

PLL tillämpning för mjukvarudefinierad FM-radio

John Främling

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Elektroteknik
Identifikationsnummer:	3530
Författare:	John Främling
Arbetets namn:	PLL tillämpning för mjukvarudefinierad FM-radio
Handledare (Arcada):	Rene Herrmann
Uppdragsgivare:	
<p>Sammandrag:</p> <p>Syftet med arbetet är att utveckla en mjukvarudefinierad FM-radio med PLL-struktur, där ett evalueringsbräde för frekvenssyntetisatorn ADF4360-8 tillsammans med evalueringsbrädets mjukvara installerad på en dator, används som sändare. Mottagarsidan består av en FM-radiomottagare som via en A/D-signalkonverter är kopplad till en dator med mjukvara för signalanalys. I arbetet undersöks även hur man kunde implementera kryptering för kommunikationen. En slumpalsgenerator programmeras för att skapa en serie pseudoslumptal som möjliggör kryptering av signalen med hjälp av frekvenshopp. Krypteringen har här bemärkelsen att kanalfrekvensen hoppar och kan därför inte avlyssnas utan att känna till frekvenshoppsmönstret.</p> <p>I detta examensarbete dokumenterades hur denna hård- och mjukvara passar sig för ovannämnda applikation, samt beskrivs principen för hur en kryptering kunde förverkligas med ovannämnda apparatur.</p> <p>I mätningen av hårdvarans frekvenshoppsegenskaper framgår att hårdvaran har restriktioner som måste kompenseras om man vill använda kryptering med hjälp av frekvenshopp. Utan restriktioner i hårdvaran skulle resultatet från mätningarna vara $y = x$. Mätningarna ger ändå ekvationen $y = 0,94x - 8,25$, vilket betyder att frekvensen är 6 % längre än den teoretiska frekvensen och att fördröjningen som huvudsakligen härstammar från restriktioner i kommunikationen mellan USB och frekvenssyntetisator, leder till ett frekvensfel på ca 8 Hz. Dessutom är frekvenssyntetisatorn långsammare än väntat. Spektralmätningarnas noggrannhet försämras av att samplingsfrekvensen endast är 48 kHz för mätningarnas hårdvara. Vid låg modulationshastighet är det relativa felet stort medan det vid höga modulationshastigheter är relativt lägre. Resultatet visar ändå att denna utrustning lämpar sig för användning som mjukvarudefinierad radio, samt att det är möjligt att implementera kryptering med frekvenshopp för den.</p>	
Nyckelord:	Mjukvarudefinierad radio, SDR, kryptering, frekvenssyntetisator, faslåst slinga, PLL, evalueringsbräde, frekvenshopp.
Sidantal:	33
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada – University of Applied Sciences	
Degree Programme:	Electrical Engineering
Identification number:	3530
Author:	John Främling
Title:	PLL- tillämpning för mjukvarudefinierad FM-radio.
Supervisor (Arcada):	Rene Herrmann
Commissioned by:	
<p>Abstract:</p> <p>The aim for this thesis is to develop a software-defined FM-radio with PLL-structure. The main hardware components consist of an evaluation board for the frequency synthesizer ADF4360-8, and a PC with the software for the evaluation board. These two connected together works as the transmitter. The receiver side comprise of a FM-receiver connected to a PC through an analog-to-digital converter. The receiver-side PC has software for signal analysis installed. In this thesis will also be introduced an implementation of encryption for the communication. A random number generator is programmed to create a series of pseudo-random numbers that will enable encryption of the signal using frequency hopping. This encryption using frequency hopping cannot be listened in on without knowing the exact frequency hopping pattern.</p> <p>This thesis documents how this soft- and hardware suits the foregoing application. It also describes the principle of how the encryption could be implemented with the mentioned equipment.</p> <p>Measurements of the frequency hopping capabilities shows that the hardware has restrictions which has to be compensated for if used in conjunction with the frequency hopping encryption. Without the hardware restrictions the result from the measurements would be $y=x$. The measurements gives an equation of $y = 0,94x - 8,25$, meaning that the frequency is 6% lower than the theoretical frequency, and the delays - that origin mainly from restrictions in the communication between the USB-connection and the frequency synthesizer - leads to a 8 Hz frequency error. The accuracy of the spectral measurements is negatively affected by the sampling rate of only 48 kHz for the equipment used. In addition, the frequency synthesizer in not as fast as predicted. Low time delay will give relatively less error than a higher time delay. Regardless of these inaccuracies, the results proves that this equipment is suited to be used for as a software-defined FM-radio, and also that it is possible to implement an encryption using frequency hopping in the communication.</p>	
Keywords:	Software-defined radio, SDR, encryption, frequency synthesizer, phase-locked loop, PLL, evaluation board, frequency hopping.
Number of pages:	33
Language:	Swedish
Date of acceptance:	

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Sähkötekniikka
Tunnistenumero:	3530
Tekijä:	John Främling
Työn nimi:	PLL tillämpning för mjukvarudefinierad FM-radio.
Työn ohjaaja (Arcada):	Rene Herrmann
Toimeksiantaja:	
<p>Tiivistelmä:</p> <p>Opinnäytteen tarkoituksena on kehittää FM-ohjelmistoradio PLL-rakenteella. Lähettimenä käytetään taajuussyntetisaattori ADF4360-8 evaluointilevyä, sekä siihen kuuluvaa PC-ohjelmistoa. Vastaanottimena toimii A/D-konverterrin kautta tietokoneeseen kytketty FM-radiovastaanotin. Tietokoneelle on asennettu signaalianalyysiin tarkoitettu ohjelmisto. Opinnäytteessä tutkitaan myös kuinka salaus olisi mahdollista toteuttaa kommunikoinnissa. Satunnaislukugeneraattori ohjelmoidaan pseudosatunnaissarjan luomiseksi. Tämä mahdollistaisi signaalin salauksen taajuushyppelyn avulla. Salauksen periaatteena on että kanavataajuus hyppii, eikä sen takia voida salakuunnella ilman että taajuushyppelykuvio on tiedossa.</p> <p>Tässä opinnäytteessä dokumentoitiin miten kyseinen ohjelmisto ja laitteisto soveltuu yllä mainittuun sovellukseen, sekä kuvataan periaatetta kuinka salauksen voisi toteuttaa edellä mainitulla laitteistolla.</p> <p>Laitteiston taajuushyppelyominaisuuksien mittauksissa käy esille, että laitteistolla on rajoituksia joita on kompensoitava jos halutaan käyttää taajuushyppelysalausta. Ilman laitteiston rajoituksia olisi mittausten tulos $y = x$. Mittausten tulokset antavat yhtälön $y = 0,94x - 8,25$, mikä tarkoittaa että taajuus on 6 % alempi kuin teoreettinen taajuus ja että viive, joka pääasiassa aiheutuu USB:n ja taajuussyntetisaattorin välisestä kommunikointirajoituksesta, aiheuttaa 8 Hz:n taajuusvirheen. Mittausten tarkkuuteen vaikuttaa huomontavasti myös mittauslaitteiston alhainen näytteenottotaajuus (48 kHz). Tämän lisäksi taajuussyntetisaattori on odotettua hitaampi. Alhaisella modulointinopeudella suhteellinen virhe on iso, kuin taas isoilla modulointinopeuksilla suhteellisesti pienempi. Tulos kuitenkin osoittaa, että edellä mainittu laitteisto soveltuu käytettäväksi ohjelmistoradiona ja myös että salaus on mahdollista toteuttaa taajuushyppelyä käyttäen.</p>	
Avainsanat:	Ohjelmistoradio, SDR, salaus, taajuussyntetisaattori, vaihelukittu silmukka, taajuushyppely.
Sidantal:	33
Språk:	Svenska
Datum för godkännande:	

INNEHÅLL

1	INTRODUKTION	7
2	TEORI	9
	2.1 Superheterodynmodulatorn	11
	2.2 En faslåst slinga, PLL	11
	2.3 Evalueringsbrädet EVAL-4360-8EBZ1	12
	2.4 Mjukvarudefinierad radio, SDR	13
	2.5 Kryptering med frekvenshopp	14
3	EXPERIMENTELL GENOMFÖRNING	16
	3.1 Digital modulation med evalueringsbrädet	20
	3.2 Demodulering av en digital radiosignal	20
	3.3 Slumptalsgenerator för frekvenshopp	22
4.	RESULTAT	25
5	SAMMANDRAG	31
	Källor	32

FIGURER

Figur 1. Mjukvarudefinerad radio

Figur 2. Traditionell sändare med transceiver

Figur 3. Blockschema för en FM-mottagare med varierande sändarfrekvens

Figur 4. Multiplexer

Figur 5. Principen för FSK [2]

Figur 6. Signalerna $f(A)$ och $f(B)$ med sidoband vars avstånd från mittfrekvensen beror av tiden

Figur 7. Blockschema för superheterodynen [3]

Figur 8. Blockschema för frekvenssyntetisator

Figur 9. Traditionell mjukvarudefinerad radio [7]

Figur 10. Evalueringsbrädets graf för minimum respektive maximum frekvens vs. värde för den externa induktansen [10]

Figur 11. Blockschema för komponenter som användes i arbetet

Figur 12. Användargränssnitt för EVAL-ADF4360-8-9EBZ1

Figur 13. Användargränssnitt för Audacity

Figur 14. Blockschema för enkelriktad kommunikation, simplex

Figur 15. Blockschema för halv duplex

Figur 16. Blockschema för full duplex

Figur 17. Koden för en slumpvalsgenerator

Figur 18. Blockschemat för experimentet

Figur 19. Användargränssnitt för Clio 10 [12]

Figur 20. Relativa felets korrelation med experimentella frekvensen

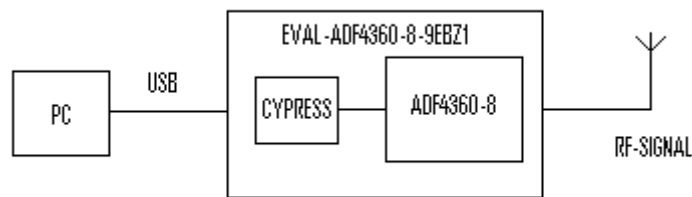
Tabell 1. Utrustning för att utföra experimentet

Tabell 2. Teoretisk modulationshastighet som funktion av tiden

Tabell 3. Resultatet från mätningen

INTRODUKTION

Syftet med arbetet är att tillämpa en frekvenssyntetisator med PLL-struktur (faslåst slinga, Phase Locked Loop) för att demonstrera mjukvarudefinierad FM-radio med smal bandbredd. Småbandsradioteknologin används för telemetridatatransmission eller automationsuppgifter. Mjukvarudefinierad radio är ett sätt att m.h.a en PC och en enkel hårdvarumodul skapa radiosignaler utan att det behövs en skild transceiver.



Figur 1. Mjukvarudefinierad radio.

Figur 1 visar hur sändaren i arbetet är uppbyggd. En PC är kopplad via en USB-kabel till evalueringsbrädet, och ut fås en rf-signal. I en traditionell sändare är PC'n med ljudkort kopplat till en dyr transceiver.



Figur 2. Traditionell sändare med transceiver.

I arbetet presenteras frekvenssyntetisatorn ADF4360-8, som är en integrerad heltals-N frekvenssyntetisator med en spänningsstyrd oscillator (VCO). VCO's frekvensband sträcker sig från 65 till 400 MHz. I standard PLL lösningar är låsdetektorkretsen analog eller digital. Låsdetektorn i ADF4360-8 är en digital krets (DLD) som ger ett logiskt

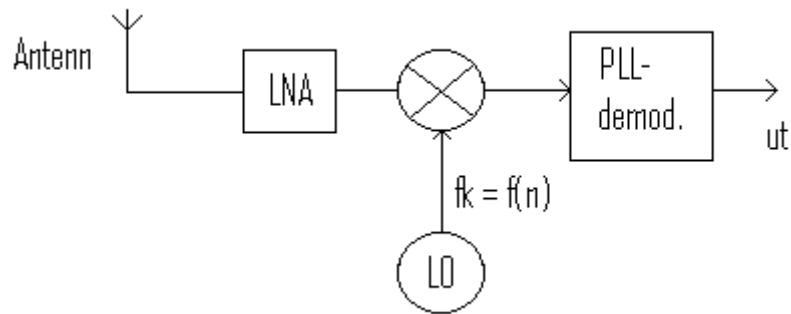
högt som utsignal då fasfelet för tre (eller vid behov fem) på varandra följande fasdetektorcykler är lägre än 15 ns. Den visar ett logiskt högt så länge fasfelet är lägre än 25 ns för två cykler.

I arbetet används ett evalueringsbräde för frekvenssyntetisatorn ADF4360-8 för att demonstrera mjukvarudefinierad FM-radio, dess bandbredd i förhållande till mjukvaruparameterintervall, radiosignalmottagning och kommunikationsfördröjningen mellan PC och evalueringsbrädet. I arbetet presenteras också hur en enkel kryptering kan skapas med frekvenshoppning och vad det skulle innebära för radiomottagarens hårdvara.

Den digitala radiosignalen skapas genom att välja bestämda frekvenshopp i PLL-kretsen. Hoppastigheten är modulationshastigheten, hoppfrekvenserna är definitioner på de digitala värdena. Medeltalet av frekvenserna är kanaldefinitionen.

Vid digital mottagning är den lokala oscillatoren matad av frekvensen som är medeltalsfrekvensen, alltså kanaldefinitionen. Är den mottagna frekvensen mindre än medeltalet så mottas signal A, om den däremot är högre så mottas i stället B.

Kryptering är möjligt genom att definiera radiosignalen så, att A svarar mot en negativ frekvensskillnad från medelfrekvensen och B som en likadan positiv skillnad från medelfrekvensen. Själva kanalfrekvensen som då är medelfrekvensen kan dock nu försees med vilket som helst värde, m.a.o. kan den byta enligt ett hemligt mönster. Innebörden är då att mottagaren som vet om hoppmönstret, mottar vid dessa hoppande frekvenser en negativ eller positiv frekvensskillnad, A eller B. Krypteringen har här bemärkelsen att kanalfrekvensen hoppar och kan därför inte avlyssnas utan att känna till frekvenshoppningsmönstret.



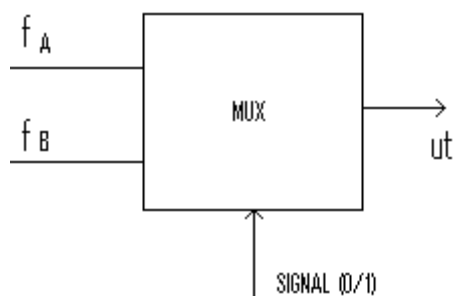
Figur 3. Blockschema för en FM-mottagare med varierande sändarfrekvens.

Sändaren hoppar enligt ett mönster $f(n)$ som mottagaren känner till. Detta mönster genereras av en slumpvalsgenerator.

FM-demodulation används vid dekodning av informationen i en FM-radiosignal.

2. TEORI

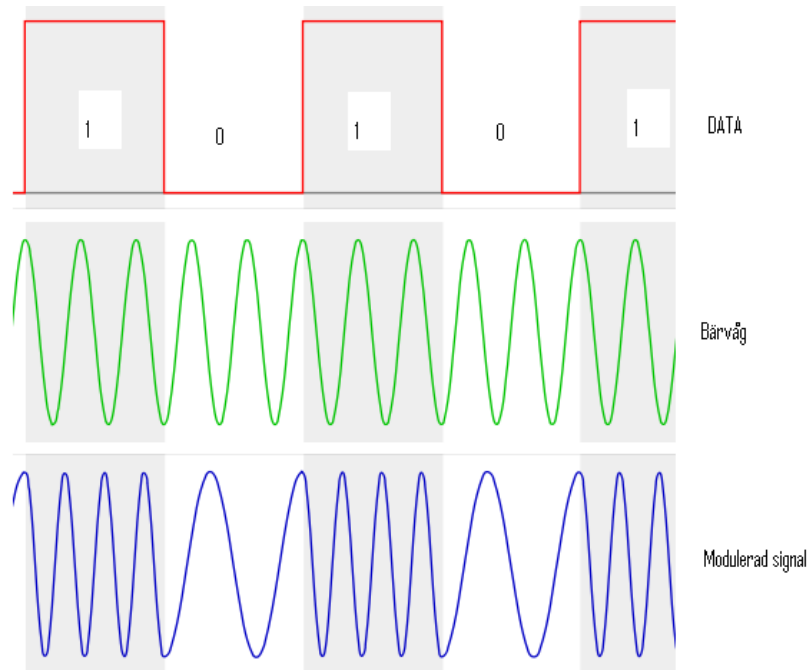
Digitala FM-signaler kan skapas till exempel genom en multiplexer som väljer mellan två olika frekvenser. Hastigheten med vilken de byts heter modulationshastigheten. Ett exempel av en multiplexer ses i figuren nedan:



Figur 4. En fax är ett typiskt exempel på hur digitala signaler kan skapas med hjälp av en multiplexer.

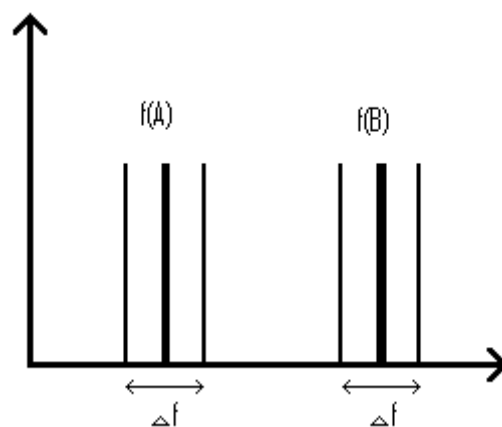
En 2-till-1 multiplexer som kunde användas i en fax är en enhet där utsignalen ändras beroende på om en pixel är vit eller svart. Tekniken tillåter också att flera användare eller processer delar på samma resurs [1].

Frequency-shift keying (FSK) är en frekvensmodulationsteknik där digital information överförs med två olika frekvenser som motsvarar 0 respektive 1. [2]



Figur 5. Illustration för FSK-principen [2].

Bärvågen moduleras beroende på data som skall sändas, dvs. om det är "0" eller "1". I detta fall leder en etta (låt oss kalla den för $f(B)$) till en högre frekvens på den modulerade signalen än en nolla ($f(A)$).

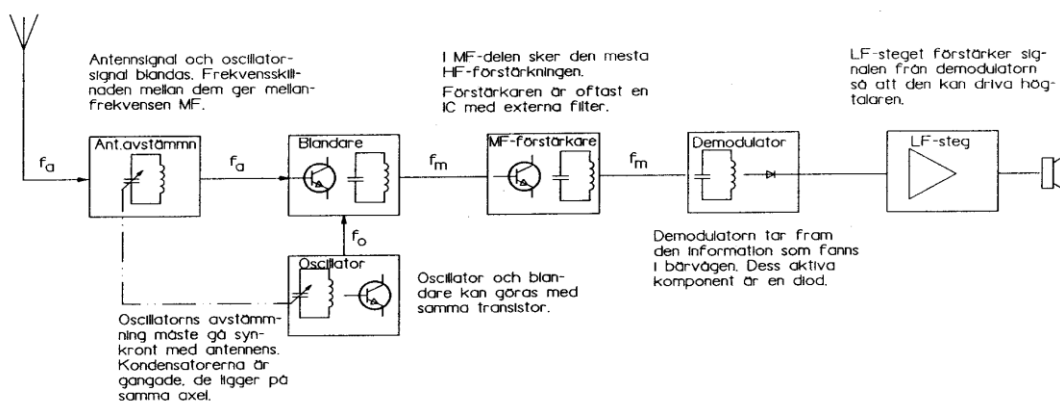


Figur 6. Signalerna $f(A)$ och $f(B)$ med sidoband vars avstånd från mittfrekvensen beror av tiden.

Figur 6 visar två signaler, A och B, vars sidobands avstånd från själva mittfrekvensen beror på modulationshastigheten.

2.1 Superheterodynprincipen

Superheterodynmottagaren utnyttjar heterodynprincipen för att omvandla en radiofrekvens till mottagarens förutbestämda mellanfrekvens. En viktig fördel med en låg mellanfrekvens är att man lättare kan stänga ut radiostationer man inte vill höra som ligger nära den önskade kanalen. Man slipper också göra flera frekvensberoende kretsar i mottagaren och ändra alla synkront när man vill lyssna på en annan sändarfrekvens. [4]



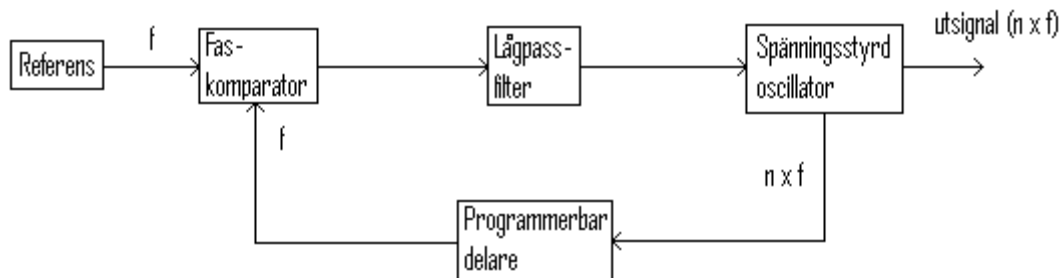
Figur 7. Superheterodynens blockschema [3].

Då signalen kommer till mottagaren via antennen, skickas den till en blandare som blandar den med en signal genererad i en lokal oscillator (LO). Blandningen ger ut två frekvenser, dvs. summan och differensen av de två signalerna. Om oscillatornsignalen hålls på ett konstant frekvensavstånd från antennsignalen blir skillnaden mellan dessa också konstant. Denna nya frekvens kallas mellanfrekvens (MF) och är standardiserad (10,7 MHz för FM-bandet).

2.2 En faslåst slinga, PLL

En faslåst slinga (på engelska Phase Locked Loop) är en sluten återkopplad krets för att

låsa en genererad signal till en viss frekvens som motsvarar en yttre referensfrekvens. Genom jämförelse och justering mellan referensen och den internt genererade signalen så kommer den genererade signalen att till slut att matcha referensen och låser då vid denna med både rätt frekvens och fas. [5]



Figur 8. Blockschemata för frekvenssyntetisator.

Många frekvenssyntetisatorer är baserade på en faslåst slinga (PLL, phase locked loop). För att en PLL skall kunna konverteras till en frekvenssyntetisator behöver den en frekvensdelare mellan VCO'n och fasdetektorn (se figur 8). (Banerjee 2006 s.10)

FM-demodulation med PLL FM-demodulatorer på senare tid blivit populärt. PLL FM-detektorer kan lätt framställas med hjälp av en mängd integrerade PLL kretsar som finns till handa, och därmed kan man också hitta PLL FM-demodulatorer i många typer av radioutrustningar. Integrerade kretsar för PLL FM-demodulation uppkom då teknologin för integrerade kretsar blev så långt utvecklad att man kunde tillverka analoga RF-kretsar. [14]

2.3 Evalueringsbrädet EVAL-ADF4360-8EBZ1

I slutarbetet används ett evalueringsbräde för frekvenssyntetisatorn ADF4360-8. Till evalueringsbrädet hör mjukvara som används för programmeringen av frekvenssyntetisatorn. Evalueringsbrädets mjukvara används bl.a. för att programmera referensfrekvensen och dess delare, ströminställningar till laddningspumpen, frekvensinställningar för utfrekvensen, samt för frekvenshopp mellan frekvenser.

Evalueringsbrädet är designat för att låta användaren utvärdera egenskaperna för frekvenssyntetisatorn ADF4360-8. Den består av frekvenssyntetisatorn ADF4360-8BCPZ, en USB-port och SMA kontakter för RF utsignal. Den innehåller också ett kretsfilter för att fullända PLL'n. Med hjälp av en medföljande kabel kan evalueringsbrädet kopplas till en PC för programmering med hjälp av mjukvara.

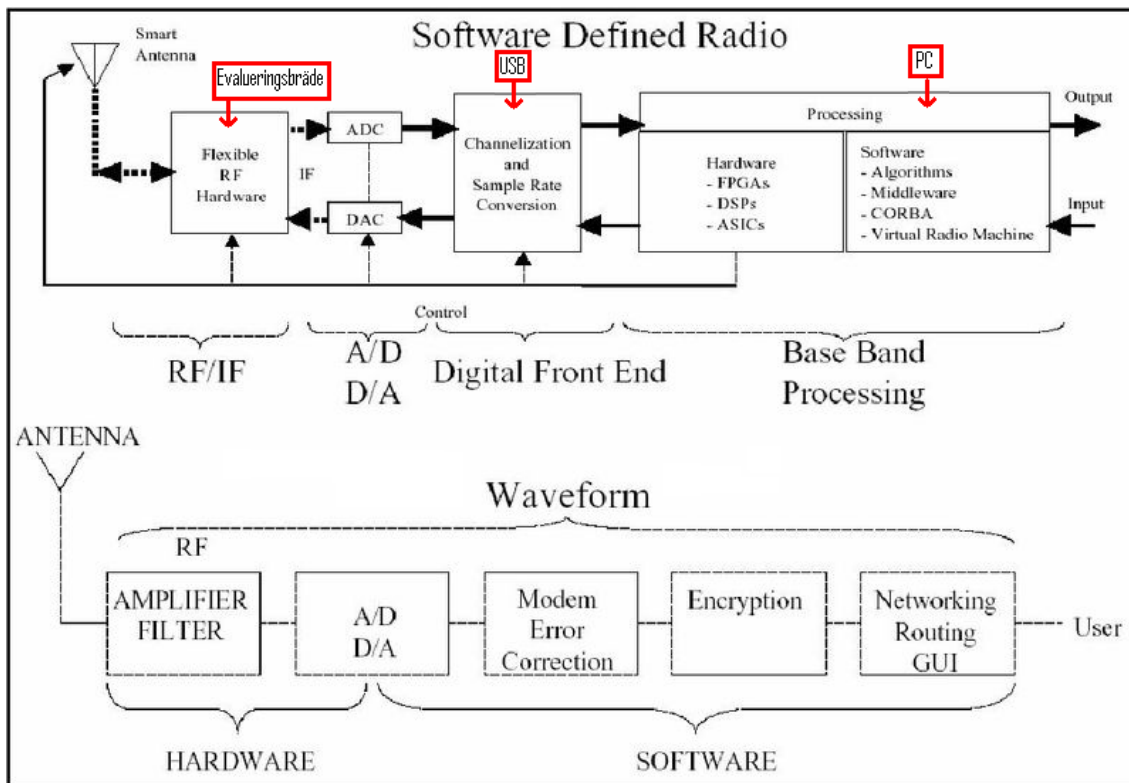
Brädet får sin ström från ett 9 V batteri eller via USB-kabel. En extern referenssignal kan användas eller också den inbyggda 10 MHz TCXO'n (Temperature Corrected Crystal Oscillator). Designparametrarna för kretsfiltret är för en 120 MHz centerfrekvens, en PFD frekvens på 1 MHz och en bandbredd för lågpassfiltret på 40 kHz. För andra frekvenser kan man med hjälp av programmet ADIsimPLL simulera hurdana komponenter som behövs.

2.4 Mjukvarudefinierad radio, SDR

Mjukvarudefinierad radio betyder att man i stället för ett traditionellt system där man med komponenter som implementerats med hårdvara i stället implementerar behövliga komponenter med mjukvara på en dator.

Traditionella hårdvarubaserade radioapparater har en begränsad funktionalitet och kan endast modifieras genom att ändra på hårdvaran, vilket resulterar i högre produktionskostnader och minimal flexibilitet. Dessa problem löses med mjukvarudefinierad radio, eftersom man endast med att omprogrammera mjukvaran för SDR kan ändra på dess egenskaper.

Fördelar med den mjukvarudefinierade radion är att man kan skapa en stor skala olika produkter via en gemensam plattform vilket gör att nya produkter snabbare fås ut på marknaden. Då samma mjukvara kan användas i flera olika radioapplikationer sjunker också utvecklingskostnaderna. En viktig fördel är också att man kan fjärrprogrammera trådlöst, vilket gör att t.ex. fel i mjukvaran kan korrigeras eller uppdateras trots att radion är i användning.[6]



Figur 9. Traditionell mjukvarudefinierad radio [7].

Figur 9 ovan visar blockschemat för en traditionell mjukvarudefinierad radio. I arbetet har strukturen förenklats så, att ”antennen” är evalueringsbrädet, USB-kabeln är bildens kanalisering och i datorn finns mjukvaran för evalueringsbrädet.

2.5 Kryptering med frekvenshopp

Kryptering består av två delar, en algoritm och en nyckel. När man skall kryptera information behöver man både en publik algoritm och en hemlig nyckel. Då man dekrypterar behöver man ha tillgång till den hemliga nyckeln och algoritmen. I arbetet används en s.k. symmetrisk kryptering, vilket betyder att man använder samma nyckel vid kryptering och dekryptering. Både mottagaren och sändaren måste ha tillgång till samma nyckel. Ett problem med denna är hur man skall överföra nyckeln på ett bra sätt från sändaren till mottagaren. [8] I vårt fall är detta inget problem eftersom sändare och mottagare är samma. Vi använder oss av en slumpgeneratorfunktion för att få tillstånd en nyckel. Nyckellängden är 8 bitar vilket betyder att det totala antalet nycklar är

. Den förverkligas genom c-programmering (se figur 17).

Med endast matematiska funktioner är det omöjligt att skapa en helt slumpmässig serie tal eftersom datorer inte kan "gissa" slumpmässigt, de kan endast följa instruktioner. De får i stället en slumpmässig serie till stånd med hjälp av ett frö. Detta frö är ett nummer som tas från en källa som är oförutspåbar, t.ex. exakta tiden från datorn. Tack vare fröet undviker man också att en slumpmässgenerator inte producerar samma serie tal varje gång den körs.

För att sedan få någonting ut av fröet behövs en algoritm. En skenbart slumpmässig serie tal som genereras av en algoritm kallas för pseudoslumpmässgenerator. Exempel på algoritmer för slumpmässgenerering är t.ex. von Neumanns metod eller Blum Blum Shub [9].

Eftersom alla program som används på datorer bara har tillgång till en begränsad mängd minne, kan en talserie inte heller anta en oändlig mängd värden. Detta leder till att en pseudoslumpmässsekvens i längden kommer att börja upprepa sig. Antalet tal som kan produceras innan pseudoslumpmässsekvensen börjar upprepa sig kallas för period.

För en bra slumpmässgenerator krävs det bl.a. att perioden bör vara lång, fördelningen bör vara så gott som likformig samt nära efterföljande värden bör vara oberoende av varandra. Speciellt viktigt är förstås också att slumpmässigen inte har någon regelbundenhet.

3 EXPERIMENTELL GENOMFÖRNING

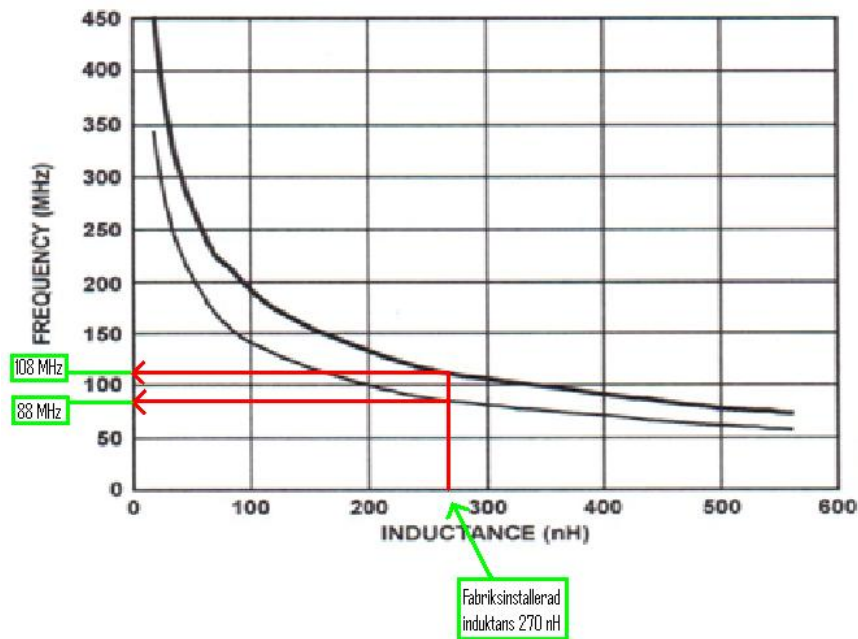
Till den experimentella undersökningen används evalueringsbrädet, två datorer, en FM-radiomottagare samt signalanalys- och annan mjukvara enligt följande tabell:

Utrustning	Typ	Version
PC 1	Intel Pentium 4 CPU 2.80GHz 2.86GHz, 1,00 GB RAM	
PC 2	Asus Intel Core2 Duo CPU T7300 @ 2.00 Ghz	
Evaluation board	EVAL-ADF4360-8/-9EBZ1	rev. A
USB-kabel	28AWG/2C+28AWG/2C	
Kabel radio-konverter	Microphone Cable CME 220	
Mjukvara	Analog Devices Evaluation Software ADF4360-x	rev. 4.0.3.
	Microsoft Windows XP Pro	Version 2002 SP3
	Clio 10	Ver. 10.31, build 10.3.1.1
Radiomottagare	Denver TWR-802	
A/D-audio konverter	Clio fw FW-01	

Tabell 1. Utrustning för att utföra experimentet.

3.1. Digital modulation med evalueringsbrädet

Det tillgängliga frekvensbandet vid evalueringsbrädet är 65 – 400 MHz. Med externa induktanser är det möjligt att använda hela frekvensbandet. Varför just frekvenssyntetisatorn ADF4360-8 valdes till arbetet beror på att det med fabrikskonfigurationen utlovas ett frekvensband från 88 till 108 MHz, vilket räcker till för laborationen. Utfrekvensen kan ställas in genom att via mjukvaran ställa in frekvenserna, men behövs ett annat frekvensband än 88-108 MHz är man tvungen att löda externa induktanser till evalueringsbrädet. Hur induktanserna då bör väljas framgår i figur 10 nedan.



Figur 10. Evalueringsbrädets graf för minimum respektive maximum frekvens vs. värde för den externa induktansen [10]

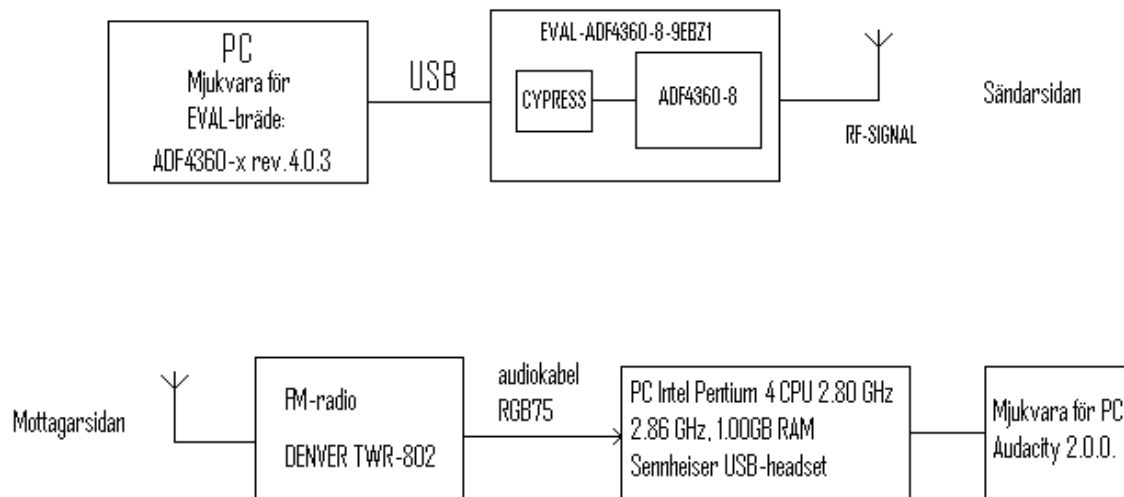
Ur figur 10 ovan kan man läsa hur den på evalueringsbrädet fabriksinstallerade induktansen (270 nH) leder till det ovan nämnda frekvensbandet 88-108 MHz för frekvenssyntetisatorn som används i laborationen.

Enligt mjukvarans tillverkare Analog Devices, kan fördröjningen mellan ett frekvensbyte ställas in från 0 ms uppåt. Mjukvaran har tre register som styr frekvenssyntetisatorn; N-räknaren, Control och R-räknaren. I samtal med tillverkaren framgår även att då mjukvaran gör frekvenshopp, skriver den kontinuerligt i de tre registren via USB-förbindelsen. Varje register ”skrivning” är en USB-operation, och alla tre register skrivs vid varje hopp. För varje USB-operation är fördröjningen ca 1 ms, vilket betyder att den totala fördröjningen för varje frekvensbyte är ca 3 ms. Då tiden för frekvensbytet i programvaran ställs till 0, är USB-operationen den enda fördröjningen. Tillverkaren konstaterar också att genom att modifiera mjukvaran är det möjligt att få den att endast skriva N-räknaren vilket således skulle leda till att fördröjningen skulle vara endast ca 1 ms. Den teoretiska modulationshastigheten för fördröjningarna 1-10 ms är då enligt tabellen nedan.

Fördröjning [ms]	fteoretisk [hz]
1	1000
2	500
3	333
4	250
5	200
6	167
7	143
8	125
9	111
10	100

Tabell 2. Teoretisk modulationshastighet som funktion av tiden.

Mätningar utfördes med en FM-radiomottagare och demodulationsmjukvara. Den faktiska modulationshastigheten blev som väntat långsammare än den teoretiska. Skillnaden orsakas av fördröjningen i kommunikationen mellan PC och PLL-chipen.



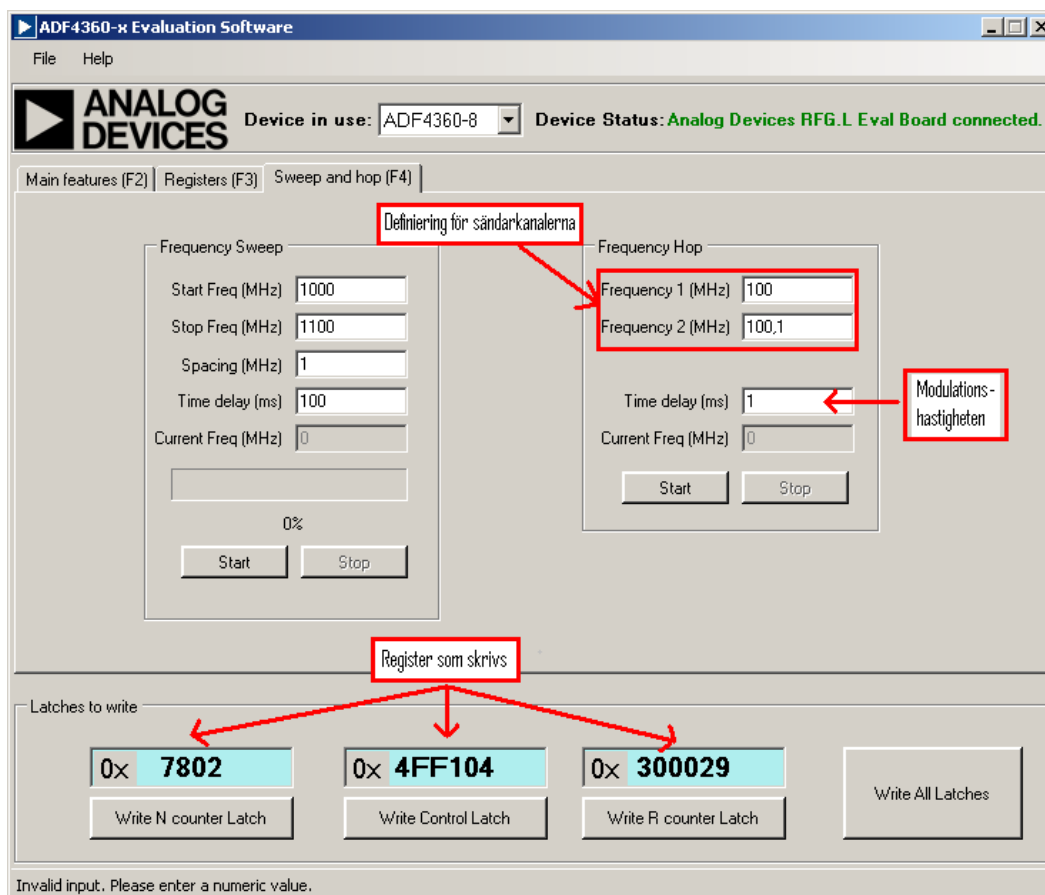
Figur 11. Blockschema för komponenter som användes i arbetet. Övre delen beskriver sändarsidan, nedre delen mottagarsidan.

På sändarsidan använder vi en dator med mjukvaran ADF4360-x Evaluation Software för evalueringsbrädet. Datorn är kopplad till evalueringsbrädet med en USB-kabel.

Mjukvaran för evalueringsbrädet räknar ut de hexadecimalvärden som skall skrivas till PLL'n. Dessa värden skickas från datorn till mjukvaran för USB'n som finns på evalueringsbrädet, som i sin tur skriver och skickar värdet vidare till PLL'n. För att få en bättre signal till radiomottagaren används en koaxialkabel som antenn från evalueringsbrädet.

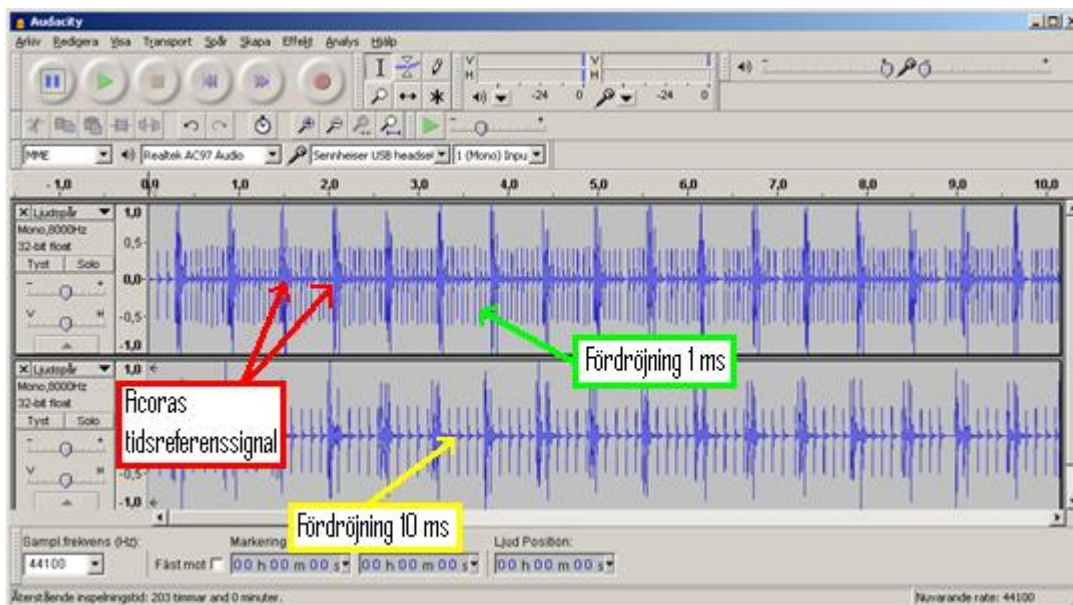
På mottagarsidan fungerar en FM-radio av märket Denver TWR-802 som mottagare. Radion är kopplad till en bärbar dators ljudkort med en kabel. Med programmet Audacity (version 2.0.0) som är installerat på datorn får man fram den dekodade signalen. Sändar- och mottagarsidan använder i detta fall samma dator.

Figur 12 nedan visar hur användargränssnittet ser ut för evalueringsbrädets mjukvara. Då man i praktiken testar evalueringsbrädet, ställs ett frekvenshopp in (i detta fall 100,0 MHz och 100,1 MHz), och också en fördröjningstid (time delay) för hur lång tiden skall vara mellan hoppen. Då man ställer in tidsfördröjningen till 1 ms och sedan ändrar den till 10 ms kan man med bara örat höra att en tydlig förändring sker. Detta bevisas även med programmet Audacity på mottagarsidan.



Figur 12. Användargränssnitt för EVAL-ADF4360-8-9EBZ1.

För att dekodra sändarsignalen används programmet Audacity. Figur 13 nedan visar ett test med två signaler med olika fördröjning, 1 ms respektive 10 ms.



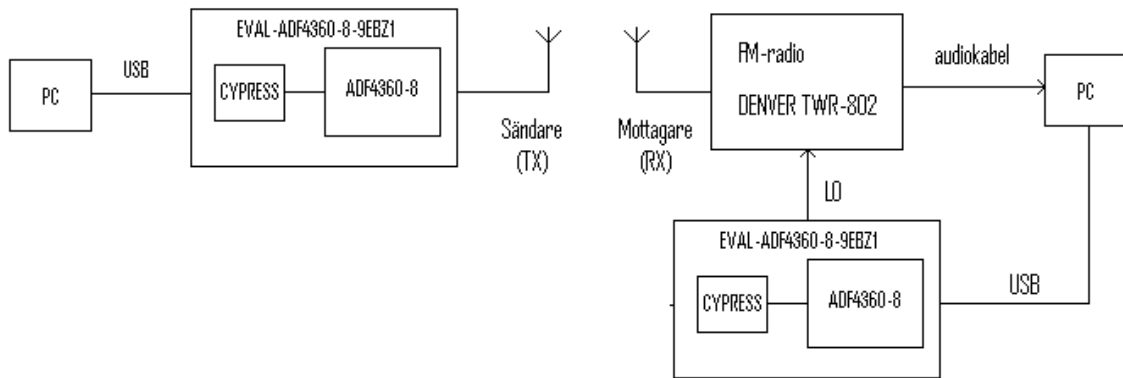
Figur 13. Användargränssnittet för Audacity.

I figur 13 ovan ses hur en förändring av fördröjningen ändrar på spektret för signalerna. Den övre har en fördröjning på 1 ms medan den nedres fördröjning är 10 ms. Den gröna pilen visar det betydligt tätare spektret för den övre signalen, medan den gula pilen visar det glesare spektret för den nedre signalen. De röda pilarna visar Ficoras 0.5 sekunders tidsreferenssignal som sändes på samma frekvens som användes här.

3.2. Demodulering av en digital radiosignal

En FM-superheterodynmodulator har en principiell struktur enligt figur 7. Strukturen kan förenklas genom att ersätta den lokala oscillatoren med en PLL. Detta leder till olika möjliga länkstrukturer. Nedan beskrivs strukturer för enkelriktad-, halv duplex och full duplex radiotransmission.

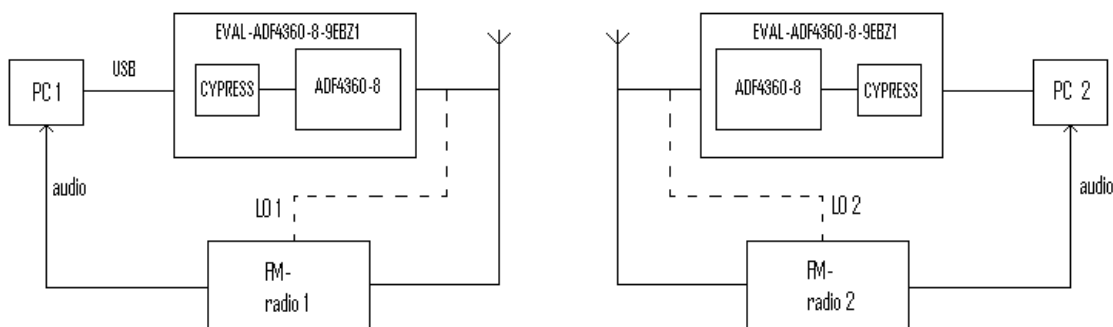
Enkelriktad radiotransmission, även kallad simplex, har den enklaste länkstrukturen för radiotransmission. Radioprogram är ett exempel på enkelriktad transmission. Det finns en sändare och en mottagare och de kan endast fungera som sändare respektive mottagare.



Figur 14. Blockschema för enkelriktad kommunikation, simplex.

I figuren ovan är sändaren (TX) placerad till vänster och mottagaren (RX) till höger. I arbetet används de ovanstående komponenterna så, att sändar- och mottagarsidan använder samma PC och evalueringsbräde. Om man har infört kryptering för sändningen ändrar lokaloscillatorn frekvens enligt slumpmönstret $f(n)$, förutsatt att mönstret är känt för mottagaren.

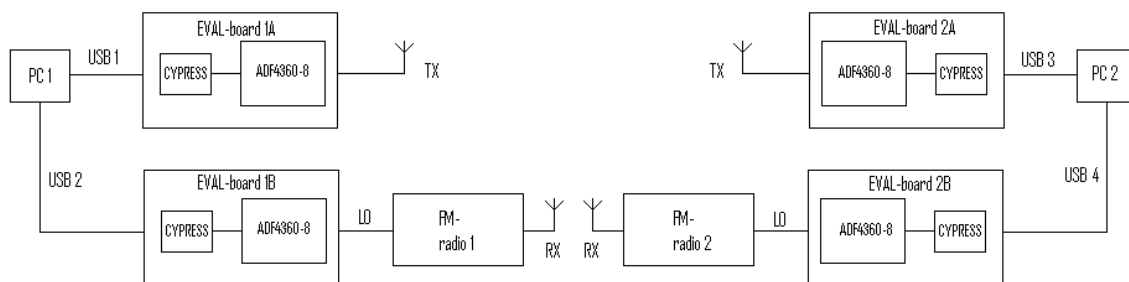
En halv duplex länk använder en enda kanal för att sända och ta emot. Systemet är turvis dubbelriktat, och kan fungera t.ex. med hjälp av en push-to-talk -knapp, såsom radiotelefonen som endast kan sända då PTT-knappen är intryckt. Dessa är ofta system som använts redan länge, som till exempel den amplitudmodulerade VHF-radion som fortfarande används inom den kommersiella flygtrafiken. Fördelen är att det inte kräver någon annan infrastruktur än två radion för att kunna kommunicera med varandra. Som nackdel är den begränsade räckvidden; de båda radioapparaterna bör vara inom varandras räckvidd.



Figur 15. Blockschema för halv duplex.

I figur 15 ovan visas länkstrukturen för en halv duplex. Då PC 1 sänder ut en signal är LO 1 inte kopplad till FM-radio 1, men LO 2 är kopplad till FM-radio 2. Om länken är krypterad hoppar LO 2 enligt slumpталsserien $f(n)$. Länken är alltså dubbelriktad men inte samtidigt.

En full duplex länk sänder och tar emot på olika kanaler. En full duplex är ett system som är samtidigt dubbelriktat. Här kan radion både sända och ta emot samtidigt, ett exempel är mobiltelefonen. [11]



Figur 16. Blockschema för full duplex.

Vid kryptering byter lokaloscillatorn frekvens enligt ett mönster $f(n)$.

3.3. Slumptalsgenerator för frekvenshopp

I arbetet används en enkel slumptalsgenerator för att generera en serie slumpmässiga tal som skall användas för att göra ett schema för hur frekvenshoppet skall ske. Tack vare dessa frekvenshopp kan i praktiken ingen annan än den som har den s.k. nyckeln avlyssna informationen som skickas. Avlyssning innebär att sändaren byter frekvenser enligt mönstret $f(n)$ och mottagarens dekryptering att frekvenshoppet kan följas.

Med koden nedan programmeras en slumptalsgenerator för frekvenshoppet i arbetet. I detta fall fås tre olika slumpmässiga talserier med 255 tal, som antar värden mellan 0-255. Dessa talserier kan sedan användas för att konfigurera en nyckel över hur sändaren och mottagaren skall hoppa mellan frekvenser.

```

#include <stdio.h>
#include <math.h>

// demonstrerar slumpalsserier
// gcc -o slumptal slumptal.c

main()
{
  unsigned char seed=0; //start - bankkoden
  unsigned char m=255; //antal olika tal
  unsigned char a=3; //helst printal
  unsigned char c=7; //helst printal
  unsigned char i=0;
  unsigned char j=0;
  unsigned char slumptal=seed;

  while(j<3)
  {
    while(i<255)
    {
      i++;
      slumptal=(a*slumptal+c)%m;
      printf("%d;", slumptal);
    }
    printf("\n");
    i=0;
    j++;
    seed++;
    slumptal=seed;
  }
}

```

Figur 17. Koden för en slumpalsgenerator.

Frötalet bestämmer vilken talserie man kommer att använda. Byter man frötaal byter man också talserie. Talserien som fås av frötalet bestämmer frekvenshoppen. Om man har ett frötaal och en nyckel med 255 slumpmässiga tal mellan 1-255, vet man hur frekvenshoppen kommer att ske. Men dessutom behöver man även bestämd tid för när frekvenshoppen börjar, för om man endast snappade upp informationen på en frekvens vid en tidpunkt vet man inte med säkerhet till vilken frekvens man nästa gång bör hoppa. Detta för att samma frekvens kan förekomma flera gånger i en frötaalsekvens.

Om vår serie t.ex. är 25/203/54 osv. kommer bärvågsfrekvensen att multiplicera 100 kHz med i fråga varande tal, i detta fall blir resultatet för dessa tre första talen i den 255 tals långa serien 2,5 MHz/20,3 Mhz/5,4 MHz osv. Dessa är frekvenser som kommer att läggas till den lägsta möjliga frekvensen.

Om det tillgängliga frekvensbandet är t.ex. 88-108 MHz, skulle 20 MHz delat med 255

st kanaler ge en kanalbredd på ca 78 kHz vilket inte är särskilt praktiskt. Om man i stället har ett minsta frekvenshopp på 100 kHz skulle 255 olika hopp kräva en bandbredd på 25,5 MHz.

Eftersom kanalbredden för FM-bandet är 200 kHz, kunde en avlyssnare med bara en radiomottagare i teorin kunna snappa upp två efterföljande frekvenshopp då $\Delta f < 100$ kHz. Avlyssnaren kunde trots allt inte veta vilken kanalfrekvensen är, dvs. det vore svårt att kunna avgöra om det som sänds är en etta eller en nolla, eftersom det för detta krävs att den exakta kanalfrekvensen för just det hoppet är känd. Om Δf är 100 kHz eller lägre (låt oss säga 50 kHz), och ett frekvenshopp mot förmodan är efterföljande frekvenser (100,05 och 100,10) - vilket är möjligt men osannolikt - skulle en avlyssnare som snappat upp dessa bägge signaler inte med säkerhet veta vilken kanalfrekvensen var, dvs. avlyssnaren skulle nog höra något men vet inte vad det är den hör.

För att bedöma chansen att ett avlyssningsförsök ändå lyckas, kan det konstateras att det krävs att ett sannolikt mönster lyckas avlyssnas och sedan faktiskt uppträder. Om hopp sker efter varje bit är detta praktiskt taget hopplöst. Säkerheten för algoritmen är hög då hoppmönstrets framtid bara kan gissas om ett antal historiska värden är känd. I analysen används ett enkelt 8 bits mönster. Detta innebär att hoppen upprepar sig redan efter 255 tecken eller bitar, om nyckeln inte ändras. Man kan undersöka dessa 255 olika hoppmönster på likheter i hoppvärdena för att avgöra om något av hoppmönstren är mera sannolikt att hittas på slump. Detta kallas för mönsteranalys.

Insvängningstiden för evalueringsbrädets frekvenssyntetisator ADF4360-8, beror på hur den ställts in. Med mjukvaran är det möjligt att bestämma antalet cykler som krävs för att frekvenssyntetisatorn skall anse sig vara låst. Denna kan väljas till tre eller fem cykler, i arbetet används fem. I ADF4360-8 är låsdetektionen digital (DLD, digital lock detection). Frekvenssyntetisatorn övervakar fasfelet i fasfrekvensdetektorn (Phase Frequency Detector, PFD) och använder en tolerans på 15 ns för att bestämma om den är låst till en frekvens. Då fasfelet för fem efterföljande cykler är inom 15 ns toleransen, skickar frekvenssyntetisatorn ut en etta.

Frekvenserna som används i arbetet är 88-108 MHz. 100 MHz skulle leda till att en

svängning tar 10 ns, vilket då är den teoretiska maximihastigheten för frekvensbyte vid 100 MHz.

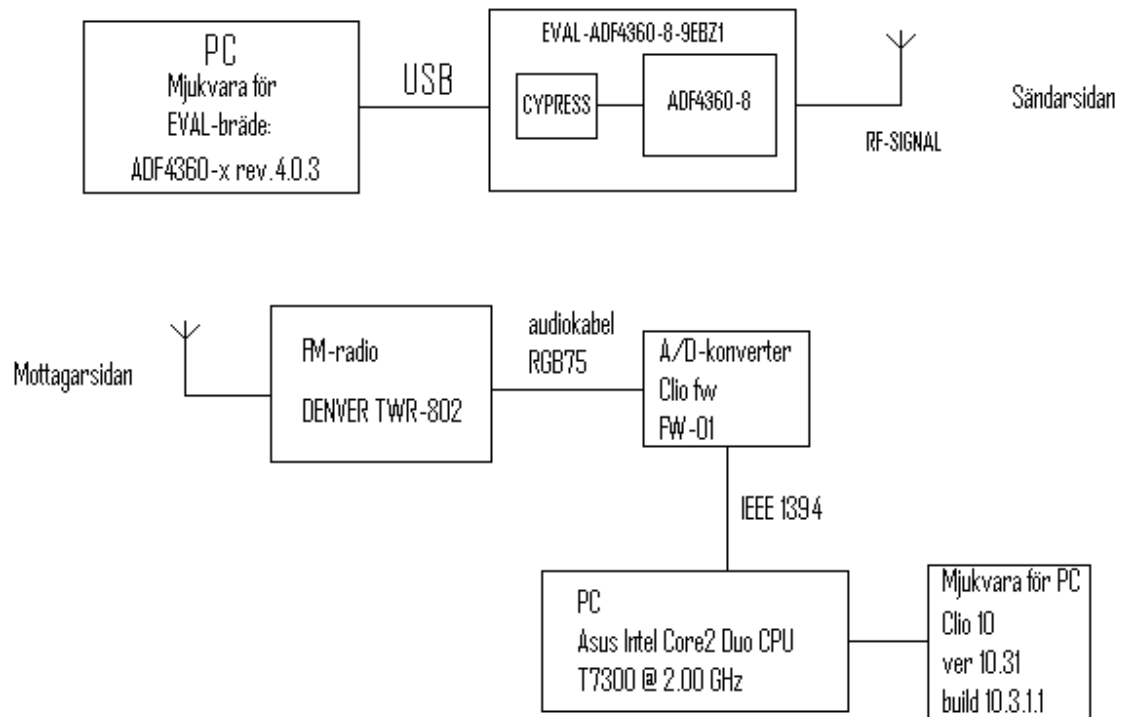
Använder man konstant samma nyckel, kan man byta bärvågsfrekvens efter varje bit, men man bör senast byta efter 255 bitar. Om man skulle använda sig av en längre bitlängd än 255 skulle slumpalsserien börja upprepa sig och sändningen bli lättare för en utomstående att avlyssnas.

Bärvågsfrekvensen hoppar enligt det slumpalssgenererade mönstret. Då en frekvens hoppar från t.ex. 100,0 MHz till 105,0 MHz, ändrar sig förstås även frekvensen för en nolla respektive etta. Minimihopp för bärvågsfrekvensen rekommenderas vara 100 kHz, eftersom ett ”kortare” hopp i praktiken skulle underlätta avlyssning.

Antalet nycklar kunde i praktiken väljas mellan 1 och 255 st. Man kunde t.ex. tänka sig att varje veckodag har sin egen nyckel, vilket förstås då skulle leda till sju stycken olika nycklar eller frekvenshoppsmönster. Ju oftare detta mönster byts desto svårare blir avlyssningen.

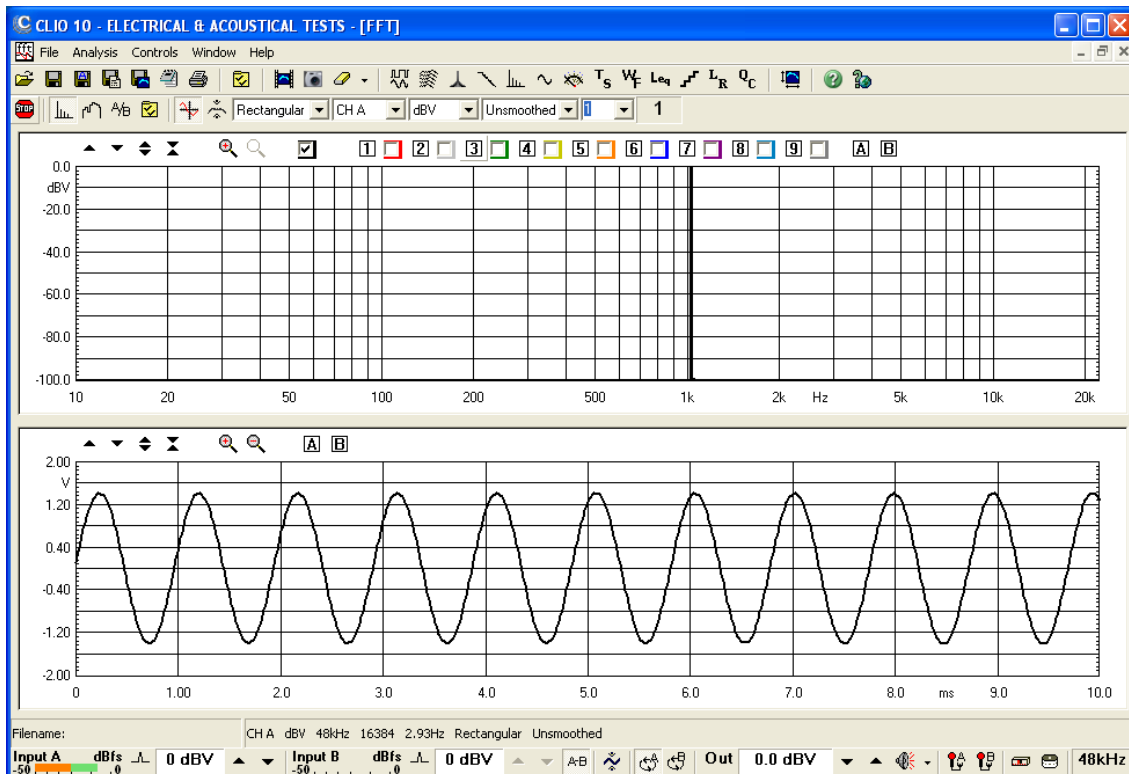
4. RESULTAT

Experimentet genomförs med utrustning enligt tabell 1. På sändarsidan kopplas evalueringsskivan via en USB-kabel till en dator med evalueringsskivans mjukvara installerat. Mottagarsidan består av en enkel FM-radiomottagare som via en audiokabel är kopplad till en A/D-konverter, varifrån en utsignal körs via en seriell högkommunikationskoppling av typen IEEE 1394 till en bärbar dator. På denna dator är mjukvaran Clio 10, för analysering av bl.a. smalbands FFT (Fast Fourier Transform), installerad.



Figur 18. Blockschemat för experimentet.

Frekvenshoppet programmeras via mjukvaran för evalueringsbrädet. I experimentet görs frekvenshopp från 100,0 till 100,1 MHz, dvs. det minsta frekvenshoppet som kan rekommenderas vid kryptering då man vill försvåra avlyssning.



Figur 19. Användargränssnitt för Clío 10 [12].

Det visar sig att digital FM-modulation kan skapas med evalueringsbrädet med hjälp av dess mjukvara. Denna utgör då en mjukvarudefinierad radio. Radion använder frekvensbandet 88-108 MHz, som endast kan ändras genom att ändra på induktanserna på evalueringsbrädet.

Radions egenskaper kan utökas om PC mjukvaran kan ändras så att frekvenshopp mellan flera olika frekvenser vore möjligt, vilket skulle tillåta en enkel kryptering av det som sänds.

Mätningarna utförs med en FM-mottagare som är kopplad till en enhet som ändrar den inkommande analoga signalen till en digital för att kunna presenteras i mjukvaran CLIO10. Enheten är kopplad via en IEEE-1394 länk till datorn.

Experimentet görs genom att mäta den verkliga fördröjningen av den dekodade signalen. med evalueringsbrädet samt mjukvaran Clío10. Resultaten för olika fördröjningar mäts upp och jämförs med det teoretiska värdet för frekvenshoppet. På så

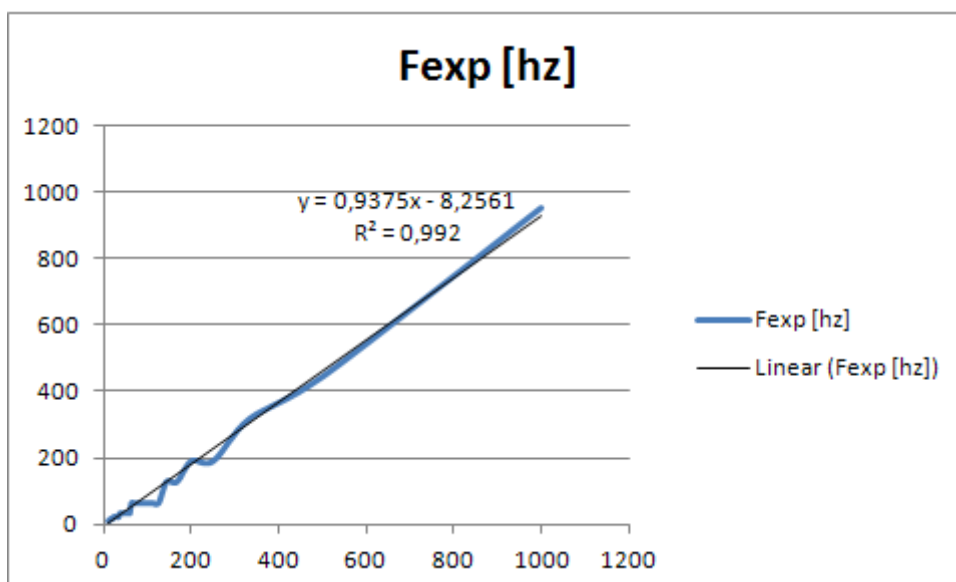
vis kommer vi förhoppningsvis att kunna räkna ut hur stort felet från hårdvaran är och vad som påverkar felet. Totalt görs 48 mätningar och den verkliga frekvensen läses ut med hjälp av mjukvaran Clio10. All uppmätt data har skrivits in i nedanstående tabell, och uträkningar gjort med programmet Excel.

delay [ms]	fteoretisk [hz]	fexp [hz]	delta	exp/teor	ref.error [%]
1	1000	954	46	0,9540	4,6
2	500	445	55	0,8900	11,0
3	333	318	15	0,9540	4,6
4	250	190	60	0,7600	24,0
5	200	190	10	0,9500	5,0
6	167	127	40	0,7620	23,8
7	143	127	16	0,8890	11,1
8	125	63	62	0,5040	49,6
9	111	63	48	0,5670	43,3
10	100	63	37	0,6300	37,0
11	91	63	28	0,6930	30,7
12	83	63	20	0,7560	24,4
13	77	63	14	0,8190	18,1
14	71	63	8	0,8820	11,8
15	67	63	4	0,9450	5,5
16	63	63	0	1,0000	0,0
17	59	32	27	0,5440	45,6
18	56	32	24	0,5760	42,4
19	53	32	21	0,6080	39,2
20	50	32	18	0,6400	36,0
22	45	32	13	0,7040	29,6
24	42	32	10	0,7680	23,2
26	38	32	6	0,8320	16,8
28	36	32	4	0,8960	10,4
30	33	21	12	0,6300	37,0
32	31	21	10	0,6720	32,8
34	29	21	8	0,7140	28,6
36	28	21	7	0,7560	24,4
38	26	21	5	0,7980	20,2
40	25	21	4	0,8400	16,0
42	24	21	3	0,8820	11,8
44	23	21	2	0,9240	7,6
46	22	21	1	0,9660	3,4
48	21	16	5	0,7680	23,2
50	20	16	4	0,8000	20,0
54	19	16	3	0,8640	13,6

58	17	16	1	0,9280	7,2
62	16	16	0	0,9920	0,8
66	15	13	2	0,8580	14,2
70	14	13	1	0,9100	9,0
74	14	13	1	0,9620	3,8
78	13	13	0	1,0000	0,0
82	12	11	1	0,9020	9,8
86	12	11	1	0,9460	5,4
90	11	11	0	0,9900	1,0
94	11	9	2	0,8460	15,4
98	10	9	1	0,8820	11,8
100	10	9	1	0,9000	10,0

Tabell 3. Resultatet från mätningen.

Resultatet visar att denna utrustning lämpar sig för användning som mjukvarudefinierad radio. Konstateras bör, att fördröjningen som huvudsakligen härstammar från kommunikationen mellan USB och frekvenssyntetisatorns chip, leder till ett frekvensfel på ca 8 Hz som bör kompenseras. Dessutom är den frekvenssyntetisatorn inte heller så snabb som påstås.



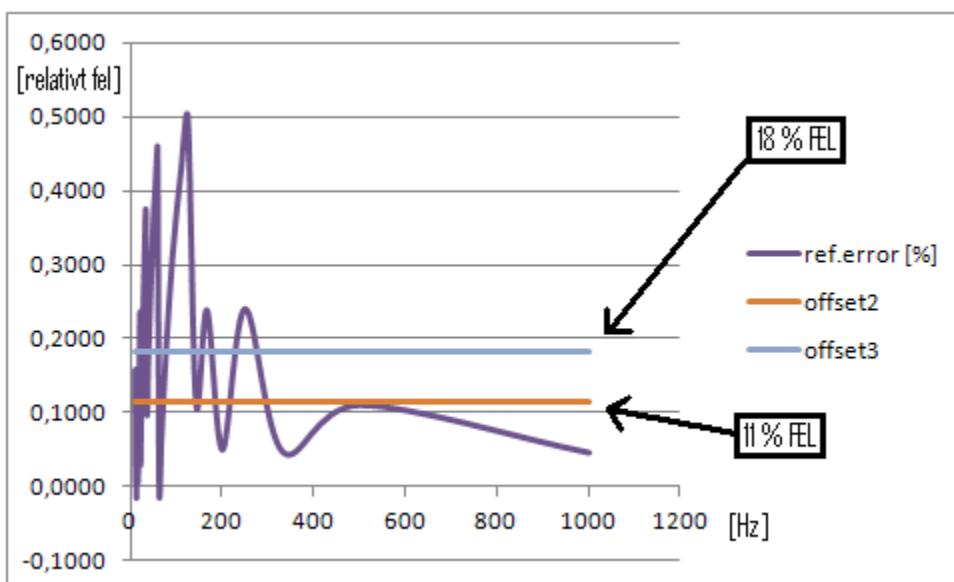
Figur 19. Den experimentella frekvensens korrelation med den teoretiska frekvensen.

Som resultatet i figur 19 ovan visar, motsvarar signalen från sändarsidan inte den teoretiska frekvensen som fördröjningen skulle motsvara. Detta beror på att det finns

restriktioner i hårdvaran. Utan dessa restriktioner, skulle resultatet teoretiskt vara en rät linje där frekvensen direkt motsvarar tiden för fördröjningen mellan frekvenshopp, dvs. sträckan i figur 19. vore $y = x$. Om resultatet från mätningarna räknas ut och om de sedan görs till en linjär form fås en sträcka med ekvationen $y = 0,94x - 8,25$ vilket betyder att fördröjningens inverkan vid låga frekvenser är stor men på högre frekvenser mindre. Detta framgår tydligare i figur 20.

Determinationskoefficienten anger hur stor del av variationerna i y som kan förklaras med ändringar i x . Ur figur 19 kan man se att den är 99,2 %, vilket betyder att determinationskoefficienten är hög, eftersom mindre än 1 % av variationerna i y beror på annat som den matematiska modellen inte kan förklara.

Om man analyserar resultatet ur ett annat perspektiv, d.v.s. hur det relativa felet korrelerar med den experimentella frekvensen (figur 20) kan man se att felet tydligt är relativt större för lägre frekvenser och vice versa. Ur figuren nedan kan man urskilja att frekvenser lägre än ca 300 Hz ger ett större fel än högre frekvenser. Frekvenserna under 300 Hz har ett fel på omkring 18 % medan frekvenserna över 300 Hz ligger närmare ett fel på 11 %.



Figur 20. Relativa felets korrelation med experimentella frekvensen.

5. SAMMANDRAG

Mjukvarudefinierad radio med ovan nämnda prestanda kunde användas för att sända meddelanden i form av korta textmeddelanden eller telemetridata.

Före mätningarna var det klart att ett fel skulle uppstå mellan den teoretiska frekvensen och den experimentella. Det intressanta i experimentet var att reda ut hur stort detta fel är och om det finns någon regel för felet. Om denna utrustning skulle användas för mjukvarudefinierad radio med inbyggd kryptering, borde detta fel vara noggrant uppmätt och dessutom bör man kompensera det. Mätningarna visar att ett väntat fel uppstår som huvudsakligen beror på kommunikationsfördröjningar mellan evalueringsbrädets cypress-kontroller och chip. Frekvensfelet som uppstår är 8 Hz då fördröjningen mellan frekvenshoppen ligger mellan 10 och 1000 Hz. Detta betyder att det relativa felet för frekvenser under låga frekvenser (<200 Hz) är stora, men den testade hård- och mjukvaran kunde trots allt användas för mjukvarudefinierad radio, eftersom fördröjningen för en möjlig kryptering inte skulle behöva vara så snabb.

Om man kunde styra evalueringsbrädet direkt via dess mikrokontroller, vilket enligt tillverkaren också är möjligt, skulle man slippa de restriktioner som den nuvarande kommunikationen mellan USB och frekvenssyntetisator-chip har. Då skulle man inte över huvud taget använda sig av evalueringsbrädets mjukvara, utan den i hårdvaran inprogrammerade mjukvaran för evalueringsbrädet.

Källor / References

1. <http://sv.wikipedia.org/wiki/Multiplexer>
2. http://en.wikipedia.org/wiki/Frequency-shift_keying
3. Arcada, Kaj Grahns kursmaterial i telekommunikationselektronik 2006
<http://people.arcada.fi/~grahn/KE2000/K1.doc>
4. <http://sv.wikipedia.org/wiki/Superheterodynmodulator>
5. http://sv.wikipedia.org/wiki/Fasl%C3%A5st_slinga
6. http://www.wirelessinnovation.org/benefits_of_sdr
7. <http://fi.wikipedia.org/wiki/Ohjelmistoradio>
8. <http://sv.wikipedia.org/wiki/Kryptering>
9. http://en.wikipedia.org/wiki/Pseudorandom_number_generator
10. Datablad för ADF4360-8, Analog Devices
(http://www.analog.com/static/imported-files/data_sheets/ADF4360-8.pdf)
11.
http://www.tml.tkk.fi/Opinnot/T110.2105/2007/Forelasningar/02.Fysiska_skiktet.pdf
12. <http://www.audiomatica.com/download/clioman10.pdf>

12. <http://www.audiomatica.com/download/clioman10.pdf>
13. Banerjee, Dean. 2006, PLL Performance, Simulation and Design, 4 uppl., Dog Ear Publishing LCC, 344 s.
14. <http://www.radio-electronics.com/info/rf-technology-design/fm-reception/fm-pll-detector-demodulator-demodulation.php>