

Matti Pernu

Linuxin käyttöönotto LIIKE-kortilla

Insinöörityö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2013



| | |
|---|---|
| Koulutusala Tekniikan ja liikenteen ala | Koulutusohjelma Tietotekniikan koulutusohjelma |
| Tekijä(t) Matti Pernu | |
| Työn nimi Linuxin käyttöönotto LIIKE-kortilla | |
| Vaihtoehdot Ohjelmointi | Ohjaaja(t) Arto Partanen Toimeksiantaja Markku Karppinen / Kajaanin ammattikorkeakoulu |
| Aika Kevät 2013 | Sivumäärä ja liitteet 25 |
| <p>Tämän työn tavoitteena oli modifioida ja saada Linux-käyttöjärjestelmä käynnistymään LIIKE-mittauskortilla, joka oli muunneltu tarkoitukseen sopivaksi käyttäen pohjana AT91SAM9260-EK-mikrokontrollerikorttia. Kortin tarkoituksena on toimia mittalaitteena hiihtovalmennuksen tukena.</p> <p>Työn alussa on esitelty käytetyt laitteet eli LIIKE-mittauskortti sekä SAM-ICE-debuggeri. Ohjelmiston osalta on esitelty Linuxin Ubuntu-distribuutio, SAM Boot Assistant -ohjelma sekä VMware Player -virtualisointiohjelmisto.</p> <p>Työssä käytettiin hyväksi Git-versionhallintaohjelmalla haettuja valmiita esimerkkikoodeja, joita muokattiin kortille sopivaksi. Työn suorittamiseen tarvittiin kernelin lisäksi Bootstrap, U-Boot sekä root filesystem, jotka muokattiin ja käännettiin Ubuntu Linuxissa toimivaksi kokonaisuudeksi. Windows 7 -käyttöjärjestelmän alla toimineella SAM Boot Assistant -ohjelmalla saatiin käännettyt tiedostot siirrettyä kortin muistiin oikeille paikoilleen, jolloin kortti saatiin käynnistymään käännettyyn Linux-käyttöjärjestelmään.</p> <p>Testaus suoritettiin Windows 7 -käyttöjärjestelmään asennetulla Tera Term -terminaali-ohjelmalla, jolla saatiin luettua mikrokontrollerikortin sarjaportista kernelin käynnistymisen yhteydessä lähetetyt tiedot. Lopputuloksena kortille saatiin käynnistymään Linux-käyttöjärjestelmä.</p> | |
| Kieli | Suomi |
| Asiasanat | Linux, käyttöjärjestelmät, sulautettu tietotekniikka |
| Säilytyspaikka | <input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto |

| | |
|---|---|
| School Engineering | Degree Programme Information Technology |
| Author(s) Matti Pernu | |
| Title Deployment of Linux on LIIKE-card | |
| Optional Professional Studies Software Engineering | Instructor(s) Mr Arto Partanen |
| | Commissioned by Mr Markku Karppinen, Development Engineer / Kajaani University of Applied Sciences |
| Date Spring 2013 | Total Number of Pages and Appendices 25 |
| <p>The purpose of this Bachelor's thesis was to modify and start up a Linux operating system on the LIIKE measurement card, which is based on the AT91SAM9260-EK microcontroller card and was modified to suit its purpose. The card is meant to support ski coaching as a measuring device.</p> <p>The beginning of this thesis introduces the used equipment, the LIIKE measurement card and a SAM-ICE debugger. The software introduced in this thesis is the Linux Ubuntu distribution, a SAM Boot Assistant and VMware Player virtualization software.</p> <p>The existing code samples received via the Git distributed revision controlling program were modified to suit the purposes of the card. In addition to the kernel, a Bootstrap, a U-Boot and a root file system were modified and compiled in order to make a functioning package. The card was able to boot up a functioning operating system after the compiled files were transferred to the memory of the card with the SAM Boot Assistant which was running under the Windows 7 operating system.</p> <p>The testing was performed with a Tera Term terminal emulator installed in the Windows 7 operating system. It was used to read the data sent via serial port of the microcontroller card during the boot of the kernel.</p> | |
| Language of Thesis | Finnish |
| Keywords | Linux, operating systems, embedded information technology |
| Deposited at | <input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences |

SISÄLLYS

| | |
|---|----|
| 1 JOHDANTO | 1 |
| 2 LAITTEET JA OHJELMISTOT | 2 |
| 2.1 Työssä käytetyt laitteet | 2 |
| 2.1.1 AT91SAM9260 LIIKE -mittauskortti | 2 |
| 2.1.2 Atmel SAM-ICE | 2 |
| 2.2 Työssä käytetyt ohjelmat | 2 |
| 2.2.1 VMware Player | 2 |
| 2.2.2 Ubuntu Linux | 3 |
| 2.2.3 SAM-BA | 3 |
| 3 KÄÄNTÄMISYMPÄRISTÖN KÄYTTÖÖNOTTO | 4 |
| 3.1 SAM-BA:n asennus | 4 |
| 3.2 Tera Termin asennus | 4 |
| 3.3 VMware Playerin asennus | 5 |
| 3.4 Ubuntu Linuxin asennus virtuaalikoneena | 6 |
| 3.5 LIIKE-kortin kytkentä | 9 |
| 4 YTIMEN KÄÄNTÄMINEN | 10 |
| 4.1 Bootstrap | 12 |
| 4.2 U-Boot | 13 |
| 4.3 Linux-kernel | 15 |
| 4.4 Root filesystem | 16 |
| 5 YTIMEN SIIRTO KORTILLE | 18 |
| 6 TULOKSET JA TESTAAMINEN | 21 |
| 7 YHTEENVETO | 24 |
| LÄHTEET | 25 |

SYMBOLILUETTELO

| | |
|---------|---|
| ARM | Advanced RISC Machines, 32-bittinen RISC-proessoriarkkitehtuuri |
| GNU | GNU's Not Unix. Projekti, jonka tavoitteena on kehittää täysin vapaista ohjelmista koostuva käyttöjärjestelmä |
| GNU GPL | GNU General Public License. GNU yleinen lisenssi, vapaa ohjelmistolisenssi |
| GIT | Linuxissa käytettävä versionhallintaohjelma |

1 JOHDANTO

Insinööriyö tehtiin Kajaanin ammattikorkeakoululle. Se on vuonna 1992 perustettu Kajaanin kaupungin omistama liikelaitos, jossa opiskelee n. 2000 opiskelijaa. Koulutusta tarjotaan 5 eri koulutusosalalla ja 12 eri koulutusohjelmalla, joista 3 on englanninkielistä. Vuosittain Kajaanin AMK:ssa aloittaa n. 400 uutta perustutkinto-opiskelijaa, ja vuosittain aikuisopiskelijoita on n. 280. [1.]

Insinööriyön tavoitteena on kääntää ja ottaa käyttöön Linux-käyttöjärjestelmä AT91SAM9260-EK-mikroprosessorikortin pohjalle rakennetulla LIIKE-kortilla. LIIKE-kortti on toteutettu tutkimuskäyttöön hiihdon mittauksiin LIIKE-hankkeessa. LIIKE-kortilla voidaan mitata 12 EMG-signaalia ja 24 voimasignaalia. Tulokset voidaan lähettää reaaliajassa kortilla olevan RF-moduulin avulla serverille.

Ytimen eli kernelin kääntämistä varten tehdään siihen sopiva ympäristö asentamalla PC-tietokoneelle VMWare-virtuaalikone, johon asennetaan Linuxin Ubuntu-versio. Ubuntulle haetaan ytimen lähdekoodi ja käännetään se käyttäen mikrokontrollerikortille sopivia asetuksia. Tämän jälkeen ydin siirretään mikrokontrollerikortille ja testataan toimivaksi.

2 LAITTEET JA OHJELMISTOT

2.1 Työssä käytetyt laitteet

2.1.1 AT91SAM9260 LIIKE -mittauskortti

Työssä käytettiin LIIKE-mittauskorttia, joka oli muunneltu käyttötarkoitukseensa sopivaksi AT91SAM9260-EK-kortin pohjalle. Laitteeseen kuului ARM9-pohjainen AT91SAM9260-mikroprosessori n. 200 MHz ARM926EJ-S-ytimellä, ja muisteina toimivat 64 MB SDRAM sekä 128 MB NANDFLASH. Lisäksi löytyi paikka mahdolliselle SD-muistikortille. Muita mainitsemisen arvoisia toiminnallisuuksia kortilla ovat GPS-moduuli, ilmanpainemittari, RF-radiomoduuli mittaustulosten reaaliaikaiseen lähetykseen sekä 2 kappaletta ADS1296-muunninta EMG-mittauksiin ja 3 kappaletta ADS1298-muunninta lihasten mittauksiin.

2.1.2 Atmel SAM-ICE

Työssä käytettiin debuggerina Atmel SAM-ICE (AT91SAM-ICE) JTAG -emulaattoria, joka on suunniteltu ARM -prosessoreille. ICE yhdistetään USB-väylän kautta PC-tietokoneeseen ja 20-pinnisellä JTAG-liittimellä LIIKE-korttiin. Yhdistetyn ICE:n kautta siirretään käännetty Linux-ydin PC-tietokoneelta LIIKE-kortille.

2.2 Työssä käytetyt ohjelmat

2.2.1 VMware Player

VMware Player on internetistä löytyvä kotikäyttöön ilmainen virtualisointiohjelma. Versiosta 3.0 lähtien VMware Playerillä on voitu luoda virtuaalikoneita, mihin ennen tarvittiin erillinen ohjelma. Tähän työhön valittiin VMware Playeristä uusin versio, eli 5.0.1 build-894247.

2.2.2 Ubuntu Linux

Linuxin kehitys alkoi vuonna 1990. Käyttöjärjestelmän isänä tunnettu Linus Torvalds ei ollut tyytyväinen MS-DOS- ja Windows-käyttöjärjestelmiin. Unix kiinnosti häntä, sillä siinä oli sellaisia ominaisuuksia ja vakautta, jota hän oli hakenut. Unix oli käyttöjärjestelmänä kuitenkin liian raskas ja kallis PC-käyttöön. Opiskeluaikanaan Linus sai käsiinsä Minixin, PC:lle tarkoitettun Unix-tyylisen käyttöjärjestelmän, joka oli suunniteltu opetuskäyttöön ja jonka lähdekoodi oli käytettävissä.

Käyttäen Minix:iä pohjana ja ottaen siitä hyviä ideoita Linus aloitti Linux-käyttöjärjestelmän luomisen. Tarkoituksena ei ollut tehdä käyttöjärjestelmää koko maailmalle, vaan ainoastaan hänen omiin tarpeisiinsa. Vuonna 1991 valmistui ensimmäinen toimiva käyttöjärjestelmän ohjelmakoodi ja Linus päätti laittaa sen vapaasti kaikkien saataville internetiin. [2.]

Sittemmin Linuxiin on tullut sen vapaan lähdekoodin, ns. GNU-projektin, ansiosta lukuisia eri jakeluversioita, joista yksi on tähän työhön valittu Ubuntu Linux. Ubuntu'n ensimmäinen julkaisu ilmestyi 20. lokakuuta 2004.

Linux-ydin on julkaistu GNU GPL- eli General Public -lisenssillä (GNU yleinen julkinen lisenssi). Lisenssin on alun perin kirjoittanut Richard Stallman GNU-projektille vuonna 1989. GNU GPL on vapaan levitysoikeuden omaava lisenssi ohjelmistoille ja muille töille. Lisenssi vaatii siitä periytyvien töiden julkaisemista saman lisenssin alaisena, vaikka lisenssin alaista työtä olisi muutettu tai lisätty.

2.2.3 SAM-BA

SAM-BA eli SAM Boot Assistant on ohjelma, jolla saadaan siirrettyä helposti AT91-laitteiden muistiin erilaisia ohjelmia. Korttien kanssa kommunikointiin SAM-BA voi käyttää RS232:sta, USB:tä tai JTAG:ia. SAM-BA:a voidaan käyttää joko komentoriviltä tai graafisen käyttöliittymän kautta. Työhön valittiin SAM-BA:n versio 2.8.

3 KÄÄNTÄMISYMPÄRISTÖN KÄYTTÖÖNOTTO

Kääntämisympäristö asennettiin PC-tietokoneelle, jonka tärkeimmät tiedot näkyvät alla olevassa taulukossa 1.

Taulukko 1. Asennusalustana toimineen PC-tietokoneen tiedot

| | |
|-------------------|----------------------------------|
| Proessori | AMD Phenom II X6 1055T, 3,15 GHz |
| Keskusmuisti | 8 GB |
| Näytönohjain | NVIDIA Geforce GTX460 |
| Käyttöjärjestelmä | Windows 7 Professional, 64-bit |

3.1 SAM-BA:n asennus

SAM-BA:n asennus aloitettiin hakemalla internetistä siihen asennuspaketti. Versioksi valittiin Windows 7 -käyttöjärjestelmään käypä Atmelin AT91-ISP v. 1.12 -paketti, joka sisälsi SAM-BA:n version 2.8 [3]. Asennus tapahtui suoraviivaisesti oletusasetuksilla painelemalla Next-nappia. Asennuksen jälkeen PC käynnistettiin uudestaan, jotta ohjelma toimisi ongelmitta.

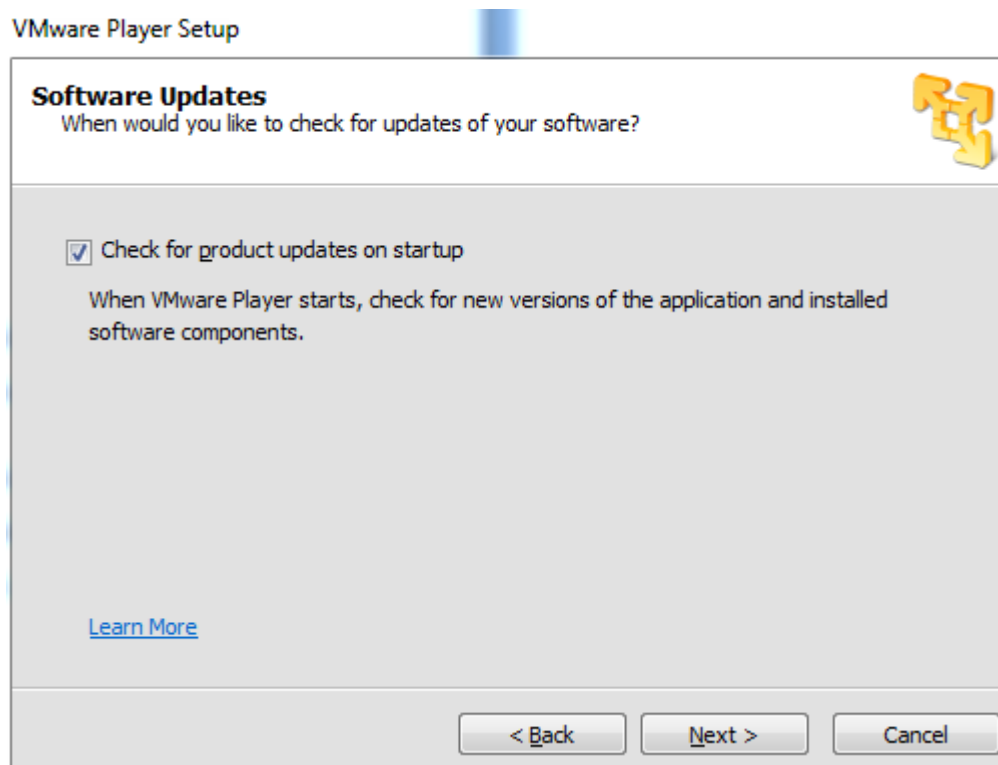
3.2 Tera Termin asennus

Tera Term on yksinkertainen ilmainen terminaaliohjelma, jolla saadaan otettua Telnet, SSH tai sarjaporttityhteys kohdelaitteen kanssa. Tera Termin asennus aloitettiin hakemalla internetistä asennustiedosto. Versioksi valittiin 4.77 [4]. Tera Term asennettiin Windows 7 -käyttöjärjestelmälle oletusasetuksilla SAM-BA:n tavoin painelemalla ”Next”-nappia.

3.3 VMware Playerin asennus

VMware Playerin asennus aloitettiin hakemalla internetistä siihen asennuspaketti. Tuorein 32- ja 64-bittisille Windows-käyttöjärjestelmille sopivan paketin versio oli 5.0.1 [5].

VMware Playerin ohjattu asennus oli hyvin yksinkertainen. Aluksi valittiin asennuskansio, johon ohjelma haluttiin. Tämän jälkeen sai valita, halusiko ohjelman käynnistyksen yhteydessä tarkistaa mahdolliset päivitykset. Tähän jätettiin rasti ruutuun kuvan 1 mukaisesti, jotta pystyttiin varmistumaan ohjelman olevan ajan tasalla.



Kuva 1. VMWare-playerin ohjattu asennus

Seuraavaksi valittiin, haluttiinko auttaa VMware Playerin kehittämisessä. Mikäli rasti jätetään kohtaan ”Help improve VMware Player”, ohjelma lähettää nimettömänä dataa ja käyttöstatistiikkoja ohjelman kehittäjälle. Tässä tapauksessa rastia ei jätetty ruutuun.

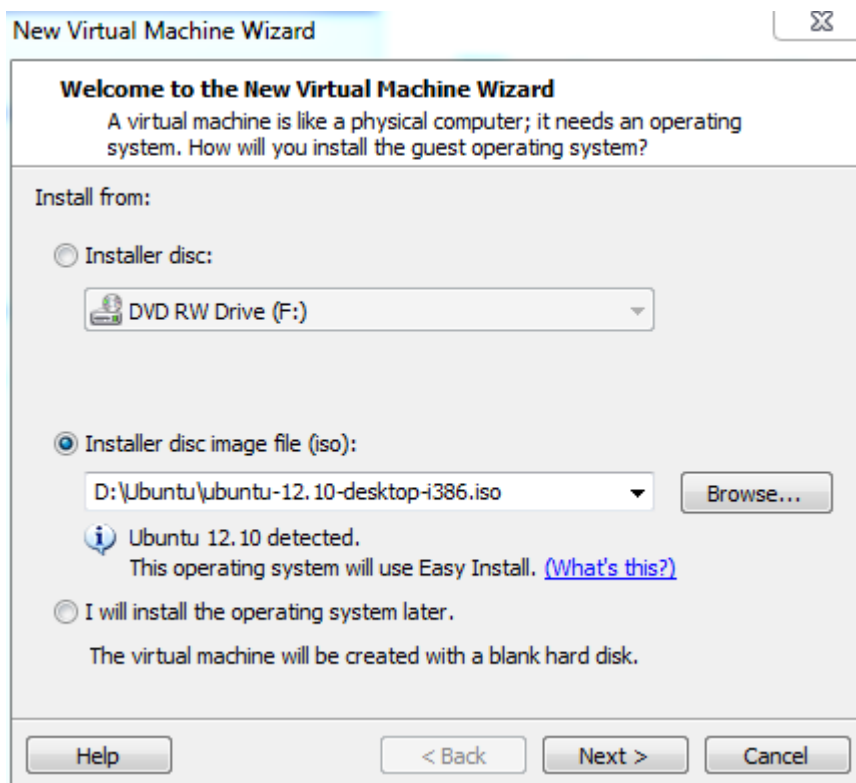
Tämän jälkeen valittiin asennuksen yhteydessä työpöydälle sekä käynnistä-valikkoon lisättäväksi pikakuvakkeet ohjelman käynnistämiseen.

3.4 Ubuntu Linuxin asennus virtuaalikoneena

Kun ohjelma virtuaalikoneiden asennusta varten oli saatu asennettua, oli aika siirtyä asentamaan Linux-kernelin kääntämiseen sopivaa alustaa. Tähän valittiin Ubuntu Linux, jonka asennus virtuaalikoneeksi tehtiin seuraavasti.

Linux avoimen lähdekoodin ja GNU GPL:n alaisena käyttöjärjestelmänä on helppo löytää internetistä ladattavaksi kotikoneelle. Linux Ubuntuun versioksi valittiin 32-bittinen 12.10 [6]. Ubuntu ladattiin valmiina levykuvana eli imagena, jolloin se saatiin kätevästi asennettua suoraan virtuaalikoneeksi.

Asentaminen aloitettiin käynnistämällä VMware Player ja valitsemalla uuden virtuaalikoneen luominen näppäinyhdistelmällä Ctrl+N. Asennuslähteeksi valittiin levykuva (.iso) ja ”Browse”-napista selattiin lähteeksi imuroitu levykuva kuvan 2 mukaisesti.

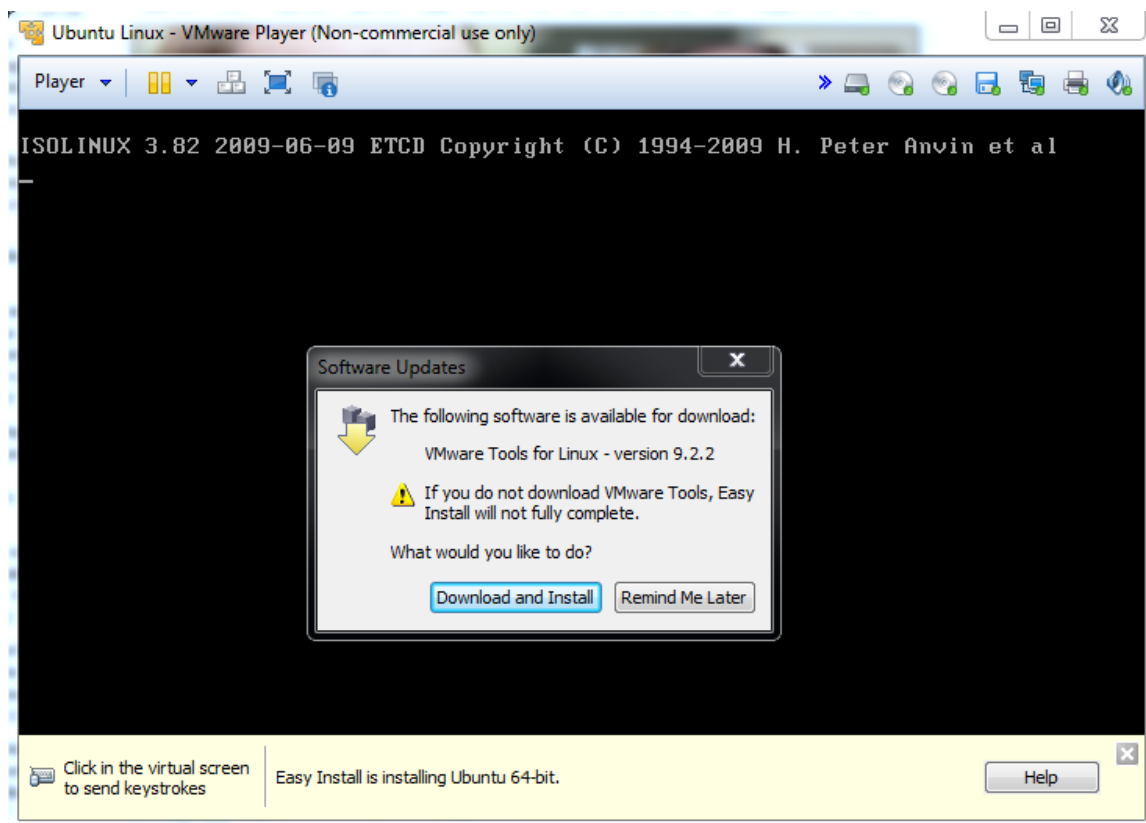


Kuva 2. VMware Playeriin valittiin Ubuntu Linuxin .iso levykuva.

Seuraavaksi annettiin asennettavaan käyttöjärjestelmään käytettäväksi oma nimi, käyttäjänimi sekä salasana. Tämän jälkeen annettiin virtuaalikoneelle kuvaava nimi, tässä tapauksessa ”Ubuntu Linux”, ja määritettiin tietokoneelle kansio, jota VMware Player käyttää virtuaalikoneen asetuksiin ja tiedon taltiointiin.

Tämän jälkeen määritettiin virtuaalikoneelle annettavan kovalevyosion koko. Tässä tapauksessa sille lohkaistiin 20 GB:n kokoinen osa.

Jotta virtuaalikone toimisi sulavasti, muutettiin ”Customize Hardware”-kohdasta virtuaalikoneelle käyttöön annettavaksi 2 prosessoriydintä sekä 2 GB keskusmuistia. ”Finish”-napin jälkeen VMware Player ehdottaa asennettavaksi VMware Tools for Linux -lisäpakettia kuvan 3 mukaisesti, johon vastattiin myöntävästi.

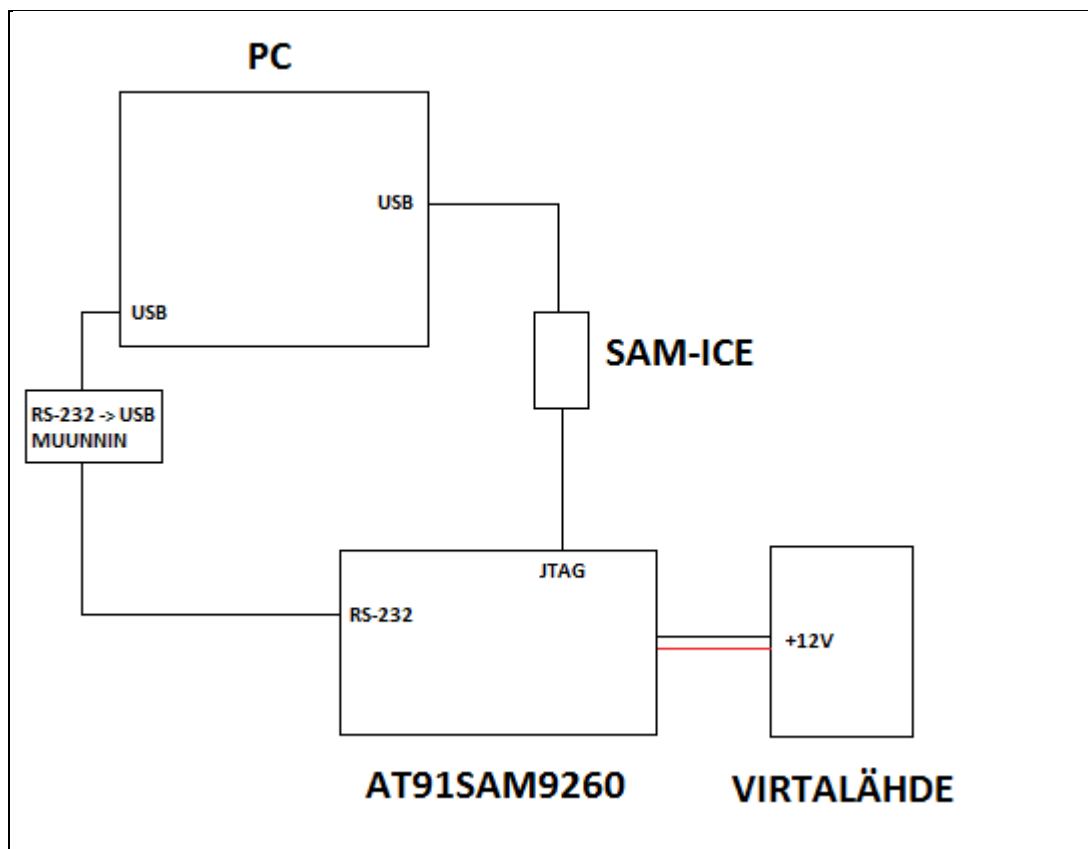


Kuva 3. VMware Tools for Linux

Tämän jälkeen Ubuntu jatkoi asentumistaan virtuaalikoneeksi.

3.5 LIIKE-kortin kytkentä

Kytkenässä käytettiin LIIKE-kortin lisäksi SAM-ICE debuggeria, tietokonetta, säädettävää virtalähdettä sekä tietokoneen puuttuvasta sarjaportista johtuen USB-sarjaporttimuunninta. Tietokoneen USB-porttiin kytkettiin SAM-ICE, joka puolestaan kytkettiin Atmelin kortin JTAG-liitäntään. Tätä kautta saatiin syötettyä tietokoneessa olleella SAM-BA-ohjelmalla kortin muistiin ohjelmat. Atmelin kortti kytkettiin kortin sarjaportin kautta takaisin tietokoneen USB-väylään, josta saatiin kortin käynnistyessä luettua dataa Tera Term -ohjelmalla. Kortin virtajohdot kytkettiin virtalähteeseen, joka säädettiin antamaan n. 12 voltin jännitettä. Kytkenä esitetty kuvassa 4.

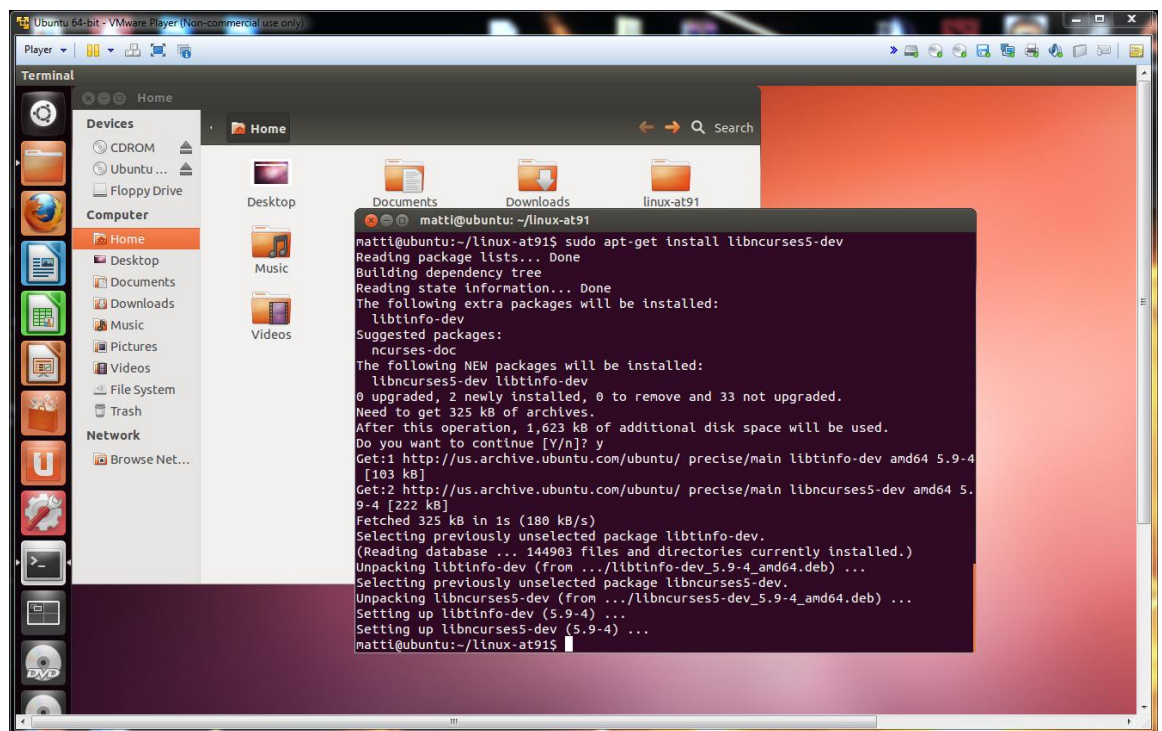


Kuva 4. AT91SAM9260-kortin kytkentä

4 YTIMEN KÄÄNTÄMINEN

Ytimen kääntäminen aloitettiin asentamalla Ncurses-niminen ohjelma, jota Ubuntussa ei ollut valmiina asennettuna. Tämän paketin avulla Linux-ytimen asetusten säätäminen saatiin vähän graafisempaan muotoon, jolloin niiden muuttaminen helpottui.

Ncurses asennettiin Linuxille avaamalla Terminal-ohjelma, joka vastaa Windows-käyttöjärjestelmissä komentokehotetta. Käskeyksi annettiin komentokehotteessa ”sudo apt-get install libncurses5-dev”, jolla asennus saatiin suoritettua. Asennus näkyy kuvassa 5.



Kuva 5. Ncurses asennus Linuxiin.

Seuraavaksi asennettiin versionhallintaohjelma Git, jolla päästiin hakemaan itse Linux-ydin ja muut tarvittavat osat koneelle. Versionhallintaohjelman asennus tapahtui komennolla ”sudo apt-get install git”.

Seuraava tarvittava paketti oli Linux-kääntäjään lisäosa, jolla saatiin toimimaan arm-ristiinkäännös. Tämä asennettiin komennolla ”sudo apt-get install gcc-arm-linux-gnueabi”.

Koska Linux-kernelin käynnistämiseen käytettiin työssä U-Bootia, piti se kääntää uImage-muotoon ja tähän tarvittiin oma työkalunsa nimeltä mkimage. Se asennettiin komennolla ”sudo apt-get install uboot-mkimage”.

Myös buildroot:lla rootfs:n kääntämistä varten piti asentaa muutama lisäpaketti. Ensin asennettiin parseri eli jäsenin, Bison. Asennus tapahtui komennolla ”sudo apt-get install bison”. Toisena asennettiin Flex komennolla ”sudo apt-get install flex”. Viimeinen tarvittava paketti oli Texinfo, joka asentui komennolla ”sudo apt-get texinfo”.

Kun tarvittavat paketit projektiin oli asennettu, voitiin siirtyä itse kernelin ja sen käynnistämiseen vaadittavien osien kääntämiseen.

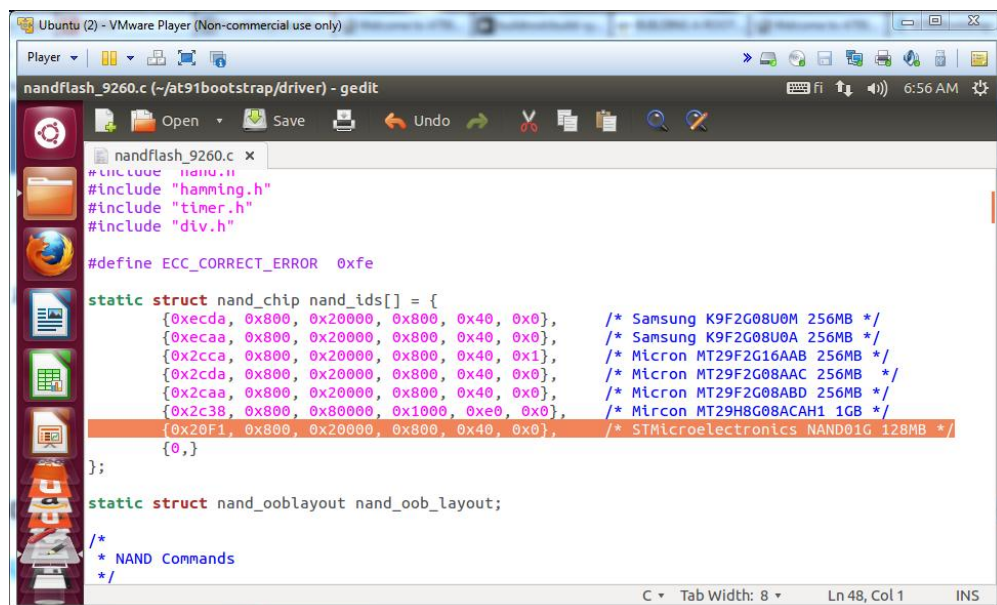
4.1 Bootstrap

LIIKE-kortin ensimmäinen toimintaan vaadittava osio oli Bootstrap. Sen tehtävä on tarjota valikoima erilaisia algoritmeja kortin laitteiston hallintaan, kuten kellotaajuuksien säätö, DRAM-muistin alustus ja seuraavan ohjelman hakeminen valitulta muistilta (tässä tapauksessa NANDFLASH-muistilta) kortin muistiin ja sen käynnistäminen.

Bootstrap haettiin ensin Git versionhallintaohjelmalla Linuxiin komennolla ”git clone git://github.com/linux4sam/at91bootstrap.git”. Käytettävä versio Bootstrapista oli 3.5.2.

Seuraavaksi Bootstrap-kansiossa annettiin komento ”make ARCH=arm at91sam9260eknf_uboot_defconfig”, jolla saatiin otettua käyttöön ennalta laaditut oletusasetukset at91sam9260-korttiin jossa käytetään NANDFLASH-muistia.

Koska työssä käytettävään korttiin oli vaihdettu NANDFLASH-muisti, piti Bootstrapille kertoa uuden muistin parametrit, jotta se toimisi. Tämä tapahtui muuttamalla at91bootstrap/driver/nandflash.c-tiedostoa. Sinne lisättiin kuvan 6 mukaisesti rivi ”{0x20F1, 0x800, 0x20000, 0x800, 0x40, 0x0}, ”, jolloin käytettävä muisti saatiin tunnistettua.



```

nandflash_9260.c (~at91bootstrap/driver) - gedit
#include "nandu.h"
#include "hamming.h"
#include "timer.h"
#include "div.h"

#define ECC_CORRECT_ERROR 0xfe

static struct nand_chip nand_ids[] = {
    {0xecda, 0x800, 0x20000, 0x800, 0x40, 0x0}, /* Samsung K9F2G08U0M 256MB */
    {0xecaa, 0x800, 0x20000, 0x800, 0x40, 0x0}, /* Samsung K9F2G08U0A 256MB */
    {0x2cca, 0x800, 0x20000, 0x800, 0x40, 0x1}, /* Micron MT29F2G16AAB 256MB */
    {0x2cda, 0x800, 0x20000, 0x800, 0x40, 0x0}, /* Micron MT29F2G08AAC 256MB */
    {0x2caa, 0x800, 0x20000, 0x800, 0x40, 0x0}, /* Micron MT29F2G08ABD 256MB */
    {0x2c38, 0x800, 0x80000, 0x1000, 0xe0, 0x0}, /* Micron MT29HG08ACAH1 1GB */
    {0x20F1, 0x800, 0x20000, 0x800, 0x40, 0x0}, /* STMicroelectronics NAND01G 128MB */
    {0,}
};

static struct nand_ooblayout nand_oob_layout;

/*
 * NAND Commands
 */

```

Kuva 6. Muutos nandflash.c tiedostoon

Itse Bootstrapiin ei tämän jälkeen tarvittu muutoksia, joten se voitiin kääntää kortille sopivaksi tiedostoksi. Tämä tapahtui bootstrap-kansiossa komennolla ”make CROSS_COMPILE=arm-linux-gnueabi-”, joka teki at91bootstrap/binaries/-kansioon työssä käytettävän ”at91sam9260ek-nandflashboot-uboot-3.5.2.bin” -tiedoston.

4.2 U-Boot

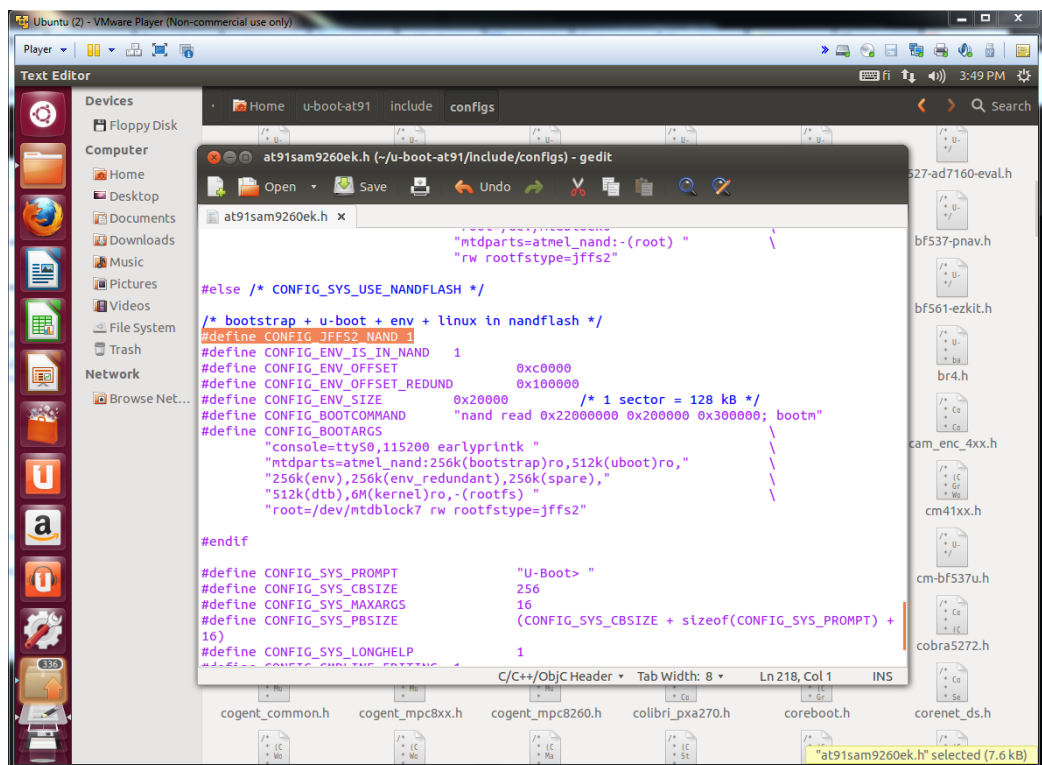
U-Boot on Bootstrapin käynnistämä osio, jonka tehtävänä on konfiguroida kortin päärajapinnat sekä käynnistää Linux-kernel. U-Bootissa on kohtalaisen paljon säätömahdollisuuksia, mutta tässä työssä tarvittiin säätöjä vain ethernetin poistamiseen käytöstä sekä käytettävän tiedostojärjestelmän esittely (JFFS2).

U-Boot haettiin Git-versionhallintaohjelmalla Linuxiin komennolla ”git clone git://github.com/linux4sam/u-boot-at91.git”. U-Bootin versio oli 2012.10.

Kuten Bootstrapissakin, U-Bootissa ladattiin ensin käytettäväksi oletusasetukset at91sam9260-kortille komennolla ”make at91sam9260ek_nandflash_config”. Tämän jälkeen U-Bootista otettiin pois ethernetin määrittelyt, koska kortissa ei sitä ollut ja niiden käyttö aiheutti ongelmallisia sivuvaikutuksia kuten virrankulutuksen hyppiminen. Ethernet saatiin pois käytöstä tekemällä muutoksia sekä U-Bootiin että kernelin asetuksiin, jotka käydään läpi luvussa 4.3. U-Bootissa muutettiin tiedostoa polussa /u-boot-at91/include/configs/at91sam9260ek.h. Ensimmäinen muutos oli laittaa kommentteihin ethernetin esittelyrivit (kuva 7), ja toisena lisättiin jffs2 tiedostomuodon tuki NANDFLASH-muistille lisäämällä tiedostoon rivi ”#define CONFIG_JFFS2_NAND 1” (kuva 8).



Kuva 7. Ethernetin poisto käytöstä U-Bootissa.



Kuva 8. Otetaan käyttöön JFFS2-tiedostojärjestelmä NANDFLASH-muistissa.

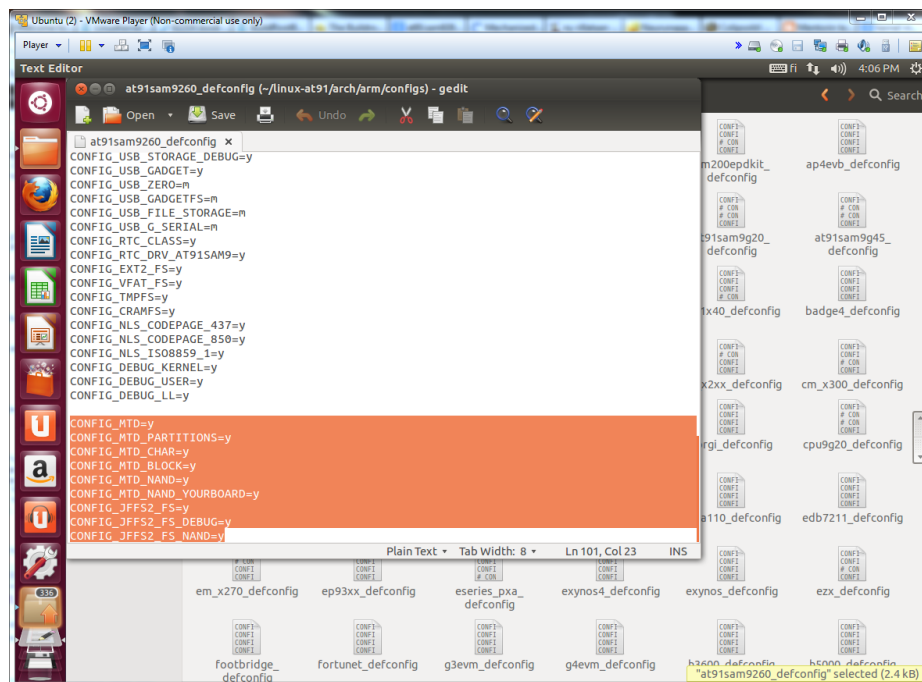
Muutosten jälkeen U-Boot käännettiin komennolla ”make CROSS_COMPILE=arm-linux-gnueabi-”. Valmis kortille syötettävä tiedosto löytyy tämän jälkeen polusta /u-boot-at91/u-boot.bin.

4.3 Linux-kernel

Linux-kernel oli koko projektin keskeisin osa, joka sisälsi käyttöjärjestelmän ytimen, erittäin suuren määrän mahdollisuuksia säätää ytimen toimintaa sekä tarvittavat ajurit kortin osien toimintaan. Työssä käytetty Linux-ytimen versio oli 3.6.9.

Linux-kernel haettiin Git-versionhallintaohjelmalla komennolla ”git clone git://github.com/linux4sam/linux-at91.git”.

Tämän jälkeen tehtiin kernelin oletusasetustiedostoon muutos, jolla saatiin otettua käyttöön jffs2 tiedostojärjestelmä ja muistin osiointiin tarvittavat ajurit. Tiedosto, jota muutettiin, sijaitsi polussa /linux-at91/arch/arm/configs/at91sam9260_defconfig. Tiedostoon tehtiin kuvan 9 mukaiset muutokset.



Kuva 9. Kernelin oletusasetusten muutos.

Muutosten jälkeen ladattiin kernelin oletusasetukset komennolla ”make ARCH=arm at91sam9260_defconfig”. Tämän jälkeen säädettiin loput kernelin asetukset kuntoon. Asetuksiin päästiin käsiksi komennolla ”make ARCH=arm menuconfig”. Kerneliin tehtävät muutokset listattuna:

- Kernel Features -> Use the ARM EABI to compile the kernel (rasti ruutuun)
- Device Drivers -> Memory Technology Device (MTD) support -> NAND Device Support -> Support for NAND FLASH / SmartMedia on AT91 and AVR32 (rasti ruutuun)
- Device Drivers -> Network device support -> Ethernet driver support (rasti pois)
- Device Drivers -> Network device support -> Wireless LAN (rasti pois)

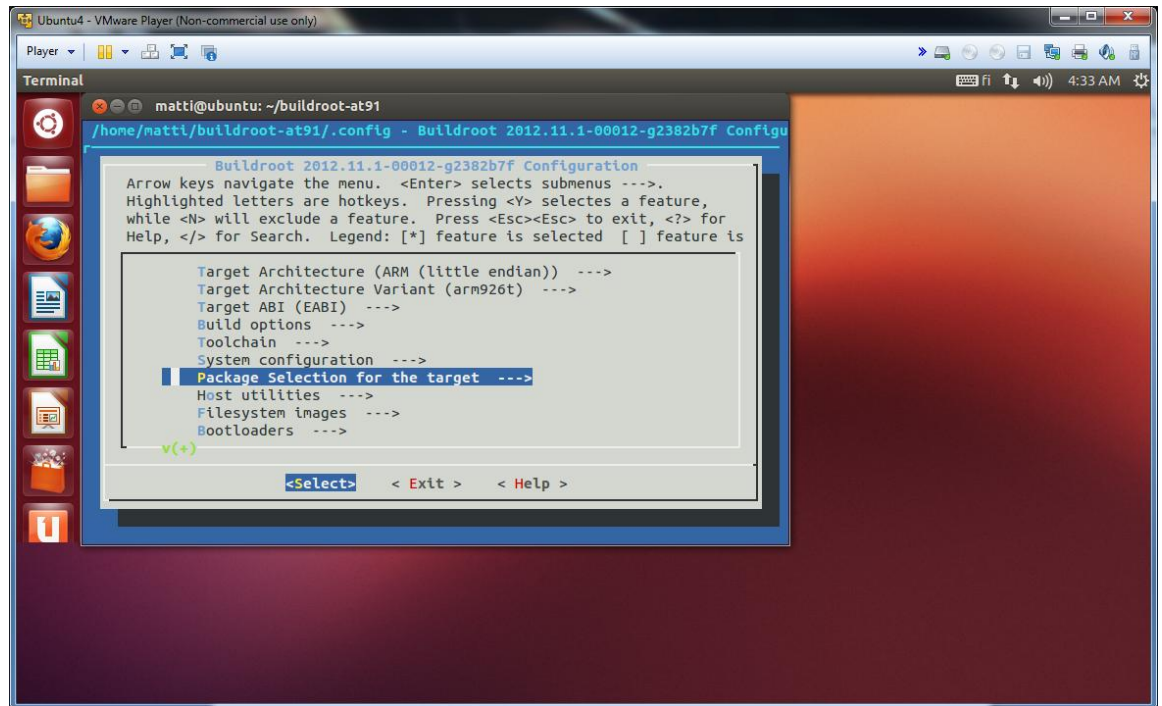
Tämän jälkeen tallennettiin muutokset ja käännettiin kernel kortilla käytettäväksi uImageksi komennolla ”make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- uImage”. Käännetty tiedosto löytyi polusta /linux-at91/arch/arm/boot/uImage.

4.4 Root filesystem

Root filesystem sisältää nimensä mukaisesti juuren tiedostojärjestelmän, johon sisältyy kansiot ja mahdolliset ohjelmat. Työssä käytettiin rootfs:n kääntämiseen Buildroot-nimistä ohjelmapakettia. Buildrootilla olisi ollut mahdollista tehdä myös muiden osien kääntäminen, mutta sitä ei hyödynnetty tässä tapauksessa.

Kuten aiemmissakin tapauksissa, kääntämisprosessi aloitettiin lataamalla Buildroot komennolla ”git clone git://github.com/linux4sam/buildroot-at91.git”. Tämän jälkeen asetettiin kortille sopivat oletusasetukset, mutta koska nandflashia käyttävälle kortille ei sellaisia ollut valmiina, jouduttiin käyttämään oletusasetuksia dataflash-asetuksia ja muuttamaan niitä myöhemmin. Oletusasetukset asetettiin komennolla ”make at91sam9260dfc_defconfig”.

Seuraavaksi päästiin säätämään asetuksia asetusvalikosta komennolla ”make menuconfig”. Asetusvalikon päävalikko näkyvässä kuvassa 10.



Kuva 10. Buildrootin asetusvalikko

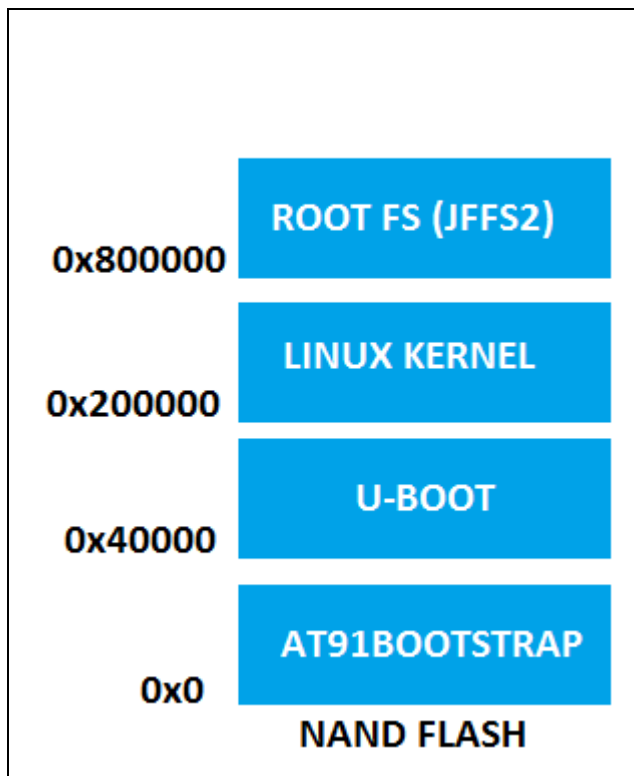
Tehdyt muutokset listattuna alla:

- Package Selection for the target -> Filesystem and flash utilities -> mtd/jffs2 utilities (rasti ruutuun)
- Filesystem images -> jffs2 root filesystem (rasti ruutuun)
- Filesystem images -> Flash Type (NAND flash with 2kB Page and 128kB erasesize) (rasti ruutuun)
- Bootloaders -> AT91Bootstrap (rasti pois)
- Bootloaders -> U-Boot (rasti pois)
- Kernel -> Linux Kernel (rasti pois)

Asetusten muuttamisen jälkeen tiedostojärjestelmä käännettiin komennolla ”make”. Käännetty tiedosto löytyi hakemistosta /buildroot-at91/output/images/rootfs.jffs2.

5 YTIMEN SIIRTO KORTILLE

Kun tarpeelliset tiedostot oli saatu käännettyä, voitiin ne siirtää LIIKE-kortille. Siirtoa varten otettiin selvää, mistä muistipaikoista Bootstrap, U-Boot sekä Kernel hakivat seuraavaksi käynnistettävää osaa. Nämä muistipaikat oli koodattu suoraan kyseisten osien lähdekoodiin. Muistikartta tarvittavien muistipaikkojen osalta esitetty kuvassa 11.

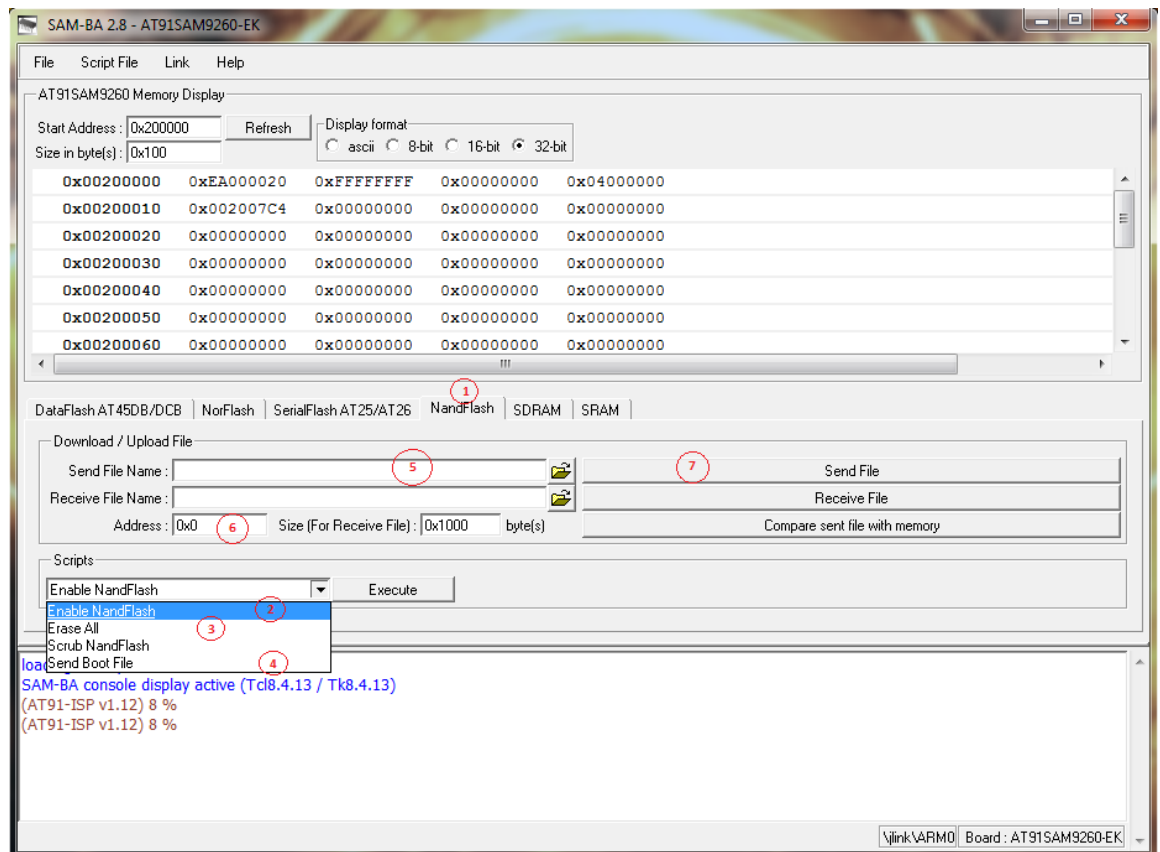


Kuva 11. Muistikartta

Kun tarvittavat muistiosoitteet olivat selvillä, voitiin aloittaa tiedostojen siirtäminen kortin muistiin SAM-BA-ohjelmalla.

Kortin JTAG-väylään kytkettiin ensimmäiseksi PC:n USB-porttiin yhteydessä oleva SAM-ICE, ja kortin virtajohdot yhdistettiin virtalähteeseen, joka oli säädetty 12 volttiin. Tämän jälkeen käynnistettiin SAM-BA.

Ensimmäiseksi SAM-BA:n käynnistyessä valittiin yhteysmuoto, jolla korttiin haluttiin olla yhteydessä. Tässä tapauksessa siis pudotusvalikosta valittiin kohta ”\jlink\ARM0”, sekä alemmasta valikosta valittiin käytettäväksi kortiksi ”AT91SAM9260-EK”. Tämän jälkeen painettiin ”Connect”-nappia, jolloin SAM-BA otti yhteyden korttiin ja eteen avautui graafinen käyttöliittymä. Tiedostojen siirto tapahtui numeroidussa järjestyksessä, joka on esitetty kuvassa 12.



Kuva 12. SAM-BA:n käyttöliittymä

Tiedostojen siirto aloitettiin tarkistamalla, että kuvassa näkyvä välilehti ”NandFlash” (1) oli valittuna. Tämän jälkeen kortin NandFlash-muisti otettiin käyttöön ajamalla alasvetovalikosta löytyvä ”Enable NandFlash”-skripti (2). Seuraavaksi kortin muisti tyhjennettiin ”Erase All”-skriptillä, jotta sinne ei vahingossa ollut jäänyt mitään tiedostonrippeitä (3).

Ensimmäinen kortille siirrettävä tiedosto oli Bootstrap. Tämä saatiin siirrettyä valitsemalla ”Send Boot File”-skripti (4), ja painamalla ”Execute”-nappia. Aiemmin käännetty Bootstrabin sisältävä tiedosto at91sam9260ek-nandflashboot-uboot-3.5.2.bin etsittiin tallennuskansiosta ja lähetettiin avautuneen ikkunan ”Open”-napilla.

Toisena lähetettävistä tiedostoista oli U-Boot. U-boot.bin-tiedosto haettiin ”Send File Name”-kohtaan (5) ja ”Address”-kenttään (6) annettiin osoitteeksi 0x40000, jonka jälkeen tiedosto lähetettiin ”Send File”-napilla (7) kortille.

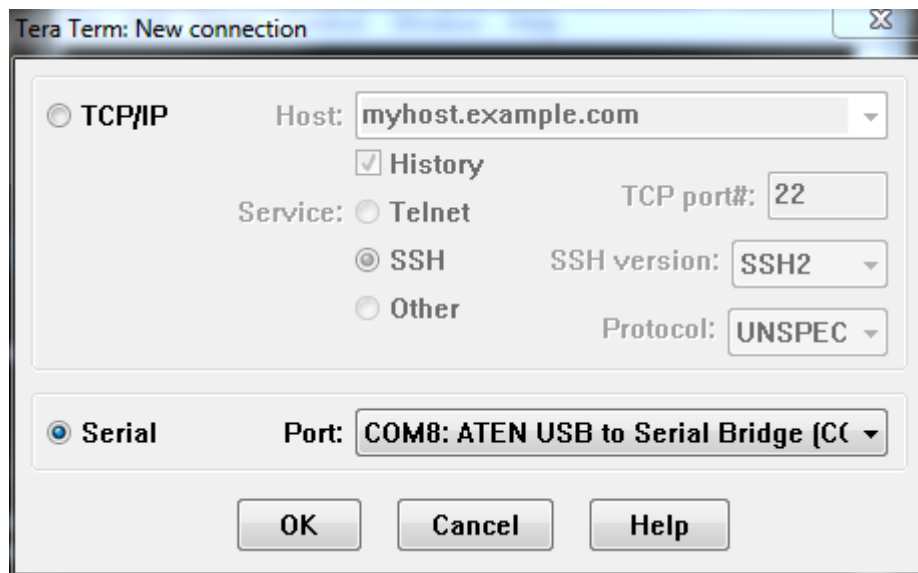
Kolmantena oli vuorossa Linux Kernel. U-Bootin tapaan aiemmin tehty uImage-tiedosto haettiin ”Send File Name”-kohtaan, ”Address”-kohtaan annettiin osoitteeksi 0x200000 ja kernel lähetettiin kortille ”Send File”-napilla.

Viimeisenä kortille siirrettiin tiedostojärjestelmän sisältävä rootfs.jffs2-tiedosto. Tuttuun tapaan se haettiin ensin ”Send File Name”-kohtaan. Osoitteeksi valittiin ”Address”-kenttään 0x800000, ja tiedostojärjestelmä lähetettiin ”Send File”-napilla.

Nyt Linux-ydin tarvittavine käynnistystiedostoineen ja tiedostojärjestelmineen oli siirretty kortin muistiin ja SAM-BA-ohjelma voitiin sammuttaa sekä SAM-ICE ottaa irti kortin JTAG-liitännästä. Kun korttiin kytkettiin käyttöjännitteet, Linux-käyttöjärjestelmä lähti käynnistymään ja sen toimintaa päästiin testaamaan.

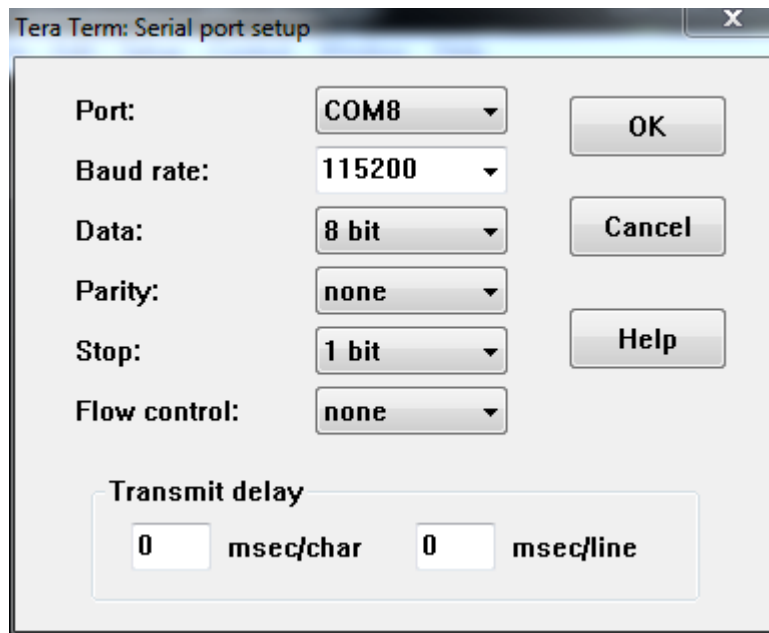
6 TULOKSET JA TESTAAMINEN

Kortin toiminnan testaaminen aloitettiin Bootstrapin, U-Bootin, kernelin sekä rootfs:n siirtämisen jälkeen laittamalla sarjakaapelin toinen pää kortin RS-232-liittimeen ja toinen pää RS-232 – USB-muuntimen kautta PC:n USB-väylään kiinni. Tämän jälkeen käynnistettiin tietokoneella oleva Tera Term -ohjelma ja valittiin sarjayhteys valitsemalla ”Serial”-kohta. Portiksi valittiin COM8, johon kortilta tuleva sarjakaapeli oli USB-muuntimen kautta kytketty (kuva 13).



Kuva 13. Sarjaportin valinta Tera Term -ohjelmassa

Tämän jälkeen muutokset hyväksyttiin ”OK”-napilla. Jäljelle jääneestä Tera Term -ikkunasta valittiin ”Setup”-valikko ja sieltä ”Serial port...”-kohta. Jotta tiedonsiirto kortin ja tietokoneen välillä toimisi, piti ”Baud rate:”-kohtaan valita 115200 (kuva 14) koska korttiin siirrettyjen tiedostojen asetuksissa oli määritelty kyseinen nopeus.



Kuva 14. Tera Term -ohjelman asetukset

Seuraavaksi kortin virtalähteelle menevää +-kaapelia käytettiin irti ja laitettiin takaisin, jolloin kortti lähti käynnistämään muistissa olevaa Bootstrapia. Bootstrap lähti hyvin käyntiin ja käynnisti seuraavaksi järjestyksessä U-Bootin, kernelin sekä rootfs:n. Kirjautumistiedoiksi Tera Termiin syötettiin käyttäjänimeksi ”root”, ja koska salasanan kyselyä asetettu Buildrootin asetuksista, sitä ei tarvinnut syöttää. Kuvassa 15 on esitettyinä kernelin käynnistymisen jälkeen kortilla oleva toimiva Linux ja sen hakemistot.

```

COM8:115200baud - Tera Term VT
File Edit Setup Control Window Help
NET: Registered protocol family 2
TCP established hash table entries: 2048 (order: 2, 16384 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
TCP: reno registered
UDP hash table entries: 256 (order: 0, 4096 bytes)
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
NetHinder Floating Point Emulator V0.97 (double precision)
jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
msgmni has been set to 120
io scheduler noop registered (default)
at91sam9260-usart.0: ttyS0 at MMIO 0xfffff200 (irq = 17) is a ATMEL_SERIAL
console [ttyS0] enabled
at91sam9260-usart.1: ttyS1 at MMIO 0xffffb000 (irq = 22) is a ATMEL_SERIAL
at91sam9260-usart.2: ttyS2 at MMIO 0xffff4000 (irq = 23) is a ATMEL_SERIAL
brd: module loaded
atmel_nand: Use On Flash BBT
atmel_nand atmel_nand: No DMA support for NAND access.
ONFI param page 0 valid
ONFI flash detected
NAND device: Manufacturer ID: 0x20, Chip ID: 0xf1 (ST Micro NAND016H3B2CN6), page size: 2048, OOB size: 64
Bad block table found at page 65472, version 0x01
Bad block table found at page 65408, version 0x01
8 cmdlinepart partitions found on MTD device atmel_nand
Creating 8 MTD partitions on "atmel_nand":
0x000000000000-0x000000040000 : "bootstrap"
0x000000040000-0x0000000c0000 : "uboot"
0x0000000c0000-0x000000100000 : "env"
0x000000100000-0x000000140000 : "env_redundant"
0x000000140000-0x000000180000 : "spare"
0x000000180000-0x000000200000 : "dtb"
0x000000200000-0x000000800000 : "kernel"
0x000000800000-0x000000800000 : "rootfs"
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 1
at91_ohci at91_ohci: irq 36, io mem 0x00500000
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mousedev: PS/2 mouse device common for all mice
rtc-at91sam9 rtc-at91sam9.0: rtc core: registered rtc-at91sam9 as rtc0
rtc-at91sam9 rtc-at91sam9.0: rtc0: SET TIME!
i2c /dev entries driver
at91sam9_udt: sorry, watchdog is disabled
at91sam9_udt: the watchdog has been disabled
TCP: cubic registered
NET: Registered protocol family 17
rtc-at91sam9 rtc-at91sam9.0: hctosys: unable to read the hardware clock
jffs2: Empty flash at 0x000fbb00 ends at 0x000fc000
VFS: Mounted root (jffs2 filesystem) on device 31:7.
Freeing init memory: 128K
Starting logging: OK
Initializing random number generator... done.
Starting network...

Welcome to Buildroot
buildroot login: root
# cd ..
# ls
bin      etc      lib      media   opt      root     sbin     tmp     var
dev      home    linuxrc  mnt     proc     run      sys     usr
#

```

Kuva 15. Tera Term -ohjelman ikkuna kortin käynnistyttyä

7 YHTEENVETO

Työn tarkoituksen oli muokata ja asentaa toimiva Linux-kernel käynnistykseen vaadittavine tiedostoineen LIIKE-kortille. Kernelin oli käynnistytävä automaattisesti, kun korttiin kytkettiin käyttöjännite. Työ tehtiin Kajaanin ammattikorkeakoululle.

Työ oli erittäin mielenkiintoinen ja haastava, koska kokemukseni sulautettujen järjestelmien käytännön sovelluksista oli aika suppea sekä tuntemukseni Linuxista olematon. Työhön oli tarkoitus lisätä vielä automaattisesti kernelin käynnistyttyä avautuva Hello World -sovellus, mutta aikapulan takia en ehtinyt saada sitä asennettua suoraan kortin tiedostojärjestelmään.

Jotta kernelin käynnistämiseen vaaditut Bootstrap ja U-Boot saatiin toimimaan, vaati se syvällistä lähdekoodien tutkimista, joka oli koodin runsaudesta johtuen kohtalaisen aikaa vievää. Myös kernelin ja rootfs:n tutkimiseen ja asetusten säätöön meni pitkään, ennen kuin sain ne säädettyä kortille sopiviksi ja toimiviksi.

Lopputuloksena oli kuitenkin toimiva kokonaisuus, jossa Bootstrap, U-Boot, kernel sekä rootfs käynnistivät järjestyksessä toisensa ja käyttäjälle jäi valmis paketti, johon voidaan asentaa kortin käyttöä ajatellen hyötyohjelmia.

LÄHTEET

- 1 Kajaanin ammattikorkeakoulun toiminta. Viitattu 24.2.2013. [WWW-sivusto]
<http://www.kamk.fi/fi/Esittely/Toiminta>
- 2 Nikkanen, T.Linuxin tarina. Jyväskylä: Gummerus Kirjapaino Oy, 2000. 256 s.
ISBN 951-762-990-7
- 3 Download AT91-ISP_v1.12. Viitattu 30.4.2013 [WWW-sivusto]
http://www.atmel.com/Images/Install_AT91-ISP_v1.12.exe
- 4 Download Tera Term v4.77. Viitattu 1.5.2013 [WWW-sivusto]
<http://sourceforge.jp/projects/ttssh2/downloads/58215/teraterm-4.77.exe>
- 5 Download VMWare Player 5.0. Viitattu 6.3.2013 [WWW-sivusto]
https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/5_0
- 6 Lataa Ubuntu 12.10. Viitattu 2.5.2013 [WWW-sivusto]
<http://linux.softpedia.com/dyn-postdownload.php?p=87208&t=0&i=1>