

Metropolia Ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Pekka Kulpakko

**Ala-aseman prosessori- ja ohjelmistokehitysympäristön
kartoitus**

Opinnäytetyö 7.9.2008

Ohjaaja: technology manager Kim Malmberg
Ohjaava opettaja: yliopettaja Antti Piironen

Tekijä Otsikko	Pekka Kulpakko Ala-aseman prosessori- ja ohjelmistokehitysympäristön kartoitus
Sivumäärä Aika	57 sivua 7.9.2008
Koulutusohjelma	tietotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	technology manager Kim Malmberg yliopettaja Antti Piironen
<p>Opinnäytetyössä kartoitettiin sähköverkkojen verkostoautomaatioon soveltuvaa prosessoriteknologiaa ja ohjelmistokehitysympäristöjä. Työ tehtiin Netcontrol Oy:lle, joka on kehittämässä uutta Netcon100-nimistä ala-asemaa. Opinnäytetyön tavoitteena oli löytää kolme potentiaalista vaihtoehtoa Netcon100-ala-aseman I/O-yksikön mikrokontrolleriksi sekä testata eri ohjelmistokehitysympäristöjä.</p> <p>Netcon100 suunnitellaan IEC 61850 -standardin mukaisesti. IEC 61850 sekä ala-asemalta vaadittava pitkä elinkaari aiheuttavat mikrokontrollerille vaatimuksia, jotka rajaavat pois valtaosan markkinoilla olevista mikrokontrollereista.</p> <p>Varsinaiseen kartoitukseen valittiin lopulta mukaan kuusi eri valmistajan mikrokontrolleria. Näistä kuudesta mikrokontrollerista valittiin suunnitelman mukaisesti kolme tarkemmin tutkittavaksi ja testattavaksi. Valitut mikrokontrollerit olivat Analog Devicesin ADSP-BF536, Freescalen MCF52235 ja Atmelin AT32UC3A0512. Kyseisille mikrokontrollereille tilattiin kehitysalustat, joita käytettiin testisovelluksien tekemiseen. Testisovelluksissa kaikille kehitysalustoille tehtiin Remote Port -sovellus, joka toimii Ethernet/sarjaportti-muuntimena. Testisovelluksien tärkein tarkoitus oli kuitenkin tutustua eri ohjelmistokehitysympäristöihin.</p> <p>Opinnäytetyössä testattiin neljää eri ohjelmistokehitysympäristöä. Testatut ympäristöt olivat Analog Devicesin VisualDSP++ 5.0, Freescalen Codewarrior Development Studio, IAR:n IAR Embedded Workbench ja Atmelin AVR32 Studio 2.0. Kaikissa testatuissa ohjelmistokehitysympäristöissä on suurin piirtein samat perusominaisuudet. Suurimmat erot eri ohjelmistokehitysympäristöjen välillä ovat yleisessä käytettävyydessä ja valmistajien tarjoamissa dokumentaatioissa.</p> <p>Opinnäytetyössä saatiin rajattua markkinoilla olevasta suuresta mikrokontrollerivalikoimasta ala-asemaan soveltuvat vaihtoehdot. Opinnäytetyön kartoituksen ja Netcontrolin lisätestauksen perusteella tehtävän mikrokontrollerivalinnan onnistuminen selvinnee kuitenkin vasta tulevaisuudessa, kun nähdään, kuinka pitkäikäiseksi valittu mikrokontrolleri osoittautuu.</p>	
Hakusanat	prosessori, mikrokontrolleri, ohjelmistokehitysympäristö, ala-asema

Author	Pekka Kulpakko
Title	A study of the processor technology and the Integrated Development Environment of the Remote Terminal Unit
Number of Pages	57
Date	7 September 2008
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Kim Malmberg, Technology Manager Antti Piironen, Principal Lecturer
<p>The goal of this degree thesis was to study the processor technology and the Integrated Development Environments that are applicable to Substation Automation Systems. The thesis was commissioned by Netcontrol, a company developing a new Remote Terminal Unit called Netcon100. The main objective of this thesis was to find three potential microcontrollers for an I/O-unit of the Netcon100 and test different Integrated Development Environments.</p> <p>Netcon100 will be developed using the IEC 61850 standard. Most of the microcontrollers on the market are not applicable because of the requirements of IEC 61850 and the long lifecycle expectancy of the Remote Terminal Unit.</p> <p>Finally six microcontrollers from different manufacturers were included in the study and three of them were selected for closer examination. Those three microcontrollers were ADSP-536 from Analog Devices, MCF52235 from Freescale and AT32UC3A0512 from Atmel. Test applications were made for the evaluation boards of each microcontroller. In the test application, the evaluation boards were programmed to act as Ethernet-to-Serial port converters. However, the main objective of the test applications was to become acquainted with different Integrated Development Environments.</p> <p>VisualDSP++ 5.0 from Analog Devices, Codewarrior Development Studio from Freescale, IAR Embedded Workbench from IAR and AVR32 Studio 2.0 from Atmel were the Integrated Development Environments which were tested. All of them had similar basic features but there were some differences in documentation.</p> <p>The degree thesis outlines the microcontrollers applicable to Netcon100. The final decision about the microcontroller will be based on this thesis and some additional tests.</p>	
Keywords	processor, microcontroller, Integrated Development Environment, Remote Terminal Unit

Sisällys

Tiivistelmä

Abstract

Lyhenteet, käsitteet ja määritelmät

1 Johdanto	7
2 SCADA-järjestelmän ja ala-aseman toimintaperiaate	8
3 Mikrokontrollerit.....	10
3.1 Vaatimukset.....	10
3.2 Vaihtoehdot	12
3.3 Analog Devices ADSP-BF536.....	15
3.4 Freescale MCF52235	19
3.5 Atmel AT32UC3A0512	25
4 Käytännön testaus	30
4.1 Laitteisto.....	30
4.2 Testisovellus.....	31
5 Ohjelmistokehitysympäristöt	34
5.1 Vaatimukset ja vaihtoehdot.....	34
5.2 VisualDSP++ 5.0	35
5.3 Codewarrior Development Studio.....	41
5.4 IAR Embedded Workbench	45
5.5 AVR32 Studio 2.0.....	49
6 Yhteenveto	51
Lähteet.....	53
Liitteet	
Liite 1: Analog Devices Blackfin -tuoteperhe	55
Liite 2: Atmel UC3 -tuoteperhe	56
Liite 3: Freescale ColdFire -tuoteperhe.....	57

Lyhenteet, käsitteet ja määritelmät

ALU	Mikrokontrollerin ytimeä löytyvä aritmeettis-looginen yksikkö, jota käytetään operaatioiden suorittamiseen.
DMA	Direct Memory Access. Mikrokontrollereiden ominaisuus, joka mahdollistaa itsenäisen tiedonsiirron muistin ja oheislaitteiden välillä vähentäen keskusyksikön kuormitusta.
DSP	Digital Signal Processing. Digitaalinen signaalinkäsittely.
EMI	Electromagnetic Interference. Sähkömagneettinen häiriö.
IDE	Integrated Development Environment. Ohjelmistokehitysympäristö, jossa ohjelmistokehityksen kaikki vaiheet on yhdistetty saman ympäristön alle.
IP	Internet Protocol. OSI-mallin verkkokerroksen protokolla, jota käytetään päätelaitteiden osoitteistamiseen ja pakettien reitittämiseen.
JTAG	Mikrokontrollereissa käytetty virheentestausportti.
lwIP	Lightweight IP. Sulautettuihin järjestelmiin kehitetty avoimen lähdekoodin periaatteella toimiva TCP/IP-pino.
MAC	Media Access Controller. Toteuttaa OSI-mallin siirtokerroksen.
MII	Media Independent Interface. Rajapinta Ethernet MAC- ja PHY-tasojen välillä.
PHY	Ethernetin fyysisen tason liitäntä. Toteuttaa OSI-mallin fyysisen kerroksen.
RISC	Reduced Instruction Set Computer. Mikrokontrollerin käskykannan suunnittelutapa, jossa käskyt pyritään pitämään rakenteeltaan yksinkertaisina.
RMII	Reduced Media Independent Interface. Signaalien määrä MAC- ja PHY-tasojen välillä pienempi kuin MII-rajapinnassa.
RTOS	Real-time operating system. Reaaliaikakäyttöjärjestelmä, jota voidaan käyttää esimerkiksi mikrokontrollerisovelluksissa, jotka tarvitsevat moniajtoa.
RTU	Remote Terminal Unit. Ala-asema, jonka avulla SCADA-järjestelmä liitetään valvottavaan tai ohjattavaan kohteeseen.

SCADA	Supervisory Control And Data Acquisition. Järjestelmä jonka avulla sähköverkoja voidaan valvoa ja ohjata.
TCP	Transmission Control Protocol. OSI-mallin kuljetuskerroksen tietoliikenneprotokolla.
UART	Universal Asynchronous Receiver Transmitter. Sarjaliikenneportti, jota käytetään sarjamuotoisessa tiedonsiirrossa.
UDP	User Datagram Protocol. OSI-mallin kuljetuskerroksen tietoliikenneprotokolla.
VDK	VisualDSP++ Kernel -niminen reaaliaikakäyttöjärjestelmä.

1 Johdanto

Työn tavoite ja sisältö

Tämän opinnäytetyön tarkoitus on kartoittaa sähköverkkojen verkostoautomaatioon soveltuvaa prosessoriteknologiaa ja ohjelmistokehitysympäristöjä. Työ tehdään Netcontrol-nimiselle yritykselle. Netcontrol on suunnittelemassa uutta Netcon100-nimistä ala-asemaa. Netcon100 muodostuu prosessoriyksiköstä ja siihen liitettävistä I/O-yksiköistä. Prosessoriyksikössä käytettävä mikrokontrolleri on jo aiemmin päätetty. Tässä opinnäytetyössä keskitytään siis I/O-yksikön prosessoriteknologian kartoitukseen. Netcon100-ala-asema suunnitellaan IEC 61850 -standardin mukaisesti.

Opinnäytetyö on rajattu tavoitteeseen löytää kolme potentiaalista mikrokontrolleri-vaihtoehtoa, joille tilataan kehitysalustat. Jokaiselle kehitysalustalle tehdään sama testisovellus, jonka avulla tutustutaan mikrokontrollereihin ja ohjelmistokehitysympäristöihin. Opinnäytetyön selvityksen ja Netcontrolin lisätestauksen perusteella päätetään lopullisesti, mitä mikrokontrolleria Netcon100-ala-asemassa tullaan käyttämään.

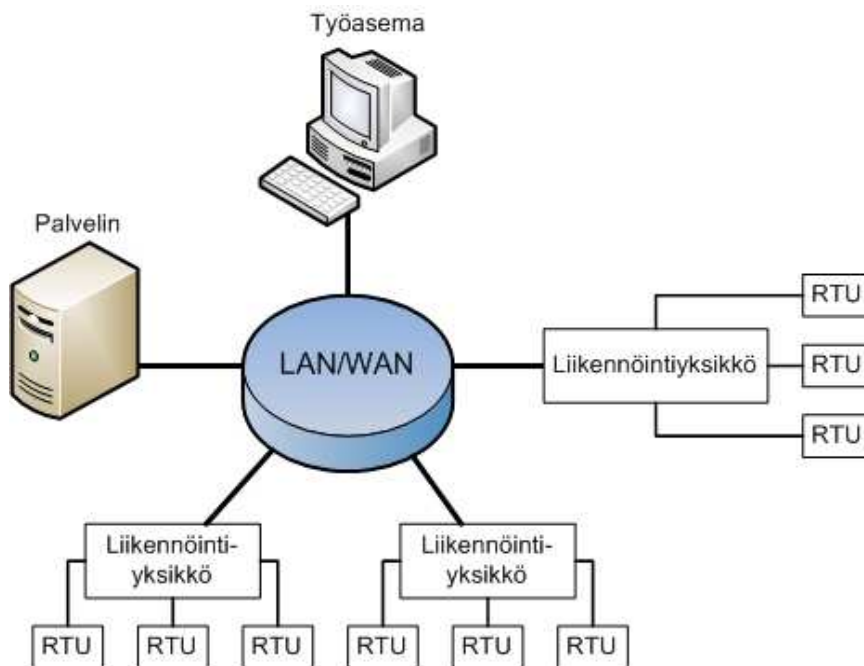
Opinnäytetyössä esitellään luvussa kaksi sähköverkkojen valvontaan ja ohjaukseen käytetyn SCADA-järjestelmän ja ala-aseman toimintaperiaate. Luvussa kolme esitellään mikrokontrollerin vaatimukset sekä eri mikrokontrollerivaihtoehdot. Neljännessä luvussa esitellään kehitysalustoille tehty testisovellus. Testisovellusten ohjelmoimiseen käytetyt ohjelmistokehitysympäristöt esitellään luvussa viisi. Ympäristöjen esittelyn yhteydessä kerrotaan myös, kuinka testisovelluksen tekeminen kyseisellä ympäristöllä onnistui.

Netcontrol Oy

Netcontrol Oy on vuonna 1991 perustettu yritys. Netcontrol kehittää ja toimittaa energiantuotannon ja energianjakelun valvonta- ja ohjausjärjestelmiä. Netcontrolin toimipisteet sijaitsevat Helsingissä, Ruotsin Västeråsissa sekä Norjan Oslossa. Tällä hetkellä yritys työllistää yhteensä noin 50 henkeä, joista Suomen toimipisteessä työskentelee noin 30.

2 SCADA-järjestelmän ja ala-aseman toimintaperiaate

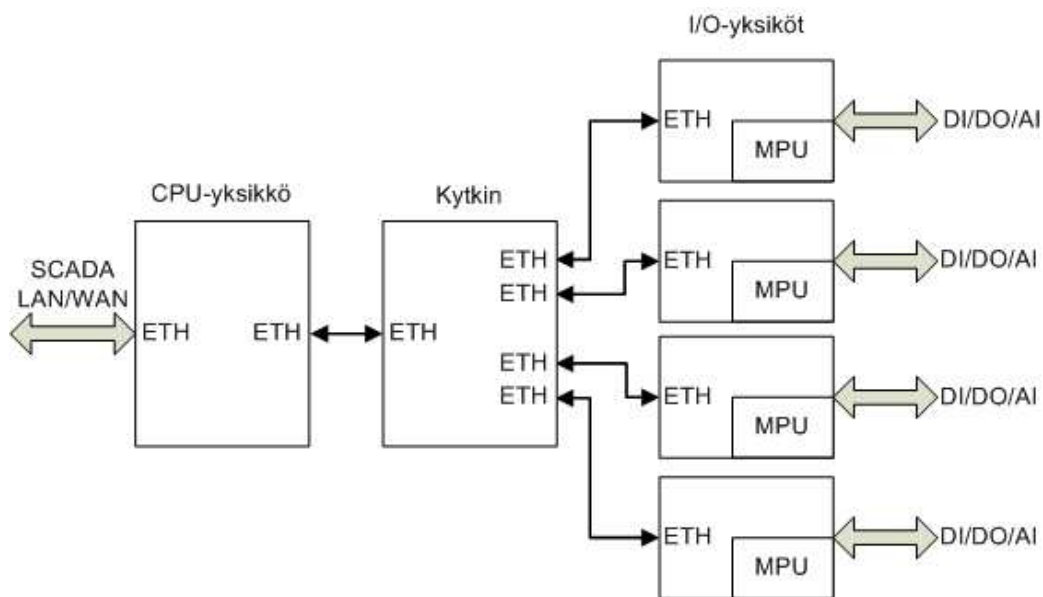
Sähköverkkoja valvotaan ja ohjataan SCADA-järjestelmien avulla. SCADA-järjestelmät muodostuvat palvelimista, työasemista, liikennöintiyksiköistä ja useista ala-asemista (RTU). Kuvassa 1 on esitetty esimerkki SCADA-järjestelmän rakenteesta. Sähköverkkoja valvotaan ja ohjataan työaseman käyttöliittymän välityksellä. Sähköverkon eri toimilaitteiden tilatiedot ja mittaustiedot tallennetaan palvelimen tietokantaan. Myös järjestelmästä toimilaitteille suoritettavat ohjaukset kulkevat palvelimen kautta. Yleensä SCADA-järjestelmien palvelimet on varmennettu siten, että käytössä on useita rinnakkain toimivia palvelimia. Myös työasemia on järjestelmässä yleensä useita. (1; 2; 3.)



Kuva 1. SCADA-järjestelmän arkkitehtuuri.

Ala-asetat muodostavat rajapinnan valvottavan tai ohjattavan laitteen kanssa. Ala-asetat toimivat mikrokontrollereilla, ja ne yhdistetään verkostoautomaation muihin laitteisiin I/O-liitäntöjen avulla. Ala-asetilla voidaan esimerkiksi suorittaa ohjauksia katkaisijoille tai välittää tilatietoja ja mitta-arvoja.

Netcon100-ala-asema, jonka lohkokaavio on kuvassa 2, koostuu prosessoriyksiköstä ja siihen liitettävistä I/O-yksiköistä. I/O-yksiköillä luetaan digitaalisia ja analogisia signaaleita (DI ja AI). Lisäksi I/O-yksiköstä löytyy digitaalisia lähtöjä (DO), joiden avulla ohjauksia voidaan suorittaa. I/O-yksiköt yhdistetään prosessoriyksikköön Ethernet-väylällä. Netcon100 suunnitellaan IEC 61850 -standardin mukaisesti. IEC 61850 -standardin vaatimuksista kerrotaan luvussa 3.1.



Kuva 2. Netcon100-ala-aseman lohkokaavio.

3 Mikrokontrollerit

3.1 Vaatimukset

IEC 61850 on kansainvälisen sähköalan standardointiorganisaation (IEC) kehittämä standardi, joka määrittää sähköverkkojen verkostoautomaatiolaitteiden vaatimukset. Aikaisemmin eri valmistajat ovat käyttäneet useita eri liikennöinti-protokollia alasemissään. Tämä on aiheuttanut eri valmistajien laitteiden ja järjestelmien välille yhteensovittamisongelmia. Yksi IEC 61850 -standardin päätavoitteista onkin mahdollistaa maailmanlaajuinen yhteentoimivuus eri valmistajien laitteille.

IEC 61850 määrittää, että prosessiväylän liikennöinnin tulee perustua Ethernetiin. Liikennöinnissä käytetään OSI-mallin ensimmäistä ja toista kerrosta eli fyysistä kerrosta ja siirtokerrosta. Ethernetin käyttö liikennöintimenetelmänä on keskeinen asia mikrokontrollerin valinnassa. Yksi mikrokontrollerin ehdottomista vaatimuksista onkin se, että siinä tulee olla integroitu 10/100Mbps:n Ethernet-portti. (1.)

Sähköverkkojen verkostoautomaatioon kehitetyillä laitteilla on huomattavasti kovemmat vaatimukset kuin esimerkiksi tavallisella kulutuselektronikalla. Standardin osiossa IEC 61850-3 on esitetty laitteiden luotettavuuteen ja toimintaympäristöön liittyviä vaatimuksia. IEC 61850-3 ei sisällä kaikkia vaatimuksia, vaan siinä viitataan myös muihin standardeihin.

Laitteiden tulee olla toimintavarmuudeltaan erittäin luotettavia. Sähköaseman toiminta ei saa vaarantua minkään verkostoautomaatioon liitetyn laitteen hajoamisen seurauksena, eikä yhden laitteen hajoaminen saa johtaa muiden laitteiden hajoamiseen. Laitteet eivät saa myöskään vikatilassa suorittaa mitään hallitsemattomia ohjaustoimenpiteitä. Toimintavarmuus tulee taata myös vaikeissa olosuhteissa, kuten alhaisessa ja korkeassa lämpötilassa sekä EMI-häiriöille alttiissa ympäristössä. IEC 61850-3 -standardissa todetaan, että yleiset teollisuusympäristöön tarkoitetut EMI-häiriönsietovaatimukset eivät sähköasemien tapauksessa ole riittäviä, joten vaatimuksia on tarkennettu IEC 61000 -standardissa. (4; 5.)

Sähköverkkojen verkostoautomaatiolaitteilta odotetaan pitkää, noin 15–25 vuoden, käyttöikää. Laitteiden pitkä elinkaari aiheuttaa omat haasteensa mikrokontrollerin valinnalle, koska valmistajat eivät takaa piirien saatavuutta ja tukea yhtä pitkällä aikavälillä. (1.)

Edellä mainitut asiat tulee ottaa huomioon ala-aseman suunnittelussa ja sitä kautta myös prosessoriteknologian valinnassa. Käyttötarkoitukseen sopivan mikrokontrollerin tulee olla elinikäennusteeltaan soveltuva pitkän elinkaaren tuotteisiin. Mikrokontrollerin tulee olla 32-bittinen ja suorituskyvyltään tehokas. Mikrokontrollerin pitää kyetä toimimaan riittävän laajalla lämpötila-alueella. Yleensä mikrokontrollereista löytyy erikseen teollisuuskäyttöön tarkoitetut mallit, joiden lämpötila-alue on $-40-85\text{ }^{\circ}\text{C}$. Mikrokontrollerissa tulee olla vahtiajastin, joka tarvittaessa kykenee käynnistämään mikrokontrollerin suorittaman ohjelman uudelleen, jos ohjelman suoritus jostain syystä jumiutuu. Lisäksi mikrokontrollerista tulee löytyä virheentestaukseen tarkoitettu liitäntä, esimerkiksi JTAG.

3.2 Vaihtoehdot

Kartoituksen yhteydessä tutkittiin seuraavien valmistajien mikrokontrollerivaihtoehtoja:

- Analog Devices
- Atmel
- Digi International
- Freescale Semiconductor
- Luminary Micro
- NXP Semiconductors
- STMicroelectronics
- Renesas Technology
- Microchip Technology
- Texas Instruments

Edellä olleessa luettelossa neljältä viimeksi mainitulta ei joko löytynyt tarkoitukseen sopivaa mikrokontrolleria tai löytyi vain turhan tehokas vaihtoehto. Listan kuudelta ensimmäiseltä valmistajalta valittiin jokaiselta yksi mikrokontrolleri tarkempaan kartoitukseen: Analog Devicesilta ADSP-BF536, Atmelilta AT32UC3A0512, Freescalelta MCF52235, Digiltä NS7520, NXP:ltä LPC2368 ja Luminary Microilta LM3S8938.

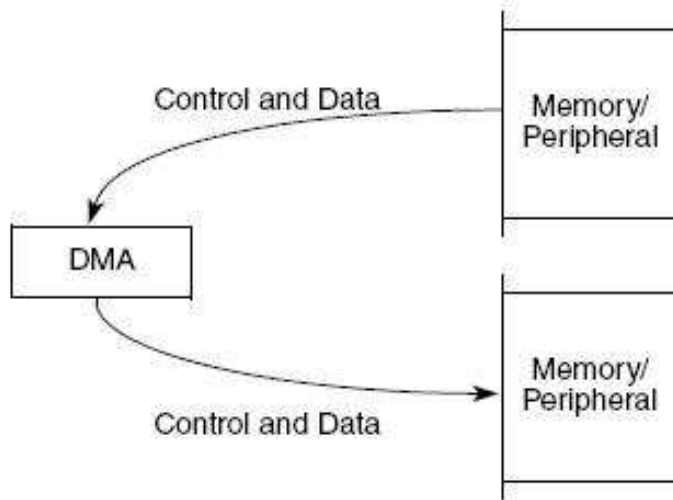
Kaikkien valmistajien tuoteperheissä on useita toisistaan hieman poikkeavia malleja. Saman tuoteperheen eri mikrokontrollerit eroavat toisistaan esimerkiksi muistin määrässä, oheislaitteiden määrässä ja kelloaajuudessa. Tässä vaiheessa uuden tuotteen kehitysprojektia ei vielä ole tarkkaa tietoa esimerkiksi mikrokontrollerin muisti-vaatimuksista, joten kartoitukseen valittiin jokaiselta valmistajalta tämän hetkisen tilanteen mukaan paras vaihtoehto vertailun helpottamiseksi.

Tarkempaan selvitykseen ja testattavaksi valittiin alkuperäisen suunnitelman mukaisesti mikrokontrollerit kolmelta eri valmistajalta: Analog Devicesilta ADSP-BF536, Atmelilta AT32UC3A0512 ja Freescalelta MCF52235. Kyseisille mikrokontrollereille tilattiin kokeilualustat, joiden avulla niihin päästiin tutustumaan käytännössä. Analog

Devicesin kokeilualusta on varustettu edellä mainitusta mikrokontrollerista poikkeavalla ADSP-BF537-mallilla, joka voi toimia korkeammalla kellotaajuudella. Testattujen mikro-kontrollereiden ominaisuuksista kerrotaan enemmän tulevissa luvuissa 3.3–3.5.

Testattavaksi valitut mikrokontrollerit ovat suurten ja tunnettujen valmistajien tuotteita. Kartoituksessa olleiden mikrokontrollereiden valmistajista Luminary Micro ja Digi ovat pienimpiä yrityksiä. Kaikki testatut mikrokontrollerit sisältävät melko monipuoliset integroidut oheislaitteet ja Freescalen mikrokontrolleria lukuun ottamatta ulkoisten muistien liittäminen niihin onnistuu hyvin sisäisen muistiohjaimen avulla. Analog Devicesin ja Digin mikrokontrollerit poikkesivat muista siten, että ne eivät sisältäneet lainkaan sisäistä Flash-muistia, jota käytetään mikrokontrollereiden ohjelmamuistina. Freescalen ja Luminary Micron mikrokontrollereiden sisäisen Flash-muistin tiedon säilymisajaksi luvataan 10 vuotta. Atmelin mikrokontrollerilla säilymisaika on 15 vuotta ja NXP:llä 20. Analog Devicesin ja Digin mikrokontrollereihin ohjelmamuistiksi voidaan valita ulkoinen Flash-muistipiiri, jonka tiedon säilymisaika on riittävän pitkä.

Testattavista mikrokontrollereista löytyy kaikista useita DMA-kanavia, joita voidaan käyttää oheislaitteiden ja muistin välisessä tiedonsiirrossa. DMA:ta käytettäessä suuretkaan tiedonsiirto-operaatiot eivät juuri kuormita itse prosessoria. Kuvassa 3 on esitetty Freescalen MCF52235-mikrokontrollerin DMA-yksikön toimintaperiaate lohkoakaaviona. Luminary Micron LM3S8938 oli kartoituksen ainoa mikrokontrolleri, josta DMA-ohjainta ei löytynyt lainkaan.



Kuva 3. MCF52235-mikrokontrollerin DMA-yksikön toimintaperiaate (6, s. 365).

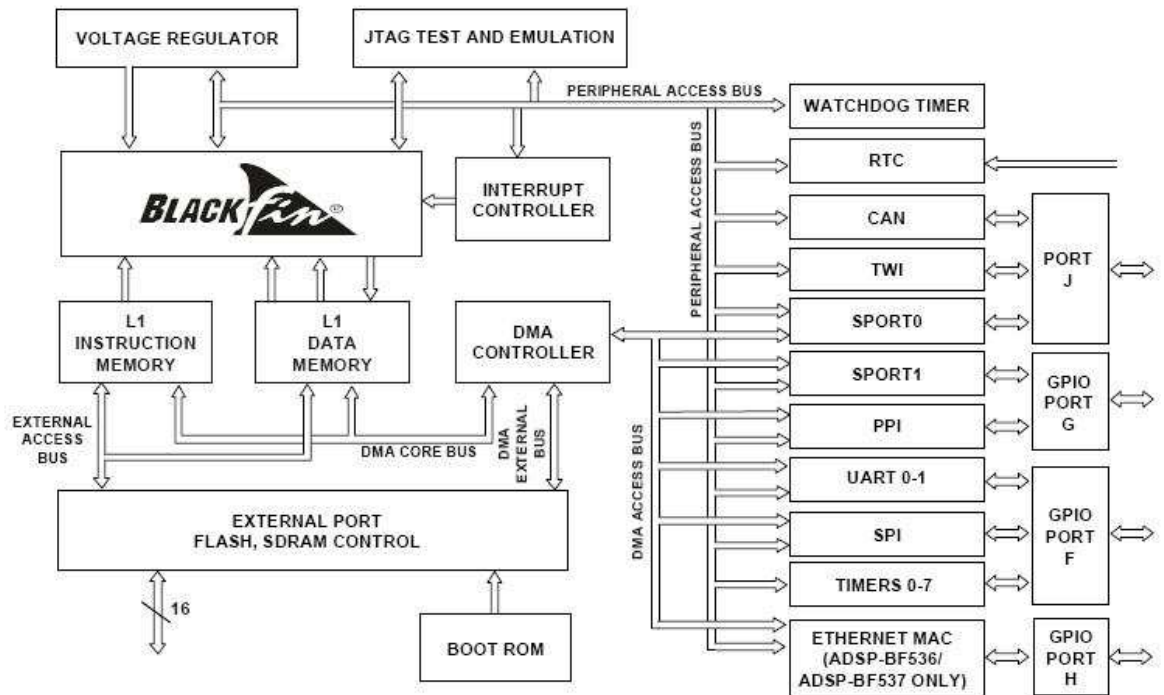
Kartoituksessa mukana olleiden mikrokontrollereiden tehonkulutuksissa oli huomattavia eroja. Atmelin AT32UC3A0512-mikrokontrolleri kuluttaa vähiten tehoa täydellä kellotaajuudella (noin 120 mW). Analog Devicesin, NXP:n ja Luminary Micron mikrokontrollereilla tehonkulutus on noin 400 milliwattia täydellä kellotaajuudella. Digin mikrokontrolleri kuluttaa noin 500 milliwattia ja Freescalen MCF52235 noin yhden watin verran tehoa täydellä kellotaajuudella. Tehonkulutuksia vertailtaessa on hyvä huomioida se, että Freescalen ja Luminary Micron tehonkulutukseen on huomioitu sisäisen Ethernet PHY:n kuluttama teho. Näin ollen muiden valmistajien mikrokontrollereiden tehonkulutuksiin täytyy lisätä 200–300 mW, joka on suurinpiirtein ulkoisen PHY-piirin tehonkulutus.

Eri valmistajien mikrokontrollereille tarjoamassa dokumentaatiossa oli jonkin verran eroavaisuuksia. Digin NS7520-mikrokontrollerille löytyi kaikista heikoimmat dokumentit, muiden valmistajien tarjoamat dokumentit mikrokontrollereille olivat varsin hyviä. Parhaat esimerkkisovellukset ja ajurit löytyivät Atmelin, Analog Devicesin ja Luminary Micron mikrokontrollereille.

3.3 Analog Devices ADSP-BF536

Analog Devicesin valmistama ADSP-BF536-mikrokontrolleri kuuluu Blackfin-tuoteperheeseen. Blackfin-arkkitehtuurin mikrokontrollerit ovat digitaaliseen signaalinkäsittelyyn suunniteltuja mikrokontrollereita. Tuoteperheeseen kuuluvat mikrokontrollerimallit on esitetty liitteessä 1. Blackfin-arkkitehtuurin ensimmäiset mikrokontrollerit on julkaistu vuonna 2000. ADSP-BF536-mikrokontrolleri on tullut markkinoille vuonna 2005.

ADSP-BF536 on 32-bittinen RISC-mikrokontrolleri, joka sisältää sisäänrakennettua SRAM-muistia 100 kilotavua. Sisäistä Flash-muistia mikrokontrollerista ei löydy lainkaan. Ulkoinen ohjelmamuisti liitetään mikrokontrolleriin EBIU (External Bus Interface Unit) -yksikön kautta. Kuvassa 4 on esitetty ADSP-BF536-mikrokontrollerin lohkokaavio. Kaaviosta käyvät ilmi mikrokontrollerin eri yksiköt, väylät ja integroidut oheislaitteet. ADSP-BF536:n ainoa kotelovaihtoehto on BGA-mallinen kotelo, josta on saatavilla 182- tai 208-pinniset versiot. BGA-kotelo on varsin hankala mahdollisten korjaus- ja piirinvaihtotoimenpiteiden yhteydessä. BGA-kotelossa juotospisteet löytyvät nimittäin piirin alta, joten käsin juottaminen on mahdotonta ja rikkinäisen piirin vaihtaminen onnistuu vain tarkoitukseen kehitetyillä laitteilla. (7, s.3; 8, s. 66.)



Kuva 4. Analog Devicesin ADSP-BF53x-sarjan mikrokontrollerin lohkokaavio (8, s. 64).

ADSP-BF536-mikrokontrollerin ydin

ADSP-BF536-mikrokontrollerin ydin koostuu kahdesta 16-bittisestä kertolaskuyksiköstä, kahdesta 40-bittisestä akusta, kahdesta 40-bittisestä aritmeettis-loogisesta yksiköstä eli ALU:sta, neljästä 8-bittisestä video ALU:sta ja 40-bittisestä siirtoyksiköstä. Kertolaskuyksiköt pystyvät suorittamaan yhden kellojakson aikana kahden 16-bittisen operandin MAC-operaation (Tässä yhteydessä MAC ei tarkoita OSI-mallin siirtokerrosta, vaan on lyhenne sanoista Multiply Accumulate, joka digitaalisessa signaalinkäsittelyssä tarkoittaa kahden luvun kertolaskua ja tuloksen yhteenlaskemista akun arvon kanssa). Myös 32-bittisten operandien kertolasku on mahdollista kertolaskuyksiköissä. Käskyjen noutaminen, tulkitseminen ja suoritus tapahtuvat ADSP-BF536:ssa 10-tasoisella liukuhihnalla. (9, s. 39–41, 106, 141–142.)

Ominaisuudet

ADSP-BF536-mikrokontrollerin oheislaitteille tarkoitettu DMA-ohjain on 12-kanavainen. Se mahdollistaa tiedonsiirron muistin ja seuraavien oheislaitteiden välillä: Ethernet MAC, SPORT0-1, UART0-1, PPI, SPI. Mikrokontrollerista löytyy lisäksi kaksi MDMA-ohjainta, jotka on tarkoitettu muistisiirtoja varten. MDMA:n avulla voidaan siirtää dataa ulkoisen ja sisäisen muistin välillä EBIU-yksikön kautta. (8, s. 140–149.)

Blackfin-mikrokontrollereissa keskeytyksiä hallitaan kahden eri tason avulla. SIC (System Interrupt Controller) -ohjain sisältää 32 sisääntuloa oheislaitteilta tulleille keskeytyksille. SIC-ohjaimen tehtävänä on ryhmitellä ja priorisoida keskeytykset ja ohjata ne edelleen 9 lähdön avulla CEC (Core Event Controller) -ohjaimelle. CEC tukee näin ollen yhdeksää yleiskäyttöistä keskeytystä kuuden ennalta määrätyn keskeytyslähteen lisäksi. Keskeytykset ohjataan SIC:stä CEC:lle neljän siirtorekisterin avulla. (8, s. 114–126.)

ADSP-BF536 sisältää sisäänrakennetun Ethernet MAC -yksikön, joka toimii 10:n tai 100 Mbps:n nopeudella kaksisuuntaisessa (full-duplex) tai vuorosuuntaisessa (half-duplex) tilassa. MAC-yksikkö tukee IEEE 802.3 -standardia. MAC-yksikkö tarjoaa ainoastaan OSI-mallin MAC-tason, ja se täytyy liittää MII- tai RMII-rajapinnan avulla erilliseen fyysisen tason (PHY) mahdollistavaan ulkoiseen piiriin. ADSP-BF536-mikrokontrollerin oheislaitteille tarkoitettu DMA-ohjain tukee MAC-yksikköä. DMA-ohjaimessa on varattu omat kanavat sekä lähtö- että vastaanottopuolelle. (8, s. 386–389.)

ADSP-BF536-mikrokontrolleriin on integroitu useita eri tiedonsiirtoportteja. Mikrokontrollerista löytyy kaksi UARTia, yksi SPI-portti, yksi CAN-portti ja yksi TWI-portti. CAN-portti täyttää CAN2.0B-standardin vaatimukset. TWI (Two-Wire Interface) -portti on Analog Devicesin käyttämä nimitys Philipsin kehittämästä, I2C-nimellä paremmin tunnetusta, portista. Mikrokontrollerista löytyy edellä mainittujen tiedonsiirtoporttien lisäksi lähinnä signaalinkäsittelyyn tarkoitettut PPI (Parallel

Peripheral Interface) -portti ja SPORT (Synchronous Serial Peripheral Port) -portti. PPI on vuorosuuntaisessa tilassa toimiva portti, jota tyypillisesti käytetään AD- ja DA-muuntimien, LCD-paneelien sekä videodekoodereiden ja -enkoodereiden liittämiseen. Kaksikanavainen SPORT-portti puolestaan tukee useita sarjamuotoisia tiedonsiirtoprotokollia. Tuettuja protokollia ovat G.711-standardin mukaiset A-law ja μ -law algoritmit, TDM (Time-Division Multiplexing), I2S ja H.100. Näiden lisäksi SPORT-porttia voidaan käyttää esimerkiksi AD-muuntimien liittämiseen tai muiden mikrokontrollereiden liittämiseen ADSP-BF536:een. (8, s. 62–63, 346, 730.)

ADSP-BF536-mikrokontrollerin ajastinyksikkö koostuu kahdeksasta identtisestä 32-bittisestä ajastimesta. Kukin ajastin on täysin itsenäinen, mutta halutessaan mikrokontrollerin käyttäjä voi käyttää ajastimia rinnakkain. Ajastimia voidaan käyttää esimerkiksi pulssinleveysmodulaatioon eli PWM:ään (Pulse Width Modulation). Mikrokontrollerista löytyy myös vahtiajastin, joka oli yksi ala-aseman mikrokontrollerille asetetuista perusvaatimuksista. ADSP-BF536:n vahtiajastin on 32-bittinen laskuri. Vahtiajastinta käytettäessä laskurin arvoa vähennetään joka kellojaksolla. Jos laskurin arvoa ei päivitetä, niin arvon saapuessa nolnaan vahtiajastin suorittaa mikrokontrollerin uudelleen käynnistyksen tai keskeytyksen. Edellä mainittujen ajastimien lisäksi mikrokontrollerista löytyy 32-bittinen ajastin jaksollisten keskeytysten tuottamiseen. Säännöllisiä keskeytyksiä voidaan tuottaa myös reaaliaikakellon eli RTC:n avulla. ADSP-BF536:n RTC koostuu neljästä laskurista, joita käytetään sekuntien, minuuttien, tuntien ja päivien laskemiseen. Päivien laskemiseen käytetty laskuri on 15-bittinen. (8, s. 890, 952–954, 962–965, 974–975.)

Tehonkulutus ja toimintatilat

ADSP-BF536 voi toimia 400 MHz:n kelloaajuudella ja 2,5 tai 3,3 voltin käyttöjännitteellä. Mikrokontrollerissa on sisäänrakennettu jänniteregulaattori, jonka avulla saadaan ytimelle sen tarvitsema 0,8–1,3 voltin käyttöjännite. Mikrokontrollerin kelloaajuutta ja ytimen toimintajännitettä voidaan laskea, jolloin mikrokontrollerin tehonkulutusta saadaan pienennettyä huomattavasti. Laskemalla ytimen jännitettä ja

kellotaajuutta 400 MHz:stä 250 MHz:iin saadaan tehonkulutus lähes puolitettyä. (7, s. 23–24; 10.)

Tehonkulutusta voidaan hallita myös eri toimintatilojen avulla. ADSP-BF536:ssa on neljä toimintatilaa, jotka on esitetty kuvassa 5. Full On -tilassa mikrokontrolleri toimii täydellä kellotaajuudella, joka mahdollistaa parhaan suorituskyvyn. Sleep- ja Deep Sleep -tiloissa tehonkulutus on huomattavasti pienempää kellosignaaleiden (CCLK ja SCLK) sammuttamisen seurauksena. Kaikista pienin tehonkulutus saadaan Hibernate-tilassa, joka ei kuitenkaan ole mikrokontrollerin varsinainen toimintatila. Hibernate-tilaan siirryttäessä kaikki säilytettävät tiedot täytyy tallentaa katoamattomaan muistiin. (8, s. 1069–1085.)

Operating Mode	Power Savings	PLL		CCLK	SCLK	Allowed DMA Access
		Status	Bypassed			
Full On	None	Enabled	No	Enabled	Enabled	L1
Active	Medium	Enabled ¹	Yes	Enabled	Enabled	L1
Sleep	High	Enabled	No	Disabled	Enabled	–
Deep Sleep	Maximum	Disabled	–	Disabled	Disabled	–

Kuva 5. ADSP-BF536-mikrokontrollerin toimintatilat (8, s. 1070).

Saatavuus

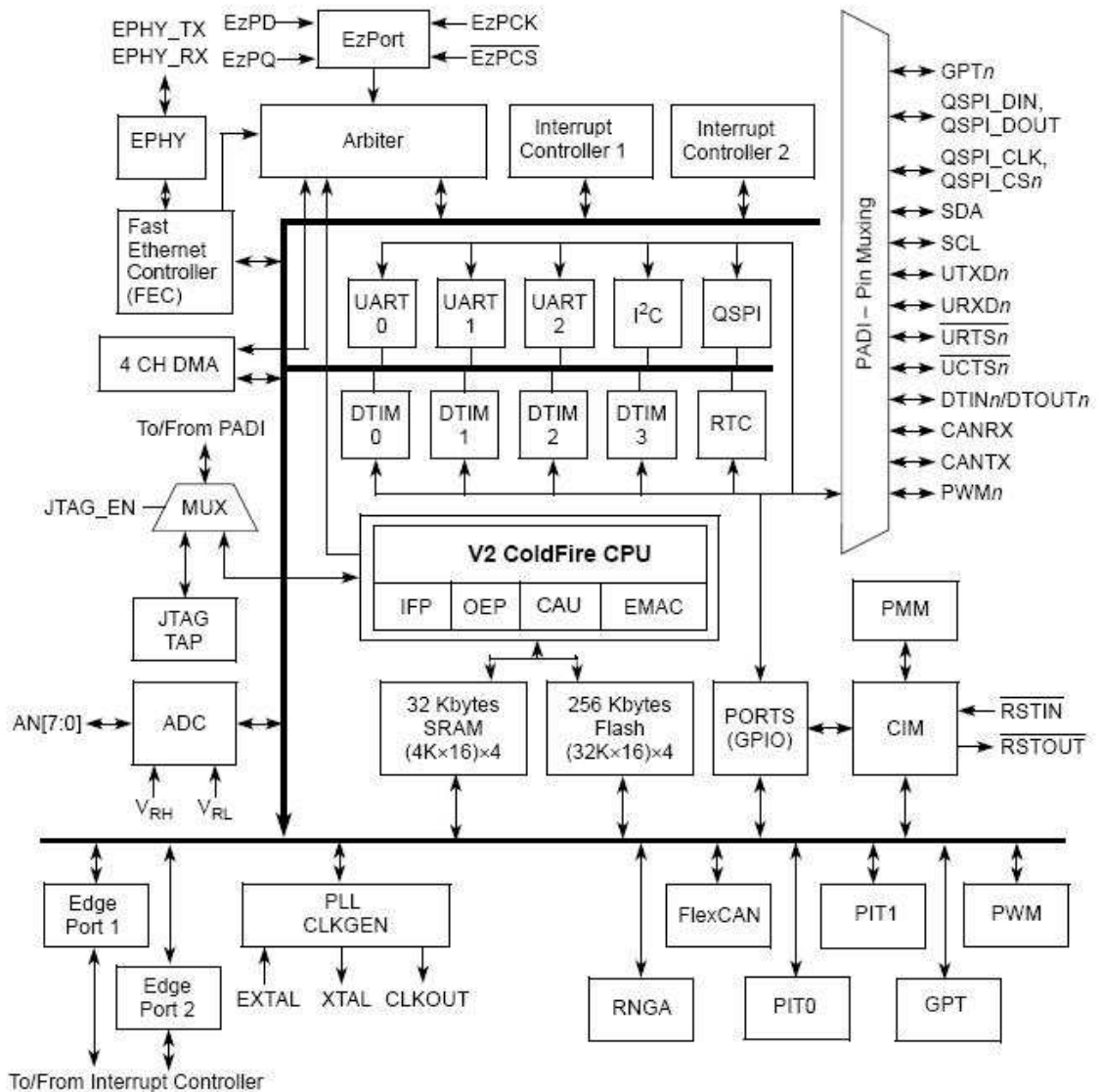
ADSP-BF536-mikrokontrolleri on ollut markkinoilla jo useamman vuoden ajan. Sen saatavuus opinnäytetyön tekohetkellä oli hyvä. Saatavuutta tiedusteltiin Arrow Finland Oy:ltä, joka ilmoitti piirin toimitusajaksi kolme viikkoa. Mikrokontrollerille löytyy Suomesta myös muita jakelijoita.

3.4 Freescale MCF52235

MCF52235-mikrokontrolleri kuuluu Freescale Semiconductorin ColdFire V2 (Version 2) -tuoteperheeseen. ColdFire-arkkitehtuuri on testattavina olleista mikrokontrollereista

selkeästi vanhin. ColdFire-mikrokontrollereita on ollut markkinoilla jo vuodesta 1994 alkaen. MCF52235-malli on ollut saatavilla muutaman vuoden ajan. Liitteessä 3 on linjaus ColdFire-sarjan mikrokontrollereista. ColdFire-tuoteperhe on selvästi laajempi kuin Blackfin- ja Atmelin UC3-tuoteperheet. Freescale on myös edelleen jatkamassa ColdFire-tuoteperheen kehittämistä ja tuomassa markkinoille uusia malleja, kuten liitteestä 3 nähdään.

MCF52235 on 32-bittinen RISC-tyyppinen mikrokontrolleri, joka sisältää paljon integroitua oheislaitteita. MCF52235 on varustettu sisäänrakennetulla 256 kilotavun Flash-muistilla ja 32 kilotavun SRAM-muistilla. MCF52235 on suunniteltu yleiskäyttöiseksi mikrokontrolleriksi. Mikrokontrolleria on saatavilla kahdessa eri kotelossa. Kotelovaihtoehdot ovat 112-pinninen LQFP tai 121-pinninen MAPBGA. Kuvassa 6 on esitetty MCF52235-mikrokontrollerin lohkokaavio. Kuvasta käy ilmi mikrokontrollerin eri yksiköt sekä integroidut oheislaitteet. (6, s. 11–13.)



Kuva 6. MCF52235-mikrokontrollerin lohkokaavio (6, s. 13).

MCF52235-mikrokontrollerin ydin

ColdFire V2 -sarjan mikrokontrollerin ydin sisältää kaksi erillistä kaksitasoista liukuhihnaa. IFP (Instruction Fetch Pipeline) -liukuhihnaa käytetään käskyjen noutamiseen ja OEP (Operand Execution Pipeline) -liukuhihnaa käskyjen suorittamiseen. (6, s. 41–42.)

Mikrokontrollerin ydin sisältää myös EMAC (Enhanced Multiply-Accumulate Unit) -yksikön, jota voidaan käyttää DSP-operaatioiden suorittamiseen. EMAC toimii neljäntasoisella liukuhihnalla. Sen toiminta on optimoitu 32-bittisten operandien kertolaskuihin, ja se sisältää neljä 48-bittistä akkua. ColdFire-ytimeistä löytyy myös kryptausyksikkö CAU (Cryptographic Acceleration Unit), joka tukee seuraavia kryptausalgoritmeja: DES, 3DES, AES, MD5 ja SHA-1 (6, s. 73, 93.)

Muisti ja ominaisuudet

MCF52235-mikrokontrollerista löytyy sisäänrakennettua Flash- ja SRAM-muistia. Flash-muisti on haihtumatonta 32-bittistä muistia, jota voidaan käyttää ohjelma- ja datamuistina. Mikrokontrollerin 32 kilotavun SRAM-muisti on tarkoitettu datamuistiksi. Datan lukeminen SRAM-muistista onnistuu yhden kellojakson aikana. Muistista lukeminen voidaan toteuttaa myös DMA:ta käyttäen. SRAM on jaettu kahteen osaan, mikä mahdollistaa prosessorin ja DMA:n samanaikaiset lukuoperaatiot muistista. (6, s. 175, 247–249.)

Sisäänrakennettu Flash-muisti kestää 100 000 kirjoitusoperaatiota. Yksi MCF52235-mikrokontrollerin ehdottomista heikkouksista on kuitenkin se, että Freescale ilmoittaa Flash-muistin datan säilymisajaksi vain kymmenen vuotta. (11.)

DMA-yksikkö, joka MCF52235-mikrokontrollerista löytyy, koostuu neljästä itsenäisestä ja toiminnaltaan samankaltaisesta kanavasta. DMA mahdollistaa tiedonsiirron muistin ja oheislaitteiden välillä. Tiedonsiirto voi olla joko tiedon lukemista tai kirjoittamista. (6, s. 365.)

Keskeytyksiä hallitaan MCF52235-mikrokontrollerissa kahden keskeytysohjaimen (INTC0 ja INTC1) avulla. Molemmat keskeytysohjaimet voivat ohjata 63:a keskeytyslähdettä eli yhteensä mikrokontrolleri pystyy hallitsemaan 126:ta keskeytystä. Jokaiselle keskeytykselle löytyy oma rekisteri. Rekistereiden avulla keskeyksiä voidaan ohjata esimerkiksi prioriteetteja asettamalla. (6, s. 221.)

MCF52235-mikrokontrollerista löytyy integroitu FEC (Fast Ethernet Controller) -yksikkö. Ethernet-yksikön MAC tukee 10:n ja 100 Mbps:n Ethernet/IEEE 802.3 -verkkoja. MAC voi toimia sekä kaksisuuntaisessa että vuorosuuntaisessa tilassa. FEC saadaan liitettyä DMA-ohjaimeen ja se yhdistetään fyysiselle tasolle MII-rajapinnan avulla. MCF52235:n Ethernet-liitäntä eroaa Atmelin ja Analog Devicesin mikrokontrollereista siten, että mikrokontrolleriin on edellä mainituista poiketen integroitu myös fyysisen tason (PHY) liitäntä. Muissa mikrokontrollereissa fyysiselle tasolle liittäminen joudutaan toteuttamaan erillisen piirin avulla. MCF52235-mikrokontrollerin MAC toimii vain integroidun fyysisen tason liitännän kanssa, eli sitä ei voida liittää ulkopuoliseen piiriin. (6, s. 279–282, 331.)

MCF52235-mikrokontrollerissa on useita portteja, joiden avulla se voidaan liittää muihin laitteisiin. Mikrokontrolleriin on integroitu kolme UARTia, yksi SPI- ja yksi I2C-yksikkö. Jokainen UART saadaan tarvittaessa liitettyä mikrokontrollerin DMA-yksikköön ja keskeytyslogiikkaan. SPI-yksikkö sisältää neljä piirinvalintasignaalia, joten mikrokontrolleri voidaan sen avulla liittää 15:een eri laitteeseen. Lisäksi mikrokontrollerissa on FlexCAN-yksikkö, joka mahdollistaa CAN-väylässä tapahtuvan tiedonsiirron. FlexCAN-yksikkö tukee protokollaa 2.0B. (6, s. 428, 443, 555.)

MCF52235 sisältää kahdeksan itsenäistä pulssinleveysmodulaatioon tarkoitettua kanavaa. PWM:ää varten mikrokontrollerissa on neljä kelloa. Kellot on jaettu PWM-kanavien käytettäväksi siten, että neljä PWM-kanavaa voi käyttää kahta kellosignaalia. Jokainen kanava ei siis voi käyttää omaa kellosignaalia, kuten esimerkiksi Atmelin AT32UC3A0512-mikrokontrollerissa. MCF52235:ssa on myös sisäänrakennettu ADC-yksikkö, joka koostuu kahdesta 12 bitin resoluutioon kykenevästä AD-muuntimesta. (6, s. 493, 531–532, 543–544.)

MCF52235-mikrokontrolleri sisältää reaaliaikakellon. Reaaliaikakello päivittää sekunnit ja minuutit 6-bittisiin laskureihin. Tunnit päivittyvät 5-bittiseen laskuriin ja päivät 16-bittiseen laskuriin. Reaaliaikakello toimii 1 Hz:n taajuudella, ja sitä voidaan käyttää keskeytysten käynnistämiseen. Mikrokontrollerista löytyy lisäksi 32-bittinen

vahtiajastin. Vahtiajastimella voidaan uudelleenkäynnistää mikrokontrolleri, jos sen suorittama ohjelma jää jumiin. (6, s. 23, 131–132.)

Tehonkulutus ja toimintatilat

MCF52235-mikrokontrolleri toimii 3,3 voltin käyttöjännitteellä. Mikrokontrolleri kuluttaa täydellä kelloaajuudella noin yhden watin verran tehoa. Verrattaessa tehonkulutusta muihin mikrokontrollereihin on hyvä muistaa, että MCF52235 sisältää integroidun Ethernet PHY:n.

MCF52235 voidaan asettaa neljään erilaiseen toimintatilaan, joiden avulla voidaan hallita mikrokontrollerin tehonkulutusta. Run-tila on normaali toimintatila, jossa tehonkulutus on suoraan riippuvainen käytetystä kelloaajuudesta. Wait-tilassa prosessorin ja muistien kellot ovat pysähdyksissä, kunnes mikrokontrolleri saa joltain oheislaitteeltaan keskeytyksen, joka käynnistää kellot uudelleen. Doze-tila on hyvin pitkälti Wait-tilan kaltainen, mutta siinä jokaisen oheislaitteen toiminta määritetään erikseen. Viimeinen toimintatila, johon MCF52235 voidaan asettaa on Stop-tila, jossa tehonkulutus on kaikista pienin. Silloin kaikki mikrokontrollerin kellot on pysäytetty ja oheislaitteiden toiminta lopetettu. (6, s. 155–156.)

Kuvassa 7 on esitetty MCF52235-mikrokontrollerin eri toimintatilojen virrankulutus. Kyseiset virta-arvot saadaan, kun mikrokontrolleri suorittaa ohjelmakoodia Flash-muistista ja kaikkien muiden oheislaitteiden paitsi Flash-yksikön kellot on sammutettu. (12, s. 33.)

Mode ²	PLL @25 MHz (typical) ³	PLL @60 MHz (typical) ³	PLL @60 MHz (peak) ⁴	Unit
STOP mode 3 (STPMD[1:0]=11)	0.2		1.0	mA
STOP mode 2 (STPMD[1:0]=10)	7		—	
STOP mode 1 (STPMD[1:0]=01)	10	12	—	
STOP mode 0 (STPMD[1:0]=00)	10	12	—	
WAIT	16	27	—	
DOZE	16	27	—	
RUN	25	45	—	

Kuva 7. MCF52235-mikrokontrollerin toimintatilojen virrankulutus (12, s. 33).

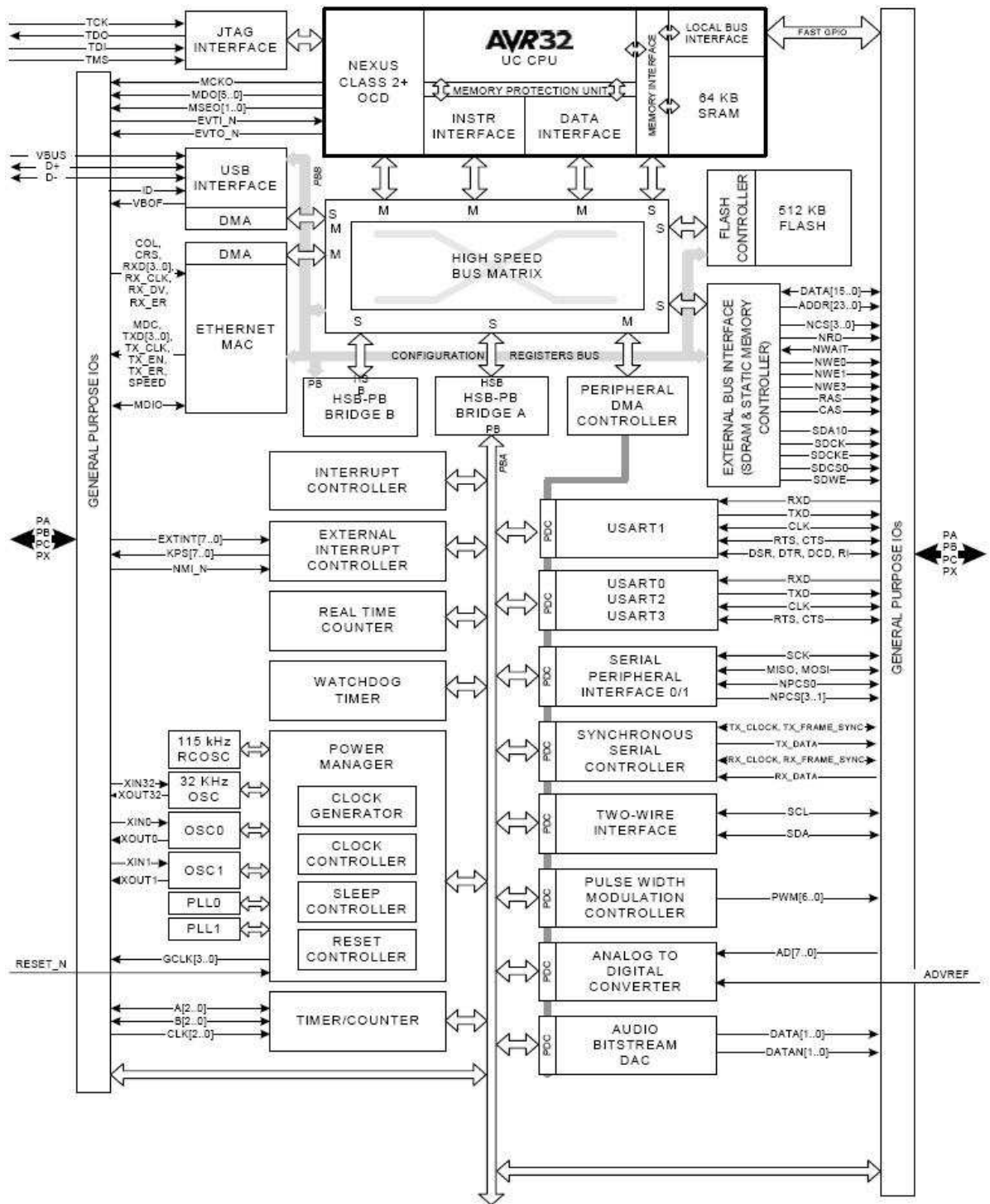
Saatavuus

Freescalen MCF52235-mikrokontrollerin saatavuus näyttäisi olevan erittäin hyvä opinnäytetyön tekohetkellä. Saatavuutta tiedusteltiin Arrow Finland Oy:ltä ja Future Electronicsilta. Molemmilla jakelijoilla kyseisen mallin saatavuus oli tiedusteluhetkellä hyvä, ja mikrokontrollereita löytyi suoraan jakelijoiden varastoista. Edellä mainittujen jakelijoiden lisäksi Freescalen tuotteille löytyy Suomesta myös muita jakelijoita.

3.5 Atmel AT32UC3A0512

AT32UC3A0512 kuuluu UC3-sarjan mikrokontrollereihin, jotka perustuvat Atmelin vuonna 2006 julkaisemaan AVR32-arkkitehtuuriin. UC3 on järjestyksessään toinen AVR32-arkkitehtuuriin perustuva mikrokontrollerisarja. UC3-sarja muodostuu tällä hetkellä UC3A- ja UC3B-mikrokontrollereista, joista UC3A-sarjasta löytyy käyttötarkoitukseen sopivat ominaisuudet. AT32UC3A0512 on opinnäytetyössä testatuista mikrokontrollereista kaikista uusin. Ensimmäiset näytekappaleet mikrokontrollerista on julkaistu vuonna 2007, ja varsinaiseen tuotantoon mikrokontrolleri on saatu vuonna 2008. UC3-sarjaan kuuluvat mikrokontrollerit löytyvät liitteessä 2 olevasta kuvasta. Kuten kuvasta käy ilmi, niin UC3A ja UC3B eivät ole jäämässä sarjan ainoiksi mikrokontrollereiksi, vaan Atmel on tulevaisuudessa laajentamassa UC3-sarjaa uusilla mikrokontrollerimalleilla.

UC3-sarjan mikrokontrollerit ovat todella alhaisen tehonkulutuksen omaavia 32-bittisiä RISC-mikrokontrollereita. AT32UC3A0512-malli sisältää 512 kilotavua sisäänrakennettua Flash-muistia, 64 kilotavua SRAM-muistia ja paljon samalle piirille integroitua oheislaitteita. Mikrokontrolleriin kuuluvat eri yksiköt, väylät ja oheislaitteet näkyvät kuvan 8 lohkokaaviossa. AT32UC3A0512-mikrokontrolleri on saatavissa 100-pinnisessä TQFP- ja 144-pinnisessä LQFP-kotelossa. (13, s 1.)



Kuva 8. AT32UC3A0512-mikrokontrollerin lohkokaavio (13, s. 5).

AT32UC3A0512-mikrokontrollerin ydin

Mikrokontrollerin ytimen liukuhihna muodostuu kolmesta eri tasosta. IF (Instruction Fetch) -tasoa käytetään käskyjen noutamiseen ja ID (Instruction decode) -tasoa käskyjen tulkitsemiseen. Käskyjen suoritus tapahtuu Ex (Instruction Execute) -tasolla, joka voidaan vielä jakaa kolmeen rinnakkaiseen osaan, joita ovat MUL, ALU ja LS. Suurin osa datan käsittelyoperaatioista suoritetaan ALU:ssa. MUL-yksikössä suoritetaan prosessorin kertolasku- ja MAC-operaatiot. MUL-yksikössä on mahdollista suorittaa 16- ja 32-bittisten operandien kertolasku- ja MAC-operaatioita. LS-yksikkö puolestaan hoitaa datansiirto-operaatiot. (14, s. 17–18.)

Muisti ja ominaisuudet

AT32UC3A0512-mikrokontrollerista löytyy 512 kilotavua sisäistä Flash-muistia, joka liitetään 32-bittiseen HSB (High Speed Bus) -väylään FLASHC (Flash Controller) -ohjaimen avulla. Flash-muistille voidaan kirjoittaa 100 000 kertaa, ja Atmel ilmoittaa, että Flash-muistiin kirjoitettu data säilyy 15 vuotta. 15 vuoden säilyvyys on mikrokontrolleriin integroidulle ohjelmamuistille varsin hyvä aika. Sisäänrakennettua SRAM-muistia mikrokontrollerissa on 64 kilotavua. Ulkoisten muistipiirien liittämistä varten AT32UC3A0512-mikrokontrollerissa on EBI (External Bus Interface) -liitäntä, johon voidaan liittää viisi ulkoista muistipiiriä. (13, s. 32, 144.)

Mikrokontrolleriin on liitetty 15-kanavainen DMA-ohjain, josta käytetään nimitystä PDCA (Peripheral DMA Controller). PDCA:n avulla voidaan siirtää dataa oheislaitteelta sisäiseen tai ulkoiseen muistiin ja päinvastoin lukea dataa muistista oheislaitteelle vähäisellä prosessorin kuormituksella. (13, s. 1, 152.)

AT32UC3A0512-mikrokontrollerin keskeytyksiä hallitaan ITC (Interrupt Controller) -yksikön avulla. ITC muodostuu 64:stä keskeytysryhmästä. Jokaisessa ryhmässä voi olla 32 keskeytyspyyntölinjaa, jotka on yhdistetty mikrokontrollerin oheislaitteille. AVR32-arkkitehtuurissa keskeytykset voidaan priorisoida rekistereiden avulla neljälle eri

tasolle. Mikrokontrollerissa on lisäksi EIC (External Interrupt Controller) -yksikkö, jota käytetään ulkoisten keskeytysten ohjaamiseen. (13, s. 98, 104.)

MACB-yksikkö, joka AT32UC3A0512-mikrokontrolleriin on integroitu, on Ethernet/IEEE 802.3 -yhteensopiva. Se toimii kaksisuuntaisessa tai vuorosuuntaisessa tilassa 10:n ja 100 Mbps:n nopeudella. MACB-yksikkö on ominaisuuksiltaan hyvin samankaltainen Blackfin-mikrokontrollerin MAC-yksikön kanssa. MACB pitää liittää samaan tapaan ulkoiseen PHY-piiriin MII- tai RMII-rajapinnan välityksellä. MACB-yksikkö voidaan liittää mikrokontrollerin DMA-ohjaimen. Lähetys- ja vastaanottopuoli saadaan liitettyä eri DMA-kanaviin. (13, s. 431–432.)

AT32UC3A0512-mikrokontrollerissa on neljä USARTia (Universal Synchronous/Asynchronous Receiver Transmitter), kaksi SPI-yksikköä ja yksi TWI-yksikkö. Kaikki edellä mainitut voidaan liittää mikrokontrollerin DMA-ohjaimen. SPI-yksiköt sisältävät neljä piirinvalintasiignaalia, jotka mahdollistavat sen, että yhteen SPI-väylään voidaan liittää 15 laitetta. TWI on Atmelin käyttämä nimitys I2C-väylästä.

Sarjamuotoisia tiedonsiirtoprotokollia varten AT32UC3A0512:ssa on myös SSC (Synchronous Serial Controller) -ohjain. SSC tukee muun muassa I2S-, Short Frame Sync- ja Long Frame Sync -protokollia. Atmelin mikrokontrollerissa ei ole lainkaan CAN-väylää, joka on integroitu Analog Devicesin ja Freescalen mikrokontrollereihin. AT32UC3A0512 puolestaan sisältää USB-yksikön, jota muissa testatuissa mikrokontrollereissa ei ole. (13, s. 1, 257.)

USB-yksikkö täyttää USB 2.0 -määritykset, mutta ei tue high speed -nopeutta (480 Mbps). USB-yksikkö sisältää myös OTG (On-The-Go) -ominaisuuden. Perinteisesti USB-liitäntää käyttävät laitteet, joita esimerkiksi PC:hen on liitetty, ovat olleet oheislaitteita, jolloin PC on ollut isäntälaitte. Kannettavien laitteiden USB-liitäntöjen ja laitteiden liittämistarpeiden lisääntyminen johtivat siihen, että USB-määrityksiin lisättiin OTG-ominaisuus. OTG-laitteet voivat toimia sekä isäntälaitteina että oheislaitteina. AT32UC3A0512-mikrokontrollerin USB-yksikkö voidaan muiden oheislaitteiden tapaan liittää DMA-ohjaimen. (13, s. 491; 15.)

Pulssinleveysmodulaatiota varten AT32UC3A0512-mikrokontrolleriin on liitetty PWM-ohjain. Ohjaimen avulla on mahdollista käyttää seitsemää itsenäistä PWM-kanavaa. Jokaista PWM-kanavaa varten on varattu oma 20-bittinen laskuri. Mikrokontrollerissa on myös sisäänrakennetut AD- ja DA-muuntimet. AD-muunnin mahdollistaa kahdeksan analogisen kanavan käytön ja se tukee 8 ja 10 bitin resoluutiota. (13, s. 670, 696.)

AT32UC3A0512-mikrokontrollerista löytyvä reaaliaikakello päivittää 32-bittistä laskuria. Laskurin päivittymisnopeus voidaan säätää 32 kHz:n oskillaattorin tai sisäisen RC-oskillaattorin taajuuden avulla. Laskuri voi kuitenkin päivittyä korkeintaan 16 kHz:n taajuudella. Reaaliaikakelloa voidaan käyttää säännöllisten keskeytysten tuottamiseen. Myös Atmelin mikrokontrollerissa on vahtiajastin, joka tarvittaessa käynnistää mikrokontrollerin uudelleen. (13, s. 85–87, 93.)

Tehonkulutus ja toimintatilat

AT32UC3A0512 voi toimia 3,3 voltin käyttöjännitteellä, jolloin sisäisen jännite-regulaattorin avulla saadaan mikrokontrollerin ytimelle, muisteille ja oheislaitteille niiden tarvitsema 1,8 voltin jännite. Toisessa vaihtoehdossa mikrokontrollerille voidaan suoraan syöttää 3,3 ja 1,8 voltin jännitteet. AT32UC3A0512-mikrokontrollerissa on kartoituksessa mukana olleista mikrokontrollereista pienin tehonkulutus. (13, s. 13–14.)

PM (Power Manager) -yksikköä käytetään AT32UC3A0512-mikrokontrollerin oskillaattoreiden ja PLL-yksiköiden hallintaan. PM-yksikön avulla voidaan asettaa mikrokontrolleri erilaisiin tehonsäästötiloihin. Normaalin toimintatilan lisäksi mikrokontrolleri voidaan asettaa kuuteen eri toimintatilaan, joilla sen tehonkulutusta saadaan pienemmäksi. Eri tehonsäästötilat on esitetty kuvassa 9.

Index	Sleep Mode	CPU	HSB	PBA,B GCLK	Osc0,1 PLL0,1	Osc32	RCOsc	BOD & Bandgap	Voltage Regulator
0	Idle	Stop	Run	Run	Run	Run	Run	On	Full power
1	Frozen	Stop	Stop	Run	Run	Run	Run	On	Full power
2	Standby	Stop	Stop	Stop	Run	Run	Run	On	Full power
3	Stop	Stop	Stop	Stop	Stop	Run	Run	On	Low power
4	DeepStop	Stop	Stop	Stop	Stop	Run	Run	Off	Low power
5	Static	Stop	Stop	Stop	Stop	Stop	Stop	Off	Low power

Kuva 9. AT32UC3A0512-mikrokontrollerin tehonsäästötilat (13, s. 5).

Static-toimintatilassa on sammutettu kaikki mikrokontrollerin oskillaattorit, joten tehonkulutus on silloin kaikkein pienimmillään. Normaalisissa toimintatilassa virrankulutus on 60 MHz:n kellotaajuudella 37 milliampeeria, mutta Static-tilassa päästään pienimmillään 14 mikroampeerin virrankulutukseen. Eri toimintatiloista voidaan palata normaalitilaan esimerkiksi ulkoisten keskeytysten avulla. (13, s. 52, 59–60, 760.)

Saatavuus

Kuten aiemmin on todettu, AT32UC3A0512-mikrokontrolleri on aivan uusi piiri, jonka tuotanto on vasta aloitettu. Tästä johtuen tuotteen saatavuus ei vielä ole hyvä.

Saatavuutta tiedusteltiin kahdelta jakelijalta, Arrow Finland Oy:ltä ja Acte Oy:ltä.

Molemmat ilmoittivat mikrokontrollerin toimitusajan olevan opinnäytetyön tekohetkellä yli kolme kuukautta.

4 Käytännön testaus

4.1 Laitteisto

Ohjelmistokehitysympäristöjä ja mikrokontrollereita kokeiltiin käytännössä rakentamalla sama testisovellus jokaiselle tilatulle kehitysalustalle ja testaamalla sovelluksen toiminta Netcontrolin valmistamilla ala-aseilla. Ala-asemat, joita

testauksessa käytettiin, ovat nimeltään Netcon Gateway 502 (GW502) ja Netcon RTU8. GW502 on SCADA-järjestelmän liikennöintiyksikkö, johon saadaan liitettyä ala-asemia. GW502 tukee useita eri liikennöinti-protokollia, ja se sisältää kolme sarjaporttia, kaksi Ethernet-porttia ja 5-porttisen Ethernet-kytkimen. RTU8 on erotinasemien ohjaukseen ja valvontaan kehitetty ala-asema, joka liitetään SCADA-järjestelmään sarjaportin avulla. Testisovelluksen tärkein tarkoitus oli tutustua eri ohjelmistokehitysympäristöihin. (16; 17.)

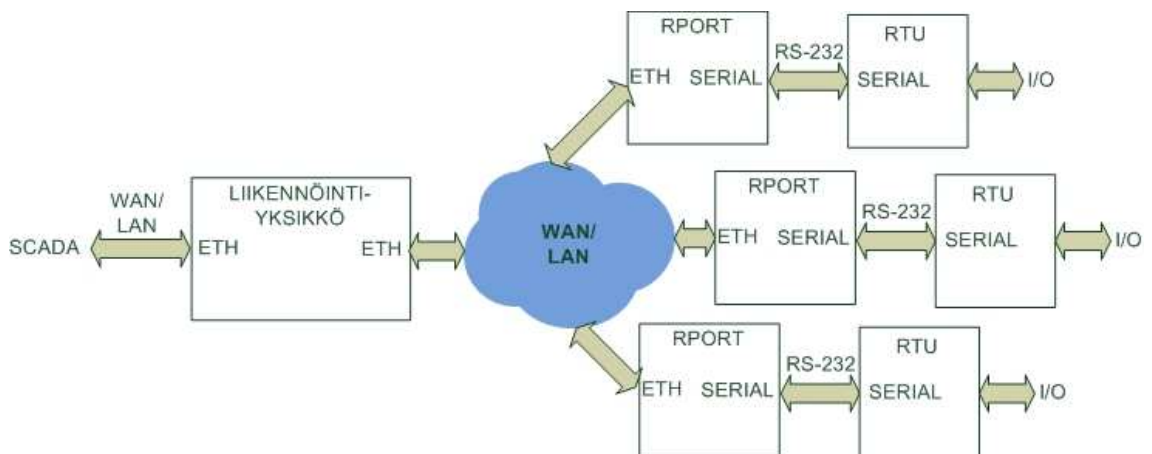
4.2 Testisovellus

Sovelluksen idea oli luoda eri mikrokontrollereiden kehitysalustoille niin sanottu Remote Port -sovellus, joka toimii Ethernet/sarjaportti-muuntimena. Kuvassa 10 on esitetty järjestelmä, jossa ala-asema liitetään sarjaportin kautta liikennöintiyksikköön.



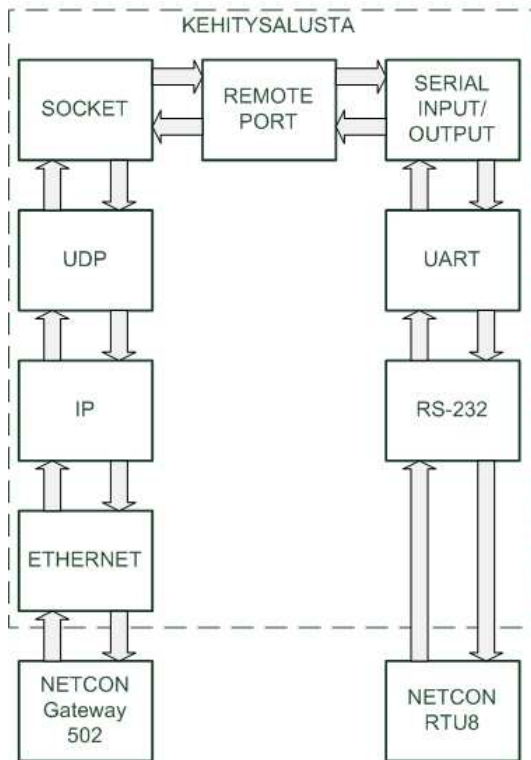
Kuva 10. Sarjaportin avulla liikennöintiyksikköön liitetty RTU.

Kuvan 11 järjestelmässä puolestaan käytetään Ethernet-porttia ala-asemien liittämiseen. Remote Port (RPORT) -sovellus toteuttaa Ethernet-portin ja sarjaportin välisen muunnoksen. Kuvan 11 järjestelmässä voidaan täten liittää useita ala-asemia Ethernet-portin kautta liikennöintiyksikköön. Testisovelluksessa rakennettiin kuvan 11 tilanne käyttäen liikennöintiyksikkönä GW502:ta ja ala-asemana yhtä RTU8:aa.



Kuva 11. Ethernet-portin avulla liikennöintiyksikköön liitetyt RTU:t.

Testisovelluksessa kehitysalustan mikrokontrollerin oli tarkoitus vastaanottaa UDP-viestejä Ethernet-portin kautta GW502:lta ja lähettää vastaanottamansa data eteenpäin sarjaportin kautta RTU8 ala-asemalle. Sovelluksen tuli myös kyetä toimimaan päinvastoin eli lähettämään sarjaportista RTU8:lta saamansa data Ethernet-portista UDP-viesteinä eteenpäin GW502:lle. GW502 oli asetettu lähettämään säännöllisin väliajoin kysely RTU8:lle. Jos RTU8:n I/O-tiloissa oli tapahtunut muutoksia, niin tieto muutoksista lähetettiin GW502:lle, muussa tapauksessa lähetettiin viesti, että muutoksia ei ollut tapahtunut. GW502 ja RTU8 käyttivät tiedonsiirrossa IEC 60870-5-101-standardin mukaista liikennöinti-protokollaa. Testisovelluksen toteutuksessa hyödynnettiin prosessorivalmistajien tarjoamia valmiita TCP/IP-pinoja sekä Netcontrolista löytyvää valmiita puskurointimallia UART:lle. Sovelluksen perusajatus on esitetty lohkokatasolla kuvassa 12.



Kuva 12. Testisovelluksen lohkokaavio.

Kaikkien testattujen kehitysalustojen TCP/IP-pinoesimerkit toimivat reaaliaikakäyttäjärjestelmien päällä. Kaikilla mikrokontrollereilla testisovellus toteutettiin samaa ajatusta käyttäen. Reaaliaikakäyttäjärjestelmä suorittaa kolmea eri tehtävää vuorotellen. Yksi tehtävistä vastaanottaa UDP-viestejä, toinen lähettää niitä ja kolmas lukee sarjaportista ja kirjoittaa sarjaporttiin, jos dataa on luettavissa tai kirjoitettavissa. Eri tehtäviä täytyy suorittaa riittävän nopeassa tahdissa. Tehtävät tulee olla ohjelmoitu siten, että missään tehtävässä ei ole ehtoa, joka pysäyttää tehtävän suorituksen. Edellä mainittujen tehtävien lisäksi reaaliaikakäyttäjärjestelmä suorittaa mikrokontrollerista riippuen muita välttämättömiä tehtäviä, kuten esimerkiksi IP-pinon ylläpitoa.

Testauksessa hyödynnettiin Netcontrolin kehittämää Windowsissa toimivaa NSA (Netcon Serial Analyzer) -sovellusta. NSA:lla pystyttiin seuraamaan GW502:n lähettämiä ja vastaanottamia UDP-viestejä. Tämän avulla saatiin varmistus siitä, että testisovellus oikeasti kykenee vastaanottamaan ja lähettämään dataa halutulla tavalla. Testisovelluksien toteutuksista eri mikrokontrollereilla ja ohjelmistokehitysympäristöillä kerrotaan luvussa viisi.

5 Ohjelmistokehitysympäristöt

5.1 Vaatimukset ja vaihtoehdot

Analog Devicesin, Atmelin ja Freescalen kehitysalustojen mukana tuli kaikissa valmistajan tarjoama ohjelmistokehitysympäristö. Analog Devicesin Blackfin-mikrokontrolleri oli ainoa, jolle ei löytynyt lainkaan ulkopuolisen yrityksen tarjoamaa ohjelmistokehitysympäristöä. Useat ympäristövaihtoehdot ovat mikrokontrollerin pitkää käyttöikää silmällä pitäen hyvä asia. Näin ollen pystytään tarvittaessa vaihtamaan toiseen ohjelmistokehitysympäristöön, jos tarkoitukseen valittu ympäristö käytössä osoittautuukin heikoksi tai esimerkiksi valmistajan tuki tuotteelle loppuu. Tarkoituksena kuitenkin on heti löytää mahdollisimman hyvä ympäristö.

Tärkeimmät vaatimukset ohjelmistokehitysympäristölle ovat helppokäyttöisyys ja monipuoliset virheentestausominaisuudet. Ohjelmakoodissa esiintyvien muuttujien arvoja tulisi pystyä helposti tarkkailemaan. Ohjelmakoodia pitää pystyä ajamaan eteenpäin rivi kerrallaan ja siihen pitää myös pystyä asettamaan keskeytyspisteitä, joissa ohjelman suoritus pysäytetään.

Ohjelmistokehitysympäristöjen vertailu pelkkien kirjattujen ominaisuuksien perusteella on kuitenkin hankalaa. Niinpä opinnäytetyössä kokeiltiin käytännössä kehitysalustojen mukana tulleita ohjelmistokehitysympäristöjä sekä yhtä ulkopuolisen valmistajan tarjoamaa ympäristöä. Analog Devices:lta testattavana oli VisualDSP++ 5.0 ja Freescalelta Codewarrior Development Studio. Atmelin kehitysalustaa testattiin kahdella eri ohjelmistokehitysympäristöllä. Atmelin omalla AVR32 Studio 2.0:lla ja IAR Embedded Workbenchillä. Eri ympäristöistä kerrotaan tarkemmin tulevissa luvuissa.

Kaikki testattavana olleet ohjelmistokehitysympäristöt ovat kokonaisuuksia, joissa sovellukseen kuuluvien tiedostojen editoiminen, kääntäminen, yhdistäminen, sekä mikrokontrollerille siirtäminen ja virheentestaus on yhdistetty saman ympäristön alle.

Näin ollen ohjelmakehitysympäristöistä voidaan käyttää myös nimitystä IDE. Ohjelmistokehitysympäristön toimintaperiaate on se, että kääntäjällä käännetään yksi tai useampi ohjelmakoodia sisältävä tiedosto. Useat tiedostot yhdistetään samaksi tiedostoksi linkittimen avulla. Tiedostojen yhdistämisen seurauksena syntynyt tiedosto voidaan siirtää suoritettavaksi mikrokontrollerin muistiin. Kun sovellusta suoritetaan mikrokontrollerin muistista virheentestauslaitteessa, voidaan ohjelmistokehitysympäristön ja erillisen virheentestauslaitteen avulla tarkkailla sovelluksen suoritusta ja muuttujien arvoja.

Kaikista testatuista ohjelmistokehitysympäristöistä löytyi hyvin samankaltaiset perusominaisuudet. Atmelin AVR32 Studio 2.0 on testatuista ympäristöistä ainoa, joka on täysin ilmainen. Ympäristön dokumentaatio (käyttöohjeet) AVR32 Studiolla oli kuitenkin kaikista niukin. Parhaat dokumentaatiot löytyivät VisualDSP:lle ja IAR:n ympäristölle. Molempien tarjoama dokumentaatio oli todella laaja. Analog Devicesin VisualDSP++ 5.0 sisältää muista ympäristöistä poiketen myös digitaalisen signaalinkäsittelyyn suunnattuja ominaisuuksia.

5.2 VisualDSP++ 5.0

Analog Devicesin ADSP-BF537 EZ-KIT LITE -kehitysalustan mukana tuli VisualDSP++ 5.0. Kyseinen integroitu ohjelmistokehitys- ja virheentestausympäristö on tarkoitettu Analog Devicesin digitaalisten signaaliprosessoreiden ohjelmistokehitykseen. Kehitysalustan mukana tulleessa versiossa on 90 päivän ajan täysi lisenssi. 90 päivän kuluttua käytön aloittamisesta ympäristön simulaattoria ja emulaattoria ei voi enää käyttää. Myös mikrokontrollerin ohjelmamuistista on tämän jälkeen ohjelmoitavissa enää 25 %, ellei ohjelmistokehitysympäristöön hankita uutta lisenssiä.

Ohjelmistokehitysympäristön simulaattori on tarkoitettu ohjelmistokehitykseen, jossa ei käytetä lainkaan mikrokontrolleria. Emulaattoria puolestaan käytetään mikrokontrollerin virheentestaukseen erillisen virheentestauslaitteen välityksellä. Kehitysalusta ei sisältänyt virheentestauslaitetta, joten ympäristöstä käytettiin

Evaluation-versiota, jossa kehitysalustan mikrokontrolleria testattiin kehitysalustaan suoraan liitettävän USB-kaapelin välityksellä.

Testisovelluksessa ei tarvinnut käyttää Blackfin-mikrokontrollerin digitaaliseen signaalinkäsittelyyn liittyviä ominaisuuksia. Tästä johtuen myös ohjelmisto-kehitysympäristön digitaaliseen signaalinkäsittelyyn suunnatut ominaisuudet jäivät opinnäytetyössä käytännön tasolla testaamatta. Analog Devicesin tarjoamat käyttöohjeet VisualDSP++ 5.0:lle ovat laajat ja hyvät.

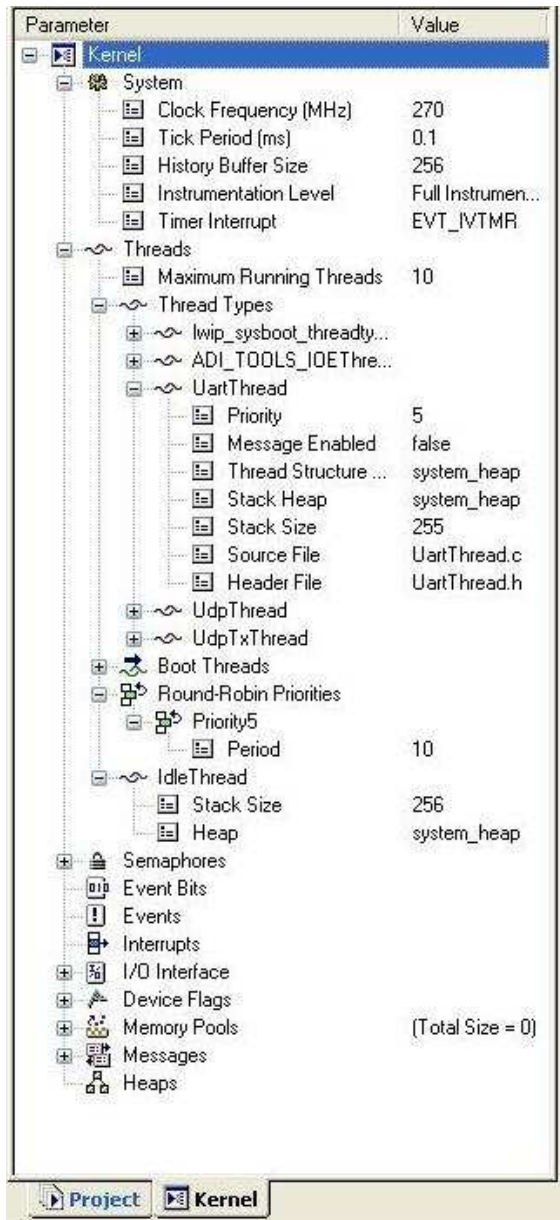
Perusominaisuudet

VisualDSP++ 5.0:ssa voidaan käyttää C- ja C++-ohjelmointikieliä. Myös ohjelmointi suoraan assembly-kielellä on mahdollista. Assembly-kielisten käskyjen lisääminen C-koodin sekaan onnistuu myös. Ohjelmistokehitysympäristön linkittimen tekemän tiedoston avulla voidaan sovellusta suorittaa RAM-muistista. Jos sovellus halutaan ladata Flash-muistiin, joka mahdollistaa sovelluksen suorittamisen myös ilman USB-liitäntäistä virheentestauskaapelia, täytyy lataajan avulla tehdä tiedosto, jonka mikrokontrolleri osaa käynnistää Flash-muistista. Ohjeet sovelluksen lataamiseen kehitysalustan Flash-muistille olivat heikot ja vaativat hieman enemmän asiaan perehtymistä ennen lataamisen onnistumista. (18, s. 38.)

Uuden sovelluksen ohjelmointi VisualDSP++ 5.0:ssa aloitetaan projektin luomisella. Projektien luominen tapahtuu projektivelhon avulla. Valittavana on neljä vaihtoehtoista sovellustyyppiä: perussovellus, kirjastosovellus, lwIP Ethernet -sovellus ja VDK-sovellus. VDK on lyhennys VisualDSP++ Kernelistä, joka on Blackfin-mikrokontrollerissa toimiva reaaliaikakäyttöjärjestelmä. Projektivelhoon syötetään projektiin liittyviä perustietoja, kuten esimerkiksi mikrokontrollerimalli ja ohjelmointikieli. Kyseisiä tietoja voidaan myös jälkeinpäin helposti muokata. (18, s. 80–83.)

Projektiin kuuluvat tiedostot löytyvät puumaisesti kansioihin jaoteltuina projekti-ikkunan alta. Projekti-ikkunan kautta on mahdollista lisätä ja poistaa projektiin kuuluvia

tiedostoja. Projekti-ikkunaan voidaan samanaikaisesti myös avata useita eri projekteja. Eri projekteja voidaan yhdistää luomalla niiden välille riippuvuussuhteita. Jos projektiin kuuluu VDK, niin sen ominaisuuksia on helppo muokata projekti-ikkunan Kernel-välilehdeltä. Kernel-välilehdellä, joka on esitetty kuvassa 13, voidaan esimerkiksi luoda uusia tehtäviä, joita reaaliaikakäyttöjärjestelmässä suoritetaan. (18, s. 100–114.)



Kuva 13. Näkymä VisualDSP++ 5.0:n projekti-ikkunan Kernel-välilehdeltä.

VisualDSP++ 5.0 luo automaattisesti uudelle projektille Debug- ja Release-konfiguraatiot. Omien konfiguraatioiden lisääminen on helppoa. Eri konfiguraatioiden projektiasetukset voidaan muuttaa erilaisiksi. (18, s. 92–93.)

Ohjelmakoodin kirjoittaminen tapahtuu editori-ikkunassa. C-kielellä ohjelmoitaessa editori-ikkunassa näkyy oletuksena ohjelmoijan sinne kirjoittama C-koodi. Halutessaan VisualDSP++ 5.0:ssa voidaan kuitenkin saada samaan editori-ikkunaan näkyviin myös sovelluksen kääntämisen seurauksena syntynyt assembly-koodi. Kuvassa 14 while-lauseke on C-koodia ja sitä seuraavat rivit esittävät kääntäjän tuottamaa assembly-koodia kyseisestä C-koodista. (18, s. 118–119.)

```

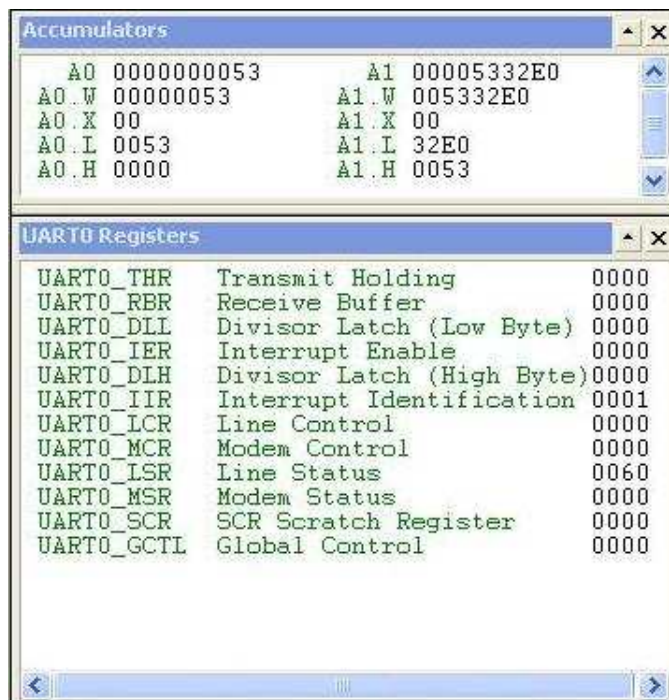
// try sending bytes to UART
while ((*pUART0_LSR & 0x20) && nTxBufCount > 0)
[300DB4E] R0 = W [ P5 ] ( Z ) ;
[300DB50] CC = ! BITTST ( R0 , 0x5 ) ;
[300DB52] IF CC JUMP 76 /*0x300DB9E*/ ;
[300DB54] P2.L = 0x67a0 ;
[300DB58] P2.H = 0xff80 ;
[300DB5C] R0 = [ P2 ] ;
[300DB5E] CC = R0 <= 0 ;
[300DB60] IF CC JUMP 62 /*0x300DB9E*/ ;
{

```

Kuva 14. Esimerkki editori-ikkunan näkymästä

Sovelluksen kääntäminen onnistuu VisualDSP++ 5.0:ssa yhdellä napin painalluksella. Kääntämisen onnistumisesta, virheistä ja varoituksista saadaan tuloste niille varattuun ikkunaan. Virheilmoitusta napsauttamalla päästään suoraan hypättyä oikeaan tiedostoon riville, jossa virhe tapahtui. Tulosteikkunaan voidaan avata myös konsolinäkymä. Konsolinäkymässä voidaan esimerkiksi nähdä C- ja C++-standardikirjastojen tulostusfunktioiden tulosteita (18, s. 128).

Blackfin-mikrokontrollerin eri rekistereiden tarkkailu on helppoa VisualDSP++ 5.0:ssa. Mikrokontrollerin ja sen oheislaitteiden rekisterit ja niiden arvot saadaan helposti näkyviin kehitysympäristöön. Kuvassa 15 on esimerkki ADSP-BF537-mikrokontrollerin akkujen ja UART0-sarjaliikenneväylän rekistereistä. Ohjelmistokehitysympäristössä on myös mahdollista luoda oma rekisteri-ikkuna ja poimia sinne haluamiaan rekistereitä tarkkailtavaksi. (18, s. 176–177, 179.)



Kuva 15. Näkymä akkujen ja UART0-rekistereiden arvoista.

Rekisteri-ikkunoiden lisäksi VisualDSP++ 5.0:ssa on paljon muita sovelluksen virheentestauksessa hyödyllisiä ikkunoita. Funktioiden paikallisia muuttujia voidaan tarkkailla Locals-ikkunassa. Sovelluksen yleismuuttujat voidaan puolestaan lisätä Expressions-ikkunaan. Muuttujista näytetään oletuksena niiden nimi ja arvo. Muuttujista saadaan myös tarvittaessa näkyviin osoite, tyyppi, koko ja muoto. Muuttujien arvot eivät kuitenkaan päivitty reaaliaikaisesti, vaan sovelluksen suoritus täytyy aina keskeyttää päivitystä varten. VisualDSP++ 5.0:n emulaattorissa on kuitenkin olemassa myös BTC (Background Telemetry Channels) -ominaisuus, jonka avulla pystytään JTAG-liitännän kautta seuraamaan muistia reaaliajassa. BTC vaatii toimiakseen erillisen virheentestauslaitteen. (18, s. 148–150, 152–153, 171.)

Ohjelmakoodiin saadaan VisualDSP++ 5.0:ssa lisättyä keskeytyspisteitä. Kun ohjelmakoodi on suoritettu keskeytyspisteeseen asti, sen suoritus pysäytetään. Keskeytyspisteeseen voidaan myös asettaa pysäytysehtoja. Ehto voi olla esimerkiksi se, että sovelluksen suoritus pysäytetään vasta kun keskeytyspisteeseen saavutaan viidennen kerran. Keskeytyspisteiden lisäksi VisualDSP++ 5.0:ssa on mahdollista luoda tarkistuspisteitä. Tarkistuspisteessä ohjelmakoodin suoritus keskeytetään kun

esimerkiksi ohjelmoijan määrittämä rekisteri saa tietyn arvon. Tarkistuspisteet ovat kuitenkin käytössä vain simulaattorissa. Keskeytyspisteen jälkeen ohjelmakoodin suoritus voidaan käynnistää uudelleen tai koodia voidaan suorittaa askel kerrallaan. Askel voi olla yksi C-, C++- tai assembly-kielinen rivi kerrallaan. (18, s. 224, 231–234.)

VisualDSP++ 5.0:ssa on paljon, varsinkin digitaaliseen signaalinkäsittelyyn liittyviä, ominaisuuksia joita ei tämän opinnäytetyön puitteissa ehditty testata. Ohjelmisto mahdollistaa esimerkiksi useiden erilaisten kuvaajien piirtämisen.

Testisovelluksen toteutus VisualDSP++ 5.0:lla

Blackfin-mikrokontrolleri oli ensimmäisenä testausvuorossa. Testisovelluksen rakennus aloitettiin tekemällä lwIP-pino VisualDSP++ 5.0:n projektivelholla. lwIP-pino on sulautettuihin järjestelmiin kehitetty avoimen lähdekoodin TCP/IP-pino. Blackfin-mikrokontrollerissa lwIP-pino oli tehty toimimaan vain VDK:n päällä. VDK:sta johtuen ohjelmakoodin virheentestaus askel kerrallaan oli hankalaa, koska VDK suorittaa paljon reaaliaikakäyttöjärjestelmän ylläpitämiseen liittyvää ohjelmakoodia. VisualDSP++ 5.0:n mahdollisuus tulostaa konsoli-ikkunaan auttoi virheentestauksessa huomattavasti varsinkin testisovelluksen alkuvaiheessa ennen kuin UARTin ajurit saatiin liitettyä projektiin.

VDK vaati perehtymistä, koska aluksi piti selvittää miten omaa ohjelmakoodia voidaan lisätä VDK-projektiin. Dokumentointi VDK:lle on kuitenkin hyvä ja varsin pian VDK:n perusajatus selvisi. IP-pino osoittautui heti IP-osoitteen asettamisen jälkeen toimivaksi, sillä kehitysalustaa pystyttiin pingaamaan PC:ltä. Myös UDP-viestien lähettäminen ja vastaanotto onnistui varsin nopeasti pitkälti VisualDSP++ 5.0:n mukana tulleiden esimerkkien ansiosta.

ADSP-BF537:n UARTille ei löytynyt valmista testisovelluksessa käyttökelpoista ajuria, joten ajuri päätettiin tehdä itse. Kun yksinkertainen UARTin ajuri saatiin liitettyä lwIP-projektiin, niin voitiin todeta että luvussa 4.2 esitelty testisovellus toimi ainakin kohtuullisesti. GW502 joutui välillä lähettämään UDP-viestinsä kaksi kertaa ennen kuin

siihen saatiin vastaus RTU8:lta, mutta muuten data kulki halutulla tavalla. Testi-sovellusta ei kuitenkaan viimeistelty, koska opinnäytetyön aikataulun kannalta tärkeämpää oli ehtiä testata kaikkia kehitysalustoja.

Kaiken kaikkiaan VisualDSP++ 5.0:sta jäi varsin hyvä kuva. Ohjelmistokehitysympäristö on selkeä käyttää. Siihen ja VDK:hon löytyy hyvät dokumentit ja niiden avulla ympäristöstä pääsee hyvin perille. Huomattava ero muihin ohjelmistokehitysympäristöihin verrattuna VisualDSP++ 5.0:ssa on, että reaaliaikakäyttöjärjestelmä on todella hyvin integroitu ympäristöön ja näin sen käyttäminen on helpompaa. Muut prosessorivalmistajat käyttävät TCP/IP-esimerkeissä avoimen lähdekoodin periaatteella toimivia reaaliaikakäyttöjärjestelmiä, jotka eivät ole valmistajien itsensä kehittämisiä.

5.3 Codewarrior Development Studio

Freescalelta testattavan oli M52235EVB-kehitysalusta ja sen mukana tullut Freescalen mikrokontrollereille tarkoitettu ohjelmistokehitysympäristö, joka on nimeltään Codewarrior Development Studio. Kehitysalustan mukana tulleen Codewarriorin version ja kehitysalustaan liitettävän virheentestauslaitteen kanssa ilmeni ongelmia jo ennen kuin niitä päästiin edes käyttämään. Codewarrior ilmoitti projektia käännettäessä viallisesta lisenssistä, vaikka kyseiseen ohjelman versioon ei tarvitse lisenssiä. Lopulta Codewarriorista haettiin valmistajan verkkosivuilta uusi versio, joka saatiin toimimaan. Virheentestauslaite ei puolestaan suostunut ohjelmoimaan mikrokontrolleria ennen kuin sille haettiin uusimmat ajurit valmistajan, P&E Microcomputer Systemsin, verkkosivuilta. Codewarriorin testikäytössä ollut versio mahdollistaa ohjelmoinnin vain C- ja assembly-kielillä. Koodin koko on myös rajoitettu 128 kilotavuun. Codewarriorista on saatavilla kolme eri maksullista versiota. Maksullisissa versioissa ohjelmistokehitysympäristöön saadaan lisää ominaisuuksia ja rajoitukset koodin määrässä poistuvat.

Perusominaisuudet

Myös Codewarriorissa ohjelmointi tapahtuu projektien kautta. Uusi projekti luodaan projektivelhon avulla. Projekteja voidaan luoda myös make-tiedostoja hyödyntäen. Codewarriorissa on mahdollista yhdistää projekteja liittämällä varsinaiseen projektiin aliprojekteja. Projektivelhossa syötetään aluksi perustietoja, kuten ohjelmointikieli, mikrokontrollerimalli ja projektiin mahdollisesti liitettävät tiedostot. Projektivelhossa on mahdollista suorittaa mikrokontrollerin oheislaitteille ja keskeytyksille alustustoimenpiteitä. Tässä tapauksessa projektiin liitetään valmiit alustuskoodit ohjelmoijan valitsemille oheislaitteille. Tätä ominaisuutta kokeiltiin testisovelluksen yhteydessä hyvällä menestyksellä UARTin alustuskoodin tekemiseen. Automaattisten alustuskoodien lisäksi on mahdollista luoda oheislaitteille valmiita ajureita.

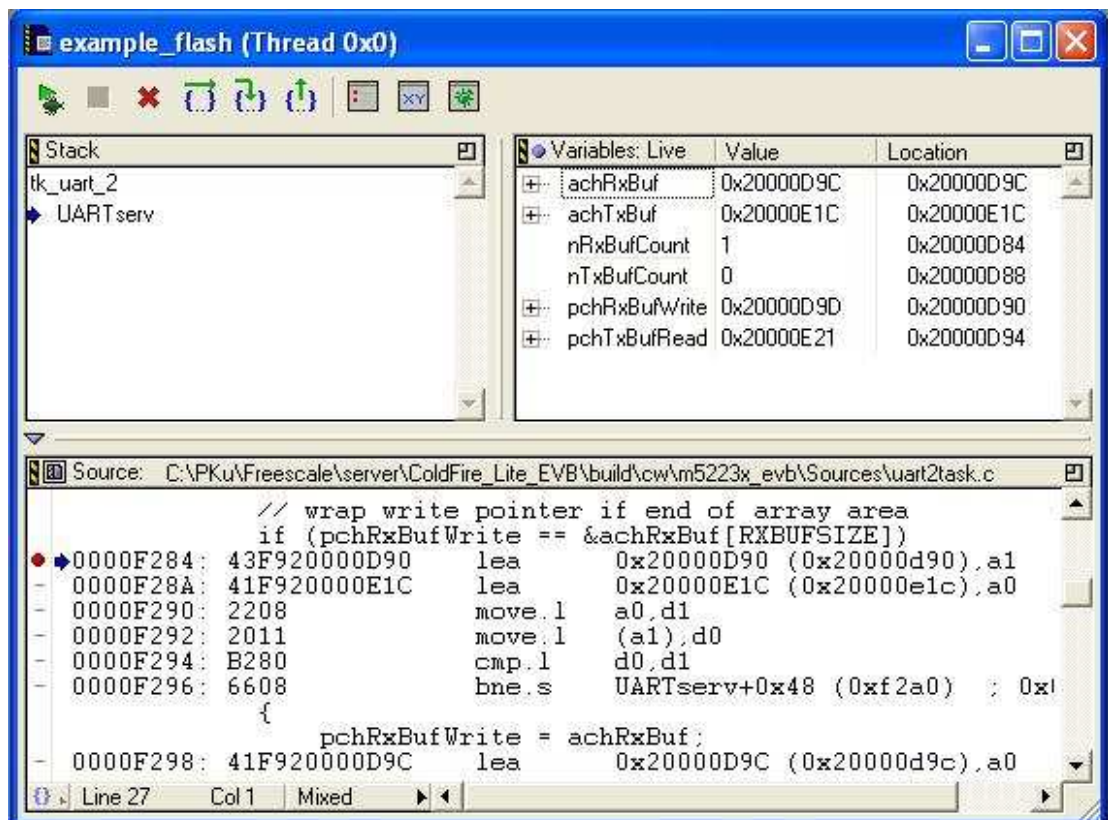
Projektia hallitaan projekti-ikkunan kautta. Projekti-ikkuna on jaettu kolmeen välilehteen. Projektiin kuuluvat tiedostot löytyvät puumaisesti kansioista projekti-ikkunasta Files-välilehdeltä. Uusien tiedostojen ja kansioiden lisääminen projektiin onnistuu samalta välilehdeltä. Link Order -välilehdeltä nähdään järjestys, jossa projektiin kuuluvat tiedostot yhdistetään. Yhden projektin sisällä voi olla useita erilaisia projektiasetuksia. Erilaiset asetukset esitetään projekti-ikkunan Targets-välilehdellä. Projekti-ikkunasta löytyy myös pikanäppäimet projektin kääntämiseen ja yhdistämiseen, virheentestaustilan avaamiseen ja projektin suorittamiseen. (19, s. 41–47.)

Jos projektin kääntämisen yhteydessä on ilmennyt virheitä tai varoituksia, niin Codewarrioriin aukeaa erillinen ikkuna, jossa virheet tai varoituksen esitetään. Virhettä napsauttamalla päästään siirtymään virheen sisältäneen tiedoston riville, jossa virhe ilmeni. Codewarriorilla voidaan suorittaa koodia RAM- tai Flash-muistilta. Sovelluksen ohjelmointi M52235EVB-kehitysalustan mikrokontrollerin Flash-muistille on Codewarriorissa huomattavasti helpompaa kuin VisualDSP++ 5.0:ssa.

Ohjelmakoodin virheentestaus tapahtuu Codewarriorissa virheentestausikkunan kautta. Virheentestausikkunan kautta voidaan ohjelmakoodin suoritus käynnistää ja pysäyttää.

Pysäytyksen jälkeen voidaan ohjelmakoodia suorittaa C- tai assembly-kielinen rivi kerrallaan. Kun virheentestausikkuna on avattu, voidaan avata muita testauksessa hyödyllisiä ikkunoita, kuten esimerkiksi rekisteri-ikkuna tai muisti-ikkuna.

Ohjelmakoodin suorituksen pysäytyessä, joko ohjelmoijan toimesta tai sen saavuttaessa keskeytyspisteen, näkyy virheentestausikkunassa pysäytyskohdan ohjelmakoodi ja suoritettavana olevan funktion paikalliset muuttujat, yleismuuttujat ja näiden arvot. Jos ohjelmakoodi on kirjoitettu C-kielillä, niin se voidaan näyttää virheentestausikkunassa myös assembly-kielisenä tai sekamuodossa, jossa C-koodia seuraa kääntäjän tuottama assembly-vastine. Kuvan 16 virheentestausikkunassa C-kielistä if-lausetta seuraavat kuusi riviä ovat assembly-koodia. Kuvassa 16 näkyy myös lista UARTserv-funktiossa käytettävistä paikallisista muuttujista ja yleismuuttujista. (19, s. 187–194.)



Kuva 16. Näkymä Codewarriorin virheentestausikkunasta.

Ohjelmistokehitysympäristön käyttäjä voi lisätä ohjelmakoodiin keskeytyspisteitä, tarkistuspisteitä ja tapahtumapisteitä, jotka keskeyttävät ohjelmakoodin suorittamisen. Codewarriorissa on käytössä kolme erilaista keskeytyspistettä: tavallinen, ehdollinen ja

väliaikainen. Tavallinen keskeytyspiste pysäyttää ohjelmakoodin suorituksen kyseiselle riville saavuttaessa. Ehdollinen keskeytyspiste tekee samoin, mutta vain jos pysäytykselle asetettu ehto täyttyy. Väliaikainen keskeytyspiste keskeyttää ohjelmakoodin suorituksen, mutta häviää yhden pysäytyskerran jälkeen, eli saavutessa uudelleen samaan ohjelmakoodin kohtaan suoritusta ei enää pysäytetä. (19, s. 203–214.)

Tapahtumapisteet eroavat keskeytyspisteistä siten, että kun ohjelmakoodin suorituksessa kohdataan tapahtumapiste, niin suoritusta ei pysäytetä. Tapahtumapisteiden avulla voidaan esimerkiksi ajaa skriptejä, kerätä dataa tai hypätä suoraan rivin yli suorittamatta siinä olevaa ohjelmakoodia. Tarkistuspuisteet pysäyttävät ohjelmakoodin suorituksen keskeytyspisteiden tapaan, mutta pysäytyksen ehtona on se, että tarkistuspuisteeseen määrittämään muistialueeseen kirjoitetaan tai sieltä luetaan dataa. Kehitysalustan mikrokontrollerille pystyttiin asettamaan samanaikaisesti vain yksi tarkistuspuiste. (19, s. 214–228.)

Testisovelluksen toteutus Codewarrior Development Studio:lla

Freescalen kehitysalustan mukana tuli esimerkkiprojekti, jossa käytettiin Interniche Technologiesin kehittämää ilmaista TCP/IP-pinoa ja NicheTask-reaaliaikakäyttöjärjestelmää. TCP/IP-pinon dokumenteissa kerrottiin, että pino voidaan saada toimimaan reaaliaikakäyttöjärjestelmän kanssa tai silmukalla. Koska esimerkki oli rakennettu kokonaan NicheTaskin päälle ja dokumentaatio silmukan käytöstä oli huono, niin päädyttiin testisovellus rakentamaan NicheTaskia käyttäen. NicheTask ei ole samalla tavalla liitetty Codewarrior-ympäristöön kuin VisualDSP++ 5.0:ssa VDK.

Esimerkissä oli valmiit sovellukset UDP-viestien lähettämiseksi ja vastaanottamiseksi. Suurimmaksi ongelmaksi muodostui se, että esimerkkiprojektin eri sovellukset olivat omia kokonaisuuksiaan. Niitä pystyi kääntämään vain yhden kerrallaan, ja ne suorittivat aina samaa NicheTaskin tehtävää. UDP-viestien lähetys ja vastaanotto piti siis saada yhdistettyä samaan sovellukseen eli tehtäviä piti luoda lisää. Lisäksi sovellukseen piti lisätä myös UARTille oma tehtävä. Freescalen TCP/IP-pinon ja NicheTaskin dokumentaatio oli heikompi kuin Analog Devicesin tarjoama dokumentaatio.

Kehitysalustan esimerkkiprojektille löytyi vain muutama suppea dokumentti. Internichen verkkosivuilla olleista dokumenteista ei myöskään ollut apua. Selvitystyön jälkeen projektiin saatiin kuitenkin lisättyä tehtäviä, joten UDP-viestien lähetys ja vastaanotto sekä UART saatiin toimimaan samassa sovelluksessa. UART-tehtävässä käytettiin samaa puskurointimallia kuin Blackfin-mikrokontrollerin testisovelluksessa.

M52235EVB-kehitysalustalla tehty testisovellus toimi kokeiltaessa paremmin kuin ADSP-BF537 EZ-KIT LITE -kehitysalustalla tehty sovellus, koska GW502 sai nyt jokaiselle lähettämälleen kyselylle vastauksen RTU8:lta. Usean vuorokauden mittaisen kokeilun yhteydessä havaittiin kuitenkin, että mikrokontrollerin suorittama testisovellus lakkaa toimimasta. Tämän ongelman ratkaisemiseen ei kuitenkaan opinnäytetyön yhteydessä käytetty aikaa.

Codewarrior Development Studiosta löytyi suunnilleen kaikki samat ominaisuudet kun kaikista muistakin ohjelmistokehitysympäristöistä. Testisovelluksen virheentestauksessa käytetty ohjelmakoodin suorittaminen askel kerrallaan toimi Codewarriorilla tehdyssä sovelluksessa paremmin kuin muiden ympäristöjen kanssa. Mikrokontrollerin Flash-muistia tyhjennettäessä tapahtui Codewarriorissa usein virhe, joka vaati ohjelmistokehitysympäristön uudelleen käynnistämisen, jotta muisti saatiin tyhjennettyä.

5.4 IAR Embedded Workbench

Atmelin EVK1100-kehitysalustaa kokeiltiin kahdella eri ohjelmistokehitysympäristöllä. Toinen näistä on IAR:n tarjoama ohjelmistokehitysympäristö IAR Embedded Workbench. IAR kehittää ohjelmistokehitysympäristöjä useille eri mikrokontrollereille, eikä se kuulu minkään piirivalmistajan alaisuuteen. AT32UC3A0512-mikrokontrollerin lisäksi testattavina olleista mikrokontrollereista Freescalen MCF52235 on tuettu IAR:n toimesta. Muista kartoituksessa mukana olleista mikrokontrollereista myös NXP:n ja Luminary Micron mikrokontrollereille on saatavilla IAR Embedded Workbench.

IAR Embedded Workbenchiä testattiin ilmaisen 30 päivän kokeiluversion kanssa. IAR:n ohjelmistokehitysympäristöstä on AVR32-mikrokontrollerille saatavilla kolme

ominaisuuksiltaan hieman poikkeavaa versiota. AT32UC3A0512-mikrokontrolleria pystytään IAR Embedded Workbenchissä ohjelmoimaan ja testaamaan käyttäen samaa AVR32 JTAGICE mkII -virheentestauslaitetta, joka toimii myös AVR32 Studion kanssa. Atmel tarjoaa AT32UC3-sarjan mikrokontrollereille varsin kattavan ajuripaketin ja esimerkkiprojekteja. Kaikista ajureista ja esimerkeistä on olemassa IAR Embedded Workbenchin kanssa yhteensopivat versiot.

Perusominaisuudet

IAR:n ohjelmistokehitysympäristössä on mahdollista kehittää sovelluksia C-, C++- tai assembly-kielillä. Assembly-kielisiä käskyjä voidaan myös lisätä esimerkiksi C-koodin sekaan. IAR:llä luotavat sovellukset muodostuvat projekteista. Projekteja hallitaan työtilan avulla. Työtilassa näkyvät puumaisesti projektiin kuuluvat tiedostot. Uusien tiedostojen lisääminen projektiin voidaan suorittaa esimerkiksi työtilasta. Projekti voidaan liittää useampaan kuin vain yhteen työtilaan. Jos esimerkiksi kahdessa eri projektissa tarvitaan samoja tiedostoja, voidaan yhteisistä tiedostoista tehdä oma projekti joka liitetään kahteen eri työtilaan. Jos yhteisiin tiedostoihin tehdään muutoksia, niin ne päivittyvät molempiin projekteihin niiden kääntämisen yhteydessä. (20, s. 108.)

Täysin uuden projektin luominen IAR:llä eroaa hieman muista testatuista ohjelmistokehitysympäristöistä, koska IAR:ssä ei ole edellä mainittujen kaltaista projektivelhoa, johon syötetään asetuksia. IAR:llä uuden projektin luomisessa riittää että valitaan ohjelmointikieli ja projektikansio. Projektin asetuksia päästään muokkaamaan projektin luomisen jälkeen. Uuden projektin luomisessa tehdään projektille automaattisesti Debug- ja Release-asetukset. Omien projektiasetusten luominen onnistuu myös helposti. (19, s. 109–113.)

IAR Embedded Workbench mahdollistaa projektin kääntämisen IDE-ympäristön lisäksi myös komentoriville syötettävien käskyjen avulla. IDE-ympäristöä käytettäessä projektin kääntämisessä havaituista virheistä ja varoituksista saadaan ilmoitus erilliseen

ikkunaan. Ilmoitusta napsauttamalla päästään tässäkin ympäristössä siirtymään suoraan koodiin riville, jossa esimerkiksi virhe on havaittu. (20, s. 120.)

IAR:n ohjelmistokehitysympäristö käyttää testauksessa C-SPY-virheentestausohjelmaa. Sovellusta voidaan C-SPY:llä testata joko simulaattoria tai mikrokontrolleria ja virheentestauslaitetta käyttäen. Testaus tapahtuu samasta editori-ikkunasta, jossa ohjelmakoodia muokataan. Sovellusta voidaan suorittaa askel kerrallaan neljää erilaista askellusmenetelmää käyttäen. Yhden askeleen voi muodostaa esimerkiksi C- tai assembly-kielinen rivi. Sovellus voidaan tietysti myös käynnistää normaalisti suoritettavaksi, jolloin se voidaan pysäyttää erillisellä pysäytyspainikkeella tai keskeytyspisteillä. Keskeytyspisteitä voidaan asettaa ohjelmakoodiriveille tai muistiosoitteille. Keskeytyspisteille voidaan määrittää erilaisia toteutumisehtoja. Kaikki asetetut keskeytyspisteet listataan ikkunassa, jossa niitä voidaan muokata tai ne voidaan poistaa. (20, s. 155, 164.)

IAR Embedded Workbench sisältää paljon sovelluksen virheentestausta helpottavia ikkunoita. Erilaisiin ikkunoihin voidaan avata esimerkiksi mikrokontrollerin ja siihen integroitujen oheislaitteiden rekistereitä sekä sovelluksessa käytettyjä muuttujia. Watch-ikkunaan voidaan lisätä sovelluksen yleismuuttujat tarkkailtaviksi. Funktioiden paikallisille muuttujille on käytössä oma ikkuna. Kaikkien virheentestausikkunoiden arvot päivittyvät kun sovelluksen suoritus pysäytetään. IAR:n ohjelmistokehitysympäristö mahdollistaa edellisten ympäristöjen tapaan kääntäjän tuottaman assembly-koodin näyttämisen Disassembly-ikkunassa. Tämän ikkunan avulla voidaan sovellusta suorittaa assembly-kielinen rivi kerrallaan.

Testisovelluksen toteutus IAR Embedded Workbenchillä

Atmelin esimerkkisovellusten joukossa oli EVK1100-kokeilualustalle tehty TCP/IP-pinoesimerkki, jonka pohjalta testisovellus rakennettiin. Esimerkki toimii FreeRTOS-reaaliaikakäyttöjärjestelmän päällä. FreeRTOS on ilmainen avoimen lähdekoodin periaatteella rakennettu käyttöjärjestelmä. TCP/IP-pino, jota Atmelin esimerkissä käytettiin, oli sama lwIP-niminen pino, jota käytettiin myös Analog Devicesin Blackfin-

mikrokontrollerin testisovelluksen yhteydessä. Tämän seurauksena Atmelin testisovelluksessa voitiin osittain hyödyntää samoja ohjelmakoodeja UDP-viestien lähettämiseen ja vastaanottamiseen kuin Blackfin-mikrokontrollerin kokeilun yhteydessä.

Kaikista Atmelin esimerkkisovelluksista on olemassa eri versiot IAR Embedded Workbenchille ja AVR32 Studiolle. Eri versioissa ohjelmakoodit ovat samoja, mutta projektitiedostot eroavat, koska käytössä on eri ohjelmistokehitysympäristöt. TCP/IP-pinoesimerkki AVR32 Studiolle piti tehdä make-tiedoston avulla ja tämä aiheutti selvitystyötä, josta kerrotaan luvun 5.5 lopussa. Niinpä Atmelin testisovellusta ruvettiin aluksi tekemään IAR:n ohjelmistokehitysympäristössä.

Atmelin esimerkkisovelluksesta IAR:lle tehdystä versiosta saatiin avattua työtila, josta löytyi lwIPdemo-projekti. Kun projekti käännettiin, niin saatiin selaimen kautta avattua Web Server, jossa näkyi kaikki reaaliaikakäyttöjärjestelmän suorittamat tehtävät. Seuraavassa vaiheessa piti selvittää, kuinka reaaliaikakäyttöjärjestelmään saadaan lisättyä omia tehtäviä ja poistettua turhia tehtäviä. Atmelin lwIPdemo-esimerkin dokumentaatio oli suppea, mutta FreeRTOS oli dokumentoitu paremmin kuin Internichen reaaliaikakäyttöjärjestelmä. Kuitenkaan dokumentaatio ei ollut VDK:n dokumentaation veroinen.

Testisovellus saatiin rakennettua siihen vaiheeseen, että GW502:n lähettämien UDP-viestien välittäminen sarjaportin kautta RTU8:lle onnistui hyvin, mutta toiseen suuntaan data kulki vain yksi tavu kerrallaan. Toteutuksessa auttoi paljon se, että Atmelilta löytyi valmiit UARTin ajurit ja UDP-viestien lähetyksessä ja vastaanotossa voitiin käyttää osittain samaa ohjelmakoodia, jota käytettiin Blackfin-mikrokontrollerin kokeilun yhteydessä. Noin viikon käytön jälkeen IAR:n ohjelmistokehitysympäristön lisenssi ei yhtäkkiä enää ollutkaan voimassa, vaikka kokeiluversiossa piti olla 30 päivän lisenssi. Tässä vaiheessa piti keskeyttää testisovelluksen tekeminen IAR:n ympäristöllä ja alkaa tutkia, miten se saadaan tehtyä AVR32 Studiolla, koska uuden kokeilulisenssin saaminen kesti muutaman päivän. Testisovellus tehtiin loppuun AVR32 Studiolla, mutta

muokatut ohjelmakoodit kopioitiin IAR:llä tehtyyn versioon ja IAR:llä käännetyn sovelluksen toimivuus myös kokeiltiin.

5.5 AVR32 Studio 2.0

AVR32 Studio 2.0 on Atmelin tarjoama ohjelmistokehitysympäristö AVR32-sarjan mikrokontrollereille. AVR32 Studio eroaa aiemmin esitellyistä ohjelmistokehitysympäristöistä eniten siinä, että se on täysin ilmainen ja vapaasti ladattavissa Atmelin verkkosivuilta. Se on rakennettu Eclipsen CDT-ympäristön ja GNU C -kääntäjän (GCC) sekä virheentestausohjelman (GDB) pohjalta. Eclipse ja GNU ovat avoimeen lähdekoodiin perustuvia projekteja, jotka ovat käyttäjille täysin ilmaisia. AVR32 Studiossa on mahdollista ohjelmoida C-, C++- ja assembly-ohjelmointikielillä. Assembly-koodia voidaan myös lisätä esimerkiksi C-koodin sekaan, kuten kaikissa muissakin testatuissa ympäristöissä.

AVR32 Studiolle ei ole, ainakaan vielä, saatavilla samanlaista kattavaa käyttöopasta, joka on saatavilla muille testatuille ohjelmistokehitysympäristöille. AVR32 Studiolle löytyvän pikakäyttöoppaan ja ympäristön Help-valikon avulla pääsee kuitenkin käytössä alkuun.

Perusominaisuudet

Ohjelmistokehitysympäristössä käytetty GCC on komentorivipohjainen kääntäjä, joten projektien kääntäminen onnistuu myös pelkkää komentoriviä käyttäen. IDE-ympäristöä käytettäessä täytyy ympäristön käyttäjän aluksi luoda työtila, johon kaikki projektit tallennetaan. Kaikki työtilaan luodut projektit näkyvät tiedostoineen Project Explorer-ikkunassa. AVR32 Studiossa projektit luodaan projektivelholla. Projektit voidaan tehdä valmiista make-tiedostosta tai siten, että AVR32 Studio luo uudelle projektille make-tiedoston. Työtilassa olevien projektien välille voidaan projektivelhossa luoda riippuvuuksia. Riippuvuuksien luominen onnistuu jälkeinpäin projektin ominaisuuksista.

Projektivelhosta voi myös suoraan luoda esimerkkiprojekteja. Esimerkkiprojektit ovat samoja Atmelin tarjoamia esimerkkejä, jotka olivat saatavilla IAR:n ohjelmistokehitysympäristöön. Mikrokontrollerin oheislaitteiden valmiita ajureita voidaan projektiin lisätä Software Framework Wizardin avulla. Kaikki tuetut oheislaitteet on listattu Wizardiin, ja käyttäjä voi valita haluamansa laitteet, jolloin ajuritiedostot liitetään projektiin ja tiedostojen funktiot ovat ohjelmoijan käytettävissä. Ajureiden lisäksi Software Framework Wizardista voidaan liittää projektiin esimerkiksi avoimen lähdekoodin FreeRTOS reaaliaikakäyttöjärjestelmä.

Projektin kääntämisen yhteydessä ilmenneistä virheistä ja varoituksista saadaan samankaltainen ilmoitus tiedostosta ja rivistä kuin kaikissa muissakin testatuissa ohjelmistokehitysympäristöissä. Ilmoitusta napsauttamalla päästään siirtymään kyseiselle riville. Uudelle projektille luodaan tässäkin ohjelmistokehitysympäristössä automaattisesti Debug- ja Release-asetukset.

AVR32 Studioissa voidaan sovellusta testata simulaattorilla tai mikrokontrollerilla virheentestauslaitteen välityksellä. Kun projekti käynnistetään virheentestaustilassa, saadaan auki erillinen virheentestausnäkyvä. AVR32 Studiosta löytyy muiden testattujen ympäristöjen kanssa hyvin samankaltaiset virheentestausominaisuudet. Erilaisten ikkunoiden avulla voidaan tarkkailla esimerkiksi mikrokontrollerin muistia, rekistereitä ja suoritettavan ohjelmakoodin muuttujia. Disassembly-ikkunassa nähdään kääntäjän tuottama assembly-koodi C-koodin jälkeen. Sovellusta voidaan suorittaa eteenpäin joko C- tai assembly-kielinen askel kerrallaan. Sovelluksen suorittaminen voidaan keskeyttää keskeytyspisteiden ja tarkistuspuisteiden avulla. Keskeytyspisteet pysäyttävät sovelluksen suorituksen kyseiselle riville saavuttaessa ja tarkistuspuisteet, kun ennalta määrättyyn muuttujaan viitataan.

Testisovelluksen toteutus AVR32 Studio 2.0:lla

Atmelin esimerkkisovelluksista kaikki muut paitsi lwIP-esimerkki löytyvät AVR32 Studioissa projektivelhon alta. lwIP-esimerkki täytyy kääntää make-tiedoston kautta.

Dokumentaatio sovelluksen tekemiseksi make-tiedostosta AVR32 Studiossa oli hyvin vähäistä. Make-tiedoston kautta tehtyä sovellusta ei myöskään voida kääntää samalla tavalla kuin muita sovelluksia, vaan kääntämistä ja Flash-muistin ohjelmoimista varten joudutaan tekemään Make Targets -komentoja. Näiden komentojen suorittamisen jälkeen esimerkkisovellus toimii muutamista AVR32 Studion ilmoittamista käännösvirheistä huolimatta.

Kun IAR:n ympäristön kanssa törmättiin lisenssiongelmiin, jatkettiin testisovelluksen tekemistä EVK1100-kehityslustalle AVR32 Studiossa. Testisovellus oli saatu jo osittain toimimaan IAR:n ohjelmistokehitysympäristössä, joten lwIP-esimerkin muokkaaminen AVR32 Studiossa samalle tasolle IAR:n projektin kanssa oli varsin nopea toimenpide sen jälkeen kun alkuperäinen lwIP-esimerkki saatiin aluksi onnistuneesti käännettyä. Testisovellus tehtiin loppuun AVR32 Studiolla. Sovellus toimi luotettavammin usean vuorokauden pituisessa testissä kuin Freescalen kokeilualustalla toteutettu sovellus.

AVR32 Studiossa on hyvin samankaltaiset ominaisuudet kuin muissakin testatuissa ohjelmistokehitysympäristöissä. Ongelmat make-tiedostosta rakennetun projektin kanssa antavat väärän kuvan ympäristöstä, sillä AVR32 Studio ei ole yhtään vaikeakäyttöisempi kuin muutkaan ohjelmistokehitysympäristöt. Hyvän käyttöoppaan puuttuminen tietysti jossain määrin hankaloittaa ympäristön käyttöä aluksi. AVR32 Studioon integroitu Software Framework Wizard vaikutti varsin hyvältä ominaisuudelta. Make-tiedoston kautta tehtyyn projektiin ei pystytä lisäämään Software Framework Wizardin kautta ajureita, vaan ajurit täytyy liittää projektiin tiedosto kerrallaan.

6 Yhteenveto

Opinnäytetyössä ei juurikaan törmätty opinnäytetyön etenemistä haittaaviin suuriin ongelmiin. Suurimmaksi haasteeksi osoittautui opinnäytetyön rajaaminen, koska esimerkiksi mikrokontrollereihin liittyvän dokumentaation laajuus on todella suuri.

Kartoituksen yhteydessä havaittiin, että Netcon100-ala-aseman vaatimukset täyttäviä mikrokontrollereita ei ole tällä hetkellä markkinoilla kovinkaan paljon. Rajoittavaksi tekijäksi osoittautui yleensä integroitu Ethernet-portti. Useilla piirivalmistajilla ei ole Ethernet-portin omaavia mikrokontrollerimalleja lainkaan. Usein myös Ethernet-portti oli integroitu I/O-yksikön kannalta vain turhan tehokkaiisiin mikrokontrollerimalleihin.

Testattavaksi valitut kolme mikrokontrolleria ovat arkkitehtuurin iältä varsin erilaisia. Freescalen ColdFire-arkkitehtuuri on ollut markkinoilla jo pitkään. Vastaavasti Atmelin AVR32-arkkitehtuurin mikrokontrollerit edustavat uutta tuoteperhettä. Analog Devicesin Blackfin-arkkitehtuuri sijoittuu iältään näiden kahden väliin. Piirivalmistajat eivät suostu paljastamaan mikrokontrollereiden tämän hetkistä elinikäennustetta, vaan toteavat sen riippuvan kysynnästä. Ainakin AVR32- ja ColdFire-arkkitehtuureiden tuoteperheet näyttäisivät kuitenkin tulevaisuudessa kasvavan uusilla mikrokontrollerimalleilla, joten kyseiset arkkitehtuurit tuskin ovat häviämässä markkinoilta lähiaikoina.

Eri valmistajien ohjelmistokehitysympäristöistä näyttäisi löytyvän hyvin pitkälti samat perusominaisuudet. Suurimmat erot eri ympäristöjen välillä ovat yleisessä käytettävyydessä sekä valmistajien ympäristöille tarjoamissa dokumentaatioissa ja esimerkkisovelluksissa. Opinnäytetyön puitteissa keskityttiin lähinnä valmistajien omien kehitysympäristöjen testaukseen. Markkinoilla on kuitenkin runsaasti ulkopuolisten valmistajien ohjelmistokehitysympäristöjä, varsinkin yleiskäyttöisille mikrokontrollereille.

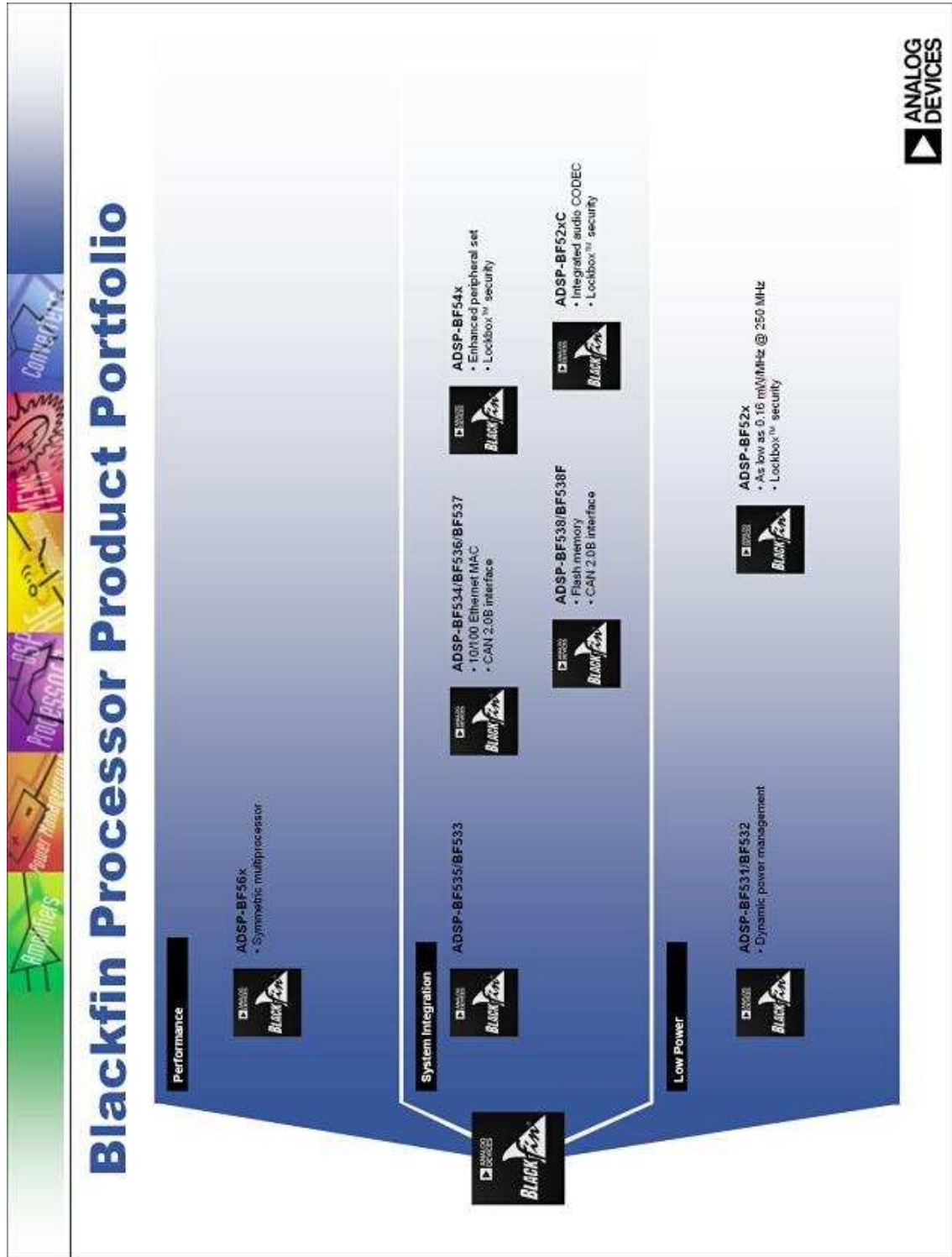
Opinnäytetyön kartoituksessa on saatu rajattua suuresta mikrokontrollerivalikoimasta ja ohjelmistokehitysympäristövaihtoehdoista ala-asemaan tällä hetkellä sopivat vaihtoehdot. Lopullisen valinnan onnistuminen selvinnee vasta tulevaisuudessa, kun nähdään, kuinka pitkäikäisiksi eri mikrokontrollerivaihtoehdot osoittautuvat.

Lähteet

- 1 Kirkman, Robert. Development in Substation Automation Systems.
- 2 Bricker, S., Gonen, T. & Rubin, L. Substation Automation Technologies and Advantages. IEEE, 2001.
- 3 McClanahan, Robert. The Benefits of networked SCADA system utilizing IP-enabled networks. IEEE, 2002.
- 4 Communication networks and systems in substations, Part3: General Requirements. IEC 61850-3.
- 5 Pozzuoli, Marzio. Ethernet in Substation Automation Applications – Issues and Requirements. Ruggedcom Inc.
- 6 Freescale Semiconductor. MCF52235 ColdFire Integrated Microcontroller Reference Manual, 2007.
- 7 Analog Devices. ADSP-BF534/ADSP-BF536/ADSP-BF537 Datasheet, 2008.
- 8 Analog Devices. ADSP-BF537 Blackfin Processor Hardware Reference, 2007.
- 9 Analog Devices. ADSP-BF53x/BF56x Blackfin Processor Programming Reference, 2007.
- 10 Analog Devices. Estimating Power for ADSP-BF534/ADSP-BF536/ADSP-BF537 Blackfin Processor. Engineer-to-Engineer Note, 2007.
- 11 Freescale Semiconductor. MCF5223x Family Specification Sheet, 2006.
- 12 Freescale Semiconductor. MCF52235 ColdFire Microcontroller Datasheet, 2007.
- 13 Atmel. AT32UC3A Series Datasheet, 2008.
- 14 Atmel. AVR32UC Technical Reference Manual, 2007.
- 15 Introduction to USB On-The-Go. (WWW-dokumentti). <www.usb.org/developers/onthego/USB_OTG_Intro.pdf>. Luettu 17.6.2008.
- 16 Netcontrol. GW502 Datasheet.
- 17 Netcontrol. RTU8 Datalehti.
- 18 Analog Devices. VisualDSP++ 5.0 User's Guide, 2007.

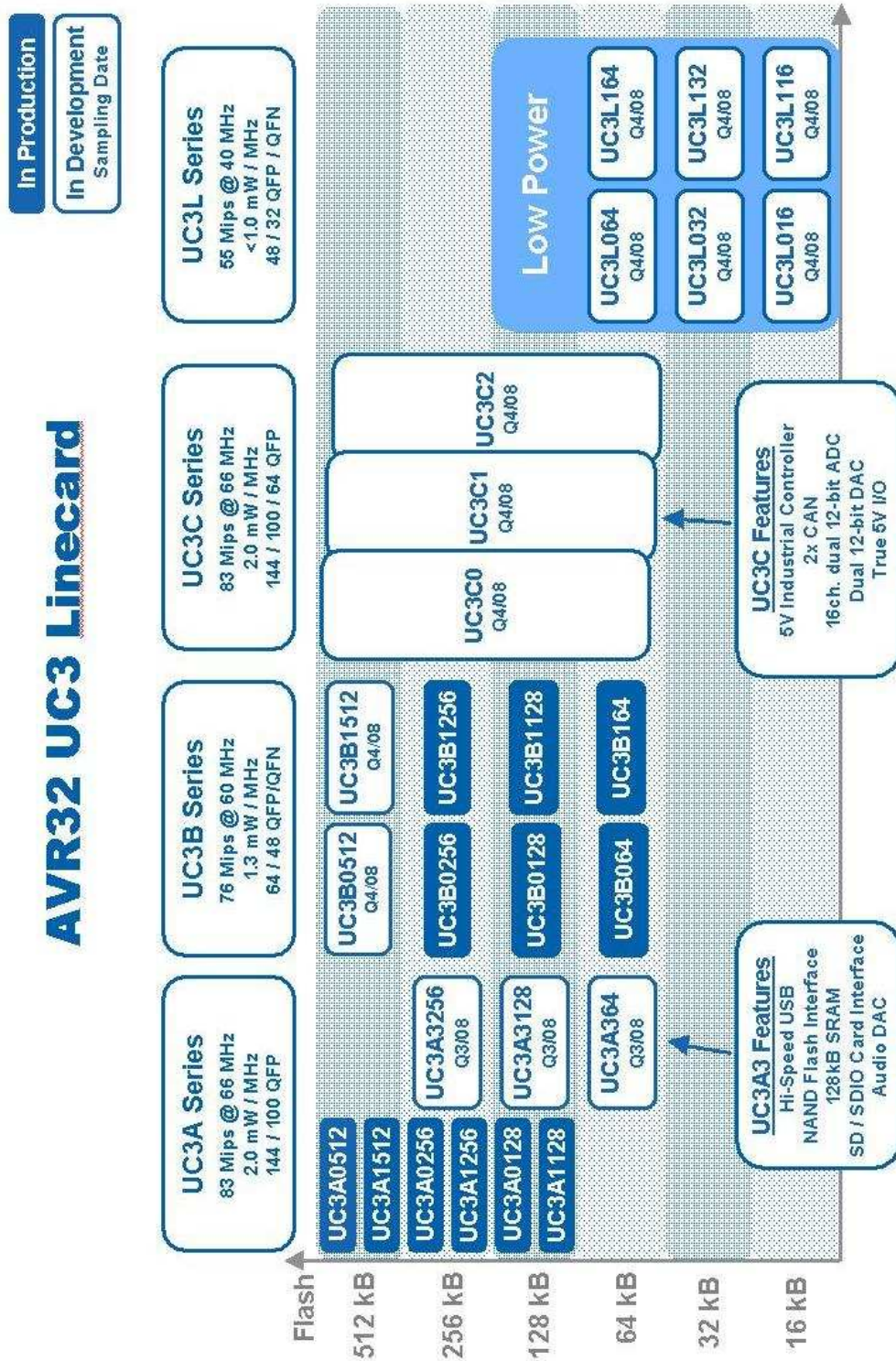
- 19 Freescale Semiconductor. CodeWarrior Development Studio IDE 5.7 User's Guide, 2007.
- 20 IAR. IAR Embedded Workbench IDE User Guide, 2006.
- 21 Analog Devices. Blackfin Processor Roadmap. (WWW-dokumentti). <http://www.analog.com/en/embedded-processing-dsp/blackfin/content/blackfin_roadmap/fca.html>. Luettu 17.6.2008.
- 22 Atmel. AVR32 UC3 Linecard. Microcontroller Seminar, 2008.
- 23 Freescale Semiconductor. The ColdFire Family of 32-bit Embedded Controllers, 2007.

Liite 1: Analog Devices Blackfin -tuoteperhe



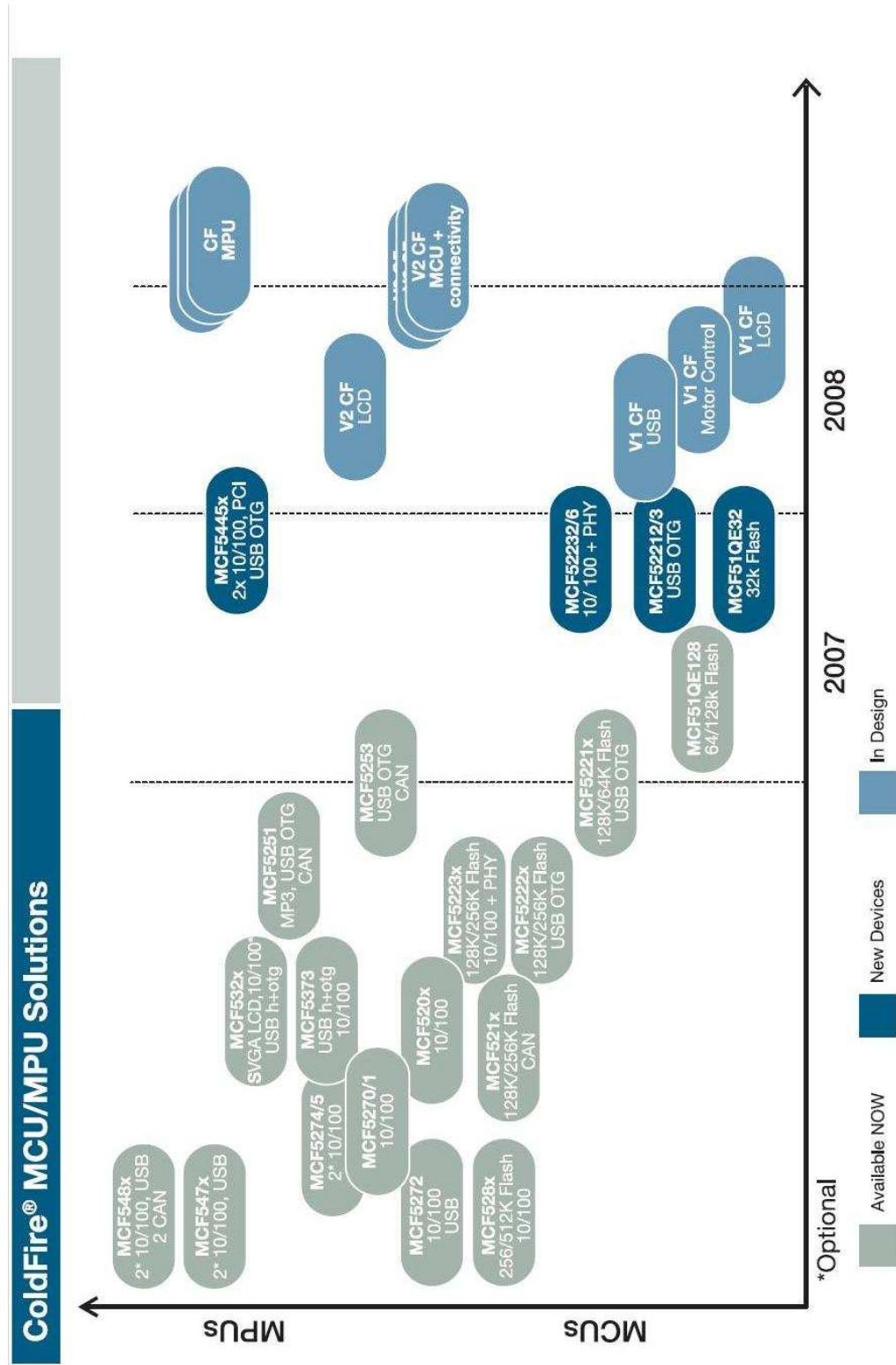
Kuva 1. Analog Devices Blackfin -sarjan mikrokontrollerit (21).

Liite 2: Atmel UC3 -tuotepihe



Kuva 1. AVR32 UC3 -tuotepiheen mikrokontrollerit (22).

Liite 3: Freescale ColdFire -tuoteperhe



Kuva 1. Freescale ColdFire -sarjan mikrokontrollerit (23).