



Henri Antero Seppälä

LINUX-OHJELMISTOKEHITYS AVR32- YMPÄRISTÖSSÄ

Tekniikka ja liikenne

2009

VAASAN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Henri Seppälä
Opinnäytetyön nimi	Linux-ohjelmistokehitys AVR32-ympäristössä
Vuosi	2009
Kieli	suomi
Sivumäärä	41 + 1 liite
Ohjaaja	Martti Mustonen

Tämän opinnäytetyön tarkoituksena on tutustua sulautetun elektroniikan ohjelmistokehitykseen Linuxilla, kohteena AVR32-ympäristö. Työssä käytetään Atmelin valmistamaa NGW100 Network Gateway kit-evaluointikorttia, joka on tehokas ja monipuolisin liitännöin varustettu kehitysympäristö sulautetun Linuxin ohjelmistokehitykseen.

Ohjelmointiin käytetään työpöytä tietokonetta, johon on asennettu Ubuntu Hardy Heron-käyttöjärjestelmä. Työn tuloksena tuotetaan NGW:llä ajettava ohjelma, joka käyttää I2C-väylää suuntatiedon lukemiseen Devantech CMPS03-elektroniselta kompassilta.

Ohjelman lisäksi työssä kirjoitetaan Vaasan ammattikorkeakoulun käyttöön ohjeistus I2C-väylän käyttämiseen NGW:llä ja esitellään jonkin verran hyväksi havaittuja tietolähteitä sekä Linuxin käyttöön yleisesti että sulautetussa elektroniikassa. Lähteet ovat suurelta osin www-sivuja, mutta myös painettuun kirjallisuuteen tutustutaan.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tietotekniikan koulutusohjelma

ABSTRACT

Author	Henri Seppälä
Title	Linux Software Development in AVR32 Environment
Year	2009
Language	Finnish
Pages	41 + 1 appendices
Name of Supervisor	Martti Mustonen

The purpose of this thesis is to become familiar with developing software for embedded electronics AVR32-environment using Linux operating system. Device that will be used is an Atmel NGW100 Network Gateway kit-evaluation card, which is a powerful development kit equipped with versatile connections for embedded Linux applications.

Programming is done with a desktop computer equipped with Ubuntu Hardy Heron-operating system. Results are a finished application for using I2C-bus with NGW, and a comprehensive guide for students of Vaasa University of Applied Sciences to study for better understanding of embedded Linux with NGW and I2C-bus.

An important aspect in this thesis is also information retrieval, which will be handled by presenting some of the most useful www-pages and written books about Linux in general, and also in embedded electronics field.

SISÄLLYS

1 JOHDANTO	5
2 KÄYTETYT LAITTEET JA NIIDEN OMINAISUUDET	7
2.1 Atmel NGW100	7
2.2 Elektroninen kompassi CMPS03	12
3 TIEDONHAKU	14
3.1 Sulautettu Linux-ohjelmistokehitys	14
3.2 AVR32 ja Buildroot	18
3.3 I2C-väylä	19
4 TYÖN ETENEMINEN	21
4.1 Tutustuminen aiheeseen	21
4.2 AVR32 Linux ohjelmointityökalujen asennus	23
4.3 Buildroot-käyttö	25
4.4 Uuden käyttöjärjestelmän siirtäminen NGW:lle	28
4.5 I2C-käyttö Linuxissa ja NGW:llä	32
5 YHTEENVETO	38

LÄHTEET

LIITE 1: WikiVAMKiin kirjoitettu ohjeistus

1 JOHDANTO

Linux on käyttöjärjestelmänä vahvasti esillä tämän päivän sulautetussa elektroniikassa. Esimerkeiksi Linuxin käyttökohteista käyvät Nokian nettitabletti N810, Googlen Linux-pohjaiseen Android-käyttöjärjestelmään perustuva HTC G1-matkapuhelin, TomTom-autonavigaattorit sekä monet tietoliikenneverkon aktiivilaitteet aina ADSL-modeemeista palvelimiin. Linuxin etuja sulautetussa elektroniikassa ovat muun muassa lisenssimaksuttomuus, monipuolinen tuki erilaisille prosessoreille ja oheislaitteille, skaalautuvuus sekä saatavilla olevien, usein ilmaisten, sovellusten valikoima./1/

Tämän opinnäytetyön tarkoituksena on tutustua sulautetun elektroniikan ohjelmistokehitykseen Linux-käyttöjärjestelmällä Atmelin AVR32-ympäristössä. Käytössä on työpöytä tietokone, johon on asennettu Ubuntu Hardy Heron-käyttöjärjestelmä, sekä Atmelin valmistama NGW100 Network Gateway Kit-evaluointikortti. Kyseessä on lähinnä verkkosovellusten kehittämiseen tarkoitettu laite, mutta monipuolisten liitäntöjen ansiosta se toimii mainiosti myös yleiskäyttöisenä ohjelmointialustana. Työn tuloksena tuotetaan dokumentaatio tarvittavista ohjelmointityökaluista, tietolähteistä ja valmiista ohjelmasta Philipsin I2C-väylän käyttämiseksi Linux-käyttöjärjestelmään pohjautuvassa sulautetussa laitteessa. I2C-väylällä ohjataan ja luetaan Devantech CMPS03-elektronista kompassia. Lisäksi työn tuloksena kirjoitetaan ohjeistus I2C-väylän käytöstä Vaasan ammattikorkeakoulun käytettäväksi.

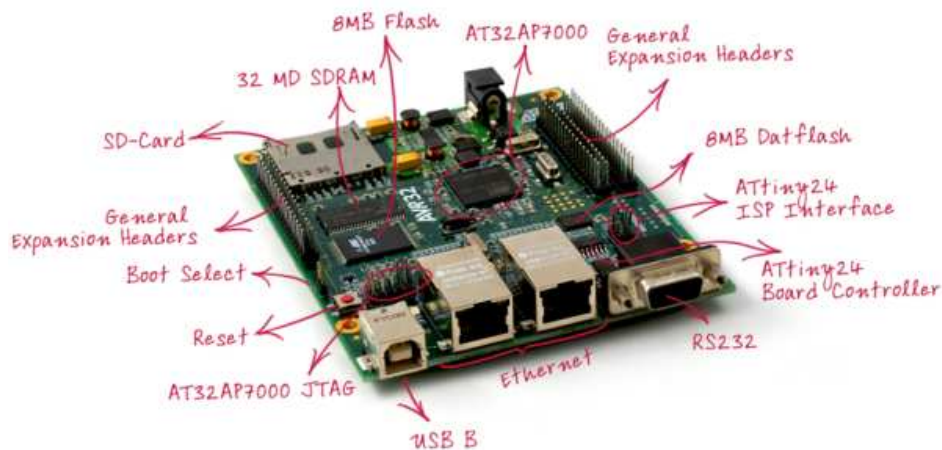
Työ aloitetaan lähes täysin ilman aiempaa kokemusta Linuxista edes työpöytäkäytössä, joten tiedonhaulla aiheesta on suuri merkitys työn onnistumiselle. Tiedonhauille onkin omistettu kokonainen luku tekstistä. Linux-käyttöjärjestelmästä löytyy tietoa internetistä ja painetusta kirjallisuudesta lähes rajattomasti, mutta tarkan ja ajantasaisen tiedon löytäminen on hankalampaa. Työssä tullaan esittelemään joitain tietolähteitä ja pohtimaan hieman niiden luotettavuutta ja käyttömahdollisuuksia.

Viimeisessä kappaleessa käydään aloittelijaystävällisesti läpi tarvittavat toimenpiteet toimivan ohjelman kehittämiseksi AVR32-ympäristölle. Tärkeimpiä aiheita ovat ohjelmointityökalujen asennus isäntäkoneelle, uuden päivitetyn käyttöjärjestelmän luominen ja siirto NGW:lle sekä tarvittavat toimenpiteet I2C-väylän käyttämiseksi omasta ohjelmasta. Lisäksi käsitellään aloittelijan näkökulmasta työssä tarvittavia Linux-komentoja ja Linux-käyttöjärjestelmän ominaisuuksia. Oman ohjelman koodi ja sen toiminta käydään läpi kohta kohdalta. Tarkoituksena on, että viimeisen kappaleen ohjeilla opiskelija pystyy aloittamaan sulautetun elektroniikan ohjelmistokehityksen Linuxilla.

2 KÄYTETYT LAITTEET JA NIIDEN OMINAISUUDET

2.1 Atmel ATNGW100 Network Gateway kit

NGW100 Network Gateway Kit (myöhemmin NGW) on Atmelin valmistama evaluointikortti AVR32-pohjaiselle AT32AP7000-mikro-ohjaimelle. Se on suunniteltu lähinnä Linux-pohjaisten verkkosovellusten kehittämiseen, mutta kuvan 1 mukaisin monipuolisin liitännöin varustettuna se toimii myös hyvänä yleiskehitysympäristönä AT32AP7000-prosessorille. NGW:n hinta on Suomessa tällä hetkellä n. 100 €, joten se sopii myös edullisuutensa ansiosta hyvin sulautetun elektroniikan ohjelmistokehitykseen pienelläkin budjetilla [2]. Atmelin tarjoama kalliimpi vaihtoehto on STK1000-evaluointikortti, joka on varustettu samalla prosessorilla, mutta lisäksi muun muassa QVGA-resoluutioisella LCD-näytöllä. Hintaluokka STK1000-kortilla on noin 600 € [2],[3].



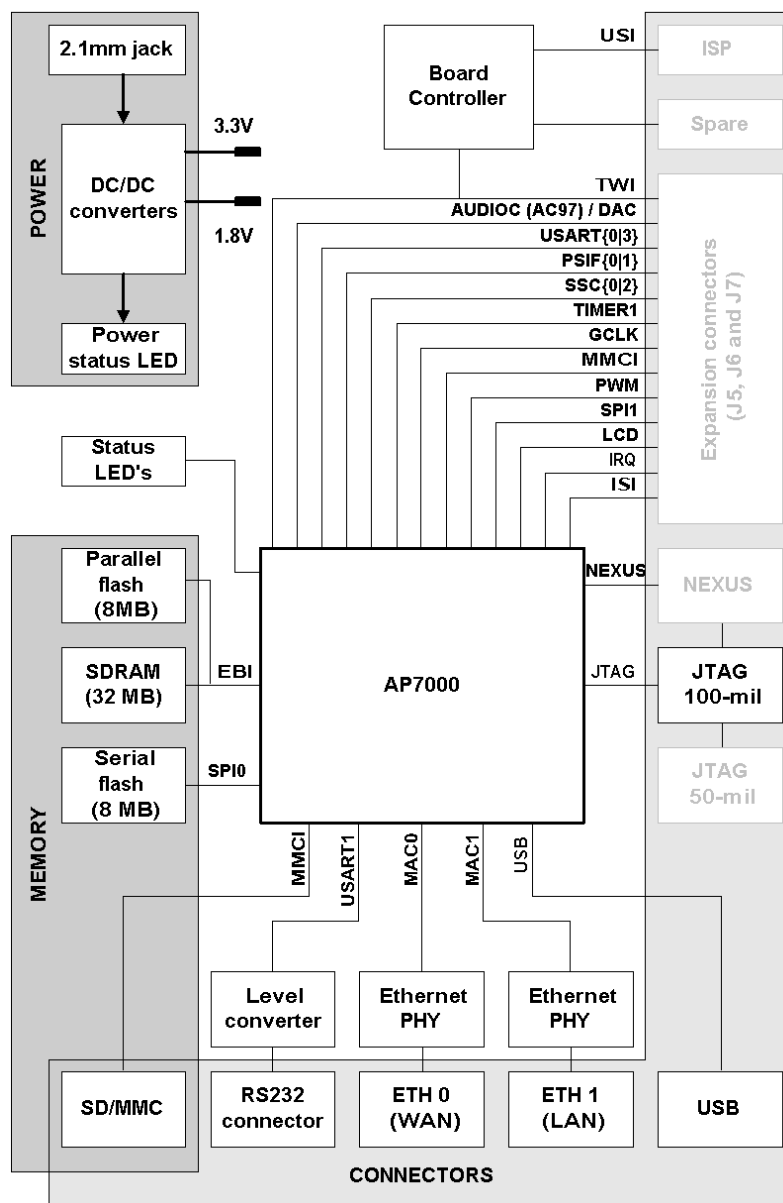
Kuva 1. Atmel NGW100-kortti, sen liitännät ja tärkeimmät osat [5]

AT32AP7000 on 32-bittinen, RISC-arkkitehtuuriin perustuva DSP-käskykannalla varustettu mikro-ohjain. Sen ilmoitettu huippusuorituskyky on 210 DMIPS (Dhrystone MIPS) 150 MHz:n taajuudella. Prosessorin suunnittelussa on painotettu erityisesti matalaa tehonkulutusta sekä korkeaa kooditiheyttä ja sovellustehokkuutta. Muistinhallintayksikön ja joustavasti ohjelmoitavien keskeytysohjausten ansiosta

prosessori sopii käytettäväksi modernien ja myös reaaliaikaisten käyttöjärjestelmien kanssa. Lisäksi DSP-käskykanta tehostaa prosessorin käyttöä multimedia- ja tietoliikennesovelluksissa./4/

Prosessoriin on sisäänrakennettu 16 kt käsky- ja datavälimuistit, 32 kt SRAM-datumuisti sekä enemmän muistia vaativille sovelluksille ulkoinen 16-bittinen SRAM-väylä. DMA-ohjain (Direct Memory Access) mahdollistaa tiedonsiirron eri muistien ja oheislaitteiden välillä ilman prosessorin osallistumista. 256-jalkaiseen CTBGA-koteloon on pakattu paljon ominaisuuksia: rajapinnat ulkoisille laajennusmuisteille (SDRAM, DataFlash, SRAM, MMC, SD, Compact Flash, Smart Media, NAND Flash), ohjelmoitava LCD-näytönohjain, joka mahdollistaa resoluutioiden käytön aina 2048x2048 asti, CMOS-rajapinta tukemaan enintään 12-bittisellä dataväylällä varustettuja kameroita, USB 2.0, verkkoyhteyksiin 2 kpl Ethernet MAC-rajapintoja, 3 kpl synkronoitua sarjaprotokolla-ohjainta, 4 kpl USART-ohjaimia, I2C-yhteensopiva TWI, sekä 16-bittinen stereo audio DAC. /6/

NGW-evaluointikortti on varustettu prosessorin mahdollistamilla monipuolisilla liitäntänoilla, joihin kuuluu kaksi ethernet-liitintä (LAN ja WAN), RS232-sarjaportti, USB-liitin, JTAG-liitin kortilla olevan FLASH-muistin ohjelmointia tai ohjelmien vianetsintää varten sekä useita yleiskäyttöön tarkoitettuja laajennusnastoja (kuva 2).



Kuva 2. NGW lohkokkaavio /5/

Käyttömuistina kortilla on 32 Mt SDRAM-muisti, joka on yhdistetty prosessoriin 16-bittisellä dataväylällä. 16 Mt FLASH-muisti on jaettu kahteen osaan, 8 Mt sarjamuotoiseen FLASHiin, sekä samankokoiseen rinnakkaismuistiin. Jälkimmäiseen on nopeampana vaihtoehtona tehtaalla esiasennettu U-Boot-käynnistyslatain ja sulautettu Linux-ydin hitaampaan sarjamuistiin Linux-tiedostojärjestelmä. Lisäksi kortille on sijoitettu SD- tai MMC-muistikortinlukulaite, jonka avulla kortilla olevaa FLASH-muistia on mahdollista laajentaa. Muita tärkeitä komponentteja kortilla ovat

virtaliitettä, joka hyväksyy 9-15 V tasajännitteen, merkkivalot jännitteensyötölle ja kortin statukselle sekä reset-kytkin. Virtakytkintä kortilla ei ole. /5/

NGW-kortin FLASH-muistiin on tehtaalla esiasennettu U-Boot-käynnistyslatain ja Linux-käyttöjärjestelmä, sekä työkalupaketti nimeltä Busybox. Kortti on ohjelmiston osalta valmis käyttöön suoraan laatikosta otettuna.

Käynnistyslatain on pieni ohjelma, jonka ensisijainen tehtävä on aloittaa käyttöjärjestelmän lataus laitteiston käynnistyksen jälkeen. Käynnistyslataimen asetuksissa on tieto käyttöjärjestelmän tai käyttöjärjestelmien tiedostojen sijainnista, sekä tieto siitä mitä oheislaitteita käynnistämiseen tarvitaan. Käynnistyslataimen muunneltavuus mahdollistaa käyttöjärjestelmän sijoittamisen erityyppisiin muisteihin, kuten NGW:llä joko kortilla olevaan FLASH-muistiin, muistikortille tai jopa verkkoasemalle. Erityisesti pöytätietokoneissa käynnistyslatainta voidaan hyödyntää usean käyttöjärjestelmän laitteissa, jolloin käyttäjä voi konetta käynnistäessään valita mitä järjestelmää käytetään. /7/

U-Boot on yleiskäyttöinen GPL-lisenssin alainen käynnistyslatain sulautetun elektroniikan järjestelmiin. Ensimmäinen U-Boot-versio julkaistiin jo vuonna 2002 ja nykyään se on useiden valmistajien evaluointikorttien mukana toimitettava oletuskäynnistyslatain. U-Bootin suunnittelun lähtökohtina ovat olleet helppo muunneltavuus uusille ohjelmistoympäristöille ja prosessoreille, helppo vianetsintä, mahdollisimman pieni koko ja luotettavuus sekä kaikkien ominaisuuksien muunneltavuus tarpeen mukaan. Käyttöliittymänä toimii kuvan 3 mukainen, yksinkertainen tekstipohjainen komentokehote, jota yleensä käytetään sarjaportin kautta. /8/

```
U-Boot 1.1.4-at0 (Jan 3 2007 - 10:30:09)

U-Boot code: 00000000 -> 000144f7 data: 24000000 -> 24002d80
SDRAM: 32 MB at address 0x10000000
Testing SDRAM...OK
malloc: Using memory from 0x11fc0000 to 0x12000000
Flash: 8 MB at address 0x00000000
DRAM Configuration:
Bank #0: 10000000 32 MB
In: serial
Out: serial
Err: serial
Net: macb0, macb1
Press SPACE to abort autoboot in 1 seconds
Uboot> _
```

Kuva 3. U-Boot-käyttöliittymä

Käyttöjärjestelmän ja lähinnä sen ytimen tärkein tehtävä on tarjota järjestelmän eri osien ja oheislaitteiden palvelut sovellusohjelmien käyttöön sekä huolehtia käyttöoikeuksien ja resurssien jaosta eri prosessien ja käyttäjien kesken. Käyttöjärjestelmän ansiosta jokaiseen ohjelmaan ei tarvitse kirjoittaa koodia, esimerkiksi sarjaportin käyttämiseen vaan käyttöjärjestelmä tarjoaa sarjaportin palvelut ohjelmille. Tämä helpottaa huomattavasti muun muassa sulautetun elektroniikan ohjelmointia ja mahdollistaa koodin siirtämisen järjestelmästä toiseen. /9/

NGW-kortille on asennettu Linux-käyttöjärjestelmä sekä Busybox-työkalupaketti. Yhdessä nämä vievät tilaa kortilta alle 7 Mt, josta työkalupaketin osuus on noin 1 Mt. Suurin ero kortille asennetun Linuxin ja perinteiseen työpöytäkäyttöjärjestelmän välillä on graafisen käyttöliittymän puuttuminen, koska näyttöäkään ei ole. Tiedostojärjestelmässä navigoidaan ja käytetään työkaluja tekstipohjaisella komentokehoteella, jota syötetään ja luetaan isäntäkoneella sarjaporttiliitännän kautta (kuva 4).

```

* iptables postrouting ... [ OK ]
* iptables incoming traffic ... [ OK ]
* iptables outgoing traffic ... [ OK ]
* starting dnsmasq ... [ OK ]
* running ntpdate ... [ FAILED ]
* starting dropbear ... [ OK ]
* starting inited ... [ OK ]
* starting nmbd ... [ OK ]
* starting smbd ... [ OK ]
* starting winbindd ... [ OK ]
* starting httpd ... [ OK ]
* get board type for GPIO ---- 'NGH'
* setup GPIO boot LED ... [ OK ]
* setup GPIO LED A ... [ OK ]
* setup GPIO LED B ... [ OK ]

Network Gateway ready

BusyBox v1.4.2 (2007-04-17 15:34:55 CEST) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

#

```

Kuva 4. Käyttöliittymänäkymä ensimmäisen käynnistyksen jälkeen

NGW-kortilla olevan Linux-ytimen versio on 2.6.18-atngw, eli se on Atmelin tähän käyttöön konfiguroima erikoisversio. Linux-ytimen lähdekoodi on käännetty AVR32-ympäristölle ja siihen on lisätty sopivat laiteajurit, jotka mahdollistavat sen toimimisen AT32AP7000-prosessorilla sekä prosessorin eri ominaisuuksien ja oheislaitteiden palvelujen käytön suoraan käyttäjän omista ohjelmista. Käytännössä mikä tahansa tehtaalla asennettu ydin on tuotteen saapuessa loppukäyttäjälle jo vanhentunut, joten se kannattaa päivittää ensimmäisenä. Atmel tarjoaa helpon tavan uuden, viimeisimmän ytimen mukaisen käyttöjärjestelmän luomiseen Buildroot-ohjelmistolla.

Busybox on ”sulautetun Linuxin sveitsiläinen linkkuveitsi”. Siihen on sisällytetty monia UNIX-järjestelmän työkaluja erittäin pieneen tilaan. Lisäksi sitä voi helposti muokata jättämällä pois tai lisäämällä osia oman tarpeen ja resurssien mukaan. Kuten U-Boot ja Linux-ydin myös Busybox on lisensoitu GPL-lisenssillä. /10/

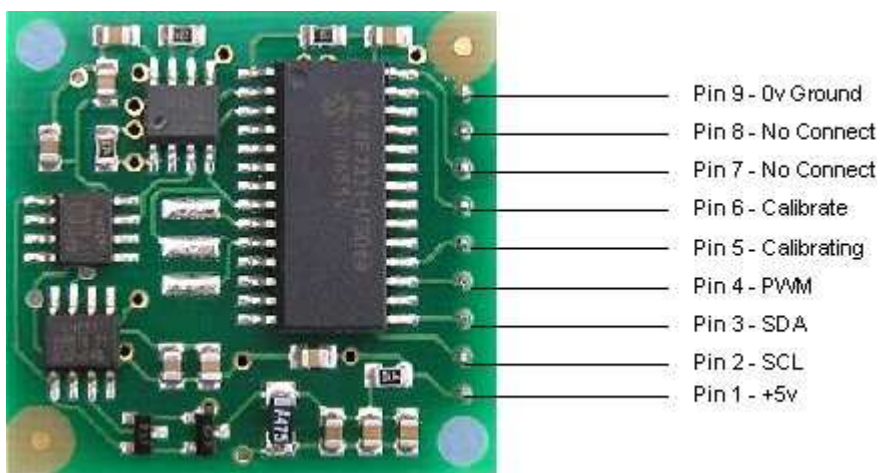
2.2 Elektroninen kompassi Devantech CMPS03

Devantechin valmistama CMPS03-elektroninen kompassi mittaa maapallon magneettikentän suuntaa. Se on erityisesti suunniteltu käytettäväksi roboteissa, jotka tar-

vitsevat suuntatiedon navigointia varten. Suunta on luettavissa joko I2C-väylää käyttäen tai PWM-muodossa./11/

Suunnan mittaus perustuu kahteen Philipsin KMZ51-magneetikentän voimakkuusanturiin, jotka on asetettu tiettyyn kulmaan toisiaan vasten. Tällöin niiden mittaus tuloksen erotuksesta voidaan laskea magneetikentän suunta. Laskennan hoitaa kortille sijoitettu 8-bittinen PIC16F872-mikro-ohjain, joka on lisäksi ohjelmoitu lähettämään tieto joko I2C-väylää käyttäen tai PWM-muodossa eteenpäin./11/

Kompassi mittaa magneetikenttää jatkuvasti ja tallentaa suuntatiedon eri rekistereihin. Kytkemällä kompassiin I2C-isäntälaitte kuvan 5 osoittamiin liittimiin, on rekistereistä mahdollista lukea suunta joko 8- tai 16-bittisenä arvona. I2C-protokolla on sama kuin yleisissä EEPROM-muisteissa. Kompassi voidaan myös kalibroida I2C-väylän kautta erilaisten maantieteellisten sijaintien mukaan. /11/



Kuva 5. Devantech CMPS03-elektroninen kompassi ja sen liittimet /11/

3 TIEDONHAKU

3.1 Sulautettu Linux-ohjelmistokehitys

Aloitettaessa Linux-pohjaista ohjelmistokehitystä lähes täysin ilman aiempaa Linux-kokemusta, on tarpeellista tutustua jonkin verran tämän käyttöjärjestelmän perusominaisuuksiin. Koska Linux kehittyy ja päivittyy jatkuvasti, on tarkan ja ajantasaisen tiedon hankkiminen painetusta kirjallisuudesta kohtalaisen hankalaa ja usein turhan vaivalloista. Kuitenkin painetulla sanalla on olemassa auktoriteetti, joka internetissä julkaistavalta tiedolta usein puuttuu.

Linuxista on vuosien saatossa kirjoitettu huomattava määrä kirjallisuutta, niin vastaalkajille kuin kokeneemmillekin käyttäjille. Englanniksi kirjoitettu tarjonta on luonnollisesti huomattavasti runsaampaa kuin suomeksi. Suurin osa kirjoista on tarkoitettu normaalin työpöytäkäyttöjärjestelmän käyttöoppaiksi, mutta Linuxin ydin on sama PC-tietokoneella ja sulautetulla laitteella, joten niistäkin on hyötyä. Joka tapauksessa sulautetun Linuxin ohjelmistokehitys tehdään todennäköisesti Linux-pohjaisella työpöytäkoneella, joten sen käyttö ja toiminta on myös ymmärrettävä.

Yksi vaihtoehto Linuxin peruskäytön opiskeluun kirjasta on Linux-koulutuspaketti (Kapanen 2004). Kirja on tarkoitettu käytettäväksi oppikirjana joko oppilaitoksissa tai itseopiskelussa, joten se sisältää paljon käytännön harjoituksia ja esimerkkejä. Kirjassa on noin 700 sivua. Kirjan tärkeimpiä aiheita ovat muun muassa Linuxin asennus ja ohjelmistojen käyttöönotto, Linuxin komentorivin käyttö, käyttöjärjestelmän ylläpito sekä Open Office-toimisto-ohjelmisto. Lisäksi kirjan alussa käsitellään yleisluontoisesti varsinkin Linux-käyttöjärjestelmällä varustetun tietokoneen käyttöä ja tiedonhallintaa. Muita kappaleita ovat WWW ja sähköposti, kuvankäsittely, tietokannan käsittely ja ohjelmoinnin perusteet. Hyvän asiasanahakemistonsa ansiosta kirjaa voi käyttää myös hakuteoksena etsittäessä ratkaisua tai ohjeita tiettyyn ongelmaan./12/

”Kuvittele maailma, jossa kaikilla on vapaa pääsy kaikkeen inhimilliseen tietämykseen. Se on tehtävämme.”-Jimmy Wales, Wikipedian perustaja

Wikipedia, ilmainen vapaan sisällön tietosanakirja, on yksi internetin ilmiöistä, johon tietoa internetistä hakeva ei voi olla törmäämättä. Ensimmäisen, alkuperäisen Wikipedian perustivat vuonna 2001 Internet-yrittäjä Jimmy Wales ja filosofi Larry Sanger, jonka jälkeen siitä on kehittynyt satoja versioita eri kielillä. Suomenkielinen Wikipedia sisältää tällä hetkellä 219 208 artikkelia, ollen maailman 14:ksi suurin. Lisäksi monet eri aihepiireihin keskittyneet yhteisöt, oppilaitokset ja yritykset ylläpitävät omia wiki-sivustojaan. Yleisluonteisten wikien lisenssiehtojen mukaan niiden sisältöä saa vapaasti kopioida, muokata ja levittää, kunhan edelleenjaettavien tietojen lisenssiehdot ovat samat ja Wikipedia mainitaan selkeästi lähteeksi./13/

Wikipediat perustuvat wiki-ohjelmistolla luotuun verkkosivustoon, jonne käyttäjät voivat yleensä vapaasti lisätä omia artikkeleitaan sekä korjailta muiden kirjoituksia. Juuri tässä vapaudessa on sekä wiki-sivustojen suurin vahvuus että heikkous. Ihanteellisessa tilanteessa wikiin kirjoittavat asiantuntijat, jotka käyttävät luotettavaa kirjallisuutta tai omia tutkimuksiaan lähteinä. Huonoimmassa tapauksissa wikissä esiintyvät häiriköt, jotka kiusanteon, omien mielipiteiden edistämisen tai puhtaan tietämättömyyden nimissä vääristelevät ja vahingoittavat wikin sisältöä.

Linux-yhteisöt ovat jo Linuxin avoimen luonteen takia taipuvaisia tekemään hyvää yhteistyötä wiki-tyyppisissä projekteissa. Linux on suurimmalta osin kehitetty vapaaehtoistoimintana, johon myös wiki-sivustot perustuvat. Häiriköitä ei Linux-yhteisöissä suvaita, joskin fanaatikot saattavat joskus olla hieman yliampuvia. Kuitenkin Linux-wikeistä löytyvä tieto on suurimmaksi osaksi tarkkaa ja täsmällistä, mikä lisäksi saattaa olla saatavissa ainoastaan, tai ainakin vähimmällä vaivalla, juuri kyseisiltä sivustoilta.

Ongelmia tietoteknisissä wikeissä aiheuttavat laitteistojen ja ohjelmistojen nopea päivittyminen, jonka takia tieto vanhentuu nopeasti. Sama tieto pitäisikin kirjoittaa ainakin osittain uudelleen aina laitteiden kehittyessä. Tästä johtuen kaikki wiki-si-

vustoilta saatava tieto olisi hyvä tarkistaa joko tutkimalla käytettyjä lähteitä tai kokeilemalla itse. Vanhentuneen tiedon lisäksi ongelmia saattaa joissain tapauksissa aiheuttaa wiki-ohjelmistojen luonteesta johtuva kokonaisrakenteen puute. Wiki on kokoelma yksittäisiä sivuja, joita käyttäjät lisäävät tarpeellisiksi katsomistaan aiheista. Oppikirjamaiseen, kokonaisuuden hallintaan tähtäävään tiedonhakuun wikit eivät välttämättä sovellu, tai ainakaan kokonaisuus ei ole helposti hallittavissa.

Verkko-osoitteesta <http://www.linux.fi/wiki> löytyy suomenkielinen Linux-wiki-sivusto, joka on avattu vuonna 2005. Artikkeleita löytyy viimeisimmän sivustolla olevan ilmoituksen mukaan noin 1200 kpl ja eri kävijöitä sivustolla on kuukausittain noin 30 000. Wikin aihealueet on etusivulla jaettu seuraavasti:

-uusi Linux-käyttäjä?

-perustietoa Linuxista

-ohjelmat

-laitteisto

-järjestelmä

-verkot ja Internet

-järjestäytyminen.

Linux.fi/wiki on hyvä esimerkki siitä, mitä wiki-sivustot voivat parhaimmillaan olla. Vaikka käyttäjä ei tietäisi tarkalleen mistä ohjelmasta tai Linux-komennosta tarvitsee tietoa, on sen löytäminen kohtalaisen helppoa aihealueiden jakamisen ansiosta. Myös haku tuntuu toimivan hyvin, ja muutenkin sivusto latautuu nopeasti. Wikin lisäksi sivustolla on keskustelualue, jossa voi sekä esittää kysymyksiä Linuxia koskien sekä keskustella wikin artikkeleista ja sivustosta yleisesti. Artikkelien teksti on yleensä hyvän kieliopin mukaista, ja vaikka tekstin sujuvuus ei ehkä aina ole julkaistavan kirjallisuuden tasoista, on se silti riittävän selkeää ratkaisun löytämiseksi ongelmiin./14/

Toinen esimerkki tämän työn aiheeseen liittyvästä wiki-sivustosta löytyy osoitteesta <http://avrfreaks.net/wiki>. Englanninkielinen sivusto keskittyy Atmelin AVR- ja

AVR32-tuoteperheiden mikro-ohjainten ominaisuuksiin ja niiden ohjelmistokehitykseen. Sisällöntuotannosta ja sen oikeellisuuden valvonnasta vastaavat aktiiviset käyttäjät./15/

Sivuston sisältö on etusivulla jaettu selkeästi kahteen alueeseen: AVR-(8-bittiset) ja AVR32-(32-bittiset) mikro-ohjaimet. Oikeastaan tähän sivuston rakenteen selkeys loppuukin sillä vaikka, esimerkiksi AVR32-osiossa on käytetty paljonkin alaotsikoita, on niiden sisällössä huomattavia päällekkäisyyksiä. NGW100 User guide-otsikon alta löytyy tietoa muun muassa aiheista *Downloading tools, tools installation* ja *Linux development on STK1000 and NGW*. Samoja asioita käsitellään kuitenkin myös ylemmällä otsikkotasolla aiheissa *AVR32 Tools* ja *AVR32 Software*. Osa aiheista sisältää toisaalla vanhentunutta tietoa, toisaalla taas samat asiat on käsitelty hieman eri sanoilla ja esitystavoilla. Rakenteen puuttumisen lisäksi tekstissä on ongelmia. Artikkeleita sivustolle tuottavat useaa eri kansallisuus- ja kieliryhmää edustavat ihmiset, joiden kielitaito ja kirjoitetun esityksen taso on jokseenkin vaihtelevaa. Kirjoitusvirheitä esiintyy jonkin verran ja ohjeiden eteneminen on joissain artikkeleissa epäloogista. Kaikista puutteistaan huolimatta sivusto on kuitenkin ainut paikka, josta aiheeseen liittyvää tietoa on kattavasti saatavilla. Wikin lisäksi sivustoon kuuluu keskustelualue, jolla ainakin ajoittain esiintyy todellista asiantuntemusta kokeneiden käyttäjien antaessa tyhjentäviä vastauksia kokemattomampien kysymyksiin./15/

Sulautetun elektroniikan Linuxin ja työpöytäkäyttöjärjestelmän ohjelmointi eivät juurikaan eroa toisistaan. Käytettävät kirjastot ja kääntäjä ovat eri versiota, mutta hyvin kirjoitetun C-kielisen ohjelman voi siirtää eri ympäristöön ilman suuria muutoksia itse koodiin. Siksi sama ohjelmointikirjallisuus on käyttökelpoista sekä sulautetun että työpöytä-Linuxin ohjelmia kirjoitettaessa.

Hyvä perusteos C-ohjelmointiin on Simo Silanderin Ohjelmointi – Pro training (Satku-Kauppakaari 2000). Oman kuvauksensa mukaan kirja on täydellinen itseopiskelupaketti ohjelmointiin. Kirja käsittelee noin 340 sivullaan ohjelmoinnin käsitteitä ja rakenteita, käy läpi sekä rakenteellisen että modulaarisen ohjelmoinnin ominais-

piirteet sekä tutustuu erilaisiin tietorakenteisiin ja C-kieleen erityisesti./16/

Toinen hyvä tietolähde on sekä internetistä pdf-muodossa että painettuna kirjana löytyvä englanninkielinen *Advanced Linux Programming* (New Riders Publishing 2001). Kirjoittajina ovat toimineet Mark Mitchell, Jeffrey Oldham ja Alex Samuel, CodeSourcery LLC yhtiöstä. Nimensä mukaisesti kirja sopii hieman edistyneempään käyttöön kuin Silanderin perusteos ja keskittyykin huomattavissa määrin hyvän, tietoturvallisen ja Linux-ympäristöä tehokkaasti hyödyntävän koodin luomiseen./17/

3.2 AVR32 ja Buildroot

Linuxin käyttöön AVR32-ympäristössä on hyvä tietolähde aiemmin mainitun AVRFreaks.netin lisäksi verkko-osoitteessa <http://www.avr32linux.org>. Kyseessä on myös wiki-sivusto, joten sen tarjoamiin tietoihin tulee suhtautua oikealla tavalla.

Sivuston Howto-osiosta löytyvät ohjeet monen aloittelevan AVR32-ympäristön käyttäjän ongelmiin, muun muassa AVR32 Linux-kehitysympäristön asennukseen, uuden käyttöjärjestelmäytimen kääntämiseen sekä moniin eri tapoihin muokata ja asentaa käyttöjärjestelmä laitteelle. Näiden ohjeiden lisäksi sivustolle on kerätty suurehko määrä erilaisia AVR32 Linuxiin liittyviä päivityspaketteja, sovellusohjelmia ja kehitystyökaluja. Yhtenä erikoisuutena löytyvät ohjeet Linuxin asentamiseen Windows-tietokoneelle VMware Player-virtuaalikoneen avulla./18/

Tämän raportin kirjoittamisen aikaan avr32linux.org-sivusto on ollut hakkerin hyökäyksen johdosta poissa käytöstä useita kuukausia. Sivuston tiedot ja ohjeet ovat edelleen luettavissa osoitteessa test.avr32linux.org, mutta niitä ei voi muokata. Monet artikkelit ovatkin jo hieman vanhentuneita ja niiden tiedot tai mahdolliset uudet tavat tehdä samoja asioita tulisi tarkistaa myös toisesta lähteestä, esimerkiksi avrfreaks.netin wikistä tai keskustelupalstalta.

Buildroot on Atmelin kehittämä työkalu täydellisen käyttöjärjestelmäpaketin luomi-

seen halutuilla argumenteilla STK1000- tai NGW100-järjestelmään. Virallinen dokumentaatio ohjelmiston käyttöön löytyy Atmelin Norjan yksikön kotisivuilta osoitteesta <http://www.atmel.no/buildroot./19/>

Hyvä tietolähde Buildrootin kanssa alkuun pääsemiseksi on Atmelin Application note AVR32003. Kyseessä on virallinen dokumentaatio, josta vastaa Atmel, joten se on wiki-sivustoihin verrattaessa luotettava ja validi tietopaketti. Dokumentissa kerrotaan selkeästi mistä Buildroot koostuu, mitä sen käyttämiseksi vaaditaan järjestelmältä ja mitä sillä voi tehdä. Jo pelkän Getting Started-luvun ohjeilla käyttäjä saa sekä asennettua Buildrootin että luotua uuden käyttöjärjestelmäpaketin haluamalleen laitteelle. Aloittelijan ohjeiden jälkeen dokumentissa käydään tarkasti läpi Buildrootin tiedostorakenne eli kuinka luotavaa käyttöjärjestelmäpakettia voidaan muokata omiin tarpeisiin sekä kuinka valmiita tiedostoja käytetään uuden käyttöjärjestelmän siirtämiseksi, esimerkiksi NGW-laitteelle. Dokumentti ei vastaa kysymykseen ”Kuinka saan ominaisuuden X otettua käyttöön?”, vaan se opastaa perusasetusten tekemisen sekä sen mistä näitä asetuksia voi muuttaa. Käyttäjän vastuulle jää sekä löytää eri ominaisuuksien sijainti Buildrootin valikoissa että päättää mitä niistä tarvitaan./20/

3.3 I2C-väylä

I2C (Inter-Integrated Circuit bus) on alun perin Philipsin kehittämä yksinkertainen kaksijohtiminen sarjaväylä, jota käytetään yleensä mikro-ohjaimen ja sen oheislaitteiden väliseen tiedonsiirtoon. Täydellinen määrittely väylälle löytyy Philipsin verkkosivuilta osoitteesta <http://www.nxp.com>. 46-sivuisessa dokumentissa esitellään tarkasti väylän versiohistoria, sen hyödyt käyttäjille, tekniset ominaisuudet sekä Philipsin tarjoamat erilaiset kehitystyökalut I2C-väylälle.

Käyttäjän ei aina ole välttämättä tarpeen tuntea käyttämänsä tiedonsiirtotavan jokaisista teknistä yksityiskohtaa. Tällöin edelläkuvatun kaltainen täydellinen määrittely on usein turhan raskasta ja aikaa vievää luettavaa siitä saatuihin hyötyihin nähden.

I2C:n tyyppiseen, yleiskäyttöiseen teknologiaan on saatavilla huomattavasti yksinkertaisemmin esitettyjä tietopaketteja, monilla eri kielillä. Tästä esimerkkinä voi toimia Jari Koskisen Mikrotietokonetekniikka-oppikirja, jossa on käsitelty myös I2C-väylää tarpeellisessa määrin sen käyttämisen mahdollistamiseksi. /21/,/22/

4 TYÖN ETENEMINEN

4.1 Tutustuminen aiheeseen

NGW-korttiin tutustuminen aloitettiin jo ennen kuin itse laite oli käytettävissä. Avrfreaks.net wikistä kortin ominaisuuksia ja erilaisia liitännävaihtoehtoja tutkimalla saatiin kohtalaisen hyvä käsitys siitä mihin kortti pystyy. Myös työpöytäkoneelle asennettiin Windows XP:n rinnalle Linux-käyttöjärjestelmä (Ubuntu Hardy Heron) ja tutustuttiin sen peruskäyttöön jonkin verran.

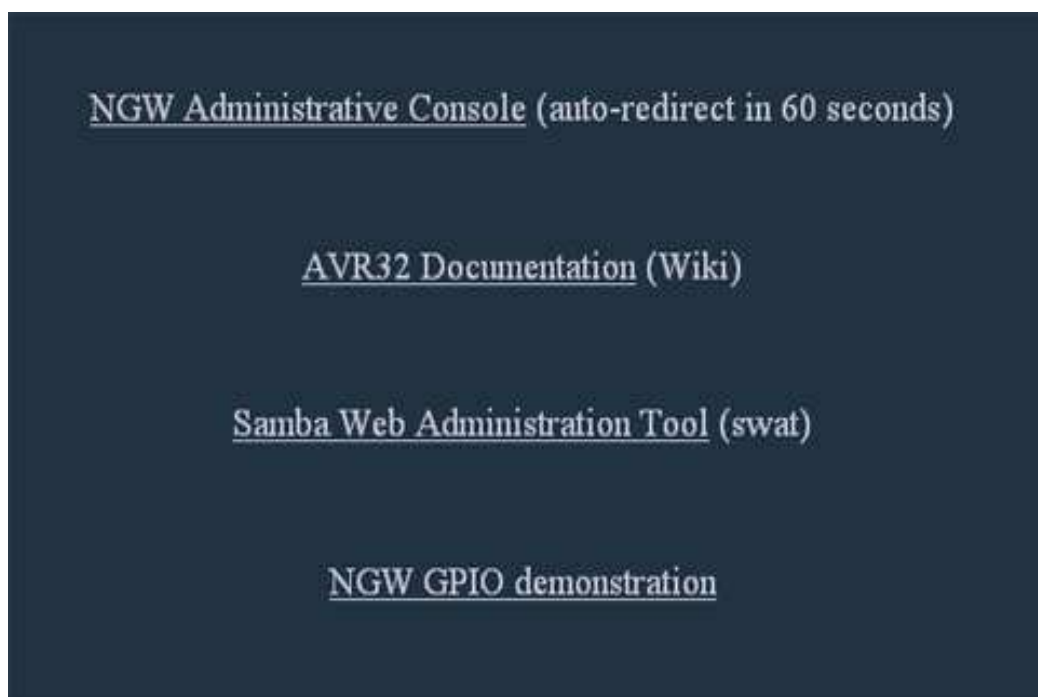
Kortin saavuttua se kytkettiin ADSL-modeemin ja tietokoneen väliin reitittimeksi verkkoliitännöjen kautta. Lisäksi oli hankittu USB-RS232-kaapeli, koska käytettävissä tietokoneessa ei ollut normaalia sarjaporttiliitaintä. Tällä kytkettiin tietokone NGW:n sarjaporttiin. Virtajohtoa tai muuntajaa kortin mukana ei tullut, mutta tehollähteeksi kelpaa mikä tahansa yleiskäyttöinen 9-15 voltin tasajännitelähde, jonka napaisuudella ei ole merkitystä. Myöskään virtakytkintä kortilla ei ole, mutta RESET-nappi löytyy.

Verkkoyhteys toimi heti moitteetta, ja tietokone sekä NGW pystyttiin yhdistämään Telnetin kautta. Kortin IP-osoite on oletuksena 10.0.0.1. Avrfreaks.net wiki-sivuilta saadun vaikutelman mukaan verkkoyhteys ei välttämättä toimisi kortin halutessa toimia myös DNS-palvelimena, mutta tällaisia ongelmia ei esiintynyt. Sarjaporttia käytettiin Ubuntun GtkTerm-ohjelman avulla, jonka yhteysasetuksissa nopeudeksi tulee valita 115200 bps, databittejä 8, ei pariteettia ja stopbittejä 1. Myös NGW:n muistikortinlukija testattiin: lukijaan asetettu 1 Gt muistikortti näkyi Linuxin tiedostojärjestelmässä hakemistossa /media/mmcblk0p1. Kortti oli alustettu työpöytäkoneessa EXT2-tiedostojärjestelmälle.

Telnetin ja sarjaportin kautta pystyttiin tutustumaan NGW:lle asennetun Linuxin tiedostojärjestelmään, joka on melko samankaltainen kuin työpöytäkäyttöjärjestelmäs-

sä. Sarjaportin kautta näkee myös mitä kortin käynnistyessä tapahtuu, koska oletusasetuksilla se lähettää kaikki Linuxin konsoliin tulevat ilmoitukset sarjaportille.

NGW toimii myös www-palvelimena. Kortin ollessa kytkettynä tietokoneen verkkoliitännään on normaalilla verkkoselaimella mahdollista navigoida joko numeeriseen IP-osoitteeseen 10.0.0.1 tai tutummalla tavalla osoitteeseen <http://ngw.example.net>. Osoitteesta löytyvät linkit sekä reitittimen että Samba-hallinnan asetuksiin, kortille tallennettuun versioon avrfreaks.net wikistä, sekä kortille sijoitetun LED-valon vilkuttamiseen kykenevään demo-ohjelmaan (kuva 6). Reitittimen ja Samban asetukset eivät tässä työssä kaivanneet huomiota, LED vilkkui, ja avrfreaks.net wikiä on parempi lukea alkuperäisestä osoitteesta, jossa tietoja päivitetään tarpeen mukaan. Voitiin kuitenkin todeta että laite toimii kuten sen pitääkin myös näiltä osin.

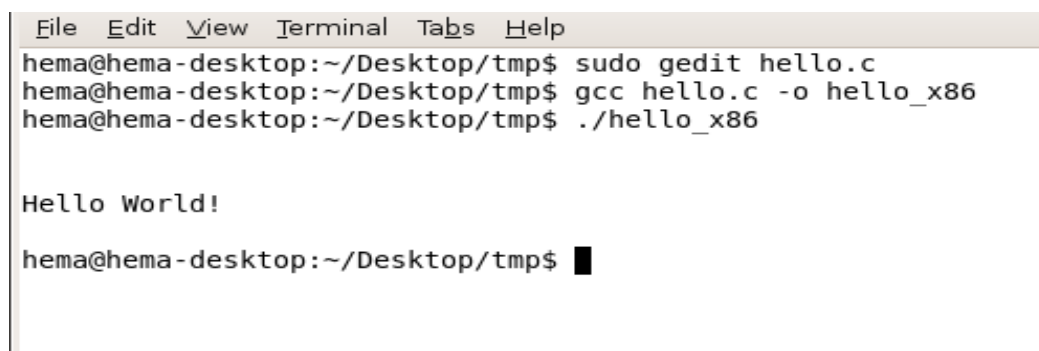


Kuva 6. NGW:n www-palvelimen näkymä osoitteessa <http://ngw.example.net>

4.2 AVR32 Linux ohjelmointityökalujen asennus

Kortin osoittauduttua toimivaksi, oli aika siirtyä miettimään sen ohjelmistoa. Tyypillinen tapa aloittaa ohjelmointi uudessa ympäristössä on kirjoittaa ja kääntää pieni *Hello World*-ohjelma, jolla voidaan todeta, että kääntäjä ja ympäristö pääpiirteissään toimivat. Ohjelman ainut toiminto on tulostaa Hello World-teksti näytölle tai tässä tapauksessa konsoliin.

GCC-kääntäjä asennettiin Ubuntuun komennolla *sudo apt-get install build-essential*. Gedit-tekstieditorilla kirjoitettiin muutaman rivin ohjelma *hello.c*-tiedostoon, joka käännettiin GCC-kääntäjällä ja ajettiin komentokehotteesta (kuva 7).



```
File Edit View Terminal Tabs Help
hema@hema-desktop:~/Desktop/tmp$ sudo gedit hello.c
hema@hema-desktop:~/Desktop/tmp$ gcc hello.c -o hello_x86
hema@hema-desktop:~/Desktop/tmp$ ./hello_x86

Hello World!

hema@hema-desktop:~/Desktop/tmp$ █
```

Kuva 7. Hello World-ohjelman käänös ja ajo x86-alustalla

Ohjelma toimi oikein, mutta tämä käänös oli tehty vasta x86-alustalle eli normaalille PC-tietokoneelle. Toimiakseen AVR32-alustalla, kuten NGW:llä, tarvittiin AVR32 Linux-versio GCC-kääntäjästä, eli AVR32 Linux-ohjelmointityökalut (engl. *toolchain*).

Atmel on tehnyt ohjelmointityökalujen asennuksesta kohtalaisen helppoa. Kaikki tarvittavat ohjelmapaketit on lisätty Atmel-ohjelmavarastoon (engl. *repository*), joka asentamalla kaiken pitäisi toimia. Kyseinen ohjelmavarasto ei kuulu Ubuntun alkuperäiseen listaan lähdekoodivarastoista, joten se on lisättävä sinne käsin. Käytännös-

sä tämä tapahtuu lisäämällä tekstieditorilla tiedostoon */etc/apt/sources.list* seuraava rivi:

```
deb http://www.atmel.no/avr32/ubuntu/hardy binary/
```

Tämän jälkeen Ubuntuun asennusohjelma *apt* osaa hakea Atmelin ohjelmointityökalut oikeasta osoitteesta seuraavilla komennoilla:

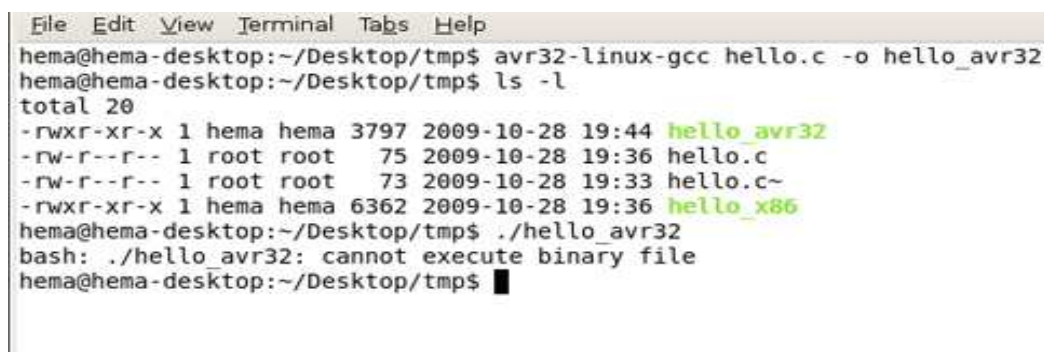
```
sudo apt-get update
```

```
sudo apt-get install avr32-gnu-toolchain
```

Mikäli virheilmoituksia ei tule, työkalut ovat valmiita käytettäväksi. Ne on myös helppo päivittää, koska nyt *apt* osaa automaattisesti etsiä niihin uusia versioita. Seuraavaksi käännettiin sama, aiemmin kirjoitettu *hello.c*-tiedosto *avr32-linux-gcc*-kääntäjällä komennolla:

```
avr32-linux-gcc hello.c -o hello_avr32
```

Käännös onnistui ja yritettäessä ajaa ohjelmaa työpöytä-Linuxin komentokehotteesta ohjelma ei toiminut, koska nyt oli kyseessä AVR32-alustalle käännetty ohjelma (kuva 8).



```
File Edit View Terminal Tabs Help
hema@hema-desktop:~/Desktop/tmp$ avr32-linux-gcc hello.c -o hello_avr32
hema@hema-desktop:~/Desktop/tmp$ ls -l
total 20
-rwxr-xr-x 1 hema hema 3797 2009-10-28 19:44 hello_avr32
-rw-r--r-- 1 root root 75 2009-10-28 19:36 hello.c
-rw-r--r-- 1 root root 73 2009-10-28 19:33 hello.c~
-rwxr-xr-x 1 hema hema 6362 2009-10-28 19:36 hello_x86
hema@hema-desktop:~/Desktop/tmp$ ./hello_avr32
bash: ./hello_avr32: cannot execute binary file
hema@hema-desktop:~/Desktop/tmp$
```

Kuva 8. Hello World-ohjelman käännös AVR32 Linux-työkaluilla

Hello_avr32-ohjelman siirrettiin SD-muistikortilla NGW:lle ja ajettiin siellä sarjaportin kautta. Nyt ohjelma toimi kuvan 9 mukaan odotetusti:


```

~ # cd media/mmcblk0p1/softa2
/media/mmcblk0p1/softa2 # ls -l
-rwxr-xr-x  1 root  root           3797 Oct 28 18:42 hello_avr32
-rwxr-xr-x  1 root  root           5504 Sep 23 15:47 i2c_heman
/media/mmcblk0p1/softa2 # ./hello_avr32

Hello World!

/media/mmcblk0p1/softa2 # █
devtty@980: 115200 B.N.1 DTR RTS

```

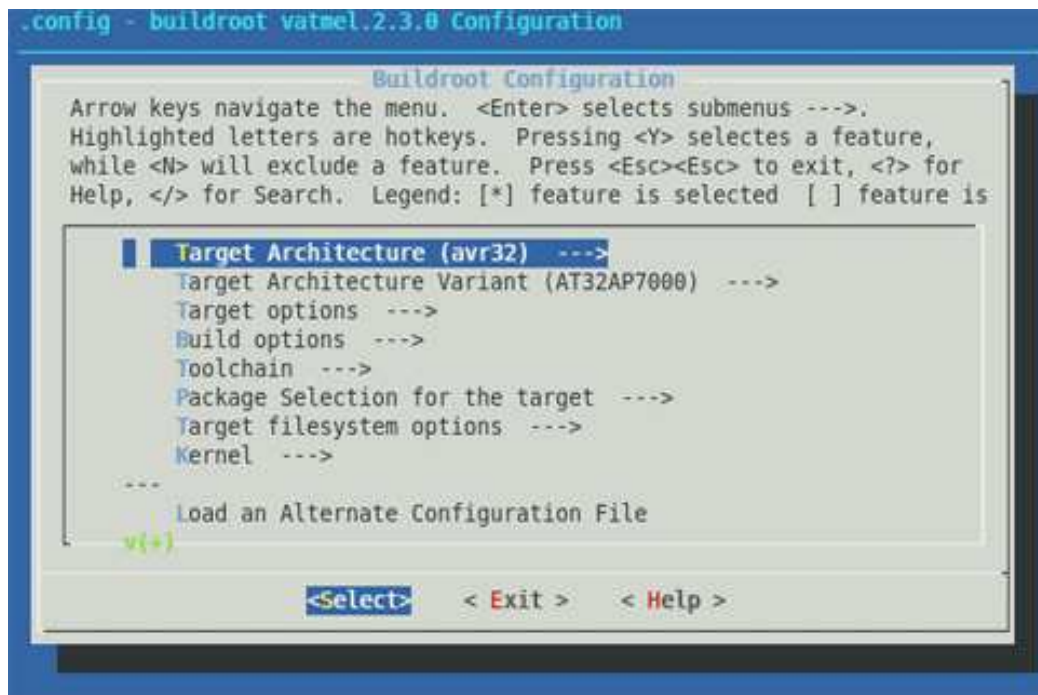
Kuva 9. Hello World-ohjelman ajo NGW:llä

4.3 Buildroot-käyttö

Jo kohtalaisen varhaisessa vaiheessa aiheeseen tutustumista törmättiin ohjelmistoon nimeltä Buildroot. Kyseessä ei ole varsinainen ohjelmoitu ohjelma vaan sarja Makefile-komentoja, komentosarjoja sekä valikkojärjestelmä, joiden avulla on mahdollista luoda kokonaan uusi tiedostojärjestelmä kohdelaitteelle, joka voi olla NGW tai STK1000. Tärkeimmät käyttökohteet Buildrootilla ovat uuden, päivitetyn käyttöjärjestelmän luominen kohdejärjestelmään sekä omalle sovellukselle tai laitteelle tarpeellisten ominaisuuksien käyttöönotto Linuxissa. Asentaminen on helppoa: haetaan paketti osoitteesta www.atmel.no/buildroot ja puretaan sen sisältö mihin tahansa sopivaan hakemistoon. Buildroot on välittömästi käytettävissä komentokehotteesta.

Käytön aloittaminen Atmelin dokumentin *AVR32003: AVR32 AP7 Linux Buildroot* mukaan on yksinkertaista. Siirrytään komentokehotteesta hakemistoon, johon Buildroot-paketti puretaan ja ladataan asetustiedostoon (kconfig) kohdejärjestelmän oletusasetukset. NGW:n ollessa kyseessä tämä tehdään komennolla `make atngw100_defconfig`. Tämän jälkeen ohje suosittelee aloittamaan tarvittavien pakettien lataamisen ja kääntämisen komennolla `make`, mutta ennen sitä kannattaa kiinnittää huomiota pariin asiaan.

Buildrootin ja sen luoman tiedostojärjestelmän yleisiä asetuksia pääsee muuttamaan komennolla `make menuconfig`, joka tuo näytölle kuvan 10 mukaisen valikon:

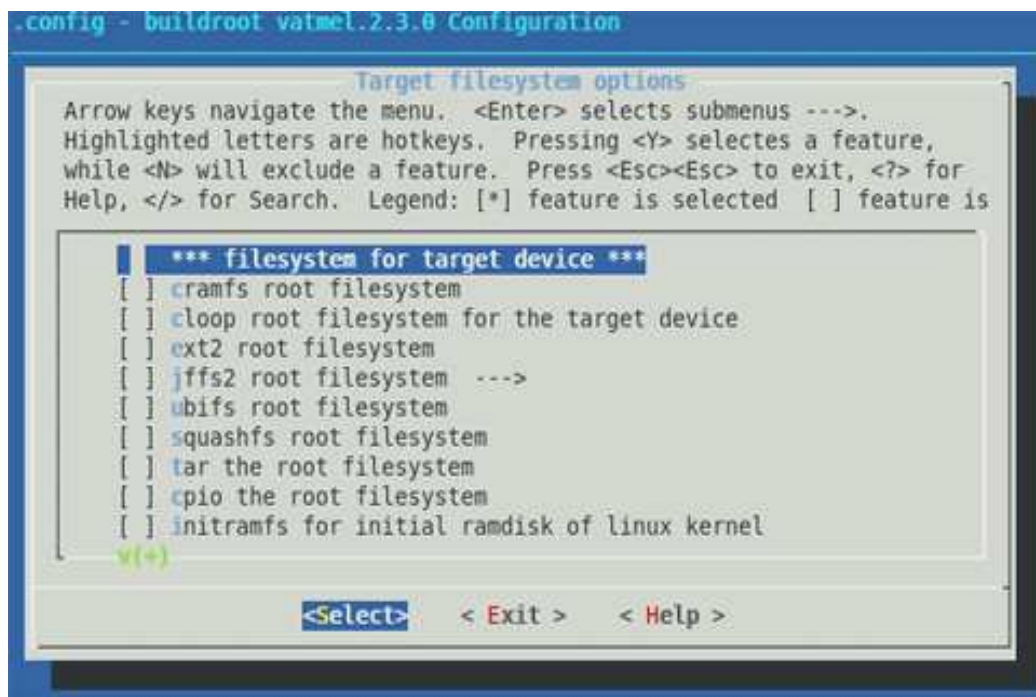


Kuva 10. Buildroot menuconfig

Tässä valikossa voidaan muun muassa valita kohdearkkitehtuuri ja laitteen prosessori, syöttää mahdolliset projektin tiedot sekä tehdä valintoja liittyen Buildrootin tapaan hakea asennuspaketteja ja kääntää niitä. Lisäksi valikosta löytyy vaihtoehtoja koskien ladattavia ohjelmapaketteja, Linux-ytimen asetuksia sekä luotavan tiedostojärjestelmän tyyppiä.

Työstä saadun kokemuksen mukaan menuconfig-valikossa on heti alkuvaiheessa syytä tehdä kaksi muutosta. Atmel on poistanut NGW:lle asennettavat wiki-dokumentit alkuperäisestä sijainnistaan, josta Buildroot yrittää niitä hakea. Tästä syystä make-komento antaa virheilmoituksen, joka estää käännöksen suorittamisen. Helppo ratkaisu on poistaa wiki-dokumentit asennettavien pakettien joukosta. Menuconfig-valikossa siirrytään kohtaan *Package Selection for the target* ja poistetaan sen valinnoista rasti ruudusta *avr32-wiki-docs*. Paketti on myös mahdollista hakea ja asentaa erikseen, mutta siihen ei nähty tarvetta, koska kaikki dokumentit ajantasaisempina versioina ovat saatavilla avrfreaks.net-sivustolla.

Toinen muutos liittyy luotavan järjestelmän tiedostomuotoon. Jos tässä vaiheessa projektia tiedetään, että järjestelmä on tarkoitus siirtää NGW:lle käyttäen esimerkiksi muistikorttia, kannattaa sille sopiva tiedostojärjestelmä valita heti. Tämä valinta tehdään valikossa *Target filesystem options* (kuva 11):



Kuva 11. Target filesystem options-valikko

Jos tarkoitus on kirjoittaa uusi järjestelmä suoraan NGW:n omaan muistiin, käytetään jffs2-tiedostojärjestelmää. SD- tai MMC-muistikorttia käytettäessä paras valinta on ext2, koska tällöin myös isäntäkoneella voidaan tehdä muutoksia suoraan kortilla oleviin tiedostoihin. Myös ext3 on käyttökelpoinen, mutta koska se on journaloiva tiedostojärjestelmä, eli kirjoittaa jatkuvasti lokia tapahtumista, se kuluttaa muistikortin rajallisia kirjoituskertoja.

Muita muutoksia menuconfig-valikossa ei ainakaan tässä vaiheessa tarvitse tehdä, vaan valikosta voidaan poistua tallentaen muutokset. Komentokehotteessa voidaan tarkastaa tehtyjen valintojen toimivuus komennolla *make dependencies*, jonka pitäisi palauttaa "Ok" jokaiseen kohtaan. Tämän jälkeen uuden järjestelmän pakettien haku

ja kääntäminen voidaan aloittaa komennolla *make*. Ennen tätä on mahdollista erikseen hakea tarvittavat paketit komennolla *make source*, ja suorittaa varsinainen käännös vasta tämän jälkeen. Pakettien haku ja käännös kestää ensimmäisellä kerralla pitkään mutta myöhemmin tehtäviä muutoksia varten suurin osa tarvittavista paketeista on nyt haettu, eikä sitä tarvitse tehdä uudestaan.

Käännöksen jälkeen Buildroot-hakemistosta */binaries/atngw100* löytyy valmis tiedostojärjestelmä sekä levykuvana *rootfs.avr32.ext2* että gzip-pakattuna tiedostona *atngw100-linux-2.6.27.6.gz*. Levykuvan voi Linuxissa liittää olemassa olevaan hakemistoon, josta se on mahdollista kopioida muistikortille tai pakatun tiedoston voi purkaa suoraan kortille.

Buildrootin yleisiä asetuksia koskevan menuconfig-valikon lisäksi Buildrootissa on valikko nimeltä linux26-menuconfig, jossa voidaan määritellä luotavan Linux-ytimen asetuksia. Näihin kuuluvat muun muassa asennettavat laiteajurit, käyttöjärjestelmän tukemat tiedostojärjestelmät sekä monia muita käyttöjärjestelmäytimen toimintaan vaikuttavia asetuksia. Näitä valintoja muokkaamalla on myös mahdollista pienentää käyttöjärjestelmän tilantarvetta sekä ottaa käyttöön ominaisuuksia joita saatetaan tarvita itse valmistetuilla laitteilla, joiden ominaisuudet eroavat Atmelin NGW- ja STK1000-korteista.

4.4 Uuden käyttöjärjestelmän siirtäminen NGW:lle

Kun uusi käyttöjärjestelmä on luotu Buildrootilla se täytyy siirtää kohdelaitteelle. NGW:llä mahdollisuuksia on kolme: kortilla oleva FLASH-muisti, SD- tai MMC-muistikortti sekä verkon kautta käynnistyvä käyttöjärjestelmä. Käytössä ei ollut kortilla olevan muistin ohjelmoimiseen tarvittavaa laitteistoa, eikä verkon kautta käynnistämistä pidetty tarpeellisena ominaisuutena, joten käytettäväksi tavaksi valittiin SD-muistikortti. Verkon kautta käynnistyvä järjestelmä voi olla käytännöllinen, esimerkiksi tehtäessä raskasta ohjelmistokehitystä, jolloin kirjoitetun ohjelman jokaista eri versiota ei tarvitse erikseen siirtää laitteelle. Lisäksi verkon kautta käyn-

nistyminen saattaa tulla tarpeeseen joissain tietyissä valmiissa sovelluksissa. SD-muistikortin käyttö on verrattain yksinkertaista ja varsinkin edullista, jonka lisäksi se sopii oppilaitoskäyttöön hyvin, koska itse laitteen muistissa olevaan järjestelmään ei juurikaan tarvitse tehdä muutoksia. Tällöin laite voidaan siirtää opiskelijalta toiselle ilman suuria toimenpiteitä ja jokaisen opiskelijan omat työt säilyvät omilla muistikorteilla.

Aloittelevalle Linux-käyttäjälle valmiin käyttöjärjestelmän siirtäminen muistikortille ei ole niin yksinkertainen tehtävä kuin miltä se saattaa kuulostaa. Asiaa tutkittua todettiin yksinkertaisimmaksi tavaksi Buildrootin tuottaman levykuvan liittämisen tätä varten luotuun hakemistoon, esimerkiksi työpöydällä ja tiedostojen kopioimisen sieltä kuvan 12 mukaan:

```
hema@miniubuntu:~/Desktop$ mkdir tmp
hema@miniubuntu:~/Desktop$ cd buildroot-avr32-
v2.3.0/binaries/atngw100/
hema@miniubuntu:~/Desktop/buildroot-avr32-
v2.3.0/binaries/atngw100$ sudo mount -o loop rootfs.avr32.ext2
~/Desktop/tmp
```

Kuva 12. Buildrootin luoman levykuvan liittäminen työpöydällä olevaan hakemistoon

Käytössä oli 1 Gt SD-muistikortti, joka piti alustaa ext2-tiedostojärjestelmälle. Tämä tapahtuu Linuxissa, esimerkiksi mkfs-ohjelmalla (kuva 13).

```

hema@miniubuntu:~/Desktop/tmp$ sudo umount /dev/sdb1
hema@miniubuntu:~/Desktop/tmp$ sudo mkfs -t ext2 /dev/sdb1
mke2fs 1.40.8 (13-Mar-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
62976 inodes, 251489 blocks
12574 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=260046848
8 block groups
32768 blocks per group, 32768 fragments per group
7872 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to
override.

```

Kuva 13. EXT2-tiedostojärjestelmän luominen muistikortille /dev/sdb1

Muistikortin ja levykuvan sisältävän hakemiston ollessa valmiina, on jäljellä enää käyttöjärjestelmän tiedostojen kopioiminen muistikortille. Linuxissa tämä tapahtuu siirtymällä lähdehakemistoon ja käyttämällä cp-komentoa seuraavilla parametreillä:

```
sudo cp -r -f* /media/disk-1
```

Huomioon otettavaa on, että */media/disk-1* on oletettu muistikortin liitoskohta, joka voi olla jokin muukin. Asian voi tarkastaa komennolla *df*. Tämän jälkeen *ls -l* komennolla voidaan tutkia tiedostojen löytyminen muistikortilta.

Muistikortilla on nyt valmis järjestelmä, mutta NGW ei vielä osaa käynnistää sitä. Koska kyseessä ei ole varsinainen plug'n'play-laite, täytyy sille määritellä mistä käyttöjärjestelmä löytyy sekä eräitä muita asetuksia. Tämä tapahtuu muokkaamalla U-Boot-käynnistyslatainta. Koko U-Boot-ohjelman päivittäminen vaatisi ohjelmointilaitteen jota ei ollut käytössä. Onneksi tämä ei ollut tarpeen, vaan kortille asennettun ohjelman asetusten muuttaminen riitti.

U-Bootin asetusten muuttamiseksi NGW täytyy kytkeä isäntäkoneeseen sarjaportin kautta. Käytetyssä työpöytäkoneessa ei ollut sarjaporttia, joten liittämiseen käytettiin USB-RS232-kaapelia. Terminaali-ohjelmaan, Linuxissa esimerkiksi GtTerm, valitaan asetuksiin yhteyden nopeudeksi 115200 bps, kahdeksan databittiä, ei pariteettia ja yksi stop-bitti. Tämän jälkeen NGW:n voi käynnistää.

U-Boot käynnistyy heti virran kytkemisen jälkeen, joten tässä vaiheessa täytyy olla melko nopea. Ruudulle ilmestyy sekunnin ajaksi teksti *Press SPACE to abort autoboot in 1 seconds*, jolloin välilyöntiä painamalla päästään U-Bootin valikkoon. Käytössä olevat asetukset, jotka kannattaakin kirjoittaa talteen, voi tulostaa ruudulle komennolla *printenv*. Näin ne on helppo palauttaa, jos jokin menee vikaan.

U-Bootin tarvitsemia tietoja ovat käyttöjärjestelmän tiedostojärjestelmä, sen sijainti muistiavaruudessa sekä konsolin osoite. Lisäksi asetuksiin määritellään muistikortinlukijan käynnistäminen käynnistyksen yhteydessä sekä tähän tarvittava pieni viive. Nämä asetukset annetaan seuraavilla komennoilla:

```
Uboot>askenv bootcmd
```

```
Please enter 'bootcmd':mmcinit; ext2load mmc 0:1 0x10400000 /boot/uImage;  
bootm
```

```
Uboot>set bootargs 'console=ttyS0 root=/dev/mmcblk0p1 rootwait'
```

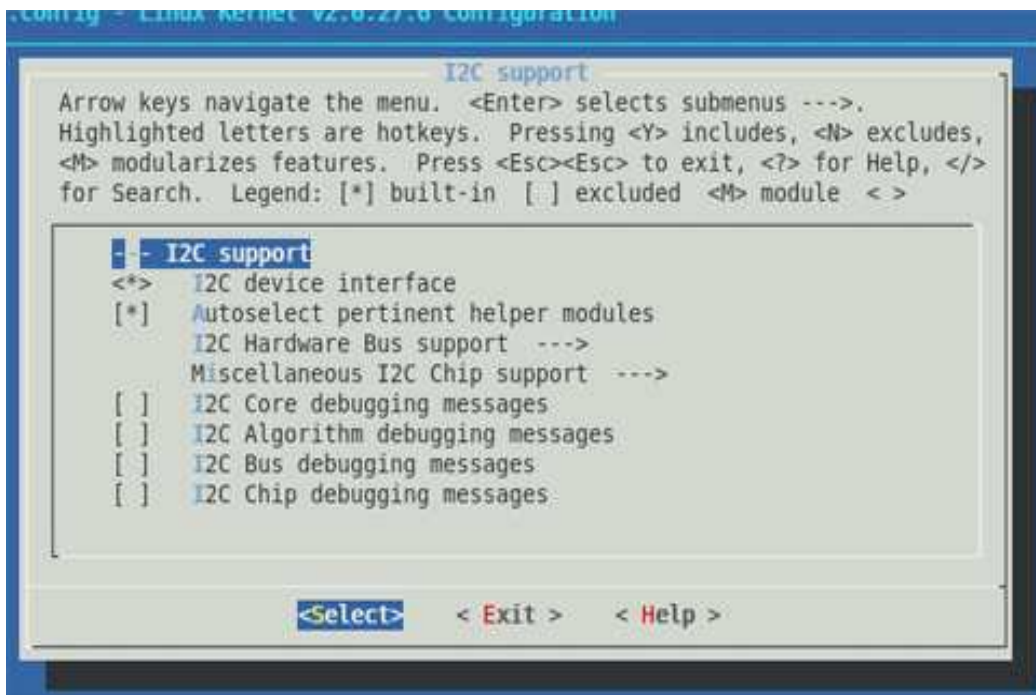
```
Uboot>saveenv
```

```
Uboot>boot
```

Komentojen antamisen jälkeen NGW käynnistyy uudestaan ja lataa käyttöjärjestelmän muistikortilta. Käynnistymisvaiheessa voi huomata joitain eroja alkuperäiseen verrattuna, mutta varmuuden käyttöjärjestelmän versiosta saa esimerkiksi tarkistamalla sen komentokehotteesta komennolla *uname -r*.

4.5 I2C-käyttö Linuxissa ja NGW:llä

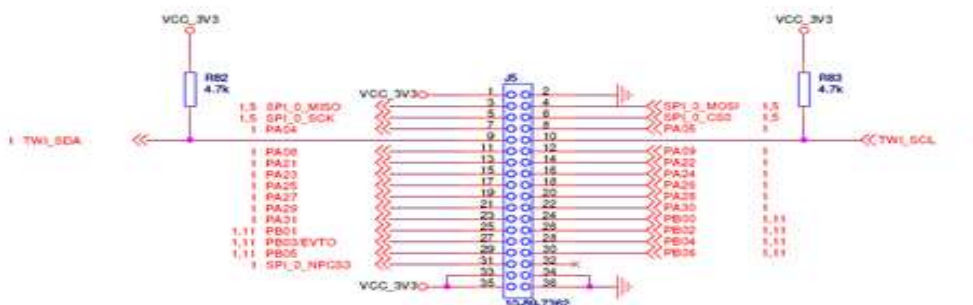
Linuxissa oheislaitteita ja järjestelmän sisäisiä toimintoja, joihin I2C-väylä kuuluu, käytetään laitetiedostojen kautta. Laitetiedostot sijaitsevat tiedostojärjestelmässä /dev-hakemistossa kuten esimerkiksi normaalin PC-tietokoneen kiintolevyjen laitetiedostot /dev/sda1 ja /dev/sda2. Laitetiedosto lisätään ottamalla kyseisen laitteen laiteajurit käyttöön Linuxin ytimen asetuksissa. Buildroot-ohjelmalla tämä on helppoa: komennolla *make linux26-menuconfig* päästään oikeaan valikkoon, josta otetaan käyttöön *Device Drivers, I2C support, I2C device interface*, sekä vielä tämän alavalikosta *I2C Hardware Bus support, GPIO-based bitbanging I2C* (kuva 14). Näiden valintojen jälkeen käännetyllä ytimellä käynnistetyn järjestelmän /dev-hakemistosta löytyy laitetiedosto i2c-0, eli laitteen I2C-väylä on valmis käytettäväksi.



Kuva 14. I2C-valinnat Buildrootin linux26-menuconfig-valikossa

NGW:n piirikaavion mukaan I2C-väylän tarvitsemien kello- ja dataväyliä

liitinnastat ovat J5-liittimessä numerot 9 ja 10 (kuva 15). Nämä kytketään CMPS03-kompassin nastoihin 3 ja 2. Lisäksi kompassille tulee järjestää viiden voltin käyttöjännite, joka otettiin vastuksen kautta samalta muuntajalta kuin NGW:n 12 voltia. Näiden kytkentöjen jälkeen kompassi ja NGW ovat valmiita käytettäväksi, tarvitaan vain ohjelma joka osaa niitä käyttää.



Kuva 15. NGW:n piirikaavio I2C-liitintien osalta (TWI_SDA ja _SCL) /3/

Ennen oman ohjelman suunnittelua kannattaa laitteiston toiminta testata valmiilla ohjelmalla. Lisäksi valmiin ohjelman lähdekoodista saa hyviä esimerkkejä omaa ohjelmaa varten. Tällaisiksi esimerkiksi soveltuvat osoitteesta <http://lm-sensors.org/wiki/I2CTools> löytyvät I2C-työkalut. Ladatusta paketista löytyy lähdekoodit kaikille mukana tuleville ohjelmilla sekä Makefile-tiedosto, jolla niiden kääntäminen onnistuu.

LM-Sensors-ohjelmat toimivat monien Linux-ohjelmien tapaan erilaisilla alustoilla. Paketin mukana tulevan Makefile-tiedoston asetukset on kuitenkin tehty normaalille PC-tietokoneelle kääntämistä varten, joten sitä täytyy muokata jotta käännettävät ohjelmat toimisivat AVR32 Linuxissa. Muutos on pieni ja helppo: avataan Makefile-tiedosto tekstieditoriin, ja korvataan rivi `CC := gcc` rivillä `CC := avr32-linux-gcc`. Nyt Makefile käyttää ohjelmien kääntämiseen aiemmin asennettua AVR32 Linux-kääntäjää. Käännetään ohjelmat komennolla `make`.

Käännöksen jälkeen `tools`-hakemistosta löytyy muun muassa ohjelma nimeltä

i2cdetect. Tällä ohjelmalla voidaan selvittää i2c-väylään liitettyjen laitteiden osoitteet sekä tutkia mitä ominaisuuksia itse väylä tukee. Seuraavaksi kopioidaan ohjelma muistikortille ja ajetaan se komennolla `./i2cdetect 0`. Nollaa käytetään, koska `/dev-`hakemistossa i2c-laitetiedosto on järjestysnumeroltaan myös nolla, eli `i2c-0`.

Jos väylään ei ole kytketty laitteita, kaikki osoitteet välillä 03..77 näyttävät pelkkää viivaa. Kompassin ollessa kytkettynä nähdään laite osoitteessa 60, kuten kuvassa 16. Vaikka kompassin tiedoissa sen osoitteeksi kerrotaan C0, on 60 oikein, koska Linuxin käyttämät laiteajurit muuntavat osoitteen muotoon, jossa osoitteen bittejä siirretään yhden kerran oikealle. Tällä tavalla laiteajurit huolehtivat I2C-luku- tai –kirjoitusbitin valinnasta käytetyn funktion mukaan, eikä sitä tarvitse ottaa huomioon osoitettaessa laitteen osoitetta omassa ohjelmassa.

C0 = 1100 0000

60 = 0110 0000

```

/softa2 # ./i2cdetect 0
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-0.
I will probe address range 0x03-0x77.
Continue? [Y/n] Y
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
10:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
20:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
30:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
40:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
50:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
60: 60  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
70:  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
/softa2 #

```

Kuva 16. *i2cdetect*-ohjelman käyttö

Jos kompassin tai muun käytetyn laitteen osoite näkyy tulosteessa oikein, voidaan todeta että laitteisto toimii ja siirtyä kehittämään omaa ohjelmistoa.

Kirjoitettavan ohjelman toimintaa ei alun perin määritelty kovin tarkasti, ainoat vaatimukset olivat, että yhteys kompassin ja NGW:n välillä toimii ja kompassilta saadaan suuntatieto I2C-väylää pitkin ohjelmalle. Näistä lähtökohdista johtuen itse ohjelma muodostui melko yksinkertaiseksi, eikä varsinaista koodia tarvinnut kirjoittaa paljon. Kuten aiemmin on todettu, oheislaitteita käsitellään Linuxissa lai-

tiedostojen kautta ja, näin ollen myös I2C-väylän käyttö on pohjimmiltaan vain tiedoston käsittelyä, kunhan oikeat kirjastot ja laiteajurit on asennettu. Seuraavassa kappaleessa käydään tarvittavan koodin rakenne läpi kohta kohdalta.

Aluksi määritellään tarvittavat otsikkotiedostot. Kolme ensimmäistä ovat Linuxin ja C-kielen standarditiedostoja, ainut erikoisuus on *i2c-dev.h*, joka otetaan mukaan *lm-sensors*-ohjelmapaketista.

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include "i2c-dev.h"
```

Seuraavaksi aloitetaan *main*-funktio, joka onkin ohjelman ainut. Alustetaan muuttajat tiedoston avaukselle ja nimelle sekä I2C-laiteajurille, joka on tässä tapauksessa nolla. Näiden jälkeen kirjoitetaan tiedostonimi luotuun *filename*-taulukkoon *sprintf*-funktioilla ja avataan tiedosto luku- ja kirjoitustilaan *open*-funktioilla.

```
int main (void) {
    int file;
    char filename[20];
    int adapter_nr=0;

    sprintf(filename, "/dev/i2c-%d", adapter_nr);

    if((file=open(filename, O_RDWR))<0) {
        printf("\nVirhe laitetiedoston avaamisessa\n");
        exit(1);
    }
}
```

Viimeisessä *if*-lausekkeessa tarkastetaan tiedoston avaamisen onnistuminen tutkimalla *open*-funktion palauttamaa arvoa, joka on avauksen onnistuessa nolla. Jos avaus ei onnistu, poistutaan ohjelmasta.

Nyt I2C-laitetiedosto on valmiina ohjelman käytettäväksi. Seuraavassa vaiheessa määritetään käytettävän laitteen osoite.

```

int addr = 0x60;

if(ioctl(file,I2C_SLAVE,addr)<0) {
    printf("\nVirhe laitteen osoitteen valinnassa\n");
    exit(1);
}

```

Ioctl-funktio varaa valitun osoitteen käytössä olevan laitetiedoston käyttöön, jolloin mahdolliset muut I2C-isäntälaitteet eivät pääse sekoittamaan väylän liikennettä. Myös tämä funktio palauttaa toimiessaan nollan.

Kompassin rekisterin lukeminen toimii I2C-väylälle tyypilliseen tapaan: isäntälaitte eli NGW kirjoittaa väylälle halutun rekisterin numeron ja orjalaite eli kompassi, vastaa lähettämällä kyseisen rekisterin sisällön.

```

char buf[10];

buf[0]=0x01;

if(write(file,buf,1) !=1) {
    printf("\nVirhe kirjoituksessa\n");
    exit(1);
}

if(read(file,buf,1) !=1) {
    printf("\nVirhe lukemisessa\n");
    exit(1);
}
else {
    printf("\nKompassilta luettu arvo on %d\n",buf[0]);
}

close(file);
exit(0);
}

```

Aluksi luodaan taulukko jonka ensimmäiseen alkioon syötetään luettavan rekisterin numero. CMPS03-kompassilla rekisterissä 1 on suuntatieto 8-bittisenä arvona, joka on tällä kertaa riittävän tarkka (0-255). *Write*-funktioilla kirjoitetaan tiedostoon *file*, eli I2C-laitetiedostoon yksi tavu taulukon *buf*-alusta, eli rekisteri 1. Onnistuessaan funktio palauttaa kirjoitettujen tavujen määrän, tässä tapauksessa 1. Seuraavaksi luetaan *read*-funktioilla samasta tiedostosta saman taulukon ensimmäiseen alkioon palautuva luku, eli kompassin suuntatieto. Lopuksi tulostetaan arvo konsoliin.

Ohjelma on näin yksinkertainen, tarvitaan vain yksi funktio joka kirjoittaa I2C-

väylälle luettavan rekisterin numeron ja lukee sieltä kyseisen rekisterin arvon. Todellisuudessa varsinaisesta väylän käytöstä huolehtii käyttöjärjestelmä laitetiedoston kautta. Käyttäjälle jää ainoastaan tiedoston käsittely. Ohjelmasta on jatkossa helppo luoda vaikkapa funktio, jota käyttämällä laajempi järjestelmä, esimerkiksi navigointitietoa tarvitseva robotti voi ohjata itseään oikeaan suuntaan.

5 YHTEENVETO

Työn tarkoituksena oli tutustua sulautetun elektroniikan ohjelmistokehitykseen Linuxilla. Atmelin Network Gateway Kit AT32AP7000-prosessorilla varustettuna osoittautui tähän tarkoitukseen erittäin sopivaksi. Monipuoliset liitännät, tehokas prosessori ja edullinen hinta ovat kortin suurimpia etuja. Laite tuntui myös toimivan luotettavasti, esimerkiksi noin vuoden yhtämittäinen toiminta reitittimenä työpöytätietokoneen ja ADSL-modeemin välillä ei katkennut kertaakaan laitteesta johtuen. Ainut puutteeksi mahdollisesti luettava ominaisuus laitteessa on oman näytön puuttuminen, mutta sekään ei muodostunut suureksi ongelmaksi sarjaporttityhetyden toimiessa luotettavasti.

Omassa työssäni kortin mahdollistamista ominaisuuksista jäi käyttämättä suurin osa, muun muassa verkkoyhteyksiä ei hyödynnetty mitenkään. Yksi esimerkki käyttötavasta voisi olla kodin lämpötilaa, kosteutta ynnä muuta mittaava laite, joka siirtäisi nämä tiedot omalle verkkopalvelimelleen.

Tiedonhaun merkitystä ei tässä työssä voi väheksyä, sillä siihen kului huomattavan paljon aikaa. Linuxia ennalta tuntematon voi joutua hakemaan yksinkertaisiinkin ongelmiin ratkaisuja kirjallisuuden tai internetin kautta. Silloin on hyvä olla tiedossa luotettavia lähteitä joita käyttää. Tässä työssä on tuotu esiin joitain vaihtoehtoja tiedonhauulle sekä kirjallisuudesta että internetistä.

Atmelin kotisivuillaan tarjoama dokumentaatio AVR32-laitteille ei osoittautunut kovin luotettavaksi. Monet dokumentit sisältävät vanhentuneita tietoja ja esimerkiksi NGW:n dokumentaatio kehoitetaan hakemaan *avrfreaks.net/wiki*-sivustolta. Tuolla sivustolla kuitenkin heti alussa ilmoitetaan että löytyvä dokumentaatio ei ole virallista, joten ilmeisesti virallista dokumentaatiota esimerkiksi NGW:lle ei ole saatavilla. Virallisuus ei kuitenkaan käytännössä merkitse paljon ja *avrfreaks.net/wiki*-sivustolta löytyy paljon hyvää tietoa. Ongelmia aiheutti joissain tapauksissa saatavilla olevan tiedon vanhentuneisuus sekä erilaiset epämääräiset esitystavat joilla tietoa oli

tarjolla.

Varsinaista ohjelmointia työhön kuului loppujen lopuksi yllättävän vähän ja suuri osa työhön käytetystä ajasta kuluikin erilaisiin käyttöjärjestelmää ja ohjelmointityökaluja valmistaviin toimenpiteisiin. Linuxin tapa käyttää oheislaitteita laitetiedostojen kautta oli merkittävä asia ja Buildroot-ohjelmistoon tutustuminen on varmasti hyödyllistä kenelle tahansa sulautetun Linuxin opiskelijalle. Valmiita funktioita Linuxille löytyy lähes joka tarpeeseen, joten ohjelmoinnissa olennaiseksi osoittautui kyky lukea ja ymmärtää toisten kirjoittamaa koodia. Tämä osoittaa myös koodin perinpohjaisen kommentoinnin hyödyn.

Linuxin käyttö sulautetussa elektroniikassa tuntuu erittäin hyödylliseltä. Prosessorien ja muistien hintojen laskiessa ohjelmistokehitys vie yhä suuremman osan resursseista uusia laitteita suunniteltaessa, jolloin avoimen ja siirrettävän käyttöjärjestelmän ja lähdekoodin käyttö perustehtävien hoitamiseen järjestelmässä on hyvä tapa vähentää ohjelmoinnin työvaiheita. Kokonaisuudessaan työ onnistui omasta mielestäni kohtuullisesti, joskin kokemattomuus Linuxin käytössä aiheutti vaikeuksia joissain työn vaiheissa. Oppimiskokemuksena työ toimi erittäin hyvin ja uskon, että monet sulautetun elektroniikan opiskelijat kokevat tästä työstä saadut kokemukset hyödyllisiksi sulautettuun Linuxiin tutustuessaan.

LÄHTEET

- /3/ Atmel Corporation 2009, ATNGW100 Network Gateway Kit, [online]. [viitattu 11.10.2009].
<URL: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4102>
- /4/ Atmel Corporation 2009, AVR32 32-bit MCU - AP7 Application Processors, [online]. [viitattu 11.10.2009].
<URL: <http://www.atmel.com/products/AVR32/ap7.asp>>
- /6/ Atmel Corporation 2009, AT32AP7000 Preliminary summary, [online]. [viitattu 11.10.2009].
<URL: http://atmel.com/dyn/resources/prod_documents/32003S.pdf>
- /19/ Atmel Norway AS 2009, Buildroot for AVR32, [online]. [viitattu 17.10.2009].
<URL: <http://www.atmel.no/buildroot/>>
- /20/ Atmel Norway AS 2009, Buildroot for AVR32 – Application notes, [online]. [viitattu 17.10.2009].
<URL: http://www.atmel.com/dyn/resources/prod_documents/doc32062.pdf>
- /18/ AVR32 Linux 2009, [online]. [viitattu 17.10.2009].
<URL: <http://test.avr32linux.org>> tai <URL: <http://avr32linux.org>>
- /5/ AVR Freaks wiki 2009, NGW User Guide – Hardware reference [online]. [viitattu 11.10.2009].
<URL: http://www.avrfreaks.net/wiki/index.php/Documentation:NGW/NGW100Hardware_reference>
- /15/ AVR Freaks wiki 2009, [online]. [viitattu 17.10.2009].
<URL: <http://avrfreaks.net/wiki>>
- /17/ Codesourcery LLC 2001, Advanced Linux Programming, [online]. [viitattu 17.10.2009].
<URL: <http://www.advancedlinuxprogramming.com/downloads.html>>
- /8/ DENX Software Engineering 2009, The Universal Boot Loader, [online]. [viitattu 11.10.2009].
<URL: <http://www.denx.de/wiki/U-Bootdoc/Presentation>>
- /11/ Devantech Ltd (Robot Electronics) 2009, CMPS03 - Compass Module, [online]. [viitattu 11.10.2009].
<URL: <http://www.robot-electronics.co.uk/htm/cmeps3tech.htm>>
- /9/ Haikala, Ilkka-Järvinen, Hannu-Matti 2003. Käyttöjärjestelmät. Jyväskylä. Talentum.

- /12/ Kapanen, Timo 2004. Linux-koulutuspaketti. 1 p. Helsinki. IT Press.
- /22/ Koskinen, Jari 1998. Mikrotietokonetekniikka. 1 p. Helsinki. Otava.
- /1/ Kuutti, Wille 2008. Näkymätön Linux. Tietokone. Nro 1, 38-41
- /14/ Linux.fi wiki 2009, [online]. [viitattu 17.10.2009].
<URL: <http://fi.wikipedia.org/wiki/Wikipedia:Etusivu>>
- /7/ Linux.fi wiki 2009, Käynnistyslatain, [online]. [viitattu 11.10.2009].
<URL: <http://linux.fi/wiki/K%C3%A4ynnistyslatain>>
- /21/ NXP Semiconductors 2009, The I2C-Bus Specification, [online]. [viitattu 21.10.2009].
<URL: http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf>
- /16/ Silander, Simo 2002. Ohjelmointi Pro Training. 4 p. Jyväskylä. Satku.
- /2/ Tietopetri Oy 2009, Atmel-tuotteet, [online]. [viitattu 11.10.2009].
<URL:<http://www.tietopetri.fi>>
- /10/ Vlasenko, Denys 2009. BusyBox: The Swiss Army Knife of Embedded Linux, [online]. [viitattu 11.10.2009].
<URL: <http://www.busybox.net/about.html>>
- /13/ Wikipedia (Suomi) 2009, Tietoja, [online]. [viitattu 17.10.2009].
<URL: <http://fi.wikipedia.org/wiki/Wikipedia:Tietoja>>

WikiVAMKiin kirjoitettu ohjeistus

Uusi käyttöjärjestelmä NGW:lle SD-kortilla

Asenna Buildroot ja hae NGW:on oletusasetukset edellisen ohjeen mukaan.

Muuta Buildrootin menuconfig-valikossa tiedostojärjestelmäksi ext2, koska se sopii parhaiten muistikorttikäyttöön:

```
/buildroot-avr32-v2.3.0$ make menuconfig
```

```
Target filesystem options => ext2 root filesystem
```

Käännä uudestaan make-komennolla. Nyt */buildroot-avr32-v2.3.0/binaries/atngw100*-hakemistosta löytyy levykuva *rootfs.avr32.ext2*

Liitä levykuva esimerkiksi työpöydälle väliaikaishakemistoon:

```
/buildroot-avr32-v2.3.0/binaries/atngw100$ sudo mount -o loop rootfs.avr32.ext2 ~/Desktop/tmp
```

Nyt hakemistossa työpöydällä pitäisi olla valmis Linux-järjestelmä.

Aseta väh. 256 Mt muistikortti muistikortin lukijaan, ja alusta se ext2-tiedostojärjestelmälle (oletetaan kortin sijainniksi */dev/sdb1*, tarkista *df*-komennolla):

```
~/ $ sudo umount /dev/sdb1
~/ $ sudo mkfs -t ext2 /dev/sdb1
```

Kopioi käyttöjärjestelmän tiedostot väliaikaishakemistosta muistikortille:

```
~/Desktop/tmp$ sudo cp -r -f ~/Desktop/tmp/* /media/disk-1
```

Nyt muistikortilla on valmis käyttöjärjestelmä NGW:tä varten. Vielä pitää muuttaa U-Boot-käynnistyslatainta NGW:llä niin, että laite myös lataa käyttöjärjestelmän muistikortilta.

U-Boot-käynnistyslataimen asetukset SD-kortilta käynnistämistä varten

Käynnistyslataimen konfigurointia varten tietokone pitää kytkeä NGW:n sarjaporttiin terminaaliohjelman yhteysasetuksilla 115200bps, 8 databittiä, ei pariteettia, 1 stopbitti.

Käynnistä NGW, ja paina välilyöntiä kun konsoliin tulee ilmoitus *Press spacebar to abort autoboot in 1 seconds*. Näin pääset U-Bootin valikkoon.

Komennolla *printenv* saat näytölle käytössä olevat asetukset, jotka kannattaa kirjoittaa muistiin, jos et ole varma konfiguroinnin onnistumisesta.

Muistikortilta käynnistämistä varten U-Bootille annetaan seuraavat käskyt ja parametrit:

```
Uboot> askenv bootcmd
Please enter 'bootcmd':mmcinit; ext2load mmc 0:1 0x10400000 /boot/uImage; bootm
Uboot> set bootargs 'console=ttyS0 root=/dev/mmcblk0p1 rootwait
Uboot> saveenv
Uboot> boot
```

Nyt U-Boot käynnistyessään lataa ext2-tiedostomuodossa olevan käyttöjärjestelmän muistikortilta.

U-Boot-käynnistyslataimen asetukset NGW:n FLASH-muistilta käynnistämistä varten

Näillä asetuksilla voit palauttaa NGW:n käynnistämään kortin FLASH-muistiin asennetun käyttöjärjestelmän:

```
Uboot> askenv bootcmd
Please enter 'bootcmd':fsload;bootm
Uboot> set bootargs 'console=ttyS0 root=/dev/mtdblock1 rootfstype=jffs2'
Uboot> saveenv
Uboot> boot
```

I2C-väylän käyttö NGW:llä

Ladattaessa Buildrootiin NGW:n oletusasetukset, myös I2C-tuki on valittuna. Jos näin ei ole, voidaan asia tarkistaa Buildrootin valikosta:

```
/buildroot-avr32-2.3.0$ make linux26-menuconfig
```

Varmista että seuraavat kohdat on valittu asennettavaksi:

```
Device drivers => *I2C support => *I2C device interface
I2C Hardware Bus support => *GPIO-based bitbanging I2C
```

Jos jouduit tekemään muutoksia, käännä järjestelmä uudestaan, ja käynnistä NGW. /dev-hakemistosta pitäisi nyt löytyä I2C-laitetiedosto i2c-0

Hyvät työkalut I2C-väylän testaukseen ja käyttöön löytyvät osoitteesta <http://www.lm-sensors.org/wiki/I2CTools>

Pura paketti ja käännä työkalut make-komennolla. Huomaa muokata Makefile-tiedostoon gcc-kääntäjäksi avr32-linux-gcc:

```
CC := avr32-linux-gcc
```

Tools-hakemistosta löytyvällä i2c_detect-ohjelmalla on kätevä testata väylään liitettyjen laitteiden löytyminen. Kytke esimerkiksi elektroninen kompassi CMPS03 NGW:n I2C-pinneihin 9 (data) ja 10 (kello). Siirrä ohjelma NGW:lle muistikortilla tai Telnetillä, ja aja se konsolista komennolla `./i2c_detect 0`

Koska Linux käsittelee I2C-osoitteita ilman luku- tai kirjoitustilaa merkitsevää viimeistä LSB-bittiä, näkyy kompassin osoite C0 muodossa 60:

```
C0 = 1100 0000 >> 0110 0000 = 60
```

Jos i2c_detect löytää kompassin väylältä, on väylä todennäköisesti kunnossa ja kompassia voidaan käyttää omasta ohjelmasta.

Alla esimerkki yksinkertaisesta ohjelmasta, joka lukee elektroniselta kompassilta rekisteristä 1 suuntatiedon 8-bittisenä arvona:

```
#include <stdlib.h>
#include <fcntl.h>

//otsikkotiedosto lm-sensors.org I2C-tools paketista
#include "i2c-dev.h"

int main (void) {
    int file;
    char filename/20/;
    int adapter_nr=0;

//Avataan laitetiedosto i2c-0 ohjelman käyttöön

    sprintf(filename,"/dev/i2c-%d",adapter_nr");

    if((file=open(filename, O_RDWR))<0) {
        printf("\nVirhe laitetiedoston avaamisessa\n");
        exit(1);
    }

//Määritellään laitteen osoite (CMPS03-kompassi)

    int addr = 0x60;

    if(ioctl(file,I2C_SLAVE,addr)<0) {
        printf("\nVirhe laitteen osoitteen valinnassa\n");
        exit(1);
    }

    char buf/10/;
    buf[0]=0x01;

//Kirjoitetaan väylälle kompassilta luettavan rekisterin numero (1)

    if(write(file,buf,1) !=1) {
        printf("\nVirhe kirjoituksessa\n");
        exit(1);
    }

//Luetaan väylältä rekisterin arvo, ja tulostetaan se konsoliin

    if(read(file,buf,1) !=1) {
        printf("\nVirhe lukemisessa\n");
        exit(1);
    }
    else {
        printf("\nKompassilta luettu arvo on %d\n",buf[0]);
    }

    close(file);
    exit(0);
}
```