NTNU
Det skapende universitet

Om implementering av frekvenshopping i OpenBTS.

Morten Bremseth

Master i elektronikk

Oppgaven levert: August 2011
Hovedveileder: Lars Magne Lundheim, IET
Biveileder(e): Stig Frode Mjølsnes, IMTE
Torleif Maseng, FFI

Norges teknisk-naturvitenskapelige universitet

Sammendrag

Denne Rapporten gir en forklaring på hva programvaredefinert radio er generelt. Den omtaler en konkret realisering ved hjelp av programvaren GNU Radio og den universale radioenheten USRP (Universal Software Radio Peripheral). GSM forklares slik at leseren får den nødvendige oversikten som trengs for å forstå resten av rapporten. Frekvenshopping forklares generelt og hvordan det er gjort i GSM. Det fremstilles en mulig realisering av frekvenshopping med USRP. Programvaren som benyttes for å opprette et GSM-nettverk presenteres. Programvaren som trengs er OpenBTS, GNU Radio og Asterisk. Den fysiske oppkoblingen av USRP når OpenBTS benyttes i programvare GSM blir forklart og hvilket testmiljø eksperimentene ble utført i beskrives. Delvis vellykkede resultater presenteres og rapporten konkluderer med at det skal være mulig å implementere frekvenshopping i OpenBTS med USRP som RF-maskinvare.

Oppgaveteksten

Tittel: Om implementering av frekvenshopping i OpenBTS.

Bakgrunn

GSM-standarden for mobilkommunikasjon inneholder enn del mekanismer for sikkerhet mot avlytting og andre typer angrep. Den teknolgoiske utviklingen i senere år har gjort det stadig lettere for en tredjepart å utføre slike angrep. En masteroppgave ved NTNU fra 2010 [1] tar for seg mulige scenarier for angrep mot GSM og viser hvordan blant annet fritt tilgjengelige program- og maskinvareverktøy som USRP, OpenBTS og Airprobe kan brukes til dette.

Ekseperimentene rapportert i [1] demonstrerer bare delvis vellykkede angrep mot sikkerheten. En av hindringene var at OpenBTS ikke legger til rette for frekvenshopping.

Oppgavebeskrivelse

Kandidaten skal sette seg inn i aktuelle deler av GSM-standarden og hvordan denne er implementert i OpenBTS. Han skal videre vurderere realiserbarheten av å inkludere frekvenshopping i en SDR-implementering ved hjelp av USRP, og om mulig realisere dette.

Faglærer: Lars Lundheim, IET. Veiledere: Torleiv Maseng, FFI, Stig Frode Mjølsnes, ITEM.

Referanse: [1] M. Glendrange, K. Hove og E. Hvideberg: Decoding GSM", Maseroppgaverapport, Institutt for telematikk, NTNU, 2010.

Innhold

Figurer

Tabeller

Forkortelser

Her er forkortelser og akronymer listet opp, en god del av disse er relatert til GSM og er hentet fra [ETSa].

ADC	Analog-til-digital-omformer (Analog to Digital Converter)
ARFCN	Radiokanal (Absolute Radio Frequency Channel Number)
AuC	autentiseringssenteret (Authentication Center)
BCCH	Kringkastingskontrollkanal (Broadcast Control CHannel)
BSC	Basestasjonskontroller (Base Station Controller)
BSS	Basestasjonsystem (Base Station System)
BTS	Basestasjonsender/mottaker (Base Transceiver Station)
CA	Celleallokasjonen (Cell Allocation)
CCH	Signaleringskanal (CCH, Control CHannel)
CTS	Trådløst telefonsystem (Cordless Telephony System)
DAC	Digital-til-analog-omformer (Digital to Analog Converter)
DC	Likestrøm (Direct Current) har frekvens lik 0 Hz
DUC	Digital-opp-konverterer (Digital Up Converter)
EIR	Utstyridentitetsdatabasen (Equiptment Identity Register)
ETSI	European Telecommunications Standards Institute
FDMA	Frekvensdelt multippel aksess (Frequency Division Multile Access)
FFH	Rask frekvenshopping (Fast Frequency Hopping)
FN	Tidsrammenummer (Frame Number)
GMSC	'Inngangsporten' til mobilnettet (Gateway Mobile-services Switching Center)
GMSK	Gaussisk minimum-skift nøkling (Gaussian Minimum Shift Keying)
GNU	står for GNU's Not Unix, som er en rekursiv forkortelse som betyr GNU er ikke Unix
GRC	GNU Radio sitt grafiske utviklingsgrensesnitt (GNU Radio Companion)
GSM	Globalt System for Mobilkommunikasjon (Global System for Mobile communications eller originalt fra Groupe Spécial Mobile)
HBS	Hjemmebasestasjon (Home Base Station)
HLR	hjemmelokasjonsdatabasen (Home Location Register)
HSN	Hoppesekvensnummeret (Hopping Sequence Number)

IF	Mellomfrekvens (Intermediate Frequency)
IMEI	Internasjonalt mobilutstyridentitetsnummer (International Mobile station Equipment Identity)
IMSI	Internasjonalt mobilabonnentnummer (International Mobile Subscriber Identity)
MA	Mobilallokasjonen (Mobile Allocation)
MAI	Mobilallokasjonindeks (Mobile Allocation Index)
MAIO	Mobilallokasjonindeksoffset (Mobile Allocation Index Offset)
MCC	Mobil land kode (Mobile Country Code)
ME	Mobilutstyr, eller mobiltelefon, ekskludert SIM kort (Mobile Equipment)
MNC	Mobil nettverk kode (Mobile Network Code)
MS	Mobilstasjon, som er mobiltelefon inkludert SIM kort, (Mobile Station)
MSC	Mobilsentral (Mobile-services Switching Centre)
MSIN	Mobilabonnentidentitetsnummer (Mobile Subscriber Identification Number)
PFD	Fase frekvens detektor (Phase Frequency Detector)
RF	Radiofrekvens (Radio Frequency)
RX	Mottaker (Resceiver)
RXA	Sokkel på USRP, for RX på kort A
RXB	Sokkel på USRP, for RX på kort B
SCH	Synkroniseringskanalen (Synchronization CHannel)
SDR	Programvaredefinert radio (Software Defined Radio)
SFH	Langsom frekvenshopping (Slow Frequency Hopping)
SIM	Abonnementidentitetskort (Subscriber Identity Module)
SIP	Session Initiation Protocol
SPI	Serielt grensesnitt til ytre enheter (Serial Peripheral Interface bus)
TCH	Trafikkanal (Traffic CHannel)
TDMA	Time Division Multiple Access
TMSI	Midlertidig mobilabonnementnummer (Temporary Mobile Subscriber Identity)
TRX	Sender og mottaker i ett (Transceiver)
TX	Sender (Transmitter)
TXA	Sokkel på USRP, for TX på kort A
TXB	Sokkel på USRP, for TX på kort B
USA	Amerikas forente stater (United States of America)
USB	Universal Serial Bus
USRP	Universell radioenhet (Universal Software Radio Peripheral)
VCO	Spenningskontrollert oscillator (Voltage Controlled Oscillator)
VLR	Besøksdatabase (Visitor Location Register)
VoIP	Voice over Internet Protocol

Kapittel 1

Introduksjon

Denne rapporten forklarer hva frekvenshopping er og introduserer de delene av GSM som leseren må ha noe forståelse av for å få nytte av denne rapporten. Programvaredefinert radio blir forklart generelt før mer konkret maskinvare og programvare blir forklart, dette omfatter Universal Software Radio Peripheral (USRP), GNU Radio, OpenBTS, Asterisk og Smqueue.

1.1 Motivasjon

Motivasjonen for denne oppgaven er at i en tidligere masteroppgave gitt ved NTNU [MG10] ble eksprimentene bare delvis vellykkede, og et av hindrene var at OpenBTS ikke støttet frekvenshopping. Videre er motivasjonen for implementasjon av frekvenshopping i OpenBTS at det vil gi bedre kvalitet på overføringen mellom basestasjonen og mobilenhetene samt at det vil innføre noe sikkerhet ved at avlyttingsutstyret må støtte frekvenshopping.

1.2 Fungerende GSM basestasjon for 7000 kroner

Det er ganske utrolig at nær sagt hvem som helst nå kan anskaffe seg en fungerende GSM basestasjon ved å benytte gratis programvare på en datamaskin og en universell radioenhet. Den universelle radioenheten og de nødvendige tilleggskortene koster totalt 1285 dollar i skrivende stund fra Ettus Research LLC i USA. Når programmene er installert og konfigurert og radioenheten er tilkoblet datamaskinen kan vanlige GSM håndsett ringe og sende meldinger til hverandre gratis innenfor basestasjonens rekkevidde. Systemet kan utvides til å kunne nå videre ut av basestasjonens rekkevidde ved hjelp av Voice over Internet Protocol (VoIP) eller at systemet kan kobles til det vanlige telefonnettet. Dette kan gjøre et slikt system interessant for bedrifter med mange telefoner hvorav flere er mobiltelefoner.

1.3 Vær oppmerksom på

Frekvensbåndene som GSM benytter er det lisensiering på. Et annet moment er at når programvaren er installert og radioenheten er koblet til så fungerer systemet som en GSM-basestasjon. Dette medfører at andre mobiltelefoner kan koble seg til dette nettverket. Om dette da er et test nettverk vil det mest sannsynlig ikke støtte nødanrop selv om ukjente telefoner kan prøve å foreta et. Det kan gi fatale konsekvenser. Derfor påpekes det at et slikt nettverk bør kun opprettes inne i for eksempel et faradaybur.

1.4 Problemet

For at frekvenshopping skal kunne støttes må systemet først støtte mer enn en radiokanal. Det andre som er en utfordring er å hoppe mellom disse radiokanalene når flere er tilgjengelig.

1.5 Rapportoppbyggingen

Dette kapitlet gir en introduksjon til temaet som rapporten omhandler. Kapittel 2 gir en forklaring på hva programvaredefinert radio er generelt og omtaler en konkret realisering. Kapittel 3 forklarer GSM og gir den nødvendige oversikten som trengs for å forstå resten av rapporten. Kapittel 4 gir en forklaring på hva frekvenshopping er og en mulig realisering av frekvenshopping med USRP. Kapittel 5 omtaler programvaren som benyttes for å opprette et GSM-nettverk. Kapittel 6 forklarer den fysiske oppkoblingen av USRP og omtaler hvilket testmiljø eksperimentene ble utført i. Kapittel 7 presentere resultatene og kommenterer de. Kapittel 8 konkluderer rapporten.

Kapittel 2

Programvaredefinert radio

Dette kapitlet omhandler programvaredefinert radio generelt og en konkret realisering ved hjelp av programvaren GNU Radio og den universale radioenheten USRP (Universal Software Radio Peripheral).

2.1 Programvaredefinert radio

I programvaredefinert radio, også kalt Software Defined Radio (SDR), skal ideelt sett all prosessering av hele radiosignalet gjøres i programvare [Tho09]. Dette vil medføre at en hvilken som helst SDR vil kunne motta og sende radiosignaler etter nye standarder og protokoller kun med en oppdatering av programvaren og ingen endringer i maskinvaren. Mange radiosystemer er fortsatt avhengig av at vesentlige og store deler av maskinvaren byttes ut for å støtte nye standarder.

Med dagens teknologi er det ikke mulig å komme helt til ideell SDR, så en praktisk definisjon av en realiserbar SDR kan være:

'En programvaredefinert radio er et element i et trådløst nettverk der funksjonsmåte og parametre kan endres etter produksjonstidspunktet ved hjelp av programvare.' [Tho09]

For små produktserier, forskning og utvikling kan programvaredefinert radio være et godt alternativ for å holde kostnadene nede og at utviklingen kan gå raskt. Universal Software Radio Peripheral (USRP) og GNU Radio sammen med en Linux datamaskin vil være et eksempel på rimelig og relativt enkel SDR som kan brukes til mye forskjellig, blant annet FM-radio, GPS-mottaker og flere i tillegg til GSM basestasjon.

2.2 Universal Software Radio Peripheral

Universal Software Radio Peripheral (USRP) er en universell radioenhet som kan benyttes som en bro mellom analoge radiofrekvenssignaler (RF-signaler) i luften eller i kabler og den digitale verden i en datamaskin. I datamaskinen kan punktprøvene

av RF-signalet mottatt fra USRP via USB behandles videre i programvare for å dekode informasjonen. Programvare på datamaskinen kan også brukes til å generere punktprøver som leveres til USRP via USB slik at de blir sendt som RF-signaler. Programvaren som behandler eller genererer punktprøver definerer radioens funksjon og virkemåte. Deler av signalbehandlingskjeden i USRP er analog siden dagens teknologi ikke klarer å gjengi en ideell SDR.

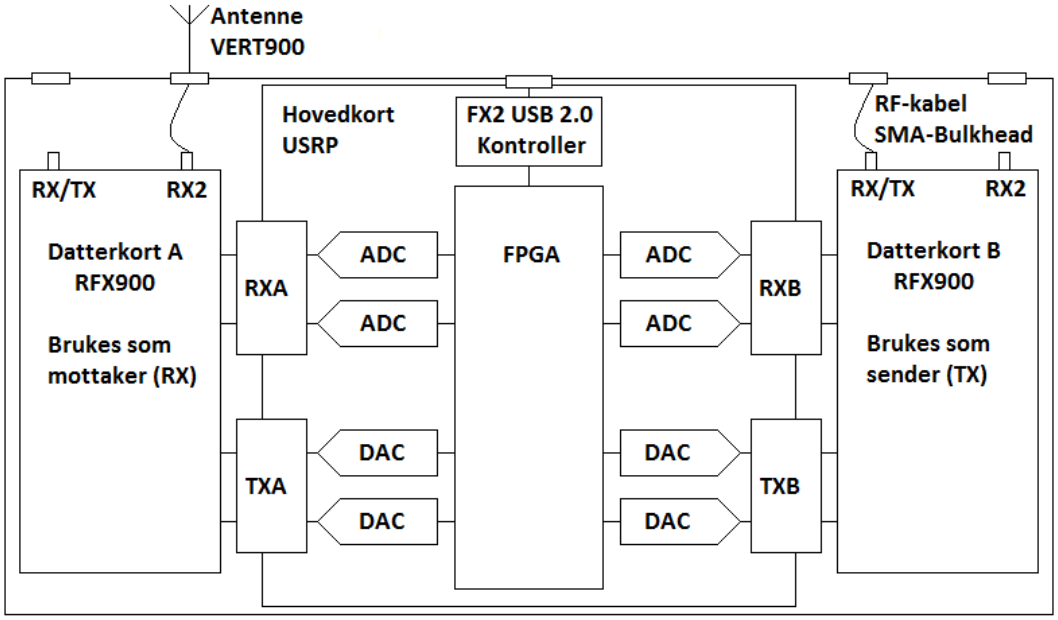
Den analoge delen kalles RF-kort eller datterkort og de monteres i USRP-enheten. Disse er nødvendig for å oppnå hele linken fra antennen til datamaskinen. Det finnes flere forskjellige datterkort som til sammen dekker store deler av frekvensspekteret mellom DC og 5.9 GHz, blant annet dekker datterkortet RFX900 frekvensene fra 800 MHz til 1.0 GHz. Skjemaet i figur 2.1 gir et overblikk av USRP med to transceiver datterkort. USRP består av et hovedkort med logikk og sokler som det kan tilkobles datterkort. Hovedkortet i USRP består blant annet av en USB 2.0 kontroller (heretter også kalt USB-kontroller), en enhet med programmerbar logikk (FPGA) som tar seg av opp- og ned-konverteringen av punktprøveraten, fire 64 MS/s (Mega-Sampler per sekund) 12 bit analog til digital-omformere (ADC), fire 128 MS/s 14 bit digital til analog-omformere (DAC) og fire sokler til datterkort [Ettc]. Det finnes flere forskjellige typer datterkort til USRP, de deles opp i sendere (TX), mottakere (RX) og transceivere (TRX). Hovedkortet har totalt fire sokler hvorav to er for TX datterkort (merket TXA og TXB) og to for RX datterkort (merket RXA og RXB). TX og RX datterkortene tar hver opp en sokkel (TXA eller TXB for TX og RXA eller RXB for RX) mens TRX datterkortene tar hver opp to sokler (TXA og RXA for kort A eller TXB og RXB for kort B) [Ettd]. USRP er designet for å fungere sammen med GNU Radio.

Det er utviklet flere USRP-utgaver i tillegg til den originale USRP. Den originale omtales både som USRP og USRP1, i denne rapporten benyttes USRP om denne enheten. De andre enhetene er USRP2, USRP E100, USRP N200 og USRP N210. Disse har alle noen forskjellige egenskaper og erstatter ikke den originale USRP.

2.3 GNU Radio

GNU Radio er en kraftig programvare som er gratis og åpen kildekode som brukes som et programvareutviklingsverktøy og utgjør grunnblokkene i SDR. GNU Radio tilbyr ferdige blokker for signalering og -behandling på vanlige hyllevare-prosessorer. Dette muliggjør rask implementasjon og utvikling av programvaredefinert radio som kan brukes sammen med forskjellig rimelig ekstern RF-maskinvare hvor USRP er et godt alternativ i mange situasjoner. Innen forskning og utvikling av produkter for trådløs kommunikasjon er GNU Radio mye brukt av skoler, private og kommersielle interesser [GNU].

Applikasjoner skrevet for GNU Radio er hovedsaklig skrevet i programmeringsspråket Python, mens den ytelseskritiske signalbehandlingen er implementert i C++. GNU Radio inkluderer også et grafisk utviklingsgrensesnitt som heter GNU Radio Companion (GRC) hvor man har tilgang på blokkene og grunnfunksjoner slik at de fleste applikasjonene kan utvikles ved å plassere blokker og koble disse sammen



Figur 2.1: Skjema over USRP og datterkort. Grunnlaget for skjemaet er hentet fra [Ettb]. Dette skjemaet viser oppkoblingen som benyttes for å gjøre USRP med to TRX datterkort til en GSM basestasjon når GSM-programvaren OpenBTS benyttes. OpenBTS benytter RXA som RX og TXB som TX, TXA og RXB benyttes ikke. OpenBTS omtales i kapittel 5.

grafisk. Dette er for mange en mye enklere måte å utvikel på enn å skrive programkoden selv. Dette lar utviklere utvikle høykapasitets sanntid radiosystemer i et utviklingsmiljø som er enkelt å bruke.

Ved å benytte tidligere lagret eller generert data kan GNU Radio brukes som et simuleringsverktøy uten RF-maskinvare for eksempel til utvikling av signalbehandlingsalgoritmer.

Kapittel 3

GSM

Globalt System for Mobilkommunikasjon (GSM) er et stort tema som beskrives med tusenvis av sider med dokumentasjon angående protokoller, signaler, standarder, og mer til via rekommandasjonene til European Telecommunications Standards Institute (ETSI). I dette kapitlet vil noen deler av GSM bli introdusert slik at leseren vil kunne danne seg et grovt overblikk, men dette blir på ingen måte en altomfattende beskrivelse av GSM. For å holde styr på de forkortelsene som brukes i denne rapporten er de listet opp i Forkortelser. Forkortelsene er i all hovedsak lik de som brukes i dokumentasjonen fra ETSI og dessverre er mange av forkortelsene svært like hverandre.

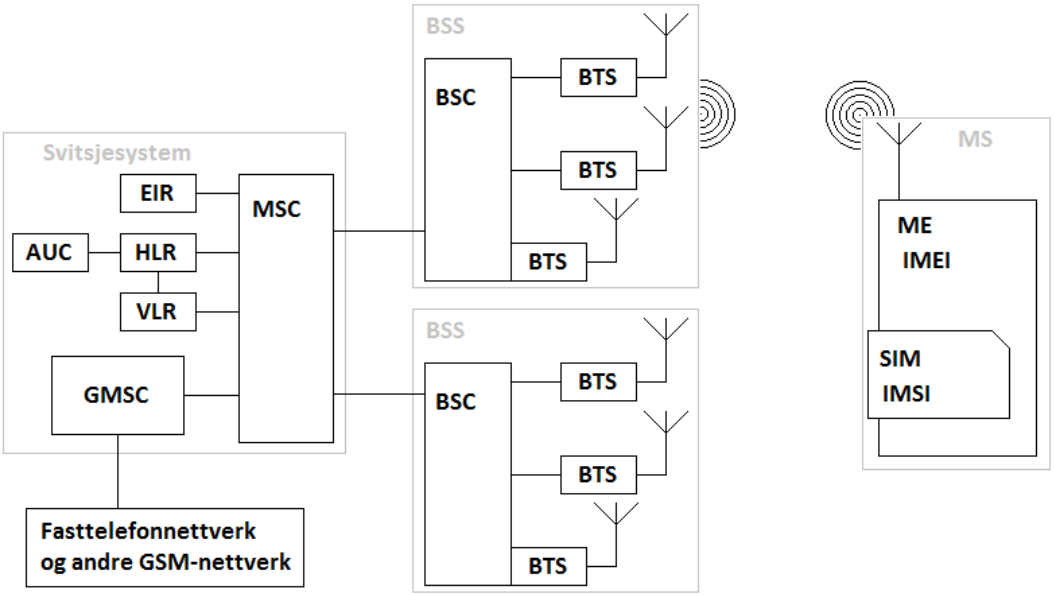
Hvert land som benytter GSM må identifisere mobilnettene i landet med en mobillandkode (MCC, Mobile Country Code) og hver mobilnettoperatør må identifisere sine mobilnett med en mobilnettverkkode (MNC, Mobile Network Code). MCC er en tresifret kode og MNC er tosifret, for begge gjelder intervallet $[0, 9]$. Norge har MCC = 242 mens mobilnettoperatørene Telenor, Netcom og Network Norway henholdsvis har MNC = 01, 02, 05 [MG10].

3.1 GSM overblikk

GSM består av flere enheter og deler som til sammen utgjør et system for digital mobilkommunikasjon. Disse er til dels kompliserte, men detaljkunnskap om alt er ikke nødvendig for å få det overblikket som trengs for å få nytte av denne rapporten. Figur 3.1 viser et skjema som gir et overblikk over de enhetene som er sammenkoblet i GSM. En grov oppdeling av GSM er svitsjesystemet, basestasjonsystemet (BSS, Base Station System) og mobilstasjonen (MS, Mobile Station) [Bju95].

3.1.1 Svitsjesystemet

Svitsjesystemet består i seg selv av flere enheter og databaser. Mobilsentralen (MSC, Mobile-services Switching Center) har som oppgave å foreta all svitsjing og handover inne i mobilnettet som innebærer å ivareta nødvendige funksjoner for at abonnenten kan forflytte seg. Alle samtaler krever et samtaleoppsett i starten av samtalen, dette



Figur 3.1: Overblikks skjema over GSM. Grunnlaget for skjemaet er hentet fra [Bju95]. Dette skjemaet viser hvilke enheter i GSM som er sammenkoblet.

styres også av MSC. MSC og GMSC (Gateway Mobile-services Switching Center) skal være leddet som binder mobilnettet sammen med andre GSM-nett og det offentlige telefonnettet, GMSC kan oppfattes som 'inngangsporten' til mobilnettet. Videre har svitsjesystemet som oppgave å holde rede på lokasjonen for abonnentene, hjemmelokasjonsdatabasen (HLR, Home Location Register) og besøksdatabasen (VLR, Visitor Location Register) benyttes til å holde styr på dette. Alle mobilabonnenter får tildelt en plass i et HLR ved abonnementoppstart. Denne plassen vil inneholde abonnementinformasjon og mobilabonnementens lokasjon innenfor GSM-systemet, det vil si at HLR inneholder hvilket VLR mobilabonnementen er tilkoblet eller sist var tilkoblet. VLR er en database som inneholder data om alle mobilstasjonene (MS, Mobile Station) som til enhver tid er innenfor dekningsområdet til MSCen. Svitsjesystemet skal også ta seg av autentisering og utstyrstilgang. Dette tas hånd om av autentiseringssenteret (AuC, Authentication Center) og utstyridentitetsdatabasen (EIR, Equiptment Identity Register). AuC håndterer sikkerhetsfasilitetene og konfidensialitet innenfor systemet, dette betyr at AuC produserer data for abonnentautentisering og kryptering av kommunikasjonen på radiostrekningen. EIR inneholder data som serienummer på typegodkjente mobilstasjoner (MS) som kan få tilgang på mobilnettet. EIR kan også brukes til å nekte adgang til enkelte mobilstasjoner som av forskjellige grunner skal nektes tilgang, stjålne enheter kan være et eksempel [Bju95].

3.1.2 Basestasjonsystemet

Basestasjonsystemet (BSS, Base Station System) består av to typer enheter, basestasjonskontroller (BSC, Base Station Controller) og basestasjonssender/mottaker (BTS, Base Transceiver Station). Et basestasjonsystem (BSS) kan bestå av en BSC og en eller flere BTSer som kan være på samme geografisk plassering som BSCen eller ikke. En mobilsentral (MSC) kan være tilkoblet en eller flere basestasjonskontrollere (BSC). BTS er essensielt en transceiver og har som oppgave å opprette en celle for å sende radiosignaler til MSer og motta radiosignaler fra MSer, men den har noen flere oppgaver. Disse oppgavene omfatter å detektere når en MS ønsker kontakt med mobilnettet, å plassere et oppkall til en MS i riktig signaleringsblokk og å detektere og rapportere til MS behovet for 'Timing Advance', altså når MS skal sende for at BTS skal motta signalet til riktig tid. BTS har videre oppgaver som å måle og rapportere til BSC interferensnivåer og blokkeringer på ledige radiokanaler (ARFCN, Absolute Radio Frequency Channel Number) og kvaliteten på overføringene på radiostrekningen til og fra hver enkelt MS, kryptere bitstrømmen som sendes på radiostrekningen til MS og dekryptere mottatt bitstrøm. En oppgave som enten kan utføres av BTS eller BSC er koding og transkoding mellom 13 kbit/s tale som benyttes på radiostrekningen og 64 kbit/s tale som benyttes i fasttelefonnettet. Basestasjonskontrolleren (BSC) er enheten med intelligens i et BSS. Denne har mer komplekse oppgaver som administrasjon av hvilke radiokanaler (ARFCN) som benyttes til trafikk og signalering over radiostrekningen, administrasjon av frekvenshopping, om dette benyttes, som innbefatter styring av frekvenshoppesekvensen og at denne blir overført til BTSene og MSene. BSC har også som oppgave å styre sendereffekten til BTSene og MSene og vil om målte data tilsier det initiere og gjennomføre en intern handover som så rapporteres til MSC. Intern handover vil si at for eksempel en samtale blir flyttet til en annen BTS som tilhører samme BSC, altså i samme BSS, motstykket er ekstern handover. Ekstern handover er at for eksempel en samtale blir flyttet til en annen BTS som tilhører en annen BSC, altså i et annet BSS, det er MSC som initierer denne. BSC oversender krypteringsnøkler til BTSene slik at kryptering og dekryptering kan foretas [Bju95].

3.1.3 Mobilstasjonen

Mobilstasjonen (MS, Mobile Station), eller selve mobiltelefonen, består av mobilutstyret (ME, Mobile Equiptment) og abonnementidentitetskortet (SIM, Subscriber Identity Module). Mobilutstyret har et internasjonalt identitetsnummer (IMEI, International Mobile station Equiptment Identity) [ETSd]. SIM-kortet inneholder blant annet et internasjonalt mobilabonnentnummer (IMSI, International Mobile Subscriber Identity), dette blir brukt av mobilsentralen (MSC) til å identifisere abonnenten og blir erstattet av ett midlertidig mobilabonnementnummer (TMSI, Temporary Mobile Subscriber Identity) for at ikke IMSI skal sendes unødvendig. IMSI er et 15 siffer langt nummer som består av mobillandkode (MCC, Mobile Country Code), mobilnettverkkode (MNC, Mobile Network Code) og et mobilabonnentidentitetsnummer (MSIN, Mobile Subscriber Identification Number), alle sifrene skal

være i intervallet $[0, 9]$. Siden TMSI kun er gyldig innenfor et område med samme VLR er dette et nummer med 32 bit. Kodingen og strukturen på disse kan bestemmes i samarbeid mellom operatør og leverandør slik at lokale behov blir ivaretatt. Det er normalt at deler av TMSI bestemmes av tidspunktet det opprettes. Videre skal ikke kombinasjonen hvor alle 32 bittene er satt til 1 tildeles en MS, dette er fordi denne kombinasjonen brukes av SIM-kortet til å indikere at det ikke er noe gyldig TMSI tilgjengelig. TMSI kan kodes ved bruk av heksadesimale tall, altså åtte siffer i intervallet $[0, F]$. IMEI er også et 15 siffer langt nummer hvor hvert siffer skal være i intervallet $[0, 9]$. De seks første sifrene er et typegodkjenningsnummer, de to neste er et produsentnummer. Så er de seks neste et serienummer som i kombinasjon med typegodkjenningsnummeret og produsentnummeret er individuelt for hver enhet. Det siste sifferet skal være null når det sendes fra MS. Det er IMEI som brukes sammen med EIR i svitsjesystemet til å identifisere mobilutstyr som skal og ikke skal få tilgang til mobilnettet. Mens for å identifisere mobilabonnenter er det IMSI og TMSI som brukes sammen med HLR, VLR og AuC.

3.2 Kanaler

I GSM er det definert mange kanaler. Disse er delt opp i to hovedkategorier, logiske kanaler og fysiske kanaler [ETSc][Bju95]. Kanalene under disse er ordnet i et hierarki. I de kommende avsnittene blir de to hovedkategoriene definert og forklart. Noen kanaler i disse blir også forklart, men ikke alle.

3.2.1 Logiske kanaler

Logiske kanaler overfører data av forskjellig karakter og disse er delt opp i to typer, trafikkanaler (TCH, Traffic CHannel) og signaleringskanaler (CCH, Control CHannel).

Trafikkanaler er definert for å overføre kodet tale og data. Overføring skjer enten i fullrate trafikkanal (TCH/F) eller halvrate trafikkanal (TCH/H). Hvor TCH/F kan overføre informasjon med en brutto datahastighet på 22.8 kbit/s og TCH/H kan overføre informasjon med den halve brutto datahastigheten, 11.4 kbit/s. Det er verd å merke seg at kodet tale og data ikke kan overføres på samme TCH. Det er likevel slik at flere TCHer kan allokeres til samme mobilstasjon (MS). Eksempel på dette er at en TCH/H for kodet tale og en TCH/H for data blir allokert til samme MS. Denne kombinasjonen overfører tilsammen med samme datahastighet som en fullrate trafikkanal og alternerer mellom de to TCH/Hene. I noen tilfeller er det behov for større overføringshastighet, og da allokeres flere TCH/F for kodet tale og data. Dette kalles multitidslukekonfigurasjon fordi denne konfigurasjonen beslaglegger flere tidsluker i den fysiske kanalen. Tidsluker blir forklart nærmere i avsnittet om fysiske kanaler 3.2.2.

Signaleringskanaler overfører synkroniseringsdata og signalering i fire typer kanaler. Disse er kringkastingskanaler, fellessignaleringskanaler, dedikerte signaleringskanaler og trådløst telefonsystem kontrollkanaler (CTS, Cordless Telephony System

control channels). Kringkastingskanalene brukes kun til å sende informasjon til MS. Denne informasjonen er blant annet generell informasjon om nettet, nabocellene, cellen MS er lokalisert i og maksimal effekt i denne cellen. Videre lytter MS til disse kanalene for å få bærebølge-, bit- og rammesynkronisering. Parametrene som trengs for frekvenshopping (FH) sendes på kringkastingskontrollkanalen (BCCH, Broadcast Control CHannel) og synkroniseringskanalen (SCH, Synchronization CHannel). Fellessignaleringskanalene brukes til å kalle opp mobilstasjonen eller basestasjonen for å motta eller opprette for eksempel en samtale, videre vil en dedikerte signaleringskanal bli tildelt MS på denne kanalen. De dedikerte signaleringskanalene blir brukt til overføring av systeminformasjon, målerapporter og signalering i forbindelse med handover. Den siste typen signaleringskanaler er i forbindelse med trådløst telefonsystem (CTS eller GSM-CTS) som er et system som gjør det mulig å koble en CTS-hjemmebasestasjon (HBS, Home Base Station) til det offentlige telefonnettet for å bruke en GSM-mobiltelefon som fasttelefon. CTS-kontrollkanalene er en noe nedstrippet utgave av de tre foregående kanaltypene og brukes i et mobilnett som er beregnet for innendørs bruk i hjemmet.

Tillate kombinasjoner av logiske kanaler til fysiske kanaler er gitt i ETSI anbefalingene i [ETSb].

3.2.2 Fysiske kanaler

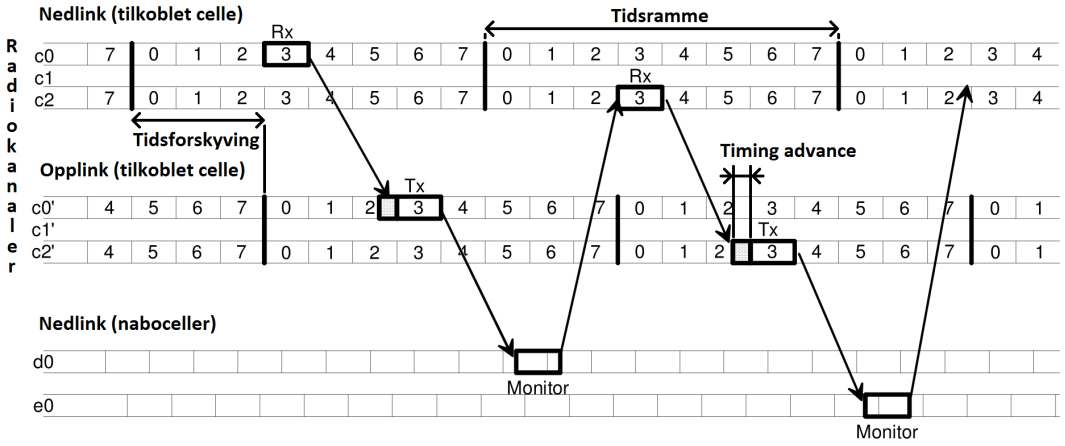
En fysisk kanal i GSM bruker en kombinasjon av frekvensdelt og tidsdelt multippel aksess og er definert av en sekvens av radiokanaler og tidsluker.

Frekvensdelt multippel aksess (FDMA, Frequency Division Multile Access) betyr at et frekvensbånd blir delt opp i kanaler som har et nummer. I GSM kalles numrene radiokanalnummer eller ARFCN (Absolute Radio Frequency Channel Number). ARFCN er et nummer i intervallet $[1, 1023]$ som bestemmer hvilke bærebølgefrekvenser en radiokanal har. En radiokanal har en opplink og en nedlink som hver har en bærebølgefrekvens. Opplink benyttes av mobilstajonen (MS) til å sende radiosignaler til basestasjonen (BTS) og nedlink benuyttes av BTS til å sende radiosignaler til MS. Disse radiosignalene bærer dataene som skal sendes mellom MS og BTS. Sammenhengen mellom bærebølgefrekvensene til radiokanalene og ARFCN er gjengitt i tabell 3.1 sammen med navn på og frekvensområdene som benyttes av forskjellige bånd i GSM. Båndbredden til en radiokanal er 200 kHz for opplink og nedlink. Avstanden mellom bærebølgene til opplink og nedlink i frekvens er forskjellig i de ulike båndene, men for alle GSM 900-båndene er avstanden 45 MHz. Avstanden mellom radiokanalene er den samme som båndbredden, altså 200 kHz. En BTS kan benytte en eller flere radiokanaler for kommunikasjon med MSer. Begrepet 'multiple ARFCN' benyttes om en BTS som benytter mer enn en radiokanal og følgelig vil ha mer enn en ARFCN assosiert med seg. Multiple ARFCN benyttes for å kunne betjene mange brukere, men er også en forutsetting for at BTSen kan benytte frekvenshopping, mer om dette temaet i kapittel 4. Figur 3.2 viser en multiple ARFCN BTS og hvordan en MS kan hoppe mellom forskjellige radiokanaler.

Tidsdelt multippel aksess (TDMA, Time Division Multiple Access) betyr at en radiokanal deles i tidsluker. Alle brukerne av radiokanalen modulerer sine data inn

Bånd	Link	Frekvensområde	ARFCN	Formler [MHz]
P-GSM 900 (Primary)	Opplink	$890 - 915$ MHz	$1 \leq n \leq 124$	$F_o(n) = 890 + 0.2 \cdot n$
	Nedlink	$935 - 960$ MHz		$F_n(n) = F_o(n) + 45$
E-GSM 900 (Extended)	Opplink	$880 - 915$ MHz	$1 \leq n \leq 124$ $975 \leq n \leq 1023$	$F_o(n) = 890 + 0.2 \cdot n$ $F_o(n) = 890 + 0.2 \cdot (n - 1024)$
	Nedlink	$925 - 960$ MHz		$F_n(n) = F_o(n) + 45$
R-GSM 900 (Railways)	Opplink	$876 - 915$ MHz	$1 \leq n \leq 124$ $955 \leq n \leq 1023$	$F_o(n) = 890 + 0.2 \cdot n$ $F_o(n) = 890 + 0.2 \cdot (n - 1024)$
	Nedlink	$921 - 960$ MHz		$F_n(n) = F_o(n) + 45$
DCS 1800	Opplink	$1710 - 1785$ MHz	$512 \leq n \leq 885$	$F_o(n) = 1710.2 + 0.2 \cdot (n - 512)$
	Nedlink	$1805 - 1880$ MHz		$F_n(n) = F_o(n) + 95$
PCS 1900	Opplink	$1850 - 1910$ MHz	$512 \leq n \leq 810$	$F_o(n) = 1850.2 + 0.2 \cdot (n - 512)$
	Nedlink	$1930 - 1990$ MHz		$F_n(n) = F_o(n) + 80$
GSM 450	Opplink	$450.4 - 457.6$ MHz	$259 \leq n \leq 293$	$F_o(n) = 450.6 + 0.2 \cdot (n - 259)$
	Nedlink	$460.4 - 467.6$ MHz		$F_n(n) = F_o(n) + 10$
GSM 480	Opplink	$478.8 - 486$ MHz	$306 \leq n \leq 340$	$F_o(n) = 479 + 0.2 \cdot (n - 306)$
	Nedlink	$488.8 - 496$ MHz		$F_n(n) = F_o(n) + 10$
GSM 850	Opplink	$824 - 849$ MHz	$128 \leq n \leq 251$	$F_o(n) = 824.2 + 0.2 \cdot (n - 128)$
	Nedlink	$869 - 894$ MHz		$F_n(n) = F_o(n) + 45$

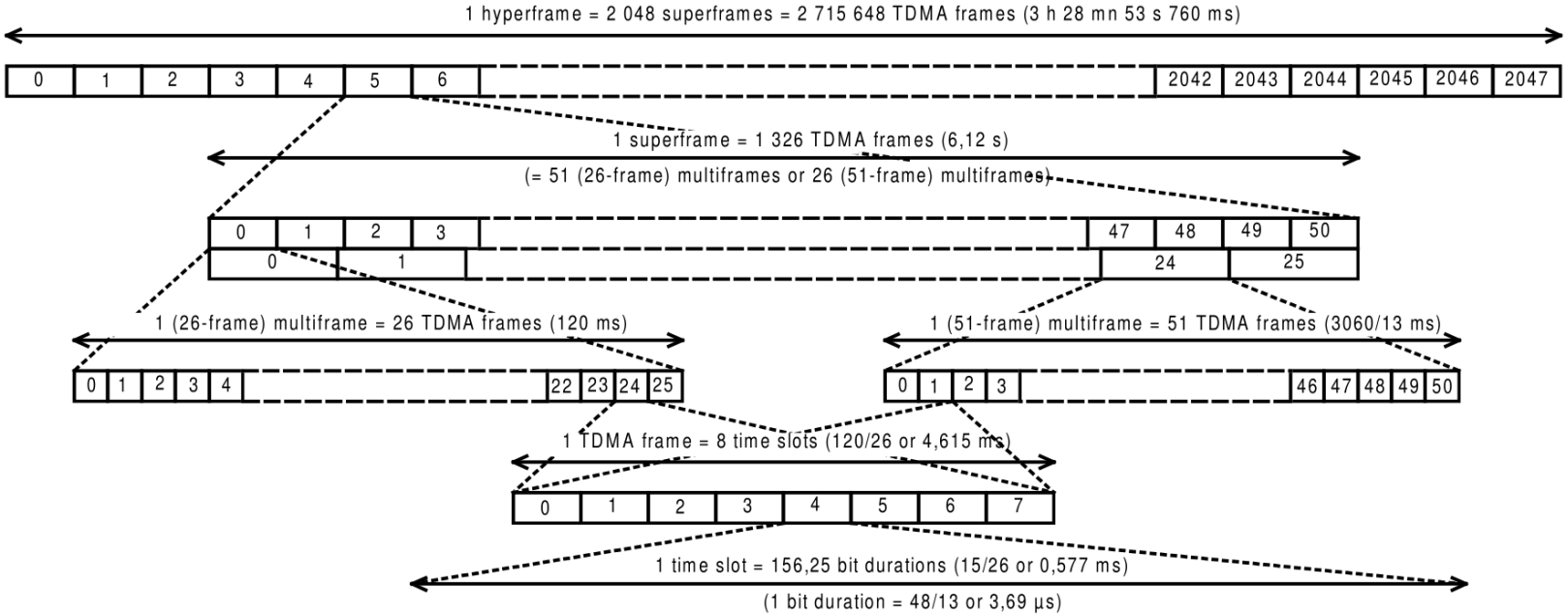
Tabell 3.1: Tabell over bærebølgefrekvensformler, frekvensområde og ARFCN i GSM. Grunnlaget er hentet fra [ETSf]. Tabellen viser navn på og frekvensområdene som benyttes i GSM sammen med formlene for å rekne ut bærebølgefrekvensene til opplink, $F_o(n)$, og nedlink, $F_n(n)$, når ARFCN er oppgitt. Både båndbredden til en radiokanal og avstanden mellom radiokanalene er 200 kHz både for opplink og nedlink.



Figur 3.2: Figur over en MS sine hopp mellom forskjellige bærebølger. Grunnlaget for skjemaet er hentet fra [ETSc]. Denne figuren viser hvordan en MS hopper mellom flere bærebølger for å motta og sende data og monitorere nabocellene. Her er radiokanalene c1, c2, c3, d0 og e0 vist og c1 og c3 benyttes av MS for data mens d0 og e0 monitoreres. I denne ilustrasjonen benyttes frekvenshopping. Det er også vist at tidsrammene i nedlink og opplink er tidsforskjøvet med tre tidsluker. Videre er timing advance indikert, altså det at MS må sende litt før hele tidsforskyvingen har gått for at tidsluken skal komme frem til riktig tid når MS er et stykke unna BTS.

på samme bærebølge, men i ulike tidsluker. Dermed kan flere brukere benytte samme frekvens for dataoverføring. Tildeling av tidsluker kan gjøres dynamisk og da er TDMA fleksibelt. I GSM benyttes det TDMA både på opplink og nedlink, og tidslukene i de to retningene er tidsforskjøvet i forhold til hverandre slik at de ikke overlapper. Dette er vist i figur 3.2. Dette gjør at MS ikke trenger å lytte samtidig som den sender. Dermed kan MS benytte felles sender- og mottakerantenne som gir designfordeler. MS må sende litt før hele tidsforskyvingen har gått for at tidsluken skal komme frem til riktig tid når MS er et stykke unna BTS, dette kalles fortenning eller timing advance som også er vist i figuren. I GSM er åtte tidsluker organisert i en tidsramme. Tidslukene betegnes som tidsluke 0 (T0) til tidsluke 7 (T7), etter T7 følger T0 i neste tidsramme. Varigheten til en tidsluke er definert til 15/26 ms eller ca. 577 µs og da blir varightene til en tidsramme 60/13 ms eller ca. 4.62 ms. Den totale datahastigheten på en bærebølge i GSM er 1625/6 kbit/s eller ca. 271 kbit/s.

Tidsrammene er organisert i multirammer, superrammer og hyperrammer. Figur 3.3 viser oppbyggingen av disse. Den lengste rammen er hyperrammen, denne har en varighet på omtrent 3 timer og 29 minutter, og består av 2048 superrammer. En hyperramme består av totalt 2 715 648 tidsrammer som er nummerert. Tidsrammenummerene (FN, Frame Number) starter på 0 og går til og med 2 715 647. En superramme varer i 6.12 sekunder og kan bestå av forskjellige varianter og antall av multirammer, men som totalt består av 1326 tidsluker. Det er tre typer multirammer, 26-multiramme, 51-multiramme og 52-multiramme som hver består av henholdsvis 26, 51 og 52 tidsrammer.



Figur 3.3: Skjema over oppbyggingen av en hyperramme. Grunnlaget for skjemaet er hentet fra [ETSe]. Dette skjemaet viser oppbyggingen fra tidsluker til en hyperramme via tidsrammer, multirammer (26-multirammer og 51-multirammer) og superrammer.

Kapittel 4

Frekvenshopping

Frekvenshopping (FH) vil si at bærebølgefrekvensen til den fysiske kanalen mellom sender og mottaker endres med faste tidsintervall etter et mønster som er kjent for både sender og mottaker slik at de tunes til de samme frekvensene til riktig tid. Hovedsaklig deles frekvenshopping opp i to typer, langsom frekvenshopping (SFH, Slow Frequency Hopping) og rask frekvenshopping (FFH, Fast Frequency Hopping). Forskjellen på SFH og FFH er hoppehastigheten, mer spesifikt om hoppingen er raskere eller langsommere enn symbolraten på sendingen [Hay01].

4.1 Frekvenshopping i GSM

I GSM kan frekvenshopping benyttes, og det er da langsom frekvenshopping (SFH) som brukes. Bærebølgefrekvensen endres da mellom hver tidsramme, som blir omtrent hvert $4,62$ ms eller ca. 217 ganger i sekundet [Jea][ETSc]. Det er to realiseringstyper av FH i GSM, basisbånd FH og syntetisert FH. Forskjellen på disse er vist i figur 4.1. Som vist i figur 3.1 kan et mobilnett ha flere basestasjonsender/mottakere (BTS). Hver BTS har en eller flere celler for kommunikasjon med mobilstasjoner (MS). Videre har hver celle en eller flere transceivere (TRX) som hver benytter en radiokanal. I figur 4.1 er det vist hvordan fysiske kanaler kan hoppe mellom radiokanaler, men også at enkelte tidsrammer eller tidsluker ikke kan hoppe. Det er den første tidsluken (T0) i den første TRXen (C0) som ikke tillates å hoppe. Dette er fordi C0T0 benyttes som kringkastningskanal. Som figuren viser har hver TRX en fast radiokanal når basisbånd FH benyttes, mens radiokanalene flyttes mellom TRXene når syntetisert FH benyttes. Dette betyr at ingen av tidslukene som er på C0 kan hoppe når syntetisert FH benyttes. T1 til T7 på C0 kan benyttes selv om FH er aktivert, men vil ikke være med på hoppingen.

Alle radiokanalene som allokeres til en celle kalles celleallokasjonen (CA, Cell Allocation), ut av CA benyttes et sett av radiokanaler til FH. Dette settet kalles mobilallokasjonen (MA, Mobile Allocation), denne kan inneholde maksimalt 64 radiokanaler. Det at en celle inneholder mer enn en radiokanal kan være relatert til at cellen er dimensjonert til å betjene mange MSer. Om FH ikke benyttes i cellen vil CA inneholde like mange radiokanaler som det er TRXer i cellen fordi hver TRX

bruker da kun en radiokanal. Tilfellet er det samme når basisbånd FH benyttes, forskjellen er at de fysiske kanalene vil hoppe mellom TRXene. Når syntetisert FH benyttes vil hver TRX få tildelt fysiske kanaler. Frekvenshoppingen oppnås ved at hver TRX hopper mellom radiokanalene som er i MA, MA er ikke begrenset til antall TRXer når syntetisert FH benyttes. MA kan da godt inneholde flere radiokanaler enn TRXer, men begrenses av CA.

Videre er det to typer frekvenshoppingsmønstre, tilfeldig hopping og syklisk hopping. Lengden på hoppesekvensen er lik antallet radiokanaler i MA, N. Det er hoppesekvensnummeret (HSN, Hopping Sequence Number) som bestemmer hvilket hoppemønster som skal benyttes. Hoppesekvensen genereres enten av algoritmen i formel 4.1, syklisk hopping når $HSN = 0$, eller av algoritmen vist i blokkdiagrammet i figur 4.2, tilfeldig hopping når $1 \leq HSN \leq 63$. Ved syklisk hopping starter hoppingen på mobilallokasjonindeksen (MAI, Mobile Allocation Index) eller indeksen lik Mobilallokasjonindeksoffsetet (MAIO, Mobile Allocation Index Offset), $MAI = MAIO$. Videre hoppes det syklisk gjennom radiokanalene tilgjengelig i MA til høyeste indeks før det hoppes til laveste indeks for så å fortsette oppover igjen. Ved tilfeldig hopping genereres hoppesekvensen av en tilfeldig-tall-generator-algoritme med HSN, FN, MA og $MAIO$ som inndata i tillegg til at oppslagstabellen i tabell 4.1 benyttes. Tidsrammenummeret (FN, Frame Number) har et område fra 0 til 2 715 647, i algoritmen benyttes reduserete tidsrammenummre $T1$, $T2$ og $T3$ (som i denne sammenhengen ikke har noe med tidsluker å gjøre). Reduksjonene er vis i formel 4.2 til 4.4. Det er hver TRX sitt individuelle MAIO som forsikrer at hoppesekvensene er ortogonale når HSN er det samme i cellene, MAIO forteller hvor i hoppesekvensen TRXen skal begynne. Dette forhindrer to TRXer i samme celle å skape et uakseptabelt høyt interferensnivå. Parametrene som trengs for frekvenshopping (FH) sendes på kringkastingskontrollkanalen (BCCH, Broadcast Control CHannel) og synkroniseringskanalen (SCH, Synchronization CHannel).

$$MAI = (FN + MAIO) \text{ modulo } N \tag{4.1}$$

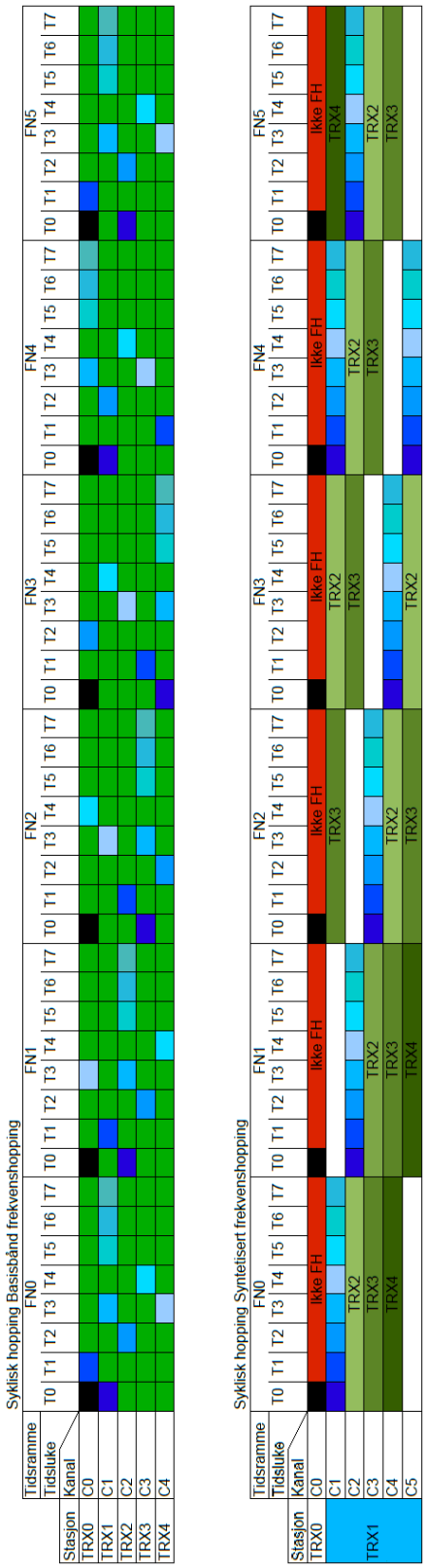
$$T1 = FN \text{ div } (26 \cdot 51) \tag{4.2}$$

$$T2 = FN \text{ modulo } 26 \tag{4.3}$$

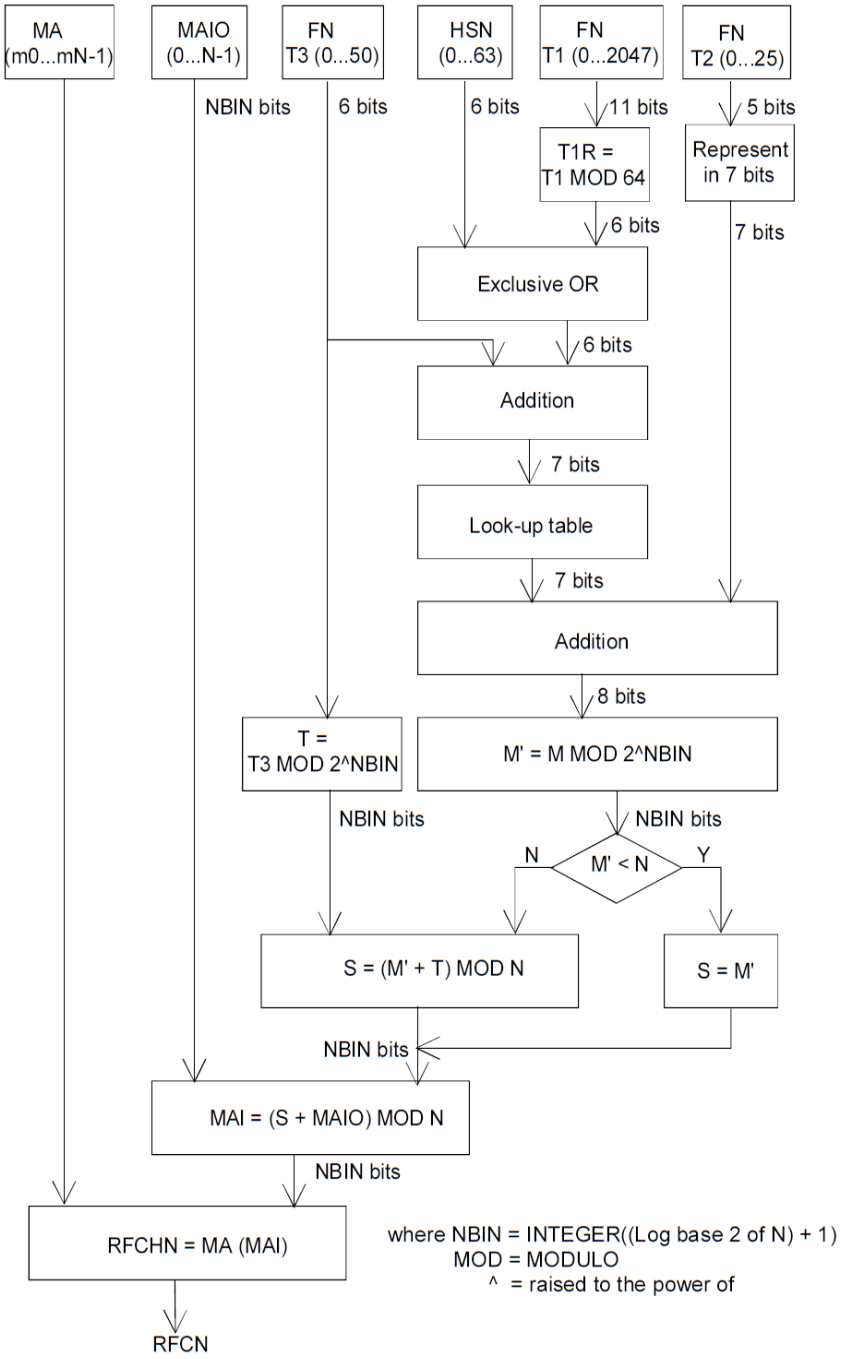
$$T3 = FN \text{ modulo } 51 \tag{4.4}$$

4.2 Frekvenshopping mulig på USRP?

For at det skal være mulig å få implementert frekvenshopping i OpenBTS med USRP som RF maskinvare må kombinasjonen støtte multiple ARFCN for at det skal være noen frekvenser å hoppe mellom [ETSc]. Da må USRP kunne representere flere TRX. I artikkelen [Dav] i seksjonen om maskinvare (3.8 Hardware) skriver forfatterne at USRP er rimelig og har nok båndbredde til å støtte frekvenshopping i



Figur 4.1: Syklisk frekvenshopping, basisbånd og syntetesert hopping. Grunnlaget for figuren er hentet fra [Jea]. Denne figuren viser hvordan fysiske kanaler vil kunne hoppe mellom radiokanaler. De forskjellige blåfargene representerer hver sin fysiske kanal. Den sorte tidsluken, C0T0, benyttes til krinkastingskanal og kan ikke hoppe.



Figur 4.2: Blokkdiagram over hoppesekvensalgoritmen for tilfeldig hopping, $HSN \neq 0$. Blokkdiagrammet er hentet fra [ETSc]. Dette blokkdiagrammet viser algoritmen som produserer hoppesekvensen til hver TRX i cellen. Algoritmen benytter HSN, FN (eller mer korrekt reduserete tidsrammenummre $T1$, $T2$ og $T3$), MA og $MAIO$ som inndata sammen med verdier fra oppslagstabellen gjenngitt i tabell 4.1.

Adresse	Verdier									
000 . . . 009	48	98	63	1	36	95	78	102	94	73
010 . . . 019	0	64	25	81	76	59	124	23	104	100
020 . . . 029	101	47	118	85	18	56	96	86	54	2
030 . . . 039	80	34	127	13	6	89	57	103	12	74
040 . . . 049	55	111	75	38	109	71	112	29	11	88
050 . . . 059	87	19	3	68	110	26	33	31	8	45
060 . . . 069	82	58	40	107	32	5	106	92	62	67
070 . . . 079	77	108	122	37	60	66	121	42	51	126
080 . . . 089	117	114	4	90	43	52	53	113	120	72
090 . . . 099	16	49	7	79	119	61	22	84	9	97
100 . . . 109	91	15	21	24	46	39	93	105	65	70
110 . . . 114	125	99	17	123						

Tabell 4.1: Oppslagstabell som hoppesekvensalgoritmen benytter. Tabellen er hentet fra [ETSc]. Tabellen viser verdiene som er med i addisjonen som produserer M i hoppesekvensalgoritmen som er vist i figur 4.2.

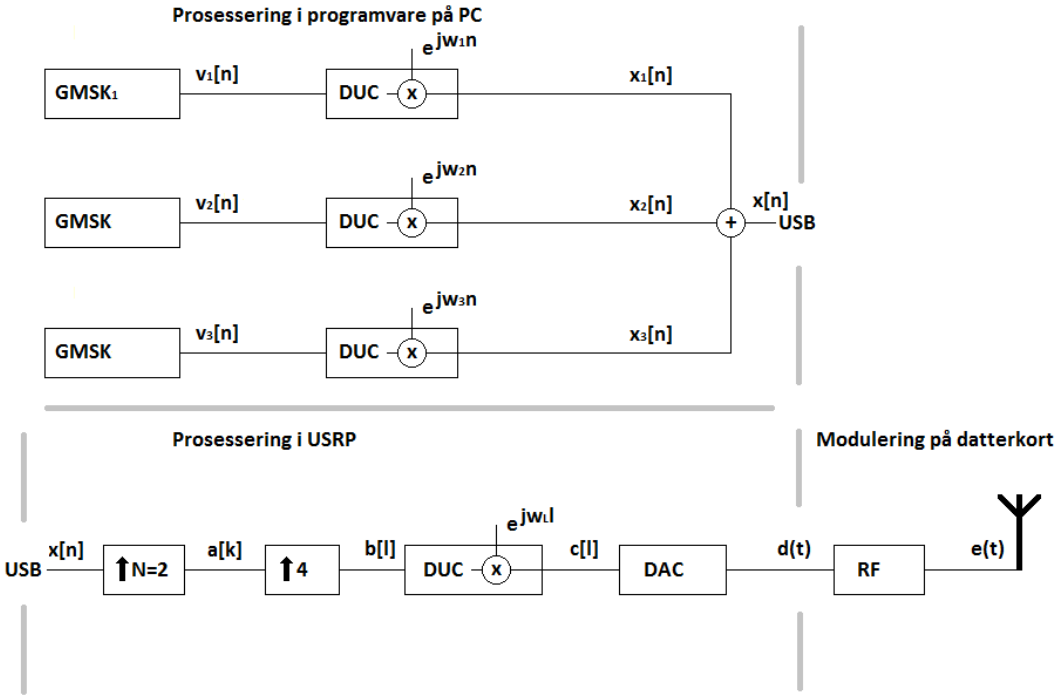
digitalt basisbånd etter hvert. I en epost fra den ene forfatteren av [Dav] får jeg vite at støtte av multiple ARFCN er implementert i en kommersiell utgave av OpenBTS sammen med ikke-USRP RF maskinvare, så OpenBTS har mulighet for flere ARFCN. OpenBTS omtales i kapittel 5.

4.3 Signalgangen for frekvenshopping med USRP

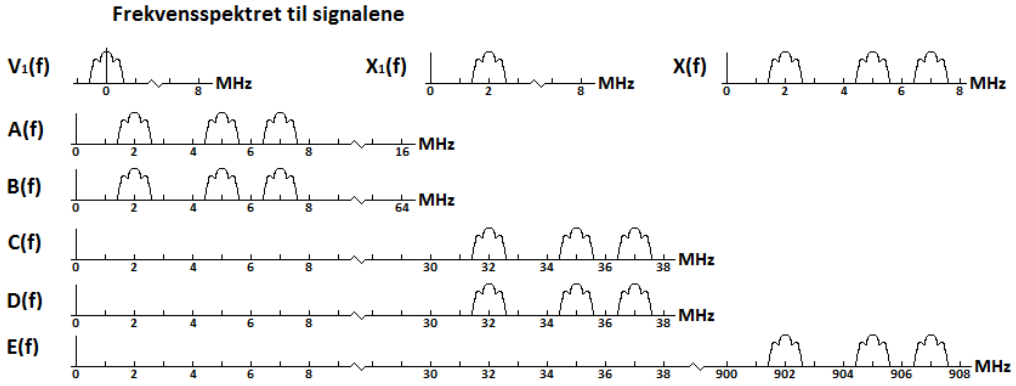
I blokkskjemaet i figur 4.3 vises en mulig løsning på signalgangen for sending (TX) med USRP hvor det er multiple ARFCN. I den offentlig tilgjengelige utgaven av OpenBTS støttes ikke multiple ARFCN når dette skrives, men en kommersiell utgave gjør [Mai]. Det blir først beskrevet hvordan systemet vil fungere med multiple ARFCNs som er en nødvendighet for å kunne støtte frekvenshopping og så forklares endringene eller forskjellene ved singel ARFCN som tilbys av den offenlige OpenBTS.

Ut fra hver GMSK-blokk i figur 4.3 kommer et ferdig GMSK-modulert signal for en radiokanal (ARFCN), $v_i[n]$. GMSK er en forkortelse for Gaussisk minimumskift nøkling (Gaussian Minimum Shift Keying). Dette signalet kan inneholde data fra flere MSer og eventuelle signaleringssignaler. Signalet er et basisbåndsignal og har en båndbredde på 200 kHz og kunne derfor hatt en samplingsfrekvens f_s ned mot 400 kHz og fortsatt oppfylt Nyquist-kriteriet, men er her satt til $f_s = 16$ MHz og gir dermed en aliasfri båndbredde på 8 MHz. I figur 4.4 er frekvensspekteret til signalene $v_1[n]$, $x_1[n]$, $x[n]$, $a[k]$, $b[l]$, $c[l]$, $d(t)$ og $e(t)$ i figur 4.3 illustrert, bredden på hver ARFCN er overdrevet noe for at de skal synes bedre på figuren.

Signalet, $v_i[n]$, fra hver GMSK blokk flyttes i frekvens med en digital-oppkonverterer (DUC), $x_i[n] = v_i[n] \cdot e^{j\omega_i n}$ hvor $\omega_i = 2\pi f_i$. I illustrasjonen er $f_1 =$



Figur 4.3: Skjema over signalgangen for sending (TX) fra et ferdig modulert GMSK basisbåndsignal for en ARFCN til RF. Frekvensspekteret til signalene $v_1[n]$, $x_1[n]$, $x[n]$, $a[k]$, $b[l]$, $c[l]$, $d(t)$ og $e(t)$ er presentert i figur 4.4, mens frekvensspekteret til $v_2[n]$, $v_3[n]$, $x_2[n]$ og $x_3[n]$ blir forklart i figurteksten.



Figur 4.4: Frekvensspekterene $V_1(f)$, $X_1(f)$, $X(f)$, $A(f)$, $B(f)$, $C(f)$, $D(f)$ og $E(f)$ til signalene $v_1[n]$, $x_1[n]$, $x[n]$, $a[k]$, $b[l]$, $c[l]$, $d(t)$ og $e(t)$ i skjemaet presentert i figur 4.3. Signalet $v_1[n]$ er et ferdig modulert GMSK basisbåndsignal for en ARFCN, bredden på hver ARFCN er overdrevet noe for at de skal synes bedre. Frekvensspekterene $V_2(f) = V_3(f) = V_1(f)$ til signalene $v_2[n]$ og $v_3[n]$ mens $X_2(f)$ og $X_2(f)$ til signalene $x_2[n]$ og $x_3[n]$ ser nesten lik ut som $X_1(f)$ bare med en større forskyvning fra 0, henholdsvis 5 og 7 mot 2 MHz.

2 MHz, $f_2 = 5$ MHz og $f_3 = 7$ MHz. Signalet fra de tre ARFCNene blir så samlet til ett signal, $x[n] = x_1[n] + x_2[n] + x_3[n]$, før signalet, samplene, sendes ut fra datamaskinen via USB til USRP. I USRP prosesseres signalet videre og det første som gjøres er at samplingsfrekvensen økes i to blokker. I første blokk blir $x[n]$ interpolert med en faktor $N = 2$ som resulterer i $a[k]$ med tilhørende $f_s = 32$ MHz. N er en konfigurerbar størrelse. Videre i andre blokk blir $a[k]$ interpolert med en fast faktor på 4 slik at $f_s = 128$ MHz for $b[l]$. Siden frekvensinnholdet ikke er endret, er ikke frekvensspekteret endret fra $x[n]$ til $b[l]$ heller. I neste omgang flyttes signalet opp i frekvens i to steg, først av hovedkortet til USRP opp til en mellomfrekvens (IF) og så av datterkortet i USRP til RF. I DUC-blokken flyttes frekvensinnholdet i $b[l]$ opp til en IF, $c[l] = b[l] \cdot e^{j\omega_L l}$ hvor $\omega_L = 2\pi f_L$. f_L er frekvensen signalet skal flyttes opp til, i illustrasjonen er $f_L = 30$ MHz. Frekvensspekteret til IF kan maksimalt være 64 MHz. Det digitale signalet $c[l]$ gjøres så om til et analogt signal $d(t)$ av en digital-til-analog-konverterer (DAC) før $d(t)$ sendes til datterkortet i USRP som tar seg av den siste flyttingen av signalet opp til ca 900 MHz som er den frekvensen RF signalet, $e(t)$, skal ha.

I den offentlig tilgjengelige utgaven av OpenBTS støttes ikke multiple ARFCN når dette skrives, altså sendes kun en ARFCN. Skjemaet i figur 4.3 vil da bare inneholde en GMSK-blokk, si GMSK1. Dette medfører at det ikke er nødvendig å flytte signalet opp i frekvens, det kan sies at $f_1 = 0$ og da blir $x_1[n] = v_1[n]$, og det er heller ingen andre signaler som skal kombineres slik at $x[n] = x_1[n] + 0 = v_1[n]$. Frekvensspekteret vil da forbli rundt 0 for $v_1[n]$, $x_1[n]$, $x[n]$, $a[k]$ og $b[l]$ mens frekvensspekteret til signalet etter DUC, $c[l]$, og DAC, $d(t)$, i USRP vil ligge rundt 30 MHz i illustrasjonen. Etter at datterkortet har gjort sitt ligger frekvensspekteret rundt 900 MHz i dette tilfellet. Siden samplingsfrekvensen f_s for signalet fra GMSK blokken, $v_1[n]$, kan ha en mye lavere verdi når kun en ARFCN behandles, si ned til 0.5 MHz, vil N kunne økes, si til 32, og da vil dataraten som overføres via USB minke til $\frac{1}{16}$ av hva raten var i tilfelle med multiple ARFCN beskrevet over.

Kapittel 5

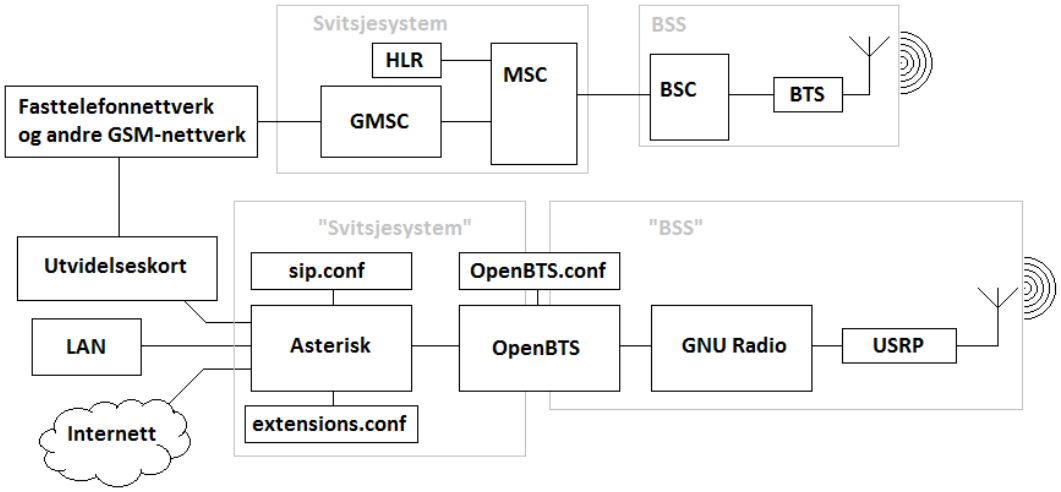
Programvare GSM

Det finnes områder og tilfeller hvor konvensjonelle GSM-nettverk ikke er kostnadseffektive og der kan programvare GSM-nettverk være et alternativ. Et programvare GSM-nettverk kan opprettes ved hjelp av gratis programvare og rimelig RF-maskinvare. Programvaren som trengs er OpenBTS, GNU Radio og Asterisk. RF-maskinvaren er USRP med to datterkort. Dette kan opprette et lukket nettverk hvor de mobilstasjonene (MS) som skal kommunisere må være tilkoblet denne cellen. Som figur 5.1 illustrerer vil MSene kunne få kontakt med omverdenen om utvidelseskort, tilgang til lokale nettverk eller Internett kobles sammen med systemet. GNU Radio og USRP utgjør en programvaredefinert radio (SDR) som styres av OpenBTS mens Asterisk programvaren brukes for å rute samtalene internt i cellen og ut til omverdenen, men da som Voice over Internet Protocol (VoIP). MSene blir representert som SIP-enheter ovenfor omverden. Session Initiation Protocol (SIP) benyttes normalt av kontortelefoner og vil ofte oppfattes som en vanlig telefon for brukeren. SIP-enheter kan kobles til et lokalt nettverk og administreres av for eksempel Asterisk.

Kombinasjonen av GSM-nettverk og VoIP kan danne grunnlaget for en ny type mobil-nettverk som kan rulles ut raskere og driftes med betydelig lavere kostnader enn med eksisterende teknologi i områder med nybygging i utviklingsland. Tanken er at dette systemet skal være kompatibelt med de fleste telefonene som allerede er i markedet til kun en tidel av kostnadene og ha en mye lavere kompleksitet enn konvensjonelle GSM-systemer. Dette gjør at denne teknologien også kan benyttes i private nettverk [BTSb].

5.1 OpenBTS

OpenBTS er en åpen kildekode Unix-applikasjon som bruker RF-maskinvaren USRP til å danne et GSM-nettverk for standard GSM mobiltelefoner. GNU Radio og USRP utgjør en programvaredefinert radio (SDR) som styres av OpenBTS. Konfigurasjonsfilen 'OpenBTS.conf' inneholder informasjon som lastes ved oppstart av OpenBTS og benyttes til oppsettet av SDR. 'OpenBTS.conf' inneholder sentrale parametre som maksimal og minimal sendestyrke, nettverk og celle identifikasjon, maksimal rekkevidde og hvilket frekvensbånd og radiokanal (ARFCN) som skal be-



Figur 5.1: Skjema over et konvensjonelt GSM-nettverk og programvare GSM-nettverk. I øverste del av figuren er en forenklet utgave av GSM-nettverket som er illustrert i figur 3.1. Nedre del av figuren gir et overblikk over hvordan de ulike applikasjonene utgjør et programvare GSM-nettverk. Nettverkenes tilkobling til omverden er illustrert til venstre i figuren.

nyttes. Disse parametrene er relatert til basestasjonsystemet (BSS) i konvensjonell GSM, men konfigurasjonsfilen inneholder også parametre som er mer relatert til svitsjesystemet. Noen av disse er parametre for å rute samtaler og data til de riktige applikasjonene, innholdet til forskjellige logiske kanaler og hvordan de skal kombineres til fysiske kanaler, teksten i velkomstmeldingen og hva som skal logges. Dette gjør at OpenBTS er med i både BSS og svitsjesystemet som illustrert i figur 5.1. OpenBTS representerer hver MS som en SIP-enhet ovenfor kommunikasjonsserveren i svitsjesystemet, i programvare GSM benyttes programvaren Asterisk som dette.

5.2 Asterisk

Asterisk-programvaren er en hoveddel av svitsjesystemet i programvare GSM, den er åpen kildekode og gratis. Asterisk benyttes av små og store bedrifter, kundesenter, operatører og statlige organisasjoner over hele verden [Ast]. Asterisk benyttes i mange sammenhenger foruten programvare GSM, den forvandler vanlig datamaskiner til kommunikasjonsservere. En kommunikasjonsserver er en datamaskin som styrer eller ruter tale, data eller video mellom brukere, for eksempel ansatte i en bedrift. Eksempler på hva programvaren brukes i er IP PBX-systemer, VoIP gateways og konferanseservere blant annet.

Når Asterisk benyttes i programvare GSM er det to konfigurasjonsfiler som er sentrale, 'sip.conf' og 'extensions.conf'. Hvor 'sip.conf' kan sammenlignes med hjemmelokasjonsdatabasen (HLR) siden den inneholder identifikasjon av brukere og sammenhengen mellom deres IMSI og telefonnummer. Det er innholdet i 'exten-

sions.conf' som bestemmer hva som skal gjøres når et telefonnummer ringes. Denne kan være ganske enkel ved at den telefonen som tilhører det ringte nummeret anropes, men kan også være komplisert. Hva som skjer kan være vesentlig mer komplisert når telefonsvar, flervalgsmenyer og andre alternativer skal implementeres.

5.3 Smqueue

For å kunne støtte tekstmeldinger (SMS) i programvare GSM må det benyttes programvare som håndterer dette. Smqueue er en SMS lagre og videresende stakk som kan benyttes til håntering av SMS. Denne har sin egen konfigurasjonsfil, 'smqueue.conf'. Smqueue er ikke med i illustrasjonen i figur 5.1, men den ville vært koblet til blokkene OpenBTS, LAN og Internett i tillegg til konfigurasjonsfilen sin. SMS håndtering diskuteres ikke noe videre i denne rapporten.

Kapittel 6

Testoppsett

Den fysiske oppkoblingen av USRP når OpenBTS benyttes i programvare GSM blir forklart i dette kapitlet. Videre er oppkoblingen den samme under eksperimenteringen med frekvenshopping (FH) på USRP, men med den ene endringen at det benyttes to antenner, en for RX og en for TX. USRP kobles til datamaskinen via USB, men får strømmen fra en omformer tilkoblet strømnettet.

6.1 Fysisk oppkobling av USRP

Den fysiske oppkoblingen av USRP er forholdsvis enkel, men må like fult være riktig. De delene som skal kobles sammen eller monteres om du vil er hovedkortet til USRP, to identiske datterkort (RFX900) og en antenne (VERT900) i tillegg til et par RF-kabler. Når OpenBTS benyttes sammen med USRP for å danne en GSM-basestasjon (sammen med noe mer programvare som er nærmere forklart i kapittel 5) er det fordelaktig å benytte to TRX-datterkort da dette gir bedre signal-støy-forhold. Disse monteres i sokler på hovedkortet til USRP, ett som kort A og ett som kort B. Hvert datterkort har en TX/RX-tilkobling og en RX2-tilkobling. Oppkoblingen er vist i figur 2.1. Kort A monteres i sokkel TXA og RXA og benyttes som RX av OpenBTS, antennen kobles til RX2-tilkoblingen på datterkortet ved hjelp av en RF-kabel og antennen plukker opp RF-signalet. Datterkortet nedkonverterer RF-signalet til et IF-signal som blir levert til hovedkortet via sokkel RXA, sokkel TXA benyttes ikke. Kort B monteres i sokkel TXB og RXB og benyttes som TX av OpenBTS, en RF-kabel kobles til TX/RX-tilkoblingen på datterkortet. Det anbefales ikke å koble til en antenne på dette datterkortet. Dette er fordi da vil signalet fra senderen gi økt støy på mottakeren og signal lekkasjen fra selve datterkortet (uten antenne) gir rekkevidde som er tilstrekkelig for kort-avstands testing, mottaker rekkevidden vil avta om en antenne monteres [BTSa]. OpenBTS leverer punktprøvene av signalet som skal sendes til USRP via USB, på hovedkortet blir punktprøvene omgjort til et IF-signal som leveres til datterkortet via sokkel TXB, RXB benyttes ikke. IF-signalet blir oppkonvertert av datterkortet til et RF-signal og sendt ut på TX/RX-tilkoblingen.

6.2 Testmiljø

Testene av programvare GSM ble utført i hjemmekontoret til forfatteren i et landlig område og i kontoret til veilederen ved NTNU Gløshaugen. Eksperimenteringen med frekvenshopping ble utført ved et annet hjemmekontor til forfatteren som ligger mer sentralt i tillegg til de samme stedene som tidligere nevnt. Det ble benyttet to mobiltelefoner under GSM testingen. En eldre enhet, Sony Ericsson K610i, og en nyere enhet, Samsung Galaxy Spica i5700.

Kapittel 7

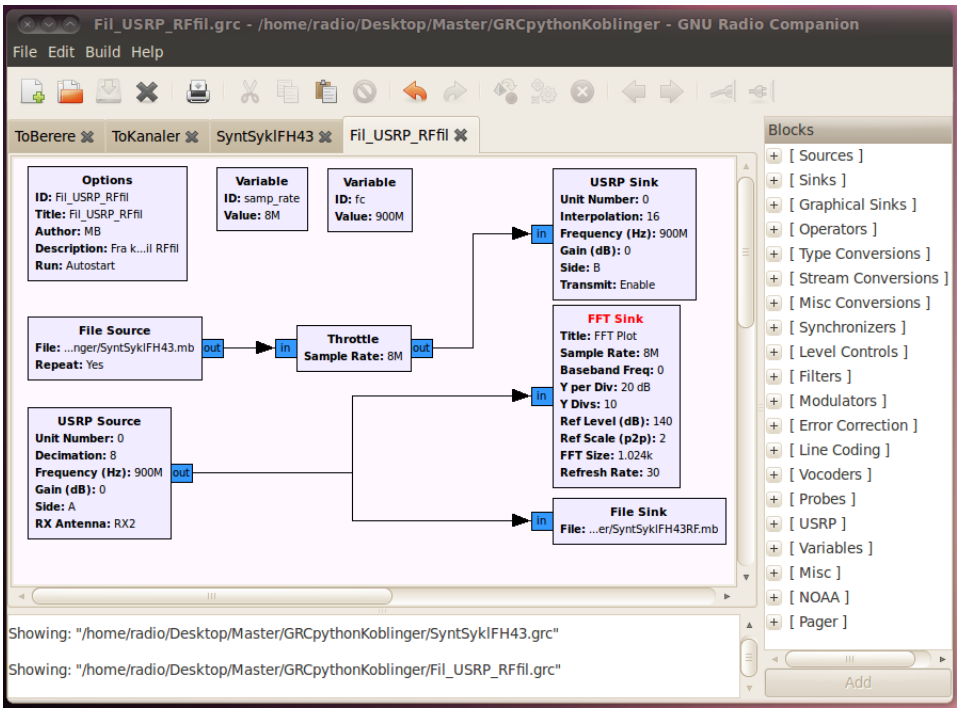
Frekvenshoppingekspriment

I dette kapittelet blir resultater for eksperimentene presenteres og fremgangsmåten blir beskrevet. Resultatene blir også kommentert.

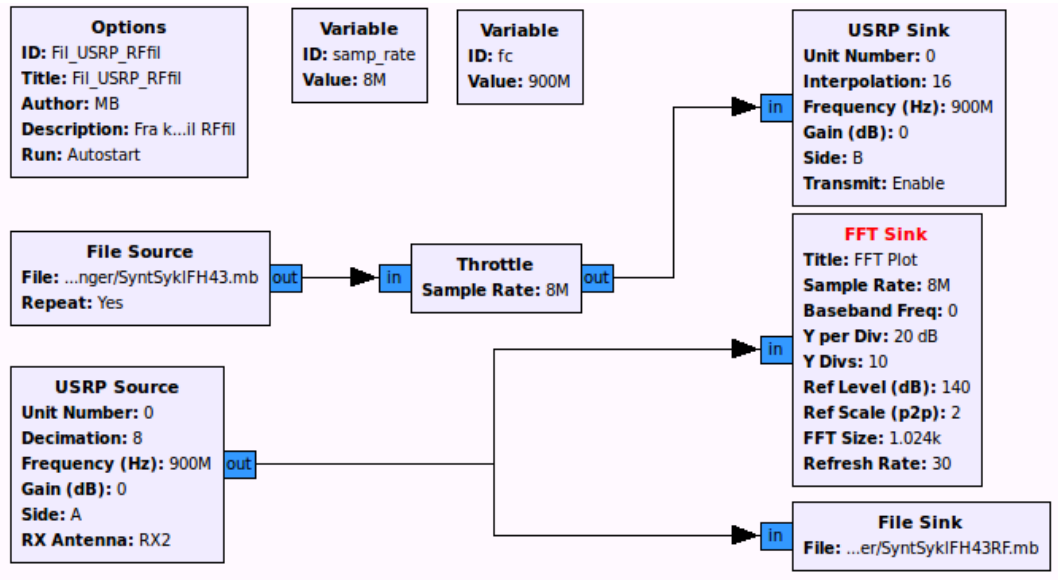
7.1 Fremgangsmåte

Det var GNU Radio og USRP med to RFX900 datterkort som ble benyttet. Oppkoblingene for sender (TX) og mottaker (RX) var lik som for oppsettet når OpenBTS benyttes, med en endring. OpenBTS-oppsettet er beskrevet i kapittel 6.1. Endringen var at det ble montert en VERT900 antenne for TX også, så to antenner ble benyttet. GNU Radio sitt grafisk utviklingsgrensesnitt som heter GNU Radio Companion (GRC) ble benyttet til å utvikle flytdiagrammene som ga grunnlaget for pythonkoden til eksperimentskriptene. Brukergrensesnittet til GRC er vist i figur 7.1. Pythonkoden genereres av GRC (ved å trykke på F5) og pythonskriptet kan kjøres direkte fra GRC (ved å trykke F6). Skriptet stoppes ved hjelp av en rød knapp med et hvitt kryss på eller ved å trykke F7.

Det er tre scenarioer som ble testet. Disse er scenarioer hvor to bærebølger, to kanaler og syntetisert syklisk frekvenshopping ble sendt og mottatt. I tillegg ble støynivået dokumentert. Det ble først generert kildefiler med signalet som skulle sendes for hver av de tre scenarioene før de ble benyttet i flytdiagrammet 'Fil_USRP_RFfil' som er vist i figur 7.2. Flytdiagrammet 'Fil_USRP_RFfil' beskriver sender og mottaker pythonskriptet som ble benyttet i eksprementeringen. Valgt signalkildefil leses og sendes via USRP. USRP mottar så RF-signalet via antennen og lagrer dette i en RF-fil, for videre analyse. Throttle-blokken i hvert flytdiagram sikrer at punktprøveraten respekteres.



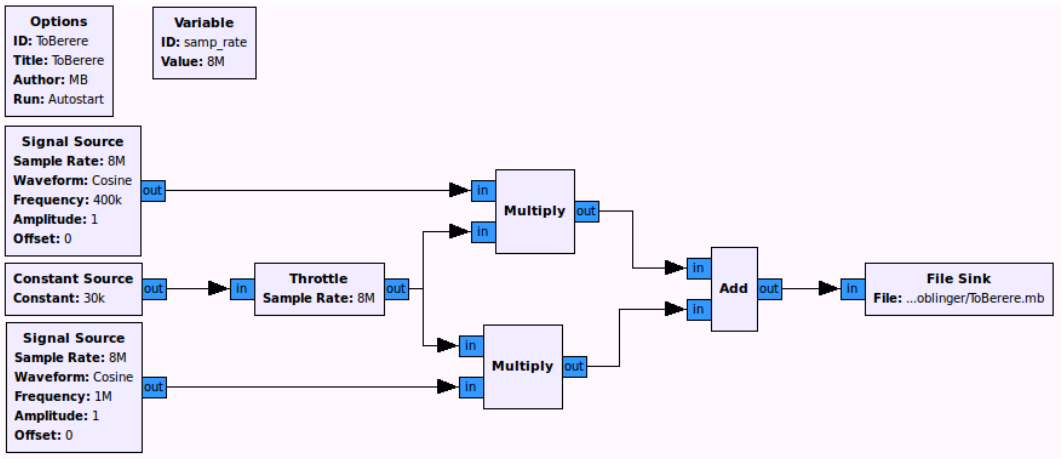
Figur 7.1: Figur av GNU Radio sitt grafisk utviklingsgrensesnitt (GRC, GNU Radio Companion).



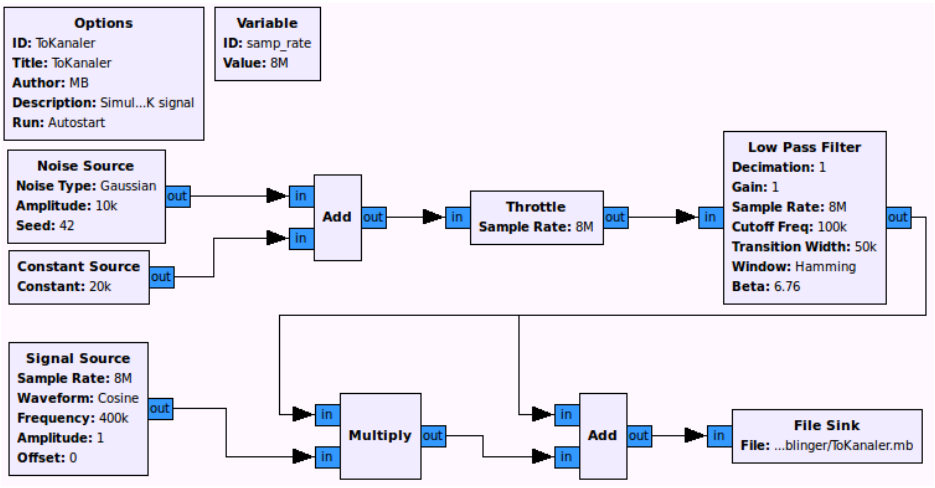
Figur 7.2: Flytdiagrammet 'Fil_USRP_RFfil' beskriver sender og mottaker python-skriptet. Skriptet ble benyttet til eksprementering. Signalkildefilen leses og sendes via USRP i øvre del, mens USRP mottar RF-signalet via antennen og lagrer dette i en RF-fil, for videre analyse, i nedre del av skjemaet. FFT-blokken genererte frekvensspekterene som ble dokumentert.

7.2 Flytdiagrammer for signalkildene

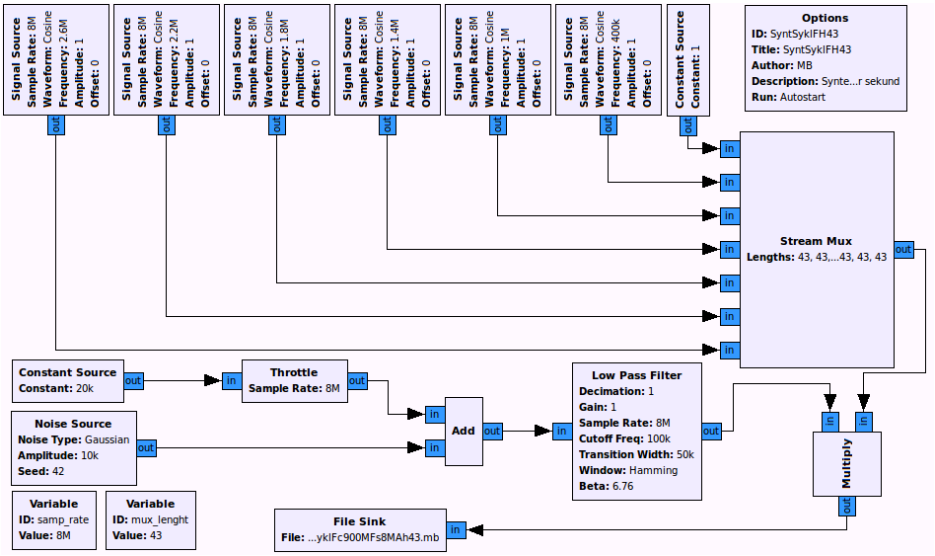
Flytdiagrammer for pythonskriptene som genererer signalkildefilene er vist i figurene 7.3, 7.4 og 7.5. Signalkildefilgenereringen i hvert flytdiagram er forklart nærmere i figurtekstene til de respektive flytdiagrammene. Alle signalkildene må ta hensyn til at maksimal amplitude som kan sendes via USRP er 32 767. Skriptene ble benyttet til å generere signalkildefiler for videre eksperimentering. Dette for å kunne sammenligne figurene bedre.



Figur 7.3: Flytdiagrammet til scenarioet to bærebølger, 'ToBerere'. Flytdiagrammet beskriver signalkildefilgenererings pythonskriptet for to bærebølger. Signalkildefilen 'ToBerere.mb' genereres ved at to cosinussignaler på henholdsvis 400 kHz og 1.0 MHz multipliseres med en konstant med verdi 30 000 og så adderes sammen.



Figur 7.4: Flytdiagrammet til scenarioet to kanaler, 'ToKanaler'. Flytdiagrammet beskriver signalkildefilgenererings pythonskriptet for to radiokanaler (ARFCN). Signalkildefilen 'ToKanaler.mb' genereres ved at Gaussisk støy og en konstant blir addert sammen og filtrert til å ha en båndbredde på omtrent 200 kHz. Siden blir det filtrerte støysignalet multiplisert med et cosinussignal på 400 kHz og addert med seg selv.



Figur 7.5: Flytdiagrammet til scenarioet syntetisert syklisk frekvenshopping, 'SyntSyklFH43'. Flytdiagrammet beskriver signalkildefilgenererings pythonskriptet for syntetisert syklisk frekvenshopping. Signalkildefilen 'SyntSyklFH43.mb' genereres ved at en konstant og flere cosinussignaler på henholdsvis 400 kHz, 1.0 MHz, 1.4 MHz, 1.8 MHz, 2.2 MHz og 2.6 MHz multiplekses og multipliseres med et filtrerte støysignal. Her hoppes det for hver 43. punktprøve, dette er omtrent 100 ganger raskere enn det GSM benytter. Støysignalet består av Gaussisk støy og en konstant som blir addert sammen og filtrert til å ha en båndbredde på omtrent 200 kHz.

7.3 Signalanalyse

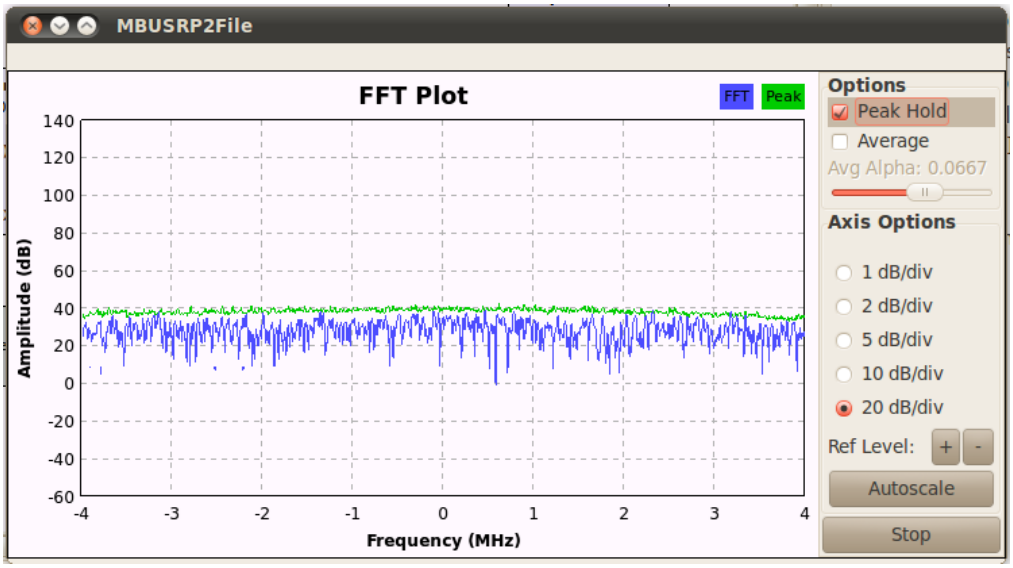
På de neste sidene blir frekvensspekteret og tid-frekvens-analysen til signalskildefilene og mottatt RF-signal for to bærebølger, to kanaler og syntetisert syklisk frekvenshopping presentert, mottatt RF-støy i et tidspunkt under eksperimenteringen presenteres også. Frekvensspekteret til et signal vises i en figur og i neste figur er tid-frekvens-analysen til samme signal vist. Aksene på alle frekvensspektrene er like slik at de kan sammenlignes direkte. I tid-frekvens-analysene er amplituden indikert med farge, mørk blå er laveste mens rød er høyeste. Til analysen er det brukt en tid-frekvens-verktøykasse fra [TFT]. Denne normaliserer amplituden så fargetonene kan ikke direkte sammenlignes mellom de forskjellige figurene. Frekvensen blir også normalisert, 0.5 Hz i figurene tilsvarer 400 kHz. Analysen er gjort på et utsnitt av signalene. Utsnittet er på 1024 punktprøver, foruten valg av utsnitt er resterende parameter like. Tid-frekvens-verktøykasse er en MATLAB-verktøykasse. Det er brukt Octave til å utføre analysen. Octave er en MATLAB-kompatibel programvare og kan benytte seg av filer skrevet for MATLAB.

Først blir frekvensspekteret og tid-frekvens-analysen til mottatt RF-støysignalet i et tidspunkt under eksperimenteringen presentert i figur 7.6 og 7.7. Støyen ser ut til å være fordelt over hele spekteret og har ingen rentoner. Amplituden ligger omtrent på 40 dB. Videre blir frekvensspekteret og tid-frekvens-analysen til både sendt og mottatt signal til de tre scenarioene presentert, da i fire figurer for hvert scenario.

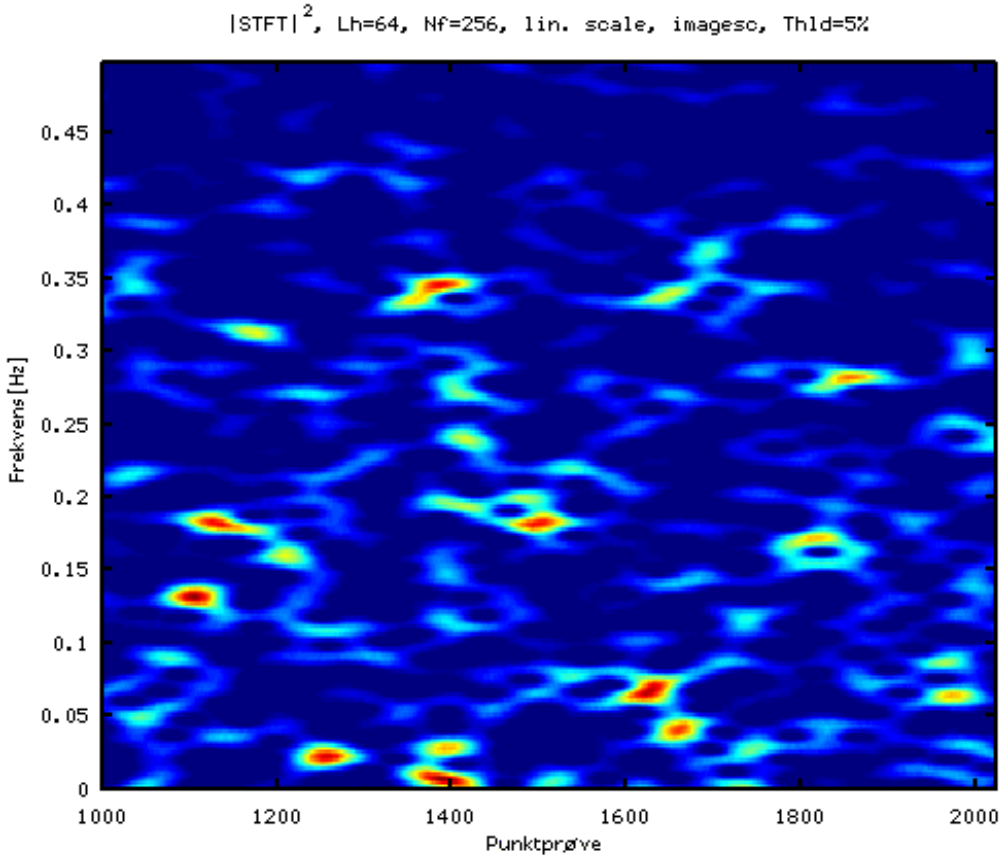
To bærebølger presenteres i figur 7.8, 7.9, 7.10 og 7.11. Kilden er fint og klart representert. Mottaket av RF-signalet er ikke så bra som forventet, men begge bærebølgene kan finnes i figur 7.11.

To kanaler presenteres i figur 7.12, 7.13, 7.14 og 7.15. Kilden er også her klart representert som forventet. Mottaket av RF-signalet er omtrent på likt nivå som for to bærebølger, men det kan tyde på at det er noe bedre. Begge kanalene kan finnes i figur 7.15, men frekvensen til en av kanalene er på feil plass, den er på 800 kHz (0.1) mot 400 kHz (0.05) som forventet. Interpoleringsverdiene og desimeringsverdiene er like for to bærebølger og to kanaler, som de er for de andre og forsåvidt. Dette finner jeg ikke noen forklaring på.

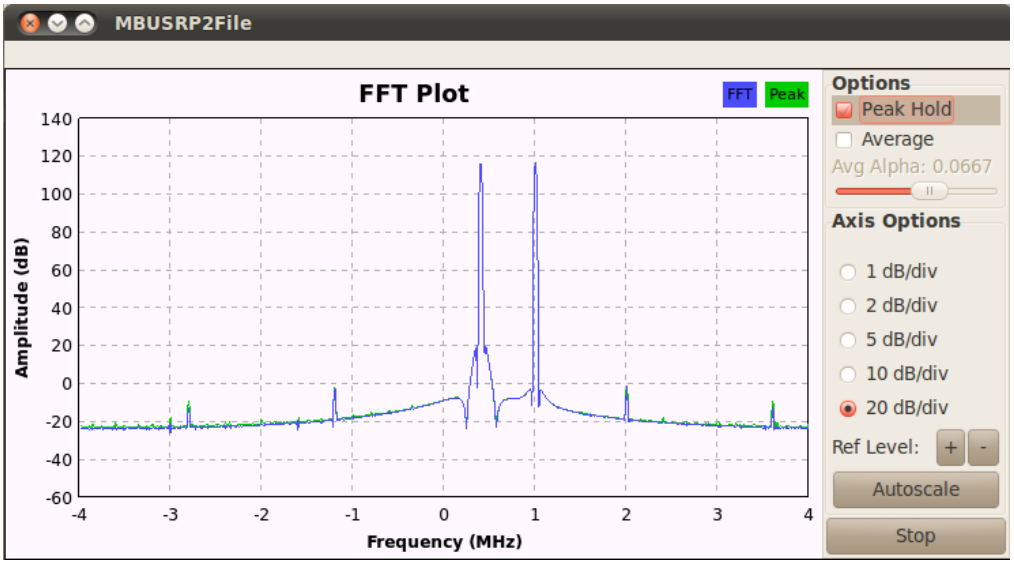
Sist presenteres syntetisert syklisk frekvenshopping i figur 7.16, 7.17, 7.18 og 7.19. Ser tydelig at det hoppes syklisk gjennom radiokanalene, og den forventede større avstanden mellom 2. og 3. radiokanal er tydelig i kilden. Like tydelig i mottatt RF-signal er det ikke, men det kan like vel tydes at det hoppes i det samme mønsteret.



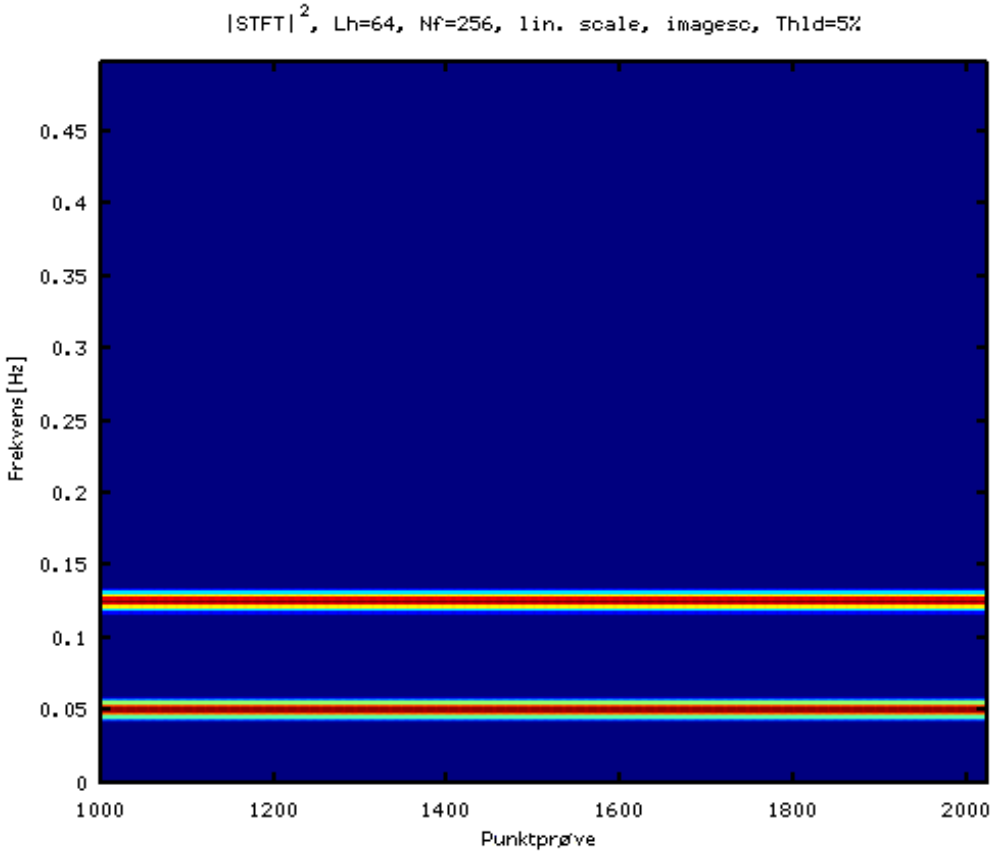
Figur 7.6: Frekvensspekteret til støyen i et tidspunkt under eksperimenteringen.



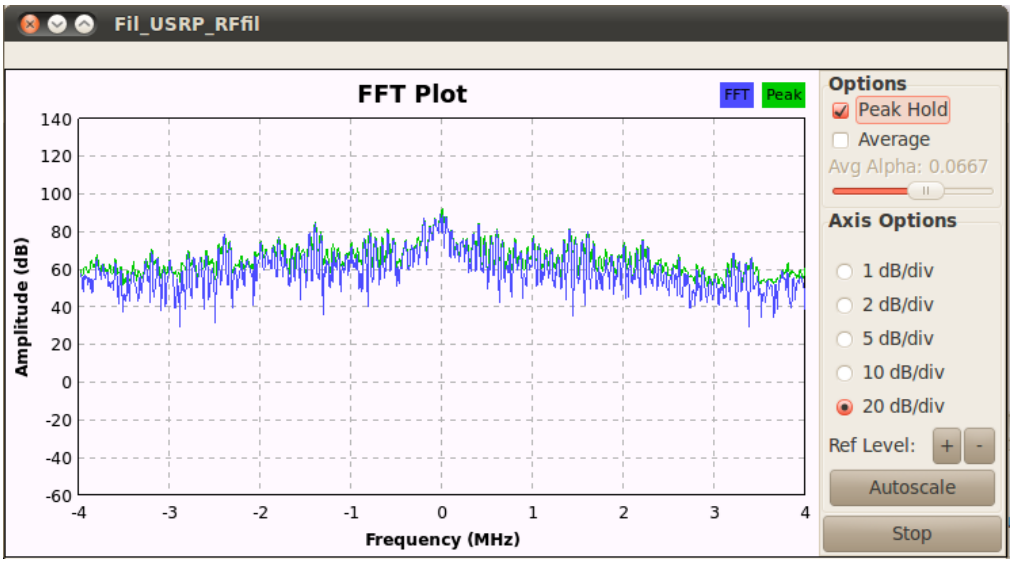
Figur 7.7: Tid-frekvens-analysen til støyen i et tidspunkt under eksperimenteringen. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



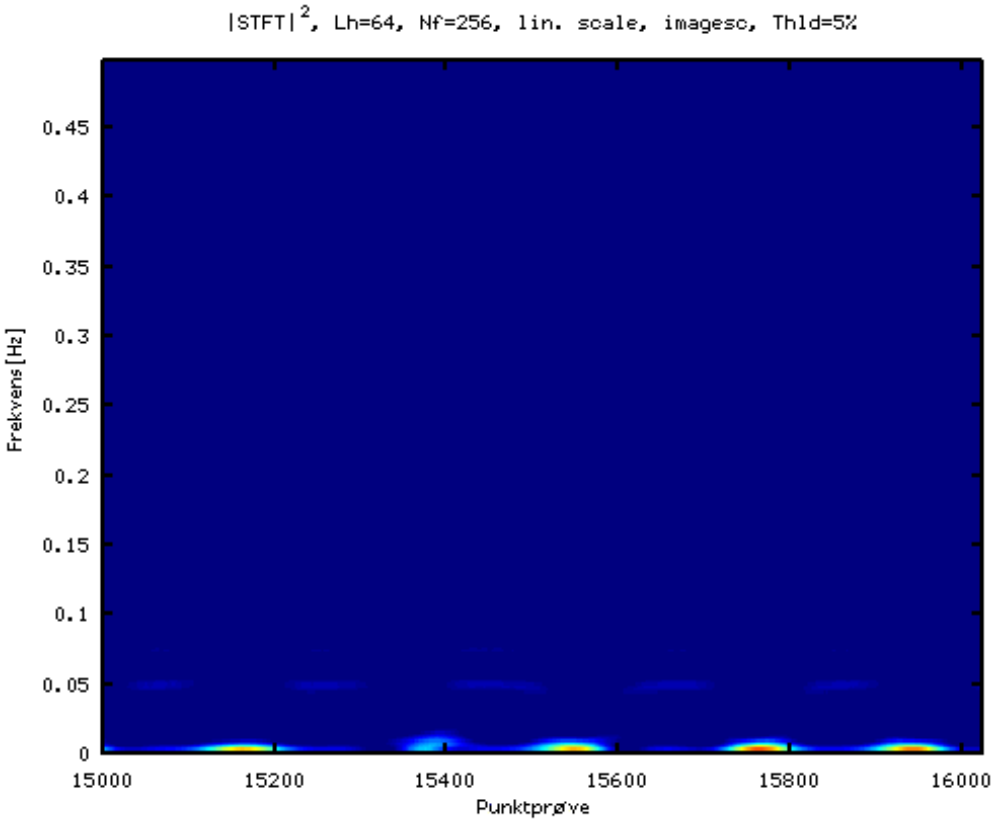
Figur 7.8: Frekvensspekteret til signalkilden for to bærebølger.



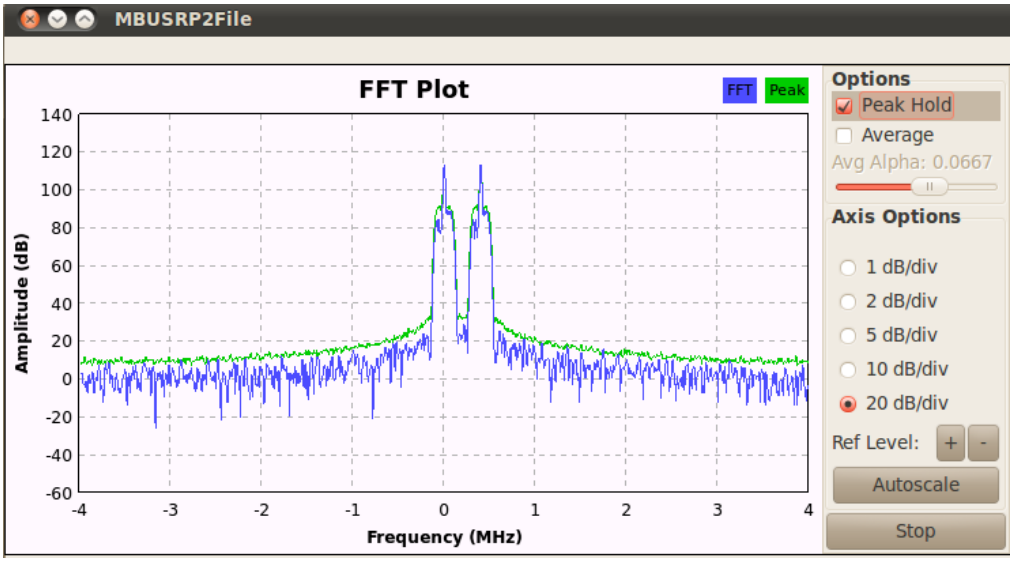
Figur 7.9: Tid-frekvens-analysen til signalkilden for to bærebølger. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



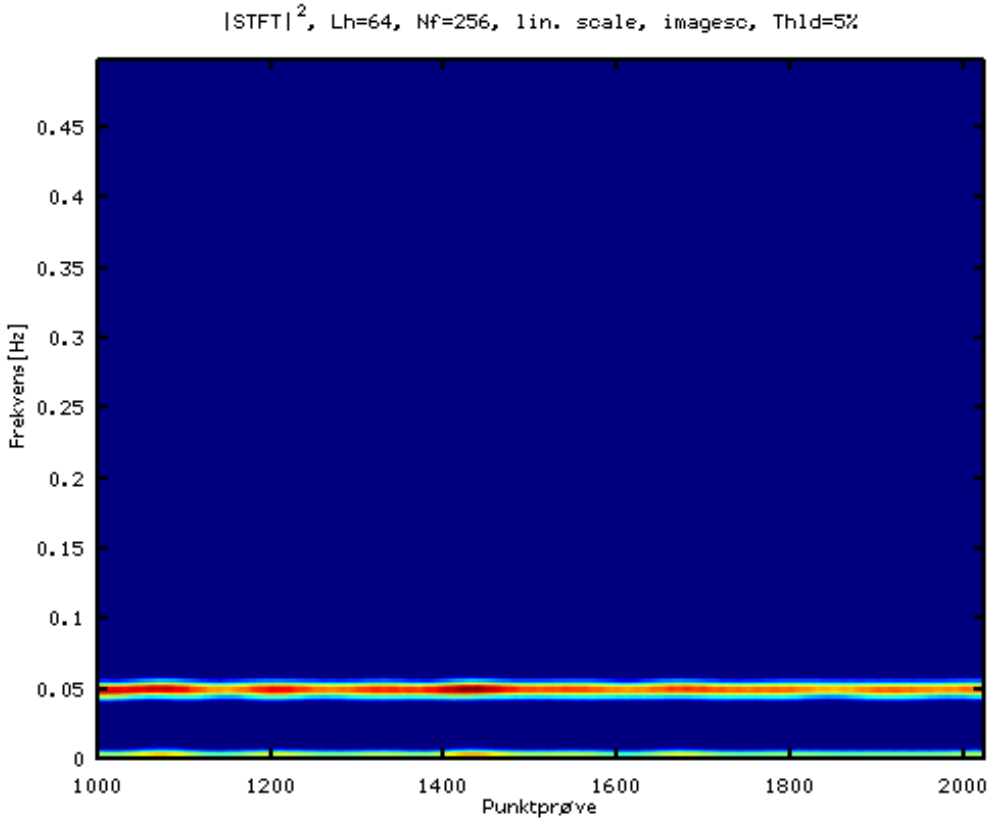
Figur 7.10: Frekvensspekteret til RF-signalet mottatt av USRP for to bærebølger.



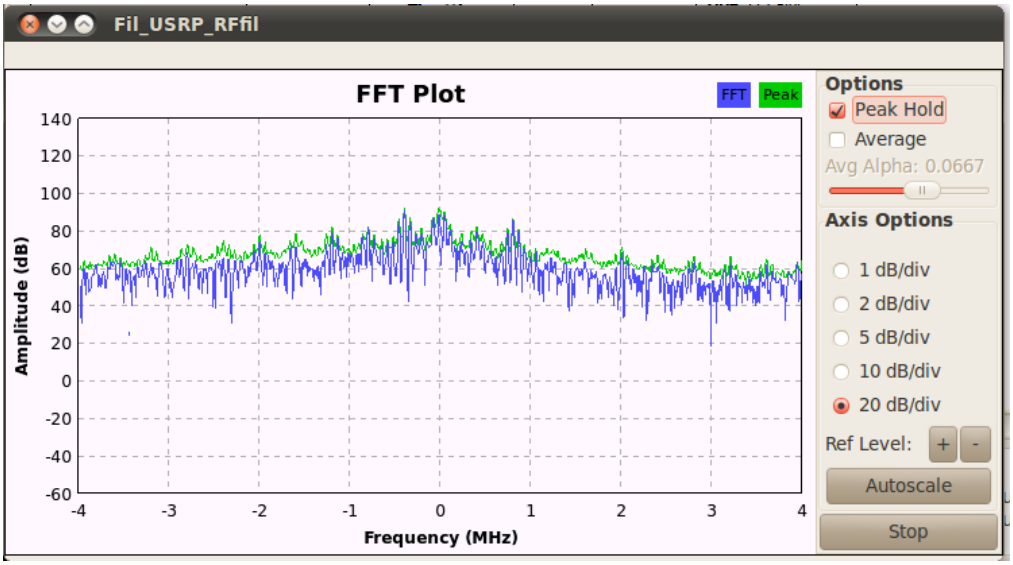
Figur 7.11: Tid-frekvens-analysen til RF-signalet mottatt av USRP for to bærebølger. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



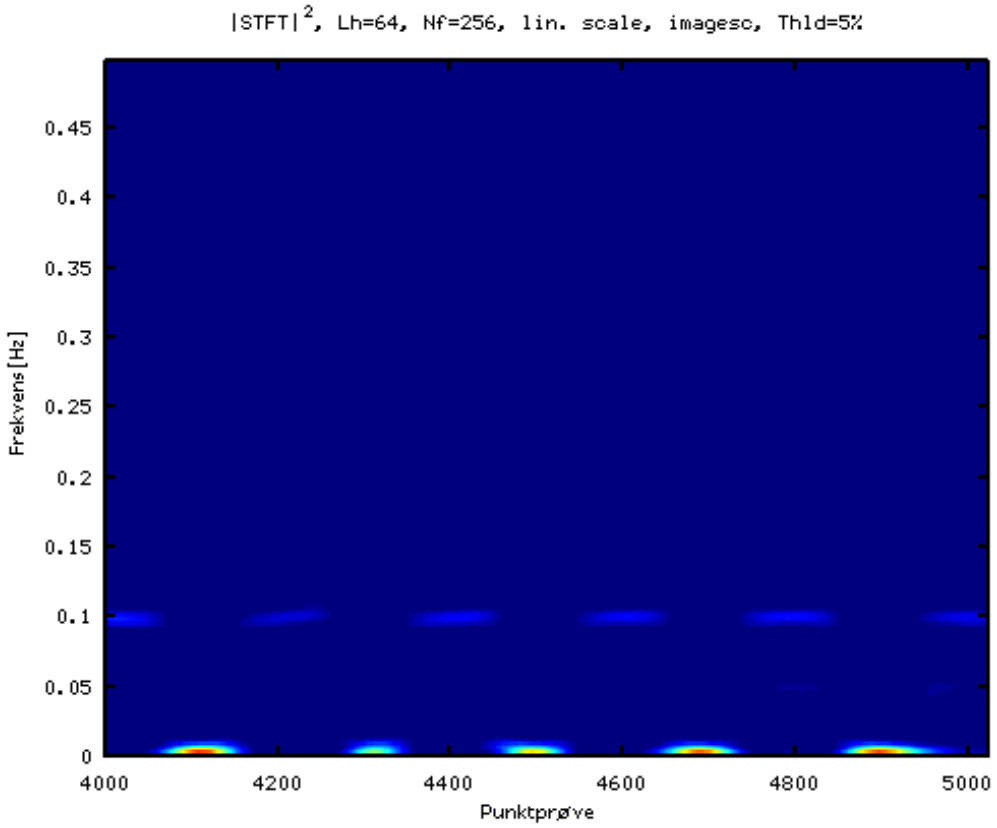
Figur 7.12: Frekvensspekteret til signalkilden for to kanaler.



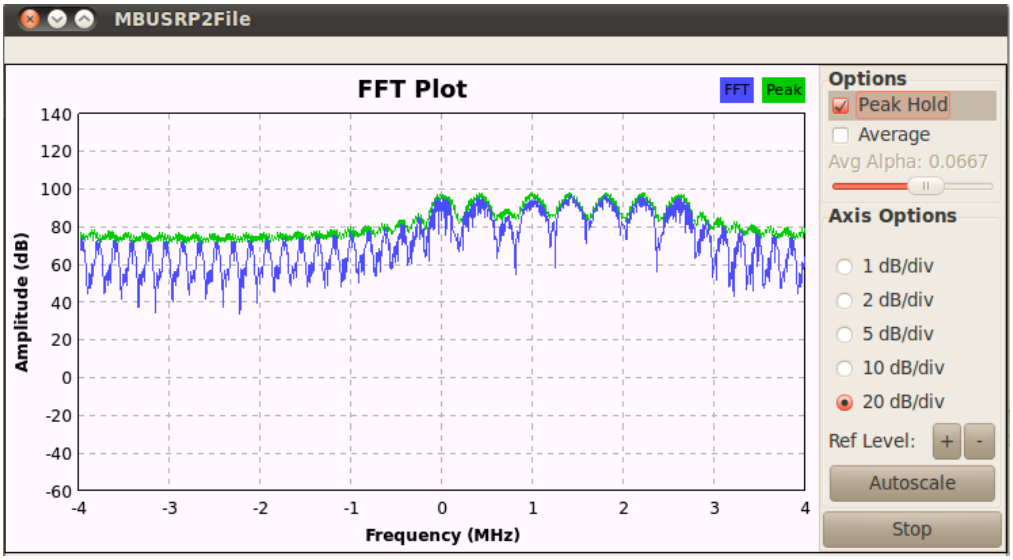
Figur 7.13: Tid-frekvens-analysen til signalkilden for to kanaler. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



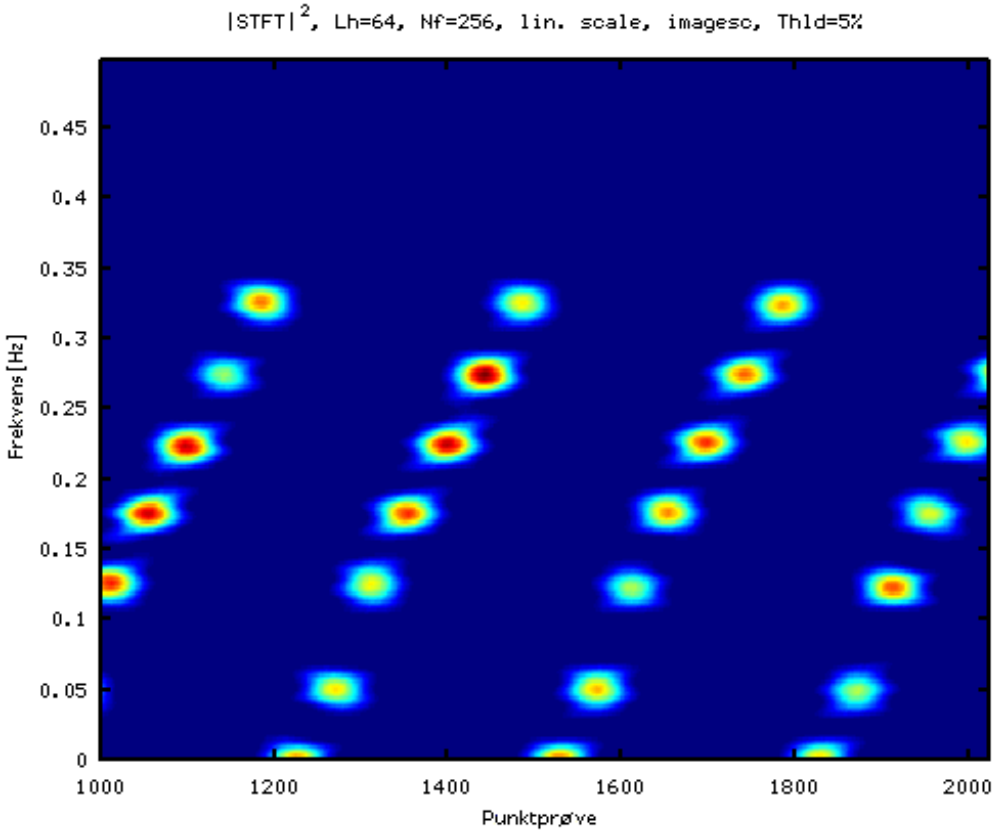
Figur 7.14: Frekvensspekteret til RF-signalet mottatt av USRP for to kanaler.



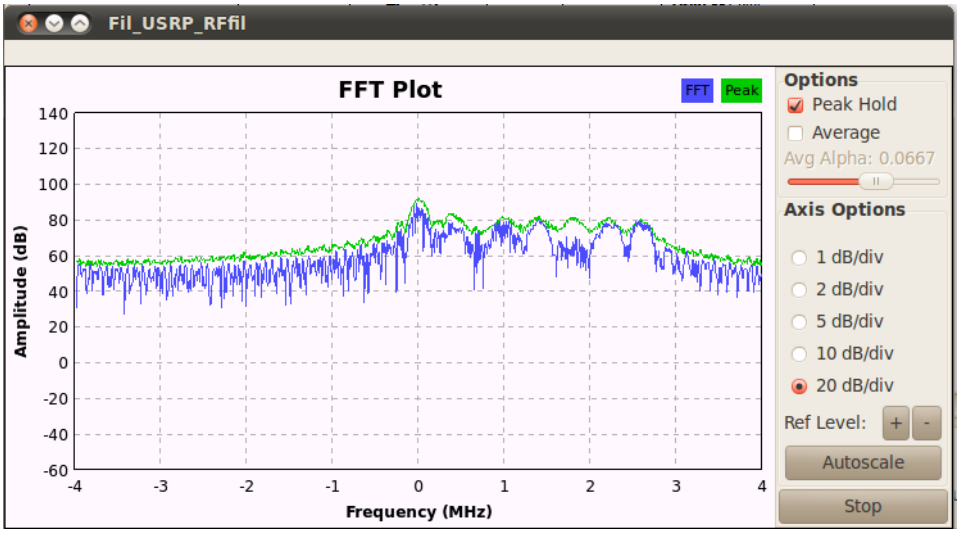
Figur 7.15: Tid-frekvens-analysen til RF-signalet mottatt av USRP for to kanaler. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



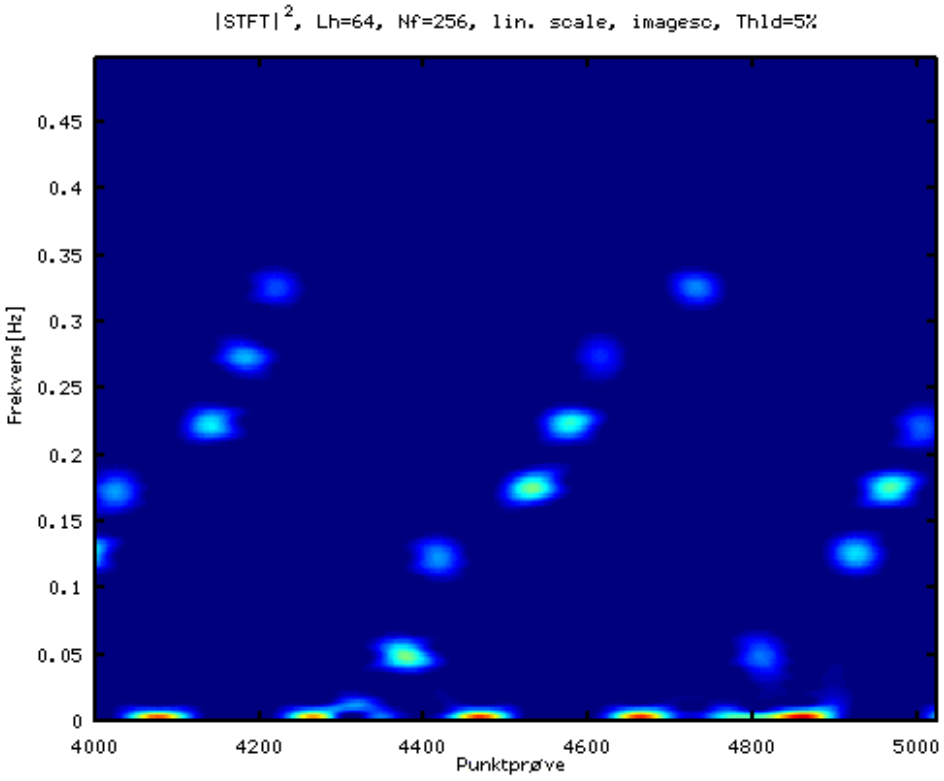
Figur 7.16: Frekvensspekteret til signalkilden for syntetisert syklisk frekvenshopping.



Figur 7.17: Tid-frekvens-analysen til signalkilden for syntetisert syklisk frekvenshopping. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.



Figur 7.18: Frekvensspekteret til RF-signalet mottatt av USRP for syntetisert syklisk frekvenshopping.



Figur 7.19: Tid-frekvens-analysen til RF-signalet mottatt av USRP for syntetisert syklisk frekvenshopping. Amplitude og frekvens blir normalisert. 0.5 Hz tilsvarer 4 MHz, altså $f_s/2$.

Kapittel 8

Konklusjon

En programvaredefinert radio som GNU Radio og USRP utgjør og som OpenBTS styrer er en god måte å realisere et enkelt GSM-nettverk på. Måten frekvenshopping skal implementeres i GSM er klar i rekomandasjonene fra European Telecommunications Standards Institute (ETSI). Selv om ikke resultatene fra eksperimentene er entydig på at frekvenshopping vil fungere må de tolkes som at om implementasjonen gjøres på en god måte er det mulig å få frekvenshopping til å fungere. Implementasjonen av frekvenshopping i OpenBTS med USRP som RF-maskinvare er en programeringsjobb hvor utvikleren må sette seg inn i et komplisert program med betydelig mengde kode. God kunnskap innen C++ og Python vil være fordelaktig.

Bibliografi

[Ast] Asterisk. `http://www.asterisk.org/`.

[Bju95] Vidar Bjugan. *GSM Det globale systemet for mobilkommunikasjon.* Tapir Forlag, Trondheim, 1995.

[BTSa] OpenBTS Desktop Test Kit. `http://gnuradio.org/redmine/wiki/gnuradio/OpenBTSDesktopTestingKit`.

[BTSb] OpenBTS Wiki Start. `http://gnuradio.org/redmine/wiki/gnuradio/OpenBTS`.

[Dat] Integrated Synthesizer and VCO ADF4360-3. `http://www.analog.com/static/imported-files/data_sheets/ADF4360-3.pdf`.

[Dav] David A. Burgess, Harvind S. Samra. The Open BTS Project. `http://www.ahzf.de/itstuff/papers/OpenBTSProject.pdf` Følger også med kildekoden.

[ETSa] ETSI. Digital cellular telekommunications system (Phase 2+); Abbreviations and acronyms (GSM 01.04 version 8.0.0 release 1999. `http://pda.etsi.org/pda/queryform.asp`.

[ETSb] ETSI. Digital cellular telekommunications system (Phase 2+); Mobile Station - Base Station System (MS - BSS) interface; Channel structures and access capabilities (GSM 04.03 version 7.0.0 Release 1998. `http://pda.etsi.org/pda/queryform.asp`.

[ETSc] ETSI. Digital cellular telekommunications system (Phase 2+); Multiplexing and multiple access on the radio path (GSM 05.02 version 8.5.1 release 1999. `http://pda.etsi.org/pda/queryform.asp`.

[ETSd] ETSI. Digital cellular telekommunications system (Phase 2+); Numbering, adressing and identification (GSM 03.03 version 5.4.1 release 1996. `http://pda.etsi.org/pda/queryform.asp`.

[ETSe] ETSI. Digital cellular telekommunications system (Phase 2+); Physical layer on the radio path; General description (GSM 05.01 version 6.1.1 Release 1997. `http://pda.etsi.org/pda/queryform.asp`.

[ETSf] ETSI. Digital cellular telekommunications system (Phase 2+); Radio transmission and reception (GSM 05.05 version 8.5.1 release 1999. http://pda.etsi.org/pda/queryform.asp.

[Etta] Skjema over RFX900. http://code.ettus.com/redmine/ettus/attachments/10/rfx900.pdf.

[Ettb] Skjema over USRP. http://code.ettus.com/redmine/ettus/attachments/11/usrp1.pdf.

[Ettc] Universal Software Radio Peripheral - The Foundation for Complete Software Radio Systems. http://www.ettus.com/downloads/ettus_ds_usrp_v7.pdf.

[Ettd] USRP User's and Developer's Guide. http://www.olifantasia.com/gnuradio/usrp/files/usrp_guide.pdf.

[GNU] GNU Radio. http://gnuradio.org/redmine/wiki/gnuradio/WikiStart.

[Hay01] Simon Haykin. *Communication System 4th Edition.* John Wilev & Sons, Inc, 2001.

[Jea] Jean-Marie Bourjolly, Souheyl Touhami, Leslie Déjoie, Ke Ding, Oumar Dioume, Michel Lominy. Optimizating Frequency Hopping in GSM Cellular Phone Networks. Levert av veileder.

[Mai] openbts-discuss. http://sourceforge.net/mailarchive/forum.php?forum_name=openbts-discuss Tråder: How to support multiple ARFCN in OpenBTS?

[MG10] K. Hove og E. Hvideberg M. Glendrange. Decoding GSM. *NTNU, Institutt for telematikk*, 2010.

[TFT] the Time-Frequency Toolbox. http://tftb.nongnu.org/.

[Tho09] Einar Thorsrud. Programvaredefinert radio, Mulige hyllevareløsninger for DSRC-anvendelser. *NTNU*, 2009.

Tillegg A

Opp- og nedkonverteringen mellom IF og RF

I tillegg B.2 presenteres et skjema over hvilke komponenter som utgjør datterkortet RFX900 og deres sammenkoblinger. Skjemaet av RFX900 er hentet fra [Etta]. Komponentene U103 og U3, ADF4360-3, utgjør de to spenningskontrollerte oscillatorene (VCO), en for TX og en for RX, som er sentrale i oppkonvertering og nedkonvertering av RF-signalene som skal sendes og mottas. Signalet som skal sendes kommer til datterkortet fra hovedkortet til USRP som et IF-signal og må derfor oppkonverteres til et RF-signal med høyere frekvens som sendes på lufta. Selve oppkonverteringen utføres av mikseren U101, AD834X, som tar inn I og Q signalene fra hovedkortet og f_c som leveres av VCOen U103 på datterkortet, f_c vil være bærebølgefrekvensen til RF-signalet ut av mikseren. Tilsvarende er signalet som mottas fra lufta et høyfrekvent RF-signal og må derfor nedkonverteres til et IF-signal før det leveres til hovedkortet. Mikseren U2, AD8347, sammen med filtrering tar seg av selve nedkonverteringen av RF-signalet, som mottas fra antennen, ned til et IF-signal og deler det i ett I og ett Q signal som leveres til hovedkortet. Mikseren trenger bærebølgefrekvensen f_c til RF-signalet for å utføre nedkonverteringen og splittingen i I og Q signal, f_c leveres av VCOen U3 på datterkortet.

La oss gå nærmer inn på hvordan f_c settes til ønsket frekvens. f_c vil være ca 900 MHz i GSM900. f_c kontrolleres av referanseklokkefrekvensen, f_{REFin}, inn på REFin og digitale data mottatt via ett SPI kompatibelt 3-tråd grensesnitt som består av CLK, DATA og LE. Referanseklokken, clock_p, kommer inn på datterkortet via koblingen J1 pin 11, for RX, og J102 pin 12, for TX, fra USRP. f_{REFin} bør være mellom 10 MHz og 250 MHz, dermed kan $f_{REFin} = 52$ MHz for GSM900 , og i ett eksempel i databladet er $f_{REFin} = 16$ MHz. De digitale dataene leveres fra USRP. For TX (kobling J102) korresponderer serieklokken på SCLK_TX pin 30 med CLK, selve dataene på SDO_TX pin 34 med DATA og last innhold på SEN_TX pin 36 med LE. For RX (kobling J1) korresponderer serieklokken på SCLK_RX pin 29 med CLK, selve dataene på SDO_RX pin 33 med DATA og last innhold på SEN_RX pin 35 med LE. Dataene gir verdi til en preskalerer (P), tre tellere (A, B og R) og en del kontrollinnstillinger deriblant DIVIDE-BY-2 (DIV2) og DIVIDE-

BY-2-SELECT (DIVSEL), de to tellerene A og B utgjør sammen med P teller N. DIV2 og DIVSEL kontrollerer to forskjellige hendelser som kan se like ut. DIV2 bestemmer om utgangsfrekvensen f_c skal ligge i området $1600 - 1950$ MHz (0) eller $800 - 975$ MHz (1), mens DIVSEL bestemmer om det er f_{VCO} (0) eller $\frac{f_{VCO}}{2}$ (1) som skal deles på N når f'_{PFD} lages. PFD står for Fase frekvens detektor (Phase Frequency Detector) som sammenligner f_{PFD} og f'_{PFD} for å vite om V_{TUNE} skal endres for å få riktig f_c ut.

Formlene er hentet fra databladet til ADF4360-3 side 9 [Dat] og utvidet.

$$f_c = \begin{cases} f_{VCO} & \text{når DIV2} = 0\text{, får da } f_c = 1600 - 1950 \text{ MHz} \\ \frac{f_{VCO}}{2} & \text{når DIV2} = 1\text{, får da } f_c = 800 - 975 \text{ MHz} \end{cases} \tag{A.1}$$

$$f'_{PFD} = \begin{cases} \frac{f_{VCO}}{N} & \text{om DIVSEL} = 0 \\ \frac{f_{VCO}}{2 \cdot N} & \text{om DIVSEL} = 1 \end{cases} \tag{A.2}$$

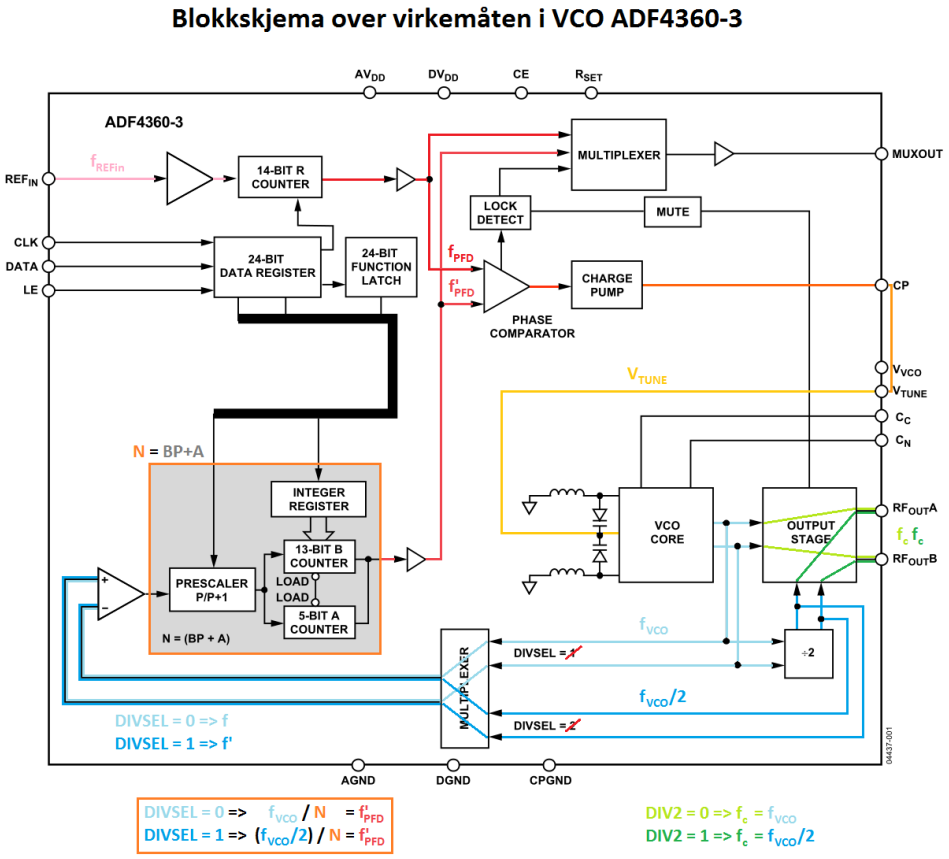
$$f_{PFD} = \frac{f_{REFin}}{R} \tag{A.3}$$

$$f_{VCO} = \frac{[(P \cdot B) + A] \cdot f_{REFin}}{R} = \frac{f_{REFin}}{R} \cdot N \tag{A.4}$$

$$N = BP + A \tag{A.5}$$

Hvor,
f_c er frekvensen til signalet som leveres ut av ADF4360-3
f_{VCO} er frekvensen til signalet ut av VCO kjernen
f'_{PFD} er frekvensen til signalet ut av N telleren
f_{PFD} er frekvensen til signalet ut av R telleren
f_{REFin} er frekvensen til signalet fra referanseklokken inn til ADF4360-3
A er verdien i A telleren, verdiområdet til $A = 0 - 31$
B er verdien i B telleren, verdiområdet til $B = 3 - 8191$
N er verdien som signalet inn i N telleren deles på,
verdiområdet til $N = 2.667 \left(\frac{8 \cdot 3}{9}\right) - 7973.7878 \left(\frac{32 \cdot 8191}{33} + 3\right)$
P er verdien til preskalereren, verdiområdet til $P = \frac{8}{9}, \frac{16}{17}, \frac{32}{33}$
R er verdien i R telleren, verdiområdet til $R = 1 - 16383$

Figur A.1 viser ett skjema over en spenningskontrollert oscillator (VCO) ADF4360-3 som det er plasser to av på datterkortet RFX900. Ett mulig oppsett for GSM900 kan være at $f_{REFin} = 52$ MHz og $R = 260$ da blir $f_{PFD} = 200$ kHz. For å få $f_c = 900$ MHz må DIV2 være satt (DIV2 = 1) og da er målet å få $f_{VCO} = 1800$ MHz som betyr at $V_{TUNE} =$ ca 1.7 V. (Om DIVSET = 0, altså ikke satt, måtte $N = 9000$ som er utenfor N sitt intervall.) Med DIVSET = 1 er det $\frac{f_{VCO}}{2}$ som deles på N og da burde $N = 4500$. Med $P = \frac{8}{9}$, $B = 4000$ og $A = 0$ blir $N = 4500$ og N kan endres med en hel ved hjelp av A mellom hver N som går opp i 9 (eller egentlig $\frac{8}{9}$), slik dekkes alle ARFCNene. (Ner som går opp i 9 er $[..., 4482, 4491, 4500, 4509, 4518, ...]$



Figur A.1: Skjema over en spenningskontrollert oscillator (VCO) ADF4360-3 som er plasser to av på datterkortet RFX900

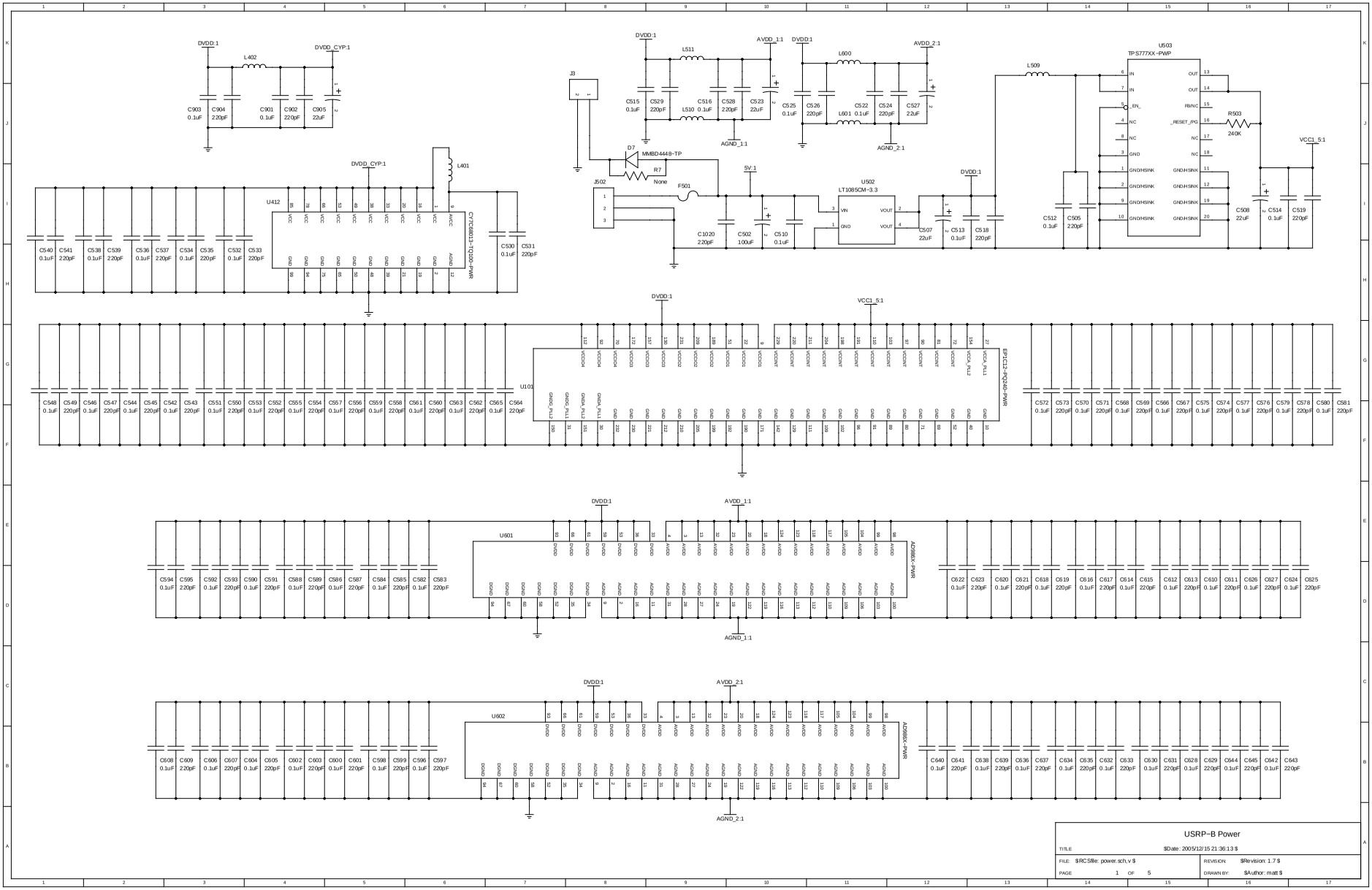
disse gir $B = [..., 3984, 3992, 4000, 4008, 4016, ...]$ og A dekker fint verdiene imellom.) f_c kan altså endres i steg på 200 kHz som er kanalbredden i GSM.

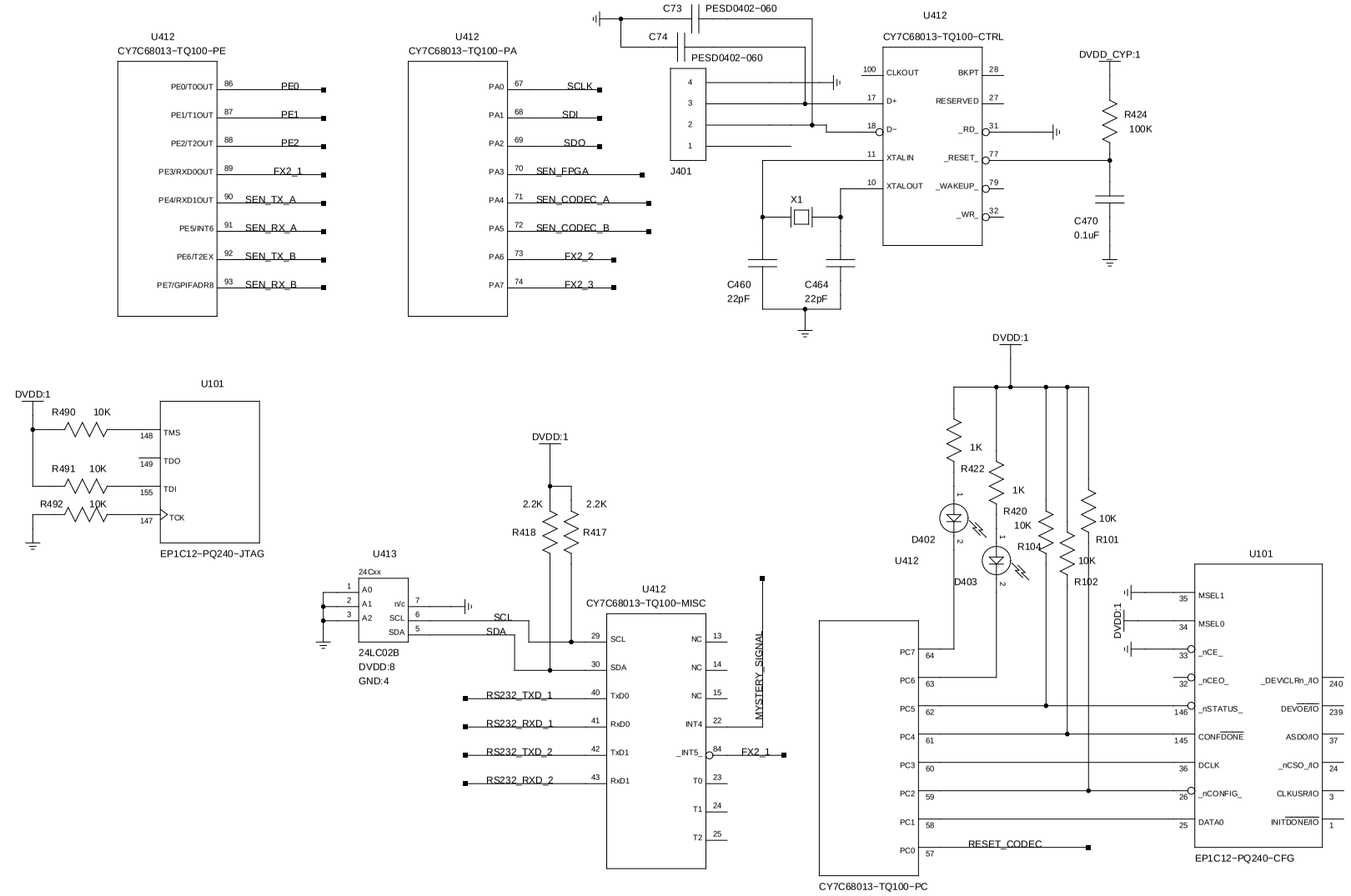
Tillegg B

Skjema

B.1 USRP

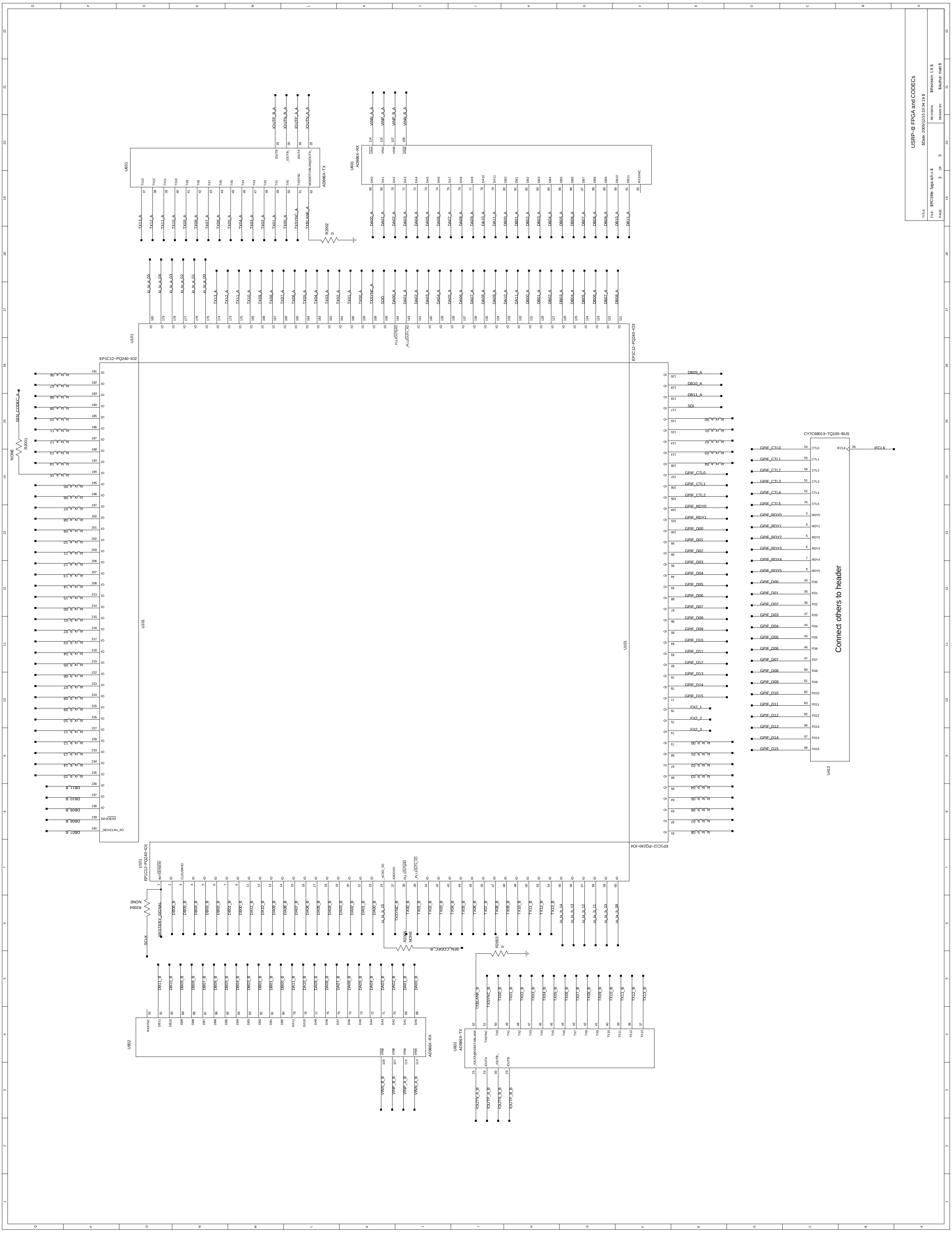
Skjema av USRP er hentet fra [Ettb]


DVDD:1
DVDD_CYP:1
L402
C903 0.1uF
C904 220pF
C901 0.1uF
C902 220pF
C905 22uF
L401
U412
CY7C68013-TQ100-PWR
C540 0.1uF
C541 220pF
C538 0.1uF
C539 220pF
C536 0.1uF
C537 220pF
C534 0.1uF
C535 220pF
C532 0.1uF
C533 220pF
C530 0.1uF
C531 220pF
J3
L511
AVDD_1:1
C515 0.1uF
C529 220pF
L510 0.1uF
C516 0.1uF
C528 220pF
C523 22uF
AGND_1:1
L600
AVDD_2:1
C525 0.1uF
C526 220pF
L601 0.1uF
C522 0.1uF
C524 220pF
C527 22uF
AGND_2:1
D7
MMBD4448-TP
R7
None
J502
F501
5V:1
C1020 220pF
C502 100uF
C510 0.1uF
U502
LT1085CM-3.3
VIN
GND
VOUT
C507 22uF
C513 0.1uF
C518 220pF
L509
U503
TPS777XX-PWP
C512 0.1uF
C505 220pF
R503
240K
C508 22uF
C514 0.1uF
C519 220pF
VCC1_5:1
U101
EP1C12-TQ240-PWR
C548 0.1uF
C549 220pF
C546 0.1uF
C547 220pF
C544 0.1uF
C545 220pF
C542 0.1uF
C543 220pF
C551 0.1uF
C550 220pF
C553 0.1uF
C552 220pF
C555 0.1uF
C554 220pF
C557 0.1uF
C556 220pF
C559 0.1uF
C558 220pF
C561 0.1uF
C560 220pF
C563 0.1uF
C562 220pF
C565 0.1uF
C564 220pF
C572 0.1uF
C573 220pF
C570 0.1uF
C571 220pF
C568 0.1uF
C569 220pF
C566 0.1uF
C567 220pF
C575 0.1uF
C574 220pF
C577 0.1uF
C576 220pF
C579 0.1uF
C578 220pF
C580 0.1uF
C581 220pF
U601
AD9865-PWR
C594 0.1uF
C595 220pF
C592 0.1uF
C593 220pF
C590 0.1uF
C591 220pF
C588 0.1uF
C589 220pF
C586 0.1uF
C587 220pF
C584 0.1uF
C585 220pF
C582 0.1uF
C583 220pF
C622 0.1uF
C623 220pF
C620 0.1uF
C621 220pF
C618 0.1uF
C619 220pF
C616 0.1uF
C617 220pF
C614 0.1uF
C615 220pF
C612 0.1uF
C613 220pF
C610 0.1uF
C611 220pF
C626 0.1uF
C627 220pF
C624 0.1uF
C625 220pF
U602
C608 0.1uF
C609 220pF
C606 0.1uF
C607 220pF
C604 0.1uF
C605 220pF
C602 0.1uF
C603 220pF
C600 0.1uF
C601 220pF
C598 0.1uF
C599 220pF
C596 0.1uF
C597 220pF
C640 0.1uF
C641 220pF
C638 0.1uF
C639 220pF
C636 0.1uF
C637 220pF
C634 0.1uF
C635 220pF
C632 0.1uF
C633 220pF
C630 0.1uF
C631 220pF
C628 0.1uF
C629 220pF
C644 0.1uF
C645 220pF
C642 0.1uF
C643 220pF
USRP−B Power
TITLE
$Date: 2005/12/15 21:36:13 $
FILE: $RCSfile: power.sch,v $
REVISION: $Revision: 1.7 $
PAGE 1 OF 5
DRAWN BY: $Author: matt $

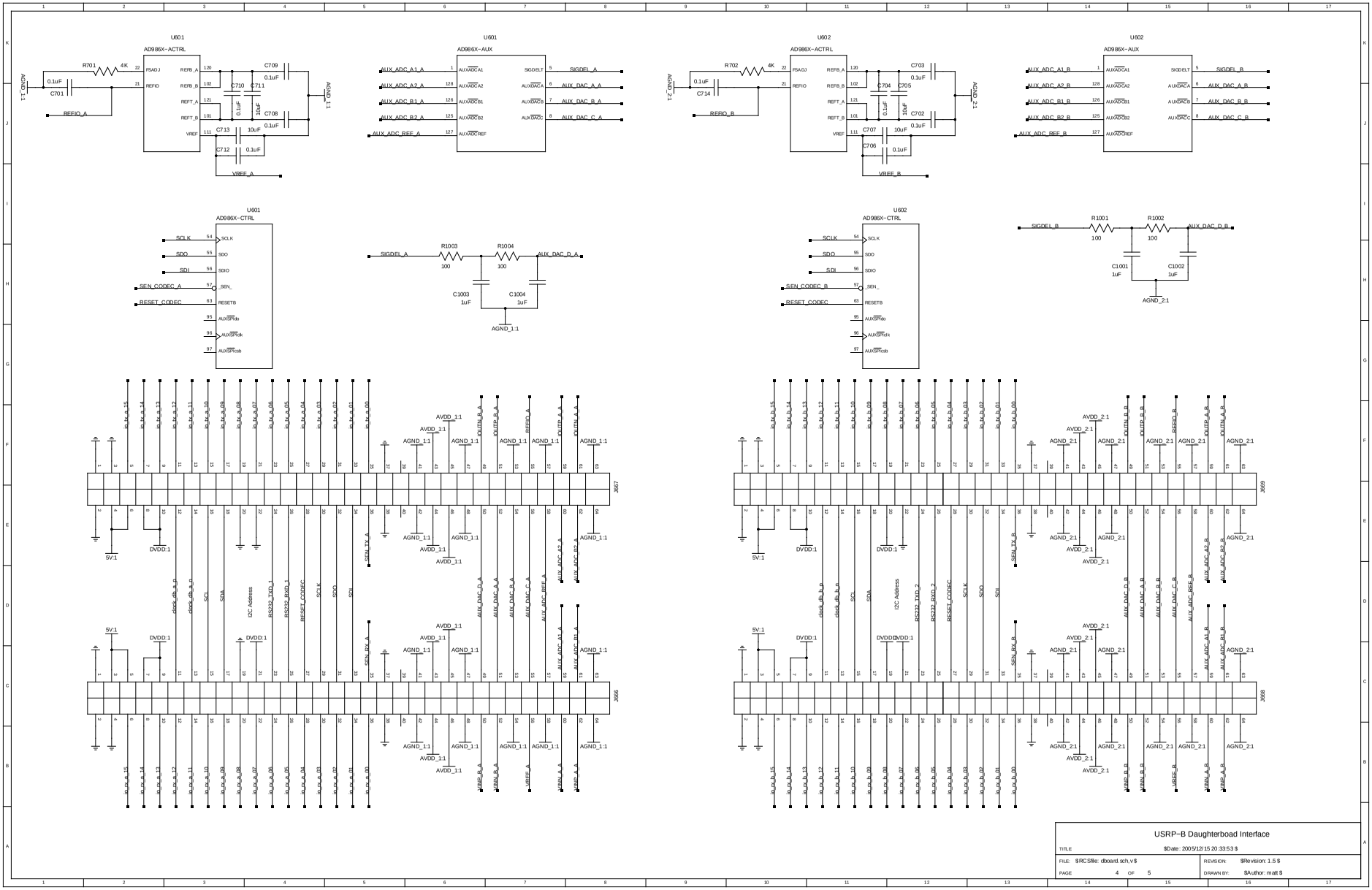


USRP-B FPGA/FX2 Interface

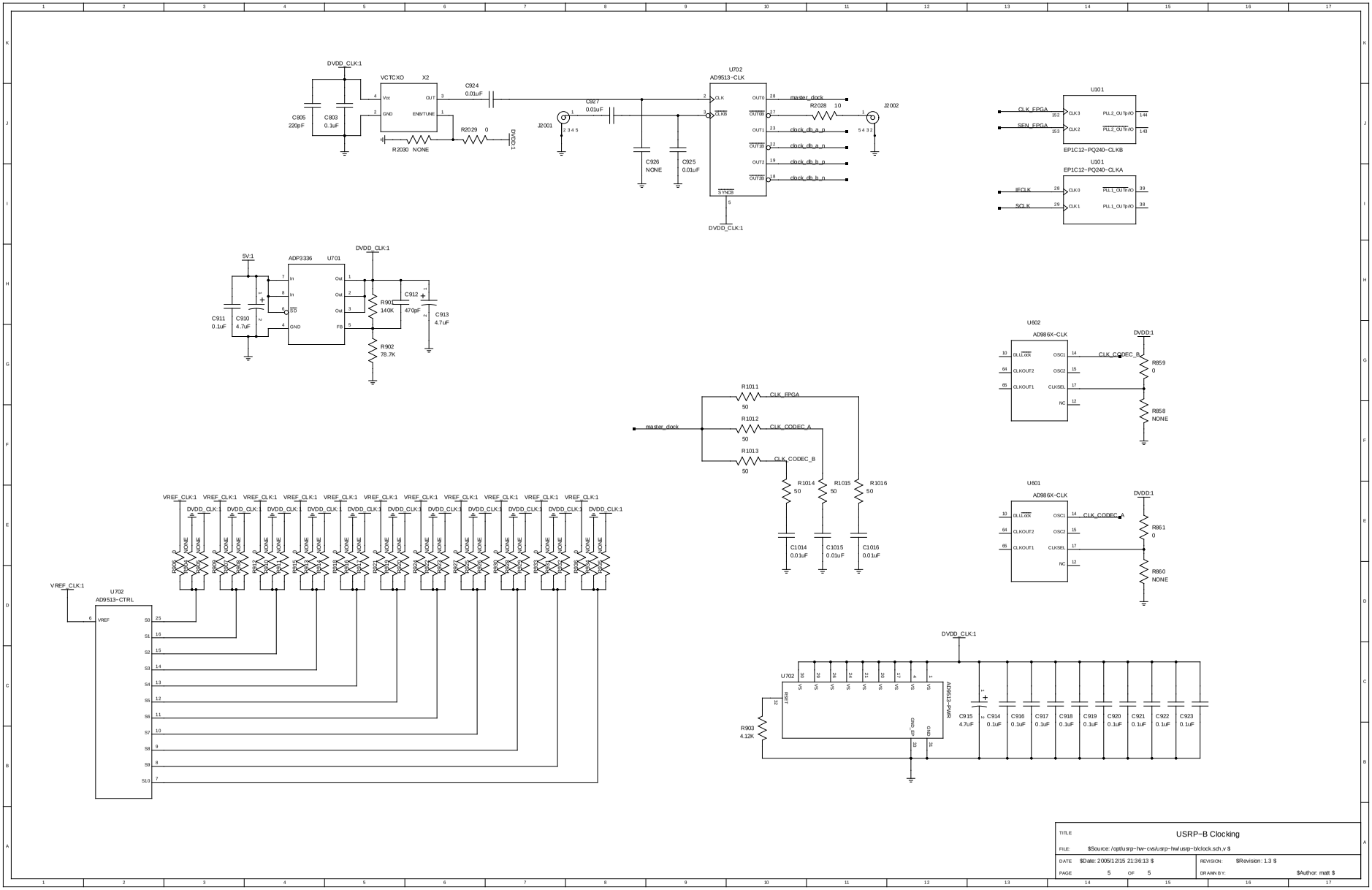
TITLE	$Date: 2005/12/15 20:34:07 $		
FILE: $RCSfile: interface.sch,v $		REVISION:	$Revision: 1.5 $
PAGE 2 OF 5		DRAWN BY:	$Author: matt $



USRP−B FPGA and CODECs
$Date: 2005/12/15 20:34:19 $
$Revision: 1.6 $
$Author: matt $
FILE: $RCSfile: fpga.sch,v $
PAGE 3 OF 5
U101 EP1C12−PQ240−IO1
U101 EP1C12−PQ240−IO2
U101 EP1C12−PQ240−IO3
U101 EP1C12−PQ240−IO4
U601 AD986X−TX
U601 AD986X−RX
U802 AD986X−TX
U802 AD986X−RX
U412 CY7C68013−TQ100−BUS
Connect others to header
IFCLK
SEN_CODEC_A
SEN_CODEC_B
MYSTERY_SIGNAL
SCLK
SDI
SDO
R2001 NONE
R2002 0
R2003 0
R2004 NONE

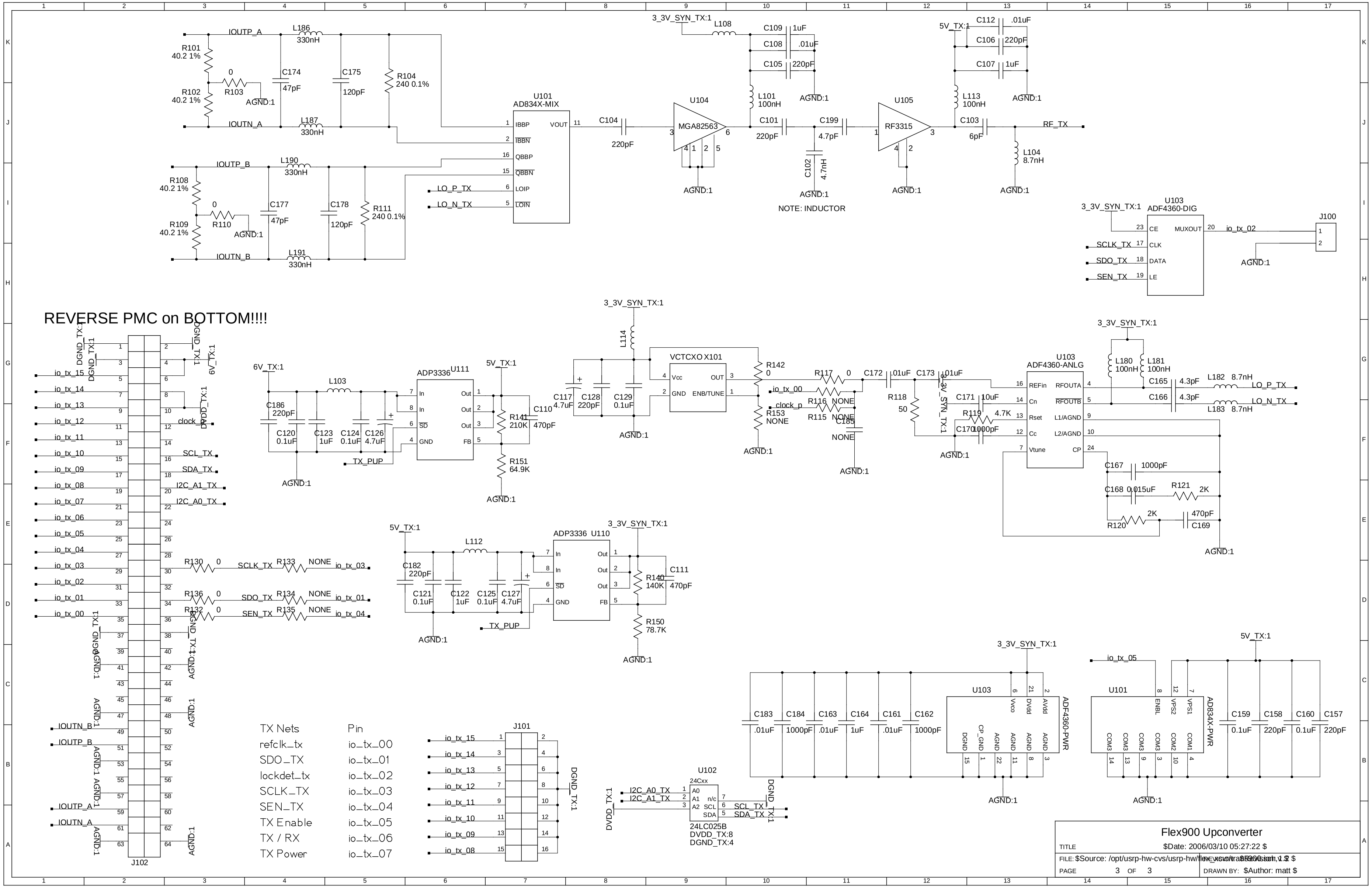


TITLE	USRP-B Daughterboad Interface
	$Date: 2005/12/15 20:33:53 $
FILE: $RCSfile: dboard.sch,v $	REVISION: $Revision: 1.5 $
PAGE 4 OF 5	DRAWN BY: $Author: matt $


DVDD_CLK:1
VCTCXO
X2
C805
220pF
C803
0.1uF
C924
0.01uF
R2029
0
R2030
NONE
U702
AD9513-CLK
C927
0.01uF
J2001
C926
NONE
C925
0.01uF
R2028
10
J2002
master_dock
clock_db_a_p
clock_db_a_n
clock_db_b_p
clock_db_b_n
SYNCB
U101
EP1C12-PQ240-CLKB
CLK_FPGA
SEN_FPGA
EP1C12-PQ240-CLKA
IFCLK
SCLK
5V:1
ADP3336
U701
C911
0.1uF
C910
4.7uF
R901
140K
C912
470pF
C913
4.7uF
R902
78.7K
U602
AD986X-CLK
CLK_CODEC_B
R859
0
R858
NONE
R1011
R1012
R1013
CLK_CODEC_A
50
R1014
R1015
R1016
5.0
C1014
0.01uF
C1015
0.01uF
C1016
0.01uF
U601
R861
R860
VREF_CLK:1
U702
AD9513-CTRL
VREF
S0
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
R903
4.12K
C915
4.7uF
C914
0.1uF
C916
0.1uF
C917
0.1uF
C918
0.1uF
C919
0.1uF
C920
0.1uF
C921
0.1uF
C922
0.1uF
C923
0.1uF
TITLE
USRP-B Clocking
FILE:
$Source: /opt/usrp-hw-cvs/usrp-hw/usrp-b/clock.sch,v $
DATE
$Date: 2005/12/15 21:36:13 $
REVISION:
$Revision: 1.3 $
PAGE
5
OF
5
DRAWN BY:
$Author: matt $

B.2 RFX900

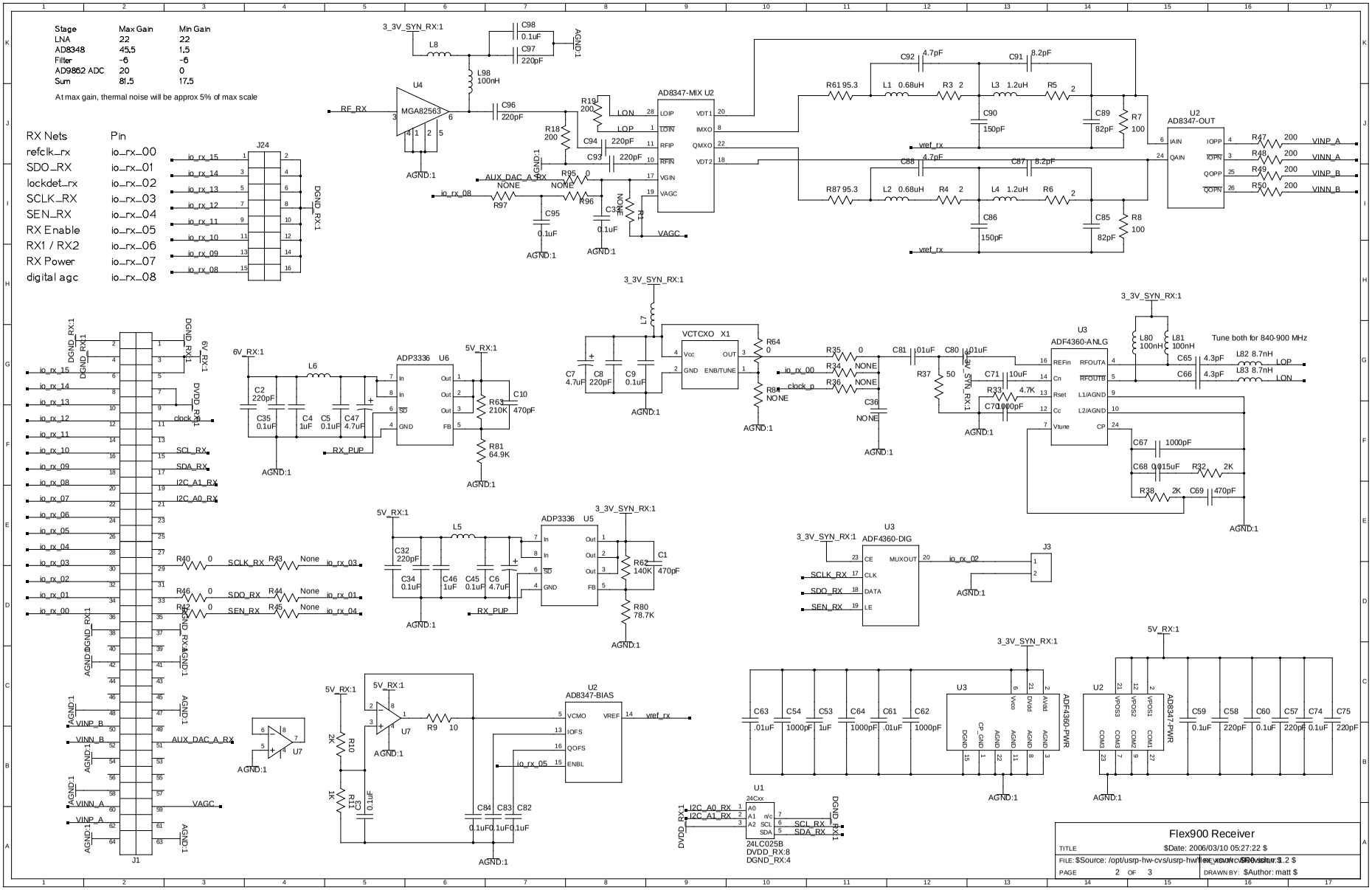
Skjema av RFX900 er hentet fra [Etta]


REVERSE PMC on BOTTOM!!!!
NOTE: INDUCTOR
TX Nets
refclk_tx
SDO_TX
lockdet_tx
SCLK_TX
SEN_TX
TX Enable
TX / RX
TX Power
Pin
io_tx_00
io_tx_01
io_tx_02
io_tx_03
io_tx_04
io_tx_05
io_tx_06
io_tx_07
U101 AD834X-MIX
U104 MGA82563
U105 RF3315
U103 ADF4360-DIG
U103 ADF4360-ANLG
U103 ADF4360-PWR
U101 AD834X-PWR
VCTCXO X101
ADP3336 U111
ADP3336 U110
U102 24LC025B
J100
J101
J102
Flex900 Upconverter
TITLE
$Date: 2006/03/10 05:27:22 $
FILE: $Source: /opt/usrp-hw-cvs/usrp-hw/flex_xcvr/flex900.sch,v $
PAGE 3 OF 3
DRAWN BY: $Author: matt $

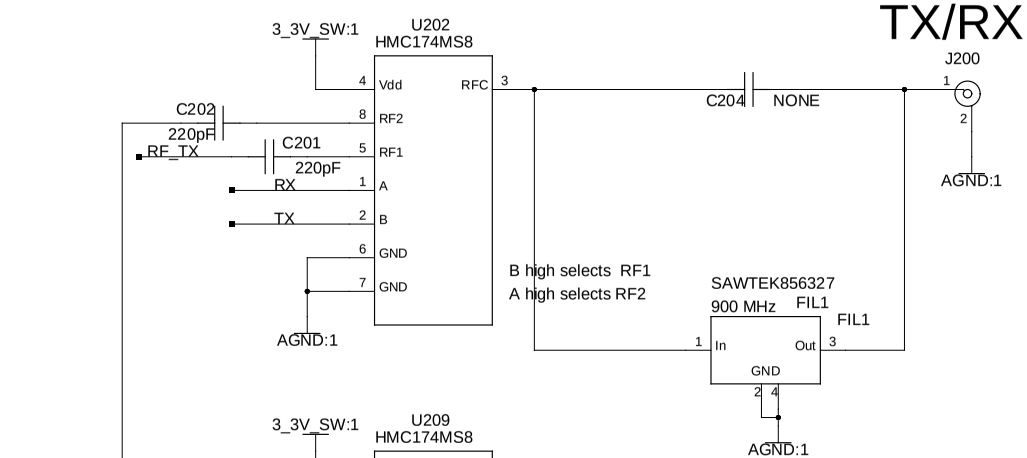
Stage	Max Gain	Min Gain
LNA	22	22
AD8348	45.5	1.5
Filter	-6	-6
AD9862 ADC	20	0
Sum	81.5	17.5

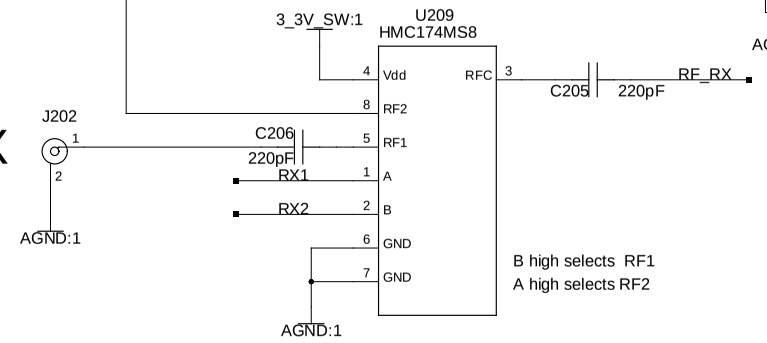
At max gain, thermal noise will be approx 5% of max scale

RX Nets	Pin
refclk_rx	io_rx_00
SDO_RX	io_rx_01
lockdet_rx	io_rx_02
SCLK_RX	io_rx_03
SEN_RX	io_rx_04
RX Enable	io_rx_05
RX1 / RX2	io_rx_06
RX Power	io_rx_07
digital agc	io_rx_08



TX/RX

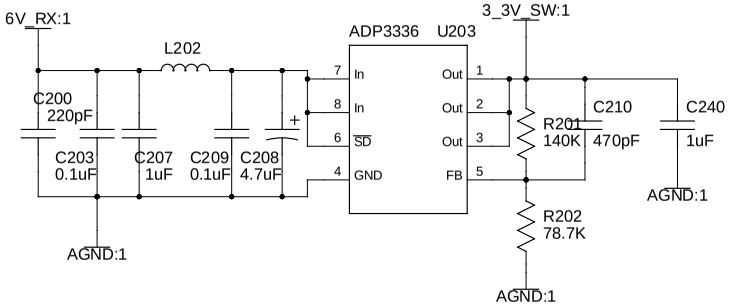


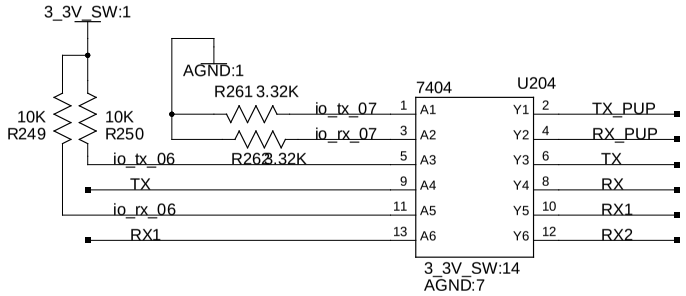


RX

A1	B1	A2	B2	
0	0	0	0	All Off
0	1	0	0	TX Only
0	1	1	0	TX Only
0	1	0	1	Full Duplex TX and RX
0	0	0	1	RX from secondary RX port
1	0	0	1	RX from secondary RX port
1	0	1	0	RX from primary TX/RX port
0	0	1	0	Nothing
1	0	0	0	Nothing
0	0	1	1	BADI
0	1	1	1	BADI
1	0	1	1	BADI
1	1	0	0	BADI
1	1	0	1	BADI
1	1	1	0	BADI
1	1	1	1	BADI

Always On





Band	Upconverter	Downconverter	Upconverter VCO	Downconverter VCO
400 - 500 MHz	AD8345	AD8348	ADF4360-7 (Div by 2)	ADF4360-3 (Div by 2)
800 - 975 MHz	AD8349	AD8347	ADF4360-3 (Div by 2)	Same
1.2 - 1.3 GHz	AD8349	AD8347	ADF4360-0 (Div by 2)	Same
2.3 - 2.4 GHz	AD8349	AD8347	ADF4360-1	Same
2.4 - 2.7 GHz	AD8349	AD8347	ADF4360-0	Same

Flex900 Common	
TITLE	$Date: 2006/03/10 05:27:22 $
FILE: $Source: /opt/usrp-hw-cvs/usrp-hw/flex_common/Flex900.sch,v $	$Revision: 1.2 $
PAGE 1 OF 3	DRAWN BY: $Author: matt $