

Nikita Pavlov

Langattoman sensoriverkon kehittäminen opetustarkoitukseen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

9.5.2013

Tekijä Otsikko	Nikita Pavlov Pienimuotoisen langattoman sensoriverkon kehittäminen
Sivumäärä Aika	37 sivua + 4 liitettä 9.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	sulautettu tietotekniikka
Ohjaaja	lehtori Anssi Ikonen
Insinöörityön aiheena oli toteuttaa langaton lähiverkko Metropolia Ammattikorkeakoulun sulautetun tietotekniikan alalle mittaaman erilaisten mittauslaitteiden tiedot ja siirtämään helposti antureiden ja päätelaitteen välillä. Insinöörityössä tutkittiin nRF24L01-lähettimien kykyjä ja ominaisuuksia asetetun tavoitteen saavuttamiseksi. Kyseisiä lähtimiä käytetään muissakin projekteissa koulun sisällä, ja opinnäytetyössä käytettiin aiemmin kerättyjä kokemuksia ja onnistumisia, sekä tutkittiin tulevaisuuden mahdollisuuksia järjestelmän käyttämisestä tietotekniikan opetuksessa. Tuloksena toteutettiin kolmella lähettimellä ja yhdellä vastaanottimella toimiva pienimuotoinen lähiverkko. Verkon toiminnan laadun todettiin olevan heikkotasoinen, ja näin ollen pohdittiin erilaisia menetelmiä järjestelmän parantamiseksi. Järjestelmän asetusten uudelleenmäärittely parantaisi yleisen toiminnan. Projektista saatiin reilusti kokemusta, ja saatua tietoa voidaan hyödyntää muissakin projekteissa uudenlaisten tavoitteiden saavuttamiseksi.	
Avainsanat	Lähetin, anturi, verkko

Author Title	Nikita Pavlov Development of small scale wireless sensor network
Number of Pages Date	37 pages + 4 appendices 9 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Embedded Systems
Instructor	Anssi Ikonen, Lecturer
The subject of the thesis was to execute a wireless communication network for embedded systems studies of Metropolia University of Applied Sciences, which based on sending measured information from various measurement devices effortlessly between sensors and terminal. In this thesis the nRF24L01 transceivers' abilities and qualities were examined in regards of achieving the set objective. These particular senders are used in other projects inside the school and earlier experiences and successes were utilized in the project. Also future possibilities were explored, and the various options of the system being used in different field of IT-education. As a result a small and compact network of three transmitters and one receiver was developed. The networks functionality was regarded as poor, and hence the ways to enhance the system are addressed. The reconfiguration of system settings improved overall functioning. The project gave a lot of new experience, and the information gain can be utilized in several other projects with accomplishment of new goals.	
Keywords	Transceiver, sensor, network

Sisällys

Lyhenteet

1	Johdanto	1
2	Langaton anturiverkko	1
2.1	Anturiverkon perusteet	1
2.2	Laitearkkitehtuuri	2
2.3	Verkko-ohjelmisto	3
2.4	RF-Lähetin	5
3	Projektin fyysiset laitteet ja niiden kuvaus	5
3.1	Cypress PSOC	5
	PSoC-arkkitehtuuri	7
	Digitaalinen järjestelmä	9
	GDI-lohko	10
	RDI-lohko	10
	Digitaaliset lohkot	11
	PSoC Designer-ohjelman toiminta	12
3.2	nRF24L01-tiedonsiirtomoduuli	14
	Moduulin rakenne	14
	Yhteyden muodostaminen nRF24L01-piiriin	15
	SPI-väylän käskykanta	17
	Sisäiset rekisterit	19
4	Pääohjelman toteutus	21
4.1	Lähetin	21
4.1.1	Käytetyt moduulit	21
4.1.2	Pääohjelma	24
4.2	Vastaanotin	27
4.2.1	Käytetyt moduulit	27
4.2.2	Pääohjelma	28
5	Laitteen käyttö	30

5.1	Esivalmistelu	30
5.2	Terminaalin käyttö	31
6	Yhteenveto	32
6.1	Projektin onnistumisen analysointi	32
6.2	Ongelmat	33
6.3	Tulevaisuuden näkymät	34
	Lähteet	36
	Liitteet	
	Liite 1. Lähettimen pääkoodi	
	Liite 2. Vastaanottimen pääkoodi	
	Liite 3. Lähettimen vuokaavio	
	Liite 4. Vastaanottimen vuokaavio	

Lyhenteet

ACK	<i>Acknowledgement</i> . Tietoverkkoteknologiassa kuittaus lähetetyn viestin vastaanottamisesta.
ADCINC	<i>Incremental Analog-to-Digital Converter</i> . Signaalin analogisen muodon muuttamisessa digitaalseksi käytettävä muunnin.
CE	<i>Chip Enable</i> . Sirunsallinta mahdollistaa valinnan eri oheislaitteiston joukosta.
COM	<i>Communication Port</i> . Sarjaportti on kahden tietokonelaitteen väliseen tietoliikenteeseen tarkoitettu tietoliikenneportti. Tätä voidaan käyttää esimerkiksi mikrokontrollerin ja tietokoneen väliseen kommunikointiin.
CRC	<i>Cyclic Redundancy Check</i> . Tarkisteavaimen luontiin tarkoitettu algoritmi.
CSN	<i>Chip Selected Not</i> . Piirinvalintasignaali Master-laite määrittelee tiedonsiirtoparametrit.
GDI	<i>The Global Digital Interconnect</i> . Globaali digitaalinen yhteenliitos.
GIE	<i>Global Input Even</i> . Parillinen globaalinen syöttöväylä.
GIO	<i>Global Input Odd</i> . Pariton globaalinen syöttöväylä.
GND	<i>Ground</i> . Maadoitus on sähkölaitteen liittäminen maahan johtimella, jolloin potentiaali laitteen ja johtimen välillä muodostuu yhtä suureksi.
GOE	<i>Global Output Even</i> . Parillinen globaalinen lähtöväylä.
GOO	<i>Global Output Odd</i> . Pariton globaalinen lähtöväylä.
IDE	<i>Integrated Design Environment</i> . Ohjelmointiympäristö on ohjelma, jolla suunnitellaan ja toteutetaan ohjelmistoa.

IRQ	<i>Interrupt request.</i> Keskeytyspyyntö sallii oheislaitteen pyytää prosessoria keskeyttämään toimintansa ja tekemän tuon laitteen pyytämiä toimenpiteitä.
LCD	<i>Liquid-Crystal Display.</i> Nestekidenäyttö perustuu nestekiteiden valonheijastusominaisuuksien muutoksiin sähkökentässä.
M8C	Parannettu 8-bittinen mikroprosessori. Se tukee 8-bittisen käskyjen toimintaa ja se on optimoitu olemaan pieni ja nopea.
MISO	<i>Master Input Slave Output.</i> Dataväylä, jossa isäntäkone vastaanottaa dataa ja orjakone lähettää dataa.
MOSI	<i>Master Output Slave Input.</i> Dataväylä, jossa isäntäkone lähettää dataa ja orjakone vastaanottaa dataa.
MUX	<i>Multiplexer.</i> Multiplekseri eli ottovalitsin, on kombinaatiopiiri, joka tiettyjen valintaehtojen mukaan yhdistää monesta binääridataisesta otosta yhden kerrallaan antoon. Valintaa ohjaa joukko valintaottoja.
PGA	<i>Programmable-gain amplifier.</i> Ohjelmoitava vahvistin.
PRS	<i>Pseudo Random Sequence.</i> Näennäissatunnaissekvenssi on sekvenssi, jossa kaikki sen osat ovat suhteellisen riippumattomia toisistaan.
PSoC	<i>Programmable System-on-Chip.</i> Ohjelmoitava yhden piirin järjestelmä, joka mahdollistaa laitteen analogiaosan sulauttamisen samaan koteloon mikrokontrollerin kanssa.
PuTTY	Ohjelma, jolla voi ottaa pääteyhteyksiä Windows-tietokoneesta muihin tietokoneisiin.
PWM	<i>Pulse Width Modulation.</i> Pulssileveysmodulaatio on modulointitapa, jolla kuormaan menevää jännitettä säädetään muuttamalla pulssisuhdetta niin, että lähtösignaalin keskiarvo värähtelynjakson ajalta laskettuna on sama kuin modulointisignaalin arvo.

RAM	<i>Random Access Memory</i> . Käyttömuisti tai keskusmuisti on ohjelmien työmuisti, johon latautuvat suoritettavat sovellukset.
RDI	<i>The Row Digital Interconnect</i> . Rivillinen digitaalinen yhteenliitos
RF	<i>Radio Frequency</i> . Radiotaajuus.
RS232	<i>Recommended Standard 232</i> . Katso: COM.
RX	<i>Reception</i> . Radiotaajuuden vastaanoton lyhenne.
SCK	<i>Serial Clock</i> . Kellopulssi, joka antaa piirille tahdin.
SPI	<i>Serial Peripheral Interface Bus</i> . Tiedonsiirrossa käytettävä laite. Kyseisessä väylässä on neljä eri signaalijohdinta.
SPIM	<i>Serial Peripheral Interconnect Master</i> . Cypress PSoC:in moduuli, joka vastaa SPI-väylän toiminnasta. Toimii nimenmukaisesti master-laitteena.
TX	<i>Transmission</i> . Radiotaajuuden lähettämisen lyhenne.
UART	<i>Universal Asynchronous Receiver Transmitter</i> . Sarjaliikennepiiri, jonka tärkein tehtävä on dataväylältä rinnakkaismuotoisen datan luku siirtorekisteriin, josta se muutetaan edelleen sarjamuotoon ja päinvastoin.
Vcc	<i>Voltage</i> . Positiivinen syöttöjännite.

1 Johdanto

Insinööriyön tavoitteena on tutkia verkkokaupasta ostettujen langattomaan datansiirtoon tarkoitettujen antureiden soveltuvuutta langattoman lähiverkon rakentamisessa. Työn tarkoituksena oli kartoittaa tekijän osaamista kyseisellä alalla ja mahdollisesti toimia esimerkkinä vastaavanlaisille projekteille.

Metropolia Ammattikorkeakoulun sulautetun tietotekniikan koulutus tarjoaa langattomia lähettimiä koskevaa opetusta ja opetuksen kannalta haluttiin suunniteltavan ja rakennettavan pienessä tilassa käytettävä langaton lähiverkko, jolla voi siirtää mitattuja data-arvoja mittauslaitteista tietokoneeseen tai vastaavalle päätteelle jälkikäsiteltäväksi.

NRF24L01-malliset lähettimet ovat käytössä koulun erilaisissa projekteissa, mutta usein niiden käyttö rajoittuu kahden lähettimen käyttöön, esimerkiksi erilaisten robottien ohjauksessa.

Aihe on valittu sen perusteella, että se liittyi tekijän opiskelu-alaan ja mahdollisuuteen käyttää hyväksi aikaisempaa kokemusta.

Insinööriyössä esitellään ensin aiheen taustaa. Seuraavaksi käydään läpi käytettyä laitteistoa ja eri komponentteja. Tämän jälkeen kerrotaan työn eri vaiheista. Lopuksi pohditaan työn onnistumista, työn etenemisessä esiintyneitä ongelmia sekä projektin tulevaisuuden näkymiä.

2 Langaton anturiverkko

2.1 Anturiverkon perusteet

Langaton anturiverkko on IT-kommunikaation tärkeä osa-alue tiedon hakemisessa. Tilanteissa, joissa johdot ja kaapelointi ovat yleensä epäkäytännöllistä sekä tarvitaan nopea, helposti asennettava ja ylläpidettävä anturiverkko, käytetään langatonta anturiverkkoa. Langaton anturiverkko muodostuu solmuista, jotka on järjestetty yhteistyöverkkoon. Näillä solmuilla on oma laskentateho, ja ne voivat lähettää ja vastaanottaa viestejä langattoman tai kaapeloidun tietoliikenneyhteyden kautta. Solmut havainnoivat

ympäristöä ja välittävät seuratusta kentästä kerättyä tietoa langattoman yhteyden kautta. [15, s.7.]

Yksittäinen verkko voi koostua useista eri topologisista aliverkoista. Verkostoja luokitellaan lähiverkoiksi (LAN) tai laajaverkoksi (WAN). Lähiverkko toimii yhden rakennuksen sisällä, kun taas laajaverkko mahdollistaa yhteydenoton rakennusten välillä.

Langattoman anturiverkon toiminnan vaatimuksiin kuuluvat sen vikasetokyky, toimintaikä, skaalautuvuus, reaaliaikaisuus, turvallisuus ja verkon tuotantokustannukset. Verkon toiminta on säilytettävä huolimatta sen dynaamisista luonteesta ja epäonnistumisista vaikeissa olosuhteissa, jotka voivat johtua niin ikään paristojen ehtymisestä tai ulkoisista häiriöistä.

Anturiverkon solmut ovat akkukäyttöiset, tai energia voidaan tuottaa ympäristöstä esimerkiksi aurinkopaneelia käyttäen. Näiden solmujen ylläpito on vaikeaa. Niinpä energiansäästö ja kuorman tasaus on otettava huomioon langattoman anturiverkon suunnittelussa ja toteutuksessa sekä sen protokollissa ja sovelluksissa. Langattomassa anturiverkossa olevien solmujen määrä on suuri, ja siten sen protokollat on käsiteltävä korkealla tiheydellä. Verkko on tiukasti sidoksissa reaali maailmaan, ja siksi tiukka ajoitus rajoittaa sen ilmaisutoimintaa, käsittelyä ja viestintää. Anturiverkkojen turvallisuuden tarve on nähtävissä erityisesti terveydenhuollossa sekä turvallisuus- ja sotilaallisissa sovelluksissa. Useimmat sovellukset välittävät tietoa, joka on henkilökohtaista tai luotamuksellista. Kuten aiemmin mainittiin, anturiverkon solmujen määrä on melko suuri ja kun solmuissa loppuu virta, ne korvataan uusilla. [14, s.5.]

Kokonaisuudessa langaton anturiverkko koostuu yksittäisistä sulautetuista järjestelmistä, jotka pystyvät interaktioon ympäristönsä kanssa käyttäen erilaisia antureita. Anturit käsittelevät paikallisesti tietoa ja kommunikoivat tätä tietoa langattomasti naapuriantureiden kanssa. [14, s.6.]

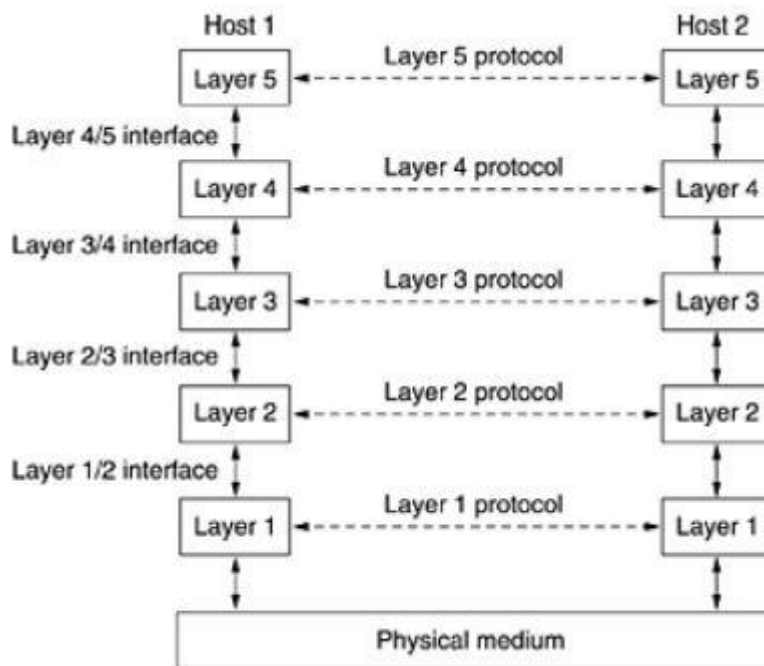
2.2 Laitearkkitehtuuri

Kaupalliset alustat tyypillisesti koostuvat kolmesta osasta joko yksittäiselle levyille tai osana sulautettuja järjestelmiä. Langattomat moduulit ovat anturiverkon avaintekijöitä, sillä ne ovat viestintävalmiita ja sisältävät ohjelmoitavan muistin, jossa sovelluksen

koodi sijaitsee. Tyypillisesti moduuli koostuu mikrokontrollerista, virtalähteestä, muistiyksiköstä ja se saattaa sisältää muutamia antureita. Anturilevy on asennettu moduulille ja se on sulautettu osaksi sensoreita. Vaihtoehtoisesti antureita voidaan liittää langattomaan moduuliin. Ohjelmointipiiri, joka tunnetaan myös yhdyskäytäväpiirinä, tarjoaa useita liitäntöjä kuten Ethernet, WiFi, USB ja sarjaportti, jotka mahdollistavat kytkennän useisiin moduuleihin ja näiden ohjelmointiin tai datan keräämiseen kyseisistä moduuleista. [14, s.7.]

2.3 Verkko-ohjelmisto

Kuvassa 1 on esitetty tyypillinen verkko-ohjelmiston rakenne. Suunnittelun monimutkaisuuden vähentämiseksi useimmat anturiverkot ovat pino-rakenteisia ja muodostuvat eri kerroksista ja tasoista, kukin aina edellisen alapuolella. Kerrosten määrä, nimi, sisältö ja sen funktio poikkeavat verkosta toiseen. Jokaisen kerroksen tehtävänä on tarjota tiettyjä palveluita ylemmille kerroksille ja suojata nämä kerrokset siltä tiedolta, miten tarjotut palvelut on otettu täytäntöön. [14, s.10.]



Kuva 1. Kuvassa on esitetty verkko-ohjelmiston rakenne. [14, s.10].

Yhdellä koneella kerros N pitää yhteyden toisen koneen vastaavanlaisen kerroksen N kanssa. Säännöt ja käytännöt tässä yhteydessä ovat yhteisesti nimeltään kerroksen N protokolla. Periaatteessa protokolla on viestinnän osapuolten välinen sopimus siitä, miten niiden välinen viestintä etenee. Todellisuudessa tietoja ei siirretä suoraan yhden laitteen kerroksesta N toisen laitteen kerrokseen N. Sen sijaan jokainen kerros välittää datan ja ohjaustiedon sen alla olevaan kerrokseen, aina alimpaan kerrokseen asti. Ykköskerroksen alapuolella on fyysinen väline, jonka avulla todellinen viestintä tapahtuu. [14, s.11.]

Fyysinen kerros määrittelee laitteen fyysiset ja sähköön liittyvät ominaisuudet. Erityisesti se määrittelee suhteen laitteen ja siirtovälineen välillä. Siirtoväline voi olla kuparinen tai optinen kaapeli. Tämä sisältää myös nastojen asettelua, jännitteen, kaapelin määrittelyn, solmukohtia, toistimet, verkkosovittimet jne. Suunnittelussa tärkeintä on varmistaa, että lähetetty 1-bitti saapuu perille 1-bittinä eikä 0-bittinä.

Tiedonsiirtoprotokollan päätehtävä on raaka-lähetyslaitoksen muuttaminen linjaksi ilman verkkokerroksen siirtovirheitä. Tämä saavutetaan sallimalla lähettäjälle syötetyn datan hajottamista datakehyksiksi (data frames), ja välittämällä peräkkäin nämä kehykset. Jos palvelu on luotettava, vastaanotin vahvistaa jokaisen kehyksen oikean vastaanoton lähettämällä takaisin kuittauskehyksen. Usein virtaussäätö ja virheiden käsittely ovat osana kerrosta. Lähetysverkoissa on lisäksi datalinkkikerroksen ongelma: miten hallitaan jaetun kanavan pääsyä. Datalinkkikerroksella on olemassa erityinen alikerros, nimeltään Medium Access Control -kerros (MAC), joka käsittelee edellistä ongelmaa. [14, s.13.]

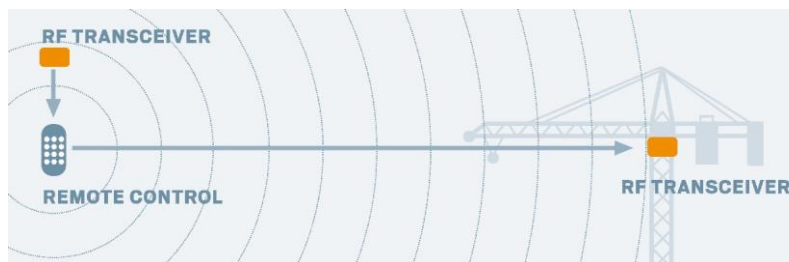
Verkkokerros ohjaa aliverkkoa. Verkon suunnittelun tärkein osuus on määrittää, kuinka paketit reitittyvät lähteestä kohteeseen. Reitit voivat olla erittäin dynaamisia, tai ne voivat määräytyvät uudelleen jokaiselle paketille, samalla näyttäen senhetkistä verkon kuormitusta. Verkkokerros näyttää myös tarjottujen palvelujen laadun, kuten esim. viiveen tai kulkuun kulutetun ajan.

Siirtokerroksen perustehtävä on hyväksyä yläpuolelta tulevaa dataa, tarvittaessa jakaa sitä pienempiin osiin, siirtää sitä verkkokerrokselle ja varmistaa, että kaikki palaset saapuvat oikein toiseen päähän. Siirtokerros määrittää myös palveluiden tarjontaa verkon käyttäjille.

Sovelluskerros sisältää erilaisia protokollia, jotka käyttäjä yleisesti tarvitsee. [14, s.14]

2.4 RF-Lähetin

RF-lähetimet (engl. RF-transceiver) ovat elektronisia laitteita, jotka voivat sekä lähettää että vastaanottaa radiosignaaleja. RF-lähetin yhdistää lähettimen ja vastaanottimen toimintoja. Nämä lähetimet ovat käytössä monissa eri video-, ääni- ja data-sovelluksissa, kuten radiossa, langattomassa puhelimessa, kännyköissä ja langattomaan verkkoon kytketyssä tietokoneessa [5]. Kuvassa 2 on esitetty yksinkertainen RF-lähettimen käyttötapa. RF-lähetin koostuu antennista, virittimestä ja halutun taajuuden ilmaisimesta. Kun lähetin toimii vastaanottimena, sen antenni kaappaa radioaaltoja, viritin eristää halutun taajuuden ja ilmaisin erottaa informaation aallosta. Kun lähetin toimii lähettimenä, sen oskillaattori luo virrasta sini-aaltoja, jotka koodataan ja lähetetään radiosignaaleina. [6].



Kuva 2. RF-lähettimen yksikertaisen toiminnan esittely.

Lähettimen etuina ovat sen edullisuus, pitkä kantama, erityisen pieni kulutus ja joustavuus. Nämä lähetimet mahdollistavat vähävirtaisen langattoman viestinnän luomisen osaksi sulautettuja tuotteita. [16].

3 Projektin fyysiset laitteet ja niiden kuvaus

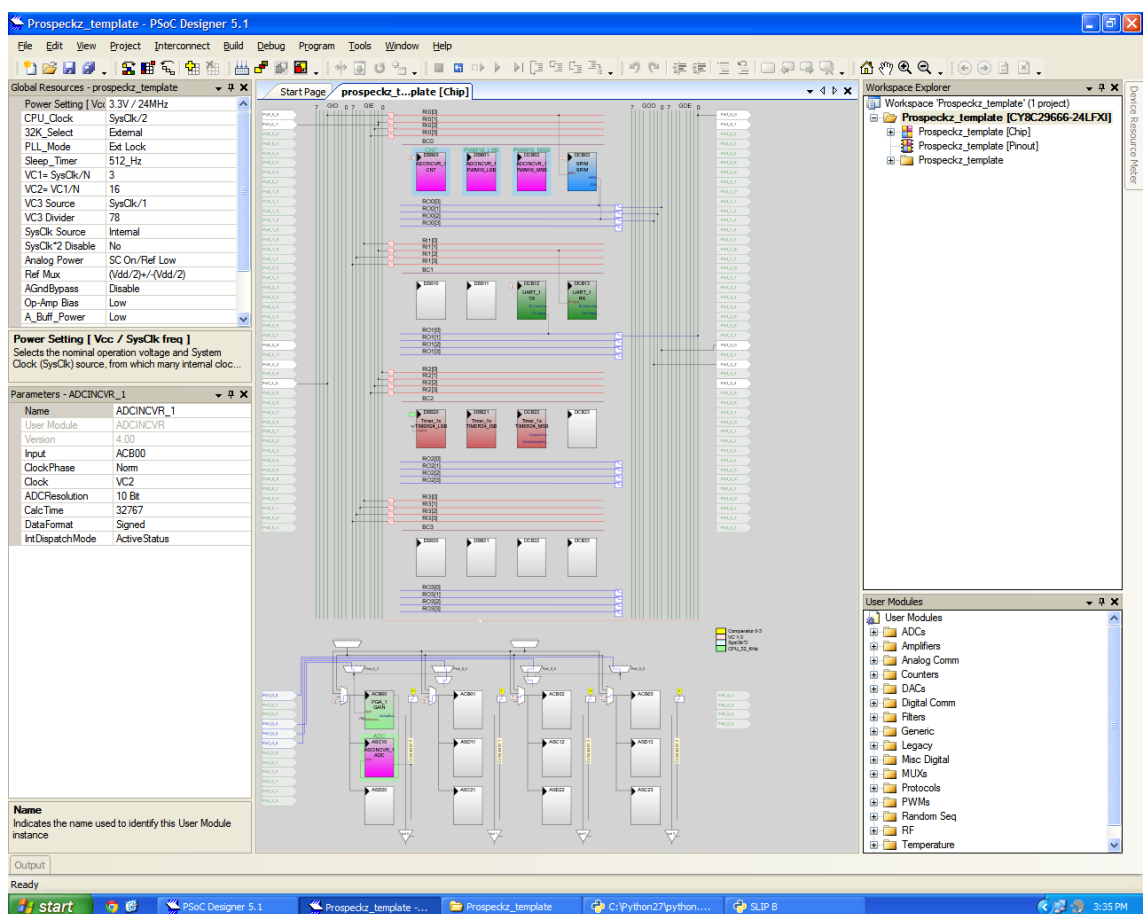
3.1 Cypress PSOC

Cypress PSoC mahdollistaa laitteen analogiaosan sulauttamisen samaan koteloon mikrokontrollerin kanssa. Piiri sisältää analogisia ja digitaalisia lohkoja, joita voidaan käyttää suunnitteluohjelmiston valmiissa tai omissa ratkaisuissa. PSoC sisältää Cyp-

ressin oman 8-bittisen M8C-ytimen. PSoC tarjoaa järjestelmän suunnittelijalle analogisen integraation, joustavan IO-reitityksen ja täydellisen virrankulutuksen hallinnan.

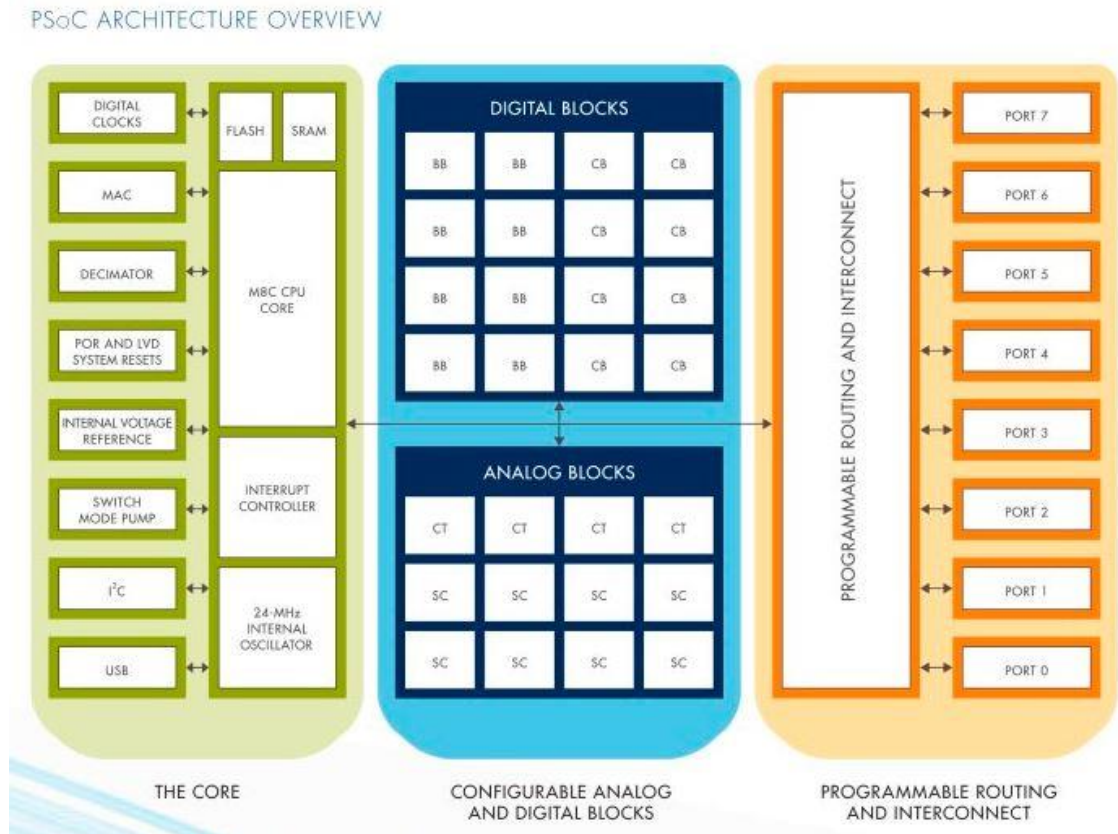
PSoC sisältää joukon analogisia lohkoja, jotka voivat toimia yhdessä tai yksin. Cypressin valmistetut moduulit on suunniteltu analogisia lohkoja varten, ja ne antavat kehittäjälle mahdollisuuden tehdä analogisia toimintoja. Moduuleita ovat muun muassa kais-tanpäästösuodattimet, A/D ja D/A-muuntimet jne. Analogisten lohkojen lisäksi PSoC:ssa on käytössä myös digitaaliset lohkot. Näille lohkoille on suunniteltu digitaali-set moduulit, kuten ajastimet, laskurit ja SPI (Serial Peripheral Interface).

PSoC muokataan vastaamaan erityisiä vaatimuksia käyttämällä Cypressin kehittämä PSoC Designer -suunnitteluympäristö. Ohjelmistolla kehitetään sovelluksia käyttämällä ennalta määrätyt analogiset ja digitaaliset oheislaitteet. Koko toiminta perustuu drag-and-drop-menetelmään, mikä nopeuttaa sovelluksen kehittämistä. Kuvassa 3 on esitetty tyypillinen näkymä ohjelmasta.



Kuva 3. PSoC-Designer -suunnitteluympäristö Windows käyttöjärjestelmässä.

PSoC-arkkitehtuuri



Kuva 4. PSoC-järjestelmän arkkitehtuuri.

PSoC koostuu neljästä eri pääelementistä. Kuvassa 4 on esitetty nämä edellä mainitut elementit. Vasemmalta oikealle ensimmäisenä on esitetty PSoC-ydin, ja sen rinnalla toimivat järjestelmäresurssit. Kuvan keskellä on esitetty digitaalinen ja analoginen lohko. Viimeisenä kuvan oikealla puolella on esitetty ohjelmoitava reititys ja yhdistäminen. [13, s.5.]

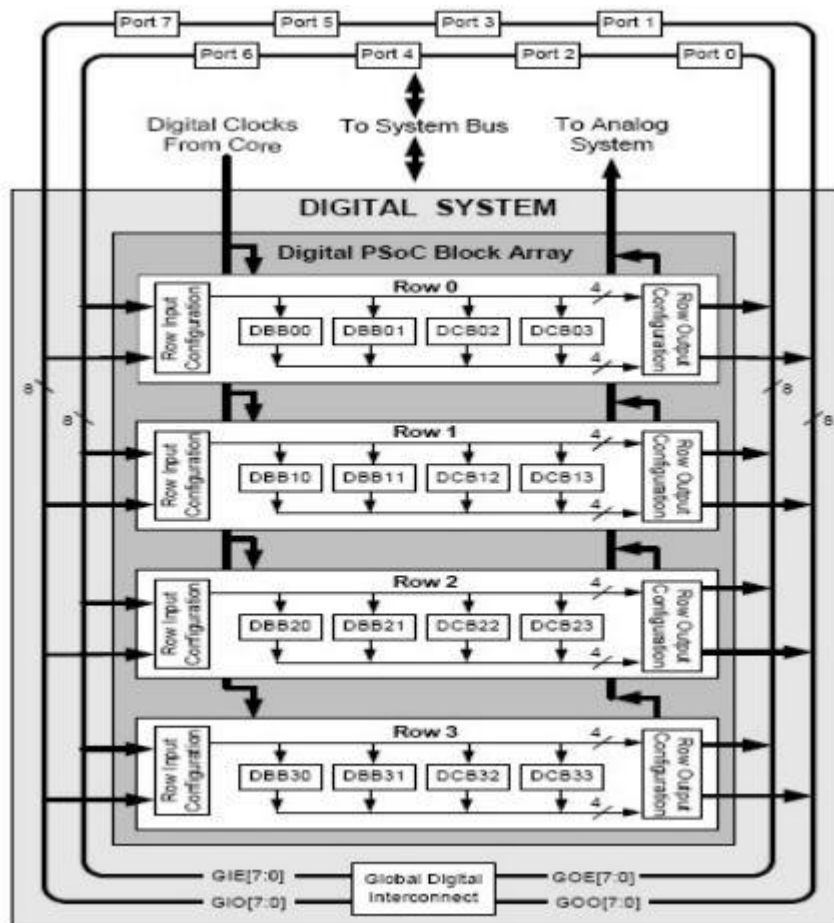
PSoC-perheen ydin on yhteinen kaikille laitteille. Ydin sisältää patentoidun M8C-suorittimen ja se toimii 24 MHz:n kellotaajuudella. Lisäksi ydin sisältää väli- ja flash-muistin, useita kellolähteitä, keskeytysohjaimen ja Watchdog-ajastimen. Järjestelmäresurssit tarjoavat lisää ominaisuuksia PSoC-ympäristöön. Näitä ovat kerto- ja yhteenlaskun suorittamiseen tarvittava moduuli, jakamiseen tarvittava moduuli, I2C-väylä, vaihtotilan kytkin ja täydellä nopeudella toimiva USB-väylä.

Digitaalinen järjestelmä koostuu useista digitaalisista lohkoista. Jokainen digitaalinen lohko voidaan määrittää suorittamaan yhden useista digitaalisista funktioista, kuten ajastin, UART (Universal Asynchronous Receiver Transmitter) tai SPI.

Analoginen järjestelmä koostuu puolestaan useista analogisista lohkoista. Samalla tavalla jokainen analoginen lohko voidaan määrittää suorittamaan yhtä useista analogisista funktioista, kuten ADC (Analog-to-Digital Converter) tai DAC (Digital-to-Analog Converter).

Kaikki edellä mainitut elementit on kytketty toisiinsa muokattavan digitaalista ja analogista järjestelmää yhdistävän kytkentäkentän kautta, mikä mahdollistaa käytettävien lohkojen liittämisen toisiinsa toiminnallisesti. Niitä voi ohjata mistä tahansa piirillä saatavilla olevasta I/O-nastasta. PSoC:in konfiguroiva ydin muodostuu analogisesta ja digitaalisesta järjestelmästä sekä niiden välisestä yhteenliittymästä. [13, s.6.]

Digitaalinen järjestelmä



Kuva 5. PSoC:in Digitaalisen järjestelmän lohkokaavio (13, s.41).

Kuvassa 5 esitetty konfiguroiva digitaalinen järjestelmä koostuu kolmesta pääkomponentista: globaalista digitaalisesta yhteydestä GDI, peräkkäisestä digitaalisesta yhteydestä RDI ja digitaalisista lohkoista.

Digitaaliset lohkot voidaan määrittää suorittamaan yhden seitsemästä saatavilla olevasta oheislaitetoiminnosta. GDI- ja RDI-komponentit muodostavat yhteyden, joka puolestaan mahdollistaa signaalin reitityksen digitaalisiin lohkoihin ja pois.

RDI-komponentti yhdistää digitaalisten lohkojen syöttö- ja lähtöväylät GDI-komponenttiin. GDI puolestaan mahdollistaa RDI:n kytkennän piiriin I/O-nastoihin. Yhdessä RDI- ja GDI-komponentit yhdistävät digitaaliset lohkot keskenään sekä mahdollistavat näiden yhdistämisen sirun I/O-nastoihin. [13, s.42.]

Digitaalisten lohkojen ja digitaalisen yhteyden määrittely tapahtuu ohjelmoimalla erilaiset RAM-pohjaiset rekisterit. Näiden rekistereiden tarkat määritelmät voidaan säilyttää piirin flash-muistissa ja ladata tarvittaessa RAM-muistiin. Tämä mahdollistaa PSoC:in dynaamisen uudelleenmäärittelyn. [13, s.7.]

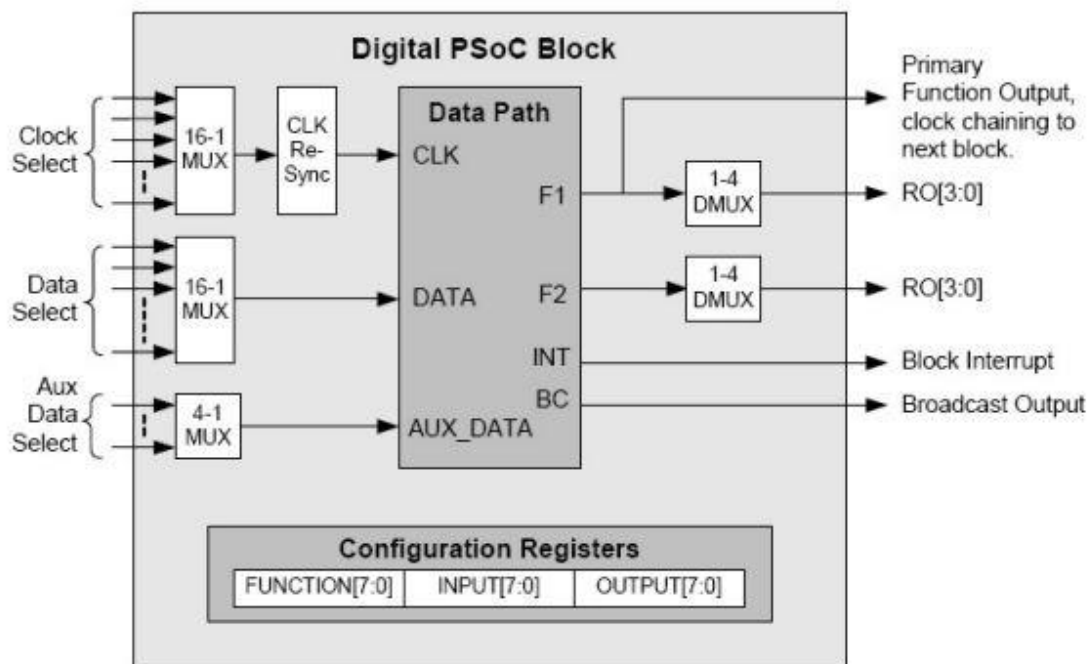
GDI-lohko

GDI-lohko koostuu neljästä 8-bittisestä väylästä. Väylät mahdollistavat signaalin reitittämisen laitteen piirin nastoihin tai poispäin. Kaksi käytettyä väylää ovat syöttöväyliä, nimeltään pariton globaalinen syöttö (GIO) ja parillinen globaalinen syöttö (GIE). Kaksi lähtöväylää ovat taasen nimeltään pariton globaalinen lähtö (GOO) ja parillinen globaalinen lähtö (GOE). Parittomat väylät ovat yhteydessä parittomiin I/O-portteihin eli portteihin 1, 3, 5 ja 7. Parilliset väylät ovat puolestaan yhteydessä parillisiin I/O-portteihin eli portteihin 0, 2, 4 ja 6. [13, s.8.]

RDI-lohko

RDI-lohko koostuu neljästä peräkkäisestä syöttöverkosta ja neljästä peräkkäisestä lähtöverkosta. Syöttöverkkojen työtehtävänä on reitittää signaalit GDI:stä digitaalisten lohkojen syöttöihin. Jokainen peräkkäinen syöttö sisältää 4-to-1-ottovalitsimen, jonka syöttöihin on kytketty neljä GDI-riviä. Ottovalitsimen lähdöt ovat nimeltään Row Inputs. Tämä mahdollistaa sen, että jokainen yksittäinen Row Input voi olla kytkettynä yhteen neljästä saatavilla olevasta GDI-linjasta. [13, s.8.]

Digitaaliset lohkot



Kuva 6. PSoC:in digitaalinen lohko määrittelyineen (13, s.10).

Kuvassa 6 on esitetty PSoC:in digitaalisen lohkon lohkokaavio. Digitaalinen lohko koostuu datapolusta (data path), syöttömultipleksereista (mux), lähtödemultipleksereista (dmux), konfiguraatiorekisteristä (configuration registers) ja ketjuttavista signaalista. RAM-pohjaisten määrittelyrekistereiden avulla voidaan määrätä suorittamaan jokin seuraavista seitsemästä toiminnasta: ajastin, laskuri, pulssileveysmodulaattori (PWM), näennäissatunnaissekvenssi (PRS), syklinen redundanssitarkistus (CRC), SPI tai kaksisuuntainen UART. Jokainen digitaalinen lohko tarjoa 8-bittisen resoluution. Useita digitaalisia lohkoja voidaan ketjuttaa tuottamaan yhä suuremman tarkkuuden toimintoja. [13, s.10.]

Digitaalisessa lohkossa on kolme syöttöä: kello, data ja avustaja. Tyypillisesti jokaisessa funktiossa on kello, sekä yksi 16 mahdollisesta datan syötöstä, 16-to-1-syöttömultiplekseria käyttäen. Avustaja valitaan 4-to-1-syöttömultiplekserin kautta. Avustajan syöttö on erityisesti suunniteltu toimimaan digitaalisen lohkon SPI Slave-Select-signaalina, ja se on määritelty Slave-laitteen SPI-tilassa. [13, s.10.]

Digitaalisessa lohkoissa on viisi ulostulolähtöä: pääulostulo, lisäulostulo, ketjuttava signaali ja lohkokeskeytys. Pää- ja lisäulostulot määritellään käyttämään RDI-riviulostuloväylää 1-to-4-lähtömultiplekserin kautta. Ketjutussignaali levittää tietoa yhdestä digitaalisesta lohkoista seuraavaan. Tämä mahdollistaa suurella resoluutiolla toimivien funktioiden toteutuksen. Lohkokeskeytys riippuu funktiosta ja tätä käytetään PSoC-ytimelle suunnatussa vektoripohjaisessa keskeytyssignaaliin. [13, s.11.]

Syöttösignaalit, lähtösignaalit ja datapolun funktio on määriteltävissä käyttäen seitsemää RAM-perusteista rekisteriä: kolmea datarekisteriä, ohjausrekisteriä, funktiorekisteriä, syöttörekisteriä ja lähtörekisteriä. [13, s.11.]

Useiden edellä mainittujen rekistereiden määrittely on funktoriippuvaista. Esimerkiksi ohjausrekisteri sisältää funktio-ohjauksen ja statusbitin valitulle funktiolle. Riippuen valitusta funktiosta määrittely jokaiselle ohjausrekisterin bitille muuttuu. Samoin kolmen datarekisterin määrittely muuttuu digitaalisen lohkon valitusta funktiosta riippuen. [13, s.11.]

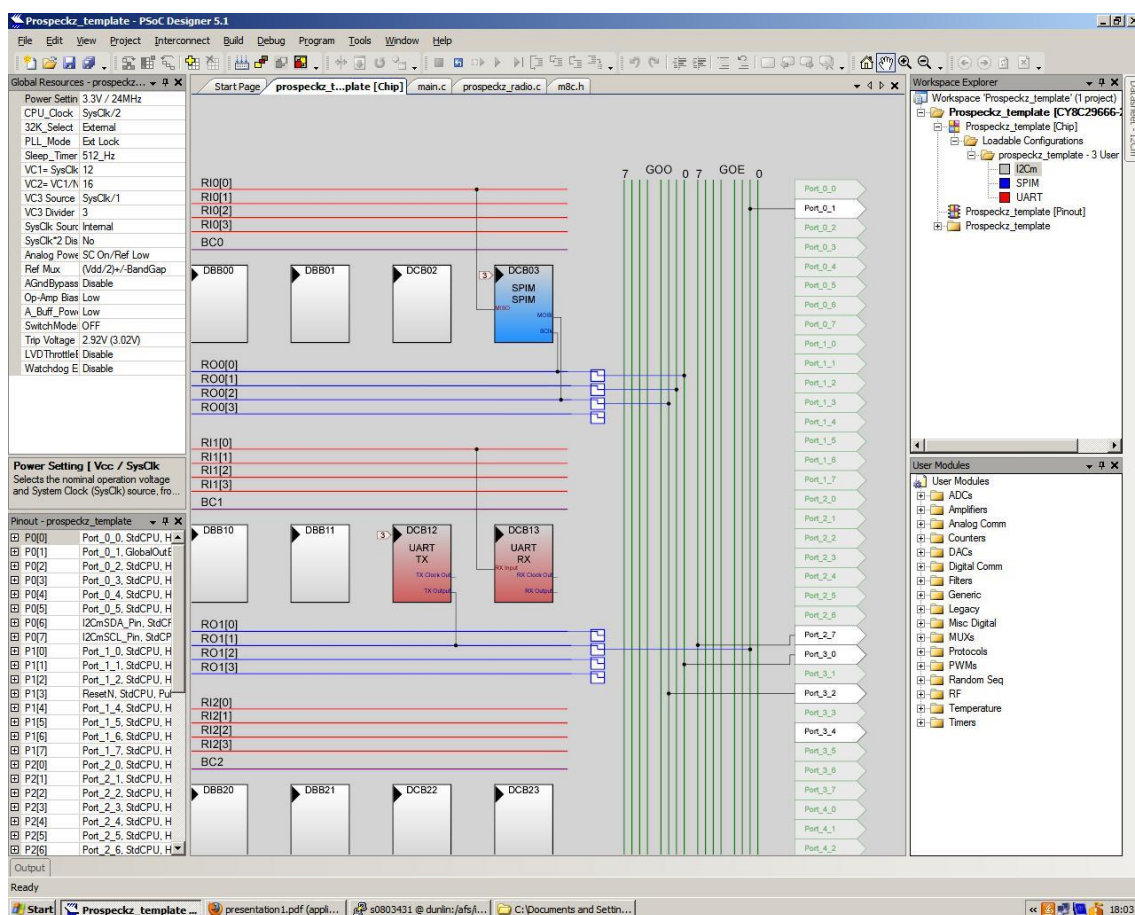
Funktiorekisteri mahdollistaa digitaalisen lohkon funktion valinnan. Syöttörekisteri mahdollistaa kolmen digitaalisen lohkon syöttösignaalin määrittelyn. Lähtörekisteri mahdollistaa digitaalisen lohkon lähtösignaalien vahvistamisen. [13, s.12.]

Koska kaikki seitsemän rekisteriä ovat RAM-pohjaisia, useat digitaalisen lohkon asetukset voidaan tallentaa flash-muistiin ja myöhemmin sovelluksen edellyttäminä ladata uudelleen. Tämä mahdollistaa digitaalisen lohkon uudelleenmäärittelyä. [13, s.12.]

PSoC Designer-ohjelman toiminta

PSoC Designer on integroitu kehitysympäristö (IDE), ja sitä käytetään sovelluksien kehittämiseen PSoC-laitteille. Ohjelmisto sisältää laite-editorin, sovelluseditorin, C- ja Assembly-ohjelmointikielten kääntäjän, sovelluksen rakentajan ja reaaliaikaisen virheenkorjaajan. [13, s.14]

Laite-editori erottaa PSoC Designer -kehitysympäristön muista mikrokontrollereihin kehittämiseen tarkoitetuista kehitysympäristöistä. PSoC Designer sisältää graafisen käyttöliittymän, ja sen avulla määritellään koko PSoC-ympäristö. Käyttöliittymä luo sovellukselle automaattisesti tarvittavan sovellusliittymän. [13, s.14.]



Kuva 7. PSoC Designer-kehitysympäristön valittavat käyttömoduulit.

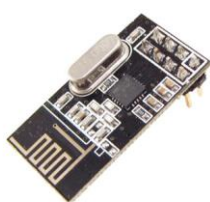
Kuvassa 7 on esitetty tilannekuva laite-editorista. PSoC-laitteen globaalit resurssit ovat määritettävissä ohjelman vasemmassa yläkulmassa. Syöttö- ja lähtöportit määritellään vasemmassa alakulmassa sijaitsevassa osiossa. Digitaaliset ja analogiset lohkot valitaan oikeasta alakulmasta ja sijoitetaan sekä reititetään editorin keskellä sijaitsevalla alueella. Kaikki tämä tapahtuu graafisella tasolla. [13, s.15.]

Kun malli on määritelty, asetettu ja reititetty, laite-editori luo kaikki malliin perustuvat sovellustiedostot. Nämä tiedostot sisältävät sovellusliittymän ja keskeytyspalvelurutiinit. Laite-editori luo myös sovelluskohtaisen datalehtiön käytetyistä laitteista ja asetuksista. [13, s.15.]

Seuraavaksi sovelluseditoria käytetään sovelluskoodin luomiseen, ja se rakennetaan hex-tiedostoon. Tämä tiedosto ladataan laitteen piiriin flash-muistiin. Flash-ohjelmointiin käytetään PSoC Programmer -työkalua, joka on integroitu osaksi PSoC Designer -kehitysympäristöä. [13, s.15.]

3.2 nRF24L01-tiedonsiirtomoduuli

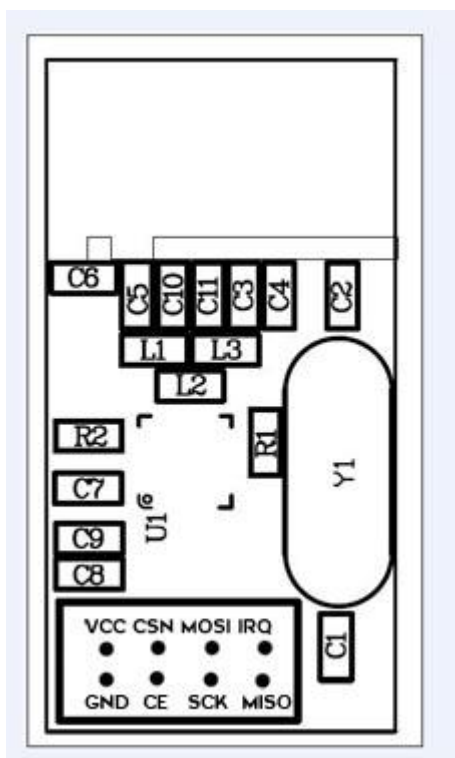
nRF24L01 on pitkälle integroitu, vähävirtainen radiolähetin. Kuvassa 8 on esitetty lähettimen fyysinen ulkomuoto. Kyseinen piiri on jatkoa suosituille nRF2401-sarjan piireille. 24L01 tuo moduulin edellisten versioiden ominaisuuksien lisäksi SPI-liitännän, laiteistolinkkikerroksen, useita dataväyliä ja paljon muita tärkeitä ominaisuuksia. Tämän kaiken lisäksi lähetin on keskimäärin halvempi kuin muut mikrokontrollerit, eikä se vaadi liikaa ulkoisia komponentteja. [8].



Kuva 8. Kuvassa nRF24L01-lähetin.

Moduulin rakenne

Kyseisessä mikropiirissä on käytössä kahdeksan nastaa yhteyden muodostamiseen – Vcc, GND, IRQ, CE – ja neljä SPI-nastaa: CSN, SCK, MISO ja MOSI. Kuvassa 9 on esitetty lähettimen nastojen paikat. Lähettimen nastat sietävät 5 V:n jännitteen, mutta itse lähetin ei voi toimia kyseisellä jännitteellä, vaan ohjekirjan mukaan käyttöjännite vaihtelee 1,9 V:n ja 3,6 V:n välillä. Tästä johtuen tarvittaessa mikropiirin rinnalla tulee käyttää sopivaa jänniteregulaattoria.



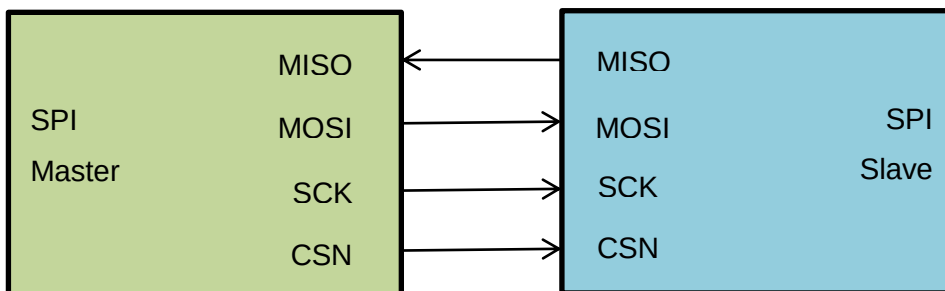
Kuva 9. NRF24L01-lähettimen nastajärjestys. Nastat on esitetty kuvan alimmassa reunassa pienillä mustilla pisteillä.

Datan vastaanottoa ja lähettämistä varten SPI-väylä vaatii toimiakseen neljä nastaa, CSN, SCK, MISO ja MOSI. Kommunikoinnissa täytyy huolehtia, että asetukset ovat isäntä- ja orja-laitteella samat ja että ne voi tarvittaessa muuttaa samoiksi. Jäljellä olevat nastat on sidottu käyttäjän laitteiston SPI-liitäntään vastaavanimisiin nastoihin tai portteihin. Kaksi jäljellä olevaa nastaa ovat CE ja IRQ. CE ohjaa tiedonsiirtoa, kun lähettin on TX- ja RX-tilassa. IRQ on puolestaan keskeytysnasta, ja se on "active-low"-muotoa.

Yhteyden muodostaminen nRF24L01-piiriin

Yhteyden muodostamiseen mikropiiriin ja käyttäjän mikrokontrollerin välillä vaaditaan SPI-liitäntä. SPI-käyttöliittymän avulla voidaan lukea tai kirjoittaa rekisterit sekä lähettää ja vastaanottaa tietoa. Datalehdessä on ristiriitainen tieto maksimaalisen SPI-väylän datanopeuden kannalta. Sivulla 1 todetaan, että suurin mahdollinen nopeus on 8 Mbps, kun taas sivulla 19 suurin mahdollinen nopeus on 10 Mbps. Yksikertaisuuden vuoksi kannattaa asettaa SPI-väylän nopeudeksi 1 Mbps. Nopeuden kasvattaminen on kan-

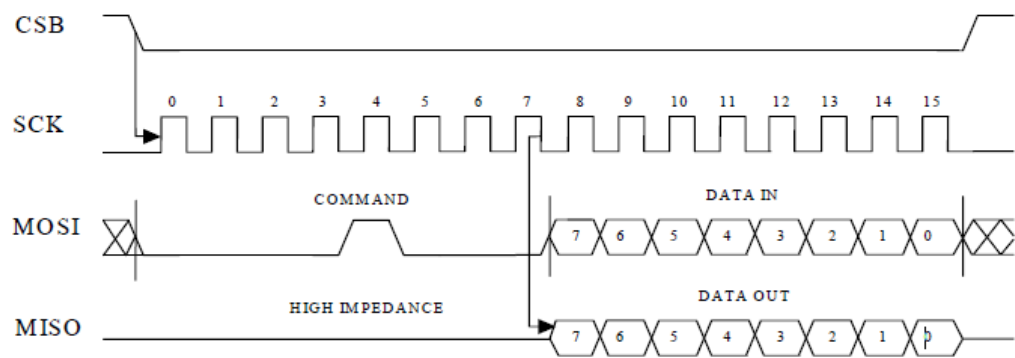
nattavaa, kun tarpeena on siirtää valtava määrä dataa. SPI-väylä täytyy myös asettaa 8-bittisen tiedon siirtämiseen, ja väylän tulee olla määriteltä Master-tyyppiseksi.



Kuva 10. Yksinkertainen SPI-yhteys.

Kuvassa 10 esitetyssä SPI-yhteydessä on käytössä neljä linjaa: MOSI, MISO, SCK ja CSN. Kytkeytetyt laitteet ovat kahta muotoa: Master ja Slave. Master-laite luo signaalin MOSI-linjalle ja Slave-laite ottaa signaalin vastaan. Puolestaan Slave-laite luo signaalin MISO-linjalle ja Master-laite vastaanottaa kyseisen signaalin. Master-laite luo myös kellopulssin SCK-linjalle synkronoidakseen datanlähetyksen kyseisen laitteen ja Slave-laitteen välillä. SS-linja informoi käytettävästä Slave-laitteesta.

Yhteyden muodostamiseen tarvitaan muutamat määrittelyt, ja kaikki toiminnot on esitetty kuvassa 11. CSN nasta on "active-low"-muotoa. Aluksi CSN-nasta täytyy olla nostettuna ylös ("1"-tila) ja tämän jälkeen kyseinen nasta tuodaan alas "0"-tilaan merkiksi siitä, että SPI-väylän kautta tuodaan tietoa 24L01-lähettimeen. CSN-nasta pysyy alhaalla koko tapahtuman ajan. Tämän jälkeen alkaa kellopulssi, jolloin MOSI ja MISO lähettävät dataa kellopulssin laskevalla reunalla. Kun kaikki data on lähetetty tai luettu, nostetaan CSN-nasta ylös "1"-tilaan. Tämän kaiken jälkeen voidaan käsitellä tietoa mikrokontrollerissa.



Kuva 11. Datan lähetys SPI-väylällä. Kuvassa CSB vastaa tekstissä käytettyä CSN (11, s.2).

SPI-väylän käytön hyötynä on sen tarjoama tuki yhtäaikaiselle lähetykselle ja vastaanotolle, mahdollisuus suureen tiedonsiirtonopeuteen, yksinkertainen laitteistotason toteutus ja vähäinen pinnien käyttötarve. Huonona puolena voidaan mainita virheenkorjauksen ja -tarkistuksen puuttuminen ja alttius häiriöille

SPI-väylän käskykanta

Taulukko 1. nRF24L01-lähettimen SPI-käskykanta (17, s.46; 17, s.47).

Käskyn nimi	Käskytavu (bittimuoto)
R_REGISTER	000A AAAA
W_REGISTER	001A AAAA
R_RX_PAYLOAD	0110 0001
W_TX_PAYLOAD	1010 0000
FLUSH_TX	1110 0001
FLUSH_RX	1110 0010
REUSE_TX_PL	1110 0011
NOP	0101 0000

NRF24L01-lähettimen SPI-väylän käskykanta muodostuu kahdeksasta käskystä. Taulukossa 1 on esitetty nRF24L01-lähettimen käytettävät toimintakäskyt ja näiden bittiarvot.

1. R_REGISTER-käskyllä voidaan lukea kaikki 24L01:n rekisterit. Se on bittimuotoa "000AAAAA" missä "AAAAA"-kohdan tilalle kirjoitetaan rekisterin osoite jonka halutaan lukea. Käytössä olevat osoitteet ovat arvoltaan välillä 0x00 ja 0x17. [17, s.46.]
2. W_REGISTER-käskyllä voidaan kirjoittaa kaikkiin 24L01:n rekistereihin. Tämä käsky vastaan rakenteeltaan käskyä R_REGISTER. Käsky esiintyy binäärimuodossa "001AAAAA" ja tässäkin "AAAAA"-kohdan tilalle kirjoitetaan rekisterin osoite, jonne halutaan kirjoittaa. [17, s.46.]
3. R_RX_PAYLOAD-käskyllä voidaan lukea vastaanotetun payloadin sisältö jos kyseinen tietopaketti on vastaanotettu. Tässä tilassa tapahtuu RX_DR-keskeytys ja CE-nasta on käytössä käskyn koko toiminnan ajan. Kun vastaanotetaan lähetetty datapaketti, CE-nasta on ylhäällä koko toiminnan ajan. Paketin vastaanotettua, CE-nasta kytketään alas ja tällöin vastaanottotila poistetaan käytöstä. Seuraavaksi nollataan RX_DR-keskeytys ja CE-nasta asetetaan uudelleen ylös. Tällöin nasta valvoo ilmassa kuljetetut datapaketit ja aiemmin vastaanotettu datapaketti päättyy käsiteltäväksi. [17, s.46.]
4. W_TX_PAYLOAD-käskyä käytetään yleisesti silloin, kun laite on TX- eli lähetystilassa. Tässä lähetetään ensin "0xA0"-arvoinen tavu ja sen perässä payload-paketti. On tärkeää huomioida, että paketin pituuden täytyy vastata sekä lähettimessä että vastaanottimessa. Lähetäkseen payload-paketti CE-nastan täytyy pysyä ylhäällä vähintään 10 mikrosekunnin ajan. Tiedonsiirron päätettyä TX_DS keskeytys nollataan. [17, s.46.]
5. FLUSH_TX-käskyllä tyhjennetään TX FIFO. [17, s.46.]
6. FLUSH_RX-käskyllä tyhjennetään RX FIFO. [17, s.46.]
7. REUSE_TX_PL-käskyllä voidaan jatkuvasti lähettää tietopaketti niin kauan, kuin CE-nasta on HIGH-tilassa. [17, s.46.]
8. NOP-käsky käytetään lukeakseen STATUS-rekisteriä. [17, s.47.]

NRF24L01-lähettimen toimintaa ohjataan edellä luetelluilla käskyillä, joita lähetetään SPI-väylän kautta. Käskyn lähettäminen tapahtuu seuraavassa järjestyksessä:

1. Ensin asetetaan CSN-nasta alas.
2. Tämän jälkeen lähetetään käskytavu argumentoineen. Jos argumenttia ei ole, lähetetään puolestaan argumenttitavu.
3. Jos käsky palauttaa datan, lähetetään NOP-käskyn tavu (0xFF) ja luetaan saatu vastaus. Argumenttia siirrettäessä lähetetään ensin vähiten merkitsevä tavu, ja tavuja lähettäessä lähetettäessä ensin eniten merkitsevä bitti.
4. Kun edellä kuvatut toiminnot on suoritettu, asetetaan CSN-nasta ylös.

Sisäiset rekisterit

Taulukko 2. nRF24L01-lähettimen käytettävät rekisterit ja muistiosoitteet. (17, s. 53; 17, s.58)

Muistisääntö	Osoite (hex)
CONFIG	00
EN_AA	01
EN_RXADDR	02
SETUP_AW	03
SETUP_RETR	04
RF_CH	05
RF_SETUP	06
RX_ADDR_Px, (x=[0..5])	0A – 0F
TX_ADDR	10
FIFO_STATUS	17

NRF24L01-lähetin voidaan konfiguroida ja valvoa sen toimintaa käyttämällä taulukossa 2 esitettyä rekisterikarttaa SPI-väylän kautta luku- ja kirjoituskäskyillä.

CONFIG-rekisteri on tärkein kaikista 24L01-rekistereistä. PWR_UP-bittiiä asetettaessa itse lähetin aloittaa toiminnan. PRIM_RX-bitti asettaa laitteen vastaanotin-tilaan ja bitin nollaamalla sen voi asettaa puolestaan lähetin-tilaan. [17, s.53.]

EN_AA-rekisterin avulla voidaan ottaa tai poistaa käytöstä automaattinen kuittaus. Tämän rekisterin ollessa käytössä tietopaketin lähettämisen jälkeen lähetin muuttuu automaattisesti vastaanottotilaan. Kun tietopaketti on vastaanotettu, alkuperäinen vastaanotin lähettää kuittauksen. Kun vastaanottimeksi muutettu lähetin vastaanottaa kyseisen kuittauksen, sen toiminta muuttuu takaisin lähettimeksi, ja toiminta jatkuu normaalilla tavalla. Jos kuittauksen vastaanottaminen on epäonnistunut ennalta määrätyn aikavälin aikana, odottava vastaanotin muuttuu takaisin lähettimeksi ja lähettää samanlaisen tietopaketin uudelleen. Tämän jälkeen se muuttuu jälleen vastaanottimeksi odottamaan kuittausta. [17, s.53.]

EN_RXADDR-rekisterissä otetaan käyttöön tarvittavat dataväylät tietopakettien siirtoa varten [17, s.53].

SETUP_AW-rekisterissä määritellään osoitteen leveyttä [17, s.54].

SETUP_RETR-rekisteri on riippuvainen EN_AA-rekisterin käytöstä, sillä sitä muokkaamalla voidaan määrittää, kuinka monta uudelleenyritystä tapahtuu, jos kuittaus ei onnistu. Samalla voidaan määrittää aikaviive edellä mainitulle tapahtumalle.[17, s.54.]

RF_CH-rekisterissä määritellään RF-kanavan asetuksia. Tässä voidaan valita kanava, joka on arvoltaan välillä 0 – 127 ja näiden kanavien väli on tyypillisesti 1 MHz:n luokkaa [17, s.54].

RF_SETUP-rekisteri sisältää kaikki tarvittavat asetukset RF-osa-alueen määrittelyyn. Rekisterin avulla voidaan valita muun muassa siirtonopeus ja tehotaso. [17, s.54.]

STATUS-rekisteri on nimenmukaisesti tilarekisteri. Kun IRQ-nasta asetetaan LOW-tilaan, kyseisestä rekisteristä nähdään, mitä on tapahtunut. Rekisteristä voidaan poimia tietoa kaikista HIGH-tilassa olevista keskeytyksistä ja tietoväylistä. [17, s.55.]

RX_ADDR_P0-RX_ADDR_P5-rekisterit määrittävät vastaanotto-osoitteet jokaiselle tietoväylälle [17, s.55].

TX_ADDR-rekisteri on samantapainen kuin RX_ADDR-rekisteri. Tässä määritellään tietoväylän leveyttä [17, s.56].

FIFO_STATUS-rekisterissä tarkistetaan, ovatko TX ja RX FIFO:n tyhjiä tai ei. Rekisterin käyttö on hyödyllistä, kun lähetetään dataa suurella nopeudella ja halutaan käyttää FIFOa maksimaalisella tasolla. [17, s.57.]

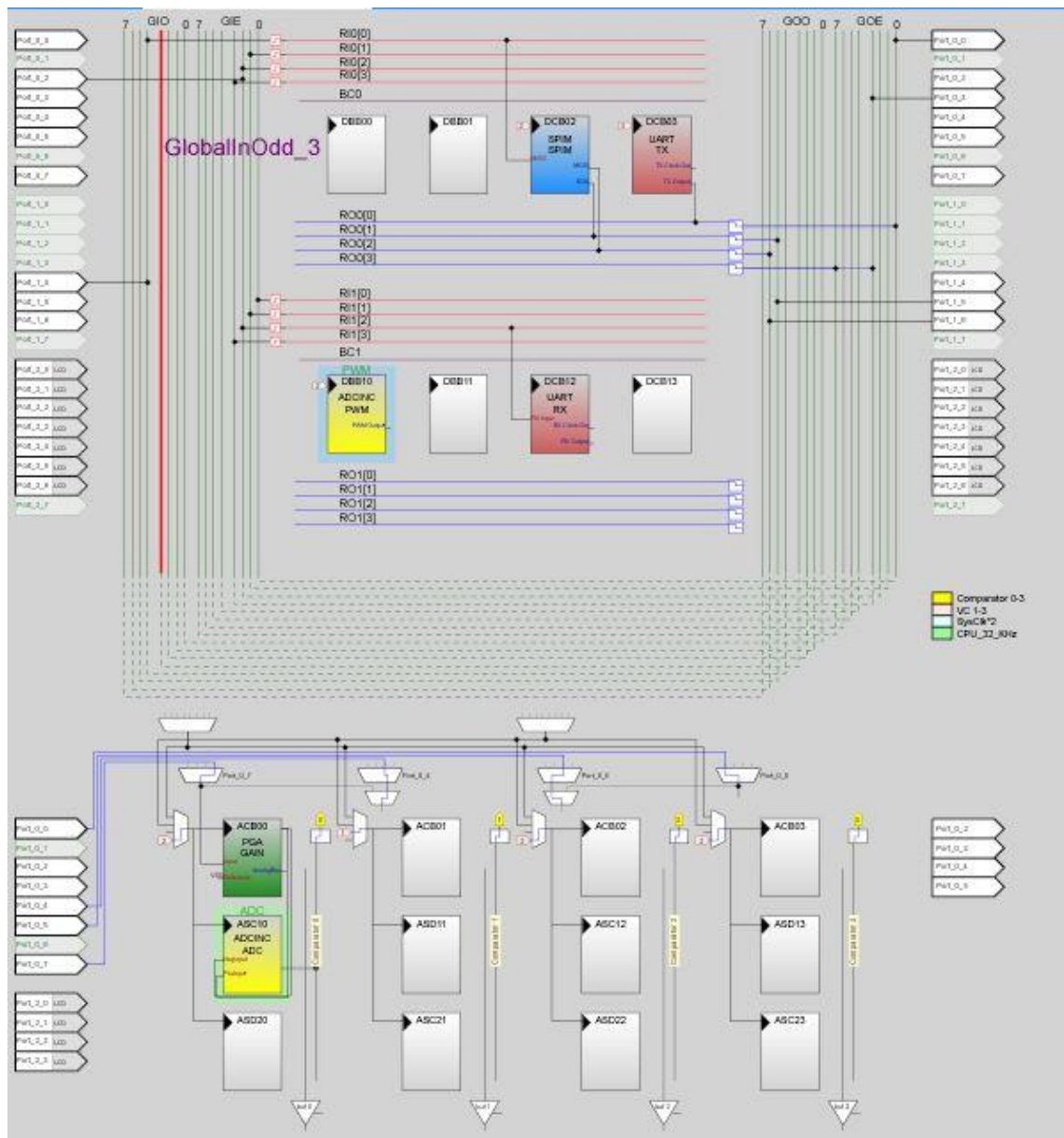
4 Pääohjelman toteutus

4.1 Lähetin

Lähettimen toiminta on esitetty vuokaaviossa (ks. liite 3). Kaaviossa on esitetty toiminnan kannalta tärkeät vaiheet ja toiminnot. Lähetin on rakennettu PSOC-alustalle ja ohjelmointiin on käytetty PSOC Programmer -ohjelmistoa. Prosessorina on CY8C27443-24PXI-piiri. Kyseinen piiri on valittu sen ominaisuuksien ja myös sen helpon saatavuuden takia.

4.1.1 Käytetyt moduulit

Ohjelman toiminnan kannalta on käytössä seuraavat moduulit: SPIM, UART, PGA, ADCINC, SLEEPTIMER ja LCD. Kuvassa 12 on esitetty edellä mainittujen moduulien käyttöönotto PSoC Designer -kehitysohjelmassa.



Kuva 12. Lähettimenä toimivan mikrokontrollerin moduulit ja näiden kytkentätapa.

SPIM-moduuli vastaa SPI-väylän toiminnasta, jolla kommunikoidaan mikropiiriin ja nRF24L01-lähettimen välillä. Kyseisen väylän kautta kulkevat kaikki lähettimelle tarkoitetut asetukset sekä lähetettävät että vastaanotettavat datapaketit. Datalehden mukaan SPIM-moduulin kellotaajuus täytyy olla asetettuna kaksinkertaiseksi halutusta bittinopeudesta. Järjestelmässä käytetty bittinopeus on 1 Mbps ja näin SPIM-moduulin laskeutu kellotaajuus 2 MHz.

UART-moduuli vastaa RS232-tietoliikenneportin toiminnasta, ja se mahdollistaa sarjamuotoisen tietoliikenteen mikrokontrollerin ja mittauslaitteen kanssa. UART käyttää

kahta PSoC:in lohkoa nimeltään TX ja RX, jotka toimivat itsenäisesti. Moduulin käyttöönoton jälkeen määritellään toimiva kellotaajuus sekä sisääntulo- ja ulostulo-portit. Kellotaajuuden on oltava kahdeksan kertaa halutusta bittinopeudesta, ja taajuuden toleranssi on $\pm 2\%$. Sarjaportin nopeus vaihtelee välillä 75 bit/s ja 115200 bit/s. PuTTY:in tyypillinen käyttämä nopeus sarjaportille on 9600 bit/s. Kaavioissa 1 ja 2 on esitetty tarvittavan kellotaajuuden laskeminen. Kaavassa 3 on esitetty edellä laskettujen taajuuksien välinen toleranssi. Laskun mukaan toleranssi on määrätyn vaatimuksen rajoissa. Moduulin muut asetukset ovat ennalta määrättyt eikä arvoja muuteta.

$$f = 9600bps * 8 = 76800Hz \quad (1)$$

$$f = 24MHz \div 12 \div 2 \div 13 \approx 76923 Hz \quad (2)$$

$$D = \frac{76923Hz - 76800Hz}{76800Hz} * 100 \approx 0,16\% \quad (3)$$

ADCINC- ja PGA-moduulit vastaavat analogisesta sisääntulosta. Tätä yhdistelmää käytetään muun muassa silloin, kun halutaan lukea LM35-anturin mitatut lämpötilan arvot. ADCINC on analogia-digitaalimuunnin, joka kerää useat analogiset näytteet ja tuottaa 8-bittiset digitaaliset arvot. Kellotaajuuden täytyy olla arvoltaan 125 kHz:n ja 8 MHz:n välillä. Näytteenottotaajuus lasketaan kaavan 4 mukaan, ja sen sallittu vaihteluväli on 125 sps:n ja 31,25 ksp/s:n välillä. ADCINC-moduulin sisääntulot on reititetty PGA-moduulilta.

$$f_s = \frac{24 MHz \div 12 \div 2}{256 \times (2^8 - 6 + 1)} = 781,25 sps \quad (4)$$

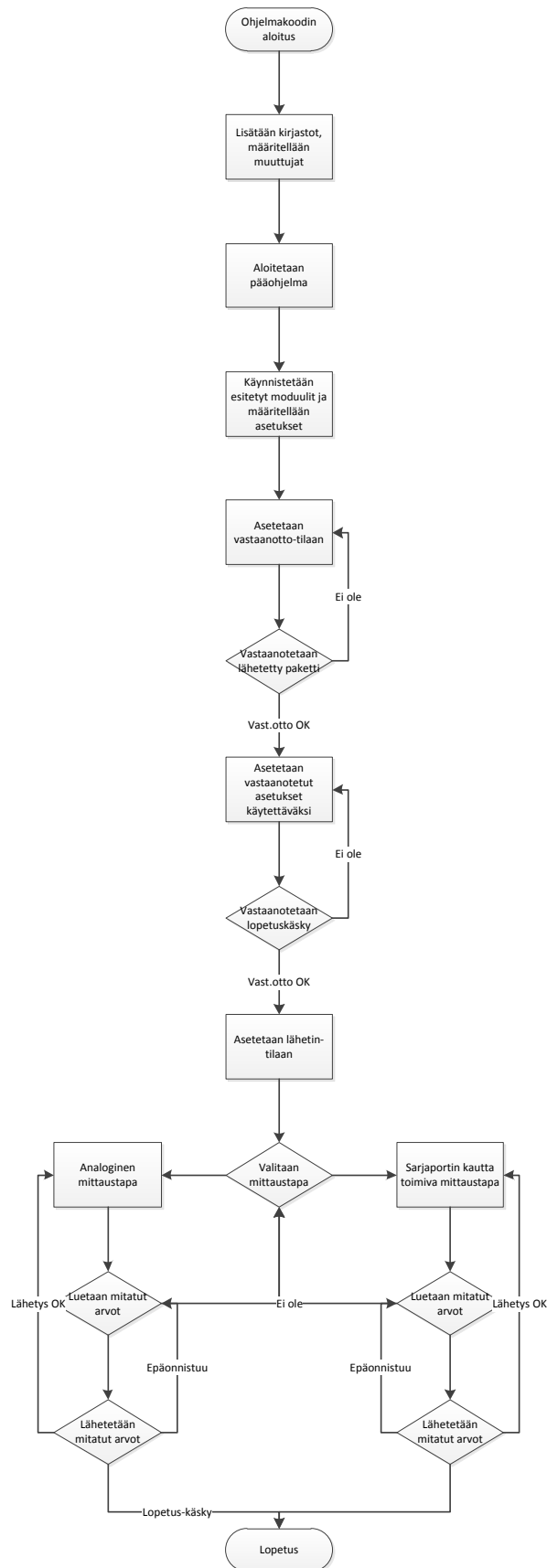
PGA-moduuli on lisätty antamaan signaalille korkean tuloimpedanssin ja se tuo enemmän joustavuutta signaalin reititykselle. PGA-moduulin vahvistusarvon määrittäminen sijoittuu 0,062:n ja 48:n välille. Tässä järjestelmässä sisääntulojännitteenä käytetään tasajännitettä arvoltaan välillä 0 V – 5 V, ja näin tulevan jännitteen vahvistus on välttämätön. Piirin ominaisuutena on, että maksimaalisin vahvistettu jännitearvo ei voi ylittää esimääritettyä 5 V:a.

LCD-moduuli nimenmukaisesti sallii LCD-näytön toimintaa ja näytölle tulostetaan ohjelman toiminnasta kertovat viestit ja tilapäivitykset.

SLEEPTIMER-moduuli tarjoaa perustason ajanottotoimintoja ilman arvokkaiden digitaalisten lohkojen käyttöä. Kyseisellä moduulilla saadaan ohjelman lähettämät mitatut arvot vastaanottimelle 1 Hz:n taajuudella. Moduulin keskeytysnopeuden arvoksi voi valita 1 Hz, 8 Hz, 64 Hz ja 512 Hz. Tästä sarjasta valitaan ensimmäinen arvo.

4.1.2 Pääohjelma

Pääkoodi on rakennettu mahdollisimman yksinkertaiseksi, selkeäksi ja toimivaksi. C-koodi on esitetty liitteessä 1. Kuvassa 13 on esitetty koodin toimintaa kuvaava vuokaavio.



Kuva 13. Lähettimessä käytetyn ohjelman vuokaavio.

Ohjelmaan on poimittu erilaisia toimintoja lähettimen esimerkkikoodista. Ohjelman toiminnan edistyminen määräytyy "while" -toimintojen kautta, ja kaikenlaiset valinnat ovat toimivuuden kannalta toteutettu "switch-case"-rakenteella.

Pääohjelman rinnalle on lisätty nRF24L01.c sekä nRF24L01.h-kirjastot. Kyseiset kirjastot kehitti Metropolia Ammattikorkeakoulun lehtori Kimmo Sauren ja hän on sallinut niiden käytön tässä projektissa. NRF24L01.h-kirjasto sisältää 24L01-lähettimen toiminnan kannalta tarvittavat SPI-käskyt, sekä tiedostossa on myös määritelty kommunikointifunktiot, rekisteriosoitteet, IRQ-keskeytysten ja STATUS-rekisterin maskit. Edellä mainitut funktioita ja määrittelyjä on nRF24L01.c-koodissa.

nRF24L01.c-koodi sisältää erilaisia funktioita lähettimen asetuksien määrittämiseen. Näitä funktioita käytetään pääohjelmassa. Alkuperäistä esimerkkikoodia on muokattu työtä varten funktioiden eri kohdissa. RF_SETUP-rekisterissa on määritetty SPI-väylänopeudeksi 1 Mbps. RX_ADDR_P0 ja TX_ADDR-rekisterit on määritetty vastamaan toisiaan määrittämälle niille samanlaiset lähetys- ja vastaanotto-osoitteet.

Pääohjelma rakentuu kolmesta eri osasta. Aluksi koodiin sisällytetään erilaiset ohjelman toiminnan kannalta tärkeät header-tiedostot, muun muassa edellä mainittu nRF24L01.h-tiedosto. Samalla esitellään globaalit muuttujat, joiden arvot saattavat muuttua ohjelman edetessä. Tässä kohdassa määritellään myös rs232-yhteyden toimintaan tarvittavat muuttujat.

Main-osiossa käynnistetään aiemmin esitetyt analogiset ja digitaaliset moduulit, otetaan käyttöön globaalit keskeytykset ja määritellään datapaketin koon ja sen sisältämät arvot. Payload on vähintään kahden alkion taulukko, jonka ensimmäinen alkio määrittää lähettimen ID-tunnuksen, ja toiseen alkioon voidaan kirjoittaa siirrettävä data.

Tämä jälkeen esitellään päättymätön silmukka, jossa tapahtuu lähettimen datan käsittely. Vuokaavion mukaan aluksi asetetaan lähetin vastaanottotilaan vastaanottamaan päälaitteelta tulevia asetuksia. Vastaanotto-funktio on toteutettu käyttäen "while"-silmukkaa, jossa tutkitaan ilmateitse kulkevaa payload-dataa. Silmukassa vertaillaan laitteen ID-tunnusta payloadin sisältämään tunnuksen. Kun tunnukset vastaavat toisiinsa, ohjelma poistuu silmukasta ja siirtyy pääohjelmassa eteenpäin. Tämän jälkeen LCD-näytölle tulostetaan tilapäivitys saapuneista asetuksista ja odotetaan lopetuskäskyä. Tätä varten on rakennettu vastaavanlainen "while"-silmukka, jossa käsitellään

payloadin ensimmäisen alkion arvoa. Kun lähettimen ja payloadin arvot vastaavat toisiinsa, ohjelma poistuu "while"-silmukasta. Tämän jälkeen lähetin asetetaan lähetystilaan.

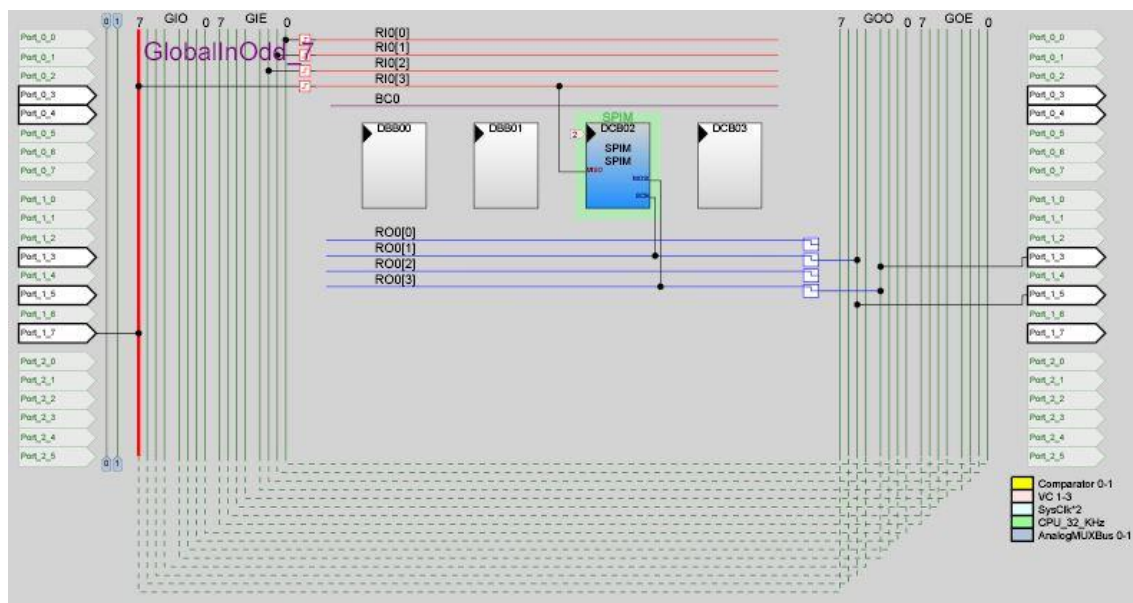
Edellä vastaanotetut asetukset määrittävät mikrokontrollerin mittaustapaa. Projektin toiminnan kannalta on esitetty kaksi mittaustapaa. Yksi näistä on analoginen tapa, esimerkiksi LM35-anturia varten. Toinen mittaustapa käyttää COM-tiedonsiirtoväylää mittaustiedon lukemiseen. Valinta tapahtuu "switch-case"-rakenteella, jossa vertaillaan payloadin toisen alkion arvoa ohjelman esimäärittämiin arvoihin. Molemmat "case"-tapaukset on rakennettu samankaltaisella tavalla. Kun ohjelma siirtyy tapaukseen, tulostetaan LCD-näyttölle päivitys valitusta mittaustavasta ja käynnistetään mittaustavoil- le ominaiset moduulit. Mitattu data tallennetaan uuteen payloadiin laitteen ID-tunnuksen rinnalle ja koko paketti lähetetään vastaanottimelle. Payloadin lähettä- seen käytetään "while"-silmukkaa. Kun payload on onnistuneesti lähetetty, siirtyy ohjelma pois silmukasta ja aloittaa uudelleen "case"-tapauksen toiminnan. Tämä jatkuu siihen asti, kunnes mikrokontrollerista kytketään virtalähde pois tai se käynnistetään uudelleen.

4.2 Vastaanotin

Vastaanottimen toiminta on esitetty vuokaaviossa (ks. liite 4). Vastanotin on rakennettu PsoC-alustalle, mutta lähetimestä poiketen tässä käytetään erilaista piiriä. Piiriksi on valittu CY7C64215-28PVXC sen projektin toimintaan vaadittujen ominaisuuksien pe- rusteella.

4.2.1 Käytetyt moduulit

Vastaanottimessa käytetään seuraavia moduuleja: SPIM, SLEEPTIMER ja USBUART. SPIM- ja SLEEPTIMER-moduulit ovat asetukseltaan vastaavanlaiset lähettimenä toi- mivassa mikrokontrollerissa. Kuvasta 14 näkyy, että ainoa asetettava moduuli on SPIM.

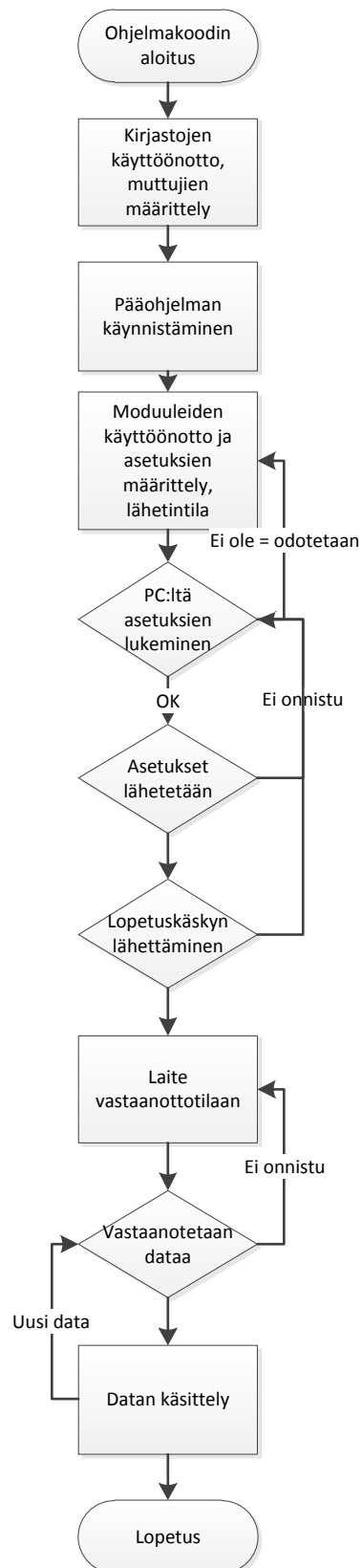


Kuva 14. Kuvassa on esitetty SPIM-moduulin kytkentäkaava.

USBUART-moduuli jäljittää mikrokontrollerin USB-liitäntää COM-portiksi eli sarjaportiksi. Yksinkertainen tapa kytkeä mikrokontrolleri USB-liitäntään on yhteyden jäljittäminen RS232-muotoon yli USB-väylän. Suuri etu tässä menetelmässä on, että PC-sovellukset käyttävät USB-yhteyttä RS232:n COM-yhteytenä, mikä mahdollistaa sen helpon virhekorjauksen. Kun uusi laite muodostaa ensimmäisen kerran yhteyden Windows PC -tietokoneeseen, Windows pyytää käyttäjää asentamaan ajurit. Tarvittavat ajurit luodaan ohjelman kääntämisen yhteydessä, ja ne sijaitsevat projektin LIB-kansiossa. Kun ajurit on asennettu, mikrokontrolleri luetaan COM-portina ja sen toimintaa voidaan ohjata PuTTY-pääätteeltä.

4.2.2 Pääohjelma

Vastaanottimen käyttämä koodi on rakenteeltaan vastaavanlainen lähettimessä käytetävän koodin kanssa. Koodi on esitetty liitteessä 2. Main.c-tiedoston rinnalla käytetään nRF24L01.c- ja nRF24L01.h-kirjastoja. Nämä tiedostot ja niiden rakenne on esitelty aiemmin. Kuvassa 15 on esitetty vastaanottimen pääohjelman rakenne vuokaaviona.



Kuva 15. Vastaanottimen pääohjelman vuokaavio.

Pääkoodi sisältää ohjelman koodin ja sen alussa määritellään tarvittavat kirjastot sekä globaaliset muuttujat jotka käytetään ohjelman eri funktioissa. Main-funktion alussa otetaan käyttöön aiemmin määritetyt moduulit, käynnistetään globaaliset keskeytykset ja otetaan käyttöön uniajastin.

Käyttäjä kommunikoi mikrokontrollerin kanssa tietokoneelle asennetun PuTTY-päänteen avulla. Yhteys tapahtuu käyttäen USB-väylää, ja sitä varten on käytössä USB-UART-moduuli. Kommunikoinnin toimintoon on laadittu oma funktio. Funktio koostuu "while"-silmukasta ja "switch-case"-rakenteesta. "While"-silmukassa luetaan USB:n kautta saadut arvot ja vertaillaan niitä esimääritettyyn lopetusarvoon. Jos tietokoneelta saatu arvo ei vastaa lopetusarvoa, ohjelma siirtyy käyttämään "switch-case"-rakennetta. Jälkimmäisessä rakenteessa tallennetaan payloadiin valitun laitteen ID-tunnus ja sille laitteelle lähetettävät asetukset. Kaikkien asetuksien määrittelyjen jälkeen ohjelma poistuu "while"-silmukasta ja asettaa vastaanottimen vastaanotin-tilaan. Tämä toiminto on rakennettu myös "switch-case"-rakenteella. Tässä tilassa vastaanotetaan kaikki mahdolliset payloadit ja tarkistetaan niiden sisältö yksi kerrallaan. Payloadin kahdesta alkiossa ensimmäinen sisältää lähettimen laitekohtaisen ID-tunnuksen, ja toinen alkio lähetetyn datan. "Switch-case"-rakenne tarkistaa ID-tunnukset ja lähettää tietokoneelle tämän tunnuksen ja sen mukana tulleen data-arvon. Arvot tulostuvat PuTTY-päänteelle. Ohjelman toiminta jatkuu niin kauan, kunnes mikrokontrollerilta katkaistaan virta.

5 Laitteen käyttö

5.1 Esivalmistelu

Järjestelmän käyttämiseen käyttäjä tarvitsee seuraavanlaiset oheislaitteet ja tarvikkeet:

1. Vähintään 2 kpl nR24L01-lähetintä
2. Mikrokontrolleri, jossa on CY8C27443-24PXI-mallinen piiri. Tämä mikrokontrolleri toimii jatkossa lähettimenä.
3. Mikrokontrolleri, jossa on CY7C64215-28PVXC-mallinen piiri. Tämä mikrokontrolleri toimii jatkossa vastaanottimena.

4. USB-kaapeli
5. Tietokone tai vastaavanlainen päätelaite
6. 5V virtalähde (vähintään 3 kpl sauvaparistoa)
7. LCD-näyttö.

Kun käytössä on kaikki edellä mainitut komponentit ja laitteet, käyttäjä voi aloittaa järjestelmän käytön. Saatavilla olevat nRF24L01-lähettimet kytketään niille määrätyille paikoille saatavilla oleviin mikrokontrollereihin. Vastaanottimena toimiva kontrolleri kytketään USB-kaapelilla käyttäjän tietokoneeseen. Lähettimenä toimivaan mikrokontrolleriin kytketään valittu mittauslaite sille määrättyyn porttiin. Käyttäjällä on mahdollisuus valita yksi kahdesta saatavilla olevasta portista: analoginen portti tai I2C-portti. Kun mittauslaite on kytketty mikrokontrolleriin, voidaan kyseinen laite kytkeä toimintaan. Tämä mikrokontrolleri muuttuu ohjelman mukaisesti aluksi vastaanottimeksi odottamaan tietokoneeseen kytketystä mikrokontrollerista tulevia käskyjä.

Tietokoneeseen kytketty mikrokontrolleri toimii aluksi lähettimenä, ja se nimenmukaisesti lähettää käyttäjän valitut asetukset. Kontrolleri saa virtansa USB-portista ja sen toiminta ohjautuu PuTTY-päätteen kautta. Käyttäjä käynnistää tietokoneelta PuTTY-päätteen. Yhteyden muodostamiseksi ohjelmasta valitaan yhteystyyppinä "serial". Tämän jälkeen painetaan "open"-valinta. Jos mikrokontrolleri on yhdistetty oikein USB-porttiin, yhteys laitteeseen toimii ja käyttäjälle avautuu terminaali-ikkuna.

5.2 Terminaalin käyttö

Mikrokontrollerin toiminta alkaa käyttäjän painaessa näppäimistöllä "s"-näppäintä. Tällöin ruudulle tulostuu teksti "Valitse laite:". Tähän käskyyn voidaan vastata vain numeronäppäimellä, joka on arvoltaan mikrokontrollerin muistiin ohjelmoitujen laitteiden määrä. Kun laitteen ID-tunnus on syötetty, ohjelma tulostaa

Valitse mittaus: a:Analog i:I2C

Tässä käskyssä käyvät ainoastaan näppäimet "a" ja "i". "a"-näppäimellä valitaan mittaustavaksi analoginen sisääntuloportti ja "i"-näppäimellä rs232-sisääntuloportti. Kun valinta on tehty, ohjelma tulostaa seuraavaksi tekstin

Valittu laite: X Valittu mittaus: Y Paina L lähettääksesi

Tässä X-arvo vastaa valitun laitteen ID-tunnusta ja Y-arvo valitun laitteen mittaustapaa. Painamalla "l"-näppäintä kyseiset asetukset lähetetään ilmateitse mittausmikrokontrollereille. Kun tämä on, tehty terminaaliiin tulostetaan

Asetukset on lähetetty... S: jatkaa / Q: vastaanotto

Tässä käyttäjän painaessa "s"-näppäintä päästään ohjelman alkuun ja näin voidaan lähettää asetuksia useammille mittausmikrokontrollereille. "Q"-näppäintä painamalla lähetetään kaikille mittausmikrokontrollereille lopetuskäsky, jolloin mikrokontrollerit muuttuvat lähettimiksi ja koneeseen kytketty mikrokontrolleri muuttuu vastaanottimeksi. Tämän jälkeen PuTTY-terminaaliin tulostetaan rivi kerrallaan vastaanotettujen laitteiden ID-tunnukset ja niiden mukana tulleet data-arvot. Arvot tulostetaan hex-muodossa.

Lopettaakseen järjestelmän toimintaa, käyttäjä katkaisee laitteiden virranlähdet poistamalla käytöstä paristot tai irrottamalla mikrokontrolleri USB-portista.

6 Yhteenveto

6.1 Projektin onnistumisen analysointi

Projekti onnistui kokonaisuudessaan hyvin. Alkuperäisestä ideasta saatiin toteutettua monet vaatimukset ja toiveet. Järjestelmässä sovellettiin sille määritellyjä ominaisuuksia ja toimintoja. Järjestelmän ollessa käytössä mikrokontrollereiden kautta voidaan siirtää mittausantureilta saadut arvot tietokoneelle jälkikäsiteltäväksi. Tietokoneelta voidaan myös lähettää langattomasti asetuksia. Järjestelmän käyttöönotto on osittain monimutkaista, koska järjestelmän fyysinen ulkomuoto ei vastaa alun perin suunniteltua helppokäyttöistä kokonaisuutta.

Projektin eteneminen oli suurimmalta osin tasaista, lomia lukuun ottamatta. Projektiin tutustuminen alkoi syyskuussa 2012 ja lokakuun lopussa saatiin ensimmäiset testitulokset nRF24L01-lähettimien toiminnasta. Työt keskeytyivät joululoman ajaksi. Kiireisten asioiden takia projektin tekeminen jatkui vasta helmikuussa ja jatkui aina maaliskuun loppuun asti. Huhtikuussa aloitettiin toinen intensiiviprojekti ja kevään lähestyessä projekti päätettiin lopettaa koulusta valmistumisen takia.

Järjestelmän toimimiseen vaadittavat komponentit olivat helposti saatavissa. Tämä johtui osittain siitä, että projektissa käytetyt mikrokontrollerit ja niihin upotetut piirit ovat vastaavanlaisina käytössä koulun järjestämällä kursseilla. NRF24L01-lähettimiä oli aluksi saatu neljä, joista kolme kappaletta vahingoittui toimintakyvyttömäksi. Uusia lähettimiä oli tilattu iso erä eBay-verkkokaupasta. Kirjastosta oli mahdollisuus lainata pitkälle ajanjaksolle ”PSoC programmer”-ohjelmointilaite, jolla mikrokontrolleri ohjelmointiin. Koulun tietotekniikan opiskelijoiden käytössä olevissa laboratoriotiloissa oli käytössä useampi tietokone, joihin oli asennettu kehitysohjelmisto. Samoissa tiloissa oli lisäksi tarpeelliset työkalut, kuten juotinlaitteet, mittausslaitteet ja erilliset jännitelaitteet.

Projektin kehittämisen aikana olen kehittänyt C-ohjelmointikielen osaamista ja oppinut SPI-väylän käyttöönottoa RF-lähettimen käyttämiseen.

6.2 Ongelmat

Projektin suurin ongelma on koko järjestelmän toiminnan laatu. Yhteyden muodostaminen lähettimien välillä on sitä epävakaampaa, mitä enemmän lähettimiä on käytössä. Testien perusteella pelkästään vastaanottimen ja yhden lähettimen välinen toiminta onnistuu lähes jokaisen käytön aikana. Yhteys toimii molempiin suuntiin: asetukset saapuvat lähettimeen ja se puolestaan lähettää mitatut data-arvot vastaanottimelle. Kun yhden lähettimen rinnalle kytketään toinen lähetin, verkon toiminnan laatu heikkenee huomattavasti. Käyttäjä joutuu lähettämään asetuksia lähettimeen useamman kerran, ja verkossa tapahtuu paljon datahäviötä. Kyseinen tilanne pahentui sitä mukaan, mitä enemmän lähettimiä on käytössä samanaikaisesti. Järjestelmässä on käytetty datansiirtoon yhtä siirtokaistaa, mikä mahdollisesti aiheutti kommunikointiongelmat, sillä kaistalla liikkuu samanaikaisesti suuri määrä dataa. Ongelmaa on yritetty ratkaista ohjelmoimalla lähettimiä käyttämään omia kaistojaan. Ratkaisu ei kuitenkaan onnistunut, sillä vastaanotin lakkasi toimimasta kanavia vaihdettaessa. Toiminnan epäonnis-

tuminen johtui mahdollisesti vääränlaisesta ajastuksesta. Ongelmaan ei ole kirjoitushetkellä löydetty ratkaisua. Yhden kaistan ongelman ratkaiseminen voi onnistua ajastamalla oikeaoppisesti datansiirtoa ja tarkistamalla kaistan kuormitus ennen datasiirtoa.

Lopullinen tuote ei vastannut täysin tilaajaan vaatimaa loppumuotoa. Alkuperäisen suunnitelman mukaan piirille piti suunnitella ja toteuttaa oma kompakti piirilevy, jossa olisi toiminnan kannalta tarvittavat portit, erillinen virtalähde ja suojakotelo. Kirjoitushetkellä järjestelmä oli toteutettu koulusta saatavilla olevalle piirilevylle. NRF24L01-lähetin käytti toimiakseen sille erikseen suunniteltua adapteria, joka voitiin turvallisesti ja helposti kytkeä käytettävälle piirilevylle. Virtalähteenä toimii erillinen jännitegeneraattori, tai tarvittaessa voidaan käyttää "PsoC"-ohjelmointilaitetta.

Aikataulujen suhteen projektin tuli olla valmis helmikuussa 2013. Näin ei kuitenkaan tapahtunut, ja projekti jäi keskeneräiseksi. Tämä johtuu osittain siitä, että projektille tehty alkuperäinen aikasuunnitelma kehityksen eri vaiheille oli suunniteltu liian tiukaksi ja toteutukseen kului odotettua enemmän aikaa. Suurimman osan ajasta vei tutustuminen alkuperäisiin lähettäjiin. Alkuperäisen suunnitelman mukaan järjestelmässä käytettäisiin nRF24L0E1-lähettäjiä, ja näiden ohjelmointiin käytettäisiin sille suunniteltu ohjelmointilaite. Ohjelmointilaite on vieras Metropolia ammattikorkeakoulun opetuksessa, ja sen opetteluun kuluisi useampi viikko. Kuukauden tutustumisjakson jälkeen otettiin käyttöön useasti koulun opetuksessa käytetty "Cypress PSoC"-kehitysalusta ja lähettämeksi nRF24L01.

Projektia tehtiin suurimmalta osin koulun rakennuksen sisällä, sieltä saatavien työkalujen takia. Työtila valittiin sen perusteella, miten kyseisessä tilassa oli ollut opetusta. Tämän takia kaikki tarvikkeet ja komponentit on jouduttu kuljettamaan mukana, ja järjestelmän pystyttäminen vei aikaa.

6.3 Tulevaisuuden näkymät

Projektin tärkein kehityskohde on radioverkon lisätestaus todellisessa sovelluksessa. Järjestelmä toimii epävakaaasti tällä hetkellä tuntemattomista syistä, ja seuraavana tavoitteena on parantaa verkon toimivuutta. Tämä vaatii käytössä olevien asetuksien uudelleen läpikäyntiä ja lisäsuunnittelua sekä lisätestien tekemistä erilaisissa olosuhteissa.

Puuttuviin ominaisuuksiin luettelisin dynaamisen payloadin käytön. Payloadin pituuden voisi määrittää käyttämällä lähetysfunktiossa pituusargumenttia, ja sen toimesta kaikenkokoiset rakenteet voitaisiin tehokkaasti lähettää kaikkiin osoitteisiin. Myös ACK payloadin käyttö tehostaisi kaiken siirrettävän datan perille saapumisen tarkistamista. Tällä tavalla voidaan seurata mahdollista tietoliikenteen häviötä ja parantaa järjestelmän toimivuutta.

Vastaanottimelle suunniteltua käyttöjärjestelmää on kehitettävä tai parannettava. Parannuksiin luettelen järjestelmän monipuolisemman asetusten laatimisen, vastaanottotilan jälkeiset toimenpiteet, kuten lähettimien uudelleenmäärittely sekä mahdollisten virheiden seurannan ja etäisen uudelleenkäynnistämisen ongelmatilanteissa.

Parhaillaan lähettimet kommunikoivat ainoastaan vastaanottimen kanssa. Lähettimet voidaan ohjelmoida toimimaan monipuolisesti myös keskenään. Näin saadaan pitkäkantaisia verkkoja, joissa lähettimet voivat toimia siltoina datansiirrossa, sillä käytössä olevat RF-lähettimet voivat toimia asetuksista riippuen vuorotellen lähettimenä tai vastaanottimena.

Lähteet

- 1 Van Ess David. 2010. Using the ADCINC Analog to Digital Converter. 2007 - 2010 Verkkodokumentti. Cypress Semiconductor Corporation. <<http://www.cypress.com/?docID=36701>>. Luettu 30.4.2013.
- 2 SleepTimer datasheet. 2012. Verkkodokumentti. Cypress Semiconductor Corporation. <<http://www.cypress.com/?docID=4025>>. Päivitetty 5.10.2012. Luettu 29.4.2013.
- 3 PSoC Designer Assembly Language User Guide. 2001. Verkkodokumentti. Cypress Microsystems Inc. <<http://www.ele.uva.es/~jesman/BigSeti/ftp/Microcontroladores/PSoC%20Microcontrollers/AssemblerUserGuide.pdf>>. Päivitetty 30.5.2001. Luettu 1.5.2013.
- 4 Introduction to PSoC. 2006–2011. Verkkoartikkeli. PIAP Technical Consulting. <<http://www.psocdeveloper.com/articles/what-is-psoc.html>>. Luettu 30.4.2014.
- 5 RF Transceivers Information. 1999–2013. Verkkoartikkeli. GlobalSpec. Inc. <http://www.globalspec.com/learnmore/communications_rf_microwave/rf_microwave_components/rf_transceivers>. Luettu 2.5.2013.
- 6 What is an RF Transceiver?. 2003–2013. Verkkoartikkeli. Conjecture Corporation. <<http://www.wisegeek.com/what-is-an-rf-transceiver.htm>>. Luettu 2.5.2013.
- 7 RF Transceiver. 2013. Verkkoartikkeli. <<http://www.embelectronip.com/index.php/en/rf-transceiver.html>>. Luettu 2.5.2013.
- 8 2.4GHz Wireless nRF24L01+ Transceiver Module. 2013. Verkkoartikkeli. <http://www.mdfly.com/index.php?main_page=product_info&products_id=81>. Luettu 29.4.2013.
- 9 A/D-muuntimet. 2013. Verkkodokumentti. <<http://users.jyu.fi/~jupeihal/Kontrollerit3.pdf>>. Luettu 3.5.2013.
- 10 Jämsä Lauri. Mikä olisi paras kehitysalusta?. 2011. Verkkoartikkeli. <<http://www.ruuvipenkki.fi/2011/04/10/mika-olisi-paras-kehitysalusta>>. Luotu 10.4.2011. Luettu 2.5.2013.
- 11 SPI Interface Specification. 2005. Verkkodokumentti. VTI Technologies OY. <http://www.murata-mems.fi/sites/default/files/documents/tn15_spi_interface_specification.pdf>. Päivitetty 19.9.2005. Luettu 2.5.2013.

- 12 Prosentti käsitteenä. 2013. Verkkoartikkeli. Yleradio.
<<http://abitreenit.yle.fi/treenaa/lukio/aine/14/80/219/217>>. Luettu 2.5.2013.
- 13 Abdullah Ismail. 2007. Programmable System-on-Chip Technology Semiconductor Digital Systems Architecture Project Report. Verkkodokumentti.
<<http://carg.site.uottawa.ca/doc/PSoCAbdallahIsmail.pdf>>. Päivitetty 3.9.2007.
Luettu 3.5.2013.
- 14 La Malfa Salvatore. 2010. Wireless Sensor Networks. Verkkodokumentti.
<<http://www.dees.unict.it/users/bando/files/wsn.pdf>>. Päivitetty 19.1.2010. Luettu 8.5.2010.
- 15 Puccinelli Daniele, Haenggi Martin. 2005. Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing. Verkkodokumentti.
<<http://faculty.kfupm.edu.sa/COE/mayez/ps-coe499/project/Sensor-network/XSCALE-STARGATE/1-1-challenges-sensor-networks.pdf>>. Päivitetty 2005. Luettu 8.5.2013.
- 16 RF Module (Transmitter & Receiver). 2012. Verkkoartikkeli.
<<http://www.engineersgarage.com/electronic-components/rf-module-transmitter-receiver>>. Päivitetty 2012. Luettu 2.5.2013
- 17 NRF24L01 Product Specification V2.0. 2007. Verkkodokumentti. Nordic Semiconductor.
<http://www.nordicsemi.com/eng/nordic/download_resource/8041/1/58493979>. Päivitetty 7.2007. Luettu 28.4.2013.

Lähettimen pääkoodi

```
//-----  
// Lähetin-mikrokontrolleri  
// Tekijä: Nikita Pavlov  
// 27.4.2013  
//-----  
  
#include <m8c.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include "PSoCAPI.h"  
#include "PSoCGPIoint.h"  
#include "nRF24L01.h"  
  
// Erilaisten muuttujien esittely  
int valinta, mittausValinta, valmius;  
unsigned char status = 0, rxData = 0;  
  
// Pääohjelman esittely  
void main(void)  
{  
    BYTE payloadData[2] = {1, 0}; // payloadin käyttöönotto.  
    // Ensimmäinen alkio sisältää ID-tunnuksen, toinen alkio sisältää lä-  
    hetettävän datan  
    BYTE analogValue; // tähän tallennetaan analogiset mittausarvot  
  
    M8C_EnableGInt; // globaaliset keskeytykset  
  
    UART_DisableInt(); // UART:n keskeytyksien poistaminen käytöstä  
    UART_Start(UART_PARITY_NONE); // UART:n toiminnan salliminen  
  
    PGA_Start(PGA_MEDPOWER); // PGA-moduulin käyttöönotto  
    SPIM_Start(SPIM_SPIM_MODE_0 | SPIM_SPIM_MSB_FIRST); // SPIM-moduulin  
    käyttöönotto  
  
    // SleepTimer-moduulin käyttöönotto  
    SleepTimer_Start();  
    SleepTimer_SetInterval(SleepTimer_1_HZ);  
    SleepTimer_EnableInt();  
    SleepTimer_SyncWait(1, SleepTimer_WAIT_RELOAD);  
  
    // LCD-moduulin käyttöönotto  
    LCD_Start();  
    LCD_Position(0, 0);  
    LCD_PrCString("Trans1");  
  
    // Ohjelman toiminta alkaa tästä  
    for(;;) {  
        // Mikrokontrolleri vastaanotto-tilaan  
        nRF24L01_commonConfiguration();  
        nRF24L01_SetAsReciever();  
  
        // Vastanotetaan payload-paketti ja vertaillaan ID-tunnukset  
        while(valinta != 0x31){
```

```
        if( nRF24L01_is_RX_DR() == 1 ){
            nRF24L01_readPayload( R_RX_PAYLOAD, payloadData, 2);
            nRF24L01_clear_RX_DR();
            valinta = payloadData[0];
            mittausValinta = payloadData[1];
        }
    }

    LCD_Position(0,8);
    LCD_PrCString("Aset. Ok");

    // Lopetuskäskyn odottaminen
    while(valmius != 0x71){
        if( nRF24L01_is_RX_DR() == 1 ){
            nRF24L01_readPayload( R_RX_PAYLOAD, payloadData, 2);
            nRF24L01_clear_RX_DR();
            valmius = payloadData[0];
        }
    }

    // Mikrokontrolleri lähetin-tilaan
    nRF24L01_commonConfiguration();
    nRF24L01_SetAsTransmitter();

    // Mittaustavan valinta
    switch(mittausValinta){
        // Analoginen mittaustapa
        case 'a':
            SleepTimer_SyncWait(1, SleepTimer_WAIT_RELOAD);
            payloadData[0] = 1;
            ADCINC_Start(ADCINC_MEDPOWER);
            ADCINC_GetSamples(1);

            // Analogisen arvon lukeminen
            analogValue = ADCINC_iClearFlagGetData();

            // Mitatun datan tulostaminen
            payloadData[1] = analogValue;
            LCD_Position(1,0);
            LCD_PrCString("ID: " );
            LCD_Position(1,4);
            LCD_PrHexByte(payloadData[0]);
            LCD_Position(1,7);
            LCD_PrHexInt(analogValue);

            // Mitatun datan lähettäminen
            while( nRF24L01_is_TX_DS() != 1 ){
                nRF24L01_sendPayload(W_TX_PAYLOAD, payloadData, 2);
                nRF24L01_transmitPayload();
                if( nRF24L01_is_MAX_RT() ){
                    nRF24L01_clear_MAX_RT();
                    break;
                }
            }
            nRF24L01_clear_TX_DS();
            break;

        // RS232-mittaustapa
        case 'i':
```

```
SleepTimer_SyncWait(8, SleepTimer_WAIT_RELOAD);

// Mitatun datan tulostaminen
LCD_Position(1,0);
LCD_PrCString("ID: " );
LCD_Position(1,4);
LCD_PrHexByte(payloadData[0]);
LCD_Position(1,7);
LCD_PrHexInt(rxData);

if(status & UART_RX_REG_FULL) {
    rxData = UART_bReadRxData();
    payloadData[1] = rxData;
    UART_SendData(rxData);

    // Mitatun datan lähettäminen
    while( nRF24L01_is_TX_DS() != 1 ){
        nRF24L01_sendPayload(W_TX_PAYLOAD, payloadData, 2);
        nRF24L01_transmitPayload();
        if( nRF24L01_is_MAX_RT() ){
            nRF24L01_clear_MAX_RT();
            break;
        }
    }
    nRF24L01_clear_TX_DS();
}
break;
} // Switch-case tapauksen lopetus
} // For-silmukan lopetus
} // Pääohjelman lopetus
```


Vastaanottimen pääkoodi

```
//-----  
// Vastaanotin-moduli  
// Tekijä: Nikita Pavlov  
// 27.4.2013  
//-----  
  
// Kirjastojen liittäminen  
#include <m8c.h>  
#include <stdlib.h>  
#include "PSoCAPI.h"  
#include "PSoCGPIoint.h"  
#include "nRF24L01.h"  
  
// Muuttujien esittely  
int i, laiteID;  
BYTE Len;  
  
void main(void)  
{  
    // Lisämuuttujien esittely  
    BYTE payloadData[2];  
    BYTE vastaanOtettuData[2];  
  
    M8C_EnableGInt ;  
  
    // USBUART-moduulin käyttöönotto  
    USBUART_Start(USBUART_5V_OPERATION);  
    while(!USBUART_Init());  
  
    // SPI-moduulin käyttöönotto  
    SPIM_Start(SPIM_SPIM_MODE_0 | SPIM_SPIM_MSB_FIRST);  
  
    SleepTimer_Start();  
    SleepTimer_SetInterval(SleepTimer_8_HZ);  
    SleepTimer_EnableInt();  
  
    // Pääohjelman esittely  
    while(1){  
        // Ohjelma pysyy lähetystilassa kunnes annetaan lopetuskäsky  
        while(x != 'q'){  
  
            // Luetaan näppäimistö  
            Len = USBUART_bGetRxCount();  
            if (Len){  
                x = USBUART_ReadChar();  
                while (!USBUART_bTxIsReady());  
  
                switch(x){  
  
                    case 's':  
                        USBUART_CPutString("Valitse laite: ");  
                        break;
```

```
case '1':
payloadData[0] = x;
USBUART_PutChar(payloadData[0]);
USBUART_PutCRLF();
USBUART_CPutString("Valitse mittaus: a:Analog i:I2C");
break;

case '2':
payloadData[0] = x;
USBUART_PutCRLF();
USBUART_CPutString("Valitse mittaus: a:Analog i:I2C");
break;

case '3':
payloadData[0] = x;
USBUART_PutCRLF();
USBUART_CPutString("Valitse mittaus: a:Analog i:I2C");
break;

case 'a':
payloadData[1] = x;
USBUART_PutCRLF();
USBUART_CPutString("Valittu laite: ");
USBUART_PutChar(payloadData[0]);
USBUART_CPutString(" Valittu mittaus: ");
USBUART_PutChar(payloadData[1]);
USBUART_CPutString(" Paina L lähettämiseksi ");
nRF24L01_SetAsTransmitter();
break;

case 'i':
payloadData[1] = x;
USBUART_PutCRLF();
USBUART_CPutString("Valittu laite: ");
USBUART_PutChar(payloadData[0]);
USBUART_CPutString(" Valittu mittaus: ");
USBUART_PutChar(payloadData[1]);
USBUART_CPutString(" Paina L lahetäkseen ");
nRF24L01_commonConfiguration();
nRF24L01_SetAsTransmitter();
break;

case 'u':
payloadData[1] = x;
USBUART_PutCRLF();
USBUART_CPutString("Valittu laite: ");
USBUART_PutChar(payloadData[0]);
USBUART_CPutString(" Valittu mittaus: ");
USBUART_PutChar(payloadData[1]);
USBUART_CPutString(" Paina L lahetäkseen ");
nRF24L01_commonConfiguration();
nRF24L01_SetAsTransmitter();
break;

// L-näppäimellä lähetetään asetukset
case 'l':
while( nRF24L01_is_TX_DS() != 1 ){
nRF24L01_sendPayload(W_TX_PAYLOAD, payloadData, 2);
nRF24L01_transmitPayload();
```

```
        if( nRF24L01_is_MAX_RT() ){
            nRF24L01_clear_MAX_RT();
            break;
        }
    }
    nRF24L01_clear_TX_DS();
    payloadData[0] = 0;
    payloadData[1] = 0;
    USBUART_PutCRLF();
    USBUART_CPutString("Asetukset on lahetetty... S: jatkaa / Q:
vastaanotto");
    break;

    // Q-näppäin asettaa lähettimet lähetystilaan ja vastaanotti-
men vastaanottotilaan
    case 'q':
        payloadData[0] = 0x71;
        while( nRF24L01_is_TX_DS() != 1 ){
            nRF24L01_sendPayload(W_TX_PAYLOAD, payloadData, 2);
            nRF24L01_transmitPayload();
            if( nRF24L01_is_MAX_RT() ){
                nRF24L01_clear_MAX_RT();
                break;
            }
        }
        nRF24L01_SetAsReciever();
        break;
    } // Switch-rakenteen lopetus
} // Len-funktion lopetus
} // While-silmukan lopetus

// Mitatun datan vastaanottaminen
if( nRF24L01_is_RX_DR() == 1 ){
    nRF24L01_readPayload( R_RX_PAYLOAD, payloadData, 2);
    nRF24L01_clear_RX_DR();
    laiteID = payloadData[0];
}

SleepTimer_SyncWait(1, SleepTimer_WAIT_RELOAD);

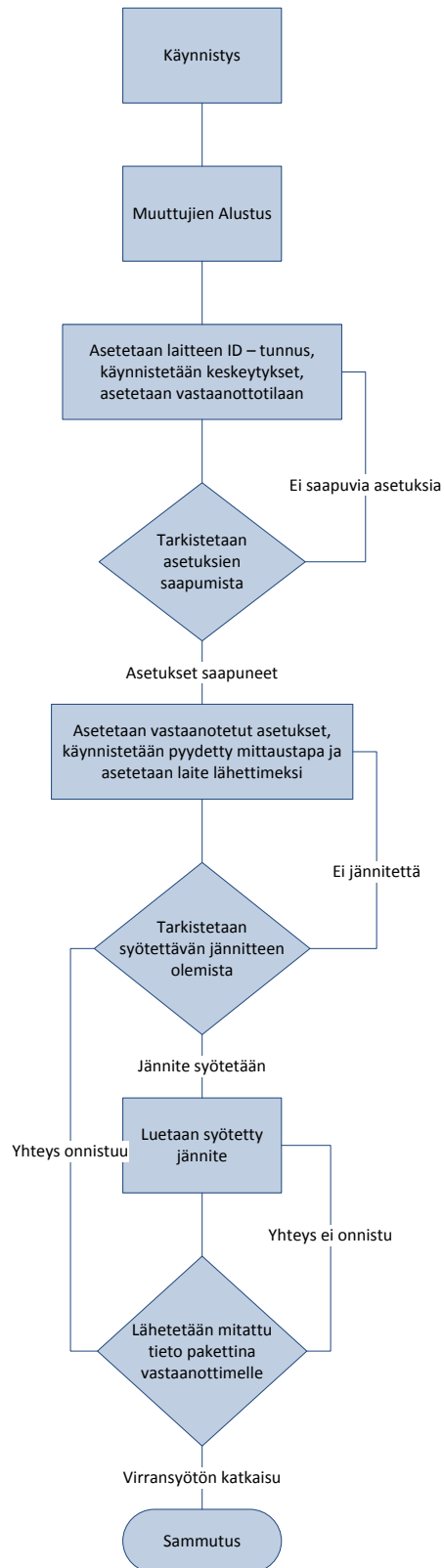
// Vastaanotetut datan tulostus PuTTY-päätteelle
switch(laiteID){

case 1:
    while (!USBUART_bTxIsReady());
    USBUART_CPutString("ID:");
    USBUART_PutSHexByte(payloadData[0]);
    USBUART_CPutString("Data:");
    USBUART_PutSHexByte(payloadData[1]);
    USBUART_PutCRLF();
    break;

case 2:
    while (!USBUART_bTxIsReady());
    USBUART_CPutString("ID:");
    USBUART_PutSHexByte(payloadData[0]);
    USBUART_CPutString("Data:");
    USBUART_PutSHexByte(payloadData[1]);
```

```
USBUART_PutCRLF();  
break;  
  
case 3:  
while (!USBUART_bTxIsReady());  
USBUART_CPutString("ID:");  
USBUART_PutSHexByte(payloadData[0]);  
USBUART_CPutString(" Data:");  
USBUART_PutSHexByte(payloadData[1]);  
USBUART_PutCRLF();  
break;  
    } // Switch-tilan lopetus  
    } // While-silmukan lopetus  
} // Pääohjelman lopetus
```

Lähettimen vuokaavio



Lähettimen vuokaavio

