"Apuntes sobre "Radio Definida por Software SDR"

Por

XE1YDC Diego Varela

01 de Junio 2016 en el "Boletín Tecnológico ARJAC"

Radio Definida por Software (Software Defined Radio SDR) es un aspecto del dominio de procesamiento digital de señales (Digital Signal Processing DSP) donde señales de radio son digitalizadas y procesadas para obtener modulación o demodulación de los distintos modos. Para comenzar a entender SDR es necesario primero tener nociones de DSP.

Apuntes sobre Radio Definida por Software

Compiladas por Ing. Pedro E. Colla – LU7HZ
Radio Club Córdoba
Córdoba-Argentina

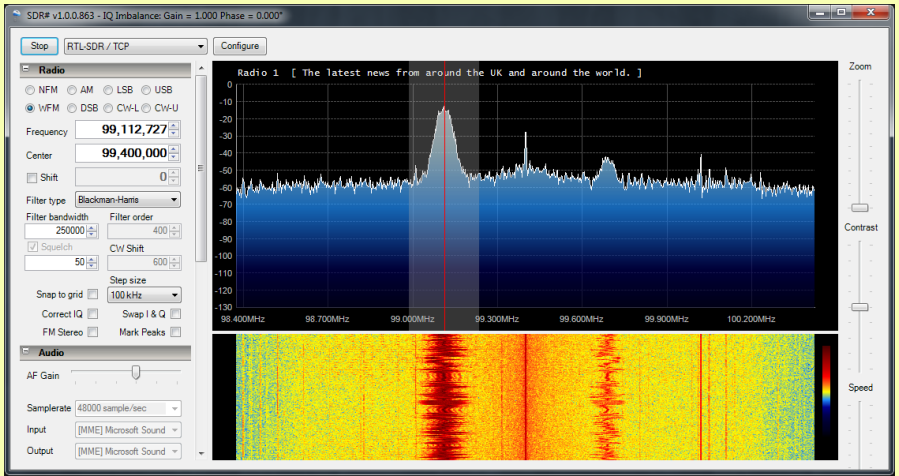
Radio Definida por Software

Introducción.

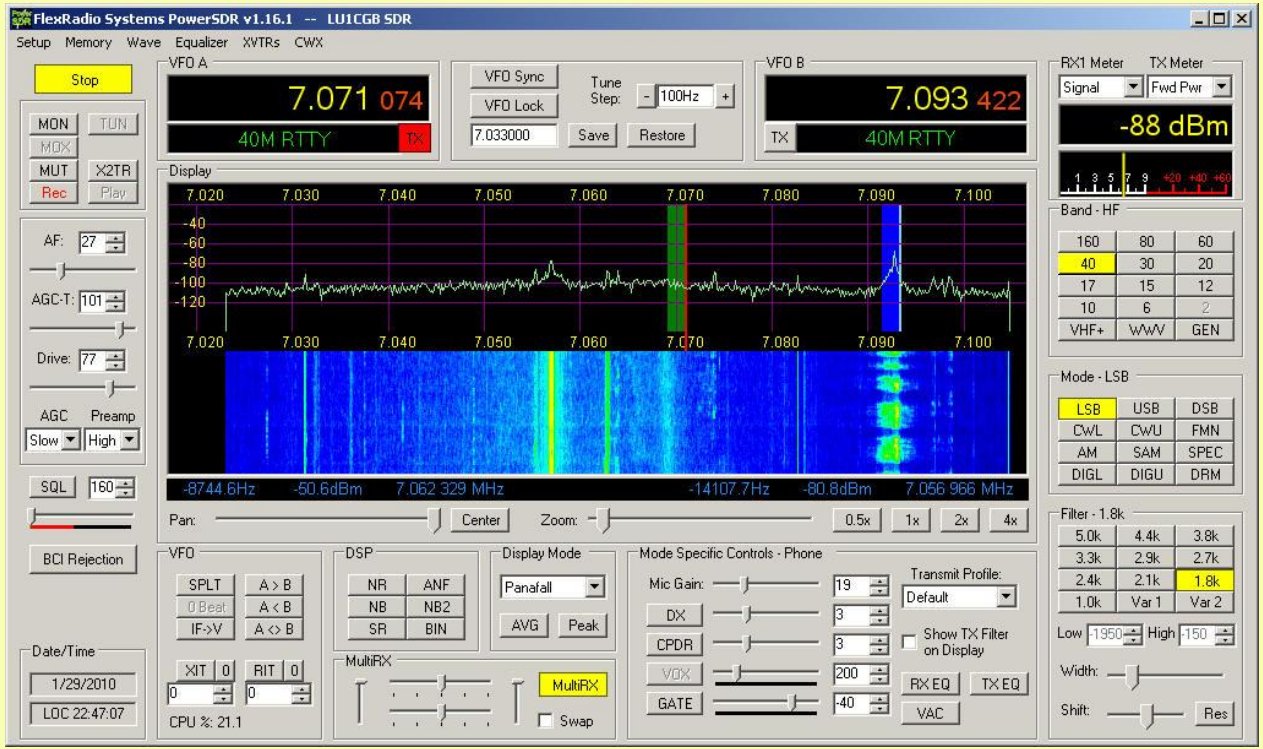
Radio Definida por Software (*Software Defined Radio SDR*) es un aspecto del dominio de *procesamiento digital de señales* (*Digital Signal Processing DSP*) donde señales de radio son digitalizadas y procesadas para obtener modulación o demodulación de los distintos modos. Para comenzar a entender SDR es necesario primero tener nociones de DSP.

Sigo encontrando sorprendente que uno de los mejores tutorials sobre DSP en el campo profesional sea el contenido en el Capítulo 16 del *ARRL Radio Amateur Handbook*; si bien hay libros mejores y mas completos es sorprendente la sintesis de los aspectos relevantes que contienen estas pocas páginas y por lo tanto su lectura es muy recomendada.

Desafortunadamente este tutorial está en inglés lo que limita su difusión. Para superar ese problema hace algún tiempo traduje una introducción al DSP disponible en Internet en el site http://www.bores.com que cubre los aspectos mas relevantes del proceso de digitalización y procesamiento de señales en general.



Hay por supuesto libros a cualquier profundidad sobre el tema, uno que encuentro tiene un buen equilibrio entre disponibilidad gratuita y profundidad de cobertura es el libro del Dr. S.W.Smith. Este libro cubre en general todos los aspectos del procesamiento digital de señales (DSP) del cual solo una parte será de utilización directa en SDR. Otro tutorial también completo es el disponible en el site de la Dra. Langton.



SDR para Radioaficionados

Los usos del procesamiento digital de señales (DSP) en radio son múltiples y no son recientes; aplicaciones destinadas a operar con modos digitales (CW, RTTY, SSTV, Packet y otros) utilizando la placa de sonido de la PC han estado disponibles en grado creciente de sofisticación por al menos una década en la mayor parte de las plataformas de PC (Windows y Linux mayormente).

Basicamente todos estos usos implican tomar la señal de audio del receptor y procesar la codificación contenida en esta para reconstruir el mensaje digital originalmente codificado en este por el proceso, a menudo inverso, en el otro extremo de la comunicación.

La cantidad de paquetes disponibles para los distintos modos es enorme, algunos de los más comunes pueden verse en el este enlace de Internet del Dr. Oliver Welp, DL9QJ (en inglés y alemán,…)

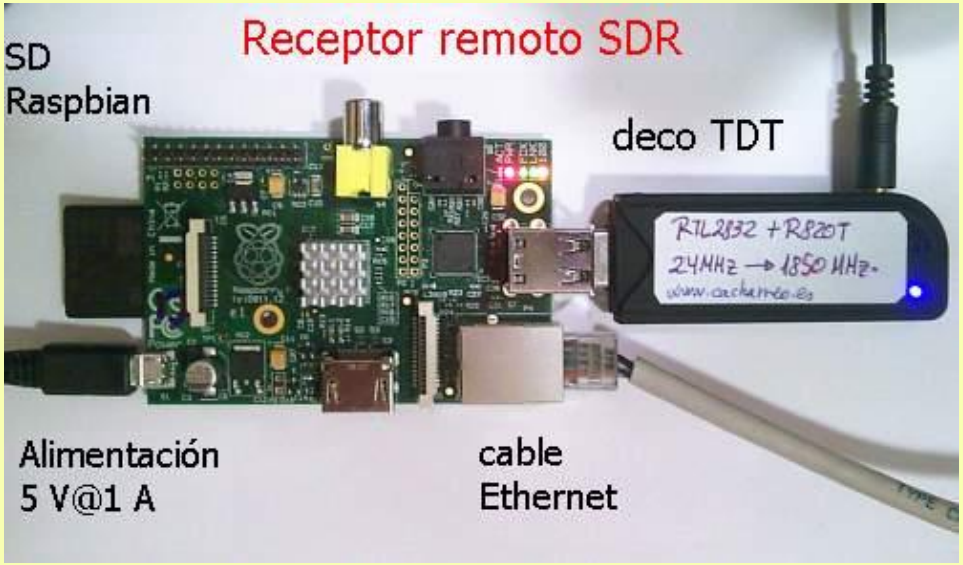
En estos usos existe una doble modulación de la señal, primero a nivel del audio y luego el resultado contenido en una banda base donde es modulada de alguna forma pre-definida (tipicamente SSB o FM); si bien se utiliza procesamiento digital de señales para recuperar la modulación contenida en el audio (demodularla) el procesamiento de de la banda base se sigue haciendo en el receptor. El proceso de generación de las señales es exactamente el inverso donde es común el uso de DSP para generar la modulación a nivel de audio pero luego esto es inyectado en una banda base que es procesada y emitida por un transmisor convencional.

Quizás con algo de arbitrariedad no se considera a estos usos como Software Defined Radio.

Que es entonces Software Defiined Radio? Hay varias definiciones formales pero coloquialmente se trata de tomar una señal de RF en bruto, digitalizarla y realizar todo el proceso de demodulación de la misma mediante técnicas de DSP.

Idealmente se tomaría la señal directamente desde la antena al dispositivo que digitaliza y de ahí en adelante todo el procesamiento es digital. Si bien esto es teóricamente posible y en algunos casos a los costos correspondientes también factible en la práctica el enfoque tiene una serie de problemas relacionados con los niveles de señales involucrados (en transmisión por altos y en recepción por bajos).

Una primera visión introductoria sobre SDR puede verse en el artículo breve de Marcelo Franco (N2UO) también disponible en su sitio Web adjunto



Algunas limitaciones en la práctica

Cuando tratamos de implementar la definición “teórica” de SDR enfrentamos un número de problemas; los dos principales son la velocidad de muestreo y la capacidad de manejo de señales (altas o bajas) del dispositivo digitalizador; ya se há visto que el procesamiento digital de señales introduce sus propios problemas y distorsiones en la señal que no existen en su contrapartida analógica.

Los limites expresados por el criterio de *Nyquist* (visto en la sección anterior) donde la velocidad de muestreo debe ser el doble o más que la mayor frecuencia de la señal a procesar definen las caracteristicas plausibles de la señal a ser procesada.

Por lo tanto una placa de sonido standard puede procesar una frecuencia de muestreo de 48Khz por lo que podrá manejar señales que tengan hasta unos 24 Khz de ancho de banda, algunas placas de sonido especiales pueden muestrear a 96 Khz o incluso más por lo que su capacidad de manejar anchos de banda termina siendo del orden de los 50 Khz.

Notese que los anchos de banda típicamente manejados en aplicaciones de aficionado oscilan entre unos pocos Hz (CW) hasta unos 6 Khz (FM de banda angosta) pasando por unos 3 Khz en el caso de SSB; incluso señales de AM o FM comercial están contenidas en un ancho de banda de 50 Khz o menos por lo que pueden ser procesadas por una placa de sonido de costo razonable.

Pero en general estas señales vienen moduladas en una banda base de mucho mayor frecuencia, por ejemplo una señal de CW en la banda de 40 mts ocupa unos pocos Hz de ancho de banda pero su frecuencia es de 7 Mhz, se necesitaría muestrear al doble de esa frecuencia (14 Mhz) para procesarla directamente desde la antena (supuesto que el digitalizador pueda capturar una señal tan baja como la disponible en la antena).

El costo de los dispositivos digitalizadores (llamados *conversores A/D* por “Analogico a Digital”) se incrementa con su velocidad de muestreo y existen límites tecnológicos en cuan rápido un digitalizador puede procesar una señal.

Para dar ordenes de magnitud un conversor profesional de baja gama puede procesar 2 Mhz (dos millones de muestras por segundo, capaz entonces de procesar señales de hasta 1 Mhz de ancho de banda) mientras que uno de alta gama puede manejar 20 Mhz (veinte millones de muestras por segundo, capaz entonces de procesar señales de hasta 10 Mhz de ancho de banda).

Un conversor profesional de baja gama podrá procesar señales muy complejas de telefonía celular pero se necesitará uno de alta gama para procesar una señal de TV convencional (analógica o digital), los costos de un conversor de este tipo con comparables a equipos de comunicaciones de alta gama (varios miles de dolares).

Existen dispositivos de muestreo avanzados que pueden ir más alto en su capacidad de muestreo hasta gamas de 100 Mhz o más (pudiendo entonces manipular señales de hasta 50 Mhz de ancho de banda) pero ni su costo ni su disponibilidad son accesibles facilmente por lo que se los considera en este trabajo fuera del alcance bajo discusión.

Notese que aún un conversor profesional de alta gama puede solo procesar directamente señales en una gama baja de frecuencias de HF (hasta la banda de 10 Mhz) por lo que no sería útil para operar en frecuencias más altas; pero al mismo tiempo notese que el conversor puede manejar cualquier señal comprendida entre 0 Hz (algo más en términos prácticos) y su frecuencia más alta (10 Mhz en este caso).

Una etapa mezcladora de RF puede llevar la frecuencia de la banda base desde niveles inmanejables para el digitalizador hasta frecuencias que le son manejables; por ejemplo, si a una señal de CW disponible en 7.020 Mhz la mezclamos con una señal de oscilador local de 7.000 Mhz obtenemos una señal de 14.020 Mhz (suma) y otra de 20 Khz (diferencia), filtrando la primera la segunda está dentro de la gama que un digitalizador económico tal como una placa de sonido común de una PC puede procesar.

La solución para el nivel de señales es agregar etapas de amplificación de RF, sea de señal debil en recepción como de alta potencia en transmisión (si bien es más comun usar SDR en recepción puede utilizarse tanto para demodular una señal como para modularla!).

Software Defined Radio en la Práctica

En general los diseños de SDR “prácticos” empiezan por definir el digitalizador disponible y construyen el sistema alrededor de el. Se alejan entonces de la teoría en cuanto a no capturar la señal desde la antena (o para el caso, desde su fuente de generación) sino que es sometido a uno o más pasos de amplificación y mezcla (conversión) para llevarla a anchos de banda asi como nivel de señal que pueda manejar el digitalizador disponible.

La implementación de SDR en usos prácticos tendrá entonces tres componentes;

Un dispositivo de hardware que toma la señal desde su fuente y la deja en un rango de frecuencia y amplitudes que el digitalizador puede manejar.

Un dispositivo que es capaz de tomar la señal y digitalizarla a una determinada velocidad máxima de muestreo, lo que a su vez define el ancho de banda máximo de la señal que es capaz de procesar.

Un software que es capaz de procesar la señal una vez que la misma fue digitalizada para operar sobre ella para obtener la señal demodulada final que se desea.

Software Defined Radio en la Práctica

En general los diseños de SDR “prácticos” empiezan por definir el digitalizador disponible y construyen el sistema alrededor de el. Se alejan entonces de la teoría en cuanto a no capturar la señal desde la antena (o para el caso, desde su fuente de generación) sino que es sometido a uno o más pasos de amplificación y mezcla (conversión) para llevarla a anchos de banda asi como nivel de señal que pueda manejar el digitalizador disponible.

La implementación de SDR en usos prácticos tendrá entonces tres componentes;

Un dispositivo de hardware que toma la señal desde su fuente y la deja en un rango de frecuencia y amplitudes que el digitalizador puede manejar.

Un dispositivo que es capaz de tomar la señal y digitalizarla a una determinada velocidad máxima de muestreo, lo que a su vez define el ancho de banda máximo de la señal que es capaz de procesar.

Un software que es capaz de procesar la señal una vez que la misma fue digitalizada para operar sobre ella para obtener la señal demodulada final que se desea. Implementar SDR en la práctica consistirá en elegir y hacer trabajar en conjunto estos componentes.

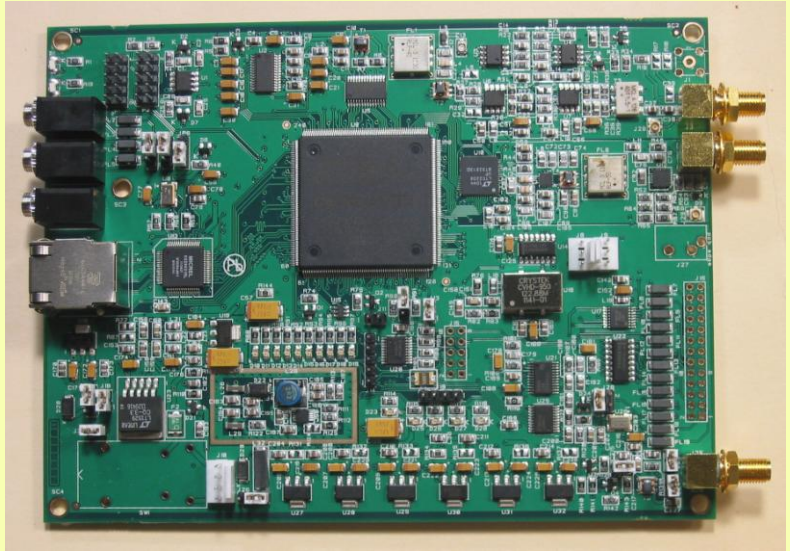
Tipos de SDR

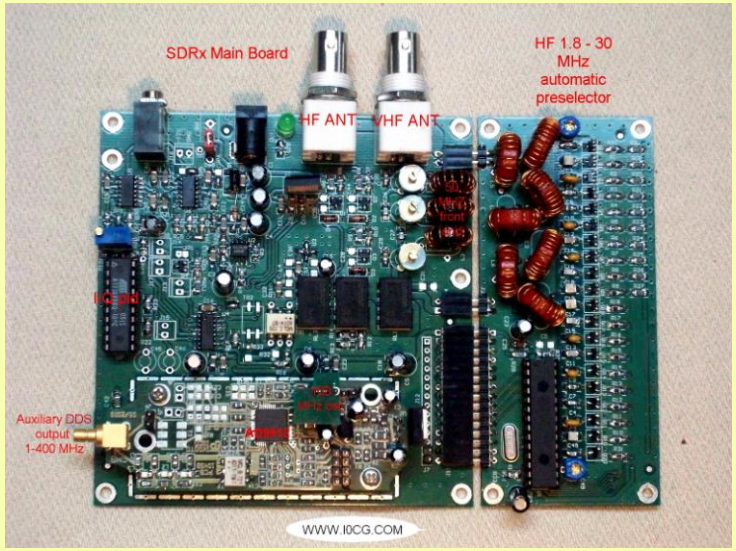
Me gusta la taxonomía de aplicaciones SDR que se utiliza en el sitio web de Christophe F4DAN donde el divide las distintas implementaciones de SDR según la siguiente clasificación:

Tipo I. Son en general implementaciones que implican la utilización de una placa de sonido convencional de una PC como digitalizador y software convencional de PC como elemento de procesamiento. Es probablemente el tipo que es más accesible para radioaficionados y por lo tanto el que tiene más interés para estos apuntes.
Tipo Ia. El SDR se implementa alimentando a la placa de sonido la salida de audio de un receptor convencional de comunicaciones, con alguna excepción esto comenté unos párrafos atrás que no es estrictamente SDR la mayor parte de las veces, pero como veremos hay algunas excepciones.
Tipo Ib. El SDR se procesa introduciendo a la placa de sonido una señal mono que representa una frecuencia intermedia de aproximadamente 12 Khz.
Tipo Ic. El SDR se procesa introduciendo a la placa de sonido una señal I+Q que representa una frecuencia intermedia en el rango de frecuencias que la placa de sonido puede manejar. Este es el tipo quizás más potente, con mejor relación costo-prestación y de mayor atractivo para los aficionados en el estado actual de tecnología.
Tipo Id. El SDR se procesa introduciendo una señal I+Q en un digitalizador y procesador de señales especializado (no una placa de sonido).

Tipo II. El SDR se implementa con un dispositivo especial que captura la señal desde la antena y la procesa a partir de allí.
Tipo III. El SDR se implementa con un dispositivo especial que captura la señal desde una IF analógica y la procesa a partir de alli.
Tipo IV. El SDR es implementado por receptores especiales que toman la señal directamente desde su fuente en la frecuencia de trabajo y la procesan en toda la cadena.
Tipo V. SDR online, es un SDR que está implementado en un servidor capaz de proveer parcial o totalmente la capacidad de procesamiento digital de señales, tiene cierta utilidad práctica para radioaficionados por lo que lo veremos con algún detalle.

Solo veremos con algun detalle los **Tipos I** y **V** que son los que mayor interés práctico pueden tener para radioaficionados mientras que los restantes pueden ser consultados a partir de los aplicativos y tutoriales disponibles en el site de Christophe.





Tipo Ia-Procesamiento de Audio

Consiste en tomar el audio de un receptor (o generar el audio para un transmisor, dependiendo del sentido) a partir de operar utilizando una placa de sonido; esta modalidad viene siendo usada por radioaficionados por al menos una década para generar y decodificar CW, RTTY, SSTV, Packet y más recientemente modos digitales como PSK31. También puede utilizarse para decodificación de modos no tan frecuentes en radioafición como DRM (Digital Radio Mondiale), señales APT de satelites metereológicos y modos digitales especiales por ejemplo.

Como dijimos antes esto no es estrictamente SDR porque la señal ya está procesada previamente por el transceiver convencional y demodulada desde su banda base por medios convencionales.

Sin embargo es bueno tenerlo en cuenta porque al utilizar dos de los tres componentes necesarios para SDR (la placa de sonido como digitalizador y el software que procesa este resultado) es un excelente punto de entrada al tema.

Bajo algunos escenarios es algo más en realidad. Si el receptor tiene un ancho de banda de FI y audio grande (digamos unos 20 Khz) y es capaz de recibir en SSB el resultado es indistinguible de un amplificador y mezclador como los que veremos en otros tipos y con limitaciones puede utilizarse como front-end; esto se puede hacer facilmente con receptores digitales como el ICOM PCR-1000 o similares por ejemplo. El ancho de banda será muy pequeño, del orden de 10-12 Khz pero podrá experimentarse SDR con hardware disponible y familiarizarse con los programas.

Tipo Ib-Procesamiento de FI de 12 Khz.

En este modo de procesamiento se dispone de una cadena de RF que es capaz de tomar la señal de la antena, amplificarla y mezclarla hasta obtener una FI de 12 Khz (o, en el caso de transmisión de partir de una señal en 12 Khz y mezclarla/amplificarla hasta su frecuencia final).

Notese que un receptor convencional típicamente tendrá un punto en la cadena (justo antes de demodular) donde habrá una señal de frecuencia intermedia de estas caracteristicas; pero desafortunadamente esta estará normalmente por encima de las frecuencias que la placa de sonido puede manejar (por ejemplo 455 Khz o 10.7 Mhz). Sin embargo es posible introducir un mezclador adicional en ese punto que introduzca un cambio de frecuencia adicional al rango deseado.

Otra forma de obtener el mismo resultado, quizás en forma más sencilla, es utilizar un diseño simple de conversión directa donde la frecuencia de entrada es mezclada con un oscilador local en una frecuencia próxima (de forma que la diferencia entre ambos sea 12 Khz).

Históricamente este es el enfoque que primero se utilizó y posiblemente el más accesible porque sea que se parta de un receptor existente o que se haga un diseño simple de conversión directa son ambos enfoques muy al alcance de los aficionados, incluso aquellos con menores habilidades constructivas o disponibilidad de instrumental.

Hay varias referencias de este enfoque (el que también se utiliza para otros usos tales como decodificación de DRM) en la página de F4DAN de las cuales el que más me gusta en su enfoque experimental es el de Takafumi JI3GAB quien hace una implementación muy cruda de un transceiver de SSB en 12 Khz en Java; como casi todo el material de SDR el mismo está en inglés por lo que hice la traducción para ponerlo al alcance de más gente.
El enfoque de Takafumi es quizás obsoleto en su concepción pero es extremadamente didáctico y pueden apreciarse en el los principales bloques constructivos que se utilizarán en implementaciones más recientes; por eso será mencionado varias veces como referencia en las secciones siguientes de estos apuntes.

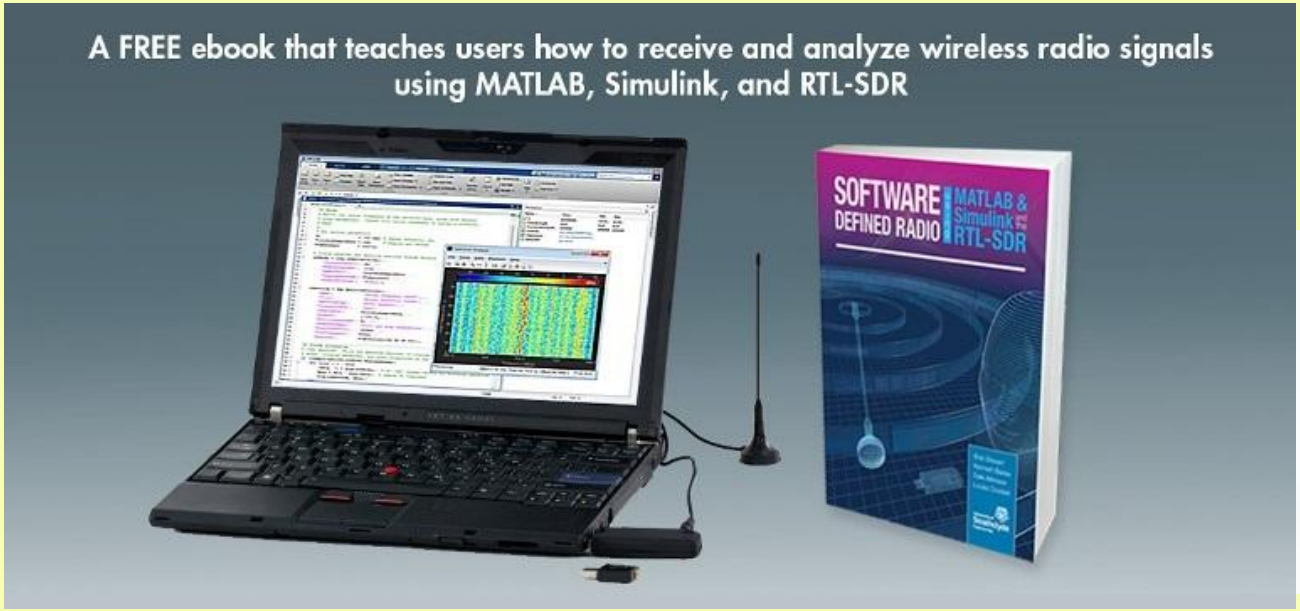
Al final de la traducción se agregan un par de diseños simples de conversores de FI convencional de 455 Khz a 12 Khz de bajo costo con lo que este enfoque puede ser experimentado a muy bajo costo y simplicidad constructiva.

Un experimento no muy util, pero al menos divertido, es implementar el software de Takafumi y alimentar la placa de sonido con la salida de audio de un transceiver con ancho de banda suficiente (unos 20 Khz); podrán demodularse tanto señales de AM, como SSB con facilidad y será bien evidente que lo está haciendo el software y no el receptor porque la sintonía indicará 12 Khz de diferencia con donde está la señal en realidad si la estuvieramos demodulando con el receptor.

Si se disponen dos PCs (o una PC con dos placas de sonido) otro experimento también ilustrativo consiste en escuchar la salida del programa “transmisor” en SSB (por ejemplo con auriculares a la salida de la placa de sonido) y verificar que el audio que genera no es inteligible; sin embargo cuando se alimenta a la entrada de otra placa de sonido (en otra PC) sobre la cual está corriendo el programa receptor puede verse que la señal es demodulada perfectamente y que oimos con claridad el audio. Se trata en definitiva de un excitador de SSB de 12 Khz y nada impediría mediante amplificación y conversión llevar esa señal al rango de frecuencias que nos haga sentido. De hecho Takafumi incluye en su artículo como modificar un FT747 con ese propósito (***cosa que por ninguna razón se recomienda hacer!!!***).

Pero el enfoque de este tipo de SDR tiene un fallo fundamental de base, la implementación tiene los mismos problemas de pobre rechazo de imagen que tienen los receptores de conversión directa a los que tanto se le parecen; la frecuencia deseada tan cercana a la del oscilador local hace que tanto la señal suma como la diferencia estén presentes en la salida y no es facil filtrar una de otra; los diseños tienen un comportamiento pobre en este sentido.

Con espiritu de radioaficionado y personalmente habiendo operado receptores de conversión directa de distinto tipo por décadas también sé que es un diseño que es confiable para comunicaciones la mayor parte del tiempo (excepto con las bandas muy pobladas o en condiciones de competencia o aperturas extremadamente significativas) por lo que se puede hacer mucho con ellos a costos y complejidades constructivas mucho más accesibles que diseños más complejos.



Tipo Ic-Procesamiento de Señales I+Q

Los problemas del Tipo Ib visto anteriormente se abordan y solucionan satisfactoriamente recurriendo al uso de señales "*complejas*"; la complejidad no reside en su "*dificultad*" sino en que se trata de una señal donde se informa tanto amplitud como fase y por lo tanto se la expresa como un número complejo (si bien la matemática es un tanto áspera es accesible con conocimientos matemáticos de nivel secundario).

La discusión sobre en que consiste una señal I/Q y porque exactamente el tenerla hace mucho más faciles las cosas en SDR es un tema que es explorado con cierto detalle en el siguiente artículo de *Ireland y Harman* que hé traducido para hacerlo más accesible

En el mismo se discuten con cierto detalle los algoritmos para demodular tanto AM y SSB como PSK y FM asi como la ventaja fundamental de disponer de señales I/Q y como las mismas son generadas.

En la retrospectiva puede verse que en la implementación de Takafumi de la sección anterior el utilizaba las señales I/Q generadas totalmente en software porque las mismas son al final necesarias para implementar los algoritmos de decodificación. No obstante en su caso se partia de una misma señal donde las relaciones de fase e imágenes espurias de mezcla ya estaban incorporadas.

El objetivo de los mezcladores con salida I/Q será entonces preservar la información de amplitud y fase ANTES de realizar la mezcla, de tal manera que pueda operarse matemáticamente luego con esa información.

Otra fuente de referencia, lamentablemente en inglés, a menudo citada donde se explica con cierto detalle las señales I/Q (complejas) y porque el tenerlas facilita la implementación es el artículo de Richard Lyons adjunto.

Dado que detectar los componentes I/Q de una señal, introducirlos en la placa de sonido stereo (para poder procesar las señales I y Q cada una en un canal distinto) y procesarlas con software apropiado de PC es la implementación *state-of-the-art* preferida por aficionados me detendré en diferentes enfoques de grado creciente de dificultad de implementación asi como algunas consideraciones de tipo práctica.

Conceptos Básicos

Las aplicaciones de SDR en radio han sido descriptas extensivamente en numerosas fuentes, la realizada por *Gerald Youngblood (AC5OG)* en una serie de números de la revista QEX sigue siendo una de las que encuentro más detalladas y accesibles, la Parte 1 de esa serie donde se cubren algunos aspectos básicos se adjunta a continuación:

Si bien algunos de los conceptos, en especial aquellos referidos al componente de software del SDR, han sido superados por mejores implementaciones la descripción conceptual y del hardware involucrado sigue siendo la base sobre la que se construyen todas las implementaciones de SDR. La serie completa puede obtenerse aquí.

Otra serie de articulos, aparecidos también en la revista QEX que vale la pena referir son los escritos por el autor del paquete LinRad, *Leif Asbrink (SM5BSZ)* quien no solo cubre los aspectos de implementación con buen grado de detalle sino que también se aboca a describir al SDR desde una perspectiva más sistémica en cuanto a su uso a aplicaciones concretas de radioaficionado; el hecho que el mismo sea un entusiasta practicante de EME (Tierra-Luna-Tierra) sesga fuertemente sus diseños para que sean de alta performance y bajo ruido. Un artículo relevante a las consideraciones de diseño de software puede verse en la parte 3 adjunta:

La serie completa de articulos puede obtenerse aquí.

Implementación de Gintaras (LY1GP)

La implementación hecha por Gintaras está profusamente documentada en su Web site; desafortunadamente el mismo se encuentra en inglés por lo que hé traducido algunos segmentos que resultan relevantes a esta introducción y que serán de utilidad para discutir distintos aspectos de la implementación práctica de sus diseños.

En este material se observan dos implementaciones, la denominada TinySDR (SDR diminuto) y ZeroSDR, cada una con sus particularidades en las que nos detendremos.

El TinySDR es probablemente la implementación más simple de una interfaz SDR que hé visto; tiene circuitos sintonizados y posiblemente requiere algún cuidado en su ajuste para funcionar lo que lo hace más dificil de construir.

Sin embargo es posible construirlo por prácticamente nada en términos de costo de materiales y el circuito es realmente muy simple.

En esencia se basa en un diseño simple de receptor de conversión directa para 3.5 Mhz donde se obtienen las necesarias señales defasadas del oscilador local mediante un circuito LC; se utiliza un mezclador balanceado a diodo como detector de cuadratura.

El circuito no es extremadamente sensible, pero es más que adecuado para operación local o de señales fuertes y por cierto que su simplicidad es dificil de batir!!!

El ZeroSDR (el Zero proviene de "*Cero Sintonía*")es un circuito mucho más elaborado desde el punto de vista complejidad, pero mejora drásticamente la simplicidad de armado y ajuste a partir de utilizar componentes digitales y no contar con ningun circuito sintonizado.

Gintaras utiliza una fuente externa como oscilador local lo que limita un poco la generalidad del diseño (no todos tienen un generador de señales entre su instrumental por cierto, y el oscilador local tiene lo suyo en términos de construcción).

Notese que otro de los factores de ajuste críticos del diseño TinySDR es el obtener señales de oscilador local defasadas en 90 grados; lo que para el caso de intentarlo con un circuito LC requiere cuidadoso y extensivo ajuste (y probablemente mucho cuidado constructivo para que sea estable); en el diseño ZeroSDR este problema se elimina totalmente porque las señales defasadas se obtienen digitalmente a partir de una única señal de oscilador local.

Este enfoque agrega una enorme simplicidad constructiva (el defasaje no requiere ajuste) pero agrega otra y es que el oscilador debe operar en 4x la frecuencia final; esto es si queremos dos señales de 7 Mhz para utilizar como oscilador local defasadas en 90 grados tenemos que empezar con una señal de 7 Mhz x 4 = 28 Mhz; el tener un oscilador simple que además sea estable se complica a medida que aumenta la frecuencia por lo que estos diseños son fáciles de implementar en las bandas bajas de HF (1.8, 3.5 y 7 Mhz) y se hacen crecientemente dificiles en las altas (10 Mhz en adelante).

Implementación de Tasic (YU1LM)

Quizás uno de los autores más prolíficos en diseños de SDR accesibles a radioaficionados "*de a pié*" es Tasic (YU1LM), su site web contiene literalmente docenas de circuitos con instrucciones detalladas de diseño, construcción y ajuste.
De todos sus diseños tomaré uno que considero contiene los principales ingredientes para esta discusión, sin embargo aliento a que el resto de los diseños sean también explorados con tiempo.
Se trata de un receptor compacto de HF cuyos detalles constructivos hé traducido del original.
Notese que el receptor está optimizado para operar en cualquier parte del segmento de HF (0.30 Mhz) pero **no es** en realidad un receptor de banda corrida; la frecuencia en la que opera dependerá de la frecuencia del oscilador local que se le construya.

Si se dispone de un oscilador externo el rango de recepción es efectivamente de toda la gama de HF; sin embargo en su versión más pedestre donde el oscilador local es un simple circuito basado en cristal el receptor operará sobre una segmento de unos 50 Khz en el entorno del oscilador local.

Se utiliza una configuración llamada "Detector Tayloe" como detector de cuadratura para procesar las señales I+Q.
Esto significa que si el oscilador local opera en 7.00 Mhz este circuito proveerá señales que permitirán recibir el segmento de 7.00 a 7.05 Mhz con una placa normal de sonido.

Acá es necesario detenerse algunas cuestiones que pueden lucir confusas sin aclaraciones adicionales.

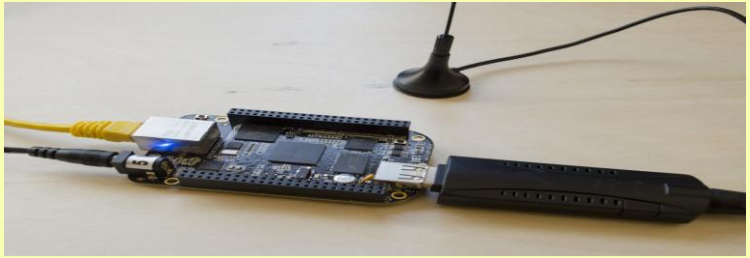
La primera de ella es cual es el ancho de banda de este circuito y la segunda es como se relaciona esto con la velocidad de muestreo en la placa de sonido; sin aclaraciones adicionales pueden aparecer en contradicción con definiciones anteriores.

El circuito no es sintonizado, por lo tanto cualquier señal en su entrada mezclada con el oscilador local queda disponible para su proceso por la placa de sonido; en las placas de sonido comunes (las que tiene casi toda PC) de 16 bits las mismas operan con una frecuencia máxima de muestreo de 48 Khz y por lo tanto pueden procesar señales de hasta unos 24 Khz de ancho de banda (criterio de Nyquist).

Para evitar problemas de alias en el proceso de conversión estas placas tienen un filtro pasabajos que elimina todo componente por encima de 24 Khz por lo que el ancho de banda del sistema está definido por estos filtros (a pesar de no haber circuitos sintonizados previos).

Sin embargo en el ejemplo “casual” anterior comento que con un oscilador local de 7.00 Mhz se puede obtener un segmento de 7.00 a 7.05 Mhz, es decir 50 Khz!! Como es esto posible si el filtro en la placa de sonido limita la banda pasante a la mitad? El secreto está en el disponer de señales I+Q donde cada canal lleva la mitad del ancho de banda por lo tanto es como que se duplica la capacidad de procesamiento de la PC en terminos de frecuencia. Esto ciertamente no ocurría cuando procesabamos una banda base monoseñal (como en los tipos Ia y Ib).

Finalmente, un circuito como este nos permite abordar un tercer tópico relacionado con la placa de sonido y es su ancho de banda; las placas de sonido “standard” son de 16 bits y tienen capacidad de manejar un ancho de banda de aproximadamente 50 Khz cuando son alimentadas con señales I+Q



Placas de sonido de 24 bits, no ya tan comunes pero aún dentro de costos muy razonables para un radioaficionado, pueden manejar frecuencias de muestreo de hasta 96 Khz; esto es permiten manejar anchos de banda de hasta 48 Khz que al duplicarse por la disponibilidad de señales I y Q permiten manejar casi 100 Khz de un segmento de banda con una frecuencia de oscilador local fija! Hay placas de mayor cantidad de bits que permiten anchos de banda de 120 y 150 Khz pero su costo ya no es tan bajo.

Si bien las bandas de HF son más anchas que esto aún asi no es descabellado acomodar los gustos operativos con una cuidadosa elección de la frecuencia del oscilador local de forma que con una única frecuencia fija en éste tengamos los 100 Khz de la banda donde transcurre la mayor parte de nuestro tiempo (donde nuestra antena funciona mejor, la que tiene los modos que nos interesan, etc).

Implementación Elektor

El diseño anterior de YU1LM provee una performance comparable a la de cualquier receptor comercial en la gama de HF a una ínfima fracción de su costo y dentro de limites razonables de complejidad constructiva.

Parte de su simplicidad sin embargo deriva de utilizar un oscilador a cristal de frecuencia fija como oscilador local, este hecho hace más facil la construcción y calibración pero limita el receptor a un segmento de 50 a 100 Khz de la banda que a uno le interese (dependiendo de la placa de sonido utilizada).

La construcción de osciladores variables estables há sido un tópico de interés para los radioaficionados desde siempre; y sabemos que no es sencillo obtener un diseño facil de implementar y estable. Los circuitos SDR necesitan además mucha estabilidad tanto de frecuencia como de fase para operar correctamente.

Los diseños basados en PLL pueden ser adecuados si se implementan con cuidado pero pueden tener caracteristicas de ruido de fase significativos que no los hacen siempre adecuados para su utilización en SDR; la complejidad constructiva de un oscilador basado en PLL dista de ser baja además estando tanto su armado como calibración normalmente fuera del alcance de un principiante.

Sin embargo la reciente, y creciente, disponibilidad de circuitos integrados de síntesis digital de señales (DDS) empieza a hacer atractiva la construcción de circuitos osciladores que son al mismo tiempo sencillos constructivamente y extremadamente estables.

Esto fue aprovechado en el diseño adjunto que apareció en la revista Elektor donde una placa de procesamiento SDR genera las señales I+Q en banda corrida sobre el
Espectro de HF en base a controlar la frecuencia del DDS; la complejidad constructiva es mediana y los materiales no son de disponibilidad masiva, pero el diseño permite una implementación de un receptor de alta gama a un costo ridículamente bajo y aún al alcance si no de un principiante por lo menos de alguien con conocimientos constructivos modestos.

El artículo y los negativos del impreso puede ser obtenidos en el site de la revista y se adjunta aquí para referencia.

Notese que el diseño es lo suficientemente avanzado como para incorporar una interfaz USB para realizar la interfaz respectiva con la PC con la que opera; el artículo refiere al software SDRadio como controlante, este software no es parte del paquete sino otro componente de la operación SDR que será explorado por separado

Implementación SoftRock

Distintos diseños han sido propuestos para implementar SDR alrededor de los mismos principios básicos; habrá diferencias en como la señal es pre-procesada, como es mezclada, como se genera la frecuencia del oscilador local y como se logran los distintos defasajes y filtros necesarios.

Pero todos los diseños tienen en común que al final de su trabajo quedan disponibles dos señales, la I y Q, con el cual puede procederse a la digitalización y procesamiento propiamente dicho.

En el tiempo han aparecido diseños a escala comercial, sea en forma de kit como en la forma de productos terminados, que pueden ser utilizados por radioaficionados; una búsqueda en Google rápidamente arrojará nombres como SoftRock. SDR-1000, WinRadio o Perseus que no serán explicados aquí en detalle pero se trata de diferentes placas o dispositivos de front-end.

Sin embargo es necesario detenerse en el diseño *SoftRock* no solo por ser probablemente la implementación más popular y exitosa de SDR sino porque además tiene un costo tan razonable y preserva bien el espiritu experimental de la radio.

Todos los detalles sobre Softrock como oferta, sus costos, modelos y disponibilidad pueden obtenerse de su autor Tony Parks. Una versión preliminar de la placa (ya discontinuada) y su funcionamiento pueden verse en el site de AMQRP , en el mismo existen referencias sobre como contactar a Tony Parks por información comercial; la placa tiene grupos en Yahoo para su discusión y soporte.

Software para SDR

En lo visto inicialmente un sistema SDR tiene tres componentes de los cuales me hé referido con cierta extensión a uno de ellos (procesamiento preliminar de señales por hardware) y queda claro que la placa de sonido de la PC será en la mayoría de los casos el segundo (dispositivo digitalizador).

Sin embargo el corazón mismo del sistema, es decir el software que procesa digitalmente las señales es un aspecto al que me hé referido muy tangencialmente; es necesario profundizar en el para completar un panorama de SDR aplicado a radioaficionados.

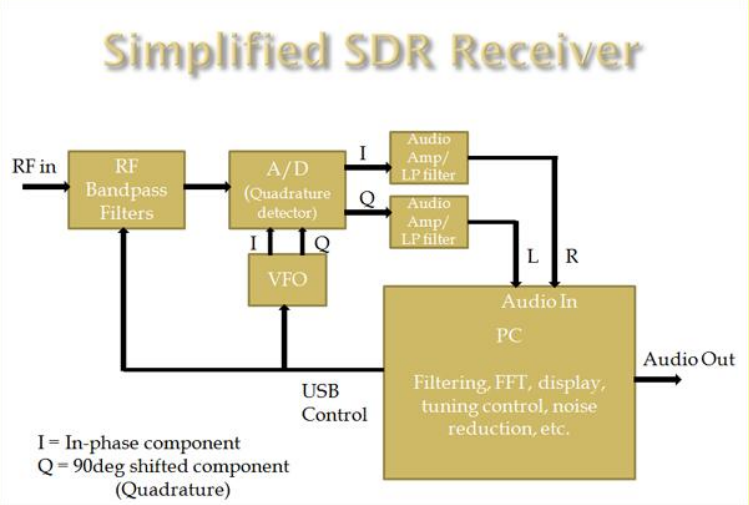
Es quizás razonable asumir que un aficionado principiante abordará el uso de SDR mayormente desde la perspectiva de construir u obtener el hardware como el descripto antes, dotar a su PC de la mejor placa de sonido que pueda y utilizar algunos de los paquetes de software disponibles para implementar el sistema SDR.

Sin embargo la implementación del software es una fuente de experimentación tan rica e inagotable como su contrapartida de hardware; en realidad y desde una perspectiva personal lo es más, pues una misma plataforma básica de hardware SDR puede dar lugar a múltiples proyectos de software en la medida que se implementan diferentes modos y protocolos de comunicación asi distintos enfoques de pre y post-procesamiento de las señales.

Hay varias formas de clasificar los paquetes de software, una posible que encuentro adecuada es:

Paquete Cerrado: implementa SDR de una forma en particular, agregando algoritmos y funciones fijas. Puede soportar una o más plataformas de hardware. Puede o no tener el codigo fuente disponible.
Paquete Semi-Abierto (o Semi-Cerrado): implementa SDR de una forma en particular pero provee facilidades, normalmente en la forma de un *Application Program Interface (API)*, para que determinados aspectos del funcionamiento puedan ser implementados externamente, el funcionamiento interno modificado o agregar soporte a plataformas de hardware no sostenidas nativamente por el paquete en su forma original.
Paquete Abierto: implementa los bloques constructivos de SDR tales como mezcladores, filtros, generadores, etc. El paquete implementa algún mecanismo mediante el cual los bloques constructivos pueden interconectarse para formar distintas configuraciones de sistemas SDR.

Se verán con mayor detalle los distintos enfoques de implementación en las secciones siguientes.

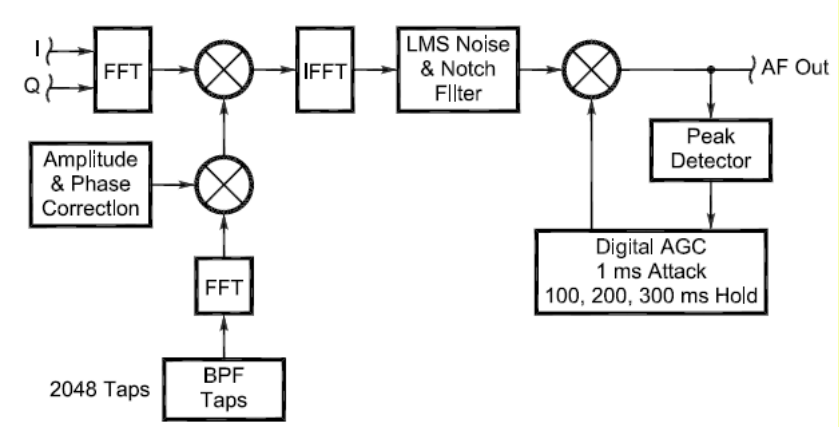


Principios Fundamentales

Casi todos los paquetes de software asumen disponer de la versión digital de las señales I y Q disponibles como punto de partida; el resto de las funciones de control del hardware serán accesorias a la operación SDR misma y dependerán de cada implementación.

En su forma más básica las señales I y Q son utilizadas para demodular el modo que se desee tal como fue descripto en señales anteriores; es un buen ejercicio seguir la implementación de JI3GAB previamente discutida para observar como se procesan las señales I y Q en los distintos pasos; al mismo tiempo es posible observar como funciones auxiliares que son dificiles de comprender en abstracto tales como la *decimación* y la *interpolación* son utilizadas para compensar por diferentes tasas de muestreo y en diferentes partes de la cadena.

Cada paquete de software implementará su propia versión de que pasos de pre-procesamiento y post-procesamiento le agrega a la demodulación (o modulación) básica de las señales I/Q, por ejemplo en el ejemplo de Youngblood (AC5OG) previamente discutido el esquema de procesamiento será:



Notese que la señal I/Q es transformada del dominio del tiempo al dominio de frecuencia mediante una *transformada rápida de Fourier* (FFT) que es un procedimiento matemático básico optimizado para procesamiento digital de señales.

La señal es demodulada con el algoritmo que corresponda aportando una serie de correcciones en amplitud y fase (muchas de ellas que se corresponden con simplificaciones en el hardware).

La señal demodulada es transformada nuevamente al dominio del tiempo mediante la *transformada inversa de Fourier* (IFFT) y luego procesada en sus caracteristicas de banda pasante, comportamiento de ruido y de magnitud de señal mediante algoritmos de LMS Noise, Notch Filter y AGC digital.

Puede observarse que en soluciones como la de SM5BSZ previamente vista el procesamiento puede ser considerablemente más complejo e involucrando más etapas de procesamiento.

Es obvio que cuantas más etapas de procesamiento se desee introducir más potente tiene que ser el procesador que las realice; con algoritmos razonablemente optimizados es posible realizar SDR con máquinas cuya velocidad de CPU sea 500 Mhz o superior (relativamente modestas) para señales con anchos de banda procesables por una placa de sonido (algunas decenas de Khz).

No hay una fórmula o algoritmo para determinar la capacidad necesaria para procesar una determinada señal, es en forma última un mecanismo de prueba y error.

Sin embargo no siempre los paquetes asumen o pueden asumir empezar por señales I/Q, en algunos casos es una opción de configuración no hacerlo, en otros casos las señales I/Q directamente no son soportadas.

Nuevamente el ejemplo de JI3GAB es instructivo al respecto. Puede verse que en esa implementación se comienza por una señal de audio “normal” y digitalmente se la mezcla con un oscilador local (ficticio, en software) con dos señales defasadas en 90 grados entre si (mediante también procesamiento en software); las señales I/Q existen entonces en un momento intermedio del cálculo y alli pueden ser aplicados los algoritmos del caso.

La ventaja de los paquetes que no requieren las señales I/Q (o que pueden opcionalmente funcionar sin ellas) es que pueden ser alimentados por esquemas de recepción como los discutidos previamente del Tipo Ia y Ib. Como hemos visto este tipo de enfoques tiene serios problemas de ruido y rechazo de imagen pero aun asi es posible aprovechar en oportunidades funciones de demodulación y filtrado disponible en aplicaciones SDR para aumentar prestaciones que el receptor originalmente no tiene.

Paquete Cerrado

Es la implementación más difundida, facil de instalar y utilizar. Tipicamente se asume que las señales I/Q son alimentadas a los canales derecho e izquierdo de la placa de sonido (la que, por lo tanto, debe ser stereo).

Los paquetes implementan típicamente una o más de las siguientes funciones:
Demodulación del o de los modos que soporta, tipicamente CW, AM, SSB, NBFM, WFM y otros.
Distintas funciones de filtrado pasabajo, pasabanda, pasalto, notch, etc.
Funciones de control automático de ganancia (AGC), de frecuencia (AFC), Noise Blanker y otros.
Distintas facilidades para calibrar, adaptar y compensar diferentes defectos en el proceso de digitalización y procesamiento dependiendo de la tarjeta de sonido utilizada.
Funciones para controlar el hardware utilizado, tanto sea para compensar simplificaciones de diseño como para controlar aspectos de su funcionamiento tal como frecuencia del oscilador local o ganancia de la cadena amplificadora/mezcladora.
Una interfaz con el usuario capaz de interactuar con las funciones anteriores asi como proveer medios de ayuda a la operación tales como indicadores de señal, información de espectro (espectrograma o interfaz “waterfall”) y otros.
Mas allá de los más o menos aspectos que el autor decide dejar habilitados para su configuración por el usuario estos paquetes implementan los algoritmos de decoficicación con un esquema fijo y pre-definido.

Notese que esto no cambia por el hecho que el autor haga disponible los fuentes del aplicativo, el modificar el codigo fuente además de tener su complejidad termina con otro paquete cerrado que funciona diferente.

En su operación soportan el hardware que sus autores definieron u operan con hardware genérico (por ejemplo tomando las señales I/Q por la placa de sonido pero sin controlar en forma alguna el proceso de generación de las mismas).

Su instalación es típicamente muy sencilla y si bien la documentación disponible varía considerablemente de paquete en paquete las funciones básicas suelen poderse operar con un minimo de esfuerzo de configuración.

Hay paquetes tanto para Linux (por ejemplo LINRAD) como para Windows (entre otros Rocky, SDR-1000, KGKSDR, PowerSDR o DSPRX entre otros); personalmente encuentro Rocky una implementación muy sencilla, general y potente.
Se refuerza la idea que si bien casi todos están optimizados para alguna placa en particular también incluyen en general la posibilidad de operar con placas genéricas simplemente procesando señales I/Q alimentadas a la placa de sonido permitiendo operar con implementaciones como la TinySDR o las de YU1LM por caso.

Paquete Semi-Abierto
En la medida en que las implementaciones de SDR proliferan sin que exista un standard de hardware es crecientemente imposible para los autores de software dar soporte no solo a las nuevas implementaciones que emergen luego que el software fue escrito sino incluso a las existentes durante su diseño.

Al mismo tiempo desde el punto de vista de hardware puede ser importante o interesante modificar algún aspecto del procesamiento de las señales realizado por el programa sin necesariamente tener que re-escribir todo el paquete (con su mayoría de funciones útiles); esto es tanto para experimentar con nuevos modos de comunicación o para mejorar los existentes.

Los paquetes en esta categoría proveen una *Application Program Interface (API)* donde es posible desde un módulo externo que puede ser escrito por el usuario final, típicamente siguiendo la estructura de una *dynamic linked library (dll)* en el ambiente Windows, modificar parcial o totalmente el comportamiento del programa, agregar nuevas funciones, agregar soporte a nuevo hardware y otras.

El paquete más notable en esta categoría es el escrito por Alberto (I2PHD) denominado WinRad.

El API definido por el autor es usualmente documentado junto con el programa, no hay un standard definido para estas implementaciones por lo que el autor define que llamadas incluye, que parámetros permite intercambiar y que funciones estarán sujetas a modificación externa; supongo que con el tiempo emergerá algún standard para estas API, sea que todos los paquetes apliquen el de alguno en particular o que se cree un standard al efecto.

Por ejemplo el documento de definición de Winrad es el siguiente:

Puede verse que el API no solo considera las funciones mediantes las cuales el programa inicializa la dll y interactua con un hardware definido por esta sino que también permite acceso al flujo de información digitalizada de audio y funciones de servicio que mediante callback al programa principal permite la modificación del comportamiento básico.

El desarrollo típicamente podrá realizarse en VB, C++ o Delphi aunque cualquier otro lenguaje que permita generar dlls también resultará apropiado.

Por su naturaleza esta categoría de paquetes aplicativos es probablemente el mas ajustado a las necesidades de experimentación de aficionados porque mientras permite su uso en forma directa (modos convencionales) al ser instalado también permite prácticamente cualquier grado de experimentación (modos agregados) a esfuerzos de desarrollo razonables.

Dado que solo se necesita implementar el código necesario para sostener la nueva función implementada o soportar el nuevo hardware al mismo tiempo que todo el resto de las funciones se utilizan tal como originalmente fueron concebidas por el autor.

Paquete Abierto

Esos paquetes son en realidad plataformas de desarrollo donde los bloques funcionales para realizar DSP se hacen disponibles en la forma de módulos cuyas interfaces están definidas y que deben ser interconectados en configuraciones definidas según la implementación que se desee.

Aunque puedan existir ejemplos de implementación que se pueden utilizar como guias no suelen ser ambientes que están disponibles para su uso en radio con solo instalarlos; es usualmente necesario conocer en profundidad su funcionamiento, configuracióny programación para realizar implementaciones útiles con ellos.

A nivel profesional MatLab es un ambiente muy sofisticado y complejo que se puede utilizar con este propósito; su costo y complejidad (su propósito es vastamente más grande que implementar SDR o incluso DSP) lo ubica en el extremo de alta gama disponible.

El ambiente GNURadio, desarrollado por la iniciativa del mismo nombre, implementa un ambiente para exploración de SDR de propósito general de mucha potencia donde bloques constructivos provistos por el paquete (escritos típicamente en C++) hacen disponibles fuentes, filtros, demoduladores y otros bloques constructivos de SDR. Estos bloques constructivos son típicamente escritos para maximizar su performance.

Los bloques constructivos son integrados para una implementación en particular mediante un script escrito en un lenguaje llamado Python (un lenguaje bastante común en el ambiente Linux).

Estos ambientes y otros permiten extender en forma prácticamente ilimitada la plataforma mediante el agregado de bloques constructivos para sostener hardware procesador de señales, de digitalización y diferentes algoritmos.

Si bien la placa de sonido de la PC es soportada como digitalizador estas plataformas son típicamente utilizadas con digitalizadores de otro tipo y mayor potencia de procesamiento.

En general este tipo de plataformas tienen una potencia y complejidad de uso que las coloca por encima de lo que es necesario para utilizar SDR en aplicaciones de radio pero es útil conocer su existencia.

Una implementación más reciente es BasicDSP creado por un enfoque muy novedoso es el propuesto por Pieter-Tjerk PA3FWM y Niels PA1DSP quienes son los responsables atrás del receptor público WebSDR montado en la Universidad de Twente en Holanda. Ellos crearon un paquete de software que permite experimentar con SDR con una simplicidad asombrosa.
En esencia el programa provee bloques pre-armados para acceder a las señales de la placa de sonido (entrada y salida, mono y stereo) u otras fuentes tales como archivos .WAV, generador sinusoidal y generador de ruido, algunas funciones matemáticas básicas muy utilizadas en DSP, la posibilidad de implementar filtros FIR y la posibilidad de controlar externamente la operación mediante hasta 4 controles deslizantes (slider) que pueden ser asignados programáticamente.

Para implementar un algoritmo cualquiera de SDR bastará tipear un script sencillo con la manipulación que es necesario realizar en la señal para luego poner el resultado en la salida que se desee. El programa permite trabajar por separado los canales derecho e izquierdo, tanto en la entrada como la salida.

La metáfora de funcionamiento es que cada muestra que realiza la placa de sonido (que puede controlarse en su cadencia mediante la directiva *samplerate*) es sometida a cada operación en el script hasta que finalmente se envia a la salida. Una versión experimental, cuya principal novedad es la implementación de la función matemática atan(x) por lo que permite implementar algoritmos relacionados con FSK/PSK/NBFM puede ser encontrada en el area de archivos del grupo Yahoo BasicDSP. Este programa es equivalente a una "regleta" para SDR del lado del software, vale la pena jugar con el.

Aplicaciones SDR basadas en Servidor

El poner un receptor disponible en una red local o en el Internet no es algo novedoso; en cuanto hubo aplicaciones Web de cierta complejidad y plataformas de transmisión (*streaming*) de audio junto con usuarios con anchos de banda razonables hubo aficionados que pusieron sus receptores en la Web para que otros pudieran sintonizar los mismos. De acuerdo a la implementación el usuario puede o no controlar los parametros de recepción tales como modo, frecuencia y otros parámetros.

Más allá de lo interesante que puede ser este tipo de implementaciones (quien no los há usado en condiciones de propagación para escucharse “al otro lado del mundo”?) no constituyen SDR; es simplemente un receptor con un transporte de audio donde hay poco o ningun procesamiento digital como tal.

Sin embargo a partir de la disponibilidad de plataformas SDR basadas en Web, siendo WebSDR quizás la más importante, la captura de la señal puede realizarse en una localización pero el procesamiento de la banda base realizarse en otra.

De esa manera multiples usuarios no están ya limitados a escuchar el mismo modo y frecuencia que uno de ellos controla o que están definidos en el receptor fuente sinó que cada uno recibe un segmento de una banda y procesa la misma independientemente del resto.

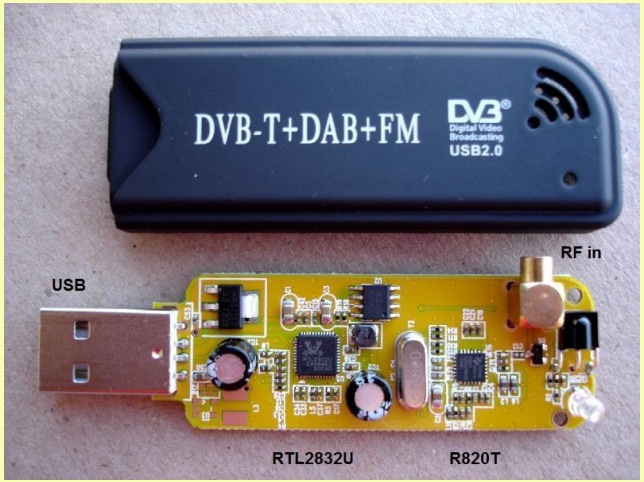
De esa forma docenas de usuarios pueden concurrentemente usar el receptor cada uno sintonizando la frecuencia y modo de su elección. En la taxonomía inicial estas implementaciones fueron referidas como **Tipo V**.

Si bien estas modalidades no han sido desarrolladas para radioaficionados sino para otros usos como radioastronomía por ejemplo, las bandas de aficionados proveen un buen público donde evaluar las plataformas.

Un ejemplo notable de una implementación es, mencionado en la sección anterior, el receptor implementado en la Universidad de Twente (Holanda) donde segmentos de la banda de 40, 80 y recientemente 20 metros son accesibles con esta modalidad.

Si bien paquetes como WebSDR están ligeramente fuera del alcance de usuarios individuales, en particular principiantes, ciertamente están dentro del alcance de usuarios más sofisticados e instituciones por lo que las prácticas operativas a partir de una estación multioperada pueden adquirir una dimensión absolutamente sin precedentes.

No es dificil visualizar que esta tecnología puede, comunitariamente, ser utilizada para compartir hardware complejo que quizás sea imposible disponer individualmente.



Fuentes:

www.qsl.net/l/lu7did/docs/SDR/SDR_**Apuntes**_v1.1.doc

http://ea8ate.blogspot.mx/2009/01/10-consejos-para-operar-en-psk31.html