sekä yrityksen työntekijöiltä. Etenkin yrityksen työntekijöiden neuvot auttoivat suurelta osin työn etenemistä.

Kohtasin muutamia evaluointialustaan ja ohjelmointiympäristöön liittyviä ongelmia työn aikana, mutta sain selvitettyä ne tutkimalla ja kollegoiden neuvojen avulla.

Olen tyytyväinen työn lopputulokseen. Onnistuin saavuttumaan työlle asetut tavoitteet ja sain hyvää kokemusta työelämästä. Työssä sain käytettyä hyväkseni itsenäisellä tiedonhaulla oppimiani tietoja sekä kollegoiden ja ohjaavan opettajan mielipiteitä ja neuvoja.

Evaluointityökalu tarjoaa useita mahdollisuuksia jatkaa kehitystä eteenpäin ja tämä työ oli vain pieni osa sen tarjoamista mahdollisuuksista. Evaluointityökalulle on jo kaavailtu uusia ominaisuuksia, jotka mahdollisesti lisätään tulevaisuudessa työhön.

LÄHTEET

1. JOT – JUST ON TIME. JOT Automation. Saatavissa: http://www.jotautomation.com/en/about-jot.html Hakupäivä 14.5.2014

2. Thakur Anshul. Field Programmable Gate Array (FPGA). Saatavissa: http://www.engineersgarage.com/articles/fpga-tutorial-basics?page=1 Hakupäivä 22.5.2014

3. Xillinx, Inc. History. Funding Universe. Saatavissa: http://www.fundinguniverse.com/company-histories/xilinx-inc-history/ Hakupäivä 9.5.2014

4. What is a FPGA?. Xillinx. Saatavissa: http://www.xilinx.com/fpga/index.htm Hakupäivä 18.3.2014

5. FPGA Fundamentals. National Instruments. Saatavissa: http://www.ni.com/white-paper/6983/en/ Hakupäivä 16.5.2014

6. Introduction to FPGA Technology: Top 5 Benefits. National Instruments. Saatavissa: http://www.ni.com/white-paper/6984/en/ Hakupäivä 20.5.2014

7. CAN History. CAN in Automation (CiA). Saatavissa: http://www.can-cia.de/index.php?id=161 Hakupäivä 12.5.2014

8. Controller Area Network (CAN) Overview. National Instruments. Saatavissa: http://sine.ni.com/np/app/main/p/ap/icomm/lang/en/pg/1/sn/n17:icomm,n21:17/fmid/2834/#toc3 Hakupäivä 16.5.2014

9. CAN Bus Description. Leroy Davis. Saatavissa: http://www.interfacebus.com/CAN-Bus-Description-Vendors-Canbus-Protocol.html Hakupäivä 16.5.2014

10. Richards Pat. A CAN Physical Layer Discussion. Microchip Technology Inc. Saatavissa: http://ww1.microchip.com/downloads/en/AppNotes/00228a.pdf hakupäivä 16.5.2014

11. Parikh, Bijal. CAN Protocol - Understanding the Controller Area Network Protocol. Saatavissa: http://www.engineersgarage.com/article/what-is-controller-area-network?page=4 hakupäivä 8.5.2014

12. Controller Area Network (CAN) Tutorial. National Instruments. Saatavissa: http://download.ni.com/pub/devzone/tut/can_tutorial.pdf Hakupäivä 3.4.2014

13. Bosch CAN Specification – Version 2.0. Robert Bosch GmbH. Saatavissa: http://esd.cs.ucr.edu/webres/can20.pdf Hakupäivä 12.5.2014

14. LabVIEW User Manual. National Instruments. Saatavissa: http://www.ni.com/pdf/manuals/320999b.pdf Hakupäivä 20.5.2014

15. Company History. National Instruments. Saatavissa: http://www.ni.com/company/our-vision/history.htm Hakupäivä 8.5.2014

16. NI LabVIEW FPGA. National Instruments. Saatavissa: http://finland.ni.com/fpga Hakupäivä 17.3.2014

17. NI LabVIEW Real-Time Module. National Instruments. Saatavissa: http://www.ni.com/labview/realtime/ Hakupäivä 8.5.2014

18. LabVIEW Real-Time Module User Manual. National Instruments. Saatavissa: http://www.ni.com/pdf/manuals/322154d.pdf Hakupäivä 13.5.2014

19. NI LabVIEW RIO Evaluation Kit. National Instruments. Saatavissa: http://sine.ni.com/nips/cds/view/p/lang/fi/nid/205721 Hakupäivä 17.3.2014

20. NI sbRIO-9636. National Instruments. Saatavissa: http://sine.ni.com/nips/cds/view/p/lang/fi/nid/210421 Hakupäivä 17.3.2014