

Teemu Rautiainen

ARM-7-kehitysympäristön käyttöönotto

Insinöörityö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2008



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Tekniikan ja liikenteen ala	Koulutusohjelma Tietotekniikan koulutusohjelma
Tekijä(t) Teemu Rautiainen	
Työn nimi ARM-7-kehitysympäristön käyttöönotto	
Vaihtoehtoiset ammattiopinnot Sulautetut järjestelmät	Ohjaaja(t) Arto Partanen Toimeksiantaja Kajaanin AMK/Markku Karppinen
Aika Kevät 2008	Sivumäärä ja liitteet 34+4
<p>Tämän insinööriyön tarkoituksena oli ARM-7-kehitysympäristön käyttöönotto. Työssä tutkittiin käytettävän Atmelin AT91SAM7X-EK-kortin ominaisuuksia, toimintaa sekä sen käyttöä erilaisilla ohjelmistoympäristöillä opetuskäytössä. Tarkoituksena oli myös tarkastella kortin soveltuvuutta opetuskäyttöön sekä tehdä korttiin liittyvää opetusmateriaalia. Opetusmateriaalin tuottaminen oli tarkoitus tehdä Techsmithin Camtasia-studiolla, jolla voitaisiin tehdä opetusmateriaalia videon muodossa. Kirjallinen opetusmateriaali oli tarkoitus tuottaa Microsoftin PowerPointia apuna käyttäen. Työ tehtiin Kajaanin ammattikorkeakoulun tietotekniikan laboratoriolle. Kortin perustana on AT91SAM7XC256-mikrokontrolleri.</p> <p>Ohjelmointiympäristöinä käytettiin IAR Systemsin IAR Embedded Workbenchiä sekä Eclipseä, joka perustuu vapaaseen lähdekoodiin. Eclipse sisältää monia lisäominaisuuksia ja sitä voidaan käyttää erilaisilla ohjelmointikie- lillä. Tuettuja ohjelmointiympäristöjä ovat esimerkiksi Java, C/C++ sekä PHP. Eclipse on eräänlainen graafinen editor, joka koostuu erilaisista osista. Käyttäjä voi räätälöidä itselleen tarkoituksenmukaisen ohjelmiston Eclipseä pohjana käyttäen.</p> <p>Eclipsen käyttö osoittautui erittäin haastavaksi, jonka vuoksi muut suunnitellut tehtävät jäivät joiltain osin toteutumatta. Eclipsen käytöstä voitaisiinkin tehdä erillinen, oma opinnäytetyö.</p>	
Kieli	Suomi
Asiasanat	ARM-7 mikrokontrollerikortti, Open-Source työkalut
Säilytyspaikka	<input type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Kajaani University of Applied Sciences	Degree Programme Information technology
Author(s) Teemu Rautiainen	
Title Inauguration of ARM-7 Development Environment	
Optional Professional Studies Embedded systems	Instructor(s) Arto Partanen Kajaani University of Applied Sciences
	Commissioned by Markku Karppinen Kajaani University of Applied Sciences
Date Spring 2008	Total Number of Pages and Appendices 34+4
<p>The objective of this bachelor's thesis was the inauguration of the ARM-7 development environment. The main goal was to examine the attributes, operation and usage of Atmel's AT91SAM7X-EK card in different programming environments. The card will be used in the Kajaani University of Applied Sciences for educational purposes. The card is based on the AT91SAM7XC256 microcontroller belonging to the ARM7/TDMI product-family, which are used, for example, in mobile phones and in industry.</p> <p>Open-source based Eclipse was used as the programming environment. Open-Source means that you can copy, fix and use the source code freely. Eclipse has various additional attributes and it can be used with many different programming languages. Supported programming languages include, for example Java, C/C++ and PHP.</p> <p>The usage of Eclipse turned out to be very challenging and, consequently, the thesis plan was not carried out as a whole. In fact, the usage of Eclipse could be a topic for a separate bachelor's thesis.</p>	
Language of Thesis	Finnish
Keywords	ARM-7 microcontroller, Open-Source tools
Deposited at	<input type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Tämä insinöörityö on tehty Kajaanin ammattikorkeakoulun tietotekniikan laboratoriolle.

Haluaisin kiittää ohjaajana toiminutta Arto Partasta hyvistä sekä asiantuntevista neuvoista.

Työ oli mielenkiintoinen, opettavainen sekä haastava.

SISÄLLYS

1 JOHDANTO	1
2 SULAUTETUT JÄRJESTELMÄT	2
3 AVOIN LÄHDEKOODI	5
3.1 Avoimeen lähdekoodin perustuvat ohjelmat	6
3.2 Työssä käytettävät avoimeen lähdekoodiin perustuvat ohjelmat	8
4 MIKROPROSESSORI	11
5 AT91SAM7XC256-MIKROKONTROLLERI	18
6 OPETUSKORTTI AT91SAM7X-EK	20
7 KEHITYSTYÖKALUJEN KÄYTTÖ	23
7.1 IAR Embedded workbench	23
7.2 Eclipse	24
7.3 Korjausehdotukset	29
8 TULOSTEN ANALYSOINTI	30
9 YHTEENVETO	31
LÄHTEET	32

KÄYTETYT TERMIT

AES (Advanced Encryption Standard)

AES on symmetrinen salausalgoritmi, joka käyttää 128, 192 tai 256 bitin salausavainta.

ALU (Arithmetic Logic Unit)

Mikroprosessorissa oleva toiminnallinen yksikkö, joka suorittaa laskutoimituksia ja käsittelee bittejä nollaamalla, asettamalla tai vertaamalla niitä toisiinsa.

CKGR (Clock generator)

Kytkenä, joka tuottaa ajastussignaalin piirin synkronoitua toimintaa varten.
(vapaa suomennos)

DBGU (Debug unit)

Debugausyksikkö

Flash-muisti

Muistityyppi, jonka sisältöä voidaan muuttaa ja joka säilyttää sisältönsä myös ilman sähköä.

PDC (Peripheral DMA Controller)

Oheislaite DMA-kontrolleri. (vapaa suomennos)

PIO (Parallel Input/Output Controller)

Rinnakkainen tulo/lähtöohjain

PWMC (Power Width Modulation Controller)

Virranleveysmodulaatio-ohjain (vapaa suomennos)

RISC (Reduced Instruction Set Computer)

Proessori, jonka konekielikäskyt ovat yksinkertaisia mutta jotka toteutetaan erittäin nopeasti, tyypillisesti yhden kellojakson aikana ja ilman mikrokoodia.

ROM (Read-Only Memory)

Lukumuisti

SPI (Serial Peripheral Interface)

Synkroninen sarjamuotoisen datan linkki, joka toimii full duplex -tilassa.

SRAM (Static RAM)

Muistipiirien valmistustekniikka, jossa kukin bitti tallennetaan kahdesta transistorista koostuvaan flip-flop-kiikkuun. Sähkö virtaa kiikun jommaltakummalta puolelta bitin arvosta riippuen.

TDES (Triple-Des)

DES-salausmenetelmän seuraaja, jossa avaimen pituus on 112 bittiä tai 168 bittiä.

TWI (Two-wire Serial interface)

Sama kuin I²C, lisenssisyistä eri nimitys.

USART (Universal Synchronous/Asynchronous Receiver-Transmitter)

Sarjaliikenteen lähetys- ja vastaanottopiiri, joka muuntaa rinnakkaismuotoisen datan sarjamuotoiseksi dataksi.

WDT (Watchdog timer)

Ajastin, joka seuraa laitteen tai ohjelman toimintaa ja antaa hälytyksen, jos viive tai toimimattomuus on kestänyt ennalta määrätyn ajan.

1 JOHDANTO

Sulautetut järjestelmät kehittyvät nopeaa tahtia, joten Kajaanin ammattikorkeakoulu on hankkinut Atmelin AT91SAM7XC–evaluointisarjan. Sarja sisältää AT91SAM7X-EK-kortin, jonka perustana on AT91SAM7XC-mikrokontrolleri. Kortti on hankittu nykyaikaistamaan Kajaanin ammattikorkeakoulun tietotekniikan laboratoriota. Muut käytössä olevat opetuskortit, jotka perustuvat 8051,- AVR,- sekä PowerPC-tuoteperheisiin, ovat toki vielä ajankohtaisia, mutta hankitulla Atmelin kortilla voidaan tehdä monipuolisempia sovelluksia. Oppilaan erikoistuessa sulautettuihin järjestelmiin koulun tulisi tarjota mahdollisimman ajanmukainen opetusvälineistö ajatellen nykyisiä sulautettuja järjestelmiä. Evaluointisarja mahdollistaa vaativien sovellusten tekemisen, joihin opiskelijat väistämättä törmäävät työelämässä.

Kortilla voidaan tehdä erilaisia sovelluksia kortin monipuolisten liitäntöjen sekä komponenttien avulla. Tämän insinööriyön tarkoituksena on edellä mainitun kortin käyttöönotto. Käyttöönoton tavoitteena oli tutkia edellä mainitun kortin ominaisuuksia, liitäntöjä sekä sen käyttöä erilaisilla ohjelmointiympäristöillä. Tavoitteena oli myös tutkia kortin soveltuvuutta opetuskäyttöön sekä tehdä mahdollisia opetuskäyttöön soveltuvia ohjelmia ja myös opetuskäyttöön soveltuvaa materiaalia. Opetuskäyttöön tulevan materiaalin tuottaminen oli tarkoitus tehdä käyttäen Techsmithin Camtasia Studio -ohjelmaa, jolla saataisiin aikaan eräänlainen ”opetusvideo”. Tarkoituksena oli tehdä myös luettavaa materiaalia, jonka toteutus hoidettaisiin Microsoft PowerPointin avulla.

Ohjelmointiympäristöistä oli tarkoitus käyttää IAR-yhtiön embedded workbenchia sekä vapaaseen lähdekoodiin perustuvaa ohjelmointiympäristöä. IAR on johtavin markkinoilla oleva sulautettujen järjestelmien ohjelmistotoimittaja, jonka vuoksi kyseisen yhtiön ohjelmisto valittiin käytettäväksi. Vapaaseen lähdekoodiin perustuvista ohjelmistoista valittiin Eclipse monipuolisuutensa vuoksi. Vapaaseen lähdekoodiin perustuvat ohjelmat kehittyvät jatkuvasti kyseisten ohjelmien käyttäjien ansiosta, joten ohjelmistoihin on saatavilla runsaasti erilaisia päivityksiä sekä parannuksia.

2 SULAUTETUT JÄRJESTELMÄT

Sulautetut järjestelmät ovat nykyään osa jokapäiväistä elämää. Esimerkiksi kännykkä, jota käytetään lähes joka päivä, on sulautettu järjestelmä. Sulautettu järjestelmä koostuu ohjelmistosta ja laitteistosta. Ohjelmisto suunnitellaan yleensä kehitysohjelmia apuna käyttäen, sekä laitteiston ytimenä toimii yleensä mikrokontrolleri.

Kehitysohjelmilla voidaan testata laitteiston toimintaa sekä ajaa laitteistoon ohjelmia. Kehitysohjelmat helpottavat testaajan työtä suuresti, koska niillä voidaan tutkia mahdollisia ohjelmistovirheitä sekä myös laitteiston haluttua toimintaa.

Sulautetuissa järjestelmissä käytetään nykyään lähes poikkeuksetta mikrokontrolleria, varsinkin, jos laite on hiemankaan monimutkaisempi. Mikrokontrollerin käyttö on suosittua, koska siinä yhdistyvät pieni virrankulutus, hyvä suorituskyky sekä pieni koko. Erityisesti viihdeelektronikka vaatii kaikkia kyseisiä ominaisuuksia. [1.]

Mikrokontrolleri on komponentti, johon on mikroprosessorin lisäksi yhdistetty muita mikro-tietokoneen osia, kuten esim. laskureita tai ajastimia, muistia, I/O-liitäntöjä, sarjaliittimiä sekä A/D-muunnin. Mikrokontrollerit on alun perin kehitetty juuri sulautettujen järjestelmien tarpeisiin, ja siksi alkuaikoina niitä joskus kutsuttiinkin sulautetuiksi ohjaimiksi. [2, s. 13.]

Sulautettujen järjestelmien kehitystyökalut

Sulautetuille järjestelmille on saatavana lukuisia erilaisia kehitystyökaluja. Käytettävien kehitystyökalujen tarve riippuu siitä, minkälaiseen käyttöön laite tulee. Koska kehitystyökalujen määrä on suuri, seuraavassa on kerrottu kyseisessä työssä käytettävistä kehitystyökaluista sekä niitä tarjoavista yhtiöistä.

Kaupalliset kehitystyökalut

IAR Systems on tuottanut kehitystyökaluja sulautettuihin järjestelmiin vuodesta 1983 lähtien. Yhtiö tarjoaa C/C++-kääntäjiä ja -debuggereita. IAR:n tuottama käännös/debuggaus -työkalu on IAR Embedded Workbench, jota olemme käyttäneet myös opiskelussamme. Tilakonetyökaluista on saatavana IAR Visual State Machine, joka on UML-yhteensopiva. RTOS-, TCP/IP- ja USB-työskentelyyn on saatavilla IAR Powerpac. Yhtiö tarjoaa myös kehitystyökalupaketteja sekä debuggaus-mittapäitä. [3.]

IAR Embedded Workbench

IAR Embedded Workbench on työkalu sulautettujen järjestelmien C/Embedded C++-ohjelmointiin. Käyttöönotto on helppoa helpon käyttöliittymän ansiosta. Myös ohjelmointi on helppoa käytettävästä mikrokontrollerista riippumatta. Ohjelma on maksullinen, joten halpaa tai ilmaista ratkaisua hakeville kyseistä työkalua voi suositella varauksin. AT91SAM7X-EK-evaluointisarjan mukana tulleen IAR Embedded Workbench Version 4.30A Evaluation-version työkalun käyttöä rajoittaa myös se tekijä, että kääntäjään on asetettu koodin koolle rajoitus, joka on 32 kilotavua. Käytettäessä erillistä debuggeria kyseinen ohjelmaversio tarvitsee Atmelin SAM-ICE-debuggerin. Toisten valmistajien debuggerit toimivat vain 5.xx-versioiden kanssa. Yritystoiminnassa sekä opetuskäytössä kyseinen ohjelmointityökalu on hyvä aiemmin mainitun helpon käytön ja teknisen tuen saatavuuden takia.

Muut kaupalliset kehitysohjelmat

Saatavilla on myös muita kehitysohjelmia, joita ovat esimerkiksi Keilin μ Vision, sekä myös Green Hillillä on oma kehitysohjelmansa. Kyseisiin ohjelmiin ei kuitenkaan perehdytty tässä työssä ajanpuutteen takia.

Avoimeen lähdekoodiin perustuvat kehitysohjelmat

Avoimeen lähdekoodiin perustuva Eclipse on esitelty myöhemmin luvussa 3.2.

3 AVOIN LÄHDEKOODI

Lähdekoodi on ohjelman perusta, jonka avulla ohjelma luodaan ja sitä voidaan myöhemmin parantaa tai korjata. Open Source eli avoin lähdekoodi on kaikkien saatavana, ja se on kaikille ilmaista. Ohjelmaa voidaan siis vapaasti muokata, korjata tai levittää. Kaupallisten ohjelmien lähdekoodi on suljettu, jolloin ohjelman muokkaaminen tai korjaaminen omiin tarpeisiin soveltuvaksi on mahdotonta. [4.]

Historia

Vielä 1960-luvulle saakka ohjelmistotuotteet kuuluivat ostettavan laitteiston hintaan ja niissä oli lähes poikkeuksetta ohjelmistokoodi mukana. Yliopistomaailmassa avoimuus säilyi, mutta vuonna 1965 IBM lopetti lähdekoodin jakamisen käyttöjärjestelmien mukana ja uudet lait ja tekijänoikeussäännökset kielsivät ohjelmien levittämisen ilman lupaa. Ilmaiset lähdekoodit katosivat nopeasti ja kaupalliset ohjelmistotuottajat ottivat markkinat hallintaansa.

Vuonna 1984 Richard Stallman sai tarpeekseen kaupallisesta kehityksestä ja julkaisi avoimen ohjelmistopakettin, GNU:n, joka sisälsi käyttöliittymän sekä erilaisia ohjelmointityökaluja. GNU:n suosio kasvoi nopeasti ja Stallman perusti kumppaniensa kanssa Free Software Foundationin kehittämään avoimen lähdekoodin ohjelmistoja. Tätä varten luotiin GNU General Public License (GPL), joka takasi kaikille oikeuden käyttää ja levittää ohjelmaa ja tehdä siihen muutoksia. GNU oli pelkkä kokoelma ohjelmia, josta puuttui ydin, kernel, jota ilman se ei voinut toimia itsenäisenä käyttöjärjestelmänä.

Vuonna 1991 Linus Torvald kehitteli avoimen Minix-käyttöjärjestelmän pohjalta oman versionsa kernelistä, jolle hän antoi nimeksi Linux. Pitkän kehitystyön tuloksena Linux on nykyisin yleisin avoimeen lähdekoodiin perustuvien käyttöjärjestelmien ydin. Vuonna 1998 perustettiin Open Source Initiative -yhteisö, jonka tehtävänä on hallita ja edistää avoimen lähdekoodin käyttöä yleiseksi hyväksi. [4.]

3.1 Avoimeen lähdekoodin perustuvat ohjelmat

Avoimeen lähdekoodiin perustuvia ohjelmia on saatavana nykyään lähes joka tarpeeseen. Avoimeen lähdekoodiin perustuvia ohjelmia on saatavana koti- teollisuus- sekä opetuskäyttöön. Avoimen lähdekoodin ohjelmia on saatavalla runsaasti, jos tietää mistä etsiä.

Avoimeen lähdekoodiin perustuvien ohjelmien etsiminen esim. internetiä apuna käyttäen vaatii oman aikansa, varsinkin tietotekniikkaan perehtymättömälle. Ohjelmiston kehittäjät eivät mainosta tuotteitaan medioissa, koska he tekevät pääosin töitä ilmaiseksi, joten mainontaan ei ole rahaa. Avoimeen lähdekoodiin perustuvat ohjelmat ovat yleensä monimuotoisempia, niissä on runsaasti ominaisuuksia, jonka takia ohjelmien käyttäminen aloittelijalta voi olla hankalampaa. Seuraavassa taulukossa on esitelty ohjelmia, joita voidaan käyttää esimerkiksi yritys- tai opetusikäytössä.

Taulukko 1. Open Source testaustyökalut [5.]

Open Source testaustyökalut		
Työkalu	Kuvaus	Osoite
Bugzilla	Bugien hallinta	https://bugzilla.mozilla.org/
Selenium	Testaus automatisointi	http://www.openqa.org/selenium/
Pamie	Testaus automatisointi	http://pamie.sourceforge.net/
Watir	Testaus automatisointi	http://wtr.rubyforge.org/
Jmeter	Suorituskyky testaus	http://jakarta.apache.org/jmeter/
Grinder	Suorituskyky testaus	http://grinder.sourceforge.net/
OpenSTA	Suorituskyky testaus	http://www.opensta.org/
Staf	Testien ajo framework	http://staf.sourceforge.net/index.php
Fitnessse	Testien ajo framework	http://fitnessse.org/
AllPairs	Testien generointi	http://www.satisfice.com/tools.shtml
Testlink	Testien hallinta	http://testlink.sourceforge.net/docs/testLink.php
Paros	Tietoturva	http://www.parosproxy.org/index.shtml
Snort	Tietoturva	http://www.snort.org/
Ethereal	Tietoturva	http://www.ethereal.com/
Nessus	Tietoturva	http://www.nessus.org/
Junit	Yksikkötestien automatisointi	http://www.junit.org/news/extension/index.htm

Yrityskäyttöön soveltuvat ohjelmat ovat yleensä kalliita, joten pienempi tai suurempikin yritys voi säästää huomattavat summan käyttämällä avoimeen lähdekoodiin perustuvia työkaluja.

Kotikäyttöön on saatavilla myös runsaasti avoimeen lähdekoodiin perustuvia ohjelmia. Avoimeen lähdekoodiin perustuvista ohjelmista saadaan todella kattava paketti kotikäyttöön ja vieläpä ilmaiseksi. Kaupallisia ohjelmistoja käytettäessä kotikoneen käyttäjälle ohjelmistot voivat maksaa pahimmassa tapauksessa jopa yli puolet tietokoneen hankintahinnasta.

3.2 Työssä käytettävät avoimeen lähdekoodiin perustuvat ohjelmat

Työssä käytettiin pääasiallisena ohjelmointiympäristönä Eclipseä, joka perustuu avoimeen lähdekoodiin.

Eclipse (IDE)

Eclipse on ohjelmointiympäristö, joka tukee mm. Java- PHP- sekä C/C++-ohjelmointikieltä. IDE lisäys tarkoittaa ”Integrated Development Enviroment”, eli vapaasti suomennettuna ”Yhdistetty sovelluskehitysympäristö”. Eclipse on eräänlainen graafinen selain, jolla käytetään siihen liitettyjä erilaisia komponentteja, eli ”plug-in:ejä”. Kyseisessä työssä tarvittavat ”plug-in:it” on esitelty myöhemmin taulukossa 6.

Eclipsen käyttö alkaa sopivien plug-inien valinnalla. Esimerkiksi C/C++-ohjelmoinnissa tärkein plug-in on CDT, jolla mahdollistetaan C/C++-kielten kääntäminen. Jokaiselle tuetulle ohjelmointikielelle on olemassa omat tarvittavat plug-init.

Ohjelmistoa kehitetään avoimen lähdekoodin lisenssillä, joten sen käyttö on maksutonta. Sen suurin heikkous aloittelijan kannalta onkin sen suuri monimuotoisuus. Käyttöönotto ei ole helppoa aloittelijalle vaan vaatii suuria ponnisteluja. Jotain ohjelmoinnin laajuudesta kuvaa esimerkiksi James P. Lynchin ”Using Open Source Tools for AT91SAM7S Cross Development” -käyttöopas, joka on 198-sivuinen. Toki tarkempi perehtyminen palkitaan jo vähintäänkin säästyneessä rahamäärässä. Ohjelmointiympäristön toinen heikkous on sen teknisen tuen puute. Ongelmien ilmaantuessa käyttäjä joutuu itse etsimään viat. Jonkinlaista teknistä tukea voi saada esim. keskustelufoorumeilta, joissa on toisia Eclipsen käyttäjiä, mutta yritystoiminnassa kyseinen vaihtoehto ei ole sen epävarmuuden ja mahdollisen hitaan vastauksen takia mahdollista.

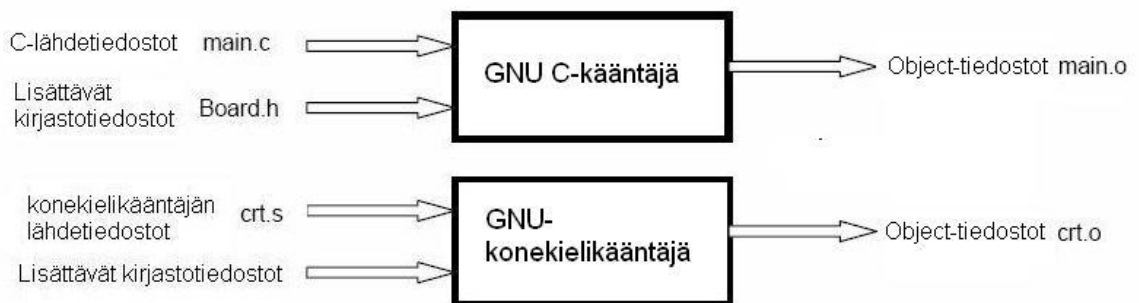
Eclipsen toiminta

Seuraavissa kuvissa on selostettu Eclipsen toimintaa sekä sitä, miten tarvittavat ohjelmat tekevät toimintonsa.



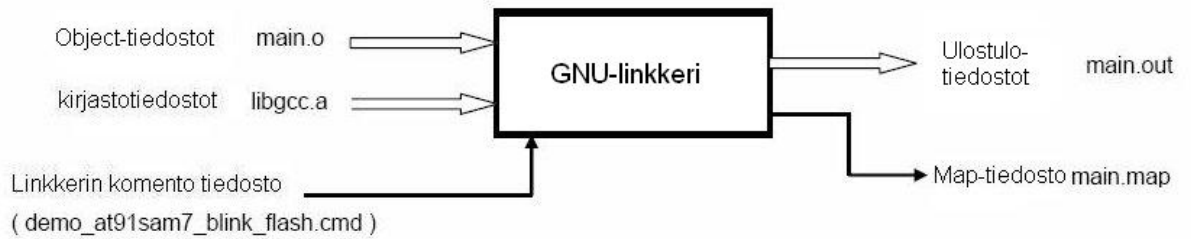
Kuva 1. Eclipse-editori

Eclipse tekee tarvittavat C- sekä konekielikääntäjän lähde- ja lisättävät tiedostot. Eclipse tekee myös tarvittavat ”make”- sekä linkkerin tiedostot. [6.]



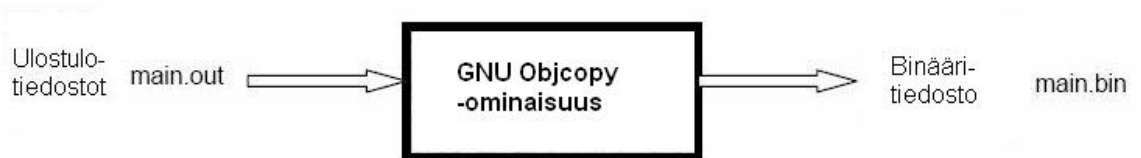
Kuva 2. GNU-kääntäjä – ja -konekielikääntäjä

GNU-kääntäjä ja GNU-konekielikääntäjä kääntävät ja ”kokoavat” lähdetiedostot. Kääntämisen tuloksena syntyvät object- tiedostot, jotka ovat .o-loppuisia. Object-tiedostot ovat lähellä mikrokontrollerin tarvitsemia lopullisia tiedostoja sillä erotuksella, että niistä puuttuvat osoitteet. [6.]



Kuva 3. GNU-linkkeri

GNU-linkkeriä käytetään keräämään tehdyt object-tiedostot sekä tarvittavat object-moduulit, ratkaisemaan tarvittavat osoitteet ja yhdistämään edellä mainitut asiat ladattavaksi ulostulo-tiedostoksi. Kyseinen tiedosto sisältää .out-päätteen. Linkkerin komentotiedosto määrittelee luotujen object-moduulien muistipaikan sekä järjestelee tarvittaessa ne. [6.]



Kuva 4. Flash -tiedoston tuottaminen

Flash-sovelluksia käännettäessä Gnu objcopy-ominaisuus valmistaa käytettävälle debuggerille sen tarvitseman .bin-tiedoston. Kyseistä tiedostoa tarvitaan flash-sovelluksen kääntämisessä. [6.]

4 MIKROPROSESSORIT

Mikroprosessori on tietokoneen keskusyksikkö eli prosessori, joka on integroitu yhdelle piisirulle. Mikroprosessorin sisäiseen rakenteeseen ei käyttäjä pysty tekemään minkäänlaisia muutoksia. Käyttäjän on tyydyttävä ominaisuuksiin, jotka mikroprosessorin valmistaja on siihen tehnyt.

Samat toiminnot, jotka aikaansaadaan mikroprosessorilla, voidaan saada aikaan myös prosessorilla, jonka ympärille on koottu useista erillisistä komponenteista sama kokonaisuus kuin mikroprosessorissa. Prosessori, jonka ympärille on erillisten piirien myötä koottu sama kokonaisuus kuin mikroprosessorissa, on käyttöönotto kynnykseltään suurempi kuin mikroprosessori. Se vie suuremman tilan, on epäluotettavampi ja kuluttaa enemmän tehoa kuin mikroprosessori. Vastineeksi voi kuitenkin saada paremman suorituskyvyn ja omiin tarpeisiin räätälöidyn ratkaisun. [2, s. 13.]

Mikroprosessorin kolme tärkeintä sisäistä osaa ovat rekisterit, ALU sekä ohjausosa.

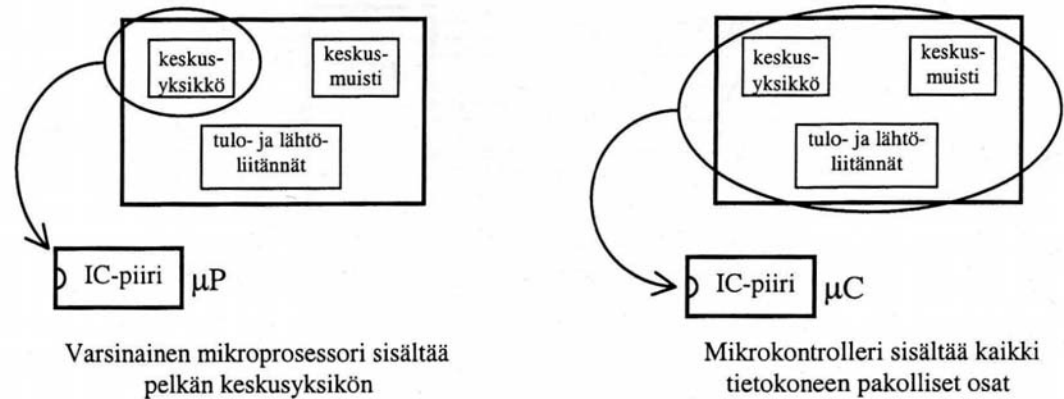
Tietokonetta, jonka keskusyksikkönä on mikroprosessori, kutsutaan nimellä mikrotietokone. Jos prosessorina on jokin muu kuin mikroprosessori, niin kyseessä ei ole mikrotietokone, vaan yleensä vain pelkkä tietokone.

Kaikkia mikrotietokoneita ei pystytä tai ei ole järkevää rakentaa yhden ja samanlaisen mikroprosessorin ympärille. Mikroprosessoreja onkin olemassa hyvin monenlaisia, koska eri käyttökohteissa eri ominaisuudet ovat ratkaisevia. Ei ole olemassa vain yhtä ja ainoaa mikroprosessoria, jota voisi käyttää kaikissa tilanteissa. Eri firmojen valmistamien mikroprosessorien paremmuus on suurelta osin makuasia. Jota on totuttu käyttämään, myös käytetään.

Yleiskäyttöiset mikroprosessorit voidaan jakaa kahteen pääryhmään:

- varsinaiset mikroprosessorit eli keskusyksiköt, μP
- mikrokontrollerit eli mikro-ohjaimet, μC

Alla olevassa kuvassa on esitetty, mitkä tietokoneen pakollisista osista on integroitu yhdelle piirille kyseisissä kahdessa pääryhmässä. [2, s. 13.]



Kuva 5. Mikroprosessorin ja mikrokontrollerin erot [2, s. 13.]

Yleiskäyttöisten mikroprosessorien lisäksi on olemassa erikoistehtäviin tarkoitettuja mikroprosessoreja. Yleisimpiä ovat DSP-prosessorit, jotka on tarkoitettu digitaaliseen signaalinkäsittelyyn.

Yleiskäyttöisten mikroprosessorien pääpiirteet ovat hyvin samanlaisia. Suurempibittiset prosessorit sisältävät samat perusasiat kuin pienempibittiset, lisättyinä omilla erikoispiirteillään. [2, s. 14.]

Mikrokontrollereilla on yleensä rajatummat ominaisuudet verrattuna mikroprosessoreihin. Esim. nopeus sekä muistinosoituskky ovat pienempiä. Mikrokontrollereilla tehdyn toteutuksen komponenttimäärä on kuitenkin pienempi sekä tehonkulutus on myös pienempi. Mikrokontrollereita käytetään yleensä vaatimatonta laskutehoa tarvitsevilla ohjaussovelluksissa sekä kulutuselektronikassa. Mikrokontrolleria voidaan myös käyttää isommissa tietokoneissa jonkin osatehtävän suorittamiseen. Esimerkiksi PC-koneessa mikrokontrolleri suorittaa näppäimistön lukemisen ja merkin lähetyksen pääprosessorille. [2, s. 14.]

Mikroprosessorin rekisterit toimivat keskusyksikössä tiedon väliaikaisina varastopaikkoina. Ne ovat kuin muistipaikkoja, joilla jokaisella on oma nimensä. Yleiskäyttöiset rekisterit ovat täysin prosessorin käyttäjän hallinnassa, eli ohjelmoija voi päättää, mitä tietoa niihin tallennetaan. Prosessori ei muuta yleiskäyttöisten rekisterien sisältöä, ellei sille anneta nimenomaan käskyä, jolla sisältöä halutaan muuttaa. Yleiskäyttöisiä rekistereitä on yleisesti viidestä kahteenkymmeneen. [2, s. 18.]

Omiin tehtäviinsä varattuja rekisterejä prosessori käyttää joihinkin omiin erikoistehtäviinsä. Tämä tarkoittaa että prosessori voi muuttaa näiden rekisterien sisältöä automaattisesti muiden käskyjen suorituksen yhteydessä ilman, että siihen tarvittaisiin omia käskyjä. Erikoiskäyttöisiä rekistereitä voidaan yleensä lukea sekä niihin pystytään myös kirjoittamaan käyttäjän toimesta. [2, s. 19.]

Rekisterit

Akku eli rekisteri A (accumulator) on prosessorin käytetyin rekisteri. Kyseessä on sisäisen dataväylän ”solmupiste”, jonka kautta iso osa tiedoista kulkee. Kun ALU:ssa tehdään jonkinlainen toimenpide, niin operandi on yleensä aina akussa. [2, s. 19.]

Lippurekisteri eli liput (flags) eli statusrekisteri eli PSW (program/processor status word) on yksittäisten yhden bitin tietojen eli lippujen yhteenliittymä. Lippurekisterin jokainen bitti yksinään vastaa tiettyä tietoa. Käytetyt yksittäiset liput vaihtelevat prosessorikohtaisesti. Joitakin esimerkkejä käytetyistä lipuista ovat:

- C eli Carry, joka ilmoittaa mm. yhteenlaskussa mahdollisesti syntyneen ylivuodon
 - Z eli Zero, joka ilmoittaa operaation vastauksen olevan nolla
 - P eli Parity, joka ilmoittaa, onko akussa ”ykkösiä” pariton vai parillinen määrä.
- [2, s. 19.]

Ohjelmalaskuri eli PC (Program Counter) on rekisteri, jonka sisältönä on aina osoite, josta seuraava käsky haetaan.

Käskyrekisteri eli IR (Instruction Register) on rekisteri, johon prosessori lukee suoritettavan käskyn muistista. Käskyrekisteri on näkymätön käyttäjälle, eikä sitä voi lukea tai kirjoittaa.

Pino-osoitin eli SP (Stack Pointer) on rekisteri, jonka sisältönä on pinon huipun osoite eli se osoittaa pinon huippua. Pino on keskusmuistissa oleva muistirakenne, jota prosessori käyttää toimintaansa. [2, s. 19.]

Puskurit ovat rajapintana prosessorin sisäisten ja ulkoisten signaalien välillä. Puskurit antavat prosessorin ulkoisille väylille niin ”tehokkaat” signaalit, että samalle väylälle voidaan kytkeä joitakin ulkoisia piirejä. Jos ulkoisia piirejä kytketään paljon, ei prosessorin sisäistenkään puskureiden teho riitä, vaan joudutaan käyttämään tehokkaampia ulkoisia puskuripiirejä. [2, s. 19.]

ALU

ALU eli aritmeettis-looginen yksikkö on prosessorin osa, jossa laskutoimitukset ja operaatiot tehdään. ALU esitetään kuvissa yleensä V:n tai Y:n muotoisena kuviona. Tämä kuvaa sitä, että ALU:un voi olla tuloina kaksi ja lähtönä yksi operandi. ALU:ssa tehtävän operaation perusteella asetetaan lippurekisterin liput. [2, s. 20.]

Ohjausosa

Ohjausosa toimii prosessorin sisäisenä ohjauskeskuksena. Se hoitaa prosessorin sisäisen ja ulkoisen ajoituksen. Ohjausyksikön tuloina ovat ennen kaikkea muistista luetut käskyt, mutta niiden lisäksi myös jotkin ulkoiset ohjauslinjat. [2, s. 20.]

Kun prosessori on lukenut suoritettavan käskyn käskyrekisteriin, se purkaa käskyn omiksi toimintaohjeikseen käskydekooderissa. Tavallisesti se on ROM-muisti, jonka sisällöksi on koodattu prosessorin mikrokoodi. Mikrokoodi on käyttäjälle täysin näkymätöntä, eikä sen sisältöön voida mitenkään vaikuttaa. Mikrokoodin rakenne vaikuttaa siihen, miten tehokas ja nopea mikroprosessori on. [2, s. 20.]

Ohjauslogiikka muodostaa käskydekooderilta saatujen ohjeiden mukaisesti tarvittavat ohjaussignaalit prosessorin sisäiseen sekä ulkoiseen ohjaamiseen ja ajoitukseen. Sisäiset ohjaussignaalit ovat käyttäjälle näkymättömiä, mutta ulkoisten ohjaussignaalien avulla suoritetaan prosessorin liittäminen ympäryspiireihin. Tyypillisiä ohjaussignaaleja ovat luku- ja kirjoituspulssi. [2, s. 20.]

Käskykanta tarkoittaa sitä konekielisten käskyjen kokoelmaa, jonka prosessori ymmärtää. Jokainen mikroprosessori omaa oman käskykantansa, eivätkä toisen prosessorin käskyt sovi

suoraan toiselle prosessorille. Koska jokainen käsky esitetään 8-bittisellä operaatiokoodilla, kuuluu 8-bittisen prosessorin käskykantaan enintään 256 käskyä. [2, s. 37.]

Joissakin tapauksissa prosessorien käskykannat ovat ylöspäin yhteensopivia niin, että uusi tehokkaampi prosessori kykenee suorittamaan myös vanhemmalle prosessorille kirjoitettuja ohjelmia. Esimerkiksi Intelin 80865,- 80286,- 80386,- 80486-Pentium–prosessoriperhe omaa yhteensopivat ohjelmat. Käskykannan määrää viimeisenä prosessorin ohjausyksikössä oleva käskydekooderi, joka tulkitsee keskusmuistista luettavat käskyt. [2, s. 37.]

ARM

ARM on lyhenne nimestä Advanced Risc Machines Ltd. Kyseisen yhtiön nimi kehittämät prosessorit kantavat myös samaa nimeä. Yhtiö on perustettu vuonna 1990, ja omistajina ovat olleet Acorn, Apple ja VLSI. Yhtiö tunnettiin ennen nimenmuutosta nimellä Acorn. Acorn kehitti vuonna 1985 ensimmäisen kaupallisen RISC-prosessorin. Vuonna 1987 Acorn toi markkinoille ARM-prosessorin, joka oli ensimmäinen RISC-prosessori, jota käytettiin halvissa PC-tietokoneissa. [7, 8.] ARM on yksi maailman eniten lisensöity ja siten levinnein prosessoriydin. Alla olevassa taulukossa on esitetty kaikki valmistajat, jotka ovat hankkineet lisenssin ARM-7-prosessoriytimeen.

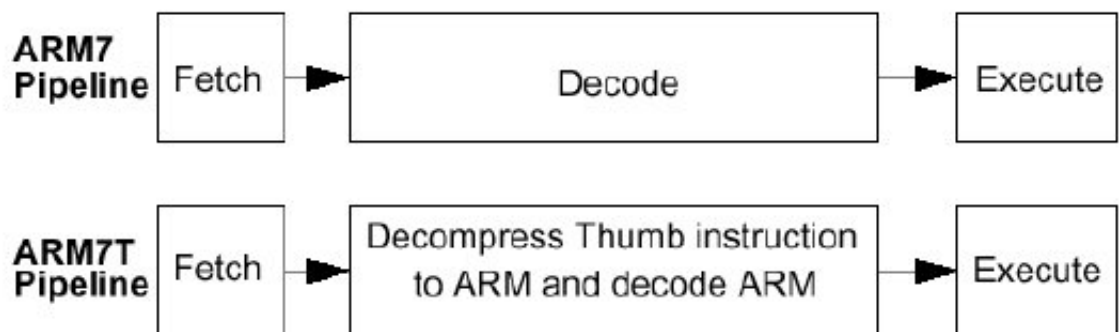
Taulukko 2. ARM-7-lisenssit [9.]

Accent	iSine	Shanghai Research ICC
Analog Devices Inc.	Industrial Tech Research Institute	Sharp
AMI Semiconductor	KawasakiMicroelectronics	Shenzen State Microelectronics (SMIT)
Atmel Corporation	LSI Corporation	SiRF Technology
Broadcom Corporation	Matsushita	Silicon Integrated Systems Corp.
Canadian Micro Corporation	Mediatek	Skyworks
Chartered Semiconductor	Micronas	SMIC Corporation
Cirrus Logic	National Asic CRC (SEU)	Socle
Conexant Systems Inc.	NEC Electronics Corporation	Sony
DSPG	Neomagic Corporation	Spreadtrum Communications Inc.
eSilicon Corporation	nVIDIA	STMicroelectronics
Flextronics	OKI	Texas Instruments
Freescale Semiconductor	Oxford Semiconductor	Toshiba Corporation
Fujitsu	NXP	UMC
Global Unichip Corporation	Pixim	Verisilicon
Hong Kong IC Design Centre	Qualcomm	Via Telecom/Via Technology
Huawei Technologies	Rohm	Winbond Electronics Corp.
Indilinx	Samsung Electronics	Xi'an
Infineon Technologies	Sanyo	Yamaha Corporation
Intel Corporation	Seiko Epson	Zarlink Semiconductor
Intellon Corporation	Shanghai Fudan	Zoran Corporation
Intrinsix Corporation		

ARM:ia käytetään erityisesti kannettavissa laitteissa pienen virrankulutuksensa vuoksi sekä suhteellisen hyvän tehosuhteensa takia. Useat matkapuhelinvalmistajat käyttävät matkapuhelimissaan ARM-ytimiä, Kyseisiä valmistajia ovat esimerkiksi Sony Ericsson, Nokia, Samsung sekä LG. [10.] ARM:iin on saatavilla erilaisia laajennuksia, joita ovat esim. Thumb, El Segundo sekä Jazelle. [8.]

ARM Thumb

ARM T(Thumb) -laajennus pienentää ARM:in käskykannan 16-bittiseksi sananleveydeltään, jolloin saavutetaan 35–40%:n muistinsäästö 32-bittiseen nähden. Laajennus mahdollistaa huomattavasti yksinkertaisemman järjestelmän toteutuksen, sillä käskyt vievät vain puolet normaalin 32-bittisen tilan vaatimasta muistimäärästä. Prosessorin suorituskyky ei kuitenkaan heikenny merkittävästi. [8.] Alla olevassa kuvassa on havainnollistettu miten Thumb-laajennus käyttäytyy.



Kuva 6. Thumb-laajennus [8.]

ARM-arkkitehtuuri

ARM-arkkitehtuuri on 32-bittinen RISC-prosessoriydin sisältäen 16 käskyä. Se sisältää 37 kappaletta 32-bittistä kokonaislukurekisteriä, joista 16 on käytettävissä. Prosessoriydin on liukuhihnoitettu, jossa on 3 erilaista vaihetta. Prosessoriydin voi olla myös varustettu väli-muistilla riippuen toteutuksesta. ARM 7 sisältää Von Neumann-tyyppisen väylärakenteen ja ARM 9 Harvard-tyyppisen väylärakenteen, joka sisältää 8-, 16-, tai 32- bittiset datatyytit. ARM 7:ssä on käytettävissä 7 erilaista toimintatilaa, joita ovat usr, fig, irq, svc, abt, sys sekä und. [8.]

Alla olevassa taulukossa on esitelty ARM 7-prosessorin toimintatilat.

Taulukko 3. ARM7:n toimintatilat [8.]

- **ARMin toimintatilat:**

- User (usr): Normaali ARM-ohjelmien suoritustila
- FIQ (fiq): Tiedonsiirtotila (fast irq, DMA-tyyppinen siirto)
- IRQ (iqr): Käytetään yleisten keskeytyksien palvelemiseen
- Supervisor (svc): Suojattu tila käyttöjärjestelmää varten
- Abort mode (abt): Valitaan datan tai käskyn haun keskeytyessä
- System (sys): Käyttöjärjestelmän 'etuoikeus'-tila käyttäjälle
- Undefined (und): Valitaan kun kohdataan määrittelemätön käsky

ARM7:ssä on 37 erilaista rekisteriä, joista käytettävissä on vaihteleva määrä pankitettuna rekistereinä riippuen käytettävästä toimintatilasta. Kuusitoista 32-bittistä kokonaislukurekisteriä R0-R15 ovat käytettävissä ARM-tilassa (usr). R0-R12-rekisterit ovat yleiskäyttöisiä rekistereitä. R13 on pinon osoitin (SP). R14 on aliohjelma-linkki rekisteri, joka säilyttää R15:n arvon, kun BL-käsky suoritetaan. R15 on ohjelmalaskuri (PC). R16 on tilarekisteri (CPSR, Current Program State Register). [8.] Liitteessä 1 on esitelty ARM7-rekisterikanta.

ARM7:n käskykanta on täysin 32-bittinen natiivissa käyttötilassa. Käskysana on 32-bittiä pitkä. Kaikki käskyt ovat ehdollisia, normaalissa käskyn suorituksessa AL- ehdolla (Always). Käskykanta on RISC -prosessoriksi monipuolinen erilaisine osoitetiloineen. Se sisältää 36 kappaletta erilaisia käskymuotoja. [8.] Liitteessä 2 on esitelty ARM7-käskykanta.

5 AT91SAM7XC256-MIKROKONTROLLERI

AT91SAM7XC-mikrokontrolleri pohjautuu ARM7TDMI-proessoriarkkitehtuuriin. ARM7TDMI on 32-bittinen ARMv4T Von Neumann RISC-arkkitehtuuriin perustuva prosessoriperhe. Kontrollerissa on 16-bittinen käskykanta sekä se toimii 55 Mhz:iin saakka saavuttaen 0,9 MIPS (Million of Instruction Per Second)/MHz suorituskyvyn, jolloin maksimaaliseksi suorituskyvyksi saadaan 49,5 MIPS. Käyttäjää voi valita käytettävän suorituskyvyn taajuuden mukaan. Kaikissa sovelluksissa ei maksimaalinen suorituskyky ole tarpeen, vaan tarvitaan esim. pienempää virrankulutusta, joten voidaan valita pienempi taajuus, koska virrankulutus kasvaa taajuuden noustessa.[1, 11.]

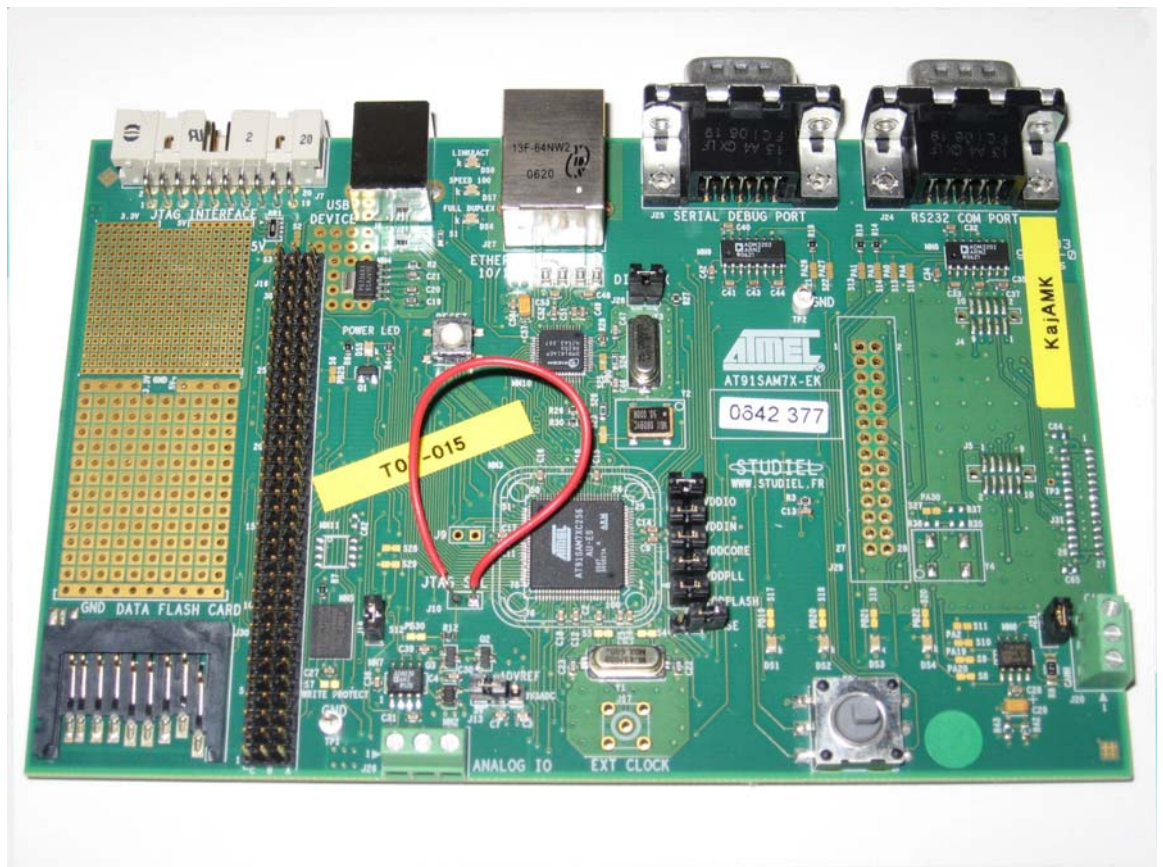
AT91SAM7XC-256-mikrokontrolleri sisältää monipuoliset sisäiset ominaisuudet. Alla olevassa taulukossa on esitelty AT91SAM7XC256-mikrokontrollerin tärkeimpiä sisäisiä ominaisuuksia.

Taulukko 4. AT91SAM7XC256-mikrokontrollerin tärkeimmät sisäiset ominaisuudet

- 256 kilotavua sisäistä flash-muistia
- 64 kilotavua sisäistä SRAM-muistia
- Kellogeneraattori (CKGR, Clock Generator)
- Debuggausyksikkö (DBGU, Debug Unit)
- Ikkunoitu Watchdog (WDT, Windowed Watchdog)
- Kaksi rinnakkaista Input/Output-ohjainta (PIO, Parallel Input/Output Controller)
- Seitsemäntoista oheislaite-DMA-ohjainkanavaa (PDC, Peripheral DMA Controller)
- Kehittynyt kryptaus järjestelmä (AES, Advanced Encryption System)
- Kolminkertainen datan kryptausjärjestelmä (TDES, Triple Data Encryption System)
- USB 2.0 laiteportti (nopeus 12 Megabittia/sekunti)
- Ethernet MAC 10/100 base-T
- Part 2.0A ja Part 2.0B yhteensopiva CAN-ohjain
- Kaksi universaalia synkronoitua/asynkronoitua vastaanotinlähettäjä (USART, Universal Synchronous/Asynchronous Receiver Transmitters)
- Kaksi isäntä/orja sarjallista lisälaiterajapintaa (SPI, Serial Peripheral Interface)
- 4 -kanavainen 16-bittinen virranleveysmodulaatio-ohjain (PWMC, Power Width Modulation Controller)
- Two-wire-rajapinta (TWI, Two-wire Interface)
- 8-kanavainen 10-bittinen analogi/digitaalimuunnin, 4 kanavaa multipleksattu digitaalisten I/O:en avulla.
- SAM-BAT TM käynnistysavustaja
- IEEE 1149.1 JTAG boundary Scan kaikkiin digitaalisiin pinneihin
- 5 voltin yhteensopivat I/O:t mukaan lukien 4 suurivirtaista (16mA) I/O:ta

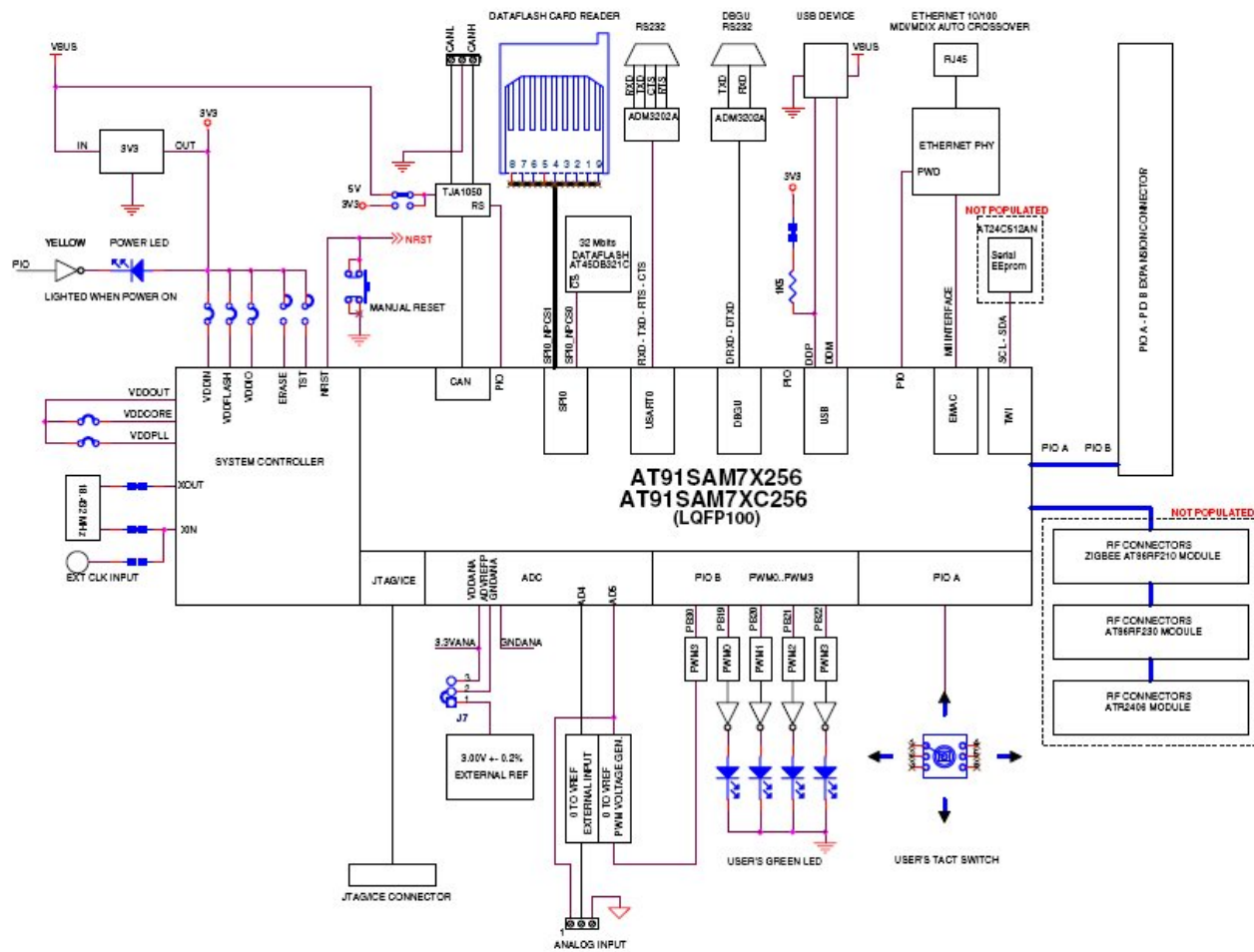
6 OPETUSKORTTI AT91SAM7X-EK

Opetuskortin käyttöönotto aloitettiin tutustumalla itse korttiin. AT91SAM7X-EK-opetuskortti on Atmelin tuotantoa. Aluksi tarkistettiin, tuleeko kortille virrat päälle sekä onko tarvittavat hyppykytkimet oikein. Tarvittavia hyppykytkimiä ovat esimerkiksi hyppykytkin J10, jolla valitaan JTAG-tai ICE-moodi. Toinen usein tarvittava hyppy-kytkin on hyppykytkin J8, jolla voidaan tyhjentää kortin flash-muisti. Hyppykytkimien luettelointi on esitetty liitteessä 1 ja 2/1. Huomattiin, että korttiin pitää tinata hyppylanka jtag-ice debuggauksen sallimiseksi. Alla olevassa kuvassa on esitelty opetuskortti sekä siihen tehty muokkaus.



Kuva 8. AT91SAM7X-EK-opetuskortti

Tutkittiin myös AT91SAM7X-EK-opetuskortin toiminnallista lohkokaaviota, joka on esitelty alla olevassa kuvassa.



Kuva 9. AT91SAM7X-EK-opetuskortin lohkokaavio [11.]

Tutkittiin myös AT91SAM7X-EK-opetuskortin ominaisuuksia sekä liitäntöjä. Alla olevassa taulukossa on esitelty opetuskortin tärkeimmät liitännät.

Taulukko 5. AT91SAM7X-EK opetuskortin tärkeimmät liitännät [11.]

- sarja-CAN-liitäntä
- Dataflash-korttipaikka
- DBGU-liitäntä
- RS232-liitäntä
- Ethernet 10/100 -liitäntä
- JTAG/ICE-liitäntä
-USB-liitäntä
- Puskuroitu analoginen sisääntuloliitäntä ja PWM-ulostulo.
- Kytkenärima ulkoisia kytkentöjä varten

7 KEHITYSTYÖKALUJEN KÄYTTÖ

Opetuskorttia käytettiin kahdella erilaisella kehitysohjelmalla, IAR Embedded Workbenchillä sekä Eclipsellä. Seuraavassa on kerrottu kyseisten kehitysohjelmien käytöstä sekä siitä, min-kälaisia tuloksia kehitysohjelmien kanssa saatiin.

7.1 IAR Embedded workbench

Opetuskorttia suunniteltiin alun perin käytettäväksi IAR embedded workbenchillä. Kyseisellä ohjelmalla oli tarkoitus tehdä opetuskäyttöön soveltuvia esimerkkiohjelmaa sekä testata opetuskortin toimivuutta.

Käyttö aloitettiin tutustumalla opetuskortin mukana tulleeseen DVD:hen. Kyseiseltä DVD:ltä etsittiin Atmelin valmistama ”user guide”. Kyseinen user guide käännettiin suomen kielelle, sekä siitä tehtiin PowerPoint-esitys helpottamaan omaa sekä muiden mahdollisten käyttäjien käyttöä.

PowerPoint-esityksen valmistuttua asennettiin IAR embedded workbench sekä etsittiin DVD:ltä valmis esimerkkiohjelma, jolla voitaisiin testata kortin toimivuutta. Esimerkki ohjelma oli nimetty nimellä ”led swing”. Kyseisen ohjelman tarkoituksena oli sytyttää kortilla olevat ledit DS1-DS4, jolloin saataisiin aikaan ”juokseva” led-rivistö.

Kun ohjelma oli ladattu sekä asennettu IAR ebddeded workbenchiin, oli vuorossa sovelluksen kääntäminen. Opetuskortti yhdistettiin USB-liitännän kautta tietokoneeseen, jossa IAR Embedded Workbench oli.

Käännöksessä huomattiin ohjelman ilmoittama virhe, joka ilmoitti, ettei jtag-ice ole kytketty. Kyseinen ilmoitus aiheutti hämmennystä, koska ”user guiden” mukaan kääntämisen pitäisi onnistua myös USB-liitännän avulla.

Pohdittuani asiaa kytkin kortin Olimexin valmistaman jtag-icen avulla tietokoneeseen. Kyseinen jtag-ice:n puuttumisilmoitus toistui taas. Tutkittuani asiaa sain selville että käyttämäni IAR embedded workbenchin versio ei ollut yhteensopiva muun kuin Atmelin valmistaman

jtag-icen kanssa. Samalla selvisi myös, että DVD:n mukana tulleessa IAR embedded workbenchissä oli kääntäjän kääntämän koodin koko rajoitettu 32 kilotavuun.

Keskusteltuani kyseisistä ongelmista ohjaavan opettajan kanssa tulimme siihen tulokseen, ettei IAR embedded workbench ollut sopiva työväline tähän käyttöön johtuen pääasiassa juuri kääntäjän kääntämän koodin koon rajoituksen vuoksi. Opetuskäytössä edellä mainittu rajoitus olisi aiheuttanut ongelmia mielekkäiden ohjelmien teossa, koska käännettävät ohjelmat olisivat jääneet yksinkertaisiksi. Kortin antama potentiaali haluttiin kuitenkin hyödyntää, joten kyseisen kehitysohjelman käyttäminen hylättiin.

7.2 Eclipse

Tutkittuani muita mahdollisia opetuskäyttöön soveltuvia kehitysohjelmia päädyimme Eclipseen. Kyseisen kehitysohjelman vapaaseen lähdekoodiin perustuvuus ja sitä kautta ilmaisuus olisi opetuskäytössä erittäin hyvä ominaisuus. Aloitin Eclipseen käytön tutkimalla James P Lynchin kirjoittamaa “Using Open Source Tools for AT91SAM7S Cross Development revision C” opasta. [6.] Kyseisen oppaan laajuuden vuoksi käänsin oppaan suomen kielelle poimien sieltä tärkeimmät kohdat.

Oppaasta valmistettiin myös PowerPoint-esitys, jotta myös muiden käyttäjien olisi helpompaa käyttää kyseistä ohjelmaa. PowerPoint-esityksen valmistuttua aloin etsiä tarvittavia ohjelmia, jotta Eclipseen käyttö olisi mahdollista. Käytettävien ohjelmien tarve riippuu myös siitä tekijästä, käytetäänkö USB-OCD-debuggeria vai ”wiggler”-debuggeria. Alla olevassa kuvassa on esitelty sekä USB-OCD-debuggeri että ”wiggler”-mallin debuggeri.



Kuva 10. ARM-USB-OCD sekä wiggler-tyyppinen debuggeri [12.]

Etsittyäni sopivat ohjelmat Eclipsen käyttöä varten asensin ne. Alla olevassa taulukossa on esitelty ohjelmat, jotka tarvitaan ARM-7-mikrokontrolleria varten käytettäessä ARM-USB-OCD-tyyppistä debuggeria. Wiggler-tyypin debuggeria käytettäessä tarvitaan myös Atmelin SAM-BA-ohjelmisto. Kyseinen ohjelma mahdollistaa flash-muistiin kirjoituksen käytettäessä kyseisentyyppistä debuggeria.

Taulukko 6. Eclipseä varten tarvittavat ohjelmat

Eclipse IDE (versio 3.2 tai uudempi)
Eclipse CDT plugin (C/C++ kääntämistä varten)
Native Gnu C++/C compiler suite for ARM targets(C++/C- kielen kääntämistä varten)
OpenOCD (JTAG-KEY tai ARM-USB-OCD debuggausta varten)

Alla olevassa taulukossa on esitelty kyseisten ohjelmien internet-latausosoitteet.

Taulukko 7. Ohjelmien internet-latausosoitteet

Eclipse IDE for C/C++ Developers: http://www.yagarto.de/#download
Eclipse CDT plugin http://www.yagarto.de/#download
Native Gnu C++/C compiler suite for ARM targets http://www.yagarto.de/#download
OpenOCD (JTAG-KEY tai ARM-USB-OCD debuggausta varten) http://www.yagarto.de/#download

Asennettuani sopivat ohjelmat aloitin sopivien konfigurointitiedostojen etsimisen, jotta Eclipsen käyttö opetuskortilla olisi mahdollista. Käytettävästä debuggerista riippuen tarvitaan jokaiselle debuggerille oma konfigurointitiedosto, .cfg-tiedosto. Kyseinen tiedosto voidaan

tehdä itse esim. notepadillä, tai kyseinen tiedosto voidaan etsiä internetiä apuna käyttäen. Alla olevassa kuvassa on esimerkki valmiista .cfg-tiedostosta.

Open OCD .cfg file

```
#define our ports
telnet_port 4444
gdb_port 3533

#commands specific to the Olimex ARM-USB-OCD
interface ft2232
ft2232_device_desc "Olimex OpenOCD JTAG A"
ft2232_layout "olimax-jtag"
ft2232_vid_pid 0x15BA 0x0003
jtag_speed 2
jtag_nsrst_delay 200
jtag_ntrst_delay 200

#reset_config <signals> [combination] [trst_type] [srst_type]
reset_config srst_only srst_pulls_trst

#jtag_device <IR length> <IR capture> <IR mask> <IDCODE instruction>
jtag_device 4 0x1 0xf 0xe

#daemon_startup <'attach'|'reset'>
daemon_startup reset

#target <type> <endianess> <reset_mode> <jtag#> [variant]
target arm7tdmi little run_and_init 0 arm7tdmi_r4

#run_and_halt_time <target#> <time_in_ms>
run_and_halt_time 0 30

# commands below are specific to AT91sam7 Flash Programming
# -----

#target_script specifies the flash programming script file
target_script 0 reset script.ocd

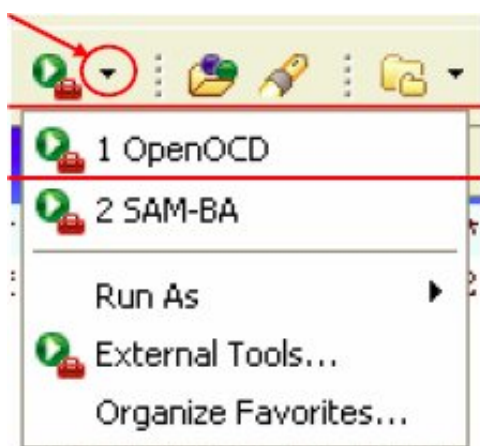
#working_area <target#> <address> <size> <'backup'|'nobackup'>
working_area 0 0x40000000 0x4000 nobackup

#flash bank at91sam7 0 0 0 0 <target#>
flash bank at91sam7 0 0 0 0
```

Kuva 11. .cfg-tiedosto.

Kyseiseen tiedostoon määritellään käyttöön tarvittavat parametrit sekä Eclipsen käyttämät portit. Kyseisellä tiedostolla voidaan esimerkiksi määrittellä jtagin nopeus sekä prosessoriin liittyvät asetukset.

Seuraavaksi aloitin Eclipsen käytön valmistelun. Eclipsen käytössä on monia erilaisia pieniä ominaisuuksia, jotka on hyvä tehdä, jotta käytöstä tulisi mahdollisimman helppoa. Joissakin tapauksissa käyttö ei ole edes mahdollista, jos tarvittavia asetusten muutoksia ei tehdä. Esimerkki edellä mainituista asetuksista on käyttöperspektiivin vaihto. Kyseinen perspektiivi on vaihdettava C/C++-perspektiiviksi, jos ohjelmointikielenä käytetään jompaakumpaa edellä mainituista kielistä. Käytettäessä erilaista koodikieltä sovelluksen kääntö ei onnistu. Käyttöä helpottavia asetuksia ovat esimerkiksi suosikkityökalujen järjestäminen pikavalikkoon. Alla olevassa kuvassa on esimerkki toteutetusta kyseisestä toiminnosta.



Kuva 12. Työkalujen järjestely

Tehtyäni käyttöä helpottavat/ ja mahdollistavat asetukset aloitin sopivien esimerkkiohjelmien etsimisen. Sopiva esimerkki löytyi myös James P Lynchin valmistamana. Kyseinen ohjelma ei vaatinut muita muutoksia kuin prosessorin ja itse kortin kirjastojen vaihtamisen sopivanlaiseksi.

Käännettävän sovelluksen tekeminen aloitetaan luomalla projekti, johon voidaan tuoda valmiit tiedostot jostakin toisesta projektista Import-komennolla. Tarvittavat tiedostot voidaan tehdä myös itse, mutta aloittelijan on helpompaa käyttää valmista projektia pohjana, jotta käyttökynnys Eclipseen ei tulisi liian korkeaksi. Projektissa tarvitaan monia erilaisia tiedostoja riippuen siitä ollaanko tekemässä flash- vai RAM-sovellusta. Esimerkkejä kyseisistä tiedostoista ovat Eclipseen ”objcopyn” luoma main.out-sekä main.bin-tiedosto. Ensin mainittua tiedostoa tarvitaan RAM-sovelluksessa sekä jälkimmäistä flash-sovelluksessa.

Eclipsellä pystytään kääntämään sekä RAM- että flash-sovelluksia. RAM-sovellusten kääntämisessä on myös omat nikkinsä. Kyseisten toimintojen jättämättä tekeminen tekee sen, että kääntäminen ei ole mahdollista. Alla olevassa kuvassa on esimerkki kyseisistä toiminnoista, ”Run Commands”.

```
monitor soft_reset_halt
monitor armv4_5 core_state arm
monitor mww 0xffffffff60 0x00320100
monitor mww 0xfffffd44 0xa0008000
monitor mww 0xffffc20 0xa0000601
monitor wait 100
monitor mww 0xffffc2c 0x00480a0e
monitor wait 200
monitor mww 0xffffc30 0x7
monitor wait 100
monitor mww 0xffffd08 0xa5000401
set remote memory-write-packet-size 1024
set remote memory-write-packet-size fixed
set remote memory-read-packet-size 1024
set remote memory-read-packet-size fixed
monitor mww 0xffffd00 0xa5000004
monitor mww 0xfffff00 0x01
monitor reg pc 0x00000000
monitor arm7_9 sw_bkpts enable
load
continue
```

Kuva 13. RAM-sovelluksen ajokäskyt.

Kun tarvittavat muutokset koodiin oli tehty, oli vuorossa ohjelman kääntäminen ja ajaminen opetuskorttiin. Muutaman korjaus- sekä ajokerran jälkeen debuggeri ei antanut enää virheilmoituksia itse koodista. Ohjelma- debuggerin käynnistyttyä käynnistin myös jtag-icen yhteyden korttiin, jolloin ohjelma antoi virheilmoituksen. Vikaa yritettiin korjata etsimällä mahdollisia virheitä koodista sekä itse Eclipsen asetuksista. Tutkittuamme asiaa päädyimme siihen tulokseen, ettei Olimexin jtag-ice ollut jostain syystä yhteensopiva opetuskortin kanssa. Tämä oli kummallista, koska aiemmin oli katsottu, että kyseisen jtag-icen pitäisi olla yhteensopiva kaikkien ARM-prosessoriydinten kanssa. Kortin käyttö jäi tältä osin kyseiseen pisteeseen ajan puutteen takia.

7.3 Korjausehdotukset

Opetuskortin toimivuuden kannalta soveliaain vaihtoehto olisi ottaa yhteyttä yritykseen, jolta Olimexin jtag-ice tilattiin. Kyseiseltä yritykseltä voisi tiedustella, onko vastaavia ongelmia esiintynyt ja mitkä olisivat korjausehdotukset. Jos yrityksen mielestä kyseinen jtag-ice on so- piva, seuraava vian etsimisen kohde tulisi olla jtag-icen konfigurointitiedosto tai RAM- sovelluksen ajokäskyjen tarkastaminen.

8 TULOSTEN ANALYSOINTI

Työn alussa eniten vaivaa tuotti sopivan kehitysohjelman valinta. Mukana tullessa IAR embedded workbench -kehitysohjelmassa oli se ongelma, että sen kääntämän koodin koko oli rajoitettu 32 kilotavuun. Kyseisestä ohjelmasta päätettiin luopua juuri edellä mainitun ongelman takia sekä osaltaan myös siksi, ettei Olimexin tilattu jtag-ice-debuggeri pystynyt toimimaan kyseisen ohjelman kanssa. Käytettäväksi kehitysohjelmaksi valittiin Eclipse, joka oli soveliain vaihtoehto monipuolisuuden sekä ilmaisuutensa takia. Eclipsen sekä tarvittavien apuohjelmien asennus sujui vaivattomasti, mutta itse Eclipsen käyttö olikin jo haastavampaa.

Eclipsen käyttöönottoa helpotti James P. Lynchin tekemä opas, mutta senkään lukeminen ei ollut hetken asia, koska kyseinen opas sisälsi n. 200 sivua. Juuri kyseiselle opetuskortille soveltuvan käyttöoppaan puuttuminen hankaloitti asioita, koska mahdollisia eroja oppaassa käytetyn kortin sekä opetuskortin välillä piti pohtia.

Ohjelmointia varten tarvittavien erilaisten kirjastojen etsimisessä meni myös aikansa, kuten myös sen selvittämisessä, mitä kirjastoja tarkalleen ottaen tarvitaan. Kirjastojen sisältöön tutustuminen vei myös oman aikansa.

Kirjallista opetusmateriaalia saatiin tuotetuksi IAR Embedded Workbenchin sekä Eclipsen käytöstä. Camtasia Studion käyttö ei toteutunut johtuen siitä, ettei opetuskortilla saatu ohjelmoitua sovelluksia, joiden tekoon Camtasia Studiota olisi käytetty.

Eclipsen laajuus osoittautui työn vaikeutta lisääväksi tekijäksi, eikä opetuskorttia saatu sovelusten tekemisen kannalta käyttöönotettua sovitun ajan puitteissa.

9 YHTEENVETO

Opetuskortin käyttöönoton tarkoituksena oli käyttää tiettyä kehitysohjelmaa sekä tehdä sille jonkinlaisia esimerkkiohjelmia, joita voitaisiin käyttää opetuksessa. Tarkoituksena oli myös tehdä muuta opetusmateriaalia liittyen käytettäviin työvälineisiin. Opetusmateriaalia oli tarkoitus apuna käyttäen Microsoft PowerPointia sekä Techsmithin Camtasia Studiota.

Opetuskortin käyttöönotto jäi vaillinaiseksi ohjelmoinnin kannalta, koska opetuskortille ei saatu ajettua yhtään ohjelmaa kovasta yrityksestä huolimatta. Eclipseä saatiin tehtyä testiohjelma opetuskorttiin, mutta ohjelmaa ei saatu ajettua kortille. Kyseiseen ilmiöön on syynä valitun Eclipse-kehitysohjelman vaatima suuri työpanos, jonka takia aika loppui siltä osin kesken. Kyseisen kehitysohjelman vaatii suuren määrän opiskelua asetusten sekä tarvittavien asetustiedostojen tekemisen myötä.

Opetuskortille tehtiin tarvittavat korjausehdotukset, jotta opetuskortti saataisiin toimimaan halutulla tavalla.

Työssä käytettävistä IAR Embedded Workbench sekä Eclipse kehitysohjelmista saatiin valmistettua kirjallista materiaalia opetuskäyttöön.

LÄHTEET

- 1 TIE 530 Prosessorit/Kontrollerit [WWW-dokumentti]
<<http://users.jyu.fi/~jupeihal/Kontrollerit1.pdf>>
(luettu 15.3.2008)
- 2 Rantala, P. Tietokonetekniikka osa 2. Mikrotietokonetekniikka.

Tietokotka.119 s.

ISBN-951-559-258-5
- 3 Embedded development tools from IAR Systems [WWW-dokumentti]
<<http://www.iar.com/website1/1.0.1.0/14/1/index.php>>
(luettu 29.4.2008)
- 4 Mikä avoin lähdekoodi on? [WWW-dokumentti]
<http://savonlinna.joensuu.fi/eom/tekno/R2_verkkojulkaisu/avoin/avoin_lahdekoodi.html>
(luettu 29.4.2008)
- 5 Open source testaustyökalut [WWW-dokumentti]
<http://www.swbusiness.fi/uploads/attachments/1158750778_Open_Source_Testaus_tyokalut_Ilkka_Myllyla_200906.pdf>
(luettu 29.4.2008)
- 6 Using Open Source Tools for AT91SAM7S Cross Development

[WWW-dokumentti]
<http://gnuarm.alexthegreek.com/atmel/Using_Open_Source_Tools_for_AT91SAM7S_Cross_Development_revision_C.pdf>
(luettu 20.2.2008)
















- 7 ARM Milestones [WWW-dokumentti]
<<http://www.arm.com/aboutarm/milestones.html>>
(luettu 15.3.2008)
- 8 Pietikäinen, V.2001.ARM arkkitehtuuri [WWW-dokumentti]
<http://tisu.it.jyu.fi/embedded/TIE345/vanhat_luentokalvot/ARM_3.pdf>
(luettu 15.3.2008)
- 9 Processor Licensees [WWW-dokumentti]
<<http://www.arm.com/products/licensing/licencees.html>>
(luettu 16.3.2008)
- 10 Other Voice Phones-ARM Powered Products [WWW-dokumentti]
<http://www.arm.com/markets/mobile_solutions/armpp/835.html>
(luettu 16.3.2008)
- 11 Atmel Corporationin mukana tullut AT91SAM7X-EK–tuotepakettiin sisältyvä DVD
- 12 Development and Prototype boards and tools for pic avr and msp 340 [WWW-dokumentti]
<<http://www.olimex.com/dev/index.html>>
(luettu 24.3.2008)

LIITTEET

1 ARM7-rekisterikanta


2 ARM7-käskykanta

3 Atmel AT91SAM7X-EK konfigurointi hyppy-kytkimet

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	 R8_fiq	R8	R8	R8	R8
R9	 R9_fiq	R9	R9	R9	R9
R10	 R10_fiq	R10	R10	R10	R10
R11	 R11_fiq	R11	R11	R11	R11
R12	 R12_fiq	R12	R12	R12	R12
R13	 R13_fiq	 R13_svc	 R13_abt	 R13_irq	 R13_und
R14	 R14_fiq	 R14_svc	 R14_abt	 R14_irq	 R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

ARM State Program Status Registers

CPSR	CPSR  SPSR_fiq	CPSR  SPSR_svc	CPSR  SPSR_abt	CPSR  SPSR_irq	CPSR  SPSR_und
------	--	--	--	--	--

 = banked register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Cond	0	0	1	Opcode				S	Rn	Rd	Operand 2						<i>Data Processing / PSR Transfer</i>					
Cond	0	0	0	0	0	0	A	S	Rd	Rn	Rs	1	0	0	1	Rm	<i>Multiply</i>					
Cond	0	0	0	0	1	U	A	S	RdHi	RdLo	Rn	1	0	0	1	Rm	<i>Multiply Long</i>					
Cond	0	0	0	1	0	B	0	0	Rn	Rd	0	0	0	0	1	0	0	1	Rm	<i>Single Data Swap</i>		
Cond	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	0	1	Rn	<i>Branch and Exchange</i>
Cond	0	0	0	P	U	0	W	L	Rn	Rd	0	0	0	0	1	S	H	1	Rm	<i>Halfword Data Transfer: register offset</i>		
Cond	0	0	0	P	U	1	W	L	Rn	Rd	Offset			1	S	H	1	Offset	<i>Halfword Data Transfer: immediate offset</i>			
Cond	0	1	1	P	U	B	W	L	Rn	Rd	Offset						<i>Single Data Transfer</i>					
Cond	0	1	1												1		<i>Undefined</i>					
Cond	1	0	0	P	U	S	W	L	Rn	Register List						<i>Block Data Transfer</i>						
Cond	1	0	1	L	Offset											<i>Branch</i>						
Cond	1	1	0	P	U	N	W	L	Rn	CRd	CP#	Offset				<i>Coprocessor Data Transfer</i>						
Cond	1	1	1	0	CP Opc			CRn	CRd	CP#	CP	0	CRm	<i>Coprocessor Data Operation</i>								
Cond	1	1	1	0	CP Opc			L	CRn	Rd	CP#	CP	1	CRm	<i>Coprocessor Register Transfer</i>							
Cond	1	1	1	1	Ignored by processor											<i>Software Interrupt</i>						

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Section 4

Configuration Straps

4.1 Configuration Straps

Table 4-1. Configuration Jumpers and Straps

Designation	Default Setting	Feature
J8	Opened	Erases all internal Flash memory when the board is powered. To do that, the user will have to close the J8 at least 200 ms.
J9	Opened	Do not use: Factory test mode. J9 is not populated
J10	Opened	Selects ICE mode or JTAG mode (Closed). J10 is not populated.
J11	Closed	VDDIN Jumper ⁽¹⁾
J12	Closed	VDDFLASH Jumper ⁽¹⁾
J13	1-2	ADVREF Jumper select 1-2: 3.00V Voltage reference 2-3: VDDANA
J14	Closed	VDDCORE Jumper ⁽¹⁾
J15	Closed	VDDIO Jumper ⁽¹⁾
J17	Opened	External XIN clock input. S4 and S5 must be open. J17 is not populated.
J18	Closed	VDDPLL Jumper ⁽¹⁾
J19	Closed	Enables the use of the NPC500 (PA12).
J21	Closed	Enables 120 ohms CAN bus resistance termination.
J28	Closed	Enables Ethernet Auto MDIX control.
S1	Closed	Enables permanent pull up on USB DP.
S2	Closed	The System Reset signal (NRST) is connected to the ICE/JTAG socket (J7, pin 15).
S3	Opened	Disables 5V (VUSB) power supply on J16 extension connector.

Configuration Straps

Table 4-1. Configuration Jumpers and Straps (Continued)

Designation	Default Setting	Feature
S4 - S5	Closed	Enables the use of 18.432MHz crystal. Must be open if an external clock is used.
S6	Closed	Enables the Power Led control (PB25).
S7	Opened	Disables Serial DataFlash write protect.
S8	Closed	Enables the use of the TXD CAN transceiver (PA20)
S9	Closed	Enables the use of the RXD CAN transceiver (PA19)
S10	Closed	Enables control of the Standby/Normal mode for CAN transceivers (PA2)
S11	Opened	Enables control of the Standby/Normal mode for CAN transceivers (PA2). If S11 is closed, S10 must be open.
S12	Closed	Enables the use of PWM3 Analog Output (PB30)
S13	Closed	Enables the use of the TXD0 signal (PA1)
S14	Closed	Enables the use of the RTS0 signal (PA3)
S15	Closed	Enables the use of the RXD0 signal (PA0)
S16	Closed	Enables the use of the CTS0 signal (PA4)
S17	Closed	Enables the use of the User LED DS1 (PB19)
S18	Closed	Enables the use of the User LED DS2 (PB20)
S19	Closed	Enables the use of the User LED DS3 (PB21)
S20	Closed	Enables the use of the User LED DS4 (PB22)
S21	Closed	Enables the use of the DBGU TXD signal (PA28)
S22	Closed	Enables the use of the DBGU RXD signal (PA27)
S23	Opened	ETHERNET MII is the default mode. To evaluate the RMII mode, the user change S23 to S26 configuration in the following way: S23 Closed, S24 Opened, S25 Closed, S26 Opened
S24	Closed	
S25	Opened	
S26	Closed	
S27	Opened	Reserved
S28	Closed	Enables the use of the SCL of MN11 (PA11)
S29	Closed	Enables the use of the SDA of MN11 (PA10)
TP1	N.A	GND Test point.
TP2	N.A	GND Test point.

Note: 1. These jumpers are provided for measuring power consumption. By default, they are closed. To use this feature, the user has to open the strap and insert an anmeter.