

Marko Huhta

**Radiopiirin ohjelmointiympäristön kehitys ja  
langattoman verkon tahdistus**

Loppuraportti

Syksy 2009

Tekniikan yksikkö

Tietotekniikan koulutusohjelma

Sulautetut järjestelmät



## SEINÄJOEN AMMATTIKORKEAKOULU

### Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietotekniikan koulutusohjelma

Suuntautumisvaihtoehto: Sulautettujen järjestelmien suuntautumisvaihtoehto

Tekijä: Marko Huhta

Työn nimi: Radiopiirin ohjelmointiympäristön kehitys ja langattoman verkon tahdistus

Ohjaaja: Heikki Palomäki

Vuosi: 2009

Sivumäärä: 64

Liitteiden lukumäärä: 4

---

Tässä työssä tutkitaan uuden nRF24LE1-radiopiirin rakennetta ja ohjelmointia, sekä kehitetään radiopiirin sisäiselle mikrokontrollerille oma ohjelmointiympäristö käyttäen apuna Nordic Semiconductorin valmistamaa aloituspakettia. Työssä on myös tarkoitus kehittää pienivirtainen verkko ja tutkia sen hyötyä virransäästön parantamiseksi. Radiopiirille suunnitellaan oma pieni piirilevy, jota voidaan hyödyntää tulevaisuudessa sekä opetuksissa että projekteissa.

Mikrokontrollerin ohjelmointi on toteutettu SPI-väylän avulla käyttäen muuntimena Atmelin USB-kontrolleria, joka muuntaa tietokoneelta tulevan tiedon SPI-väyläiseen muotoon. Tietokoneella C-ohjelmakoodin kääntäminen on toteutettu SDCC-ohjelmalla ja ohjelman lataus piirin muistille tapahtuu omalla ohjelmalla, joka on ohjelmoitu C#-ohjelmointikielellä. Työssä on onnistuttu uuden radiopiirin ohjelmoinnissa ja käytössä, sekä langaton yhteys on saatu muodostettua.

Pienivirtaisen verkon toteutuksessa on hyödynnetty laitteiden samanaikaista toimintaa, eli tahdistamista. Tämä toiminto mahdollistaa pienivirtaisen verkon toiminnan siten että tiedonsiirtonopeus on hyvin säädeltävissä ja verkko on samalla hyvin muokattavissa erilaisiin sovelluksiin. Tuloksena on saatu erittäin pienivirtainen verkko, jonka käyttömahdollisuudet ovat laajat.

Avainsanat: nRF24LE1, Nordic Semiconductor, 8051, Atmel, Tahdistus

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## Thesis abstract

Faculty: School of Technology  
Degree programme: Information Technology  
Specialisation: Embedded Systems

Author: Marko Huhta

Title of thesis: Developing the programming environment of a radio chip and the synchronizing of a wireless network

Supervisor: Heikki Palomäki

Year: 2009

Number of pages: 64

Number of appendices: 4

---

This thesis examines the new nRF24LE1-radio chip architecture and programming, and also develops the programming environment of the radio chip's microcontroller using a Nordic Semiconductor's starter kit. The work is also intended to develop a low-power network and to explore its possibilities to save power. A small circuit board has been designed for the radio chip and it can be exploited in teaching and projects in the future.

The programming of the microcontroller is carried out through the SPI bus using Atmel's USB controller, which converts information from a computer to an SPI bus-format. The computer C-code compiling is made by an SDCC program and the compiled C-code is downloaded to a processor using a new LE1-programming software which has been programmed in a C# programming language. The programming of the new radio chip is working successfully and the wireless connection has been established.

The implementation of the low-power network has been made using time synchronization. This feature allows low-power operation and the data transfer rate is also well regulated. The network is also very customizable for various applications. The result is a network with a very low power consumption and wide usability.

Keywords: nRF24LE1, Nordic Semiconductor, 8051, Atmel, synchronizing

## SISÄLTÖ

OPINNÄYTETYÖN TIIVISTELMÄ .....	2
THESIS ABSTRACT .....	3
KUVIO- JA TAULUKKOLUETTELO .....	6
KÄYTETYT TERMIT JA LYHENTEET .....	8
1 JOHDANTO .....	10
1.1 Taustaa .....	10
1.2 Työn tavoitteet.....	11
1.3 Työn rakenne .....	12
1.4 GENSEN-hanke.....	12
2 NRF24LE1-RADIOPIIRI .....	13
2.1 Yleistä .....	13
2.2 Lähetinvastaanotin .....	13
2.3 Mikrokontrolleri.....	14
2.4 Muisti.....	15
2.4.1 Ohjelmamuisti .....	16
2.4.2 Työmuisti.....	16
2.4.3 Muut muistit.....	16
2.5 Liitynnät.....	17
3 OHJELMOINTIYMPÄRISTÖ .....	19
3.1 Rajapinnat.....	19
3.1.1 SPI-väylä.....	19
3.1.2 USB-väylä .....	20
3.1.3 nRF24LE1 ohjelmointiväylä .....	21
3.2 Ohjelmointi .....	22
3.3 SDCC-kääntäjä .....	23
3.4 Intel HEX -tiedosto .....	23
3.5 Ohjelman lataaminen .....	24
3.5.1 LE1-programmer.....	25
3.5.2 InfoPage-asetukset.....	26
3.6 USB-tikku .....	28

3.6.1	Toiminta .....	28
3.6.2	Muistin luku ja vianetsintä .....	30
<b>4</b>	<b>PIIRILEVYN SUUNNITTELU .....</b>	<b>31</b>
4.1	Piirilevy .....	31
4.2	Ominaisuudet .....	32
4.3	Käyttömahdollisuudet .....	32
<b>5</b>	<b>VERKON TAHDISTUS .....</b>	<b>34</b>
5.1	Ongelma .....	34
5.2	Tahdistuksen käyttö GENSEN-projektissa .....	35
5.3	Tahdistuksen teoria .....	36
5.3.1	Perusidea .....	36
5.3.2	Laajennettu verkko .....	37
5.3.3	Eri tahtien estäminen .....	39
5.3.4	Virrankulutus .....	40
5.4	Toteutus .....	41
5.4.1	Laskuri ja tilakone .....	41
5.4.2	Tahdin laskeminen .....	44
5.4.3	Eri verkkojen etsiminen .....	45
5.4.4	Virransäästö .....	46
5.5	Testaus .....	47
<b>6</b>	<b>TULOKSET JA ARVIOINTI .....</b>	<b>50</b>
6.1	Tulokset .....	50
6.1.1	Ohjelmointiympäristö .....	50
6.1.2	Tahdistus ja virrankulutus .....	50
6.2	Arviointi .....	55
6.2.1	Työ .....	55
6.2.2	Jatkokehitys .....	55
6.2.3	Käyttömahdollisuudet .....	56
	<b>LÄHTEET .....</b>	<b>58</b>

## KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. Kehysrakenteen kuvaus.....	14
Kuvio 2. Lähetinvastaanottimen lohkokaavio .....	14
Kuvio 3. nRF24LE1-radiopiirin mikrokontrollerin lohkokaavio .....	15
Kuvio 4. Tässä työssä käytettävän radiopiirin nastajärjestyskuva.....	18
Kuvio 5. SPI-väylän rinnakkaismuotoinen kytkentä.....	20
Kuvio 6. Työn alussa käytetty USB-ohjelmointilaite, sekä aloituspaketin mukana tullut radiopiiri.....	22
Kuvio 7. Intel HEX -muotoisen tiedoston yhden rivin osat.....	24
Kuvio 8. Rajapintakuvaus .....	24
Kuvio 9. Kuvankaappaus LE1-programmer-ohjelman käytöstä .....	26
Kuvio 10. LE1-programmer-ohjelman vuokaavio .....	27
Kuvio 11. Tilakaavio USB-tikun ohjelmakoodin tilakoneesta.....	29
Kuvio 12. Piirilevy on kytkettynä USB-ohjelmointitikkuun.....	32
Kuvio 13. Piirilevyn pieni koko mahdollistaa asennuksen moniin pieniin kohteisiin .....	33
Kuvio 14. Yksinkertaisen tahdistuksen toimintaperiaate .....	37

Kuvio 15. Laitteiden aktiiviaika on jaettu useampaan eri ikkunaan, jolloin tietoa saadaan siirrettyä kerralla enemmän .....	38
Kuvio 16. Esimerkki tilanteesta, jossa tahdin aikana ovat kahden eri verkon aktiiviajat eri kohdissa .....	40
Kuvio 17. Tilakaavio radiopiirin vertailukeskeytyksen vaiheista .....	43
Kuvio 18. Esimerkkitalanne oikean tahdin löytämisestä.....	45
Kuvio 19. Kuva testauslaitteistosta .....	48
Kuvio 20. Kuvankaappaus logiikka-analysaattorin PC-ohjelmasta.....	49
Kuvio 21. Kuvaaja prosessorin aktiiviajan virrankulutuksesta .....	52
Kuvio 22. Keskimääräisen virrankulutuksen ja akun varauksen suhde jakson pituuteen .....	53
Taulukko 1. nRF24LE1-radiopiirin virrankulutukset eri tiloissa eri toimintojen ollessa päällä. ....	41
Taulukko 2. Tahdistuksen virrankulutuksen laskuesimerkki käytännön tilanteessa. ....	54
Taulukko 3. Tahdistuksen virrankulutuksen laskuesimerkki optimoiduilla tahdistuksen arvoilla. ....	54

## Käytetyt termit ja lyhenteet

<b>RF</b>	Radio Frequency. Radiotaajuus, jota yleisesti käytetään ilmaisemaan langatonta tiedonsiirtoa.
<b>8051</b>	Sulautetuissa järjestelmissä käytetty Intelin mikrokontrollerityyppi.
<b>SoC</b>	System-on-Chip. Komponentti, joka voi sisältää monia eri sovelluksia ja toimintoja.
<b>Kide</b>	Komponentti, jolla saadaan muodostettua kellopulssi erilaisten mikropiirien käyttöön.
<b>JTAG</b>	Joint Test Action Group. Tietoliikenneväylä, jota käytetään erityisesti erilaisten piirien ohjelmointiin ja vianetsintään.
<b>SPI</b>	Serial Peripheral Interface. Neljän signaalin nopea sarjavyylä, jota käytetään laitteiden sisäiseen kommunikointiin.
<b>RS232</b>	Kahden tietokoneen väliseen tiedonsiirtoon käytettävä tietoliikenneväylä.
<b>UART</b>	Universal Asynchronous Receiver Transmitter. Yleinen tiedonsiirtoväylä, jota käytetään paljon yhdessä RS232-väylän kanssa.
<b>2-wire</b>	Kahdella johtimella toimiva sarjavyylä, jonka toisessa johtimessa kulkee kellopulssi ja toisessa tieto.
<b>CRC</b>	Cyclic Redundancy Check. Tehokas virheentarkistusalgoritmi.



<b>SDCC</b>	Small Device C Compiler on erityisesti pienien prosessorien käyttöön suunniteltu C-ohjelmakoodin kääntäjä.
<b>ANSI C</b>	American National Standards Institute C. Ohjelmointikieli.
<b>Intel HEX</b>	Intelin standardoima tiedostoformaatti, joka sisältää ohjelmakoodin lisäksi muuta oheistietoa.
<b>ISM</b>	Industrial, Scientific and Medical. ISM on maailmanlaajuisen radiotaajuuskaista, jonka käyttö ei vaadi erillistä lupaa.
<b>IO-nasta</b>	Input/Output, Mikrokontrollerin nasta, joka voi toimia tulona ja lähtönä.
<b>Full-duplex</b>	Väylän ominaisuus, jolla tieto voi kulkea molempiin suuntiin samanaikaisesti.

(Nordic Semiconductor 2009.)

# 1 JOHDANTO

## 1.1 Taustaa

Tekniikan kehittyessä ja vaatimusten kasvaessa tarvitaan jokaisella alalla koko ajan uutta osaamista ja asioiden kehittämistä. Isona osana on langattomuus, joka luo uusia haasteita monille eri aloille. Laitteistojen langattomuus onkin nykyään suuressa suosiossa, koska sillä saadaan parannettua käyttömukavuutta ja asennusnopeutta. Monet alat, kuten lääketiede, teollisuus, hyvinvointi ja liikunta, ovat soveltaneet langattomuutta omaan käyttöönsä (Basavaraju, Puttamadappa & Sarkar 2008, 297). Langattomalla tekniikalla on tulevaisuudessa varmasti enemmän kysyntää ja sitä kautta myös alan ammattitaidolla.

Seinäjoen ammattikorkeakoulun Tekniikan yksikössä on tutkittu langatonta tekniikka jo usean vuoden ajan. Koulussa on tehty monia opinnäytetöitä liittyen langattomaan tiedonsiirtoon ja langattomien laitteiden rakentamiseen. Myös koululla on huomattu tekniikan nopea kehittyminen, mikä vaatii paljon työtä, jotta pysytään tekniikan kehityksessä mukana. Koulu tarjoaa hyvät mahdollisuudet uusien laitteiden tutkimiseen ja opettelemiseen, niitä on käytetty osaksi myös opetuksessa.

Useimmin Tekniikan yksikössä käytettävät langattomat lähetin vastaanottimet ovat integroituna yhteen piiriin, joka vaatii ulkoisen mikrokontrollerin niitä ohjaamaan. Tällainen tilanne vaatii usein paljon komponentteja ja ison tilan piirilevyiltä. Markkinoilta löytyy kuitenkin liitteessä 1 esitelty SoC-komponentti, johon on integroituna mikrokontrolleri sekä langaton lähetin vastaanotin. Langaton lähetin vastaanotin on myös suuri virrankuluttaja, joka lisää haasteita langattoman tekniikan jokaiselle osa-alueelle. Pienien ja lähes näkymättömien langattomien laitteiden suurin ongelma onkin ollut paristojen suuri koko, joka ei viime vuosina ole kehittynyt yhtä suurella nopeudella kuin muu tekniikka. Tämä lisää pakottavia haasteita ohjel-

moinnin ja muun elektroniikan suunnitteluun, jotta virrankulutus ja paristojen koko saataisiin minimoitua.

## 1.2 Työn tavoitteet

Tässä opinnäytetyössä on tarkoitus tutkia markkinoille juuri tullutta Nordic Semiconductorin valmistamaa radiopiiriä, käyttäen apuna saman valmistajan valmistamaa aloituspakettia. Kyseiselle piirille on ohjelmointiympäristönä ainoastaan valmistajan maksullinen aloituspaketti, joten työssä tavoitteena on myös rakentaa oma ohjelmointiympäristö mahdollisesti ilmaisista ohjelmista ja helposti rakennettavasta elektroniikasta. Työ on suunniteltava niin, että sitä voitaisiin tulevaisuudessa käyttää myös opetuksessa, joten ohjelmistojen ja laitteistojen tulisi olla soveltuvia koulun opetuslaitteistolle.

Työssä on tarkoitus myös suunnitella oma piirilevy. Seinäjoen ammattikorkeakoulun Tekniikan yksikössä on hyvät mahdollisuudet kehittää mahdollisimman pientä elektroniikkaa, joten tämänkin työn piirilevyn tulisi olla mahdollisimman pienikokoinen ja monikäyttöinen. Monikäyttöisenä nRF24LE1-radiopiiri olisi hyödynnettävissä sekä opetuksessa että mahdollisena tuotteena tulevaisuuden projekteissa. Ohjelmointiympäristö on testattava niin, että ohjelman kääntäminen ja lataaminen piiriin mikrokontrollerille onnistuisi ongelmitta. Myös langattoman radioliikenteen vaatimat ohjelmakirjastot täytyy ohjelmoida ja testata, että langaton yhteys voidaan muodostaa.

Uuden radiopiirin mahdollisuudet pienivirtaiseen verkkoon on tutkittava ja toteutettava siten, että se olisi vertailukelpoinen jo markkinoilla oleviin vastaaviin laitteisiin. Pienivirtainen verkko on tarkoitus toteuttaa käyttäen apuna laitteiden samanaikais- ta toimintaa, eli niin sanottua tahdistusta. Tahdistuksen avulla laitteet on saatava pienivirtaiseen tilaan ja vastaavasti myös isovirtaiseen lähetys- ja vastaanottotilaan samanaikaisesti. Tällä saataisiin laitteiden virrankulutusaika mahdollisimman pieneksi, mutta kuitenkin niin että tiedonsiirto jatkuisi. Tahdistus on testattava niin, että uusien laitteiden lisääminen ja poisto eivät sekoita tahdistusta, ja että yksit-

täisten laitteiden vioittuminen ei heikennä verkon toimintaa. Verkosta on saatava itsensä korjaava.

### **1.3 Työn rakenne**

Työ aloitetaan tutkimalla Nordic Semiconductorin valmistamaa nRF24LE1-radiopiiriä ja sen ominaisuuksia. Ominaisuuksista käydään läpi ne osa-alueet, jotka ovat tärkeitä piirin ohjelmoinnin ja tahdistuksen toteuttamiselle. Lisäksi käydään läpi olennaisia osia ohjelmointiympäristön eri komponenteista ja toteutuksista.

Toiseksi tässä työssä tutkitaan radiopiirin mikrokontrollerin ohjelmakoodin kirjoittamiseen ja koodin kääntämiseen liittyviä asioita ja ratkaisuja. Esitellään myös oman ohjelmointi-ohjelman rakenne ja toiminta.

Pienivirtaisen verkon tarkoitusta ja mahdollisuuksia tutkitaan itse suunnitellulla piirilevyllä. Tutkitaan verkon toteuttamista ja toteutetaan verkko käyttäen apuna tahdistusta.

Lopuksi kerrotaan työn lopputulokset ja arvioidaan työ kokonaisuutena. Lisäksi kerrotaan jatkokehitysmahdollisuuksista ja käyttökohteista.

### **1.4 GENSEN-hanke**

Generic Sensor Network Architecture for Wireless Automation on hanke, jossa tarkoituksena on kehittää hyvin monikäyttöisiä sovelluksia anturiverkkotekniikalla. Kehiteltävien sovelluksien on oltava helposti muunneltavissa moniin eri laitteisiin ja verkkoihin. Hanke on jaettu moniin pienempiin osa-alueisiin, joista yksi on Seinäjoen ammattikoreakoululla kehitettävä pienivirtainen langaton anturiverkko. Hankkeessa on mukana Seinäjoen ammattikorkeakoulun lisäksi kaksi yliopistoa, sekä myös yrityksiä pohjanmaan alueelta. (GENSEN 2009, 1.) Myös tämä työ on tehty GENSEN-hankkeelle.

## 2 NRF24LE1-RADIOPIIRI

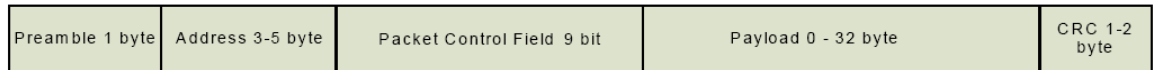
### 2.1 Yleistä

nRF24LE1-radiopiiri on Nordic Semiconductorin valmistama SoC-komponentti, joka sisältää 8051-arkkitehtuurisen mikrokontrollerin ja lisäksi 2,4 GHz:n taajuudella toimivan radio-lähetinvastaanottimen. Toimintojen yhdistämisellä saavutetaan monia etuja tekniikan suunnittelussa ja toteutuksessa. Erityisesti virrankulutus ja virransäästötilat saadaan minimoitua tässä radiopiirissä. Ulkoisten komponenttien määrä vähenee, jonka seurauksena piirilevystäkin saadaan kompaktimpi. Piiristä on saatavilla kolmea eri kokoa, joiden ulkoisen liitynnän määrät vaihtelevat. Tässä työssä on kuitenkin käytetty vain pienintä versiota. Valmistaja ehdottaa piirille useita eri käyttökohteita, kuten tietokoneen oheislaitteet, lelut ja hyvinvointiin liittyvät sovellukset. (Nordic Semiconductor 2009, 10.)

### 2.2 Lähetinvastaanotin

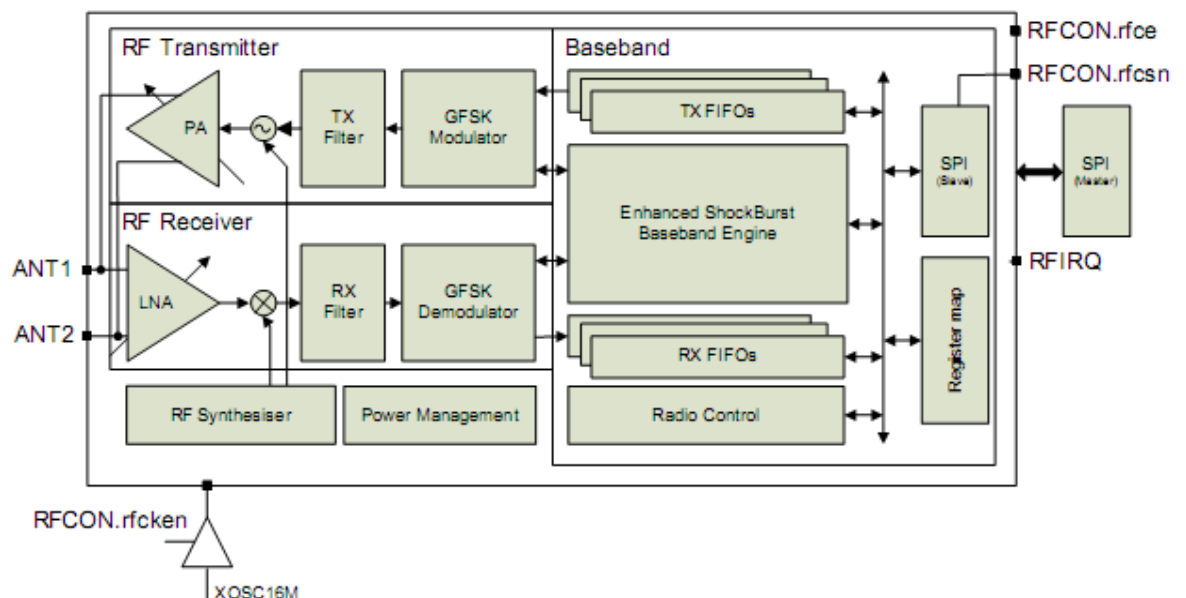
Lähetinvastaanotin on sisäänrakennettuna nRF24LE1-radiopiirille, jonka toiminnot ovat täysin samat kuin saman valmistajan nRF24L01+-versiossa. Lähetinvastaanotin on yhteydessä mikrokontrolleriin sen sisäisen SPI-väylän avulla. Radiopiirissä käytetty lähetinvastaanotin käyttää maailmanlaajuisesti vapaassa käytössä olevaa 2,4 GHz:n ISM-taajuusaluetta, joka on todella hyvin soveltuva pienen virrankulutuksen laitteisiin. Lähetinvastaanotin on säädettävissä nopeuksille 0.25 Mb/s, 1 Mb/s ja 2 Mb/s ja lähetysteho on valittavissa 0, -6, -12 ja -18 dBm vaihtoehtoja. (Nordic Semiconductor 2009, 16.)

Piirissä on sisäisenä kolme 32-tavun puskuria, joihin automaattisesti lisätään kehysrakenne. Kehysrakenne sisältää yhden tavun mittaisen synkronointitiedon, 3 - 5 tavun mittaisen osoitteen, 9 bitin mittaisen ohjauspaketin, 0 - 32 tavua tietoa sekä 1 - 2 tavun CRC-tarkistussumman. Kuviossa 1 kuvataan kehysrakenteen tietojen järjestys. (Nordic Semiconductor 2009, 23.)



Kuvio 1. Kehyrakenteen kuvaus. (Nordic Semiconductor 2009, 23.)

nRF24LE1-radiopiirin lähetinvastaanottimen antennisovitus on toteutettu niin että samalla antennilla voidaan sekä lähettää että vastaanottaa tietoa. (Nordic Semiconductor 2009, 16). Kuvion 2 lohkokaaviosta näkee lähetinvastaanottimen radio- sekä tiedonkäsittelylohkot.

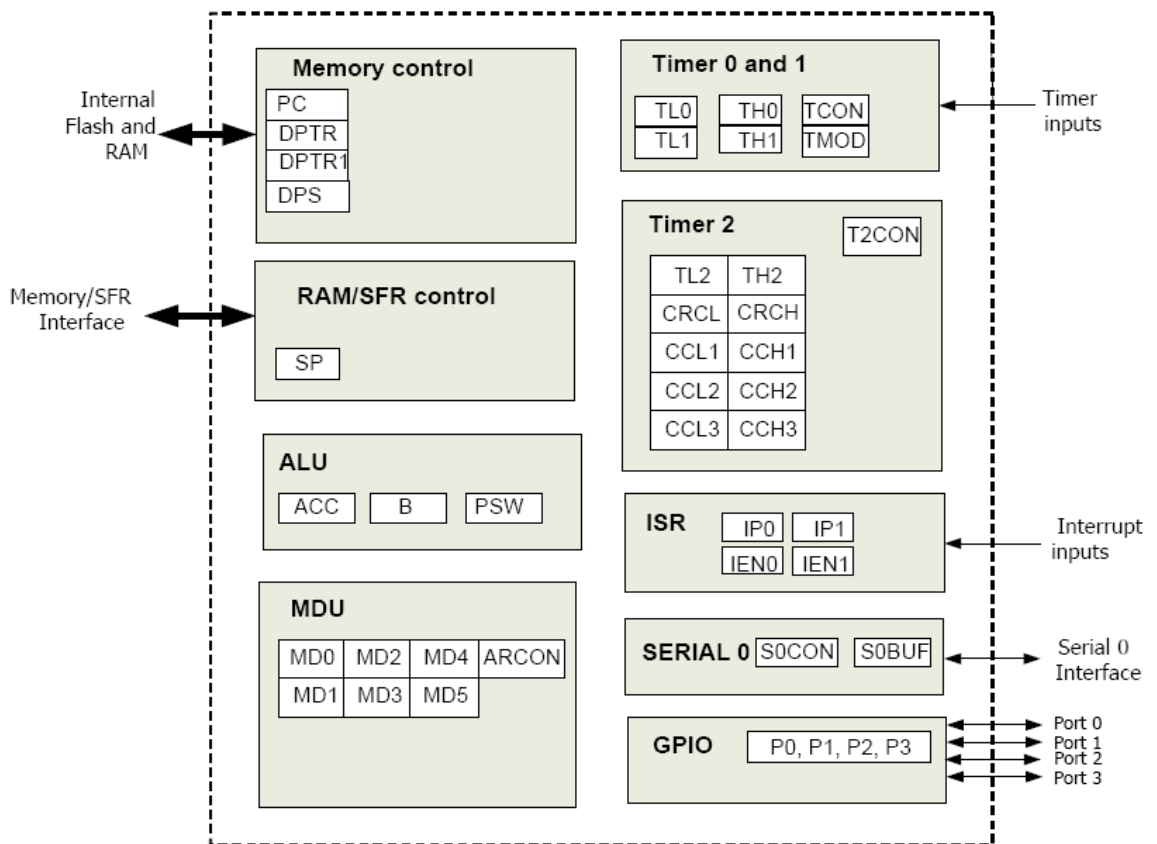


Kuvio 2. Lähetinvastaanottimen lohkokaavio. (Nordic Semiconductor 2009, 17.)

### 2.3 Mikrokontrolleri

nRF24LE1-radiopiiri sisältää 8051-arkkitehtuurisen mikrokontrollerin, joka on 1980-luvulla kehitetty sulautettuihin järjestelmiin. Mikrokontrollerin sisältämä 8051-prosessoriperhe on 1990-luvun lopulla jäänyt kokonaan pois laitteista niiden pienen muistin ja vaikean käytön takia. Nykyään pieni muisti ja vähäiset ominaisuudet riittävät moniin käyttötarkoituksiin, joten monet prosessorinvalmistajat ovat ottaneet sen uudestaan tuotantoon. (Waclawek, 1 - 2)

Kuvion 3 lohkokaaviosta näkee mikrokontrollerin eri osat. Jokaisesta osasta on lisäksi merkittynä niiden rekisterit ja ulkoiset liittynät.



Kuvio 3. nRF24LE1-radiopiirin mikrokontrollerin lohkokaavio (Nordic Semiconductor 2009, 55.)

## 2.4 Muisti

nRF24LE1-radiopiirin mikrokontrollerissa on monia muistialueita eri käyttötarkoituksia varten. Alueista käyttäjälle tärkeimpiä ovat 16 kt:n ohjelmamuisti, 256 tavun sisäinen työmuisti ja 1 kt:n ulkoinen työmuisti. Kontrollerista löytyy lisäksi 1 kt pysyvää ohjelmamuistia sekä 512 tavua pysyvää ohjelmamuistia laajennetulla käyttöiällä. (Nordic Semiconductor 2009, 62.)

### **2.4.1 Ohjelmamuisti**

16 kt:n ohjelmamuisti on alue, jonne käyttäjän ohjelmakoodi tallentuu. Ohjelmamuisti on ns. pysyvää muistia eli muisti ei tyhjene virtakatkoksen jälkeen. Tässä ohjelmamuistissa on lisäksi ominaisuus, jonka avulla käyttäjä pystyy jakamaan muistin kahteen haluamaansa osaan. Käyttäjä voi määritellä nämä ohjelmamuistin alueet suojattuun ja ei-suojattuun alueeseen. Suojatulla muistialueella tarkoitetaan aluetta, jonka muuttamisen voi tehdä ainoastaan ohjelmoinnin yhteydessä. Tämä on hyvä ominaisuus, jos esimerkiksi joitain ohjelman osia halutaan suojata sisäisiltä muutoksilta. Esimerkkinä tilanne, jossa ohjelmakoodi voi ohjelmoida itsensä uudestaan. (Nordic Semiconductor 2009, 73.)

### **2.4.2 Työmuisti**

Mikrokontrolleri sisältää sekä sisäistä, että ulkoista työmuistia. Nämä muistit ovat ns. häviävää muistia eli niiden sisältö tyhjenee virtakatkoksessa. Osa ulkoisesta työmuistista katoaa myös tietyissä virransäästötiloissa. Sisäistä työmuistia käytetään mikrokontrollerin sellaisiin toimintoihin, jotka vaativat erityistä nopeutta. Isommat ohjelmakoodin muuttajat voidaan tallentaa ulkoiseen työmuistiin, jotta sisäiselle muistille jäisi tilaa. (Nordic Semiconductor 2009, 83.)

### **2.4.3 Muut muistit**

Muita muistialueita ovat 1 kt:n pysyvä ohjelmamuisti sekä 512 tavun laajennettu pysyvä ohjelmamuisti. Nämä muistialueisiin voidaan ohjelmakoodin lisäksi tallentaa tietoja, jotka eivät tyhjene virtakatkoksen eivätkä ohjelmoinnin jälkeen. (Nordic Semiconductor 2009, 71.)



## 2.5 Liitynnät

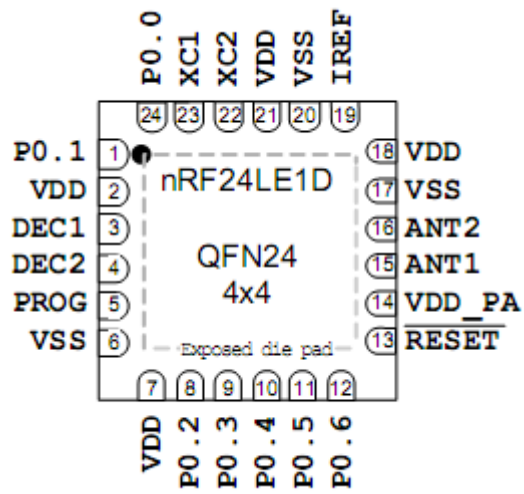
Pienimmässä nRF24LE1-radiopiirin kotelossa on seitsemän nastaa, joilla on monipuoliset käyttövaihtoehdot. Kuviossa 4 näkyville seitsemälle eri P0-portin nastoilta voidaan erikseen ohjelmoida tietty toiminto, joka estää saman nastan muiden toimintojen päällekkäisen käytön. Tämä on hyvä ominaisuus, koska sillä saadaan säästettyä piirin nastojen määrää ja samalla pienennettyä kokoa. Myös mikrokontrollerin sisäiset ohjelmamuistit ohjelmoidaan käyttäen hyväksi näitä nastoja. Kahdessa isommassa koteloversiossa toimintojen määrä pysyy pienimpään verrattuna samana, ainoastaan tavallisten IO-nastojen määrä vaihtelee.

Yleisimpiä nRF24LE1-radiopiirin tukemia ominaisuuksia ovat

- ulkoisen kellokiteen mahdollisuus
- JTAG-väylä
- SPI-väylä
- UART-väylä (sarjaportti)
- 2-wire-väylä
- 12 bittinen AD-muunnin
- tulot ja lähdöt

(Nordic Semiconductor [Viitattu 10.5.2009].)

Liitteessä 1 on listattu tarkemmin laitteen ominaisuudet. Laitteen pienimmässäkin versiossa kaikkia näitä edellä mainittuja ominaisuuksia voidaan käyttää seitsemän nastan avulla.



Kuvio 4. Tässä työssä käytettävän radiopiirin nastajärjestyskuva. (Nordic Semiconductor 2009, 14.)

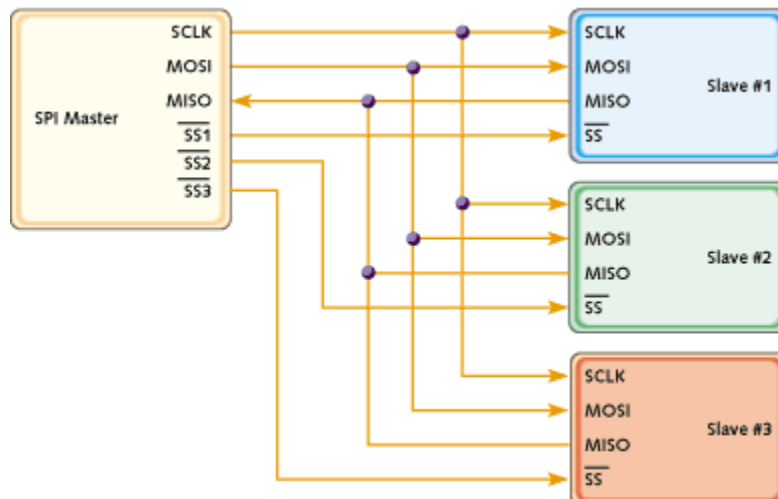
## 3 OHJELMOINTIYMPÄRISTÖ

### 3.1 Rajapinnat

Ohjelmointiympäristön käyttö vaatii monien eri rajapintojen tuntemusta. Tässä työssä käytettävät rajapinnat on esitelty seuraavaksi.

#### 3.1.1 SPI-väylä

SPI-väylä on hyvin yleinen laitteiden sisäinen väylä, jolla ei yleensä ole liityntää laitteistoista ulos. Väylä on isäntä-palvelija-tyyppinen eli väylällä on aina yksi laite, joka hoitaa kommunikoinnin muiden laitteiden kanssa. SPI-väylä on myös todella nopea verrattuna muihin laitteistojen sisäisiin väylätyyppeihin, ja lisäksi se on hyvin räätälöitävissä erilaisiin sovelluksiin. Nykyään monet SPI-väyläiset laitteet tukevat jopa 30 MHz:n nopeuksia. SPI-väylällä ei myöskään ole omaa standardoitua protokollaa, joten joissain tilanteissa hyödyttömät liikenteet saadaan näin vähennettyä ja samalla lisättyä tiedonsiirron nopeutta. SPI-väylä vaatii käyttöjännitteiden lisäksi 4 signaalia, mikä on kaksin verroin enemmän kuin monet muut sarjamuotoiset väylät. 4 signaalihohtimen ansiosta väylä on full-duplex-tyyppinen ja mahdollistaa sarja- ja rinnakkaistyyppiset kytkennät. Kuviossa 5 on esimerkkinä kolmen asiakkaan ja yhden isännän rinnakkaismuotoinen kytkentä (Kalinsky & Kalinsky 2002.)



Kuvio 5. SPI-väylän rinnakkaismuotoinen kytkentä. (Kalinsky & Kalinsky 2002.)

### 3.1.2 USB-väylä

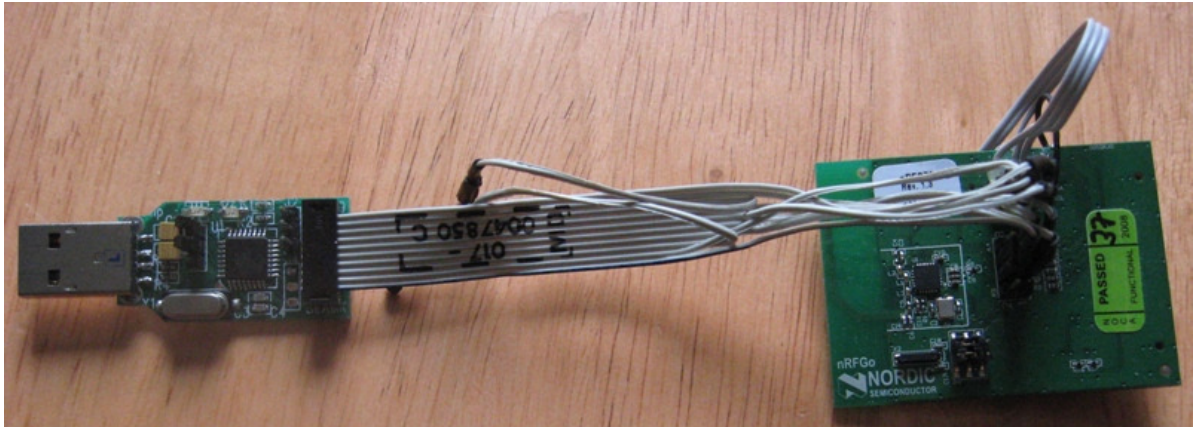
USB-väylä on erittäin suosittu tietokoneen ja oheislaitteiden välinen väylä. USB-väylään voi liittää laitteita tietokoneen ollessa käynnissä, mikä tekeekin väylästä suosittun mm. digitaalisten kameroiden ja muistitikkujen liittämisessä. Väylä on julkaistu vuonna 1995 ja ensimmäisiin tietokoneissa sitä on käytetty vuodesta 1997 lähtien. USB-väylää on jatkuvasti kehitetty nopeammaksi ja luotettavammaksi, joten se on syrjäyttänyt monet muut väyläteknikat. (Tech-FAQ [Viitattu 10.5.2009].)

Tekniikaltaan USB-väylä on isäntäohjaava, eli väylässä voi olla ainoastaan yksi isäntälaitte. Väylässä muut laitteet toimivat palvelijana, jotka toimivat ainoastaan isännän komennoista. Asiakkaiden määrä väylässä on rajoitettu 127 kappaleeseen. Väylä sisältää 4 johdinta, jotka ovat käyttöjännite +5V, maa, datalinja- ja datalinja+. Datalinjoissa tieto siirtyy erojännitteenä eli toisen datalinjan signaali on vastasuuntainen toiseen datalinjaan. Tällä saadaan linjasta hyvin kestävä ulkoisille häiriöille sekä väylän pituutta voidaan kasvattaa. (Pinouts.ru 2009)

### 3.1.3 nRF24LE1 ohjelmointiväylä

nRF24LE1-radiopiirin ohjelmointi tapahtuu SPI-väylän avulla. Koska SPI-väylä on suunniteltu laitteiden sisäiseksi väyläksi, niin siksi sitä ei ole tietokoneissa valmiina käytettävissä. Tästä syystä täytyy väylä saada jotenkin liitettyä tietokoneeseen. Vaihtoehtoja on monia, mutta työssä on kuitenkin päädytty tekemään muunnos käyttäen Atmelin valmistamaa USB-kontrolleria. Tämän valmistajan AT90USB162-mallinen mikrokontrolleri tukee suoraa USB- ja SPI-väyliä, ja se sisältää lisäksi ohjelmamuistia. (Atmel Corporation, 2008. 1.) Koska tietokoneissa on lähes aina USB-väylä, niin laite saadaan toimimaan mahdollisimman monissa tietokoneissa.

Seinäjoen ammattikorkeakoulussa AT90USB162-mikrokontrollereita on käytössä jo muissakin sovelluksissa, joten sen ottaminen mukaan tähän työhön on perusteltua. Mikrokontrollerin aikaisempi tuntemus koulussa helpottaa sen käyttöönottoa ja jatkossa muutosten tekemistä. Kuviossa 6 näkyvä USB-tikku sisältää AT90USB162-kontrollerin, USB-liittimen sekä liittimen ulkoisille laitteille. Tikku on suunniteltu Seinäjoen ammattikorkeakoulussa, ja on ohjelmoitavissa moniin eri käyttötarkoituksiin. Tässä työssä ohjelmoinnin testauksessa USB-tikkuun on nRF24LE1-radiopiiri kytketty erillisellä kaapelilla. Koska piiristä ei työn tässä vaiheessa ole ollut käytössä omaa piirilevyä, niin USB-tikun kaapeli on jouduttu juotamaan suoraan kiinni valmistajan aloituspaketin mukana tulleeseen testikorttiin. Tämä on altistanut ohjelmoinnin testauksen monille ulkopuolisille häiriöille, joita voi aiheutua esimerkiksi juotoksista, liittimistä tai kaapelin pituudesta.



Kuvio 6. Työn alussa käytetty USB-ohjelmointilaite, sekä aloituspaketin mukana tullut radiopiiri.

### 3.2 Ohjelmointi

Intel 8051 -mikrokontrollerille ohjelmointia voi tehdä joko C++-, C- tai assembly-kielillä. Assembly on paljon laiteläheisempää ohjelmointia kuin C, jonka takia se on myös paljon vaikeampaa. Vaikka assembly-kielillä ohjelmoiminen onkin hidasta, niin hyvä ohjelmoija osaa tehdä sillä tehokkaampaa koodia, kuin mitä C-kielillä voi tehdä. (Barr 2000.)

C-kieli on monille nykyisin tutumpaa, siksi se on ajateltukin ottaa pohjaksi myös tässä työssä. Seinäjoen ammattikorkeakoulussa on aiemmin käytetty ohjelmointityökaluna ConText-nimistä ohjelmointiympäristöä, joka tarjoaa hyvät mahdollisuudet omien ohjelmien liittämiseksi ohjelmointiympäristöön, etenkin kun kyseessä on ohjelmien kääntämistä ja lataamista. ConText tukee monia eri ohjelmointikieliä, sekä osittaista projektien hallintaa. Lisäksi se sisältää myös muut tekstin muokkaamiselle tarvittavat työkalut. (ConText. 2007.)

### 3.3 SDCC-kääntäjä

SDCC-kääntäjä on vanha ja monipuolinen ANSI C -kielen kääntäjä. SDCC-kääntäjällä voidaan C-ohjelmointikieltä kääntää monille eri mikrokontrollereille, kuten esimerkiksi tässä työssä käytettävälle Intel 8051 -prosessorille. Kääntäjä ei itsessään sisällä minkäänlaista graafista käyttöliittymää, vaan kääntäminen ja muutkin toiminnot hoidetaan tietokoneen komentorivillä. Tämä mahdollistaa kääntäjän käyttämisen monissa eri tilanteissa ja antaa jatkokehittäjillä hyvät mahdollisuudet lisätä SDCC-kääntäjä myös omiin projekteihinsa. SDCC-kääntäjä on myös täysin ilmainen. (SDCC. 2009.)

Tähän työhön SDCC-kääntäjä on hyvä, koska se tukee nRF24LE1-piirissäkin käytettävää Intelin 8051-arkkitehtuuria. SDCC:n ilmaisen käytön takia se soveltuu hyvin opiskelijoiden, aloittelijoiden sekä opetuksen tarpeisiin. Koska SDCC-kääntäjä on tekstipohjainen, niin se on myös helposti liitettävissä ConText-ohjelmointiohjelmaankin. SDCC:llä ohjelmakoodin kääntäminen on paljon laiteläheisempää kuin monilla muilla kääntäjillä. Se tekee kääntäjästä monimutkaisemman käyttää, mutta samalla antaa käyttäjälle enemmän mahdollisuuksia muokkaila oman ohjelman muistinkäyttöä ja muistialueita.

### 3.4 Intel HEX -tiedosto

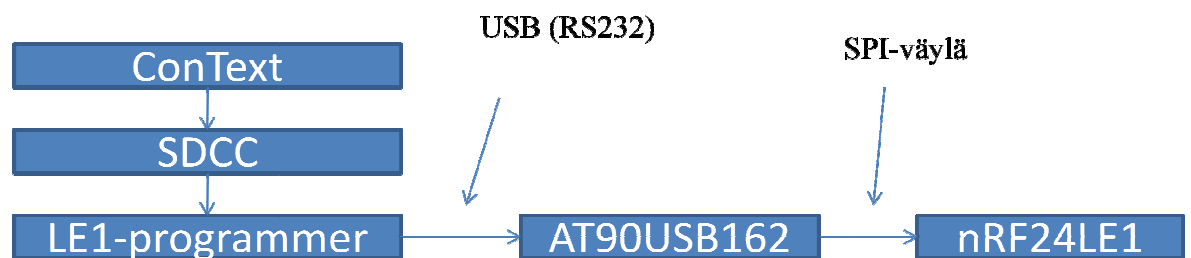
Ohjelmakoodin kääntämisessä SDCC luo muiden kääntäjien tapaan tiedoston, joka sisältää yleensä ohjelmakoodin heksamuodossa. Tämä tiedosto on SDCC:llä luokiteltu Intel HEX -nimiseksi tiedostoksi ja se sisältää tietoa muun muassa ohjelmakoodin osista, niiden sijainnista kohdelaitteen muistissa, tiedon pituuden sekä virheentarkistussumman. (Intel 1988.) Tarkemmin tiedoston sisältö on esitettyinä kuviossa 7.

RECORD MARK ' : '	RECLLEN	LOAD OFFSET	RECTYP	INFO or DATA	CHKSUM
1-byte	1-byte	2-bytes	1-byte	n-bytes	1-byte

Kuvio 7. Intel HEX -muotoisen tiedoston yhden rivin osat. (Intel 1988.)

### 3.5 Ohjelman lataaminen

Koska nRF24LE1-radiopiirillä ei ole ohjelman lataamiseen muuta mahdollisuutta kuin Nordic Semiconductorin valmistamalla aloituspaketilla, täytyi ohjelmointiin suunnitella oma laitteisto ja ohjelmat. Edellä mainituista rajapinnoista ei tietokoneohjelmissa tarvitse huomioida kuin USB-rajapinta, jolla ohjelmointikortti on kytkettynä tietokoneeseen. Ohjelmointikortissa käytetty AT90USB162-mikrokontrolleri on ohjelmoitu käyttämään valmistajalta saatua ohjelmakoodia, joka luo tietokoneelle virtuaalisen RS232-sarjaportin. Tämän portin käyttö on tietokoneella helppoa ja nopeaa. Virtuaalinen sarjaportti näkyy tietokoneohjelmille täysin samanlaisena kuin oikea sarjaporttikin, mutta tiedonsiirtonopeus on paljon sarjaporttia nopeampi ja luotettavampi. Lisäksi sarjaporttia käyttävät ohjelmat toimivat suoraan USB-tikun käyttämän virtuaalisen sarjaportin kanssa. Kuviossa 8 on kuvattu tiedon kulku laitteiden välisten rajapintojen sekä väylien välillä.



Kuvio 8. Rajapintakuvaus.



### 3.5.1 LE1-programmer

SDCC-kääntäjän luoman Intel HEX -muotoisen tiedoston tulkitsemiseen on C#-ohjelmointikielellä ohjelmoitu konsolipohjainen ohjelma. Tämän työn LE1-programmer-nimisen ohjelman tehtävänä on paloitella Intel HEX -tiedoston sisältö, muuttaa se AT90USB162-mikrokontrollerin ymmärtämään muotoon ja lähettää se USB-portin kautta USB-tikulle virtuaalisen sarjaportin kautta. Lisäksi ohjelma osaa lukea ulkoisen InfoPage-asetustiedoston, jossa on sisällytettyinä muutamia ohjelmointiin oleellisia asetuksia.

LE1-programmer-ohjelma on tehty siten, että sille annetaan käynnistämisen yhteydessä komentoriviargumenttien arvoina sarjaportin numero, asetustiedoston nimi sekä Intel HEX -muotoisen ohjelmatiedoston nimi. Koska ohjelma on konsolipohjainen, niin komentoriviargumenttien käyttö on ainoa tapa antaa ulkoisia ominaisuuksia. ConText-ohjelma tukee myös näiden konsolipohjaisten ohjelmien käyttöä niin, että ohjelmille annettavien arvojen käyttö onnistuu komentoriviargumenttien avulla.

Kuviossa 9 LE1-programmer tulostaa konsoliin ohjelmoinnin eri vaiheet ja asetetut InfoPage-asetukset. Konsoliin tulostuu myös viestit onnistuneesta ohjelmoinnista tai virheellisestä CRC-tarkistussummasta.

```

> "C:\Program Files\LE1-programmer\LE1Programmer.exe" m:\nrf24le1-kehitys\nrf_dev\nrf_net\main.ihx COM10 InfoPage.txt
*****
nRF24LE1 programming tool
Made by: Marko Huhta
*****
Opening serial port... port COM10 opened
InfoPage.txt -file opened

TARGET = 0x00
INFOPAGE = 0x01
NUPP = 0xFF
RDISMB = 0xFF
ENDEBUG = 0xFF

Starting to program InfoPage... done!
Starting to program data to unprotected memory... done!

> Process Exit Code: 0
> Time Taken: 00:04

```

Kuvio 9. Kuvankaappaus LE1-programmer-ohjelman käytöstä.

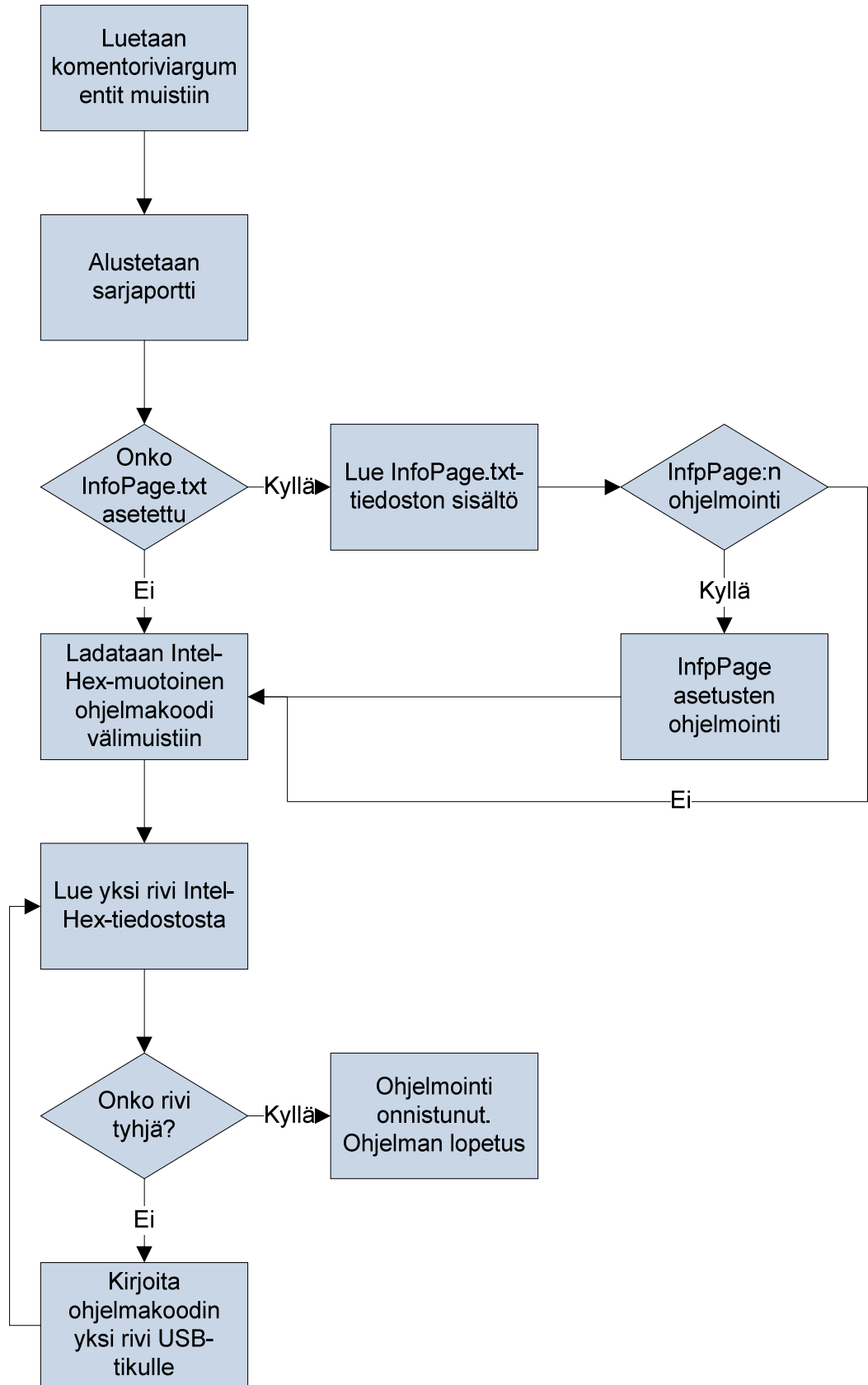
### 3.5.2 InfoPage-asetukset

Ohjelmoinnin yhteydessä LE1-programmer-ohjelma lukee ulkoisen InfoPage-asetustiedoston. Nämä asetukset asetetaan erillisillä komennoilla radiopiiriin mikrokontrollerille, mutta LE1-programmer-ohjelmaan nämä asetukset on liitetty tiedoston kautta, joka helpottaa käyttäjää asetusten muokkauksessa. USB-tikku on suunniteltu niin, että se osaa automaattisesti muuttaa ja tallentaa tiedoston sisältämät asetukset. Kuvion 10 vuokaaviosta selviää, miten LE1-programmer tarkistaa InfoPage-asetustiedoston ja ohjelmoi radiopiiriin. Liite 2 sisältää InfoPage-asetustiedoston sisällön.

Tiedosto sisältää seuraavia radiopiiriin mikrokontrollerin asetuksia

- InfoPage-asetusten ohjelmoinnin sallinta.
- Ohjelmoidaanko Intel HEX -tiedoston sisältö muistin suojatulle vai suojaamattomalle alueelle.
- Suojaamattoman muistialueen koko.
- Ohjelmamuistin salliminen ulkoisilta liitynnöiltä.
- Sallitaanko laitepohjainen JTAG ohjelmointi ja testaus.

(Nordic Semiconductor 2009, 71 - 72.)



Kuvio 10. LE1-programmer-ohjelman vuokaavio.

### 3.6 USB-tikku

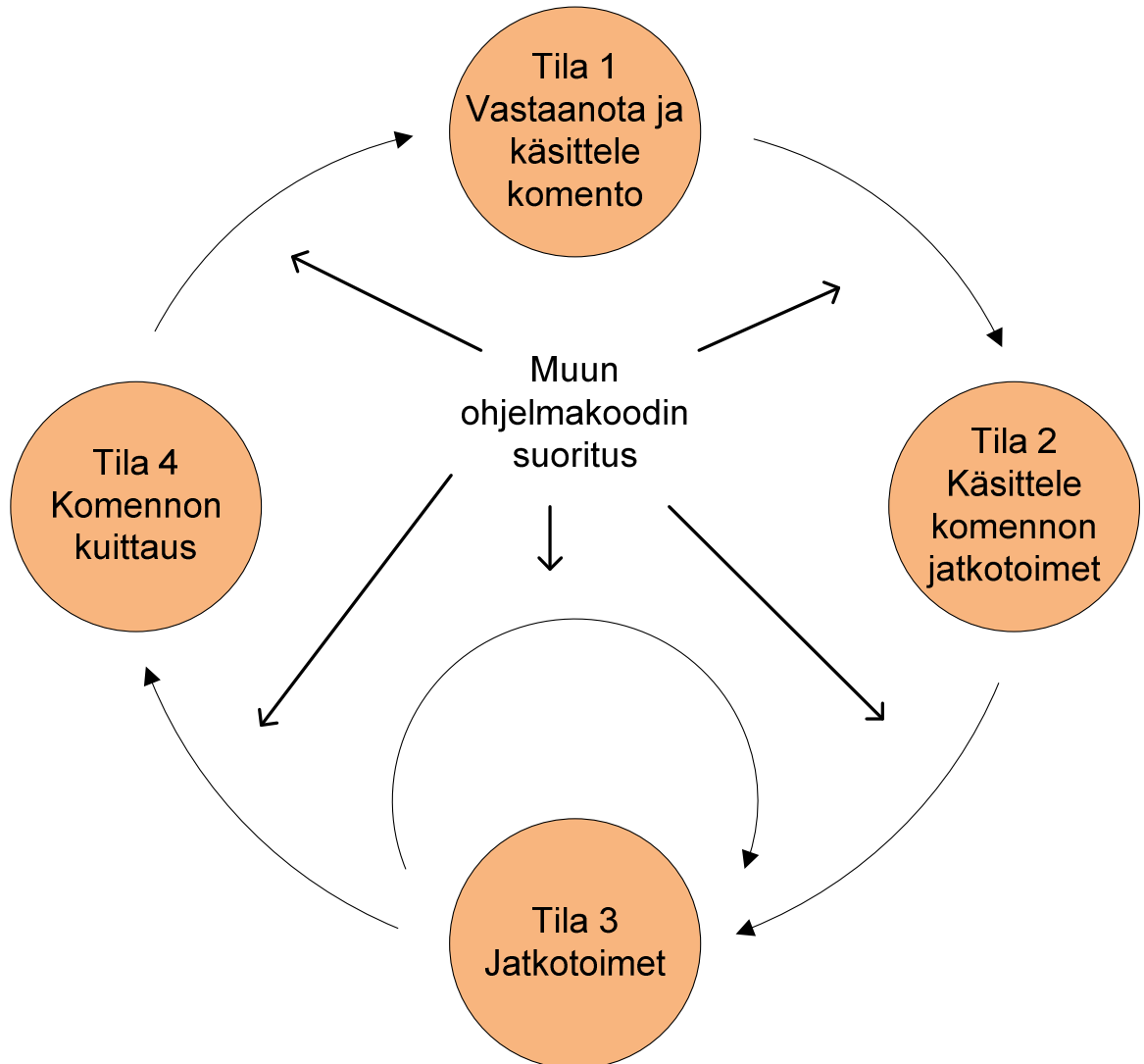
USB-tikun ohjelmassa on pohjana käytetty Atmelin jakamaa valmista ohjelmakoodia, joka käsittelee USB- ja sarjaporttiväylien käytön. Tämä valmis ohjelmakoodi on toteutettu reaaliaikaohjelmoinnilla, eli ohjelma on jaettu eri osatehtäviin ja jokaiselle osatehtävälle annetaan suoritusvuoro esimerkiksi sen prioriteetin mukaan. USB-tikussa on muutamia osatehtäviä, jotka hoitavat USB-yhteyden ylläpitoa. Tämän vuoksi mitkään muut osatehtävät eivät saa olla pysäyttäviä, vaan niiden on annettava suoritusaikaa myös muille osatehtäville.

#### 3.6.1 Toiminta

USB-tikun ohjelmaan on jouduttu luomaan radiopiirin mikrokontrollerin ohjelmointia varten uusi osatehtävä, jossa luetaan mahdollisesti tietokoneelta tulleet komennot ja käsitellään komentoja. Tietokoneelta tulevia komentoja voi olla useita ja tietokoneen lähettämä data joudutaan vastaanottamaan useissa eri osissa, joten niiden käsittely ei voi olla USB-tikun muun ohjelmakoodin pysäyttävä. Jotta USB-tikun muillekin osatehtäville voitaisiin antaa suoritusaikaa, on tietokoneelta tulevien tietojen käsittelyyn jouduttu tekemään tilakone. Tilakoneessa USB-tikku vastaanottaa komennon tietokoneelta ja asettaa tilakoneen seuraavaksi tilaksi komennon määrittävän tilan. Mitään komennon mukaisia toimintoja ei vielä suoriteta, vaan tässä vaiheessa annetaan USB-tikun muille osatehtäville aikaa. Kun USB-tikun ohjelmakoodi antaa uudestaan aikaa tälle osatehtävälle, niin kyseisen tilan toiminto suoritetaan ja mahdollisesti seuraavan vaiheen tila asetetaan tai palautetaan aloitustilaan.

Kuviossa 11 on tilakaaviona esitetty USB-tikun tilakone. Tästä selviää, että ohjelmakoodin ohjelmoinnin ensimmäisessä vaiheessa USB-tikku vastaanottaa tietokoneelta ohjelmoinnin komennon. Tämän jälkeen se valmistautuu ohjelmoimaan seuraavalla osatehtävän vuorolla. Kun vuoro tulee, niin USB-tikku vastaanottaa yhden rivin Intel HEX -tiedoston sisällöstä, käsittelee ja laskee virhesumman, jonka jälkeen se ohjelmoi yhden rivin tiedot mikrokontrollerille. Seuraavalla osatehtä-

vän vuorolla vastaanotetaan, käsitellään ja ohjelmoidaan seuraava rivi. Näitä kutsutaan ohjelmointikomennon jatkotoimiksi, jotka keskeytyvät kun kaikki rivit on ohjelmoitu ja komento lopetuksesta saatu.



Kuvio 11. Tilakaavio USB-tikun ohjelmakoodin tilakoneesta.

USB-tikku tarkistaa jokaisesta saapuvasta rivistä CRC-tarkistussumman havaitsemat mahdolliset virheet. Jos virheitä ei tule, niin USB-tikku siirtää ohjelmakoodin SPI-väylän kautta nRF24LE1-radiopiirin mikrokontrollerin muistiin. Tämä tapahtuu myös tilakoneen avulla. Jokainen Intel HEX -tiedoston rivi käsitellään erikseen osatehtävässä, joten jokaisen rivin käsittelyn ja ohjelmoinnin välissä tilakone antaa USB-tikun prosessorin muille osatehtäville aikaa esimerkiksi USB-yhteyden ylläpi-

toa varten. Näin kokonaisuudesta saadaan luotettava. USB- ja SPI-väylät ovat todella nopeita, joten radiopiirin koko ohjelmamuistin ohjelmoiminen vie aikaa vain muutaman sekunnin.

### **3.6.2 Muistin luku ja vianetsintä**

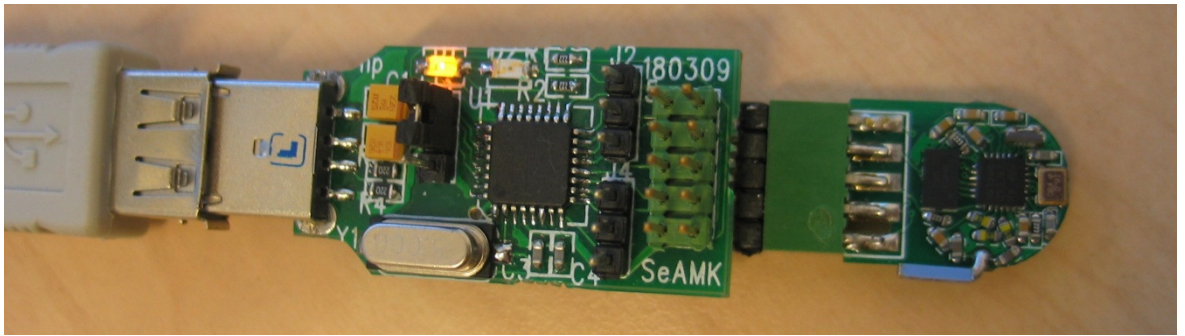
Vianetsintätarkoitukseen ja testaukseen on USB-tikun ohjelmaan lisätty muutamia komentoja, joilla voidaan radiopiirin muistia ja InfoPage-asetuksia tarkastella kaikilla yleiskäyttöisillä sarjaporttiterminaali-ohjelmilla. Radiopiirin mikrokontrollerilta voidaan eri komennoilla lukea ja tyhjentää koko ohjelmamuistin sisältö. Lisäksi voidaan lukea InfoPage-asetusmuistin sisältö sekä kirjoittaa laitevalmistajan antamat InfoPage-asetusmuistin oletusasetukset. Näitä toimintoja on käytetty apuna myös tämän työn aikana ja ne on jätetty käytettäväksi myös jatkokehityksiä varten.

## 4 PIIRILEVYN SUUNNITTELU

### 4.1 Piirilevy

Työssä on tarkoitus suunnitella oma piirilevy tätä työtä ja jatkokehityksiä varten. Valmiin aloituspaketin mukana tullutta radiopiirin piirilevyä on hankala käyttää ison koon ja huonon liitynnän takia, eikä sitä ole sellaisenaan käytettäväksi tarkoitettu-kaan. Tähän asti työssä käytetty piirilevy on alun perin ollut tarkoitus käyttää ainoastaan saman valmistajan aloituspaketin kanssa.

Kuvion 12 ja 13 näkyvässä piirilevyssä on yhdelle sivulle asetettu erittäin yleiskäytöinen 10-nastainen liitin, jolla piirilevyn liittäminen ohjelmointitikkiin ja mahdollisesti omiin sovelluksiin on helppoa. Liittimelle on tuotu kaikki ohjelmointiin liittyvät toiminnot sekä kaikki muut mikrokontrollerin tukemat ominaisuudet lukuun ottamatta kellokiteen nastoja, jotka menevät kiinteästi kiteelle. Radiopiirin ja liittimen välissä on Freescale Semiconductorin MMA7360-kiikkyvyysanturi. Piirilevyn oikeassa reunassa on liityntä antennille, joka voidaan juottaa piirilevyn kylkeen tai piirilevyn läpi mahdollistaen monenlaisen käyttökohteen. Tämä on tärkeää, sillä antennin sijoittelu erilaisten kotelo- ja komponenttimateriaalien lähelle voi aiheuttaa epätoivottuja tilanteita (Schoepke 2006, 1 - 2).



Kuvio 12. Piirilevy on kytkettynä USB-ohjelmointitikkuun.

## 4.2 Ominaisuudet

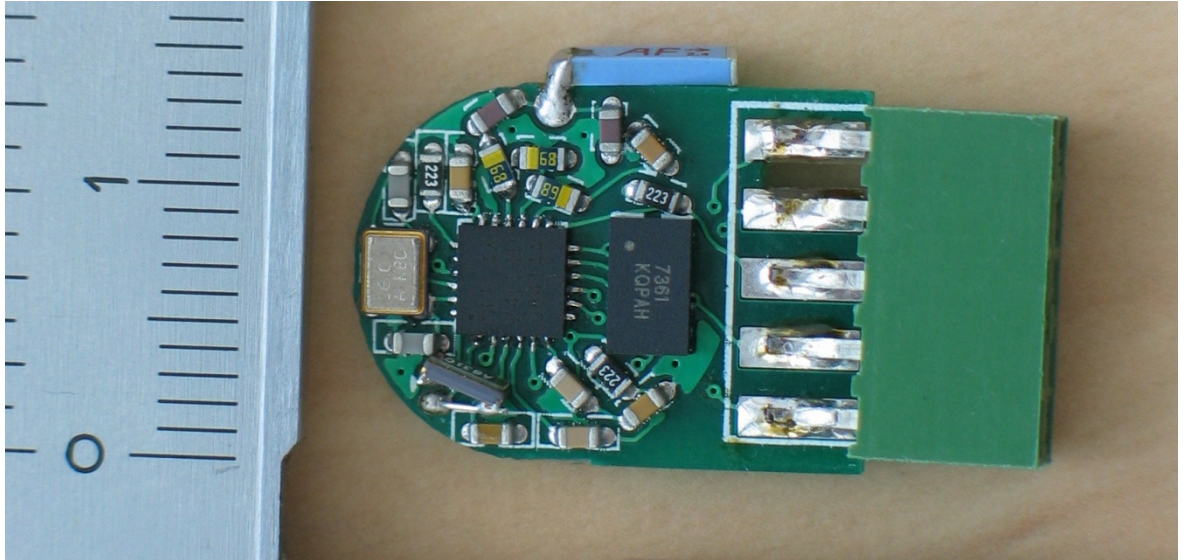
Piirilevyn kytkentä mahdollistaa lähes kaikkien mikrokontrollerin ominaisuuksien käytön. Ainoastaan kellokide varaa piirin kaksi IO-nastaa. Piirilevyllä oleva kellokide mahdollistaa laskureiden käytön 32,768 kHz taajuudella, eli se toimii samalla millisekuntilaskurina. Piirilevyllä olevalla kiihtyvyyssanturilla voidaan mitata kiihtyvyyksiä  $\pm 1,5$  g ja  $\pm 6$  g mitta-alueilla kolmelta eri akselilta. Kaikki kolmen akselin mittaustiedot ovat analogisia ja ne ovat kytkettyinä nRF24LE1-radiopiirin analogiatuloihin sekä myös liittimelle. Piiriltä on lisäksi kytketty kaksi nastaa ohjaamaan kiihtyvyyssanturin virransäästötilaa sekä mitta-alueen valintaa. Piirilevyn kytkentä on liitteenä 3.

## 4.3 Käyttömahdollisuudet

Kun piirilevy on suunniteltu siten, että mistään ominaisuuksista ei ole luovuttu ja vain pakolliset komponentit on sijoitettu piirilevyllä, voidaan sitä käyttää monissa eri tilanteissa. Tässä työssä suunniteltu piirilevy on monikäyttöinen esimerkiksi Seinäjoen ammattikorkeakoulun hankkeissa ja erilaisissa projekteissa, koska niissä usein vaaditaan pientä virrankulutusta, pientä kokoa sekä helppoa käyttöönottoa. Myös koulun omissa opetustarkoituksissa voidaan radiopiirin tekniikoihin tutustua. Piirilevyn pyöreä muoto ja 14 mm:n halkaisija mahdollistaa sen käytön



esimerkiksi AA-kokoisten paristojen kanssa, koska AA-kokoisen pariston halkaisija on myös noin 14 mm. Piirilevyllä olevan kiihtyvyyssanturin ansiosta, sitä voidaan käyttää monissa tärinän ja liikkeen mittauksissa. Ulkoiset väylät mahdollistavat erilaisten anturipiirien kytkennät tai vaikka sarjaliikenteen tietokoneen kanssa.



Kuvio 13. Piirilevyn pieni koko mahdollistaa asennuksen moniin pieniin kohteisiin.

Piirilevy koostuu pääosin yleiskomponenteista, joten sen hinta on erittäin kilpailukykyinen. Tässä työssä käytettävän piirilevyn hinta on niin alhainen, että sitä voitaisiin käyttää jopa kertakäyttötuotteissa.

## 5 VERKON TAHDISTUS

### 5.1 Ongelma

Langaton verkko sisältää aina vähintään kolme laitetta, jotka pystyvät kommunikoimaan keskenään langattomasti. Koska kyseessä on langaton verkko, täytyy laitteiden olla helposti liikuteltavissa, vaihdettavissa sekä monesti piilotettavissa. Sillä muutoin verkko olisi kannattavampaa toteuttaa langallisena. Langallisen verkon toteuttaminen olisi halvempaa ja tehokkaampaa, mutta estää laitteiden liikuteltavuuden ja muokattavuuden. Langattoman verkon yhden laitteen helppo asennus esimerkiksi luontoon, vaatteisiin tai huonekaluihin vaatii laitteelta pientä kokoa ja pitkää huoltoväliä. Yleisimmiksi ongelmiksi niissä onkin tulleet akkujen suuri koko sekä niiden lyhyt kesto, sillä tiedon siirtäminen langattomien laitteiden avulla kuluttaa virtaa huomattavasti enemmän kuin langallisilla laitteilla. Tämä johtuu laitteen antennin tarvitsemasta suuresta virrasta. Yleensä lähetysteho on verrannollinen laitteen kuluttamaan virtaan. (Basavaraju, Puttamadappa & Sarkar 2008, 8 - 10.)

Jotta langattoman laitteen käyttö saataisiin tehokkaammaksi, tulisi virrankulutusta saada pienennettyä. Langattomalla laitteella yleensä tiedon vastaanottaminen kuluttaa hieman enemmän virtaa kuin tiedon lähettäminen. Jotta laite voisi vielä vastaanottaa tietoa, niin pitää sen kuunnella jatkuvasti, onko tietoa saatavilla. Tällä tavalla laite kuluttaisi jatkuvasti täyden määrän virtaa ja akun kesto olisi todella lyhyt. Tehokkainta olisi, jos laite tietäisi etukäteen, koska tietoa olisi saatavilla ja näin asettaisi isovirtaisen vastaanottotilan päälle juuri oikeaan aikaan. Laite lukisi näin tietoa vain sen ajan, kun sitä on saatavilla ja sammuttaisi vastaanottotilan heti kun kaikki tieto on vastaanotettu. Verkossa tällainen toteutus on todella haastavaa, koska verkon jokaisen laitteen tulisi tietää jokin yhteinen ajankohta, jolloin tietoja voitaisiin lähettää ja vastaanottaa. Lisäksi yksinkertaisimmat laitteet eivät voi lähettää sekä vastaanottaa samaan aikaan, joten jokaisen laitteen tulisi myös tietää yhteiset vuorot tietojen lähettämiseksi ja vastaanottamiseksi. Vastaanottotilan ollessa pois päältä, laite olisi mahdollisimman pienivirtaisessa tilassa. Tämä tila tulisi olla huomattavasti aktiivista aikaa pidempi, jotta kokonaisvirrankulutus saataisiin

merkittävästi pienemmäksi. Langattomissa anturiverkoissa ei tarvita suurta tiedon-  
siirtonopeutta, joten tällainen toteutus olisi mahdollista. (Basavaraju, Puttamadap-  
pa & Sarkar 2008, 214 - 218.)

## 5.2 Tahdistuksen käyttö GENSEN-projektissa

GENSEN-projektin yhtenä osa-alueena on itsenäisen langattoman verkon suunnit-  
telu. Tässä työssä ei kuitenkaan ole tarkoitus kehittää toimivaa laitteistoa tiettyihin  
kohteisiin, vaan suunnitella yleiskäyttöinen tahdistus, jota voitaisiin käyttää pohja-  
na kaikissa mahdollisissa projekteissa. (GENSEN 2009, 4 - 5.)

GENSEN-projektin projektisuunnitelmassa on määriteltynä verkolle seuraavia  
käyttökohteita

- sähkölaitteiden valvonta
- lehmien valvonta
- rakenteiden kunnan valvonta
  - tuulivoimaloiden siipien värinän mittaus
  - siltojen ja rakennusten värinän mittaus
- kasvihuoneen valvonta
- energiantuotantoalueen valvonta ja ohjaus.

(GENSEN 2009, 5.)

Näistä edellä mainituista kohteista on tähän työhön otettu lähtökohdaksi lehmien ja  
tuulivoimaloiden siipien valvonta, koska näissä tapauksissa Tekniikan yksikössä  
kehitetty laitteisto toimii sellaisenaan. Myös laitteistoon suunnitellut mittauskom-  
ponentit käyvät molempiin kohteisiin.

Langattomissa laitteissa käytettävän tahdistuksen takia mittauksien välinen aika  
on suuri. Tämä tekee mittauksista epärealistisia hetkellisesti, mutta pitemmällä  
aikavälillä mittauksista voidaan päätellä värinän, liikkeiden sekä asentojen muutok-

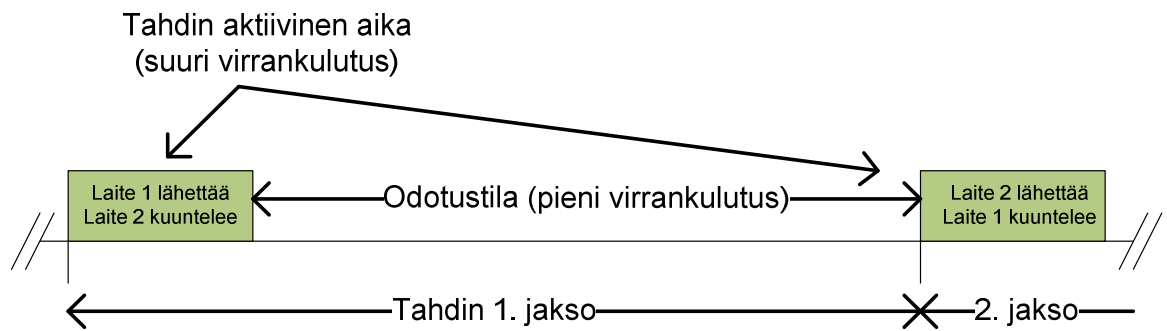
sia sekä vertailla arvoja muiden vastaavien laitteiden arvoihin. Tällainen mittaaminen on hyvinkin toteutettavissa esimerkiksi lehmien levottomuuksia mitattaessa.

### **5.3 Tahdistuksen teoria**

Hyvin toimivan tahdistuksen tärkein ominaisuus on samanaikainen tahti, joka on mahdollisuus toteuttaa vain mikrokontrollerin tarkalla laskurilla ja kellokiteellä. Verkon kaikkien laitteiden laskurien on oltava samanaikaiset, jotta pitkällä taukovälillä ei tapahdu suuria tahdin eroavaisuuksia, ja näin tiedon lukeminen ei tapahtuisi väärään aikaan. Jokaisen laitteen käynnistäminen verkossa tai uuden laitteen lisääminen verkkoon tuo tahdistamisessa ongelmia, koska laitetta ei käytännössä mahdollista käynnistää samaan aikaan jonkun muun laitteen kanssa niin että tahti olisi sama. (Basavaraju, Puttamadappa & Sarkar 2008, 242.)

#### **5.3.1 Perusidea**

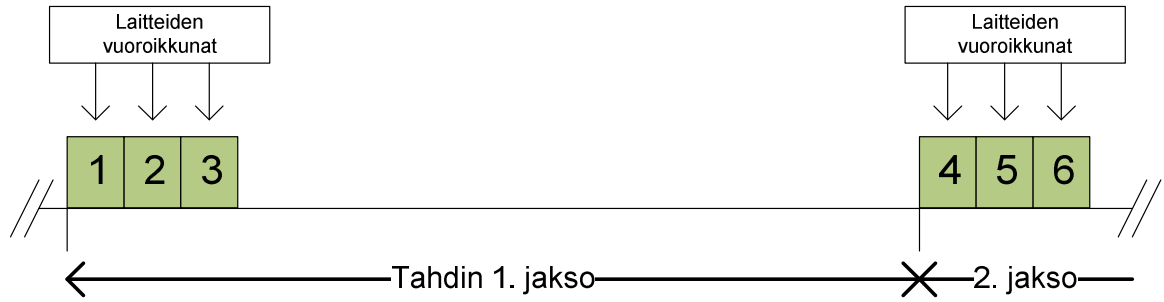
Tahdistuksessa on käytettävä laskutoimenpidettä, jolla saadaan laskettua oikea tahti. Jokaisen uuden laitteen on kuunneltava käynnistyksen yhteydessä hetken muita lähetettäviä laitteita ja siten tahdistaa oma laskuri samaan tahtiin kuin viestin lähettäjän laskuri. Tämä on mahdollista laskemalla matemaattisesti viestin lähettämiseen, siirtämiseen sekä vastaanottamiseen kuluvat ajat. Lisäksi pitää kirjata ylös oman laskurin arvo viestin vastaanottohetkellä. Lasketuista ajoista ja niitä omaan laskurin aikaan vertaamalla saadaan tietää lähettäjän tahdin eroavaisuus omaan tahtiin, ja näin omaa tahtia muuttamalla eroavaisuuden verran päästään lähettäjän kanssa samaan tahtiin. Kuviossa 14 on esimerkki tilanteesta, jossa samassa tahdissa olevat laitteet lähettävät ja kuuntelevat vuorotellen. Lähetettävän tiedon täytyy sisältää vuoronumero, jonka perusteella vastaanottaja tietää kenen vuoro on seuraavalla tahdilla lähettää. (Basavaraju, Puttamadappa & Sarkar 2008, 243.)



Kuvio 14. Yksinkertaisen tahdistuksen toimintaperiaate.

### 5.3.2 Laajennettu verkko

Edellä mainittu tapa toimisi hyvin, jos laitteita olisi vain vähän ja yksi laite voisi lähettää yhden tahdin aikana. Tavoitteena on kuitenkin kehittää isompi verkko, joka voisi sisältää noin 30 – 250 laitetta, joten useammalle laitteelle on annettava lähetysvuoro samaan tahtiin ja vielä niin, että tietojen lähetys ja vastaanotto eivät mene päällekkäin. Isommassa verkossa laite pääosin kuuntelee tahtien aikana muiden laitteiden lähettämiä viestejä ja oman lähetysvuoronumeron perusteella laskee, millä tahdilla laitteella on vuoro itse lähettää viesti, jolloin muut laitteet kuuntelevat. Tätä lähetysvuoromenetelmää on nopeutettu vielä lisäämällä yhteen tahtiin esimerkiksi kolmelle eri laitteelle omat vuoroikkunat. Kuvion 15 mukaisesti on yhden tahdin aikana kolmella laitteella aikaa lähettää yksi viesti ja vastaanottaa kaksi viestiä, sillä jos laitteella on vuoro lähettää, niin lähettämiseen kuluva aika estää tiedon vastaanottamisen. Jokainen laite osaa omasta lähetysvuoronumerostaan laskea tahdin jolloin viesti lähetetään. Lähetysvuoronumerosta voidaan laskea myös tahdin sisältämä ikkunavuoro.



Kuvio 15. Laitteiden aktiiviaika on jaettu useampaan eri ikkunaan, jolloin tietoa saadaan siirrettyä kerralla enemmän.

Koska yhden tahdin aikana tietoja lähettää nyt useampi laite, ja välttämättä aina ei esimerkiksi tahdin ensimmäisellä ikkunalla oleva laite lähetä viestiä, niin täytyy vastaanottajan tietää miten tahdistaa oman laskurinsa oikeaksi. Viestin vastaanottaja ei tiedä, millä vuoroikkunalla viesti on vastaanotettu, joten tahtia ei välttämättä saada kohdistettua niin, että lähettäjän ja vastaanottajan kaikki kolme vuoroikkunaa osuvat kohdalleen. Vastaanotetun viestin lähetysvuoronumerosta voidaan kuitenkin päätellä lähettäjän lähetysvuoron lisäksi sen vuoroikkuna. Vuoroikkunan perusteella tiedetään, monenko ikkunan kuluttaman ajan verran laskuria pitäisi siirtää. Viestin vastaanottajan täytyy laskea lähetysvuoronumerosta vuoron tunnus, joka on kyseisen vuoron ensimmäisen vuoroikkunan tunnus.

Vuoron tunnus lasketaan seuraavalla kaavalla

$$N_w = \left\lfloor \frac{N_t - 1}{3} \right\rfloor * 3 + 1 \quad (1)$$

jossa  $N_t$  on viestin lähettäjän lähetysvuoro  
 $N_w$  on vuoron tunnus

Tahdistuksen aikaero voidaan nyt laskea seuraavalla kaavalla

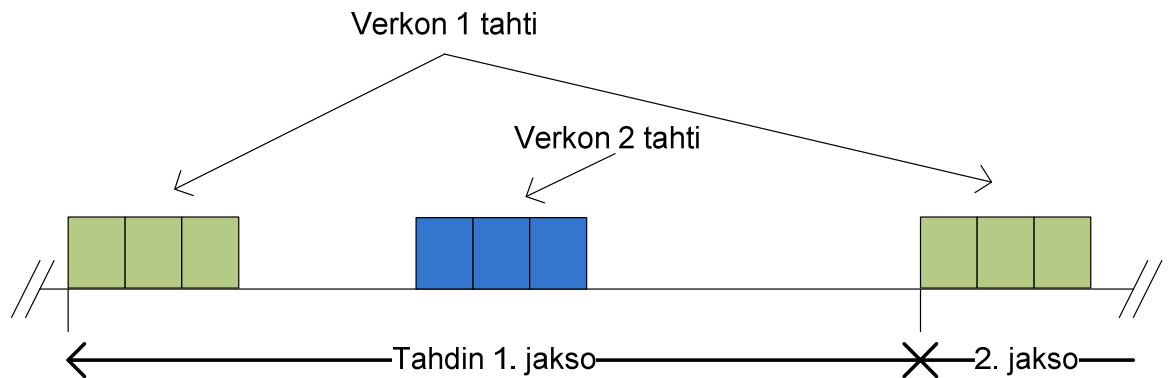
$$\Delta t = t_{int} - (t_1 + ((N_t - N_w) * t_2) + t_3) \quad (2)$$

jossa	$\Delta t$	on	tahtien aikaero
	$t_{int}$	on	tahdin aika viestin saapumishetkellä
	$t_1$	on	tahdin alkutoimiin kuluva aika
	$t_2$	on	vuoroikkunan aika
	$t_3$	on	radioliikenteeseen kuluva aika

Isoissa verkoissa kaikki laitteet eivät ole yleensä suorassa yhteydessä toisiinsa. Tämän vuoksi verkossa voi olla maksimissaan 15 eri lähetysvuoronumeroa, mutta laitteita jopa 250. Yhden laitteen suorassa yhteydessä voi olla maksimissaan 15 laitetta.

### 5.3.3 Eri tahtien estäminen

Laite tahdistaa oman laskurinsa ensimmäisen vastaanotetun viestin mukaisesti ja sen jälkeen sammuttaa vastaanottotilan pois päältä odotellessaan seuraavaa tahtia. Tämä voi aiheuttaa usean eri tahdin verkkoon. Kuviosta 16 näkee miten eri tahdit voivat sijoittua jaksolle. Jos kaksi itsenäistä ja eri tahdissa olevaa verkkoa yhdistetään, niin lopullisessa verkossa on kaksi eri tahtia, sillä laitteet eivät pysty kuuntelemaan muita vastaanottotilan ollessa pois päältä. Saman tahdin laitteiden on pidettävä kirjaa yhteisestä tahdin tunnuksesta, joka on sama kaikilla samassa tahdissa olevilla. Alkutilanteessa tahdin tunnus voi olla esimerkiksi laitteen oma tunnus. Kahdella eri tahdissa olevalla verkolla on siis eri tahdin tunnukset. Jokaisen laitteen on satunnaisin väliajoin jätettävä vastaanottotila päälle kahden tahdin väliseksi ajaksi, jotta laite voisi kuunnella onko muita mahdollisia tahtia saatavilla. Jos laite huomaa toisen tahdin olemassaolon niin se vertailee saapuvasta viestistä tahdin tunnuksia ja tekee päätöksen siirtymisestä toiseen tahtiin. Päätös tahdin muutoksesta voidaan tehdä esimerkiksi tahdin tunnusnumeron suuruuden mukaan.



Kuvio 16. Esimerkki tilanteesta, jossa tahdin aikana ovat kahden eri verkon aktiivijat eri kohdissa.

#### 5.3.4 Virrankulutus

Koska suurin osa radiopiirin mikrokontrollerin toiminnoista on integroituna itse piiriin, on ulkoisten komponenttien määrä vähentynyt ja sen seurauksena myös erilaisten komponenttien vaikutus virrankulutukseen pienentynyt. Koska toiminnot ovat piirin sisällä, on niitä myös helpompi ohjata ohjelmakoodin avulla. Esimerkkinä vanhempi Nordic Semiconductorin radiopiiri, joka ei sisältänyt omaa mikrokontrolleria. Tätä vanhempaa radiopiiriä ohjataan ulkoisen SPI-väylän kautta, joka on huomattavasti hitaampaa kuin integroituna. Koska väylä hidasti kommunikointia, niin lisäsi se myös radiopiirin aktiivisena olevaa aikaa, joka taas lisää virrankulutusta.

Mikrokontrollereissa on usein monia erilaisia virransäästötiloja, joiden virrankulutus muuttuu sen mukaan miten paljon ominaisuuksia on päällä tai pois. Taulukosta 1 nähdään viestien vastaanottotilan suuri virrankulutus, joka on tärkeää saada mahdollisimman pieneksi.



Taulukko 1. nRF24LE1-radiopiirin virrankulutukset eri tiloissa ja eri toimintojen ollessa päällä. (Nordic Semiconductor 2009, 182 - 183.)

<b>Mikrokontrollerin ytimen virrankulutukset</b>		
Syvä unitila	0,5	µA
Rekistereiden säilytys, Ulkoinen kellokide päällä	3	µA
Käyttövalmius, Ulkoinen 16 MHz kide päällä	1	mA
Aktiivinen tila, 8 MHz, 4 MIPS	4	mA
<b>Mikrokontrollerin toimintojen virrankulutukset</b>		
Ohjelmamuistiin kirjoitus	1,8	mA
Ohjelmamuistin tyhjennys (kokonaan)	0,8	mA
Viestin lähetys, täysi lähetysteho	11,1	mA
Viestin lähetys, pienin lähetysteho	6,8	mA
Viestin vastaanotto, 2 Mt/s	13,3	mA
Viestin vastaanotto 250 kt/s	12,4	mA

## 5.4 Toteutus

Tämän työn mukaista tahdistusta on tämän työn tekijä kokeillut aiemmin Atmelin ATTiny84-mikrokontrollerin kanssa, jolla idea on saatu jotenkin toimimaan. Tässä työssä tarkoitus on kuitenkin tutkia tahdistuksen toteuttamista uudelle nRF24LE1-radiopiirille, joka antaa paremmat lähtökohdat pienivirtaiselle verkolle.

### 5.4.1 Laskuri ja tilakone

Aluksi on tutkittu mikrokontrollerin mahdollisuuksia tarkkaan laskuriin, josta saataisiin myös tarkat keskeytykset prosessorin eri toimintojen aloittamiseen. nRF24LE1-radiopiirin mikrokontrolleri sisältää lähes tällaista tahdistusta varten tehdyn laskurin. Tämä laskuri toimii ulkoisella 32,768 kHz:n kellokriteellä, jota voidaan käyttää myös tarkkana mikrosekuntilaskurina. Laskurin ansiosta voidaan verkon tahti ja mittausten väliaika säätää halutusti noin 30 µs: tarkkuudella. Tahdistuksen tarkkuus on suoraan verrannollinen kellokriteen taajuuteen. Koska kellokriteen taajuus on 32,768 kHz niin se on 32 768 kellopulssia sekunnissa. Tahdistus toimii yhden kellopulssin tarkkuudella eli noin 30,5 µs.

Tarkkuus voidaan laskea seuraavalla kaavalla

$$Tarkkuus = \frac{1s}{Kellotaajuus} \quad (3)$$

Edellistä kaavaa käyttämällä saadaan tarkkuudeksi

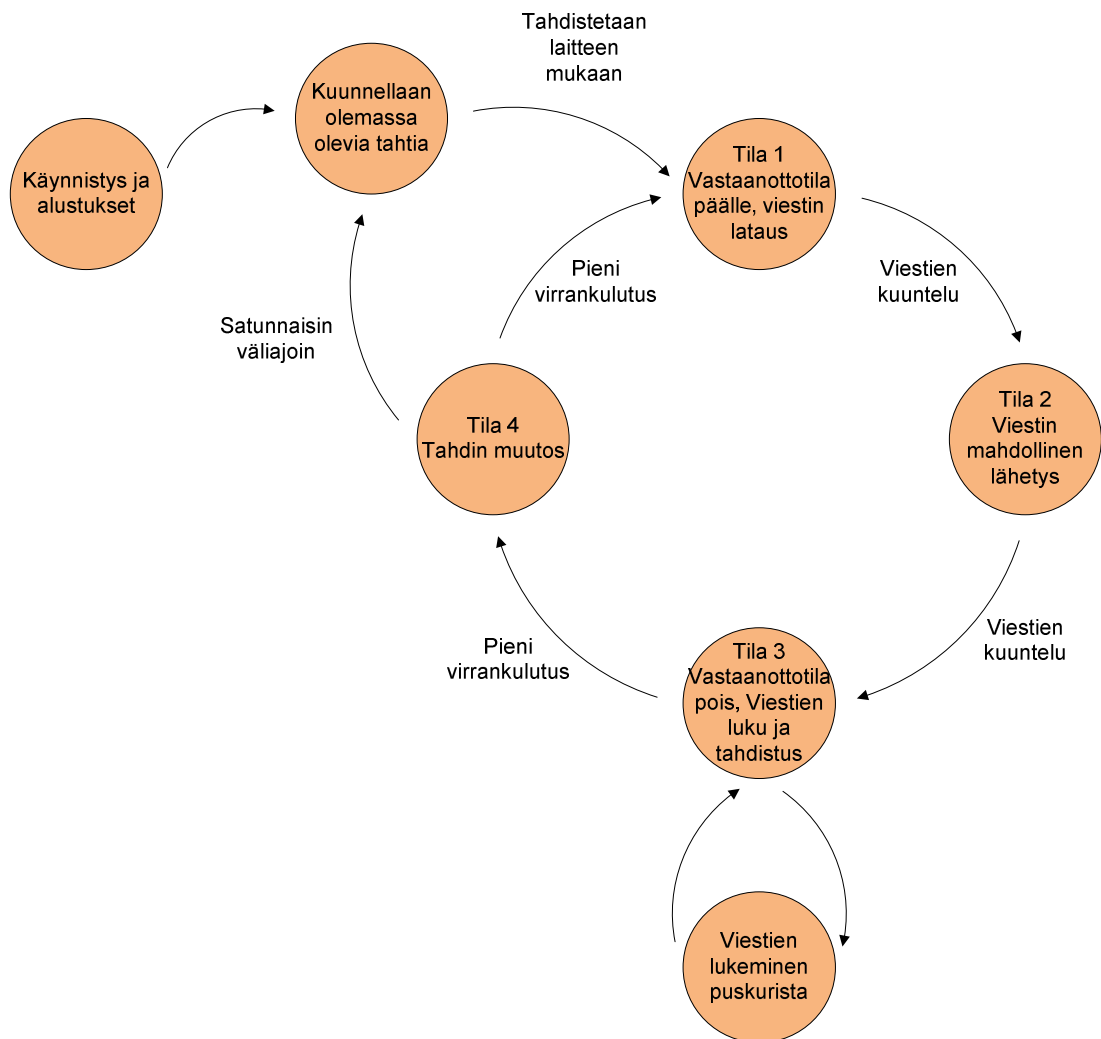
$$\frac{1 s}{32768 Hz} = 0,00003052 s = 30,5 \mu s$$

Mikrokontrollerin toimintaan kuuluu useita eri vaiheita, joita täytyy laskurin tietyillä arvoilla suorittaa. Monista kontrollereista löytyy laskureiden vertailukeskeytyksiä, joiden avulla mikrokontrollerin prosessoriytimelle voidaan antaa tarkka ajankohta toimenpiteiden suorittamiselle. nRF24LE1-mikrokontrollerin laskurissa ei ole kuin yksi vertailukeskeytys, jonka arvoa voi muuttaa. Siksi mikrokontrollerin ohjelmakoodiin on jouduttu tekemään tilakone. Piirin ollessa odotustilassa, laskurin arvo kasvaa 32,768 kHz:n taajuudella. Kun laskurin arvo kasvaa asetetun vertailukeskeytyksen arvoon, piiri lopettaa odotustilan ja aloittaa aktiivisen tilan.

Aktiivisen tilan alussa mikrokontrollerin prosessori siirtää seuraavaksi lähetettävän viestin radiopiirin lähetyspuskuriin ja laskee itselleen mahdollisen lähetysvuoron vuoroikkunan. Prosessori asettaa seuraavaksi vertailukeskeytyksen ajankohdaksi vuoroikkunan mukaisen lähetysvuoron ajan ja käynnistää samalla radioyksikön vastaanottotilan päälle mahdollisten saapuvien viestien varalle. Kun prosessorin ohjelmakoodi on suorittanut vertailukeskeytyksen, niin se antaa aikaa ohjelmakoodin pääsilmukalle. Seuraava vertailukeskeytys tapahtuu oman lähetysikkunan alussa, jolloin prosessori katsoo onko sen vuoro lähettää vai ei. Mahdollinen lähetys on vain lähetysignaalin antaminen lähetyspuskurissa olevalle viestille, joten toiminto ei kuluta paljoa aikaa. Seuraavan vertailukeskeytyksen arvo asetetaan ja ohjelmakoodi palaa pääsilmukkaan. Laskuri keskeyttää vertailukeskeytyksellä seuraavaksi, kun kaikki kolme lähetysikkunaa on käyty läpi. Tässä keskeytyksessä

prosessori sammuttaa vastaanottotilan ja käy yksitellen läpi kaikki saapuneet viestit lukemalla ne vastaanottopuskurista.

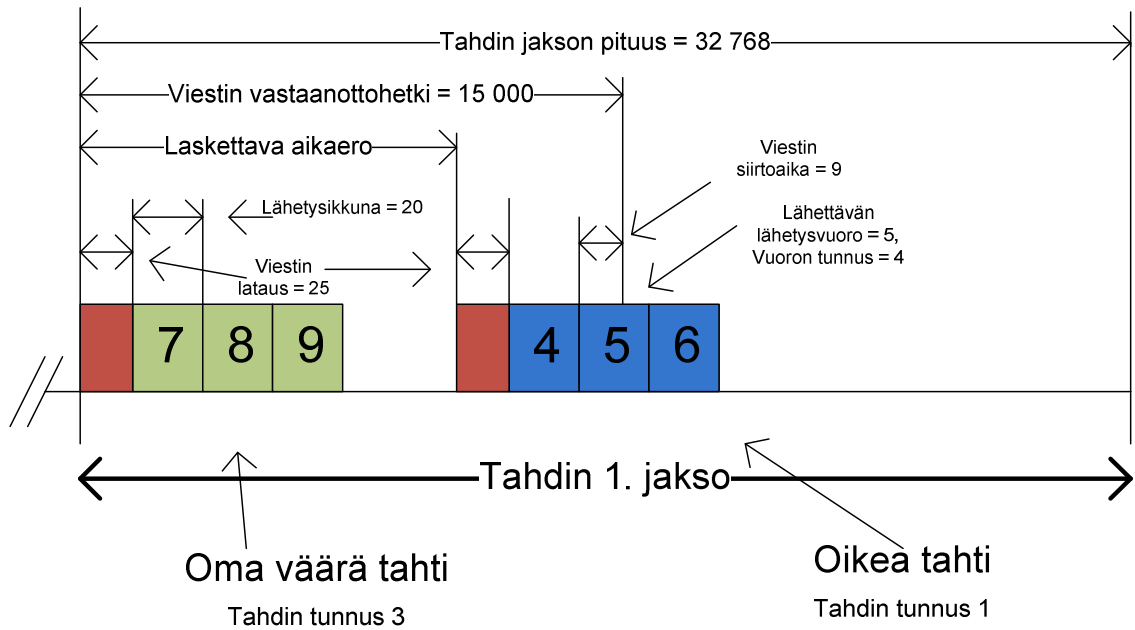
Prosessori tarkistaa ja mahdollisesti muuttaa tahdistusta vain ensimmäisen viestin mukaan, eli tekee hienosäätöä. Tässä tilassa myös lasketaan seuraavia lähetyksenvuoroja, pidetään kirjaa muiden laitteiden lähetyksenvuoroista, tarkistellaan vuorojen päällekkäisyyksiä sekä kerätään mahdollisesti muiden laitteiden lähettämiä mittautustietoja. Lopuksi vielä asetetaan yksi vertailukeskeytyks, jonka ajankohta on riippuvainen mahdollisesta laskurin tahdin muuttamisesta. Tässä keskeytyksessä asetetaan lopuksi pitkä odotusaika, jonka jälkeen uusi tahdin alku alkaa. Tilakoneen toiminta on esitettyä kuvion 17 tilakaaviossa.



Kuvio 17. Tilakaavio radiopiirin vertailukeskeytyksen vaiheista.

### 5.4.2 Tahdin laskeminen

nRF24LE1-radiopiirin mikrokontrollerin sekunti-laskurista löytyy ominaisuus, jolla laskurin aika voidaan kaapata erittäin tarkasti viestin vastaanottohetkestä. Tätä ominaisuutta on käytetty selvittämään se, millä oman laskurin ajankohdalla saapuvat viestit ovat vastaanotettu. Viestien vastaanottopuskureita on kolme, joten radiopiiri pystyy vastaanottamaan kolme viestiä siten, että mikrokontrolleri ei niitä vielä siirrä muistiin. Tämän takia yhden tahdin vuoroikkunoiden määrä on jätetty kolmeen. Lähettäjä voi lähettää viestin kolmessa eri ikkunassa tahdin aikana, siksi sen on laskettava tahdin aikaero käyttäen hyväksi lähettäjän lähettämää lähetysvuoronumeroa. Jokaiselle laitteelle on tallennettu muistiin kiinteät ajat viestin lataamiselle, ikkunan koolle, viestin siirtoajalle sekä tahdin jakson pituudelle. Näitä kiinteitä arvoja, viestin saapumisajankohtaa sekä lähetysvuoronumeroa hyväksi käyttämällä vastaanottaja pystyy laskemaan, mikä on tarkka tahdin aikaero ja siirtämään oman laskurin arvoa aikaeron mukaisesti, jolloin tahti muuttuu samaksi. Kuvion 18 esimerkkitalanteesta selviää eri vaiheiden aikojen suhde kokonaisuutenaan. Kun yksi aikayksikkö on noin  $30,5 \mu\text{s}$ , vastaa  $32\,768$  aikayksikköä noin yhtä sekuntia.



Kuvio 18. Esimerkkutilanne oikean tahdin löytämisestä.

Kuvion 18 mukaisen tilanteen esimerkkilasku kaavaa 1 käyttäen

$$\Delta t = 15\,000 - (25 + ((5 - 4) * 20) + 9)) = 14\,946$$

Kaavan esimerkkilaskusta saatu aikaero summataan tai vähennetään tahdin jakson pituuden kanssa riippuen sen suuruudesta. Tahdin alkamisajankohdan keskeytyks alkaa seuraavassa jaksossa samaan aikaan kuin viestin lähettäjän tahti. Kun laite on vastaanottanut muutamia viestejä oikealla tahdilla, niin laite olettaa olevansa oikeassa tahdissa ja lopettaa muiden tahtien kuuntelemisen. Muita tahteja kuitenkin kuunnellaan satunnaisen ajan välein, että löydettäisiin mahdollisesti uusi parempi tahti.

### 5.4.3 Eri verkkojen etsiminen

Jokainen laite pitää kirjaa tahdin tunnuksesta, joka käynnistyksen yhteydessä alustetaan samaksi kuin oma yksilöllinen tunnus. Viestiä vastaanottaessa ja tahtia

asettaessa laite vertaa vastaanotetusta viestistä saapuvaa tahdin tunnusta omaan tunnuksensa. Jos saapuva tunnus on pienempi kuin oma, niin tämä pienempi tunnus tallennetaan. Jokainen laite lähettää aina viestissään oman tahdin tunnuksensa. Tämä tunnus on verkossa kuitenkin vain luku, joka ideaalitalanteessa on aina sama kuin verkon pienimmän tunnuksen omaavan laitteen tunnus. Toimintoon on vielä varmuudeksi lisätty ajastin, jonka avulla laite pääsee uuteen tahtiin, jos oikean tahdin laitteet katoavat. Uutta tahtia etsittäessä laite alustaa oman tahdin tunnuksen arvoksi 255, jotta se voisi tahdistaa kaikkiin mahdollisiin tahteihin ja etsiä niistä pienintä tahdin tunnusta. Näin saadaan verkosta itsensä korjaava.

#### **5.4.4 Virransäästö**

Kun laite olettaa olevansa tahdissa eikä sen enää tarvitse etsiä jatkuvasti uusia tahteja, aloittaa laite pienivirtaisen tilan. Laite siis käynnistää radioyksikön vastaanottotilan juuri ennen ensimmäisen ikkunan alkamista ja sammuttaa sen heti viimeisen ikkunan loputtua. Tällä saadaan turha virrankulutus pois. Saapuvan viestin käsittely ja tahdin laskeminen suoritetaan heti viimeisen lähetysikkunan jälkeen, mutta tässä vaiheessa päällä on ainoastaan mikrokontrolleri. Kun laskut on laskettu ja viestit luettu, kytkee mikrokontrolleri kaikki ulkoiset IO-nastat pois päältä, lukuun ottamatta kellokiteen nastoja. Tämän jälkeen se sammuttaa kaikki mahdolliset toiminnot ja jättää ainoastaan sekuntilaskurin ja sen vertailukeskeytyksen seuraamisen päälle. nRF24LE1-radiopiirillä on todella pieni virrankulutus tilassa, jossa ainoastaan sekuntilaskuri on käynnissä.

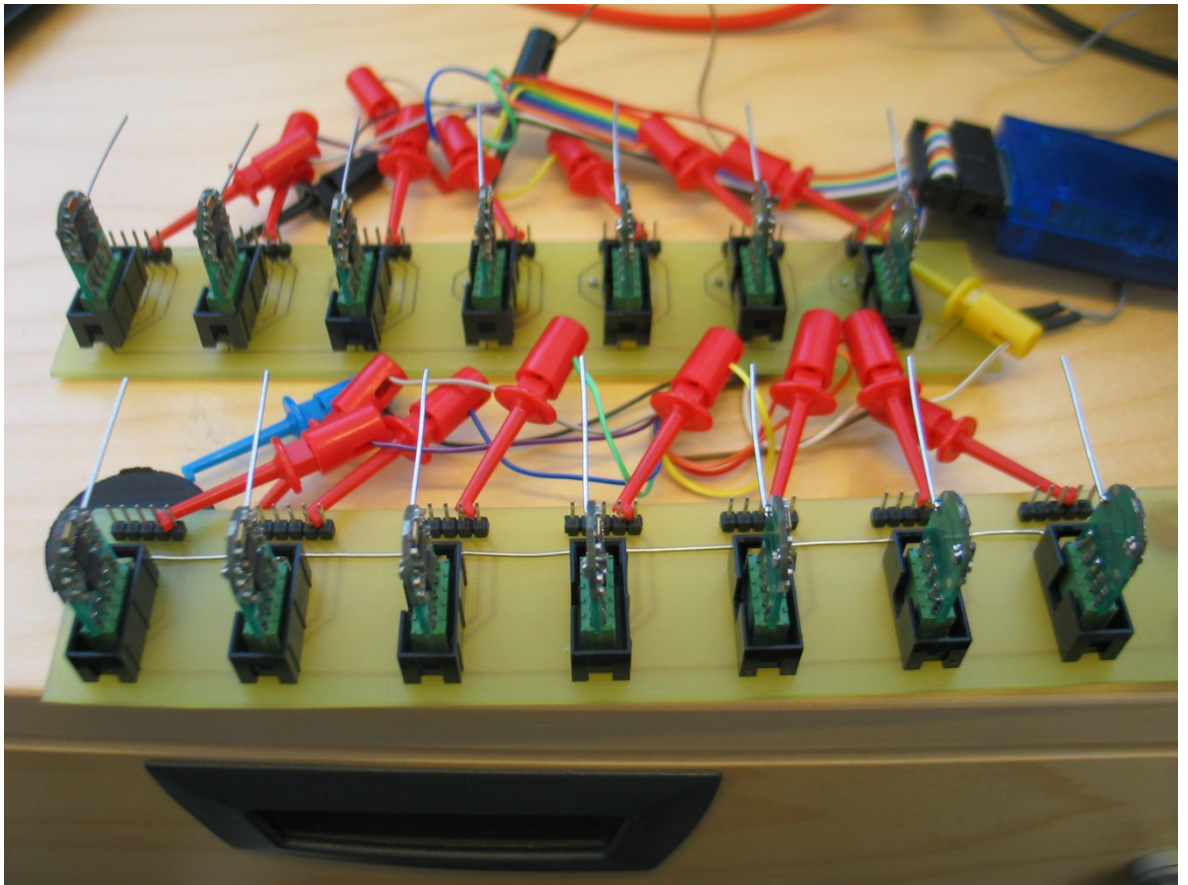
Virransäästöä mitattaessa on mittauslaitteistona käytetty Fluken yleismittarin 289 mallia, joka pystyy lukemaan mikroampeerin virtoja 0,06 % tarkkuudella (Fluke 2009). Virrankulutuksen profiilia mitattaessa on työhön valmistettu oma pieni vahvistinkytkentä, jolla voidaan mitata virrankulutuksien suhteita ohjelmakoodin eri vaiheilla. Vahvistinkytkentä on liitteessä 4. Periaatteessa kytkennällä pystyttäisiin mittaamaan myös todelliset virrat eri vaiheissa, mutta kytkennän ja komponenttien tarkkuus voi erittäin pienillä virroilla vaihdella. Mittaustiedot on luettu vahvistinkytk-

kennästä Tektronix TDS210 -oskilloskoopilla, josta ne on kaapattu WaveStar-ohjelmalla Exceliin.

## 5.5 Testaus

Ohjelmointiympäristön ja USB-tikun toimintoja on voitu testata käyttäen apuna USB-tikulle ohjelmoituja testauskomentoja. Näillä komennoilla on pystytty seuraamaan ohjelmakoodin siirtymistä radiopiirin mikrokontrollerin muistiin, sekä muistin tyhjentämistä. USB-tikun laskeman CRC-tarkistussumman toimivuus on testattu muokkaamalla lähetettävän ohjelmakoodin tietoja siten että CRC-summa ei ole paikkansapitävä. Tällaisessa tilanteessa USB-tikku lopettaa ohjelmoinnin ja ilmoittaa tapahtuneesta CRC-virheestä.

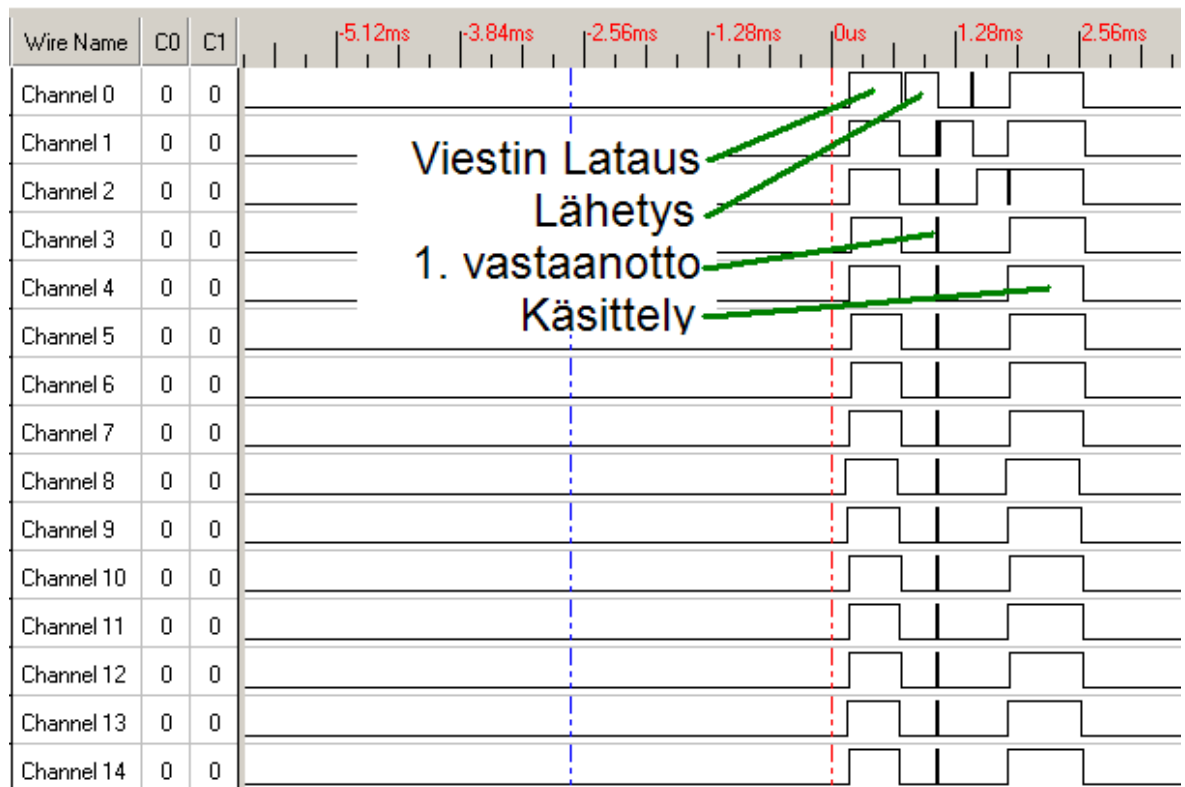
Tahdistusta on testattu käyttämällä täysin identtisiä laitteita. Kuviossa 19 olevat laitteet ovat olleet kiinnitettynä niille valmistettuihin telineisiin, että välttyttäisiin laitteiston virheiltä. Jos jokainen laite olisi esimerkiksi pöydällä kaikkien johtojen ja työkalujen seassa olisivat oikosulut ja komponenttien irtoaminen mahdollista. Telineiden avulla varmistettiin myös se, että jokainen laite saa varmasti kaikilta viestien, sillä liian kaukana toisistaan olevat laitteet eivät välttämättä pysty kommunikoimaan keskenään. Myös yhden virtalähteen käyttö varmisti sen, että kaikki laitteet saavat samaa jännitettä. Kuviossa 19 on kehityksessä käytetty tilanne, jossa 14 laitetta on kiinni niille tehdyissä telineissä. Logiikka-analysaattorin johdot ovat kytketty jokaiseen laitteeseen



Kuvio 19. Kuva testauslaitteistosta.

Testauksessa laitteita on ollut käytössä 14 kappaletta, joissa jokaisessa on sama ohjelmakoodi lukuun ottamatta jokaisen laitteen yksilöllistä tunnusta. Nämä tunnukset ovat kiinteästi ohjelmoituna radiopiirin mikrokontrollerin ohjelmamuistiin, joten ne eivät voi sekaantua. Jokaiseen laitteeseen on jouduttu ohjelmoimaan yhden IO-nastan ohjaus sen mukaan, missä ohjelmakoodin vaiheessa koodin suoritus menee, koska laitteissa ei ole mitään näyttöä eikä sarjamuotoinenkaan tiedon siirto ole järkevää sen hitauden takia. Jokaisen laitteen yksi IO-nasta on ollut kytkettynä tietokoneen USB-porttiin asennettavaan 16-kanavaiseen logiikka-analysaattoriin. Tämän avulla tietokoneen näytöltä on pystytty seuraamaan jokaisen laitteen tahtia ja vaiheita sekä hienosäätämään tahdistuksen laskutoimituksia. Kuviossa 20 on kuvankaappaus logiikka-analysaattorin ohjelmasta, josta näkee jokaisen laitteen olevan samassa tahdissa.





Kuvio 20. Kuvankaappaus logiikka-analysaattorin PC-ohjelmasta.

Itse tahdistusta ja sen ominaisuuksia on testattu seuraavilla tavoilla

- jokaisen laitteen yhtäaikainen käynnistäminen
- yksittäisten laitteiden sammuttaminen ja käynnistäminen
- kahden eri tahdin luominen ja näiden yhdistäminen
- tahdin tunnuksen seuranta ja merkittävien laitteiden poisto verkosta
- tahdistuksen testaus jakson eri pituuksilla
- laitteiden pienivirtaisen tilan virrankulutus.

Jokainen edellä mainittu ominaisuus on kehitetty testausympäristössä ja lopullisessa versiossa on kaikkia laitetta testattu yhdessä. Jokainen ominaisuus on testattu ja tutkittu niin kauan, kunnes mistään ei ole löytynyt virheitä.

## 6 TULOKSET JA ARVIOINTI

### 6.1 Tulokset

Tässä osiossa on ensin kerrottu ohjelmointiympäristön tuloksia, jossa kerrotaan myös työssä suunnitellun piirilevyn toimivuudesta. Lisäksi kerrotaan tahdistuksen toimivuudesta ja mitatuista virrankulutusarvoista. Lopuksi esitellään erilaisten optimointien vaikutusta virrankulutukseen ja akun kapasiteettiin.

#### 6.1.1 Ohjelmointiympäristö

Työssä tulokseksi saatiin toimiva ohjelmointiympäristö nRF24LE1-radiopiirille. Ohjelmointiympäristö on tavoitteidensa mukaisesti ilmainen, lisäksi käytettävät laitteistot ja komponentit ovat Seinäjoen ammattikorkeakoululta saatavissa helposti. Tahdistuksen kehityksen aikana on samalla jouduttu käyttämään työn alussa kehitettyä ohjelmointiympäristöä, jonka luotettavuus on todistettu. Ohjelmointiympäristön ohjeet ja materiaalit ovat Internetissä muiden ladattavissa, ja sen englanninkielisestä käännöksestä on joillain alan keskustelusivustoilla oltu kiinnostuneita.

Piirilevyn suunnittelussa ja toteutuksessa on onnistuttu erittäin hyvin. Piirilevystä on saatu pienikokoinen, sekä hyvin muokattavissa moniin sovelluksiin. Piirilevyä on otettu käyttöön jo muissakin Seinäjoen ammattikorkeakoulun projekteissa, sillä käyttöönotto on nopeaa ja helppoa. Virheitä piirilevyn kytkennästä ei ole löytynyt.

#### 6.1.2 Tahdistus ja virrankulutus

Vaikka GENSEN-hankkeelle ei tässä työssä vielä valmista laitteistoa ollut tarkoituskaan kehittää, on työssä saatu kuitenkin toimiva ja todella monikäyttöinen pohja laitteiston ja tahdistuksen osalta. Tahdistuksen avulla on päästy todella pitkiin ak-

kujen kestoihin, ja tahdistuksen jakson pituuden muuttamisella voidaan akkujen kestoa ja tiedonsiirron nopeus säädellä sopivaksi jokaiseen käyttökohteeseen.

Tahdistuksessa on päästy odotettua parempiin lopputuloksiin, koska radiopiirin ominaisuudet tukivat hyvin tahdistukseen tarvittavia toimintoja. Virrankulutuksen eri vaiheet saatiin tarkasti mitattua erilaisilla laitteistoilla ja kytkennöillä. Kuviossa 21 on kuvaaja radiopiirin aktiivisen ajan virrankulutusprofiilista ja sen vaiheista. Näissä testaustuloksissa on profiilia mitattaessa mikrokontrollerille laitettu tehtäväksi mittaustietojen lukemista ja käsittelyä. Työn tuloksissa on haluttu, että laite käsittelee paljon mittaustietoja, jotta profiilista saadaan todenmukainen. Tahdistuksen eri vaiheiden ajat ovat myös hieman pienintä mahdollista isommat, koska halutaan tahdistuksen toimivan varmemmin.

Kuvion 21 profiilista voidaan virrankulutuksen mukaan lukea seuraavia vaiheita:

1. 0,5 - 1,5 ms. Mikrokontrolleri käynnistää sisäisen 16 MHz: kiteen, jonka avulla se käynnistää ulkoisen 16 MHz: kiteen.
2. 1,5 - 2,0 ms. Ulkoisen kiteen käynnistäminen.
3. 2,0 - 2,5 ms. Ohjelmakoodin suoritus alkaa keskeytyksestä ja viestin siirto lähetysohjelmaan alkaa.
4. 2,5 - 5,0 ms. Vastaanotto-tila aktiivisena.
5. 5,0 - 9,5 ms. Mikrokontrolleri käsittelee saapuvaa viestiä ja tallentaa tiedot muistiin.
6. 9,5 - 10 ms. Mikrokontrolleri palaa virransäästötilaan.
7. Esimerkki virtaprofiilista kun laitteella on lähetysohjelma. Virtapudotus tulee kun laite muuttaa vastaanotto-tilansa lähetysohjelmaan.

Tahdin jakson pituuden ollessa kaksi sekuntia, niin aktiivisen vastaanotto-tilan käytösuhde on 0,125 %.

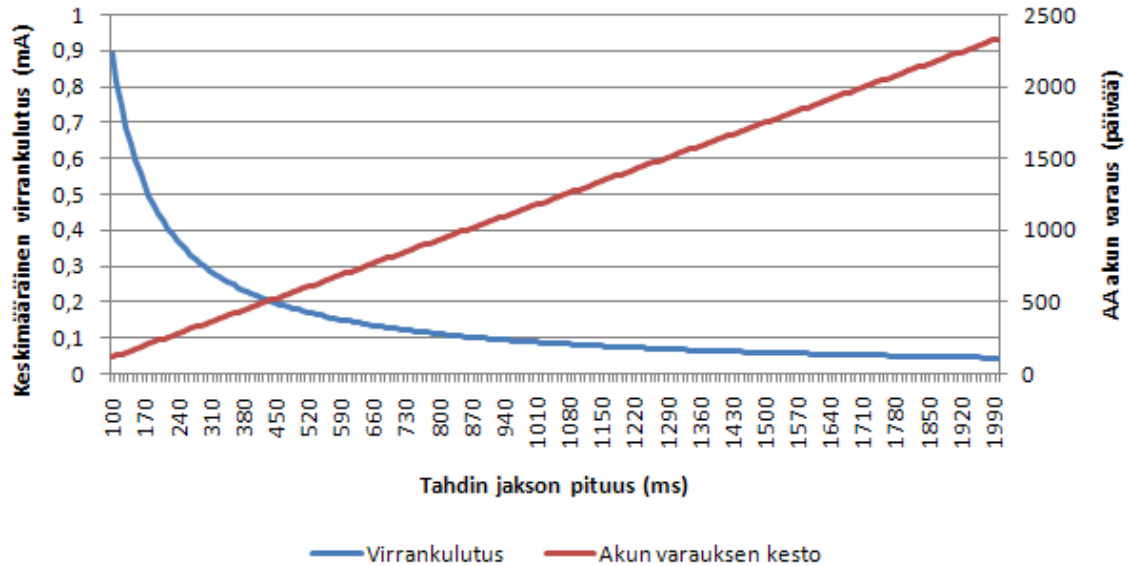


Kuvio 21. Kuvaaja prosessorin aktiiviajan virrankulutuksesta.

Kaikki profiilin virrankulutustilat ovat yhteneviä myös laitteen valmistajan antamiin virrankulutustietoihin. Profiilista ei pysty tarkasti tulkitsemaan virransäästötilan virrankulutusta, mutta Fluken 289-yleismittarilla on virrankulutukseksi saatu mitattua noin  $3 \mu\text{A}$ :n verran. Tämä virransäästötilan mitattu virrankulutus on sama kuin valmistajan ilmoittama.

Virrankulutus on yksittäisissä projekteissa piirin aktiiviajalla aina sama. Mutta laitteiden jokaiseen vaiheeseen kuluva aika voidaan muuttaa halutun mittaiseksi. Optimoimalla kaikkien vaiheiden ajat pienimmäksi mahdolliseksi, voidaan keskimääräistä virrankulutusta pienentää. Lisäksi tahdistuksen virransäästötilaa eli tahdin jakson pituutta kasvattamalla saadaan virrankulutusta pienemmäksi. Tämä tietysti vaikuttaa myös tiedonsiirron nopeuteen hidastavasti. Kuviossa 22 on esitetty tahdin jakson pituuden vaikutusta keskimääräiseen virrankulutukseen ja AA-akun varauksen keston.

## Tahdin jakson pituuden vaikutus keskimääräiseen virrankulutukseen



Kuvio 22. Keskimääräisen virrankulutuksen ja akun varauksen suhde jakson pituuteen.

Taulukoissa 2 ja 3 on laskettu esimerkkinä akkujen kapasiteetin kestoja. Ensimmäisessä taulukossa on esimerkki tässä testauksessa käytettävillä tahdin asetuksilla ja jälkimmäisessä taulukossa on kaikki laitteen aktiivivaiheita optimoituina pienimpään mahdolliseen. Lisäksi laite käsittelee paljon tietoja mm. kiihtyvyyssanturin mittaustietojen lukemista, sekä paikannukseen liittyvän tiedon tallentamista ja laskemista. Taulukon laskuissa on akkuna käytetty kahta tavallista AA-akkua, joiden nimellinen virtakapasiteetti on 2500 mAh. Tällöin voidaan havaita tahdistuksen pienen muutoksen vaikutus akkujen varauksen kesto. Käytännössä akkujen varaus pienenee itsestään vuosien mittaan, mutta sitä ei ole huomioitu taulukoiden laskuissa. Paristoja tarvitaan kaksi, koska yksi paristo tuottaa 1.5 V jännitteen ja kaksi sarjankytkettyinä 3.0 V. Radiopiiri tarvitsee käyttöjännitteekseen vähintään 1.9 V.

Taulukko 2. Tahdistuksen virrankulutuksen laskuesimerkki käytännön tilanteessa.

<b>Vähennetty optimointi ja lisätty tietojen käsittely</b>			
<b>Vaihe</b>	<b>Virta (mA)</b>	<b>Aika (ms)</b>	<b>Kulutus (mA * ms)</b>
1	3,25	1	3,25
2	2,3	0,5	1,15
3	5,8	0,5	2,9
4	21	2,5	52,5
5	5,9	4,5	26,55
6	0,003	991	2,973
<b>Summa</b>	<b>38,253</b>	<b>1000</b>	<b>89,323</b>
<b>Keskimääräinen virrankulutus (mA)</b> 0,089323 <b>AA akun nimellinen kapasiteetti (mAh)</b> 2500 <b>Laitteen käyttöaika (2 x 1.5 V AA akku)</b> 27988 tuntia 1166 päivää 3,2 vuotta			

Taulukko 3. Tahdistuksen virrankulutuksen laskuesimerkki optimoiduilla tahdistuksen arvoilla.

<b>Optimoitu tahdistus ilman tietojen käsittelyä</b>			
<b>Vaihe</b>	<b>Virta (mA)</b>	<b>Aika (ms)</b>	<b>Kulutus (mA * ms)</b>
1	2,4	1	2,4
2	1,4	1	1,4
3	4,8	0,5	2,4
4	18,6	1,12	20,832
5	4,2	1	4,2
6	0,003	991	2,973
<b>Summa</b>	<b>31,403</b>	<b>995,62</b>	<b>34,205</b>
<b>Keskimääräinen virrankulutus (mA)</b> 0,034355 <b>AA akun nimellinen kapasiteetti (mAh)</b> 2500 <b>Laitteen käyttöaika (2 x 1.5 V AA akku)</b> 72769 tuntia 3032 päivää 8,3 vuotta			

## 6.2 Arviointi

Tässä osiossa arvioidaan työn suoritusta ja tuloksia. Lisäksi kerrotaan ohjelmointiympäristön jatkokehitysmahdollisuuksista. Lopuksi mainitaan muutamia käyttökohteita ja mahdollisuuksia työssä käytetyille laitteistolle sekä kehitetylle tahdistukselle.

### 6.2.1 Työ

Työn lähtökohtia ja tavoitteita ajatellen tässä työssä on onnistuttu hyvin. Työ oli yleisesti haastava ja vaati hyvät pohjatiedot ennen projektin aloittamista. Työssä vaadittiin monien ohjelmointikielien hallitsemista, sekä paljon elektroniikan tunteista. Tämä teki työstä mielenkiintoista, koska yhden asian kehittäminen mahdollisti uuden asian aloittamisen.

Nordic Semiconductorin valmistamalla aloituspaketilla on päästy hyvin alkuun projektissa ja saatu tietoa tarvittavista kytkennöistä ja rajapinnoista. Ohjelmointiympäristö ei siis vaatinut aloituspaketin lisäksi mitään muita hankintoja.

Tahdistuksen kehittäminen on ollut todella opettavaista. Monimutkaisen tahdistuksen toteutus on pakottanut käyttämään mikrokontrollerin ominaisuuksia tehokkaasti. Apua ongelmiin on jouduttu kysymään valmistajalta ja yleisiltä keskusteluvustoilta. Pienien osien testaus on työlästä ja joskus suuren verkon testaamisessa on hankala paikantaa virheitä. Työn lopputuloksien perusteella tämän työn tekijä on työstään tyytyväinen, sillä uuden laitteen käyttö monimutkaisissa soveluksissa on usein vaikeaa.

### 6.2.2 Jatkokehitys

Ohjelmointiympäristön työn tuloksilla on jatkossa todella hyvät kehittämismahdollisuudet. LE1-programmer-ohjelmaa on mahdollisuus kehittää toimivaksi myös

graafisella käyttöliittymällä, jolloin sillä voisi selkeämmin ohjelmoida erilaisia ase-  
tuksia ja säädellä sarjaportin nopeuksia. Ohjelmaan voisi jatkossa miettiä myös  
ohjelmakoodin lukemista radiopiiriltä tietokoneelle esimerkiksi Intel HEX  
-tiedostomuotoon.

Tässä työssä suunniteltu piirilevy antaa hyvät mahdollisuudet moniin eri sovelluk-  
siin. Radiopiiri on niin edistysellinen, että se tulee varmasti olemaan tulevaisuu-  
dessa monessa projektissa mukana, joten sille suunniteltu piirilevy ja ilmainen oh-  
jelmointiympäristö ovat tarpeen.

Tahdistusta ei itsessään voi kovin pitkälle enää kehittää. Kaikki toimivaan tahdis-  
tukseen tarvittavat toiminnot on tehty. Tahdistus on kuitenkin suunniteltu siten, että  
sitä voidaan pienillä muokkauksilla käyttää myös monissa muissa mikrokontrolle-  
reissa ja myös eri sovelluksissa. Tässä työssä tehty tahdistus ei välttämättä tarvit-  
se olla langattoman verkon käytössä, vaan sitä voidaan soveltaa vaikkapa yhden  
laajan sarjaväylän laitteiden väliseen vuorotteluun. Toimivan tahdistuksen pohjalle  
on hyvä kehittää jatkossa esimerkiksi mittausanturiverkkoja, joita on tarkoitus  
GENSEN-hankeessakin toteuttaa.

### **6.2.3 Käyttömahdollisuudet**

Tässä työssä kehitetylle laitteelle on jo sellaisenaan monia hyviä käyttökohteita.  
Laitteen pieni koko ja monipuolinen mikrokontrolleri mahdollistaa laajan muokatta-  
vuuden räätälöitäviin sovelluksiin. Työssä kehitetty tahdistus lisää monien laittei-  
den käytön samassa verkossa sekä kasvattaa laitteen käyttöaikaa.

Vaikka GENSEN-hankkeelle tästä työstä on jo paljon käyttökohteita, niin voidaan  
tätä sovellusta käyttää esimerkiksi seuraaviin tarkoituksiin

- lasten valvonta päiväkodissa
- vuokrausjärjestelmä



- pelit ja leikit
  - jääkiekon kiihtyvyys
  - keilojen asento
  - piiloleikit
- golf-radan valvonta
- älyranneke
  - tilaisuuksien pääsylippu
  - maksuväline.

Lisäksi laite voidaan kytkeä osaksi suurempaa kokonaisuutta, esimerkiksi rajapintana pienemmän anturiverkon ja tietoliikenneverkon välillä.

## LÄHTEET

- Atmel Corporation. 2008. AT90USB82/162 datasheet. [Pdf-julkaisu]. USA: San Jose. [Viitattu 10.5.2009]. Saatavissa: [http://www.atmel.com/dyn/resources/prod\\_documents/doc7707.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc7707.pdf)
- Barr, M. 2000. Assembly vs. C. [www-julkaisu]. Netrino. [Viitattu 10.5.2009]. Saatavissa: <http://www.netrino.com/Embedded-Systems/How-To/Assembly-vs-C-Comparison>.
- Basavaraju, T.G., Puttamadappa, C. & Sarkar S.K. 2008. Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications. Boca Raton: Auerbach Publications.
- Context. 2007. Text Editor Features. [www-julkaisu]. UK: Coventry. [Viitattu 10.5.2009]. Saatavissa: <http://www.contexteditor.org/features.html>.
- Fluke. 2009. Fluke 289 True-rms Industrial Logging Multimeter with TrendCapture. [www-julkaisu]. [Viitattu 26.10.2009]. Saatavissa: <http://us.fluke.com/usen/products/Fluke+289.htm?&trck=289specifications>.
- GENSEN. 12.8.2009. Research Plan. Generic Sensor Network Architecture for Wireless Automation. [Hanke-ehdotus].
- Intel. 6.1.1988. Hexadecimal Object File Format Specification. [Pdf-julkaisu]. [Viitattu 27.10.2009]. Saatavissa: <http://microsym.com/editor/assets/intelhex.pdf>.
- Kalinsky, D. & Kalinsky, R. 1.2.2002. Introduction to Serial Peripheral Interface. [www-julkaisu]. Embedded System Design. [Viitattu 10.5.2009]. Saatavissa: [http://embedded.com/columns/beginnerscorner/9900483?\\_requestid=466795](http://embedded.com/columns/beginnerscorner/9900483?_requestid=466795).
- Nordic Semiconductor. 2009. nRF24LE1 Product Specification. [Pdf-julkaisu]. Norway: Trondheim. [Viitattu 8.5.2009]. Saatavissa: [http://www.nordicsemi.no/files/Product/data\\_sheet/nRF24LE1\\_Product\\_Spec\\_v1\\_2.pdf](http://www.nordicsemi.no/files/Product/data_sheet/nRF24LE1_Product_Spec_v1_2.pdf).
- Nordic Semiconductor. Product Brief nRFgo Development Kit for Nrf24LE1. [Pdf-julkaisu]. Norway: Trondheim. [Viitattu 10.5.2009]. Saatavissa: [http://www.semiconductorstore.com/pdf/newsite/nordic/Prod\\_brief\\_Devtools\\_nRFgo\\_DK\\_nRF24LE1](http://www.semiconductorstore.com/pdf/newsite/nordic/Prod_brief_Devtools_nRFgo_DK_nRF24LE1)

Pinouts.ru.19.10. 2005. USB pinout and wiring. [www-julkaisu]. [Viitattu 12.9.2009]. Saatavissa: [http://pinouts.ru/Slots/USB\\_pinout.shtml](http://pinouts.ru/Slots/USB_pinout.shtml).

Schoepke, E. 1.2006. Chip Antenna Layout Considerations for 802.11 Applications. [Pdf-julkaisu]. Johanson Technology. [Viitattu 20.10.2009]. Saatavana: [http://www.johansontechnology.com/images/stories/tech-notes/ip/JTI\\_Antenna-Mounting\\_2006-11.pdf](http://www.johansontechnology.com/images/stories/tech-notes/ip/JTI_Antenna-Mounting_2006-11.pdf).

SDCC. 10.5.2009. What is SDCC. [www-julkaisu]. [Viitattu 10.5.2009]. Saatavissa: <http://sdcc.sourceforge.net/index.php#Who>.

Tech-FAQ. [www-julkaisu]. What is USB. [Viitattu 10.5.2009]. Saatavissa: <http://www.tech-faq.com/usb.shtml>.

Waclawek, J. The unofficial history of 8051. [Pdf-julkaisu]. [Viitattu 26.10.2009]. Saatavissa: <http://www.efton.sk/t0t1/history8051.pdf>.

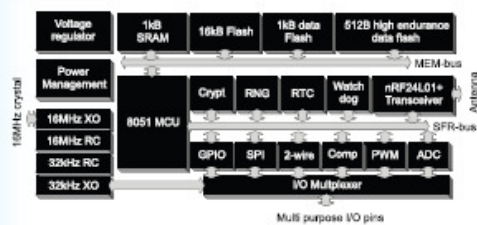
(Nordic Semiconductor [Viitattu 10.5.2009].)

### Single chip ultra low power wireless

#### Ultra low power wireless system-on-chip solution

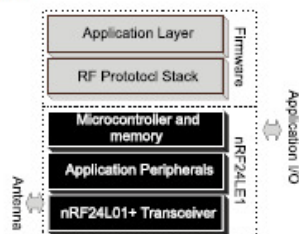
The nRF24LE1 is a unique solution offering a complete ultra low power (ULP) wireless system-on-chip (SoC) solution. It integrates the industry best nRF24L01+ 2.4GHz transceiver core, an enhanced 8051 microcontroller, flash memory and a wide range of analog and digital peripherals.

#### Block diagram



The 8-bit microcontroller is powerful enough to run both the RF protocol stack and the application layer, enabling a true single chip implementation of ULP wireless applications.

#### System diagram



### Optimized for ultra low power wireless

#### Advanced power management and on-chip peripherals

The nRF24LE1 is optimized to provide a single chip solution for ULP wireless applications. The combination of processing power, memory, low power oscillators, real-time counter, AES encryption accelerator, random generator, plus a range of power saving modes provides an ideal platform for implementation of RF Protocols. Benefits include tighter protocol timing, security, lower power consumption and improved co-existence performance. For the application layer the nRF24LE1 offers a rich set of peripherals including: SPI, 2-wire, UART, 8 to 12-bit ADC, PWM, and an ultra low power analog comparator for voltage level system wake-up.

### Three package sizes for different applications

#### Same core but different package and I/O count

The nRF24LE1 comes in three different package variants:

- An ultra compact 4x4mm 24-pin QFN (7 generic I/O pins)
- A compact 5x5mm 32-pin QFN (15 generic I/O pins)
- A 7x7mm 48-pin QFN (31 generic I/O pins)

The 4x4mm 24-pin QFN with 7 generic I/O pins is ideal for low I/O count applications where small size is key. Examples include wearable sports sensors and watches. The 5x5mm 32-pin QFN with 15 generic I/O pins is ideal for medium I/O count applications such as wireless mice, remote controls and toys. The 7x7mm 48-pin QFN with 31 generic I/O pins is for high I/O count products like wireless keyboards.



RF Silicon Software Reference Design Development Tools

### PRODUCT BRIEF

## nRF24LE1

nRF24LE1-F16Q24  
nRF24LE1-F16Q32  
nRF24LE1-F16Q48

### KEY FEATURES

- Fully featured ultra low power nRF24L01+ 2.4GHz transceiver core
- Worldwide 2.4GHz ISM band operation
- Enhanced ShockBurst™ hardware link layer
- 250 kbps, 1 Mbps and 2 Mbps on-air data rate options
- Air compatible with nRF24L01; nRF24L01+; nRF24LU1; and nRF2401A, -02, -E1 and -E2
- Low cost external  $\pm 60$ ppm 16MHz crystal
- Enhanced 8-bit 8051 compatible microcontroller
- 32-bit multiplication-division unit
- AES encryption/decryption accelerator
- 16 kbytes on-chip flash memory
- 1 kbyte on-chip data flash memory
- 512 bytes high-endurance data flash memory
- 1 kbytes on SRAM plus 256 bytes of IRAM
- Low power 16MHz crystal and RC oscillators
- Ultra low power 32kHz crystal and RC oscillators
- Flexible real-time counter and three 16-bit timers/counters
- Ultra low power analog comparator for system wake-up
- Rich set of digital interfaces including: SPI master/slave, 2-wire master/slave, and UART
- 2-channel PWM
- Programmable resolution ADC: 6, 8, 10, or 12-bits
- Random Number Generator based on thermal noise
- Supports the Nordic nRFProbe hardware debugger
- Programmable generic I/O pins
- Three package options:
  - 4x4mm 24-pin QFN (7 Generic I/O pins)
  - 5x5mm 32-pin QFN (15 Generic I/O pins)
  - 7x7mm 48-pin QFN (31 Generic I/O pins)

### APPLICATIONS

- PC peripherals – mice, keyboards and remotes
- Gaming controllers
- RF remote controls for consumer electronics devices such as set-top boxes, media players and TVs
- Sports and healthcare sensors
- Sport watches, bike computers, and gym equipment
- Remote controlled toys
- Active RFID

### A complete development platform

#### For firmware and hardware engineers

The nRF24LE1 is supported by a complete development platform enabling designers to easily develop hardware and firmware for the chip. The platform comprises two key elements: the nRFgo Starter Kit and the nRF24LE1 Development Kit. One of each is required to get started with nRF24LE1.



nRFgo Starter Kit

nRF24LE1 Development Kit

The nRFgo Starter Kit provides a generic development platform including motherboards with sockets for radio modules, and the nRFgo Studio evaluation PC application. The nRF24LE1 Development Kit comes in three versions: one for each package variant. These include the nRF24LE1 radio modules, complete Software Development Kit (SDK), and nRFProbe hardware debug support.

#### For more information

Please visit [www.nordicsemi.com](http://www.nordicsemi.com) for the complete product specification and more information about this or any other ULP wireless products.

#### About Nordic Semiconductor ASA

##### Ultra low power RF silicon solutions

Nordic Semiconductor is a fabless semiconductor company specializing in ULP short-range wireless communication. Nordic is a public company listed on the Norwegian stock exchange.

Nordic provides RF Silicon Solutions for ultra low power wireless including:

- Highly integrated RF silicon
- Sophisticated and flexible development tools
- Application specific communication software
- Complete reference designs

#### Worldwide office locations

##### Headquarter

Trondheim, Norway  
Telephone: +47 72 89 89 00  
[www.nordicsemi.com](http://www.nordicsemi.com)



### SPECIFICATIONS

Frequency band	2.4GHz ISM (2.40000 – 2.4835GHz)
On-air data rate	250 kbps, 1 Mbps or 2 Mbps
Modulation	GFSK
Channel bandwidth	1MHz for 250 kbps and 1 Mbps mode, 2MHz for 2 Mbps mode
RF channels	126
Output power	Programmable: 0, -6, -12 or -18 dBm
External crystal	16MHz ±60ppm
Hardware Link layer	Enhanced ShockBurst™
Microcontroller	Enhanced 8-bit 8051 compatible microcontroller <ul style="list-style-type: none"> <li>- Reduced instruction cycle time</li> <li>- Up to 16MHz operation</li> <li>- 32-bit multiplication/division unit</li> </ul>
Program Memory	16 kbytes flash (1k endurance)
Data memory	1 kbytes (1k endurance) plus 512 bytes high endurance flash
RAM	1 kbytes SRAM plus 256 bytes IRAM
Oscillators	16MHz crystal oscillator 16MHz RC oscillator 32kHz crystal oscillator 32kHz RC oscillator
Hardware Security	8 x 8 Gaolis field multiplier for AES acceleration
Digital I/O	Flexible general purpose data port <ul style="list-style-type: none"> <li>- Hardware SPI master/slave</li> <li>- 2-wire master/slave</li> <li>- UART</li> </ul>
Analog peripherals	Programmable resolution ADC 2-channel PWM Analog comparator Random Number Generator
General purpose I/O	7 for 24-pin 4x4mm variant 15 for 32-pin 5x5mm variant 31 for 48-pin 7x7mm variant
Hardware debuggers support	nRFProbe, System Navigator from First Silicon Solutions (FS2)
Voltage regulator	On-chip 1.9 to 3.6V operation
Package options	RoHS compliant 24-pin 4x4mm QFN (nRF24LE1-F16Q24) 32-pin 5x5mm QFN (nRF24LE1-F16Q32) 48-pin 7x7mm QFN (nRF24LE1-F16Q48)

### Related Products

nRF6700	nRFgo Starter Kit
nRF24LE1-F16Q24-DK	nRFgo compatible Development Kit for 4x4mm 24-pin nRF24LE1
nRF24LE1-F16Q32-DK	nRFgo compatible Development Kit for 5x5mm 32-pin nRF24LE1
nRF24LE1-F16Q48-DK	nRFgo compatible Development Kit for 7x7mm 48-pin nRF24LE1

Visit [www.nordicsemi.com](http://www.nordicsemi.com) for Nordic Semiconductor sales offices and distributors worldwide.

## LIITE 2: InfoPage-asetustiedosto

```
-----
# Program memory target
# 0x00 = Programming to unprotected memory area
# 0x01 = Programming to protected memory area

TARGET = 0x00
#-----

# Program InfoPage
# 0x00 = Disable InfoPage programming
# 0x01 = Enable InfoPage programming

INFOPAGE = 0x01
#-----

#####
|
# The InfoPage is a separate page (512bytes) of flash memory that contain Nordic system tuning parame-
# ters and the configurable options of the flash memory. Any changes to the flash memory configuration
# must be done by updating this page. The InfoPage content is as follows:

# In address 0x00 starts DSYS (Device system) register (32 bytes) and it is reserved for device use only.

#-----
# Number of unprotected pages: NUPP (page address of start of protected area)
# Name: NUPP
# Size: 1 byte
# Address: 0x20
# Comment: Read out to register FPCR during start up
# NUPP=0xFF: all pages are unprotected

NUPP = 0xFF
#-----

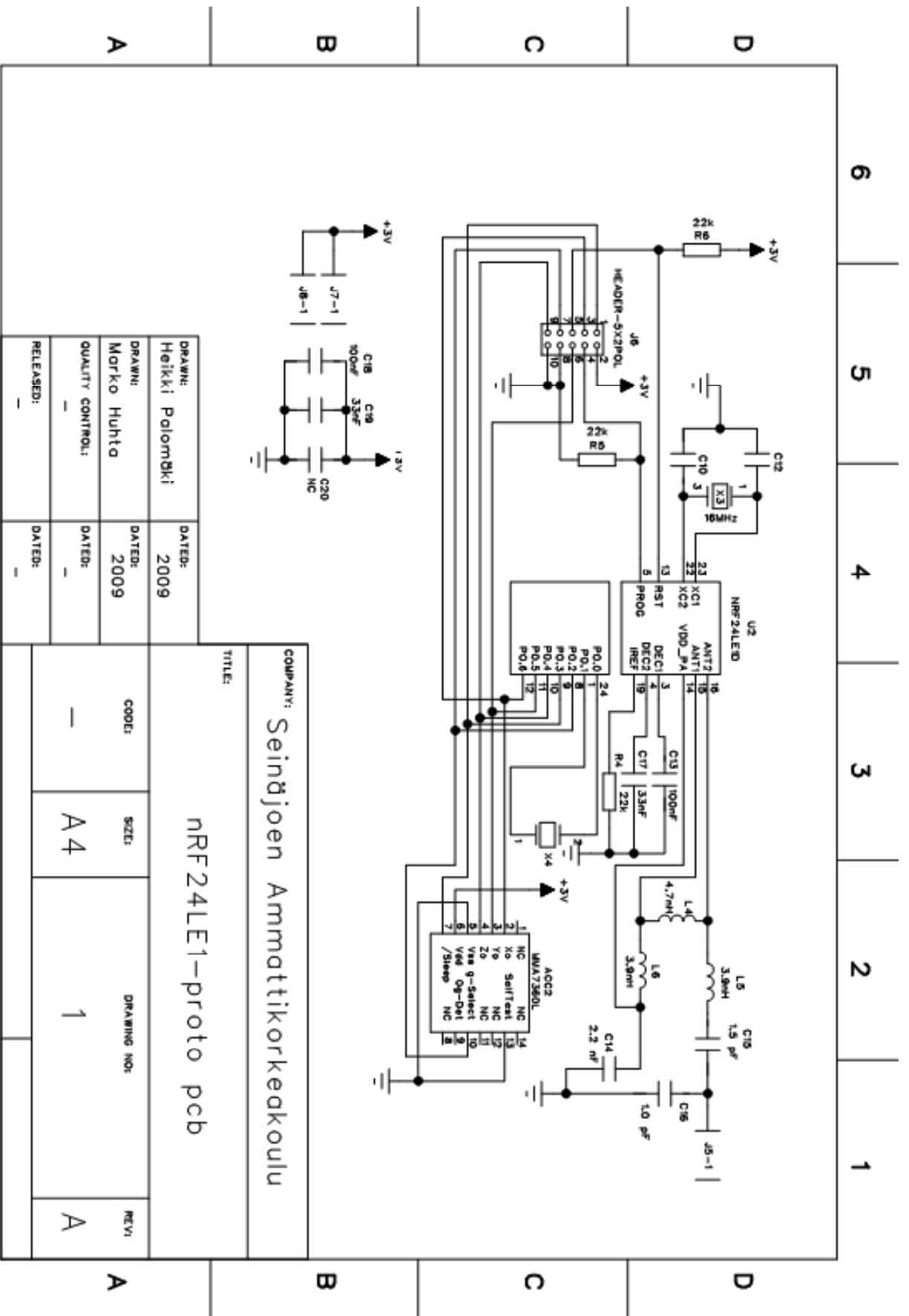
# Flash main block read back protect
# Name: RDISMB
# Size: 1 byte
# Address: 0x23
# Comment: Disable flash main block access from external interfaces (SPI, HW debug).
# Byte value:
# • 0xFF: Flash main block accessible from external interfaces
# • Other value: No read/erase/write of flash main block from external interfaces. Only read of info page
# Can only be changed once by SPI command RDISMB. Can only be reset by SPI command ERASE ALL

RDISMB = 0xFF
#-----

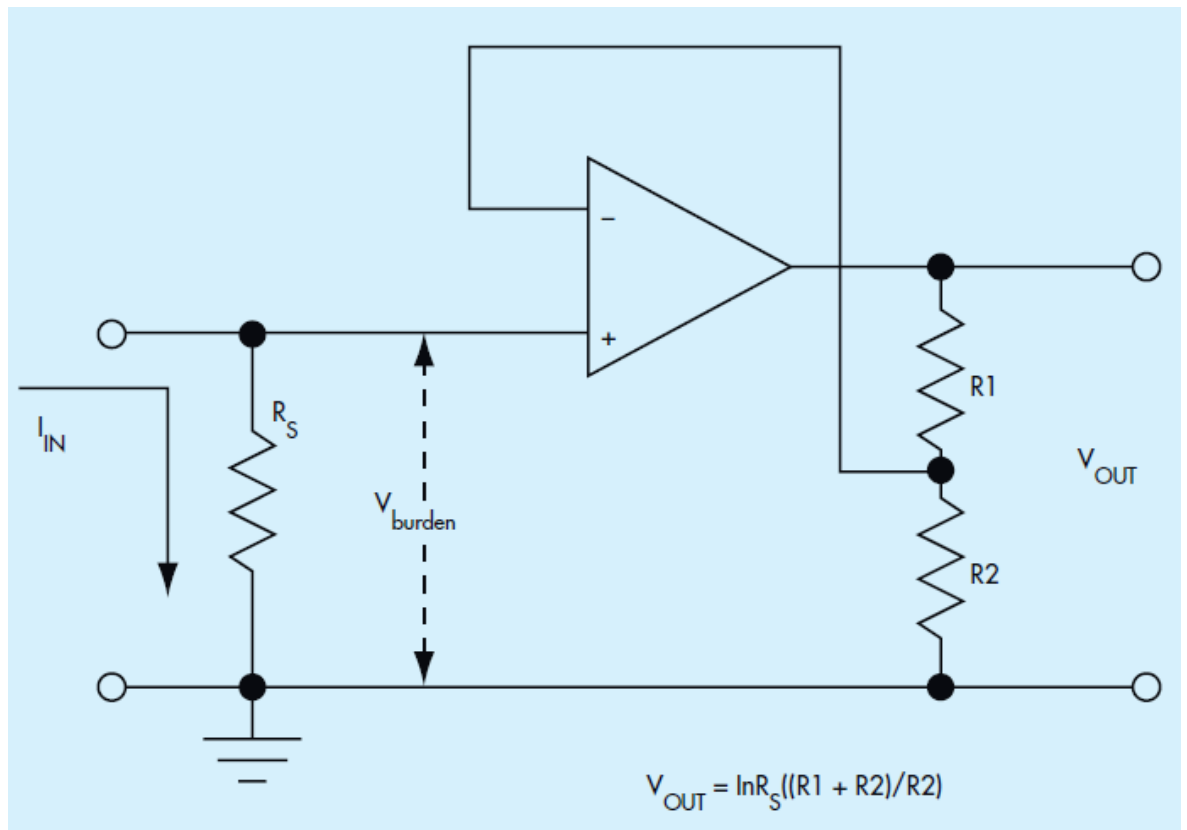
# Enable Hw debug
# Name: ENDEBUG
# Size: 1 byte
# Address: 0x24
# Comment: Enable on chip Hw debug features and JTAG interface.
# Byte value:
# • 0xFF: Hw debug features disabled
# • other value: Hw debug features and
# JTAG interface enabled

ENDEBUG = 0xFF
#-----
```

LIITE 3: Piirilevyn kytkentä



#### LIITE 4: Virtamittauskytkentä



LÄHDE:

Frenzel, L.E., 15.2.2007. Basics of design, Accurately measure nanoampere and picoampere currents. [Pdf-julkaisu]. Saatavissa: [http://electronicdesign.com/Files/29/14692/14692\\_01.pdf](http://electronicdesign.com/Files/29/14692/14692_01.pdf).