

Mikael Oja

# Radiovirittimen käyttöönotto ja ohjelmistoradiodemon ohjelmointi

Insinööri (AMK),

tietotekniikka

Kevät 2017



KAJAANIN  
AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

## Tiivistelmä

**Tekijä(t):** Oja Mikael

**Työn nimi:** Radiovirittimen käyttöönotto ja ohjelmistoradiodemon ohjelmointi

**Tutkintonimike:** Insinööri (AMK), tietotekniikka

**Asiasanat:** ohjelmistoradio, sulautettu järjestelmä

Opinnäytetyön tavoitteena oli ottaa käyttöön STMicroelectronicsin kehittämä radiovirittinlaajennuskortti STMicroelectronicsin kehittämälle Accordo5 -kehitysalustalle sekä ohjelmoida laitteistoa hyödyntävä ohjelmistoradio valmiiden laiteohjainkirjastojen avulla. Tavoitteena oli myös tutustua radioteknologiaan ja ohjelmistokehitykseen sulautetuille järjestelmille.

Työtä suorittaessa hyödynnettiin useita eri työkaluja ja tekniikoita, esimerkiksi virtualisointia väliaikaisen verkkopalvelimen pystyttämiseen MediaWiki-palvelua varten ja Yocto-projektia kustomoidun Linux käyttöjärjestelmän kääntämiseen.

Työssä toteutettu ohjelmistoradio on toimeksiannon määritelmien mukainen ja täyttää asetetut vaatimukset. Ominaisuuksiltaan se on toimiva FM-radio, jolla voi virittäytyä halutulle taajuudelle, siirtyä seuraavalle tai edelliselle kanavalle, listata kaikkien vastaanotettavien kanavien taajuudet, ja näyttää radiolähetysten mukana tulevaa tekstidataa.

## **Abstract**

**Author(s):** Oja Mikael

**Title of the Publication:** Radio tuner integration and programming a software defined radio demo

**Degree Title:** Bachelor of Engineering, Information Technology

**Keywords:** software defined radio, embedded system

The aim of this Bachelors thesis was to integrate a radio tuner to an Accordo5 development platform, both developed by STMicroelectronics, and program a software defined radio using provided device control libraries. The aim was also to familiarize oneself with radio technology and development for embedded systems.

Many tools and techniques were utilized while working on the radio, such as virtualization to host a temporary webserver for device documentation implemented with the MediaWiki wiki engine and Yocto-project to compile a customized Linux operating system.

The software defined radio produced during the thesis matches the specifications of the assignment and fulfills the set requirements. It is an FM radio, that can be used to tune to a desired frequency, switch to next or previous channel, list all receivable radio channel frequencies and show text data transmitted by the radio channel.

## Sisällys

1	Johdanto.....	1
2	Radion tiivistetty historia.....	2
3	Radioteknologian teoriaa lyhyesti.....	4
3.1	Radion toiminta yksinkertaistettuna.....	4
3.2	Signaalimodulointi.....	4
3.2.1	Amplitudimodulointi.....	5
3.2.2	Taajuusmodulointi.....	6
3.2.3	Vaiheavainnus.....	6
3.3	Ohjelmistoradio.....	6
4	Työssä käytettyjä työkaluja.....	7
4.1	Yocto-projekti.....	7
4.2	MediaWiki.....	8
4.3	Virtualisointi.....	8
5	Toteutusalueista.....	10
5.1	Laitteisto.....	10
5.2	Ohjelmisto.....	12
6	Suoritus.....	13
6.1	Tavoite.....	13
6.2	Suunnitelma.....	13
6.3	Dokumentaation pystyttäminen.....	13
6.4	Levykuvakkeen kääntäminen Yocto-projektilla.....	15
6.5	Ohjelman toteutus.....	16
6.6	Yhteistyö ST:n kanssa.....	18
6.7	Lopputulokset.....	18
7	Yhteenveto.....	19
	Lähteet.....	20

## Symboliluettelo

AM	Amplitude Modulation
BSP	Board Support Package
DAB	Digital Audio Broadcast
FM	Frequency Modulation
IF	Intermediate Frequency
LO	Local Oscillator
PSK	Phase Shift Keying
RF	Radio Frequency
ST	STMicroelectronics

## 1 Johdanto

Työn toimeksiantaja, Symbio Finland Oy, on suomalainen ohjelmistoyritys, jonka toiminta keskittyy paljon sulautettujen järjestelmien ympärille. Symbio on jo pitkään tarjonnut palveluitaan automarkkinoilla ja kehittänyt autojen keskusjärjestelmäsovelluksia. Autojen kehitys liikkuu koko ajan enemmän ohjelmistopainotteiseen suuntaan. Keskusjärjestelmät eivät enää koostu pelkästä radiosta, vaan ovat yhä useammin pieniä tietokoneita, jotka tarjoavat radion lisäksi useita palveluita, kuten navigaattori, peruutustutka ja muiden auton sensorien datan välittäminen kuljettajalle. Vaikka palvelut ovatkin lisääntyneet niin radio on vielä olennainen osa keskusjärjestelmän ominaisuuksista.

Radioteknologia on kehittynyt ensimmäisistä radioista niin paljon, että radion kuunteluun tarvittava elektroniikka mahtuu nykyään pienelle mikropiirille, radioviritinkortille. Koska radioviritinkortti voi tukea useita eri lähetysteknologioita, se voi olla ominaisuuksiltaan hyvin monipuolinen. Sen ohjainohjelmisto sisältää paljon toimintoja, jotka ovat loppukäyttäjän kannalta epäolennaisia, mutta silti radion toiminnan kannalta tarpeellisia.

Työssä keskitytään ohjelmistoradion käyttörajapinnan ja sitä hyödyntävän demon luomiseen. Sovellus kehitetään STMicroelectronicsin (ST) kehittämälle keskusjärjestelmäalustalle. Alustan lisävarusteena saatava ST:n kehittämä radioviritinkortti sisältää integroidut piirit erilaisten radiosignaalien käsittelyyn. Työssä perehdytään uuden laitteen käyttöönottoon laitteen mukana tulevien ohjainkirjastojen avulla. Lisäksi työssä tutustutaan radioteknologiaan ja uuden tuotteen suunnitteluun ja prototyypin tekemiseen.

## 2 Radion tiivistetty historia

Radioaallot löydettiin vuonna 1887, kun saksalainen fyysikko Heinrich Hertz suoritti sarjan kokeita todistaakseen Maxwellin yhtälöiden ennustamien sähkömagneettisten aaltojen olemassaolon. Hertzin koelaitte oli varsin alkeellinen, eikä hän onnistunut vastaanottamaan aaltoja 20 metriä kauempaa. Hertz ei ymmärtänyt löytönsä merkitystä ja totesi aaltojen olevan käytännön kannalta turhia. [1.]

Radioaaltojen soveltamisen kommunikointiin keksi italialainen Guglielmo Marconi vuonna 1895. Marconi jäljitteli, yhdisteli ja paranteli muiden keksijöiden tekemiä löydöksiä ja useiden kokeiden jälkeen hän onnistui vastaanottamaan radioaaltoja yli kolmen kilometrin päästä. Hakiessaan rahoitusta Italian viestintäministeriöltä häntä kutsuttiin hulluksi ja Marconi muuttikin pian laitteistonsa kanssa Englantiin. Siellä hänen keksintönsä vakuutti useita viestintäalan henkilöitä ja hän sai tukea jatkaakseen kokeitaan. Marconi tunnetaan kehittämästään Marconin laista, jonka mukaan signaalin kantomatka on verrannollinen antennin korkeuden neliöön, sekä radiosähkötyksen keksijänä. [2.] Sähkötyös oli radioaaltojen ensimmäinen kaupallinen sovellus.

Kanadalainen tutkija Reginald Fessenden ymmärsi aiempien lankapuhelukokeidensa perusteella, että äänen lähettämiseen tarvittaisiin jatkuva-aalto. Vuonna 1900 hän teki ensimmäisen AM-lähetyksen, jonka kantoaallon lähteenä toimi erikoistarkoitukseen suunniteltu 10 kHz kipinäinduktori. Lähetys oli kuitenkin erittäin huonolaatuinen ja lähetetty ääni peittyi lähes kokonaan kipinän aiheuttamaan kohinaan. Fessenden tunnetaan kehittämästään teknologiasta, jolla pystyisi lähettämään jatkuvaa sinimuotoista aaltoa. Hän kehitti myös teorian sekoittamisesta (englanniksi heterodyning), jota kymmenen vuotta myöhemmin elektroniputkien kehittyessä alettiin käyttää radiosignaalien erottelussa. Fessendenin keksinnöt toimivat AM-lähetysten perustana. [3.]

Vaikka elektroniputket keksittiin jo 1800-luvun puolella, erilaiset elektroniputkia hyödyntävät 1900-luvulla tehdyt keksinnöt paransivat radioaaltojen lähetystä ja vastaanottoa:

- Diodi: John Ambrose Flemingin vuonna 1904 keksimän tasasuuntaajan avulla vastaanotettu vaihtovirtainen radiosignaali pystyttiin muuttamaan yksisuuntaiseksi aiempia kidekonevastaanottimia tehokkaammin [4][5].
- Audion: Lee De Forestin 1906 keksimän alkeellisen triodin signaalia vahvistavat ominaisuudet löydettiin vasta 1912. Audionin avulla heikko radiosignaali pystyttiin

vahvistamaan, parantaen signaalin kantavuutta. De Forestin Audionin alkuperäinen malli poistui kuitenkin nopeasti käytöstä, kun siitä paranneltu versio kehitettiin. Paranneltua mallia kutsutaan yksinkertaisesti triodiksi. [4.] [6. s. 9]

- Oskillaattori: Edwin Armstrong keksi vuonna 1912 varhaisen oskillaattorin, joka oli halpa ja pienikokoinen tapa muodostaa jatkuvaa aaltoa [7].

Näitä keksintöjä yhdistelemällä ja kehittämällä AM-radiolähetykset ja näin myös äänen lähetys mahdollistuivat. Säännölliset viihdetarkoitukselliset radiolähetykset alkoivat 1920-luvulla ja radiosta tuli pian kodin pääviihdyttäjä. [8.]

Vuonna 1947 keksitty transistori mullisti elektroniikan kehityksen. Suurehkot, kömpelöt ja paljon energiaa käyttävät elektroniputket korvattiin pienillä ja energiatehokkailla transistoreilla. Enää ei tarvittu useita suuria paristoja ja radion koko pieneni huomattavasti. [9.]

1970-luvun alussa yleistyneet mikropiirit pienensivät elektroniikkaa vielä entisestään ja aloittivat niin sanotun informaation aikakauden [10]. Nykyään radion kuuntelemiseen ei tarvitse kuin antennin, analogidigitaalimuuntajan ja pienen tietokoneen, kuten puhelimen, jolla digitaalinen radiosignaali käsitellään ohjelmistotasolla.



### 3 Radioteknologian teoriaa lyhyesti

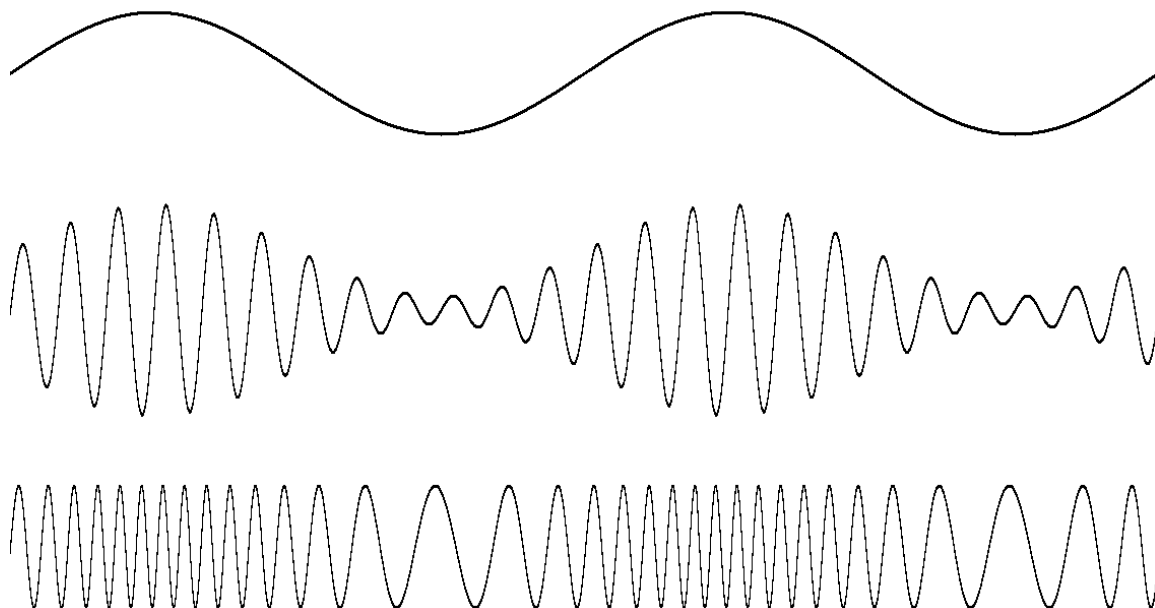
#### 3.1 Radion toiminta yksinkertaistettuna

Kun radioaalto kohtaa antennin, radioaallon kantama sähkökenttä aiheuttaa antennin joh-  
timessa pienen sähkövirran. Sähkövirran suunta ja voimakkuus vaihtelevat radioaallon  
vaiheen mukaan, ja laitetasolla tätä vaihtelun taajuutta kutsutaan radiotaajuudeksi (Radio  
Frequency, RF). Tässä vaiheessa RF-signaali sisältää kaikki antennin vastaanottamat  
taajuudet. Oskillaattorilla (Local Oscillator, LO) valitaan radion läpipäästötaajuus ja muut  
taajuudet suodatetaan pois RF-signaalista. Lisäksi LO:lla valittu taajuus demoduloidaan  
RF-signaalista, eli signaalista poistetaan kantaalto ja saadaan tuloksena välitaajuinen  
(Intermediate Frequency, IF) signaali.

IF-signaalista yleensä suodatetaan pois erilaisia häiriöitä ja heijastumia signaalin paran-  
tamiseksi. Tämä suodatusprosessi voi sisältää useita eri vahvistimia ja demoduloijia. Lo-  
puksi IF-signaalista kuitenkin on jäljellä alkuperäinen lähetetty datasiignaali, joka voi olla  
analogista tai digitaalista bittivirtaa. Tämä data prosessoidaan sovelluksesta riippuen esi-  
merkiksi stereovahvistimen läpi kaiuttimille. [8.]

#### 3.2 Signaalimodulointi

Signaalin modulointi tarkoittaa kahden signaalin yhdistämistä. Toinen signaali on alkupe-  
räisen datan sisältävä signaali, ja toinen on korkeataajuinen kantaalto. Kantaalto muo-  
kataan sisältämään datasiignaali siten, että kantaallon korkea taajuus säilyy. Kuvassa 1  
on esimerkkejä analogisen signaalin moduloinnista. Eritaajuisia kantaaltoja käyttämällä  
samankaltaista dataa voidaan lähettää samalla tekniikalla ilman signaalien sekoittumista.



Kuva 1. Analogisen signaalin modulointi. Ylimpänä alkuperäinen datasignaali, keskimmäisenä amplitudimoduloitu signaali ja alimpana taajuusmoduloitu signaali.

### 3.2.1 Amplitudimodulointi

Amplitudimoduloinnissa (Amplitude Modulation, AM) lähetettävä data moduloidaan kanta-aallon amplitudiin eli aallon "korkeuteen" tai "voimakkuuteen". Kanta-aaltona käytetään pitkiä ja keskipitkiä aaltoja. AM-signaalin demodulointi on yksinkertaista, ja tästä syystä vastaanottimet ovat halpoja valmistaa. Ensimmäiset vastaanottimet olivatkin niin sanottuja kidekone radioita, jotka eivät vaadi erillistä virtalähdettä, vaan toimivat radioaallon sähkökentän tuottamalla sähkövirralla.

AM-lähetyksillä on kuitenkin suuria heikkouksia. Koska alkuperäinen signaali on moduloitu kanta-aallon amplitudiin, kohina ja muut sähkömagneettiset häiriöt vaikuttavat suoraan äänenlaatuun ja signaalin heiketessä häiriöiden vaikutus kasvaa huomattavasti. Tästä syystä lähetyksissä pyritään käyttämään mahdollisimman voimakkaita aaltoja. Amplitudin kasvattaminen vaikuttaa kuitenkin suoraan virrankulutukseen ja laskee tehosuhtetta. Ongelmista huolimatta AM-lähetyksiä käytetään vieläkin esimerkiksi harrastelijakäytössä. [11.]

### 3.2.2 Taajuusmodulointi

Taajuusmoduloinnissa (Frequency Modulation, FM) alkuperäinen signaali moduloidaan kanta-aallon taajuuteen. Tästä syystä FM käyttää suurempaa taajuusalueita yhdelle kanavalle kuin AM. Alun perin ajateltiin, että kohina kasvaa suurilla taajuusalueilla ja FM-tekniikan kehityksen jatkaminen hylättiin. Tällöin ei kuitenkaan vielä ymmärretty FM-signaalin olevan luonnostaan vastustuskykyinen kohinaa vastaan. Mitä laajempaa taajuusalueita FM-signaalille käytetään, sitä vähemmän kohina häiritsee signaalin demodulointia. Kohina vaikuttaa ainoastaan signaalin amplitudiin, jolloin se voidaan suodattaa pois, kunhan signaali on tarpeeksi voimakas. Samasta syystä signaalin voimakkuuden vaihtelut eivät vaikuta lähetyksen laatuun, tiettyyn raja-arvoon asti. Näiden etujen ansiosta FM on osittain korvannut AM-lähetykset ja on erittäin laajalti käytetty yleisradiolähetyksissä sekä radiokommunikoinnissa. [12.] [13.]

### 3.2.3 Vaiheavainnus

Vaihemodulointia (Phase Shift Keying, PSK) käytetään digitaalisen signaalin modulointiin. Kanta-aallon vaihetta siirretään datasympolin eli bittiarvojen mukaan. PSK-tekniikoita on useita erilaisia, käyttökohteen tarkoituksen ja tarpeiden mukaan. Yleisiä muuttujia ovat datasiirron nopeus, signaalin kantavuus ja virheellisten symbolien määrä suhteessa vastaanotetun datan määrään. [14.]

## 3.3 Ohjelmistoradio

Ohjelmistoradio on radiosovellus, joka ei teoriassa tarvitse muuta radiolaitteistoa kuin antennin ja muuntajan. Analoginen RF-signaali muutetaan digitaaliseksi IF-signaaliksi, joka käsitellään ohjelmistotasolla. Yleensä IF-signaalin käsittelyyn käytetään kuitenkin siihen erikoistunutta mikropiiriä. Koska signaalin demodulointi tehdään ohjelmistasolla, ohjelmistoradiolla voi teoriassa kuunnella kaikkia radiolähetyksiä, jotka antenni voi vastaanottaa. Kustannuskuluja saadaan vähennettyä, kun tarvitaan vähemmän tiettyyn teknologiaan erikoistuvia fyysisiä komponentteja ja ohjelmistoradiota on helpompi kopioida massatuotantoon. Lisäksi uusia ominaisuuksia voidaan lisätä ohjelmistopäivityksillä ja näin pidennetään radion elinikää. [15.]

## 4 Työssä käytettyjä työkaluja

### 4.1 Yocto-projekti

Vuonna 2010 perustettu Yocto-projekti on Linux Foundationin hallinnoima ja ylläpitämä avoimen lähdekoodin yhteistyöprojekti eri laite- ja ohjelmistoyritysten kanssa. Sen tavoitteena on kerätä resursseja ja työkaluja yhden projektin alle yksinkertaistamaan kustomoidun sulautetun Linux-käyttöjärjestelmän luontia. Yocton resurssien avulla on helppoa rakentaa oma levykuvake sekä luoda kuvakekohtainen ohjelmistokehityspaketti. [16.]

Avoimen lähdekoodin ansiosta kuka tahansa voi käyttää Yocton tarjoamia työkaluja ja osallistua projektin kehittämiseen. Yocton johto-organisaatio koostuu 21 jäsenyrityksestä [17], joiden lisäksi Yocton tukiorganisaatioon kuuluu 40 yritystä [18]. Organisaatioiden lisäksi yksittäiset henkilöt voivat liittyä osaksi Yocton yhteisöä ja olla mukana kehittämällä resursseja ja raportoimalla tai korjaamalla ohjelmointivirheitä [19].

Yoctoa apuna käyttäen laitekohtaisen levykuvan kehityksen aloittaminen on yksinkertaisempaa ja nopeampaa kuin alusta kehittäminen, kun suuri osa työstä on yhteisön ansiosta jo valmiiksi tehtynä. Käyttämällä Yoctoa levykuvakkeen kehityspohjana valmiita resursseja voi yhdistellä siten, että levykuvake sisältää ainoastaan tarpeellisia ominaisuuksia ja tarpeen tullen ominaisuuksia on helppo lisätä jälkikäteen. Omalle laitteelle voi luoda oman laitetikipaketin (Board Support Package, BSP), jota ylläpitämällä ja jakamalla voidaan mahdollistaa nopea ohjelmistokehittämisen aloittaminen laitteelle. [20.]

Yocto-projektin kääntöympäristö perustuu OpenEmbedded-projektin OE-ytimeen, joka tarjoaa suuren määrän metadataa ja työkaluja sulautettujen Linux-levykuvakkeiden luontiin. OpenEmbedded-projekti käyttää BitBake nimistä ohjelmaa suoritettavien tehtävien hallinnointiin. BitBaken ohjeistamiseen käytetään reseptejä. Reseptit ovat tiedostoja, jotka sisältävät ohjeita pakettien konfigurointiin ja rakentamiseen. Reseptin suoritus voi olla riippuvainen toisista resepteistä, tai jopa koostua kokonaan toisten reseptien suorittamisesta. [20.]

## 4.2 MediaWiki

MediaWiki on ilmainen avoimen lähdekoodin palvelinohjelma wikipalveluiden luontiin. Se on php-ohjelmointikielellä toteutettu verkkosivumoottori, jonka avulla wikin sivuja voi helposti luoda ja muokata. MediaWiki kehitettiin alun perin korvaamaan UseModWiki Wikipedian moottorina vuonna 2002. UseModWiki tallentaa sivut tekstitiedostoihin, ja Wikipedian kasvava käyttäjämäärä vaati tehokkaampaa ratkaisua. MediaWiki on suunniteltu toimimaan sivuilla, jotka sisältävät valtavia määriä dataa ja prosessoivat tuhansia palvelinpyyntöjä sekunnissa. Vaikka MediaWiki on jo pitkään toiminut erillisenä projektina Wikipediasta, se toimii edelleen MediaWikin kehitystarpeiden suurimpana kohteena. [21.]

MediaWiki tallentaa sivujen sisältämän informaation tietokantaan. Kaikilla wikin käyttäjillä on mahdollisuus osallistua muokkaamaan wikin sisältöä ja muokkaukset tallennetaan omana objektinaan, poistamatta vanhoja sivuja. Tämä mahdollistaa vanhojen sivujen helpon palauttamisen vandalismin varalta. [22.]

## 4.3 Virtualisointi

Virtualisoinnilla tarkoitetaan käytännössä uuden resurssin luomista, jakamalla tietokoneen fyysisiä resursseja. Tämä resurssi voi olla verkkokortti, kovalevy tai kokonaan toinen tietokone, eli virtuaalikone. Virtuaalikone ei omaa yhtään omaa laitteistoa, vaan kaikki sen osat ovat joko fyysisen isäntäkoneen osia, tai puhtaasti ohjelmapohjaisia oikean laitteen toimintaa emuloivia ohjelmia. [23.]

Virtualisoinnilla voidaan saavuttaa useita etuja. Virtuaalikone voidaan viedä (englanniksi export) virtualisointiympäristöstä, jolloin virtuaalikoneen asetukset ja ominaisuudet pakataan yhdeksi tiedostoksi. Tiedosto voidaan siirtää toiselle tietokoneelle ja tuoda (englanniksi import) se uuteen virtualisointiympäristöön, jolloin tiedosto puretaan ja luodaan uusi identtinen virtuaalikone. Tätä voidaan hyväksikäyttää esimerkiksi kehitysympäristöjen jakamisessa. Joskus kehitysympäristön pystyttäminen vaatii useita ohjelmistokirjastoja, ohjelmia ja asetusten muokkauksia. Virtualisoinnilla ympäristön pystyttäminen tarvitsee tehdä vain kerran, jonka jälkeen virtuaalikone voidaan jakaa kaikille kehitysympäristöä tarvitseville. Näin säästetään aikaa ja helpotetaan työskentelyn aloittamista. [24.]

Virtuaalikonetta voi käyttää myös testiympäristönä, eräänlaisena hiekkalaatikkona. Tästä hyödytään silloin, kun ei haluta paisuttaa ja sotkea fyysisen koneen järjestelmää erilaisilla

ohjelmistoilla ja asetuksilla, jotka hidastaisivat koneen toimintaa tai jopa estäisivät sen kokonaan. Virtuaalikone on helppo poistaa ja korvata. Virtuaalikoneen varmuuskopiointi tai uudelleen asentaminen on usein nopeampaa kuin fyysisen koneen. [24.]

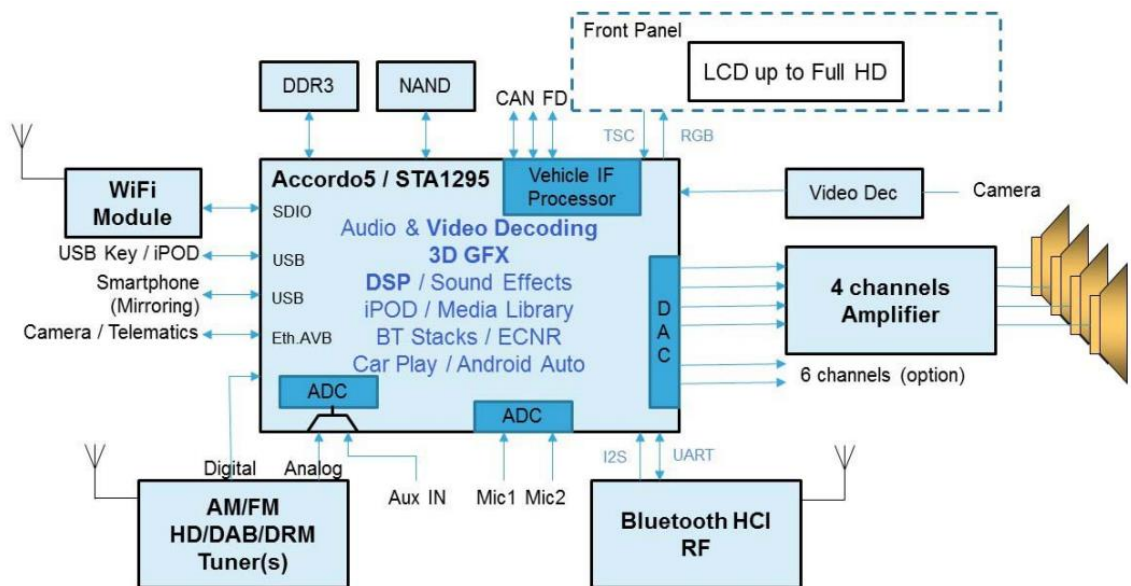
Virtualisoinnilla on kuitenkin huonotkin puolensa. Koska monia laitteistoja emuloidaan isäntäkoneen prosessorilla, virtuaalikoneella on usein hitaampaa työskennellä. Virtuaalikone ei pysty hyödyntämään kaikkia fyysisen koneen resursseja tehokkaasti, sillä emulointi ja isäntäjärjestelmän pyörittäminen varaavat oman osansa. Tästä syystä raskaat tehtävät, kuten suurien ohjelmistojen kääntäminen, ovat usein hitaampia suorittaa virtuaalikoneella. Virtualisointi ei myöskään sovellu graafisesti vaativiin tehtäviin, sillä näytönohjaimien virtualisointi tuo mukanaan omia ongelmiansa. [24.]

Useat näistä ongelmista voidaan korjata käyttämällä niin sanottua bare-metal hypervisor-virtualisointiympäristöä. Tällaisessa ympäristössä laitteiston ja virtualisointijärjestelmän välissä ei ole erillistä käyttöjärjestelmää vaan virtualisointiohjelmisto tavallaan toimii sen roolissa. Näin laitteiston resurssit voidaan jakaa virtuaalikoneiden käyttöön erittäin tehokkaasti. Tällaisen järjestelmän pystyttäminen ja ylläpito vaativat kuitenkin enemmän vaivaa, kuin käyttöjärjestelmän päällä toimiva virtualisointiympäristö. [24.]

## 5 Toteutusaluista

### 5.1 Laitteisto

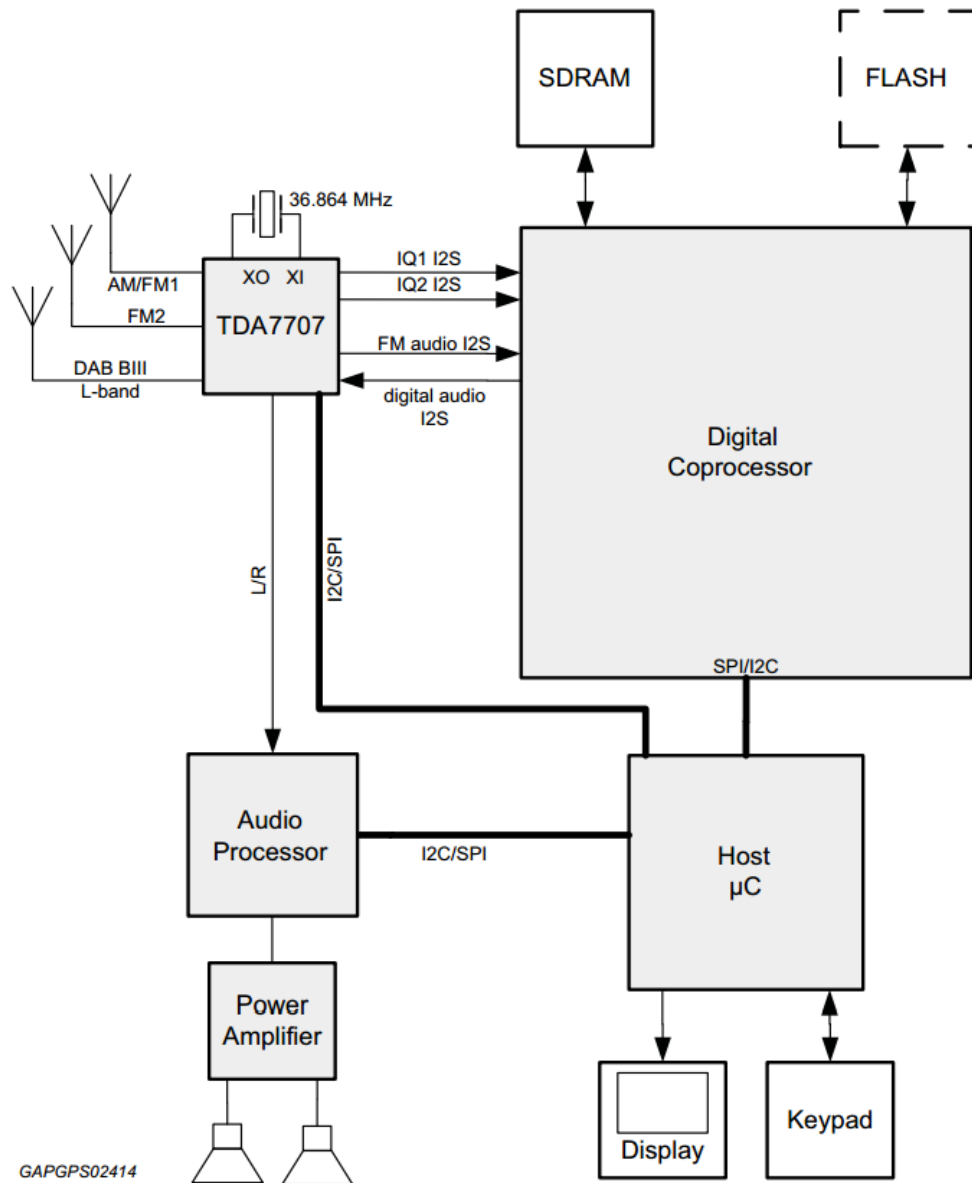
Opinnäytetyön toteutus tehdään ST:n STA1295 Accordo5-prosessoriin perustuvalla keskusjärjestelmälaitteella. Laitteen ominaisuuksia on kuvattu yleisellä tasolla kuvassa 2. Laitte on suunniteltu toimimaan auton multimediakeskuksena sekä elektronisten järjestelmien hallintalaitteena. STA1295 on tehokas ratkaisu näytöllisille äänijärjestelmille (englanniksi display audio systems).



Kuva 2. Yleiskuva STA1295 järjestelmästä. [25.]

Radioviritin ei ole osa päälaitetta vaan on erillinen ST:n kehittämä lisävaruste, joka liitetään päälaitteessa olevaan sitä varten suunniteltuun liittimeen. Radioviritinlaajennuskortti sisältää kaksi ST:n valmistamaa radiosignaalin käsittelyyn erikoistunutta TDA7707-mikropiiriä sekä digitaalisten lähetysten dekodaukseen erikoistuneen digitaalisen lisäprossessorin. Kohdemaasta riippuen lisäprossessori on joko DAB-lähetysten käsittelyyn erikoistunut STA660 tai HD-radion käsittelyyn erikoistunut STA680. Suomessa ei ole digitaalisia radiolähettyksiä, joten lisävarusteen valinnassa sillä ei varsinaisesti ole väliä. DAB-lähettykset kuitenkin yleistyvät ympäri maailmaa ja HD-radio on käytössä ainoastaan Yhdysvalloissa. Tulevaisuutta ajatellen lienee siis viisaampaa valita DAB-lähettyksiä tukeva lisävaruste.

Radioantennit liitetään laajennuskortissa oleviin SMB-liittimiin (SubMiniature version B). TDA7707-piirit muuntavat radiosignaalin digitaalseksi välisignaalksi eli bittivirraksi, joka käsitellään lähetyksestä riippuen STA660-piirillä (DAB-lähetykset) tai ohjataan suoraan päälaitteen käsiteltäväksi (FM- ja AM-lähetykset). Lopuksi äänidata ohjataan laitteen ääniohjaimelle, joka muuntaa sen digitaaliansalogimuuntimella takaisin analogiseksi signaaliksi, joka toistetaan äänijärjestelmällä. Laajennuskortin toimintaa on kuvattu yksinkertais-  
 tetulla yleistasolla kuvassa 3.



Kuva 3. Yleiskuva radioviritin laajennuskortin toiminnasta. [26.]



## 5.2 Ohjelmisto

Käyttöjärjestelmänä laitteessa toimii ST:n STA1295-laitetukipaketista Yocto-projektilla käännetty Linux. Radiovirittimen laiteohjaimena toimii ST:n kehittämä radiovirittimen abstrahointikerros, joka tarjoaa kehittäjälle koherentin ja vakaan ohjelmointirajapinnan. Se koostuu useista eri tasojen ohjelmistokirjastoista, kuten käyttöjärjestelmäkohtaisten kirjastojen abstrahointikirjastosta, radiovirittimeen ladattavista firmwareista, laiteohjainkommunikointiprotokollakerroksista ja muista vastaavista.

## 6 Suoritus

### 6.1 Tavoite

Työn tavoitteena on tutustua radioteknologiaan yleisellä tasolla ja ohjelmoidaan yksinkertainen radion käyttörajapinta ST:n ohjainkirjastojen avulla, sekä käyttörajapintaa hyödyntävä ohjelmistoradio FM-radion kuunteluun. Työn alustana toimii ST:n STA1295 Accordo5 prosessoriin perustuva auton keskusjärjestelmän kehityslaitte ja TDA7707-radioviritinmikropiiriä hyödyntävä laajennuskortti. Työn aikana tuotettu koodi luovutetaan insinööryön päätteeksi työn toimeksiantajalle ST:n tarjoamien resurssien prototyyppi implementaationa.

### 6.2 Suunnitelma

Työn suorittamiseen tarvitaan ST:n STA1295-kehityspaketti. Kehityspaketti sisältää STA1295 Accordo5-kehityslaitteen ja siihen kuuluvat johdot ja virtalähteen, radioviritinkortin, ST:n Yocto-projektilaitetukipaketin, ST:n Flash Loader -työkalun sekä laitteen ja työkalujen dokumentaatiot. Kehitysympäristönä käytetään 64bit Ubuntu 16.04-käyttöjärjestelmällistä tietokonetta.

Työssä ohjelmoitavan käyttörajapinnan mallina käytetään tavallista yksinkertaista autordiota. Toteutettavia ominaisuuksia ovat seuraavalle ja edelliselle kanavalle virittäminen, tietyille taajuuksille virittäminen, radiokanavan teksti-informaation näyttäminen, kaikkien saatavilla olevien kanavien haku ja tallentaminen listaan, josta ne voidaan hakea. Näiden lisäksi toteutetaan pakolliset initialisointi ja de-initialisointifunktiot. Testausta varten ohjelmoidaan käyttörajapintaa hyödyntävä komentoriviltä käytettävä sovellus.

### 6.3 Dokumentaation pystyttäminen

Laitteen käyttöohjeet ja dokumentointi on toteutettu MediaWiki ohjelmistoa käyttävänä wikisivustona. Koska dokumentaatiota tarvitsee toistaiseksi vain yksi henkilö, paikallinen hostaus on riittävä ja luultavasti yksinkertaisin ratkaisu. Dokumentaatio on kuitenkin toteutettu käyttämällä vanhaa MediaWiki-versiota, joka pohjautuu PHP:n vanhentuneeseen

5.x-versioon eikä toimi uudemman 7.x version kanssa. PHP 5.x ei ole saatavilla virallisista Ubuntu 16.x-repositorioista.

Yksinkertainen ratkaisu on pystyttää virtuaalikone, johon asennetaan LAMP-ohjelmistokokonaisuus (Linux, Apache, MySQL, PHP). Virtualisointia hyödyntämällä kehitysympäristöä ei sotketa vanhentuneilla ohjelmistoilla ja työn päätteeksi verkkopalvelimen poistaminen on helppoa. Virtualisointiohjelmia on useita, ja kaikki ovat epäilemättä näin yksinkertaiseen tarkoitukseen yhtä hyviä. Työssä valittiin VirtualBox, koska se oli valmiiksi asennettuna kehityskoneelle. Ohjeet VirtualBoxin asentamiseen ja käyttöön löytyvät osoitteesta [www.virtualbox.org/manual/UserManual.html](http://www.virtualbox.org/manual/UserManual.html). Virtuaalikoneelle asennetaan Linux-käyttöjärjestelmä ja tarpeelliset ohjelmistot MediaWiki palvelun hostaamiseen. Debian on ilmainen Linux-jakelu, ja ohjeet sen lataamiseen ja asentamiseen löytyvät Debianin virallisilta verkkosivuilta osoitteesta [www.debian.org](http://www.debian.org). Tietoa MediaWikin tarvitsemista ohjelmistoista löytyy osoitteesta [www.mediawiki.org/wiki/Manual:Installation\\_requirements](http://www.mediawiki.org/wiki/Manual:Installation_requirements). ST:n MediaWiki toteutus käyttää Apache2-verkkopalvelinta ja MySQL-tietokantaa.

MediaWiki dokumentaatio on pakattuna tar.gz-tiedostona, joka on helppo siirtää virtuaalikoneelle scp:llä tai VirtualBoxin shared folder-ominaisuudella (vaatii VirtualBoxin lisäominaisuus paketteja). Työssä päädyttiin ensimmäiseen vaihtoehtoon. Asennetaan virtuaalikoneelle openssh-server komennolla

```
sudo apt-get install openssh-server
```

VirtualBoxin virtuaalikoneen asetuksissa ohjataan jokin host-koneen portti, esimerkiksi 5522, virtuaalikoneen porttiin 22. Tämä voidaan tehdä host-koneen komentoriviltä komennolla

```
vboxmanage modifyvm vmname --natpf1 "ssh,tcp,,5522,,22",
```

jossa vmname on virtuaalikoneelle asetettu nimi. Nyt tiedostoja voidaan siirtää virtuaalikoneelle käyttämällä host-koneella komentoa

```
scp -P 5522 filename.tar.gz username@127.0.0.1:/home/username
```

Lisäksi virtuaalikoneen käyttöliittymää ei tarvitse enää pitää auki, vaan sitä voidaan hallinnoida ssh-yhteydellä komennolla

```
ssh -P 5522 username@127.0.0.1
```

Jottei dokumentaatiota tarvitsisi lukea virtuaalikoneen terminaalin kautta, ohjataan myös host-koneen portti 5580 virtuaalikoneen porttiin 80

```
vboxmanage modifyvm vmname --natpf1 "http,tcp,,5580,,80"
```

MediaWiki-paketti puretaan ajamalla virtuaalikoneessa komento

```
tar -xzf MediaWikiPackage.tar.gz
```

Puretuissa tiedostoissa on MySQL-tietokanta, joka sisältää wikin datan, MediaWikin php-moottorin sekä asennusskriptin. Muokataan asennusskriptiin oikea MySQL-tietokannan salasana sekä Apache2-palvelimen asetuksiin asetettu www-sivuhakemisto (oletusarvolta */var/www/html*) ja ajetaan skripti

```
./install_linux.sh
```

Skriptin valmistuessa wikisivu voidaan avata siirtymällä hostin selaimella osoitteeseen *localhost:5580/MediaWiki\_name*, jossa *MediaWiki\_name* on virtuaalikoneen www-sivuhakemistoon syntynyt kansio. Sivun auetessa MediaWikin initialisointi alkaa. Initialisoinnin aikana MediaWiki tarkistaa järjestelmän ohjelmistojen yhteensopivuusongelmien varalta sekä ottaa käyttöön oikean tietokannan. Initialisoinnin päätteeksi dokumentaation etusivu aukeaa ja wiki on valmis luettavaksi.

Dokumentaatiota tutkimalla kävi ilmi, ettei sen sisältö ole niin kattava kuin olisi toivottu. Sisältö on enimmäkseen laitteen levykuvakkeeseen sisällytettyjen ohjelmointirajapintojen ja tiettyjen laiteominaisuuksien kuvausta sekä levykuvakkeen yksinkertaistetut kääntöohjeet. Vaikka radiovirittimen ohjelmointirajapinnan dokumentaatio oli kattava, itse radiovirittintä käsittelevä dokumentaatio oli erittäin suppea, hieman sekava ja käytännössä rajoittui laitteen ja viritinkortin kytkentäohjeeseen. Myöhemmin radiovirittimen käyttöönotossa syntyi ongelmia, eikä dokumentaatiosta ollut paljoa apua niiden ratkaisemisessa.

#### 6.4 Levykuvakkeen kääntäminen Yocto-projektilla

Yocto-projekti konfiguroidaan ajamalla ympäristön pystytysskripti, jonka avulla Yocto asetetaan kääntämään oikeanlainen levykuvake. Skripti luo build-hakemiston, johon levykuvake käännetään. Levykuvakkeen kääntö käynnistetään komennolla

*bitbake imagename -k,*

jossa imagename on halutun levykuvakkeen nimi, esimerkiksi core-image-minimal. Koska kuvakkeen kääntäminen koostuu tuhansista eri vaiheista, kääntöprosessi kestää useita tunteja. Normaalisti BitBake keskeyttäisi kääntöprosessin yhdenkin virheen kohdallaan, mutta lisäämällä -k komennon perään BitBake yrittää suorittaa mahdollisimman monta vaihetta, virheistä huolimatta. Näin yksi virhe ensimmäisen kymmenen minuutin aikana ei keskeytä koko monen tunnin prosessia ja sen voi jättää etenemään vaikkapa yön ylitse. Prosessin valmistuessa käännettyt levykuvat ilmestyvät build-hakemiston polkuun `../builddirectory/tmp/deploy/images/targetname`, jossa targetname on aiemmin ympäristön pystytysskriptissä asetettu kohdelaitte.

Yocto-projektin avulla käännetään myös ohjelmistokehitystyökaluketju (englanniksi software development toolchain) koodin ristiin kääntämistä varten. Työkaluketjun kääntäminen onnistuu BitBaken komennolla

*bitbake -c populate\_sdk imagename*

Prosessin valmistuessa työkaluketjun asennusskripti ilmaantuu build-hakemiston polkuun `../[builddir]/tmp/deploy/sdk`. Sillä on automaagisesti generoitu nimi joka on mallia `_${DISTRO}_${TCLIBC}_${SDK_ARCH}_${IMAGE_BASENAME}_${TUNE_PKGARCH}-toolchain-<SDK_VERSION>.sh`, jossa jokainen `_${...}` vastaa jotain Yoctoon asetettua muuttujaa. Asennusskripti ajetaan ja ensimmäisenä skripti kysyy mihin työkaluketju halutaan purkaa. Työkaluketju koostuu referenssi sysroot -hakemistoista, kääntöohjelmista sekä kääntöympäristön pystytysskriptistä, joka asettaa tarpeellisia ristiin kääntömuuttujia kääntöprosessin suorittamista varten. Kun käännettävän ohjelman makefilet tai vastaavat käännön automatisointiavustajat ohjelmoidaan käyttämään näitä muuttujia, ristiin kääntö voidaan suorittaa normaalisti komentoriviltä GNU make -ohjelmaa kutsumalla. Kattavat ohjeet Yocto-projektin ominaisuuksista, toiminnasta ja käytöstä löytyvät osoitteesta [www.yoctoproject.org/docs/2.2.1/mega-manual/mega-manual.html](http://www.yoctoproject.org/docs/2.2.1/mega-manual/mega-manual.html).

## 6.5 Ohjelman toteutus

Ohjelman toteutus aloitettiin tutustumalla abstrahointikerroksen dokumentointiin sekä esimerkkiohjelmien lähdekoodiin. Saatua hyvä kuva ohjelmointirajapinnasta ja ohjelman elinkaaresta siirryttiin käyttörajapinnan suunnitteluun kysymällä "Mitkä ovat radion tärkeim-

mät toiminnallisuudet?" Radiota käytettäessä tärkeimmät ominaisuudet lienee seuraavalle kanavalle, edelliselle kanavalle, tai jollekin tietylle taajuudelle virittäytyminen. Kanavat lähettävät usein jotain tekstidataa äänen lisäksi, kuten kanavan nimi, radio-ohjelman nimi, tai mainoksia. Näillä ominaisuuksilla radion normaalikäyttö onnistunee.

Ohjelmointi suoritettiin C-kielellä. Käyttörajapinta toteutettiin dokumentaation ja suunnitelmien avulla funktionaalisuus kerrallaan. Kun käyttörajapinnan ominaisuudet olivat valmiit, ohjelmoitiin niitä hyödyntävä yksinkertainen demo-ohjelma. Koodin saavuttaessa teoreettisesti toimivan tilan ohjelman kääntö aloitettiin Yoctolla käännetyllä työkaluketjulla. Tässä vaiheessa kävi ilmi, ettei radiovirittimen abstrahointikerrosta ole käännettyyn mukaan referenssi sysroot -hakemistoihin. Lyhyellä tutkimustyöllä selvisi, että laiteohjainkirjastojen koodi käännetään ja linkataan osaksi ST:n radiodemo-ohjelmaa, eikä siitä tehdä erillistä kirjastoa itse levykuvakkeeseen. Abstrahointikerroksen BitBake-reseptiä tutkimalla kirjasto ja sen headerit kuitenkin löytyivät.

Demo-ohjelma käännettiin ja siirrettiin laitteelle testausta varten. Ohjelma ei kuitenkaan löytänyt radiovirittinlaitetta ollenkaan, vaikka kortti oli kiinnitettynä. Tällä varmistettiin, ettei sen hetkinen implementaatio toimi. Toimeksiantajan kanssa asiaa pohdittua sovittiin, että demoimplementaatio viedään niin pitkälle kuin opinnäytetyössä suunniteltiin ja otetaan yhteyttä laitevalmistajaan radiovirittimen käyttöönoton ongelmista.

Radion käyttörajapinta toteutettiin loppuun dokumentaation avulla. Suunnitelman mukaisesti ohjelmoitiin myös käyttörajapintaa hyödyntävä radiodemo, jonka käyttö onnistuu komentoriviltä. Koodi siistittiin ja kommentoitiin jatkokehitystä helpottavilla *TODO* -merkinnöillä, jotka useimmat C-koodieditorit tunnistavat ja korostavat. Radio-ohjelman kääntämistä helpottamaan tehtiin makefile-tiedosto, johon linkattavat kirjastot, kääntökomennot, kääntöobjektien poistaminen ja muut ohjelman kääntämiseen liittyvät toiminnot ovat valmiiksi tehtynä. Näin ohjelman kääntö onnistuu komentoriviltä make-ohjelmaa kutsumalla ja hakemiston siivoaminen lisäämällä kutsun perään *clean*

*make ja make clean*

Koodin kääntyvyys varmistettiin työkaluketjulla ja siitä korjattiin kääntäjän havaitsemat virheet.

## 6.6 Yhteistyö ST:n kanssa

Päädyimme toimeksiantajan kanssa poistamaan ylimääräiset välikädet laitevalmistajan ja ohjelmoijan välisestä kommunikaatiosta ja päädyin keskustelemaan laiteongelmista ST:n yhteyshenkilön kanssa. Virittimen tunnistusongelma johtui siitä, ettei tarpeellista kernel-moduulia aktivoitu oletuksella ja se täytyi aktivoida erikseen laitteen käynnistyksen yhteydessä Linuxin modprobe -ominaisuutta käyttämällä.

Uusia ongelmia syntyi, kun kehityslaite syystä tai toisesta lopetti toiminnan erään viikonlopun aikana, eikä enää käynnistynyt, eikä uuden levykuvakkeen flashays onnistunut. Tämä on ST:n yhteyshenkilön mukaan yleinen ongelma ja voi ilmaantua useastakin erisyystä, joita voi olla erittäin hankala paikantaa. Ongelman etsintään ja selvitykseen meni reilu viikko, jonka aikana kokeiltiin flashaysympäristön toimintaa ja kehityslaitteen eri muistien flashaysta. Useiden testien jälkeen ongelma eteni sen verran, että U-Boot saatiin käynnistymään suhteellisen luotettavasti. U-Bootin avulla saatiin selville, että laitteen muisti on todennäköisesti korruptoitunut tai muutoin hajonnut kriittisiltä alueilta, eikä laitteen korjaaminen sähköpostin välityksellä onnistuisi. Laite paketoitiin ja lähetettiin ST:n toimistolle Ranskaan tutkittavaksi ja korjattavaksi tai vaihdettavaksi. Laitteen saavuttua Ranskaan ongelman lähteeksi varmistui kärähtänyt muistipiiri.

## 6.7 Lopputulos

Symbiolla oli Kuopiossa toinen uudempaa revisiota oleva kehityslaite, joka oli toistaiseksi hyllyllä toimeettomana. Laite lähetettiin Ouluun, jotta opinnäytetyötä voitaisiin jatkaa. Laitteen saavuttua toimistolle ohjelmaa testattaessa ilmeni kuitenkin lisäongelmia. Ohjainkirjaston ei onnistunut kommunikoida radiovirittimen kanssa. Tämä johtui radioviritinkortilla olevan jumpperin väärästä asennosta. Jumpperin asento oli vaihdettu vastaamaan viritinkortin dokumentaatioissa olevaa kuvaa. Siirtämällä jumpperi alkuperäiseen asentoonsa demo-ohjelma alkoi toimia.

## 7 Yhteenveto

Opinnäytetyön suoritus kuvasi hyvin insinöörin työtehtäviä. Otetaan käyttöön vieras laitteisto ja ohjelmisto. Tutustutaan niiden ominaisuuksiin ja demonstroidaan niitä. Joskus kuitenkin tulee ylitsepääsemättömiä ongelmia laitteiston kanssa. Tällöin päästään tekemään yhteistyötä laitevalmistajan kanssa ja etsimään ongelman lähdettä. Jos ongelma löytyy, niin voidaan jatkaa töitä ja varmistaa tehdyn työn tulokset. Aina näin ei kuitenkaan käy ja ongelmaa ei voida korjata, tai ei löydetä ollenkaan.

Työn teoriapohja on melko laaja. Työn aikana tutustuttiin monipuolisesti eri työkaluihin ja teknologioihin. Opinnäytetyössä käytetyn ST:n laitteen ja radiovirittimen laitetason toiminnan ymmärtäminen ei kuitenkaan ole tarpeellista toimeksiannon suorittamisen kannalta, niin mielenkiintoista kun se onkin.

Ohjelmistoradion ohjelmointi valmiiden ohjainkirjastojen avulla ei vaadi radioteorian tuntemista vaan riittää, että omaksuu tarjotun ohjelmointirajapinnan ja on joskus käyttänyt radiota. En kokenut ohjainkirjaston ja käyttörajapinnan välisen kerroksen suunnittelua ja ohjelmointia kovin teknisesti haastavaksi suoritukseksi. Työn haastavuus tulikin sen teoreettisesta laajuudesta, tehtävien tärkeyden arvioinnista ja ajankäytön tasapainotuksesta aikataulussa pysymisen kannalta. Koska tehtävänä oli demon ohjelmointi, niin kaikkien radiovirittimen toimintojen toteuttaminen ja hyödyntäminen ei ollut olennaista. Jos kyseessä olisi lopullinen tuote, niin asia voisi olla toisin: Pitäisi varmistaa, että kaikki tarvittavat ja tarjolla olevat ominaisuudet on varmasti implementoitu ja toimivat kunnolla.

Työn aikana huomasin, miten ehdottoman tarpeellista selkeä ja kattava dokumentointi voi olla. Jos kehityskohdetta ei ole dokumentoitu huolellisesti, sen toiminnasta ja käyttöönotosta on helppoa saada vääristynyt kuva. Tällöin dokumentaation tekemättä jättämisellä säästetty työaika joudutaan käyttämään myöhemmin niin omalta kuin asiakkaankin osalta.

Aiheen teoriapohjan laajuuden ja aikataulun huomioiden olen tyytyväinen opinnäytetyön tulokseen ja koen oppineeni tärkeitä puolia ohjelmistokehityksestä ja insinöörin työstä.



## Lähteet

1. Heinrich Hertz - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: [https://en.wikipedia.org/wiki/Heinrich\\_Hertz](https://en.wikipedia.org/wiki/Heinrich_Hertz). Accessed 25.2.2017, 2017.
2. History of radio - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: [https://en.wikipedia.org/wiki/History\\_of\\_radio](https://en.wikipedia.org/wiki/History_of_radio). Accessed 25.2.2017, 2017.
3. Radio's First Voice - Mervyn C. Fry Available at: [https://www.ieee.ca/millennium/radio/radio\\_birth.html](https://www.ieee.ca/millennium/radio/radio_birth.html). Accessed 25.2.2017, 2017.
4. How Vacuum Tubes Work - Eric Barbour Available at: [http://www.vacuumtubes.net/How\\_Vacuum\\_Tubes\\_Work.htm](http://www.vacuumtubes.net/How_Vacuum_Tubes_Work.htm). Accessed 25.2.2017, 2017.
5. Sir Ambrose Fleming: Father of Modern Electronics - Jerry Bergman Available at: <http://www.icr.org/article/sir-ambrose-fleming-father-modern-electronics>. Accessed 25.2.2017, 2017.
6. Thomas H. Lee, The Design of CMOS Radio-Frequency Integrated Circuits. Cambridge University Press 2004.
7. Edwin Armstrong: Pioneer of the Airwaves - Columbia Magazine Available at: <http://www.columbia.edu/cu/alumni/Magazine/Spring2002/Armstrong.html>. Accessed 25.2.2017, 2017.
8. Radio Receiver - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: [https://en.wikipedia.org/wiki/Radio\\_receiver](https://en.wikipedia.org/wiki/Radio_receiver). Accessed 25.2.2017, 2017.
9. Transistor - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: <https://en.wikipedia.org/wiki/Transistor>. Accessed 25.2.2017, 2017.
10. The History of the Integrated Circuit - Nobel Media AB Available at: [http://www.nobelprize.org/educational/physics/integrated\\_circuit/history](http://www.nobelprize.org/educational/physics/integrated_circuit/history). Accessed 25.2.2017, 2017.

11. What is Amplitude Modulation - Adrio Communications Ltd. Available at: <http://www.radio-electronics.com/info/rf-technology-design/am-amplitude-modulation/what-is-am-tutorial.php>. Accessed 25.2.2017, 2017.
12. What is FM: Frequency Modulation Tutorial - Adrio Communications Ltd. Available at: <http://www.radio-electronics.com/info/rf-technology-design/fm-frequency-modulation/what-is-fm-tutorial.php>. Accessed 25.2.2017, 2017.
13. Frequency Modulation Advantages & Disadvantages - Adrio Communications Ltd. Available at: <http://www.radio-electronics.com/info/rf-technology-design/fm-frequency-modulation/advantages-disadvantages.php>. Accessed 25.2.2017, 2017.
14. What is PSK, Phase Shift Keying - Adrio Communications Ltd. Available at: <http://www.radio-electronics.com/info/rf-technology-design/pm-phase-modulation/what-is-psk-phase-shift-keying-tutorial.php>. Accessed 25.2.2017, 2017.
15. Software Defined Radio, SDR, Tutorial - Adrio Communications Ltd. Available at: <http://www.radio-electronics.com/info/rf-technology-design/sdr/software-defined-radios-tutorial.php>. Accessed 25.2.2017, 2017.
16. About - Yocto Project, A Linux Foundation Collaborative Project Available at: <https://www.yoctoproject.org/about>. Accessed 16.4.2017, 2017.
17. Member Organizations - Yocto Project, A Linux Foundation Collaborative Project Available at: <https://www.yoctoproject.org/ecosystem/member-organizations>. Accessed 16.4.2017, 2017.
18. Supporting Organizations - Yocto Project, A Linux Foundation Collaborative Project Available at: <https://www.yoctoproject.org/ecosystem/supporting-organizations>. Accessed 16.4.2017, 2017.
19. Participation - Yocto Project, A Linux Foundation Collaborative Project Available at: <https://www.yoctoproject.org/about/participation>. Accessed 16.4.2017, 2017.
20. Yocto Project Mega-Manual - Scott Rifenbark, Intel Corporation Available at: <http://www.yoctoproject.org/docs/2.2.1/mega-manual/mega-manual.html>. Accessed 16.4.2017, 2017.
21. MediaWiki - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: <https://en.wikipedia.org/wiki/MediaWiki> Accessed 16.4.2017, 2017.

22. What is MediaWiki - MediaWiki. Wikimedia Foundation, Inc. Available at: [https://www.mediawiki.org/wiki/Manual:What\\_is\\_MediaWiki](https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki) Accessed 16.4.2017, 2017.
23. Virtualization - Margaret Rouse and Brian Kirsch Available at: <http://searchservervirtualization.techtarget.com/definition/virtualization> Accessed 16.4.2017, 2017.
24. Hardware virtualization - Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. Available at: [https://en.wikipedia.org/wiki/Hardware\\_virtualization](https://en.wikipedia.org/wiki/Hardware_virtualization) Accessed 16.4.2017, 2017.
25. STA1295 Product Specifications v1.0 - STMicroelectronics Available at: [http://www.st.com/resource/en/data\\_brief/sta1295.pdf](http://www.st.com/resource/en/data_brief/sta1295.pdf) Accessed 16.4.2017, 2017.
26. TDA7707 Product Specifications v1.0 - STMicroelectronics Available at: [http://www.st.com/resource/en/data\\_brief/tda7707.pdf](http://www.st.com/resource/en/data_brief/tda7707.pdf) Accessed 16.4.2017, 201