

PROCESAMIENTO DE SEÑALES CON TECNOLOGÍA SDR: PRIMEROS PASOS CON EL RECEPTOR PERSEUS

Karel Toledo de la Garza¹, Liset Martínez Marrero², Alejandro Cabrera Aldaya³, Ernesto Cruz Pereda⁴, Luis Miguel Gato Díaz⁵, Enildo Sánchez Rodríguez⁶, Ernesto González Zamora⁷, Yuneisy Bengochea Arias⁸, Jorge Torres Gómez⁹

^{1,3,4}Complejo de Investigaciones de Tecnologías Integradas (CITI), Calle 114 e/Rotonda y Ciclovía, Boyeros,

^{2,5,9}CUJAE, Dpto. de Telecomunicaciones y Telemática, Calle 114 e/Rotonda y Ciclovía, Boyeros,

^{6,7}Centro de Investigación y Desarrollo Técnico (CIDT), Calle 114 e/Rotonda y Ciclovía, Boyeros,

⁸Centro de Control Aéreo, Empresa Cubana de Aeropuertos y Servicios Aeronáuticos (ECASA)

⁹e-mail: jorge.tg@electrica.cujae.edu.cu

RESUMEN

La tecnología de Radio Definido por Software (SDR) es empleada en las comunicaciones digitales con múltiples aplicaciones. Su principal ventaja se encuentra la capacidad de integrar múltiples soluciones en un mismo dispositivo mediante la reutilización de código abierto. El presente artículo expone los resultados alcanzados en la configuración del receptor Perseus de la firma Microtelecom acoplado a una tarjeta Raspberry Pi. Mediante el receptor Perseus se digitaliza el espectro radioeléctrico en una banda de 40MHz, mientras que con la tarjeta Raspberry Pi se controla la operación del receptor. En el lado remoto se conformó una interfaz gráfica en el lenguaje C con el compilador QT para visualizar los datos y cambiar los modos de trabajo del receptor.

PALABRAS CLAVES: Perseus, Raspberry, SDR.

ABSTRACT

Software Defined Radio (SDR) technology is commonly used in digital communications with several practical applications. The main advantage is focused on the availability of various solutions and the reusability of open source codes. The current article addresses the results obtained in the configuration of the Perseus receiver together with the Raspberry Pi processor. Through the use of the Perseus receiver the spectrum is sensed in a bandwidth of 40MHz. On the other hand, the Raspberry Pi set the Perseus configuration properly. Besides, a graphical user interface was developed in order to depict the spectrum received as well as for changing the receiver parameters.

KEY WORDS: Perseus, Raspberry, SDR.

INTRODUCCIÓN

Los servicios de comunicaciones requeridos en la actualidad son de rápida evolución hacia el aumento de interconectividad y rapidez de transmisión. Esta característica requiere del desarrollo de nuevos esquemas de transmisión-recepción que hagan uso integral de dimensiones del espacio de radio tales como la frecuencia, el tiempo, el espacio y la codificación [1], el cual toma la definición de Hiper-espacio [2-3].

Es en este escenario donde emerge el desarrollo del Radio Definido por Software (SDR, Software Defined Radio), como elemento flexible [4] que explota el uso de estas dimensiones, pero a la vez necesita de la implementación de técnicas de sensado en la búsqueda de oportunidades de transmisión. Estas técnicas de sensado se desarrollan con mayor interés en el área espectral, con el objetivo de identificar intervalos de frecuencia libres para la transmisión de información. Entre las técnicas de sensado se puede mencionar el detector de energía [5-6], el sensado coherente [7] y la ciclo-estacionariedad [8-10]. Estas técnicas requieren de la aplicación de la Transformada Rápida de Fourier (FFT) [11] para identificar de forma estadística los intervalos de ocupación del espectro.

Otra de las técnicas para describir con más detalle la ocupación del espectro se basa en la identificación de la tecnología que emite la información [12]. Con esta descripción no solo se puede determinar su ancho de banda sino además su alcance, incluso es posible limitar las comunicaciones sólo con tecnologías específicas.

Las características de transmisión varían según el canal de comunicaciones empleado. En algunas bandas de frecuencia se transmite con modulaciones analógicas como la AM o FM [13], mientras que en otras se transmite con modulaciones digitales como la OFDM [14]. Para aprovechar al máximo las características del canal es necesario que el sistema cambie de forma adaptativa sus parámetros de comunicación. La tecnología SDR permite esta operatividad a partir de definir en software sus principales funciones, en hardware queda solamente el acondicionamiento y la digitalización de la señal, en software el conjunto de aplicaciones para clasificar y demodular.

El presente artículo describe la arquitectura y control del Perseus, el cual hace función de receptor para la digitalización, su control se realiza mediante la tecnología Raspberry Pi. Esta configuración representa el elemento base para la recepción de señales y conforma un punto de partida para las aplicaciones SDR.

DESCRIPCIÓN DEL SISTEMA

El sistema que se propone para conformar una aplicación SDR está conformado por el Perseus en conjunto con el Raspberry Pi. El primero para hacer la captura y digitalización de la señal, el segundo para controlar al Perseus y hacer procesamiento de señales. La figura 1 muestra el sistema completo, el Raspberry Pi se emplea con tres funciones: para controlar al Perseus, para hacer procesamiento y para enviar los datos por vía Ethernet hacia una aplicación remota. La opción del Raspberry Pi como controlador resulta atractiva por sus bajos costos y consumo de potencia, esto en comparación con sistemas de cómputo tradicionales tales como computadoras personales.

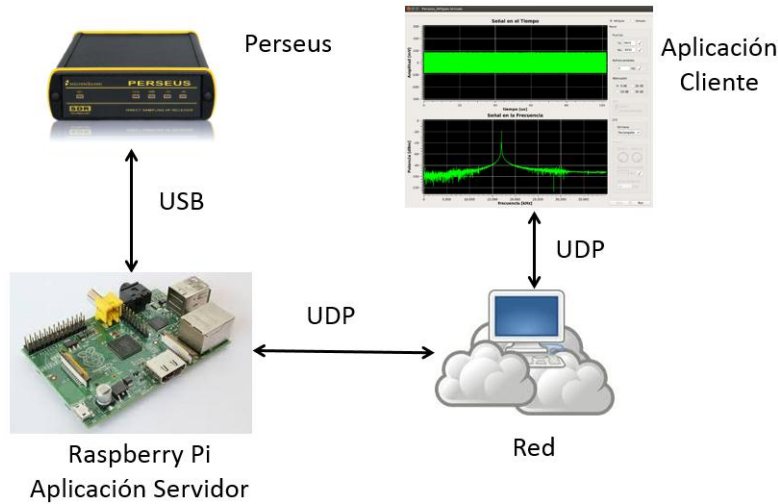


Figura 1: Descripción del Sistema

El Perseus está conformado para la captura de la señales por un banco de filtros, un convertor A/D, un FPGA y un Microcontrolador como se muestra en la figura 2, los parámetros de trabajo se indican en la tabla 1. El microcontrolador es la interfaz entre la PC controladora y el circuito de recepción del Perseus mediante USB. Esta configuración funciona en dos modos de trabajo, uno para aplicaciones espectrales (HFSSpan) y otro para la demodulación de las señales (Stream). En el modo HFSSpan se reciben del Perseus bloques de 16 384 muestras de forma asíncrona. En este modo se muestrea la señal con una frecuencia de muestreo de 80MHz, y con una longitud de datos de 16 bits, lo cual requiere de una rapidez de transmisión de datos de 1 280 Mbits/s. Esta velocidad es superior a las posibilidades de salida por USB de 480 Mbits/s, este modo no se emplea para la recepción, sino más bien para la aplicación de sensado de espectro.

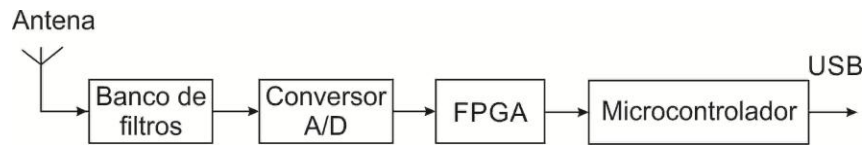


Figura 2: Estructura interna del FPGA.

Por otro lado, en el modo Stream el envío de datos del Perseus se hace de forma síncrona, en este caso el ancho de banda de la señal se especifica como máximo en 2 MHz como se indica en la tabla 1. Para el modo Stream en el FPGA se programa un circuito de down-conversion como se muestra en la figura 3. Con este circuito se obtiene la representación banda-base de señales pasa-banda, esto en general en forma compleja mediante las señales I y Q. Es a partir de las mismas que se procede a demodular la señal recibida.

Tabla 1: Parámetros de trabajo del receptor Perseus

| Parámetros | Valores |
|------------------------------|--|
| Rango de Frecuencia | 10 kHz – 40 MHz |
| Atenuadores | 0, 10, 20, 30 dB |
| Convertor A/D | 14 bits, 80 MHz |
| FPGA | Xilinx's Spartan III XC-3S250E |
| Sensibilidad | -131 dBm |
| Ancho de Banda (Modo Stream) | 48 kHz, 95 kHz, 96 kHz, 125 kHz, 192 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz |
| USB | USB 2.0, 480 Mbits/s |

Por su parte el Raspberry Pi es una placa con un procesador BCM2835, con 512 Mbytes de memoria RAM, con periféricos HDMI, S-Video, Ethernet, USB 2.0 y 120 pines de entrada/salida. Esta placa de pequeñas dimensiones resulta de muy bajo coste y con poco consumo de potencia. El sistema operativo instalado es el Raspbian con base en la distribución Debian y se instala en una SD-Card con capacidad para 4 GBytes.

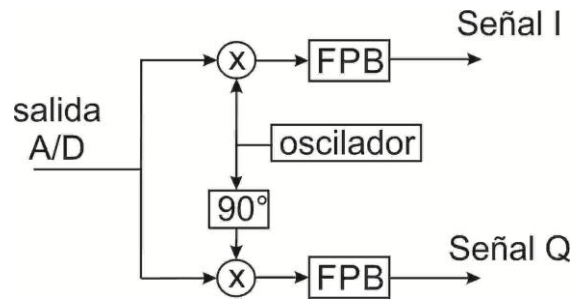


Figura 3: Estructura interna del FPGA.

CONTROL DEL PERSEUS DESDE EL RASPBERRY PI

Para controlar el Perseus desde el microcontrolador empotrado en la placa Raspberry Pi es necesario desarrollar una aplicación software que se ejecute sobre el sistema operativo Linux (Raspbian). Este sistema operativo es el que viene instalado para el Raspberry Pi. Esta aplicación debe ser capaz de realizar las mismas operaciones que realizan las aplicaciones que brinda el fabricante del Perseus para el sistema operativo Windows, mediante la configuración de este dispositivo en modo HFSpan a través de la interfaz USB. El fabricante del Perseus brinda ejemplos de aplicación para su configuración tanto para el sistema operativo Windows como para Linux, pero para este último no existe soporte en la

configuración del dispositivo en el modo HFSSpan. Se hace necesario desarrollar una aplicación para el modo HFSSpan en C en analogía con la solución para Windows, dado que el Raspberry Pi opera en Linux.

Para configurar el Perseus es necesario programar a través de la interfaz USB el firmware del microcontrolador y el bitstream del FPGA correspondientes a la configuración/modo de trabajo deseado. La aplicación de ejemplo para configurar el Perseus desde Windows utiliza ficheros bitstreams cifrados por el fabricante, esto con el objetivo de que no se utilice un bitstream que pueda dañar el dispositivo. Por otra parte, la aplicación equivalente para Linux utiliza ficheros bitstream sin cifrar. Esta diferencia evita que sea posible utilizar directamente los bitstreams que son proporcionados para Windows para configurar el dispositivo en modo HFSSpan desde Linux.

Tras el estudio y observación de las transferencias realizadas en Windows para configurar el dispositivo en el modo HFSSpan se identificó que un componente en la aplicación software en Windows descifraba el bitstream y lo enviaba hacia el dispositivo. El componente encargado de esta funcionalidad es una biblioteca de enlace dinámico (DLL) que brinda el fabricante del dispositivo. La aplicación de configuración para Windows utiliza una función de esta biblioteca para descifrar un bitstream y descargarlo hacia el dispositivo. En la aplicación de ejemplo se ejecuta esta función con un parámetro que es la clave de descifrado que debe utilizarse, pero con esta clave solamente no es posible descifrar el bitstream debido a que se desconoce el algoritmo de cifrado empleado por el fabricante del dispositivo para proteger el bitstream.

Debido a que el código de la DLL que contiene a la función de descifrado no está disponible, fue necesario realizar ingeniería inversa a la misma con el objetivo de identificar el algoritmo utilizado, esto permite descifrar el bitstream y emplearlo directamente en la aplicación para Linux del Raspberry Pi. Tras desensamblar el fichero DLL con el software IDA Disassembler, fue posible identificar el código ensamblador correspondiente a la función encargada de descifrar el bitstream. Luego del estudio de la misma fue posible identificar la porción de código encargada de esta funcionalidad la cual utiliza un algoritmo de cifrado de flujo para realizar esta tarea.

Los algoritmos de cifrado de flujo utilizan una función para la generación de números pseudo-aleatorios a partir de una clave inicial. De esta forma, para cifrar un fichero por cada byte del mismo se ejecuta esta función para generar un byte pseudo-aleatorio y aplicar la función XOR entre este y el byte del fichero correspondiente. Para el descifrado se realiza el mismo procedimiento, pero la entrada del procedimiento es el fichero cifrado y la misma clave. La función de generación de números pseudo-aleatorios utilizada contiene aproximadamente 25 instrucciones en ensamblador y al parecer es un algoritmo creado por la misma compañía para esta aplicación. Una vez identificadas las instrucciones en ensamblador de la función pseudo-aleatoria para descifrar un bitstream, se programa en un lenguaje conocido las funcionalidades de estas instrucciones de tal forma que sea posible obtener en un fichero el bitstream descifrado.

Luego de realizar esta tarea se obtuvo un fichero bitstream que poseía los campos que identifican a un bitstream válido para el FPGA que posee el Perseus. Para la verificación experimental del procedimiento de descifrado se contaba con los bitstreams para el modo de trabajo Stream tanto cifrados (en la versión

para Windows) como descifrados (en la versión para Linux), se aplicó el algoritmo de descifrado a estos últimos para comparar los ficheros resultantes con los originales y en todos los casos los contenidos de los ficheros coincidían, lo cual permitió confiar en el proceso de descifrado y que la descarga del bitstream descifrado para el modo HFSSpan desde Linux no dañaría al dispositivo.

Una vez obtenido el fichero bitstream del FPGA para el modo HFSSpan descifrado se desarrolló una aplicación para Linux a modo de configurar el Perseus y obtener las muestras en el Raspberry Pi. El modo de comunicación con el Perseus se logra a través de USB, lo cual requiere del empleo de la biblioteca libusb disponible para Linux utilizando el compilador gcc. Los pasos de programación y de configuración del Perseus son los siguientes:

1. Escaneo de los puertos USB de la PC: El escaneo se implementa con el objetivo de identificar los números de serie del IDVendor del Perseus. Con ello se reconoce además el total de receptores conectados a la PC.
2. Programación del microcontrolador: En el caso del modo Stream y modo HFSSpan los firmwares del microcontrolador (ficheros .hex) estaban disponibles. Solo se hizo necesaria la re-adaptación en el modo de representar esta información, con el objetivo de reutilizar el código disponible para el modo Stream. Una vez obtenido el fichero en la forma apropiada, la descarga se realiza mediante la función `libusb_bulk_transfer`.
3. Programación del FPGA: Al igual que el microcontrolador se requiere de la descarga de un fichero de programación (bitstream). El bitstream utilizado fue el obtenido a través del procedimiento de descifrado descrito anteriormente.
4. Configuración de los atenuadores: Luego de programado el microcontrolador y el FPGA del receptor Perseus se procede a configurar los filtros según la banda de trabajo que se desee. En el caso del modo HFSSpan estos filtros no son configurados, dado que se captura toda la banda de recepción. En el caso del modo Stream se configura el centro de filtro y la banda de paso en función del ancho de banda seleccionado (ver Tabla 1).
5. Configuración del Conversor A/D: El parámetro principal para la configuración del conversor A/D está dado por el ancho de banda a seleccionar para el caso del modo Stream. La velocidad del conversor A/D depende de los valores de ancho de banda mostrados en la Tabla 1. Para el modo HFSSpan el conversor A/D se ubica por defecto en la máxima velocidad de muestreo dado por 80 MHz
6. Recepción de datos: Luego de configurados todos los bloques del receptor Perseus se procede a la captura de los datos. La recogida de datos que envía el Perseus se realiza en el modo Stream por interrupción y en el modo HFSSpan por encuesta. La interrupción es generada por los paquetes USB que envía el Perseus y en el modo Stream se atiende por interrupción para que no exista pérdida de paquetes, de lo contrario no se pudiera proceder con la demodulación.

APLICACIONES Y RESULTADOS OBTENIDOS SOBRE RASPBERRY PI

Desarrollo de la aplicación gráfica

La interfaz gráfica de usuario es desarrollada en la aplicación Qt Creator para la configuración remota del Perseus, recepción, visualización y procesamiento de las muestras recibidas a través de una aplicación Servidor implementada en el Raspberry Pi. Ambas aplicaciones están conectadas vía Ethernet utilizando el protocolo UDP.

La aplicación desarrollada se muestra en las figura 4 y 5 para los modos HFSpan y Stream, respectivamente. Esta interfaz consta de 3 paneles fundamentales: visualización en el dominio del tiempo, visualización en el dominio de la frecuencia y configuración de ambos modos de trabajo del Perseus. La recepción de las muestras fue realizada utilizando la biblioteca socket.h, para la visualización en el tiempo. En la frecuencia se emplea la biblioteca qwt6.1.0 que se acopla al Qt Creator. La Transformada Rápida de Fourier se implementa utilizando la biblioteca fftw3, todas disponibles para sistemas operativos Linux y Windows. El panel de configuración consta de los parámetros disponibles en el Perseus y de otras funcionalidades que se exponen a continuación:

1. Modos de Trabajo: HFSpan o Stream.
2. Puertos de Transmisión y Recepción: utilizados para la comunicación con el Servidor.
3. Refrescamiento: Permite controlar la demora entre envíos de los bloques de datos recibidos del Perseus (Solo para el modo HFSpan).
4. Atenuador: Permite configurar el atenuador de entrada a la señal recibida por el Perseus en el caso de que posea mucha potencia y saturar al receptor.
5. ADC: Permite activar/desactivar una pre-amplificación y Dither a la señal recibida por el Perseus.
6. Ventana: Permite aplicar diferentes ventanas a la señal recibida en la aplicación Cliente para una mejor visualización de la Transformada Rápida de Fourier.
7. Dials: Permite mover las señales en fase y cuadratura (sumarle una componente de directa a la señal original) (Solo para el modo Stream).
8. Frecuencia Central (Solo para el modo Stream).
9. Ancho de Banda (Solo para el modo Stream).
10. Botones de Run y Stop: permite detener/iniciar la visualización en el dominio del tiempo y de la frecuencia.

De esta forma la aplicación al ser ejecutada comienza recibir muestras en el modo HFSpan y automáticamente se inicia la visualización de la señal en el tiempo y en la frecuencia hasta 40 MHz como se muestra en la figura 4. En este modo se permite cambiar al modo Stream, los puertos de transmisión y recepción, el refrescamiento, aplicar las funcionalidades del conversor analógico digital y la aplicación de diferentes ventanas para el cálculo de la Transformada Rápida de Fourier.

Al seleccionar el modo Stream, la aplicación de forma remota ordena al Perseus a cambiar de modo de trabajo y comienza a recibir la señal en fase y cuadratura que se visualiza en el tiempo y en el dominio de la frecuencia. En este modo se puede realizar todo lo antes expuesto para modo HFSpan, excepto el refrescamiento. Se adicionan también los dials para mover las gráficas de la señal en fase y en cuadratura, la selección de la frecuencia central deseada y el ancho de banda o frecuencia de muestreo. Las gráficas en este modo difieren del modo HFSpan debido a que se visualizan en el tiempo la señal en fase y cuadratura como se observa en la figura 5.

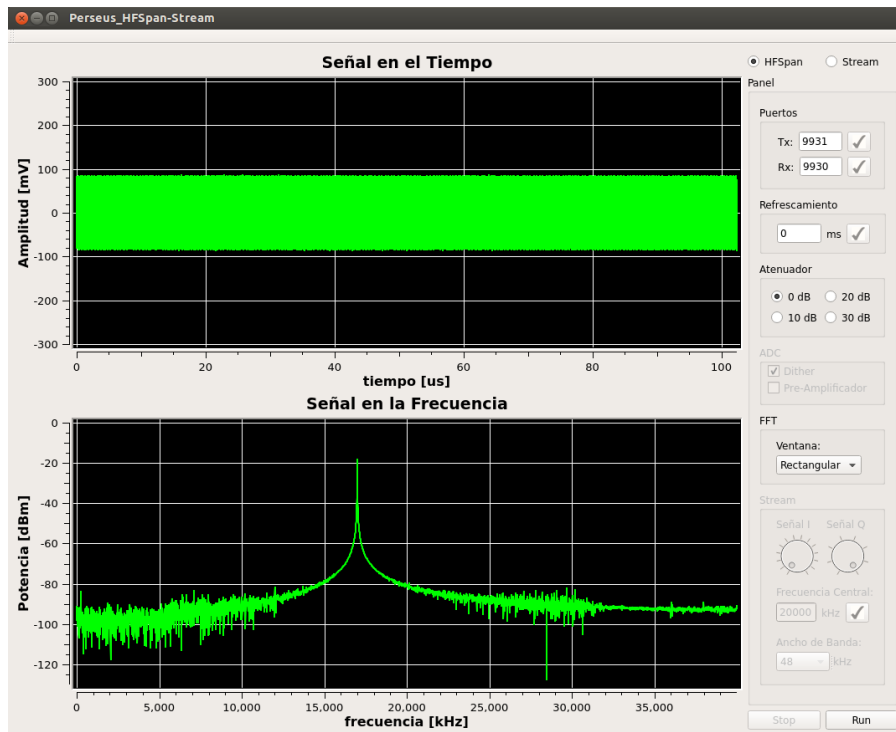


Figura 4: Aplicación gráfica Modo HFSpan.

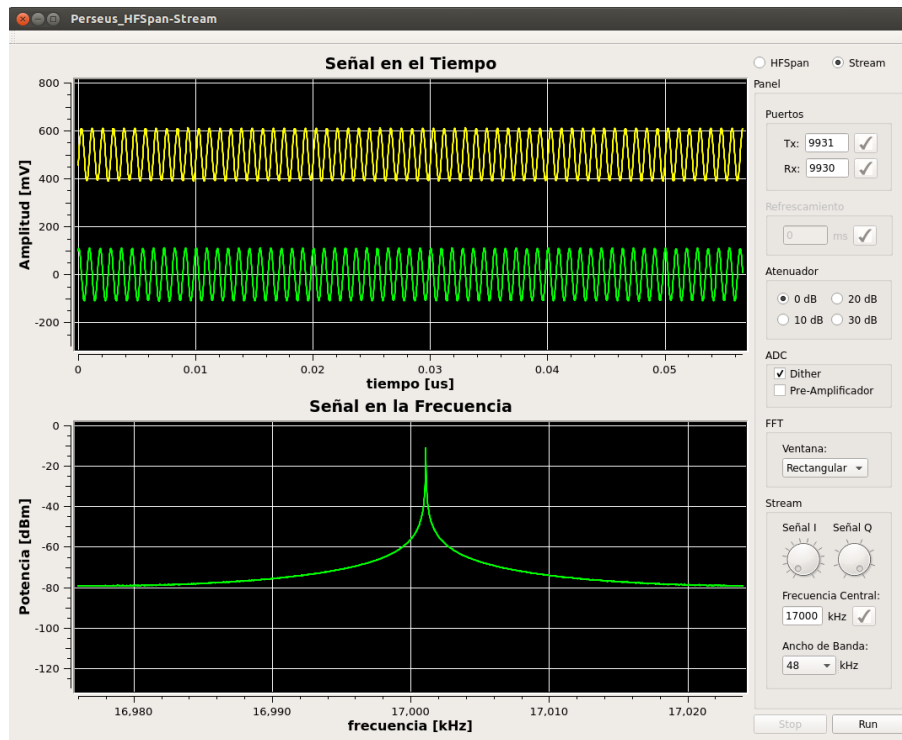


Figura 5: Aplicación gráfica Modo Stream.

Desarrollo de la aplicación gráfica

La Transformada Rápida de Fourier es implementada en la interfaz gráfica de usuario utilizando la FFTW (Transformada de Fourier). Esta biblioteca contiene una colección exhaustiva de algoritmos acelerados en lenguaje C para el cálculo de la DFT (Transformada Discreta de Fourier) y otros casos particulares.

Sus principales características se muestran a continuación:

1. FFTW calcula la DFT de datos complejos, datos reales, datos reales con simetría par o impar (usualmente conocida como la transformada del coseno o del seno) y la transformada discreta de Hartley (DHT) de datos reales.
2. Los datos de entrada pueden tener longitud arbitraria.
3. FFTW soporta datos multidimensionales arbitrarios.
4. FFTW soporta los juegos de instrucciones SSE, SSE2, AVX, AltiVec, y MIPS PS.
5. FFTW3.3 incluye el cálculo paralelo (multi-hilos).

Para la ejecución efectiva de la FFTW, no se usa un algoritmo fijo para el cálculo de la transformada sino que se adapta al hardware en el cual se está ejecutando para maximizar el rendimiento. El cálculo de la FFT se divide en dos fases: configurar un plan solo una vez y luego ejecutar el plan cuantas veces se necesite.

Estas dos fases para el cálculo de la FFT son utilizadas en la aplicación gráfica para la señal recibida. En el modo HFSPAN a través del cálculo de la FFTW de datos reales y en el modo Stream se parte de datos complejos debido a la recepción de la señal en fase y cuadratura. Los resultados obtenidos para ambos modos se observan en las figuras 4 y 5.

CONCLUSIONES.

En el presente artículo se describió de modo general el sistema conformado entre el receptor Perseus, el Raspberry Pi y la aplicación cliente. La solución obtenida permite la colocación del receptor Perseus en zonas remotas donde se requiera de mayor autonomía energética. Al realizarse el control del receptor por la placa Raspberry Pi se obtiene una solución factible con pocos recursos de cómputo y con un mínimo de costos. Este sistema permite además su acceso remoto dado que se establece comunicación por la interfaz Ethernet, lo cual posibilita compartir la información recibida por el Perseus entre múltiples clientes. Por otro lado, la interfaz cliente es la que desarrolla los algoritmos más complejos con la visualización de los resultados y el cálculo de la transformada de Fourier. La propuesta desarrollada permite además establecer en software los pasos de detección, clasificación y demodulación de las señales obtenidas por el receptor. El sistema desarrollado ofrece una variante de plataforma de radio cognitivo, en la cual se pueden desarrollar técnicas de sensado de espectro y cooperación.

REFERENCIAS.

1. YUCEK, T.; ARSLAN, H. "A survey of spectrum sensing algorithms for cognitive radio applications". *IEEE Communications Surveys Tutorials*, 2009, vol. 11, núm. 1, pp. 116–130.
2. DROZD, A. L.; KASPEROVICH, I. P.; CARROLL, C. E.; BLACKBURN, A. C. "Computational electromagnetics applied to analyzing the efficient utilization of the RF transmission hyperspace". En actas de Proc. IEEE/ACES International Conference on Wireless Communications and Applied Computational Electromagnetics, Honolulu, Hawaii, USA, 2005, pp. 1077–1085.
3. MATHESON, R. "The electrospace model as a frequency management tool". En actas de International Symposium on Advanced Radio Technologies, Boulder, Colorado, USA, 2003, pp. 126–132.
4. GONZALEZ, C. R.; DIETRICH, C. B.; REED, J. H. "Understanding the software communications architecture," *IEEE Communications Magazine*, 2009, vol. 47, no. 9, pp. 50–57.
5. YUAN, Y.; BAHL, P.; CHANDRA, R.; CHOU, P. A.; FERRELL, J. I.; MOSCIBRODA, T.; NARLANKA, S.; WU, Y. "KNOWS: Cognitive radio networks over white spaces". En actas de Proc. IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland, 2007, pp. 416–427.

6. GHASEMI, A.; SOUSA, E. "Optimization of spectrum sensing for opportunistic spectrum access in cognitive radio networks". En actas de Proc. IEEE Consumer Communication and Networking Conference, Las Vegas, Nevada, USA, 2007, pp. 1022–1026.
7. MISHRA, S. T. B. S. M.; MAHADEVAPPA, R.; BRODERSEN, R. W. "Cognitive technology for ultra-wideband/WiMax coexistence". En actas de Proc. IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland, 2007, pp. 179–186.
8. SAFATLY, L.; AZIZ, B.; NAFKHA, A.; LOUET, Y.; NASSER, Y.; EL-HAJJ, A.; KABALAN, K. Y. "Blind spectrum sensing using symmetry property of cyclic autocorrelation function: from theory to practice". *EURASIP Journal on Wireless Communications and Networking*, 2014, vol. 2014, núm. 1, p. 26.
9. ARKOULIS, S.; ANIFANTIS, E.; KARYOTIS, V.; PAPAVALASSIOU, S.; MITROU, N. "Discovering and exploiting spectrum power correlations in cognitive radio networks: an experimentally driven approach". *EURASIP Journal on Wireless Communications and Networking*, 2014, vol. 2014, núm. 1, p. 17.
10. LI, S.; BI, G. "Time domain averaging and correlation-based improved spectrum sensing method for cognitive radio". *EURASIP Journal on Advances in Signal Processing*, 2014, vol. 2014, núm. 1, p. 46.
11. OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-time signal processing*. 3 Ed. Prentice Hall, 2010. 1136 pp. ISBN 0131988425.
12. YUCEK T.; ARSLAN, H. "Spectrum characterization for opportunistic cognitive radio systems". En actas de Proc. IEEE Military Communication Conference, Washington, D.C., USA, 2006, pp. 1–6.
13. CARLSON, A. B.; CRILLY, P. B.; RUTLEDGE, J. C. *Communication Systems: An introduction to Signals and Noise in Electrical Communication*, 4 Ed. McGraw-Hill, 2002.
14. FUQING, X. *Digital Modulation Techniques*, 2 Ed. Artech House. London 2006.
15. RASPBERRY PI FOUNDATION. *Computer Module Development Kit* [en línea]. [ref de 22 junio 2014] Disponible en Web: <http://www.raspberrypi.org/products/compute-module-development-kit/>.
16. ASTRO RADIO S.L. *Receptor SDR Perseus* [en línea]. [ref de 13 junio 2014]. Disponible en Web: <http://www.astro-radio.com/219001.html>.