

TEKNILLINEN KORKEAKOULU

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Ari Sassi

ZIGBEE-TEKNOLOGIA JA SEN OPPILAITOSKÄYTTÖÖN TÄHTÄÄVÄT  
SOVELLUKSET

Diplomityö, joka on jätetty opinnäytteenä tarkistettavaksi diplomi-insinöörin  
tutkintoa varten Espoossa 29.5.2009

Työn valvoja:

Prof. Heikki Saikkonen

Työn ohjaaja:

TkL Matti Welin

Tekijä: Ari Sassi		
Työn nimi: Zigbee-teknologia ja sen oppilaitoskäyttöön tähtäävät sovellukset		
Päivämäärä: 29.5.2009	Kieli: Suomi	Sivumäärä: 6+58
Tiedekunta: Elektroniikan, tietoliikenteen ja automaation tiedekunta		
Professori: Sulautetut järjestelmät		Koodi: Tik-76
Valvoja: Prof. Heikki Saikkonen		
Ohjaaja: TkL Matti Welin		
<p>Tässä opinnäytetyössä keskitytään langattomien likiverkkojen Bluetooth- ja Zigbee-tekniikoihin. Molemmat tekniikat pitävät sisällään varsin yhtenäisiä piirteitä, mutta niiden teknisissä toteutuksissa on merkittäviä eroja. Opinnäytetyön tavoitteena on ollut selvittää Bluetooth- ja Zigbee-verkkojen toiminnalliset erot niin verkon muodostumisen, verkkoon liittymisen kuin itse tiedonsiirron kannalta. Myös itse liityntöjen toteutusta tarkastellaan sekä elektroniikan, että ohjelmiston näkökulmasta. Zigbee on osoittautunut selvästi Bluetoothia yksinkertaisemmaksi ja halvemmaksi ratkaisuksi toteuttaa langaton likiverkkoliityntä. Opinnäytetyön käytännön osuus on yksinomaan Zigbee-liityntän elektroniikan ja ohjelmiston toteutuksessa ja tiedonsiirron analysoinnissa. Tehdyillä testikorteilla ajatut ohjelmat osoittavat Zigbee-verkon muodostumisen, siihen liittymisen ja tiedonsiirron nopeiksi tapahtumiksi.</p>		
Avainsanat: Likiverkot, Zigbee, Bluetooth, Atmel MAC, AT86RF230		

## Esipuhe

Haluan kiittää Professori Heikki Saikkosta ja ohjaajaani Matti Weliniä hyvästä ja kärsivällisestä opinnäytetyön ohjauksesta.

Otaniemi, 29.5.2009

Ari Sassi

## Sisältö

1	Johdanto ja tausta .....	1
2	Bluetooth .....	3
2.1	Yleistä likiverkoista .....	3
2.2	Bluetooth-arkkitehtuuri .....	5
2.3	Bluetooth-protokollapino .....	8
2.3.1	Bluetooth-laitteiden osoitteet .....	10
2.3.2	Bluetooth-linkit .....	11
2.3.3	Bluetooth-paketit .....	12
2.3.4	Bluetooth-laitteen tilat .....	14
2.3.5	Bluetooth-protokollapinin ylemmät kerrokset .....	15
2.3.6	Bluetooth-laitteiden profiilit .....	16
2.4	Yhteenveto .....	17
3	Zigbee .....	18
3.1	Yleistä .....	18
3.2	Zigbee vs Bluetooth .....	18
3.3	Zigbee-verkon komponentit .....	19
3.4	Zigbee-verkon topologiat .....	19
3.4.1	Tähtiverkko .....	19
3.4.2	Puuverkko .....	20
3.4.3	Mesh-verkko .....	20
3.5	Zigbee-verkkoon pääsy .....	21
3.6	Zigbee-protokollapino .....	21
3.6.1	Zigbee PHY .....	22
3.6.2	Zigbee MAC .....	24
4	Zigbee radion integrointi osaksi digitaalista tuotetta .....	29
4.1	Yleistä .....	29
4.2	Zigbee-piiriratkaisut .....	29
4.3	AT86RF230 piirin tilat .....	30
4.3.1	Perustilat .....	30
4.3.2	Laajennetut tilat .....	32
4.4	Atmel MAC .....	34
5	Zigbee prototyyppien rakentaminen ja verkkodemonstraatiot .....	36
5.1	Zigbee käytännössä .....	36
5.1.1	ATMega1281 .....	37
5.1.2	AT86RF230 .....	37
5.2	Piirikaavio- ja levysuunnittelu sekä piirilevyvalmistus .....	39
5.3	PAN-coordinator ja device-sovellukset .....	39
5.4	Zigbee-liikenteen analysointi- ja testaussuunnitelma .....	40
5.4.1	PAN-koordinaattorin toiminta .....	41
5.4.2	Laitteen toiminta .....	41
5.5	Yhteenveto .....	43
6	Tulokset .....	44
7	Yhteenveto .....	47

## Symbolit ja lyhenteet

MP3	Motion Picture experts group
SIG	Special Interest Group
CAN	Controlled Area Network
FHSS	Frequency Hopping Spread Spectrun
GFSK	Gaussian Frequency Shift Keying
PSK	Phase Shift Keying
O-QPSK	Offset Quadrature Phase Shift Keying
8-DPSK	Differential Phase Shift Keying
IMS	Industrial Medical Scientific
ACL	Asynchronous Connection-Less
SCO	Synchronous Connection Oriented
LMP	Link Manager Protocol
L2CAP	Logical Link Control and Adaption Protocol
DM	Data Medium rate
DH	Data High rate
HCI	Host Command Interface
EIA	Electronic Industrial Associates
TIA	Telecommunication Industrial Associates
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
SPI	Serial Peripheral Interface
SDP	Service Discovery Protocol
TSC	Telephone Control and Signalling
QoS	Quality of Service
WAP	Wireless Application Protocol
OBEX	Object Binary Exchange
IrDa	Infrared Data Associates
DSSS	Direct Sequence Spread Spectrum
PAN	Personal Area Network
LR-WPAN	Low Rate Wireless Personal Area Network
ZC	Zigbee Coordinator
ZR	Zigbee Router
ZED	Zigbee End Device
FFD	Full Feature Device
RFD	Reduced Feature Device
CSMA-CA	Carrier Sense Multiple Access Collision Avoidance
SAP	Service Access Point
RF-SAP	Radio Frequence Service Access Point
CCA	Clear Channel Assesment
ED	Energy Detection
LQI	Link Quality Detection
MIB	Management information Base
PHY	Physical layer
MAC	Media Access Control

PLME	PHY Link Management Entity
MLME	MAC Link Management Entity
MPDU	MAC Protocol Data Unit
PSDU	PHY Service Data Unit
CAP	Contention Access Period
CFP	Contention Free Period
GTS	Guaranteed Time Slot

# 1 Johdanto ja tausta

Langattomuus tiedonsiirrossa kiehtoo ihmisiä ja organisaatioita tänä päivänä yhä enemmän. Fyysisen kaapeloinnin kustannukset ja muutoksia jäykistävä luonne on luonut tarpeen kehittää langattomia tiedonsiirtotekniikoita. Langaton tiedonsiirto mahdollistaa myös ratkaisuja, joissa kaapelointi ei tule kysymykseen. Matkapuhelimet ja kulutuselektroniikkaan liittyvät kauko-ohjaimet ovat tuttuja kaikille. Kiinteän liittymän puhelimet ovat vähentyneet. Langattomia ratkaisuja on kehitetty vuosien saatossa eri käyttötarkoituksia silmällä pitäen. Tämän opinnäytetyön kannalta mielenkiinto kohdistuu niin sanottuihin likiverkkoihin, joissa tiedonsiirron kantomatka rajoittuu muutamiin kymmeniin metreihin.

Likiverkon tiedonsiirtoa on kehitetty sekä ihmisten, että automaation näkökulmasta. Ihmisen tarpeisiin jo jonkin aikaa tuotteissa ollut Bluetooth-liityntä löytyy lähes jokaisesta modernista matkapuhelimesta. Bluetoothin käyttö rajoittuu matkapuhelimessa pääsääntöisesti kuvien ja asiakirjojen siirtoon matkapuhelimien tai tietokoneiden välillä, mutta tulevaisuudessa matkapuhelimella voi kenties ohjata kodin elektroniikkalaitteita. Muissa kuin matkapuhelimeissa koko laitteen toimintaperiaate voi perustua Bluetooth-tiedonsiirtoon. Tällaisia ovat esimerkiksi Bluetooth-näppäimistöt, tulostimet, kamerat ja sairaaloiden mittalaitteet. Teollisuus automaatiassa liikkuvien kohteiden tiedonsiirto voi olla toteutettu Bluetoothilla.

Toinen likiverkkojen tiedonsiirtotekniikka Zigbee on kehitetty alun perin koti- ja teollisuusautomaation näkökulmasta. Zigbee on vielä yleisesti tuntematonta tekniikkaa ja sitä ei esimerkiksi matkapuhelimesta ainakaan vielä löydy. Bluetooth- ja Zigbee-tekniikat ovat infrapunatiedonsiirron ohella likiverkkojen tiedonsiirtotekniikat. Infrapuna-tekniikkaa ei tässä opinnäytetyössä käsitellä.

Bluetooth-tekniikka on yleisestä tunnettua niin käyttöominaisuuksiltaan kuin teknisenestikin, koska sen kehitysjuuret juontavat 1990-luvun alkupuolelle. Zigbee-tekniikka on kehitetty 2000-luvun alkupuolella ja vasta viime vuosina on piiriteknikkaa ollut saatavilla. Aihe on siis varsin ajankohtainen ja mielenkiintoinen. Molempien tiedonsiirtotekniikoiden laitteissa on käytön kannalta samaa se, että niiden käyttö on lupavapaata, mikä tarkoittaa sitä, että tiedonsiirto tapahtuu lupavapaalla taajuuskaistalla. Koska molemmat tekniikat tarjoavat käytön kannalta lähes samat ominaisuudet, herää kysymys. Miksi on kehitetty uusi tiedonsiirtotekniikka, joka ei näytä tuovan Bluetoothiin verrattuna mitään uutta vai tuoko Zigbee-tekniikka ainoastaan vaihtoehdoisen tekniikan Bluetoothin rinnalle?

Tämän lopputyön tavoitteena on selvittää, mitkä ovat Bluetooth- ja Zigbee-verkon muodostumisen erot, verkkojen toiminnalliset erot, miten Zigbee-tekniikka on toteutettu piiri- ja ohjelmistotasolla, mihin Zigbee-tekniikka parhaiten soveltuu ja mitä rajoitteita tekniikka sisältää. Lopputyön avulla saadaan myös opetus- ja harjoitusmateriaalia elektroniikka asentajan ammatti- ja erikoisammattitutkinnon opetukseen koulu-

tuskeskus Salpauksessa. Nämä tutkinnot pitävät sisällään osan langaton tiedonsiirto, johon on sisältynyt 433/900MHz AM- ja FM-modulaatiopohjaisia tiedonsiirtoratkaisuja ja Bluetoothia. Tarkoitus on korvata opetuksessa AM- ja FM-pohjaiset tekniikat Zigbeellä. Lisäksi tehdään kolme mikro-ohjainkorttia, joilla muodostetaan Zigbee-verkko, tiedonsiirron analysointia varten.

Opinnäytetyön jaottelu on seuraava. Luvussa kaksi käsitellään likiverkkojen yleisiä ominaisuuksia ja rajoituksia sekä Bluetoothia. Luku kolme keskittyy Zigbeehin. Luvussa neljä paneudutaan tarkemmin Zigbee-liitynnän toteutukseen ja luvussa viisi kuvataan prototyyppikorttien ja testiohjelmien toteutus. Luvussa kuusi on opinnäytetyön tuloksien arviointia. Lopuksi luvussa seitsemän on yhteenveto ja Zigbeeen nykytilan arviointi.



## 2 Bluetooth

### 2.1 Yleistä likiverkoista

Likiverkolla tarkoitetaan lyhyen kantaman langatonta kommunikointia. Tekniikkoina on olemassa infrapuna-, Bluetooth- ja Zigbee-ratkaisut. Vaikka tämä opinnäytetyö keskittyy käytännön osalta täysin Zigbee-tekniikkaan, pätee likiverkkojen yleiset ominaisuudet monelta osin myös Bluetooth-tekniikkaan. Langattomia lähiverkkoja ei tässä lopputyössä käsitellä. Likiverkkojen keskeisin ominaisuus on langattomuus ja lyhyet siirtomatkat, mikä tarkoittaa enintään muutamia kymmeniä metrejä hyvissä olosuhteissa ilman vaimentavia esteitä. Teollisuuden liikkuvat kohteet ovat yksi esimerkki tiedonsiirron langattomuuden tarpeesta.

Likiverkoille on tyypillistä lähiverkkoja vaatimattomammat siirtokaistat. Käytännön siirtonopeudet ovat alle 1Mbit/s. Bluetoothin teoreettinen maksimisiirtokaista on 723,2Kbit/s ja Zigbeeen 250Kbit/s. Kuitenkin näistäkin pienemmät siirtonopeudet ovat käyttökelpoisia laitteissa, joihin liittynyt on tarkoitettu. On ymmärrettävää, että langallinen siirto sopii suurten tietomäärien siirtämiseen paremmin. Bluetoothin avulla on mahdollista siirtää MP3 pakattua ääntä 128Kbit/s nopeudella ja ääni siirtyy hyvälaatuisena vastaanottajalle. Laitteissa, joihin Bluetooth ja Zigbee on tarkoitettu, siirretään varsin pieniä tietomääriä. [1,s. 3-5]

Virrankulutus on keskeinen tekijä likiverkon laitteissa, koska ne toimivat usein akulla tai pattereilla. Tämä asettaa piiriteknikalle vaatimuksia. Käytännössä laitteissa, joihin ei tuoda erillistä virtasyöttöä, on valmiustilan ja toiminnallisen tilan suhde pyrittävä pitämään mahdollisimman suurena. Valmiustilan osuus tulee huomioida ohjelmiston suunnittelussa. [1,s. 9-11]

Likiverkkoliittynän tarvetta voidaan laitteissa tarkastella kahdelta näkökulmalta. Likiverkkoliittynällä saadaan laitteeseen lisäarvoa. Laitteen käyttötarkoitus on jokin muu kuin likiverkkotoiminnat, mutta liittynä lisää laitteen käyttömahdollisuuksia. Matkapuhelimiin on lisätty Bluetooth-liittymä, jotta käyttäjä voisi kommunikoida myös muiden laitteiden kuin matkapuhelimien kanssa. Toisaalta koko laitteen käyttötarkoitus voi perustua likiverkkoliittynään. Tästä ovat esimerkkinä kulutuselektronikan kaukosäätimet ja teollisuusautomaation mittalähettimet. [1,s. 11-14]

Kun tuotteeseen harkitaan likiverkkoliittynää, joudutaan miettimään, mitä liittynällä halutaan ja hyväksytäänkö liittynän ominaisuudet. Seuraavat tuoteominaisuudet tukevat likiverkkoliittynän käyttöä. Likiverkkoliittynä lisää käyttömahdollisuuksia tai ja mukavuutta tai helpottaa laitteen käyttöä. Kytöntänopeus laitteiden välillä ei vaikuta laitteiden ensisijaiseen tehtävään ja saavutettava tiedonsiirtokaista on riittävä. Virran-

syöttötapa saadaan ratkaistua tai se on ilmeinen. Lisäksi liittynnän kantomatka on vä. [1,s. 8-9,15-20]

Zigbee kilpailee Bluetoothin ohella lyhyen kantaman langattomien tiedonsiirtoteknologioiden markkinoilla. Zigbeeen Bluetoothia edullisempi hinta ja yksinkertaisempi piiritekniinen toteutus ohjaa sen käyttöä edullisiin ja yksinkertaisempiin laitteisiin, joihin Bluetooth on liian kallis tai monimutkainen ratkaisu. Zigbee on myös uutta ja kiinnostavaa tekniikkaa Bluetoothin rinnalla. Zigbeeen kehityksen taustalla on ollut tarve saada valmistusteknisesti yksinkertaisempi ja halvempi piiritekniinen ratkaisu, mitä Bluetooth on tähän asti tarjonnut. Moniin lyhyen kantaman langattomiin sovelluksiin Bluetooth on vaatimuksiin nähden ylimitoitettu. Näissä tapauksissa Zigbee tarjoaa hyvän vaihtoehdon.

Infrapunasiirtoa lukuun ottamatta on kahdelle muulle tekniikalle tyypillinen piirre kahden kommunikoivan laitteen yhteydenmuodostusvaihe, joka on epädeterministinen vaihe ja otettava huomioon laitesuunnittelussa. Yhteydenmuodostusvaiheessa laitteiden on ensin löydettävä toisensa. Vaikkakin kohdelaite olisi itsestään ilmoittanut, voi tämä vaihe kestää kymmeniä sekunteja. Yhteydenmuodostusvaiheen jälkeen laitteet voivat kommunikoida. Kahden Bluetooth-laitteen välinen yhteydenmuodostus kaikkineen vie tyypillisesti aikaa neljä viisi sekuntia, mutta enimmillään tähän saattaa mennä 30 sekuntia. Toisaalta taas jokin laite on voitu asettaa tilaan, jossa sitä ei voida verkosta löytää. Kohtalaisen hidas yhteyden muodostusvaihe tekee sekä Bluetoothista, että Zigbeestä käyttökelvottomia tiedonsiirtotekniikoita sovelluksiin, joissa tarvitaan nopeaa vasteaikaa laitteilta. Kun jatkossa puhutaan Bluetooth- tai Zigbee-tekniologiasta, käytetään pelkästään termejä Bluetooth ja Zigbee. [1,s. 21-24]

Likiverkoissa laitteiden suunnittelun keskeisimpiä lähtökohtia on verkon peittoalueen määrittely. Zigbee-laitteiden kantomatka rajoittuu muutamaan kymmeneen metriin, koska tällä saadaan lähetysteho ja virrankulutus minimoitua. Bluetooth-standardiin on määriteltä kolme teholuokkaa. Luokan yksi määrittely ulottaa toimintasäteen maksimissaan sataan metriin ja lähetysteho on 100mW. Luokan yksi kohdalla ei varsinaisesti enää puhuta likiverkosta. Luokka kaksi määrittää kantomatkan kymmenen metriin ja lähetysteho on 2.5mW. Luokka 3 maksimilähetysteho on 1mW ja kantomatka yksi metri. Luokan kaksi Bluetooth- ja Zigbee-laitteiden voidaan katsoa edustavan kantomatkaltaan ja lähetysteholtaan likiverkon käsitettä. [3,s. 21]

Standardi 802.15 kattaa pienen siirtonopeuden ja tehonkulutuksen määrittämiä langattomille likiverkoille. Standardia yllä pitää yhteisö nimeltä Task Group 4, eli lyhyemmin TG4. Standardin 802.15 keskeisimmät reunaehdot ovat seuraavat. [1,s. 6-9]

- tiedonsiirtonopeudet 250, 40 ja 20 kbit/s
- 64- ja 16-bittiset laiteosoitteet
- CSMA-CA siirtotien varausmenetelmä
- verkon automaattinen muodostuminen

- sekä yhteydellinen, että yhteydetön tiedonsiirto
- virranhallintaominaisuudet
- 16 kanavaa 2.4GHz ja 10 kanavaa 915MHz lupavapailla kaistoilla
- yksi kanava 868MHz kaistalla

Tämän opinnäytetyön kannalta tärkeät 802.15 alistandardit ovat 802.15.1 Bluetooth- ja 802.15.4 Zigbee-standardi.

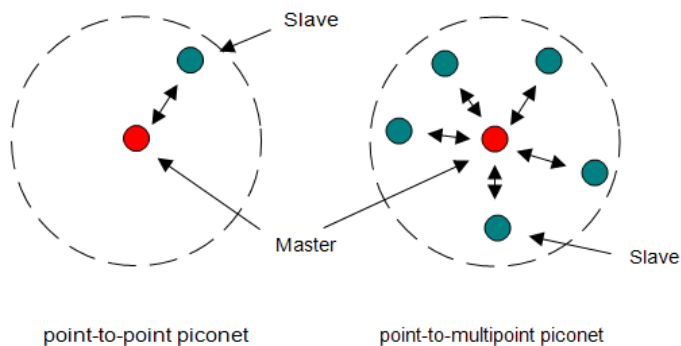
Erickssonin 1990-luvun alkupuolella kehittämä Bluetooth ja nykyisin SIGin ylläpitämä sekä kehittämä langaton lyhyen kantaman tekniikka on kuvattu IEEE standardissa 802.15.1. Alun perin Bluetooth-tekniikassa tavoiteltiin edullista ja vähän virtaa kuluttavaa ratkaisua matkapuhelimien ja niiden lisälaitteiden väliseen kommunikaatioon. Bluetooth-piirit eivät myöskään saaneet maksaa enempää kuin laitteiden välinen kaapeli. 1990-luvun lopussa perustettuun SIG:iin liittyy IBM, Intel, Nokia ja Toshiba. Nykyisin mukana on lähes kaikki keskeiset matkapuhelin-, tietokone- ja autoelektroniikan valmistajat. Vaikka Bluetooth alun perin kehitettiin matkapuhelinympäristöön, on se levinnyt nykyisin lähes kaikkiin lyhyen kantaman langattomiin tiedonsiirtosovelluksiin. Teollisuus on havainnut Bluetoothin käyttömahdollisuudet. Hyvinä esimerkkeinä on erilaisten laitteiden ohjaaminen, laitteiden tilakyselyt ja asetukset sekä Bluetoothin yhdistäminen auton CAN-väylään. [2, s.6-9]

## 2.2 Bluetooth-arkkitehtuuri

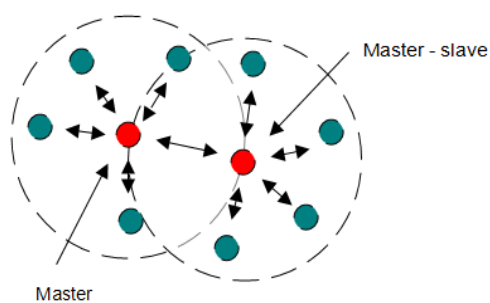
Bluetooth-laitteiden kommunikointi perustuu kahteen perusasiaan. Ensin laitteen on selvitettävä, mitä muita Bluetooth-laitteita peittoalueella on. Tätä kutsutaan discover-toiminnoksi. Kun peittoalueella oleva laite on löytynyt, on siihen muodostettava yhteys ennen kuin tiedonsiirto laitteiden välillä on mahdollinen. Laitteiden välinen kommunikointi perustuu master–slave asetelmaan. Ryhmä laitteita muodostaa piconetin siten, että piconetiin kuuluu yksi master- ja enintään seitsemän slave-laitetta. Piconetissä kommunikointi perustuu siihen, että master määrittää kommunikointiin liittyvän taa-juushyppelysekvenssin, kanavoinnin, ajoituksen ja ohjauksen jota kommunikointi vaatii. Näin olleen piconet on vahvassa masterin ohjauksessa ja samalla alueella voi toimia useampia piconettejä. Huomattavaa on, että slave-laitteet keskustelevat vain master- laitteen kanssa. [1, s. 77]

Master pitää samanaikaisesti yhteyttä useampaan slave-laitteeseen, jos piconetissä on useampia kuin yksi slave-laite. Slave-laitteet muodostavat aina yhteyden master-laitteen kanssa, koska ne eivät saa muodostaa suoraa yhteyttä toiseen slave-laitteeseen. Masterin tiedonsiirto on osoitettu aina vain yhdelle slave-laitteelle. Masterin lähetyksiä slave-laitteille voidaan kuitenkin multipleksoida ja näin syntyy vaikutelma yhtäaikaista yhteyksistä. Liikenteeseen on määritelty myös levitysviestilähetys, joka käsitellään kaikissa slave-laitteissa. Piconetin voi muodostaa myös kaksi laitetta point-to-point tyyppisesti, mutta tässäkin asetelmassa jompikumpi laite toimii master- ja toinen slave-laitteena. Tällainen tilanne on käytännössä yleinen. [2, s.77-78]

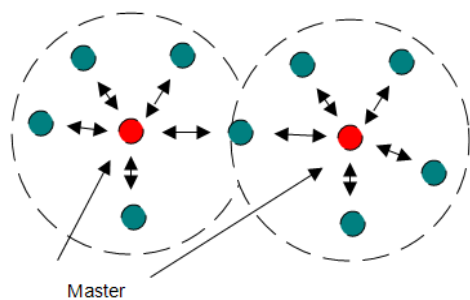
Kuvassa yksi vasemmalla näkyy kahden laitteen muodostama piconet ja oikealla usean laitteen muodostama piconet. Kuvassa kaksi on kaksi piconet-verkkoa, jossa yksi laite esiintyy toisen verkon master- ja toisen verkon slave-laitteena.



Kuva 1: Kahden ja useamman laitteen piconet



Kuva 2: Yksi master- ja yksi master-slave laite

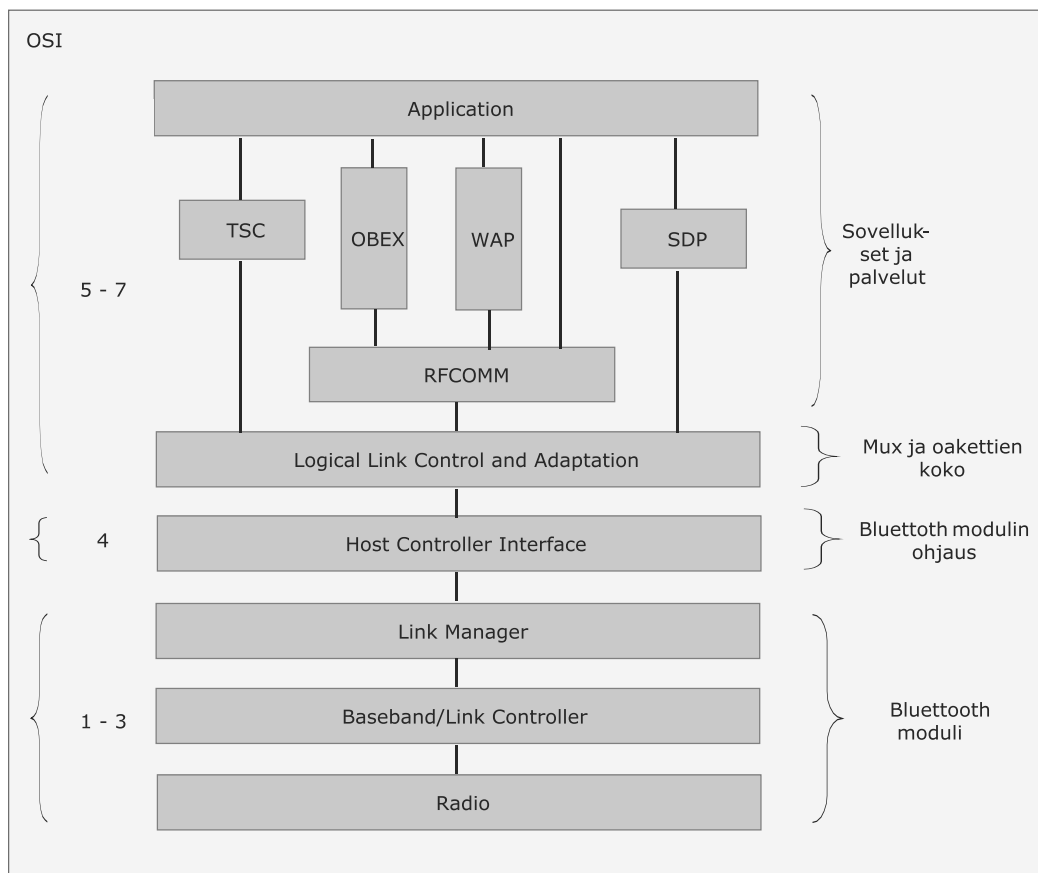


Kuva 3: Yksi slave-laite kahdessa piconetissä.

Kuvassa 3 oleva slave-laite kuuntelee kumpaakin master-laitetta.

Piconet Bluetooth-verkko rajoittaa kommunikoivien laitteiden määrän kahdeksaan. Laajempi verkko voidaan rakentaa scatternetin avulla. Tässä ratkaisussa jokin laite toimii toisessa piconetissa master- ja toisessa slave-laitteena. Laite voi olla slave-laitteena myös kahdessa piconetissä. Mikäli yksi laite kuuluu kahteen piconetiin, tulee sen kyetä kahden piconetin rinnakkaiseen toteutukseen taajuushyppelyiden ja ajoitusten suhteen. Kahden piconetin lähetykset voivat mennä ajallisesti päällekkäin, josta syntyy rajoitteita. Myös tiedonsiirtonopeudet ja radio-osan säätäminen heikentää vasteaikoja. Sama laite ei voi toimia kahdessa piconetissä master-laitteena, koska tällöin taajuushyppelysekvenssit ja tiedonsiirron ajoitukset olisivat samanlaisia limittäin olevissa piconeteissa. Jonkin slave-laitteen on toimittava silta-laitteena kahden piconetin välisessä liikenteessä. Bluetooth scatternet on standardissa löyhästi määritelty ja sitä on tarkennettu Zigbeetä määritettäessä. Master-laite on vastuussa slave-laitteiden pollaamisesta ja yhteyden kaistan varaamisesta tai yhteyden estämisestä. Master myös asettaa piconetin synkronointikellon sekä päättää taajuushyppelysekvenssistä. [3, s.8-9]

## 2.3 Bluetooth-protokollapino



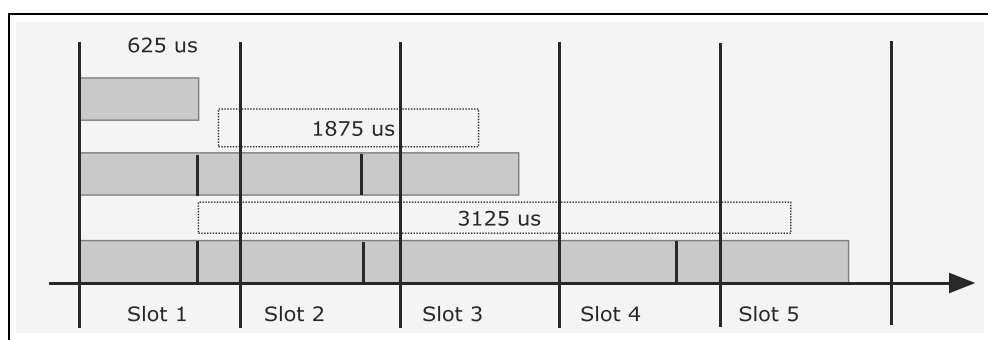
Kuva 4: Bluetooth-protokollapino (Hytönen 2003)

Kuva neljä esittää Bluetooth-protokollatasot. Bluetooth-pinon fyysinen kerros perustuu GFSK-modulaatiopohjaiseen taajuushyppelyyn. GFSK-modulaatiossa tuotetaan binääri yksi positiivisella taajuusjaolla ja nolla negatiivisella taajuusjaolla. GFSK ei vaadi monimutkaista demodulaatiota ja laitteen radio-osa on yksinkertaisempi. GFSK tuottaa maksimi tiedonsiirtonopeudeksi 1Mbit/s. Nopeampi lähetyskaista saadaan käyttämällä PSK-modulaatiota, mutta tässäkin tapauksessa otsikko lähetetään käyttäen GFSK-modulaatiota. Tällä varmistetaan jokaisen kehyksen vastaanoton onnistuminen radio-osassa. PSK-modulaatiolla on mahdollisuus saavuttaa 2Mbit/s käyttäen /4-DQPSK modulaatiota. 3 Mbit/s lähetysnopeus saadaan käyttämällä 8-DPSK-modulaatiota. [2, s. 79-80]

Bluetooth käyttää 2.4 GHz lupavapaata IMS-taajuusalueita. Taajuuskaista on 2.400–2.4835GHz. 83.5MHz kaista on jaettu 79 1MHz kanavaan. Näissä kanavissa käytettävä symbolinopeus on 1M symbolia/s. Informaatio lähetetään paketeissa ja jokaisen pa-

ketin jälkeen vaihdetaan käytettävää taajuutta taajuushyppelyn FHSS mukaisesti. Paketti voi olla pituudeltaan 1, 3 tai 5 aikaväliä. Aikaväli time slot on 625 $\mu$ s. Näin ollen maksimihyppymäärä on 1600 taajuushyppyä/s. Inquiry- ja paging-toiminnot käyttävät vain puolet aikavälistä ja näiden kohdalla tapahtuu 3200 taajuushyppelyä sekunnissa. Koko taajuuskaistan 79 kanavaa käytetään taajuushyppelyssä tai vähintään 20 kanavaa.

Kanavahyppely tarjoaa perussalauksen siirtoon ja mikäli jollakin kanavalla on kohinaa tai etenemisiongelmiä, ei sitä käytetä pitkään aikaa. Bluetooth-laite käyttää tavallisesti vai yhtä kanavaa yhdellä kertaa, mutta usean kanavan rinnakkainen käyttö on myös mahdollista. Pienten pakettikokojen ja lyhyiden aikavälien takia on radio-osien ja oskillaattorien valmistaminen tarkkaa. Lisäksi taajuushyppely edellyttää nopeaa vaihtamista ja säätämistä eri taajuuksille. Radio-osalle esitetään siis huomattavia vaatimuksia verrattuna kantataajuusjärjestelmiin. [2, s. 80-81][3, s. 16-19]



Kuva 5: Bluetooth-pakettien koot ja kestot.

Master-laite ilmoittaa slave-laitteille oman kellotietonsa ja laiteosoitteensa, joista slave-laitteet laskevat käytettävän hyppelysekvenssin. Slave-laitteen kytkeytyessä master-laitteeseen, ilmoittaa master slavelle edellä mainitut tiedot. Menetelmästä johtuu, että vierekkäiset piconetit käyttävät eri hyppelysekvenssejä ja samoilla taajuuksilla tapahtuvia lähetyksiä tapahtuu vähän. Siirto on tahdistettu master-laitteen kelloon. On tärkeää, että slave-laite vastaa masterille samaa taajuutta käyttäen, mitä master käytti. Tämän jälkeen vaihdetaan taajuutta sekvenssin mukaisesti. Master-laite määrittää myös siirto-kapasiteetin käytön. Datalähetyksissä vastataan masterin lähettämiin paketteihin. Slave ei saa alkaa itse lähettämään. Äänen lähetyksessä master allokoii slave-laitteelle tietyn määrän aikavälejä, esimerkiksi joka kolmannen, lähetystä varten. Slave lähettää tällä tavalla, vaikka yhteys masteriin olisi poikki. Ensimmäisessä Bluetooth-standardissa käytettiin kiinteää kelloasetusta, mutta myöhemmin on määritelty erimittaisia aikavälejä. Vaikka tässä yhteydessä puhutaan pakettikäsitteestä, on kyseessä kuitenkin normaali siirtoyhteyskerroksen kehyslähetyksen ja vastaanotto toiminta. Kuten muissakin siirtojärjestelmissä, on myös Bluetooth-siirrossa oma siirtoyhteyskehysrakente, joka alkaa ns. access code-osalla, joka ilmentää kehysalkua. [3, s. 18-21]

Fyysisen kerroksen siirrossa yksi looginen siirtokanava vastaa yhtä fyysistä siirtotaajuutta. Taajuuksia on siis 79 tai 23 siirtokanavalla. Ne laitteet, joilla on sama taajuushyppelysekvenssi, muodostavat piconetin. Jokainen kanava on jaettu aikaväleihin. Jo-

kainen aikaväli on liitetty kanavaan eli tiettyyn taajuuteen. Näin ollen kaksi peräkkäistä aikaväliä tarkoittaa kahta taajuutta hyppelysekvenssin mukaisesti. Ensin master-laite käyttää lähetyksessään slave-laitteelle aikaväliä ja slave-laite vastaa käyttäen heti seuraavaa aikaväliä. Multipleksatussa lähetyksessä master lähettää parillisissa aikaväleissä ja slave parittomissa. Ainoastaan discovery-toiminnassa kaksi slave laitetta voi lähettää toisilleen. [2, s. 81-82] [3, s. 16-17]

Kun Bluetooth-laite käynnistyy, se lähettää itsestään tietoja. Nimi, teholuokka, laitteen tukemat palvelut, teknisiä tietoja, laitteen tunnistetiedot, master-valmius ja minkä määrittelyn mukainen laite tiedot lähettää. Mahdollisesti myös muita tietoja lähetetään. Laite, joka ensimmäisenä lähettää tietonsa, asettuu masteriksi, mikäli master-laitetta ei vielä ole olemassa. On mahdollista, että laite ei halua olla master, jolloin masteriksi asettuu jokin muu laite. Mikäli master-laite sammutetaan, se poistuu kantoalueelta tai se ei jaa lähetyksivuoroja, on master-laite valikoitava uudelleen. [3, s. 19-20]

### 2.3.1 Bluetooth-laitteiden osoitteet

Bluetooth-laitteissa käytetään neljää eri osoitetta. [2, s. 84-85]

**BD\_ADDR:** Laitteen kiinteä valmistajan asettama yksilöllinen 48-bitin MAC-osoite. Tämä on ainut kiinteä osoite laitteessa ja sitä käytetään vain indikoimaan viestinnän aloitusta. Tällä osoitteella laitteet erottuvat toisistaan verkon luontivaiheessa. Sivulla 11 kuvassa kuusi on esitetty osoitteen rakenne.

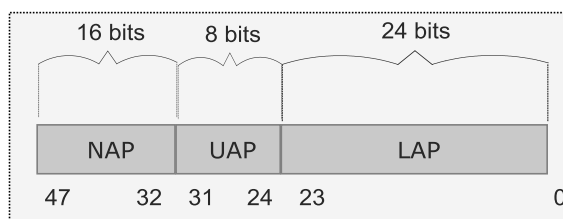
**LT\_ADDR:** 3-bittinen masteriin kytkeytyneen laitteen osoite. Osoite on voimassa vain sen aikaa, kun laite kuuluu piconetiin. Osoite tarvitaan masterin slave-lähettyksiin.

**PM\_ADDR:** 8-bitin osoite, joka on voimassa vain kun laite on parked-tilassa, eli ei aktiivisena piconet jäsenenä. Master antaa osoitteen slavelle, kun laite siirtyy parked-tilaan.

**AR\_ADDR:** Masterin antama tilapäinen parked-tilan osoite. Tällä osoitteella master ilmoittaa slavelle seuraavan aikavälin, jolla asianomainen slave voi lähettää.

IEEE 48-bitin MAC-osoite koostuu kolmesta osasta kuvan kuusi mukaisesti. NAP (Non Significant Address Part) osaa käytetään salauksen muodostuksessa. UAP (Upper Address Part) osaa käytetään taajuushyppelyssä sekä virheen tarkastuksessa. LAP (Lower Address Part) osaa käytetään taajuushyppelyssä ja synkronointisanan luonnissa.





Kuva 6: Bluetooth IEEE MAC-osoite

### 2.3.2 Bluetooth-linkit

Tiedonsiirto master- ja slave-laitteiden välillä hoidetaan paketeilla. Näistä paketeista muodostuu tiedonsiirtolinkit. Linkin tyyppi määräytyy tiedonsiirron luonteen perusteella yhteyden muodostusvaiheessa. Bluetooth-yhteydessä on kahden tyyppisiä linkkejä: [3, s. 33-39]

ACL-linkki vastaa pakettikytkentäistä yhteyttä

SCO-linkki vastaa piirikytkentäistä yhteyttä.

ACL-linkissä tiedonsiirtoyhteyden nopeus saattaa vaihdella, koska lähetettäviä paketteja kuitataan. Myös mahdolliset uudelleenlähetykset hidastavat tiedonsiirtoa ja voivat aiheuttaa viivettä. SCO-linkissä sovitaan yhteydenmuodostusvaiheessa tiedonsiirtonopeus ja muita siirtoparametreja. Nopeuden sopimisella pyritään pitämään tiedonsiirron viiveet samoina. Näin ollen kuittauspaketit ja uudelleenlähetykset eivät tule kysymykseen SCO-linkissä. SCO-linkin luonteen perusteella se sopii reaaliaikaiseen siirtoon.

Seuraavassa on koottu ACL-linkin tärkeimmät ominaisuudet.

- master voi ylläpitää montaa ACL-linkkiä yhtä aikaa
- master päättää tiedonsiirrosta, ajasta ja suunnasta slavelle
- masterin ja slaven välillä voi olla vain yksi ACL-linkki
- kun yhteyden on luotu, on ACL-linkki olemassa
- koska ylemmiltä kerroksilta saadaan tieto lähetettäväksi, on lähetys satunnaista
- levitysviestit käyttävät ACL-linkkiä
- sisältää virheentarkastuksen ja uudelleenlähetysten
- ACL-linkin sisällä kuljetetaan LMP (ohjaustietoja) tai L2CAP (varsinaista user dataa, DM- tai DH-paketeissa) -paketteja

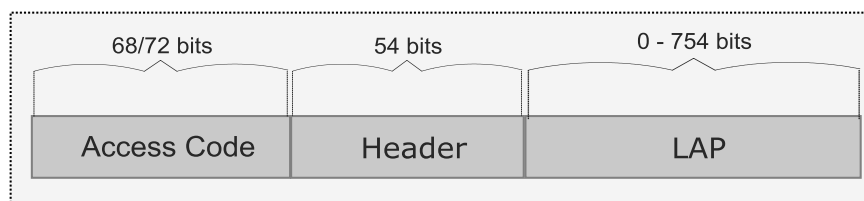
ACL-linkki on toteutettu kaikissa Bluetooth-laitteissa.

Vastaavasti SCO-linkin tärkeimmät ominaisuudet on lueteltu seuraavassa.

- ei uudelleenlähetyistä, varattu siirtokapasiteetti ja aikaväli
- master voi ylläpitää enintään kolmea rinnakkaista SCO-linkkiä yllä ja ne voivat olla samaan slave-laitteeseen
- linkin tiedonsiirtonopeus ylläpidettävä
- SCO-linkin liikenteen voi syrjäyttää vain Link Management-ohjaus
- kaikissa laitteissa ei ole välttämättä SCO-linkkiä toteutettu, jos puhetta tai ääntä ei tarvitse siirtää

### 2.3.3 Bluetooth-paketit

Bluetooth-paketti kulkee joko ACL- tai SCO-linkissä. Kuvassa seitsemän on esitetty paketin rakenne. Access Code on piconet kohtainen ja se liittää paketin oikeaan masteriin tai sitä käytetään inquiry-vaiheessa. Access Code koostuu 4-bitin tahdistuskuvioista (0101 tai 1010), synkronointiosasta ja trailerista. Tahdistusbittien tarkoitus on tahdistaa vastaanottimet. 64-bittinen synkronointisana on muodostettu laiteosoitteesta matemaattisesti. Mikäli tämä bittikuvio on sama kuin laitteessa oleva talletettu synkronointisana, syntyy laitteessa korrelaatiopiikki. Mikäli tämä syntyy, jatkaa laite paketin kuuntelua, muussa tapauksessa laite sammuttaa radio-osan. Elektroniikassa tapahtuva korrelaatiopiikki on merkki siitä, että paketti on laitteelle tarkoitettu ja laite ottaa koko paketin vastaan. [3, s. 42-48]

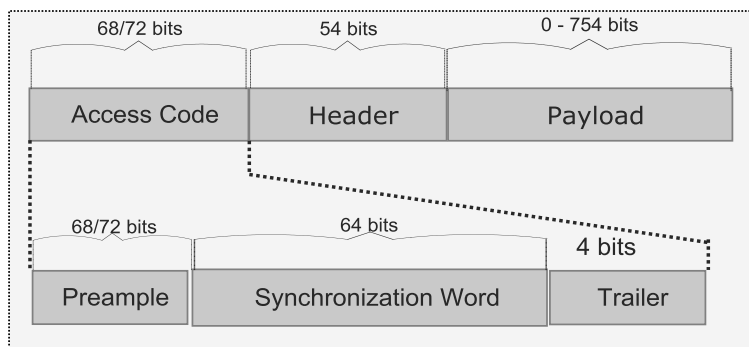


Kuva 7: Bluetooth-paketin rakenne

Access Code on tärkeä, koska sen sisältö määrittää, mikä on paketin tarkoitus. Seuraavassa on eri koodeja toiminnoille:

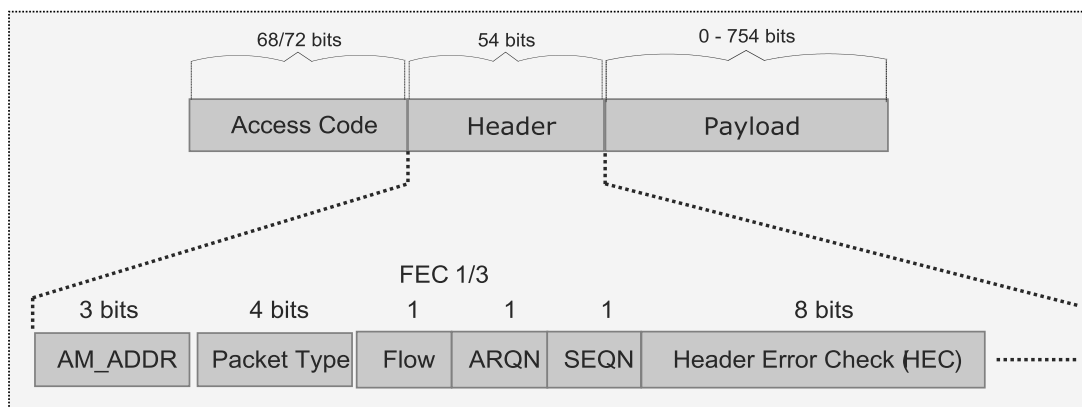
- CAC (Channel Access Code), tiedonsiirtokoodi piconeissä
- DAC (Device Access Code), page- ja page scan-toimintojen koodi
- GIAC (General Inquiry Access Code), inquiry-toiminnon koodi
- DIAC (Dedicated Inquiry Access Code), kun haetaan inquiryllä tietyillä ominaisuuksilla olevaa laitetta

Kun laite on saanut vastaan Access Coden, se päättää jatkaako se paketin vastaanottamista.



Kuva 8: Access Code-osan rakenne

Access-osan jälkeen tulee paketin otsikko eli header. Kuvassa yhdeksän on paketin otsikon rakenne. Ensimmäisenä on 3-bittinen AM\_ADDR eli laitteen masterilta paging-vaiheessa saama osoite. Tämä tarvitaan sekä masterin ja slaven lähettämien pakettien tunnustamiseen. Mikäli tämä osoite on nolla, se tarkoittaa levitysviestiosoitetta, jonka kaikki laitteet käsittelevät.

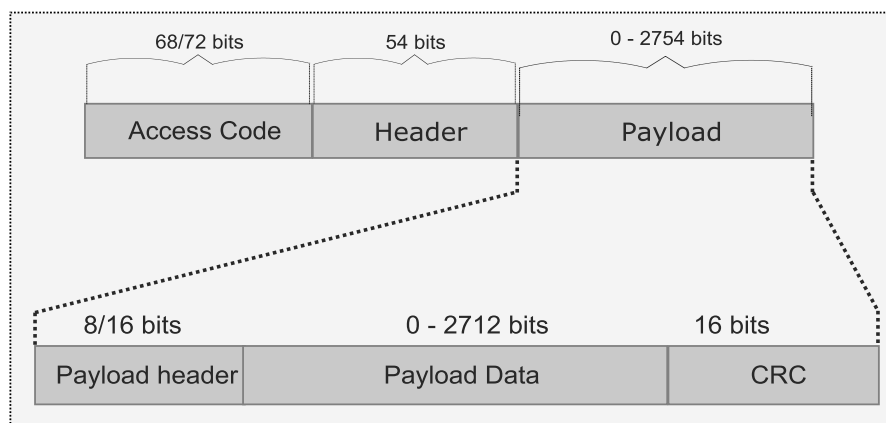


Kuva 9: Packet-Header rakenne

Packet type-kenttä ilmoittaa paketin tyyppin, pituuden ja dataosan virheentarkistuksen. ACL- ja SCO-, NULL tai POLL-paketeilla on omat tunnisteet. Packet type kertoo vastaanottajalle myös dataosan virheentarkastusmenetelmän. [3, s. 45]

Flow-bitti antaa vastaanottajalle mahdollisuuden hidastaa lähettäjän lähetystä. ARQN-bitillä vastaanottaja kuittaa virheettömästi vastaanotetun paketin. SEQN-bitti invertoidaan jokaiseen lähetykseen, jotta vastaanottaja erottaisi uudelleenlähetetyn paketin peräkkäisistä lähetyksistä. Kuvassa yhdeksän on esitetty Packet-header rakenne.

ACL-hyötykuormaosuudessa on oma otsikko, joka määrittää, onko kyseessä data- vai ohjausviesti eli L2CAP- vai LMP-viesti? Otsikossa on myös tieto, onko data sarjan viimeinen vai ensimmäinen osa ja paljonko hyötykuorman pituus on. Lopussa on virheentarkistustieto CRC. Kuvassa 10 on esitetty ACL-hyötykuormaosan rakenne.



Kuva 10: ACL-hyötykuormaosa

SCO-paketin rakenne on sama kuin ACL-paketin, mutta hyötykuormaosa on vakio-  
mittainen 240-bittinä, koska SCO-lähetysnopeus on sovittu. SCO-liikenteessä ei ole  
vuonohjausta eikä uudelleenlähetystä.

Bluetooth-liikenteeseen on määritelty myös erikoispaketteja. [3, s. 49]

- ID: pakettia käytetään ennen kuin linkki on luotu ja siinä kulkee tietona vain tietona access code.
- NULL: pakettia käytetään pakettien kuittauksessa tai vuon ohjauksessa. Sisältää tietona vain access coden.
- POLL: pakettia käyttää master-laite slave-laitteen tarkistukseen. Tällä master varmistaa, että slave-laite on olemassa. Paketti on kuitattava ja rakenne on sama kuin NULL-paketissa.
- FHS: paketissa kulkee taajuushyppelyyn ja ajoitukseen liittyvät tiedot. Tällä paketilla lähetetään master-slave roolin vaihdot, master-roolin ilmoitus tai inquiry-paging-toiminnot.

### 2.3.4 Bluetooth-laitteen tilat

Bluetooth-laitteilla on kolme vähävirtaista tilaa. Eniten virtaa vievä säästötila on Sniff. Tässä tilassa laite vähentää verkkoaktiiviteettia ja kuuntelee harvakseltaan, onko masterilla asiaa. Laite synkronoi kellonsa kuitenkin masterin kelloon. Hold-tilassa laite ainoastaan kuuntelee masterin ohjetta poistua Hold-tilasta. Muuta verkkoaktiiviteettia tässä tilassa ei ole, paitsi kelloahdistus. Viimeinen säästötila on Parked. Tässä tilassa laite ei siirrä dataa, tahdistaa harvoin itsensä masterin kelloon, mutta kuuntelee kaikille laitteilla tarkoitettuja levitysviestejä. Kun yhteys masteriin on luotu ja tiedonsiirto on alkanut masterin ja slaven välillä, ovat laitteet Active-tilassa. Standby-tilassa datasiirtoa ei tapahdu, radio-osa on sammutettu, laite ei vastaa kutsuun ja laite on passiivinen. In-

quiry scan-tilassa laite vastaa inquiry-kyselyyn. Inquiry-tila mahdollistaa laitteiden löytymisen. Laitteen ohjelmisto ei välttämättä siirrä laitetta milloinkaan inquiry scan-tilaan ja silloin laitetta ei löydetä. Tällöin laite ei halua kommunikoida Bluetooth:lla.

Kun yhteys halutaan muodostaa, tarvitaan tiloja page- ja page scan. Page-tilassa oleva laite ottaa yhteyttä page scan-tilassa olevaan laitteeseen. Laitteet aina aika ajoin menevät page scan-tilaan, jotta niihin saadaan luotua yhteys. [2, s. 37-39]

### 2.3.5 Bluetooth-protokollapinon ylempät kerrokset

Kuvassa neljä sivulla kahdeksan näkyy, miten edellä kuvatut toiminnot sijoittuvat OSI-malliin. Fyysisellä kerroksella toimivat radio- ja kantataajuusosat. Radio-osan tehtävä on välittää tieto radiotaajuuden avulla siirtomedian eli tässä tapauksessa ilmatien yli. Kantataajuusosan hoidettavaksi jää kanavien ja siirtoon liittyvien ajoitusten hallinnat. Siirtoyhteyskerroksella toimiva Link Controller hoitaa siirtotien, linkin ohjausta Link Managerilta saamiensa ohjeidenmukaisesti. Verkkokerroksella toimii Link Manager. Nämä kolme alinta kerrosta hoitavat Bluetooth-protokollapinossa seuraavat toiminnot. Sivun kahdeksan Bluetooth-protokollapinokuvassa nämä vastaavat kerroksia 1-3. Verkkokerroksen tehtävät ovat seuraavat: [1, s. 70-75]

- pakettien välitys ilmatien yli
- kanavien koodaus ja purku
- osoitteistus
- datasiirron ajoitus, virheentarkistus, ja -korjaus

Protokollapinossa kuljetuskerrokselle, eli kerrokselle neljä, on määritelty HCI (Host Controller Interface). HCIn tarkoitus on erottaa alempien kerroksien aikakriittiset toiminnot omaksi kokonaisuudekseen ja määrittää standardoitu rajapinta. Tällöin alempien kerroksien toiminnot voidaan toteuttaa omalla prosessorilla vaikkapa omana modulina, jota ohjataan HCI-rajapinnan kautta. Tämä mahdollistaa modulaarisen suunnittelun ja selkeyttää laitesuunnittelua. HCI on toteutettu käytännössä kaikissa laitteissa, vaikka alemmat kerrokset olisivatkin kiinteää toteutusta ylempien kanssa. HCI:ssä on määritelty seuraavat pakettityypit: [3, s. 54-58]

- Command Packet. Tällä ohjataan modulin toimintaa ja saa selville modulin tila
- Event Packet. Tällä moduli informoi ylempää kerrosta tapahtumista
- Data Packet. Tällä paketilla välitetään ääntä tai dataa moduliin tai moduliin, SCO- tai ACL-linkeissä.

HCI-paketit ovat kuljetuskerroksen toteutusta ja liityntään on määritelty kolme rajapintaa: EIA/TIA 232, UART ja USB. Niistä voidaan valita, mikäli päädytään modulitoteutukseen. Koska kuljetuskerros päättyy, millä tavalla yhteyttä toiseen laitteeseen otetaan tai pidetään, on HCI:ssä joukko käskyjä näitä toimintoja varten. Alla on mainittu tärkeimpiä HCI-toimintoja. [3, s. 54-58]

- linkkien luomiseen ja ohjaamiseen liittyvät käskyt
- roolinvaihdot ja laitteen eri tilat
- datan siirto moduulin välillä ja kantataajuusosan säätäminen
- HCI-modulin tilatietojen käsittely

L2CAP-yhteyksikäytäntö liikuttaa dataa joko HCI-kerrokselle tai LM:lle. Tätä yhteyksikäytäntöä käytetään kaikessa datasiirrossa, mutta ei äänen siirrossa. Data saadaan tai annetaan joko sovellukselle tai ylemmälle kerrokselle. L2CAPin alapuolella on siis HCI tai LM ja yläpuolella joko sovellus tai SDP- tai TSC-yhteyksikäytännöt. Seuraavassa on L2CAP:n tärkeimmät tehtävät. [3, s. 59]

- multipleksoida ylempien kerroksien lähetykset numeroimalla ne ja vastaanottava laite jakaa lähetyksen. Tähän käytetään samaa ACL-linkkiä.
- datapakettien segmentointi ja kasaus vastaanotossa..
- yhdensuuntainen multicast-lähetystuki.
- ylempien kerroksien QoS-parametrien toteutus.

OSI-mallin mukaisesti kerroksille viisi-seitsemän on määritelty joukko palvelu- ja sovellusyhteyksikäytäntöjä. Alla on mainittu lyhyesti jokaisen yhteyksikäytännön tarkoituksesta. [3, s. 60-65]

- RFCOMM. Sovelluksen tai fyysisen laitteen sarjaporttiliityntämahdollisuus
- SDP. Laitteiden resurssien selvittämiseen kantosäteellä, jokaisessa laitteessa palvelin- ja asiakasosat tätä varten
- WAP. Datapalvelujen käyttämiseen, tarvitsee RFCOMM:ia
- OBEX/IrDA. Molemmat on tarkoitettu tiedonsiirtoon
- TCS. Puhelun välittämiseen tarkoitettu yhteyksikäytäntö

### 2.3.6 Bluetooth-laitteiden profiilit

Bluetooth-laitteiden suunnittelun yhtenä lähtökohtana ovat profiilit, joka määrittävät laitteen kyvyt. Näillä kyvyillä tarkoitetaan ominaisuuksia, jotka tietyllä profiililla varustetussa laitteessa on toteutettava. Profiilien avulla laitteiden välinen kommunikointi on helpompi toteuttaa ja ne määrittävät valmiuksia laitteille. Laitteen käyttötarkoituksen mukaan toteutetaan haluttu tai haluttuja profiileja laitteessa. Laitteen suunnittelija hakee profiilien toteutuksella laitteelle myös yleiskäyttösyöttä ja laajentaa laitteen käyttömahdollisuuksia. Profiileja voidaan pitää laitteen sovelluksina. Generic Access-profiili on kaikkien profiilien pohjana, koska se määrittää laitteen valmiudet yhteydenmuodostukseen ja eri tiloihin sekä kaikille laitteille yhteiset ominaisuudet. [1, s. 43]

## 2.4 Yhteenveto

Bluetooth- ja Zigbee-tekniikat ovat toiminnaltaan varsin lähellä toisiaan, joten Bluetooth toimii hyvänä vertailutekniikkana Zigbeelle. Erityisesti kolmen alimman OSI-kerroksen toimintojen erot ovat mielenkiintoisia samoin kuin itse verkon muodostamisen erot.

## 3 Zigbee

### 3.1 Yleistä

Zigbee on Bluetoothia pienemmällä kaistalla, pienemmällä virralla, halvemmalla elektroniikalla toteutettu likiverkkotekniikka automaatio- ja ohjaussovelluksiin. Zigbee-laitteiden käyttöaika akuilla on tyypillisesti vähintään kuukausia, ellei vuosia, sillä pienemmät siirtonopeudet ja yksinkertaisempi elektroniikka kuluttavat Bluetoothia selvästi vähemmän virtaa. Zigbee-liitynnän elektroniikkakustannukset ovat TIA/EIA 232-liitynnän hintatasoa. Zigbee-verkon kattavuus voi ainakin teoriassa olla Bluetooth-verkkoa laajempi. IEEE ja Zigbee Alliance ovat yhdessä määrittäneet tekniikan käyttönimeksi Zigbee. IEEE on lisäksi määritellyt termit PAN (Personal Area Network), LP-PAN (Low Rate PAN), ZC (Zigbee Coordinator), ZR (Zigbee Router) ja ZED (Zigbee End Device) liittyväksi Zigbee-tekniikkaan. [4, s. 5-8]

### 3.2 Zigbee vs Bluetooth

Zigbeeen ja Bluetoothin ominaisuuksien vertailu tuo esiin tekniikoiden erot parhaiten. Seuraavassa on vertailtu tekniikoita käytön kannalta: [6, s. 7-9]

- Zigbeeen toimintasäde ilman vahvistinta on 10-75m, Bluetoothissa 10m
- Zigbeeen lähetysopeudet ovat 250 kbit/s (2.4 GHz), 40 kbit/s (915 MHz), 20 kbit/s (868 MHz) kun Bluetoothissa puhutaan 1 Mbit/s tai hieman sen alle
- Zigbee-verkko on perus master-slave tähtiverkko, jossa laitteet lähettävät pieniä paketteja. Laitteita voi olla enintään 254 kpl. Perus Bluetooth-piconet sisältää enintään kahdeksan laitetta.
- koska Zigbee-verkot on tarkoitettu automaatio- ja sensoriverkoiksi, on tiedonsiirto-protokolla Bluetoothia paljon yksinkertaisempi. Bluetooth-protokollan täytyy pysyä kuljettamaan sekä ääntä että dataa.
- heräävä Zigbee-laite kykenee lähettämään ja vastaanottamaan dataa alle sadassa millisekunnissa, kun taas Bluetooth-laitteen pelkkään yhteydenmuodostusvaiheeseen menee vähintään 2-3 sekuntia.

Zigbee-laitteen odotetaan toimivan kahdella AA-paristolla kuudesta kuukaudesta kahteen vuoteen. Suurimman osan ajasta Zigbee-laite on nukkumassa.



### 3.3 Zigbee-verkon komponentit

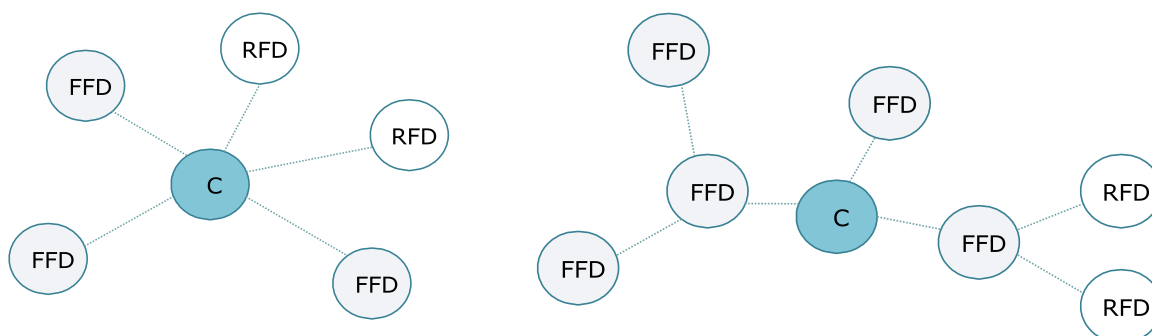
Zigbee-verkon keskeinen komponentti on laite. Laite voi olla täysillä (FFD) tai typistetyillä (RFD) ominaisuuksilla oleva. Täysillä ominaisuuksilla oleva laite voi toimia verkon PAN-koordinaattorina tai laitteena. Typistettyjen ominaisuuksien laite on tiedon siirron kannalta yksinkertainen laite, joka lähettää ja vastaanottaa pieniä tietomääriä. Typistettyjen ominaisuuksien laite voi kommunikoida vain täyden ominaisuuksien laitteiden kanssa, kun taas täyden ominaisuuksien laitteet voivat kommunikoida molempien kanssa. [2, s. 110-112]

### 3.4 Zigbee-verkon topologiat

Zigbee-verkon rakentamiseen on tarjolla kolmea eri topologiaa, tähti, laajennettu tähti ja mesh. Kaikilla Zigbee-laitteilla on verkon topologiasta riippumatta yhtäläinen pääsy verkkoon. [6, s. 25-26] [4, s. 6-7]

#### 3.4.1 Tähtiverkko

Tähtiverkossa laitteet kommunikoivat yhden laitteen kanssa, joka toimii PAN-koordinaattorina. PAN-koordinaattori toimii verkon solmupisteenä, jonka kautta kaikki liikenne laitteiden välillä kulkee. Laitteet ovat tiedonsiirron kannalta riippuvaisia PAN-koordinaattorista. PAN-koordinaattori allokoii verkko-osoitteet laitteille. Tähtiverkosta käytetään myös nimityksiä one-hop tai single-hop multi-access verkko, koska laitteet lähettävät aina PAN-koordinaattorille toiselle laitteelle menevät sanomat. PAN-koordinaattorin on oltava FFD laite. Kuvassa 11 on tähtiverkko, jonka keskellä on PAN-koordinaattori. Tummat ympyrät kuvaavat FFD- ja valkoiset RFD-laitetta. [4][5]



Kuva 11: Vasemmalla tähtiverkko ja oikealla puuverkko

Tähtiverkon PAN-koordinaattori on täysaikaisesti sähköt päällä oleva laite kun taas muut laitteet toimivat usein pattereilla tai akuilla ja nukkuvat suurimman osan ajasta. Tähän luetaan esimerkiksi kotiautomaatio-, tietokone- ja pelilaitteet. Kun täysillä ominaisuuksilla oleva laite käynnistyy, se voi halutessaan muodostaa verkon ja olla verkon PAN-koordinaattori. Kun verkko käynnistyy, valitaan PAN-tunniste, joka ei ole peitto-alueella käytössä. Zigbee tähtiverkko vastaa Bluetooth piconet-verkkoa. [4, s. 4-5]

### 3.4.2 Puuverkko

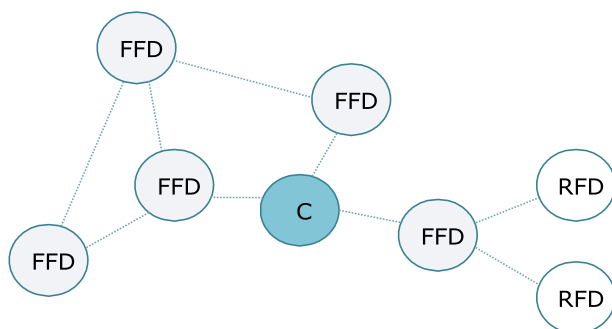
Tässä topologiassa laitteet voivat liittyä verkkoon koordinaattorin tai reitittimenä toimivan FFD-laitteen kautta. Verkon reitittimellä on kaksi tehtävää. Laitteet, jotka ovat kantomatkan ulottumattomissa koordinaattoriin, voivat liittyä verkkoon reitittimen kautta. Koska laitteet eivät voi tässäkään kommunikoida suoraan keskenään, osaa reititin kommunikoida toisten reitittimien ja verkon koordinaattorin kanssa. Verkon muodostus tapahtuu seuraavasti.

Koordinaattori muodostaa ensin varsinaisen tähden ja ottaa itselleen tähden pään roolin. Tämän jälkeen laite valitsee käyttämättömän PAN-tunnisteen ja lähettää levitysviestillä Beacon-kehyksen muille laitteille. Kun toinen laite saa Beacon-kehyksen, se voi pyytää liittymistä Beaconin lähettäneen laitteen verkkoon. Mikäli tämä sallitaan, lisää koordinaattori uuden laitteen omaan neighbour-listaan. Uusi liitetty laite lisää koordinaattorin omaan vieraslistaansa isäntänä sekä alkaa lähettää Beacon-kehyksiä, jotta mahdolliset uudet laitteet voivat liittyä tähän laitteeseen. Koordinaattori voi tarvittaessa ehdottaa laitteelle reitittimen roolia, mikäli laite ei ala lähettää Beacon-kehyksiä.

Kyseessä on yhdellä tai useammalla reitittävällä FFD-laitteella toteutettu laajennettu tähtiverkko, joka multi-hop muotoa, koska laitteen ja koordinaattorin välillä voi olla useampia reittejä. Kuvassa 11 oikealla on puuverkko.

### 3.4.3 Mesh-verkko

Kun puuverkon FFD-laitteiden annetaan kommunikoida suoraan keskenään, saadaan mesh-verkko. RFD-laitteiden täytyy edelleen kommunikoida koordinaattorin tai reititystä tekevän FFD-laitteen kanssa. Tätä verkkoa voidaan kutsua multi-hop multi-access verkoksi. Kuvassa 12 on mesh-verkko. [5]



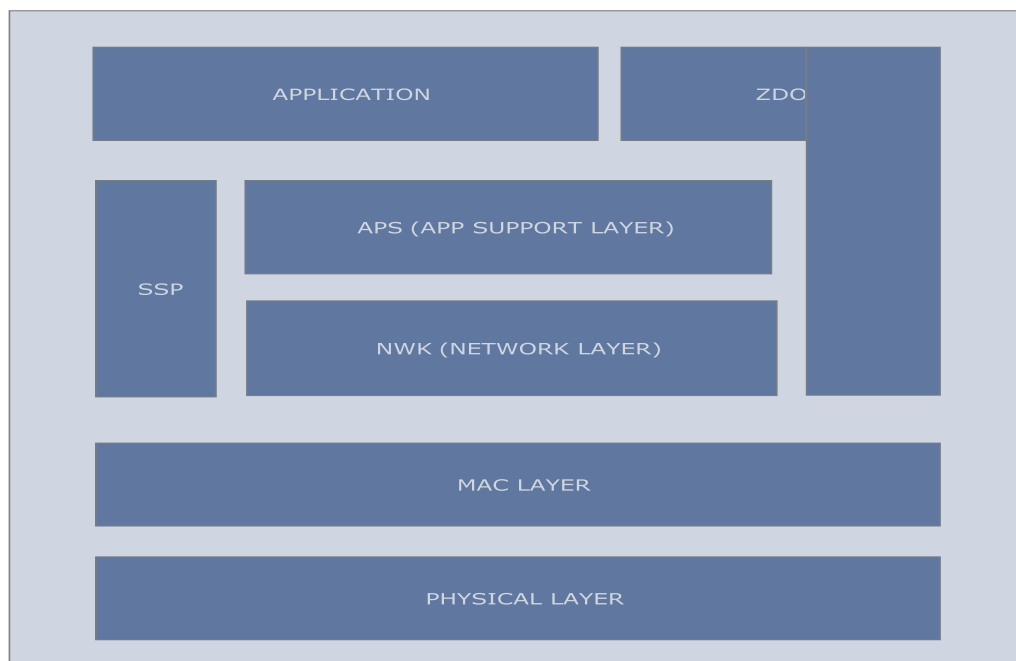
Kuva 12: Mesh-verkko

### 3.5 Zigbee-verkkoon pääsy

Laitteille on olemassa Zigbee-verkoissa kaksi kommunikointitapaa, Beacon ja Non-Beacon. NonBeacon-verkossa laite voi halutessaan lähettää, kunhan mikään muu laite ei lähetä samaan aikaan. Beacon-verkossa laite voi lähettää vain ennalta määrätyn aikajakson aikana. Beacon-verkon koordinaattori muodostaa säännöllisin väliajoin superkehysten, jota kutsutaan Beacon-kehykseksi. Jokaisen laitteen on synkronoiduttava tähän Beacon-kehukseen, jonka perusteella laite saa oman kommunikointiaikajaksonsa. Tämän aikajakson aikana laite voi lähettää tai vastaanottaa tietoa. Usein laite synkronoi toimintansa koordinaattorin Beacon-kehysiin ja herää juuri ennen kuin Beacon muodostetaan. Oman aikavälinsä kuluessa laite pyrkii tekemään tehtävänsä kommunikoinnin suhteen, palaa nukkumaan eli odottamaan seuraavaa aikaväliä. Niin sanotussa superkehyksessä voi olla aikajakso, joka ei ole minkään laitteen käytössä. Tämän aikavälin aikana noudatetaan CSMA-CA tekniikkaa ja mikä tahansa laite voi kommunikoida. [4, s. 29-42]

### 3.6 Zigbee-protokollapino

IEEE-802.15.4 määrittää OSI-mallin kaksi alinta kerrosta Zigbee-protokollapinosta, eli kerrokset PHY ja MAC. Ylempien kerroksien määrittelyt ovat Zigbee Alliancen tekemiä. [6, s. 13]



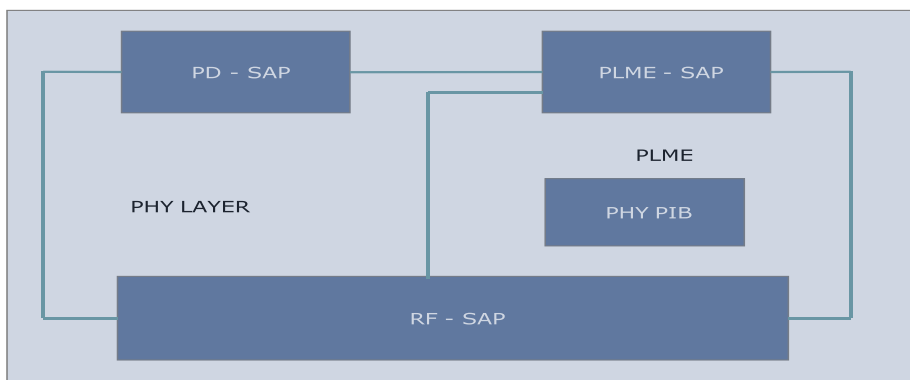
Kuva 13: Zigbee protokollapino

### 3.6.1 Zigbee PHY

PHY-kerroksen tehtävät on lueteltu seuraavassa:

- tiedon lähetys ja vastaanotto
- CCA ja CSMA-CA
- radio-osan aktivointi ja deaktivointi
- ED, kanavakohtainen voimakkuuden tunnistaminen
- kanavataajuuden valinta
- LQI, kanavan laadun tunnistaminen

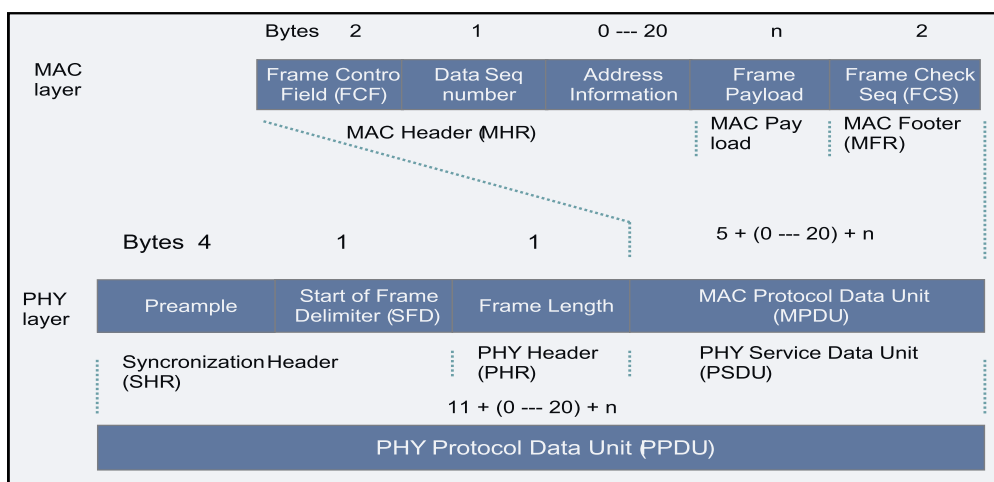
PHYn tehtävä on tarjota liityntä MAC-kerroksen ja fyysisen radiokanavan väliin. Tätä varten PHY:ssä on määritelty sekä data-, että hallintatoiminnot, jotka on liitetty PLME-hen. PHY-datapalvelu mahdollistaa PHY:n PPDU:ien lähettämisen radiokanavan yli. RF-SAP on toteutettu radio-osan elektroniikalla ja sitä ohjaavalla ohjelmakoodilla. PHY-hallintatoiminnot on toteutettu PD-SAP- ja PLME-SAP-osissa. Näiden kahden osan toiminnoilla MAC-kerros saa PHY-kerroksen lähetys- ja vastaanottopalvelut käyttöönsä. PHY-osaan kuuluu lisäksi PHY-MIB, joka sisältää PHYn hallintaobjekteja ja niitä pidetään yllä PLME-osassa. [4, s. 9-22]



Kuva 14: PHY-kerroksen komponentit

Standardi määrittää kaksi taajuuskaistaa. 250 kbit/s 2.4 GHz, 40 kbit/s 915 MHz ja 20 kbit/s 868 MHz. Molemmat kaistat käyttävät suorasekvenssiirtoa. Taajuusalueella 868–868.6 MHz on käytössä yksi kanava, alueella 902.0–928.0 MHz kymmenen kanavaa ja alueella 2.4–2.4835 GHz kuusitoista kanavaa. Useampi kanava eri taajuusalueilla helpottaa kanava-allokointia. Standardiin on määritelty myös dynaaminen kanavan valinta, koska laitteen tulee voida etsiä Beacon-kehystä eri kanavilta. Dynaamiseen kanavan valintaan sisältyy myös kanavan laadun ja lähetysoimakkuuden tulkinta. [4, s. 27-29]

Zigbee-protokollapinon kahden alimman kerroksen enkapsulointi ja PHY-kehysrakennne näkyy kuvassa 15.



Kuva 15: Zigbee-kehys

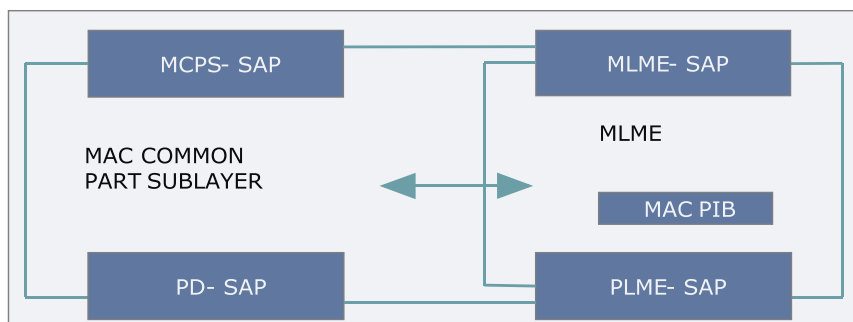
Kehys alkaa viiden tavun tahdistusbittikuviolla, jonka avulla vastaanottimet lukittuvat oikealle vastaanottotaajuudelle. Viiden tavun neljä ensimmäistä tavua ovat varsinaiset synkronointitavut ja viides on kehyksen alkamista indikoiva tavu. Nämä viisi tavua muodostavat kehyksen synkronointiosan SHR. Synkronointiosan perään tulee kehyksen pituustavu PHR. Kehyksen dataosassa on MAC- kerros eli MPDU, jota kutsutaan myös nimellä PSDU. Koko kehyksen nimenä käytetään PPDU. [4, s. 13]

### 3.6.2 Zigbee MAC

Zigbee MAC on vastuussa kaikista toiminnoista, jotka liittyvät fyysiseen radiokanavaan ja siirtoon. Seuraavassa listassa on MAC-kerroksen hoitamat tehtävät:

- tarjota luotettava linkki kahden laitteen välille
- PAN assosiointi ja disassosiointi
- Beacon-kehyksen generointi ja synkronointi
- GTS (Guaranteed Time Slot) mekanismista huolehtiminen

Kuten PHY, myös MAC sisältää hallinta-, että datatoiminnot. Kuvassa 16 on MAC kerroksen rakenne.



Kuva 16: MAC-kerroksen komponentit

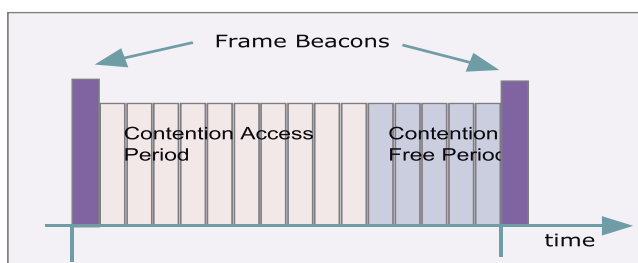
MAC-datatoimintoja käytetään MCPS-SAP-palvelujen kautta, kun taas hallintapalvelut on toteutettu MLME-osassa. PD-SAP ja PLME-SAP hoitavat data ja hallintatoiminnot MAC- ja PHY-kerroksien välillä. Verrattuna PHY-kerrokseen, on MAC-kerroksessa yksi liityntä lisää. Tämä liityntä mahdollistaa MLME:lle MAC-datapalvelujen käytön [4, s.23-25]

Superframe-kehyksellä tarkoitetaan kahden peräkkäisen Beaconin välistä aikaa. Superframe-kehys on jaettu 16 yhtä suureen aikaväliin ja Beacon lähetetään aina ensimmäisessä aikavälissä. Superframe kehyksen rakenteen määrittää ja lähettää koordinaattori. Halutessaan koordinaattori ei ole pakko lähettää superframe-kehystä. Beaconin tarkoitus on identifioida verkko, tahdistaa verkkoon kuuluvat laitteet ja määrittää super-

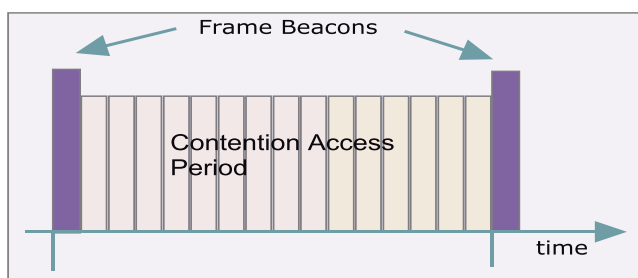
frame- kehyksien rakenne. Superframe-kehys voi sisältää sekä passiivisen, että aktiivisen osan.

Mikäli passiivinen osa on mukana, tämän aikana koordinaattori ei kommunikoi verkossa ja on mahdollisesti virransäästötilassa. Kehyksen aktiivinen osa jakautuu osiin CAP ja CFP. CAP-osan aikana mikä tahansa laite voi yrittää kommunikoida slotted CSMA-CA-menetelmällä. CFP-osa sisältää GTS-aikajaksoja, jotka on sijoitettu aina superframe-kehysten loppuun välittömästi CAP-osan jälkeen. Kuvassa 17 on esitetty Superframe-kehys, jossa on CAP- ja CFP-osat mukana. Kuvassa 18 on esitetty Superframe-kehys, jossa on vain CAP-osa. [4, s.29-41]

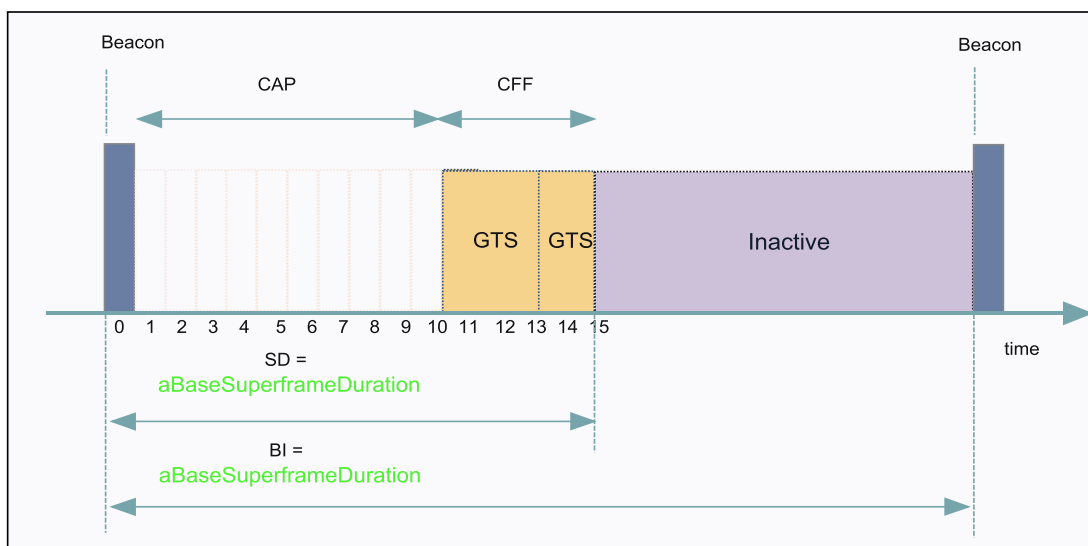
GTS aikajaksot on tarkoitettu sovelluksille, jotka tarvitsevat aikakriittisempää tiedonsiirtoa. Koska superframe-kehys lähetetään säännöllisin väliajoin, voidaan reaaliaika-sovelluksille taata lähetyksen mahdollisuus jokaisessa superframe-jaksossa. Koordinaattori voi allokoita enintään seitsemän GTS-jaksoa jokaiseen superframe-kehykseen. Koska GTS-aikajaksot on sijoitettu CAP-osan perään, on kaikki CAP-aktiiviteetti loputtava ennen CFP-osan alkua ja CFP-osa on loputtava ennen uuden Beaconin alkamista. Superframe-kehyksestä voidaan jättää CFP-osa kokonaan pois, jolloin kehys vie kahden Beaconin välisen ajan. Superframe-kehykseen voidaan luoda myös kuollut osa, joka sijoittuu CFP-osan perään ennen seuraavaa Beaconia. Kuollut osa järjestetään kasvattamalla kahden Beaconin välistä aikaa. Kuvassa 19 on esitetty superframe-kehys, jossa esiintyy dead zone-osa. Zigbee-verkko voi toimia myös ilman säännöllisiä Beaconeita, jolloin kommunikointi ei perustu allokoituihin aikaväleihin eikä superframe-kehysiin. [4, s.29-41]



Kuva 17: Superframe-kehysten rakenne, jossa CAP- ja CFP-osat

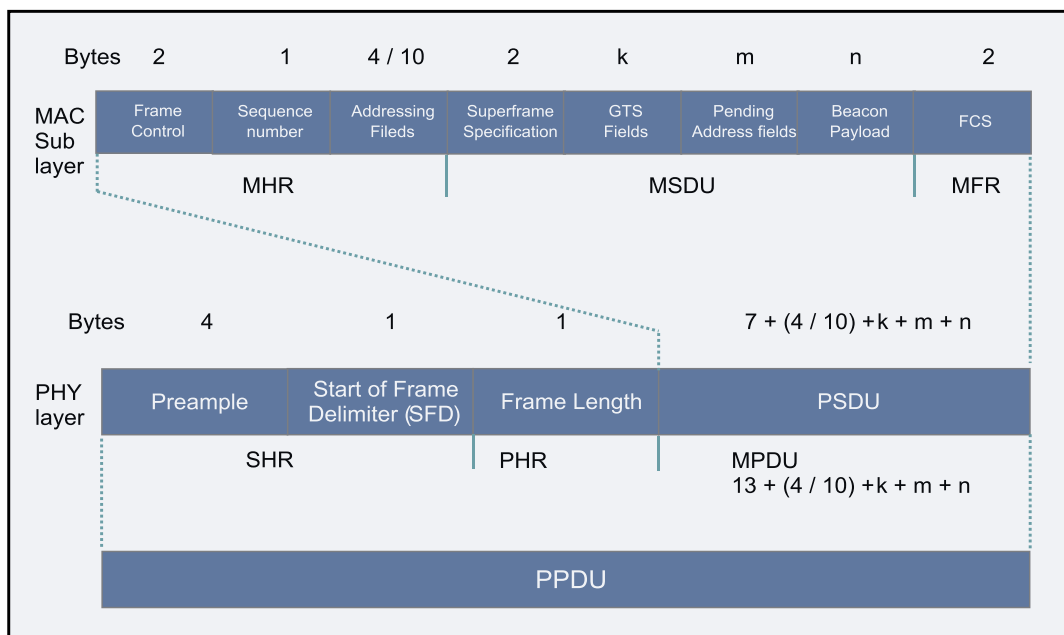


Kuva 18: Superframe-kehys, jossa vain CAP-osa mukana.



Kuva 19: Superframe-kehys, jossa dead zone mukana.

Alla on esitetty Beacon-kehysten rakenne.



Kuva 20: Beacon-kehys

Siirtoyhteyskehys alkaa neljän tavun tahdistuskuvilla, jota seuraa yhden tavun mittainen erotintavu. Kehyksen pituus tavu kertoo kehysten pituuden tavuina. Kaksi ensimmäistä kenttää muodostavat yhdessä SHR-osan ja kehysten pituus PHR-osan. Näiden



kenttien jälkeen tulee siirtoyhteyskerroksen kehysen PSDU-osa, joka voi olla normaali Zigbeeen kehysrakenne tai Beacon-kehys. Molemmat alkavat kahden tavun mittaisella Frame Control-osalla. Alla on esitetty kentän bittien merkitykset: [4, s.32-40]

#### Frame Control:

....	....	....	xxx.	=	Frame Type-kehys tyyppi. Beacon 000, Data 000 Ack 010, Command 011
....	....	....	x...	=	Security Enabled, salaus sallittu. ei 0, sallittu 1
....	....	...x	....	=	Frame Pending, lisää kehyksiä. Ei 0, kyllä data 1
....	....	..x.	....	=	Ack required, kuittaus vaaditaan. Ei 0, kyllä 1
....	....	..x..	....	=	Intra PAN, PAN sisäinen. PAN ulkoinen 0, PAN sisäinen 1
....	..xx	x...	....	=	Reserved
....	xx..	....	....	=	Dest Address Mode, kohdeosoitteen rakenne. PAN identifier not present 0, present 1, address field not present 0, present 1
..xx	....	....	....	=	Reserved
xx..	....	....	....	=	Source Address Mode) lähdeosoitteen rakenne. PAN identifier and address field not present 00, Address field contains 16-bit address 10, address field contains 64-bit address 11

Tämän jälkeen kehyksessä seuraa yhden tavun mittainen juokseva kehysnumero Sequence Number. Osoitekenttiin sisältyy PAN-tunniste (Source PAN Identifier) kaksi tavua ja lähettäjän osoite, joka voi olla kaksi tai 8 tavua Beacon-kehyksessä ja 0/20 tavua muissa kehyksissä. Tämän jälkeen seuraa MacPayload-osa. Muissa kuin Beacon-kehyksessä tämä sisältää NWK (Network)-kerroksen dataa. Beacon-kehyksessä MacPayload-osa alkaa kahden tavun mittaisella Superframe Specification-osuudella. Seuraavassa on esitetty tämän osan bittien merkitykset. [4, s.32-40]

#### Superframe Specification:

....	....	....	xxxx	=	Beacon Order 0x000F
....	....	xxxx	....	=	Superframe Order 0x00F0
....	xxxx	....	....	=	Final CAP Slot 0x0F00
...x	....	....	....	=	Battety Life Extension
..x.	....	....	....	=	Reserved
..x..	....	....	....	=	PAN Coordinator. lähettäjä on PAN-koordinaattori
x...	....	....	....	=	Assosiation Permit: (Coordinator Accepts Assosiation requests) koordinaattori hyväksyy varauskyselyt

Yhden tavun mittainen GTS Specification (k) sisältää:

## GTS Specification:

.... .xxx = GTS Description Count. GTS laskuri  
 .xxx x... = Reserved  
 x... .... = GTS Permit. koordinaattori ei hyväksy GTS varauksia 0, hyväksyy 1

## Pendins Address (m) yhden tavun kenttä sisältää:

.... .xxx = Number of short addresses pending  
 .... x... = Reserved  
 .xxx .... = Number of extended addresses pending  
 x... .... = Reserved

Beacon Payload (n) sisältää ylemmän kerroksen dataa. Tyypillisesti sisältönä on verkkokerroksen yhteyskäytäntö eli Zigbee NWK. Kentän rakenne määräytyy NWK mukaisesti. Kentän rakenne on seuraavassa. [4, s.32-40]

## Zigbee NWK:

protocol ID = 0x00 = Zigbee NWK  
  
 .... .... .... xxxx = Stack Profile  
 .... .... xxxx .... = nwkcProtocolVersion  
 .... ..xx .... .... = Reserved  
 .... .x... .... .... = Router Capacity: False 0, true 1  
 .xxxx... .... .... = Device Depth  
 x... .... .... .... = End Device Capacity: False 0, true 1

Kehyksen päättää FCS eli tarkistussumma. [4]

## 4 Zigbee radion integrointi osaksi digitaalista tuotetta

### 4.1 Yleistä

Zigbee-liitynnän toteuttamiseen laitteissa on olemassa kaksi vaihtoehtoa. Ensimmäisessä ratkaisutavassa liityntä integroidaan vahvasti muun laitteiston osaksi elektroniikan suunnitteluvaiheessa. Näin ollen Zigbee-liityntä näkyy järjestelmän keskusyksikölle ohjelmoitavana I/O-piirinä muisti- tai I/O-avaruudessa, johon sisältyy useita luettavia ja kirjoitettavia rekistereitä. Laitteen keskusyksikön suorittama ohjelma sisältää ohjelmakoodin Zigbee-liitynnän alustamiseen sekä lähetys- ja vastaanottotoimintoihin. Tarjolla on sekä maksullisia, että vapaan lähdekoodin tiedonsiirtopinoja, joilla on toteutettu Zigbee-liitynnän ohjelmallinen toiminnallisuus. Mikäli Zigbee-liityntää ei sijoiteta suoraan keskusyksikön muisti- tai I/O-avaruuteen, voidaan kommunikointi hoitaa jollakin yleisesti tuetulla liitynnällä, joka löytyy lähes kaikista mikro-ohjaimista. Yleisin ratkaisu on SPI-liityntä. Siinä mikro-ohjaimen ja Zigbee-liitynnän välinen kommunikointi tapahtuu sarjamoitaisesti. SPI-liityntää voidaan käyttää hyvin lyhyillä matkoilla laitteen piirilevyllä. Liitynnässä käytetään +5V tai +3.3V jännitetasoa ja enintään 8Mbit/s lähetys- ja vastaanottonopeutta. [7][8]

Toisessa ratkaisutavassa Zigbee-liitynnän toteutus on erotettu muun laitteiston elektroniikasta. Tässä tapauksessa Zigbee-liityntä liittyy muuhun järjestelmään niin ikään jonkun yleisen liityntästandardin mukaisesti, mutta Zigbee-liityntä ei ole kiinteästi osa laitteen elektroniikkaa. Toisin sanoen, Zigbee-liityntä voidaan erottaa laitteistosta laitteiston muuta toiminnallisuutta häiritsemättä. Tyypillisesti liityntänä käytetään USB-, UART- tai EIA/TIA 232-liityntöjä. Toinen merkittävä ero ensimmäiseen ratkaisuun on se, että liitynnän käyttöön tarvittava ohjelmisto voi olla osa laitteen koko ohjelmistoa tai sitten Zigbee-liityntä sisältää ohjelmiston. Jälkimmäisessä vaihtoehdossa Zigbee-liityntä näkyy muulle järjestelmälle modulina, jota ohjataan komennoilla, jotka moduli toteuttaa. Modulin sisältämän ohjelmiston tekijä on päättänyt, mitä toimintoja moduli tukee ja millaisia komentoja käyttäen tuettuja toimintoja ohjataan. Moduliratkaisusta käytetään myös nimitystä Zigbee-modemi, koska se näkyy muulle laitteistolle komennoilla ohjattavana modemina. PC-mikroiin USB-liitynnällä olevat Zigbee tikut ovat hyvä esimerkki ratkaisusta, jossa liitynnän ohjelmistosta on osa laitteiston käyttöjärjestelmässä ja osa itse modulissa. [4]

### 4.2 Zigbee-piiriratkaisut

Zigbee-transceiver piirejä ja moduleita on markkinoilla muutamilta valmistajilta. Alla on mainittu muutamia piiri- ja moduli tuotteita.

Atmel AT86RF230

- IEEE 802.15.4 tranceiver
- SPI-väyläliityntä

Texas Instruments / Chipcon CC2420

- IEEE 802.15.4 tranceiver
- SPI-väyläliityntä

Zigbee-moduleita:

Freescale / MaxStream XBee

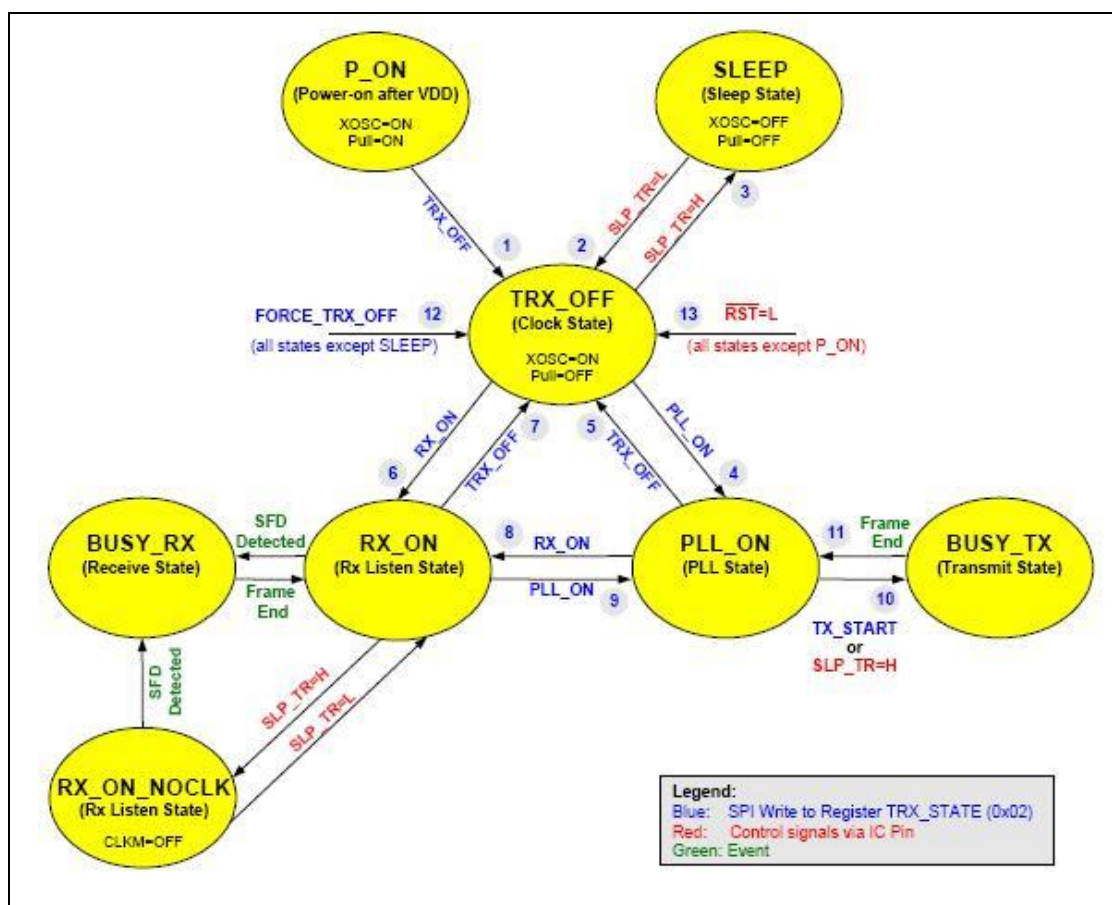
- EIA/TIA 232-liityntä
- Freescale MC13193 tranceiver
- Freescale M908GT60 mikro-ohjain

Lisäksi ainakin Silicon Laboratories, Freescale ja Atmel tekevät kehityskortteja helpottamaan Zigbee-tekniikan käyttöönottoa ja tuotekehitystä. [4]

## **4.3 AT86RF230 piirin tilat**

### **4.3.1 Perustilat**

Kuvassa 22 näkyy AT86RF230 piirin perustilat. Piirin sisäistä tilaa voidaan ohjata kahdella ulkoisella signaalilla ja SPI-väylän kautta ohjelmallisesti. Piirissä on kaksi rekisteriä TRX\_STATE ja TRX\_STATUS. SPI-väylän kautta kirjoitetaan rekisteriin TRX\_STATE ja tämän jälkeen kyseisen toimenpiteen onnistuminen tai epäonnistuminen voidaan tarkistaa lukemalla TRX\_STATUS rekisteri. [4][8]



Kuva 22. AT86RF230 perustilojen tilakaavio

Ulkoisella yläaktiivisella signaalilla SLP\_TR piiri saadaan SLEEP-tilaan ja vastaavasti tuomalle signaali alas, piiri siirtyy TRX\_OFF-tilaan. Muulla tavalla piiriä ei saada SLEEP-tilaan tai pois SLEEP-tilasta. Piiri saadaan pakotettua TRX\_OFF-tilaan ohjaamalla signaali RST alatilaa, mikäli piiri ei ole P\_ON-tilassa. Tällöin piirin konfigurointirekisterit ja SPI-liityntä alustetaan reset-arvoihin. Kaikki muut ohjaustoimenpiteet ovat ohjelmallisia. [4][8]

Sähkökatkon jälkeen piirin oskillaattori käynnistyy ja 128us päästä kellosignaali näkyy CLKM-lähdössä. Viiveen aikana sisäinen oskillaattori ehtii stabiloitua. Piiri on tilassa P\_ON. Ohjaavan mikro-ohjaimen tehtävänä on sähkökatkon jälkeen ohjata tulo SLP\_TR alas ja nostaa pienen viiveen jälkeen RST tulo ylös ennen kuin piiri voidaan ohjata TRX\_OFF-tilaan TRX\_OFF-komennolla. Vain tällä komennolla piiri saadaan TRX\_OFF-tilaan P\_ON-tilasta. Koska ohjaavaan mikro-ohjaimen digitaali I/O-signaalit ovat sähkökatkon jälkeen määrittelemättömässä tilassa jonkin aikaa, on P\_ON-tilassa piirissä kaikissa SPI-väylän signaaleissa sisäiset ylösvetovastuksen käytössä, kuten myös RST- ja SLP\_TR-tuloissa. Tällä pyritään estämään mikro-ohjaimen tahaton tilaohjaus välittömästi sähkökatkon jälkeen. [4][8]

Kuten tilakaaviosta selviää, TRX\_OFF-tilassa piirin digitaaliosa, oskillaattori ja sisäinen 1.8V regulaattori ovat toiminnassa, mutta sisäiset ylösvetovastukset ovat pois käytöstä. Piirin siirtyessä TRX\_OFF-tilasta PLL\_ON-tilaan, analoginen jänniteregulaattori aktivoidaan. Piiri aktivoi myös IRQ-lähdön, josta ohjaava mikro-ohjain tietää piirin siirtyneen PLL\_ON-tilaan ja PLL on käynnistynyt sekä lukittunut. PLL\_ON-tilan kautta päästään varsinaisiin lähetys- ja vastaanotto-tiloihin. PLL\_ON-tilassa komento RX\_ON siirtää piirin RX\_ON-tilaan. Kun tämä tila on saavutettu, analogia- ja digitaalivastaanottimet sekä taajuussyntetisoija käynnistyvät. Piiri on RxListen-moodissa. Kun saapuva kehys tunnistetaan, digitaalivastaanotto piirissä aktivoituu ja piiri siirtyy BUSY\_RX-tilaan. Jos ohjaavalle mikro-ohjaimelle on vedetty CLKM-signaali kello-signaaliksi, aktivoituu se sekä RX\_ON, että BUSY\_RX-moodeissa. Perustiloissa on oletuksena, että CLKM signaali on vedetty ohjaavalle mikro-ohjaimelle. Ajatuksena on, että kun Zigbee-aktiiviteettia ei ole, voi mikro-ohjain mennä virransäästötilaan. Jos CLKM-signaali on mikro-ohjaimen kellolähde, niin RX\_ON\_NOCLK-tila on hyödyllinen. Piiri menee RX\_ON\_CLOCK-tilaan, mikäli SLP\_TR-signaali menee ylös samaan aikaan kun tila on RX\_ON. Mikro-ohjaimella on aikaa 35 kellojaksoa toimia ennen kuin se menee virransäästötilaan. Piirin ollessa RX\_ON\_NOCLK-tilassa, tuleva kehys aiheuttaa RX\_START-keskeytyksen mikro-ohjaimelle. Tämän jälkeen piiri käynnistää kello-signaalin CLKM-lähtöön ja vaihtaa tilaksi BUSY\_RX. Keskeytyksen tulee herättää mikro-ohjain, mikäli se oli virransäästötilassa. Kun kehys on otettu vastaan, antaa piiri TRX\_END-keskeytyksen ja piiri siirtyy RX\_ON-tilaan. Jotta piiri voisi mennä uudelleen RX\_ON\_NOCLK-tilaan, on SLP\_TR-signaalin käytävä alhaalla. [8]

PLL\_ON-tilan kautta päästään siis sekä lähettämään, että vastaanottamaan kehyksiä kunhan piiri on TRX\_OFF-tilassa. PLL\_ON-tilassa signaalin SLP\_TR nostaminen ylös tai komennon TX\_START antaminen muuttaa piirin tilaksi BUSY\_TX ja kehyksen lähetys käynnistyy. Kun piiri on saanut kehyksen lähetettyä, vaihtaa se tilakseen takaisin PLL\_ON. Jos signaalia SLP\_TX käytetään kehyksen lähetyksen liipaisuun, on se käytettävä alhaalla välillä, jotta uusi kehylähetys olisi mahdollinen. [4][8]

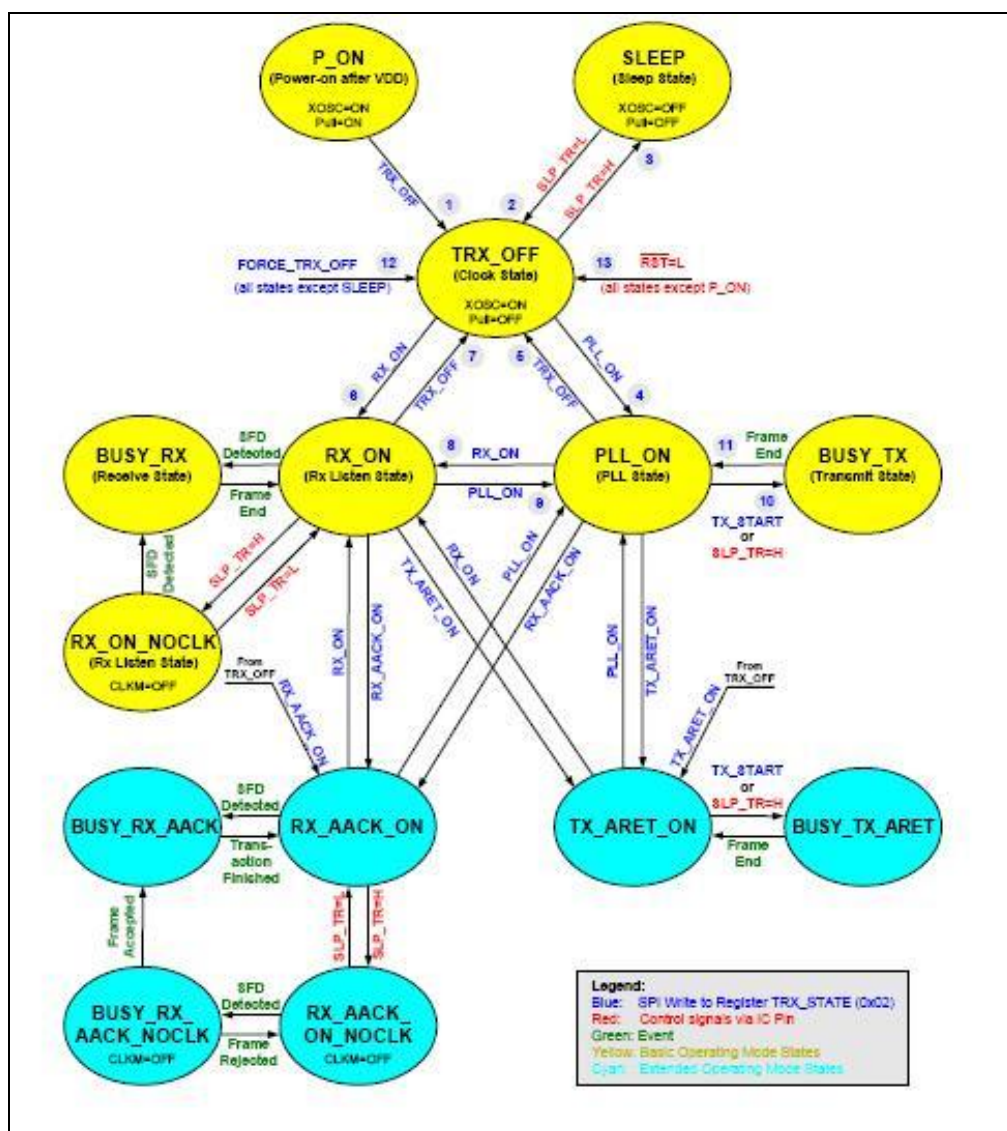
FORCE\_TRX\_OFF-komento keskeyttää kehyksen lähettämisen tai vastaanottamisen, mikäli piiri on tilassa BUSY\_RX tai BUSY\_TX ja pakottaa piirin TRX\_OFF-tilaan. Sen sijaan TRX\_OFF-komento ei keskeytä kehyksen lähetys tai vastaanotto-tilaa, vaan kyseinen operaatio suoritetaan loppuun ja sen jälkeen TRX\_OFF-komento toteutetaan ja tilaksi vaihtuu TRX\_OFF. Tämä näkyy tilakaaviossa siten, että BUSY\_RX- ja BUSY\_TX-tiloista ei ole polkua tilaan TRX\_OFF, vaan meneillään oleva operaatio on saatettava loppuun, jotta tilaksi vaihtuu RX\_ON tai PLL\_ON ja TRX\_OFF komento on mahdollinen. [8]

### 4.3.2 Laajennetut tilat

Kuvassa 23 näkyvät laajennetut tilat sinisellä värillä. Tiloihin TX\_ARET\_ON ja RX\_AACK\_ON päästää suoraan tiloista RX\_ON ja PLL\_ON, mutta myös tilasta TRX\_OFF SPI-komennoilla. Näistä kahdesta tilasta päästään perustiloihin RX\_ON ja

PLL\_ON niin ikään SPI-komennoilla kuten tilakaaviosta käy ilmi. TX\_ARET\_ON-tilassa piiri yrittää lähettää uudelleen kehystä. Piiri antaa uudelleenlähetyksestä keskeytyskoodin, joka kertoo lähetyksen onnistumisen tai syyn epäonnistumiseen.

RX\_AACK\_ON-tilassa piiri vastaanottaa kehysten, tarkastaa tarkistussumman ja suorittaa osoitetarkistuksen. Mikäli kehys on tullut virheettömästi, osoite on oma ja kehukseen pyydetään kuittausta, lähettää piiri automaattisesti kuittauskehysten. Piiri antaa myös keskeytyksen, mikäli kehys läpäisi osoitesuodatuksen. Lähettäjä, joka saa automaattisen kuittauskehysten, toimii TX\_ARET\_ON-tilassa ja tarkistaa automaattisen kuittauskehysten tarkistussumman ja sekvenssinumeron, joka pitäisi täsmätä automaattikuittauksen aiheuttaneeseen kehukseen. [8]



Kuva 23: Laajennetut tilat

TX\_ARET\_ON- ja RX\_AACK\_ON-tilojen käyttö on samanlaista kuten perustilojenkin. TRX\_STATE- ja TRX\_STATUS- rekisterit pitävät yllä tilatiedot myös näissä tiloissa. Tilakaaviosta selviää laajennettujen tilojen väliset tilamuutostekijät. Tilassa RX\_AACK\_ON piiri hylkää kehykset, joissa tarkistussumma tai osoite ei täsmää. Jos kehys hyväksytään vastaanotossa, TRX\_END-keskeytystieto annetaan mikro-ohjaimelle, joka siirtää kehyksen omaan muistiinsa. Jos vastaanotetussa kehyksessä pyydetään kuittausta, piiri lähettää kuittauksen automaattisesti heti vastaanotetun kehyksen perään. [4][8]

Zigbee-laitteen mikro-ohjain kirjoittaa lähetettävän kehyksen datan piirin muistiin, joka CSMA-CA-algoritmin mukaisesti alkaa lähettää kehystä. Kehyksen lähetys on tilan TX\_ARET\_ON tehtävä. Lähetys liipaistaan nostamalla SLP\_TR ylös vähintään 1us ajaksi. Lähetys voidaan käynnistää myös SPI-komennolla TX\_START, kuten tilakaaviosta selviää. Atmel suosittelee, että lähetettävän kehyksen data kirjoitetaan piirin kokonaisuudessaan, ennen kuin lähetys liipaistaan. Piiri hyväksyy myös kehyksen tahdistusbittien lähettämisen aloituksen samaan aikaan kun lähetettävää dataa kirjoitetaan piiriin. [4]

Kun piirin lähetys liipaistaan, piiri tunnistaa, onko lähetyksen kanava vapaa. Jos on, se alkaa lähettää. Jos ei ole, piiri suorittaa CSMA-CA back-off-algoritmin mukaisesti odotuksen, jonka jälkeen yritetään uudestaan. Tämä jatkuu kunnes kehys saadaan lähetettyä tai maksimiuusintalähetykertojen määrä tulee täyteen. Tässä tapauksessa piiri antaa keskeytyksen ja TRX\_STATUS-rekisterin bitti CHANNEL\_ACCESS\_FAILURE asetetaan. Kun piiri lähettää TRX\_ARET\_ON-tilassa, se tutkii, onko kuittaus lähetettävään kehykseen vaadittu. Jos on, piiri vaihtaa onnistuneen kehyslähetyksen jälkeen tilaa ja jää odottamaan kuittausta. Jos kuittausta ei tule määrättyssä ajassa, alkaa CSMA-CA-lähetys samalle kehykselle uudestaan. Tämä toistuu, kunnes kuittaus saadaan tai maksimilähetysmäärä tulee täyteen. Tässä tapauksessa generoidaan keskeytys tilakoodilla NO\_ACK. Jos kaikki menee hyvin, generoidaan keskeytys tilakoodilla SUCCESS. [4][8]

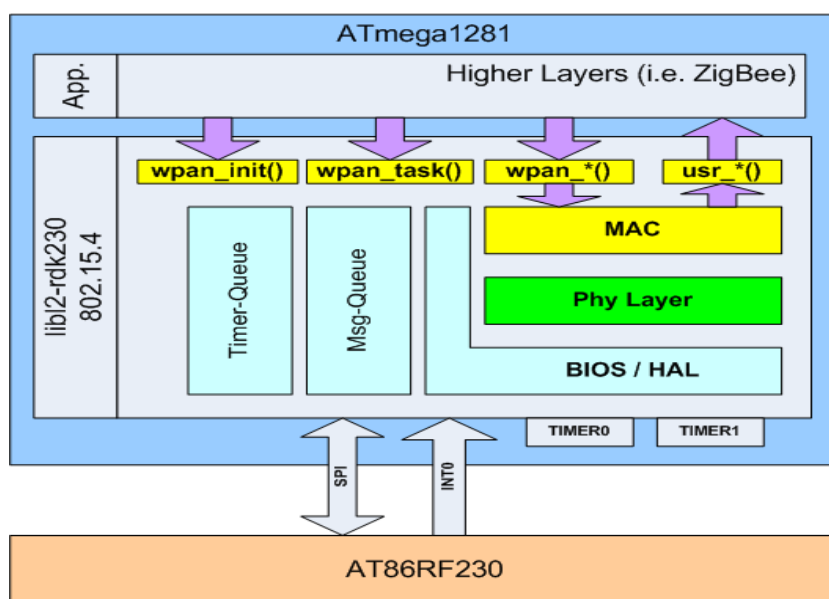
RX\_AACK\_NOCLK-tila on muuten samanlainen kuin perustila TX\_ON\_NOCLK, mutta RX\_AACK\_NOCLK tilassa piiri kuuntelee kehyksien Start-Of-Frame-Delimiter osaa ja kun se on tunnistettu, piiri siirtyy RX\_AACK\_NOCLK-tilaan ja alkaa ottaa kehystä vastaan. CLKM-signaalia ei tässä vaiheessa generoida, joten ohjaava mikro-ohjain jatkaa nukkumista, jos se oli nukkumassa. Jos kehyksen osoitekenttä täsmää, piiri siirtyy tilaan BUSY\_RX\_AACK, jossa CLKM-signaali aktivoidaan ja ohjaava mikro-ohjain pystyy lukemaan vastaanotettavan kehyksen. [8]

## 4.4 Atmel MAC

AT86RF230 piiri toteuttaa Zigbee-protokollapinon kaksi alinta kerrosta. NWK-kerroksesta ylöspäin on ohjelmallista toteutusta. Atmelilta löytyy Atmel MAC, joka osaa käyttää AT86RF230 piirillä toteutettua 802.15.4 liityntää ja toteuttaa Zigbee-protoko-



kollapinon MAC-kerroksen. Kuvassa 24 on Atmel MAC:in periaatteellinen toiminta-kaavio. [9]



Kuva 24. Atmel MAC periaatekaavio

Atmel MAC on toteutettu kirjastolla libl2\_rdk230 ja sillä voidaan toteuttaa FFD-laite. Kirjastossa toteutettuja API-kutsuja käytetään normaaliin tapaan osana ohjelmistoprojektia [9]. Periaatekaaviosta selviää, että Atmel MAC kommunikoi SPI-väylällä AT86RF230 piirin kanssa ja AT86RF230 antaa keskeytyksen mikro-ohjaimen INT0-tuloon. Näin AT86RF230 on kytketty myös testikorteilla. BIOS/HAL kaaviossa ilmestää Atmel MACin koodia, joka toteuttaa SPI-kommunikaation. PHY-osan toteutus sisältää PLME- kutsuja, joilla toteutetaan muun muassa CCA- (Clear Channel Assessment) ja ED- (Energy Detection) toiminnat sekä PLME SET- ja PLME GET-pyynnöt. [4][8].

Atmel MAC sisältämä Timer-Queue sisältää ajastimia, joita sovelluksen tekijä voi komennoilla käyttää. Ajastimia voi käynnistää ja pysäyttää. Jokaiseen ajastimeen on liitetty oma funktio, jolla ajastimien umpeutumista indikoidaan sovelluksen tekijälle. Msg-Queue sisältää kaikki viestit, jotka Atmel MAC saa. wpan\_task funktio käsittelee Msg-Queue jonossa olevat viestit, jonka vuoksi tätä funktiota tulisi kutsua niin usein sovelluksesta kuin mahdollista. Funktio palauttaa boolean arvon TRUE jokaisen käsitellyn viestin jälkeen. Jos funktio ei saa käsiteltävää viestiä jonosta, se palauttaa boolean arvon FALSE. [10]

Atmel MAC sisältämät funktiot jaetaan kolmeen ryhmään. Yleiset, Request ja Callback funktiot. Yleisillä funktioilla alustetaan Atmel Mac ja toteutetaan MACin toiminnot. Myös ajastimia käytetään yleisillä funktioilla. Yleiset funktiot ovat:

- wpan\_init ja wpan\_task

- wpan\_start\_timer ja wpan\_stop\_timer sekä usr\_timer\_trigger

Atmel MAC alustetaan funktiolla wpan\_init. Tämä funktio alustaa Atmel MAC:in muuttujat ja sitä on kutsuttava ennen muiden funktioiden kutsumista. Funktiota wpan\_task on kutsuttava sovelluksesta mahdollisimman usein, koska se ylläpitää viestijonon käsittelyä. Atmel MAC:ssä on käytettävissä 256 ohjelmallista ajastinta numeroilla 0...256, joita käsitellään funktioilla wpan\_start\_timer ja wpan\_stop\_timer. Funktiota usr\_timer\_trigger Atmel MAC kutsuu silloin, kun jokin käynnistetty ajastin umpeutuu ja näin tieto välittyy sovellukseen. Sovelluksen tekijän on kirjoitettava tämän funktion toteutus. Valtaosa Request funktioista liittyy MLME-osan toteutukseen, joiden avulla sovelluksen tekijä voi lähettää MAC-viestejä. MCPS-osan toteutukseen liittyy funktiot wpan\_mcps\_data\_request ja wpan\_mcps\_purge\_request. Ensimmäinen funktio muodostaa dataa käsittelevän viestin ja sijoittaa sen viestijonoon. Jälkimmäinen funktio luo pyyntejä viestijonon viestien purkamista varten. Funktio wpan\_task purkaa jälkimmäisen funktion tekemien pyyntien perusteella viestijonoa round-robin menetelmällä. Atmel MAC sisältää lisäksi paljon wpan\_mlme-alkuisia viestejä, joilla hallitaan muun muassa yhteyden muodostusta, kanavien etsintä ja kytkeytymistä Zigbee-verkoon sekä muutetaan MAC PIB-attribuutteja. [4][10]

Atmel MAC:ssä ei ole toteutettu kaikkia MAC-kerrokseen määriteltyjä toimintoja. Beacon-verkko ei ole tuettu, mikä tarkoittaa, että säännöllisiin Beaconeihin tahdistettu tiedonsiirto ei ole mahdollinen, vaan siirtotien varaus perustuu vain CSMA-CA-menetelmään, johon on toteutettu backoff-toiminta. Näin ollen CAP-, CFP- ja GTS-tiedonsiirtoanalysointiin ei ole mahdollisuutta. RFD-laitetta ei ole toteutettu myöskään. Kolmas puuttuva ominaisuus on reititys puu- ja mesh-verkoissa. Laajennetussa tähti-verkossa yhden tai useamman FFD-laitteen tulisi kyetä reitittämään RDF-laitteiden kehyksiä PAN-koordinaattorille. Tämä tarkoittaa sitä, että tähtitopologinen Zigbee-verkko on toteutettavissa Atmel MAC:llä.

## 5 Zigbee prototyyppien rakentaminen ja verkkodemonstraatiot

### 5.1 Zigbee käytännössä

Zigbee-kommunikaation tutkimista varten on rakennettu kolme samanlaista korttia, joiden avulla saadaan testiverkko muodostettua. Testiverkko koostuu yhdestä coordinator- ja kahdesta device-laitteesta. Liikenteen analysointiin käytetään USB-liitynnällä olevaa Zigbee-tikkua ja Windows-ohjelmistoa. Liitteessä A on esitetty ensimmäisen kortin kytkentäkaavio. Kortti rakentuu Atmel Atmega1281 mikro-ohjaimen ympärille. Zigbee tranceiverinä käytetään Atmel AT86RF230 piiriä, joka toteuttaa standardin 802.15.4. Liitteessä A esitetyn mukainen piirikortti ei osoittautunut toimivaksi tiedonsiirron vastaanoton kannalta. Tämä johtui siitä, että balun-piiri oli jäänyt pois kytkennästä ja sen jälkikäteen lisääminen ei tuottanut toimivaa ratkaisua. Sekä tranceiver- että balun-piirien juottaminen piirilevyille vaatii ammattitason laitteet johtu-

en kotelointirakenteesta. Tämä on huomioitava myös oppilaitosympäristössä. Liitteessä B on toisen tehdyn kortin piirikaavio. Toisessa kytkennässä käytetään Atmelin tekemää Zigbee-minikorttia, jossa on tranceiver, balun ja antenni valmiina omalla piirilevyllä. Tämä levy liitetään pistoliittimellä liitteen B kytkentään. Toinen kortti osoittautui toimivaksi kaikilta osin. Kortilla on myös korjattu joitakin ensimmäisen kortin reititysvaiheissa syntyneitä virheitä. Liitteessä D on molempien tehtyjen korttien piirilevy layoutit.

### 5.1.1 ATMega1281

ATMega1281 mikro-ohjaimen sisältö on lyhyesti seuraava [7]

- AVR 8-bitin RISC-keskusyksikkö
- 128kT FLASH, 8kT SRAM ja 4kT EEPROM
- FLASH ohjelmointi SPI-väylän kautta
- JTAG, IEEE 1149.1-liityntä
- kaksi 8 bitin ja yksi 16 bitin ajastinta
- 2 UART-liityntää, analogia komparaattori sekä 10-bitin AD-muunnin
- SPI-väylä ja sisäinen kello-oskillaattori
- kuusi virransäästötilaa ja yleiskäyttöisiä I/O-linjoja

### 5.1.2 AT86RF230

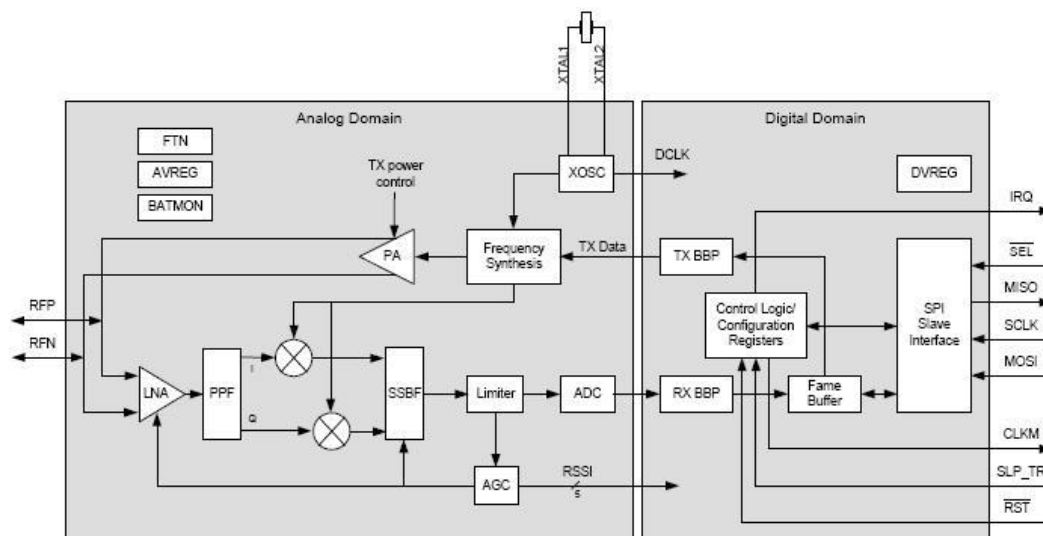
Atmel AT86RF230 piiri tarjoaa selkeän ja kompaktin ratkaisun 802.14.5-liitynnän toteutukseen. Piiri on edullinen ja hyvin saatavilla. Näiden syiden takia sitä käytetään tässä työssä. Piiri kommunikoi ympäristön kanssa SPI-väylällä, joten se on varsin helppo liitettävä ja ohjattava. SPI-väylän tiedonsiirto tapahtuu seuraavasti. Mikro-ohjain SPI-liityntä alustetaan master-tilaan ja Zigbee tranceiver toimii aina slave-tilassa. Tällöin mikro-ohjain syöttää SPI-väylän kellon, joka on signaali SCK kytkennässä. SPI-väylässä kello generoidaan vain silloin kun master-laite lähettää slave-laitteelle. Masterin lähetyssignaali on MOSI. Piirivalintesignaali on kytkennässä SS. Kun master-laite haluaa lukea tranceiveriltä tietoa, se lähettää tyypillisesti kaksi tavua. Ensimmäisessä tavussa master lähettää rekisterin, josta se haluaa lukea ja toinen tavun kirjoitus siirtää kyseisen rekisterin sisällön masterin SPI-tuloon, joka on kytkennässä signaali MISO. Toisen lähetettävän tavun sisällöllä ei ole merkitystä. SPI-väylän liityntä on osa tranceiverin digitaaliosaa. Kuvassa 21 on piirin sisäinen lohko-kaavio. Kytkennässä on lisäksi kytketty tranceiveriltä keskeytyslinja mikro-ohjaimelle, joka on signaali IRQ. Mikro-ohjaimen digitaalilähdöt PB4 ja PB5 on kytketty tranceiverin tuloihin SLP\_TR ja RSTN. Aktivoimalla tulo SLP\_TR, saadaan tranceiver SLEEP-tilaan, jossa virrankulutus on hyvin pientä, 0.1uA. Tulo on yläaktiivinen, joten kun tulo ohjataan alas mikro-ohjaimelta, tranceiver palaa normaaliin toimintatilaan. Kun ohjaimen lähtö PB5 pidetään alhaalla, pakotetaan tranceiver TRX\_OFF tilaan. CLKM on kello-signaali tranceiveriltä. Kun tranceiver saa käyttöjännitteet, käynnistyy piirin sisäinen oskillaattori.

tori. 125us oskillaattorin jälkeen käynnistymisen jälkeen, CLKM-lähtö antaa ulos piirin master clock-kellosignaalin. [8]

AT86RF230 tarvitsee ulkoisia komponentteja hyvin vähän toimiakseen. Piiri tarvitsee ulkoisen kiteen sisäistä oskillaattoria varten ja se kytketään piirin XTAL1 ja XTAL2-signaaleihin. Molemmista signaaleista on lisäksi kondensaattorit maahan, joilla pyritään varmistamaan sisäisen oskillaattorin käynnistyminen sähkökatkon jälkeen. Oskillaattorin lisäksi piirissä on vaihelukitulla silmukalla toteutettu kellogenerointi. [8]

Signaaleihin RFP ja RFN kytketään ulkoinen antenni. Antenniliitynnän lähtö on piirissä differentiaalinen. Differentiaaliseen lähtöön on kytkettävä balun, joka tekee impedanssisovituksen lähdon ja antennin välille. Antenni voi olla joko piirilevylle toteutettu symmetrinen dipoli tai Single Ended-tyyppinen. Dipolin ideaalinen impedanssi olisi mahdollisimman lähellä 73 ohmia. Käytännön piirilevylle toteutetun dipoliantennin impedanssi pyritään hakemaan lähelle 100 ohmia, koska piirin lähdon differentiaalinen impedanssi on 100 ohmia. Tällainen dipoliantenni voidaan toteuttaa piirilevylle. Tässä työssä käytetään piirilevylle toteutettua symmetristä dipoliantennia. Liitteissä A ja B on esitetty mikro-ohjaimen ja AT86RF230-piirin väliset kytkennät.[10]

Kuvassa 21 näkyy AT86RF230 piirin sisäinen lohkokaavio.



Kuva 21: AT86RF230 lohkokaavio

Kortille on lisäksi laitettu kaksi diagnostiikkaledyä, sekä EIA/TIA 232-puskuri tiedonsiirtoa varten. Käyttöjännitteiden tuottamiseen on käytetty normaalia sarjaregulointiin perustuvaa jännitteenalennuskyskentää.

## 5.2 Piirikaavio- ja levysuunnittelu sekä piirilevyvalmistus

Piirikaaviosuunnittelu on tehty Pads Power Logic 4.0 ohjelmalla ja piirilevysuunnittelu PADS Power PCB 5.0 ohjelmalla. Molemmat ovat Windows- ohjelmia. Suunnittelu etenee seuraavasti. Ensin suunnitellaan piirikaavio. Tämän jälkeen siirretään piirikaavioon tehty kytkentäinformaatio niin sanotun vetolistatiedoston avulla piirilevysuunnitteluohjelmaan. Tässä vaiheessa voidaan siirtää myös mahdolliset suunnittelusäännöt piirikaaviosta piirilevysuunnitteluvaiheeseen. Piirikaavioon piirretty kytkentäinformaatio näkyy piirilevysuunnittelussa komponenttien välisinä kumilankoina, jotka reititetään toisistaan irti. Kortissa käytetään kaksipuolista piirilevyä. Liitteessä A ja B on testikorttien kytkentäkaaviot. Liitteessä C on kuva molemmista testikorteista ja liitteessä D on esitetty molempien korttien molempien puolien reititykset.

Itse piirilevyt on valmistettu kaksipuolisesti lakatusta levystä positiivimenetelmällä. Piirilevysuunnitteluohjelmasta tulostetaan molempien puolien reititetyt kuvat muovikalvoille. Tämän jälkeen piirilevy valotetaan kalvojen välissä molemmilta puolilta UV-valolla. Valotusaika on lakan paksuudesta ja UV-lamppujen etäisyydestä riippuen minuutista neljään minuuttiin. Valotuksen jälkeen valottunut lakkaosuus kehitetään pois emäksisellä nesteellä, esimerkiksi lipeällä. Tässä vaiheessa molemmilta puolelta levyä on saatava lakka pois kokonaan niiltä osilta, joista kupari tulee poistumaan. Kehitysvaiheen jälkeen levy huuhdotaan vedellä. Seuraavaksi levy syövytetään natriumpersulfaattipohjaisessa liuoksessa, jonka jälkeen jäljelle jää piirilevysuunnitteluohjelmassa reititetyt alueet ja vedot. Käyttämättömällä liuoksella syövytys kestää alta kymmenen minuuttia, mikäli liuos on tehty oikeassa suhteessa. Syövytyksen jälkeen levy huuhdotaan runsaalla vedellä ja kuivataan sekä pyyhitään sinolilla, jotta jäljelle jääneiden kuparien päältä poistuu lakka. Levyn molemmille puolille on ruiskutettava kontaktiaine, jotta komponentit saadaan juotettua levyyn ja kupari ei ala hapettumaan. Levyyn on vielä porattava tarvittavat reiät ja juotettava läpiviennit. Tämän vaiheen jälkeen varsinainen piirilevy on valmis ja siihen voidaan kiinnittää komponentit.

## 5.3 PAN-coordinator ja device-sovellukset

Liitteiden jälkeen ovat PAN-coordinator- ja device-sovelluksien ohjelmakoodit. Sovellusta, jossa on mukana PAN-coordinator koodi, ajetaan yhdessä testikortissa. Tämän testikortin sovellus kokonaisuudessaan toteuttaa satunnaislukupalvelimen, jolta kaksi muuta testikorttia pyytää satunnaislukuja. PAN-coordinator muodostaa Zigbee-verkon, jonka jälkeen device-laitteet voivat liittyä uuteen verkkoon. Sekä PAN-coordinator, että device-laitteet ovat samalla FFD Atmel MAC-koodilla.

Atmel MAC on ladattavissa Atmelin sivuilta lähdekoodimuodossa. Paketti sisältää seuraavat kansiot:

Bios: mm. ajastimien ja viestijonon toteutusfunktiot.  
 Docs: Doxygen generoimat ohjetiedostot

Inc: Otsikkotiedostoja funktioita varten.  
 Mac: Funktiot Zigbee-liikenteen hallintaan ja ylläpitoon  
 Misc: usr-alkuiset primitiivifunktiot MACin toteuttamiseen  
 Phy: Funktiot AT86RF230-piirin käsittelemiseen

Edellä mainittujen kansioiden lisäksi paketista purkautuu Makefile, joka on lähdekoodien kääntämistä varten make-työkalulla. Makefile on valmis, joten sitä ei tarvitse Atmel MACin kääntämistä varten muokata. Kääntämistä varten tarvitaan järjestelmään AVR-ristikäntäjä. Tässä työssä käytetään ilmaista WinARV-kääntäjäohjelmistoa, joka sisältää gcc-kääntäjän. Samalla ohjelmistolla käännetään myös sovelluksien ohjelmat, joihin käännetty Atmel MAC on otettu mukaan. Atmega1281 flashrom ja eeprom saadaan ohjelmoitua CodeVision kääntäjällä.

## 5.4 Zigbee-liikenteen analysointi- ja testaussuunnitelma

Zigbee liikenteen testaussuunnitelma on esitetty alla [16].

1. Johdanto  
 Testauksessa analysoidaan Zigbee-verkon muodostumista ja tiedonsiirtoa kolmen laitteen testiverkolla.
2. Testauksen kohde ja tavoitteet  
 Testauksessa on tarkoitus saada näkyviin analysaattoria käyttäen Zigbee-verkon muodostumisen ja verkkoon liittymisen mekanismit sekä näihin vaiheisiin liittyvät kehykset. Tämän jälkeen analysoidaan Zigbee-laitteiden välistä liikennettä ja verrataan siirtoyhteyskehyksiä standardissa ilmoitettuihin kehyksiin. Lisäksi tavoitteena on saada käsitys Zigbee-verkon vasteajoista.
3. Testausympäristö  
 Testausympäristö on normaali huoneympäristö ilman signaalia vaimentavia tekijöitä. Laitteiden väliset etäisyydet ovat enintään 30m.
4. Testauksen organisointi  
 Testausta varten on tehty kolme Zigbee-liitynnällä olevaa testikorttia, joista yhdessä toimiva ohjelma toteuttaa Zigbee-coordinaattorin ja kaksi muuta toteuttavat asiakaslaitteet. Kaikissa kolmessa toimiva ohjelmisto toteuttaa Zigbee-tiedonsiirron kanalta FFD-laitteen. Testauksen suorittaa yksi henkilö.
5. Testausstrategia  
 Testauksessa analysoidaan Zigbee-liikennettä Flexipanel Zigbee-analysaattori-ohjelmistolla satunnaislukupalvelimen ja kahden asiakkaan välillä. Korttien ohjelmistaukset ovat liitteissä 4 ja 5. Flexipanel Zigbee-analysaattori käyttää moduuliratkaisulla toteutettua USB-Zigbee tikkua ja Windows-ohjelmistoa.

6. Testattavat toiminnot  
Testaus kattaa testikorttien ja Atmel MACin toiminnot.
7. Ei testattavat ominaisuudet  
Testauksessa ei testata varsinaisesti korttien ohjelmistoista niitä osia, jotka eivät liity Zigbee-liikenteeseen.

#### 5.4.1 PAN-koordinaattorin toiminta

Kun PAN-koordinaattori käynnistyy, on sen valittava kanava taajuuskaistasta, jolla se toimii. 2440 MHz kaistalla on 16 kappaletta 5 MHz kanavia alkaen taajuudesta 2405 MHz taajuuteen 2480 MHz. Näitä kanavia vastaa Atmel MACissä kanavanumerot 11–26. Koordinaattori alkaa lähettää levitysviestinä BeaconRequest kehystä alkaen ensimmäisestä kanavasta. Mikäli BeaconRequestiin saadaan vastauksena Beacon-kehys, tarkoittaa se, että kyseessä olevalla kanavalla on koordinaattori ja uusi koordinaattori lähettää seuraavalla kanavalle uuden BeaconRequest-kehysten. Tätä proseduuria jatketaan kunnes löytyy kanava, jolla ei ole koordinaattoria. Koordinaattori valitsee ensimmäisen vapaan kanavan, jolla se alkaa toimia. Mikäli vapaata kanavaa ei löydy, alkaa proseduuria alusta. Näin olleen jokaisella kanavalla voi toimia vain yksi koordinaattori. Kun vapaa kanava on löytynyt, muodostuu Zigbee-verkko tälle kanavalle. Testiohjelman koordinaattori toimii kiinteästi kanavalla 18 eli 2440MHz.

#### 5.4.2 Laitteen toiminta

Kun Zigbee-laite, joka ei ole koordinaattori, käynnistetään, suorittaa se ActiveScan-toiminnon kaikille kanaville 11-26. ActiveScan tilassa laite lähettää levitysviestinä BeaconRequest-kehystä. Laite vastaanottaa kaikilta kanavilta mahdollisesti saadut Beacon-kehysten informaatiot. Tämän jälkeen laite lähettää Association Request-kehysten sille kanavalle, josta se löysi oikean PAN ID:n. Tällä tavalla laite haluaa liittyä tiettyyn PAN-verkkoon. Mikäli PAN-koordinaattori hyväksyy liittymisen eli assosioinnin, se lähettää laitteelle osoitteen. Assosiointivaiheessa laite käyttää osoitteena EEPROM-muistin osoitteissa 0-7 olevaa osoitetta, joka on oltava kaikilla laitteilla yksilöllinen. Testiohjelmassa laite lähettää kehysiä vain kanavalle 18 eli 2440MHz. Laitteiden käynnistysjärjestyksellä ei ole merkitystä assosiointivaiheen kannalta. Beacon pohjaisessa-verkossa koordinaattori lähettää Beacon-kehystä säännöllisin väliajoin, jolloin laitteet voivat myös tällä tavalla saada informaatiota olemassa olevista verkoista, mutta tällä hetkellä Atmel MAC ei tue Beacon-pohjaista koordinaattoritoimintaa, vaan koordinaattori vastaa tuleviin BeaconRequest kehysiin lähettämällä Beaconin. Liitteessä E näkyy assosiointivaiheen kehukset.

Assosiointivaiheen aloittaa ei koordinaattorilaite BeaconRequest kehyksellä. Kehyksen tyyppi on komento. DstPAN ja DstAdd ovat FFFF. FrameControl-osasta ilmenee, että

salausta ei käytetä hyötyosassa, lisää kehyksiä ei kuulu lähetykseen, kuittausta ei odoteta ja sekvenssinumero on 9C. Punaisella näkyy komennon tyyppi. RSSI on signaalin vahvuutta ilmentävä arvo, joka tässä on -43dBm. Linkin laatua kertova indikaattoriluku on 97%. RSSI ja LQI saadaan liittynän PHY-osasta. Kehyksen lopussa analysaattori näyttää kehyksen tarkistussumman olevan oikein.

Koordinaattori vastaa Beacon-kehyksellä. Kehyksen tyyppi on Beacon ja FrameControl-osa on ilmeinen. Tämän jälkeen koordinaattori kertoo SrcPanin olevan CAFE ja koordinaattorin SrcAdd olevan CABA. Näiden kenttien jälkeen seuraa informaatiota PAN:sta. Beacon ja Superframe Order ovat 0F, eli kyseessä on NonBeacon- pohjainen PAN. Cap Slot (CAP aikaväli) arvo on 0F, BLE on eli koordinaattori käy verkkovirralla, PAN eli koordinaattori sekä AP eli PAN assosiointi on sallittu. Seuraavat kaksi kenttää kertovat, että GTS aikajaksot eivät ole käytössä. SP ja LP kertovat, ettei muita tulevia osoitteita ole.

Laite lähettää AssociationRequest-kehyyksen saatuaan Beacon-kehyyksen, jossa laite haluaa liittyä CAFE nimiseen PAN:iin ja kehys on lähetetty laitteelle CABA, eli koordinaattorille. Tässä kehyksessä lähettäjän osoite on laitteen kiinteä eeprom-muistin kahdeksaan ensimmäiseen muistipaikkaan talletettu 64-bitin osoite. Kehyksessä on tietoja liittyvästä laitteesta:

- laite pyytää käyttöönsä 16 bitin lyhyttä osoitetta
- laite on aina sähköt päällä ja ei käytä salausta
- laite on FFD ja alternate pan field kenttää ei käytetä

Tämän kehyksen koordinaattori kuittaa, jonka jälkeen laite lähettää omalla kiinteällä osoitteella DataRequest-kehyyksen, jonka koordinaattori kuittaa. Tämän jälkeen koordinaattori lähettää omalla ja laitteen kiinteällä osoitteella AssociationResponse-kehyyksen, jossa se ilmoittaa, että assosiointi on tehty ja laitteelle myönnetty osoite 0000. Tämän laite kuittaa.

Nyt laitteen assosiointi PAN-verkkoon nimeltä CAFE on tehty. Tästä eteenpäin laite käyttää osoitetta 0000 kommunikoidessaan. Koska kyseessä on NonBeacon-pohjainen verkko, tapahtuu lähetyksen CSMA-CA periaatteen mukaisesti. Kun laitteen painonappia painetaan, lähettää laite satunnaislukukyselyn koordinaattorille, johon koordinaattori vastaa lähettämällä luvun.

Toisen laitteen assosiointi verkkoon tapahtuu samalla tavalla ja koordinaattori antaa tälle dynaamisen osoitteen 0001.

Assosiointivaiheen jälkeen laitteet voivat aloittaa kommunikoinnin, josta on kehyksiä ja vasteaikoja liitteessä F.



## 5.5 Yhteenveto

Zigbee-liittymän toteuttamiseen laitteissa on tarjolla tranceiver-pohjainen, elektroniikkaan vahvasti integroitu ratkaisu, jossa laitteen ohjelmistoon on lisätty liittymän toiminnallisuus. Tässä vaihtoehdossa liittymä on mukana elektroniikan suunnittelun alkuvaiheista asti. Toinen vaihtoehto on lisätä laitteeseen Zigbee-liittymä modulina. Tämä mahdollistaa liittymän poiston tarvittaessa laitteesta puuttumatta laitteen ohjelmistoon tai elektroniikkaan muilta osin. Moduliratkaisua voidaan soveltaa myös integroituna laitteeseen, jolloin moduli on integroitu laitteeseen, mutta ei sido laitesuunnittelua siinä määrin kuin tranceiver-ratkaisu.

Tranceiver-ratkaisussa tarvitaan lisäksi ohjelmisto, joka toteuttaa vähintään MAC-kerroksen Zigbee-protokollapinosta. AT68RF230-piirin kanssa on luontevaa käyttää Atmel MAC-ohjelmistoa, joka toteuttaa MAC-kerroksen.

Mikäli laitteeseen halutaan toteuttaa koko Zigbee-protokollapino, jossa on mukana SSP (Security Service Provider), APS (Application Support ) ja ZDO (Zigbee Device Object) toiminnot, on näiden ohjelmakoodit lisättävä Atmel MAC:iin tai hankittava protokollapino, jossa niiden toteutus on mukana. APS lisää protokollapinon API-kutsuja sovelluksen tekijöille. SSP lisää protokollapinon autentikointi- ja salaustuen. ZDO lisää protokollapinon Bluetooth-kaltaisia profiileja, jotka kertovat laitteen kyvyt.

## 6 Tulokset

Ensimmäinen tavoite tälle opinnäytetyölle oli selvittää Bluetooth- ja Zigbee-verkon muodostumisen ja toiminnan erot. Bluetooth-verkon master-laite määritetään dynaamisesti verkkoa muodostettaessa, eli jokaisen laitteen ohjelmistossa on sekä master- että slave-valmiudet. On mahdollista, että master-valmiuksia ei jossakin laitteessa ole toteutettu, jolloin se ei voi muodostaa verkkoa. Käytännössä ensimmäisenä käynnistetty Bluetooth-laite asettuu masteriksi. Zigbee-verkossa laitteiden ohjelmistossa on päätetty toimiiko laite PAN-koordinaattorina, FFD- tai RFD-laitteena. Zigbee-ohjelmisto tukee molempia toimintoja, mutta laite toimii joko PAN-koordinaattorina tai FFD-RFD-laitteena. Samalla protokollapinolla voidaan toteuttaa kaikki Zigbee-verkon laitteet, jos protokollapino niitä vain tukee.

Kun Bluetooth-verkon master-laite on valittu, on slave-laitteiden pystyttävä tahdistumaan master-laitteen kelloon ja noudattamaan taajuushyppelysekvenssiä, jonka ne muodostavat masterilta saamiensa tietojen perusteella. Tämän jälkeen slave-laitteiden lähetykset ovat vahvasti master-laitteen ohjauksessa, koska master-laite jakaa lähetyksvuorot ja päättää lähetykskaistan käytöstä. Kun Zigbee-koordinaattori käynnistyy, se etsii kanavan, jolla ei vielä ole koordinaattoria ja näin Zigbee-verkko on olemassa. Beacon-pohjaisessa verkossa, koordinaattori alkaa lähettää Beacon-kehysjä tasaisin väliajoin ja mikä tahansa laite voi lähettää CSMA-CA-menettelmällä kahden Beacon-kehysvälikäytön välissä olevana aikana. NonBeacon-verkossa mikä tahansa laite voi lähettää CSMA-CA-menettelmän mukaisesti.

Merkittävin toiminnallinen ero Bluetooth- ja Zigbee verkkojen välillä löytyy siirtotien käytöstä. Kun Bluetooth-verkossa käytetään siirtokaistan kaikkia kanavia master- ja slave-laitteiden väliseen tiedonsiirtoon yhdellä peittoalueella taajuushyppelysekvenssin mukaisesti, niin Zigbee-verkossa yksi PAN-koordinaattori toimii kiinteästi yhden kanavan taajuudella. Näin ollen samalla 2.4GHz IMS-kaistalla peittoalueella voi toimia 16 Zigbee-verkkoa. Kun Zigbee-laitteet assosioituvat tiettyyn Zigbee-verkkoon, määräytyy niiden käyttämä lähetyks- ja vastaanottotaajuus kanavan mukaisesti. Koska Zigbee-liitynnässä ei käytetä taajuushyppelyä, on Zigbee-liitynnän elektroniikka yksinkertaisempaa ja näin ollen halvempaa kuin Bluetoothiin.

Molemmissa verkoissa assosiointivaiheessa sekä master-, että PAN-koordinaattorilaitteet antavat dynaamiset osoitteet laitteille. Assosiointivaiheessa laitteet käyttävät kiinteitä laitekohtaisia osoitteita. Zigbee-verkossa laitteella on vain kiinteä ja koordinaattorilta saatu 16-bitin dynaaminen osoite. Bluetooth-verkossa laitteella voi olla kiinteän osoitteen lisäksi dynaaminen kolmen bitin osoite ja kaksi dynaamista parked-tilan osoitetta, jotka ovat kahdeksan bittisiä.

Toinen tavoite oli selvittää, miten Zigbee-liityntä on toteutettavissa laitteisiin. Niin Bluetooth- kuin Zigbee-liityntä voidaan toteuttaa joko tranceiver- tai modulipohjaisesti. Ensimmäisessä toteutuksessa liityntä näkyy keskusyksikölle ohjelmoitavina rekistereinä. Tällöin liityntä on kiinteästi osa laitteen elektroniikkaa. Tranceiver-ratkaisu kattaa

OSI.-mallin fyysisen ja siirtoyhteyskerroksien toteutukset, joten verkkokerrokselta ylöspäin tarvitaan ohjelmallinen protokollapino. Valitun protokollapinon kommunikointi on sovitettava valittuun tranceiveriin.

Modulipohjaisen liittynän toteutus perustuu komentoihin, jotka moduli toteuttaa. Tällöin laitesuunnittelussa ei tarvitse keskittyä tranceiverosan ohjaamiseen, vaan liittynä näkyy keskusyksikölle komennoilla ohjattavana modemina. Tässä vaihtoehdossa on tyydyttävä niihin ominaisuuksiin, jotka modulin tekijä on toteuttanut. Bluetooth- ja Zigbee-liittynät toteutetaan hyvin samantyyllisesti, mutta Bluetoothin monimutkaisuus näkyy sekä elektroniikan, että protokollapinon toteutuksissa. Zigbee on onnistuttu toteuttamaan sekä elektroniikalta, että protokollapinolta Bluetoothia yksinkertaisemmin, mikä on ollut alun perin Zigbeeen tarkoituskin.

Kolmas tavoite oli löytää Zigbeelle sopivia käyttökohteita. Zigbee soveltuu pienten tietomäärien siirtämiseen ja sovelluksiin, joissa tarvitaan nopeaa assosiointivaihetta. Tällaisia ovat esimerkiksi teollisuuden anturiverkot ja kotiautomaation sovellukset. Zigbee-siirtoa laitteen kannalta voidaan pitää epädeterministisenä, koska sekä Beacon-, että NonBeacon-pohjaisessa verkossa lähetys perustuu CSMA-CA-menetelmään. Zigbee-tekniikassa ei ole tavoiteltu reaaliaikaisen median siirtoon tai siirrettävän tiedon saulaukseen tai suojaukseen liittyviä ominaisuuksia. Mikäli näitä ominaisuuksia tarvitaan, on valittava Bluetooth. Zigbee näyttää saavuttaneen ne tavoitteet, mitä sille on alun perin asetettu.

Neljäs tavoite liittyy opetuksen kehittämiseen koulutuskeskus Salpauksessa. Elektroniikka-asetajan ammatti- ja erikoisammattitutkintoja ajatellen lopputyö antoi runsaasti lisää osaamista likiverkoista. Piirikaavio- ja piirilevysuunnitteluvaihe ovat olleet sinällään opetuksessa mukana jo vuosia, mutta ohjelmisto tulee vaihtumaan PADS-tuotteista Eagle-tuotteeseen. Eaglen käyttöön siirtymisellä haetaan laajempaa käytettävyyttä, koska tämä ohjelma on rajoitetussa muodossa ilmainen. Piirilevyvalmistuksen ja kasaamisen kannalta Zigbee- ja balun-piirien koteloinnit vaativat huomiota. Näiden liittäminen piirilevyyn vaatii käytännössä ammattilaisvälineet. Keskeisin sisältö tässä lopputyössä on opetuksen näkökulmasta kuitenkin elektroniikan ja ohjelmiston rajapinta. Elektroniikkasuunnittelun kannalta tämä tarkoittaa Zigbee-tranceiverin, balunin ja antennin sisällyttämistä osaksi digitaalista tuotetta tai valmiin modulin käyttöä. Ohjelmistosuunnittelun näkökulmasta elektroniikan ohjaamista ja Zigbee-liittynän käyttämisestä muun ohjelmiston kannalta. Se, miten Zigbee-liittynä hyödynnetään myöhemmin tuotteissa, jää sovelluksen suunnittelijan hartioille. Opetusta tukevaa tietoa saatiin myös vertailemalla Bluetoothia ja Zigbeetä. Nyt Bluetoothin rinnalle saatiin toinen käyttökelpoinen likiverkkoliittynä kattamaan opetuksen aluetta langaton tiedonsiirto.

Kortteihin liittyi aluksi monenlaisia ongelmia. Aivan ensimmäinen protokortti tehtiin Atmel 128-mikro-ohjaimella, mutta tämä kortti oli hylättävä, koska kävi selville, että piireissä Atmel 128 ja Atmel 1281 on eroavaisuuksia joidenkin ajastinrekisterien kohdalla. Atmel MAC on testattu nimenomaan Atmel 1281 piirin kanssa. Kun toiseen korttiin lisättiin balun-piiri, saatiin tämä kortti lähettämään Zigbee-kehyksiä. Analysoinnilla huomattiin, että kehyksiä korruptoitui varsin paljon ja analysointori sai vain noin joka viidennen kehyksen oikein vastaan. Tämä kortti olisi uudelleen tehtynä ja ba-

lun-piirillä varustettuna toiminut erittäin todennäköisesti, mutta tässä vaiheessa päätettiin toiseen testikorttiin laittaa valmis liitännäiskortti, joka sisälsi tranceiverin, balunin ja antennin. Liitännäiskortti sisälsi aivan samoilla mitoilla olevan piirilevydipoli-antennin kuin ensimmäinen korttikin. Tämän toisen kortin kanssa ei elektroniikkaongelmia enää ilmennyt.

Atmel MACin kääntämisessä ja sisällyttämisessä osaksi ohjelmistoa ei ollut ongelmia. WinAVR-kääntäjästä oli käytettävä versiota, jolla Atmel MAC oli käännetty. Uudemmalla WinAVR-versiolla kääntäminen ei ainakaan suoraan onnistunut. Kääntämiseen tehty makefile toimi ilman muutoksia.

Jotta teoria ja käytäntö lopullisesti kohtaisivat, oli kolmesta protokortista muodostettava testiverkko. Verkon liikennettä analysoimalla selvisi, miten Atmel MACin koordinaattori toimii käytännössä. Zigbee-standardin mukaisesti koordinaattori etsii taajuuskaistalta vapaan kanavan ja alkaa tällä taajuudella vastata BeaconRequest-kehysiin tai lähettämään säännöllisin väliajoin Beacon-kehystä. Koska Atmel MAC ei tue Beacon-verkkoa, se siis kykenee vain vastaamaan BeaconRequest- ja AssociationRequest-kehysiin. Analysaattorilla nämä vaiheet olivat selkeästi nähtävissä. Edelleen Zigbee-standardin mukaisesti toimiva laite lähettää kaikille kanaville BeaconRequest-kehysten tai kuuntelee kanavilta säännöllisiä Beacon-kehysiä selvittääkseen peittoalueella olevat Zigbee-verkot. Näin laitteen käyttöliittymässä voidaan käyttäjälle näyttää löytyneet Zigbee-verkot ja käyttäjä valitsee verkon, johon haluaa liittyä. Atmel MAC lähettää kiinteästi koodissa määritetyille kanaville BeaconRequest-kehysten ja odottaa Beacon-kehystä, jonka saatuaan lähettää Association-kehysten. Nämä Zigbee-verkkoon liittymisvaiheen kehykset ovat niin ikään nähtävissä analysaattorilla. Kehyksiä tarkemmin tarkasteltaessa näkyy myös laitteen saaman dynaamisen verkko-osoitteen myöntäminen. Näiden vaiheiden jälkeen laitteet voivat kommunikoida koordinaattorin kautta.

Zigbee-verkkoa testatessa kokeiltiin molemmat variaatiot. Ensimmäisessä testissä käynnistettiin ensin laitteet ja viimeisenä koordinaattori. Laitteet lähettivät säännöllisesti BeaconRequest-kehysiä, kunnes käynnistynyt koordinaattori alkoi vastata ja assosiointivaiheet käynnistyivät. Toisessa testissä käynnistettiin ensin koordinaattori ja sen jälkeen laitteet. Molemmissa lopputulos oli sama, toimiva Zigbee-verkko. Koska Zigbee-verkko noudattaa CSMA-CD siirtomedian varausta, oli toisinaan varsin pitkiä aikoja, ennen kuin toinen laitteista pääsi lähettämään. Koska koordinaattorissa ja laitteissa ei ollut juuri muuta ajettavaa koodia kuin Atmel MAC, oli vasteajat saatuihin kehyksiin varsin nopeita.

Tämän opinnäytetyön pohjalta on hyvä jatkaa Zigbeeen parissa. Kaikki tarvittava tieto ja osaaminen Zigbee-liityntöjen toteuttamiseen on olemassa.

## Yhteenveto

Tälle opinnäytetyölle asetetut tavoitteet saavutettiin, vaikka Zigbee-liikenteen testausta varten jouduttiin rakentamaan kahdet testikortit ja tästä aiheutui viivästymistä. Molempien testikorttien avulla saatiin kuitenkin arvokasta tietoa liittynän elektroniikan toteutusvaihtoehtoista teoreettisen tiedon tueksi. Tämän työn pohjalta olisi hyvä jatkaa Zigbeeen tutkimista paneutumalla tarkemmin Atmel MACin toteutukseen ja siirtämiseen muille prosessoriarkkitehtuureille. Tämä tarkoittaisi sitä, että Atmel MACin sisältämät ajastintoteutukset olisi muokattava kohdeympäristön ajastimille soveltuviksi. Koska Atmel MAC on rakennettu kokonaan Atmel AT86RF230-piirin ympärille, ei se ole juurikaan käyttökelpoinen muiden Zigbee-tranceiverien kanssa. Mikäli tätä kuitenkin yritettäisi, olisi suurin osa Atmel MACista ohjelmakoodista kirjoitettava uudelleen. Toinen jatkokehityskohde olisi toteuttaa Beacon-verkko ja Zigbee Router-tuki Atmel MACiin.

Lopuksi on hyvä katsoa, mitä Zigbee-maailmalle kuuluu tällä hetkellä? Zigbee Alliance julkaisee neljännesvuosittain katsauksen Zigbee-maailman sen hetkisestä tilasta. Alliancen kuuluu tällä hetkellä noin 290 jäsenyritystä, joiden joukossa on kaikki merkittävät televiestintäalan yritykset. Tavoitteena on saavuttaa vuoden 2008 loppuun mennessä 300 jäsenyrityksen raja. Yrityksien mielenkiinnosta Zigbeetä kohtaan kuvastaa kasvavan jäsenmäärän lisäksi myös sertifioitujen Zigbee-tuotteiden määrä vuosittain. Luku vaihtelee, mutta se on kuitenkin kymmeniä, ellei muutamia satoja vuosittain. Erittymisesti Zigbee Smart Energy konsepti on saanut mielenkiintoa osakseen. Zigbee Alliance järjestää vuosittain jäsenilleen useita konferensseja ja seminaareja maailmanlaajuisesti, joissa ajankohtaisia asioita ja tuotteita tuodaan esiin. Vuonna 2008 oli kaiken kaikkiaan 17 tapahtumaa.[17] [18]

Zigbee-teknologia liitetään usein ympäristöystävällisessä mielessä uusiin tai olemassa oleviin tuotteisiin. Lyhyen kantaman tiedonsiirto- ja ratkaisujen pienellä energiankulutuksella on hyvä markkina-arvo ja erityisesti kilpailija Bluetoothiin verrattuna luvut ovat Zigbeelle varsin suotuisia. Miksi siis käyttää virtaa kuluttavaa Bluetooth-ratkaisua, jos langaton tiedonsiirto saadaan hoidettua vaatimuksiin nähden yksinkertaisemmalla ja halvemmalla Zigbee- tekniikalla? Seuraavassa on agenda Chicagon 2008 tapahtumasta:

- Building Automation & Controls
- Green Building Initiatives/Sustainability
- Information Technology & Business Management
- Specialized Climate Control Systems
- Radiant Heating & Hydronic Systems
- Ventilation & Air Movement Products

- Energy Efficiency
- Indoor Air Quality

Listasta näkyy Zigbeeen suhde automaatioon, ympäristöön ja energiatalouteen. Zigbee-tekniologialla näyttää tällä hetkellä olevan hyvä tilanne ja piiri- ja tuoteratkaisuja valmistavat tahot ovat varsin luottavaisia ja levollisia tulevaisuuden Zigbee-kysyntään ja markkinoihin. [17] [18]

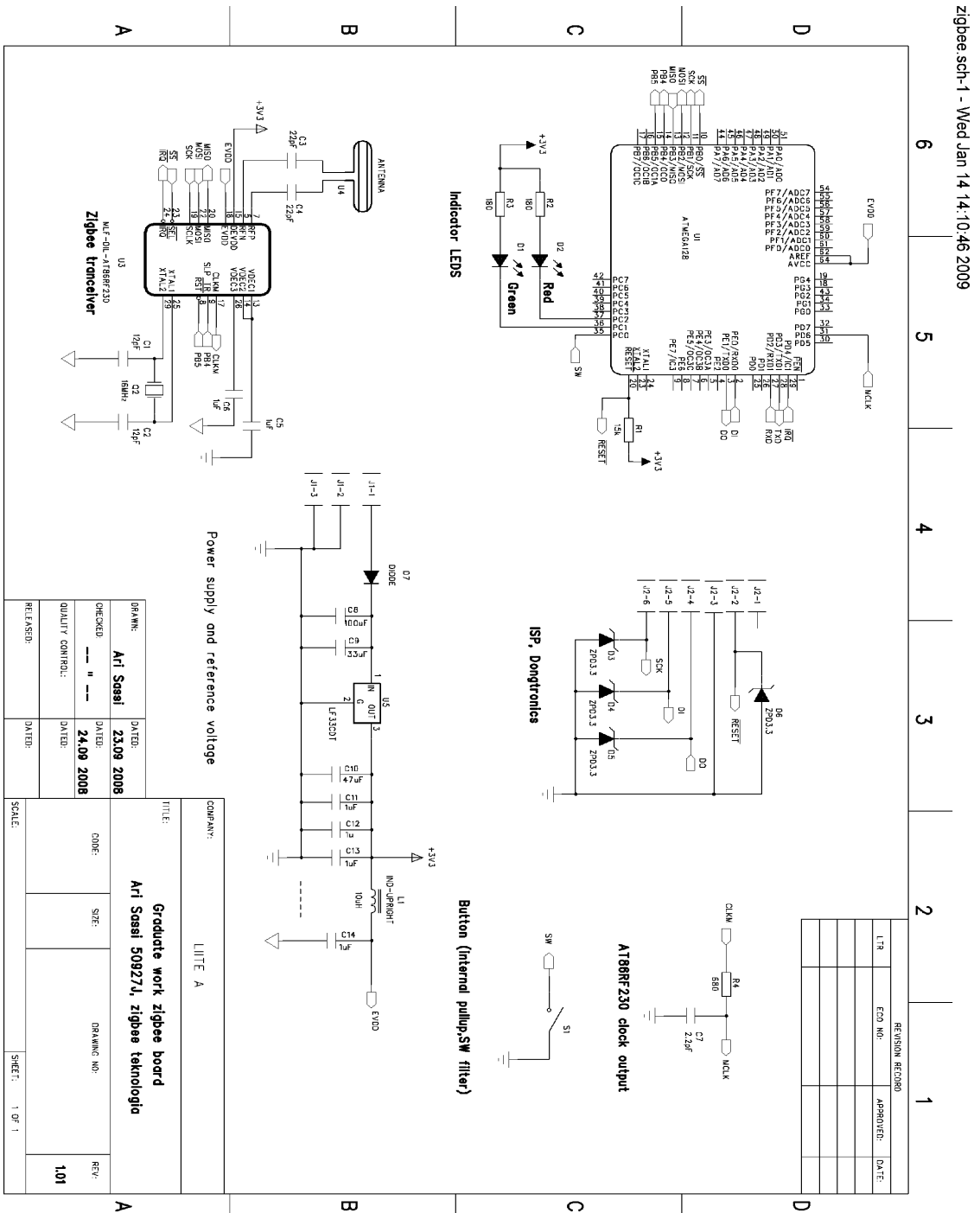
## Viitteet

- [1] Kammer, D., McNutt, G., Senese, B., *Bluetooth. Application Developer's Guide*, USA, Syngress, 2002.
- [2] Labiod, H., Afifi, H., De Santis, C., *Wi-Fi Bluetooth Zigbee and WiMax*, Netherlands, Springer, 2007.
- [3] Hytönen, T., *Bluetooth, Käyttö teollisuuden laiteohjauksessa, opetusmoniste*, Jyväskylä, 2003.
- [4] Eady, F., *Hands-On Zigbee, Implementing 802.15.4 with microcontrollers*, USA, Newnes 2007.
- [5] Sinem, E., *ZigBee/IEEE 802.15.4 Summary*, USA, Berkeley, September 2004
- [6] Dohl, J., Diehm, F., Grosa, P., *Zigbee*, Technische Universität Dresden, Dresden, November 2006
- [7] Atmel Corporation, Datasheet 2467S.pdf, *ATmega128(L) Summary*, USA, August 2007
- [8] Atmel Corporation, Datasheet doc5131.pdf, *AT86RF230 Zigbee/IEEE 802.15.4-tranceiver*, USA, June 2007
- [9] Atmel Corporation, Application note doc8087.pdf, *AVR2001: AT86RF230 Software Programmer's Guide*, USA, July 2007
- [10] Atmel Corporation, *Reference Manual for the IEEE 802.15.4 MAC library*, USA, Jun 2007
- [11] Atmel Corporation, Application note doc8095.pdf, *AVR2006: Design and characterization of the Radio Controller Board's 2.4GHz PCB Antenna*, USA, August 2007
- [12] Davies, P. *The indispensable Guide to C With Engineering Applications*, Allison-Wesley, England 1995
- [13] Tikkanen, H. *PADS Piirilevysuunnitteluopas 2*, Gummerus Kirjapaino Oy, Jyväskylä 2004
- [14] Barr, M, *Programming Embedded Systems in C and C++*, O'Reilly, USA, January 1999

- [15] Sickle, T.V. *Programming Microcontrollers in C*, LHH Technology Publishing, USA, 2001
- [16] Haikala I, Märijärvi J, *Ohjelmistotuotanto*, Suomen ATK-kustannus Oy, Espoo, 1997
- [17] Heile, B., Zigbee Alliance. Newsletter, verkkotiedote, 2008, heinäkuu, nro 3. Viitattu 15.9.2008. Saatavissa: [http://www.zigbee.org/en/newsletters/2008\\_07\\_Newsletter/0708-External.asp#newsletter](http://www.zigbee.org/en/newsletters/2008_07_Newsletter/0708-External.asp#newsletter)
- [18] Zigbee Alliance. Newsletter, verkkotiedote, 2008, syyskuu, numero 6. Viitattu 10.9.2008. Saatavissa: [http://www.zigbee.org/en/events/industry\\_events.asp](http://www.zigbee.org/en/events/industry_events.asp) 15.09.2008
- [19] Atmel Corporation, Application note doc5183.pdf, *ATAVRRZ200 Demonstration Kit User Guide*, USA, July 2006

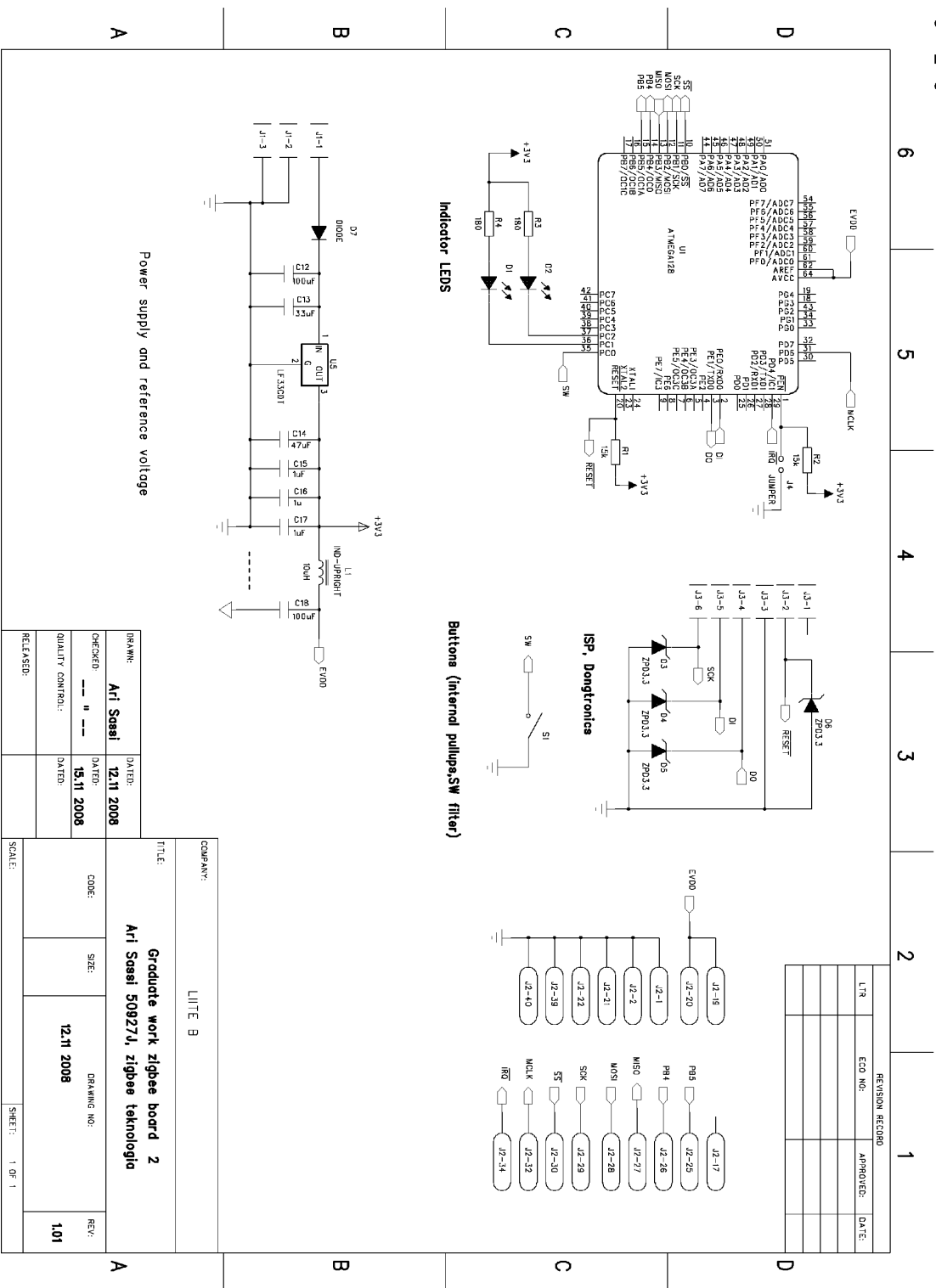


LIITE A



Kuva A1: Ensimmäisten testikorttien kytkentäkaava

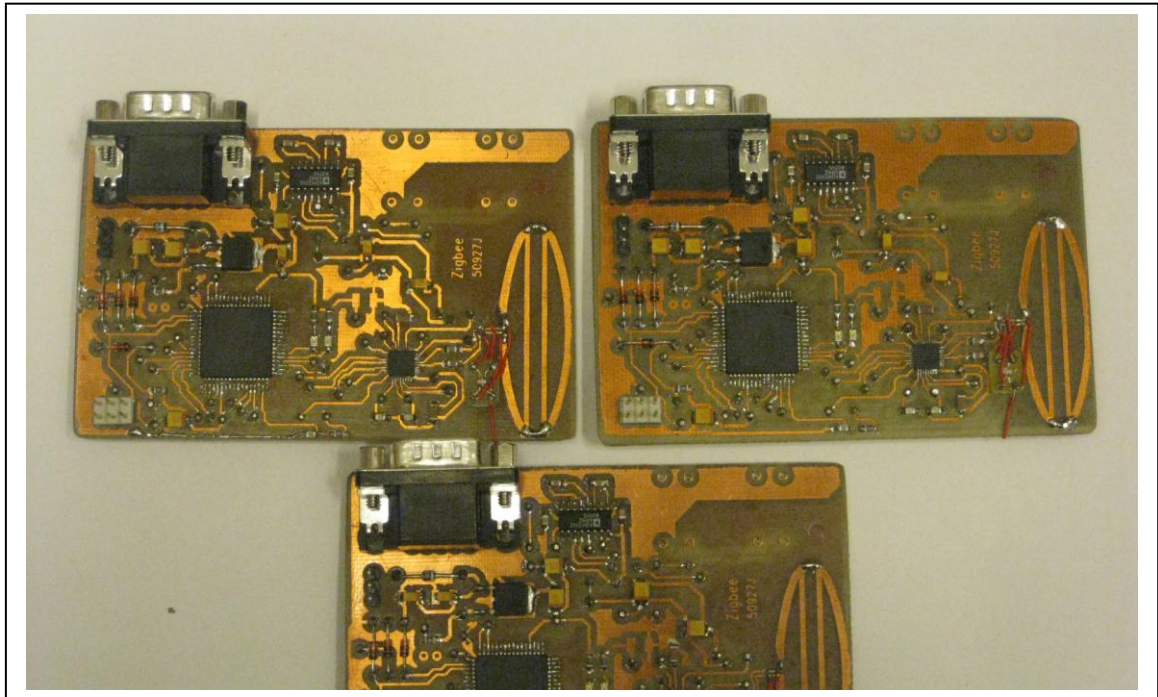
Huomaa balun-piiriin puuttuminen trnceiverin ja piirilevydipoliantennin välistä.



LIITE B

Kuva B1: Toisten testikorttien kytkentäkaava

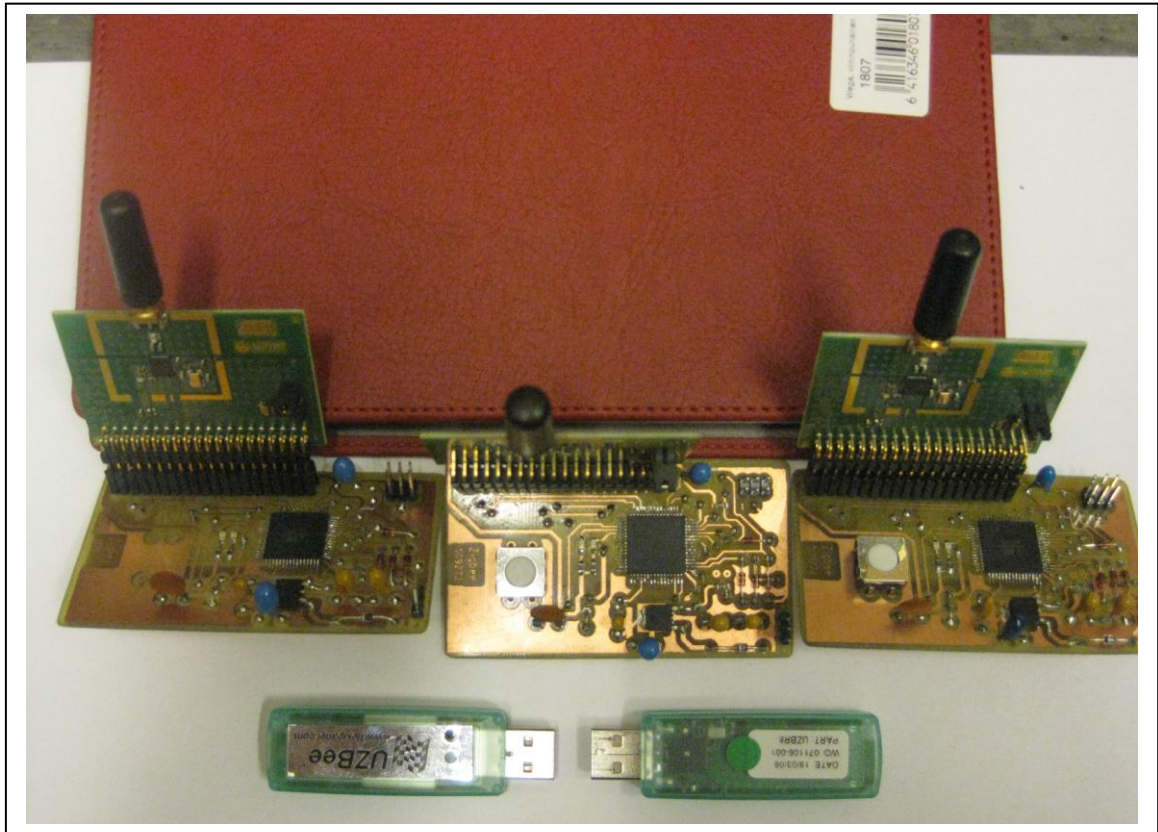
## LIITE C



Kuva C1: Ensimmäiset testikorit

Puuttuva balun-piiri on lisätty jälkikäteen omana pienenä piirilevynä piirilevydipoliantennin vasemmalle puolelle.

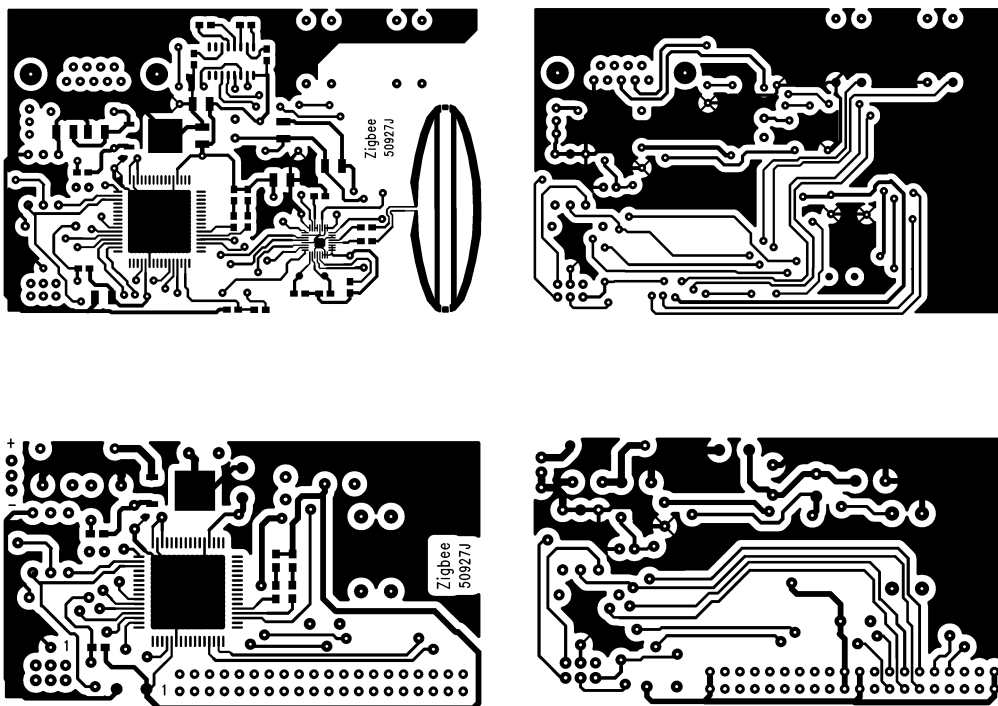
## LIITE C



Kuva C2: Toiset testikortit

Näissä korteissa tranceiver, balun ja single ended-tyyppinen antenni ovat omalla piirilevyllään, joka on liitetty testikorttiin.

## LIITE D



Kuva D1: Testikorttien lay-out-paneeli

Panelin ylärivissä näkyy ensimmäisten testikorttien päily- ja alapuolen reititykset. Alemmassa näkyy toisten testikorttien vastaavat puolet. Piirilevyn valotusvaiheessa piirilevy asennetaan kalvojen väliin.

## LIITE E

The screenshot displays the MACChongle / Pico IEEE 802.15.4 Sniffer interface. The top right shows a list of channels from 2405MHz to 2477MHz. The main window shows a list of captured frames with columns for Time (T), Direction (DT), Length (Len), Type, Error (Enr), Pad, Ack, Seq, Destination PAN (DestPAN), Destination Address (DestAdd), Source PAN (SrcPAN), Source Address (SrcAdd), and Frame Check Sequence (FCS). The bottom left shows a node list with Node 1 at 97% and Node 2 at 03%.

**General Statistics:**

Name	PAN	Add	LOI
General	FFFF	FFFF	
Node 1	CAFE	0001	97%
Node 2	CAFE	CA...	03%
Node 3			

**Frame 1 Details:**

T	DT	Len	Type	Enr	Pad	Ack	Seq	DestPAN	DestAdd	SrcPAN	SrcAdd	FCS					
08:54:45:0:028	29	Cnd	No	No	Yes	A3	CAFE	0202020202020202	CAFE	0101010101010101	Association Addr Status	Response	0000	OK	-58	97%	OK

**Frame 2 Details:**

T	DT	Len	Type	Enr	Pad	Ack	Seq	DestPAN	DestAdd	SrcPAN	SrcAdd	FCS					
08:54:45:0:039	5	Ack	No	No	No	A3	CAFE	0202020202020202	CAFE	0101010101010101	Association Addr Status	Response	0000	OK	-58	97%	OK

Kuva E1: Ensimmäisen laitteen assosiointivaiheen kehukset



LIITE E

The screenshot displays the MACdangle / Probe IEEE 802.15.4 Sniffer interface. At the top, there are controls for 'Port / Channel Select' (Import, Scan) and 'Field Select' (Application layer). Below this, 'Frame Type' and 'Delivery method' are visible. The main window shows a list of captured frames with columns for Time, Len, Type, Encr, Pad, Ack, Seq, and various protocol-specific fields. A detailed view of a selected frame (08:53:420:042) is shown below the list, including fields like SrcPAN, DestAdd, and Rssi Iqi. The interface also includes a 'Statistics' panel on the left showing 8 packets and 0 errors with a 0.000% error rate.

Time	Len	Type	Encr	Pad	Ack	Seq	SrcPAN	DestAdd	Rssi Iqi	FCS	
08:53:41	10	Cmd	No	No	9C	FFFF	FFFF	Request	-64	97%	OK
08:53:41.0	13	Bcn	No	No	8C	CAFE	CABA	Request	00	00	00
08:53:41.0	21	Cmd	No	No	9D	CAFE	CABA	Request	00	00	00
08:53:41.0	25	Cmd	No	No	9D	CAFE	CABA	Request	00	00	00
08:53:42.0	16	Cmd	No	No	9E	CAFE	0303030303030303	Request	-68	97%	OK
08:53:42.0	5	Ack	No	Yes	9E	-58	98%	OK			
08:53:42.0	29	Cmd	No	No	A2	CAFE	0303030303030303	Request			
08:53:42.0	5	Ack	No	No	A2	-69	97%	OK			

Kuva E2: Toisen laitteen assosiointivaiheen kehukset

## LIITE F

The screenshot displays the MACdongle / Probe IEEE 802.15.4 Sniffer application. The interface is divided into several sections:

- Top Panel:** Includes 'Port / Channel Select' (Import, Scan), 'Field Select', and 'APL (Application layer)'.
- Left Panel:** Contains 'Prets' (PHY, MAC, NWK, APL, AF, ZDO, ZigBee, Malbox) and 'Frame Select' (PHY Errors, MAC Beacon, MAC Command, MAC ACK, NWK, APL).
- General Settings:** Lists nodes: Node 1 (CAFE 0001, 97%), Node 2 (CAFE CA..., 03%), and Node 3 (CAFE 0000, 99%).
- Statistics:** Shows 12 packets, 0 errors, and a 0.000% error rate.
- Main Table:** A detailed frame capture log with columns for Time (T), Duration (dt), Length (Len), Type, Encr, Pad, Ack, Seq, DestPAN, DestAdd, SrcPAN, SrcAdd, RSSI, and LOI. The table shows several frames, including ACKs and data packets, with the last one being highlighted in green.
- Right Panel:** A frequency grid from 2405MHz to 2471MHz, showing ED N/A and 0 Beasons for most channels.

T	dt	Len	Type	Encr	Pad	Ack	Seq	DestPAN	DestAdd	SrcPAN	SrcAdd	RSSI	LOI	FCS
08:56:19	51	51	Data	No	No	Yes	9F	CAFE	CABA	Same	0000	-63	97%	OK
08:56:19	0.018	5	Ack	No	No	No	9F	-59	98%	OK				
08:56:19	0.014	12	Data	No	No	No	A4	CAFE	FFFF	Same	CABA	-60	98%	OK
08:56:42	23	197	Data	No	No	Yes	9F	CAFE	CABA	Same	0001	-77	97%	OK
08:56:42	0.027	5	Ack	No	No	No	9F	-62	97%	OK				
08:56:42	0.023	12	Data	No	No	No	A5	CAFE	FFFF	Same	CABA	-63	97%	OK
08:56:47	4.610	51	Data	No	No	Yes	A0	CAFE	CABA	Same	0001	-72	97%	OK
08:56:47	0.036	5	Ack	No	No	No	A0	-63	97%	OK				
08:56:47	0.029	12	Data	No	No	No	A6	CAFE	FFFF	Same	CABA	-62	97%	OK
08:56:50	3.425	51	Data	No	No	Yes	A0	CAFE	CABA	Same	0000	-76	96%	OK
08:56:50	0.051	5	Ack	No	No	No	A0	-58	98%	OK				

Kuva F1: Tiedonsiirtoa laitteiden ja koordinaattorin välillä