



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

**MATTI YLINEVA**  
**RADIOLAITTEEN OHJELMOIMINEN CE- JA**  
**FCC-SERTIFIOINTEJA VARTEN**  
Diplomityö

Tarkastaja: Prof. Hannu-Matti Järvi-  
nen

Tarkastaja ja aihe hyväksytty 1.  
marraskuuta 2017

## TIIVISTELMÄ

**MATTI YLINEVA:** Radiolaitteen ohjelmoiminen CE- ja FCC-sertifiointeja varten  
Tampereen teknillinen yliopisto  
Diplomityö, 62 sivua  
Tietotekniikan DI-tutkinto-ohjelma  
Pääaine: Ohjelmistotuotanto  
Tarkastaja: Prof. Hannu-Matti Järvinen

Avainsanat: sertifiointi, radiolaitte, python, C, CE, FCC

Diplomityössä käsiteltävää radiolaitetta halutaan myydä Euroopan unionin talousalueella ja Yhdysvalloissa. Ennen kuin laitetta voidaan viedä näiden kahden alueen markkinoille, laite tulee sertifioida. Työn tarkoituksena on luoda käytettävälle radiolaitteelle ohjelma, jonka avulla radiolaitteelle voidaan suorittaa sertifiointitestaus. Sertifiointitestauksen suorittaa kolmas osapuoli. Testaus on osa radiolaitteen CE- ja FCC-sertifioimisprosessia. Jotta radiolaitteelle voi saada edellä mainitut sertifikaatit, tulee osoittaa, että se täyttää näiden kahden sertifikaatin sille asettamat vaatimukset. Sertifiointitestauksessa laitetta testaan monissa eri testeissä ja tässä työssä esitetään radiolaitteelle tehtäviä testejä. Kun tiedetään, millaisia testejä radiolaitteelle tullaan suorittamaan, saadaan selville radiolaitteelta vaadittu toiminnallisuus näiden testien tekemiseen.

Työssä käytettävälle radiolaitteelle luodaan ohjelma, jolla voidaan kommunikoida langattoman lähiverkon ylitse muun radiolaitteen kanssa. Ohjelman avulla laitteesta voidaan luoda tukiasema, laitteella voidaan ottaa yhteys toiseen tukiasemaan, lähettää ja vastaanottaa viestejä langattoman lähiverkkoyhteyden kautta, lähettää ja vastaanottaa viestejä langattomassa lähiverkossa muodostamatta yhteyttä mihinkään laitteeseen, lukea ja kirjoittaa laitteen flash-muistiin sekä kommunikoida tietokoneen kanssa sarjaliikenneväylän ylitse. Lisäksi työssä luodaan erillinen tietokoneohjelma, jota käytetään apuna radiolaitteen testauksessa. Tietokoneohjelmassa on graafinen käyttöliittymä, jotta sen käyttö olisi mahdollisimman intuitiivista. Tietokoneohjelma kommunikoi radiolaitteen kanssa ja ohjelmalla voidaan valita, mikä toiminnallisuus radiolaitteella halutaan käynnistää. Sertifiointitestauksen tekevä henkilö saa näin radiolaitteen suorittamaan haluamansa toiminnallisuuden.

Kolmannelle osapuolelle toimitettiin testaukseen vaaditut materiaalit ja ohjelmat. Radiolaitteelle ohjelmoitu toiminnallisuus ohjelmoitiin kahteen radiolaitteeseen ja nämä laitteet toimitettiin testaavalle osapuolelle. Lisäksi toimitettiin yksi kannettava tietokone, johon asennettiin luotu tietokoneohjelma. Näillä ohjelmilla kolmas osapuoli pystyi suorittamaan vaaditut testitapaukset ja sertifioimisprosessia pystyttiin jatkamaan testaustulosten saamisen jälkeen.

## ABSTRACT

**MATTI YLINEVA:** Programming radio equipment for CE and FCC certifications  
Tampere University of Technology  
Master of Science Thesis, 62 pages  
Degree Programme in Information Technology, MSc (Tech)  
Major: Software Engineering  
Examiner: Professor Hannu-Matti Järvinen

Keywords: certification, radio equipment, python, C, CE, FCC

Radio equipment discussed in this master's thesis will be marketed in European Union and in the United States of America. Before equipment can be sold in the markets, the equipment must be certificated. The purpose of this master's thesis is to program a software to the radio equipment, so certification testing can be done. Certification testing is done by a third party. Certification testing is part of the CE and FCC certification process. The equipment must demonstrate fulfilment of FCC certification's requirements to get this certifications. The manufacturer of CE certificated product is responsible to provide necessary documents and test results if asked. Certification testing involves many different test cases and these cases are introduced in this thesis. When one knows what kind of test cases are done to the radio equipment, he or she can conclude which features are needed to complete the test cases.

A software is programmed for the radio equipment, which uses wireless local area network to communicate with other radio equipments. By using this software, it is possible to deploy it as an access point, contact it to an existing access point, send and receive data to connected radio equipment or without connecting to any other device. It is also possible to read and write to the equipment's flash-memory and communicate with computer by using serial connection. In this thesis, a separate PC software is also programmed. The latter software is done to support certification testing and make it possible to control radio equipment's features. The PC software has a graphical user interface for easy and intuitive usage. The software communicates with radio equipment via serial connection. By using the software the user can control, which feature is run in radio equipment. In this way, the person doing the certification testing can run whatever feature is necessary to complete the current test case.

All necessary software and other material were delivered to the third party. Radio equipment's software were flashed on two radio equipments and these were delivered to the third party. The computer software was installed on a laptop, which was also delivered. By using these two software the third party was able to perform radio tests needed by certifications.

## ALKUSANAT

Ensimmäisenä haluan kiittää diplomityöni tarkastajaa Hannu-Matti Järvistä ja ohjaajaani Sakari Junnilaa rakentavista kommentteista. Haluan myös kiittää Wapice Oy:tä diplomityöni aiheesta, sekä tilaisuudesta osallistua projektiin. Lisäksi haluan kiittää vielä ystävääni Tuomas Räikköstä kommentteistaan ja huomioista diplomityön kieliasuun liittyen. Viimeisenä haluan kiittää avopuolisoani Riitta Niemistä, joka potki minut pois sängystä työtä kirjoittamaan myös silloin, kun en sitä itse olisi halunnut tehdä.

Tampereella, 7.5.2018

Matti Ylineva

## SISÄLLYSLUETTELO

1.	JOHDANTO .....	1
2.	KÄYTETTÄVÄ RADIOLAITE JA SERTIFIOINTI .....	4
2.1	Sertifioitava radiolaite ja laitteen ominaisuudet.....	4
2.2	EMC-sertifioinnin tarkoitus .....	6
2.3	Radiolaitetta koskevat direktiivit Euroopassa .....	7
2.3.1	Radiolaitedirektiivi .....	7
2.3.2	Sähkömagneettista yhteensopivuutta koskeva direktiivi .....	8
2.3.3	Conformité Européenne sertifiointimerkki .....	8
2.4	Federal Communication Commission.....	9
3.	RADIOLAITTEEN KÄYTTÖ ERI TESTEISSÄ .....	10
3.1	Sähkömagneettisen yhteensopivuuden testaaminen .....	10
3.2	Säteilevän emission testaaminen.....	11
3.3	Johtuvan emission testaaminen.....	12
3.4	Häiriöiden sietotestaus .....	12
3.5	Ympäristötestaus.....	13
3.6	ALT- ja HALT-testaus .....	13
3.7	Radiolaitetestaus .....	14
4.	SERTIFIOINTITESTIEN SUORITTAMISEEN VAADITTU TOIMINNALLISUUS .....	16
4.1	Datan lähettäminen ja vastaanottaminen langattoman lähiverkkoyhteyden ylitse .....	17
4.2	Datan jatkuva lähettäminen ja vastaanottaminen langattoman lähiverkkoyhteyden ylitse .....	17
4.3	Muodostetun yhteyden tarkkaileminen.....	18
4.3.1	Radiolaitteen asettaminen tukiasemaksi ja yhdistäminen tukiasemaan .....	19
4.3.2	Luodun yhteyden katkeamattomuuden varmistaminen .....	19
4.4	Radiolaitteen flash-muistille kirjoittaminen.....	19
4.5	Jatkuva kirjoittaminen flash-muistille.....	20
4.6	Radiolaitteen maksimikuormittaminen.....	20
4.7	Käyttöliittymän kenttien lokittaminen .....	21
5.	RADIOLAITTEELLE OHJELMOITU TOIMINNALLISUUS .....	22
5.1	Toteutettu toiminnallisuus käyttäen C-ohjelmointikieltä.....	23
5.2	Toteutuksen arkkitehtuuri .....	23
5.3	Tietokoneen ja radiolaitteen välinen kommunikointi .....	25
5.4	Säikeiden välinen kommunikointi .....	27
5.5	Verkkoprosessorin käynnistäminen .....	28
5.6	Radiolaitteen asettaminen tukiasema- ja asiakasmoodiin.....	31
5.7	Datan lähettäminen ja vastaanottamien tukiasemana ja asiakkaana .....	33

5.8	Datan jatkuva lähettäminen yhdistyneeseen radiolaitteeseen .....	36
5.9	Datan jatkuva lähettäminen ja vastaanottaminen radiolaitteella yhteystömästi.....	36
5.10	Kirjoittaminen ja lukeminen radiolaitteen flash-muistille .....	40
5.11	Jatkuva muistille kirjoittaminen ja lukeminen .....	43
5.12	Radiolaitteen kuormittaminen.....	43
6.	TESTAUKSEN TUEKSI TOTEUTETTU TIETOKONEOHJELMA .....	45
6.1	Testauksen asettamat vaatimukset käyttöliittymälle .....	45
6.2	Valittu ohjelmointikieli ja käytetyt kirjastot .....	46
6.3	Tietokoneohjelman arkkitehtuuri .....	48
6.4	Tietokoneohjelman säikeiden välinen kommunikointi .....	50
6.5	Käyttöliittymän käyttäminen ja kommunikoiminen radiolaitteen kanssa...	51
6.6	Käyttöliittymän tarjoama tuki radiolaitteen käyttämiseen.....	52
6.6.1	Kahden radiolaitteen välisen yhteyden tarkkailu.....	53
6.6.2	Muistille kirjoittamisen ja lukemisen onnistumisen seuraaminen	53
6.6.3	Viestien kirjaaminen lokeihin .....	54
7.	TULOKSET JA KEHITYSEHDOTUKSET .....	55
7.1	Sekvenssien luominen ja ajaminen .....	56
7.2	Dynaaminen muistinhallinta säikeiden välisessä kommunikoinnissa .....	57
7.3	Verkkoprosessorin käyttö.....	58
8.	YHTEENVETO.....	59
	LÄHTEET.....	60

## KUVALUETTELO

<b>Kuva 1.</b>	<i>Kuvaus radiolaitteen ja siihen kiinnittyneiden laitteiden välisestä viestinnästä.....</i>	5
<b>Kuva 2.</b>	<i>Säteilevän emission testaamiseen käytettäviä antenneja. ....</i>	11
<b>Kuva 3.</b>	<i>Testiohjelman arkkitehtuuri esitetty komponenttikaaviona. ....</i>	24
<b>Kuva 4.</b>	<i>Sarjaliikenneväylää ja langatonta lähiverkkoyhteyttä pitkin saatujen viestien välittäminen ProductionTester-säikeelle. ....</i>	25
<b>Kuva 5.</b>	<i>Käyttöliittymän luominen. ....</i>	47
<b>Kuva 6.</b>	<i>Tietokoneohjelman arkkitehtuuri esitetty komponenttikaaviona. ....</i>	48
<b>Kuva 7.</b>	<i>Viestin lähettäminen tietokoneohjelmasta sarjaliikenneyhteyttä pitkin radiolaitteelle. ....</i>	50
<b>Kuva 8.</b>	<i>Sertifiointitestausta varten luotu käyttöliittymä. ....</i>	52

## OHJELMALUETTELO

<i>Ohjelma 5.1.</i>	<i>Ohjelman main-lohko.</i>	29
<i>Ohjelma 5.2.</i>	<i>Verkkoprosessorin käynnistäminen.</i>	30
<i>Ohjelma 5.3.</i>	<i>Socket_run-lohko.</i>	32
<i>Ohjelma 5.4.</i>	<i>sl_WlanSet-metodin käyttäminen.</i>	33
<i>Ohjelma 5.5.</i>	<i>l_WlanConnect-funktio.</i>	34
<i>Ohjelma 5.6.</i>	<i>Datan lähettäminen ja vastaanottaminen.</i>	35
<i>Ohjelma 5.7.</i>	<i>Datan lähettäminen yhteydettömästi.</i>	38
<i>Ohjelma 5.8.</i>	<i>Datan vastaanottaminen yhteydettömästi.</i>	39
<i>Ohjelma 5.9.</i>	<i>Tiedoston olemassaolon varmistaminen.</i>	40
<i>Ohjelma 5.10.</i>	<i>Kirjoittaminen flash-muistille.</i>	41
<i>Ohjelma 5.11.</i>	<i>Lukeminen flash-muistilta.</i>	42



## LYHENTEET JA MERKINNÄT

ALT	<i>engl. accelerated life testing</i> , Laitteen heikkouksien tunnistamiseen käytetty testaus
CE-merkki	Conformité Européenne -sertifiointimerkki
EMC	<i>engl. electromagnetic compatibility</i> , sähkömagneettinen yhteensopivuus
EMC-sertifikaatti	<i>engl. electromagnetic compatibility certification</i> , sähkömagneettisen yhteensopivuuden sertifikaatti
ESD-testi	<i>engl. electrostatic discharge testing</i> , sähköstaattisen purkauksen sietotesti
FCC	Federal Communication Commission
Flash-muisti	uudelleen ohjelmoitava puolijohdemuisti
HALT	<i>engl. highly accelerated life testing</i> , Laitteen heikkouksien tunnistamiseen käytetty testaus
IC	Industry Canada
IP-osoite	<i>engl. internet protocol address</i> , kytketyn verkkosovittimen yksilöivä osoite
MAC-osoite	<i>engl. Media Access Control address</i> , verkkosovittimen yksilöivä osoite
RE-direktiivi	<i>engl. Radio Equipment Directive</i> , Radiolaitedirektiivi
RF-laite	<i>engl. radio frequency equipment</i> , radiotaajuuksia käyttävä laite
SAR-arvo	<i>engl. speific absorption rate, SAR</i> , Ominaisabsorptionopeus
SSID	<i>engl. service set identifier</i> , Langattoman lähiverkon tunnus
TELEC	Terminal Equipment Conformity Certification Service
WPA	<i>engl. Wi-Fi Protected Access</i> , Langattomassa lähiverkossa käytettävä salausalgoritmi
WPA2	<i>engl. Wi-Fi Protected Access II</i> , Langattomassa lähiverkossa käytettävä salausalgoritmi

# 1. JOHDANTO

Nykyisten säädösten ja standardien suojaamassa yhteiskunnassa ei uutta tuotetta voida heti viedä kauppoihin myytäväksi. Ennen kuin tuotetta voidaan myydä kuluttajille, tuotteen oikeanlaisesta ja turvallisesta toiminnasta tulee varmistua. Oikeanlaista toimintaa on tuotteen haluttu ja virheetön toiminnallisuus. Yleensä tuotteen tekijä vastaa näiden asioiden testaamisesta ja varmistamisesta. Tuote voi kuitenkin aiheuttaa sivuvaikutuksia muihin tuotteisiin tai jopa terveyshaittoja, vaikka sen primäärinen toiminta oltaisiinkin testattu. Oikeanlaiseen toimintaan kuuluu myös, ettei tuote häiritse toiminnallaan muita tuotteita tai vaaranna ihmisen terveyttä. Tuote tulee sertifioida, ennen kuin sitä voidaan myydä asiakkaille. Sertifioimisella varmistetaan, että kuluttajien saatavilla olevat tuotteet ovat turvallisia ja ne noudattavat markkina-alueen lakeja ja määräyksiä. Sertifioidulle tuotteelle annetaan sertifioimismerkki, joka laitetaan tuotteen pakkaukseen tai tuotteeseen. Suomalaisille tunnetuin sertifiointimerkki lienee CE-merkki. Valmistaja on vakuuttanut, että tämän merkin omaava tuote täyttää Euroopan unionin asettamat vaatimukset. Siksi tuotetta voidaan myydä Euroopan unionin alueella.

Tämän diplomityön tarkoituksena on selvittää lukijalle, mitä toiminnallisuuksia langaton lähiverkkoa käyttävällä radiolaitteella tulee pystyä tekemään, jotta se voidaan sertifioida. Ennen kuin radiolaitteelle voidaan luoda tarvittu toiminnallisuus, tulee tietää täytettävät vaatimukset. Nämä vaatimukset tulevat esimerkiksi eri direktiiveistä, standardeista ja laeista. Sertifiointitestauksessa laitteelle suoritetaan sellaiset testitapaukset, joilla voidaan todentaa laitteen täyttävän annetut vaatimukset. Jotta radiolaitte voidaan testata, tulee siihen ohjelmoida toiminnallisuus, joka mahdollistaa sertifiointitestauksen. Ohjelmoitava toiminnallisuus saadaan selville, kun tiedetään radiolaitteelle suoritettavat sertifioimistestit.

Työssä kerrotaan esimerkkien avulla, miksi sertifioiminen on tärkeää, sekä esitellään direktiivejä ja standardeja, jotka asettavat vaatimuksia radiolaitteelle. Tämän lisäksi tuodaan esille, millaisilla testeillä radiolaitetta testataan. Työn keskeisenä sisältönä on luoda sertifioitavalle radiolaitteelle toiminnallisuus, jonka avulla radiolaitteelle voidaan suorittaa sertifiointitestaus. Testauksen tukemiseksi luodaan myös tietokoneohjelma, jonka avulla sertifiointitestauksen aikana radiolaitetta pystytään kontrolloimaan.

Sertifioimisprosessi ja vaatimukset riippuvat paljon sertifioitavasta tuotteesta ja haetuista sertifikaateista. Tässä työssä käsitellään ainoastaan työssä käytettävän radiolaitteen sertifioimista. Työssä otetaan kantaa radiolaitteen piirilevyn suunnittelun vaikutuksista sertifioimisprosessiin niiltä osin, kuin se liittyy sertifioimistestaamiseen. Työssä ei myöskään käsitellä sertifioimisprosessia kokonaisuudessaan, vaan huomio kohdistuu ainoastaan radiolaitteeseen kohdistuvien sertifioimistestien suorittamisen mahdollistavan toiminnalli-

suuden toteuttamiseen. Työssä ei käsitellä sertifiointitestauksesta saatuja mittaustuloksia tai esitellä raja-arvoja joidenka alle mittaustulosten tulisi olla, jotta radiolaitte voitaisiin sertifioida.

Työn tavoitteena on määrittää ja luoda radiolaitteen sertifiointia varten vaadittu toiminnallisuus. Työssä käytettävä metodologia on konservatiivinen. Ensimmäiseksi työssä esitellään radiolaitteen sertifiointiin liittyviä termejä, standardeja, direktiivejä ja lakeja. Tämä jälkeen työssä käydään läpi radiolaitteelle suoritettavat testit. Seuraavaksi määritellään millainen toiminnallisuus radiolaitteella tulee olla ohjelmoituna, jotta testit voidaan suorittaa. Määrittelyn jälkeen esitellään luotu esimerkkiratkaisu. Viimeisenä kerrotaan työn tulokset, arvioidaan esimerkkiratkaisun toteutus sekä annetaan kehitysehdotukset.

Luvussa 2 käydään läpi mitä sertifiointilla tarkoitetaan, kuvataan tässä diplomityössä käytettävä radiolaitte, esitellään radiolaitteen sertifiointiin liittyviä direktiivejä ja standardeja. Luvussa myöskin määritetään mitä tarkoitetaan CE- ja FCC-sertifioinneilla ja mitä näiden sertifiointien saamiseksi tarvitaan.

Luvussa 3 esitellään radiolaitteelle tehtävät sertifiointitestit. Näillä testeillä voidaan varmistaa, että radiolaitte täyttää sille asetetut vaatimukset. Jokaisesta esitellystä sertifiointitestistä kerrotaan pintapuolisesti, mitä varten sertifiointitestausta tulee tehdä, sekä mitä testissä radiolaitteelle tehdään.

Luvussa 4 määritellään vaadittu toiminnallisuus, joka radiolaitteella tulee pystyä tekemään. Tässä luvussa kerrotaan millaista toiminnallisuutta radiolaitteelle ohjelmoitavalla ohjelmalla tulee pystyä tekemään, jotta radiolaitteelle voidaan suorittaa tarvittavat sertifiointitestit. Luvussa myöskin asetetaan vaatimus tietokoneohjelmalle, jonka avulla radiolaitetta voidaan hallinnoida sertifiointitestauksen aikana. Vaatimukset on käyty läpi niin, että radiolaitteelta ja tietokoneohjelmalta vaaditut ydintoiminnallisuudet saadaan esitellyä.

Luvussa 5 kuvataan radiolaitteelle ohjelmoidut toiminnallisuudet. Luvussa esitellään luodun ohjelman arkkitehtuurillinen ratkaisu sekä kerrotaan perustellusti työtä varten valittu ohjelmointikieli. Lisäksi käsitellään, millä tavalla radiolaitteelta vaaditut toiminnallisuudet on ohjelmallisesti toteutettu. Tietokoneohjelman ja radiolaitteen sarjaliikenneyhteyden kautta tapahtuva viestien vaihto on esitelty tässä luvussa.

Luvussa 6 esitellään luotu tietokoneohjelma sekä ohjelman arkkitehtuurillinen ratkaisu. Tässä luvussa esitellään myös tietokoneohjelmaan ohjelmoitu toiminnallisuus, sekä kerrotaan, millaista tukea ohjelma tarjoaa radiolaitteelle. Tietokoneohjelmaa varten käytettävät kirjastot sekä käytettävä ohjelmointikieli määritellään tässä luvussa.

Luvussa 7 kootaan diplomityön toteuttavan vaiheen tulokset sekä esitellään kehitysehdotukset. Luvussa kerrotaan mitä luotiin ja mitä toimitettiin kolmannelle osapuolelle sertifiointitestausta varten. Lisäksi kerrotaan, miten sertifiointitestausta tulokset. Luvussa

listataan kohteita, joihin tulisi kiinnittää huomiota, mikäli tässä työssä luotuja ohjelmia jatkokehitetään.

Luvussa 8 kootaan yhteen työn tavoitteet, mitä työssä tehtiin, mitä varten ja mitä saatiin aikaan. Luvussa esitellään työn sisältö lyhyesti, mutta ilman luotujen ohjelmien tarkempaa esittelemistä. Luvussa kerrotaan myös sertifiointitestauksen tulokset.

## 2. KÄYTETTÄVÄ RADIOLAITE JA SERTIFIOINTI

Sertifiointi on kolmannen osapuolen tekemä todistus siitä, että tuote, palvelu, henkilö tai johtamisjärjestelmä vastaa sille asetettuja vaatimuksia. Asetetut vaatimukset riippuvat haettavasta sertifikaatista. Esimerkiksi EMC-sertifikaattia (engl. electromagnetic compatibility certification) hakiessa vaatimukset kohdistuvat laitteen sähkömagneettiseen yhteensopivuuteen (engl. electromagnetic compatibility, EMC). [1]

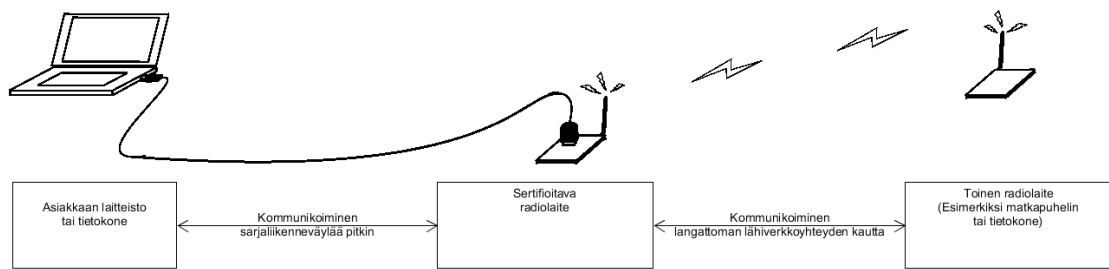
Markkinoilla oleva sertifioimaton tuote ei välttämättä vastaa ostajan olettamuksia. Tuote saattaa toimia oikein, mutta kuluttaja ei voi olla varma, että tuote toimii kuten sen pitää. Esimerkiksi elektroniset laitteet aiheuttavat sähkömagneettista häiriötä itsensä ympärille. Laitteet, jotka ovat toisten elektronisten laitteiden lähellä, joutuvat sähkömagneettisen häiriön kohteeksi. Sähkömagneettinen häiriö saattaa aiheuttaa laitteessa sen normaaliin toimintaan poikkeavaa käytöstä. Sertifioimattoman laitteen ostaja saattaa vasta laitteita käyttäessään huomata, että laitteessa esiintyy häiriötä muiden elektronisten laitteiden läheisyydessä tai se häiritsee muiden laitteiden toimintaa.

Sertifiointeja on monia erilaisia, joista jokaisella on oma soveltuvuusalueensa. Sertifiointi voi asettaa rajoitteita, mitä tuote ei saa tehdä. Esimerkiksi CE-merkin saanut radiotaajuuksia hyödyntävä tuote ei saa aiheuttaa merkitsevästi häiriötä tietyillä radiotaajuusalueilla. Laitteen aiheuttama häiriö saattaa haitata tai estää muiden laitteiden toimintaa, vaikka laite itse toimisikin oikein. Mikäli radiotaajuuksia hyödyntävää laitetta ei ole sertifioitu, saattaa laite itsessään käyttäytyä virheellisesti sen saadessa sähkömagneettista häiriötä ympäristöstä. Vastaavasti toiset laitteet saattavat käyttäytyä ennalta määräämättömällä tavalla sertifioimattoman laitteen läheisyydessä.

### 2.1 Sertifioitava radiolaite ja laitteen ominaisuudet

Diplomityössä käytetty radiolaite on mittatilaustyönä tehty asiakasta varten. Asiakas tarvitsee sellaisen radiolaitteen, jolla pystyy kommunikoimaan toisen radiolaitteen kanssa langattoman lähiverkoryhteyden kautta ja samalla kommunikoimaan myös sarjaliikenneyhteydellä asiakkaan laitteiston kanssa. Kuva 1 havainnollistaa laitteiden välistä kommunikoimista. Lisäksi laitteen tulee olla fyysiseltä kooltaan tarpeeksi pieni, jotta se mahtuu sille varattuun koteloon. Radiolaite tulee toimia vähäisellä virralla, sillä radiolaitteelle syötetään sen tarvitsema virta suoraan asiakkaan laitteistolta sarjaliikenneyhteyttä pitkin.

Markkinoilla ei ollut sellaista tuotetta, joka täyttäisi asiakkaan esittämät vaatimukset. Koska tarvittavaa laitetta ei löytynyt markkinoilla, suunniteltiin ja teetettiin kyseinen radiolaite mittatilaustyönä. Näin asiakkaan esittämiin vaatimuksiin saatiin vastattua. Tämä kuitenkin tarkoittaa, että luotu radiolaite tulee sertifioida, jotta asiakas voi myydä tuotetta



**Kuva 1.** Kuvaus radiolaitteen ja siihen kiinnittyneiden laitteiden välisestä viestinnästä.

eteenpäin. Koska laite on mittatilaustyönä tehty, ei saatavilla ollut sellaista ohjelmaa, jolla sertifiointitestausta voitaisiin suorittaa. Tuotteen sertifiointia varten radiolaitteelle tulee ohjelmoida toiminnallisuus, joka mahdollistaa sertifiointitestiä tekemisen.

Sertifiointia kannalta oleellisia asioita radiolaitteessa on sen käyttämä kommunikointitapa. Tässä työssä sertifioidun radiolaitteen tarkoitus on kommunikoida toisen laitteen kanssa käyttäen langatonta lähiverkkoa viestintäyhteytenä. Langattoman lähiverkon käyttäminen tarkoittaa, että radiolaite käyttää radiotaajuuksia (engl. radio frequency). Laitteen oikeanlainen toiminta sen käyttämällä radiotaajuuksilla tulee varmistaa. Lisäksi radiolaite on elektroninen, joten sen sähkömagneettinen yhteensopivuus tulee myös tarkastaa. Radiolaitetta on tarkoitus myydä Euroopan talousalueella sekä Yhdysvalloissa. Näiden kahden alueen lainsäädäntö vaikuttaa sertifiointiprosessiin.

Fyysiseltä kooltaan radiolaite on tavallisen pelikorttipakan levyinen ja korkeinen, mutta noin kolmasosa pakan paksuudesta. Radiolaitteessa ei ole sisäistä virtalähdettä. Liittiminä laitteessa on ainoastaan yksi D9-liitin. Kommunikoinnista langattomassa lähiverkossa vastaa Texas Instruments-nimisen yrityksen luoma SimpleLink tuoteperheeseen kuuluva mikrokontrolleri. Piirilevyyn kuuluu mikropiiriin lisäksi antenni. Nämä kaksi osaa tekevät piirilevystä radiolaitteen.

Mikrokontrolleri on Texas Instrumentsin puolesta sertifioitu. Kontrolleri omaa modulaariset FCC-, CE-, IC- ja TELEC sertifikaatit. Koska radiolaite on mittatilaustyönä tehty, se täytyy erikseen testata sertifiointia varten. Radiolaitteessa langattomaan kommunikointiin käytettävä mikrokontrolleri on malliltaan CC3200. Mikrokontrolleri sisältää Wi-Fi verkkoprosessorin ja sulautetun TCP/IP pinon. Malli tukee 802.11bng spesifikaatioita, mutta käyttää Internet protokollasta ainoastaan versiota 4. Lähetysteho on 18 desibeliä ja vastaanottoherkkyys -95,7 desibeliä. Mikrokontrolleri on luotu esineiden internet-ajattelumallin pohjalta ja se mahdollistaa kommunikoinnin langattomassa lähiverkossa 2,4–2,4835 GHz taajuusalueella [2]

Radiolaite tullaan liittämään suoraan asiakkaan laitteistoon, tai yhdistämään radiolaite laitteistoon kaapelilla. Laite saa virtansa laitteistosta, johon se on liittynyt, ja virta syötetään radiolaitteelle D9-liittimen läpi. Laitteiden välinen kommunikaatio tapahtuu D9-

liittimen kautta käyttäen RS-485 standardin mukaista sarjaliikenneyhteyttä. Laitteistoon liitetty radiolaite on stationaarinen toimintansa aikana.

## 2.2 EMC-sertifiointin tarkoitus

Markkinoilla oleva sertifioitu tuote ei ole altis sähkömagneettisille häiriöille, eikä myöskään tuota merkittävästi sähkömagneettista häiriötä. Sertifiointi on prosessi, jossa tuotetta testataan ja sen toimiminen todetaan sertifiointin asettamien vaatimuksen mukaiseksi. Tästä merkkinä tuotteelle annetaan sertifikaatti tai sertifiointimerkki. Ostaja tietää, että sertifiointimerkin omaava tuote täyttää kyseisen sertifiointin vaatimukset. Merkki toimii ostajalle lupauksena laitteen oikeanlaisesta toiminnasta. Mikäli tuotetta ei ole sertifioitu, saattaa tuotteen toiminta vaihdella riippuen ympärillä olevista asioista. Esimerkiksi puhelimissa on aikoinaan havaittu heikon sähkömagneettisen yhteensopivuuden takia virheellistä toimintaa.

Yhdysvalloissa Federal Communication Commission (FCC) valvoo radio-, televisio-, kaapeli- ja satelliittikanavien kautta tapahtuvaa kommunikaatiota. Vuonna 1994 FCC raportoi saaneensa 25 000 valitusta vuodessa. Valitukset koskivat puhelimen käyttöä radioasemien lähetyvillä. Radioasemat häiritsivät puhelimen normaalia toimintaa. Aina kun radioasema aloitti lähetyksen, lähellä olevat matkapuhelimet alkoivat toistaa radiolähetyksiä. Puhelimet altistuivat radioaseman aiheuttamalle sähkömagneettiselle häiriölle. Tapaus johti FCC:n tekemään laajempaan selvitykseen, jossa selvitettiin puhelimen sähkömagneettista yhteensopivuutta. Selvityksessä kävi ilmi, että puhelimet voitiin suunnitella niin, ettei radioasemien aiheuttama häiriö vaikuttanut niiden toimintaan. Kuitenkin komission testaamista 241 puhelimesta 68 prosenttia oli häiriölle alttiita. Näiden puhelimen piti valmistajien mukaan toimia häiriöstä huolimatta. Tuolloin häiriötä yritettiin poistaa suodattimilla, joiden toiminta todettiin riittämättömiksi. [3]

Aina sähkömagneettinen säteily ei koske vain pelkästään sähkölaitteita, vaan se vaikuttaa myös ihmisiin. Lisäksi ympäristön muuttuminen ja teknologian kehittyminen saattaa vaatia muutoksia jo olemassa oleviin ratkaisuihin. Vuonna 1998 Australian itärannikolla sijaitsevaan Homebush Bayn alueelle rakennettiin uusi luksusasuntoalue. Alueella oli jo ennestään Australian Broadcasting Corporationin omistuksessa olevia radiomastoja. Noin 14 kuukautta ennen alueen rakentamista viranomaiset huomauttivat, että radiomastot saattaisivat vaikuttaa haitallisesti sähkö- ja elektroniikkalaitteisiin sekä aiheuttaa terveyshaittoja. Vaikutukset ilmenesivät kolmannesta kerroksesta ylöspäin. Yhtenä vaihtoehtona ongelman ratkaisemiseksi pidettiin radiomastojen siirtämistä pois asuinalueen tieltä, mikä saattaisi maksaa jopa 40 miljoonaa dollaria. [4]

Sertifiointin tarkoitus on estää edellä mainittujen kaltaisten tapahtumien syntyminen, vaikka se ei olekaan aina mahdollista. Yllä mainituissa esimerkissä asuntoalueen rakentamista oltiin vastustettu radiomastojen takia, mutta ehdotus oli silti hyväksytty.

## 2.3 Radiolaitetta koskevat direktiivit Euroopassa

Euroopan Unionin yhtenä tarkoituksena on mahdollistaa tuotteiden vapaa liikkuminen markkina-alueella eli Unionin jäsenmaiden välillä. Jotta tämä olisi mahdollista niin jäsenmaiden tulee sitoutua varmistamaan, että jäsenmaissa valmistetut tuotteet täyttävät samat vähimmäisvaatimukset. Euroopan parlamentin yksi tehtävä on luoda tällaisia tuotteiden laatuun liittyviä direktiivejä, jotta vapaa kaupankäynti jäsenmaiden välillä olisi mahdollista.

Täyttämällä Euroopan parlamentin asettamat direktiivit, myös Euroopan Unionin ulkopuoliset tahot saavat tuoda tuotteitaan jokaiseen EU-jäsenmaahan. Merkinä sertifiointiin täyttämisestä tuotteen pakkaukseen tai tuotteeseen lisätään Conformance Européenne sertifiointimerkki eli CE-merkki. Tässä työssä käytettävän radiolaitteen hyväksymisen kannalta vaadittuja direktiivejä ovat muun muassa radiolaitedirektiivi ja sähkömagneettista yhteensopivuutta koskeva direktiivi.

### 2.3.1 Radiolaitedirektiivi

Radiolaitedirektiivi on vuonna 2014 alkunsa saanut direktiivi, joka määrittää radiolaitteiden vaatimustenmukaisuuden. Direktiivi kattaa turvallisuuden, terveyteen, sähkömagneettiseen yhteensopivuuteen ja radiospektrin tehokkaaseen käyttöön liittyvät välttämättömät vaatimukset ja se koskee jokaista laitetta, joka hyödyntää radiotaajuusspektriä.

Ennen radiolaitedirektiiviä oli voimassa radio- ja telepäätelaitedirektiivi, jonka voimassaolo päättyi 13.6.2017. Näin ollen kaikkien 13.6.2017 jälkeen markkinoille tuotujen tuotteiden tulee olla RE-direktiivin (engl. Radio Equipment Directive) mukaisia tuotteita. Radiolaitedirektiivi pätee vain Euroopan talousalueen ja Euroopan vapaakauppajärjestön maissa sekä Turkissa. [5] [6]

Radiolaitedirektiivin mukaan radiolaitteella tarkoitetaan elektroniikka- tai sähkölaitetta, joka tarkoituksenmukaisesti lähettää tai vastaanottaa radioaaltoja radiokommunkoinnin tai satelliittipaikannuksen takia. Radiokommunkoinnilla tarkoitetaan kommunikaatiota-paa, joka hyödyntää radioaaltoja. Satelliittipaikannuksessa objektin jokin ominaisuus määritetään käyttäen radioaaltoja. Esimerkiksi objektin nopeus ja paikka voisivat olla määritettäviä asioita. Radioaalloiksi luokitellaan sähkömagneettiset aallot, joiden taajuus on alle 3000 GHz. [7]

RE-direktiivin kannalta haitallinen häiriö on sellainen häiriö, joka saattaisi haitata radionavigointipalveluiden tai muiden turvallisuuspalveluiden toimimista. Mikäli häiriö vakavasti heikentää, estää tai toistuvasti keskeyttää sovellettavien yhteisön tai kansallisten säännösten mukaista radioviestintäpalvelua, on se myös silloin RE-direktiivin mukaan haitallinen häiriö. [8]



### 2.3.2 Sähkömagneettista yhteensopivuutta koskeva direktiivi

Sähkömagneettista yhteensopivuutta koskevalla direktiivillä tarkoitetaan Euroopan parlamentin asettamaa vaatimusta sähkölaitteiden sähkömagneettiselle yhteensopivuudelle. Direktiivi kattaa kaikki sähkölaitteet tai -asennukset, jotka vaikuttavat toisiinsa lähellä oleviin laitteisiin. Esimerkiksi televisio, matkapuhelin, radio ja kodinkoneet aiheuttavat sähkömagneettista häiriötä ympärillään oleville laitteille. Direktiivin tarkoituksena on varmistaa, että laitteissa ei esiinny sähkömagneettisen häiriön takia kohtuuttomia sivuvaikutuksia, ja että yksittäinen laite ei ole altis sähkömagneettiselle häiriölle, eikä myöskään merkittävästi tuota häiriötä. Radiolaitedirektiivissä on useita viittauksia suoraan EMC-direktiiviin (electromagnetic compatibility directive). EMC-direktiivi määrittelee radiolaitedirektiiviä tarkemmin, mitä sähkömagneettisella yhteensopivuudella tarkoitetaan ja mitä direktiivin alaiselta laitteelta vaaditaan. [9]

RF-laitteista (engl. radio frequency equipments) puhuttaessa sähkömagneettisella yhteensopivuudella tarkoitetaan laitteiston kykyä toimia tyydyttävästi sähkömagneettisessa ympäristössään ilman, että laite aiheuttaa kohtuuttomia sähkömagneettisia häiriöitä muihin samassa ympäristössä oleviin laitteisiin. Sähkömagneettisella häiriöllä tarkoitetaan sähkömagneettista ilmiötä, joka saattaa heikentää laitteiston toimintaa. Sähkömagneettinen häiriö voi olla esimerkiksi kohinaa, jokin muu ei toivottu signaali tai muutos etenemisympäristössä. RF-laitteiston aiheuttaman sähkömagneettisen häiriön tulee olla niin matala, että radio- ja telelaitteet tai muut laitteistot voivat toimia tarkoitetulla tavalla. RF-laitteiden tulee sietää häiriötä. Tämä tarkoittaa sitä, että laite pystyy toimimaan normaalilla tavalla sähkömagneettisen häiriön alaisena. Turvallisuuden parantamisella tarkoitetaan toimia ihmiselämän tai omaisuuden suojaamiseksi. [10]

### 2.3.3 Conformité Européenne sertifiointimerkki

Euroopan talousalueella myytäviltä radiolaitteilta vaaditaan Conformité Européenne -sertifiointimerkki. Tämä merkki tunnetaan yleisemmin CE-merkkinä. CE-merkintää käyttävän tuotteen tulee täyttää Euroopan Unionin turvallisuuteen, terveyteen ja ympäristön suojelemiseen asettamat vaatimukset. CE-merkintä asettaa tuotteelle samat vaatimukset oli tuote tuotettu Euroopan sisällä tai tuotu Euroopan ulkopuolelta. [11]

Valmistajien vastuulla on tehdä vaaditut teknilliset dokumentit, vaatimustenmukaisuustodistukset sekä lisätä CE-merkintä tuotteeseen, jos he haluavat tuoda tuotteen Euroopan talousalueelle myyntiin. Mikäli tuote tuodaan Euroopan ulkopuolelta, maahantuojan vastuulla on varmistaa, että tuotteesta on olemassa kyseessä olevat dokumentit, ja CE-merkintä on merkitty tuotteessa. Nämä dokumentit ja todistukset tulee pystyä pyynnöstä toimittamaan pyynnön esittäneelle taholle. [12]

CE-merkki ei takaa, että Euroopan Unioni tai kolmas osapuoli on todennut tuotteen turvalliseksi. Vaikka kuluttajalle CE-merkki toimii lupauksena toimivasta ja turvallisesta tuotteesta, eivät kaikki markkinoilla olevat CE-merkilliset tuotteet ole aitoja. Joissakin

tuotteissa on väärennetty CE-merkki. Tällaisten tuotteiden osalta ei voida olla varmoja, täyttävätkö ne CE-merkin vaatimukset. [13]

## 2.4 Federal Communication Commission

Federal Communication Commission perustettiin 1934 valvomaan Yhdysvaltojen osavaltioiden välistä radio-, televisio-, satelliitti- ja kaapelikommunikaatiota. Komission tehtäviin kuuluvat muun muassa ylläpitää kilpailua, innovaatiota ja investointeja, tukea valtion taloutta ja uudistaa mediasäännöksiä. Yhdysvalloissa radiotaajuuksia käyttävän tuotteen tulee täyttää FCC:n asettamat vaatimukset, jotta tuotetta voitaisiin myydä Yhdysvaltojen sisällä. FCC hyödyntää neljää laitteen hyväksymisohjelmaa: laitteen verifikaatio, sertifiikaatio, tavarantoimittajan vaatimustenmukaisuustodistus (engl. supplier's declaration of conformity) ja vaatimuksenmukaisuustodistus (engl. Declaration of Conformity). Kuten Euroopan talousalueella myynissä olevissa tuotteissa tulee olla CE-merkki, tulee Yhdysvaltojen alueella tuotteessa olla FCC Declaration of Conformity sertifiointimerkki eli FCC-merkki. [14] [15]

FCC:n säännösten mukaan radiotaajuuksia käyttävät elektroniikka- ja sähkölaitteet, jotka pystyvät tuottamaan radiotaajuuksia esimerkiksi säteilemällä tai johtamalla, vaativat FCC-merkin hankkimisen. Näillä laitteilla on potentiaalinen mahdollisuus häiritä radiotaajuuksia 9 kHz – 3000 GHz välillä. Tällaisten tuotteiden tulee todistaa asetusten noudattaminen käyttäen yhtä FCC:n hyväksymisohjelmaa. Hyväksymisohjelman suoritus vaaditaan, ennen kuin tuotetta voidaan myydä, maahantuoda tai edes käyttää Yhdysvalloissa. [16]

FCC-merkin hankinta vaatii tuotteesta vastuussa olevan osapuolen käyttävän akkreditoitua testauslaboratoriota, joka noudattaa asetettuja mittausprotokollia tuotteen oikeanlaisen toiminnan ja asiallisen teknillisen standardin varmistamiseksi. Kuten CE-merkin kanssa, myöskään FCC-merkkiä hakiessa ei tarvitse toimittaa hakemusta komissioon. Osapuolen on kuitenkin komission pyynnöstä toimitettava testausraportit ja muut dokumentit. Nämä dokumentit todentavat, että tuote noudattaa määrättyjä vaatimuksia. [17]

### 3. RADIOLAITTEEN KÄYTTÖ ERI TESTEISSÄ

Sertifikaatteja ja vaatimuksia on useita erilaisia. Sertifioitava tuote määrä, mitkä eri direktiivit sen tulee täyttää, ennen kuin tuote voidaan sertifioida ja tuoda markkinoille. Vaatimukset voivat olla erilaiset kahden eri tuotteen välillä. Esimerkiksi työssä käsiteltävän radiolaitteen tulee täyttää eri vaatimukset kuin esimerkiksi lasten lelun. Vaatimukset poikkeavat, vaikka molemmat tuotteet hakisivatkin esimerkiksi samaa CE-merkintää.

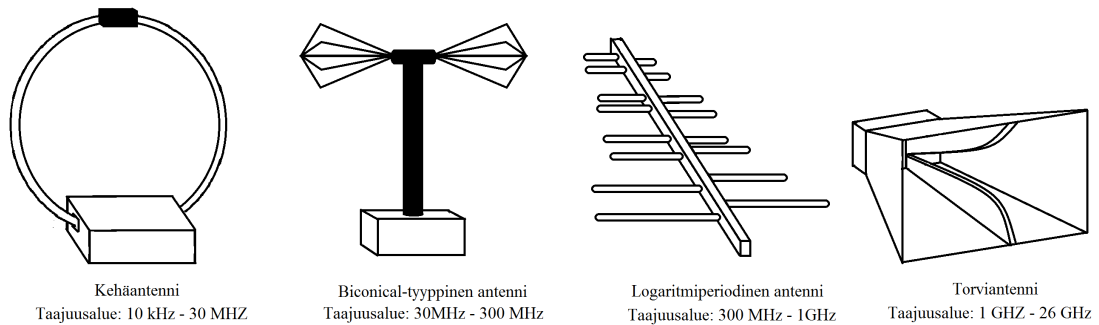
Työn kannalta merkittävin ominaisuus radiolaitteessa on sen ominaisuus käyttää radioaaltoja kommunikointiin. Tämä määrittelee laitteen radiolaitteeksi. Euroopan ja Yhdysvaltojen markkinoille tuotavien radiolaitteiden sertifiointikäytännöt ovat jonkin verran erilaisia, mutta itse sertifiointien vaatimukset ovat jokseenkin samat. Sertifioitavan radiolaitteen ominaisuuksien ja toiminnallisuuden mukaan suoritetaan kolmannen osapuolen toimesta laitteen testaaminen. Testien tulosten tulee asettua vaatimusten asettamiin rajoihin. Mikäli näin ei ole, tulee radiolaitetta muuttaa niin, että testit saadaan läpäistyä.

Vaatimukset riippuvat suoritettavasta testistä. Sertifioimista varten radiolaitteen sähkömagneettinen yhteensopivuus tulee testata. Radiolaitteet säteilevät ympäristöönsä radioaaltoja ja ihmistä lähellä olevat voimakkaat lähettimet saattavat haitata ihmisen terveyttä. Laitteen ominaisabsorptionopeus tulee testata ja sen tulee olla alle vaatimusten. Lisäksi radiolaitteelle tullaan lupaamaan vähimmäiskäyttöikä, joka saadaan arvioitua esimerkiksi ympäristötestauksen tuloksista.

#### 3.1 Sähkömagneettisen yhteensopivuuden testaaminen

Sertifioitava laite tulee todentaa olevan määräysten mukainen. Tätä varten kolmas osapuoli suorittaa laitteen tai laitteiston testaamisen. Sähkömagneettista yhteensopivuutta testataan erillisessä eristetyssä testikammiossa, jossa ei ole testin kannalta ylimääräisiä laitteita häiritsemässä testiä.

Sähkömagneettista yhteensopivuutta testaavia testejä on kahdenlaisia. Ensimmäisessä testaustavassa tarkistetaan laitteen lähettämä sähkömagneettinen häiriö. Tätä testausta kutsutaan säteilevän emission testaamiseksi. Sietotestaus on toinen käytetty testausmenetelmä ja siinä testataan laitteen toimintaa sähkömagneettisen häiriön alaisena. Yksittäinen laite saattaa emittoida sähkömagneettista häiriötä useammallakin tavalla. Sietotestaukset voidaan karkeasti jakaa kahteen osaan. Testeihin, jossa testattavaa laitetta häiritään jatkuvalla häiriöllä ja testeihin, jossa laitetta häiritään lyhyillä purskeilla. [18]



*Kuva 2. Säteilevän emission testaamiseen käytettäviä antennejä.*

### 3.2 Säteilevän emission testaaminen

Yleisin sähkömagneettista yhteensopivuutta testaava testi on säteilevien emissioiden mittaaminen. Säteilevien emissioiden testauksessa mitataan sähkömagneettisen kentän voimakkuutta. Säteilevällä emissiolla tarkoitetaan sellaista säteilyä, jota laite ei tuota tarkoituksella, vaan se syntyy laitteen toiminnan yhteydessä. Tällainen emissio aiheutuu digitaalisessa piirissä muuttuvasta virrasta. Säteilevä emissio tulee olla alle määrättyjen arvojen, sillä muuten emissio saattaa häiritä muita sähkö- tai elektroniikkalaitteita. Jokaisella suurella markkina-alueella on raja, kuinka paljon yksittäinen laite voi emittoida säteilyä. Nämä rajat ovat voimassa riippumatta laitteen tyypistä tai teollisuusalaista. [18]

Sähkömagneettiset aallot eivät leviä laitteesta tasaisesti ympyränmuotoiseksi alueeksi laitteen ympärille vaan ne suuntautuvat tiettyyn suuntaan. Tätä varten testissä käytettävien antennien kokoa ja niiden suuntautumista laitteeseen nähden tulee vaihdella. Antenni poimii sähkömagneettisia aaltoja suoraan laitteelta, mutta se saa niitä myös heijastumina eri pinnoista ja siksi esimerkiksi lattian tulee olla sähkömagneettista säteilyä hyvin absorboiva materiaalia. [18]

Avonainen testiympäristö sekä radioaaltoja heijastamaton huone ovat ensisijaisia mittausympäristöjä säteilevien emissioiden mittaamiseen. Avonainen testiympäristö on yleisin testiympäristö säteilevien emissioiden testaamiseen. Testiympäristössä antennin etäisyys on tyypillisesti 3 m, 10 m tai 30 m testattavasta laitteesta. Kaiuttomassa huoneessa seinät ovat radioaaltoja absorboivaa materiaalia, jolloin sähkömagneettinen säteily ei pääse heijastumaan seinistä ja häiritsemään mittausta. [18]

Säteileviä mittauksia mitattaessa on sellainen tilanne mahdollinen, jossa mitatut arvot ovat CE-standardin määritelmän mukaiset, mutta rikkovat FCC:n standardin vaatimuksia. Jos laite täyttää FCC:n vaatimukset säteilevien emissioiden osalta, täyttää se myös CE:n vaatimukset. Mitattavat taajuusalueet riippuvat laitteen käyttämisestä suurimmasta ja pienimmästä kellotaajuudesta. Mitattava taajuus määrittää myös testissä käytettävän antennityypin. Kuvassa 2 on muutamia erilaisia antennityyppejä, joita voidaan käyttää radiolaitteen testaamiseen. [18]

### 3.3 Johtuvan emission testaaminen

Toinen yleinen sähkömagneettinen testi on johtuvien emissioiden mittaaminen. Jokainen radiolaitte tuottaa sähkömagneettista energiaa. Johtuvalla emissiolla tarkoitetaan, että osa tuosta sähkömagneettisesta energiasta johtuu laitteessa kiinni olevia kaapeleita pitkin toisiin laitteisiin. Yksi tällainen kaapeli voi olla esimerkiksi virtakaapeli. Joskus johtuminen aiheutuu siitä, että piirin sähkömagneettista säteilyä tuottava osa on kytketty suoraan johonkin kaapeliin. Toisinaan jokin muu piirin osa johtaa sähkömagneettista energiaa. Tällöin johtavaan osaan välittyy säteilyä tuottavasta osasta yhteisen impedanssin, sähkö-, magneetti- tai sähkömagneettikentän takia. [19]

Häiriö voi johtua suoraan toiseen laitteeseen, jos laite on kiinni häiriötä tuottavassa laitteessa. Häiriö voi johtua kaapeleita tai johtimia pitkin ja näin säteillä sähkö- ja magneettiaaltoina toisiin laitteisiin, jolloin johtimet voivat toimia vahingossa antennin tavoin. Vaihtovirta aiheuttaa minkä tahansa tapaisessa johtimessa ilmaan sähkö- ja magneettiaaltoja. Näiden aaltojen kulkeuduttua noin kuudesosan niiden aallonpituudesta ne yhdistyvät sähkömagneettiseksi aalloksi. [19]

Johtuvaa emissiota mitattaessa mitataan yleensä 150 kHz – 30 MHz taajuusalue, mutta mittauksissa käytettävä taajuusalue riippuu standardista. Testin toteuttamiseen käytetään alipäästösuodatinta. Suodatin asetetaan yleensä virtalähteen ja testattavan laitteen välille, jotta tiedetään mitä impedanssia testauksessa käytetään. Koska standardi määrittelee sekä käytettävät taajuusalueet että eri impedanssit, ovat valmistajat tehneet eri arvoilla toimivia alipäästösuodattimia. Jokaisen standardin testaamiseen löytyy testaukseen sopiva alipäästösuodatin. [18]

### 3.4 Häiriöiden sietotestaus

Elektroniikka- tai sähkölaite voi joutua käyttöaikansa aikana monenlaisen häiriön vaikutuksen alle. Siksi testilaboratoriot yleensä tarjoavatkin useita erilaisia häiriöiden sietotestauksia. Säteilevien häiriöiden sietotestauksessa testataan laitteen toimintaa, kun laitteen toimintaa häiritään sen normaalin toiminnan aikana käyttäen sähkömagneettista säteilyä. Tällä tavalla testissä simuloidaan toisen laitteen luomaa häiriötä. Johtuvien RF-häiriöiden sietotestauksessa simuloidaan ulkopuolisen laitteen johtumalla tapahtuvaa häiriötä. Tällainen tilanne on esimerkiksi silloin, kun laitteen kaapeleiden vieressä menee toisen laitteen kaapeleita. [18]

Näiden kahden sietotestauksen lisäksi suoritettavia testejä on esimerkiksi ESD-testi (engl. electrostatic discharge testing), transientti- ja syöksyaaltotesti sekä jännitekatkos- ja muuttosietotesti. Näiden lisäksi testattavan laitteen häiriönsietokykyä voidaan testata magneettikentällä tai jännite- ja virtapiikeillä. Magneettikentällä voidaan esimerkiksi simuloida toisen laitteen virtalähteen luomaa häiriötä. [18]

Sähköstaattisessa purkaustestissä laitteen tai laitteiston sellaiset näkyvät osat, joita ihminen pystyisi koskea, altistetaan sähköstaattiselle purkaukselle. Transientti- ja syöksyaalto-

testauksessa simuloidaan ympäristön muuttuvia induktiivisia kuormia ja niiden aiheuttamaa häiriötä. Jännitekatkos- ja muutossietotestauksessa simuloidaan eri jännitehäiriötä ja testataan niiden vaikutusta laitteen toimintaan. Jännite- ja virtapiikkejä tapahtuu laitteen normaalissa käytössä esimerkiksi silloin, kun laite sammutetaan tai käynnistetään, laite kuormittuu yllättäen, sulake rikkoutuu tai salama aiheuttaa virtapiikin laitteeseen. [18]

### 3.5 Ympäristötestaus

Laitteen sähkömagneettisen yhteensopivuuden lisäksi tulee itse laitekin fyysisesti testata. Yksittäinen laite joutuu elinkaarensa aikana altistumaan monille eli kuormittaville tekijöille, kuten satunnaisille iskuille, tärähdyksille tai lämpötilojen vaihteluille. Mikäli laite on huonosti suunniteltu, se ei välttämättä kestä elinkaarensa aikana saamaansa kuormitusta ja hajoaa enneaikaisesti. Testattavat asiat riippuvat laitteesta itsestään, mutta myös laitteen käyttöympäristöstä. Käyttöympäristö määrittää, minkälaiselle kuormitukselle laite tulee joutumaan. Laitteen tavoiteltu käyttöikä puolestaan kertoo, kuinka paljon laitteen pitäisi kestää kuormitusta hajoamatta.

Ympäristötestauksessa voidaan testata laitteen kestävyyttä esimerkiksi iskuilla, värinällä ja lämpötilavaihtelulla. Esimerkiksi jos matkapuhelimen omistaja vahingossa tiputtaa laitteensa, joutuu matkapuhelin elinkaarensa aikana kestävänsä iskuja eri suunnasta. Matkapuhelimen kokemat lämpötilavaihtelut saattavat olla hyvinkin suuria, sillä puhelinta voidaan haluta myydä sekä lämpimillä että kylmillä alueilla, jolloin näiden kahden paikan lämpötiloissa saattaa olla suuriakin eroja. Laitteen tulisi toimia kummallakin alueella normaalisti, vaikka lämpötilat vaihtelevat pakkasesta hellelukumisiin. Laitteen kestävyyttä on helppo testata kuormittamalla sitä lyhyen aikaa, mutta pitkäkestoiset testit vaativat resursseja ja aikaa. Laitteen kestävyyttä sen koko elinkaaren aikana voi olla vaikea määrittää.

### 3.6 ALT- ja HALT-testaus

Laitteen suunnitteluvirheiden, viallisten komponenttien, ja virheellisen käyttäymisen havaitsemiseksi voidaan käyttää HALT- (engl. highly accelerated life testing) ja ALT-testejä (engl. accelerated life testing). Esimerkiksi tavallisuudesta poikkeava lämpötila tai koskeus voivat aiheuttaa laitteessa pysyviä toimivuuden alentumista tai jopa hajottaa laitteen. Hajoaminen voi tapahtua, vaikka laite toimisi tavallisissa olosuhteissa normaalisti. Usein laitteille arvioidaan elinaika, jonka ajan laitteen tulee toimia. Kuluttajille voidaan näin antaa arvio laitteen käyttöiästä. Laitteiden elinajat saattavat olla useita vuosia. Tuona aikana ne altistuvat erilaisille rasituksille. [20]

ALT-testauksen tarkoituksena on arvioida laitteen kestävyyttä pitkällä aikavälillä. Laitetta kuormitetaan testin aikana ja käytetystä kuormituksesta lasketaan testille kerroin. Kerroin kertoo, paljonko aikaa säästetään kyseistä testiä ajettaessa. Kertoimen ollessa esimerkiksi kymmenen vastaa yhden tunnin mittainen testiajo kymmentä laitteen normaalia ajotuntia. Testissä laitteelle aiheutetaan kuormitusta äärimmäisellä lämpötilalla, lämpöshokilla,

tärinällä, mekaanisella iskulla, kosteudella, muuttuvalla virralla ja jännitteellä sekä muuttamalla laitteen aktiivisuutta. Näitä kuormituksia voidaan myös yhdistää, jotta voidaan saavuttaa suurempi kerroin testille. Suuremmalla kertoimella laitetta voidaan testata esimerkiksi kuukausi ja todeta sen kestävänsä kymmenen vuoden kuormitus, mikäli saatu kerroin testille on ollut tarpeeksi suuri. [20]

HALT-testauksessa laitetta kuormitetaan yhdellä tai usealla aikaisemmin esitetyistä tavoista. Kuormitusta lisätään asteittain, kunnes ennalta määritelty maksimiraja täyttyy tai laite menee rikki. Maksimiraja voidaan valita sen mukaan, milloin laite on aikaisemmin mennyt kyseisessä testissä rikki, tai laskennallisesti laitteessa käytettyjen komponenttien ylärajoista. Jokaisen rikkoutuneen laitteen kohdalla analysoidaan, mitä laitteesta rikkoutuu ja miksi. Vika saattaa johtua huonosta komponentista tai suunnitteluvirheestä. Laitetta voidaan parantaa löytämällä laitteen heikot kohdat. Testien tuloksien analysointi saattaa olla haastavaa, sillä laitteen hajoaminen ei aina aiheudu yhdestä komponentista. Lisäksi samassa testissä laitteet voivat hajota eri syistä. Testissä on saatettu kuormittaa laitetta kovempaa, kuin mitä se lopullisessa käyttöympäristössään joutuisi kestävänsä. [20]

Jokainen laite rikkoutuu ajan saatossa. HALT- ja ALT-testeillä varmistetaan, että laite fyysisesti kestävänsä sille asetetut vaatimukset. HALT-testaamisessa on turha kiinnittää liikaa huomiota sellaisiin rikkoutuneisiin laitteisiin, jotka rikkoutuvat maksimirajan ylittyessä.

### 3.7 Radiolaitetestaus

Jokainen radioaaltoja kommunikoimiseen käytävä radiolaitte säteilee ympäristöönsä radioaaltoja. Radioaallot puolestaan sisältävät energiaa, ja törmätessään eri materiaaleihin luovuttavat energiansa törmäämäänsä materiaaliin. Riippuen materiaalista sekä radioaallon taajuudesta ja amplitudista tapahtuu materiaalissa törmäyksen takia muutoksia. Radioaaltojen tapauksessa tämä muutos on yleensä energian siirtyminen radioaallosta materiaaliin. Materiaalin saadessa radioaallon energian sen lämpötila kasvaa. Mitä tehokkaampaa lähetintä radioaaltojen lähettämiseen käytetään, sitä suurempi energia radioaallossa on.

Ominaisabsorptionopeudeksi (engl. specific absorption rate, SAR) kutsutaan sitä nopeutta, jolla radiotaajuuksissa oleva energia häviää materiaaliin jokaista massayksikköä kohti. Radiotaajuuksien aiheuttamaa lämpötilan nousua tulee rajoittaa, jotta käyttäjän kehon lämpötila ei lähde nousemaan liikaa radiotaajuuksien takia. Kehon lämpötilan nousemista on vaikea mitata. Mittausta hankaloittavat yksilöiden erilaiset fysiologiset parametrit, jotka vaikuttavat mittaukseen. Tätä varten radiotaajuuksia lähettävän laitteen maksimaalinen SAR-arvo on määritetty eri standardeissa ja direktiiveissä. Maksimiarvo riippuu sertifioidusta laitteistosta. [21]

Radiolaitteita sertifioidessa tulee tutkia, kuinka paljon radiolaitteen radioaalloista imeytyy energiaa ihmiskehoon. Euroopassa ja Yhdysvalloissa käytettävät ylärajat eivät ole samoja ja molemmilla alueilla on omat standardinsa, jotka määrittelevät SAR-arvolle ylärajan.

Yhdysvalloissa FCC on määritellyt ylärajan SAR-arvoille. Esimerkiksi matkapuhelimien SAR-arvon yläraja on 1,6 wattia per kilogramma. Matkapuhelimien on täytettävä tämä vaatimus, jotta puhelinta voitaisiin laillisesti myydä turvallisena tuotteena. Vastaava luku on 2 W/kg Euroopan Unionin alueella ja se on määritelty standardissa IEC 62209-1:2016. [22] [23]

SAR-testeissä selvitetään, paljonko ihmiskeho absorboi radiotaajuuksissa olevaa energiaa. Standardi EN 50566-2017 koskee ihmisen lähellä olevien langattomien kommunikaatio-laitteiden aiheuttaman sähkömagneettisen säteilyn rajoittamista 30 MHz – 6 GHz taajuusalueella. Standardi koskee kädessä pidettäviä tai kehoon kiinnitettäviä laitteita sekä laitteita, jotka ovat 200 mm lähempänä ihmisen kehoa. Näitä asioita koskee myös standardi IEC 62209-2:2010, joka ohjeistaa myös kuinka SAR arvot voidaan mitata niin, että mittaus on myöhemmin toistettavissa. [24] [25]

Kansainvälinen säteilysuojelutoimikunta (engl. International Commission on Non-ionizing Radiation Protection, ICNIRP) julkaisi vuonna 1998 ohjeistuksen, jonka tavoitteena on rajoittaa sähkömagneettisille kentille altistumista ja näin suojella sähkömagneettisen säteilyn aiheuttamilta terveydelle haitalliselta vaikutuksilta. Altistus aiheuttaa terveyshaittoja yksilöissä, mutta myös heidän jälkeläisissään. Ohjeistuksessa määritetään altistamisraajat, joiden ylittäminen aiheuttaa terveyshaittoja. Ohjeistus koskee sähkö-, magneetti- ja sähkömagneettisia kenttiä aina 300 GHz asti. [26]

Yhdysvalloissa radiolaitteen säteilyyn liittyvät terveysvaatimukset löytyvät liittovaltion ohjesäännöistä (engl. the Code of Federal Regulations) aihe 47, luku 1, kohta 2, pykälä 2.1093. Pykälä käsittelee radiotaajuuksien säteilylle altistumista ja koskee kannettavia laitteita. Kannettavaksi laitteeksi määritellään radiotaajuuksia lähettävä laite, jota käytetään enintään 20 cm etäisyydellä ihmiskehosta. SAR-arvot on IEEE (engl. Institute of Electrical and Electronics Engineers) määritellyt standardissa ANSI/IEEE C95.1-2005 kohdassa 4.2. [27] [28]

IEEE 1528-2013 Standardi käsittelee, miten SAR-arvot voidaan mitata. Standardissa kuvataan mittauskonsepti, mittaustekniikat, mittaukseen tarvittavat välineet, kalibraatiotekniikat ja käytetyn keinosolukon mallit sisältäen anatomisesti ihmisen pään kanssa olevan yhtenäisen mallin. [29]



## 4. SERTIFIOINTITESTIEN SUORITTAMISEEN VAADITTU TOIMINNALLISUUS

Radiolaitteen testaaminen testilaboratoriossa edellyttää, että radiolaitteella pystytään tekemään kaikki tarvittavat testitapaukset. Testauksen mahdollistavat ominaisuudet tulee ohjelmoida radiolaitteelle. Radiolaitteen varsinaista ohjelmaa ei voida käyttää, sillä se on liian spesifi testien tekemiseen, eikä varsinaisella ohjelmalla saataisi tehtyä kaikkia testitapauksia. Testien suorittamiseksi vaaditut ominaisuudet tulevat pystyä tekemään nopeasti, mieluiten yhtä nappia painamalla. Käynnistetyistä toiminnallisuuksista tulee saada selkeä indikaatio testien suorittajalle. Tämä tarkoittaa, että radiolaitetta kontrolloidaan erillisen käyttöliittymän avulla tietokoneelta. Käyttöliittymän tulee olla niin selkeä, että testien tekijä onnistuu sen avulla saamaan radiolaitteella käyntiin halutun testin tai toiminnallisuuden ilman erillistä neuvontaa. Käyttöliittymän käytöstä tulee kirjoittaa kattavat ohjeet, mikäli käyttöliittymän käyttöä ei saada tarpeeksi intuitiiviseksi.

Jotta radiolaitteella voitaisiin testata, tulee sillä tekemään seuraavat toiminnallisuudet:

- kirjoittamaan dataa radiolaitteen flash-muistille
- lukemaan dataa radiolaitteen flash-muistilta
- luomaan radiolaitteesta tukiasema, johon toinen radiolaitte voi yhdistyä
- kommunikoidaan tietokoneen kanssa
- luomaan yhteyden jo olemassa olevaan tukiasemaan
- lähettämään ja vastaanottamaan dataa langattoman lähiverkkoyhteyden ylitse
- lähettämään ja vastaanottamaan jatkuvasti dataa ilman langatonta lähiverkkoyhteyttä
- lähettämään verkkoprosessorilla dataa jatkuvasti ilman langatonta lähiverkkoyhteyttä niin, että käytetyn kanavan käyttöaste on vähintään 85 %.

Radiolaitteella tulee olla erillinen testaukseen tarkoitettu ohjelma. Tämän testaukseen tarkoitetun ohjelman pitää käynnistyä automaattisesti, kun kolmas osapuoli testaa radiolaitetta. Radiolaitteelta ei saa käynnistyä varsinaisen ohjelman, joka on räätälöity asiakasta varten. Asiakkaalle räätälöidyn ohjelman toiminta ei sellaisenaan sovellu testaustarkoitukseen.

## **4.1 Datan lähettäminen ja vastaanottaminen langattoman lähiverkkoyhteyden ylitse**

Radiolaitteen normaaliin toimintaan kuuluu kommunikointi yhden muun radiolaitteen kanssa. Datan lähettäminen tapahtuu molempiin suuntiin vuorovaikutteisesti. Kun radiolaitteen on ottanut vastaan toiselta laitteelta datan, lähettää se seuraavaksi vastauksen. Vastaavanlainen käytös tulee olla myös yhdistetyssä laitteessa. Tämä helpottaa laitteiden välisen kommunikaation rakentamista.

Vastaanottaminen ja lähettäminen vuorottelevat tasaisesti keskenään. Sellaista tilannetta ei pääse syntymään, jossa molemmat radiolaitteet olisivat lähettämässä dataa samaan aikaan. Radiolaitteen tulee vastata jokaiseen saamansa pakettiin, muutamaa erikseen määritettyä poikkeusta lukuunottamatta. Kommunikoinnin lopettaminen tulee tapahtua testatta-vaan radiolaitteeseen yhdistyneen laitteen kautta ja se tapahtuu yksinkertaisesti olemalla vastaamatta radiolaitteen lähettämään viestiin. Radiolaitteen ei tule tietää toisen, siihen yhdistyneen, laitteen tilaa, vaan sen tulee huolehtia ainoastaan omasta sisäisestä tilastaan ja vastata saamiinsa viesteihin.

Dataa tulee pystyä lähettämään valitulle kanavalle ja käytettävää kanavaa pitää pystyä vaihtamaan testien aikana. Kanavan muuttaminen tarkoittaa yhteyden katkaisua ja uuden yhteyden luomista. Tämäkin asia tulisi tapahtua mahdollisimman automaattisesti ja yksinkertaisesti. Yhteyden ylitse lähetetyistä ja vastaanotetuista paketeista tulee lähettää laitteen käyttäjälle viestinä tieto saadusta tai lähetetystä paketista. Myös yhteyden muodostamisesta tulee lähettää viesti käyttäjälle.

## **4.2 Datan jatkuva lähettäminen ja vastaanottaminen langattoman lähiverkkoyhteyden ylitse**

Radiolaitetta testattaessa testauskammiossa ei saa olla muita testin kannalta välttämättömät radiolaitteet. Ylimääräiset radiolaitteet aiheuttavat testissä häiriötä, jolloin testin tuloksia tulkitessa ei voitaisi olla täysin varmoja, paljonko toinen radiolaitte on vaikuttanut mittaustulokseen. Radiolaitteen normaaliin toimintaan kuuluvat langattoman lähiverkkoyhteyden luominen ja kommunikointi muodostettua yhteyttä pitkin toiseen laitteeseen. Kaikkien testien aikana tämä ei kuitenkaan ole mahdollista. Yhteyden muodostaminen vaatisi radiolaitteen, johon yhteys muodostettaisiin. Radiolaitteen tulee pystyä lähettämään ja vastaanottamaan jatkuvasti ennalta määrättyä viestiä langattomien lähiverkkon kanavien taajuusalueilla ilman, että radiolaitteen tarvitsee yhdistyä johonkin toiseen laitteeseen.

Testaajan tulee pystyä valitsemaan haluttu kanava, johon viestejä tullaan testin aloitettua lähettämään sekä nopeus, jolla viestejä lähetetään. Nämä tulee pystyä valitsemaan kaikista radiolaitteen tukemista lähetysopeuksista. Valintojen jälkeen testaaja voi käynnistää testin. Jatkuvan yhteydettömän lähetyksen tulee testin aikana tapahtua täydellä teholla.

Tällöin verkkoprosessorin ei tule tehdä mitään muuta, kuin lähettää ennalta määrättyä datapakettia niin nopeasti kuin mahdollista. Verkkoprosessorin tulee lähettää viestejä niin tiheästi, että käytettävän kanavan käyttöaste on yli 85 %. Testin tekijälle ei tarvitse tulla jokaisesta lähetetystä paketista ilmoitusta. Tähän prosenttiosuuteen olisi vaikea päästä, mikäli jokaisesta lähetetystä viestistä pitäisi erikseen tiedottaa testin tekijälle. Yksi ilmoitus testin alkamisesta tekijälle riittää. Verkkoprosessorin ei tarvitse testin aikana tehdä mitään muuta kuin lähettää viestiä. Tämä tarkoittaa sitä, että testin aloittamisen jälkeen verkkoprosessoria ei tarvitse ohjelmallisesti pystyä sammuttamaan. Radiolaite voidaan sammuttaa, kun testi halutaan lopettaa.

Jatkuva datan vastaanottaminen tulee tapahtua vastaavasti kuin datan jatkuva lähettäminenkin. Testin käynnistyessä testin tekijän tulee saada ilmoitus testin aloittamisesta, jonka jälkeen radiolaite jää vastaanottamaan dataa. Radiolaitteen tulee vastaanottaa data kaikilta käyttämiltään kanavilta ja jokaisella tukemallaan nopeudella. Testin aikana ei vastaanotetuista paketeista tarvitse tulla mitään ilmoitusta testin tekijälle. Verkkoprosessorin tulee ottaa viestejä vastaan niin kauan, kunnes laite käynnistetään uudestaan.

### **4.3 Muodostetun yhteyden tarkkaileminen**

Radiolaitetta testattaessa varmistetaan, ettei laite ole altis sähkömagneettiselle häiriölle. Radiolaitteen tulee pystyä jatkamaan normaalia toimintaansa sähkömagneettisesta häiriöstä huolimatta. Tällainen testi voidaan suorittaa luomalla yhteys kahden eri radiolaitteen välille, lähettämällä jatkuvasti viestejä toiselta radiolaitteelta toiselle ja altistamalla samalla laitteet sähkömagneettiselle häiriölle. Muodostettu yhteys ei saa katketa, ja jokainen lähetetty viesti tulee saapua vastaanottajalle.

Tällaisessa testissä testikammiossa tarvitaan poikkeuksellisesti kaksi radiolaitetta. Toinen laite toimii tukiasemana ja toinen yhdistyy luotuun tukiasemaan. Testin valmisteluissa toinen radiolaite asetetaan tukiasemaksi. Kun radiolaite on tehnyt itsestään tukiaseman ja on valmis palvelemaan asiakasta, tulee sen ilmoittaa tämä testin tekijälle. Seuraavaksi toinen laite voidaan yhdistää luotuun palvelimeen. Testaajalle tulee tulla ilmoitus, kun laitteet ovat saaneet muodostettua yhteyden välillensä ja palvelin on valmis kommunikoidaan asiakkaan kanssa. Molempien radiolaitteiden tulee olla identtisiä ja laitteiden erilainen toiminta tulee luoda ohjelmallisesti. Radiolaitteiden toiminta tai testin tulos ei saa muuttua, vaikka asiakas-palvelin-mallin mukaiset roolit vaihtuisivat radiolaitteiden välillä.

### **4.3.1 Radiolaitteen asettaminen tukiasemaksi ja yhdistäminen tukiasemaan**

Tukiasemaksi asetettavalle radiolaitteelle tulee antaa tieto käytettävästä verkkotunnuksesta (engl. service set identifier, SSID), sekä käytettävästä salasanasta, ja nämä tiedot annetaan käyttöliittymän kautta. Kun testaaja haluaa asettaa radiolaitteen tukiasemaksi, hän kirjoittaa käyttöliittymään käytettävän verkkotunnuksen ja salasanan. IP-osoitetta (engl. internet protocol address), porttia tai kanavaa ei tarvitse käyttöliittymässä antaa. Tämän jälkeen testaaja painaa ”Set dongle as AP” -painiketta, jolloin tukiasema luodaan sillä verkkotunnuksella ja salasanalla, jotka käyttäjä syötti käyttöliittymään. Onnistuneesta luomisesta tulee tulla ilmoitus käyttäjälle.

Jotta radiolaitte voisi yhdistyä tukiasemaan, tulee sen tietää tukiaseman verkkotunnus sekä mahdollinen salasana. Nämä tiedot tulee pystyä antamaan laitteelle käyttöliittymän kautta. Syötettyään tiedot, käyttäjä painaa ”Connect to dongle” -painiketta. Tällöin laite yhdistyy siihen tukiasemaan, jonka verkkotunnus ja salasanan käyttäjä on sille antanut. Yhteyden muodostumisesta tulee tulla ilmoitus käyttäjälle.

### **4.3.2 Luodun yhteyden katkeamattomuuden varmistaminen**

Kun kahden radiolaitteen välille on saatu luotua yhteys, voidaan luotua yhteyttä alkaa testaamaan. Testin aikana tulee asiakkaan lähettää sekunnin välein pyyntöjä palvelimelle. Palvelin välittää viestit yhdistyneeseen tietokoneeseen, joka pitää pyynnöistä kirjaa. Palvelin ei vastaa saamiinsa viesteihin lainkaan.

Tietokoneen tulee ilmoittaa testaajalle, jos palvelin ei saa sekunnin välein viestiä toiselta radiolaitteelta. Tällä tavalla testaaja tietää, onko yhteys katkennut laitteiden väliltä vai ei. Yhteyden katkeamisen ja uudelleen muodostumisen jälkeen tietokone ilmoittaa testaajalle, kuinka monta pakettia palvelimelta jäi saamatta katkon takia.

## **4.4 Radiolaitteen flash-muistille kirjoittaminen**

Normaalin käytön aikana radiolaitteen flash-muistille voidaan kirjoittaa dataa. Sähkömagneettinen häiriö voi kuitenkin aiheuttaa sen, että flash-muistille kirjoitettava data korrumpoituu. Tällöin muistilta luettu data ei ole enää sama kuin kirjoitettu. Testaajalla tulee olla mahdollisuus kirjoittaa laitteen muistille dataa käyttöliittymän avulla. Käyttöliittymässä tulee olla kenttä, johon käyttäjä voi kirjoittaa haluamansa datan, sekä toinen kenttä, johon luetaan käyttäjän aikaisemmin kirjoitettu data. Kirjoitettavan datan ei tarvitse olla määrältään suuri. Normaalin kirjoittamisen ja lukemisen lisäksi käyttöliittymässä tulee olla tuki datan jatkuvalla kirjoittamiselle ja lukemiselle testin helpottamiseksi. Datan kir-

joittamiseen radiolaitteelle on käytännössä kaksi eri vaihtoehtoa. Data voidaan kirjoittaa luomalla langaton lähiverkkoyhteys ja kirjoittaa haluttu data luodun yhteyden kautta. Toisessa vaihtoehdossa data voidaan kirjoittaa laitteelle sarjaliikenneväylän kautta. Testaajalla tulee olla mahdollisuus päättää, mitä dataa radiolaitteelle kirjoitetaan. Ensimmäinen vaihtoehto ei ole mahdollinen, sillä ylimääräisen radiolaitteen tuominen testikammioon ei ole sallittua. Lisäksi langattoman lähiverkkoyhteyden kautta kirjoittaessa olisi riski, että aiheutettu sähkömagneettinen häiriö häiritsisi langatonta lähiverkkoyhteyttä ja sitä varten kirjoittaminen radiolaitteelle epäonnistuisi. Laitteen sarjaliikennekaapelina voidaan käyttää suojattua kaapelia, joka ei ole häiriöaltis. Data säilyy korruptoitumattomana todennäköisemmin suojatussa kaapelissa kuin lähetettäessä langattoman lähiverkkoyhteyden kautta.

## 4.5 Jatkuva kirjoittaminen flash-muistille

Yksittäinen flash-muistille kirjoitus onnistuu helposti käyttöliittymän kautta. Yksittäisten kirjoituskertojen toistaminen manuaalisesti on hidasta. Mahdollinen normaalista poikkeava toiminta sähkömagneettisen häiriön aikana saadaan paremmin esille, kun radiolaitteen muistille kirjoitetaan mahdollisimman nopeasti dataa. Datan nopea kirjoittaminen kuitenkin tarkoittaa, että manuaalinen datan tarkastelu tulee olemaan työlästä, eikä testaaja ehdi vertaamaan kirjoitettua ja luettua dataa mahdollisten virheiden huomaamiseksi. Tämän takia käyttöliittymässä tulee olla tuki jatkuvalla muistiin kirjoittamiselle sekä logiikka kirjoitetun ja luetun datan automaattiselle vertailulle.

Käyttöliittymässä tulee olla kenttä, jonka avulla voidaan määritellä muistiin kirjoitettava data. Lisäksi käyttöliittymässä tulee olla painike, jota painamalla jatkuva muistille kirjoittaminen voidaan aloittaa. Jokainen epäonnistunut kirjoituskerta tulee tulla myös näkyviin käyttöliittymässä olevaan kenttään. Epäonnistuneiksi kirjoituskerroiksi katsotaan sellaiset yksittäiset kirjoitus- ja lukemisoperaatiot, joissa luettu data ei ole sama kuin kirjoitettu. Jokaisesta epäonnistuneesta testitapauksesta tulee näkyä kirjoitettu ja luettu data.

## 4.6 Radiolaitteen maksimikuormittaminen

Radiolaitetta tulee pystyä testaamaan tilanteessa, jossa sitä kuormitetaan ohjelmallisesti mahdollisimman paljon. Tällaisessa testauksessa simuloidaan laitteen toimintaa äärimmäisessä käytössä, jota harvoin normaalissa käytössä tapahtuu. Radiolaitteen maksimikuormittaminen saadaan aikaan yhdistämällä jatkuva muistiin kirjoittaminen sekä datan jatkuva lähettäminen yhteydettömästi.

Verkkoprosessoria saadaan kuormitettua lähettämällä dataa jatkuvasti mahdollisimman paljon. Vastaavasti jatkuvalla datan kirjoittamisella ja lukemisella laitteen flash-muistilta saadaan laitteen prosessoria käytetään mahdollisimman paljon. Laitteella pitää olla mahdollista ajaa näitä kahta testitapausta yhtä aikaa maksimikuormittamisen saavuttamiseksi.

## 4.7 Käyttöliittymän kenttien lokittaminen

Testauksen aikana testaaja tulee saamaan tietoa laitteen tilasta, laitteen lähettämistä ja vastaanottamista viesteistä. Tietokoneelta radiolaitteelle lähetetty tai siltä vastaanotettu viesti tulee näkyviin käyttöliittymään. Vastaavasti radiolaitteen lähettämä vastaus tulee tulla näkyviin käyttöliittymään. Testin toistettavuuden takia radiolaitteen ja tietokoneen käymä viestittely tulee lokittaa.

Lokittaminen mahdollistaa viestien tarkkailemisen myös jälkikäteen. Senkin jälkeen, kun itse testaus on jo suoritettu. Testisekvenssi voidaan tarkistaa myöhemmin tai se voidaan toistaa. Sähkömagneettisen häiriön takia radiolaitteen normaalista poikkeavat virheet ja tilat eivät välttämättä tapahdu kuitenkaan samassa kohdassa testisekvenssiä, vaikka jo kerran ajettu testisekvenssi toistettaisiinkin.

## 5. RADIOLAITTEELLE OHJELMOITU TOIMINNALLISUUS

Edellisen luvun asettamien vaatimusten pohjalta radiolaitteen ohjelman suunnittelu pystyttiin aloittamaan. Vaatimukset eivät kuitenkaan kattaneet kaikkia mahdollisia ohjelmoitavia asioita, vaan työn toteutusvaiheessa oli mahdollisuus kehittää itse omia tapoja täyttää asetetut vaatimukset. Yksi tällainen vaatimus oli esimerkiksi kommunikoinnin tekeminen radiolaitteen ja testauksessa käytettävän tietokoneen välille. Vaatimuksissa määriteltiin, että laitteen kanssa pystyytään kommunikoimaan sekä langattoman lähiverkkoyhteyden kautta että sarjaliikenneväylää pitkin. Vaatimuksessa ei kuitenkaan otettu kantaa siihen, millaisia itse viestit ovat ja liittyisikö niiden lähettämiseen esimerkiksi viestien salaamista.

Texas Instruments on valmistanut laitteessa käytettävän verkkoprosessorin ja tarjoaa prosessorin ohjelmoimiseen tarkoitetun ohjelmiston kehitysympäristön (engl. software development kit). Paketti tarjoaa vaaditut ohjelmatiedostot ohjelmointirajapinnan käyttämiseen. Tämän lisäksi ohjelmiston kehitysympäristö tarjoaa laajan valikoiman eri esimerkiohjelmiä lähdekoodeineen, joita voidaan käyttää apuna ohjelmoimisessa. Lisäksi Texas Instrumentsin kehittämää UniFlash-ohjelmaa käytetään binääritiedostojen kirjoittamiseen radiolaitteelle. Kirjoitus ei siis tapahdu laitteella olevalla ohjelmalla, vaan ulkoisen laitteen, sekä UniFlash-ohjelman avulla.

Radiolaitteella on tehdyn testiohjelman lisäksi muitakin ohjelmia. Radiolaitteessa oleva alkulatausohjelma päättää, minkä ohjelman se käynnistää radiolaitteessa. Alkulatausohjelman toiminta on automatisoitu, eikä testiohjelmassa tarvitse huomioida alkulatausohjelman olemassaoloa.

## 5.1 Toteutettu toiminnallisuus käyttäen C-ohjelmointikieltä

Testiohjelma toteutettiin radiolaitteelle käyttäen C-ohjelmointikieltä. Suurimpia syitä valittuun kieleen oli se, että Texas Instrumentsin julkaisema ohjelmiston kehitysympäristö ja sen sisältämät esimerkit on tehty samalla ohjelmointikielellä. Ennen työn aloittamista radiolaitteelle oli teetetty asiakassovellus, joka oli myös tehty C-ohjelmointikielellä. Saman kielen valitseminen testiohjelman tekemiseen oli luonnollinen valinta ja mahdollisti ohjelmien yhtenäisyyden. Saman ohjelmointikielen valinta myös mahdollisti jo olemassa olevien ratkaisujen käyttämisen testiohjelmassa. Joitain ohjelman osia pystyttiin suoraan ottamaan aikaisemmin toteutetusta asiakassovelluksesta. Tällaisia suurempia kokonaisuuksia olivat esimerkiksi sarjaliikennekommunikaatioon käytettävät metodit ja testiohjelman alussa tapahtuva laitteen alustaminen.

Työn alussa testiohjelma yritettiin pitää yksinkertaisuuden vuoksi yksisäikeisenä. Tämä ei kuitenkaan ollut mahdollista, sillä esimerkiksi langattoman lähiverkkoyhteyden luomisen jälkeen yhteyttä piti pystyä kuuntelemaan aktiivisesti ja reagoimaan saatuihin viesteihin. Rinnakkaisuus ja monisäikeisyys tuli nopeasti testiohjelman toimintaan mukaan. Sulautetussa ympäristössä rinnakkaisuuden mahdollisti FreeRTOS-käyttöjärjestelmä. FreeRTOS-käyttöjärjestelmä on sulautettuun ympäristöön erikoistunut vapaalla lisenssillä oleva käyttöjärjestelmä. Sillä pystyttiin luomaan tarvittavat säikeet, asettaamaan säikeille haluttu prioriteetti sekä käynnistämään vuorontaja. Vuorontaja antaa ajoaikaa jokaiselle säikeelle sen prioriteetin ja tarpeen mukaan.

## 5.2 Toteutuksen arkkitehtuuri

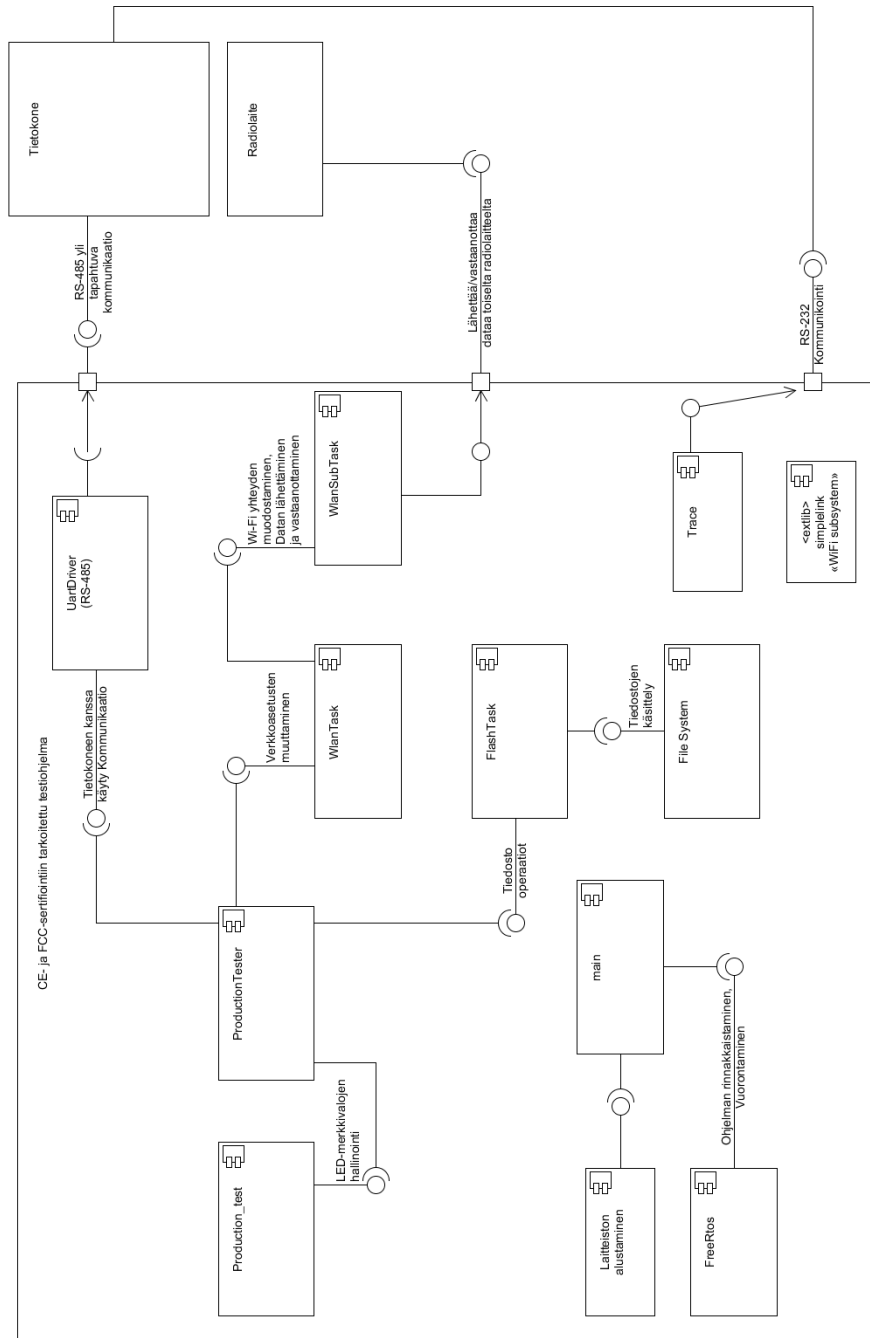
Toteutuksen aikana testiohjelmaa laajennettiin osa osalta. Aluksi testiohjelman ei pitänyt pystyä kuin vastaamaan yksinkertaisiin viesteihin. Ohjelma jaettiin neljään erilliseen osakokonaisuuteen, joilla jokaisella oli oma tehtävänsä. Lopullisesta arkkitehtuurista pystyy helposti huomaamaan nämä kokonaisuudet. Testiohjelman arkkitehtuuri on esitetty komponenttikaaviona kuvassa 3.

Testiohjelmassa on neljä erillistä säiettä vuorontajan käynnistämisen jälkeen. Nämä neljä ohjelman suurempaa osaa muodostavat testiohjelman selkärangan. Ohjelman osat ovat:

- ProductionTester
- WlanTask
- WlanSubTask
- FlashTask.

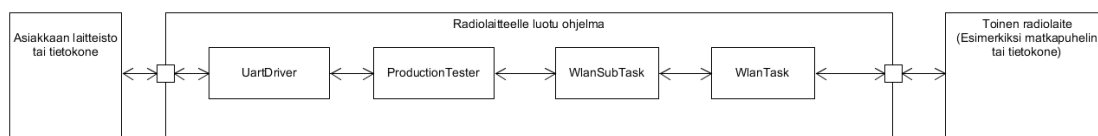
Ohjelman tärkein yksittäinen osa on *ProductionTester*. Kyseisen osa ohjelmaa huolehtii kommunikaatiosta tietokoneen kanssa ja välittää viestejä muille säikeille. Tietokoneen





**Kuva 3.** Testiohjelman arkkitehtuuri esitetty komponenttikaaviona.

kanssa käytävä kommunikaatio tarkoittaa RS-485 standardin mukaista, sarjaliikenneväylän kautta tapahtuvaa kommunikaatiota sitä varten erikseen kehitetyllä protokollalla. *ProductionTester* vastaanottaa viestit, tarkistaa niiden oikeellisuuden, jäsentää viestit ja välittää jäsenneyn datan eteenpäin oikealle säikeelle. Jokainen säie vastaa aina samaansa viestiin ja vastaus lähetetään alkuperäisen viestin lähettäjälle. *ProductionTester* odottaa tehtävältä vastausta määrittelemättömän ajan. Vastauksen saatuaan se jäsentää saadun vastauksen oikeaan muotoon ja lähettää sen sarjaliikennekaapelia pitkin tietokoneelle.



**Kuva 4.** Sarjaliikenneväylää ja langatonta lähiverkkoyhteyttä pitkin saatujen viestien välittäminen *ProductionTester*-säikeelle.

Ohjelman osat *WlanTask* ja *WlanSubTask* liittyvät läheisesti toisiinsa. *WlanTask*-säie kommunikoi *ProductionTester*- ja *WlanSubTask*-säikeiden kanssa ja välittää *ProductionTester*-ohjelmalohkolta saamansa viestit *WlanSubTask*-säikeelle. *ProductionTester* ei lähetä viestejä *WlanSubTask*-lohkolle suoraan eikä näe kyseistä lohkoa. *WlanTask*-säikeen tehtävänä on viestinvälittämisen lisäksi huolehtia verkkoprosessorin tietojen päivittämisestä, esimerkiksi MAC-osoitteen (engl. Media Access Control address) lukemisesta ja kirjoittamisesta, sekä hallinnoida *WlanSubTask*-lohkoa. Näiden kolmen osan välinen viestintä on havainnollistettu kuvassa 4. Kuvan mukaisesti *UartDrive*-ohjelmalohko huolehtii sarjaliikenneyhteyttä pitkin lähetettävistä ja vastaanotettavista viesteistä. Nämä viestit käsittelee *ProductionTester*-säie.

Säie *WlanSubTask* puolestaan hoitaa langattoman lähiverkkoyhteyden muodostamisen, pakettien lähettämisen sekä vastaanottamisen. Alun perin *WlanTask* ja *WlanSubTask* olivat yksi ohjelmalohko. Jatkuvan datan lähettäminen langattoman lähiverkkoyhteyden kautta aiheutti sen, ettei *WlanTask*-säie enää ehtinyt vastaamaan *ProductionTester*-säikeen viesteihin. *WlanTask*- ja *WlanSubTask*-säikeet eroteltiin kahdeksi erilliseksi kokonaisuudeksi, jolloin *WlanTask*-säie pystyy hallinnoimaan *WlanSubTask*-säikeen toimintaa. *ProductionTester*-säie pystyy silti kommunikoimaan *WlanTask*-säikeen kanssa, vaikka jatkuva datan lähetyks on olisikin käytössä.

*FlashTask*-säikeen tarkoitus on huolehtia flash-muistille tapahtuvasta kirjoitus- ja lukuoperaatiosta. Kaikki testiohjelmassa tapahtuva muistille kirjoitus- tai lukuoperaatiot menevät tämän säikeen kautta. Säie eriytyi erilliseksi kokonaisuudeksi sen jälkeen, kun vaatimus jatkuvasta flash-muistilta lukemisesta ja kirjoittamisesta esitettiin. Tehtävä saa käskynsä *ProductionTester*-säikeeltä ja vastaa kyseiselle säikeelle suoritettuaan halutun asian. Poikkeuksena on jatkuva kirjoitus ja lukeminen, jolloin *FlashTask*-säie aktiivisesti lukee ja kirjoittaa dataa muistille ja lähettää jatkuvasti viestejä *ProductionTester*-säikeelle, joka välittää datan tietokoneelle.

### 5.3 Tietokoneen ja radiolaitteen välinen kommunikointi

Radiolaitteelle lähetettävät ja vastaanotettavat viestit koostuvat heksadesimaaliluvuista. Viestien sisältö määrittää suoritettavan testin ja testiin tarvittavat parametrit. Esimerkiksi flash-muistille kirjoittaessa tarvitaan kirjoitettava data. Viesti koostuu kehyksestä, testitunnisteesta sekä testistä riippuvaisesta dataosasta. Kehyksen toinen tavu kertoo viestin pituuden heksadatana. Tämän tavun arvo tulee olla sama kuin viestin koko, mukaan lukien

kehysten kentät. Viestin pituus tulee pystyä esittämään yhdellä tavulla, joten yksittäisen viestin maksimi pituus on 255 tavua. Kaikkien testiohjelman ja tietokoneen käyttämien viestien tulee mahtua yhteen viestiin.

Viestin välitys tapahtuu pääsääntöisesti sarjaliikenneyhteyttä pitkin radiolaitteelle. Radiolaitteessa on muunnin, joka muuntaa saamansa datan sellaiseen muotoon, jota radiolaitteen prosessori osaa käsitellä. Laite tekee viestinvaihtoa tietokoneen kanssa. Tietokone on yhdistetty radiolaitteeseen USB-RS485-kaapelilla. Kaapelin USB-päässä tulee olla muunnin, joka muuttaa datan RS-485-standardin mukaiseksi. Tietokoneella viestien luominen ja käsittely tapahtuvat käyttämällä sarjaporttiajuria. Radiolaitteella käytetään sarjaliikenteen asetuksina:

- lähetysnopeus on 115200 bittiä sekunnissa
- paritettibitti ei ole käytössä
- viestin dataosuus on 8 bittiä
- pysäytysbittinä käytetään yhtä bittiä
- kättelyä ei käytetä.

Käyttämällä USB-RS485-adapteria voidaan radiolaitetta ohjata heti käynnistymisen alusta alkaen. Kun laite käynnistyy, tapahtuu edellä olevien asetusten mukaan viestien vaihtamista. Tämän seurauksena radiolaitte käynnistää testiohjelman.

Radiolaitteella viestien vastaanottamisesta huolehtii erillinen *UARTdriver*-ohjelmätiedosto. Tiedoston alussa määritellään sekä vastaanottavan että lähettävän puskurin koot. Molemmille puskuureille asetetaan myös laukaisin. Laukaisin laukeaa, kun puskuriin on kirjoitettu enemmän dataa, kuin mitä laukaisimen kooksi on määritetty. Laukaisin voidaan asettaa laukeamaan, kun puskurissa on 1/8, 1/4, 1/2, 3/4 tai 7/8 dataa sen koko koosta. Laukaisimen laukeaminen aiheuttaa keskeytyksen. Keskeytyksessä viestipuskuri tyhjenetään ja viesti välitetään toiseen puskuriin, josta ohjelmakoodi voi lukea ja käsitellä viestin. Radiolaitte vastaanottaa tietokoneen lähettämän viestin ja kirjoittaa sen vastaanottavaan puskuuriinsa.

Radiolaitte on jatkuvasti valmis ottamaan uutta dataa sarjaliikennekaapelia pitkin, mikäli se ei ole itse lähettämässä dataa. Sellainen tilanne, jossa molemmat kommunikaatioon osallistujat kirjoittaisivat yhtä aikaa dataa USB-RS485-kaapelin välityksellä ei ole tarkoitusten mukainen. Tätä varten kirjoitus ja lukeminen tapahtuvat normaalisti vuorotellen. Vastaavanlainen käytäntö on myös langattoman lähiverkkoyhteyden kautta tapahtuvassa viestittelyssä. Tällöin verkkoprosessori hoitaa datan lähettämisen ja lukemisen, joskin yhteys tulee muodostaa ensin ohjelmallisesti molemmissa radiolaitteissa.

Tämä lisäksi radiolaitteessa on tuki RS-232-standardin mukaiseen sarjaliikennekommunikointiin erillisen portin kautta. Tätä sarjaliikenneväylää käytetään radiolaitteen kehittämisvaiheessa informaation tulostamiseen radiolaitteelta. Tämä helpottaa testiohjelman sa virheiden etsimistä. Testiohjelmia pystytään ajamaan ainoastaan radiolaitteella, eikä

erillistä testausympäristöä ole saatavilla. Vuorovaikutuksellinen kommunikaatio tapahtuu RS-485-sarjaliikenneväylän kautta. Lopullisessa toteutuksessa radiolaite ei kirjoiteta mitään RS-232-sarjaliikenneväylään, sillä se on tarkoitettu ainoastaan ohjelman kehittämisvaiheeseen.

## 5.4 Säikeiden välinen kommunikointi

Ohjelmassa olevat neljän säiettä kommunikoivat käyttämällä FreeRTOS-kirjaston tarjoamaa viestipuskuritoimintoa. Tällä toiminnolla jokaiselle säikeelle voidaan luoda viestipuskuri. Viestipuskureita on yhtä monta kuin on säikeitäkin. Jokaiselle säikeelle annetaan vähintään kaksi viestipuskuria. Toinen puskuri on viestien lähettämistä varten ja toinen viestien vastaanottamisesta varten. Yksittäinen tehtävä pystyy lukemaan tai kirjoittamaan ainoastaan sille osoitettuihin viestipuskureihin, mutta tehtävien tarkoitus on lukea ainoastaan yhdestä puskurista ja kirjoittaa muihin. Säie ei siis pysty kirjoittamaan mihin tahansa viestipuskuriin ja esimerkiksi *FlashTask*-säie ei pysty kommunikoimaan *WlanTask*-säikeen kanssa. Näiden lisäksi arkkitehtuuriin kuuluu pienempänä kokonaisuutena *UartDriver*-ohjelmätiedosto, jonka tehtävänä on huolehtia datan vastaanottamisesta ja lähettamisestä tietokoneelle. Arkkitehtuuri ja ohjelman osien väliset yhteydet näkyvät kuvassa 3.

Sulautettuun ympäristöön ohjelmoimisessa on yleensä omat haasteensa, jotka ohjelmoitava laite asettaa. Testiohjelmaa ohjelmoidessa suurin yksittäinen rajoite on vähäinen käyttömuisti. Tämän takia säikeiden välillä tapahtuva viestien vaihto tulee toteuttaa niin, ettei muistia varata turhaan. Säikeiden lähettämät viestit vaihtelevat kooltaan, mutta yksittäisen viestin koko on kuitenkin 255 tavua. Esimerkiksi taulukko, joka pystyisi pitämään sisällään 10 viestiä, vaatisi paljon muistia.

Viestien välitys tehdään luomalla staattinen globaali muuttuja, johon lähettäjä voi kirjoittaa viestin. Kirjoittamisen jälkeen välitetään vastaanottajalle staattisen muuttujan muistiosoite lukemista varten. Tällainen staattinen ja globaali muuttuja luodaan jokaiseen tiedostoon, jossa on säikeistetty ohjelmalohko. Luomalla tällainen muuttuja voidaan viestejä lähettää ilman dynaamista muistinhallintaa. Mahdollisena ongelmana on lukemattoman viestin ylikirjoittaminen, ennen kuin vastaanottaja ehtii lukea viestin. Lähettävä säie kirjoittaa viestinsä staattiseen muuttuun ylikirjoittaen aikasemman datan. Tällainen tapa välittää viestejä on rajoittunut viestien vuorottaiseen lähettämiseen. Testiohjelmassa viestien lähetys ja vastaanotto tapahtuu aina vuorottain riippumatta säikeestä. Näin ollen tällainen tapa soveltuu testiohjelman viestienvälitykseen eikä dynaamista muistinhallintaa tarvita. Säie tarvitsee pääsyn toisen säikeen staattiseen muuttuun, jotta se voisi kirjoittaa toisen säikeen muistipuskuriin. Jokaisella säikeellä tulee olla ohjelmakooditiedostossaan määriteltynä staattinen puskuri, sekä siihen kirjoittava metodi, jota muut säikeet voivat kutsua viestin lähettämiseksi.

## 5.5 Verkkoprosessorin käynnistäminen

Verkkoprosessorin SimpleLink-ohjelmistorajapinta mahdollistaa verkkoprosessorin käyttämisen ja esimerkiksi tiedostojen lukemisen muistista. Ennen kuin verkkoprosessorilla pystytään esimerkiksi muodostamaan langaton lähiverkkoyhteys toiseen radiolaitteeseen, se tulee käynnistää erillisellä ohjelmistorajapintakutsulla. Mikäli verkkoprosessoria ei olla käynnistetty ja kutsutaan esimerkiksi tiedostoon kirjoittamiseen tarkoitettua rajapintakutsu, ei kutsutusta metodista ikinä palata. Tämä aiheuttaa ohjelman jumiutumisen. Säikeistetyssä ohjelmassa tulee aluksi luoda erillinen SimpleLink-säie, jonka tarkoituksena on huolehtia verkkoprosessorin asynkronisista tapahtumista. Asynkroninen tapahtuma on esimerkiksi uuden laitteen yhdistyminen radiolaitteesta luotuun tukiasemaan. SimpleLink-säie tulee luoda ennen vuorontajan käynnistämistä. Testiohjelman käynnistyessä *main*-lohko huolehtii radiolaitteen alustamisesta, viestipuskurien alustamisesta, haluttujen ohjelmalohkojen säikeistämisestä ja vuorontajan käynnistämisestä.

Ohjelmassa 5.1 näkyy *main*-lohkon toiminta. Lohkon alussa radiolaitte alustetaan ja radiolaitteen pinnit konfiguroidaan. Molemmat sarjaliikenneväylät alustetaan käyttöä varten. Myös tarvittavat viestipuskurit alustetaan *l\_steupQueues*-metodin sisällä. Jokainen luotu viestintäpuskuri lisätään staattiseen *all\_queues*-muuttujaan, jota käytetään säikeiden luonnissa. Seuraavaksi ohjelmassa luodaan säikeet, joista ensimmäisenä jo edellä mainittu SimpleLink-säie. Tälle säikeelle riittää annettavaksi pelkkä säikeen prioriteetti. Ennen vuorontajan käynnistämistä ohjelma luo loputkin säikeet *osi\_TaskCreate*-metodilla. Metodille annetaan parametrina säikeistettävä funktio, säikeen tunniste, pinon koko, osoitin säikeistettävän funktion parametreihin sekä säikeen prioriteetti. Kaikista korkein prioriteetti annetaan *WlanSubTask*-säikeelle, sillä sertifiointitestauksessa langattoman lähiverkon ylitse lähetettyihin viesteihin nopeasti vastaaminen on tärkeää. Säikeistettäville funktioille annetaan parametrina osoitin muuttujaan, joka pitää sisällään kaikkien säikeiden viestipuskurit. Näistä puskureista säikeet asettavat tarvitsemansa viestipuskurit toimintansa alussa.

```

1 void main()
2 {
3     // Initializing the board
4     BoardInit();
5     setupPins();
6     InitTrace();
7     initMessaging();
8     // Initializing all queues
9     l_setupQueues();
10
11    VStartSimpleLinkSpawnTask (SPAWN_TASK_PRIORITY);
12    osi_TaskCreate (productionTesting,
13                   (const signed char*) "Production_Testing",
14                   OSI_STACK_SIZE,
15                   &all_queues,
16                   1,
17                   &g_pSerialTaskHandle );
18    osi_TaskCreate (wlanTask,
19                   (const signed char*) "Wlan_Task",
20                   OSI_STACK_SIZE + 1000,
21                   &all_queues,
22                   1,
23                   &g_pWlanTaskHandle );
24    osi_TaskCreate (wlanSubTask,
25                   (const signed char*) "Wlan_Sub_Task",
26                   OSI_STACK_SIZE + 2000,
27                   &all_queues,
28                   3,
29                   &g_pWlanSubTaskHandle );
30    osi_TaskCreate (flashOperationsTask,
31                   (const signed char*) "Flash_Task",
32                   OSI_STACK_SIZE + 3000,
33                   &all_queues,
34                   1,
35                   &g_pFlashTaskHandle );
36
37    // Start the task scheduler
38    osi_start();
39 }

```

*Ohjelma 5.1. Ohjelman main-lohko.*

Verkkoprosessori voidaan asettaa toimimaan joko langattoman lähiverkon tukiasemana tai asiakkaana. Testiohjelmassa verkkoprosessorin käynnistämisestä, käytöstä ja asetusten asettamisesta vastaa *WlanTask*- ja *WlanSubTask*-säikeet. Näistä ensimmäinen säie alustaa käynnistymisensä alussa verkkoprosessorin ja alustamisen jälkeen käynnistää verkkoprosessorin asiakasmoodissa. Verkkoprosessorin käynnistäminen ja SimpleLink-rajapinnan käyttäminen vaatii ensimmäiseksi *sl\_Start*-metodin kutsumista. Verkkoprosessori voidaan

sammuttaa *sl\_stop*-metodilla. Metodille *sl\_Start* annetaan kolme parametria. Ensimmäisellä parametrilla voidaan asettaa rajapinta sellaiseksi, että sitä pystytään kutsumaan esimerkiksi sarjaliikenteen läpi. Toisella parametrilla voidaan antaa ensimmäiselle parametrille lisätietoa, esimerkiksi sarjaliikenteen asetuksia. Viimeisenä parametrina voidaan antaa osoitin funktioon. Osoittimen osoittamaa funktiota kutsutaan heti, kun metodi on suoritettu. Mikäli viimeisenä parametrina annetaan *NULL*-arvo, estää *sl\_Start*-metodi ohjelman jatkumisen metodin suorittamisen ajaksi. Näitä tuettuja ominaisuuksia ei ohjelmassa tarvita, joten *sl\_Start*-metodia kutsutaan aina kolmella *NULL*-arvoisella parametrilla.

```

1 // make WLAN a station even if its an access point, clean configures
2 // revert to STA mode
3 long lRetVal = -1;
4 lRetVal = sl_Start(NULL,NULL,NULL);
5
6 if (lRetVal < 0)
7 {
8     G_TRACE_ERROR("Failed_to_start_the_device_%d", lRetVal);
9     G_LOOP_FOREVER();
10 }
11
12 //Set wlan to station mode
13 lRetVal = sl_WlanSetMode(ROLE_STA);
14 if(lRetVal < 0)
15 {
16     G_TRACE_ERROR("%d", lRetVal);
17     G_LOOP_FOREVER();
18 }
19 // set default configures
20 l_configSLToDefaultState();
21
22 // after configures are cleaned, start sl_api
23 lRetVal = sl_Start(NULL,NULL,NULL);
24 if (lRetVal < 0)
25 {
26     G_TRACE_ERROR("Failed_to_start_the_device_%d", lRetVal);
27     G_LOOP_FOREVER();
28 }

```

### **Ohjelma 5.2.** Verkkoprosessorin käynnistäminen.

Ohjelma 5.2 havainnollistaa verkkoprosessorin käynnistämistä ja alustamista asiakasmoodiin. Alustaminen tulee tehdä joka kerta, kun *WlanTask*-säie käynnistetään. Laitteen käynnistyessä uudelleen on mahdollista, että vanhat asetukset ovat edelleen voimassa. Verkkoprosessori käynnistetään tukiasemana, jos verkkoprosessori oli asetettu tukiasemaksi ennen uudelleen käynnistämistä. Aluksi verkkoprosessori käynnistetään *sl\_Start*-metodilla, jonka jälkeen se asetetaan asiakasmoodiin *sl\_WlanSetMode*-metodilla. Metodille annettava parametri on *enum*-tyyppinen muuttuja, ja *ROLE\_STA*-muuttujan arvo on nolla, jolloin

*sl\_WlanSetMode*-metodi käynnistää verkkoprosessorin asiakkaana. Seuraavaksi verkkoprosessori asetetaan oletusasetuksiin *l\_configSLToDefaultState*-metodilla. Tämän metodin kutsuminen vaatii verkkoprosessorin uudelleen käynnistämistä ja metodi sammuttaakin verkkoprosessorin. Oletusasetuksiin asettaminen saadaan päätökseen, kun verkkoprosessori käynnistetään uudestaan *l\_configSLToDefaultState*-metodin jälkeen *sl\_Start*-funktiokutsulla.

Ainoastaan *WlanTask*- ja *WlanSubTask*-säikeissä kutsutaan *sl\_start*- tai *sl\_stop*-metodeja. Kyseisten metodien kutsuminen useammassa paikassa säikeistetyssä ohjelmassa saattaisi aiheuttaa ongelmia. Yhden säikeen käyttäessä verkkoprosessoria toinen säie voisi sammuttaa esimerkiksi verkkoprosessorin. Tämän seurauksena ensimmäinen säie saattaisi jäädä odottamaan ikuisesti vastausta verkkoprosessorilta.

## 5.6 Radiolaitteen asettaminen tukiasema- ja asiakasmoodiin

Radiolaitteen alustamisen ja säikeiden käynnistymisen jälkeen testiohjelmalla on mahdollista käynnistää ensimmäinen testisekvenssi. Laitteen asettaminen tukiasemaksi on yksi sen normaaliin toimintaan kuuluvista tehtävistä. Verkkoprosessorin asettamisesta tukiasemaksi vastaa *WlanSubTask*-säie, joka saa käskynsä *WlanTask*-säikeeltä. Verkkoprosessorin asettaminen tukiasemaksi alkaa luomalla *socketTaskObj*-tietue. Tietue sisältää tukiaseman pystyttämiseksi vaaditut konfiguraatiot ja ne saadaan jäsenneltyä samasta viestistä, jolla testisekvenssi käynnistetään. Seuraavaksi ajetaan *socket\_run-funktio*, jolle annetaan parametrina juuri luotu tietue, sekvenssin käynnistänyt viesti ja lähettämiseen sekä vastaanottamiseen tarkoitetut viestipuskurit. Tietueessa on tarvittavat tiedot verkkoprosessorin asettamiseksi tukiasemamoodiin. Tietue sisältää seuraavat konfiguraatiot:

- tukiaseman langattoman lähiverkon verkkotunnus
- tukiaseman salasana
- käytettävä kanava
- käytettävä salausero
- käytettävä portti
- käytettävä aliverkon peite
- IP-osoite
- verkkoprosessorin moodi
- lähetysteho
- lähetysnopeus.

Kaikkia näitä tietoja ei käytetä tukiaseman pystytyksessä, vaan osaa käytetään ainoastaan silloin, kun luodaan asiakkaana yhteys olemassa olevaan tukiasemaan. Funktio *socket\_run*



ajetaan riippumatta siitä, halusiko käyttäjä yhdistää radiolaitteen olemassa olevaan tukiasemaan vai asettaa sen tukiasemaksi. Vastaanotetun viestin sisältö määrittelee kumman toimenpiteen *socket\_run*-funktio suorittaa.

Verkkoprosessorin asettaminen haluttuun moodiin tapahtuu ajamalla *socket\_run*-metodi. Ohjelmassa 5.3 on esitetty kyseisen metodin toteutus. Funktion alussa parametrina annettu *socketTaskObj*-muuttuja asetetaan globaaliin *g\_socketObject*-muuttujaan, joka on myös staattinen. Loput metodille annetut parametrit liittyvät viestien välittämiseen. Toisena parametrina on osoitin toiminnallisuuden käynnistäneeseen viestiin, kolmantena osoitin vastaanottamiseen tarkoitettuun viestipuskuriin ja viimeisenä osoitin lähettämiseen tarkoitettuun viestipuskuriin. Konfiguraatioiden tallentamisen jälkeen itse konfiguroiminen voidaan aloittaa kutsumalla *socket\_init*-metodia. Metodien alustaa verkkoprosessorin konfiguraatioiden mukaisesti.

```

1 void Socket_run(void *pvParameters, sTaskMsg* message,
2                 OsiMsgQ_t* txQueue, OsiMsgQ_t* rxQueue)
3 {
4     g_socketObject = (SockObj)pvParameters;
5
6     _i16 sl_retVal = 0;
7     int32_t retVal = SUCCESS;
8     G_TRACE_DEBUG("Started!");
9
10    retVal = SUCCESS;
11
12    Socket_init(g_socketObject);
13
14    if(g_socketObject->wifiConfig.wlanMode == ROLE_STA)
15    {
16        retVal = l_WlanConnect(message, txQueue, rxQueue);
17        testWlanConnection(message, txQueue);
18
19    }
20
21    else if (g_socketObject->wifiConfig.wlanMode == ROLE_AP)
22    {
23
24        retVal = ipLeased(message, txQueue, rxQueue);
25
26        if(retVal == SUCCESS)
27        {
28            waitSocketConnection();
29            wlanReceiveMsg();
30        }
31
32    }

```

**Ohjelma 5.3.** *Socket\_run*-lohko.

Tässä vaiheessa jo suoritettuna *socket\_init*-metodin alussa verkkoprosessori sammutetaan, asetetaan oletusarvoihinsa ja käynnistetään uudelleen. Oletusarvojen mukaisesti radiolaite asetetaan asiakkaaksi. Tällöin kaikki muodostetut yhteydet katkaistaan. Ohjelmallinen toteutus on vastaavanlainen, kuin mitä ohjelmassa 5.2 on esitetty, mutta *sl\_Start*-metodia ei kuitenkaan tarvitse kutsua. Tämän jälkeen tarkistetaan parametrina annetun *g\_socketObject*-muuttujan arvoista, että asetetaanko radiolaite tukiasemaksi vai asiakasmoodiin.

Tukiasemaksi konfiguroiminen tapahtuu erillisessä *l\_configureAPMode*-metodissa, jossa *g\_socketObject*-tietueen sisältämät konfiguraatioasetukset asetetaan verkkoprosessorille käyttäen rajapintakutsua *sl\_wlanSet*. Tälle metodille annetaan tieto muutettavan konfiguraation tunnisteesta, muutettavasta muuttujasta ja lopuksi uusi arvo. Ohjelmassa 5.4 havainnollistetaan, kuinka tukiaseman verkkotunnus voidaan muuttaa *sl\_wlanSet*-metodilla käyttäen apuna *g\_socketObject*-muuttujaa. Ensimmäisenä parametrina annetaan konfiguroitavan arvon tunniste, seuraavana parametrina tulee antaa konfiguroitava valinta, jonka jälkeen annetaan konfiguroitavan arvon pituus ja lopuksi itse konfiguroitava arvo.

```

1      SockObj this = g_socketObject;
2      iRetVal = sl_wlanSet(SL_WLAN_CFG_AP_ID,
3                          WLAN_AP_OPT_SSID,
4                          strlen((const char *)&this->wifiConfig.ucSSID),
5                          (const uint8_t *)&this->wifiConfig.ucSSID);
6      G_ASSERT_ON_ERROR(iRetVal);

```

#### *Ohjelma 5.4. sl\_wlanSet-metodin käyttäminen.*

Konfiguroimisen lopuksi verkkoprosessori uudelleen käynnistetään, jolloin muutokset astuvat voimaan. Verkkoprosessori palautetaan aina oletusarvoihinsa asiakasmoodiin, kun konfigurointi aloitetaan. Mikäli konfiguraatiossa oleva moodi on tukiaseman sijasta asiakasmoodi, verkkoprosessoriin ei tarvitse muuttaa kuin käytettävä lähetysteho *sl\_wlanSet*-metodilla. Onnistuneen konfiguroimisen jälkeen ohjelman suoritus jatkuu *Socket\_run*-funktiossa. Seuraavaksi funktio yhdistää asiakkaana toimivan radiolaitteen olemassa olevaan tukiasemaan (ohjelma 5.3, rivi 17) tai jää tukiasemana odottamaan asiakkaan yhdistymistä radiolaitteelle (ohjelma 5.3, rivi 24). Konfiguraatiot määrittävät kumman toiminnallisuuden metodi tekee.

## 5.7 Datan lähettäminen ja vastaanottamien tukiasemana ja asiakkaana

Tukiasemamoodissa säie jää odottaamaan toisen radiolaitteen yhteydenottoa. Tässä tilassa ollessaan ohjelma aktiivisesti kutsuu SimpleLink-kirjaston *sl\_Accept*-metodia. Kyseinen metodi luo yhteyden sellaisen radiolaitteen kanssa, joka on yhteyden muodostamista pyytänyt. Metodi palauttaa soketin käsittelijän tai virhekodein. Paluuarvo riippuu yhteyden

muodostamisen onnistumisesta. Yhteyden muodostamisen jälkeen tukiasema on valmis kommunikoidaan toisen radiolaitteen kanssa käyttäen saamaansa käsittelijää.

Asiakasmoodissa säie yrittää muodostaa yhteyden haluttuun radiolaitteeseen. Yhteyden muodostus aloitetaan *l\_WlanConnect*-funktiossa, joka on esitetty ohjelmassa 5.5. Yhteyden muodostamista varten käyttäjän on täytynyt antaa tukiaseman verkkotunnus sekä salasana. Tukiaseman salausalgoritmi tulee olla joko WPA (Wi-Fi Protected Access) tai WPA2 (Wi-Fi Protected Access II). Metodien alussa määritellään *SlSecParams\_t*-tyyppinen *secParams*-muuttuja, johon tallennetaan konfiguraation mukaisesti käytettävä salasana, salausalgoritmi ja salasanan pituus. Yhdistäminen tapahtuu käyttämällä *sl\_wlanConnect*-rajapintakutsua. Kutsulle annetaan parametrina osittimet tukiaseman verkkotunnukseen, sen pituuteen sekä *secParams*-muuttujaan. Tietue sisältää käytettävän salaustyyppin, mahdollisen salasanan sekä salasanan pituuden. Rajapintakutsulle voidaan lisäksi antaa asiakkaan MAC-osoite sekä *SlSecParamsExt\_t*-tietue, joka määrittelee turvallisuusasetuksia tarkemmin. Tätä tietuetta tai MAC-osoitetta ei kuitenkaan ole pakko asettaa. Kutsun jälkeen radiolaitte lähettää viestin sarjaliikenteeseen toteutuneesta käskystä, jonka jälkeen se jää odottamaan yhteyden muodostumista. Onnistuneen yhteyden muodostamisen jälkeen asiakaslaitteen tulee saada tukiasemalta IP-osoite, jota se voi käyttää. Asiakaslaite saa tukiasemalta IP-osoitteen asynkronisena tapahtumana.

```

1  static long l_WlanConnect(sTaskMsg* message, OsimsgQ_t txQueue,
2                          OsimsgQ_t rxQueue)
3  {
4      SlSecParams_t secParams = {0};
5      long lRetVal = 0;
6
7      secParams.Key = (signed char*)g_socketObject->wifiConfig.ucPassword;
8      secParams.KeyLen = strlen((char*)secParams.Key);
9      secParams.Type = SL_SEC_TYPE_WPA_WPA2;
10
11     sl_WlanConnect(
12         (signed char*)g_socketObject->wifiConfig.ucSSID,  \\SSID name
13         strlen((char*)g_socketObject->wifiConfig.ucSSID), \\SSID lenght
14         0,                                                  \\ MAC-address
15         &secParams,                                       \\Security
16         0);                                               \\Enterprise
17     G_ASSERT_ON_ERROR(lRetVal);
18     // Tell the PC that board is connecting to AP
19     answerTaskMsg(*message, txQueue);
20     while((!IS_CONNECTED(g_uiStatus)) || (!IS_IP_ACQUIRED(g_uiStatus)))
21     {
22         // Wait for WLAN Event
23         osi_Sleep(TASK_SLEEP_MILLISECONDS);
24     }
25     return SUCCESS;
26 }

```

*Ohjelma 5.5. l\_WlanConnect-funktio.*

Ennen varsinaisen kommunikoinnin aloittamista asiakkaan tulee luoda soketti. Soketin avulla asiakas pystyy muodostamaan yhteyden tukiasemaan ja kommunikoimaan sen kanssa. Soketin luominen tapahtuu käyttämällä *sl\_socket*-rajapintakutsua. Tämä kutsu palauttaa luodun soketin käsittelijän. Kutsulle annetaan parametrina käytettävä IP-protokolla, käytettävä kommunikointitapa sekä kommunikointiin käytettävä protokolla. Rajapintakutsu palauttaa negatiivisen tai positiivisen arvon. Negatiivinen arvo on virhekoodi ja positiivinen arvo on kutsun palauttama käsittelijä.

Onnistuneen luonnin jälkeen yhteys tukiasemaan voidaan muodostaa *sl\_connect*-kutsulla. Tälle kutsulle tulee antaa parametrina luodun soketin käsittelijä sekä *SlSockAddrIn\_t*-tyyppiä oleva tietue. Tämä tietue sisältää tiedot käytettävästä internet protokollasta, IP-osoitteesta ja portista. Yhdistäminen tapahtuu näiden parametrien perusteella. Kutsun palauttama positiivinen arvo kertoo yhteyden muodostamisen onnistumisesta.

```

1      uint32_t i = 0;
2
3      // Send a packet to access point to indicate
4      // that client is ready to receive packets
5      iStatus = sl_Send(iSockID, &i, sizeof(i), 0);
6
7      uint8_t data = 0xFF;
8      for(;;)
9      {
10
11         int16_t status = -1;
12         status = sl_Recv(iSockID, &data, sizeof(data), 0);
13         if(status > 0)
14         {
15             memcpy(message->pTaskMsgData->data, &data, sizeof(data));
16             // Based on received payload form a message
17             message->pTaskMsgData->dataSize = sizeof(data);
18             message->pTaskMsgData->commandType = eReceivePacketsFromAP;
19             // Forward message to UART
20             answerTaskMsg(*message, txQueue);
21
22         }
23     }

```

### *Ohjelma 5.6. Datan lähettäminen ja vastaanottaminen.*

Kommunikoiminen asiakkaan ja tukiaseman välillä tapahtuu käyttämällä SimpleLink-kirjaston tarjoamia *sl\_Send*- ja *sl\_Recv*-rajapintakutsuja. Ohjelma 5.6 havainnollistaa näiden kahden metodin käyttöä. Ensimmäisellä kutsulla voidaan lähettää dataa soketin kautta yhdistyneelle laitteelle (ohjelma 5.6, rivi 5) ja jälkimmäisellä vastaanottaa dataa (ohjelma 5.6, rivi 12). Lähettämiseen ja vastaanottamiseen tarkoitettuille rajapintakutsuille tulee antaa parametrina vähintään soketin käsittelijä, osoitin puskuriiin ja puskurin pituus. Lähetettäessä annetun osoittimen osoittaman puskurin sisältö välitetään vastaanottajalle ja

otettaessa vastaan kirjoitetaan vastaanotettu data osoittimen osoittamaan muistialueeseen. Vastaanottamisen jälkeen saatu data voidaan lukea parametrina annetusta puskurista. Puskurina voi toimia taulukko tai esimerkiksi *integer*-tyyppinen muuttuja.

Ohjelmassa 5.6 näytetään, miten asiakas lähettää dataa tukiasemalle yhteyden muodostamisen jälkeen. Asiakas lähettää ensimmäiseksi yhden tavun dataa, jonka jälkeen se jää pysyvästi välittämään tukiaseman lähettämiä paketteja *ProductioTester*-säikeelle. Vastavasti *ProductioTester*-säie välittää viestit tietokoneelle sarjaliikenneyhteyden ylitse. Ohjelman 5.6 mukaista toiminnallisuutta käytetään muodostetun langattoman lähiverkkoyhteyden tarkkailemisessa.

## 5.8 Datan jatkuva lähettäminen yhdistyneeseen radiolaitteeseen

Testiohjelmalla pitää pystyä lähettämään dataa jatkuvasti yhdistyneeseen radiolaitteeseen, jotta radiolaitteiden välille muodostunutta yhteyttä voidaan testata. Muodostunut yhteys yritetään katkaista aiheuttamalla radiolaitteisiin häiriötä. Kohdassa 4.4 annettiin yhteyden tarkkailemisesta vaatimus. Tukiaseman ja asiakkaan muodostettua kommunikointiyhteys lähettää asiakas viestin tukiasemalle. Vastaanotettuaan viestin alkaa tukiasema lähettämään sekunnin välein viestejä asiakkaalle. Viestin sisältönä käytetään juoksevaa numerointia välillä 0-255. Asiakas välittää saamansa viestit sarjaliikenneportin kautta tietokoneelle.

Datan jatkuva lähettäminen yhdistyneeseen radiolaitteeseen voidaan aloittaa sen jälkeen, kun yhteys on saatu muodostettua onnistuneesti asiakkaan ja tukiaseman väliin. Yhteyden tarkkailu tehdään tietokoneella olevassa ohjelmassa. Asiakkaan tietokoneelle lähettämät viestit mahdollistavat yhteyden tarkkailun, mutta tietokoneohjelman täytyy aktiivisesti tarkistaa asiakkaalta saamiensa viestien sisältö ja pitää kirjaa, että viestien juoksevassa numeroinnissa ei tapahdu kuin yhden askeleen siirtyminen kerrallaan.

## 5.9 Datan jatkuva lähettäminen ja vastaanottaminen radiolaitteella yhteyttömästi

Yhtenä vaatimuksena testiohjelmalle on datan jatkuva lähettäminen ja vastaanottaminen radiolaitteella ilman, että langatonta lähiverkkoyhteyttä muodostetaan lainkaan. Tällä tarkoitetaan, että verkkoprosessori aktiivisesti lähettää ennalta määrättyä viestiä määrätylle langattoman lähiverkon kanavalle. Kaikki kanavaa kuuntelevat radiolaitteet saavat lähetetyn viestin. Tämän vaatimuksen toteuttamiseen voidaan käyttää ohjelmiston kehitysympäristön tarjoamia esimerkkiohjelmiä. Datan jatkuvan lähettämisen ja vastaanottamisen toteuttamiseksi voidaan hyödyntää *tranceiver\_mode*-lohkoa, joka löytyy ohjelmiston kehitysympäristön ”example”-kansion alta. Jokaiselle esimerkille löytyvät valmiiksi käännetyt binääri-tiedostot, jotka voidaan ajaa esimerkiksi Texas Instrumentsin valmistamalla Launchpad-piirilevyllä. Esimerkkejä pystytään nopeasti testaamaan ja muokkaamaan omaan käyttöön sopivaksi.

Ohjelman alussa luodaan globaali muuttuja, joka sisältää jatkuvasti lähetettävän viestin sisällön. Lähetyksen alettua tätä viestiä tullaan lähettämään kaikille samalla taajuudella oleville radiolaitteille. Esimerkissä käytetty viesti käy testiohjelman vaatimuksiin ja sitä voidaan käyttää sellaisenaan testiohjelmassa. Testiohjelma ei oletta, että tähän viestiin vastataan. Tärkeintä on, että jotain dataa saadaan lähetettyä jatkuvasti ja mahdollisimman tehokkaasti.

Testiohjelmalle tulee antaa pakettien lähettämisen aloittamisesta ilmoitus, jotta se tietää käynnistää datan jatkuvan lähettämisen tai vastaanottamisen. Tämä ilmoitus tapahtuu lähettämällä viesti sarjaliikenneyhteyden kautta radiolaitteelle. Vastaanotettu viesti käsitellään ja välitetään lopuksi *WlanSubTask*-säikeelle, jolloin säie käynnistää viestin sisällön mukaisesti datan jatkuvan lähettämisen tai vastaanottamisen. Ennen kuin *WlanSubTask-säie* asettaa verkkoprosessorille tarvittavat parametrit ja käynnistää määrätyn testin, vastaa se *WlanTask*-säikeelle. Vastausviesti välitetään *ProductionTester*-säikeelle, joka puolestaan välittää sen sarjaliikenneyhteyden ylitse tietokoneelle. Käyttäjä saa tällä tavalla indikaation testin alkamisesta.

Jatkuva datan lähettäminen yhteydettömästi alkaa konfiguroimalla verkkoprosessori asiakasmoodiin ja luomalla soketti. Ohjelmassa 5.7 näytetään esimerkkitoteutus datan jatkuvasta lähettämisestä. Tarvittava soketti luodaan käyttämällä *sl\_Socket*-rajapintakutsua. Kutsulle annetaan ensimmäiseksi parametriksi jatkuvaan yhteydettömään lähettämiseen tarkoitettu *SL\_AF\_RF*-vakio, jonka arvo on kuusi. Vakion avulla pystytään määrittelemään lähetettävä viesti kokonaisuudessaan aina otsikkokentistä alkaen. Seuraavana oleva parametri määrittelee kommunikaation semantiikan, jonka arvoksi annetaan ohjelmiston kehitysympäristössä määritelty *SL SOCK\_RAW*-vakio. Sen arvolla kolme saadaan luotua soketti, jolla pystytään lähettämään itse määriteltyä dataa. Viimeinen parametri kertoo käytettävästä protokollasta. Arvolla nolla soketti käyttää oletusprotokollaa, joka soveltuu muihin annettuihin parametreihin.

```

1  unsigned long numberOfFrames = 0;
2  iSoc = sl_Socket(SL_AF_RF, SL_SOCKET_RAW, 0);
3
4  // Continuous Transferring is on
5  answerTaskMsg(*message, txQueue);
6
7
8  sl_SetSockOpt(iSoc, SL_SOL_PHY_OPT, SL_SO_PHY_NUM_FRAMES_TO_TX,
9                &numberOfFrames, sizeof(unsigned long));
10
11  _i16 ret = sl_Send(iSoc,
12                   RawData_Ping,
13                   sizeof(RawData_Ping),
14                   SL_RAW_RF_TX_PARAMS(txConfig->m_Channel,
15                   txConfig->m_Rate,
16                   txConfig->m_TxPowerLevel,
17                   SHORT_PREAMBLE_MODE));
18
19  // Loop here until power is cut
20  for(;;)
21  {
22      osi_Sleep(TASK_SLEEP_MILLISECONDS);
23  }

```

**Ohjelma 5.7.** *Datan lähettäminen yhteydettömästi.*

Soketin luomisen jälkeen välitetään tietokoneelle testin käynnistänyt viesti, jotta tietokoneetta operoiva käyttäjä saa tiedon testin aloittamisesta. Tämän jälkeen soketti konfiguroidaan jatkuvaa lähettämistä varten käyttämällä *sl\_SetSockOpt*-rajapintakutsua (ohjelmassa 5.7 rivit 8 ja 9). Kutsulle annetaan ensimmäisenä parametrina konfiguroitavan soketin käsittelijä. Seuraava parametri kertoo konfiguroitavan protokollatason. Vakiolla *SL\_SOL\_PHY\_OPT* muokataan soketin siirtoyhteyserroksen asetuksia. Kolmannella parametrilla määritetään muokattava konfiguraatio. Lähetettävien viestien määrä voidaan määrittää käyttämällä vakiota *SL\_SO\_PHY\_NUM\_FRAMES\_TO\_TX*. Neljäs parametri kertoo tässä tapauksessa lähetettyjen pakettien määrän. Viimeinen parametri määrittää neljännen parametrin pituuden, ja sen tulee olla *sizeof(unsigned long)* suuruinen, koska konfiguraatio on määritetty *SL\_SO\_PHY\_NUM\_FRAMES\_TO\_TX*-vakiolla, . Antamalla lähetettävien viestien määräksi nolla, verkkoprosessori lähettää viestejä niin kauan kunnes radiolaitteesta katkaistaan virta.

Seuraavaksi voidaan aloittaa itse viestien lähettäminen. Lähetys aloitetaan kutsumalla *sl\_Send*-metodia, jolle annetaan parametreina käytettävän soketin käsittelijä, lähetettävä viesti, viestin pituus sekä lippumuuttuja. Lipulla voidaan määrittää käytettävä kanava, lähetysnopeus ja lähetysteho. Makro *SL\_RAW\_RF\_TX\_PARAMS* tiivistää kanavan, lähetysnopeuden ja tehon yhdeksi *signed short*-tyyppiseksi arvoksi. Viimeisenä oleva parametri *SHORT\_PREAMBLE\_MODE* määrittää käytettävän tahdistuskuvion, joka voi olla ly-

hyt tai pitkä. Tahdistuskuvion avulla vastaanottava verkkoprosessori saa selville kehyksen alkamiskohdan[30].

Rajapintakutsun jälkeen *WlanSubTask*-säie pystyy jatkamaan toimintaansa normaalisti, mutta lähetystä ei ole tarkoitus keskeyttää, eikä säieken tarvitse tehdä mitään lähetyksen alettua. Säie menee silmukkaan, josta se ei palaa. Koska *WlanSubTask*-säikeellä on kaikista suurin prioriteetti, täytyy sen aktiivisesti luovuttaa ajovuoroa muille säikeille, jottei nälkiintymistä tapahdu. Ajovuoron luovuttaminen tapahtuu kutsumalla *osi\_Sleep*-metodia, jolle annetaan parametrina nukuttava aika millisekunteina.

Datan jatkuva vastaanottaminen tapahtuu samankaltaisella tavalla. Ensimmäiseksi alustetaan tietue, jota käytetään soketin konfiguroimisessa. Seuraavaksi luodaan soketti samalla tavalla kuin jatkuvan datan lähetyksessä. Viimeiseksi parametriksi annetaan kuitenkin vastaanottamiseen käytettävä kanava. Ohjelmassa 5.8 soketin luominen näkyy rivillä 6. Soketin luomisen jälkeen käytetään *sl\_SetSockOpt*-rajapintakutsua soketin arvojen muokkaamiseen. Toisena parametrina annettu *SL\_SOL\_SOCKET*-vakio määrittää konfiguroimisen tapahtumisen soketin kuljetuskerrokseen. Kolmantena parametrina annetaan ohjelmiston kehitysympäristössä määritetty *SL\_SO\_RCVTIMEO*-vakio. Vakion avulla voidaan määrittää, että konfiguroivalla parametrilla asetetaan enimmäisaika, jonka käytetään yksittäisen paketin vastaanottamiseen. Enimmäisajaksi asetetaan 20 000 mikrosekuntia neljännen ja viidennen parametrin avulla. Tämän jälkeen voidaan aloittaa viestien vastaanottaminen kutsumalla *sl\_Recv*-metodia. Metodille annetaan parametreina käytettävä puskuri, puskurin pituus ja viimeisenä parametrina mahdollinen lippumuuttuja, jota ei kuitenkaan kyseisessä ohjelmiston kehitysympäristön versiossa tueta.

```

1  struct SlTimeval_t timeval;
2  timeval.tv_sec = 0;           // Seconds
3  timeval.tv_usec = 20000;     // Microseconds.
4  char acBuffer[1500];
5  //Open Socket on the channel to listen
6  int iSoc = sl_Socket(SL_AF_RF,SL_SOCKET_RAW,iChannel);
7
8  // Enable receive timeout
9  sl_SetSockOpt(iSoc,SL_SOL_SOCKET,SL_SO_RCVTIMEO, &timeval, \
10                sizeof(timeval));
11 sl_Recv(iSoc,acBuffer,1470,0);
12 // inform the user that Continuous Receiving is on
13 answerTaskMsg(*message, txQueue);

```

**Ohjelma 5.8.** *Datan vastaanottaminen yhteydettömästi.*

Esimerkkiohjelmassa kuuntelu jatkuu, kunnes käyttäjä keskeyttää kuuntelun. Tällöin esimerkkiohjelma tulostaa yhteenvedon pakettiliikenteestä kuuntelunsa ajalta. Yhteenvedossa näytetään kuunneltujen pakettien määrä kunkin nopeuden osalta. Radiolaitteella olevassa ohjelmassa ei tarvitse tuottaa kuuntelusta statistiikkaa, mutta sellaisen tuottaminen



olisi mahdollista *sl\_WlanRxStatStar*- ja *sl\_WlanRxStatGet*-metodeilla. Kuuntelun aloittaminen ilmoitetaan käyttäjälle lähettämällä vastausviesti *WlanSubTask*-säikeen saamaan viestiin juuri *sl\_Recv*-funktiokutsun jälkeen (ohjelma 5.8, rivi 13).

## 5.10 Kirjoittaminen ja lukeminen radiolaitteen flash-muistille

Radiolaitteen flash-muistille kirjoittamisesta huolehtii *flashOperationsTask*-säie. Säikeen avulla voidaan kirjoittaa muistilla sijaitsevaan tiedostoon dataa. Tiedoston lukeminen tai kirjoittaminen alkaa lähettämällä testin käynnistävä viesti sarjaliikenneväylän kautta radiolaitteelle. *ProductionTester*-säie ohjaa viestin *flashOperationsTask*-säikeelle. Viestin sisällä on tieto halutusta operaatiosta, sekä mahdollinen kirjoitettava data. Ohjelmassa käytetään SimpleLink-kirjaston tarjoamia tiedoston käsittelyyn tarkoitettuja rajapintakutsuja. Tiedoston käsittelyn helpottamiseksi on luotu erillinen ohjelmatiedosto, jossa tavallisimmat tiedosto-operaatiot on toteutettu käyttäen SimpleLink-kirjaston rajapintakutsuja.

Kirjoitettavan tiedoston koko on 250 tavua ja se pystyy pitämään sisällään yhden viestin. Jotta kirjoitus olisi mahdollista, tulee kirjoituksesta vastaavalle funktiolle antaa kirjoitettava viesti, viestin pituus ja kirjoitettavan tiedoston nimi. Viestin kirjoittaminen tiedostolle alkaa tiedoston olemassaolon varmistamisella. Olemassaolon varmistava metodi on esitelty ohjelmassa 5.9. Ensimmäisenä tiedostoa haetaan nimen perusteella muistista. Tiedosto luodaan flash-muistille (ohjelma 5.9, rivi 6 ja 7), jos sitä ei olla aikaisemmin luotu tai se on poistettu.

```

1  int ensureFlashTestFileExists(uint8_t* fileName)
2  {
3      if(!hd_fs_Exists(fileName, NULL))
4      {
5          // Create default file.
6          uint32_t retVal = hd_fs_Create(fileName,
7                                     L_FLASH_TESTS_FILES_MAX_SIZE);
8          if(retVal != SUCCESS)
9          {
10             G_TRACE_ERROR("%d", retVal);
11             return retVal;
12         }
13     }
14     return SUCCESS;
15 }

```

**Ohjelma 5.9.** Tiedoston olemassaolon varmistaminen.

Tiedostoon kirjoittamisesta huolehtiva funktio on esitelty ohjelmassa 5.10. Metodien alussa varmistetaan annetun tiedoston olemassaolo. Varmistamisen jälkeen tiedostolle voidaan kirjoittaa haluttu data. Kirjoittamisen mahdollistavalle funktiolle annetaan parametreina tiedoston nimi, kirjoituksen alkukohta tiedostossa, kirjoitettava data ja datan pituus

(ohjelma 5.10, rivi 10 ja 11). Radiolaitetta testattaessa riittää, että uusimman kirjoitusoperaation data on luettavissa. Kirjoitettava data voidaan kirjoittaa tiedoston alkuun kirjoittamalla samalla aikaisemman sisällön päälle. Paluarvona tulee saada arvo nolla, joka tarkoittaa kirjoitusoperaation onnistumista.

```

1 int writeToFileTelegram(uint8_t payloadLength,
2                          uint8_t* receivedPayload, uint8_t* fileName)
3 {
4     uint32_t retVal = ensureFlashTestFileExists(fileName);
5     if(retVal != SUCCESS)
6     {
7         G_TRACE_ERROR("%d",retVal);
8         return retVal;
9     }
10    retVal = hd_fs_WriteByFileName(fileName, 0, receivedPayload,
11                                   (uint32_t) payloadLength);
12    if(retVal != SUCCESS)
13    {
14        G_TRACE_ERROR("%d",retVal);
15        return retVal;
16    }
17    return SUCCESS;
18 }

```

**Ohjelma 5.10.** Kirjoittaminen flash-muistille.

Kirjoittamisen jälkeen säie suorittaa lukuoperaation juuri kirjoittamalleen datalle. Luettu data lähetetään osana vastausviestiä. Lukeminen tapahtuu erillisenä operaationa ja samainen operaatio suoritetaan, jos säiettä on käsketty lukemaan tiedoston sisältö. Tämä metodi on esitelty ohjelmassa 5.11. Metodille tulee antaa parametreina osoittimet puskuriin, *unsigned long* -tyyppiseen muuttujaan ja tiedostonimeen. Puskuriin tallennetaan luettu data ja muuttujaan tallennetaan puskuriin kirjoitettujen merkkien määrä. Kuten kirjoittaminen, myös lukeminen alkaa tiedoston olemassaolon varmistamisella (ohjelma 5.11, rivi 5). Varmistamisen jälkeen rivillä 13 luetaan tiedoston metadata. Lukemalla tiedoston metadata saadaan selville, montako merkkiä tiedostoon on viimeksi kirjoitettu. Esimerkiksi jos ensimmäisellä kirjoituskerralla tiedostoon on kirjoitettu 150 merkkiä ja jälkimmäisellä 50 merkkiä, tulee säikeen lukea ainoastaan viimeiseksi kirjoitetut merkit. Tässä tapauksessa 50 merkkiä.

```

1  int readFromFileTelegram(unsigned char* bufferToRead,
2                          unsigned long* fileSize,
3                          uint8_t* fileName)
4  {
5      uint32_t retVal = ensureFlashTestFileExists(fileName);
6      if(retVal != SUCCESS)
7      {
8          G_TRACE_ERROR("%d",retVal);
9          return retVal;
10     }
11
12     SlFsFileInfo_t pFileInfo = {0};
13     retVal = hd_fs_Info(fileName, &pFileInfo);
14     if(retVal != SUCCESS)
15     {
16         G_TRACE_ERROR("%d",retVal);
17         return retVal;
18     }
19
20     *fileSize = pFileInfo.FileLen;
21     retVal = hd_fs_ReadByFileName(fileName,
22                                 0,
23                                 bufferToRead,
24                                 pFileInfo.FileLen);
25
26     if( retVal != SUCCESS )
27     {
28         G_TRACE_ERROR("%d",retVal);
29         return retVal;
30     }
31
32     return SUCCESS;
33 }

```

*Ohjelma 5.11. Lukeminen flash-muistilta.*

Metadatan lukemisen jälkeen voidaan suorittaa lukuoperaatio, joka näkyy ohjelmassa 5.11 rivillä 21. Lukuoperaatiolle annetaan parametrina tiedoston nimi, lukemisen alku-kohta, parametrina annettu osoitin puskuriin sekä luettavan datan määrä. Luettavan datan määrä saatiin lukemalla tiedoston metadata. Osoittimen osoittamaan puskuriin tallennetaan viestin sisältö. Paluuarvo nolla kertoo onnistuneesta lukuoperaatiosta. Metodia kutsunut ohjelmalohko pystyy lukemaan tiedoston sisällön lukuoperaation jälkeen parametrina antamastaan puskurista. Parametrina annettu *fileSize*-muuttuja kertoo paljonko tiedostosta luettiin dataa puskuriin.

Datan lukemisen jälkeen *flashOperationsTask*-säie jäsentää vastausviestin, liittää siihen luetun datan ja lähettää viestin *ProductionTester*-säikeelle, joka lähettää viestin edelleen tietokoneelle. Riippumatta siitä, käskettiin *flashOperationsTask*-säiettä kirjoittamaan vai lukemaan tiedosto, suoritetaan kummassakin operaatiossa viimeisenä tiedoston lukeminen. Kirjoittamisen kohdalla käyttäjä voi nähdä, että tiedostosta luettu viesti ei vastakaan kirjoitettua viestiä ja näin ollen kirjoittamisen tai lukemisen aikana on tapahtunut virhe.

## 5.11 Jatkuva muistille kirjoittaminen ja lukeminen

Jatkuvassa kirjoitus- ja lukuoperaatiossa tiedoston lukemisesta ja siihen kirjoittamisesta huolehtii normaaliin tapaan *flashOperationsTask*-säie, mutta säikeen lähettämässä vastausviestissä on enemmän informaatiota kuin normaalissa vastausviestissä. Normaalisti säie lähettää vastausviestissä tiedostosta luetun datan, mutta jatkuvassa kirjoitus- ja lukuoperaatiossa vastausviestissä on ensimmäisenä tietosisältönä tavu, joka kertoo olivatko luettu ja kirjoitettu data samoja. Tämän jälkeen tietosisällössä on kirjoitettu data ja viimeisenä luettu data. Normaalissa kirjoitusoperaatiossa datan maksimimäärä on 250 tavua, mutta jatkuvassa kirjoitus ja lukuoperaatiossa kirjoitettavan datan maksimimäärä on 124 tavua. Tämä johtuu viestin maksimipituudesta ja siitä, että kirjoitettu ja luettu data tulee mahtua vastausviestin tietosisältöön.

Radiolaitteen kuormittamisen aikana radiolaitteelle aiheutetaan häiriöitä. Tämä saattaa aiheuttaa sen, että kirjoitus- tai lukuvaiheessa data korruptoituu, jolloin luettu data ei ole sama kuin kirjoitettu. Tätä varten lähetetään vastausviestissä luettu ja kirjoitettu data. Vastausviestissä olevan muuttujan arvo kertoo datojen vertailun tuloksen. Mikäli kirjoitetun ja luetun datan vertailu vastausviestissä ei ole sama kuin viestissä olevan vertailumuuttujan arvo, on data muuttunut matkalla radiolaittelta tietokoneelle.

## 5.12 Radiolaitteen kuormittaminen

Esimerkiksi HALT- ja ALT-testauksessa radiolaitetta tulee kuormittaa mahdollisimman paljon kuitenkin niin, että kuormittavat toiminnot ovat radiolaitteen normaalia toimintaa. Suurin kuormitus käyttäen normaaleja toimintoja saadaan aikaan, kun lähetetään verkko-prosessorilla dataa jatkuvasti, ja samalla kirjoitetaan ja luetaan flash-muistilta dataa. Luettu data lähetetään sarjaliikenneväylää pitkin tietokoneelle. Jatkuvan datan kirjoittaminen ja lukeminen tapahtuvat lähettämällä *flashOperationsTask*-säikeelle viesti, jossa on kirjoitettava data ja kyseisen testin aloittava arvo. Tällöin säie alkaa toistamaan sekvenssiä, jossa se kirjoittaa annetun datan tiedostoon, lukee datan tiedostosta ja lähettää lukemansa datan *ProductionTester*-säikeelle. Säie toistaa sekvenssiä kunnes radiolaitte sammutetaan.

Tässä tilassa ollessaan radiolaitte ei vastaa uusiin viesteihin, sillä *ProductionTester*-säie jää silmukkaan odottamaan *flashOperationsTask*-säikeen viestejä ja välittää viestit tietokoneelle poistumatta silmukasta. Radiolaitetta kuormittaessa tulee ensiksi laittaa *WlanTask*-säie lähettämään dataa jatkuvasti, jonka jälkeen *flashOperationTask* voidaan käskytää aloittamaan jatkuvan kirjoitus- ja lukuoperaatioiden suorittamisen.

## 6. TESTAUKSEN TUEKSI TOTEUTETTU TIETOKONEOHJELMA

Radiolaitteen testaamista varten toteutettiin tietokoneelle erillinen ohjelma, jonka avulla radiolaitteen sertifiointitestus saatiin suoritettua. Tällä ohjelmalla hallinnoidaan ja käskytetään radiolaitetta. Ohjelma kommunikoi radiolaitteen kanssa ja välittää käyttäjälle tietoja radiolaitteen tilasta. Kommunikoiminen ohjelman ja radiolaitteen välillä tapahtuu sarjaliikenneväylän kautta. Tietokone ja radiolaitte tulee yhdistää kaapelilla, jonka toinen pää on D9-liittimeen ja toinen tietokoneen USB-porttiin sopiva. Kaapelissa tulee olla muunnin, joka muuntaa lähetetyn ja vastaanotetun datan RS-485-standardin mukaiseksi.

Radiolaitteen ohjelmointikielen valintaan vaikutti sekä jo olemassa olevan ohjelman ohjelmointikieli että Texas Instrumentsin julkaiseman ohjelmiston kehitysympäristön ohjelmointikieli. Tietokoneeseen tuleva ohjelma luodaan alusta alkaen käyttämättä aikaisempaa ohjelmakoodia. Tämä mahdollistaa ohjelmointikielen ja -kirjastojen vapaan valitsemisen. Kielen valitsemiseen vaikuttaa ainoastaan vaatimukset käyttöliittymästä sekä kommunikoimisesta radiolaitteen kanssa. Valitulla ohjelmointikielillä pitää pystyä luomaan graafinen käyttöliittymä sekä kommunikoimaan radiolaitteen kanssa käyttäen sarjaliikenneyhteyttä.

### 6.1 Testauksen asettamat vaatimukset käyttöliittymälle

Tietokoneella ajettavassa käyttöliittymässä tulee olla mahdollista lähettää radiolaitteelle viesti, joka aloittaa halutun toiminnallisuuden suorittamisen radiolaitteella. Käyttöliittymässä on erillinen osa niille painikkeille, joilla voidaan käynnistää radiolaitteella sertifiointitestausta varten toteutettu toiminnallisuus. Käyttöliittymän avulla käyttäjän tulee pystyä:

- käynnistämään datan jatkuva yhteydetön lähetys ja valitsemaan käytettävä kanava, tehokkuus sekä lähetysnopeus
- käynnistämään datan jatkuva yhteydetön vastaanottaminen ja valitsemaan käytettävä kanava
- kirjoittamaan radiolaitteen flash-muistille dataa ja määrittämään kirjoitettava data
- lukemaan radiolaitteen flash-muistilta ja näkemään luettu data käyttöliittymässä
- käynnistämään jatkuva flash-muistille luku ja kirjoitus, määrittämään kirjoitettava data, näkemään luettu data sekä näkemään, jos luettu ja kirjoitettu data eroavat toisistaan

- käynnistämään datan jatkuva yhteydetön lähetys valitulla kanavalla, valitulla tehokkuudella sekä lähetysnopeudella, niin että kanavan käyttöaste on yli 85 %
- luomaan radiolaitteesta tukiasema ja määrittämään tukiaseman verkkotunnus, salausana sekä portti
- yhdistämään radiolaitte jo olemassa olevaan tukiasemaan
- katkaisemaan luotu yhteys toiseen radiolaitteeseen tai sammuttamaan luotu tukiasema
- aloittamaan yhteyden tarkkailemisen kahden radiolaitteen välillä.

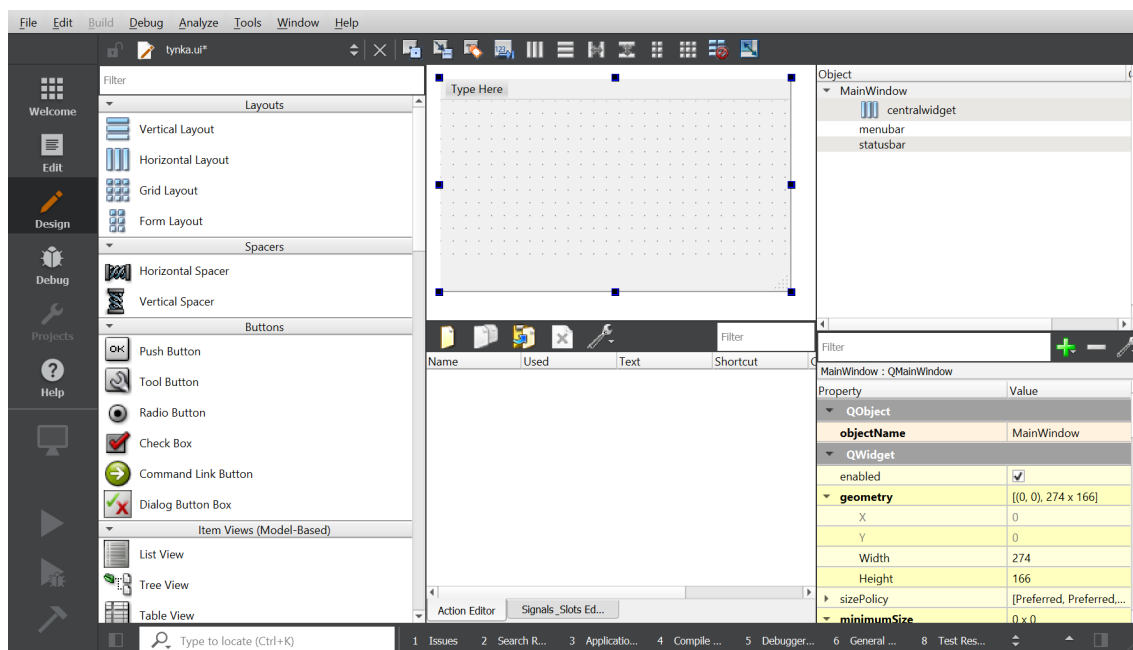
Yksittäisen painikkeen painaminen käynnistää ohjelmalohkon, joka lähettää radiolaitteelle sarjaliikenneyhteyttä pitkin toiminnallisuuden aloittavan viestin. Aloitettu toiminnallisuus riippuu painetusta painikkeesta. Jokaisen painikkeen nimen tulee olla niin selvä, että käyttäjä ymmärtää, minkä toiminnallisuuden painikkeen painaminen radiolaitteessa käynnistää. Käyttäjä pystyy vaikuttamaan joihinkin toiminnallisuuksiin. Esimerkiksi flash-muistille kirjoittaessaan käyttäjä määrittelee kirjoitettavan datan. Käyttöliittymässä tulee olla jokaiselle tällaiselle määritettävälle parametrille oma kenttänsä, johon käyttäjä voi syöttää haluamansa parametrin arvon.

## 6.2 Valittu ohjelmointikieli ja käytetyt kirjastot

Tietokoneohjelman ohjelmointikieleksi valittiin Python (versio 2.7.13) [31]. Työn tekijällä oli aikaisemmin kokemusta Python-ohjelmointikielestä sekä kirjastosta, jonka avulla graafinen käyttöliittymä voitiin toteuttaa. Ohjelmointikielen valintaan vaikutti myös aikataulliset syyt. Aikataulun takia uutta ohjelmointikieltä ei ollut mahdollista alkaa opettelemaan. Graafisen käyttöliittymän luomiseen käytettiin apuna PySide-kirjastoa [32]. PySide on Qt Projectin luoma kirjasto, jonka avulla Python-ohjelmointikielellä pystytään käyttämään Qt Projectin kehittämää Qt-sovellusalustaa. Käyttöliittymä voidaan luoda erillisellä Qt Creator -nimisellä editorilla.

Editorin suunnittelunäkymä on havainnollistettu kuvassa 5. Pohjana toimivaan ikkunaan pystyy lisäämään vetämällä eri elementtejä, kuten esimerkiksi painikkeita, tekstikenttiä, tekstiä ja eri säiliöitä. Alkuperäisen ikkunan kokoa voi muokata vapaasti. Ikkunaan on myös mahdollista lisätä widget-elementti. Tämän elementin sisälle voidaan liittää toinen erilainen ikkuna, joka on luotu erillisenä projektina Qt Creator -ohjelmalla. Tällaisen widget-elementin näkyvyys voidaan sijoittaa painikkeeseen ja elementin tilaa voidaan vaihtaa ajon aikana esimerkiksi näkymättömästä näkyväksi. Editorilla on myös mahdollista luoda rajallisesti ”signals and slots” -ajattelun mukaista toimintaa ilman, että käyttäjän täytyy ohjelmoida riviäkään.

Editori tallentaa luodun käyttöliittymän XML-merkintäkielenä tiedostoon. Luotu käyttöliittymä voidaan kääntää PySide-kirjaston avulla Python-kielelle. PySide-kirjasto tarjoaa rajapinnan Qt-sovellusalustaan, jolloin käyttöliittymä voidaan tuoda osaksi ohjelmaa.



*Kuva 5. Käyttöliittymän luominen.*

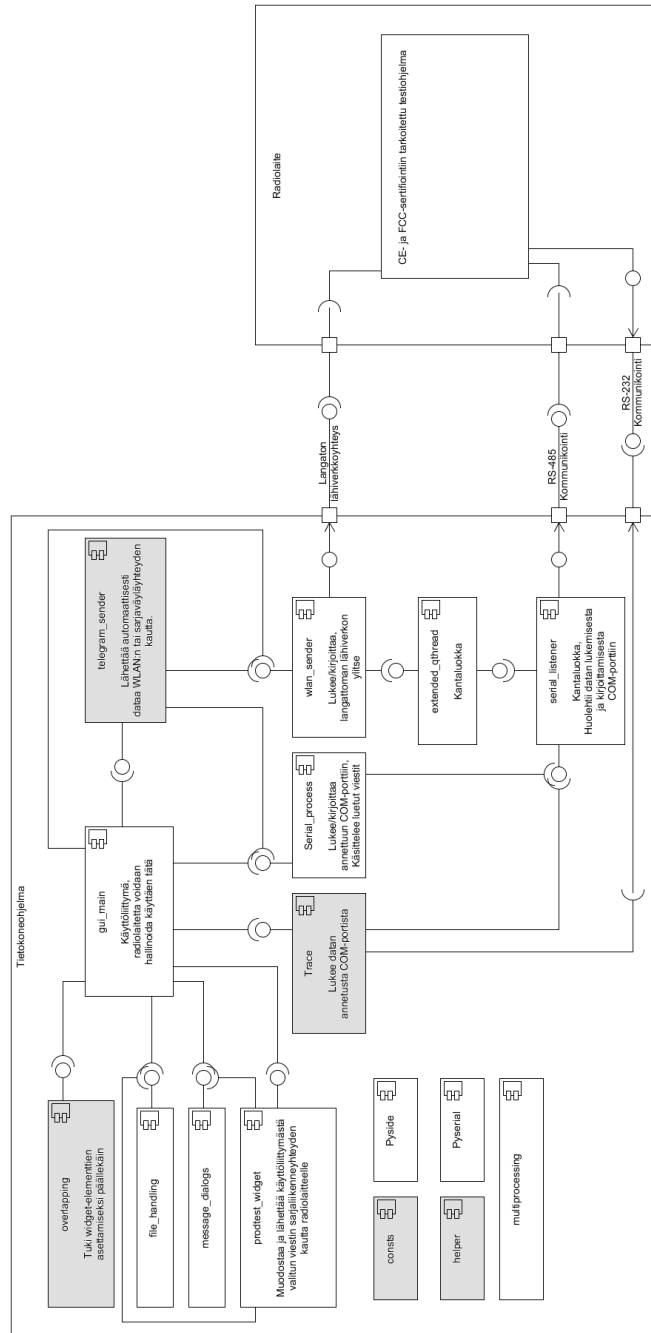
Käyttöliittymän luominen toimii pohjana ohjelmalle ja sitä voidaan muuttaa ohjelmallisesti. Esimerkiksi käyttöliittymässä oleviin painikkeisiin voidaan liittää laukaisimia ja laukaisimiin funktioita. Kun painiketta painetaan, suorittaa ohjelma laukaisimeen liitetyn funktion.

Sarjaliikennekommunikoinnista varten Python-ohjelmassa käytetään pySerial-nimistä kirjastoa. Tämä kirjasto kapseloi pääsyn sarjaliikenneporttiin ja käyttämällä kirjaston määrittämää rajapintaa voi käyttäjä lukea ja lähettää sarjaliikenneväylälle dataa. Kirjasto tukee sarjaliikenteen käyttämistä eri konfiguraatioilla. Käyttäjä voi konfiguroida seuraavia asioita:

- tietosisällön bittien määrä
- pysäytys bittien määrä
- käytettävä siirtonopeus
- pariteetin käyttö ja määrä
- kättelyn käyttö ja tapa.

Kaikki yllämainitut asiat liittyvät sarjaliikenneväylällä käytävään viestittelyyn. Ainoastaan käytettävä siirtonopeus ei vaikuta lähetettävän viestin sisältöön. Jotta kommunikointi olisi mahdollista, tulee tietokoneen ja radiolaitteen konfiguraatioiden olla samat.





Kuva 6. Tietokoneohjelman arkkitehtuuri esitetty komponenttikaaviona.

### 6.3 Tietokoneohjelman arkkitehtuuri

Tietokoneohjelma luotiin alunperin radiolaitteen kehittämisen ja testaamisen tueksi. Sertifiointitestausta varten käyttöliittymään lisättiin tuki viestien lähettämiseksi painikkeita painamalla. Käyttöliittymä sisältää myös sellaisia toiminnallisuuksia, joita ei sertifiointitestauksessa tarvita. Tällainen toiminnallisuus ohjelmoitiin käyttöliittymään radiolaitteen kehittämisen tueksi. Sertifiointitestaukseen tarkoitettussa osassa tietokoneohjelmaa on kolme säiettä. Ensimmäinen on nimeltään *GuiMain*-säie ja se huolehtii käyttöliittymä-

mään liittyvästä toiminnallisuudesta sekä hallinnoi muita säikeitä. Toisen säikeen nimi on *SerialListener* ja se huolehtii datan lukemisesta sarjaliikenneväylältä sekä kirjoittamisesta sarjaväylälle. Kolmas säie on nimeltään *SerialProcess*. Tämä säie vastaa lähetetyn ja luetun datan käsittelystä sekä parsimisesta oikeaan muotoon. Tietokoneohjelma on jakautunut näiden kolmen säikeen mukaisesti kolmeen suurempaan osaan, jotka kommunikoivat keskenään ja hyödyntävät ohjelman pienempiä osia. Tietokoneohjelman arkkitehtuuri on esitetty komponenttikaaviona kuvassa 6. Kuvassa näkyy harmaalla pohjalla osat, joita ei tarvita sertifiointitestauksessa, eikä niitä käsitellä tässä luvussa.

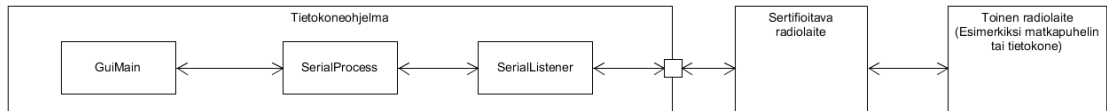
Käyttöliittymästä vastaavan säikeen keskeisin osa on *gui\_main*-näkymä. Tässä näkymässä tapahtuu käyttöliittymän kenttien päivittäminen, sekä painikkeihin liitettyjen funktioiden laukaiseminen. Toinen tärkeä osa käyttöliittymästä on *proctest\_widget*-näkymä. Tämä näkymä sisältää sertifiointitestaukseen tarkoitetut painikkeet, kentät ja funktiot, joiden avulla käyttäjä voi käynnistää radiolaitteella halutun toiminnallisuuden. Säie *GuiMain* hallinnoi näitä molempia osia ja liittää *proctest\_widget*-näkymän ajonaikaisesti osaksi *gui\_main*-näkymää.

Käyttöliittymä käyttää lisäksi kahta pienempää kooditiedostoa. Ensimmäinen on *message\_dialogs*-tiedosto, jonka tehtävä on tarjota tuki erilaisten valintaikkunoiden luomiselle ajonaikaisesti. Valintaikkuna luodaan esimerkiksi silloin, kun käyttäjä haluaa käynnistää jatkuvan muistille kirjoittamisen ja lukemisen. Tämän toiminnallisuuden käynnistäminen estää kaiken muun toiminnallisuuden käynnistämisen radiolaitteella. Valintaikkunalla varmistetaan, että käyttäjä haluaa varmasti käynnistää toiminnallisuuden.

Toinen käyttöliittymän käyttämä tiedosto on *file\_handling*-tiedosto, joka tarjoaa tuen tiedostojen käsittelemiselle. Käyttöliittymässä on asetuksia ja nämä voidaan tallentaa tiedostoon. Kun ohjelma käynnistetään, asetustiedosto luetaan, ja sen sisältämät valinnat asetetaan voimaan ohjelmassa. Käyttöliittymä näyttää radiolaitteelle lähetetyt ja vastaanotetut viestit. Nämä viestit on mahdollista tallentaa tiedostoon myöhempää lukemista varten. Koska käyttöliittymässä näytetään myös painikkeiden kautta lähetetyt viestit, voidaan sertifiointitesteissä ajatut sekvenssit myös tallennetaa tiedostoon. Näin testit voidaan myöhemmin suorittaa tarvittaessa uudelleen.

Sarjaliikennekommunikoinnista huolehtii *SerialListener*-säie, joka koostuu saman nimisestä luokasta. Säie toimii pohjana sarjaliikennekommunikoinnin luomiselle ja se sisältää toteutuksen sarjaliikenneyhteyden luomiselle ja konfiguroimiselle. Datan lähettäminen ja vastaanottaminen on myös toteutettu tämän luokan sisällä. Säikeellä ei ole muuta tehtävää kuin huolehtia sarjaliikennekommunikoinnista.

Sarjaliikennekommunikoinnissa lähetetään kahdeksan bittiä viestin tietosisällössä, joten yhden kokonaisen käskyn lähettäminen käyttöliittymästä radiolaitteelle vaatii useita viestejä. Sarjaliikenneviestien sisältöä täytyy analysoida, jotta viesteistä saadaan jäsenneltyä yksi kokonainen radiolaitteen lähettämä vastausviesti. Sarjaliikennekommunikoimisessa käytetyt viestit jäsennellään ja ne käsittelee *SerialProcess*-säie. Säie vastaa, että lähetet-



**Kuva 7.** Viestin lähettäminen tietokoneohjelmasta sarjaliikenneyhteyttä pitkin radiolaitteelle.

ty ja vastaanotettu data on rakenteeltaan oikeellista, että ja yksittäinen vastausviesti sisältää oikean määrän dataa. Sääkeessä on tuki myös tiettyihin viesteihin automaattiselle vastaamiselle, mutta tällaista toiminnallisuutta ei tarvita sertifiointitesteissä. Tuki viestien automaattiselle vastaamiselle erotti *SerialProcess*-sääkeen erilliseksi kokonaisuudeksi *SerialListener*-sääkeestä.

Sekä *SerialListener*- että *SerialProcess*-sääke on periyetty *ExtendedQThread*-luokasta. Kyseinen luokka luo perustoiminnallisuuden ja pohjan näille sääkeille, kuten sääkeiden tilanhallinnan, siihen liittyvän rinnakkaisuuden hallinnan sekä viestisignaalien määrittämisen. Luokka on puolestaan periyetty PySide-kirjaston tarjoamasta *QThread*-luokasta, joka mahdollistaa sääkeistämisen. Sääkeistettävä luokka tulee periä *QThread*-luokasta, jotta luokasta luotua instanssia voitaisiin ajaa sääkeenä.

## 6.4 Tietokoneohjelman sääkeiden välinen kommunikointi

Käyttöliittymän näkymien välinen viestintä tapahtuu osittain signaalien välityksellä. Signaalit ovat PySide-kirjaston tuoma ominaisuus. Kun *prodtest\_widget*-näkyessä painetaan painiketta, lähettää se *gui\_main*-näkykälle signaalin. Signaali on sidottu *gui\_main*-näkykän puolella sopivaan funktioon, jota kutsutaan, kun signaali saadaan otettua vastaan. Signaalit sisältävät dataa, joka puolestaan riippuu signaalista. Tämä tarkoittaa näkykän *prodtest\_widget* tapauksessa yleensä radiolaitteelle tarkoitettua viestiä. Koska *prodtest\_widget*-näkykää lisätään *gui\_main*-näkykään ajonaikaisesti, on *gui\_main*-näkykällä tieto *prodtest\_widget*-näkykämästä ja mahdollisuus käsitellä *prodtest\_widget*-näkykää ilman tarvetta signaalien lähettämiseen. Näitä molempia näkymiä hallinnoi *GuiMain*-sääke, vaikka näkymien välinen viestintä tapahtuukin osittain signaaleilla. Molemmat näkykät ovat erillisiä kokonaisuuksia ja ne on luotu Qt Editor -ohjelmalla erillisinä näkyminä.

Radiolaitteelle tarkoitettu viesti saa alkunsa *GuiMain*-sääkeeltä. Viestin välittämiseen osallistuvat sääkeet ja viestin eteneminen on kuvattu kuvassa 7. Sääke lähettää viestin *SerialProcess*-sääkeelle, joka puolestaan välittää viestin *SerialListener*-sääkeelle. Näiden kolmen sääkeen välinen kommunikointi tapahtuu käyttäen *multiprocessing*-kirjaston *Queue*-toiminnallisuutta. Sääke *SerialProcess* toimii viestiketjun välissä ja välittää *GuiMain*- ja *SerialListener*-sääkeiden toisilleen lähettämiä viestejä. Näin *SerialProcess*-sääke pystyy analysoimaan ja jäsentämään lähetettyä ja vastaanotettua dataa tarvittaessa. Tällä tavalla *GuiMain*-sääke ei tiedä *SerialListener*-sääkeen olemassaolosta. Sääke *SerialProcess* luo ja huolehtii *SerialListener*-sääkeestä.

Luokka *ExtendedQThread* määrittelee kaksi signaalia, joista toinen on tarkoitettu virheviestien lähettämistä varten ja toinen käyttöliittymän tekstikenttien päivittämiseen (*messageSignal* ja *errorSignal*). Koska *SerialProcess*- ja *SerialListener*-säikeet ovat periytetty *ExtendedQThread*-luokasta, omaavat myös ne nämä kaksi signaalia. Kaikki *SerialListener*-säikeen emittoimat signaalit menevät *SerialProcess*-säikeelle, joka emittoi ne uudelleen. Säie *GuiMain* on liittänyt *SerialProcess*-säikeen signaalit sopiviin funktioihin. Tällä tavalla *GuiMain*-säie saa myös vastaan *SerialListener*-säikeen lähettämät signaalit, vaikka nämä kaksi säiettä eivät näekään toisiaan.

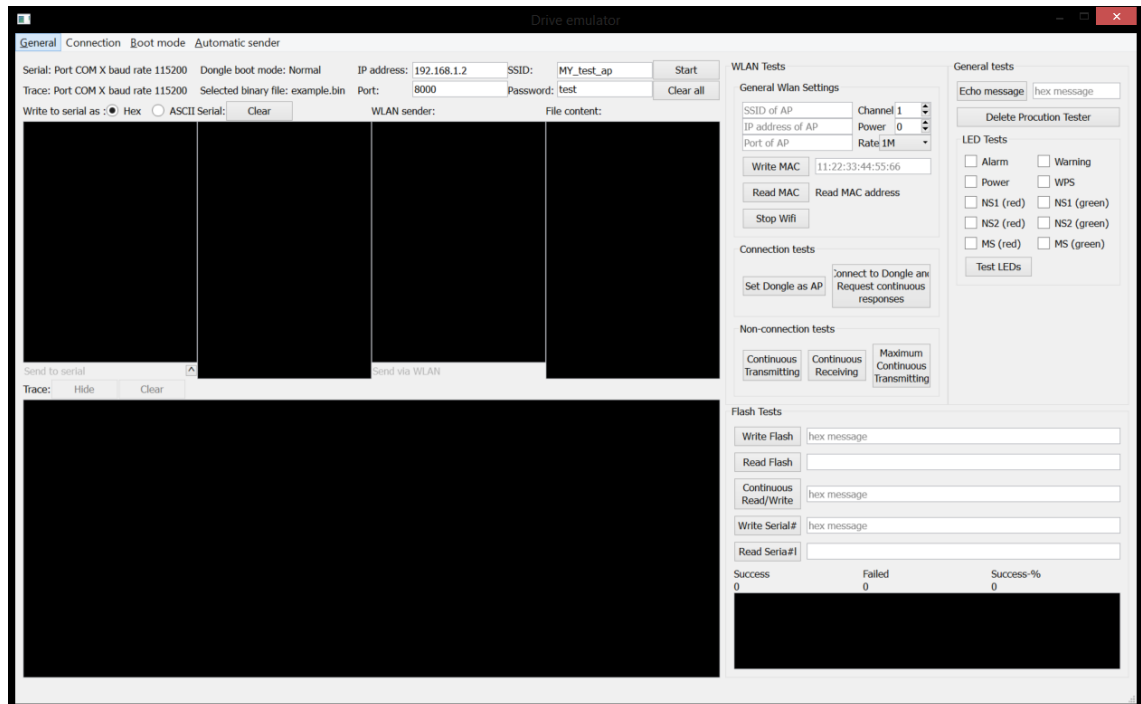
Esimerkiksi käyttäjän antamasta virheellisestä parametrissa sarjaliikenneyhteyttä alustaessa aiheutuu virheviesti. Säie *SerialListener* lähettää virheviestin signaalilla eteenpäin, joka päättyy lopulta *GuiMain*-säikeelle. Signaalin sisältämän datan mukaan luodaan valintaikkuna, joka näkyy käyttäjälle. Näin käyttäjä saa tiedon virheellisestä parametrissa. Säikeiden välillä *Queue*-toiminnolla lähetetään sellaisia viestejä, jotka lähetetään myös radiolaitteelle. Signaalit ovat puolestaan tarkoitettu käyttöliittymän päivittämiseksi tai sen muuttamiseksi.

## 6.5 Käyttöliittymän käyttäminen ja kommunikoiminen radiolaitteen kanssa

Kommunikoiminen radiolaitteen kanssa tapahtuu lähettämällä ja lukemalla viestejä sarjaliikenteen ylitse vuorotellen. Radiolaitteelle vastaa jokaiseen saamansa viestiin, muutamaa poikkeusta lukuunottamatta. Tietokoneohjelma toimii samalla tavalla, mutta tietokoneohjelma aloittaa aina kommunikoimisen radiolaitteen kanssa. Ennen kuin kommunikoiminen radiolaitteen kanssa voidaan kuitenkaan aloittaa, tulee sarjaliikenneyhteys konfiguroida ja käynnistää *SerialProcess*- sekä *SerialListener*-säikeet. Tietokoneohjelman käynnistyttyä käyttäjä voi valita käytettävän lähetyksenopeuden ja portin käyttöliittymän työkalupalkista ”Connection”-kohdan alta. Luotu tietokoneohjelman käyttöliittymä näkyy kuvassa 8. Käyttäjän tulee valita se kommunikointiportti (engl. communication port), johon USB-RS485-johto on kytketty ja lisäksi oikea lähetyksenopeus. Tämän jälkeen käyttäjä voi käynnistää muut säikeet painamalla käyttöliittymän ”Start”-painiketta. Virheettömän käynnistymisen jälkeen käyttäjä voi lähettää ja vastaanottaa viestejä radiolaitteelta.

Käyttöliittymän *proctest\_widget*-näkyvässä on painikkeita, jotka lähettävät automaattisesti viestin radiolaitteelle. Tämä näkymä on oletusarvoisesti asetettu näkymättömäksi, mutta näkymän saa työkalupalkista asetettua näkyväksi painamalla ”General”-kohdan alta ”Show/hide production tests”. Näkymä ilmestyy käyttöliittymän oikealle puolelle, jonka jälkeen näkymän painikkeita painamalla käyttäjä voi lähettää viestejä radiolaitteelle.

Painikkeen painaminen *proctest\_widget*-näkyvässä luo radiolaitteelle tarkoitetun viestin. Viesti käynnistää painikkeesta riippuvaisen toiminnallisuuden. Esimerkiksi ”Set Dongle as AP”-napin painaminen luo viestin, jolla radiolaitte tekee itsestään tukiaseman. Viestin luova funktio saattaa lukea käyttöliittymästä myös eri elementtien sisältöä ja lisätä nii-



**Kuva 8.** Sertifiointitestausta varten luotu käyttöliittymä.

den sisältämiä arvoja viestiin. Esimerkiksi ”Set Dongle as AP”-napin tapauksessa viestin muodostava funktio lukee myös ”SSID of AP”-, ”IP address of AP”-, ”Port of AP”-, ”channel”-, ”power”- ja ”rate”-kenttien arvot, koostaa tiedoista oikeanlaisen viestin ja lähettää viestin eteenpäin. Viesti päättyy lopulta radiolaitteelle, joka käsittelee viestin ja tekee sen sisältämän datan mukaisesti halutun toiminnallisuuden.

Radiolaitte lähettää myös vastausviestin, joka riippuu radiolaitteen silloisesta tilasta ja sen saamasta viestistä. Viesti lähetetään sarjaliikenneyhteyden kautta, joten *SerialListener*-säie vastaanottaa datan ja lähettää sen eteenpäin *SerialProcess*-säikeelle. Kaikki *SerialListener*-säikeen saama data lähetetään sellaisenaan aina *SerialProcess*-säikeelle. Koska vastausviesti saapuu useammassa viestissä, *SerialProcess*-säie joutuu kokoamaan vastausviestin. Jokaisessa radiolaitteen ja tietokoneen lähettämässä viestissä on toisena tavuna koko viestin pituus, joten *SerialProcess*-säie tietää montako tavua koko viestiin kuuluu. Saatuaan vastausviestin säie välittää sen *GuiMain*-säikeelle.

## 6.6 Käyttöliittymän tarjoama tuki radiolaitteen käyttämiseen

Valmiilla tietokoneohjelmalla pystytään ohjaamaan radiolaitteen toimintaa sertifiointitestausten aikana ja käynnistämään radiolaitteella haluttu toiminnallisuus kunkin testin suorittamista varten. Suurimmassa osassa sertifiointitesteistä riittää, että radiolaitteella saadaan käynnistettyä yksittäinen toiminnallisuus. Kuitenkin muutamassa testissä tietokoneohjelman tulee pystyä analysoimaan radiolaitteelta saamia viestejä. Kaksi tällaista testitapausta ovat kahden radiolaitteen välisen yhteyden tarkkaileminen sekä jatkuva muistille kirjoittamisen ja lukemisen onnistumisen seuraaminen.

Näissä molemmissa testeissä ei riitä pelkästään viestien lähettäminen ja vastaanottaminen, vaan näitä toimintoja varten käyttöliittymässä tulee olla tuki vastausten automaattiselle seuraamiselle ja lokittamiselle. Toisin kuin muut radiolaitteeseen ohjelmoidut toiminnallisuudet, näiden kahden toiminnallisuuden käynnistäminen aiheuttaa sen, että radiolaite lähettää jatkuvasti viestejä sarjaliikenneyhteyden ylitse tietokoneohjelmalle. Tällöin tietokoneohjelmalla ei ole enää turvallista tai tarkoituksenmukaista lähettää viestejä radiolaitteelle, sillä radiolaite ei vastaa lähetettyihin viesteihin. Käyttöliittymässä olevien painikkeiden käyttö estetään tällaisessa tilanteessa.

### 6.6.1 Kahden radiolaitteen välisen yhteyden tarkkailu

Käyttöliittymällä on toteutettu erikseen tuki kahden radiolaitteen välisen yhteyden tarkkailemiselle. Tällaisessa sertifiointitestauksessa tarvitaan kaksi radiolaitetta ja kaksi USB-RS485-johtoa. Tietokoneohjelmasta voidaan luoda kaksi instanssia, jolloin toinen instanssi kommunikoi tukiasemaksi asetettavan radiolaitteen kanssa ja toinen instanssi asiakkaaksi asetettavan radiolaitteen kanssa.

Radiolaitteen voi asettaa tukiasemaksi painamalla käyttöliittymästä ”Set dongle as AP” -painiketta (kuva 8). Käyttäjä voi myös määrittää verkkotunnuksen, kanavan, käytettävän tehon sekä nopeuden, jota tukiasema tulee käyttämään. Kun toisesta radiolaitteesta on onnistuttu muodostamaan tukiasema, voidaan toinen radiolaite yhdistää tukiasemaan painamalla ”Connect to Dongle and Request continuous responses” -painiketta. Tällöin radiolaite yrittää yhdistää tukiasemaan, jonka verkkotunnuksen käyttäjä on antanut käyttöliittymään. Yhteyden muodostamisen jälkeen asiakas lähettää saman tien tukiasemalle viestin muodostetun langattoman lähiverkkoyhteyden kautta. Tämän jälkeen tukiasema alkaa jatkuvasti lähettämään viestejä asiakkaalle. Asiakas ohjaa saamansa viestit tietokoneohjelmalle. Jokainen viesti sisältää tietosisältönä yhden numeron välillä 0–255 ja kahden perättäisen viestin välillä numero kasvaa aina yhdellä. Tukiasema lähettää viestin sekunnin välein.

Tietokoneohjelma näyttää käyttöliittymässä jokaisen asiakkaalta saamansa viestin ja seuraa viestissä olevaa juoksevaa numerointia. Mikäli kahden perättäisen viestin välillä on suurempi kuin yhden numeron kasvu, on asiakkaalta jäänyt yksi tai useampi viesti saamatta ja välittämättä tietokoneohjelmalle. Tällöin tietokoneohjelma luo valintaikkunan, jossa se kertoo käyttäjälle montako viestiä asiakkaalta on jäänyt saamatta. Näin sertifiointitesteissä voidaan huomata, jos radiolaitteita on onnistuttu häiritsemään niin, että muodostettu yhteys on hetkeksi katkennut.

### 6.6.2 Muistille kirjoittamisen ja lukemisen onnistumisen seuraminen

Jatkuva muistille kirjoittaminen ja lukeminen voidaan käynnistää käyttöliittymän ”Continuous read/write” -painiketta painamalla (kuva 8). Käyttäjän tulee antaa myös heksa-

datana viesti, jota radiolaite alkaa flash-muistilleen kirjoittamaan. Viesti annetaan ”Continuous read/write” -painikkeen vasemmalla puolella olevaan tekstikenttään. Painikkeen painamisen jälkeen ohjelma luo valintaikkunan, jonka avulla voidaan varmistaa käyttäjän halukkuus aloittaa kyseinen toiminnallisuus. Mikäli käyttäjä painaa valintaikkunan ”Yes”-painiketta, radiolaite aloittaa muistille kirjoittamisen ja lukemisen. Tällöin radiolaite alkaa lähettämään jatkuvan kirjoitus- ja lukuoperaation tulosta vastausviesteinä tietokoneohjelmalle.

Tietokoneohjelma aktiivisesti jäsentää ja tarkastelee radiolaitteelta saamiensa viestien sisältöä ja operaation tulosta. Käyttöliittymässä on erillinen tekstialue ”Read Serial#” -painikkeen alla. Tähän alueeseen tietokoneohjelma lisää kaikki vastausviestit, joissa operaatio on epäonnistunut. Tämän kentän yläpuolella on kolme kohtaa, joista yksi näyttää montako onnistunutta operaatiota radiolaite on suorittanut, yksi montako epäonnistunutta operaatiota ja viimeinen näyttää montako prosenttia kaikista suoritetuista operaatioista on onnistunut. Käyttäjä näkee helposti jatkuvan muistille kirjoittamisen ja lukemisen tulokset.

### 6.6.3 Viestien kirjaaminen lokeihin

Jokainen *GuiMain*-säikeen lähettämä ja vastaanottama viesti lisätään käyttöliittymässä olevaan tekstikenttään, jotta käyttäjä näkee radiolaitteen ja tietokoneohjelman käymän viestienvaihdon. Viestien edessä on aina aikaleima ja etuliite, joka kertoo lähetettiinkö vai vastaanotettiinkö viesti. Tämä tekstikenttä on sijaitsee ”Serial:”-tekstin alla (kuva 8, sivu 52).

Jokaisen tekstiä sisältävän tekstikentän sisältö tallennetaan tiedostoon, kun ohjelma suljetaan. Ohjelman sulkemisen aikaan luodaan kansio, jonka nimenä toimii sen hetkinen aikaleima. Kansio luodaan ”prodtest\_results” -nimisen hakemiston alle. Tämä hakemisto puolestaan sijaitsee ”tests”-hakemistossa, joka on tietokoneohjelman juurihakemistossa. Luoduista tiedostoista nähdään millaisia viestejä radiolaitteen ja tietokoneohjelman välillä on lähetetty. Tekstitiedostoja voidaan käyttää myöhemmin testisekvenssien toistamiseen tai niiden kattavuuden varmistamiseen.

## 7. TULOKSET JA KEHITYSEHDOTUKSET

Työn tarkoituksena oli luoda kaksi erillistä ohjelmaa, joiden avulla kehitetylle radiolaitteelle pystytään suorittamaan sertifiointitestausta. Työn tavoitteiden mukaisesti työssä myös määritettiin radiolaitteen sertifoimista varten vaaditut toiminnallisuudet. Sertifiointitestauksen suoritti kolmas osapuoli. Valmis tietokoneohjelma asennettiin kannettavalle tietokoneelle ja radiolaitteelle asennettiin työssä toteutettu ohjelma. Graafisen käyttöliittymän ulkoasu muotoutui kuvan 8 (sivulla 52) mukaiseksi. Käyttöliittymässä on myös useita ominaisuuksia, joita ei tarvita sertifiointitestaamiseen. Käyttöliittymän avulla pystytään esimerkiksi luomaan yhteys radiolaitteen ja tietokoneen välille sekä lähettämään että vastaanottamaan näiden kahden laitteen välillä paketteja. Luotu tietokoneohjelma oli ohjeistuksen kanssa tarpeeksi selkeä, jotta kolmas osapuoli pystyi käynnistämään haluamansa toiminnallisuuden. Radiolaitteelle luotu ohjelma sisälsi kaikki toiminnallisuudet, joita tarvittiin sertifiointitestauksessa. Lisäksi käyttöliittymään luodut sertifiointitestausta helpottavat ominaisuudet toimivat moitteettomasti ja nopeuttivat sertifiointitestausta.

Tietokoneella pystyttiin lähettämään radiolaitteelle komento, joka määräsi radiolaitteella käynnistettävän toiminnallisuuden. Jokaisessa sertifiointitestauksessa käytettävässä radiolaitteessa oli sama ohjelma, jota ohjattiin tietokoneohjelman käyttöliittymän avulla. Tämä mahdollisti yhden radiolaitteen käyttämisen useassa erilaisessa sertifiointitestauksessa. Tämän takia testauksen suorittavalla osapuolella ei ollut pelkoa radiolaitteiden sekoittamisesta keskenään, sillä jokaisella radiolaitteella pystyttiin tekemään haluttu sertifiointitesti. Sertifiointitestausta varten ei tarvinnut tämän takia lähettää kuin muutama radiolaitte. Mikäli yhdelle radiolaitteelle olisi aina kovakoodattu yksi toiminnallisuus, olisi radiolaitteita jouduttu lähettämään sertifiointitestausta varten useampia. Sertifiointitestauksen suorittavalle taholle toimitettiin sertifiointitestaukseen vaadittavat ohjelmat, dokumentit, laitteet ja muut tarpeelliset materiaalit, jotta testaus saataisiin suoritettua.

Ensimmäinen sertifiointitestausta ei onnistunut, sillä radiolaitteella ei pystytty lähettämään dataa langattomaan lähiverkkoon tarpeeksi nopeasti. Vaatimuksissa mainittiin, että verkkoprosessorin tulee lähettää dataa langattoman lähiverkon valittuun kanavaan niin tiheästi, että kanavan käyttöaste on yli 85 %. Ensimmäisellä toteutuksella vaadittuun lukuun ei päästy, vaan kanavan käyttöaste oli noin 40 %, kun dataa lähetettiin jatkuvasti. Ensimmäisessä toteutuksessa *WlanSubTask*-säie käski verkkoprosessorin lähettämään 100 tavun viestejä yksitellen. Kun verkkoprosessori sai lähetettyä viestin, *WlanSubTask*-säie käski verkkoprosessoria lähettämään seuraavan viestin. Verkkoprosessori joutui odottamaan *WlanSubTask*-säikeen käskyä aina lähetettyään viestin. Tämä säie ei kuitenkaan ollut välttämättä ajossa siinä kohtaa, kun verkkoprosessori sai datan lähetettyä. Tämän seurauksena verkkoprosessori joutui odottamaan suhteellisen kauan, ennenkuin se sai *WlanSubTask*-säikeeltä seuraavan käskyn viestin lähettämiseksi.



Verkkoprosessori lähettää viestin langattomaan lähiverkkoon *sl\_Send*-metodin avulla. Tälle funktiokutsulle annetaan viimeisenä parametrina toistojen määrä. Aikaisemmassa toteutuksessa toistojen määrä oli yksi jokaisella kutsukerralla. Asettamalla toistojen määräksi nolla, verkkoprosessori jatkaa annetun datan lähettämistä niin kauan, kuin verkkoprosessori on käynnissä. Tällä tavalla *WlanSubTask*-säikeen ei tarvitse käskää verkkoprosessoria kuin kerran koko toiminnallisuuden aikana.

Lisäksi radiolaitteen lähetystehoja jouduttiin ohjelmallisesti laskemaan, kun radiolaite käyttää kanavaa kuusi. Liian suuri lähetysteho kanavalla kuusi aiheutti sen, ettei laite läpäissyt sertifiointivaatimuksia. Näiden korjausten jälkeen radiolaite läpäisi sertifiointitestit onnistuneesti. Radiolaitetta ja tietokoneohjelmaa ohjelmoidessa ilmeni kohteita, joita voisi tulevaisuudessa kehittää, tai jotka voisi tehdä toisin, mikäli tietokoneohjelma tai radiolaitteella olevaa ohjelmaa kehitettäisiin jatkossa.

Sertifiointitestauksessa käytettävät ohjelmat täyttivät vaatimukset, mutta kaikki ohjelmalliset ratkaisut eivät olleet jatkokehittämisen kannalta parhaimpia mahdollisia. Nämä ratkaisut esitellään sekä niiden jatkokehittämistarve perustellaan tässä luvussa. Lisäksi tietokoneohjelman kehityksen aikana ilmeni muutamia uusia ominaisuuksia, jotka olisi haluttu osaksi ohjelmaa. Tämä ei kuitenkaan ollut mahdollista tiukan aikataulun vuoksi. Radiolaitteelle asennettavan ohjelman lopullinen arkkitehtuuri ei vastannut uusimpia vaatimuksia ja arkkitehtuurissa näkyy poistuneiden vaatimusten takia tehtyjä ratkaisuja. Yksi tällainen on esimerkiksi *WlanTask*- ja *WlanSubTask*-säikeiden olemassaolo.

Kumpaankaan ohjelmaan ei luotu minkään näköistä automaattista testausta. Automaattista testausta ei ehditty toteuttamaan tiukan aikataulun takia. Tämä tarkoittaa, että ohjelmia kehittäessä testaus on suoritettu aina manuaalisesti. Automaattisella testauksella tarkoitetaan esimerkiksi integraatio- ja järjestelmätestausta, mutta myöskään yksikkötestejä ei ole luotu kumpaakaan ohjelmaan. Kehitysprosessin aikana luotujen toiminnallisuuden varmistamiseen manuaalisella testauksella on kuluttanut paljon aikaa ja jatkokehittämisen ja ylläpidettävyyden kannalta automaattisen testauksen tekeminen olisi suotavaa. Vaikka testien kehittäminen veisikin työtunteja uusien toiminnallisuuksien kehittämiseltä, olisi se laadun varmistamisen ja koko kehitysprosessiin kannalta hyväksi.

## 7.1 Sekvenssien luominen ja ajaminen

Sertifiointitestauksessa käytettävässä tietokoneohjelmassa ei ole tukea testien automaattiselle suorittamiselle tai luomiselle. Ohjelman käyttäjä joutuu ohjaamaan radiolaitteen toimintaa käyttöliittymän nappien avulla, vaikka testauksessa käytettävä testisekvenssi olisi koko ajan sama. Yksi kehityskohde tietokoneohjelmassa voisi olla testisekvenssien luominen ja niiden ajaminen. Käyttäjä voisi määrittää käyttöliittymän nappien avulla haluamansa sekvenssin, asettaa askeleiden välille viivytyksiä ja määrittää millaisen vastauksen radiolaitteelta pitäisi saada missäkin kohtaa. Testisekvenssi voitaisiin tallentaa tietokoneelle ja käyttöliittymässä voisi olla tuki tällaisen tiedoston lataamiselle ja ajamiselle.

Nykyisessä toteutuksessa nappien painaminen luo heksadesimaaliviestin, joka lähetetään sellaisenaan radiolaitteelle. Käyttäjä näkee ohjelman lähettämän heksadesimaaliviestin. Viestien tunnistaminen pelkästään heksadesimaalijonosta on haastavaa, joten viestit tulisi näyttää käyttäjälle luodussa sekvenssissä helposti ymmärrettävässä muodossa. Käyttäjän pitäisi pystyä helposti ajamaan, muuttaamaan ja tallentamaan sekvenssejä käyttöliittymällä. Lisäksi käyttöliittymässä voitaisiin määritellä kuinka monta kertaa sekvenssi halutaan ajaa. Tällä tavalla yksinkertaista sekvenssiä ei tarvitsisi toistaa montaa kertaa, ja sekvenssejä voitaisiin käyttää apuna myös radiolaitteen mahdollisessa jatkokehittämisessä.

## 7.2 Dynaaminen muistinhallinta säikeiden välisessä kommunikoinnissa

Radiolaitteella olevassa ohjelmassa säikeiden välinen kommunikointi toteutettiin viestipuskureilla. Viestien välitys tehtiin käyttämällä staattista viestimuuttujaa, jonka seurauksena jokaisella säikeellä voi olla puskurissaan kuitenkin vain yksi viesti. Viestipuskurien tarkoituksena on mahdollistaa useiden viestien säilöminen puskurin sisällä, mutta tällaista tilannetta ei säikeistetyssä radiolaitteen ohjelmassa pääse syntymään, joten viestipuskurien käyttö nykyisessä toteutuksessa on turhaa.

Säikeiden välisessä viestinnässä tuli aluksi käyttää dynaamista muistinhallintaa. Tällöin viestin lähettäminen toiselle säikeelle alkaisi muistin varaamisella. Muistia varattaisiin lähetettävän viestin koon verran. Varattuun muistilohkoon kirjoitettaisiin viestin sisältö ja viestipuskurin kautta vastaanottavalle säikeelle lähetettäisiin muistilohkoon osoittava osoitin. Lukemisen jälkeen vastaanottava säie vapauttaisi viestin varaaman muistin, jottei muistivuotoa pääsisi syntymään. Tämän toteutuksen suurimpana etuna on skaalautuvuus. Nykyisessä toteutuksessa säikeet vastaavat ainoastaan saamiinsa viesteihin, pois lukien *ProductionTester*-säie. Arkkitehtuuri ei nykyisellään taivu tilanteeseen, jossa mikä tahansa säie voisi lähettää viestin mille tahansa säikeelle. Tämä tekee nykyisestä toteutuksesta vaikeasti skaalautuvan ja jäykän.

Dynaaminen muistinhallinta oli primäärinen tapa toteuttaa viestien välitys säikeiden välillä, mutta nykyinen toteutus oli helpompi ja nopeampi toteuttaa sekä todeta toimivaksi. Nykyisen toteutuksessa viesti tallennetaan staattiseen ja globaaliin muuttujaan, josta säie pystyy viestin lukemaan. Tämän toteutuksen jälkeen viestien välitystä ei enää muutettu. Viestipuskurien käyttäminen on jäänne aikomuksesta muuttaa nykyinen viestien välitys dynaamiseksi. Viestin vastaanottava säie lukee viestin staattisesta muuttujasta osoitteen avulla. Viestin välityksen muuttaminen dynaamiseksi vaatisi ainoastaan, että muistin dynaaminen varaaminen ja sen vapauttaminen toteutettaisiin ohjelmakoodiin. Aikataulullisista syistä tätä ei kuitenkaan ehditty toteuttamaan.

### 7.3 Verkkoprosessorin käyttö

Nykyisessä toteutuksessa *WlanSubTask*- ja *WlanTask*-säie käyttävät verkkoprosessoria. Joissain tilanteissa verkkoprosessori saatettaisiin haluta sammuttaa ja käynnistää uudelleen. Koska näillä kahdella säikeellä ei ole tietoa toistensa verkkoprosessorin käytöstä, voisi toinen säie sammuttaa verkkoprosessorin silloin, kun toinen käyttäisi sitä. Tällainen tilanne aiheuttaisi varmasti ohjelmassa virheen. Tilanne ei nykyisessä ohjelmassa ole kuitenkaan mahdollista, sillä *WlanTask*-säie välittää viestit *WlanSubTask*-säikeelle. Mikäli *WlanSubTask*-säikeen saama viesti vaatii verkkoprosessorin uudelleen käynnistämisen, käynnistää säie verkkoprosessorin, ennen kuin se lähettää vastausviestin. *WlanTask*-säie puolestaan odottaa *WlanSubTask*-säikeen vastausta kunnes saa sen. Tällöin verkkoprosessori on ehditty käynnistämään jo uudelleen, ennen kuin *WlanTask*-säie saa vastauksen. Jatkokehityksessä edellä mainitun tilanteen syntyminen saattaisi kuitenkin olla mahdollista nykyisellä arkkitehtuurilla.

Aikaisemmassa toteutuksessa *WlanTask*-säie ohjasi verkkoprosessorin toimintaa ja määräsi milloin datan jatkuva yhteydetön lähetys alkoi. Sen hetkisissä vaatimuksissa ei ollut selvyyttä, millaista toiminnallisuutta vaaditaan sertifiointitesteihin. Yhtenä vaatimuksena oli esimerkiksi mahdollisuus muuttaa MAC-osoitetta, vaikka verkkoprosessorilla lähetettäisiinkin jatkuvasti dataa. Koska *WlanTask*-säie vastasi datan jatkuvasta lähettämisestä, ei se pystynyt vaihtamaan käytettävää MAC-osoitetta. Ratkaisuna *WlanTask*-hajoitettiin kahteen osaan, joista toisella osalla pystyttiin vaihtamaan MAC-osoite, silloin kun toinen osa lähettää dataa verkkoprosessorilla. Tämä vaatimus on myöhemmin poistunut tarpeettomana ja nykyisessä toteutuksessa *WlanTask*- ja *WlanSubTask*-säikeiden toiminnallisuus voitaisiin yhdistää yhden säikeen alle. Tämä poistaisi myös edellisessä kappaleessa kerrotun ongelman, sillä muutoksen jälkeen ainoastaan yksi säie käyttäisi verkkoprosessoria nykyisen kahden sijasta.

## 8. YHTEENVETO

Työn tarkoituksena oli selvittää, mitä toiminnallisuuksia radiolaitteelle tulee ohjelmoida, jotta laitteelle voitaisiin suorittaa sertifiointitestausta CE- ja FCC-sertifiointeja varten. Työssä myös kerrottiin esimerkein, miksi sertifiointi on tärkeää sekä mitä CE- ja FCC-sertifiointien saaminen tarkoittaa ja mahdollistaa. Tämän lisäksi työssä ohjelmoitiin radiolaitteelle testejä varten haluttu toiminnallisuus sekä luotiin erillinen tietokoneohjelma, jonka avulla kolmas osapuoli pystyi testaamaan radiolaitteen. Lisäksi työssä selvitettiin lukijalle millaisia testejä radiolaitteelle tehtäisiin sertifiointitestien aikana. Työn toteutettavassa osassa kuvattiin luotuja ohjelmia, näytettiin näiden ohjelmien lopullinen arkkitehtuuri komponenttikaavion avulla sekä näytettiin ohjelmakoodiesimerkein ohjelmoitua toiminnallisuutta. Tämän lisäksi työssä käytiin läpi mahdolliset kehitysehdotukset ja mietittiin käytettyjen ratkaisujen hyödyllisyyttä. Työssä tehdyt ratkaisut olivat toimivia, mutta työn jatkokehityksen kannalta osa valituista ratkaisuista lukitsee arkkitehtuurin tietynlaiseksi. Tämä vaikeuttaa työn laajentamista ja uusien ominaisuuksien lisäämistä työhön.

Tärkeimpänä asiana työssä oli luoda sertifiointitestauksen mahdollistavat ohjelmat. Työssä luotiin kaksi erillistä ohjelmaa, joiden avulla radiolaitteelle voitiin suorittaa sertifiointitestaukset. Ensimmäinen ohjelma luotiin radiolaitteelle ja tällä ohjelmalla radiolaitteella pystyttiin suorittamaan siltä vaaditut toiminnot. Ohjelmalla pystyttiin luomaan radiolaitteesta tukiasema, yhdistämään laite olemassa olevaan tukiasemaan, lähettämään ja vastaanottamaan dataa, kirjoittamaan ja lukemaan dataa laitteen flash-muistilta sekä kommunikoimaan sarjaliikenneväylän ylitse. Tämän lisäksi radiolaitteella pystyttiin suorittamaan kahden edellä mainitun toiminnallisuuden kombinaatio ja kuormittamaan laitetta. Tietokoneelle luodulla ohjelmalla sertifiointitestauksen suorittava testaaja pystyi kommunikoimaan radiolaitteen kanssa sarjaliikenneväylää pitkin sekä täydentämään radiolaitteen toiminnallisuuksia. Esimerkiksi yhteyden tarkkailua kahden radiolaitteen välillä ei olisi pystytty suorittamaan ilman tietokoneohjelmaa, vaikka radiolaitteen ohjelma tukikin kyseisen ominaisuuden testaamista.

Luoduilla ohjelmilla pystyttiin tekemään sertifiointitestausta alusta loppuun. Tämä tosin vaati, että radiolaitteella olevaa ohjelmaa muokattiin jonkin verran, sillä kahta vaatimusta ei saatu täytettyä alkuperäisellä ohjelmalla. Korjatulla ohjelmalla testaus saatiin suoritettua loppuun, ja testauksen tuloksista voitiin päätellä, että radiolaitteelle täytetään sille asetetut vaatimukset. Tämä tarkoitti, että radiolaitteelle voitiin hakea CE- ja FCC-sertifikaatteja, ja se voidaan asettaa myyntiin Euroopan unionin talousalueella sekä Yhdysvalloissa.

## LÄHTEET

- [1] CENELEC, Frequently asked questions, verkkosivu, saatavissa (viitattu 12.10.2017), [https://www.cenelec.eu/faq/faq\\_entry.htm](https://www.cenelec.eu/faq/faq_entry.htm)
- [2] Texas Instruments, CC3200, SimpleLink™ Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU, verkkosivu, saatavissa (viitattu 26.10.2017), <http://www.ti.com/product/CC3200>
- [3] Federal Community Commission, Interference free telephones, verkkosivu, saatavissa (viitattu 15.10.2017), [https://transition.fcc.gov/Bureaus/Common\\_Carrier/News\\_Releases/1994/nrcc4019.txt](https://transition.fcc.gov/Bureaus/Common_Carrier/News_Releases/1994/nrcc4019.txt)
- [4] The Sydney Morning Herald, Planning debacle forces radio towers to seek new home, verkkosivu, saatavissa (viitattu 15.10.2017), <http://www.smh.com.au/articles/2003/02/16/1045330466812.html>
- [5] European Commission, Radio Equipment Directive (RED), verkkosivu, saatavissa (viitattu 19.10.2017), [http://ec.europa.eu/growth/sectors/electrical-engineering/red-directive\\_en](http://ec.europa.eu/growth/sectors/electrical-engineering/red-directive_en)
- [6] European Commission, Guide to the Radio Equipment Directive 2014/53/EU, verkkosivu, saatavissa (viitattu 19.10.2017), <http://ec.europa.eu/docsroom/documents/23321>
- [7] EUR-Lex, Directive 2014/53/EU of the European Parliament and of the Council, verkkosivu, saatavissa (viitattu 19.10.2017), <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014L0053>
- [8] EUR-Lex, Directive 2002/20/EC of the European Parliament and of the Council, verkkosivu, saatavissa (viitattu 19.10.2017), [http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L\\_.2002.108.01.0021.01.ENG&toc=OJ:L:2002:108:TOC](http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2002.108.01.0021.01.ENG&toc=OJ:L:2002:108:TOC)
- [9] European Commission, Electromagnetic Compatibility (EMC) Directive, saatavissa (viitattu 19.10.2017), [http://ec.europa.eu/growth/sectors/electrical-engineering/emc-directive\\_en](http://ec.europa.eu/growth/sectors/electrical-engineering/emc-directive_en)
- [10] EUR-Lex, Directive 2014/30/EU of the European Parliament and of the Council, verkkosivu, saatavissa (viitattu 19.10.2017), [http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L\\_.2014.096.01.0079.01.ENG&toc=OJ:L:2014:096:TOC](http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.096.01.0079.01.ENG&toc=OJ:L:2014:096:TOC)

- [11] European Commission, CE marking, verkkosivu, saatavissa (viitattu 20.10.2017), <http://ec.europa.eu/growth/single-market/ce-marking/>
- [12] European Commission, Manufacturers, verkkosivu, saatavissa (viitattu 20.10.2017), [https://ec.europa.eu/growth/single-market/ce-marking/manufacturers\\_en](https://ec.europa.eu/growth/single-market/ce-marking/manufacturers_en)
- [13] European Commission, EU Consumers, verkkosivu, saatavissa (viitattu 20.10.2017), [https://ec.europa.eu/growth/single-market/ce-marking/consumers\\_en](https://ec.europa.eu/growth/single-market/ce-marking/consumers_en)
- [14] Federal Communication Commission, About the FCC, verkkosivu, saatavissa (viitattu 20.10.2017), <https://www.fcc.gov/consumers/guides/about-fcc>
- [15] Federal Communication Commission, Introduction to FCC Rules and Introduction to FCC Rules and Equipment Authorization Program, verkkosivu, saatavissa (viitattu 20.10.2017), [https://transition.fcc.gov/oet/ea/presentations/files/oct05/Intro\\_to\\_FCC\\_Policy\\_GT.pdf](https://transition.fcc.gov/oet/ea/presentations/files/oct05/Intro_to_FCC_Policy_GT.pdf)
- [16] Federal Communication Commission, Equipment Authorization – Importation, verkkosivu, saatavissa (viitattu 20.10.2017), <https://www.fcc.gov/oet/ea/importation#sec1>
- [17] Federal Communication Commission, Equipment Authorization Procedures, verkkosivu, verkkosivu, saatavissa (viitattu 20.10.2017), <https://www.fcc.gov/general/equipment-authorization-procedures#block-menu-block-4>
- [18] EMC FastPass, Getting EMC Design Right The First Time, verkkosivu, saatavissa (viitattu 26.10.2017), <https://www.emcfastpass.com/emc-testing-beginners-guide/emissions/>
- [19] Incompliance, Guide to Testing Conducted Emissions (Based on the Methods in EN 55022 and EN 55011) verkkosivu, saatavissa (viitattu 27.10.2017), <https://incompliancemag.com/article/guide-to-testing-conducted-emissions-based-on-the-methods-in-en-55022-and-en-55011/>
- [20] D.G. Rahej, L.J. Gullo Design for Reliability, Wiley, Aug 2012, s. 169–180
- [21] I. Graesslin, H. Homann, S. Biederer, P. Börnert, K. Nehrke, P. Vernickel, G. Mens, P. Harvey, U. Katscher, A specific absorption rate prediction concept for parallel transmission MR, Wiley Subscription Services, Inc., A Wiley Company, Vol. 68, no. 5 2012, s. 1664–1674, <http://onlinelibrary.wiley.com.libproxy.tut.fi/doi/10.1002/mrm.24138/>
- [22] R.S. Kshetrimayum, Mobile Phones: Bad for Your Health?, IEEE, Vol. 27 no. 2 March 2008, <http://ieeexplore.ieee.org.libproxy.tut.fi/document/4475802/>

- [23] IEC, Measurement procedure for the assessment of specific absorption rate of human exposure to radio frequency fields from hand-held and body-mounted wireless communication devices - Part 1: Devices used next to the ear (Frequency range of 300 MHz to 6 GHz), IEC, Jul 2016.
- [24] BSI, Product standard to demonstrate the compliance of wireless communication devices with the basic restrictions and exposure limit values related to human exposure to electromagnetic fields in the frequency range from 30 MHz to 6 GHz: hand-held and body mount, BSI, Oct. 2017.
- [25] IEEE, Human exposure to radio frequency fields from hand-held and body-mounted wireless communication devices - Human models, instrumentation, and procedures - Part 2: Procedure to determine the specific absorption rate (SAR) for wireless communication devices used in close proximity to the human body (frequency range of 30 MHz to 6 GHz), IEEE, March 2010.
- [26] International Commission on Non-ionizing Radiation Protection, Guidelines for limiting exposure to time-varying electric, magnetic and electromagnetic fields (up to 300 GHz), ICNIRP Publication, 1998, s. 1.
- [27] The Code of Federal Regulations, Radiofrequency radiation exposure evaluation: portable devices, Verkkosivu, saatavissa (viitattu 15.1.2018), [https://www.ecfr.gov/cgi-bin/text-idx?SID=a093d967a44e66c0fbaf46c17587fe54&mc=true&node=se47.1.2\\_11093&rgn=div8](https://www.ecfr.gov/cgi-bin/text-idx?SID=a093d967a44e66c0fbaf46c17587fe54&mc=true&node=se47.1.2_11093&rgn=div8)
- [28] IEEE, IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz, IEEE, April 1999.
- [29] IEEE, Recommended Practice for Determining the Peak Spatial-Average Specific Absorption Rate (SAR) in the Human Head from Wireless Communications Devices: Measurement Techniques, IEEE, Sept. 2013.
- [30] M. Hakala, M. Vainio, Tietoverkon rakentaminen, Docendo, 2005, s. 144–145.
- [31] Python Software Foundation, Python, verkkosivu, saatavissa (viitattu 7.5.2018), <https://www.python.org/>
- [32] The Qt Company, PySide, verkkosivu, saatavissa (viitattu 7.5.2018), <https://wiki.qt.io/PySide>