



Veikko Soininen

Matalaenerginen 433 MHz:n radiolinkki

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Elektroniikan koulutusohjelma
Insinöörityö
6.4.2011

Tekijä(t) Otsikko	Veikko Soininen Matalaenerginen 433 MHz:n radiolinkki
Sivumäärä Aika	31 sivua + 2 liitettä 6.4.2011
Tutkinto	insinööri (AMK)
Koulutusohjelma	elektroniikka
Suuntautumisvaihtoehto	elektroniikkasuunnittelu
Ohjaaja(t)	Teknologiapäällikkö Marko Mattila Lehtori Janne Mäntykoski
<p>Insinööriyössä on suunniteltu ja toteutettu edullinen, monikäyttöinen ja vähän energiaa kuluttava 433 MHz:n taajuutta käyttävä radiolinkki. Työ sisältää radiolinkin piirilevy-suunnittelun, valmistuksen, ohjelmoinnin ja mittaukset. Radiolinkin suunnittelussa ja osien valinnassa huomioitiin, että linkki mukailee ISO/IEC 18000-7 -standardia valikoiduin osin. Radiopiiri valittiin siten, että sen ominaisuudet vastaavat mahdollisimman hyvin käyttötarkoitusta, eikä ohjelmoinnilla jouduta toteuttamaan monimutkaisia alhaisen tason toimintoja. Radiopiirin ja mikrokontrollerin täytyi olla myös vähävirtaisia, jotta laitteen toimintaikä pienellä jännitelähteellä voisi olla kuukausia tai vuosia.</p> <p>Radiolinkki suunniteltiin liitettäväksi tietokoneen USB-väylään, jonka avulla sillä voidaan lähettää ja vastaanottaa tietoa toisilta linkeiltä. Linkin lähetysnopeus, -teho ja synkronointitavut ovat käyttäjän ohjelmoitavissa. Synkronointitavujen avulla voidaan määritellä, mitkä linkit keskustelevat keskenään ja kuulevat halutut lähetykset. Linkkiä voidaan käyttää esimerkiksi osana sensoriverkkoa, sekä tukiasemana, että asiakkaana. Radiolinkillä on useiden satojen metrien kantama vapaassa tilassa ja sisätilassakin kymmeniä tai satoja metrejä.</p> <p>Insinööriyön lopputulokseksi saatiin helposti mukautettava pienikokoinen radiolinkki, joka on hintansa vuoksi muokattavissa ja jatkokehitettävissä moneen sovelluskohteeseen.</p>	
Avainsanat	433 MHz, radiolinkki, vähävirtainen

Author(s) Title	Veikko Soininen Low power 433 MHz radio link
Number of Pages Date	31 pages + 2 appendices 6 April 2011
Degree	Bachelor of Engineering
Degree Programme	Electronics
Specialisation option	Electronics Design
Instructor(s)	Marko Mattila, Chief Technology Officer Janne Mäntykoski, Lecturer
<p>The aim of this final year project was to design and accomplish an inexpensive, versatile and low power radio link which uses 433 MHz band. The project includes printed circuit board design, assembly, programming and measurements. The design of the radio link was carried out by following the essential parts of the ISO/IEC 18000-7 standard. The radio chip was chosen to provide as many of the needed features as possible which leads to easier programming as it is not necessary to implement all of the low level functions in software. The radio chip and microcontroller was designed to use minimal power so the device can be operational using small battery for months or even years if needed.</p> <p>The radio link was designed to be connected to a USB port which can be used to transmit and receive data and to change the settings. The transmit power, bitrate and synchronization bytes can be programmed by the user. Moreover, the synchronization bytes can be used to configure which links can communicate together. With small customizations, the link can be used as part of the sensor networks, as a base station and as a tag and it has hundreds of meters of range in free air.</p> <p>This final project was successful in creating a small sized and versatile radio link. It is inexpensive and easily customizable to many applications.</p>	
Keywords	433 MHz, radio link, low power

Sisällys

Tiivistelmä

Abstract

Sisällys

Lyhenteet ja käsitteet

1 Johdanto	1
2 Radiolinkin tekniset määrittelyt	1
2.1 Radiolinkin suunnittelun lähtökohdat	1
2.2 433 MHz:n ISM-taajuus	3
2.3 FSK-modulaatio	4
3 Mikrokontrollerin kommunikointiväylät	5
3.1 Serial Peripheral Interface -väylä	5
3.2 Universal Asynchronous Receiver/Transmitter -väylä	6
4 Radiolinkin komponentit	7
4.1 Radiolinkin lohkoakaavio	7
4.2 Radiopiiri	8
4.3 Mikrokontrolleri	10
4.4 USB-UART-muunnin	10
4.5 SAW-suodatin	11
4.6 Sovituspiirit	12
5 Piirilevyn suunnittelu	14
5.1 Piirilevyn kytkentäkaavio	14
5.2 Piirilevyn osasijoittelu	15
5.3 Piirilevyn toteutus ja kotelointi	16
6 Mikroprosessorin ohjelma ja käyttöliittymä	18
6.1 USB-UART-protokolla	18
6.2 Mikrokontrollerin ohjelmointi	19
6.2.1 Ohjelman rakenne	19
6.2.2 Lähdekooditiedostot ja funktiot	21
6.2.3 Radiopiirin toiminta	23
7 Mittaustulokset	26

7.1	Kantamamittaukset	26
7.2	Virrankulutusmittaukset	27
8	Yhteenveto	30
	Lähteet	31
	Liitteet	
	Liite 1. Piirilevyn kytkentäkaavio	
	Liite 2. Piirilevyn osasijoittelukuva	

Lyhenteet ja käsitteet

CRC	<i>Cyclic Redundancy Check</i> ; yleisesti käytetty virheentunnistusmenetelmä
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i> ; ilman virtaa tilansa pitävä uudelleenkirjoitettava muistityyppi
FSK	<i>Frequency-Shift Keying</i> ; radiosignaalin vaihemodulointimenetelmä
ISM	<i>Industrial, Scientific and Medical</i> ; alun perin teolliseen, tieteelliseen ja lääketieteelliseen käyttöön tarkoitettu radiotaajuuskaista
RSSI	<i>Received Signal Strength Indicator</i> ; signaalinvoimakkuudenilmaisoin
SPI	<i>Serial Peripheral Interface</i> ; yleiskäyttöinen tahdistettu piirilevyjen sarjaliikenneväylä
SAW	<i>Surface Acoustic Wave</i> ; pinta-aalto
UART	<i>Universal Asynchronous Receiver/Transmitter</i> ; tahdistamaton yleiskäyttöinen sarjaliikenneväylä.
USB	<i>Universal Serial Bus</i> ; nykyaikana käytetyin tietokoneiden sarjaliitäntä

1 Johdanto

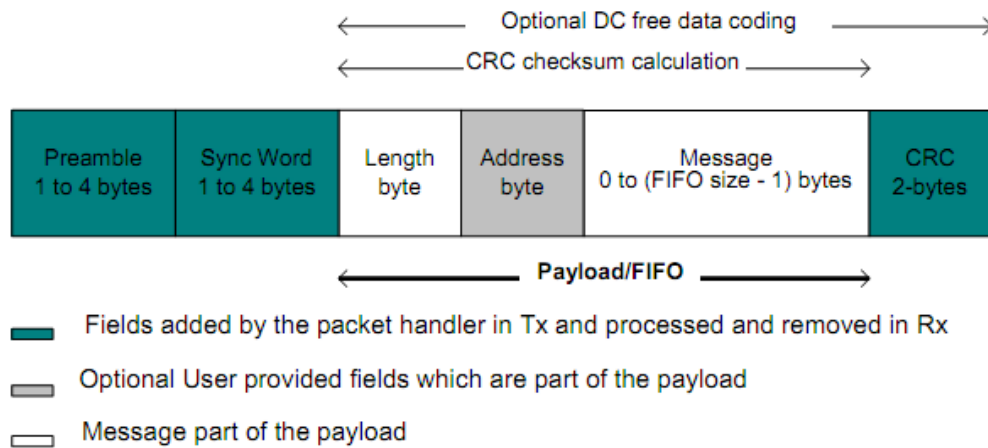
Tämä insinöörityö käsittelee edullista, yleiskäyttöistä ja matalaenergistä 433 MHz:n taajuudella toimivaa radiolinkkiä, joka liitetään USB-liitännällä tietokoneeseen. Radiolinkillä voidaan lähettää ja vastaanottaa tietoa sekä muuttaa asetuksia USB-väylän avulla esimerkiksi terminaaliohjelmaa käyttämällä. Lopullista tuotetta varten linkille voitaisiin tehdä erillinen käyttöliittymäohjelma. Radiolinkissä on säädettävä lähetysteho- ja nopeus sekä synkronointitavut. Radiolinkin keskeiset komponentit ovat mikrokontrolleri ja radiopiiri. Mikrokontrolleri on ohjelmoitu käyttäen C-ohjelmointikieltä.

Tämän työn sovelluksessa radiolinkki toimii kahden tietokoneen välisenä kommunikointikanavana, mutta se on helposti muokattavissa itsenäiseksi akku- tai paristokäyttöiseksi laitteeksi, esimerkiksi osaksi sensoriverkkoa. Sensoriverkossa voidaan hyödyntää säädettäviä synkronointitavuja. Niiden ansiosta voidaan valita, mitkä linkit kommunikoivat keskenään. Virrankulutus on minimoitu valitsemalla mahdollisimman vähävirtaiset komponentit, mutta samalla on pidetty laitteen kokonaiskustannukset mahdollisimman alhaisina. Työ tehtiin Agaidi Oy:lle, jonka toiminta keskittyy matalaenergiseen langattomaan tiedonsiirtoon. Työn tuloksia ja kehittyntä osaamista on tarkoitus hyödyntää tulevissa hankkeissa.

2 Radiolinkin tekniset määrittelyt

2.1 Radiolinkin suunnittelun lähtökohdat

Radiolinkki suunniteltiin mukailemaan ISO/IEC 18000-7 -standardia valikoiduin osin. Näin ollen keskitaajuudeksi valittiin 433,92 MHz, modulointitavaksi FSK (*Frequency-Shift Keying*), binäärien taajuusvaihteluksi ± 50 kHz ja symbolinopeudeksi n. 27 kHz tai tarvittaessa alhaisempi kuuluvuuden parantamiseksi. USB-väylän kautta laitteesta voidaan asettaa lähetysnopeus ja -teho, vastaanoton vahvistus ja synkronointitavut.



Kuva 1. Radiopiirin datapaketin rakenne [1, s. 51]

Varsinainen radiopaketti koostuu viidestä osasta (kuva 1). Preamblen pituudeksi on määritelty kolme tavua. Preamblen tarkoituksena on antaa vastaanottimelle aikaa valmistautua viestin vastaanottamiseksi ja tahdistaa ajastimensa viestin lukua varten. Tämän jälkeen seuraa yhdestä neljään ohjelmoitavaa synkronointitavua. Tämän työn tapauksessa käytössä on kaksi synkronointitavua, joilla saadaan osoitettua tarvittaessa 65 536 uniikkia linkkiä. Synkronointitavujen avulla voidaan säätää kuuluvuusalueella, mitkä laitteet vastaanottavat valitut lähetykset. Vastaanotto hylkää radiopiirin rautatasolla ne radiopaketit, joiden synkronointitavut eroavat ohjelmoiduista.

Jos tarvittaisiin tarkempaa osoittamista, voitaisiin käyttää osoitetavu-ominaisuutta, mutta tässä työssä se ei ole käytössä. Synkronointitavuilla voitaisiin osoittaa verkon osoite ja osoitetavulla yksittäisen tunnisteiden osoite. Tässä työssä ei kuitenkaan tarvita useiden verkkojen kattavaa osoitteistoa, mutta se on tarvittaessa helposti otettavissa käyttöön.

Ennen varsinaista hyötykuormaa lähetetään tavu, joka kertoo hyötykuorman pituuden. Ilman sitä vastaanottopää ei pysty tunnistamaan, mihin viesti päättyy. Seuraavana lähetetään varsinainen hyötykuorma, joka on tässä sovelluksessa 0 - 63 merkkiä. Kuitenkin pisin yhdellä paketilla lähetettävä datan pituus on 60 merkkiä, sillä dataan kuuluu myös UART-protokollan aloitusmerkki S ja loppumerkit `\r\n`. Lopulta lähetetään kaksi CRC-virheentunnistustavua. Virheentunnistus on radiopiirissä rautapohjainen. Jos virheentunnistuksessa huomataan, että vastaanotossa on tapahtunut virhe, paketti hylä-

tään, eikä mikrokontrollerille anneta keskeytystä. Näin virheelliset paketit eivät turhaan kuormita mikrokontrolleria.

Antenniliittimen impedanssisovitus tehdään 50 ohmiin, jotta voidaan käyttää edullisia 433 MHz:n taajuudelle suunniteltuja monopolianteenneja. Tarvittaessa linkkiin voidaan kuitenkin liittää mikä tahansa 50 ohmin antenni, esimerkiksi hyvin suuntaava antenni käyttökohteen niin vaatiessa. Sovitus tehdään käyttämällä diskreettejä keloja ja kondensaattoreita.

2.2 433 MHz:n ISM-taajuus

Radiotaajuuskaistat ovat hyvin säännösteltyjä, ja suunniteltaessa uutta laitetta on aina otettava huomioon paikalliset määräykset kunkin taajuuskaistan osalta. Suurimmalla osalla taajuuskaistoista ei saa lähettää lainkaan ilman lupaa, ja vapaatkin kaistat saattavat olla lähetystehoiltaan hyvin rajoitettuja.

433 MHz:n taajuus on yksi ISM-taajuusalueeseen (*Industrial, Scientific and Medical*) kuuluvista radiotaajuuskaistoista, joiden käyttö ei vaadi erillistä lupaa. Termi ISM tarkoittaa teollisuuden, tieteen ja lääketieteen laitteita, joiden tarkoituksiin kaista alun perin varattiin. Lähettäjiä ei välttämättä käytetä välittämään tietoa, vaan niitä voidaan käyttää myös kuumennusvaikutuksen aikaansaamiseksi. Esimerkiksi tavallinen mikroaaltouuni toimii 2,4 GHz:n ISM-taajuusalueella. [2.]

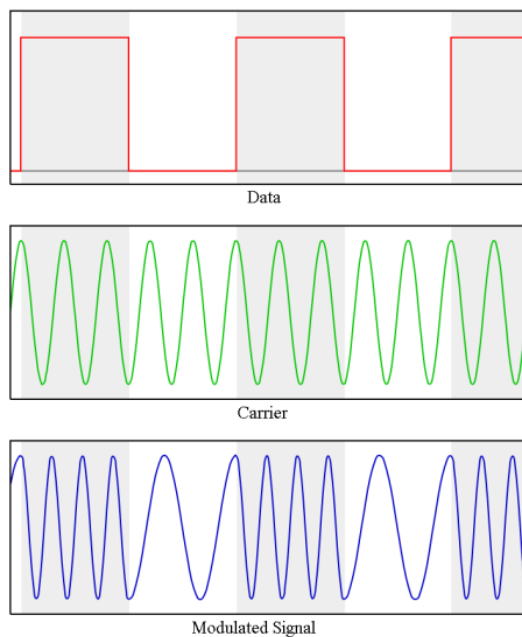
Vaikka näillä taajuuksilla lähettämiseen ei tarvita lupaa, niillä on kuitenkin tehorajoituksensa. Suomessa rajoja asettaa Viestintävirasto. Viestintäviraston määräysten mukaisesti 433 MHz:n taajuudella toimivan laitteen suurin lähetysteho saa olla 25 mW (n. +14 dBm) kymmenen prosentin toimintasuhteella, tai 1 mW (0 dBm) jatkuvana tehona. Kyseisellä taajuuskaistalla ei saa lähettää kuvaa tai ääntä, vaan se on varattu pienten tietomäärien välittämiseen. Esimerkiksi sensoriverkossa tietoa ei tarvitse yksinkertaisimmillaan välittää kuin joitain tavuja kerrallaan, ja sovelluksesta riippuen tämä voi tapahtua kerran minuutissa, tunnissa tai jopa vuorokaudessa. [3, s. 6.]

Vaihtoehtoisia taajuuksia ovat Euroopassa 868 MHz ja 2,4 GHz. Korkeammilla taajuuksilla on etuina suuremmat datanopeudet ja lyhyemmät antennit, mutta haittapuolina

huonompi kantama. Suurempi taajuus heikkenee voimakkaammin läpäistessään estettä tai heijastuessaan. Tämän työn sovelluksissa ei tarvita suurta datanopeutta, vaan suurempi merkitys on hyvällä kantomatalla.

2.3 FSK-modulaatio

Frequency-Shift Keying, eli FSK on taajuusmodulointimenetelmä, jossa dataa välitetään radioteitse käyttäen tiettyä kantoaaltoa, joka tämän työn tapauksessa on 433,92 MHz. Data välitetään kanta-aallon taajuutta muuttamalla. Tässä työssä käytetään yksinkertaisinta FSK-modulaatiotapaa, BFSK-modulaatiota eli binääristä FSK:ta.



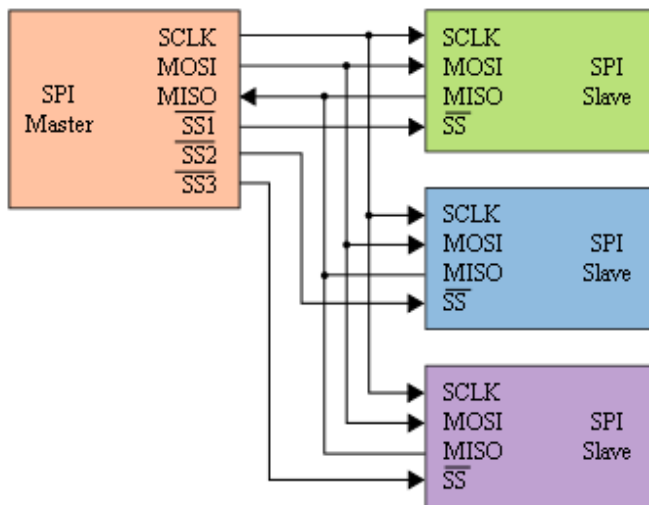
Kuva 2. BFSK-modulaatio [4]

Kuvassa 2 on esitetty, kuinka binäärisessä FSK-modulaatiossa looginen ykkönen välitetään nostamalla kanta-aaltoa määritellyn vaihtelun verran, eli tässä sovelluksessa taajuutta nostetaan 50 kHz:llä. Looginen nolla saadaan laskemalla kanta-aallon taajuutta samaisen vaihtelun verran. [4.]

3 Mikrokontrollerin kommunikointiväylät

3.1 Serial Peripheral Interface -väylä

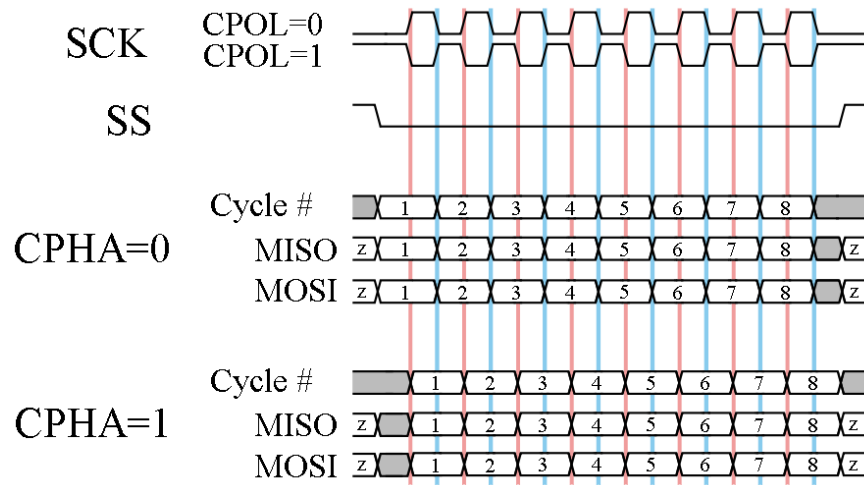
Mikrokontrolleri ja radiopiiri kommunikoivat keskenään käyttäen *Serial Peripheral Interface* -väylää eli SPI-väylää. SPI-väylä on sarjamuotoinen ja kaksisuuntainen (engl. *Full Duplex*) eli sillä voidaan lähettää ja vastaanottaa tietoa samaan aikaan. Yksi väylän laitteista on niin sanottu isäntä eli *master*, ja loput ovat orjia eli *slaveja*. Jokaisella isäntään liitettyllä laitteella on oma *slave select* -väylä, jolla isäntä signaloi kenen kanssa se haluaa kommunikoida. Dataväylät ovat kaikkien laitteiden välillä yhteisiä. *Master Output, Slave Input*, eli *MOSI*-väylällä siirretään tietoa isännältä orjalle ja *Master Input, Slave Output*, eli *MISO*-väylällä siirretään tietoa orjalta isännälle. SPI-väylällä kommunikoinnin aloite tulee aina isännältä, eikä orja voi puhua, ellei siltä ole kysytty jotain. Kaikki toiminnot ajoitetaan isännän SCLK-kelloväylän avulla.



Kuva 3. Isäntä-laite ja kolme orjaa SPI-väylällä [5]

Kuvassa 3 on kytketty SPI-väylää käyttäen neljä laitetta toisiinsa, joista yksi on määriteltä isännäksi. Nuolen suunta kuvaa liikenteen suuntaa. *Slave select* -nastat ovat alhaalla aktiivisia ja ne ovat perustilassa loogisessa ykkösessä. Kun isäntä haluaa kommunikoida orjan kanssa, se muuttaa halutun *slave select* -nastan loogiseksi nolllaksi ja aloittaa kommunikoinnin MOSI-väylällä SCLK-kellon tahtiin. Tämän jälkeen laitteesta

riippuen orjalta odotetaan vastausta MISO-väylältä tai nostetaan *slace select* -nasta heti takaisin aktiiviseksi viestinnän päättymisen merkiksi.



Kuva 4. SPI-väylän polariteetti ja -vaiheasetukset [5]

Laitteesta riippuen kellosignaali on alkutilassa joko ylhäällä tai alhaalla ja tietoa luetaan kellon nousevalla tai laskevalla reunalla (kuva 4). Kellon polariteetti- ja vaiheasetusten tulee olla asetettu yhteneviksi kaikkien kommunikoivien laitteiden välillä, jotta kommunikointi toimii [7]. Tämän työn radiopiiri käyttää yleisimmin käytössä olevia asetuksia, eli CPOL = 0 ja CPHA = 0. Tällöin data luetaan kellopulssein nousevalla reunalla ja kellosignaali on vakiotilassa loogisessa nollassa. [5.]

3.2 Universal Asynchronous Receiver/Transmitter -väylä

Tietokone ja radiolinkki kommunikoivat keskenään käyttäen UART- eli *Universal Asynchronous Receiver/Transmitter* -väylää. UART on nimensä mukaisesti asynkroninen, eli se ei tarvitse erillistä kellosignaalia, vaan nopeus on ennalta määriteltä. UART:a kutsutaan universaaliksi, koska sen datamuoto ja nopeus ovat muutettavissa. UART, kuten SPI:kin, on sarjamuotoinen ja kaksisuuntainen, eli dataa voidaan siirtää molempiin suuntiin samanaikaisesti.

UART on yleensä kahden laitteen välinen kommunikointiväylä, mutta yhtä lähetyklinjaa voi kuunnella useampikin laite. Tällöin useat kuuntelevat laitteet eivät kuitenkaan voi myös lähettää tietoa. Data lähetetään siirtorekisteriä käyttäen bitti kerrallaan ja vas-

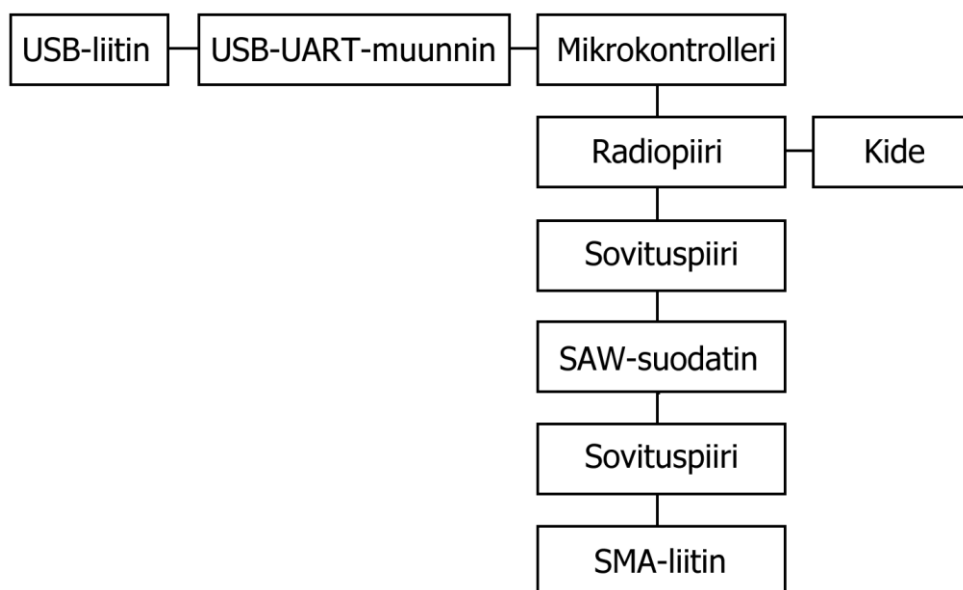
taanottopään siirtorekisteri yhdistää bitit takaisin tavuiksi. UART vaatii siis vain kaksi linjaa, lähetyksen (TX) ja vastaanoton (RX). Koska UART-väylällä ei ole erillisiä signaaliväyliä, lähetettävän merkin eteen on laitettava aloitusmerkki. Kun vastaanotto havaitsee aloitusmerkin, se rupeaa ottamaan tietoa vastaan. Näin molemmat osapuolet ovat samanarvoisia, ja kumpi tahansa voi aloittaa tiedonsiirron.

UART:a tai sen synkronista vastinetta USART:a on käytetty laajasti tietokoneisiin liitettävien lisälaitteiden kanssa, mutta nykyään sen on korvannut moninkertaisesti nopeampi USB, eli *Universal Serial Bus* -väylä. USB-laite voi olla myös hybridi, eli liikenne kulkee tietokoneelta lisälaitteelle USB-väylän yli, mutta laitteella oleva piiri muuntaa liikenteen tavalliseksi UART-liikenteeksi, eikä laite ole näin ollen täysiverinen USB-laite. Näin vältetään erillisten USB-ajurien kirjoittamiselta, ja etenkin kehitys- ja suunnitteluvaiheessa tämä on yksinkertainen tapa muodostaa yhteys tietokoneen ja prototyypin välille. [6.]

4 Radiolinkin komponentit

4.1 Radiolinkin lohkokaavio

Radiolinkin rakenne voidaan esittää lohkokaaviona (kuva 5).



Kuva 5. Radiolinkin lohkokaavio

Lohkokaaviosta nähdään, kuinka mikrokontrolleri on yhdistetty USB-väylään ja antenniliittimeen (SMA-liitin). Mikrokontrolleri on liitetty USB-UART-muuntimeen UART-väylän kautta, josta USB-UART-muunnin muodostaa virtuaalisarjaporttiyhteyden tietokoneeseen.

Mikrokontrollerin yhteys radiopiiriin toimii SPI-väylän kautta. Radiopiirin ja antenniliittimen välissä on SAW-suodatin ja impedanssisovituspiirit. Sovituspiirien avulla pyritään minimoimaan radiosignaalin häviöt. Radiopiiri tarvitsee kantataajuutensa muodostamiseen ulkoisen kiteen.

4.2 Radiopiiri

Radiopiirin valinnassa keskityttiin virrankulutuksen minimointiin, alhaiseen hintaan ja saatavuuteen. Samalla kuitenkin haluttiin saada mahdollisimman hyvä herkkyys ja laajat ominaisuudet, kuten valmis pakettimuoto ja CRC-virheenkorjaus. Radiolinkin tulee toimia tarvittaessa paristoilla tai akulla, eli piirin on tuettava matalia käyttöjännitteitä. Radiopiirin haluttiin määrittelyiden mukaisesti tukevan FSK- tai GFSK-modulaatiota. GFSK-modulaatio on FSK-modulaation kehittyneempi versio, jossa käytetään Gaussin suodatinta vaimentamaan päästökaistan reunataajuuksia.

Virrankulutuksessa suurin prioriteetti annettiin vastaanoton virrankulutukselle, koska monessa sovelluksessa radiolinkki lähinnä kuuntelee liikennettä ja tarvittaessa reagoi ajallisesti lyhyellä vastauksella. Halutuilla ominaisuuksilla varustettuja radiopiirejä on saatavana Semtechillä, Texas Instrumentsilla, Analog Devicella ja Silicon Labsilla. (Taulukko 1, ks. seur. s.)

Taulukko 1. Radiopiirien virrankulutukset

	Vastaanoton virrankulutus	Lähetyksen virrankulutus	Vastaanoton herkkyys
Semtech SX1212	3 mA	25 mA @ +10 dBm	-110 dBm, 2.0 kbps, OOK
TI CC430	15 mA	29 mA @ +10 dBm	-117 dBm, 0.6 kbps, 2-FSK
Analog Devices ADF7023	12.8 mA	24.1 mA @ +10 dBm	-116 dBm, 1.0 kbps, 2-FSK/GFSK
Silicon Labs Si4430	18.5 mA	30 mA @ +13 dBm	-121 dBm, 2.0 kbps, GFSK

Taulukosta 1 voidaan lukea, että Semtechin piiri SX1212 toimii oleellisesti pienimmällä virrankulutuksella vastaanottotilassa. Lähetyksen virrankulutukset olivat vertailussa olevilla piireillä lähellä toisiaan ja isompi vaihtelu nähdään vastaanoton virrankulutuksissa.

Vastaanoton herkkyydet on ilmoitettu optimaalisilla asetuksilla eli käytännössä hitaimmalla nopeudella ja piiristä riippuen herkimmällä modulaatiolla. Näin ollen ne eivät ole suoraan verrannollisia toisiinsa, mutta antavat silti osviittaa. Silicon Labsin piiri on herkkyydessä ylivoimainen, mutta kuluttaa myös vastaanotossa eniten virtaa.

Texas Instrumentsin CC430-piirin hyötypuolena on sisäänrakennettu mikrokontrolleri. Muut radiopiirit tarvitsevat erillisen mikrokontrollerin. Mikrokontrollerista johtuen TI CC430 on muita radiopiirejä kalliimpi ja sen käyttöönottoon vaadittaisiin myös kehitystyökalut. Uuden mikrokontrollerin opettelu olisi vienyt liikaa aikaa, joten kyseistä piiriä ei haluttu ottaa käyttöön ainakaan tässä vaiheessa.

Jäljellä olevista vaihtoehdoista päädyttiin Semtechin SX1212-piiriin, koska sillä on erittäin pieni vastaanoton virrankulutus ja hyvä saatavuus. Alhaista virrankulutusta pidettiin tässä tapauksessa tärkeämpänä kuin suurinta herkkyyttä. Semtechin piiri on myös edullinen noin kahden euron hinnallaan. SX1212 tarvitsee toimiakseen vain ulkoisen kiteen, SAW-suodattimen ja diskreettejä passiivikomponentteja. Passiivikomponentteja tarvitaan ohituskondensaattoreiksi, sovituskomponenteiksi, jänniteohjattua oskillaattoripiiriä varten, sekä ulkoista vaihelukittua silmukkaa varten. [1.]

4.3 Mikrokontrolleri

Mikrokontrollerilta vaadittiin alhaista virrankulutusta, *bootloader*-tuki, riittävästi ohjelmamuistia, SPI- ja UART-tuki, sekä riittävästi I/O-liitäntöjä merkkiledeille ja *slave select*-nastoille. Aikaisemmasta Atmel AVR -ohjelmointikokemuksesta ja kehitysalustan olemassaolosta johtuen mikrokontrolleri etsittiin Atmelin mikrokontrollereista.

Aikaisemmin käytetty vähävirtainen Atmel ATmega44Pa -mikrokontrolleri ei tukenut *bootloaderia* ja sen ohjelmamuisti oli liian rajallinen, joten mikrokontrolleriksi valittiin mallisarjan seuraava malli Atmel ATmega88Pa. *Bootloader* on mikrokontrolleriin ladattava ohjelma, jonka avulla kontrollerin ohjelma voidaan päivittää esimerkiksi sarjaporttiliitännän avulla. Näin ollen radiolinkin ohjelman päivittäminen on helpompaa, koska se ei vaadi erillistä ohjelmointilaitetta, kun *bootloader* on kerran ohjelmoitu piirille käyttäen ISP- eli *In-System Programmer*-väylää. ATmega88Pa sisältää 8 kt ohjelmamuistia, joka on kaksinkertainen määrä ATmega44Pa-mikrokontrolleriin nähden ja riittävä määrä hieman monimutkaisemmallekin ohjelmalle. ATmega88Pa sisältää myös sisäisen 8 MHz:n kellopiirin, joten se ei tarvitse ulkoista kidettä. [7.]

4.4 USB-UART-muunnin

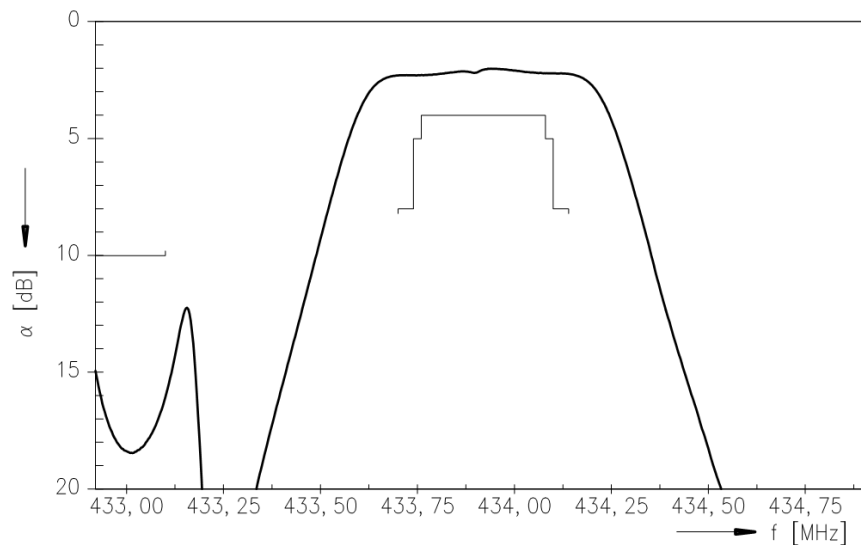
Radiolinkkiin tarvittiin myös USB-liitäntä, joten tietokoneen ja linkin välille lisättiin FTDI FT232RL -piiri, joka toimii virtuaalisena sarjaporttina käyttäen mikrokontrollerin UART-väylää. Näin radiolinkki näkyy tietokoneelle sarjaporttiin kytkettynä laitteena. Tietokone ja radiolinkki voivat kommunikoida esimerkiksi terminaaliohjelman välityksellä käyttäen FTDI:n ajureita, eli ilman tarvetta ohjelmoida erillisiä USB-ajureita. Näin radiolinkki voidaan myös ohjelmoida USB-väylällä, joka löytyy käytännössä jokaisesta tietokoneesta.

FTDI FT232RL:n EEPROM voidaan ohjelmoida käyttämällä FTDI:n valmistamaa FT_Prog-ohjelmaa. Ohjelman avulla laite saadaan tunnistumaan tietokoneessa halutun nimisenä, ja laitteelle voidaan antaa sarjanumero. Näin laitteet voidaan nimetä esimerkiksi yhtiön nimellä ja laitteen mallinumerolla ja -versiolla, ja yksilöidä sarjanumerolla. Nimeen voidaan liittää myös tieto laitteen fyysisestä ja ohjelmallisesta versiosta.

USB-UART-muunnin on kätevä tapa tehdä linkistä yhteensopiva lähes kaikkiin nykyaikaisiin tietokoneisiin. Viime vuosina valmistetuista tietokoneista ei useinkaan enää löydy RS232-sarjaliitintä, joka olisi toinen liitintävaihtoehto. FT232RL ei myöskään vaadi paljoa oheiskomponentteja, joten se ei vie paljoa tilaa piirilevytä.

4.5 SAW-suodatin

Radiopiiri tarvitsee lähetysignaalin suodattamiseksi pinta-aaltosuodattimen eli SAW-suodattimen (engl. *Surface Acoustic Wave*). SAW-suodatin on elektromekaaninen laite, jota käytetään usein radiotaajuuksien sovelluksissa. Se muuntaa sähköisen signaalin mekaaniseksi värähtelyksi ja takaisin käyttäen pietsosähköistä kidettä tai keraamia. SAW-suodattimeksi valittiin edullinen Epcos B3750, jolla on pieni vaimennus päästökaistalla. Suodattimella varmistetaan, ettei radiopiiri lähetä laittomia hajataajuuksia.



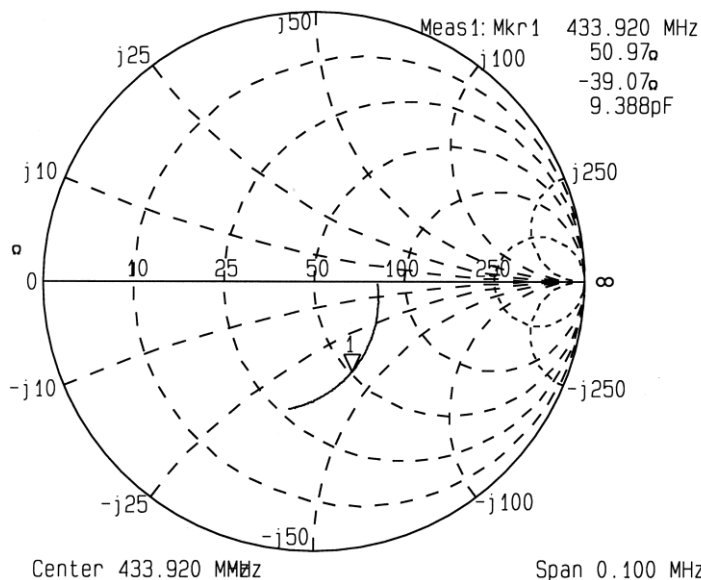
Kuva 6. Epcos B3750 -suodattimen vaimennus taajuuden funktiona [8, s. 5]

SAW-suodatin aiheuttaa ei-toivottua vaimennusta (kuva 6). Suodattimen keskitaajuus on 433,92 MHz, ja vaimennus on tällöin tyypillisesti 2,0 dB ja enintään 3,0 dB sisältäen sovituskomponentit 50 ohmiin. 3 dB:n päästökaista on tyypillisesti 0,70 MHz, joka on moninkertainen vaadittavaan kaistanleveyteen nähden. [8, s. 3.]

4.6 Sovituspiirit

Radiopiirin, SAW-suodattimen ja antennin impedanssit tulee sovittaa toisiinsa, jotta tehohäviö olisi mahdollisimman pieni. Tässä työssä impedanssit sovitetään käyttäen diskreettejä komponentteja. Koska käytettävät antennit ovat tässä sovelluksessa 50 ohmia, täytyy antenniliittimeltä SAW-suodattimelle päin näkyä sen kompleksikonjugaatti. Tässä tapauksessa kuorma on siis puhtaasti reaalinen, joten haluttu impedanssi on suoraan 50 Ohmia. SAW-suodattimen datalehdessä annetaan sovituskomponentit 50 Ohmiin, joka tarkoittaa yksittäistä 39 nH:n sarjakelaa. Koska piirilevyllä syntyy kuitenkin erilaisia hajasuureita, kuten johtimien pituuksista johtuvaa resistanssia ja induktanssia, käytäntö ei vastaa täydellisesti teoriaa.

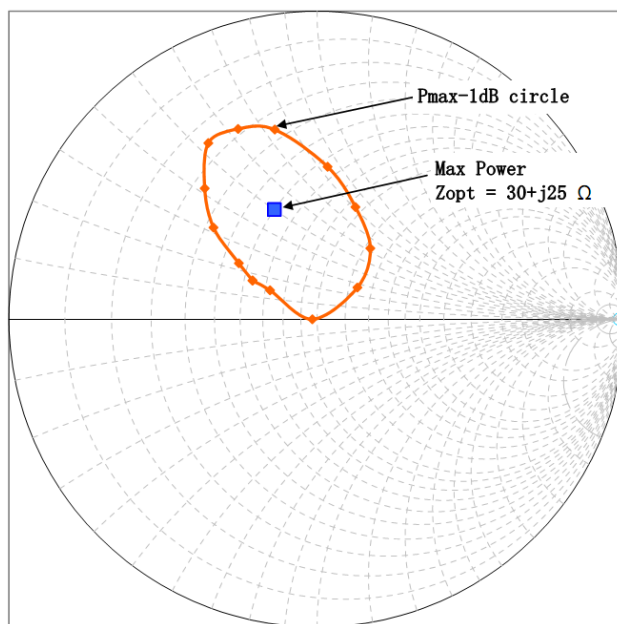
Yksinkertaisella koejärjestelyllä kokeiltiin hieman toisistaan poikkeavia sovituksia. Radiolinkki asetettiin lähettämään datapaketti, johon toinen linkki vastasi. Näiden viestien signaalinvahvuutta seuraamalla tulkittiin parasta mahdollista kuuluvuutta. Kokeet tehtiin Metropolian tutkimus- ja kehityskeskus Electrician tiloissa Teknobulevardilla, jossa toinen radiolinkin pää oli alimmassa kerroksessa ja toinen kolmannessa kerroksessa noin parikymmentä metriä sivussa.



Kuva 7. Antennin ja SAW-suodattimen sovitus 47 nH:n kelalla

Sarjakelan arvoa muutettiin hieman molempiin suuntiin, ja sen rinnalle kytkettiin myös 1,5 pF:n kapasitanssi. Erot kuuluvuudessa olivat pieniä eivätkä täysin toistettavia, sillä

pienetkin muutokset koejärjestelyissä voivat muuttaa tuloksia. Myöskään koetila ei ollut täysin staattinen rakennuksen luonteesta johtuen. Yksittäinen pykälää suurempi sovitusinduktanssi tuntui vaikuttavan positiivisesti kuuluvuuteen nähden, joten siitä tuli lopullinen sovitus (kuva 7). Impedanssi on varsin lähellä teoreettista 50 Ohmia, eli kartan keskipistettä. Mittaus on suoritettu käytössä olevalla kaistalla eli $433,92 \text{ MHz} \pm 50 \text{ kHz}$. Mitatut antennit olivat impedansseiltaan hieman kartan keskipisteen yläpuolella, joten sovitus on varsin hyvä. Edullisten monopoli-antennien tarkka mittaaminen osoittautui kuitenkin vaikeaksi, sillä antennit olivat erittäin häiriöalttiita mittaussäätelyn muutoksille.

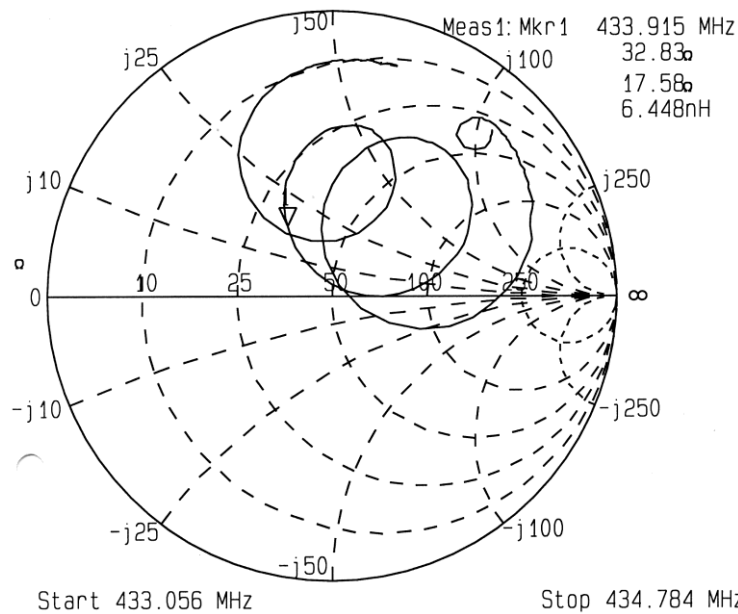


Kuva 8. Radiopiirin optimaalinen kuormaimpedanssi [1, s. 21]

Myös SAW-suodattimen ja radiopiirin välinen signaalitie on sovitettava. Radiopiirin datalehdessä nähdään radiopiirin optimaalinen kuormaimpedanssi (kuva 8). Tässä tapauksessa sovitettava impedanssi on siis annettu jo valmiiksi, kun asiaa tarkastellaan radiopiiriltä SAW-suodattimelle päin.

SAW-suodattimen datalehdessä on annettu suodattimen sisäänmenoimpedanssi, joka on $310 \Omega \parallel 2,2 \text{ pF}$. Kun taajuus tunnetaan, voidaan sisäänmenoimpedanssi muuttaa summamuotoon $70-j129$. Tämä impedanssi sovitettiin Smithin kartan avulla radiopiirin kuormaimpedanssiin. Radiopiirin datalehden mukaisesti sovitukseen on otettu mukaan

sarjakondensaattori, joka toimii niin sanottuna blokkikondensaattorina, joka estää taavirran kulun.



Kuva 9. Lopullinen mitattu sovitus radiopiiriltä SAW-suodattimelle

Sovituskomponentit juotettiin piirille ja sovitus mitattiin. Komponenttiarvoja hienosäädettiin ja vaikutuksia kokeiltiin jälleen mahdollisimman staattisella mittausjärjestelyllä. Parhaaksi todettu sovitus on esitetty kuvassa 9, jossa kolmiolla osoitettu keskitaajuus on lähellä radiopiirin optimaalista kuormaimpedanssia. Kuvasta on huomioitava laaja mittausalue, joka ylittää selkeästi käytettävän kaistan. SAW-suodattimen luonteen takia mittaustulos rupeaa voimakkaasti muuttamaan arvoaan optimaalisen kaistan ulkopuolella. Koska radiolinkki hyödyntää vain 100 kHz:n kaistaa, tästä ei aiheudu ongelmaa.

5 Piirilevyn suunnittelu

5.1 Piirilevyn kytkentäkaavio

Kytkentäkaaviossa (liite 1) on pyritty seuraamaan eri komponenttien datalehtiä mahdollisimman pitkälle. Kaikki aktiiviset komponentit on ohitettu 100 nF:n ohituskondensaattoreilla. Ohituskondensaattorin tehtävänä on toimia piensignaalinä, eli suodattaa käyttäjännitenastoista mahdollisesti häiriöitä aiheuttavaa kohinaa. USB-piirin yhtey-

dessä on 5 voltin puolella useita suodatuskomponentteja. Liittimen läheisyydessä oleva 10 nF:n kondensaattori suodattaa suuria kohinataajuuksia, jonka jälkeen käyttöjännite kulkee ferriitin läpi, joka toimii passiivisena alipäästösuodattimena kelan tapaan. Ferriitti on matalille taajuuksille oikosulku ja korkeille taajuuksille avoin, eli se toimii kuristimena.

Ferriitin jälkeen on vielä kaksi erisuuruista kondensaattoria, jotka suodattavat käyttöjännitettä laajalla taajuuskaistalla ennen komponenttia. 3,3 V käyttöjännite saadaan USB-piiriltä. Käyttöjännite on reguloitu sisäisellä regulaattorilla USB-väylän 5,0 voltin linjasta ja 3,3 voltin lähtönastassa on vielä varmuuden vuoksi 100 nF:n kondensaattori toimimassa pienenä virtapuskurina. USB-piirillä on ylösvetovastus PWREN# -nastassa, joka pitää komponentin aktiivisessa tilassa. Erittäin vähävirtaisessa sovelluksessa USB-piiri voitaisiin ajaa lepotilaan, kun sitä ei tarvita. [9.]

Mikrokontrollerin vieressä on kaksi lediä etuvastuksineen: yksi vihreä ja yksi keltainen. Ledit eivät näy ulos laitteesta, vaan ovat lähinnä testaamista varten. Mikrokontrollerin *reset*-nasta on ylösvedetty, joten kontrolleri on aina aktiivisena virtojen ollessa kytkettynä. Kytöntä painamalla ylös veto voidaan ohittaa ja asettaa *reset*-nastalle looginen nolla, jolloin mikrokontrolleri aloittaa ohjelman ajamisen alusta. Koska mikrokontrolleri ohjelmoidaan yhdistettyä ISP- ja SPI-väylää käyttäen, radiopiirin *slave select* -nastat ovat myös ylösvedettäviä, jotta radiopiiri on ohjelmoinnin ajan varmasti ei-aktiivisessa tilassa. Näin se ei häiritse ISP-ohjelmointia lähettämällä mitään.

Radiopiiri on liitetty SPI-väylällä, ja sille on sijoitettu tarvittavat suodattimet, sovituskomponentit ja biasoinnit datalehden mukaisesti. Kiteelle on myös asetettu datalehden mukaiset kuormakondensaattorit rinnankytkentänä.

5.2 Piirilevyn osasijoittelu

Osasijoittelussa lähdettiin siitä, että piirilevy asennetaan tiettyyn koteloon. Kotelo asetti vaatimuksensa etenkin piirilevyn koolle ja liittimien paikalle. Näin ollen USB-liitin ja antenni sijoitettiin piirilevyn pätyyn. Piirilevyn koko ei muodostunut ongelmaksi, vaan osasijoittelu voitiin tehdä hieman väljäksi ja näin helpommin juotettavaksi käsin. Väljyys jätti myös tilaa mahdollisille muokkaustoimenpiteille.

Piirilevyn alapuoli on maataso, jota rikkoo ainoastaan ne vedot, jotka täytyi vetää kahdella puolella. Isompi maataso antaa sähköistä vakautta. Se vähentää kohinaa ja takaa piirilevyn eri komponenteille sähköisesti mahdollisimman yhtenäisen maapotentiaalın ja parantaa antennin toimintaa. Iso maataso myös helpottaa piirilevysuunnittelua, koska maataso on signaalipuolella aina pelkän läpiviennin päässä. Kaikki vedot pyrittiin toteuttamaan mahdollisimman lyhyinä. Näin minimoidaan suuria maalenkkejä, jotka voivat aiheuttavaa häiriötä korkeammilla taajuuksilla. Pitkistä vedoista aiheutuu myös ei-toivottua vastusta. Ohituskondensaattorit on sijoitettu mahdollisimman lähelle komponentin käyttöjännitetasoja. Näin pyritään minimoimaan linjainduktanssia ja sarja-resistanssia [9].

(Ks. liite 2, osasijoittelukuva).

5.3 Piirilevyn toteutus ja kotelointi

Koteloksi valittiin edullinen ja hyvin saatavissa oleva malli, jonka mitat mahdollistivat sopivan komponenttitiheyden. Kotelo oli hieman suurempi, kuin mitä olisi tarvittu, mutta tämä voitiin kompensoida piirilevyn ulkomittoja kasvattamalla. Kotelo on valmistettu alumiinista, ja siinä on helposti muokattavat muoviset päädyt. Kotelon päälle liimattiin yrityksen logo.

Piirilevyt tilattiin suomalaiselta Prinell Piirilevy Oy:ltä. Piirilevy on toteutettu FR-4-materiaalille. FR-4 on lasikuituvahvistettu epoksilaminaatti, jota käytetään laajalti piirilevy-materiaalina. FR tarkoittaa tulenkestävyyttä (engl. *Flame Resistance*) ja tyyppinumero 4 kertoo, että kyseessä on lasikuidulla lujitettu epoksihartsia. FR-4 on ominaisuuksiltaan hyvä kompromissiratkaisu, sillä siinä on kohtuullisen hyvät mekaaniset-, sähköiset- ja lämpöominaisuudet, mutta se on silti edullinen materiaali. [10.]

Piirilevy on paksuudeltaan 1,6 mm, mikä on yleisesti käytetty piirilevyn paksuus. Se on tarpeeksi tukeva kestääkseen myös fyysistä vääntöä. Levyn paksuus ja kotelon levyura vastaavat toisiaan, joten piirilevy istuu kotelossa tukevasti eikä liiku heilutettaessa. Levyn pinnalla on 35 µm:n kuparikerros, johon komponentit juotetaan juotostinalla. Kuparikerroksen päällä on vielä juotteenestopinnoite, joka antaa piirilevylle sen vihreän värin. Juotteenestopinnoite estää juotteen leviämisen johtimia pitkin, ja siinä on reiät

vain komponenttien jalkojen ja läpivientien kohdalla. Juotteenestopinnoite on pakollinen lähinnä koneellisessa juottamisessa, mutta helpottaa siistin jäljen aikaansaamista myös käsijuotoksilla. Antennin ja USB-liittimen läpiviennit tehtiin kotelon muovipäättyyn Dremel-käsityökalulla.

Piirilevyllä voidaan tehdä myös silkkipaino, jolloin komponenttien nimet voitaisiin kalustamisen helpottamiseksi lukea suoraan piirilevyn pinnalta. Silkkipaino helpottaa myös symmetristen komponenttien oikein päin saamisessa ilman, että tarvitsee erikseen katsoa osasijoittelupiirustusta. Esimerkiksi mikrokontrollerin yläkulmaan on merkitty pieni ympyrä, joka helpottaa komponentin oikein päin asettamista, jota voisi verrata silkkipainokuvaan. Myös pienet palakomponentit voivat olla hankalia saada oikein monimutkaisessa levyssä, koska komponenttirykelmistä on vaikea nähdä mitkä padit kuuluvat millekin osalle. Silkkipaino jätettiin tässä projektissa pois kustannussyistä ja piirilevyn suhteellisen yksinkertaisesta rakenteesta johtuen.



Kuva 10. Kalustettu piirilevy ja koteloitu laite

Kuvassa 10 on kalustettu piirilevy ja koteloitu laite antennin kanssa. Ilman antennia laitteen ulkomitat ovat 88 * 72 * 19 mm, mutta tarvittaessa osasijoittelu on tehtävissä huomattavasti pienempään tilaan ja siten pienempään koteloon.

6 Mikroprosessorin ohjelma ja käyttöliittymä

6.1 USB-UART-protokolla

Laitetta voidaan käyttää kytkemällä se USB-liitännällä tietokoneeseen, minkä jälkeen laite tunnistuu tietokoneessa virtuaalisena sarjaporttina. Laite vaatii toimiakseen FTDI:n VCP-ajurin, jonka asennuksen jälkeen laite on valmis käytettäväksi. Tämän jälkeen laitteeseen voidaan ottaa yhteys millä tahansa terminaaliohjelmalla, joka tukee baudinopeutta 250 000 symbolia/sekunti.

Laite toimii asetuksilla 8-N-1, joka tarkoittaa, että merkkikoko on 8 bittiä, pariteettivirheentunnistus ei ole päällä, ja merkin jälkeen on yksi lopetusbitti. Yhteyden muodostamisen jälkeen laitteen asetuksia voidaan lukea tai muokata tai lähettää radiopaketti. Laitteen asetuksista voidaan tarpeen mukaan muuttaa lähetystehoa (TXP), vastaanoton vahvistusta (LNA), kommunikointinopeutta (BTR) ja kahta synkronointitavua (SC1/SC2). Laitteen UART-protokolla perustuu seuraaviin komentoihin:

Datan lähetys

```
['S'] [0-63 datatavua] [\r\n]
```

Vastaanottava laite näyttää

```
['S'] [datan pituus -tavu] [n datatavua] [RSSI] [\r\n]
```

datan pituus -tavu = 0x00...0x3F (= desimaaleina 0...63)

RSSI = 0x00...0xFF (RSSI / 2 = [dB])

Asetuksen syöttäminen

```
['C'] ['S'] [komento] [arvo] [\r\n]
```

komento = "TXP" / "LNA" / "BTR" / "SC1" / "SC2"

Taulukko 2. Parametrien arvot

	TXP [dbm]	LNA [dB]	BTR [kb/s]
0	12,5	0	3,3
1	9,5	-4,5	7,0
2	6,5	-9,0	11,1
3	3,5	-13,5	16,1
4	0,5		20,9
5	-2,5		26,7
6	-5,5		
7	-8,5		

Taulukossa 2 on esitetty parametrien mahdolliset arvot ja niiden vastineet. SC1 ja SC2 ovat synkronointitavuja, ja niiden arvo voi olla mitä tahansa väliltä 0x00-0xFF.

Asetuksen lukeminen

['C'] ['R'] [komento] [\r\n]

komento = "TXP" / "LNA" / "BTR" / "SC1" / "SC2"

Laite vastaa

['C'] ['R'] [arvo] [\r\n]

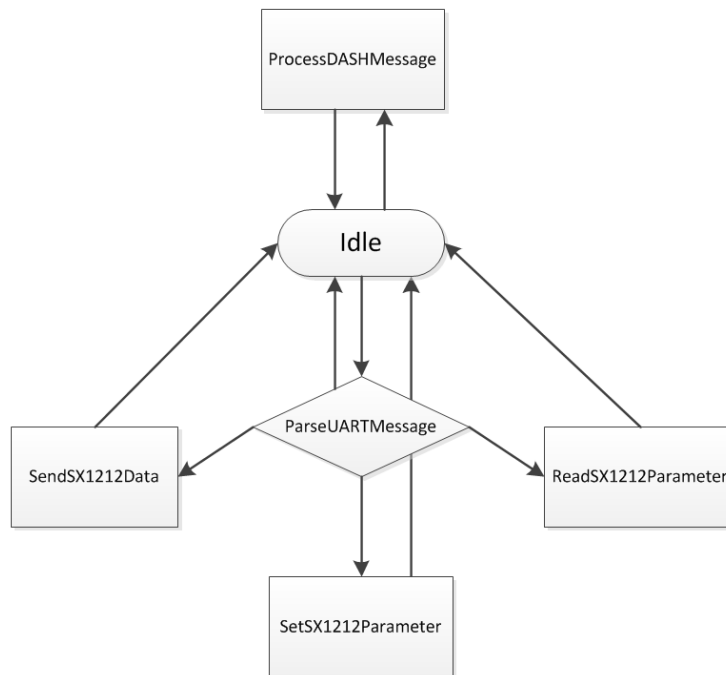
arvo = kuten komennon antamisessa

Jos komento on virheellinen, arvoksi tulee 'E'.

6.2 Mikrokontrollerin ohjelmointi

6.2.1 Ohjelman rakenne

Mikrokontrollerin ohjelma on kirjoitettu C-kielillä. Ohjelman sydän on *switch case*-tilakone ja NextState-muuttuja, joka pitää tiedon seuraavasta tilasta. Radiopiirin asetukset tallennetaan EEPROM-muistiin (*Electrically Erasable Programmable Read-Only Memory*), joka on mikrokontrollerilta löytyvää muistia, joka pitää tilansa myös ilman virtaa. Näin kerran syötettyjä asetuksia ei tarvitse antaa jokaisella käynnistyskerralla uudelleen.



Kuva 11. Ohjelman tilakaavio

Ohjelman aluksi tilakone (kuva 11) on Idle-tilassa. Idle-tila on tyhjäkäyntitila, jossa ohjelma odottaa tilanmuutosta tekemättä mitään. Tilanmuutoksen voi aiheuttaa kaksi keskeytystapaa: vastaanotettu radiopaketti tai UART-paketti.

Vastaanotettu radiopaketti aiheuttaa keskeytyksen, kun radiopiiri on vastaanottanut CRC-virheentunnistuksen läpi kulkeneen paketin. Keskeytyksen aikana luetaan paketin signaalivahvuus eli RSSI (*Received Signal Strength Indicator*) ja varsinainen hyötykuorma. Paketin lukemisen jälkeen NextState-muuttuja saa arvon ProcessDASHMessage. ProcessDASHMessage-tilassa mikrokontrolleri käsittelee puskurin sisällön ja lähettää vastaanotetun hyötykuorman ja signaalinvahvuuden UART-porttiin protokollan mukaisella kehyksellä. Tämän jälkeen palataan Idle-tilaan.

Jos radiopiiri vastaanottaa UART-väylältä komennon, se siirtyy ParseUARTMessage-tilaan, jossa määritellään minkä tyyppinen viesti on kyseessä. Jos paketin ensimmäinen kirjain on S eli lähetys (engl. *send*) on kyseessä datanlähetykspaketti, ja tilakone siirtyy SendSX1212Data-tilaan, jossa suoritetaan varsinainen radiolähetys. Tämän jälkeen palataan taas Idle-tilaan odottamaan seuraavaa tapahtumaa.

Jos paketin ensimmäinen kirjain on C eli komento (engl. *command*), tutkitaan seuraava merkki, joka kertoo, onko kyseessä asetuksen muuttaminen vai lukeminen. Jos paketin ensimmäinen merkki on S eli asettaminen (engl. *set*), siirtyy tilakone asetuksia muuttavaan SetSX1212Parameter-tilaan. SetSX1212Parameter-tilassa määritellään, mitä parametria halutaan muuttaa ja mihin arvoon. Mikrokontrolleri syöttää uuden asetuksen radiopiirille SPI-väylällä, ja asetukset tallennetaan EEPROM-muistiin. Tämän jälkeen laite siirtyy jälleen Idle-tilaan.

Jos komentoviestin toinen kirjain on R eli lukeminen (engl. *read*), siirtyy tilakone ReadSX1212Parameter-tilaan. ReadSX1212Parameter-tilassa mikrokontrolleri pyytää radiopiiriltä kysytyn asetuksen arvon. Jos annettu komento on virheellinen, mikrokontrolleri antaa vastauksena arvon E, joka kertoo käyttäjälle virheellisestä komennosta. Mikrokontrolleri käsittelee vastauksen ja lähettää sen UART-väylällä käyttäjälle. Ohjelma siirtyy Idle-tilaan. Jos paketti ei täytä mitään edellä mainituista muodoista, se hylätään ParseUARTMessage-tilassa ja siirrytään odottamaan uutta keskeytystä.

6.2.2 Lähdekooditiedostot ja funktiot

Ohjelma koostuu kuudesta lähdekooditiedostosta:

1. main.c
2. defines.h
3. uart.c
4. uart.h
5. sx1212.c
6. sx1212.h.

Kaikki säädettävät asetukset, kuten UART-nopeus ja SPI:n portit, on kerätty defines.h-tiedostoon, josta niitä on helppo yhteisesti muuttaa. Näin eri asetuksia ei joudu etsimään useista tiedostoista. Myös ohjelman kääntämiseen tarvittavat kirjastot on kerätty sinne, jolloin niitä ei tarvitse erikseen lisätä jokaiseen niitä käyttävään tiedostoon. Tämä ohjelma sisältää kaksi itse tehtyä kirjastoa eli radiopiirin sx1212-kirjaston ja USB-UART-sarjaliikennettä varten uart-kirjaston.

Näiden lisäksi on lisätty myös yleisesti tarvittavia kirjastoja, eli tässä tapauksessa I/O-, keskeytys-, viive- ja merkkijonokirjastot. Defines-tiedosto sisältää myös makroja led-merkkivaloille, uart-kirjastolle ja radiopiirille. Makroilla voidaan kirjoittaa usein tarvittavia komentoja helpommin muistettavaan muotoon. Makrot ovat hyödyllisiä myös silloin, kun useat eri komennot käyttävät samaa funktiota eri parametreilla, jolloin käyttäjän tarvitsee kirjoittaa parametrin vain kerran. Defines.h tiedostosta valitaan myös, mihin porttiin ja nastoihin radiopiiri on liitetty.

Main.c-tiedosto sisältää tilakoneen ja sen eri tilafunktiot. Ennen *switch case* -tilakoneeseen siirtymistä mikrokontrollerilla ajetaan asetusfunktio, jossa asetetaan I/O-porttien suunnat oikeiksi ja kutsutaan myös SPI:n, UART:n ja radiopiirin asetusfunktioita. Asetusfunktion lopuksi sallitaan keskeytykset ja säädetään ne toimimaan nousevalla reunalla. Tämän jälkeen vilautetaan molempia led-valoja tilakoneeseen siirtymisen merkiksi, ja laite on valmis toimimaan.

Uart.c-tiedostossa on UART-funktiot eli asetusfunktio, lähetysfunktio, vastaanoton tyhjennysfunktio ja vastaanoton keskeytysfunktio. Keskeytysfunktio kerää merkit UART:n kautta puskuriin ja siirtää tilakoneen UART-viestin purkufunktioon lopetusmerkkien jälkeen. Keskeytysfunktio osaa myös tyhjentää puskurin ylivuodon tapahtuessa. Ylivuoto voisi aiheuttaa ohjelman kaatumisen johtuen satunnaiseen muistipaikkaan kirjoittamisesta. Uart.c tarvitsee toimiakseen uart.h-otsikkotiedoston, jossa esitellään käytettävät funktiot.

Sx1212.c-tiedosto sisältää seuraavat funktiot:

- void SPI_Init()
- void SX1212_Init()
- void SX1212_Conf(uint8_t adr, uint8_t cmd)
- uint8_t SX1212_ReadReg(uint8_t cmd)
- uint8_t SX1212_ReadByte(void)
- void SX1212_Byte2Fifo(uint8_t data)
- void SX1212_Transmit(char data[], int lenght)
- void SX1212_Set_TXP(uint8_t pwr)
- void SX1212_Set_LNA(uint8_t gain)

- void SX1212_Set_BTR(uint8_t br)
- char SX1212_Get_TXP(void)
- char SX1212_Get_LNA(void)
- char SX1212_Get_BTR(void)
- ISR(IRQ_1_vect).

Init- eli asetusfunktioiden lisäksi löytyvät funktiot, joilla voidaan lukea ja kirjoittaa radiopiirin asetus- ja FIFO-rekistereitä, ja lähettää datapaketti. Set- ja Get-funktioilla voidaan lukea ja kirjoittaa määriteltyjä asetuksia. Asetukset määritellään tietyn rekisterin tietyillä biteillä, jolloin tarvitaan funktiot purkamaan tietoa koskematta muihin asetusbitteihin. Set- ja Get-funktioita käytetään UART-protokollassa. Näiden lisäksi defines.h-tiedostosta löytyy vielä makrot kahden synkronointitavun lukemiseksi ja kirjoittamiseksi. Synkronointitavut ovat erillisissä rekistereissä kokonaisina tavuina, joten niitä voidaan muuttaa ja lukea suoraan.

Lopuksi on vielä keskeytysfunktio, jonka laukaisee vastaanotettu radiopaketti, joka on käynyt läpi CRC-virheentunnistuksen. Keskeytysfunktio lukee vastaanotetun radiopaketin signaalinvahvuuden ja datan niille varattuihin muuttujiin. Ensimmäinen vastaanotettu tavu on datan pituus, jonka perusteella FIFO luetaan tyhjäksi. Tämän jälkeen tilakone siirtyy lähettämään vastaanotetun viestin UART:lla.

Myös sx1212.c tarvitsee toimiakseen otsikkotiedoston sx1212.h.

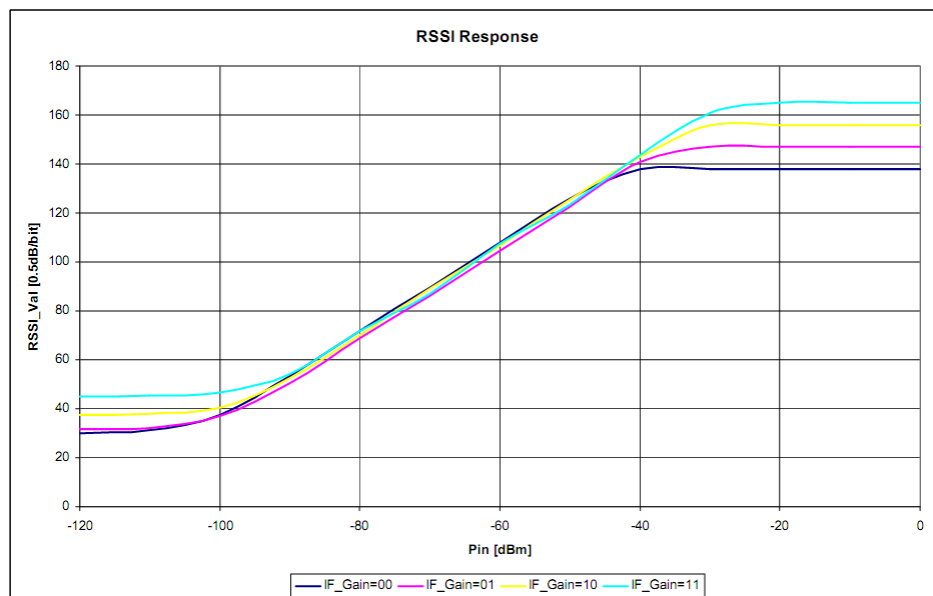
6.2.3 Radiopiirin toiminta

Radiopiirin asetukset säädetään 31 rekisterin avulla. Radiopiiri tukee FSK:n lisäksi OOK-modulaatiota (*On-Off-Keying*), joten radiopiiriltä tulee valita oikea modulaatio. OOK-modulaatio on sama kuin ASK (*Amplitude-Shift-Keying*), mutta käsittää analogisen amplitudimoduloinnin sijaan vain kaksi tilaa eli loogisen ykkösen ja nollan. Radiopiirillä on kolme rekisteriä, jotka määrittelevät kanta-aallon taajuuden. Rekisteriarvot riippuvat kanta-aallon taajuuden lisäksi käytetystä kiteestä. Koska käytössä on hieman datalehdien esimerkkititeestä poikkeava taajuus, jouduttiin arvot laskemaan erikseen.

Kantoaaltoja voidaan määrittää kaksi, jolloin taajuutta voidaan vaihtaa nopeasti valintarekisteristä. Näin saataisiin tarvittaessa kaksi eri lähetyiskanavaa. Tässä sovelluksessa käytetään kuitenkin vain yhtä taajuutta. Datalähetys on asetettu pakettimuotoiseksi ja käyttöön on otettu oikea määrä preamble- ja synkronointitavuja. Lähetysnopeus on säädettävissä ja käyttäjä voi muuttaa sitä esivalittujen arvojen välillä UART-protokollan välityksellä.

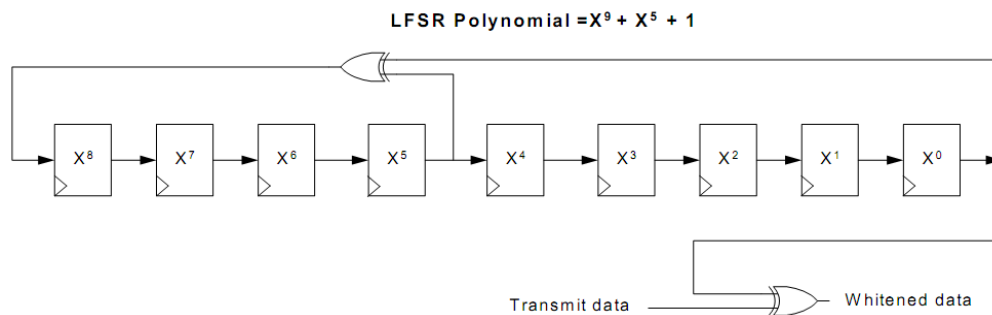
Radiopiirillä on kaksi osittain ohjelmoitavaa keskeytysnastaa: IRQ_0 ja IRQ_1. IRQ_1 on asetettu antamaan CRC_ok-keskeytys vastaanottotilassa, kun radiopiiri on vastaanottanut hyväksyttävän paketin. Tämä keskeytys aktivoi vastaanottofunktion. Lähetystilassa IRQ_1 antaa keskeytyksen lähetyksen valmistumisen merkiksi. Näin osataan odottaa oikea aika ennen radiopiirin tilan muuttamista valmius- tai vastaanottotilaan lähetyksen jälkeen. IRQ_0-keskeytys ei ole käytössä.

FIFO-rekisterin koko on määritelty suurimpaan mahdolliseen eli 64:een tavuun. Lähetystilassa piiri voidaan pitää FIFO:n täyttämisen aikana valmiustilassa. Kun FIFO on täytetty, asetetaan laite lähetystilaan. Piiri tunnistaa, milloin lähetystila on valmis ja lähettää sen jälkeen paketin. Kun paketti on lähetetty, saadaan keskeytys, jonka avulla piiri voidaan siirtää pois lähetystilasta. Näin virtaa vievässä TX-tilassa ei olla kauempaa kuin tarve vaatii ja vältetään epätarkan viivefunktion käyttämiseltä.



Kuva 12. Signaalinvoimakkuus eri vastaanoton vahvistuksilla [1, s. 27]

Datapakettiin kuuluva signaalinvoimakkuuden ilmaiseva RSSI-tavu tulee jakaa kahdella, jolloin saadaan todellinen signaalinherkkyys desibeleinä. Saatu desibeliluku kertoo kuinka paljon vastaanotettu signaali on nimellistä minimiherkkyttä voimakkaampi (kuva 12). Signaalinvoimakkuusilmaisimeen vaikuttaa käytössä oleva vastaanoton väli-taajuuden vahvistus (kuvassa IF_Gain). Signaalinvoimakkuuden ilmaisinta voidaan käyttää esimerkiksi kolmiopaikannuksessa, jos radiolinkki on usean niin kutsutun majakan kantama-alueella. Arvio radiolinkin sijainnista voidaan laskea kolmiopaikannuksella, kun majakoiden sijainnit tunnetaan. Yksinkertaisemmassa sovelluksessa voidaan käyttää myös paikan määrittämiseen solu-menetelmää, jos riittää suuntaa antava tieto lin-kin lähinnä olevasta majakasta.

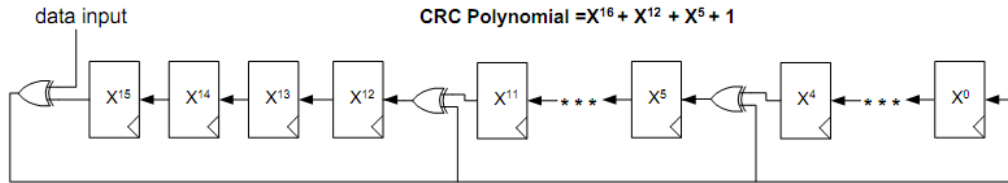


Kuva 13. Tiedon valkaisu [1, s. 54]

Radiopiiri lähettää hyötykuorman ja CRC-tavut käyttäen tiedon valkaisu (engl. *data whitening*). Valkaisu tarkoittaa, että tiedolle tehdään XOR-operaatio ennalta määritetyn satunnaisen luvun kanssa (kuva 13). Tämä on hyödyllistä, jos hyötykuormassa on useita nollia tai ykkösiä peräkkäin, jolloin vastaanoton synkronointi on haasteellisempaa. Vastaanotto poistaa valkaisuun tekemällä saman XOR-operaation ennen tiedon kirjoittamista FIFO:on, jolloin radiopiiriin käyttäjän ei tarvitse välittää valkaisuusta lainkaan.

Vaihtoehto synkronointiongelman vähentämiseksi olisi Manchester-koodaus, jossa bitit ilmaistaan tilamuutoksia käyttäen. Nouseva reuna tarkoittaa loogista ykköstä ja vastavasti laskeva reuna loogista nollaa. Näin jokainen lähetetty bitti sisältää tilanmuutoksen, jolloin hyötykuorman lisäksi vastaanotto saa jatkuvasti tahdistussignaalin. Huonona puoleena tiedon valkaisuun nähden Manchester-koodauksen käyttäminen kuitenkin puolittaa lähetysnopeuden. Tämän sovelluksen lähetysnopeuksiin päästäisiin kuitenkin

kaksinkertaistamalla nopeudensäätörekisterien arvot, sillä radiopiiri kykenee huomattavasti käytettyjä lähety nopeuksia suurempiin nopeuksiin.



Kuva 14. CRC-tavujen laskeminen [1, s. 53]

Datapaketin loppuun kuuluu kaksi CRC-virheentunnistustavua (engl. *Cyclic Redundancy Check*). SX1212-radiopiiri käyttää CCITT CRC -polynomia. Lähettävä radiopiiri laskee CRC-tarkistusavaimen (kuva 14), joka on hyötykuorman ja CRC-polynomien jakojäännös, ja lisää sen hyötykuorman perään. Vastaanottava radiopiiri jakaa hyötykuorman ja sen perään lisätyt CRC-tavut CRC-polynomilla. Jos tieto on saatu onnistuneesti vastaan, jakojäännöksen tulee olla nolla. Näin tapahtuessa radiopiiri antaa CRC_ok-keskeytyksen. Muussa tapauksessa paketti on vastaanotettu virheellisesti ja se hylätään. [11.]

CRC voi tunnistaa virheitä, mutta se ei voi korjata niitä. CRC-virheentunnistusta käyttäen voidaan huomata yksittäiset bittivirheet, mutta myös yleensä usean bitin virheet. Sen tehoon vaikuttaa valittu CRC-polynomi. CRC on virheentunnistusmenetelmänä huomattavasti tehokkaampi kuin yksinkertaisempi pariteettivirheentunnistus, joka tunnistaa vain parittoman määrän virheitä.

7 Mittaustulokset

7.1 Kantamamittaukset

Kantamamittauksia on tehty Metropolian Ammattikorkeakoulun tiloissa Technopolis Helsinki-Vantaalla. Kantamaa on arvioitu vain pääpiirteittäin, koska siihen vaikuttaa oleellisesti ympäristö ja sen muuttuvat tekijät. Näin ollen jossain muussa toimistorakennuksessa tulokset voisivat olla täysin erilaiset rakenteista riippuen. Kantamaan vaikuttaa etenkin rakenteiden metallin määrä. Myös paksut betonikerrokset vaimentavat

signaalia voimakkaasti, koska betonissa on usein käytetty teräskuitua tai –verkkoa vahvikkeena.

Kantamaan voidaan vaikuttaa lähetysnopeudella ja –teholla sekä vastaanoton vahvistusta säätämällä. Suurimmalla mahdollisella teholla kantama on parhaimmillaan, lähetyspään ollessa sisätiloissa ja vastaanotin ulkona, useita satoja metrejä. Vastaanotto ei kuitenkaan ole pitkällä matkalla enää virheetöntä, vaan luotettavan yhteyden aikaansaamiseksi tulee käytössä olla jonkinlainen uudelleenlähetysjärjestelmä.

Lähes avarassa tilassa kantama on hyvällä luotettavuudella arviolta joitain satoja metrejä antennien ollessa suorassa toisiinsa nähden. Sisätiloissa pysyessä kantama on useita kerrosvälejä, eli joitain kymmeniä metrejä tarvittavasta vakaudesta riippuen. Lähetystehon lisäksi lähetysnopeudella on suuri vaikutus vastaanoton luotettavuuteen, joten hyvin pientä dataliikennettä vaativissa kohteissa päästään huomattavasti parempaan kantomatkaan kuin suuremman tietomäärän sovelluksissa. Vähävirtaisissa sovelluksissa on aina pyrittävä käyttämään mahdollisimman pientä lähetystehoä.

Joissain sovelluksissa voidaan tarvita myös hyvin lyhyttä kantamaa. Pienellä lähetysteholla ja vastaanoton vahvistuksella kantama saadaan tarvittaessa tiputettua alle parin kymmenen metrin. Hieman vaimennusta lisäämällä matka saadaan pudotettua senkin alle.

7.2 Virrankulutusmittaukset

Laitteen kokonaisvirrankulutus mitattiin syöttämällä käyttöjännitenastoihin 3,3 voltin jännite. Näin USB-piiri ei ollut aktiivisena mittauksien aikana. Mittauksilla haluttiin nähdä, kuinka paljon laite vie virtaa akku- tai paristokäyttöisenä, koska tällöin virrankulutus on kaikkein oleellisinta. Mikrokontrolleri asetettiin *power down* -virransäästötilaan.

Taulukko 3. Radiolinkin virrankulutus radiopiirin eri tiloissa

Radiopiirin tila	Virrankulutus [mA]
STBY	0,28
SLEEP	0,20
RX	2,62
TX, 12,5 dBm	21,10
TX, -6,5 dBm	10,96

Taulukosta 3 nähdään radiolinkin virrankulutus radiopiirin ollessa valmiustilassa (STBY), virransäästötilassa (SLEEP), vastaanottotilassa (RX), lähetystilassa suurimmalla teholla (TX, 12,5 dBm) ja lähetystilassa pienimmällä teholla (TX, -6,5 dBm). Näiden mittauksien perusteella radiopiiri vie lähetys- ja vastaanottotiloissa jopa vähemmän virtaa, kuin valmistaja on datalehdessä ilmoittanut. SLEEP-tilassa virrankulutus on 0,1 μ A:a, joten käytännössä virtaa kuluu pelkästään muissa komponenteissa [1]. Vastaanoton vahvistuksen (engl. *Low Noise Amplifier* eli LNA) muuttaminen ei näkynyt virrankulutuksessa.

Taulukko 4. Mikrokontrolleri ja radiopiiri valmiustilassa

Mikrokontrollerin kellotaajuus	Virrankulutus [mA]
1 MHz	0,89
8 MHz	3,02

Myös mikrokontrollerin virrankulutus eri kellotaajuuksilla mitattiin (taulukko 4). Kellopulssin tuottaa mikrokontrollerin sisäinen 8 MHz RC-oskillaattori, ja se on ohjelmoitavissa esijakajia käyttäen tai jaettavissa kahdeksalla mikrokontrollerin niin kutsutulla sulakkeella (engl. *fuse*). Näissä mittauksissa on käytetty sulakkeella esijaettua kellopulssia. Sisäinen vastukseen ja kondensaattoriin perustuva värähtelijä ei ole absoluuttisesti kovin tarkka, joten se ei sovellu aikakriittisiin sovelluksiin. Tarkkaa ajanhallintaa vaativissa sovelluksissa mikrokontrollerille tarvitaan ulkoinen kide.

Jos mittausarvoista vähennetään radiopiirin virrankulutus valmiustilassa (n. 0,08 mA), saadaan pelkästään mikrokontrollerin käyttämä virta, jolloin pitää kuitenkin huomioida epätarkkuus johtuen piirilevyllä ja muissa komponenteissa tapahtuvista häviöistä. Mik-

rokontrolleri vie siis huomattavasti vähemmän virtaa alemmalla kellotaajuudella, joten kellotaajuus kannattaa sovittaa kulloisenkin käyttökohteen vaatimusten mukaisesti.

On huomioitava, että on energiatehokkaampaa suorittaa toimenpide nopeasti käyttämällä hetkellisesti suurta kellotaajuutta kuin tehdä sama asia pienemmällä kellotaajuudella vastaavasti pidemmässä ajassa. Tässä sovelluksessa käytetään kuitenkin suurinta 8 MHz:n kellotaajuutta ilman esijakajia, koska se mahdollistaa huomattavasti suuremman UART-nopeuden. Radiolinkki ottaa virtansa USB-liittimestä, joten muutaman milliampeerin ero virrankulutuksessa ei ole merkittävä.

Kun jännite annettiin FTDI FT232RL -piirin sisäisen regulaattorin kautta viitenä voltina, eli simuloitiin tilannetta, jossa radiolinkki on liitetty USB-porttiin, saatiin virrankulutukseksi käytännössä samat virran arvot. Suuremmasta jännitteestä johtuen tämä kuitenkin tarkoittaa suurempaa kulutettua tehoa. Ylimääräinen teho kulutetaan FT232RL-piirin regulaattorissa ja siitä muodostuu lämpöä.

Jos linkiltä odotetaan pitkää toiminnallisuutta pienen pariston tai akun kanssa, virrankulutusta on mahdollista vielä pienentää asettamalla myös mikrokontrolleri lepotilaan aina kuin mahdollista. Tällöin sen virrankulutus putoaisi murto-osaan. Mikrokontrolleri voidaan asettaa heräämään taas aktiiviseen tilaan keskeytyksestä. Keskeytyksen voi muodostaa sisäinen ajastin, joka tarkistaa tietyin väliajoin odotetaanko mikrokontrollerilta jotain toimintaa. Herättävänä keskeytyksenä voi toimia myös painonappi tai radiopiirin heräte viestin vastaanottamisen jälkeen.

Radiopiirin pitäminen jatkuvasti vastaanottotilassa vie myös suhteellisen paljon virtaa, joten sitäkin voitaisiin pitää aktiivisena sykleittäin. Esimerkiksi sensoriverkossa asiakas-kortin radiovastaanotin voitaisiin laittaa päälle sekunnin välein vain muutamaksi millisekunniksi. Tukiasema voitaisiin asettaa lähettämään ennen viestiä yli sekunnin pituinen heräte. Herätteen huomattuaan asiakas-kortti jäisi päälle kuuntelemaan tulevan viestin. Näin kortti kuulisi aina tukiaseman viestit olematta päällä koko aikaa. Tämä lähestymistapa ei toimisi nopeaa reagointia vaativissa sovelluksissa. Asiakas-kortin suunnasta tukiasemalle kulkeva viestintä tapahtuisi kuitenkin nopeasti.

8 Yhteenveto

Insinööriyössä toteutettiin edullinen yleiskäyttöinen USB-liitäntäinen radiolinkki, joka käyttää 433 MHz:n taajuutta. Radiolinkki saatiin suunniteltua ja toteutettua aikataulussa, ja se toimi käyttötarkoituksessaan halutulla tavalla. Linkki on helposti muokattavissa, ja sitä voidaan käyttää joko sellaisenaan, tai se voidaan integroida osaksi jo olemassa olevaa järjestelmää. Suhteellisen matala taajuus mahdollistaa pitkän kantomatkan, varsinkin alhaisella lähetysnopeudella.

Radiolinkki on optimaalinen vaihtoehto esimerkiksi sensoriverkon osaksi tai mihin tahansa pientä datamäärää vaativaan sovellukseen. Vähäisen komponenttimäärän ja niiden suhteellisen alhaisen hinnan johdosta sovelluskohteita on hyvin paljon. Koska piirilevysuunnittelu pohjautui aikaisemmin tehtyyn prototyyppiin, sen rakenne ei ole suurtaajuusteiden osalta täysin optimaalinen. Vastaisuudessa tulisi huomioida paremmin antennin sijainti ja radiosignaaleiden mahdollisimman lyhyt pituus.

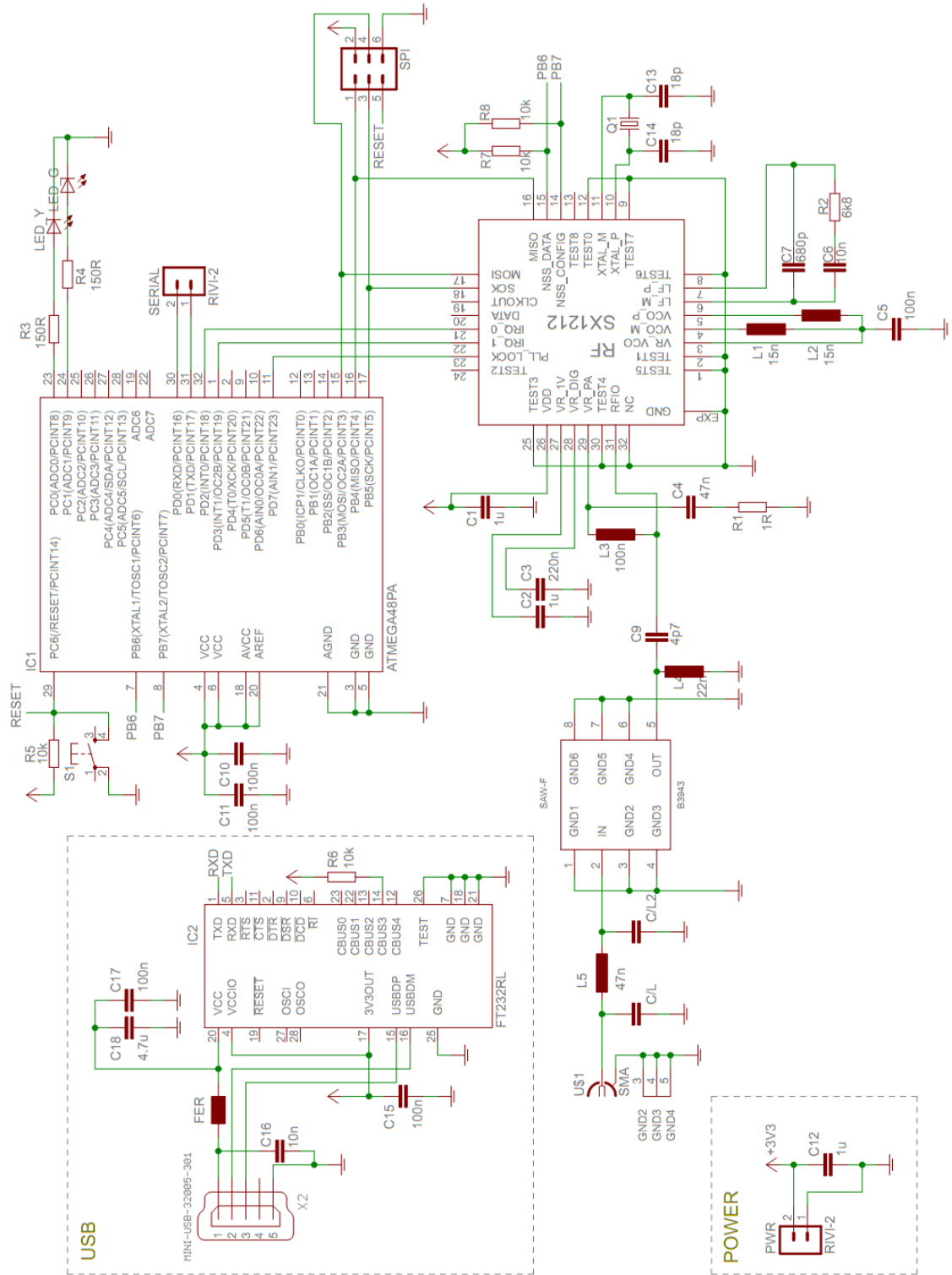
Suuri kantama ja alhainen virrankulutus mahdollistavat lukuisat sovelluskohteet ja näin ollen useita jatkokehitysvaihtoehtoja. Vähäradioliikenteinen toimintamalli ja virrankulutuksen optimoiminen mahdollistavat pienelläkin paristolla tai akulla kuukausien, tai jopa vuosien käyttöajan ilman latausta.

Lähteet

1. Semtech SX1212 datalehti. 2009. Verkkodokumentti. Luettu 6.12.2010.
<http://www.semtech.com/images/datasheet/sx1212.pdf>
2. ISM-taajuusalue. Wikipedia. Verkkodokumentti. Luettu 6.12.2010.
<http://fi.wikipedia.org/wiki/ISM-taajuusalue>
3. Viestintävirasto, Luvasta vapaiden radiolähetimien yhteistaajuuksista ja käytöstä. 2009. Verkkodokumentti. Luettu 6.12.2010.
<http://www.ficora.fi/attachments/suomiry/511x1FIIk/Viestintavirasto15Z2009M.pdf>
4. Frequency-shift keying. Wikipedia. 2010. Luettu 6.12.2010.
http://en.wikipedia.org/wiki/Frequency-shift_keying
5. Serial Peripheral Interface Bus. Wikipedia. Verkkodokumentti. Luettu 6.12.2010.
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
6. Universal asynchronous receiver/transmitter. Wikipedia. Verkkodokumentti. Luettu 6.12.2010. http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter
7. Atmel ATmega88Pa datalehti. 2010. Luettu 6.12.2010.
http://atmel.com/dyn/resources/prod_documents/doc7530.pdf
8. Epcos B3750 datalehti. 2003.
9. Decoupling capacitor. Wikipedia. Verkkodokumentti. Luettu 30.12.2010.
http://en.wikipedia.org/wiki/Decoupling_capacitor
10. FR-4. Wikipedia. Verkkodokumentti. Luettu 30.12.2010.
<http://en.wikipedia.org/wiki/FR-4>
11. Cyclic redundancy check. Wikipedia. Verkkodokumentti. Luettu 7.2.2011.
http://en.wikipedia.org/wiki/Cyclic_redundancy_check

Liitteet

Piirilevyn kytkentäkaavio



Piirilevyn osasijoittelukuva

